SANGFOR

Sangfor HCI

Technical White Paper

Product Version	6.11.1
Document Version	01
Released on	Dec. 02, 2024



Copyright © Sangfor Technologies 2024. All rights reserved.

Unless otherwise stated or authorized, Sangfor Technologies Inc. (hereinafter referred to as "Sangfor") and its affiliates reserve all intellectual property rights, including but not limited to copyrights, trademarks, patents, trade secrets, and related rights to text, images, pictures, photographs, audio, videos, charts, colors, and layouts as presented in or concerning this document and content therein. Without the prior written consent of Sangfor, this document and content therein must not be reproduced, forwarded, adapted, modified, displayed, or distributed by any other means for any purpose.

Disclaimer

Products, services, or features described in this document, whether wholly or in part, may be not within your purchase scope or usage scope. The products, services, or features you purchase must be subject to the commercial contract and terms as agreed by you and Sangfor. Unless otherwise provided in the contract, Sangfor disclaims warranties of any kind, either express or implied, for the content of this document.

Due to product version upgrades or other reasons, the content of this document will be updated from time to time. Unless otherwise agreed, this document is used for reference only, and all statements, information, and recommendations therein do not constitute any express or implied warranties. Sangfor shall not be held liable for any loss or damage arising from omissions, changes, or errors in this document.

Contact Us

Pre-sales consulting: +86 (755) 8662-7888

After-sales service: +60 12 711 7129 (7511)

Official website: www.sangfor.com

About This Document

This document describes the architecture, features, and application scenarios of Sangfor Hyper-Converged Infrastructure (HCI) to help sales, pre-sales, and technical support representatives become familiar with HCI and its application scenarios.

Intended Audience

This document is intended for:

- Product Manager
- Pre-Sale Manager
- Partner Manager

Note Icons

English Icon	Description
	Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
	Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury.
	Indicates a hazardous situation, which if not avoided, could result in settings failing to take effect, equipment damage, or data loss. NOTICE addresses practices not related to personal injury.
	Calls attention to important information, best practices, and tips. NOTE addresses information not related to personal injury or equipment damage.

Contents

Contact Us 1
1 Product Overview 16
2 Customer Value 18
3 Development Path of HCI 20
3.1 Research Initiation in 2014 20
3.2 Architecture Reconstruction in 2018 21
3.3 Continuous Development 22
4 Cloud Strategy-Orientated HCI 24
4.1 HCI Architecture 24
4.1.1 Functional Architecture 24
4.1.2 Technical Architecture 26
4.1.2.1 Component Decoupling and Servitization 26
4.1.2.2 Service Containerization Deployment 29
4.1.2.2.1 Service Containerization 29
4.1.2.2.2 Container Lifecycle Management 30
4.1.2.2.3 Container Image Management 32
4.1.2.2.4 Image Repository Optimization 36
4.1.2.3 Homogeneous Dual-Stack Architecture 36
4.1.2.4 Global HA Architecture 40
4.1.3 Deployment Architecture 42
4.1.3.1 Plane Division 42
4.1.3.2 Deployment Modes 42
4.1.4 Network Architecture 44
4.2 HCI Advantages 46
5 Key Technologies 49
5.1 aSV 49
5.1.1 Overview 49
5.1.1.1 Introduction to Hypervisor Architecture 49
5.1.1.2 Hypervisor Architecture Implementation 51
5.1.1.2.1 vCPU Mechanism 54
5.1.1.2.2 Memory Virtualization Mechanism 55
5.1.1.2.3 I/O Device Virtualization 58
5.1.1.3 SFFS and SFFS Reliability 60
5.1.1.3.1 SFFS 60
5.1.1.3.2 SFFS Reliability 62

5.1.1.3.3 UI Display 65
5.1.1.4 QCOW2 Image File 65
5.1.1.5 Kernel Hardening 67
5.1.1.6 Isolation between VM and Hypervisor 67
5.1.1.7 Resource Isolation Between VMs 68
5.1.1.8 Service Process Memory Isolation 70
5.1.1.9 PCIe Cabling for VMs 71
5.1.2 Functional Architecture 73
5.1.3 Resource Management 73
5.1.3.1 Resource Overcommitment 73
5.1.3.2 System Resource Self-Guarantee 75
5.1.3.3 Guarantee of Resource Reservation for HA 77
5.1.3.4 Memory Ballooning 77
5.1.3.5 KSM 80
5.1.3.6 Memory Hot Add 81
5.1.3.6.1 Technical Principles 81
5.1.3.6.2 UI Display 84
5.1.3.7 DRS 85
5.1.3.7.1 Technical Principles 85
5.1.3.7.2 Forecasting Algorithm and Scheduling Recommendations 87
5.1.3.7.3 Node and VM Scores 90
5.1.3.7.4 UI Display 95
5.1.3.8 DRX 97
5.1.3.9 VM Live Migration 98
5.1.3.10 Void Detection 101
5.1.3.11 GPU Virtualization 104
5.1.3.11.1 GPU Pass-Through 105
5.1.3.11.2 GPU Partitioning (NVIDIA vGPU) 107
5.1.3.11.3 GPU Scheduling 108
5.1.3.12 External Storage Shared Disk 109
5.1.3.12.1 Technical Principles 109
5.1.3.12.2 UI Display 110
5.1.3.13 64 Disks per VM 110
5.1.4 Data Protection 111
5.1.4.1 VM Snapshots 111
5.1.4.2 Quick Backup 114
5.1.4.3 Quick Restoration 117

5.1.4.4 Periodic Full Backup, Compression and Archiving, and Backup Copy 119
5.1.4.5 Backup Health Check 121
5.1.4.5.1 Technical Principles 121
5.1.4.5.2 Resource Allocation 124
5.1.4.5.3 UI Display 125
5.1.4.6 Agentless Backup 127
5.1.4.6.1 Terminologies 127
5.1.4.6.2 Workflow 128
5.1.4.6.3 Advanced Features 132
5.1.4.6.4 UI Display 136
5.1.4.7 CDP 139
5.1.4.8 One-Click Data Clearing 144
5.1.4.9 Silent Operation 145
5.1.5 Fault Prevention and Handling 146
5.1.5.1 VM Restart Upon an Error 147
5.1.5.2 HA Mechanism for VMs 148
5.1.5.2.1 HA Capabilities 149
5.1.5.2.2 Advanced Features 161
5.1.5.2.3 UI Display 164
5.1.5.3 Handling of Subhealthy Hosts 167
5.1.5.4 Monitoring and Handling of External Storage I/O Latency 170
5.1.5.4.1 Technical Principles 171
5.1.5.4.2 UI Display 173
5.1.5.5 Memory Failure Isolation and Recovery 174
5.1.5.6 RAID Card Failure and Recovery 176
5.1.5.7 System Disk Software RAID Group 177
5.1.5.7.1 Technical Principles 178
5.1.5.7.2 UI Display 180
5.1.5.8 Secure Replacement of System Disks 180
5.1.5.9 Multi-USB Mapping Between Hosts 183
5.1.5.10 UPS Integration 184
5.1.5.11 Recycle Bin 185
5.1.5.12 Process Watchdog 186
5.1.5.13 Black Box Technology 186
5.1.5.14 Host Maintenance Mode 186
5.1.5.15 System File Backup and Restore 192

5.1.5.16 Memory Snapshot 192
5.1.5.16.1 Technical Principles 192
5.1.5.16.2 UI Display 192
5.1.6 Performance and Reliability Optimization 193
5.1.6.1 VirtIO Optimization 193
5.1.6.2 NUMA Optimization for Enhanced Computing Performance 195
5.1.6.3 VM-Level QoS 198
5.1.6.4 CPU, Network and Disk QoS 199
5.1.6.5 Kunpeng BoostKit 200
5.1.6.6 Large-Core VM Scheduling 201
5.1.6.7 CPU Scheduling Load Balancing 202
5.1.6.8 CPU Scheduling and Reservation 204
5.1.6.8.1 Application Scenarios 204
5.1.6.8.2 Technical Principles 205
5.1.6.8.3 UI Display 208
5.1.6.9 Optimization of Computing Conversion Rate 209
5.1.6.10 PAT Optimization for Kylin OS 213
5.1.6.11 Enhanced File System Reliability for Kylin OS 214
5.1.6.12 Enhanced Kernel Module for Kylin OS 215
5.1.6.13 Peripheral Driver Adaptation and Enhancement for Kylin OS 217
5.1.6.14 Advanced CPU Instruction Set Support 218
5.1.6.14.1 Technical Principles 218
5.1.6.14.2 UI Display 220
5.1.7 Business Deployment 220
5.1.7.1 VM Import and Export 220
5.1.7.2 P2V and V2V Deployment 221
5.1.7.2.1 SCMT Architecture 222
5.1.7.2.2 Point-to-Point Migration (Without Proxy) 225
5.1.7.2.3 Point-to-Point Migration (with Proxy) 226
5.1.7.2.4 HA Migration 227
5.1.7.2.5 Backup Migration 229
5.1.7.3 Template-Based Deployment 230
5.1.7.4 Specifying IP Addresses or Hostnames 232
5.1.7.5 VM Scheduling Policy 234
5.1.7.5.1 Introduction to Groups 234
5.1.7.5.2 Application Scenarios of Scheduling Groups 236
5.1.7.5.3 Application Scenarios of Application Groups 237

5.1.7.5.4 Application Scenarios of Host Groups....238
5.1.7.5.5 Application Scenarios of Security Groups....239
5.1.7.5.6 Scheduling Policy Management....239
5.1.7.6 Space Release upon Data Deletion....240
5.1.7.6.1 Internal Storage Space Release....240
5.1.7.6.2 External Storage Space Release....242
5.1.7.7 VM Application Deployment Wizard....244
5.1.7.8 Full Clone....244
5.2 aSAN....245
5.2.1 Overview....245
5.2.1.1 IOPS....245
5.2.1.2 Throughput....246
5.2.1.3 Cache Disk and Data Disk....246
5.2.1.4 Disk Group....247
5.2.2 Functional Architecture....248
5.2.2.1 Logical Architecture of aSAN....249
5.2.2.2 Basic aSAN Process....251
5.2.2.3 Read I/O Process of VMs....253
5.2.3 High Performance....255
5.2.3.1 Data Sharding....255
5.2.3.2 Data Striping....256
5.2.3.2.1 Adaptive Stripe Width....257
5.2.3.2.2 Stripe Size....258
5.2.3.3 Tier....259
5.2.3.3.1 Caching System Architecture....260
5.2.3.3.2 Key Technologies....261
5.2.3.3.3 Technical Principles....266
5.2.3.4 Optimal Data Distribution....280
5.2.3.4.1 Data Localization....280
5.2.3.4.2 Utilization of More SSDs....281
5.2.3.4.3 Utilization of More HDDs....282
5.2.3.4.4 Balanced Distribution....282
5.2.3.5 I/O Path Optimization....283
5.2.3.5.1 Scheduling of VM vCPU and I/O Threads....284
5.2.3.5.2 Soft Affinity Scheduling....284
5.2.3.5.3 Read Cache of libnfs....285
5.2.3.5.4 Smart I/O Pipelining....287

5.2.3.5.5 High-Latency Submodule Optimization 290
5.2.3.5.6 Shared Library Overhead Optimization 290
5.2.3.5.7 Thread Model Optimization 290
5.2.3.5.8 IOPS Improvement 290
5.2.3.6 Zero-Copy Memory 291
5.2.3.7 Disk-Level QoS 292
5.2.3.8 RDMA 293
5.2.3.8.1 Technical Background 293
5.2.3.8.2 Technical Principles 297
5.2.3.8.3 UI Display 300
5.2.3.9 SPDK-Based Turbo Service 303
5.2.3.9.1 SPDK Overview 303
5.2.3.9.2 Terminologies 304
5.2.3.9.3 Technical Principles 304
5.2.3.9.4 Differences in I/O Processing Between SPDK VMs and Ordinary VMs 306
5.2.3.9.5 UI Display 310
5.2.4 High Reliability 311
5.2.4.1 Storage-Based Snapshot 311
5.2.4.1.1 Scenarios 311
5.2.4.1.2 Technical Principles 312
5.2.4.1.3 Consistency Group Snapshot 315
5.2.4.1.4 Scheduled Snapshot Policy 316
5.2.4.2 Clone 317
5.2.4.2.1 Full Clone 317
5.2.4.2.2 Linked Clone 318
5.2.4.2.3 Instant Full Clone 319
5.2.4.2.4 Comparison Among Three Clone Types 320
5.2.4.3 Multi-Replica Data 321
5.2.4.3.1 Strong Data Consistency Among Replicas 321
5.2.4.3.2 Replica Distribution Principles 322
5.2.4.3.3 Witness Replica Distribution Principles 323
5.2.4.3.4 Data Replica Distribution Principles for Scenarios Where Witness Replicas Exist 323
5.2.4.3.5 Fault Tolerance 326
5.2.4.4 Quorum Mechanism 334
5.2.4.4.1 How Does Split-Brain Occur? 334

5.2.4.4.2 How Does a Witness Replica Work?.....337
5.2.4.5 Silent Error Detection.....341
5.2.4.6 Data Rebuilding.....343
5.2.4.6.1 Data Rebuilding Scenarios.....344
5.2.4.6.2 Characteristics of Data Rebuilding.....350
5.2.4.7 Sub-Healthy Disk Detection and Handling.....353
5.2.4.7.1 Proactive Detection of Bad Sectors.....354
5.2.4.7.2 Proactive Bad Sector Repair.....356
5.2.4.7.3 Proactive Data Migration.....357
5.2.4.7.4 Soft Isolation Upon Failure.....357
5.2.4.8 High Reliability of Storage Links.....360
5.2.4.9 Spare Disks for Data Protection.....361
5.2.4.10 Enhanced Reliability of General I/O Processing Module for Kylin OS.....362
5.2.5 Smart O&M.....363
5.2.5.1 Bad Sector Prediction.....363
5.2.5.2 Capacity Prediction.....365
5.2.5.3 SSD Lifetime Prediction.....366
5.2.5.4 Data Self-Balancing.....367
5.2.5.4.1 Data Imbalance.....367
5.2.5.4.2 Objectives of Data Balancing.....368
5.2.5.4.3 Triggers for Data Balancing.....368
5.2.5.5 SkyOPS.....371
5.2.6 Advanced Storage Features.....373
5.2.6.1 Storage Partitioning.....373
5.2.6.1.1 Host-Level Partitioning.....373
5.2.6.1.2 Disk-Level Partitioning.....374
5.2.6.2 Storage Policies.....376
5.2.6.3 Virtual iSCSI Storage.....377
5.2.6.4 Shared Virtual Disks.....381
5.2.6.5 Virtual Disk Management.....382
5.2.7 Stretched Cluster.....385
5.2.7.1 Overview.....385
5.2.7.2 Technical Principles.....386
5.2.7.3 Two-Replica Stretched Cluster.....387
5.2.7.4 Three-Replica Stretched Cluster.....388
5.3 aNET.....391

5.3.1 Overview....391
5.3.1.1 SDN....391
5.3.1.2 Overlay Technologies....392
5.3.1.3 VXLAN Technology....396
5.3.1.4 aNET....398
5.3.1.5 Multi-Cluster Layer 2 Network....398
5.3.2 Functional Architecture....400
5.3.2.1 Logical Architecture of aNET....400
5.3.2.2 Functional Architecture of aNET....410
5.3.2.2.1 Overview....410
5.3.2.2.2 Deployment of Functional Modules....412
5.3.2.3 High Availability Design of aNET....415
5.3.2.3.1 High Availability of CMP....415
5.3.2.3.2 High Availability of CCP....416
5.3.2.3.3 High Availability of MB....418
5.3.2.4 Reliable Network Planes of aNET....418
5.3.2.5 High-Performance Architecture of aNET....420
5.3.3 Network Deployment....421
5.3.3.1 Network Interface Aggregation....421
5.3.3.2 Physical Networks....423
5.3.3.3 Physical Network Interface Reuse....425
5.3.3.3.1 Three-in-One Network Interface Reuse....426
5.3.3.3.2 Four-in-One Network Interface Reuse....427
5.3.3.4 Cluster Deployment Modes....429
5.3.3.4.1 Overview....429
5.3.3.4.2 Network Communication Domain....430
5.3.3.4.3 Single-Cluster Deployment....432
5.3.3.4.4 Multi-Cluster Deployment....432
5.3.3.5 Virtual Networks....436
5.3.3.6 IPv6 Support....438
5.3.3.7 Low Network Latency (A feature in HCI 6.9.1)....439
5.3.3.7.1 Overview....439
5.3.3.7.2 Key Technologies for Low-Latency Networks....441
5.3.4 High-Performance Network....444
5.3.4.1 DPDK-Based High-Performance Network....444
5.3.4.1.1 Disadvantages of OS Kernel-Based Data Transmission....445
5.3.4.1.2 Solutions....446

5.3.4.1.3 Implementation of Kernel Bypass 446
5.3.4.1.4 Key DPDK Technologies 447
5.3.4.2 Multiple Queues and Interrupt Binding For NICs 448
5.3.4.3 NUMA Node Affinity 450
5.3.5 Virtual Network Elements 451
5.3.5.1 Distributed Virtual Switches 451
5.3.5.2 Edge Virtual Routers 452
5.3.5.3 Edge Virtual Switches 454
5.3.6 Network Traffic Forwarding Models 457
5.3.6.1 BUM Traffic Forwarding 457
5.3.6.1.1 Unicast Replication Mode 457
5.3.6.1.2 ARP/ND Proxy 460
5.3.6.2 Unicast Traffic Forwarding 461
5.3.6.2.1 Single-Data Center, Single-Cluster Unicast Traffic Forwarding 461
5.3.6.2.2 Single-Data Center, Multi-Cluster Unicast Traffic Forwarding 474
5.3.6.2.3 Cross-Site, Multi-Cluster Unicast Traffic Forwarding 488
5.3.7 Network Visualization 491
5.3.8 East-West Distributed Firewall 493
5.3.8.1 Features 494
5.3.8.2 Application Scenarios 495
5.3.8.3 Applicable Scope 495
5.3.8.4 Policies and Rules 495
5.3.8.5 Dropped Packet Logging 496
5.3.8.6 Rule Anomaly Alert 496
5.3.9 Intelligent DFW Policy Recommendation Based on aNI 497
5.3.9.1 Architecture 497
5.3.9.2 Technical Principles 497
5.3.9.3 UI Display 498
5.3.10 Traffic Mirroring 500
5.3.11 Network O&M 503
5.3.11.1 What-You-Draw-Is-What-You-Get 503
5.3.11.2 LLDP 504
5.3.11.3 Fault Prevention and Handling 506
5.3.11.3.1 Reliability of Layer 3 Forwarding Network 506
5.3.11.3.2 Network Connectivity Test 507

5.3.11.3.3 VXLAN Connectivity Test 507
5.3.11.3.4 Network Monitoring and Alert 508
5.3.11.4 VM NIC Rate Limiting 510
5.3.11.5 Network Health Monitoring 512
5.4 aSecurity 516
5.4.1 Overview 516
5.4.2 BSI 517
5.4.3 aSecurity 519
5.4.4 VM Security 520
5.4.4.1 Overview 520
5.4.4.2 Architecture 520
5.4.4.3 Security Upon Operation 521
5.4.4.4 Asset Identification 524
5.4.4.5 Vulnerability Management 524
5.4.4.6 Ransomware Recovery 525
5.4.4.7 One-Click Threat Fixing 527
5.4.4.8 Data Protection (Correlated With Snapshots) 529
5.4.5 Cloud Firewall 530
5.4.5.1 Overview 530
5.4.5.2 Technical Architecture 531
5.4.5.3 Unified Asset Management 532
5.4.5.4 Adaptive Security Policies 533
5.4.5.5 Improved Data Plane Performance 534
5.4.5.6 HA 536
5.4.5.7 Intelligent Security Engine 536
5.4.6 vNGAF 536
5.4.6.1 IPS 537
5.4.6.2 WAF 538
5.4.7 aTrust 540
5.4.7.1 Overview 540
5.4.7.2 Architecture 541
5.4.7.3 Key Functional Modules 543
5.4.8 vAC 543
5.4.9 vSSL VPN 546
5.4.10 vAD 547
5.4.11 Data Security 547
5.4.11.1 Data Encryption 548

5.4.11.2 Commercial Cryptography 549
5.4.11.2.1 Overview 550
5.4.11.2.2 Key Technologies 551
5.4.11.2.3 Confidentiality and Integrity of Data Transmission 554
5.4.11.2.4 Commercial Cryptography Application Topology 555
5.4.12 Security O&M and Operation 556
5.4.12.1 O&M Audit (Bastion Host) (vOSM) 556
5.4.12.2 Database Audit (vDAS) 556
5.4.12.3 Vulnerability Scanning and Baseline Check (Sangfor Host Security) 557
5.4.12.4 Log Audit (SIP-Logger) 557
5.4.12.5 Security Integration (aSI) 558
5.4.12.5.1 Cyber Command Overview 558
5.4.12.5.2 Integration of Cyber Command and HCI 559
5.4.12.5.3 Scenarios 561
5.5 HCI Management 562
5.5.1 Overview 562
5.5.2 O&M Management 562
5.5.2.1 Platform Monitoring and Detection 563
5.5.2.1.1 Hardware Health Monitoring 563
5.5.2.1.2 HCI Resource Monitoring 566
5.5.2.2 Health Check 568
5.5.2.3 Alert Service 569
5.5.2.3.1 Email Alerts 570
5.5.2.3.2 SMS Alerts 570
5.5.2.4 Report Center 575
5.5.2.5 Command Line Instructions 576
5.5.3 Cluster Management 577
5.5.3.1 Cluster Communication Mechanism 577
5.5.3.2 Cluster HA 577
5.5.3.3 Dual-Host Quorum 578
5.5.3.4 Online Expansion 578
5.5.3.4.1 Host Expansion 579
5.5.3.4.2 Storage Expansion 580
5.5.4 Cluster Upgrade 581
5.5.4.1 In-Place Active Upgrade 582
5.5.4.2 Rolling Active Upgrade 583

5.5.5 Management Platform Security 584
5.5.5.1 SSH Secure Login to Host 585
5.5.5.2 HTTPS Secure Login 585
5.5.5.3 Anti-Brute Force Secure Login 585
5.5.5.4 Two-Factor Authentication for Secure Login 587
5.5.5.5 Restricted Secure Login by IP Address and MAC Address 589
5.5.5.6 Security Encryption Technology 592
5.5.5.7 User Permissions Management 592
5.5.5.8 Permission Separation 593
5.6 Typical HCI Business Processes 595
5.6.1 VM Network Traffic Forwarding 595
5.6.2 VM I/O Reading Process 596
5.7 Open Support 597
5.7.1 HCI APIs 597
5.7.2 CLI Support 598
5.8 Configuration Information and Product Compatibility 598
5.8.1 HCI Underlying Configuration Information 599
5.8.2 Server Compatibility 599
5.8.3 CPU Compatibility 599
5.8.4 GPU Compatibility 599
5.8.5 Host OS Compatibility 600
6 HCI 601
6.1 Overview 601
6.2 Private Cloud Solution 601
6.2.1 Business Scenarios and Challenges 601
6.2.2 Solution Architecture and Benefits 602
6.2.3 Customer Value 604
6.3 Disaster Recovery Solution 605
6.3.1 Business Scenarios and Challenges 605
6.3.2 Solution Architecture and Benefits 606
6.3.3 Customer Value 607
6.4 Database Management Solution 608
6.4.1 Business Scenarios and Challenges 608
6.4.2 Solution Architecture and Benefits 609
6.4.3 Customer Value 610
6.5 Server Virtualization Solution 611
6.5.1 Business Scenarios and Challenges 611

6.5.2 Solution Architecture 612
6.5.3 Customer Value 614
6.6 GPU Scenario Solution 615
6.6.1 Business Scenarios and Challenges 615
6.6.2 Solution Architecture and Benefits 616
6.6.3 Customer Value 617
6.7 VDI Solution 618
6.7.1 Business Scenarios and Challenges 618
6.7.2 Solution Architecture and Benefits 618
6.7.3 Customer Value 620
7 Cloud Computing Concepts and Terminology 621
7.1 General Concepts 621
7.2 Cloud Computing Concepts 625
7.3 Database Concepts 629
7.4 Cloud Native/Cloud Security/O&M Concepts 632
7.5 Emerging Technology Concepts 635
7.6 Sangfor Cloud Terms 638

1 Product Overview

Modern enterprises have significantly enhanced their competitiveness due to the rapid development and extensive use of IT technologies. However, these widely used technologies are accompanied by a substantial increase in both the capital expenditures (CAPEX) and operational expenses (OPEX) for the enterprises. What's worse, the prolonged business deployment cycles and isolated IT infrastructures have made the IT technologies, which should have contributed to enterprise development, a bottleneck for rapid business growth. The traditional IT framework has the following disadvantages:

1. The tight coupling of applications and resources results in an inflexible underlying architecture.
2. In the face of the rapid increase in the number of businesses and users due to business transformation, IT resources are unable to increase rapidly and flexibly to accommodate the growing needs of the business.
3. The traditional data center has a complex overall architecture and a variety of device types, resulting in slow business deployment. This makes the deployment of new businesses take several months from solution design and integrated deployment to release.
4. Difficulties in O&M and management caused by various device types from different brands in data centers make it hard for the IT department to monitor and manage the infrastructure in a unified manner. In addition, the resource usage of most devices is less than 20%. Except for some data centers that have applied compute virtualization, the storage, network, and security architectures still rely on physical device deployment, resulting in high costs for business development.
5. The IT department devotes a majority of resources and efforts to maintaining the underlying infrastructure, which leads to less concentration on innovations of combining informatization and business, thus losing the core value in the process of business transformation.

The emergence of cloud computing has disruptively changed the construction and service models of enterprise IT platforms. The online management, on-

demand access, and fast delivery features of cloud computing have gained widespread popularity among enterprises. As a result, more and more enterprises have shifted to cloud computing. According to requirements on the customer value, Sangfor provides the Hyper-Converged Infrastructure (HCI) solution as the underlying technology base for cloud computing. By leveraging its full-stack HCI software capabilities, Sangfor has built a software-defined cloud data center infrastructure that uses general commercial servers and Layer 2 and Layer 3 switches. Besides, Sangfor Cloud Platform (SCP) implements seamless integration of the cloud infrastructure and cloud applications. The synergy between HCI and SCP effectively facilitates cloud data centers to implement easy delivery and efficient O&M of IT infrastructure, which is the optimal path for cloudification.

HCI consists of aSV, aSAN, aNET, and aSecurity. SCP enables the provision of HCI virtualization resource pools as services and supports the unified management of multiple HCI clusters. SCP integrates the reliability center, monitor center, aSecurity, operations platform (aOPS), and application center, supports the function expansion of container, aDesk, big data, middleware, database, and hybrid cloud, and manages third-party physical and cloud resources. Sangfor HCI and SCP can satisfy the requirements for cloud data center construction and management of different customers. This document focuses on HCI virtualization software, hyper-converged cluster management software, and corresponding hyper-converged solutions. For details about SCP, view *Sangfor Cloud Platform Technical White Paper*.

2 Customer Value

1. Stability and Reliability

Ensuring the stability and reliability of HCI becomes more important as HCI carries more and more critical businesses. For this purpose, the overall stability and reliability of HCI is improved from multiple dimensions. The distributed architecture on which SCP and HCI are built ensures the stability of SCP in the event of a single node failure. The improved monitoring and shielding mechanism of SCP protects businesses from inevitable hardware silent errors. The disaster recovery capabilities provided by SCP enhance the reliability and stability of HCI.

2. High Performance

Higher performance is required for HCI to carry more critical businesses. HCI features high performance across multiple dimensions, including computing, storage, network, and management. For example, the storage engine for distributed storage developed by Sangfor has implemented numerous technological advancements, such as I/O striping, I/O localization, I/O unlocked pipeline, and intelligent cross-NUMA scheduling through eight years of performance optimization. With the three general servers and NVMe SSDs, this engine achieves a storage capacity of millions of IOPS, surpassing the distributed storage capacity in the industry by over 30% in database scenarios and rivaling high-end storage engines.

3. Security and Effectiveness

Security has become the fourth major infrastructure, following computing, network, and storage. Sangfor is the first in the industry to possess built-in security capabilities for cloud platforms to implement security upon business deployment and avoid complicated security management or security risks resulting from errors. Moreover, built-in security capabilities provide users with a god's-eye view to check the traffic flow directions and the security relationships between businesses, enhancing overall security effectiveness.

4. Intelligence and Convenience

The intelligence and convenience capability of HCI can optimize a single

function and ensure a convenient experience throughout the entire platform management process for users. HCI provides full lifecycle process optimization capabilities, including business migration, hyper-convergence, business launch configuration, visualized monitoring, troubleshooting, and smooth capacity expansion for users to save efforts and improve management efficiency. In addition, HCI provides SkyOPS to assist users in swiftly identifying and responding to risks to reduce their impact on business operations.

3 Development Path of HCI

3.1 Research Initiation in 2014

The fundamental concept of the HCI solution is to develop a software-defined distributed storage software that integrates commercial servers into a distributed storage resource pool.

In 2014, Sangfor initiated the R&D of distributed storage software products and drew inspiration from the design concept of GlusterFS, a popular scalable and open-source distributed file system with strong storage horizontal expansion capability. GlusterFS enables users to access application data by using standard protocols, such as GlusterFS, NFS, and CIFS, in a globally unified namespace. Compared with traditional closed and centralized storage systems that are independent and with high costs, GlusterFS uses cost-effective storage servers to deploy a centrally managed, horizontally scalable, and virtualized distributed storage resource pool with a petabyte-level storage capacity.

GlusterFS adopts a client/server (C/S) architecture where the client end uses the consistent hashing algorithm to calculate the data brick (local file system storage directory) and communicates with the corresponding server end to retrieve metadata and data. This architecture does not require a master node to manage metadata in a unified manner, which eliminates the risks of single-node failure. However, this no-master node architecture has shortcomings, such as difficulties in processing data in a unified manner and low operation efficiency, particularly when processing a large number of small-size files. In addition, file read/query operations are not frequent on the I/O path for block storage. As a result, eliminating bottlenecks concerning metadata brings only marginal benefits.

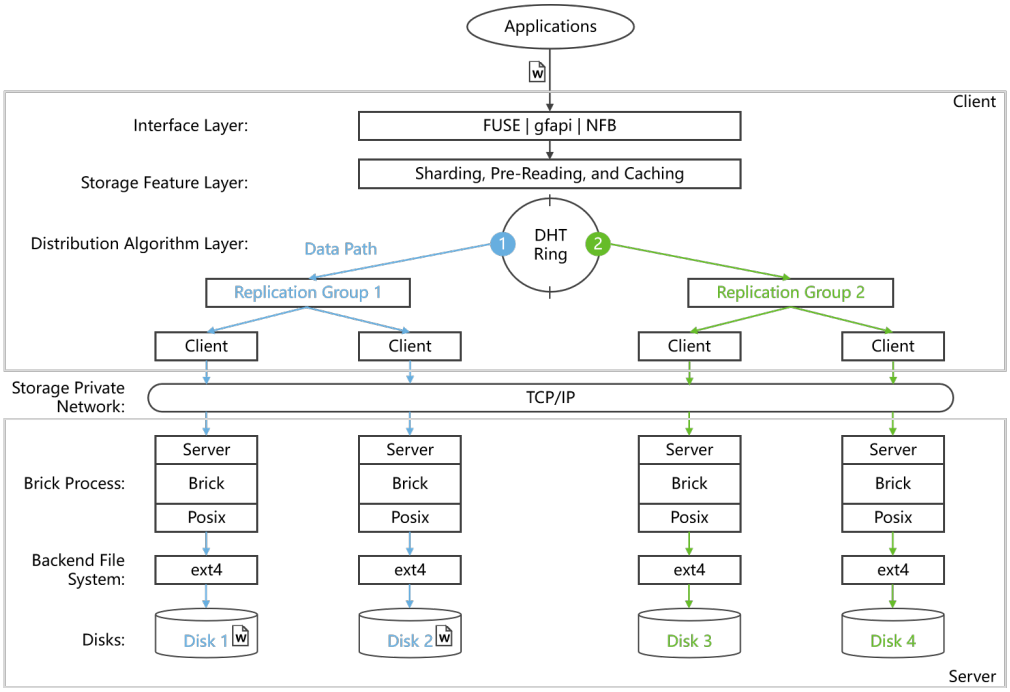


Figure 1: GlusterFS I/O Path

From 2017 to 2018, Sangfor reconstructed the architecture in an innovative manner in response to business requirement changes. By introducing the self-developed Easy File System (EFS) and central management of metadata, Sangfor addressed the deficiencies in open-source technologies and greatly enhanced product performance, particularly in terms of scalability, reliability, security, and advanced storage features. As a result, Sangfor significantly improved its product competitiveness and gained widespread market recognition.

3.2 Architecture Reconstruction in 2018

In 2018, Sangfor released aSAN 3.0.2 by adopting a brand-new, self-developed distributed storage architecture. aSAN 3.0.2 utilized the EFS local file system and supported more flexible file sharding and striping, which implemented higher performance of the storage system in data distribution, data balancing, and data rebuilding.

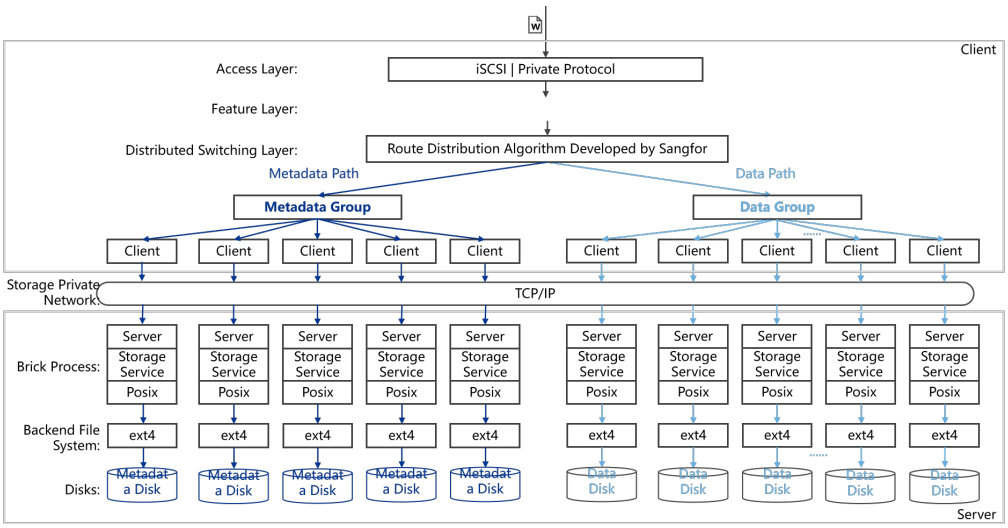


Figure 2: New Architecture

In comparison with elastic hashing algorithm-based mainstream file systems such as GlusterFS, Sangfor aSAN has the following advantages:

Built based on a new architecture, aSAN incorporates a wide range of additional advanced storage features such as storage snapshot, linked clone, and disk subhealth management.

3.3 Continuous Development

Sangfor HCI has continuously conducted iterations to comply with diversified application scenarios for customers. In 2021, Sangfor launched HCI 6.7.0, which featured layered and decoupled software and enhanced control capabilities and openness. HCI 6.7.0 optimized global management capabilities to support larger resource pools and modified the centralized management and control function of the software-defined networking (SDN) controller to support ultra-large-scale node resource management. HCI 6.8.0 was brought online in 2022. It marked the first release of several new functions, including SkyOPS, Disaster Recovery as a Service (DRaaS), business hybrid deployment, and homogeneous hybrid cloud.

In 2023, Sangfor released the latest HCI 6.9.0 with the idea of empowering products with AI-First technology. HCI 6.9.0 features software-defined reliability. It ensures uninterrupted business operations for hot migration,

active upgrade, and hot patch across QEMU kernel versions and leverages the chaos engineering toolkit to enhance the resilience and self-healing capabilities of distributed systems. In addition, both SkyOPS and aSecurity have been fully upgraded to version 2.0. The latest SkyOPS, powered by the built-in AIOps engine, provides complete risk prediction, continuous optimization, and real-time monitoring capabilities. Compared with aSecurity 1.0, aSecurity 2.0 improves the capability of security upon business deployment. It added the virtual patching function (use customized access control policies against specific vulnerabilities) and integrated with the intrusion prevention system function of distributed firewalls to protect user assets from attacks. HCI 6.9.0 puts forward the concept of Dual-Stack Consistency, which ensures uncompromised functionality and quality and supports rapid adaptation of hardware, OSs, and userspace tools through OS tools.

The latest HCI, as the technology base for cloud infrastructure, significantly enhances system performance and stability without compromising any functionalities, thus offering greater support for the construction of various private cloud resources. Furthermore, through integration with SkyOPS, aSecurity, and Managed Cloud, HCI facilitates unified management of online and offline resources for O2O Synergized Cloud, consistent security policy configuration, and business hybrid deployment, and therefore advances cloud transformation for Sangfor.

4 Cloud Strategy-Orientated HCI

As the technology base for cloud computing resources, HCI comprehensively adopts cloud strategies and supports the construction of medium and large-scale cloud resource pools. Sangfor HCI features flexible scalability in deployment, supporting the deployment from two nodes to thousands of nodes. It also provides the global resource management service, as well as advanced services such as high availability (HA) and disaster recovery (DR). Managed Cloud, with HCI as the technology base, provides customers with self-service cloud services through an online cloud service model. In addition, Managed Cloud enables quick construction of cloud infrastructure and establishment of a high cloud security baseline and supports evolution towards a homogeneous hybrid cloud. Integrating the online and offline O2O Synergized Cloud service with various advanced cloud services, Sangfor is dedicated to empowering customers in their digital transformation.

4.1 HCI Architecture

4.1.1 Functional Architecture

Sangfor HCI is designed based on a layered architecture. It abstracts and defines the computing, storage, network, and security resources as objects rather than physical node resources for management in clusters. This helps get rid of direct binding with physical node resources. HCI is independent of the platform layer for the deployment of aSV, aSAN, aNET, and aSecurity, service update, configuration, and inter-service communication. In the HCI architecture, the upper and lower layers communicate with each other through standard interfaces, and standard APIs are provided for integration with third-party systems.

During the design phase, the architecture of a software-defined data center (SDDC) is divided into several layers for loose coupling. Although the services and architecture of each layer depend on the infrastructure of the layer below, the layers interact with each other through common or industry-standard

interfaces, which makes them interrelated but independent. This loosely coupled IT architecture ensures more flexible updates and iterations and avoids potential waste, such as discarding all devices and incurring redundant investments due to technological advancements.

With an aim to meet the diverse application requirements and maintain a stable and reliable service layer, selecting an appropriate infrastructure layer plays a crucial role in the construction of an SDDC.

HCI architecture can be divided into three layers. The description and figure of the layered architecture are as follows:

1. Physical layer: It consists of infrastructure hardware, including physical resources such as CPU, memory, HBA cards, GPU, NIC, RAID cards, SSDs, and HDDs. These physical resources can be managed and controlled by the HCI cluster management software.
2. Virtualization layer: As the main component of HCI, it comprises aSV, aSAN, aNET, and aSecurity and provides virtualized computing, storage, and network resources for the upper layer and cloud layer.
3. Management and control layer: It is responsible for cluster management in HCI and functions configuration management, resource management, monitoring service, and advanced functions. This layer also supports integration with the cloud platform through northbound APIs to implement the management of cloud data centers. A layered architecture avoids deploying management layers on every node and reduces resource consumption and message interactions, thereby enhancing management efficiency.

- **Configuration management**: Used to convert or deliver configurations for aSV, aSAN, aNET, and aSecurity, which includes topology calculation, error recovery, data storage, framework distribution, and read/write paths.
- **Resource management**: Used to manage resources for aSV, aSAN, aNET, and aSecurity, which includes nodes, storage capacity, network visualization, VMs, network topology, virtual storage, and physical machines.

- **Monitoring service**: Used to monitor the components from the following aspects: usage, read/write speed, alert notification, information collection, physical machine and storage status, VM and virtual storage status, host subhealth, one-click virtual environment status check, and performance and system bottlenecks.
- **Advanced functions**: Used to offer advanced functions such as backup and continuous data protection (CDP), snapshots, HA, distributed resource scheduler (DRS), dynamic resource extension (DRX), aSecurity, and network functions virtualization (NFV).

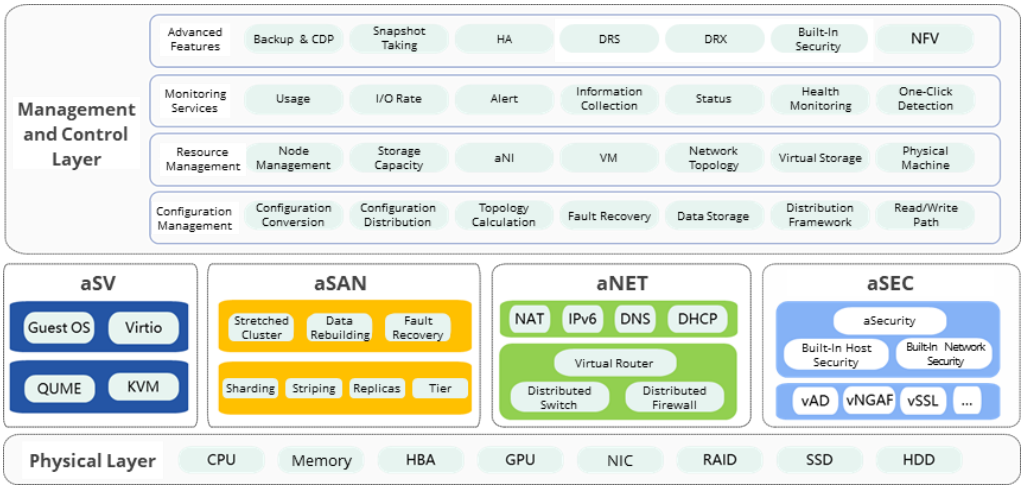


Figure 3: HCI Layered Functional Architecture

For details about aSV, aSAN, aNET, and aSecurity, refer to the following relevant chapters.

4.1.2 Technical Architecture

4.1.2.1 Component Decoupling and Servitization

HCI has encountered the following challenges during its development:

1. Data redundancy: Independent systems lead to scattered storage of similar or identical data.

2. Large system size: If all functions are tightly coupled within a single system, the system will be overloaded with codes and its maintenance, upgrade, and iteration data. As a result, the release will become arduous.
3. Code duplication: Codes with the same logic are applied to multiple systems. Worse still, highly coupled codes and data make it difficult for system extension.
4. Poor code quality: Codes with poor quality lead to interference among systems or modules, causing difficulties in maintaining the system.

In response to these issues, Sangfor has performed component decoupling and servitization for HCI. Component servitization is designed to ensure that HCI can provide standardized services and maintenance capabilities in independent scenarios such as aSV and aSAN. The main principles are illustrated in the following figure.

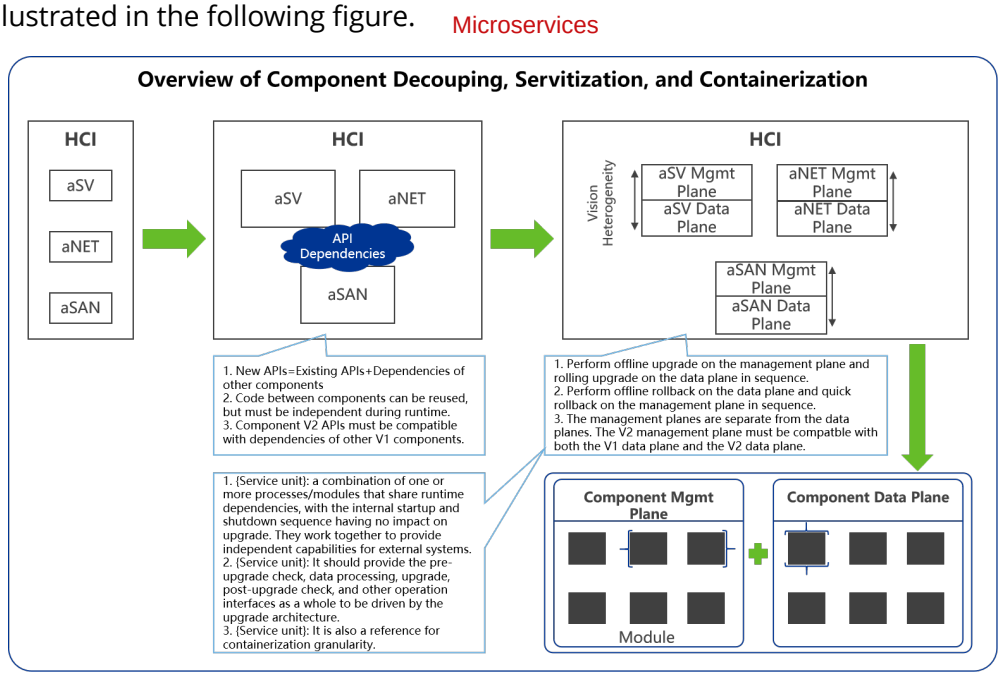


Figure 4: Component Decoupling and Containerization Overview

HCI utilizes standardized APIs to decouple aSV, aSAN and aNET and supports mutual calling between them. Also, it separates the management and data planes of the virtualization layer for version heterogeneity and decouples the business logic within the management and data planes into individual service units, each of which contains tightly coupled service logic. These service units

are then containerized and deployed.

The service decoupling process aims to decouple the dependencies between computing, storage, and network by using APIs to implement coupling, which helps standardize the interfaces and enables independent evolution, upgrades, and maintenance of the components without impacting others. This is particularly crucial in hot upgrade scenarios. Currently, the components of HCI are completely decoupled, which facilitates efficient extension and independent evolution. The following figure shows the process for service interface calling after service decoupling.

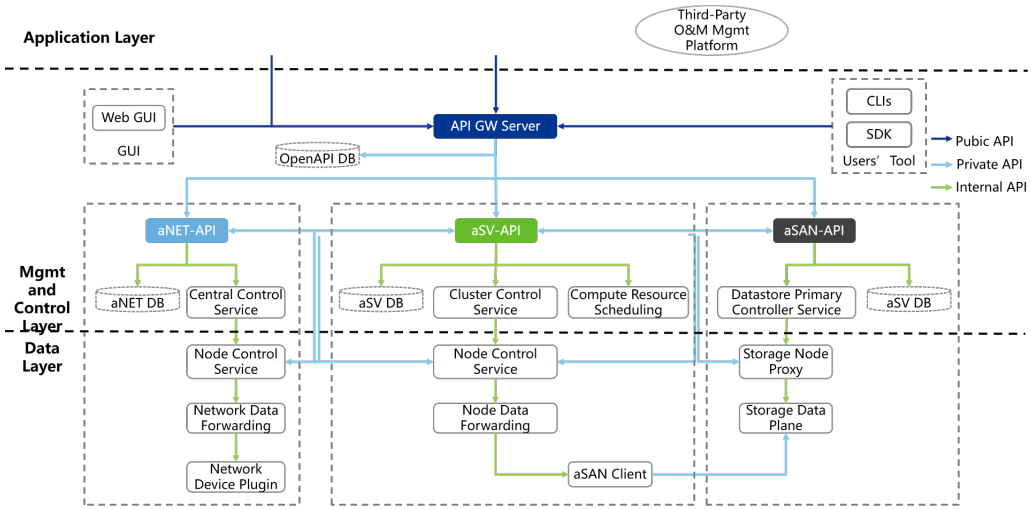


Figure 5: Decoupled HCI Services and Interface Calling

The service interfaces refer to a set of operations and corresponding parameters that are described using an interface description language. These interfaces are designed for layered architecture, with different levels of interface exposure. The specific details are as follows:

- **Public API**: Used to provide services to upper-layer applications such as GUI, CLI, VDC, and SCP. These services are provided through the built-in API gateway in HCI.
- **Private API:** Mainly used for mutual calling between components to implement various business combinations.
- **Internal API**: Specifically designed for internal component functionality.

All service interfaces are designed to be forward-compatible with earlier

versions, thus ensuring consistency of dependencies.

NOTICE

1. aSecurity and SCP are separately deployed and use the API gateway to call the aNET, aSV, and aSAN capabilities.

2. aNET, aSV, and aSAN are deployed on the host and communicate with each other through the private API.

4.1.2.2 Service Containerization Deployment

4.1.2.2.1 Service Containerization

Service containerization aims to package the components and modules of HCI into separate container images and have them deployed. Leveraging the function of one-time packaging for universal deployment of containerization, HCI ensures consistency in its development, testing, and delivery processes.

Furthermore, with the smaller granularity of service components and containerized deployment, scenarios such as HCI deployment and upgrade can be implemented based on container start, stop, and orchestration. As containers are running in an isolated manner, conflicts and dependencies that arise from sharing a common base image are eliminated, and the internal implementation at the service layer is hidden. Containers with different dependency libraries and codes can run independently without affecting others.

With the containerized deployment capability, HCI can leverage container lifecycle management and compression and management of local container images to manage its components. The service containerization architecture of HCI is illustrated in the following figure based on the granularity of containerization.

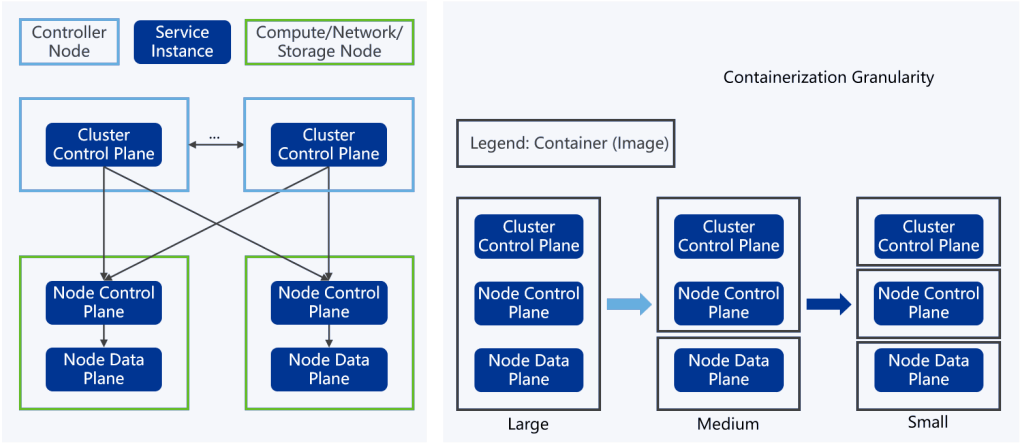


Figure 6: Service Containerization Architecture

The architecture is designed based on the characteristics of computing, storage, and network components, and the containerization granularity is demonstrated in the figure above. In a cluster architecture, computing, storage, and network resources are divided into three planes, namely, cluster control plane, node control plane, and node data plane. The resources are categorized primarily according to the different types of deployment nodes and their corresponding functions.

4.1.2.2.2 Container Lifecycle Management

Component decoupling and servitization make it easy to deploy the HCI components based on containerization. By utilizing technologies such as sf-kubelet, containerd, and runC, Sangfor has implemented lifecycle management for the containerized components. The following figure displays the container management components involved in container lifecycle management.

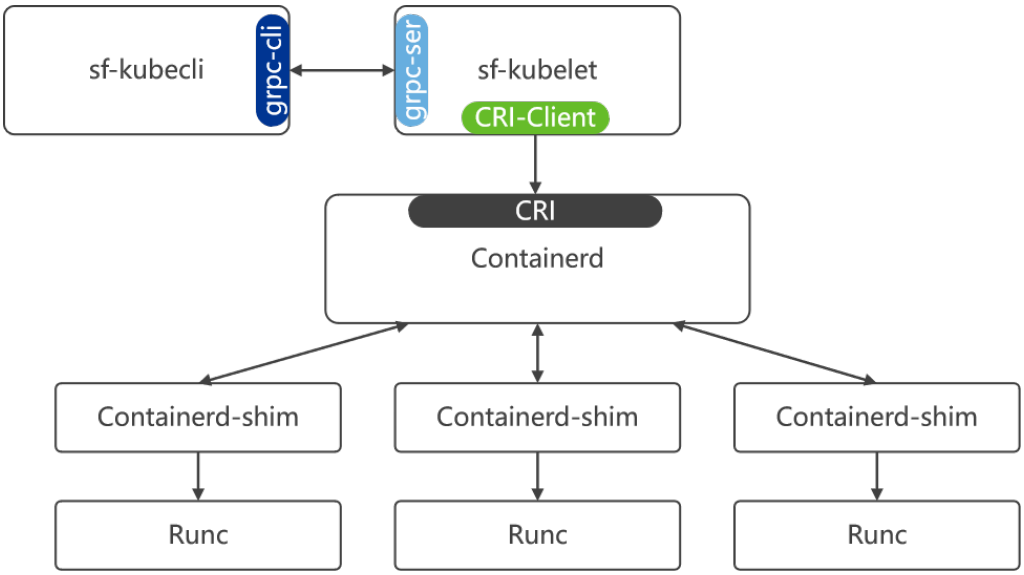


Figure 7: Container Management Components

Component description:

- **sf-kubecli**: A command line tool based on the client/server (C/S) model and interacts with sf-kubelet through gRPC. It is used to send commands, such as starting a container or stopping an operation, to sf-kubelet.
- **sf-kubelet**: A daemon process responsible for container status monitoring and container scheduling and orchestration. It interacts with containerd through the CRI client to deliver container operation commands and collect container information.
- **containerd**: It manages the complete lifecycle of a running industrial-grade container. For example, container image management, container start/stop operation control, storage management, and network management.
- **containerd-shim**: A shim between containerd and runC. Before starting a container, containerd starts a child process to call runC to start the actual container process. This child process is also responsible for monitoring the container process and reporting container runtime information to containerd.

- **runC**: In underlying runtime, it functions as a command line tool that is responsible for actually starting the container process and initializing various namespace and cgroup information.

Container status management process:

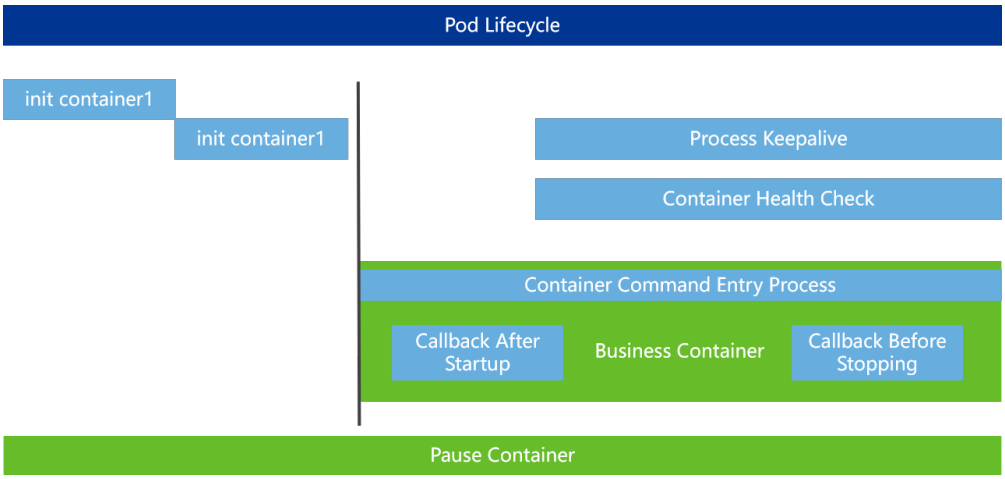


Figure 8: Container Status Management

1. Kubernetes (K8s) uses a Pod to start a pause container to perform the necessary namespace and cgroup initialization settings.
2. Before starting the actual business container in a Pod, K8s runs init containers defined in the Pod one by one in a serial manner.
3. After all the init containers have finished running, K8s creates the actual business container in Pod and executes the command defined in the YAML file to start the container process.
4. After starting the container process, K8s executes the postStart callback command defined in the YAML file.
5. While the container is running, K8s executes the liveness and container monitoring probes defined in the YAML file.
6. Before the container stops, K8s executes the preStop callback command defined in the YAML file.

4.1.2.2.3 Container Image Management

A container image is required for creating a container. The container image is a

static file that consists of multiple read-only layers.

Layered image structure:

The layered image structure of the HCI components is illustrated in the following figure.

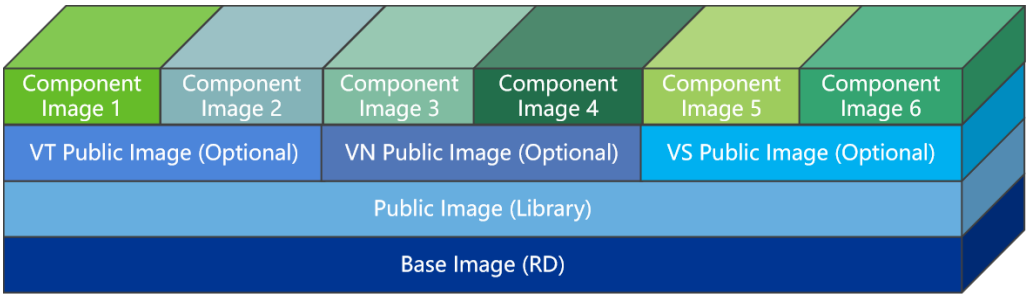


Figure 9: Layered Structure of Container Images

Advantages:

1. Independence: Each layer functions independently and focuses on a specific function.
2. Flexibility: Changes in any layer do not impact the layers above or below as long as the relationship between the interfaces of different layers remains unchanged.
3. Separability in structure: Each layer can be implemented using the most suitable technology for its specific purpose.
4. Ease of implementation and maintenance.
5. Standardization: The functions and services of each layer are precisely defined, ensuring operation standardization.

Image update:

1. Facilitates incremental updates to existing image files using service packs

When running a container, any file modifications made are recorded at the storage layer of the container. The storage layer can be saved as an image by executing the **commit** command. In other words, a new image is created by adding the storage layer on top of the existing container image. The new container image, during its runtime, contains all the file modifications made in the original container, which facilitates incremental updates to existing image

files using service packs.

2. Facilitates updated container image import in update scenarios

Only layer04 of the image is loaded locally due to the layer sharing mechanism, as shown in the following figure.

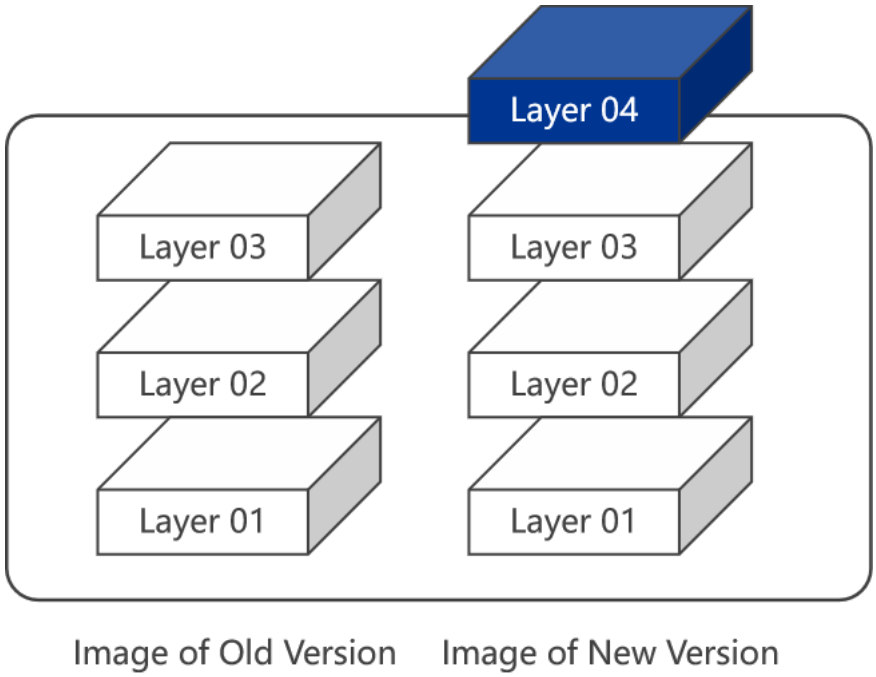


Figure 10: Container Image Update

In the scenario where layer01, layer02, and layer03 are all referenced by both the new and old images, deleting the old image only removes the references to layer01, layer02, and layer03 but does not delete the layers.

3. Implements base image sharing between the new container images and old container images

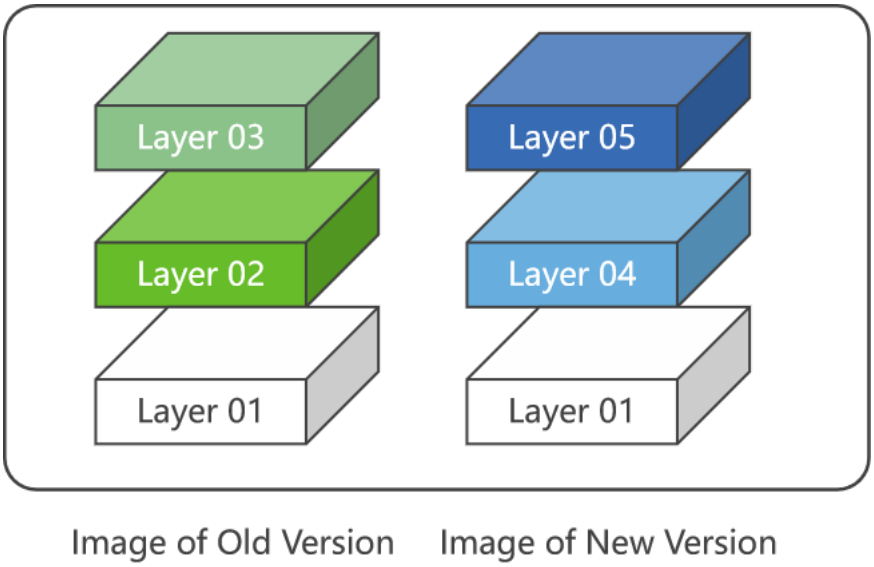


Figure 11: Base Image Sharing for Update

In the scenario where only layer01 is referenced by both the new and old images and layer02 and layer03 are not referenced by any images, deleting the old image results in the deletion of layer02 and layer03, as well as the files decompressed to the **snapshotter** directory from these layers.

4. Does not implement image layer sharing between the new container images and old container images

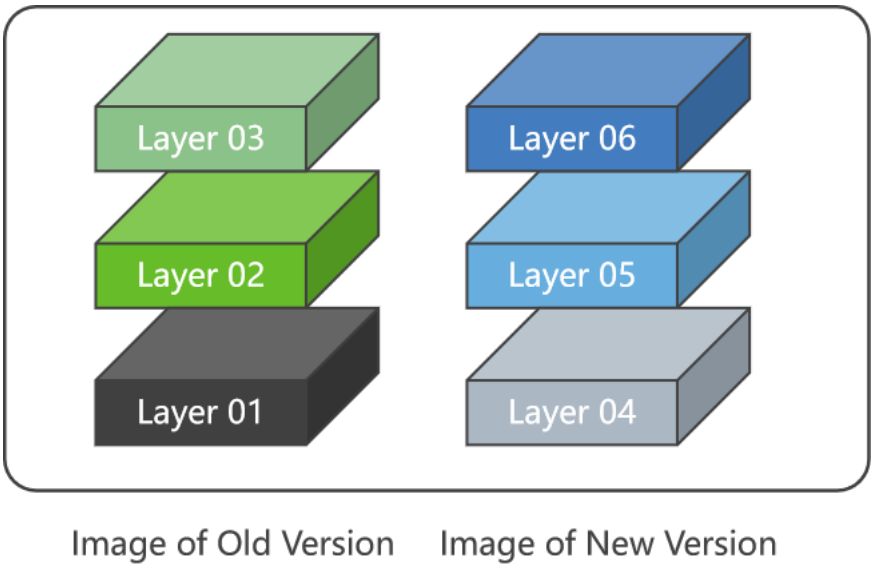


Figure 12: No Base Image Sharing for Update

In the scenario where none of the old images are referenced by new images

and layer01, layer02, and layer03 are not referenced by any images, deleting the old image results in the deletion of all layers, as well as the files decompressed to the **snapshotter** directory from these layers.

4.1.2.2.4 Image Repository Optimization

HCI 6.9.0 utilizes SquashFS to implement unified compression of container images on the disk and enable the compression and mounting of file systems. SquashFS is a compressed read-only file system based on the Linux kernel. It is capable of compressing documents, inodes (each inode stores the metadata of an object in the file system), and directories, with a maximum file size of 2^{64} bytes. The key features of SquashFS are as follows:

1. Supports the compression of data, inodes, and directories.
2. Preserves complete 32-bit UID/GID and file creation time.
3. Supports a maximum file system size of 4 GB.
4. Features highly compressed inodes and directories, with an average compressed inode length of 8 bytes.
5. Supports larger block sizes to implement higher compression rates than the 4K block size.
6. Introduces the concept of fragment blocks, which store multiple files with a size smaller than that of a block in a single block to implement higher compression rates.
7. Detects and deletes duplicate files.

4.1.2.3 Homogeneous Dual-Stack Architecture

HCI 6.8.0 and earlier versions feature a strong coupling and heavy dependency between the HCI business logic (aSV, aNET, aSAN) and the platform OS, which is manifested in the following aspects:

1. The HCI business logic has a strong dependency on the toolchain. As a result, it is required to synchronize the updates to the entire compilation environment and a series of OS library versions when the userspace

toolchain is upgraded or changed, which leads to a substantial workload for adaptation and testing.

2. The HCI business logic has a certain dependency on the critical system files such as **/proc**, **/sys**, and **cgroup** directories of the OS. Some HCI business codes directly read the content of files under the **/proc**, **/sys**, and **cgroup** directories. Consequently, kernel upgrades or modifications might result in changes to the file paths or content in these directories and lead to abnormal HCI functions.
3. The HCI business logic has a certain dependency on the underlying hardware. The upgrade or modification of the OS kernel or toolchain or adapting to new hardware can cause changes in the results returned by hardware tools, leading to abnormal HCI functions.

The latest HCI has addressed these issues by decoupling the HCI business logic from the platform OS through the OS block layer developed by Sangfor. The figure below illustrates how the block layer works.

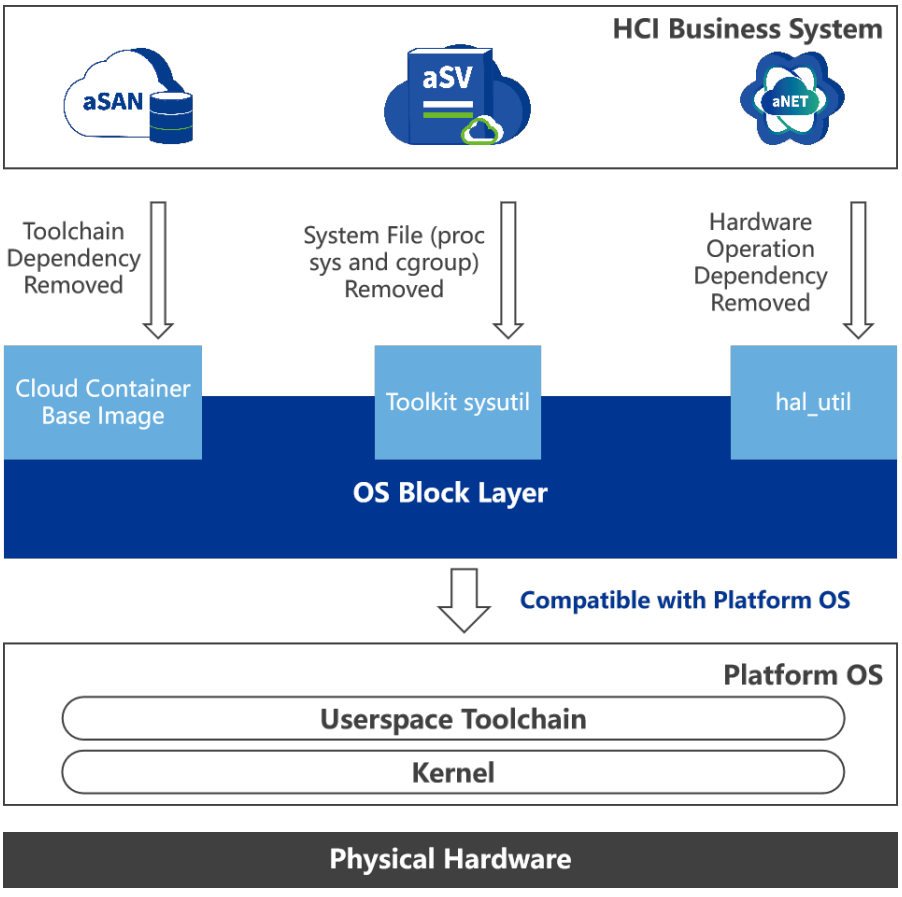


Figure 13: OS Block Layer

1. A public cloud container image is built in the OS block layer. Try not to change the cloud container base image to meet business requirements. The HCI business logic is containerized based on the cloud container base image. Therefore, upgrades or changes to the platform OS toolchain will not impact the business system, facilitating decoupling from the platform OS toolchain.
2. A cross-platform toolset called sysutil is also contained in the OS block layer, which blocks the differences in system files (under **/proc**, **/sys**, and **cgroup** directories) between various Linux versions or kernel versions and provides unified functional interfaces for the upper-layer HCI business logic. If the HCI business logic needs to retrieve specific system information, direct access to the system files (under **/proc**, **/sys**, and **cgroup** directories) is not required, and the functional interface of the sysutil toolset is called, thus ensuring decoupling from the system files (under **/proc**, **/sys**, and **cgroup** directories).
3. Furthermore, a lightweight hardware abstraction layer (HAL) solution, hal_util, is designed for cloud services in the OS block layer. The business system can call the functional interfaces provided by hal_util rather than directly call the low-layer hardware operation commands for hardware operations, which helps decouple the low-layer hardware operations.

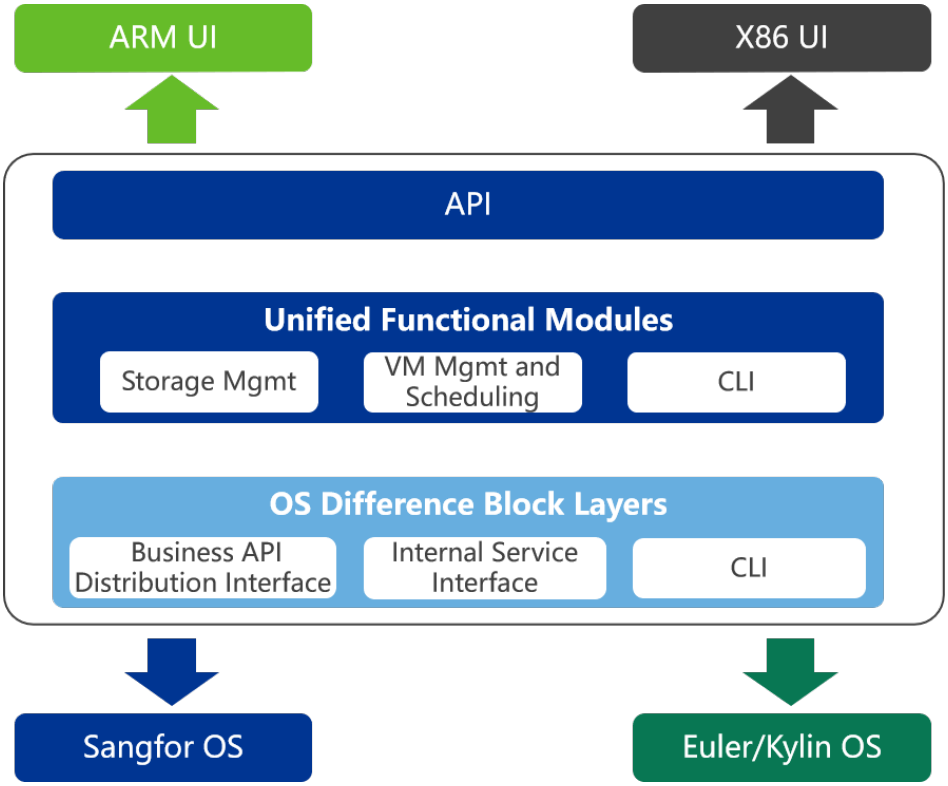


Figure 14: Unified Architecture for Arm and x86

The OS block layer enables Sangfor HCI to implement consistency in HCI business logic. Leveraging the OS block layer, HCI quickly adapts to different OSs, provides business delivery, internal service, and command-line interfaces, and effectively blocks the differences among various physical servers, thus enabling hardware compatibility for X86 and Arm servers. For instance, the functionality of reading CPU models varies across different OSs. In response to this issue, the HCI business logic transfers the interface request for reading the CPU model to the OS block layer, which then adapts to different types of CPUs and OSs at the lower layer. Therefore, no changes are required for the business logic layer.

HCI 6.9.0 boasts a set of codes. It can manage a single cluster and ensure unified management of server clusters with multiple architectures with the help of SCP.

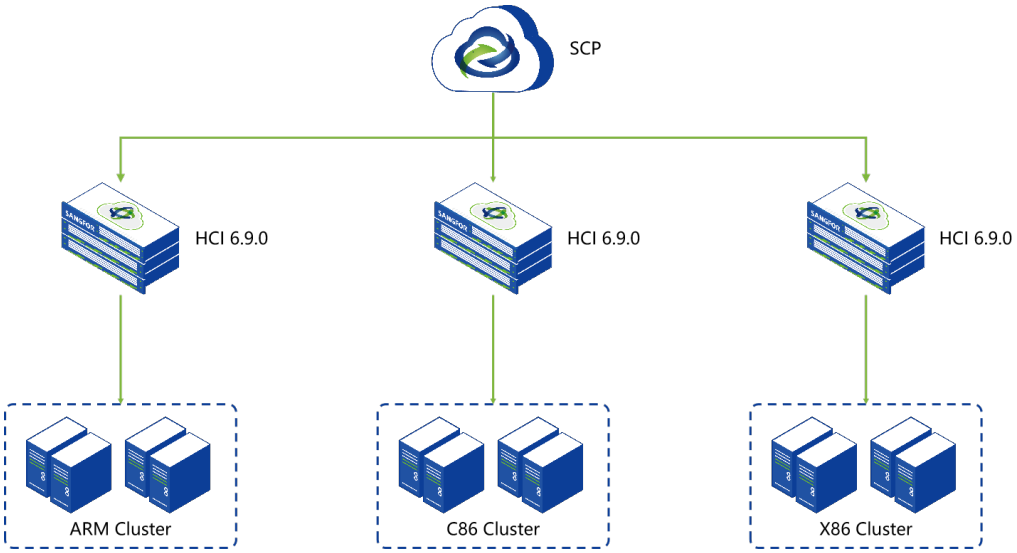


Figure 15: Deployment Architecture for Arm and x86

4.1.2.4 Global HA Architecture

HCI globally adopts a design that ensures high availability of resources. In order to achieve high availability based on the information platform, modules for data collection and local processing are deployed on each HCI node to transmit processed data to the local information platform. The control plane on each node consumes data obtained from the information platform, makes logical assessments, and takes appropriate actions to resolve issues.

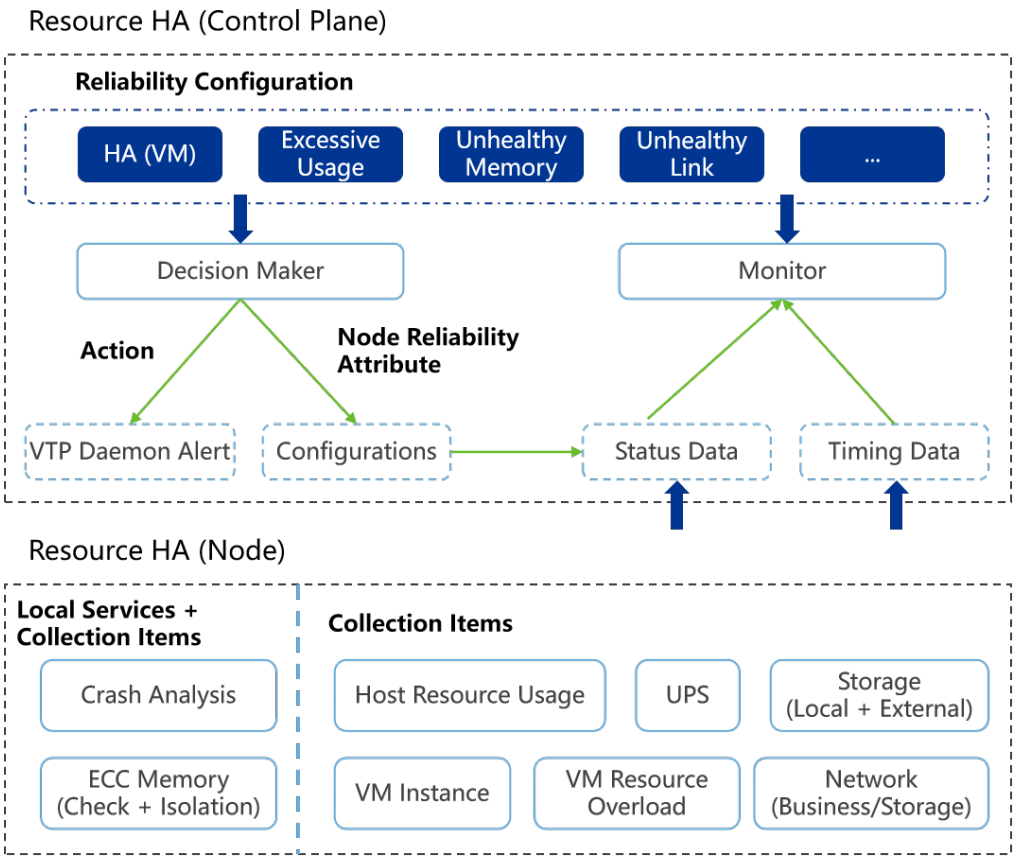


Figure 16: Resource HA Design

1. The HA subsystem is designed to implement resource HA for the entire platform. It replaces scattered modules related to reliability, including node failures, HA modules (covering various fault scenarios such as VM network unavailability and unresponsive VMs), UPS modules, exceeding load limits, and isolating subhealthy hosts.
2. The HA subsystem is designed based on the information platform. It extends the exporter to add corresponding monitoring or collection item data, which is then obtained by the control plane through the information platform with the polling mechanism. The control plane then analyzes the data and takes appropriate actions.
3. The HA subsystem is divided into two layers to cover reliability scenarios from the cluster and node aspects. The node-related scenarios include isolation based on error correction codes (ECC), event monitoring and filtering, and application process startup.

4. Resource usage, VM instances, and VM resource consumption status and metrics are collected through the built-in exporters for nodes and VMs.

4.1.3 Deployment Architecture

4.1.3.1 Plane Division

For medium and large-scale resource pools, global control and management of VMs, data protection, resource allocation, HA, and O&M are required. In this case, the architecture is divided into two planes, namely, the control plane (for cluster management) and the node plane (for management of computing and other resources). The modules of the control plane operate in a unified manner for global management, and some modules can operate in a single host-based manner for specific resource management.

- The control plane provides services in a global manner. The services include VM management, resource management and allocation, and O&M.
- The node plane provides services in a single host-based manner and is managed by the control plane. The services include actual resource management and business or task bearing.

Deployment effects:

1. The control plane is independently deployed as controller nodes.
2. Independent operating environment and resource quota are available in converged deployment scenarios, and resource isolation can be implemented.

4.1.3.2 Deployment Modes

Three deployment modes are available for different scenarios. The same MySQL is used for aSV and aNET. The figure below illustrates the cluster deployment mode of two nodes.

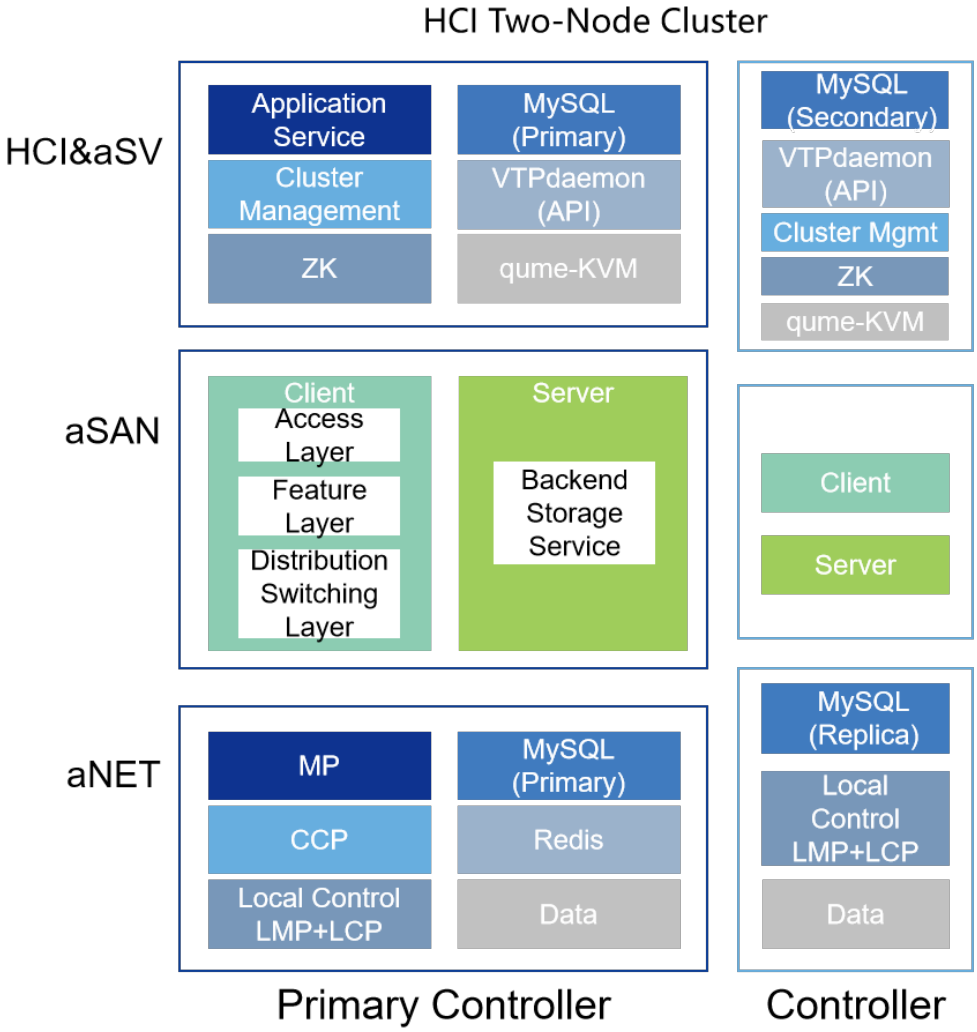


Figure 17: Two-Node Cluster Deployment Mode

The figure below illustrates the cluster deployment mode of three or more nodes.

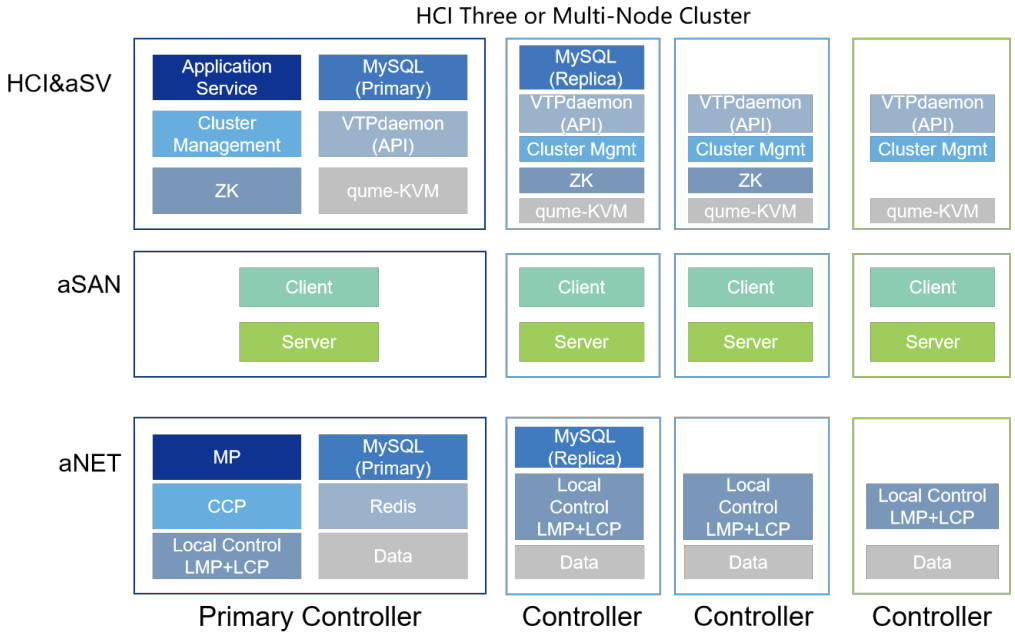


Figure 18: HCI Three-Node Cluster Deployment Mode

The figure below illustrates the deployment mode of multiple clusters.

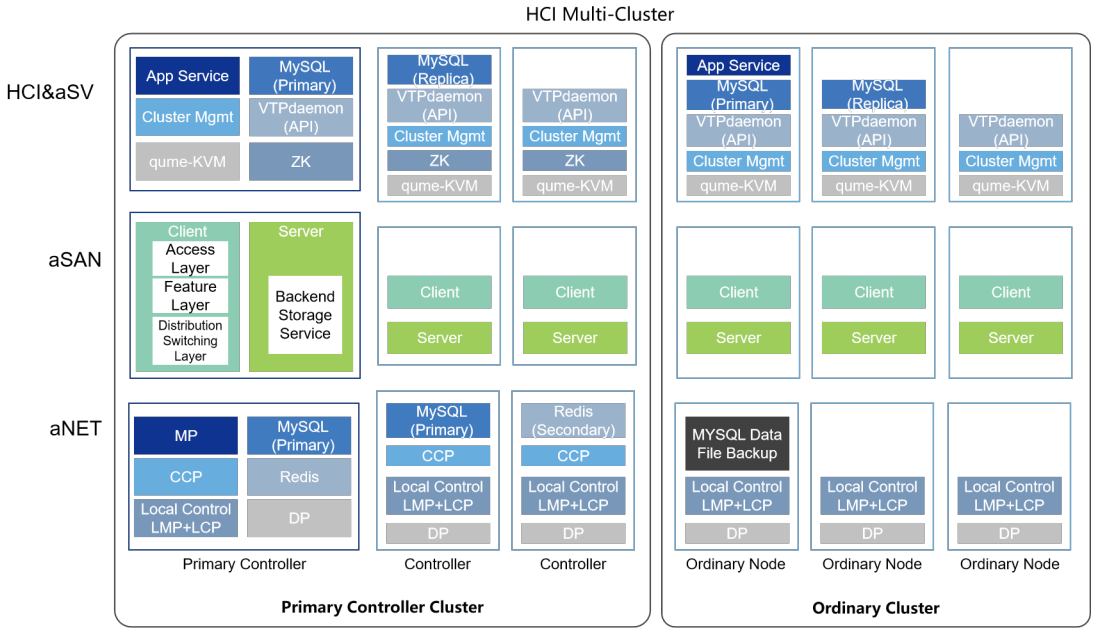


Figure 19: HCI Multi-Cluster Deployment Mode

4.1.4 Network Architecture

HCI consists of four logical network planes, and each of them can be deployed

independently or combined for sharing. The four logical network planes are the management plane, business plane, data plane (VXLAN), and storage plane.

- **Management plane**: It is designed to provide network services for HCI clusters and enables administrators to manage HCI clusters. The management network is connected to M-LAG or two stacked access switches through link aggregation. The failure of a single switch or link does not affect the stability of HCI.
- **Business plane**: It provides external network services for VMs. The business network is connected to M-LAG or two stacked switches through link aggregation to implement link redundancy. Static network interface binding can be configured for the outbound business interface. Each server can have multiple physical NICs configured as outbound business interfaces to enable VMs in the virtual network to have redundant and reliable business network connections.
- **Data plane (VXLAN)**: It handles east-west traffic between VMs and virtual network devices for network communication between them. VXLAN is connected to M-LAG or two stacked switches through link aggregation to implement link redundancy. HCI features built-in distributed virtual switches and virtual routers, and a virtual switch instance is configured for each host in the cluster. If one host goes offline, traffic routed through the virtual switch instance on this host is redirected to and handled by another host that has high availability.
- **Storage plane**: It is used for I/O data traffic storage. By establishing a dedicated private storage network, HCI ensures data security and implements link aggregation for the storage network. Link aggregation of the private storage network can implement load balancing based on TCP connections. Different TCP connections between two hosts can use different physical links to achieve maximum throughput.

The four network planes are logically isolated, and the failure of one plane does not affect the other planes.

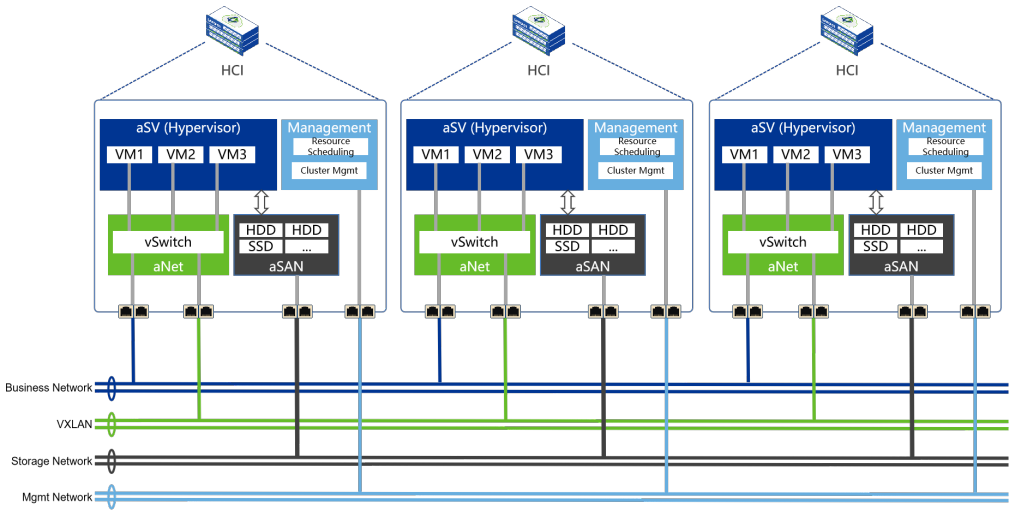


Figure 20: Network Link Aggregation

4.2 HCI Advantages

HCI boasts the following architectural characteristics and advantages:

1. Layered and Decoupled Architecture to Enhance Software Control and Openness

The business layer provides O&M capabilities such as alerts and monitoring and offers open APIs for compatibility with cloud management and third-party management. Advanced functions such as HA and disaster recovery are available globally.

The resource definition and service layer implement the software-defined storage (SDS) and SDN capabilities and facilitate integration with various types of resource pools such as computing, storage, and heterogeneous third-party systems. By calling resource APIs to request resources and encapsulating them into unified resource objects, the resource definition and service layer realize the model of Resource as a Service (RaaS).

The cluster management layer is responsible for the lifecycle management of clusters, including cluster construction, deployment, and updates.

2. Global Management to Support Larger Resource Pools

The system management capability is improved from node-based management to pool-based management as the scale expansion of customer

resources. The global control plane is required for VM management, data protection, resource allocation, HA, and O&M. The node plane on the host layer is managed by the control plane and is used to provide actual resources and bear business operations and tasks.

Centralized management of the control plane reduces resource consumption and network overhead caused by distributed control over nodes. This helps improve node performance and expand the scalability of node management.

The deployment mode of the management node and computing, storage, and network resources can be both converged or separate. Converged deployment implements rapid delivery and simplified management, while separate deployment facilitates high scalability of resource nodes.

3. Fine-Grained Resource Scheduling to Support High-Performance and Low-Latency Database Businesses

VM resource tags ensure that VM resources on corresponding nodes are scheduled. To meet the business requirements for high performance, high throughput, and low latency, the underlying mechanisms, such as big memory and resource loading, are automatically scheduled. Resources are preferentially guaranteed based on the automatic learning business model.

4. Centralized Control of SDN to Support Super-Large Scale Management

Based on the newly designed SDN architecture, business network-related configurations, such as network topology and ACL, are centrally maintained by the SDN controllers, which are clustered and deployed in HA mode to ensure business network robustness.

Physical network devices are only used for data forwarding to implement rapid scalability.

5. Multi-Cluster Layer 2 Communication Domain to Optimize Active-Active Disaster Recovery Capabilities

HCI facilitates Layer 2 network connectivity between multiple clusters or sites through VXLAN for larger-scale business capacity.

The What-You-Draw-Is-What-You-Get function ensures unified network orchestration and centralized configuration across clusters, making network

configuration management for active-active DR in data centers simpler.

5 Key Technologies

The key technologies of HCI mainly include aSV, aSAN, aNET and aSecurity, which will be introduced in detail in the following parts.

5.1 aSV

Leveraging the Linux kernel-based virtual machine (KVM) technology on a bare metal architecture, aSV incorporates advanced computing functions to provide customers with a practical, convenient, self-managed, stable, reliable, and high-performance computing platform.

5.1.1 Overview

Key technologies of aSV, represented by Sangfor file system (SFFS), kernel hardening, and resource isolation, help ensure the reliability and security of HCI.

5.1.1.1 Introduction to Hypervisor Architecture

A hypervisor, also known as a virtual machine monitor (VMM), is a computer software that runs between the physical server and the OS. It enables multiple OSs and applications to share the same underlying physical hardware. In this sense, it can be considered as the meta OS in a virtual environment. It also coordinates access to all physical devices and VMs on the server.

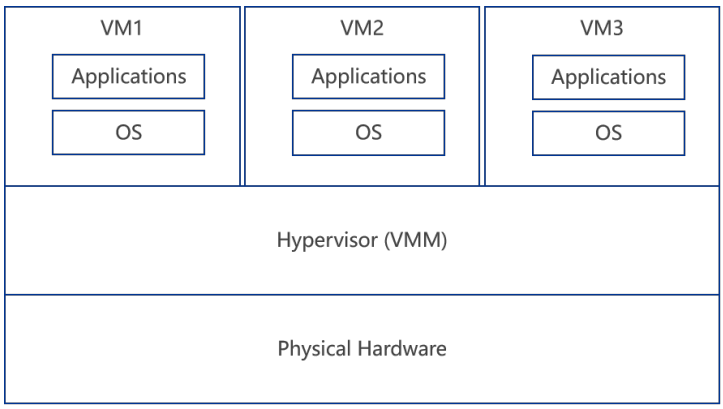


Figure 21: Hypervisor Architecture

As the core of all virtualization technologies, the hypervisor boasts a fundamental function to support the uninterrupted migration of multiple workloads. When started and executed by the server, the hypervisor allocates appropriate memory, CPU, network, and disk resources to each VM and loads the guest OSs for all VMs.

The VMM virtualizes physical resources in three parts, namely, CPU virtualization, memory virtualization, and I/O device virtualization. Among them, CPU virtualization is the most critical.

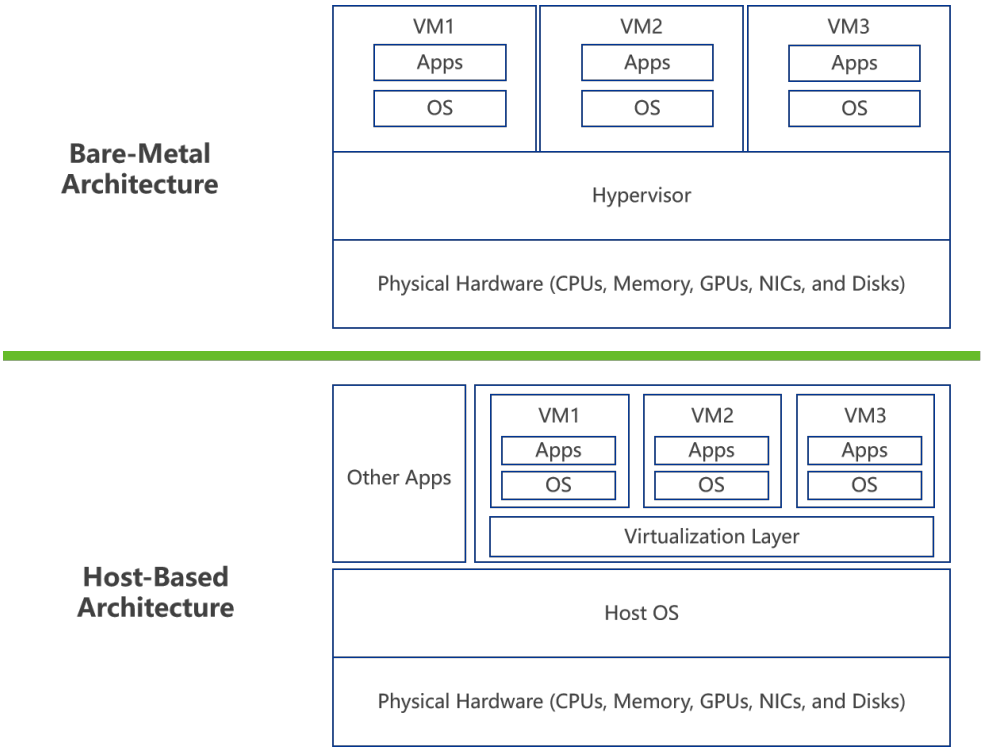


Figure 22: Hypervisor Architecture Types

Host-based virtualization and bare-metal virtualization are two virtualization technologies for major manufacturers, with the latter one being the current mainstream solution. Bare-metal virtualization is enhanced based on hardware virtualization extensions such as Intel Virtualization Technology (Intel VT).

- Bare-metal virtualization

Bare-metal virtualization indicates that a VMM operates directly on the bare machine to utilize and manage the underlying hardware resources. The guest OS needs to access the actual hardware resources through the VMM. As the

operator of the underlying hardware and installed with necessary hardware drivers, the VMM directly manages and accesses the hardware resources without requiring an underlying OS. A bare-metal VMM functions as a thin OS and has a performance between that of host virtualization and OS virtualization.

Typical bare-metal VMMs include VMware ESXi Server, Citrix XenServer, Microsoft Hyper-V, and Linux KVM.

- Host-based virtualization

Host-based virtualization indicates that an additional host OS is deployed below the VMM. Since the guest OS needs to access the hardware through the host OS, additional performance consumption is generated. However, the device drivers and underlying services provided by the host OS can be used for memory management, process scheduling, and resource management. When applications within VMs call hardware resources, the request passes through the VM kernel (i.e., the guest OS kernel) to the VMM and finally reaches the host kernel (i.e., the host OS kernel). This leads to a lower performance of host-based virtualization compared to bare-metal virtualization.

Typical host-based VMMs include VMware Server (GSX), VMware Workstation, Microsoft Virtual PC, and Virtual Server.

5.1.1.2 Hypervisor Architecture Implementation

Due to the low performance of host-based VMM, Sangfor aSV adopts the Linux KVM virtualization of bare-metal VMM for higher efficiency.

KVM is a virtualization technology based on the Linux kernel. It has been integrated into major Linux kernel versions since Linux 2.6.20. KVM utilizes the scheduler of Linux for management, so it features less core source code compared with Xen.

As a module (kvm.io) of the standard Linux kernel, KVM is responsible for CPU and memory virtualization, including creating guest VMs, allocating virtual memory, switching CPU execution modes, accessing vCPU registers, and executing vCPUs. It is necessary to load the KVM module before creating VMs.

However, only loading the KVM module is not sufficient. A userspace tool is also required for users to control the kernel. In this case, Sangfor utilizes a well-developed open-source virtualization software called QEMU.

QEMU is a regular Linux process that provides guests with the device emulation function, which can emulate BIOS, PCI/PCIe buses, disks, NICs, graphics cards, sound cards, keyboards, and mice. QEMU interacts with the KVM module running in kernel mode by using ioctl system calls. In the KVM virtualization environment, each VM can be considered as a QEMU process running within the address space of the QEMU-KVM process in Linux.

By utilizing the virtual machine extension (VMX) mode of hardware virtualization extensions such as Intel VT, KVM virtualization ensures that the CPU operates in two modes, namely, root operation mode and non-root operation mode. Each mode has its own ring 0 and ring 3 levels.

The KVM hosts run in the root operation mode, with the host kernels at the ring 0 level and userspace programs at the ring 3 level. The guest VMs run in the non-root operation mode, with their kernels at the ring 0 level and userspace programs at the ring 3 level.

When the VMM creates a guest VM, the QEMU-KVM in userspace mode on the KVM host will call ioctl to create the guest VM and vCPU through the device under **/dev/kvm**. The KVM host kernel module also makes some preparations upon receiving the request, such as loading the vCPU context into the virtual machine control structure (VMCS). Then, it drives the CPU to trigger VM_Entry to enter the VMX non-root mode and execute the guest VM code. When the guest OS of the guest VM needs to send I/O requests or network packets, VM_Exit is triggered to switch the CPU to root mode. The KVM host takes over and processes the instructions. After the processes are complete, the KVM host triggers VM_Entry again and enables instruction execution in the guest OS of the guest VM.

QEMU-KVM operates in three modes:

- **Guest mode** (ring 0 to ring 3): In this mode, the guest OS operates within the virtualization environment, and the CPU runs in VMX non-root operation mode. Guest mode has its own kernel and user modes.

- **User mode** (ring 3): In this mode, QEMU operates within the virtualization environment as a userspace tool that helps users with VM management and performs I/O on behalf of users. The CPU runs in VMX root operation mode.
- **Kernel mode** (ring 0): In this mode, the CPU and memory are emulated to facilitate the switchover of the guest mode. The KVM kernel module operates within the virtualization environment, and the CPU runs in VMX root operation mode.

The relationship between KVM and QEMU is shown in the figure below.

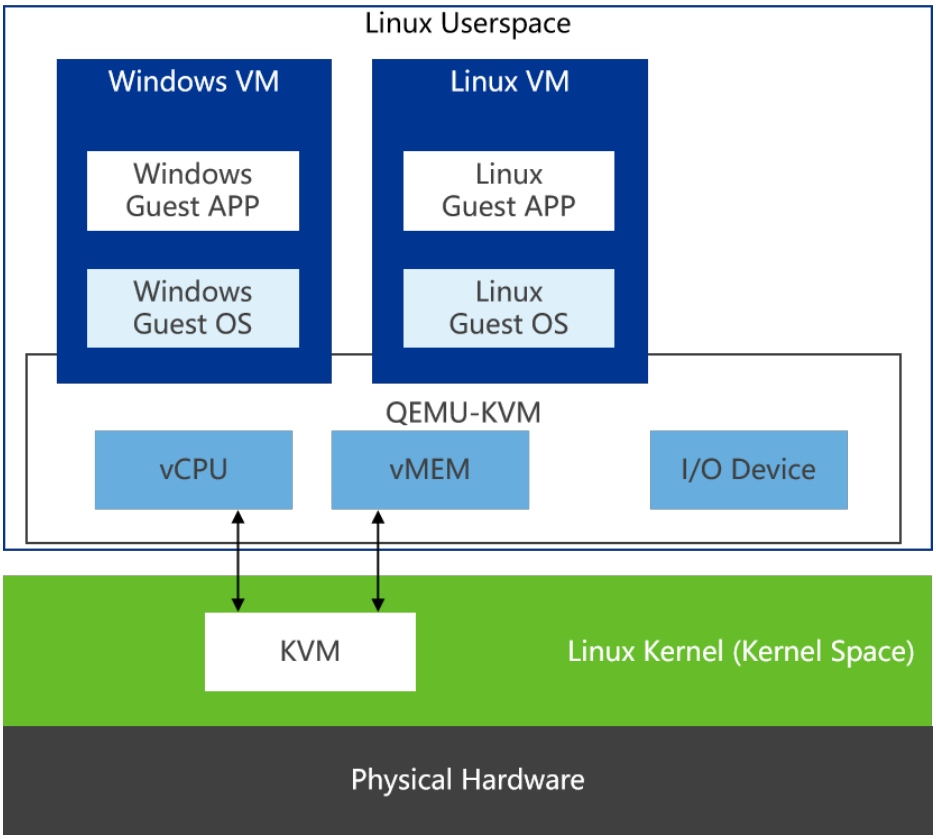


Figure 23: Relationship Between KVM and QEMU

In general, KVM consists of two parts. One is a device driver that manages virtual hardware and utilizes the character device under **/dev/kvm** as the management interface. The other is a userspace component that emulates PC hardware and is a QEMU process.

Native functions supported by QEMU-KVM:

1. CPU and memory overcommitment
2. I/O paravirtualization
3. Hot swapping (CPU, block devices, network devices, and more)
4. Symmetric multi-processing (SMP)
5. Live migration
6. PCI device allocation and single root I/O virtualization (SR-IOV)
7. Kernel same-page merging (KSM)
8. Non-uniform memory access (NUMA)

Advantages for integrating aSV with KVM:

1. Enhanced compatibility through integration into the Linux kernel
2. Improved performance with code-level resource calling
3. Simplified memory management with VM serving as a process
4. Increased scalability by supporting NUMA
5. Strong community support by maintaining an open-source development model

5.1.1.2.1 vCPU Mechanism

1. vCPU mechanism

VM does not directly interact with the pCPU. Instead, it only interacts with the vCPU assigned by the VMM and utilizes the vCPU as its computation unit. Each vCPU features a virtual machine control structure (VMCS). When a vCPU is switched off from a physical CPU, its running context is saved in its VMCS. When a vCPU is switched to run on a physical CPU, its running context is imported from the corresponding VMCS to the physical CPU. This process enables the independent operation of each vCPU.

2. vCPU scheduling mechanism

Based on the structure and function division of a VM system, the guest OS and the VMM constitute a two-level scheduling framework. The guest OS is responsible for the second-level scheduling, which involves scheduling threads

or processes onto vCPUs (mapping core threads to corresponding vCPUs). The VMM is responsible for the first-level scheduling, which involves scheduling vCPUs on physical processing units. The scheduling policies and mechanisms of the two levels are independent. The vCPU scheduler allocates and schedules physical processor resources among VMs. Specifically, it schedules vCPUs in each VM according to certain policies and mechanisms and allocates them to physical processing units to meet different requirements of VMs. vCPUs can be scheduled to one or multiple physical processing units by using either time-division multiplexing or space multiplexing to implement concurrent execution. Also, vCPUs can establish a one-to-one fixed mapping relationship with physical processing units by restricting access to a specified physical processing unit. The figure below illustrates a two-level scheduling framework of a VM in a multi-core environment.

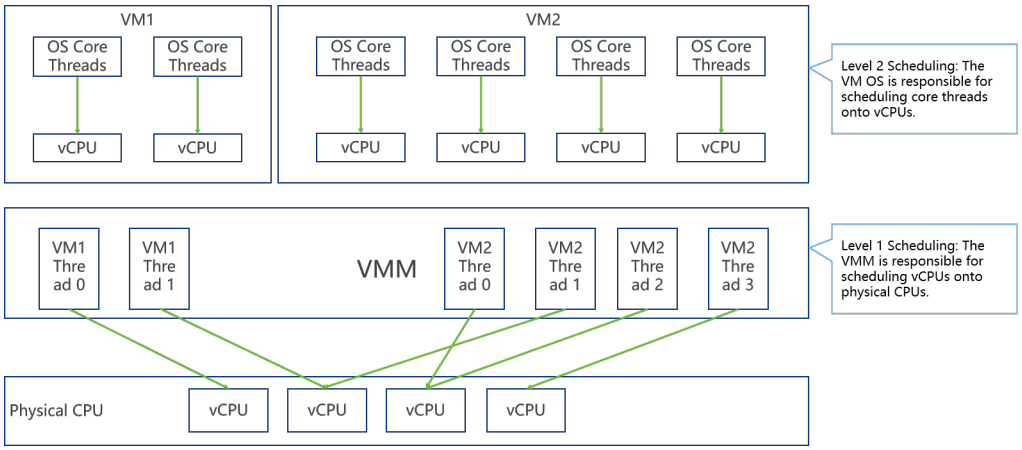


Figure 24: vCPU Scheduling Mechanism

5.1.1.2.2 Memory Virtualization Mechanism

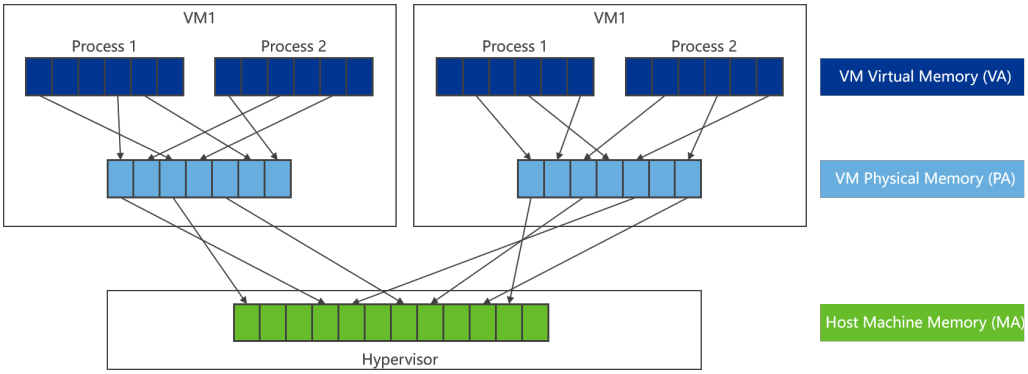


Figure 25: Memory Virtualization Model

The VMM controls all system resources, including the memory resources. Therefore, it is responsible for paged memory management and maintaining the mapping relationship between virtual addresses and machine addresses. As the guest OS features a paged memory management mechanism, the VMM-based system has an additional layer of mapping compared to a regular system.

Terms:

- **Virtual address** (VA) refers to the linear address space provided by the guest OS for its applications.
- **Physical address** (PA) refers to the pseudo address obtained by the VM after being abstracted by the VMM.
- **Machine address** (MA) refers to the actual address on the address bus.

The mapping relationships are as follows: Guest OS: PA = f(VA); VMM: MA = g(PA). The VMM maintains a set of page tables and is responsible for the mapping from PA to MA. The guest OS maintains a set of page tables and is responsible for the mapping from VA to PA. When a user program accesses VA1 during actual runtime, it is translated by the page table of the guest OS into PA1 and then translated by the VMM into MA1.

Page table virtualization:

Common memory management units (MMU) can only perform a single mapping from a virtual address to a physical address. However, the physical address obtained through MMU translation in a VM environment is not the actual machine address. It is necessary to employ the VMM to perform another mapping for obtaining the actual machine address on the address bus. However, it would be impractical and inefficient if the VMM is required for every memory access and address translation. In order to ensure efficient translation from virtual addresses to actual machine addresses, the commonly adopted solution is to generate a composite **fg** command based on the mapping of **f** and **g** and then write **fg** into the MMU. The current page table virtualization methods primarily include MMU paravirtualization and shadow

page table, with the latter being replaced by hardware-assisted memory virtualization.

- MMU paravirtualization

The basic principle of MMU paravirtualization is as follows: When the guest OS creates a new page table, it allocates a page from the free memory under its maintenance and registers the page with the VMM. The VMM then takes the write permission of the guest OS for that page table. Any subsequent write operations on that page table by the guest OS are intercepted by the VMM for validation and conversion. The VMM examines the page table to ensure that only machine pages of the VM are mapped and that no writable mappings to the pages of the page table are contained. Based on the maintained mapping relationships, the VMM replaces the physical addresses in the page table with the corresponding machine addresses and then loads the modified page table into the MMU. After these processes are complete, the MMU can directly convert the virtual addresses to machine addresses based on the modified page table.

- Hardware-assisted memory virtualization

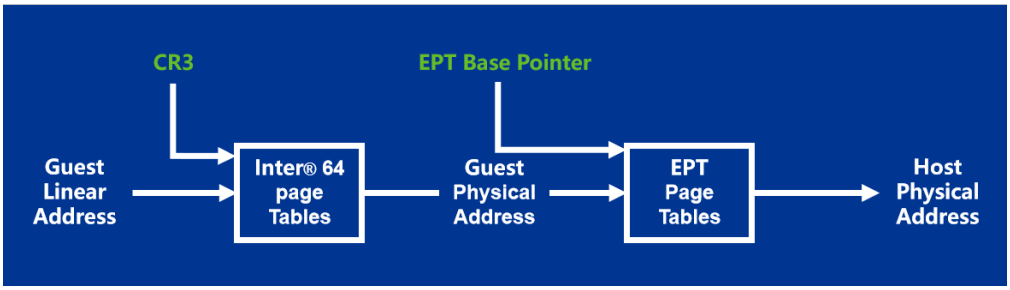


Figure 26: Hardware-Assisted Memory Virtualization Principle

Hardware-assisted memory virtualization is used to replace the shadow page table. Its basic principle is as follows: CPU hardware automatically performs two address translations from the guest virtual address (GVA) to the guest physical address (GPA) and from GPA to the host physical address (HPA), which avoids the issue of high memory consumption and poor performance in software-based solutions. Take the extended page table (EPT) of the Intel VT-x technology as an example. The VMM configures the EPT page table for converting GPA to MA to the CPU in advance. The guest OS modifies its own

page table without VMM intervention. The CPU then automatically looks up the two page tables to convert GVA to MA. By leveraging the hardware-assisted memory virtualization technology, the guest OS can run without VMM intervention, which eliminates substantial software costs and ensures that memory access performance is close to that of the physical machine. Therefore, Sangfor adopted this virtualization method.

5.1.1.2.3 I/O Device Virtualization

The VMM uses I/O virtualization to share limited external device resources. It blocks the access requests from the guest OS to I/O devices and emulates their hardware through software. Currently, three methods are available for I/O device virtualization: full device emulation, paravirtualization, and direct I/O.

Full device emulation:

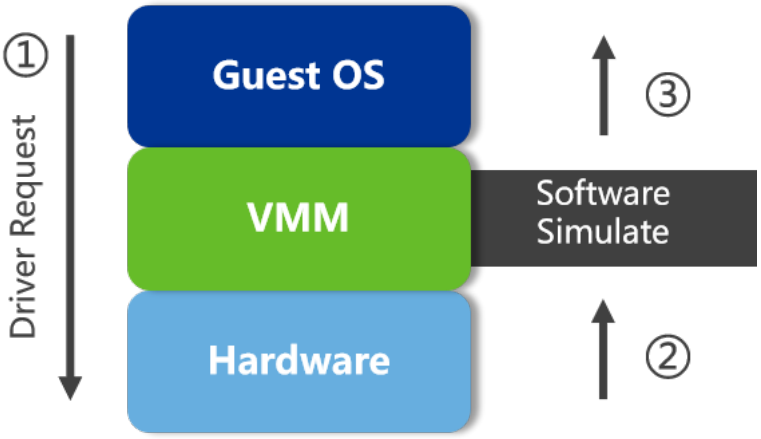


Figure 27: Full Device Emulation

An interface identical to that of a physical device is precisely emulated, which enables the driving of the virtual device without modifying the guest OS.

- **Advantages**: The existing drivers can be used without any additional hardware cost.
- **Disadvantages**: Each emulation requires the operations of multiple registers, resulting in frequent context switches because the VMM needs to block and emulate each register access. The software emulation process also leads to poor performance.

Paravirtualization:

The VMM provides a streamlined backend driver, and the guest OS offers a frontend driver, which directly sends requests from other modules to the VMM through a special communication mechanism with the guest OS. After processing the requests, the VMM backend driver sends notifications back to the frontend driver.

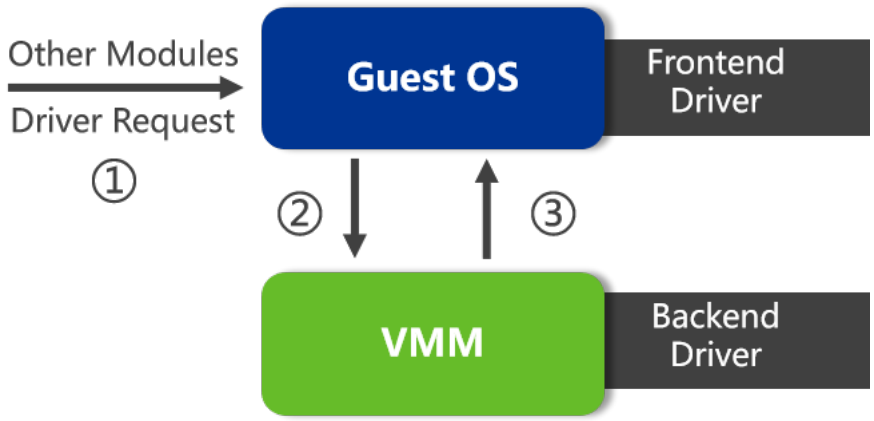


Figure 28: Paravirtualization

- **Advantages**: The transaction-based communication mechanism significantly reduces context switches and eliminates additional hardware costs.
- **Disadvantages**: The guest OS is required for the frontend driver, and the backend driver may become a bottleneck.

Direct I/O:

Physical devices are directly allocated to a specific guest OS for its access to the I/O device without the help of the VMM. I/O MMU, Intel VT-d, and PCI-SIG SR-IOV are the typical technologies designed for efficient direct I/O virtualization.

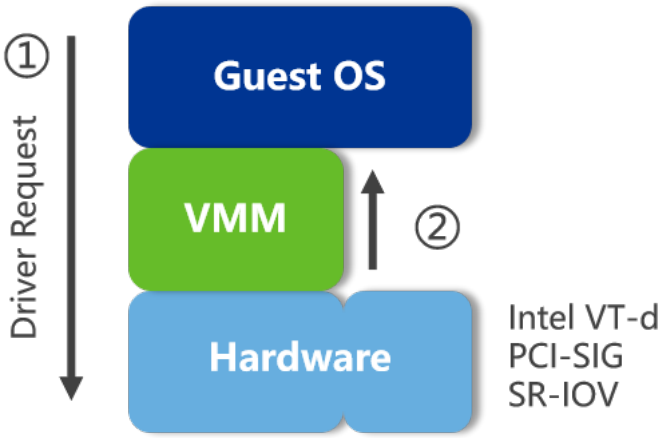


Figure 29: Direct I/O

- **Advantages**: The existing drivers can be used, and direct access reduces virtualization consumption.
- **Disadvantages**: Additional hardware is required.

Sangfor adopts different I/O virtualization methods for different hardware devices. For disks and NICs, full device emulation is used if vmTools is not installed on them, and paravirtualization is used if vmTools is installed. Direct I/O is used for raw disks. Full device emulation is used for mice.

5.1.1.3 SFFS and SFFS Reliability

5.1.1.3.1 SFFS

A cluster consists of multiple physical hosts in a hyper-converged architecture. In order to ensure the high availability of business operations, it is necessary to allow the use of external SAN for data storage. Therefore, it is important to select a suitable file system for managing the capacity of these external storage devices.

SFFS is a cluster file system developed by Sangfor for storing VM image files and configuration files in virtualization scenarios. It is used for special other than general purposes.

Its overall structure is illustrated in the figure below.

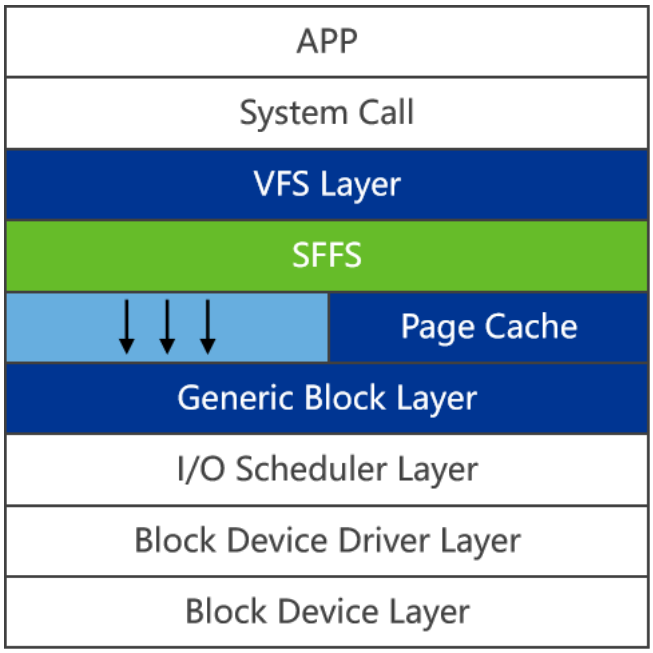


Figure 30: SFFS Structure

SFFS has the following characteristics:

1. Multiple hosts can simultaneously mount and use the same shared disk.
2. The two hosts are equal without the need for a primary controller.
3. It automatically detects the connection status of the mounted host and releases the resources occupied by the host if it is disconnected.
4. It supports online expansion of the file system.
5. It is independent of the host cluster.

SFFS also has the following limitations:

1. File system size: 1,024 TB
2. Maximum size of a single file: 256 TB
3. Maximum file name length: 100 bytes
4. Supported file types: regular files, directories, and soft links
5. Maximum number of files in a directory: 8,192 (including . and .. files)

The location of SFFS in the Linux kernel is illustrated in the figure below.

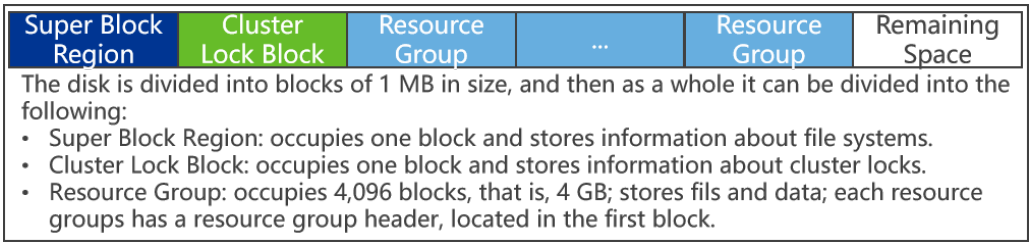


Figure 31: Location of SFFS in Linux Kernel

In virtualization scenarios where data is mostly stored as large files, global file system 2 (GFS2), with a maximum data block size of 4 KB, may result in inefficient allocation and recycling of data blocks and a higher possibility of fragmentation issues. In contrast, SFFS features a data block size of 1 MB, which saves efforts in disk allocation and recycling.

Advantages of SFFS: It establishes a cluster file system among multiple hosts, ensuring simultaneous access to external storage without mutual interference. Compared to local file systems such as ext3 and NTFS, SFFS provides the lock-based protection mechanism when multiple hosts access external storage.

5.1.1.3.2 SFFS Reliability

Currently, key clients widely adopt IT infrastructure that is integrated with both virtualization and external storage, which will remain preferred in the foreseeable future due to the potential risks associated with the transformation as well as data reliability. As HCI mounts iSCSI and FC external storage to SFFS for storage-compute separation, in version 6.10.0, Sangfor resolved and prevented problems such as SFFS corruption and block reuse to greatly improve the reliability in storage-compute separation scenarios, and added new features of discard space reclamation and 256 LUNs for external storage to enhance its capabilities.

1. Supports the flexible use of tools such as that for ATS lock and online block reuse repair for detection and maintenance to continuously ensure the reliability and security of customer data while HCI is running.

2. Supports the discard space reclamation within VMs, which deletes the files of non-pre-allocated VM disks to dynamically release the space they use and improve the usage of the external storage mounted on SFFS.
3. Supports up to 256 external storage mounts (64 in the past) and optimizes resource consumption and execution duration on the storage chain to enhance the overall experience of using external storage.

ATS Lock:

SFFS implements synchronized mutual exclusion of clusters based on the ATS (comparison and write) mechanism of the storage. If the ATS mechanism is unreliable, data problems such as block reuse will occur. Therefore, a tool independent of the file system must perform a dedicated detection on the storage to quickly identify any unexpected behaviors. The script identifies whether an ATS command is reliable by concurrently performing an ATS comparison and writes on the same sector of multiple hosts and then counting whether the number of successes matches the expectation.

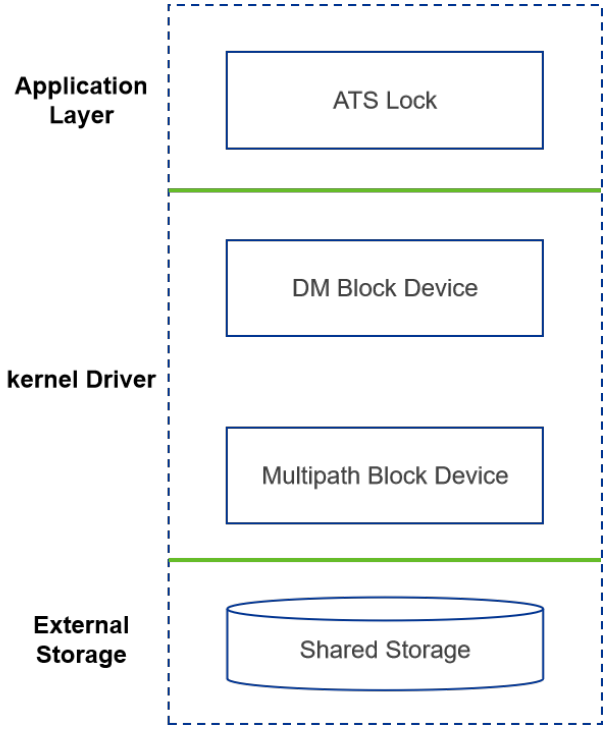


Figure 32: ATS Verification Principle

Discard Space Reclamation:

1. The space used by files stored on thin or dynamically provisioning external storage, such as Sangfor Enterprise Distributed Storage (EDS) or third-party storage, is reclaimed after the files are deleted.
2. HCI formats the external storage into SFFS for use.
3. The external storage releases the blocks of deleted files to free up space for reuse by other external storage after receiving the discard command.
4. HCI 6.10.0 supports releasing space by deleting files of a VM stored on external storage. When a file of a running VM is deleted, the VM will send a discard I/O to QEMU, which converts the I/O address space and then sends a discard I/O request to the storage mounted on SFFS, and SFFS receives the request, converts the I/O, and sends it through the kernel I/O stack to the external storage server, which releases the blocks of the deleted file.

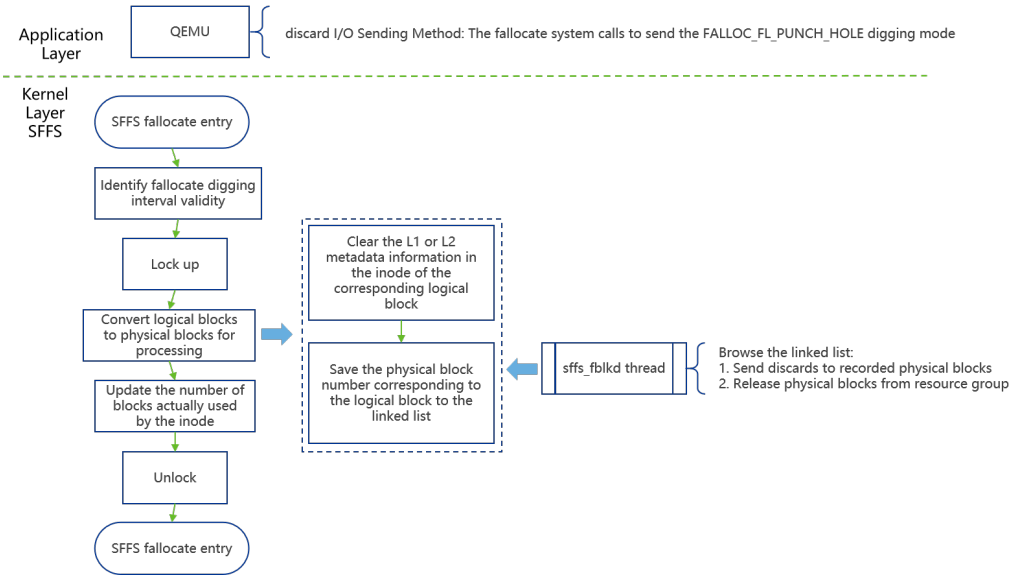


Figure 33: Discard Space Reclamation Principle

Dynamic Reclamation:

When a file is deleted, its blocks that need to be reclaimed are managed through structures such as linked lists and are slowly released in the backend. The storage server will distribute this temporarily free space to other external storage in need. This technique can prevent the I/O bandwidth from being used up in a short time to reduce the performance of other IOs, without

waiting for the release to be synchronized after the deletion.

5.1.1.3.3 UI Display

Log in to HCI, go to **Storage** > **Other Datastores**, and enable **Space Reclamation** (discard space reclamation) for the external storage mounted to SFFS to be created or being edited.

5.1.1.4 QCOW2 Image File

HCI uses QCOW2 image files as virtual disk files for VMs. QCOW2 is a virtual image format similar to RAW, VHD, and VMDK.

The QCOW2 file format has the following characteristics:

1. Supports smaller image files, even for file systems that do not support files with holes
2. Supports copy-on-write
3. Supports snapshots
4. Supports zlib-based compression
5. Supports AES encryption

Disk services are available for VMs in HCI by using QCOW2, RAW, and other formats.

In the application system, the hardware driver implements the disk storage addressing through CHS, which consists of three variables, namely, cylinder, head, and sector. When the guest OS reads or writes to the disk, it passes instructions in CHS format and translates the instructions into logical block address (LBA) values.

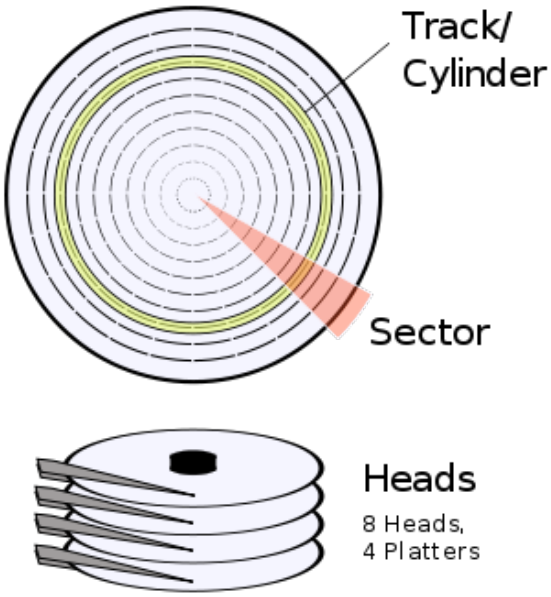


Figure 34: Disk Addressing

When disk I/O instructions are delivered from VMs, they are translated into LBA values. Then, the KVM driver translates the values into specific formats, such as QCOW2 or RAW, of virtual disk image files. Among them, QCOW2 is a commonly used virtual disk image file format. QCOW2 uses a two-level structure, consisting of an L1 table and an L2 table, to describe a file and emulate a fixed-size block device. The L1 table describes the position and offset of the L2 table, while the L2 table directly describes the position and offset of the actual user data. The two tables ensure that the VMs can use the virtual disk image efficiently. The guest OS of the VM can use a block device disk and does not need to adapt to its actual format.

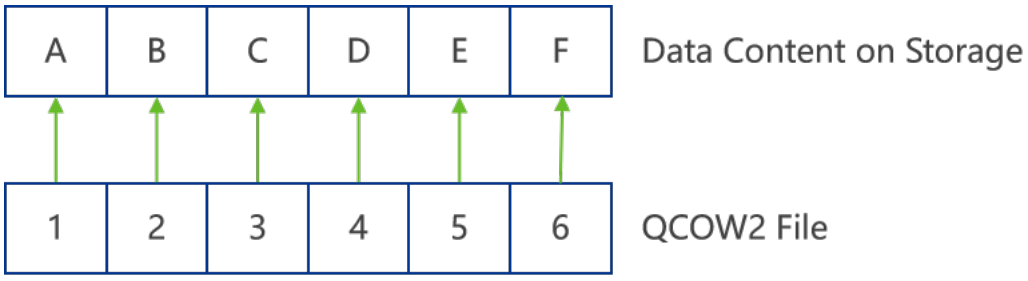


Figure 35: QCOW2 File Format

5.1.1.5 Kernel Hardening

HCI uses the Linux system as the underlying system. Therefore, it is necessary to perform basic security configurations on the Linux OS to ensure its security, which includes:

- **Minimal services**: Disable unnecessary or risky system backend processes and services, such as email agents, graphical desktops, telnet, and compilation tools.
- **Service hardening**: Improve the security of SSH and other common services.
- **Kernel parameter adjustment**: Modify kernel parameters to enhance the security of the OS, such as disabling IP forwarding.
- **File directory permission settings**: Ensure minimal file permissions based on industry hardening standards and application requirements.
- **Account password security**: Enable password complexity and validity checks and set login failure retry limits.
- **System authentication and authorization**: Configure a solution where both administrator authorization and vendor authorization code are required for root user login.
- **Logging and auditing**: Record logs of services and kernel processes with the black boxes of HCI for system operation traceback when necessary.

The above hardening measures ensure that the platform runs with the minimum set of functions in a secure, reliable, and traceable manner.

5.1.1.6 Isolation between VM and Hypervisor

HCI leverages Linux KVM virtualization for compute virtualization, where the VMM operates directly on the bare machine and manages the underlying hardware resources. Access to the actual hardware resources by the guest OS is accomplished through the VMM, which has the hardware drivers to operate on the underlying hardware.

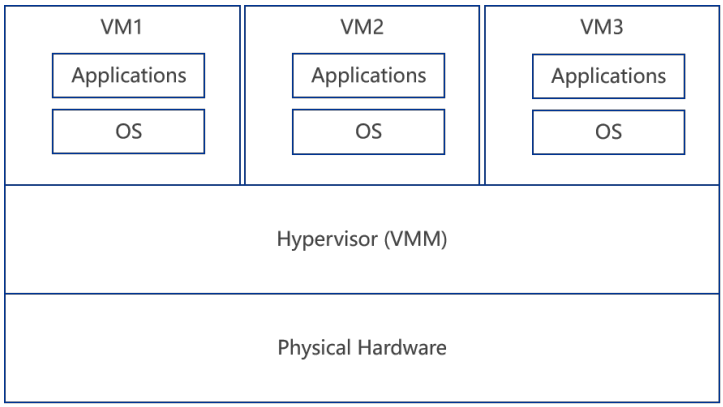


Figure 36: Bare-Metal Virtualization Technology Architecture

The VMM manages physical resources in a unified manner and ensures that each VM can obtain relatively independent physical resources. It prevents direct access of VMs to physical resources and shields virtual resource failures. QEMU emulates the independent and isolated virtual devices for VMs. This ensures that each VM can only access its own virtual resources and that a single VM failure does not affect the VMM or other VMs.

5.1.1.7 Resource Isolation Between VMs

HCI utilizes multiple technologies to isolate data at different layers, including VM network communication, VM running memory, and data I/O and image layers.

VM Network Communication:

The network packets sent by VMs are forwarded by a virtual switch (vSwitch), which has comprehensive data isolation and protection mechanisms.

VM Running Memory:

HCI develops its compute virtualization technology based on the QEMU-KVM architecture with Sangfor OS as the underlying OS. Sangfor OS is a community-based Linux system, and its logical architecture is as follows.

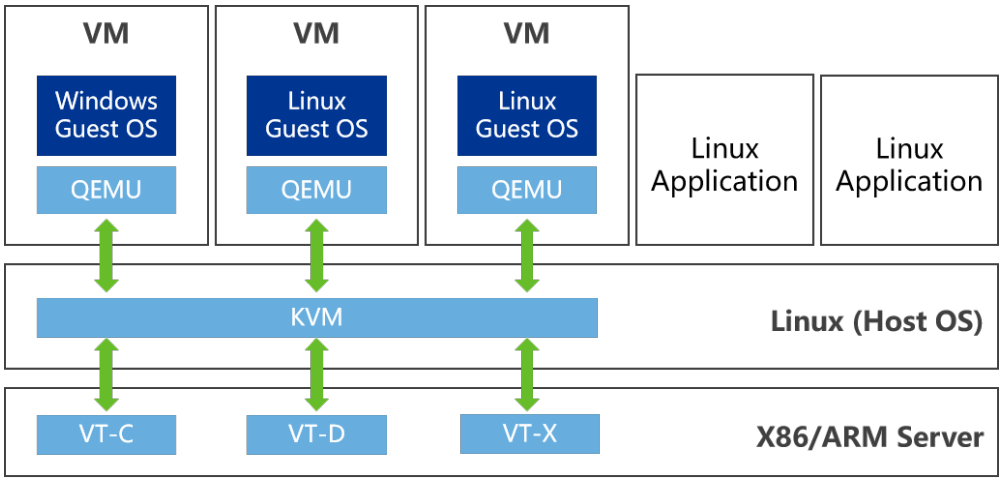


Figure 37: VM Logical Architecture for Memory Isolation

A VM is a QEMU process for the host OS and has its own independent process space as other Linux applications. Combined with the process isolation mechanism of Linux (host OS) and the memory page table mapping mechanism provided by x86 servers, specifically VT-x/EPT (which enables two-round address translations from the logical address of VM memory to the physical address of VM memory, and then to the physical address of physical server memory), this process space ensures that each VM can only access memory addresses within its own space, thus implementing space isolation for each VM.

VM I/O and Image:

Similar to the mechanism of memory isolation, this data isolation mechanism ensures that each VM process has a corresponding VM image file on the host (host OS). These images, stored in the QCOW2 format on the server disk, are independent and isolated from each other.

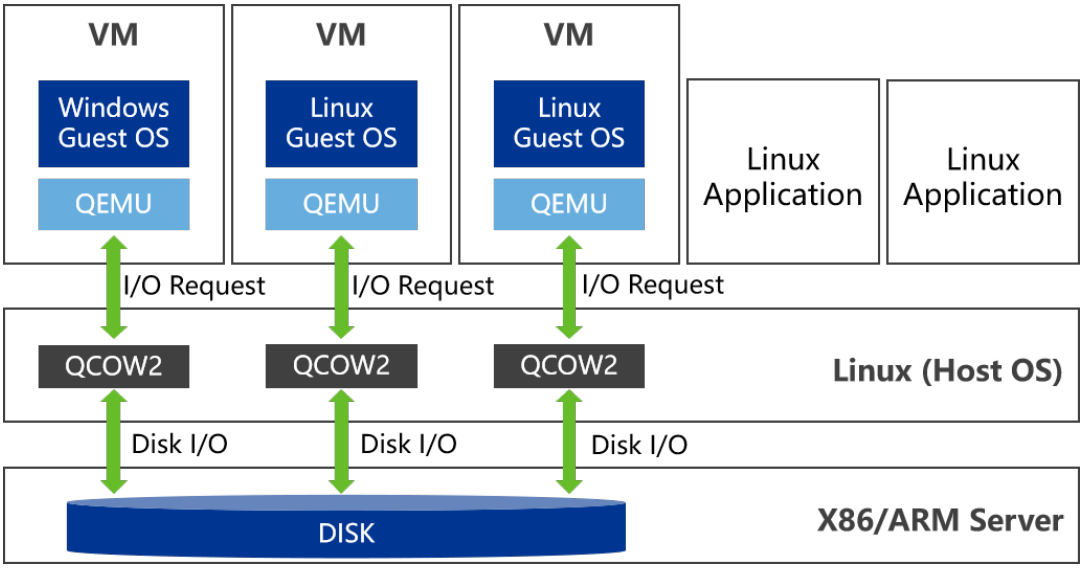


Figure 38: VM Logical Architecture for I/O and Image Isolation

5.1.1.8 Service Process Memory Isolation

The service memory isolation mechanism is designed to isolate the services in HCI, including management services, storage services, network services, and compute services, avoiding affecting each other in memory resources. For example, if the memory usage of compute services in memory overcommitment scenarios exceeds the allocated limit, such services cannot use the memory resources of other services, thus ensuring the stability of other services and HCI.

Principles:

Based on the Linux cgroups mechanism, memory isolation implements the abstraction of host resources into the following resource groups (a resource group is a collection of resources including CPU resources, memory resources, and I/O resources): HCI resource group, storage resource group, network resource group, and compute resource group. Memory resources for different services are managed through these resource groups, thereby restricting and ensuring the supply of memory resources for services accordingly.

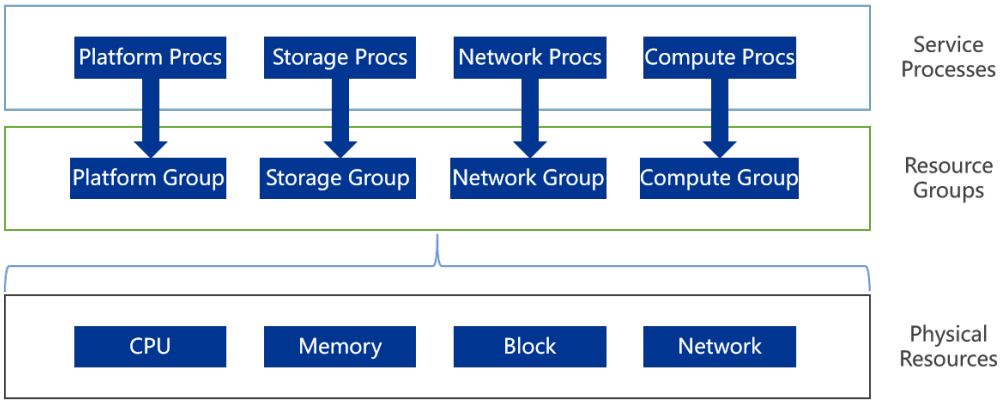


Figure 39: Service Resource Groups

Precautions:

Memory isolation supports committed-RAM overcommitment, which indicates that more VMs can be enabled without affecting the stability of HCI. However, excessive VMs will result in memory competition, deteriorated performance, and hindered business. In scenarios where VMs feature excessive swapping and frequent read/write operations, system latency may persist. In this case, some VMs need to be shut down.

NOTICE

1. Only memory isolation is supported currently.

2. For upgrades from lower versions to HCI 6.9.0 or later, memory management-related advanced functions will take effect after the host is restarted following the upgrade. Such advanced functions include critical VM memory resource guarantee and alerts on insufficient memory.

5.1.1.9 PCIe Cabling for VMs

PCIe is a high-performance bus standard widely used for connecting various components in a computer system. Compared to the PCI bus, the PCIe bus features:

1. High transfer rates: The PCIe bus supports different transfer rates, including 2.5 Gbps, 5 Gbps, 8 Gbps, 16 Gbps, and 32 Gbps. Faster than the

traditional PCI bus, it can meet high-bandwidth and low-latency data transmission requirements.

2. Scalability: The PCIe bus supports multi-device connections by increasing the bus bandwidth and number of devices through PCIe switches, meeting the requirements in various application scenarios.
3. Hot swapping: The PCIe bus supports hot swapping, and devices can be added or removed without system shutdown, which facilitates device management and maintenance.
4. Low latency: The PCIe bus adopts high-speed serial transmission technology that features low latency to enhance device responsiveness and performance.
5. High reliability: The PCIe bus employs differential signal transmission technology that features strong resistance to interference and stable and reliable transmission, thus ensuring accurate and reliable data transmission.

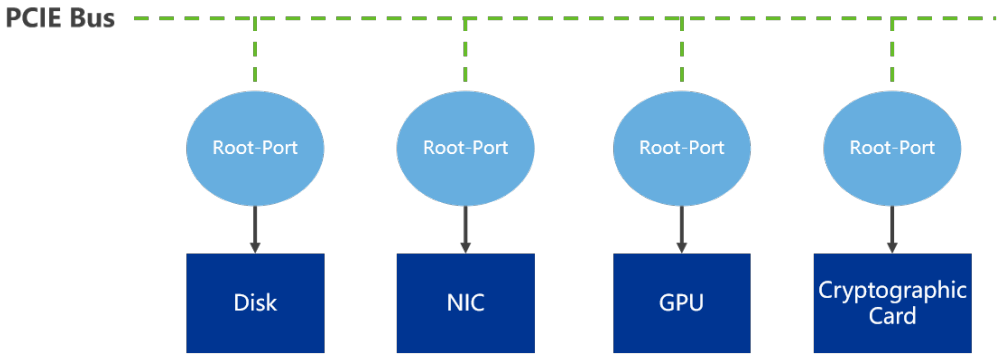


Figure 40: PCIe Bus on VMs

In HCI, virtual devices such as disks, NICs, and cryptographic cards are connected to the PCIe bus using the PCIe root port for efficient point-to-point transmission between PCIe devices. PCIe cabling makes hot swapping available to more devices. In addition, the PCIe bus meets the requirements of GPUs for data transmission and computing capabilities by supporting parallel computing of multiple GPUs for better GPU performance.

5.1.2 Functional Architecture

The functions of aSV include performance optimization, fault prevention and handling, resource management, data protection, and application deployment. The following figure shows the entire functional architecture.

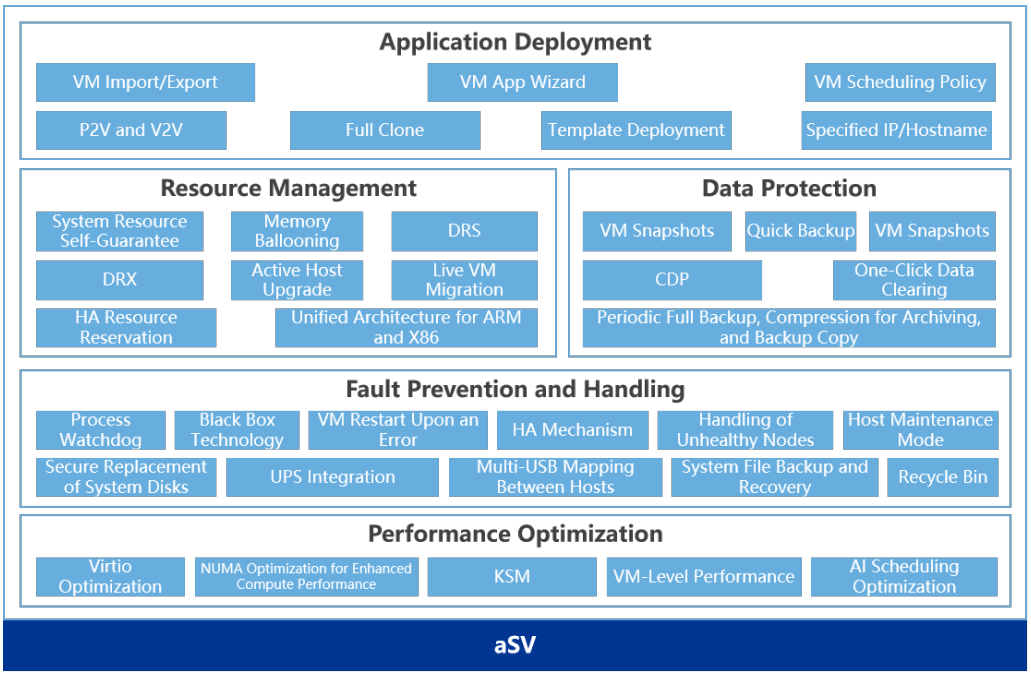


Figure 41: aSV Functional Architecture

5.1.3 Resource Management

To ensure smooth business operation, aSV uses various resource management methods to enhance business reliability and optimize resource utilization. Such methods include system resource self-guarantee, resource reservation, DRS, DRX, memory ballooning, and VM live migration.

5.1.3.1 Resource Overcommitment

Not all VMs occupy resources at all times in HCI. Instead, the resources are allocated and used dynamically, which indicates that the resources allocated to idle VMs can be occupied by busy VMs. As a result, HCI supports the overcommitment of compute resources (overcommitment indicates that the

sum of the resources allocated to VMs is greater than the resources available on the host). Be sure to note that a large number of heavily loaded VMs on a host may significantly affect the performance of VMs and even result in lag. As a result, resource overcommitment should be within an appropriate range.

- CPU Overcommitment

The recommended number of vCPUs configured should not exceed twice the number of CPU threads. (Note: The number of threads is equal to the number of physical cores. When hyper-threading is enabled on servers, the number of threads is generally twice the number of physical cores.)

- Memory Overcommitment

HCI allows users to customize the memory overcommitment ratio per physical host. It is recommended that memory overcommitment not be configured or the ratio is below 120%. However, users can make memory overcommitment up to twice the available memory, depending on their actual business situations.

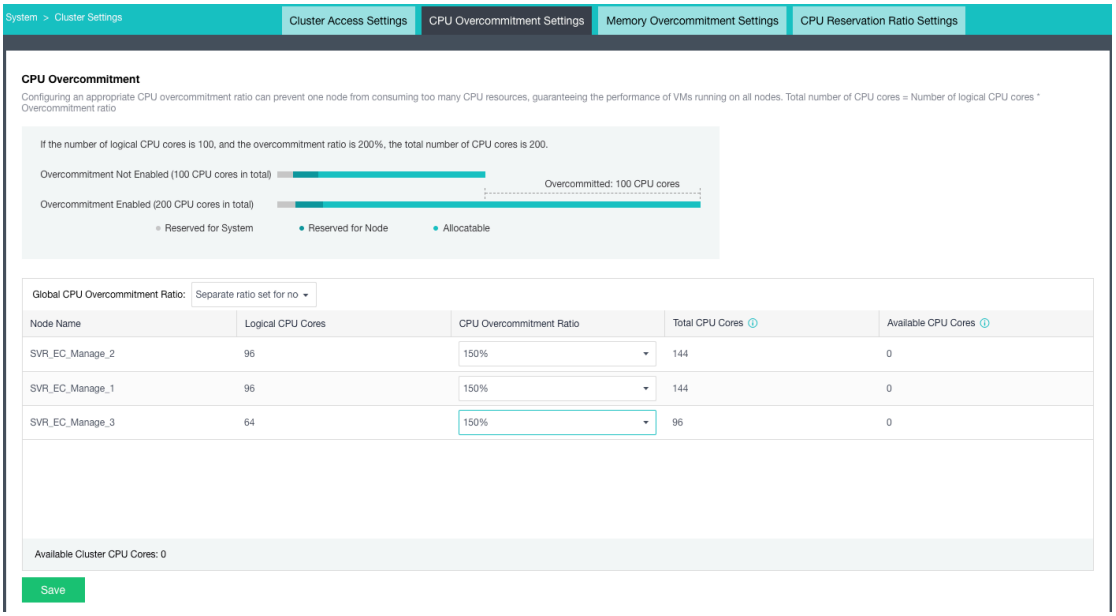


Figure 42: Memory Overcommitment Settings

Principles: Assume that 32 GB of memory is allocated to a VM, and its actual usage is only 24 GB. Theoretically, the remaining 8 GB can be reclaimed and used by other VMs. Based on this, virtual memory overcommitment is achieved

with the help of the KSM technology (when vmTools is not installed) or memory ballooning technology (when vmTools is installed).

Physical host virtual memory consists of physical memory (RAM) and swap partitions (disk partitions). In case of insufficient physical memory, the system will use the swap partitions based on the configuration (by default, HCI is configured to avoid swap usage as much as possible). HCI occupies approximately 32 GB of memory when a standard disk is configured, and the remaining physical memory of the host can be allocated to VMs.

VM memory usage consists of the QEMU process (occupying about tens of megabytes) and the VM guest OS usage (depending on the size of memory configured for VM).

Be sure to note the difference in memory management between Linux and Windows. When a Windows OS is running on a VM, the system performs the write zero operation during startup. When a Linux OS is running on a VM, the system does not set memory to zero during startup. By default, this operation triggers allocation only when it is required. (As a result, the actual memory usage of Linux is less than that of Windows by default in the initial startup phase.)



The upper limit of memory overcommitment is the sum of physical memory and swap size, which cannot be exceeded regardless of the overcommitment ratio.

5.1.3.2 System Resource Self-Guarantee

As HCI itself requires certain compute resources, it employs a system resource self-guarantee mechanism to ensure its stability and performance while bearing business operations. During system startup, HCI forcibly reserves the minimum compute and memory resources required for its operation so that VMs will not consume excessive system resources (failure to comply may result in HCI errors). Based on the HCI components enabled by users, HCI adaptively adjusts the forcibly reserved system resource.

- CPU resource allocation

The overall allocation is approximately 11-core CPU resources, where 2-core CPU resources are for the compute plane and management plane, 1-core CPU resources are for the network plane, and 8-core CPU resources are for the storage plane.

- Memory resource allocation details

8-slot configuration: a total of 27 GB memory resources, where 7 GB memory resources are for the compute plane and management plane, 7 GB memory resources are for the network plane, and 13 GB memory resources are for the storage plane.

12-slot configuration: a total of 31 GB of memory resources, where 7 GB memory resources are for the compute plane and management plane, 7 GB memory resources are for the network plane, and 17 GB memory resources are for the storage plane.

System resource self-guarantee is implemented through the resource overload protection mechanism. Too many VMs running in HCI constitute a typical resource overload scenario. The resource overload protection mechanism ensures the stable operation and management of HCI in such overload scenarios.

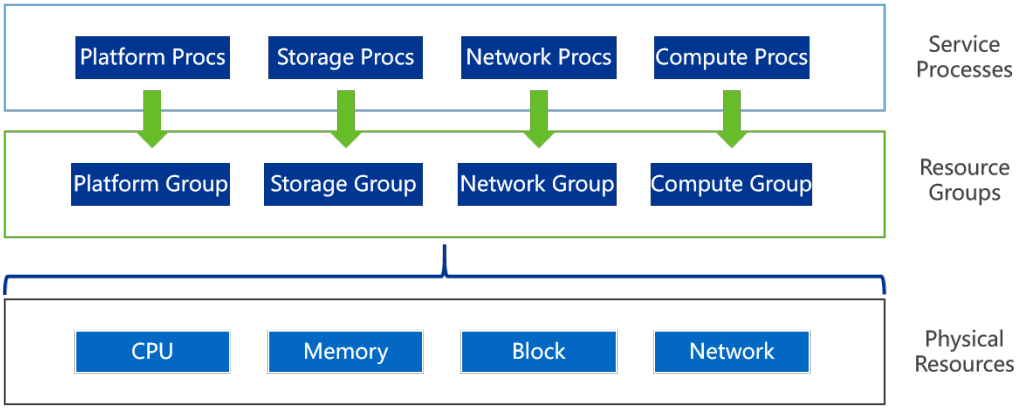


Figure 43: Resource Overload Protection Mechanism

Based on the Linux cgroups mechanism, resource overload protection implements the abstraction of the host resources into the following resource groups (a resource group is a collection of resources including CPU resources,

memory resources, and I/O resources): HCI resource group, storage resource group, network resource group, and compute resource group. Memory resources for different services are managed through these resource groups, thereby restricting and ensuring the supply of memory resources for services accordingly.

5.1.3.3 Guarantee of Resource Reservation for HA

HCI provides the resource reservation mechanism to ensure sufficient resources for executing the HA function in the event of host failures. A certain number of resources are reserved on the physical host. They are not allocated under normal circumstances. Only when certain host failure occurs, and the HA mechanism takes effect, these reserved resources can be allocated and used only when certain host failure occurs. This mechanism prevents HA failure across the entire HCI due to excessive resource utilization.

5.1.3.4 Memory Ballooning

It is necessary to deallocate idle memory from non-critical VMs when excessive memory resources of HCI are occupied by VMs. This not only increases memory resource utilization but also ensures that sufficient memory is available for the continuous and stable operation of critical businesses with sufficient protection for performance.

Generally, changing the memory allocation of a VM requires shutting down the VM, modifying its memory configuration, and restarting the VM in sequence. However, with the memory ballooning technology, HCI can adjust the physical host memory resources allocated to a VM when the VM is running without the need to shut down the VM.

With this technology, each VM is assigned a memory balloon and cannot access or use the memory within the balloon after the assignment. However, the memory can be reallocated by HCI to other processes or other VMs. A larger memory balloon indicates more memory is available. This way, HCI can deallocate and allocate memory by adjusting the size of the balloon.

HCI executes the memory deallocation policy accordingly based on the remaining system memory for a balance between memory reclamation and VM performance. When the remaining system memory is below 40%, memory deallocation is triggered, and VM memory is deallocated every hour. When the remaining system memory is below 15%, VM memory is deallocated every half an hour. When the remaining system memory is below 5%, VM memory is deallocated every 10 minutes.

Memory Deallocation Process:

When the host has insufficient available memory and VM 1 has idle memory, the host sends a memory deallocation command to VM 1, and VM 1 releases its idle memory upon receiving the command. The memory balloon of VM 1 absorbs the released memory and inflates, which indicates increased available memory for the host.

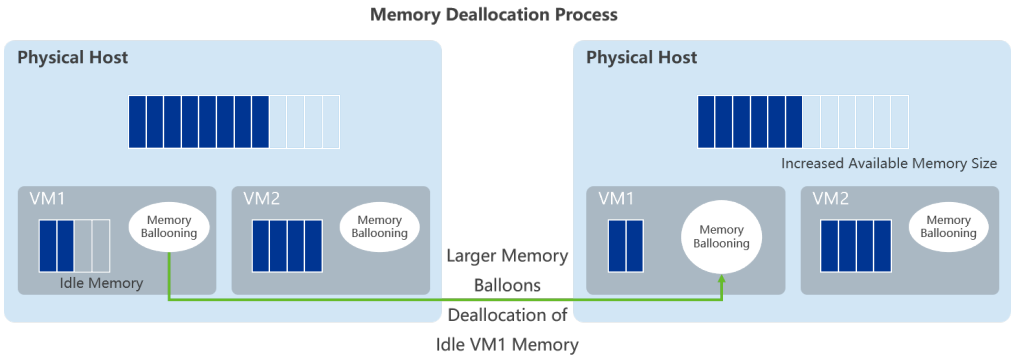


Figure 44: Memory Ballooning Diagram-Memory Deallocation

Memory Allocation Process:

When VM 2 has insufficient memory and requests memory allocation from the host, the host compresses the memory balloon of VM 2 and releases some of the memory within the balloon for VM 2 to use.

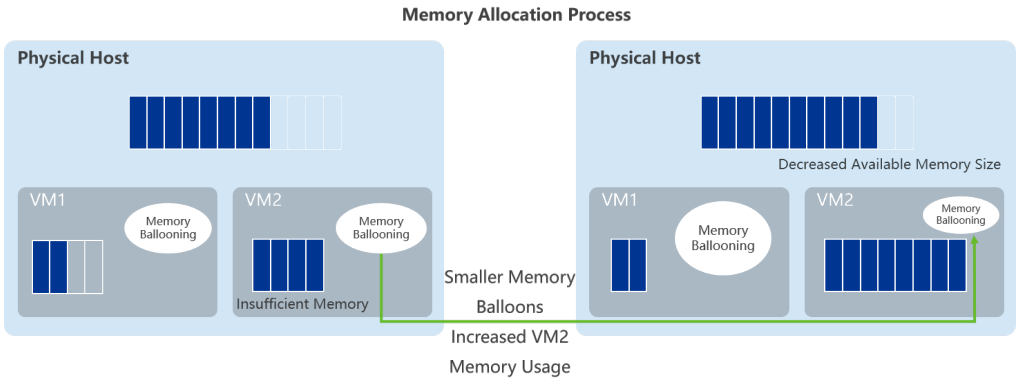


Figure 45: Memory Ballooning Diagram-Memory Release and Usage

In addition to the functions in the memory ballooning solution adopted in the open-source community, aSV has made the following optimizations:

1. New security protection mechanism:

aSV continuously checks the remaining memory of the guest system and proactively stops deallocation when the remaining memory is less than 100 MB to ensure the stable operation of the guest system.

2. New automated deallocation mechanism:

The automated deallocation mechanism eliminates the need for VMM monitoring and processing, which significantly simplifies the entire process. It accommodates large-scale deployment scenarios by resolving the issue of high scheduling pressure on the VMM and complex control policies.

3. Higher memory deallocation efficiency:

aSV optimizes the memory deallocation algorithm for a more than 30-fold increase in the deallocation efficiency.

4. Unawareness on the user side:

aSV provides users with a quality user experience by making users unaware of the change in VM memory, that is, the inflation and compression of memory balloons. This addresses users' concerns of mistaking the sudden memory increase or decrease in VM memory as system issues.

5.1.3.5 KSM

Multiple VMs running a specific business in parallel may share the same memory page because it will result in extremely high memory consumption and even waste if each VM allocates a portion of memory to store such data.

The KSM technology of HCI implements the merging of shared or identical memory pages, thereby reducing memory waste and allowing the physical host to accommodate more VMs.

KSM implements periodical scans and checks of running processes by using the kernel and comparison of their memory. If identical memory regions or pages are identified, multiple identical memory pages are merged into a single page marked as copy-on-write. This effectively reduces the overall memory usage in the system. If a process attempts to modify the merged memory page marked as copy-on-write later, a new memory page will be generated for that process.

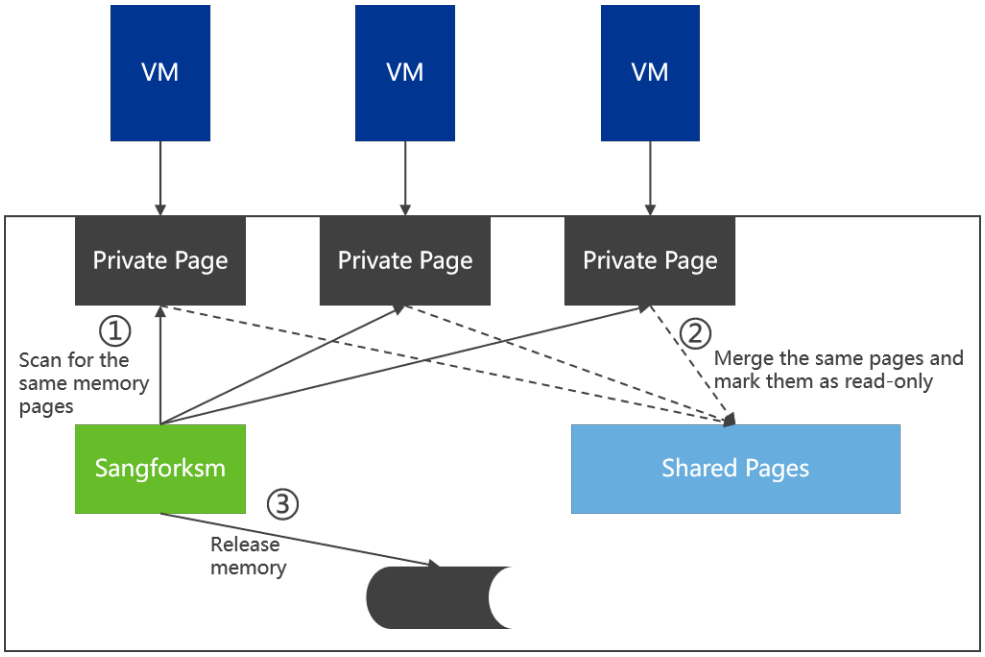


Figure 46: Technical Principles of KSM

As shown in the above figure, Sangfor KSM is the KSM processing module of HCI that periodically scans the memory pages (private pages in the figure) of VM processes. When identical memory pages for these VM processes (VMs in

the figure) are identified, it merges these pages and makes them point to a shared memory region (shared pages in the figure), which releases some memory for other VMs.

It wastes too much time in scanning and identifying the mergeable pages by using the general KSM technology. In contrast, HCI optimizes the technology by using a brand-new hash scanning algorithm for quick scanning and efficient merging of redundant data, which increases the page merging speed by more than 20 times that of native KSM. This greatly improves work efficiency and host memory usage and reduces host CPU consumption.

5.1.3.6 Memory Hot Add

Memory hot add refers to the capability to dynamically increase the memory of a running VM without interrupting its services.

5.1.3.6.1 Technical Principles

Operating System Configuration Items:

The complete hot plug capability consists of the virtualization layer and the system layer. Currently, the virtualization layer provides ACPI capability, and you can check whether the system layer supports memory hot add through the following methods:

1. Linux System:

Query the following configuration items in the /boot/config-$(uname -r) file after the VM startup:

- CONFIG_SPARSEMEM: Enables sparse memory model (the other two are flat memory and discontinuous memory).
- CONFIG_MEMORY_HOTPLUG: Enables memory hot plug.
- CONFIG_ACPI_HOTPLUG_MEMORY: Enables ACPI memory hot plug.
- CONFIG_MEMORY_HOTREMOVE: Enables memory hot removal.
- CONFIG_MIGRATION: Enables page migration.

2. Windows System:

It is not possible to check whether a Windows system supports memory hot add by querying the configuration items because its code is not open source. To check whether a Windows system supports memory hot add, you need to do a compatibility test on it. The list of Windows system versions that support memory hot add is updated with each release of Sangfor Cloud.

Server Configuration Items:

ACPI is a memory hot plug controller used by both x86 and ARM platforms. The /boot/config-$(uname -r) file of a VM that supports memory hot add must contain the following configuration items:

- CONFIG_SPARSEMEM
- CONFIG_MEMORY_HOTPLUG
- CONFIG_ACPI_HOTPLUG_MEMORY

Overall Workflow:

The following illustrates the overall workflow of memory hot add from user operation to internal processing of the VM system on HCI:

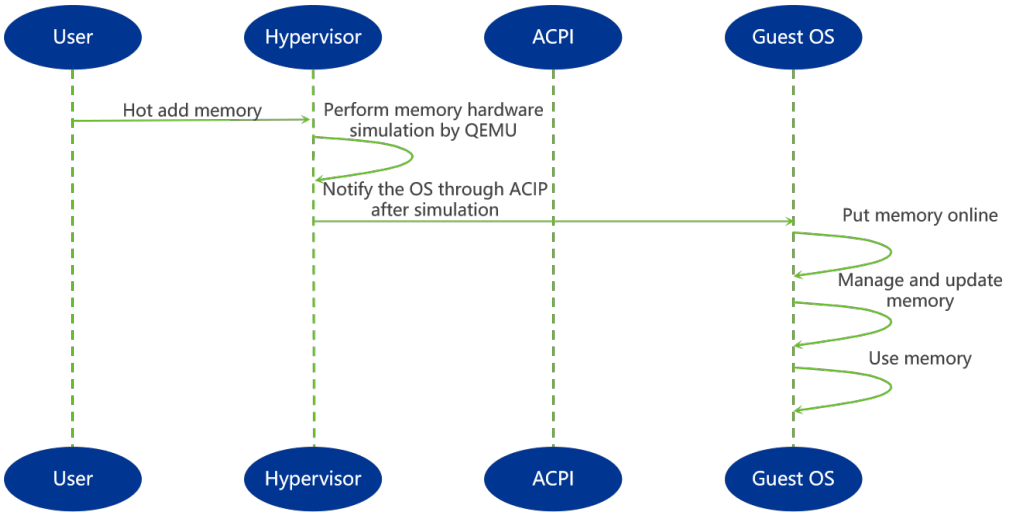


Figure 47: Memory Hot Add Process

1. The user sends a memory hot add command to the Hypervisor layer.
2. The Hypervisor layer receives the command, performs memory hardware simulation, and notifies the OS through the Advanced Configuration and Power Interface (ACPI).

3. The Guest OS receives the event notification from ACPI and puts the memory online for subsequent management, updates, and use.

- QEMU Workflow

The primary task of VM (Guest OS) memory simulation is to convert the virtual address of the VM (GVA) to the physical address (HPA) of the host on which the VM is running. With the help of a virtual hardware environment, the conversion from GVA to the physical address of the VM (GPA) can be completed by the Guest OS, and QEMU does the transition from GPA to HPA. QEMU will create a DIMM device, and send the corresponding ACPI interrupt notification to the Guest OS to write the information that has been negotiated between QEMU and the VM to a certain segment of the memory through the I/O port, allowing the Guest OS to detect the memory and obtain the information for subsequent processes.

- Guest OS Workflow

Put memory online after Guest OS startup:

During the system startup, the memory hot plug subsystem registers the event handler (a callback function for event notification) to the ACPI driver, which reads the ACPI tables saved in the BIOS and obtains the range of I/O ports corresponding to the memory and the pending contents of the corresponding events. Through the memory device driver, the work such as device initialization and putting online is gradually called and completed to finally power on the hardware.

Manage and update memory before use:

1. Initialize direct mapping page tables.
2. Adjust the entire physical memory space for direct mapping by changing the end identified by direct mapping to end+offset (range of newly added memory size) so that the kernel can access all physical memory.
3. Allocate virtual memory mapping pages.
4. Initialize virtual memory mapping page tables.
5. Request some physical memory in physical space for the address space of the newly added structured page tables and adjust the virtual mapping

space, which is an array of the memory page information identified by the kernel (structured page tables storing physical pages) for memory management. Then update the memory mapping page tables saved in the virtual mapping space based on the newly added physical space (by adding new physical address information).

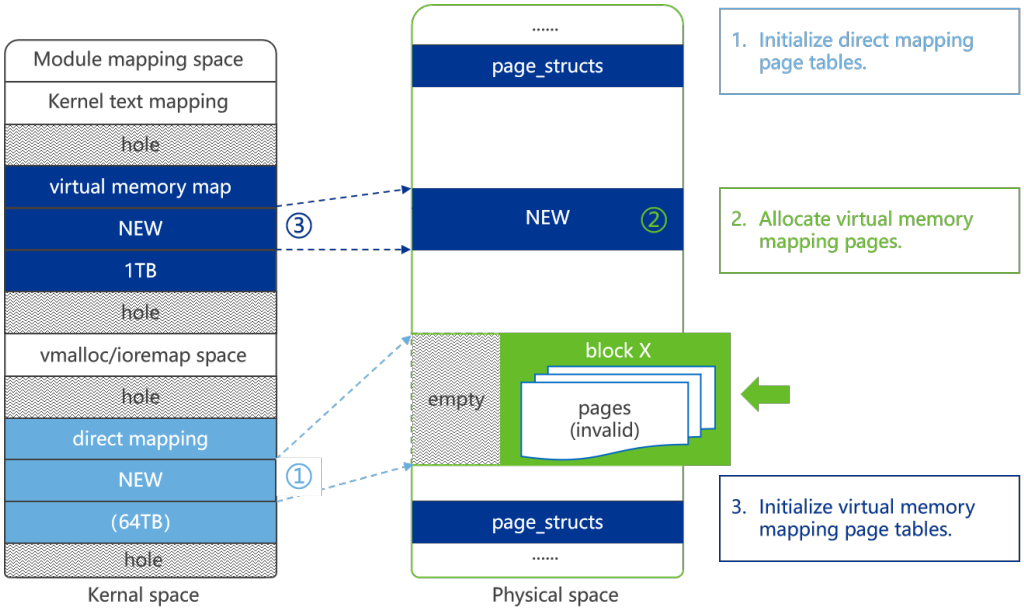


Figure 48: Memory Hot Add Procedure

5.1.3.6.2 UI Display

Log in to HCI, go to **Compute** > **Sangfor Technologies Inc. HCI**, click **More** of a **VM**, and then go to **Edit** > **Configuration** > **Processor** > **Advanced** to select **Enable CPU hot add**, and go to **Memory** to select **Enable memory hot add**.

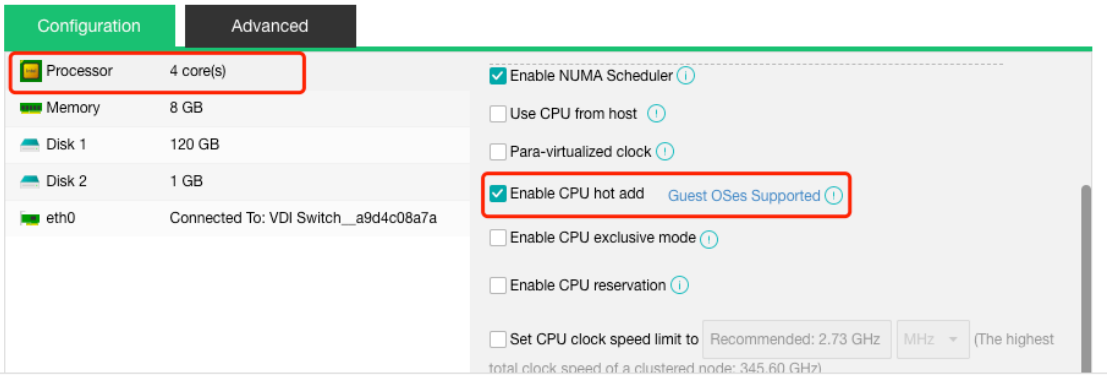


Figure 49: Enable CPU Hot Add

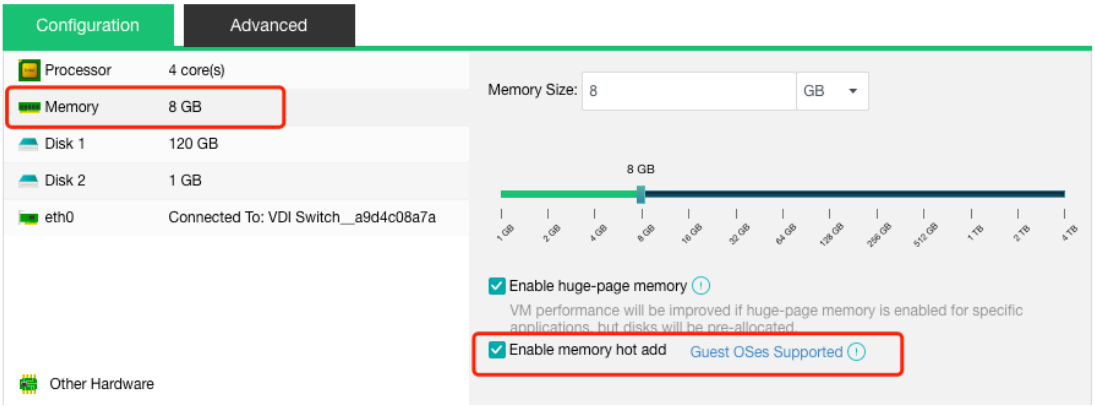


Figure 50: Enable Memory Hot

Precautions:

1. It is recommended to hot add memory when the VM memory usage is 80% rather than extremely high. The DRX provided by HCI will automatically add memory when conditions are met.
2. When the VM memory is highly pressured and severely fragmented, you can regularize the moveable free memory through echo 1 > /proc/sys/vm/compact_memory to increase the amount of 2M contiguous memory in /proc/buddyinfo.
3. When the VM memory is highly pressured and severely fragmented, you can reclaim memory by clearing page caches, directory entries, and inodes. These operations have certain risks, for example, applications that have used the caches of the file system may trigger a cache flush after the cache reclamation, resulting in a degradation of I/O performance (recoverable).

5.1.3.7 DRS

Cluster Dynamic Resource Scheduling (DRS) refers to calculating the optimal migration paths for VMs based on different scheduling modes and ensuring the cluster is in the optimal status.

5.1.3.7.1 Technical Principles

There are two scheduling modes: **Better Performance** and **Less Cost**. The

former balances the load on each node, and the latter saves node resources by centralizing VMs to run on one or more nodes as much as possible.

- Better Performance Mode

When the resource usage of the nodes in a cluster is unbalanced, DRS will migrate VMs on nodes with high loads to nodes with low loads. If the performance of the VMs is not improved after the migration to the destination node, no VM scheduling recommendations will be given.

- Less Cost Mode

If enabled, DRS will centralize the VMs on nodes with a CPU or memory usage lower than 80% to run on one or more nodes, reducing the number of nodes with high loads in the cluster.

Application Scenarios:

	Better Performance	**Less Cost**
Application Scenario	When the load on each node in a cluster is unbalanced, DRS will migrate VMs on nodes with high loads to nodes with low loads.	When there are at least two nodes with a CPU or memory usage lower than 80% in a cluster, DRS will centralize the VMs on these nodes to run on one or more nodes as much as possible.
Action	Migrates VMs on nodes that have reached the scheduling sensitivity threshold to nodes with low loads calculated based on resource forecasting.	Migrates VMs in order of node memory load from the lowest to the highest to nodes with high loads.

Table 1: DRS Scenarios

Triggering Thresholds:

1. Better Performance Mode

- **Conservative**: Triggered when the load (CPU or memory usage) of a node exceeds 70% and the load difference between each node exceeds 30% in a

cluster. Scheduling recommendations will be given and VMs on nodes with high loads will be migrated to nodes with low loads.

- **Moderate**: Triggered when the load (CPU or memory usage) of a node exceeds 60% and the load difference between each node exceeds 20% in a cluster. Scheduling recommendations will be given and VMs on nodes with high loads will be migrated to nodes with low loads.
- **Aggressive**: Triggered when the load (CPU or memory usage) of a node exceeds 50% and the load difference between each node exceeds 5% in a cluster. Scheduling recommendations will be given and VMs on nodes with high loads will be migrated to nodes with low loads.

2. Less Cost Mode

- **Conservative**: Triggered when there are at least two nodes with a CPU or memory usage lower than 80% in a cluster. VMs on nodes with low loads will be migrated to nodes with high loads, with a performance score decrease of up to 10%.
- **Moderate**: Triggered when there are at least two nodes with a CPU or memory usage lower than 80% in a cluster. VMs on nodes with low loads will be migrated to nodes with high loads, with a performance score decrease of up to 30%.
- **Aggressive**: Triggered when there are at least two nodes with a CPU or memory usage lower than 80% in a cluster. VMs on nodes with low loads will be migrated to nodes with high loads, with a performance score decrease of up to 75%.

5.1.3.7.2 Forecasting Algorithm and Scheduling Recommendations

Forecasting Algorithm:

The DRS forecasting algorithm can identify the suitable node for a VM to migrate to by calculating whether the load on the destination node will exceed the threshold within one hour after the migration through the trend forecasting of the data characterized by periodic changes or growth. As shown in the following chart, the black line refers to the actual trend while the blue

line illustrates the forecasting trend.

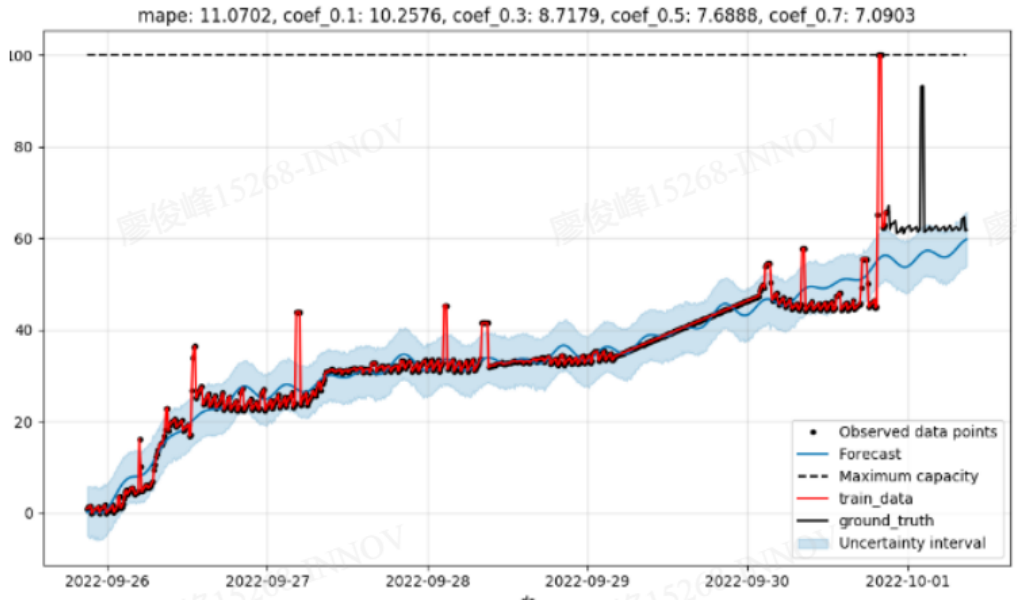


Figure 51: DRS Forecasting Trend

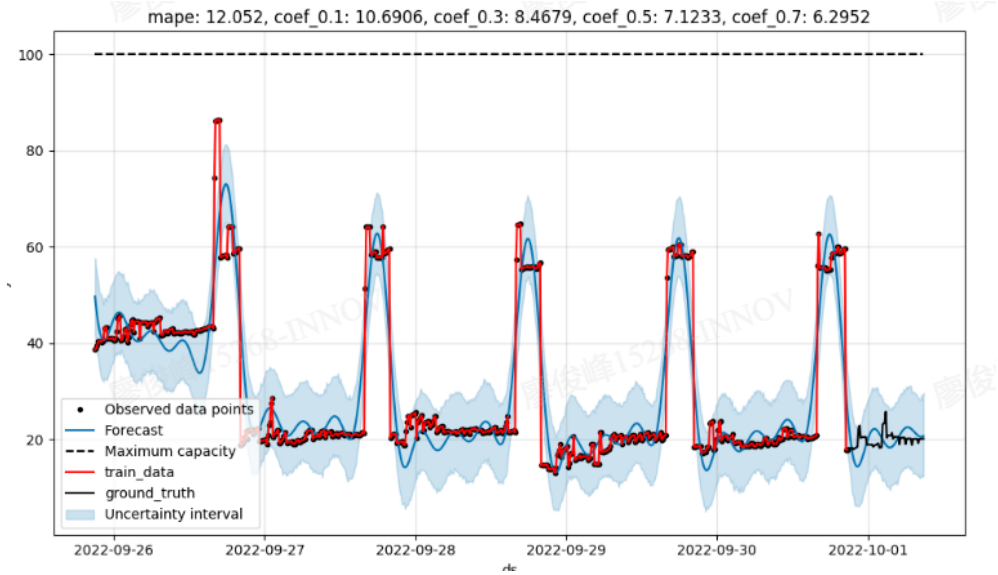


Figure 52: DRS Forecasting Trend

Difference-Based Linear Regression Scheme:

The DRS forecasting algorithm trains the model based on the load trend over the last five days (with 10-minute data point intervals) to predict the load trend for the next hour.

Self-Checking Mechanism:

The forecast engine evaluates the forecasting algorithm based on the mean absolute percentage error (MAPE) of the forecast values over the last 12 hours, which is calculated as follows:

$$\mathrm{MAPE}(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\|y_i-\hat{y}_i\|}{\|y_i\|}$$

yi and y^i respectively represent the forecast value and the actual value at time i, and n represents the total time.

Considering that the closer the forecast value is to the current time the more important its accuracy is, a time-weighted correction is introduced. When calculating MAPE, the closer the data is to the current time, the more the forecast error affects the result, which is decayed as follows:

$$w(t) = \exp(-\lambda(T-t))$$

In this exponential decay function, t means the timestamp of the data point, T means the current timestamp, and λ means a decay rate parameter. When t is equal to T, the value of the weight function is 1, indicating that the data at the current time has the highest weight; the longer t is passed from T, the smaller w(t) is, and the lower the weight of the data is. You can set λ to a positive number to ensure that the value of this function is between 0 and 1 or adjust it based on the application scenario for the best prediction.

DRS identifies node loads based on the forecast values with the MAPE lower than 10%.

Scheduling Recommendations:

Better Performance	Less Cost
Migrates VMs on nodes from high loads to nodes with low loads to improve the VM performance. The reliability level of the destination nodes cannot be lower than that of the source nodes.	Migrates VMs in order of node memory load from the lowest to the highest to nodes with high loads. The VM performance score decreases after the migration cannot exceed the threshold and the reliability level of the destination nodes cannot be lower than that of the source nodes.

Table 2: Scheduling Recommendations

5.1.3.7.3 Node and VM Scores

DRS displays the destination nodes to which VMs are scheduled based on resource scores. The Better Performance mode displays the VM performance score improvement and potential reliability improvement after migration, while the Less Cost mode displays the number of VMs on the source and destination nodes to reflect the business density increase on the destination node after migration and the potential VM reliability score improvement.

- **Node Performance Score**: calculated based on the CPU and memory loads, and network and storage performance of the node. Network performance depends on packet loss and error rates, and storage performance depends on whether the usage of all the storage devices on the node is lower than 70%, and whether there is a VM storage copy on the node.
- **Node Reliability Score**: calculated based on the severity of memory, disk, and RAID card failures on the node and displays deductions.
- **VM Performance Score**: calculated based on the CPU and memory loads, and network and storage performance of the VM. Network performance depends on packet loss and error rates, and storage performance depends on whether there is a storage copy of the VM on the node where it resides.
- **VM Reliability Score**: consistent with that of the node on which the VM resides.

Calculation Parameters:

1. Performance Score

Calculation Interval: 10 minutes.

Resource	Parameter
CPU	Actual usage
	Expected usage
	Number of logical cores

Memory	Actual usage
	Used swap partition size
Network	Packet loss rate
	Packet error rate
Storage	Whether there is a local VM copy on the node
	Whether the usage of all the disks is lower than 70%

Table 3: Performance Score Parameters

2. Reliability Score

The DRS rates the reliability scores of nodes based on their most severe failures in **Level 1** between 91-100, **Level 2** between 71-90, **Level 3** between 31-70, and **Level 4** between 0-30, and then deducts scores for other failures. The fewer the failures and the lower the level, the higher the score is. The highest reliability score is the highest score for the corresponding level (e.g., the highest score for **Level 3** is 70).

Calculation Interval: 30 minutes.

Resource	**Failure**	**Sub-Failure**	**Level**	**Deduction**	**Stateful/Stateless**	**Shut Down**	**Analysis Method**
RAID Card	RAID Card Reset	Error Code: 0x0501	1	10	Stateful	Y	Log
		Error Code: 0x2100	1	10	Stateful	Y	
		Error Code: 0x26b	1	10	Stateful	Y	
		Error Code: ../../raid/1078int.c	1	10	Stateful	Y	

		Error Code: ../../raid/utils.c	1	10	Stateful	Y	
		Error Code: 0x0d03	3	10	Stateful	Observe	
		Error Code: ./../dm/src/dm.c	3	10	Stateful	Observe	
		Other	3	10	Stateful	Observe	
	System Disk Hanging Risk		1	10	Stateless	Y/N	
	Disk Reset Failure Alert	3 disks and more	1	10	Stateful	N	
		Less than 3 disks	3	5	Stateful	N	
	Disk Reset Alert	3 disks and more	1	10	Stateful	Y	
		Less than 3 disks	3	5	Stateful	Y	
	Disk Physical Bad Sector Alert	3 disks and more	2	10	Stateful	Y	
		Less than 3 disks	3	5	Stateful	Y	
	RAID Volume Lower Disk Read-Only	Error Code: 0x32010081	1	10	Stateful	Y	
	Physical Bad Sector on RAID	Error Code: 0x31080000	3	5	Stateful	Y	

	Volume Lower Disk						
	Possible RAID Backplane Failure	Error Code: 0x30050000	2	5	Stateful	Y	
	Possible RAID Backplane and SAS Cable Failure Alert	3 disks and more	1	10	Stateful	Y	
		Less than 3 disks	3	5	Stateful	Y	
Disk	Disk Bad Sector Alert	3 disks and more (including DHA multi-disks)	2	10	Stateless	N	Smart ctl
		Less than 3 disks	3	5	Stateless	N	
	Unrecoverable Sector Exceeding Threshold	3 disks and more (including DHA multi-disks)	2	10	Stateless	N	
		Less than 3 disks	3	5	Stateless	N	
	SSD Lifetime Warning	3 disks and more (including DHA multi-disks)	2	10	Stateless	N	
		Less than 3 disks	3	10	Stateless	N	
	Insufficient SSD Space	3 disks and more (including DHA multi-	2	10	Stateless	N	

		disks)					
		Less than 3 disks	3	10	Stateless	N	
	SSD Power-On Time Too Long		3	5	Stateless	N	
	ATA Error		3	3	Stateless	N	
	Non_Medium Error		3	3	Stateless	N	
	SMART Self-Check Failure	3 disks and more (including DHA multi-disks)	2	10	Stateless	N	
		Less than 3 disks	3	10	Stateless	N	
	SMART_page_log_error		3	3	Stateless	N	
	Disk Temperature Too High		3	2	Stateless	N	
	Disk Failure Warning		3	5	Stateful (requires statistics on forecast results over a period in the past)	N	
Memory	Memory Failure Warning		1	20	Stateful	Y	Log

	Alert of Many Correctable Memory Errors		1	10	Stateful	Y	
	Alert of a Few Correctable Memory Errors		2	10	Stateful	Y	
	Correctable Memory Error Storm		1	15	Stateful	Y	
	Uncorrectable Memory Error		1	20	Stateful	Y	
	Memory Underclocking	Memory frequency is mixed	4	5	Stateless	Y	

Table 4: Reliability Score Deductions

5.1.3.7.4 UI Display

1. Log in to HCI and go to **Reliability** > **DRS** > **Scheduling Policy**.
2. Select **Automated**, **Manual**, or **Disable** for **Automation Level**.
3. Select a **Sensitivity** that corresponds to the node load threshold.
4. Select **Disable**, **Low**, **Medium**, **High**, or **Highest** for **AI Prediction Capability** based on the number of VMs. If this feature is enabled, AI algorithms will be used.
5. Configure the **Schedule** to prevent DRS from being performed during off-peak hours.

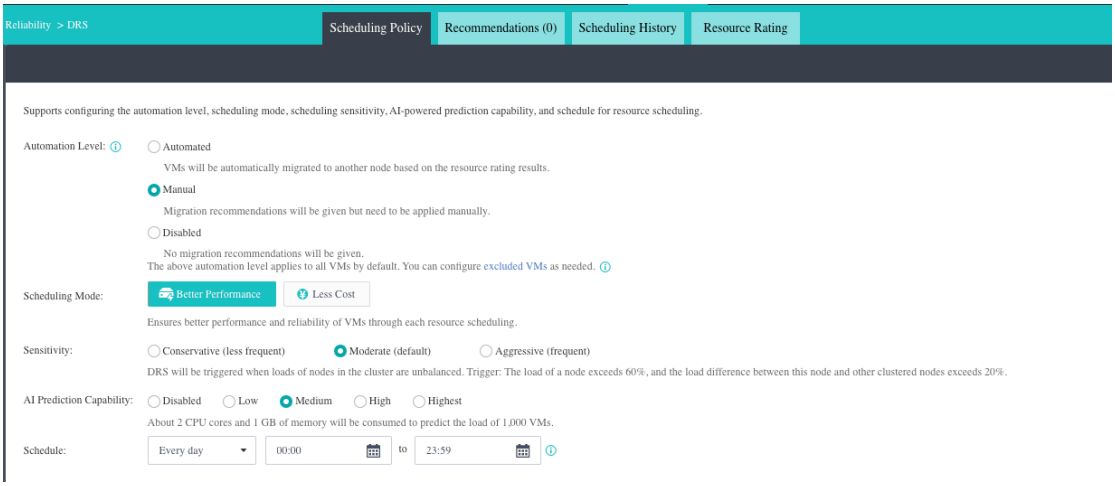


Figure 53: Select Automation Level

6. Configure **Automation Level** (**Automated**, **Manual**, **Disabled**) for VMs in special scenarios.

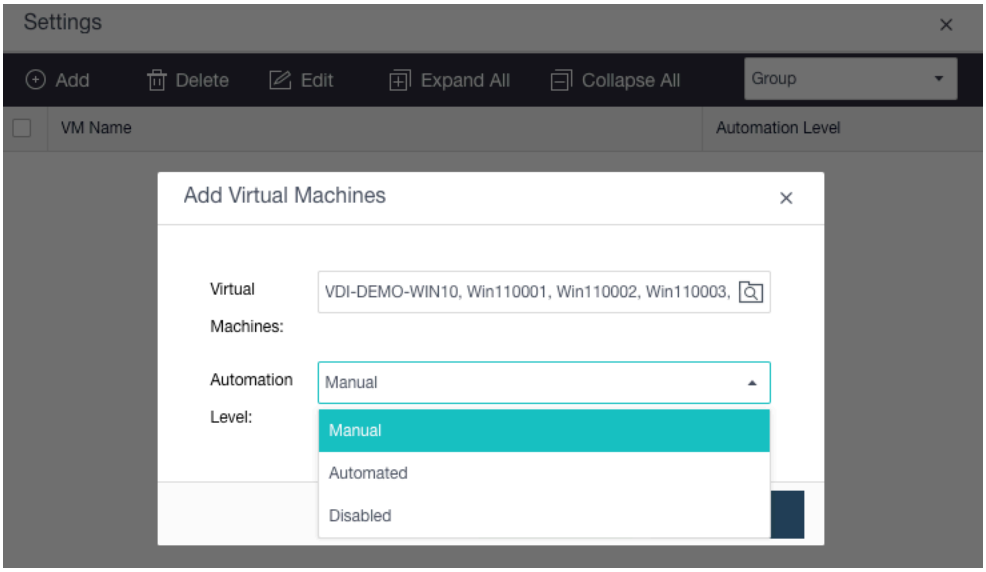


Figure 54: Select Automation Level

7. Click the **Recommendations** tab to view the scheduling recommendations from the scheduling engine based on the resource load score.
8. Click the **Scheduling History** tab to view the scheduling records.

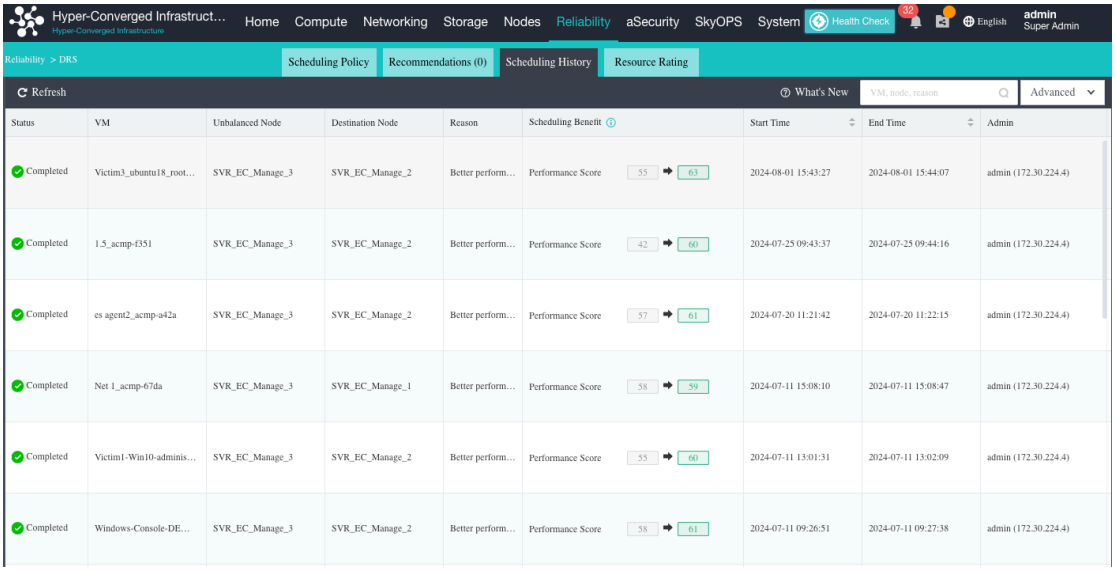


Figure 55: View Scheduling Records

9. Click the **Resource Rating** tab to view the reliability and performance scores of nodes and VMs, as well as the rating criteria.

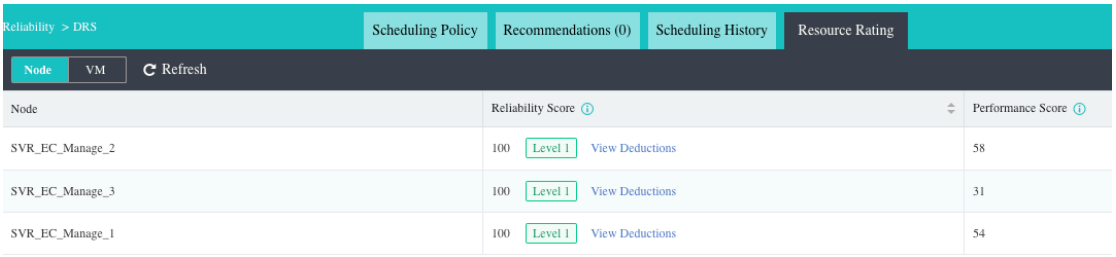


Figure 56: View Reliability and Performance Scores

5.1.3.8 DRX

For the VMs short of resources due to insufficient allocation of resources (vCPUs and vMems) in the early phase or growing business workload, aSV uses the DRX function to automatically expand the resources of VMs for continuous business operation.

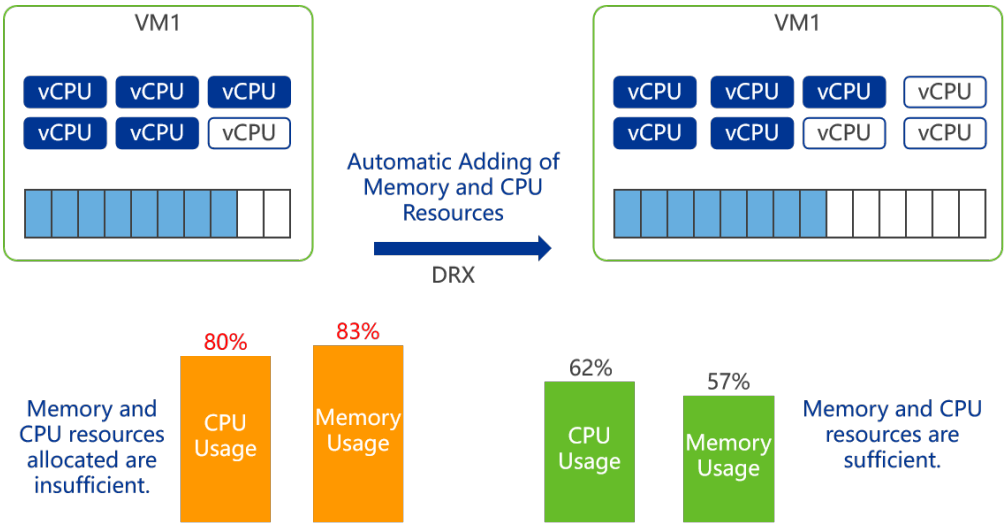


Figure 57: DRX Diagram

Principles: The system monitors the consumption of compute resources, such as CPU resources and memory resources, on business VMs in real time and verifies the remaining resources on the business host when the resource consumption reaches a custom threshold. If the host where VMs are located has sufficient remaining resources, the system adds virtual resources (vMems and vCPUs) to the business VMs without interrupting their business operation. However, if the host has insufficient remaining resources, no resource scheduling operation is performed. This can avoid affecting other businesses on that host.

5.1.3.9 VM Live Migration

VM live migration indicates migrating a VM from one physical server to another with minimal interruption to normal business operations. During device maintenance, applications can be migrated to another server through live migration, and they can be migrated back after the maintenance is completed without interrupting the applications, which reduces the planned downtime.

Generally, two traditional live migration scenarios are available. The first scenario involves a conservative throttling policy that leads to prolonged CPU throttling. In this case, the migration of dirty memory data cannot be completed in reloading scenarios, resulting in a prolonged unsuccessful

migration. The other scenario involves an extremely aggressive throttling policy that halts I/O operations directly. This approach ensures a successful migration but interrupts business operations throughout the migration process. Even worse, excessive dirty memory data may impede the migration.

To address these issues, Sangfor HCI employs a progressively enhanced CPU throttling policy, along with Sangfor4 high-efficiency compression algorithm developed by Sangfor and the technology to reduce the granularity of dirty data blocks, which achieves 100% migration with uninterrupted business operations in a reloading scenario with 1,000 concurrent Oracle users. The entire migration process only experiences performance deterioration of less than 3 seconds.

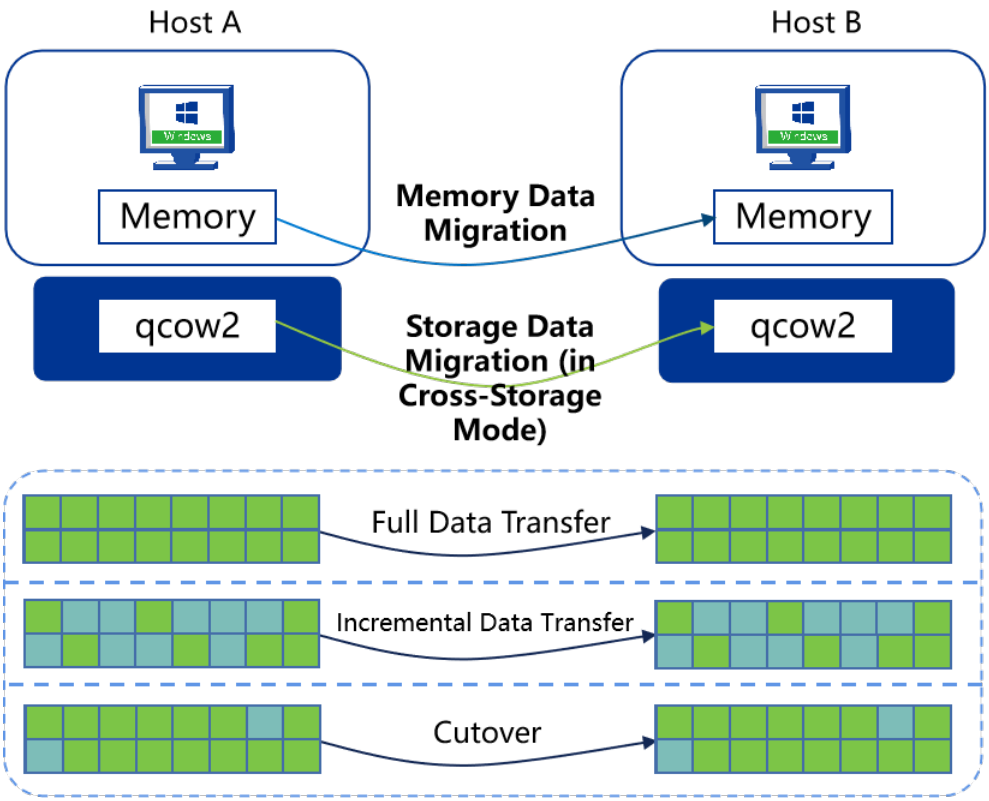


Figure 58: VM Live Migration

The live migration process involves the migration of the following three types of runtime data: VM memory data, VM disk data (optional and required for cross-storage live migration), and VM hardware status data (CPU, NIC, and GPU data).

Before migration, HCI establishes a TCP connection between the source VM and the destination VM on the destination host to transmit the memory, disk, and hardware status data on the source VM.

The data transfer during migration has the following three stages: full data transfer, incremental data transfer, and switching after shutdown. After multiple rounds of iterations, the data difference between the source end and the destination end diminishes. When the data to be transmitted reaches the threshold, the source VM is suspended to migrate the remaining data to the destination VM at a time.

The main technical points of VM live migration are as follows:

1. Sangfor4 high-efficiency compression algorithm developed by Sangfor

Unlike the inefficient native QEMU live migration compression algorithm, Sangfor4 high-efficiency compression algorithm improves the data migration efficiency by more than seven times while reducing CPU consumption by 80%.

2. Reduced granularity of data blocks for recording dirty data

The total data volume in live migration involves transmitting dirty memory pages and dirty disk pages. However, the amplification of dirty disk data may result in performance issues and hamper the entire live migration process. By reducing the granularity of data blocks for recording dirty data, this issue is addressed, and the amplification rate of dirty disk data is reduced by over 90%.

3. Optimized CPU throttling policy

The native QEMU CPU throttling algorithm has a long-term impact on business performance. If the generation rate of dirty memory data exceeds the migration rate in the incremental iterative migration stage, the migration task may not be completed. By introducing direct throttling and staged throttling during the live migration, the CPU throttling policy is optimized, and the downtime is significantly reduced to within 2 seconds.

4. Delayed migration of the frequently dirtied memory segments

Too frequently dirtied memory segments pose great challenges to the live migration process. In response to this issue, Sangfor HCI prioritizes the migration of the infrequently dirtied memory segments and delays the

migration of those with the largest number of dirty pages, which reduces the overall memory migration volume, improves the network bandwidth utilization, and optimizes the live migration performance.

5. VM hardware version management

Because of parameter optimization on the simulated hardware devices of VMs in each HCI version, their original features cannot be retained when active upgrade or live migration is performed from an earlier HCI version to a later one. Furthermore, it is difficult to modify device data or add new devices to a VM when it is running. To ensure compatibility during active upgrades and live migration to a later version for each version, HCI 6.9.0 introduces an independent VM hardware version management function, which supports compatibility with earlier HCI versions and live migration of VMs between versions while ensuring their stable operations. Boasting compatibility with multiple hardware versions, HCI 6.9.0 effectively addresses the difficulties in performing active upgrades on earlier HCI versions without interrupting VM business operations. Specifically, the VMs for an earlier HCI version are migrated to an HCI 6.9.0 cluster to ensure uninterrupted business operations, and then the original earlier-version HCI cluster is upgraded, which is followed by migrating the VMs back to the original cluster.

6. Operation in full-function mode

The VM, after live migration to HCI 6.9.0, runs in compatibility mode (that is, the later QEMU version runs the VM in command mode of the earlier version). This ensures the stable operation of the VM and the consistency in functionality and functions with the version before migration. However, new functions of HCI 6.9.0 are restricted on the VM. To allow the VM to run in full-function mode, click **Upgrade VM Compatibility** on the VM details page and restart the VM.

5.1.3.10 Void Detection

In a file system, void indicates unused space in a file. It is allocated space but does not contain any actual data. The presence of a void in a file increases the fragmentation of the file system, leading to deteriorated performance. Fiemap

is an API implementation that is used to obtain the void bitmap for the Linux file system layer. The fiemap API helps the system understand the physical locations of files and how the file system allocates these files to blocks on the disk. Fiemap can be used to optimize the file system performance and can help businesses during disk copy and backup operations that require reading the entire disk without reading any void. This will not result in extremely time-consuming execution due to void reading.

Virtual storage experiences sluggish performance when reading void, but with the fiemap API (which is supported by the file system), it no longer needs to read data when performing void detection but directly does not process the void.

Sangfor HCI makes an underlying adaptation for the fiemap API to manage and monitor the distribution of valid data on the disk, including the physical block numbers, logical block numbers, block sizes, and other information about the files. This adaptation enhances the efficiency during local backup, cross-storage migration, and cloned data merging by reducing the impact of void files on these tasks. The following describes the basic mechanism in each phase:

- During data backup, the fiemap API is called to obtain and cache the data distribution bitmap of QCOW2 images before full backup, with which void locations can be filtered out during the full backup. After the full backup is completed, if new dirty data is written into QCOW2 images, the data distribution bitmap is updated continuously to ensure that the void is skipped in each round of subsequent incremental backup.
- During cross-storage migration, the data distribution bitmap is obtained for filtering out the void locations before the first-time migration, which is similar to the backup process. The data distribution bitmap is recorded continuously, and only the valid dirty data is migrated.
- During cloned data merging, the data distribution bitmap of QCOW2 images is obtained for filtering out the void locations, which accelerates the data merging.

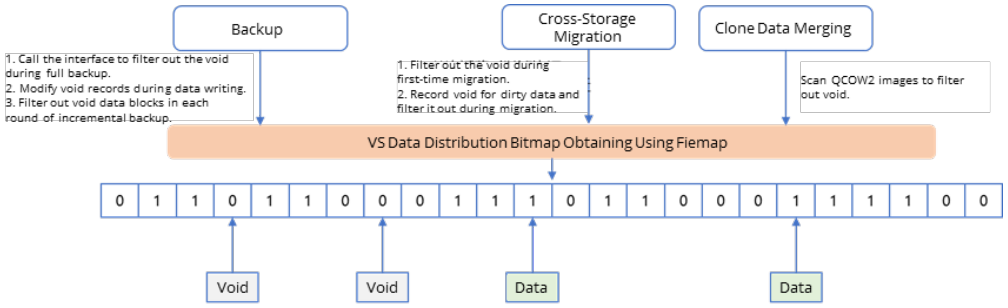


Figure 59: Void Detection Diagram

Dynamic disk allocation comes with a large amount of void and, therefore, low read/write efficiency in scenarios where reading the entire disk is required, such as VM migration and cloning. However, the virtual storage of HCI 6.8.0 or earlier uses the native NFSv3 protocol, which does not support the fiemap API for void detection. As a result, reading disk void is extremely time-consuming in these HCI versions. To address this issue, HCI 6.9.0 introduces the hole detection function using NFSv3 to optimize its performance in reading void. It significantly shortens the time required for disk read/write operations. The following table describes the test data on read/write time in the same configuration environment.

Test Data in the x86 Environment							
Business Type		Dynamic Disk Size (GB)	Actual Data Size (GB)	Void Proportion (%)	Duration for HCI 6.8.0 (s)	Duration for HCI 6.9.0 (s)	Time Consumption Optimization Margin
VM Migration	Cross-Host and Cross-Storage Live Migration for VS Ordinary VMs	150	15	90%	231	165	28.57%
		150	45	70%	474	364	23.21%
		150	75	50%	650	591	9.08%
		150	105	30%	796	792	0.50%
	Co-Host and Cross-Storage Live Migration for VS Ordinary VMs	150	15	90%	178	61	65.73%
	Cross-Host and Cross-Storage Live Migration for VS Deployed VMs	150	15	90%	2261	208	90.80%
	Co-Host and Cross-Storage Live Migration for VS Deployed VMs	150	15	90%	1458	200	86.28%
	Cross-Storage Cold Migration for VS Ordinary VMs	150	15	90%	1319	123	90.67%
	Cross-Storage Cold Migration for VS Deployed VMs	150	15	90%	2458	159	93.53%
	Cross-Storage Cold Migration for VS VM Templates	150	15	90%	1321	118	91.07%
VM Backup	Full Backup for VS Ordinary Started VMs	150	15	90%	471	196	58.39%
		150	45	70%	774	712	8.01%
		150	75	50%	874	710	18.76%
		150	105	30%	1214	1113	8.32%
	Full Backup for VS Ordinary Shut down VMs	150	15	90%	961	487	49.32%
	Full Backup for VS Deployed VMs	150	15	90%	2762	2055	25.60%
	Full Backup for VS Deployed VMs	150	15	90%	1068	450	57.87%
	Backup Copy for VS Deployed VMs	150	15	90%	3947	3635	7.90%
	Backup Archiving for VS Deployed VMs	150	15	90%	915	262	71.37%
Old VM Clone	Clone Using Disk Snapshots for VS Ordinary Shut down VMs	150	15	90%	298	70	76.51%
	Clone Using Disk Snapshots for VS Ordinary Started VMs	150	15	90%	373	287	23.06%
	Clone Using Disk Snapshots for VS Deployed Shut down VMs	150	15	90%	342	186	45.61%
	Clone Using Disk Snapshots for VS Deployed Started VMs	150	15	90%	2322	487	79.03%
VM Disk Change	VS Dynamic Disks to Pre-Allocated Disks	150	15	90%	146	67	54.11%
	VM Encryption for VS Dynamic Disks	150	15	90%	131	66	49.62%
	Adding of Dynamically Allocated Disks	150	15	90%	21	17	19.05%

Figure 60: Optimization of Read/Write Time After Void Detection Is Enabled

5.1.3.11 GPU Virtualization

HCI supports GPU pass-through and GPU partitioning to help users meet their requirements for digital transformation and unified business management in GPU applications required in image processing scenarios and AI scenarios.

5.1.3.11.1 GPU Pass-Through

This solution allows a VM to use a physical core passed through from a GPU by using direct memory access (DMA) and virtual function I/O (VFIO) technologies.

The DMA technology is used to allocate a segment of address space from the HCI kernel to a hardware device (a GPU). When an upper-layer application determines to use this hardware device, it directly puts data into the corresponding address space for direct processing by the GPU, avoiding resource consumption from CPU transit and scheduling.

The VFIO technology functions as a protective mechanism for I/O scheduling, which ensures and enhances I/O reliability. By using this technology, the KVM host recognizes and maps each physical core to a VM as virtual hardware on the VM, which is then used by the guest OS.

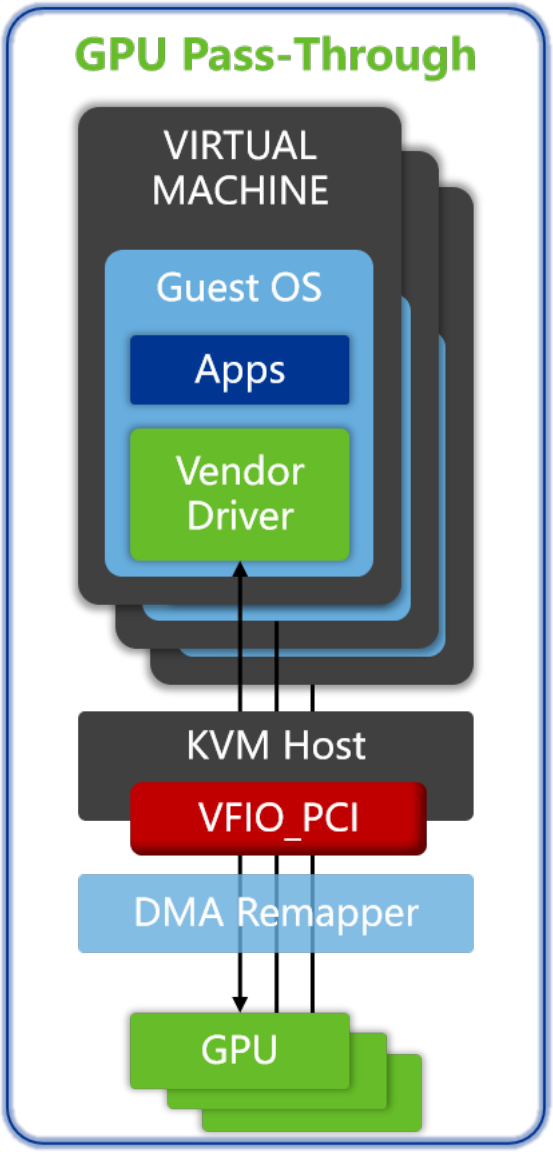


Figure 61: GPU Pass-Through

Userspace processes using a VFIO driver can directly access hardware securely under the protection of an input-output memory management unit (IOMMU) throughout the process, and even unprivileged users can use it directly. That is, VFIO is designed to provide a complete userspace driver solution that can securely display device I/O, interrupt, DMA, and other capabilities to the userspace.

The following figure shows the major VFIO kernel components.

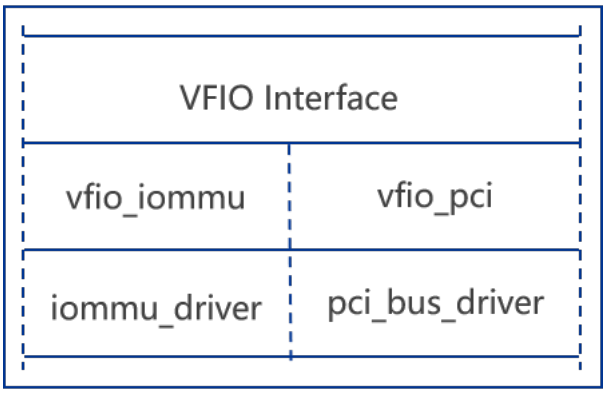


Figure 62: Major VFIO Kernel Components

The VFIO interface layer at the top provides a unified access interface, which allows the userspace to configure and call various capabilities of VFIO through agreed ioctl calls. The intermediate layers consist of the **vfio_iommu** and **vfio_pci** modules. The **vfio_iommu** module allows VFIO to encapsulate the IOMMU layer in a unified manner and is used to implement DMA remapping, which involves managing IOMMU page tables. The **vfio_pci** module allows VFIO to encapsulate the PCI devices in a unified manner and works with userspace processes to complete direct device access, specifically including PCI configuration space emulation, base address register (BAR) space redirection, and interrupt remapping. The bottom layer is the hardware driver call layer. The IOMMU driver module is an implementation related to hardware platforms. For example, it may be an Intel IOMMU driver, AMD IOMMU driver, or PPC IOMMU driver. The **vfio_pci** module calls the **pci_bus** driver on the host to implement operations such as device registration and deregistration.

5.1.3.11.2 GPU Partitioning (NVIDIA vGPU)

This solution is similar to the first solution. However, the GPU on this node is managed by the NVIDIA vGPU host driver at the KVM host layer. The driver can manage GPUs for data center use, like the NVIDIA Tesla and Quadro series, by monitoring their status and partitioning resources. After the GPU is partitioned by the NVIDIA vGPU host, multiple vGPUs are created at the VM layer, and each vGPU can be assigned to a separate VM. NVIDIA Tesla P40 is used as an example. Up to 24 vGPUs can be virtualized from a single GPU, which can be used by 24 users simultaneously. Next-generation GPU virtualization also

comes with other prominent functions, such as vGPUs of specifications varying with scenarios and load requirements and migration for GPU-supported VMs between GPU servers. NVIDIA has introduced software licensing fees to their new hardware virtualization technologies.

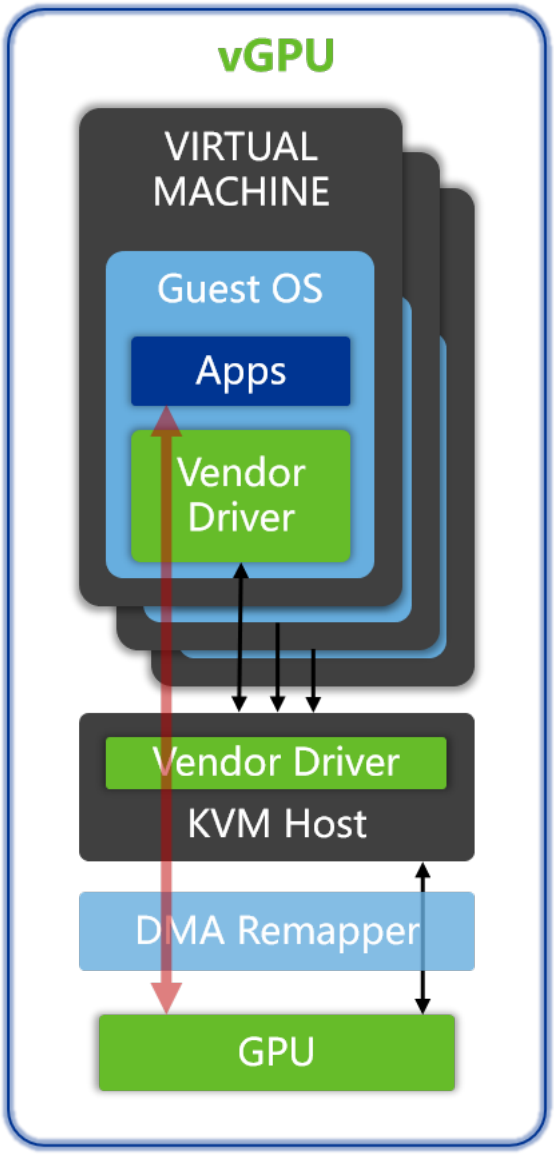


Figure 63: GPU Partitioning

5.1.3.11.3 GPU Scheduling

GPU scheduling allows automatic adjustment of GPU configurations while optimizing the utilization of underlying GPU resources. GPU scheduling modes include the following:

Performance-based scheduling: It aims to ensure balanced usage of different GPU resources and maximized performance for each GPU-enabled VM as much as possible by scheduling GPU-supported VMs to idle machines. This mode is applicable to scenarios that require high GPU performance, such as deep learning, video rendering, and scientific computing.

Density-based scheduling: In this mode, GPU-enabled VMs are scheduled to GPUs that have already been allocated. Therefore, the GPU resources can be made full use of, and GPU-enabled VMs can preferentially run on physical hosts with GPU-enabled VMs. Density-based scheduling is applicable to scenarios like AI training in educational institutions or general office environments.

5.1.3.12 External Storage Shared Disk

In the storage-compute separation scenario, if an Oracle RAC database cluster service runs on the external storage, multiple VMs will share one data disk. HCI previously did not support virtual shared disks on the external storage, and customers could only mount multiple LUNs, making management inconvenient. There is a need to support creating and mounting shared disks to multiple VMs.

5.1.3.12.1 Technical Principles

1. The shared file type is introduced to the external storage SFFS on the data plane, which can be read and written at storage mount points on different hosts.
2. The shared file works as a driver file for QEMU VM disks, enabling VMs in the cluster to use a shared disk at the same time.
3. The Shared Virtual Disk page is provided for creating, deleting, migrating, and cloning shared disks and expanding their capacity.
4. External storage shared disks can be added to VMs to be created and being edited.
5. A VM can be mounted with shared disks on different LUNs.

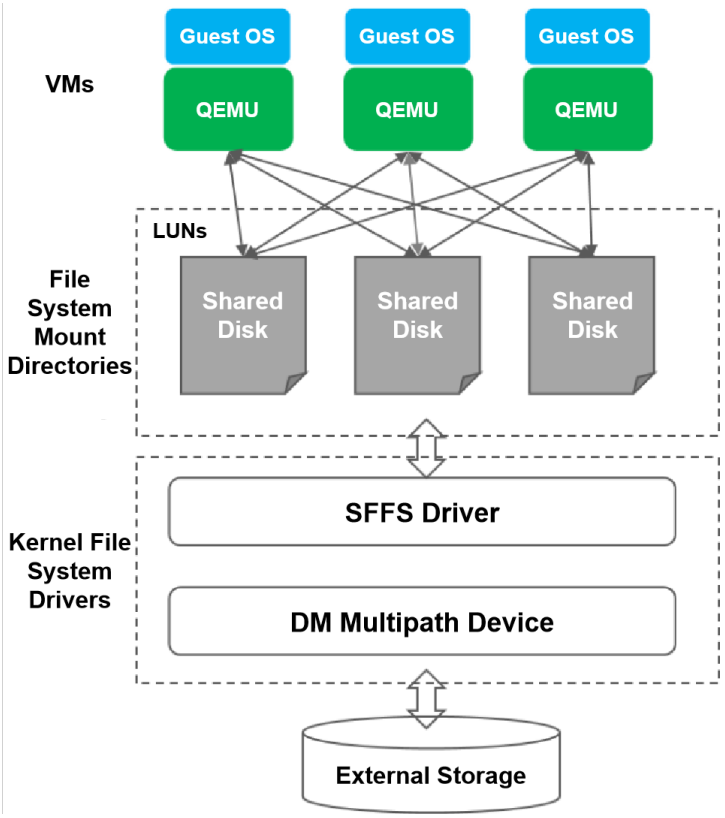


Figure 64: Principles of External Storage Shared Disks

5.1.3.12.2 UI Display

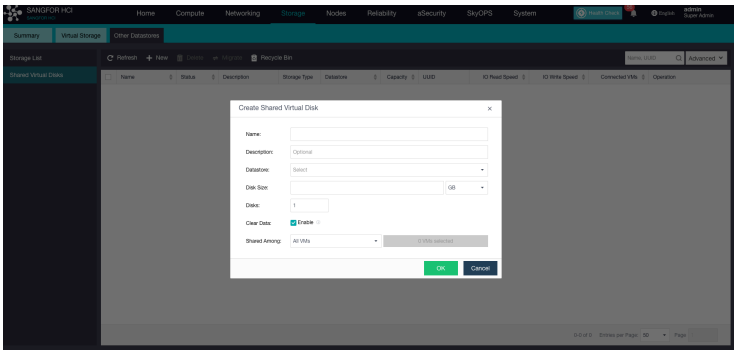


Figure 65: Shared Virtual Disk

5.1.3.13 64 Disks per VM

A bus extension enables VMs to have up to 64 disks (excluding 2 CD/DVD devices) to run the customer's multi-disk services and allows the migration of

VMs with multiple disks between HCI and VMware vCenter.

Limitation	Description
Number of Retained VM Backups	Up to 8192 files can be created in a single directory in SFFS. If the number of retained backups specified for the VM disk backup policy is greater than 8192, some backups cannot be retained.
Primary and Secondary Sites of DR for VMs with More than 16 Disks	VMs with more than 16 disks do not support disaster recovery from a primary site of HCI 6.11.1 to a secondary site with a version earlier than HCI 6.11.1.
Addition of More than 16 Disks After HCI Upgrade to 6.11.1	After upgrading HCI to 6.11.1, users need to upgrade the compatibility version of VMs and restart them before adding more than 16 disks to them.
Disk Hot Add for Linux OSes of Early Versions	Guest OSes with ISO releases earlier than 2016 or kernel versions lower than 4.0 cannot be added with more than 16 (or even more than 12) disks when they are running.
Number of Disk Drives Using IDE Disks	Up to 64 disks are supported (excluding 2 CD/DVD devices) when using VirtIO high-performance disks and up to 4 disk drives (including disks and CD/DVDs) when using IDE disks.

Table 5: Multi-Disk VM Limitations

5.1.4 Data Protection

Sangfor aSV provides multiple data protection functions, including VM snapshots for logical failures, quick backup, periodic full backup, compression and archiving, and CDP for physical failures, quick restoration during fault recovery, and one-click data clearing for confidential data protection.

5.1.4.1 VM Snapshots

According to the definition provided by the Storage Networking Industry Association (SNIA), a snapshot is a fully usable copy of a specified data set. The copy includes an image of the corresponding data at a certain time point when

the copy starts. A snapshot can be a copy or replica of the represented data. SNIA defines two types of snapshots, that is, full snapshots and incremental snapshots, each using different snapshot technologies:

1. Full snapshots: split mirror
2. Incremental snapshots:

- Copy-on-write (COW)
- Redirect-on-write (ROW)

The VM snapshot function provided by aSV is based on the QCOW2 image file. When used on a VM, this function can be used to record the disk file data at the moment when the snapshot is taken, and administrators can restore the data to the state at that moment by rolling back the snapshot.

A QCOW2 image file has multiple features, including support for multiple snapshots (images can be created based on previous ones much faster), less storage space usage (sparse features are not supported, and no storage space of the specified size is allocated in advance), optional AES encryption, and optional compression using zlib. A QCOW2 image simulates a block device with a certain size in a file system, thus making its format a disk image format supported by QEMU-KVM.

A QCOW2 image can be used to save changes made to another QCOW2 image (template image) to ensure that the original image content remains unchanged. This is called an incremental image. An incremental image seems like an independent image file, and all its data is obtained from the template image. The clusters in the incremental image will be saved only when the content of such clusters is different from that of the corresponding clusters in the template image. When reading a cluster from the incremental image, QEMU first checks whether the cluster has been allocated new data (data has changed). If not, QEMU reads the corresponding cluster from the template image.

In QCOW2 mode, the COW method is used to create VM snapshots. It copies the image of a running VM and uses the copied image as a template image without shutting down the VM. An incremental image is also generated to

record the changes made to the template image. Each incremental image references the same template image. The template image must remain in a read-only state, while the incremental image is readable and writable. In principle, an incremental image is equivalent to a snapshot of the template image. When a VM fails, the system only needs to read the smaller snapshot file and automatically call the corresponding template image, thus restoring the VM in a fast and efficient manner. Furthermore, QCOW2 snapshot files contain less content and do not occupy too much virtual disk space. As a result, each snapshot process can be completed within 1 minute. Only the 1-minute difference in all data between active and standby images exists, which ensures high consistency between active and standby VMs, thus basically implementing the synchronization of application data and disk data between the active and standby VMs.

The VM snapshot function provided by aSV is based on the QCOW2 file. When used on a VM, this function can be used to record the disk file data at the moment when the snapshot is taken, and administrators can restore the data to the state at that moment by rolling back the snapshot.

Snapshot process in aSV:

1. Snapshot 1 is generated, which records the location of all data.
2. Command writing: The data content at location 1 is changed from A to G.
3. The original data at location 1 remains unchanged but is copied to a new location and written as G.
4. Location 1 is made to point to G.

Snapshot 1 in QCOW2 format records the original location, including the data content A to which location 1 pointed originally. Similarly, snapshot 2 is generated, and other data is written at the original location.

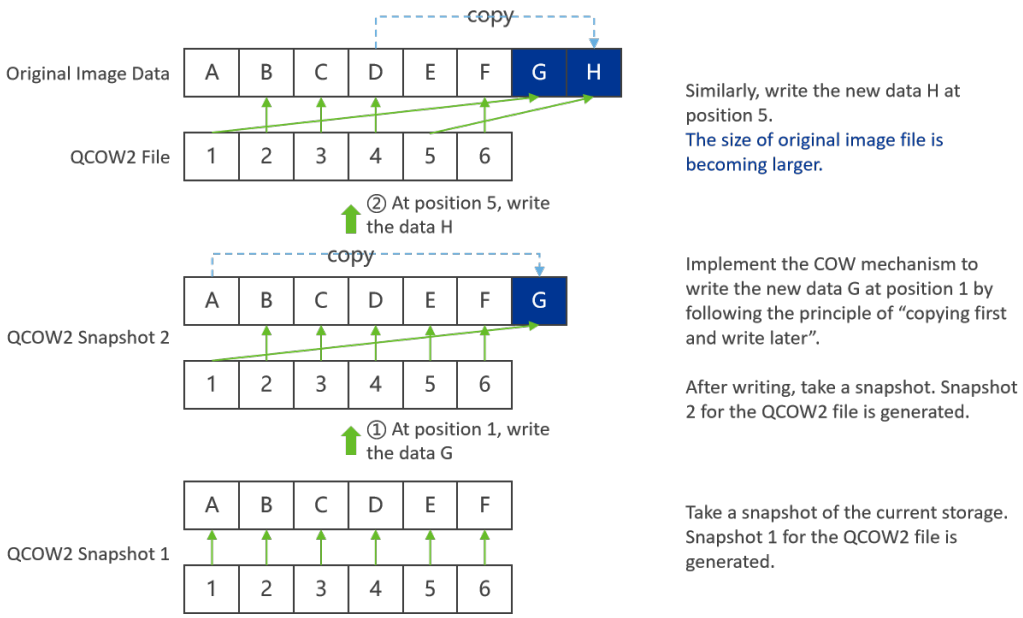


Figure 66: Snapshot Process

Snapshot 1 and snapshot 2 in QCOW2 format record the original locations and corresponding data, including the data A to which location 1 (which now points to the data G)in snapshot 1 pointed originally, and the data E to which location 5 (which now points to the data H) in snapshot 2 pointed originally. The status of the image recorded by each snapshot is complete, allowing administrators to restore the VM image data to that at any snapshot time point.

In the HCI solution, if aSV is used and external storage is connected, VM snapshots are taken using the COW method. However, if aSAN-based virtual storage is also used, VM snapshots are taken using the ROW method by default. This reduces the write amplification and I/O and improves the write speed and performance. For more information about storage snapshots, see "Storage Snapshots."

5.1.4.2 Quick Backup

Only the reliability measures of business servers (such as active-active storage, multiple replicas, HA, and automatic system restart in case of failures) are not enough to address business data failures such as deletion of data by mistake or ransomware attacks within a VM, because such failures directly damage the data source and thereby damage the replicated data as well. For example, the

recent surge in ransomware attacks has resulted in internal data corruption within business systems, rendering multiple replicas ineffective as each replica is corrupted.

In response to such failures, HCI relies on a quick backup mechanism, which uses an initial full backup, followed by incremental backups, along with the bitmap technology (which is used to mark the changed locations in the QCOW2 file, also known as dirty data locations) for quick backup. The following describes its principles.

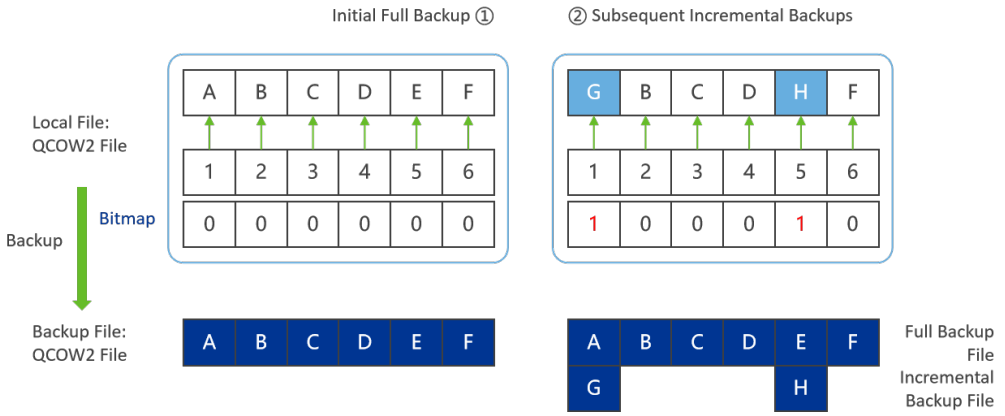


Figure 67: Backup Process

1. Full backup is preferred during VM backup, but in the case of an existing full backup, incremental backup is used directly.
2. After the full backup, new business data (G and H in the figure above) is continuously written and marked with bitmaps. It can be directly written into the QCOW2 file at the original locations, and during the next backup, only the modified data is incrementally backed up. When the backup is completed, the bitmap is reset to zero for smooth backup operation each time. New data can be directly written into the file at the original locations. COW will not occur, or it will not result in any disorder in the mapping between the QCOW2 file and the data locations. As a result, the performance of QCOW2 images is not compromised. The incremental backup accelerates the process by reducing the volume of data to be backed up each time.

3. Incremental backup files are deleted, and the data of all incremental backup files is merged in a way that all the reserved backup data is complete and usable. This frees up space quickly and implements efficient utilization of backup repository resources.

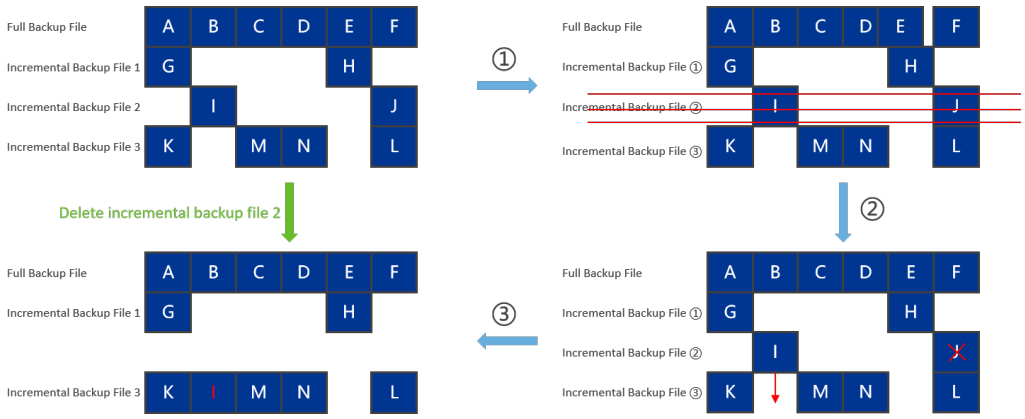


Figure 68: Backup File Deletion and Merge Process

NOTICE

The deletion operation eliminates the restore point of incremental backup file 2, but the data of incremental backup 1, incremental backup 3, and subsequent ones is all complete, accurate, and recoverable.

Quick backup also features multi-disk data consistency check in scenarios where multiple disk images of a VM or disk images of multiple VMs are related. For example, in database application scenarios, the data disks and log disks of the database (such as SQL Server or Oracle) must maintain consistency at the time of backup. Otherwise, the recovered database may still be invalid due to inconsistency. Sangfor, with its quick backup function, ensures data consistency across multiple database disks.

Straightforward backup operation in HCI:

You can add a backup policy compliant with business characteristics in four steps as prompted in the wizard.

The automatic backup period can be flexibly configured by week, day, or hour based on the actual business requirements. Automatic cleanup is available to save backup repository space to the maximum. In addition, no plugins are

required, making it simple to use.

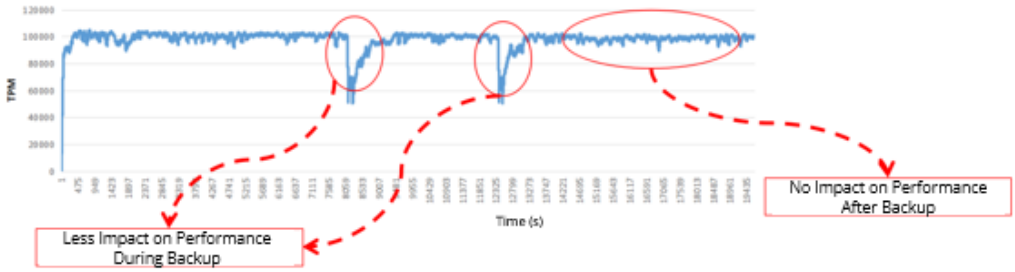


Figure 69: Performance Variation of Quick Backup

The figure above shows that the QCOW2-based backup performance deteriorates slightly during the initial full backup and subsequent incremental backup, which quickly returns to the initial level after the backup is completed.

Backup file cleanup policy:

You can customize the backup file retention policy based on business requirements, with time granularity ranging from days to years. This provides greater flexibility compared to the previous retention policy.

5.1.4.3 Quick Restoration

In the event of a business system failure, the VM can be recovered to the most recent point of normal business operation by querying the backup files if data backup has been enabled. Generally, perform the following operations for backup and restore: migrate the backup data to the running location and start the VM at the running location to resume business operations after the data migration is completed. However, substantial business data volume and long migration duration may result in a large recovery time objective (RTO). As a result, Sangfor HCI further optimizes the backup and restore function to address this issue.

Major Optimizations for Quick Restoration:

Sangfor HCI introduces a new restoration method called new VM generation. Before optimization, the VM data was restored by overwriting the original data with the backup data, which made it impossible to check whether the backup data was available before overwriting. The new restoration method effectively

solves this problem.

Under certain restrictions, quick restoration of VM backups can be achieved in just a few minutes, and the VM can then be powered on. The VM is started for quick business restoration, and the backup data is migrated to the target storage, thus resuming business operations to be online quickly.

The following figure shows the principle of quick restoration.

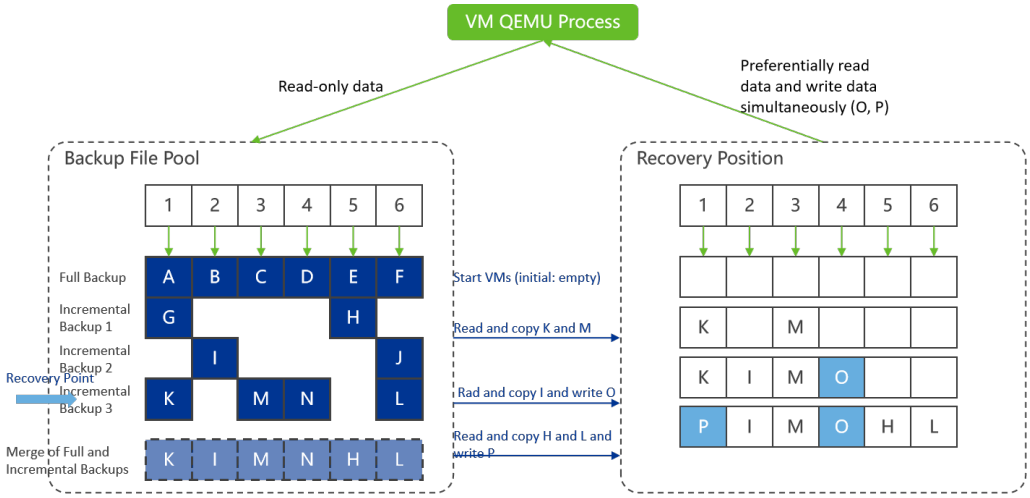


Figure 70: Quick Restoration

Procedure:

1. Start the VM for business operation quickly.

Create a blank image at the restoration location. The QEMU process starts the VM using this blank image for normal business operations.

For example, the first line for the restoration location in the above figure is a blank image.

2. Preferentially read business data from the restoration location.

During business operations, data is preferentially read from the restoration location. If no data is available there, the data needs to be migrated from the backup file pool to the restoration location before data reading.

For example, in the second line for the restoration location in the above figure, the data K at location 1 and the data M at location 3 need to be read on the business side.

3. Write business data at the restoration location.

During business operations, newly generated data from the VM is directly written into the new image.

For example, in the third and fourth lines for the restoration location in the above figure, the data O is written at location 3, and the data P is written at location 1.

4. Continuously migrate backup data from the backup file pool to the restoration location.

During VM restoration for business operations, a QEMU process continuously transfers data from the backup file pool to the restoration location silently at the backend of the VM. This process is known as backup data fetching. The VM is restored to a normal state after backup data fetching is completed.

For example, in the above figure, when the QEMU process identifies no data at location 2 in the third line for the restoration location, it migrates the data for location 2 from the backup file pool to the restoration location. When all data migration is completed, as shown by the fourth line, the VM is fully restored to a normal state.

Mechanism for Data Loss Prevention During Quick Restoration:

During quick recovery, the first step is to quickly start the VM by creating a blank image at the restoration location for quick business recovery. The second step is to preferentially restore the hot data required for business operation, thereby enhancing performance within a short period of time. Following this, migrate other data (cold data) from the backup file pool to the restoration location without affecting the operation of major businesses.

5.1.4.4 Periodic Full Backup, Compression and Archiving, and Backup Copy

aSV provides the initial full backup, followed by incremental backups, by default. However, to meet certain compliance requirements for periodic full backup and secondary data archiving in addition to incremental backups, aSV also provides the solutions for periodic full backups and compression and archiving. The following figure shows the process:

1. A user configures the automatic backup frequency of once per week, full backup frequency of once per month, and archive backup frequency of once per month.
2. At the beginning of each month, a full copy is performed according to the user configuration (the original image instead of the incremental image based on the previous backup is copied). This ensures the data reliability of the full backup. Subsequent incremental backups are performed based on this new image for the next three weeks.
3. At the end of the month, a new image is generated by merging the backup repository based on the last incremental backup and then copied to the archive datastore for archiving. Note that archiving does not indicate direct copying from the original image (production storage). Instead, data archiving is completed from the backup repository to reduce the impact on the original production environment. Given the frequency and space utilization of the archive datastore, aSV provides the compression function to optimize storage resource allocation.

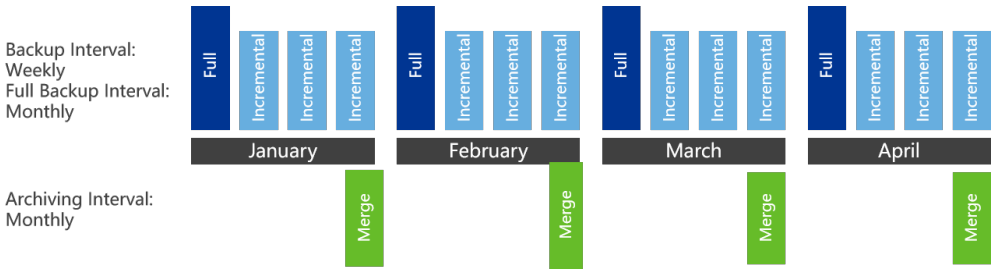


Figure 71: Periodic Full Backup

For the archive datastore, the following conditions must be met:

1. It must be distinct from the backup repository used during archiving.
2. It cannot be virtual storage.

However, conventional full backup and archiving solutions cannot meet the demanding compulsory requirements set by some special industries on security and reliability for their critical business data. To address this issue, aSV provides the backup copy function, which allows one backup and multiple backup replicas to be stored at different storage locations. In this case, the

online storage, remote storage, and offline storage requirements can also be met.

Backup copy policies support the configuration of source storage, destination storage, copy frequency, and copy rate limits. The policies are executed automatically and periodically for a backup copy from the source storage on the VM to the destination storage. Manual backup copy is also supported.

Precautions:

1. The destination storage for backup copy cannot be local storage.
2. The source storage and destination storage for backup copy cannot be the same.
3. No limits are set on the quantity of backup replicas, and backup copy policies can be continuously added. For example, if three replicas are required, one backup policy and two backup copy policies can be added.

5.1.4.5 Backup Health Check

The Backup Health Check feature supports periodic checks of the latest VM backup chain and manual backup point checks. It checks the metadata and data block readability of the backup to verify the recoverability of the backup data.

5.1.4.5.1 Technical Principles

Backup Points:

- **Backup Point**: The state of a VM at the time of creating a backup file of the VM, whose data is usually stored on blocks of multiple backup files in a backup chain.
- **Latest Backup Point**: The state of a VM at the time of creating the latest backup file of the VM, whose data is usually stored on blocks of multiple backup files in a backup chain.

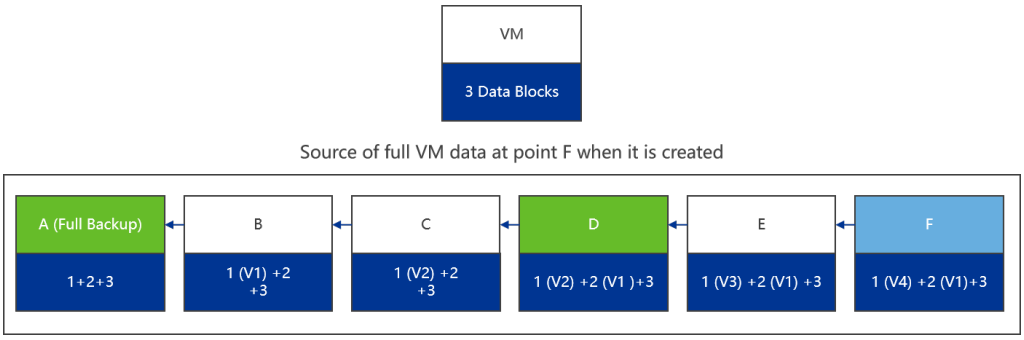


Figure 72: Backup Points

As shown above, suppose a VM has three data blocks (actually more than three) and backup points A, B, C, D, E, and F. Among them, point A has the data of all the data blocks, and point F is the latest backup point after five incremental backups. If the backups to points B, C, E, and F all modify only block 1, the backup to point D modifies only block 2, and only the first full backup at point A modifies block 3, then the full VM data at the latest point F consists of block 3 at point A, block 2 at point D, and block 1 at point F. To perform a recovery from the backup at point F, you need to ensure the integrity of the data at points A, D, and F.

Necessity:

Backup storage anomalies and malicious metadata tampering may cause backup data to become corrupted or unreadable, or backup files to be lost. If the backup data to be recovered encounters a failure, the RPO will be invalid and the RTO will be greatly increased, or even the recovery will be unavailable, bringing significant business losses.

By configuring a health check policy that applies backup data verification technology, data anomalies can be detected and recovered in time during daily backups. For unrecoverable data, a full backup is performed to generate the latest backup point for recovery, ensuring the security of the business services.

Workflow:

Automatic Backup Health Check:

The automatic backup health check checks the full backup point of the latest VM backup chain first, and then all incremental backup points based on that point in sequence.

- Check the metadata at all backup points of the latest backup chain. If there is metadata corruption, execute as follows:

1. Repair the metadata and then proceed to check the next backup point. If repair fails then execute as follows.
2. Mark the check results for the backup point with corrupted metadata and the latest backup point and all points between them as unavailable, meaning that the VM cannot be recovered from these backup points.
3. Mark the backup health check task as failed and trigger an alert log.
4. Trigger a full backup task.

- Check the readability of data blocks for the latest backup point.

5. If all data blocks are readable, then the backup health check task is completed. If there is an unreadable data block, execute as follows.
6. Mark the backup health check task as failed and trigger an alert log.
7. Trigger a full backup task.

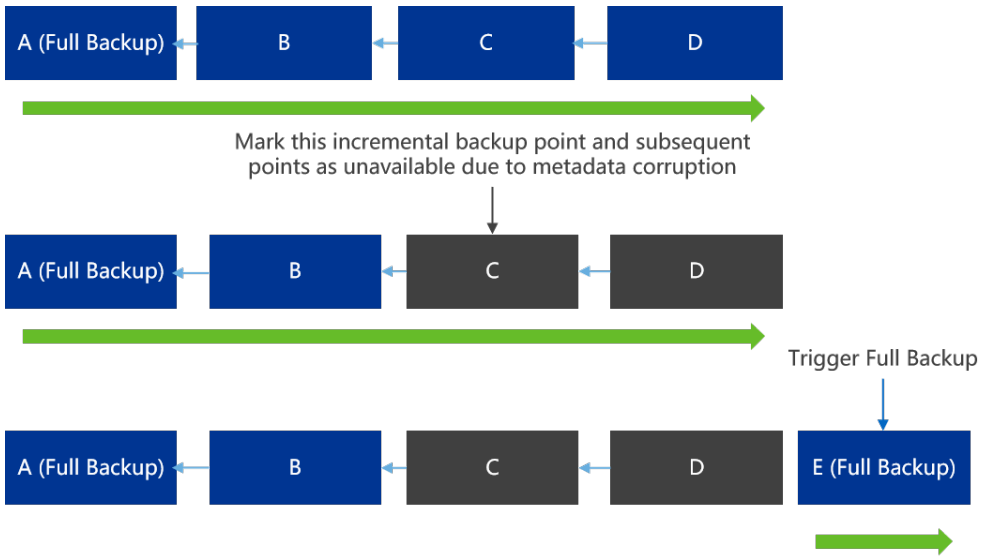


Figure 73: Metadata Check

The metadata check is illustrated in the figure above, suppose the latest backup chain contains backup points A, B, C, and D, where A is a full backup point and D is the newest backup point. To verify the metadata of point D, it is necessary to verify the metadata of points A, B, C, D, and E in sequence. If the

metadata of point C is corrupted and cannot be repaired then points C and D will be marked as unavailable and a full backup will be triggered at point F.

Manual Backup Health Check:

Along with configuring an automatic backup health check policy, you can also select a backup chain and its backup points for manual checks. The manual backup health check does not check points after the point to be checked and does not trigger a full backup. If there is a point with corrupted metadata between the point to be checked and the full backup point, then that point and all subsequent points will be marked as unavailable.

5.1.4.5.2 Resource Allocation

A backup health check requires up to 128 MB of memory size and 50% of the usage of a single CPU core. The following data was obtained in the testing environment with the maximum speed of 300 MB/s and the virtual space size of a single disk of 1 TB, which reveals that the CPU usage and memory consumption increase with the number of backups of a backup chain and their valid data size.

CPU	Clock Speed	Max I/O Read Speed for Backup Repository
14 cores, 28 threads Intel(R) Xeon(R) Gold 6132 CPU @ 2.60 GHz	2.60 GHz	471 MiB/s

Table 6: Testing Environment Configurations

Backups	Valid Data (GiB)	Avg Single CPU Core Usage	Avg Memory Consumption (MiB)
7	52	3.14	8.14
21	86	3.15	12.2
31	106	3.72	15.3

64	205	4.56	24.25
105	287	6.09	35.43
128	333	7.02	41.64
256	390	16.32	74.50

Table 7: Resource Allocation

5.1.4.5.3 UI Display

1. Select **Backup Health Check** when configuring a backup policy against backup metadata corruption or unreadable data blocks due to storage failures.

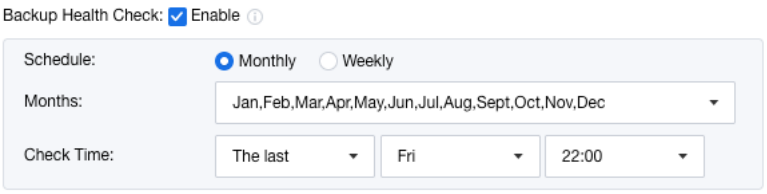


Figure 74: Configure Backup Health Check

2. Go to **Reliability** > **Scheduled Backup/CDP** > **Settings** > **Max Concurrent Tasks** to set **Backup Health Check Tasks per Datastore**. Range: 1-16. Default: 2.

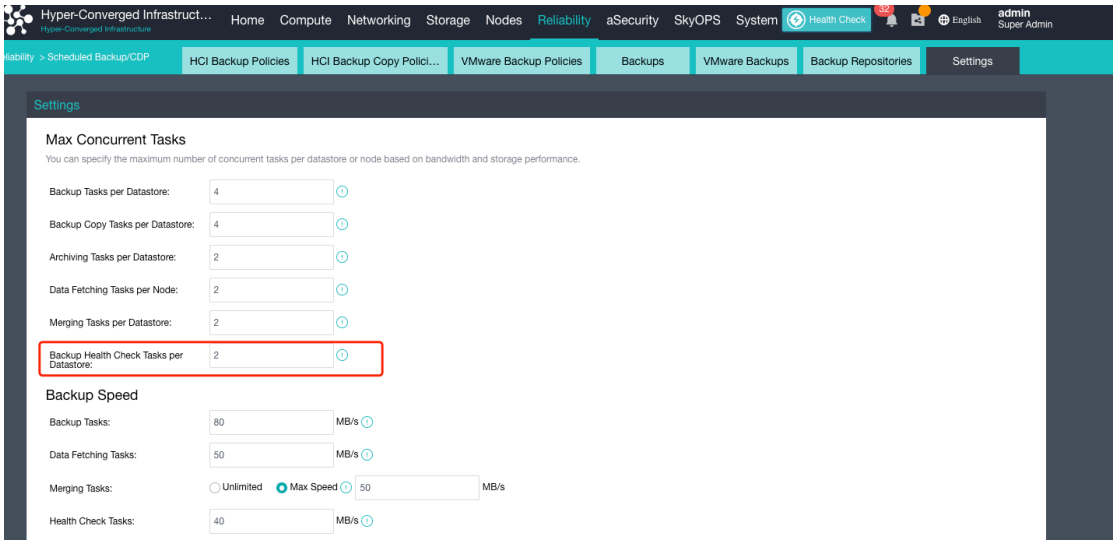


Figure 75: Set Backup Health Check Tasks per Datastore

3. Go to **Backup Speed** to set **Health Check Tasks**. Range: 1-300 MB/s. Default: 40 MB/s.

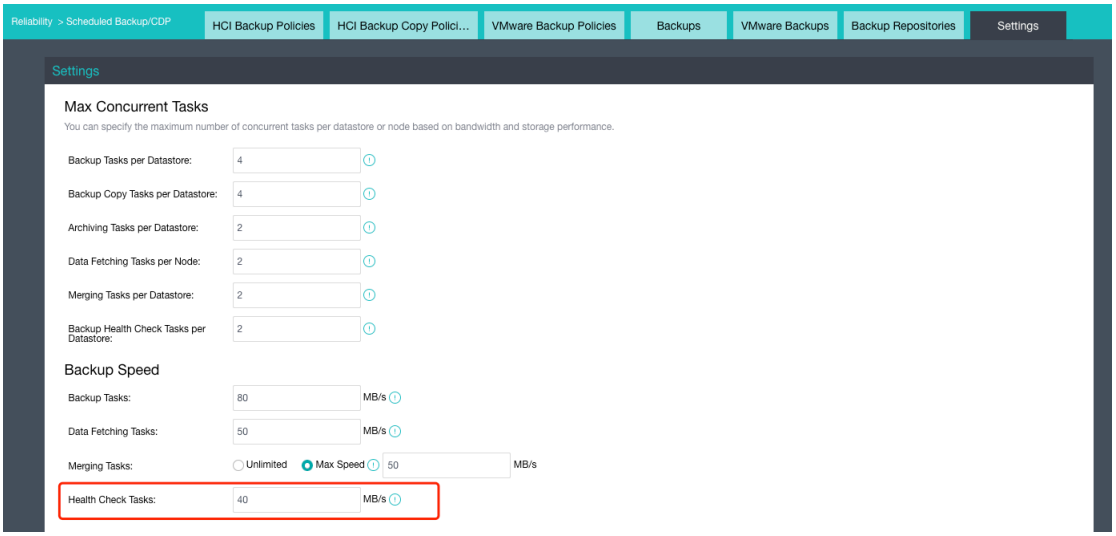


Figure 76: Set Health Check Task Speed

Precautions:

1. Backup points that have passed the backup health check are marked as passed, abnormal backup points are marked as unavailable or partly available, and those that have never been checked are marked as not checked.
2. A backup point that has passed the backup health check is not permanently available. It is recommended to perform another check on the backup point before recovery to make sure that it is recoverable.
3. VMs cannot be recovered or cloned from unavailable backup points, which cannot pass the check again.
4. VMs cannot be cloned from partly available backup points.
5. Only the available disks of a VM can be recovered from a partly available backup point by overwriting the existing VM.
6. The backup health check will use the results of the last check directly if the last check was performed less than 24 hours ago.

5.1.4.6 Agentless Backup

By using the OpenAPI and SDK for disk data access provided by Sangfor SCP, third-party backup applications can configure VMs on SCP and their disks and manipulate disk data to enable agentless VM business backups and recovery, without installing plugins on SCP or VMs.

5.1.4.6.1 Terminologies

1. Sangfor Virtual Disk Development Kit

Sangfor Virtual Disk Development Kit (SFVDDK) is a development kit consisting of SDKs and APIs written in the C programming language for accessing virtual disk files on SCP. Its APIs cover basic operations such as opening, closing, reading, and writing to virtual disk files.

2. OpenAPI

SCP provides management of basic compute, network, and storage resources, including snapshots, backups, and other resources relevant to agentless backups, so that third-party users can standardize the management, and operation and maintenance of SCP through existing APIs. The APIs are all Representational State Transfer (REST) APIs that can be called by HTTPS requests.

3. CBT

Changed Block Tracking (CBT) is a technology that improves the efficiency of agentless backups performed by third-party applications on VMs on SCP. VMs with CBT disabled support only full backups, backing up all data at the snapshot point; VMs with CBT enabled support incremental backups and generate CBT bitmaps through which bitmaps of allocated data blocks can be obtained.

When taking a snapshot of a VM with CBT enabled, a bitmap checkpoint is generated at the same time, containing bitmaps of all the changed data blocks between the current snapshot point and the previous snapshot point. SCP 6.10.0 and HCI 6.10.0 provide an API for querying the CBT bitmaps of the

changed data blocks between the specified snapshot point and the latest snapshot point.

Precautions:

1. For the first backup, query CBT bitmaps by specifying 0 for the checkpoint ID to obtain the bitmaps of all the changed data blocks of the snapshot point.
2. For subsequent backups, query CBT bitmaps by specifying the checkpoint ID of the last backup to obtain the bitmaps of all the changed data blocks between the current snapshot point and the last backup.
3. If **The VM CBT has expired. Please reset it** is prompted when querying the CBT bitmaps, it means that the VM may have been recovered from a snapshot or a backup. Please reset the CBT bitmap before performing a backup.
4. When the size of the disks being backed up is very large, the API for querying the CBT bitmaps may be unable to return the bitmap data of all the changed data blocks, and you need to query them multiple times.
5. VM disk write performance will be degraded after CBT is enabled. Please enable CBT for disks as needed.

5.1.4.6.2 Workflow

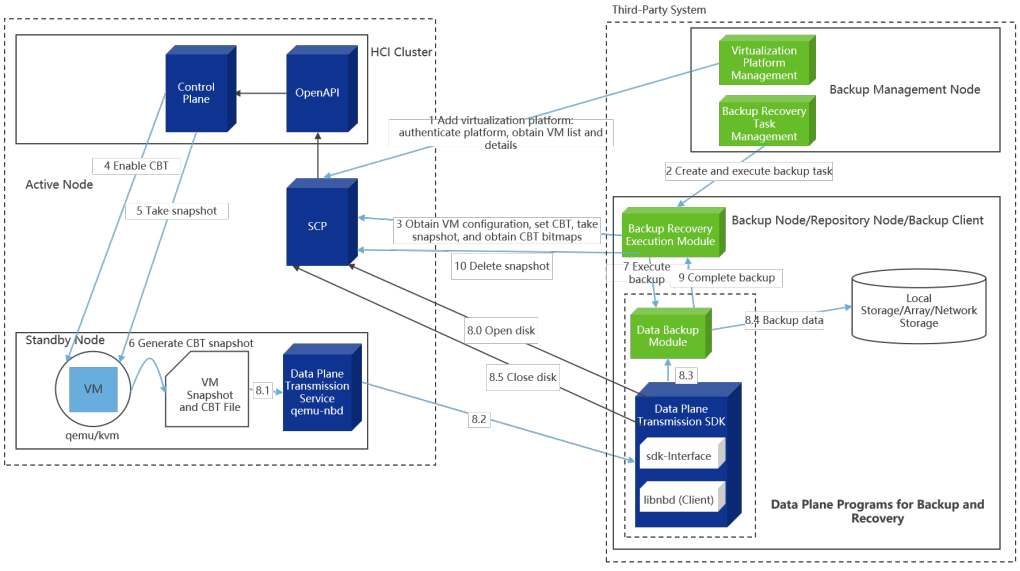


Figure 77: Agentless Backup Workflow

Backup:

- Pre-Process

1. Enable or reset CBT for a VM. An incremental VM backup can be efficiently performed by using the CBT mechanism to quickly obtain and read only the changed data blocks of the virtual disks between a snapshot point and the latest snapshot point.

- Backup

2. Take a VM snapshot.
3. Query the VM snapshot details to obtain the VM configurations and the checkpoint ID at the snapshot point.
4. Create the VM backup resource and update it periodically (once every 1-2 minutes is recommended) to prevent it from expiring. If the backup resource expires, SCP will delete it automatically and terminate this backup, and the backup application will be unable to read the related data on the virtual disks.
5. Query the CBT bitmaps to obtain the changed data blocks of the virtual disks between the checkpoint ID of a snapshot and the latest snapshot for the data area to be backed up. For a full backup, set the checkpoint ID to 0 to return the valid data area of the virtual disks.
6. Read the changed data of all disks of the VM from SCP and store it in the third-party backup storage.
7. Delete the VM backup resource.
8. Delete the VM snapshot.

Precautions:

1. Third-party backup applications need to work with OpenAPI and SFVDDK to back up VMs on SCP.
2. Disable CBT for VMs when no longer using third-party backup applications to prevent it from degrading VM I/O performance.

Recovery:

Ordinary Recovery:

In an ordinary recovery, the VM can only be started after the disk data has been recovered which takes a long time.

- Pre-Process

1. Shut down the VM.

- Recovery

2. Restore the VM Configuration. Depending on whether the VM to be recovered is new or already existing, the backup application creates a new or edits the original VM API based on the configuration and restores the VM configuration from the backup to the target VM.
3. Create the VM backup resource, set the resource type to the recovery type, and update it periodically.
4. Write the backup data to the virtual disks.
5. Delete the VM backup resource.

Rapid Recovery:

In a rapid recovery, the VM can be started immediately by directly using the backup data on the third-party backup storage as the source of data. This method eliminates the need for data recovery in scenarios that only require verification of backup availability and also supports data recovery for common backup recovery scenarios.

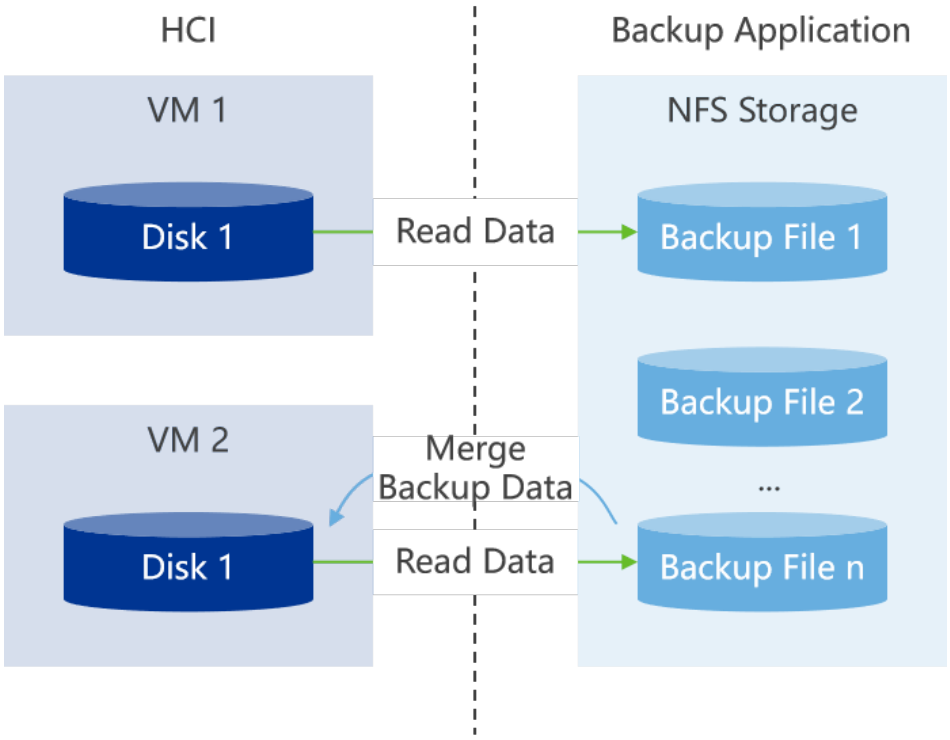


Figure 78: Rapid Recovery Principle

- Pre-Process

1. Shut down the VM.

- Recovery

2. Assemble the backup data into virtual disk files in .raw or .qcow2 format (cluster size not smaller than 64KB) and place them in the NFS storage by the third-party backup application.
3. Mount the NFS storage to HCI by calling the API that mounts it (SCP: POST /janus/20210725/storages/attachnfs; HCI: POST /hci/vt/api/v1/datastores/actions/attachnfs).
4. Call the API for modifying the VM disk base image (SCP: POST /janus/20180725/servers/{server_id}/rebase-backing-file; HCI: POST hci/vt/api/v1/vms/{id}/actions/ rebasebackingfile) to make each disk of the VM to take the virtual disk file in the third-party NFS storage as the base image.

5. The rapid recovery is completed and the VM can be normally started with its type changed to Quickly Recovered VM. To recover the VM backup data, execute as follows:
6. Call the API for enabling backup data merge (SCP: /janus/20180725/servers/{server_id}/merge-backup-data; HCI: POST hci/vt/api/v1/vms/{id}/actions/ startmergebackupdata), SCP will automatically merge the backup data from the third-party NFS storage to the production storage where the VM disks are located.
7. Query the VM details and wait for the VM type to change to Non-Quickly Recovered VM, after which the above backup data merging will be completed and the VM will no longer access the third-party NFS storage when it is running.
8. Unmount the NFS storage.

Precautions:

Quickly recovered VMs access the backup data on the NFS storage. Please do not unmount the NFS storage; otherwise, Datastore is in use will be prompted, causing VMs accessing this storage to become abnormal. To safely unmount the NFS storage, execute as follows:

1. Shut down the VM.
2. Delete the VM.
3. Unmount the NFS storage.

5.1.4.6.3 Advanced Features

Platform File Upload and Download:

You can use the APIs provided by SCP and HCI for file upload and download to back up and restore non-disk files such as custom BIOS files and TPM files.

- Upload REST API for SCP:

POST /janus/20180725/clusters/{cluster_id}/files/upload

- Upload REST APIs for HCI:

GET /hci/vt/api/v1/files/uploadinit

POST /hci/vt/api/v1/files/upload

The two interfaces must be used together to upload a file. For details on the parameters of the REST APIs, refer to the API documentation of the corresponding version.

- Download REST API for SCP:

GET /janus/20180725/clusters/{cluster_id}/files/download

- Download API for HCI:

GET /hci/vt/api/v1/files/download

Backup and Recovery of VMs with a Custom BIOS File:

A VM with a custom BIOS file must be backed up and recovered along with the BIOS file; otherwise, the VM may not be properly booted. To check whether a VM includes a custom BIOS file, you can query whether the BIOS file type (vmconf.hardware_option.bios.bios_type) of the VM is CUSTOM through the following REST APIs for VM snapshot details:

- SCP:

GET /janus/20180725/servers/{server_id}/snapshots/{snapshot_id}

- HCI:

GET /hci/vt/api/v1/snapshots/{id}

Custom BIOS File Backup:

You can use the file download APIs provided by SCP and HCI to download and back up a custom BIOS file.



1. You do not need to set the snapshot_id parameter of the REST API for the BIOS file download because the BIOS is non-volatile storage.

2. You can obtain the BIOS file name from the Content-Disposition in the HTTP response header, which will be used for recovery by overwriting.

Custom BIOS File Recovery:

You can use the file upload APIs provided by SCP and HCI to recover a custom

BIOS file through the following two methods:

1. Recovery to a newly created VM. This method requires uploading the BIOS file to the ISO directory of the specified storage, setting the BIOS type (hardware_option.bios.bios_type) of the parameters for creating a VM to custom and setting the custom BIOS path (hardware_option.bios.path) to the path of that custom BIOS file (format: storageId:storageId/path/of/file).
2. Recovery by overwriting the custom BIOS file of an existing VM. This method uploads the BIOS file directly to a directory of the target VM. Please make sure that the target VM exists and includes a custom BIOS file, and that the file name passed to the upload API is consistent with the BIOS file name of the VM.

NOTICE

1. You only need to set the VM ID parameter of the API (SCP: server_id, HCI: vmID) for uploading a custom BIOS file to a VM directory, where the new BIOS file will overwrite the original one.

2. The request method of the upload interface is POST, which uploads the file in forms (Content-type: multipart/form-data). Other parameters are passed as query parameters.

Backup and Recovery of VMs with TPM Enabled:

VMs can create and store encryption keys through the Trusted Platform Module (TPM). After a VM has TPM enabled, the TPM file must be backed up and restored; otherwise, the business services running on the VM may be abnormal. To check whether a VM has TPM enabled, you can query whether the value of vmconf.advanced_option.tpm_enabled parameter is true through the APIs for VM details or the following REST APIs for VM snapshot details:

- SCP:

GET /janus/20180725/servers/{server_id}/snapshots/{snapshot_id}

- HCI:

GET /hci/vt/api/v1/snapshots/{id}

1. TPM File Backup

You can use the file download APIs provided by SCP and HCI to back up a TPM file of the corresponding snapshot point.

NOTICE

1. You need to set the snapshot ID parameter (SCP: snapshot_id, HCI: snapshotID) for downloading a TPM file of the corresponding snapshot point.

2. If the TPM file download fails and the HTTP status code is 404, it means that the VM does not include the TPM file (this problem can occur on VMs that have never been started); if the HTTP status code is not 404 (e.g., 500), it means that the server encounters an internal error, please contact a Sangfor technical support representative.

3. The TPM file is a compressed package in .zip format that can be backed up and restored as a single unit on the original platform.

2. TPM File Recovery

You can use the file upload APIs provided by SCP and HCI to upload and recover a TPM file to a specified newly created VM or an existing VM through the same method.

NOTICE

1. TPM files can only be restored by overwriting to a directory of an existing VM, for which you must set the VM ID parameter of the upload API (SCP: server_id, HCI: vmID) and make sure that the target VM exists.

2. The request method of the upload interface is POST, which uploads the file in forms (Content-type: multipart/form-data). Other parameters are passed as query parameters.

Encrypted VM Backup:

SCP supports creating encrypted VMs whose keys are managed by the platform and cannot be exported, therefore encrypted VMs can only be recovered by overwriting the original one. To check whether a VM is encrypted, you can query whether the value of the vmconf.advanced_option.encrypted parameter is true through the following REST APIs for VM snapshot details:

- SCP:

GET /janus/20180725/servers/{server_id}/snapshots/{snapshot_id}

- HCI:

GET /hci/vt/api/v1/snapshots/{id}

1. Encrypted VM Backup

Encrypted VMs are backed up in a way that differs from ordinary VMs only in that the disk data read is in ciphertext.

2. Encrypted VM Recovery

The backup data of encrypted VMs is in ciphertext and the encryption key cannot be obtained during backup. Please do not recover an encrypted VM to a new VM; otherwise, the VM may be unable to start up.

5.1.4.6.4 UI Display

1. Log in to HCI, go to **System** > **Port Management**, select **Agentless backup data transfer service**, click its status, enter the admin password, and click **OK**.

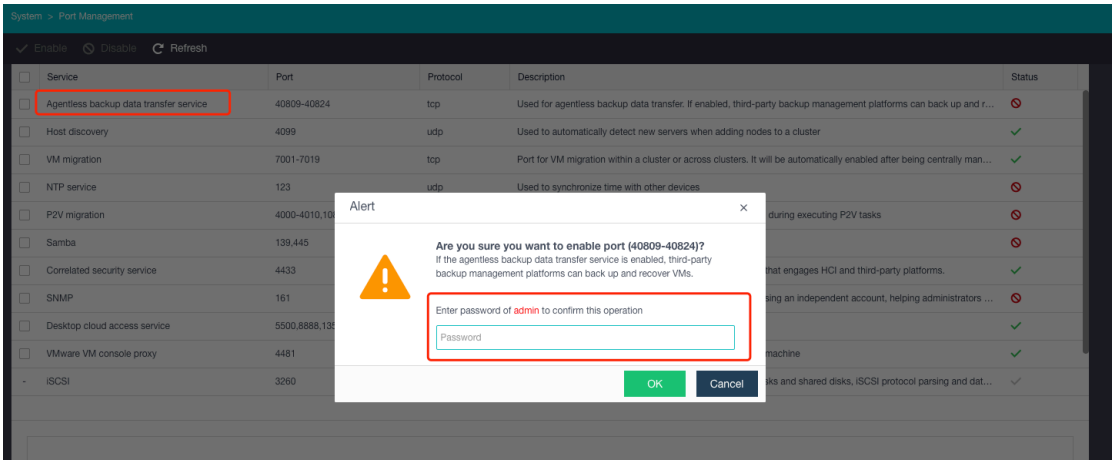


Figure 79: Enter Admin Password

2. Go to **Reliability** > **Scheduled Backup/CDP** > **Settings** > **Agentless Backup** to configure the backup data transmission interface of HCI to limit the use of HCI resources by third-party backup systems to which the agentless backup feature is connected.

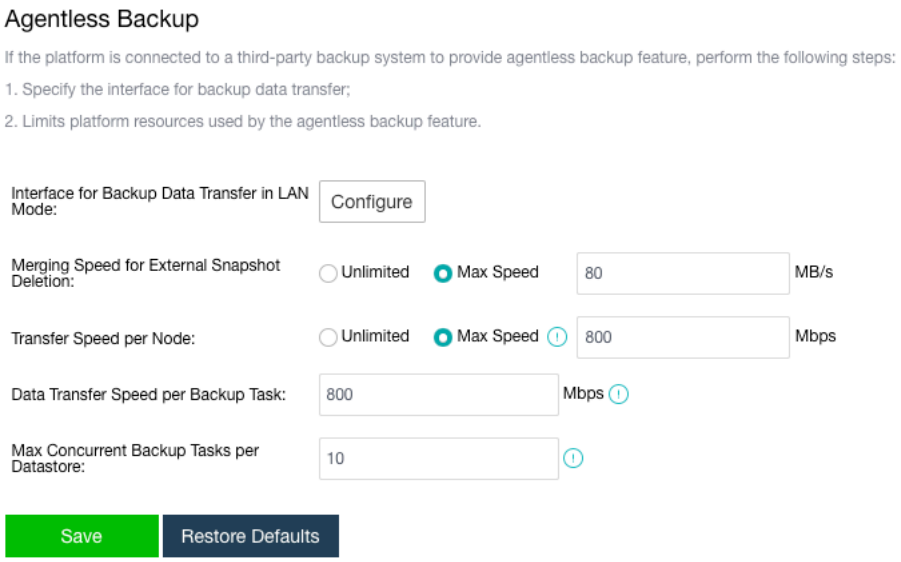


Figure 80: Configure Agentless Backup

3. Go to **System** > **Administrators and Permissions** > **Permissions** > **New** to create permissions named **Agentless Backup**, select the permissions as needed, and click **OK**.

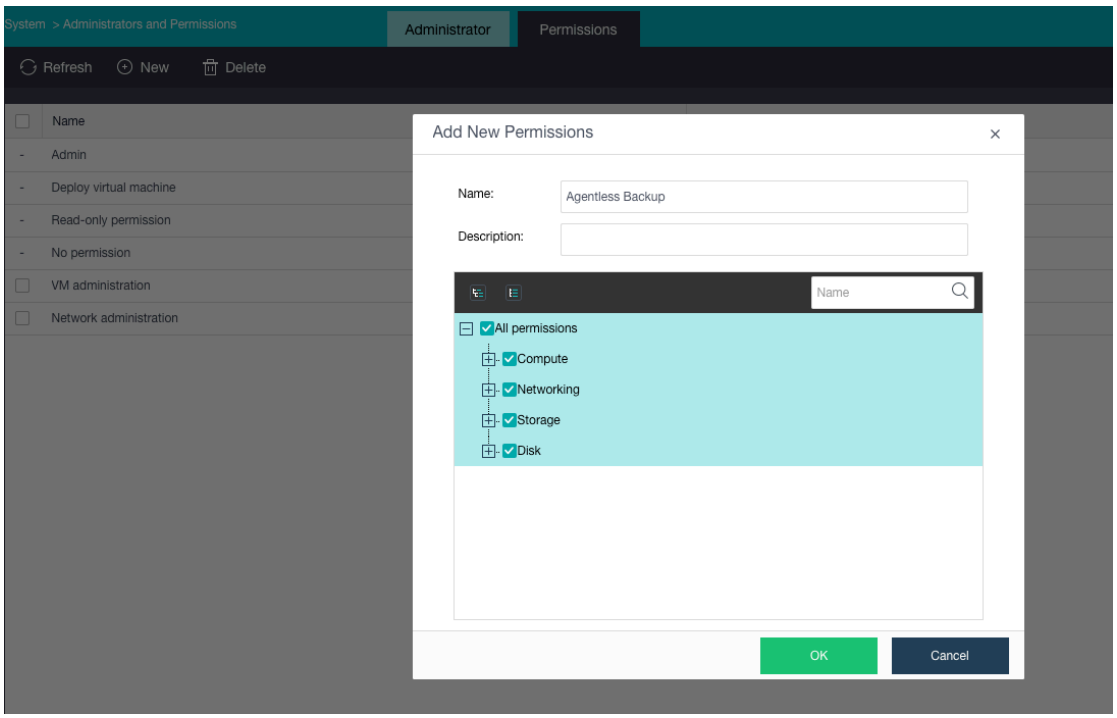


Figure 81: Create Permissions

4. Go to **Administrator** > **Add New Account**, select **System Admin** for **Role**, complete the basic information, and configure system permissions, resource permissions, and quotas.
5. Use the created account to back up and recover VMs by following the third-party platform's instructions.

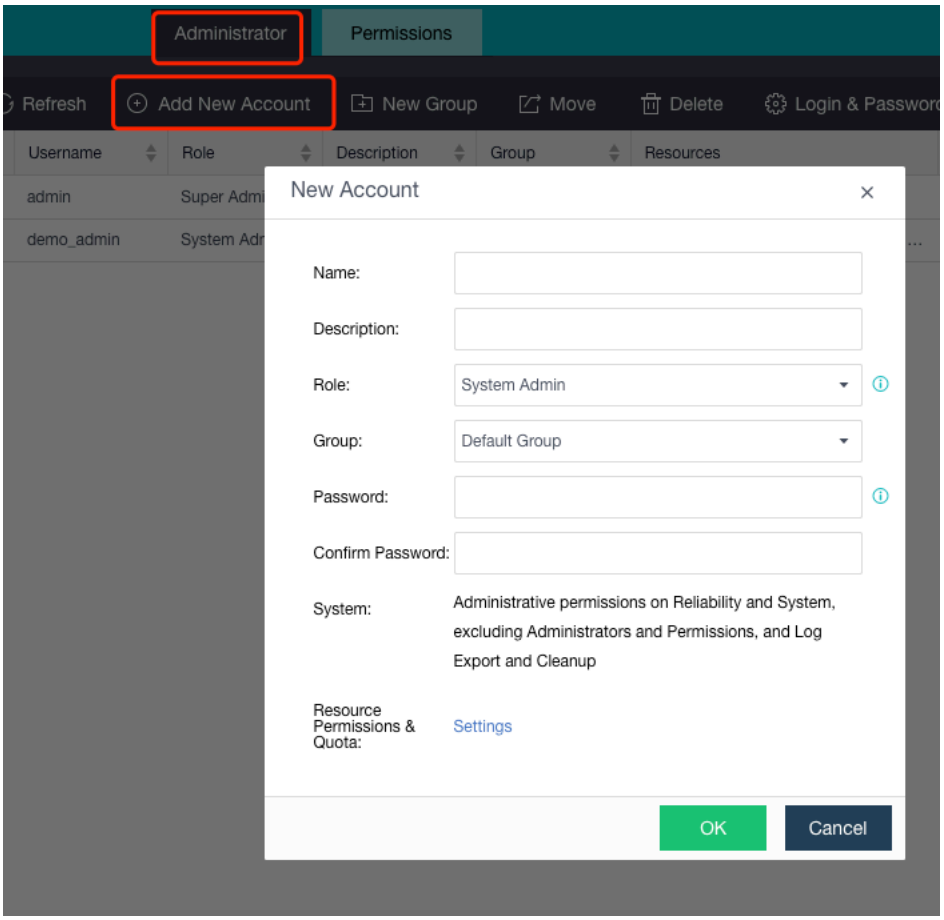


Figure 82: Add New Account

Precautions:

1. When a VM has a backup policy, CDP policy, or disaster recovery policy configured on SCP and there is an ongoing relevant task on SCP, the third-party backup task will be interrupted. It is recommended to configure either the platform backup feature or the third-party backup feature for a VM.

2. It is not supported to perform agentless backups on component-based VMs, VM templates, NFV VMs, placeholder VMs, VMs with Turbo mode enabled, and VMs that have internal disk-based snapshots.
3. It is not supported to perform agentless backups on physical disks (including shared disks of Oracle RAC), or USB disks of VMs.
4. It is not supported to perform agentless backups on storage mounted on VMs, such as iSCSI, and NAS storage.
5. An encrypted VM can only be recovered by overwriting the original VM because the call to the SFVDDK interface (sfdisklib_read) returns ciphertext data.
6. For more information, refer to the ***SANGFOR_HCI_6.10.0_User Manual_EN***.

5.1.4.7 CDP

Generally, common hour-level backup is performed, which indicates that at least 1-hour data loss is inevitable in the event of a business failure. Such data loss is unacceptable for some critical businesses. Some critical business systems require a more granular backup approach, with the recovery point objective (RPO) measured in minutes or seconds. CDP is therefore implemented to meet this requirement.

CDP is an advanced backup technology for I/O in seconds. Data protection is implemented by recording every data modification made by you and restoring data with zero data loss. To optimize CDP technology for enhanced data protection, HCI incorporates the CDP module at the QCOW2 file layer of the VM instead of using the traditional approach of embedding CDP software as an agent at the guest OS layer. HCI provides a cost-effective, easy-to-deploy, and high-value CDP solution that is more suitable for VM environments.

CDP is essentially a backup technology consisting of data backup and restore. The following describes the principles from the two perspectives:

Backup:

CDP backup data comprises restore point (RP) log files and backup point (BP) backup files. The backup process is primarily based on the generation of BPs

and RPs.

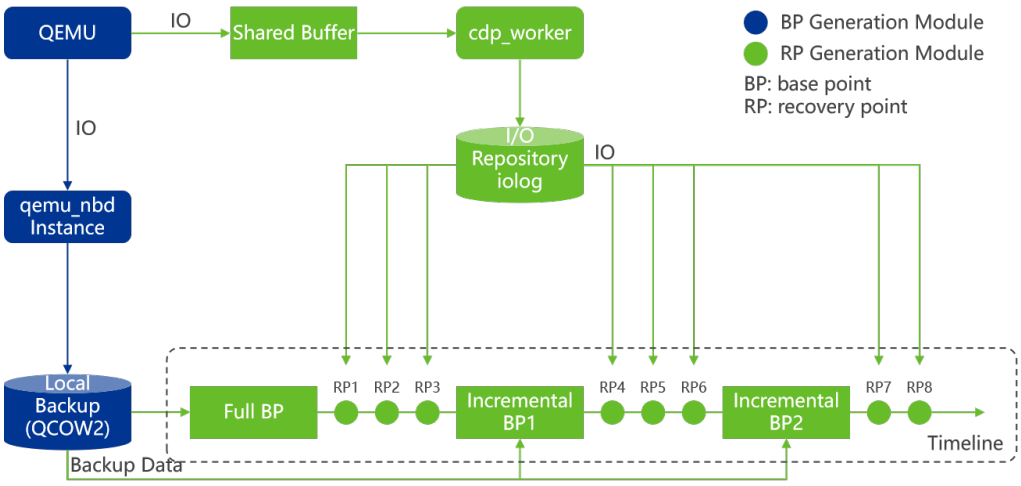


Figure 83: CDP Backup Process

1. During VM startup, both the business (QEMU) and CDP (cdp_worker) are enabled. The shared cache is allocated for them on the VM, with a default size of 2 GB.
2. cdp_worker is located at the QCOW2 file layer of the VM and can directly issue backup instructions for the QCOW2 file to generate full and incremental BPs. The initial enabling of CDP requires a full backup for BP generation, and subsequent BPs are generated periodically based on the configured periodic backup frequency (hour-level). The BPs are stored as backup files in the CDP data store.
3. Using a bypass architecture and I/O traffic distribution technology, I/O data is asynchronously copied from the main business to the CDP data store (I/O repository) through the shared cache and exists in the form of I/O logs. RPs are generated periodically based on the configured I/O log interval (the RPO is 1s/5s). However, RPs alone are not enough for data restoration. All RPs depend on their corresponding BPs.
4. The generated BPs and RPs are all timestamped to facilitate RP locating in case of failure.

NOTICE

1. The I/O log data store and the CDP data store can be at either the same location or different locations.

2. If the VM has used the common backup function, the previously generated backups can be directly used as BPs for CDP.

3. Multi-disk data consistency check is supported to ensure that each RP is valid.

Backup Consistency:

1. CDP storage has three disks, where an RP with an ID is generated for each I/O operation. RPs that have the same ID on these three disks are considered consistent RPs.
2. For example, in the following figure, the RP whose ID is 3 exists on all three vdisks. As a result, RP3 is a valid consistent RP that can be displayed on the page for VM data restoration.
3. However, the RP whose ID is 6 exists only on vdisk 1 and vdisk 3. As a result, RP6 is an invalid consistent RP that cannot be displayed on the page for VM data restoration.

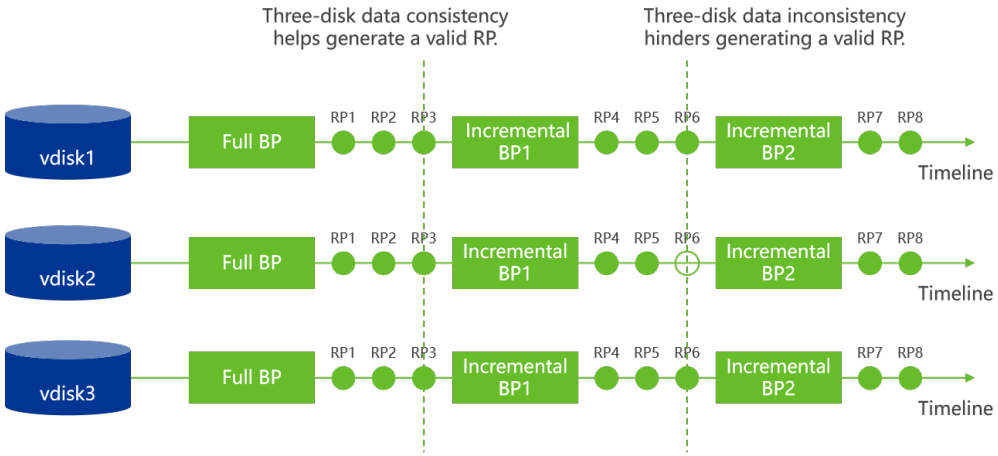


Figure 84: CDP Backup Consistency

Multi-disk data consistency check is required to ensure that each RP is valid.

Restoration:

In the scenario where a business failure occurs (such as virus attack and

deletion by mistake) and CDP backup files are required for VM data restoration, the backup files can be found according to the timestamps. Through incremental backups and innovative design of I/O repositories, quick browsing of backup files is supported for file-level and business-level data restoration.

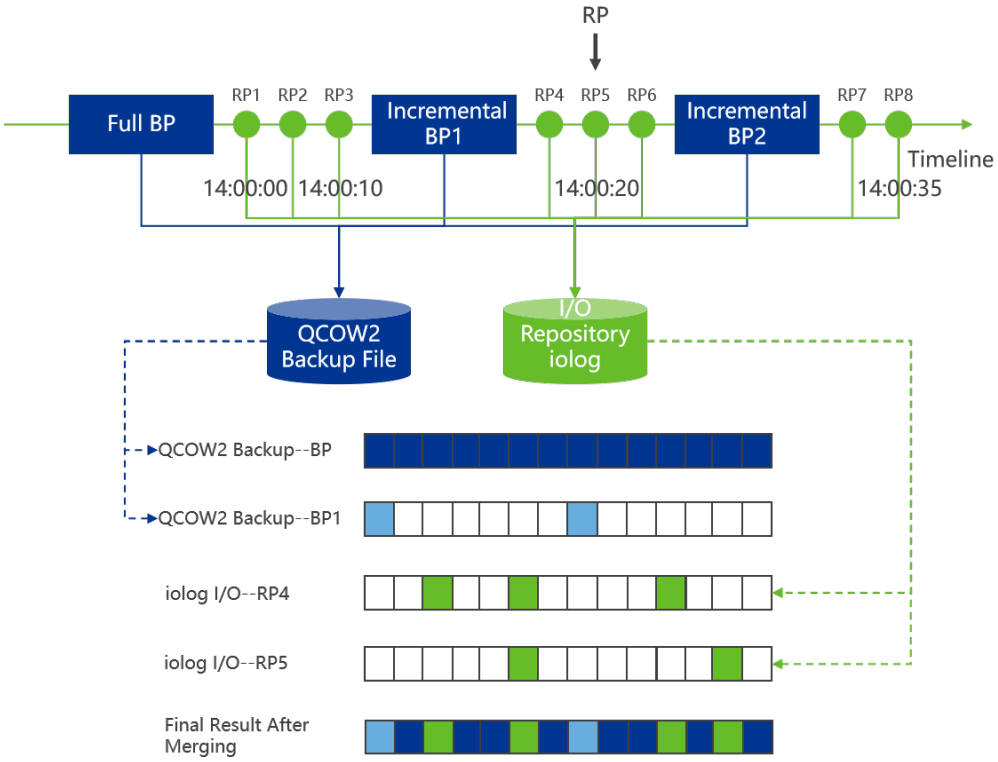


Figure 85: CDP Backup and Restore

As shown in the above figure, CDP backup files include three BP backup files (BP, BP1, and BP2) and eight RP log files (RP1 to RP8). When a specific RP is selected for restoration, a data merge is performed based on all BPs before the target RP and all RPs between the most recent BP and the target RP for business data restoration.

For example, RP5 is selected for restoration. The following describes the key steps:

1. The endpoint page sends a request to browse the I/O log (RP5) generated at 14:00:20.

2. All backup files before RP5 are extracted from the QCOW2 backup files, that is, the full BP and incremental BP1.
3. All RPs between the most recent BP and the target RP are extracted from the I/O repository, that is, RP4 and RP5.
4. Relevant backup files and RP files obtained are merged to obtain the final result for that BP.
5. You can restore specified documents or data for the entire VM at that moment by retrieving files.
6. In addition, if Sangfor CDP fails, business operations are not affected.

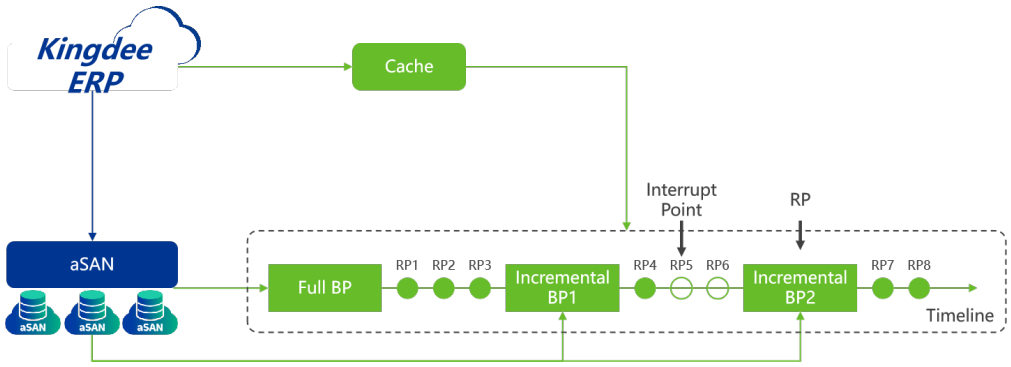


Figure 86: CDP Failure

CDP backup and restore in the Kingdee ERP system are taken as an example.

1. The production baseline backup is performed during the initial full backup. Each I/O operation is recorded for RP generation, and incremental BPs are generated periodically.
2. CDP fails after the generation of RP4. However, the business operation is normal.
3. CDP is normal after the generation of RP6, and a BP (incremental BP2) is created immediately to ensure the accuracy of subsequent backup data.

NOTICE

RP5 and RP6 are lost at this time, but other RPs remain intact and valid.

Functions and characteristics of HCI CDP:

1. Every I/O operation is recorded without any data loss. Therefore, VM data restoration can be performed from any I/O log record point with an RPO of 5s/1s (configurable).
2. The CDP function is independent of the VM QEMU business and uses an asynchronous caching mechanism. The I/O copy process has essentially no impact on business performance and no impact on business stability.
3. Multi-disk data consistency check is provided for VMs to ensure the integrity and reliability of backup data.
4. During the startup of a VM, CDP can be enabled for business operations before CDP data sync is completed. The VM I/O data backup is also performed.
5. The CDP function supports quick browsing and retrieval of backup files, making it easier for users to quickly obtain data files from the CDP backups. Detailed audit logs and permission control are also provided to ensure the security of this function.
6. It is integrated into HCI and delivered together for operation on VMs, which eliminates the need to install third-party software or plugins.
7. CDP is flexible in use. CDP can be enabled on specified VMs and can be enabled or disabled dynamically in HCI. For example, the CDP function can be enabled on a running VM without restarting or interrupting business operations.
8. The CDP function is easy to use and has no learning curve. It facilitates stable and reliable IT innovation for customers.

5.1.4.8 One-Click Data Clearing

HCI provides protection for residual data, which can be enabled by users to ensure complete data deletion and data security. HCI also supports setting the erasure rate and number of deletions. The actual erasure rate is dynamically adjusted according to storage I/O performance and usage.

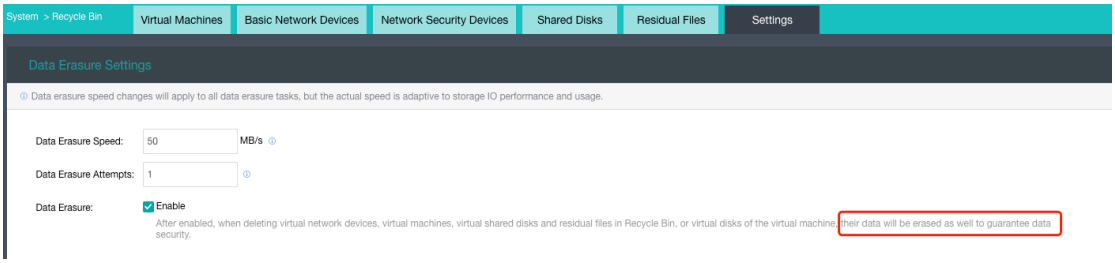


Figure 87: One-Click Data Erasing

5.1.4.9 Silent Operation

Some critical business systems, such as databases, ERP, OA, HIS, and PACS, need to ensure that all data from the OSs and applications are written into disks at the time of VM snapshots and backups. This avoids data repair (data corruption or loss may occur) because cached data is not saved to disks during VM data restoration.

Silent operation ensures the consistency of the file systems and applications. Prior to backup, applications and OSs write dirty data into the disk and enter a temporary suspended state for data consistency. When restoring data to this state by using snapshots or backups that have undergone a silent operation, both the OSs and applications feature normal startup and operation. Sangfor HCI 6.9.0 supports the silent operation function, which can be enabled in the following scenarios where the silent operation option is supported:

1. Backup: including manual backup, batch backup, and automatic backup policies
2. CDP function: baseline backup of CDP
3. Snapshot function: including manual snapshots, bulk snapshots, and snapshot policies.
4. Private cloud DR function

Principles:

- Silent operation on the Windows OS

Microsoft has provided the Volume Shadow Copy Service (VSS) since the Windows 2003 OS to support the backup and copy of system and application

data. The VSS supports three role components, that is, Requester, Provider, and Writer when providing services for external systems. Requester can request VSS to create a volume shadow replica, and the VSS uses Provider to create a snapshot. During creation, I/O refresh is blocked, and data is cached for the silent effect on the volume and OS.

- Silent operation on the Linux OS

Since version 2.6.32, the Linux kernel has provided the file system freeze function, which can be used to block all write operations to the file system, thus maintaining a stable disk environment. This function is designed to create snapshots and backups that ensure the file system consistency. This function is based on the ioctl operation on the file system mount point and is encapsulated as the **fsfreeze** command in the util-linux project.

- Silent operation for VMs in HCI

Before the creation of VM replicas or snapshot data, the host sends a system freeze request to the guest through vmTools. The system is silenced. After the creation, the host sends an unfreeze request to the guest. The guest is unsilenced.

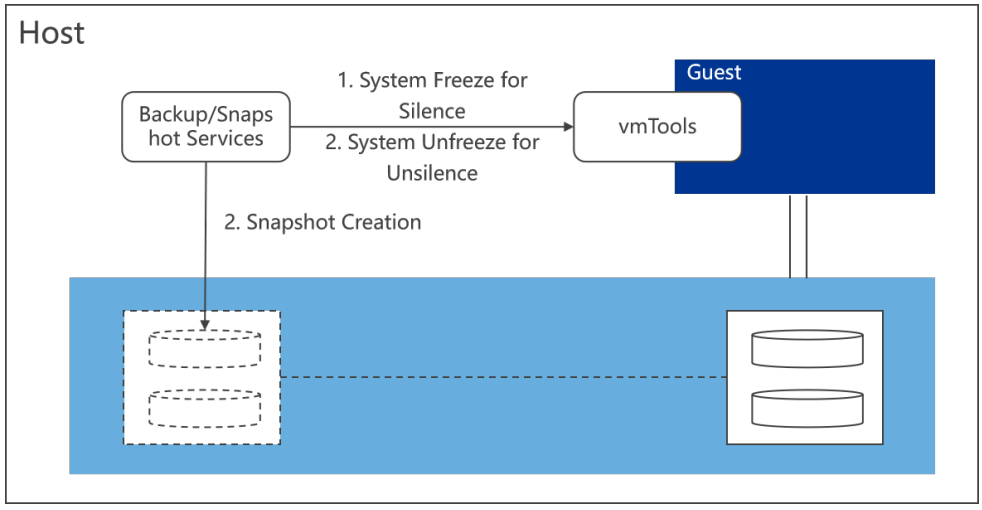


Figure 88: Silent Operation for VMs

5.1.5 Fault Prevention and Handling

Sangfor aSV provides various functions for fault prevention and handling,

including process watchdog, black boxes, VM restart upon an error, HA, subhealthy hosts, host maintenance mode, secure replacement of system disks, USB mapping between hosts, UPS power loss protection, and system file backup and restore.

5.1.5.1 VM Restart Upon an Error

Scheduling failure at the application layer (such as a blue screen or black screen) is a common business interruption issue in VM guest systems. For example, a blue screen is the most common failure at the application layer encountered by Windows OS users. It is often caused by disk failures, driver errors, CPU overclocking, improper BIOS settings, and software viruses. When such failure occurs, Windows terminates all business operations to protect computer data, resulting in a system crash. In this case, a message indicating a blue screen is displayed. Similarly, a black screen is another type of business system failure due to problems with GPU drivers, pirated software, or software viruses, which cannot be automatically recovered.

In the event of these failures, Windows does not automatically restart, and manual intervention by an administrator is required for business restart. This not only leads to uncontrollable business interruption time but also increases the O&M workload for administrators. In unattended scenarios with no administrator, the time for business recovery becomes unpredictable, which is unacceptable for enterprises sensitive to business continuity.

To address this issue, HCI provides the function of VM restart upon errors, which helps administrators in automated O&M and provides better support for the continuous operation of VM businesses.

Sangfor vmTools installed on a VM sends a heartbeat message to the host where the VM is running every several seconds. The host assesses the status of the heartbeat, disk I/O, and network traffic to determine whether scheduling at the application layer of the guest system fails. If the failure persists for several minutes, it is considered that a black screen or blue screen occurred on this VM. The VM is then shut down and restarted.

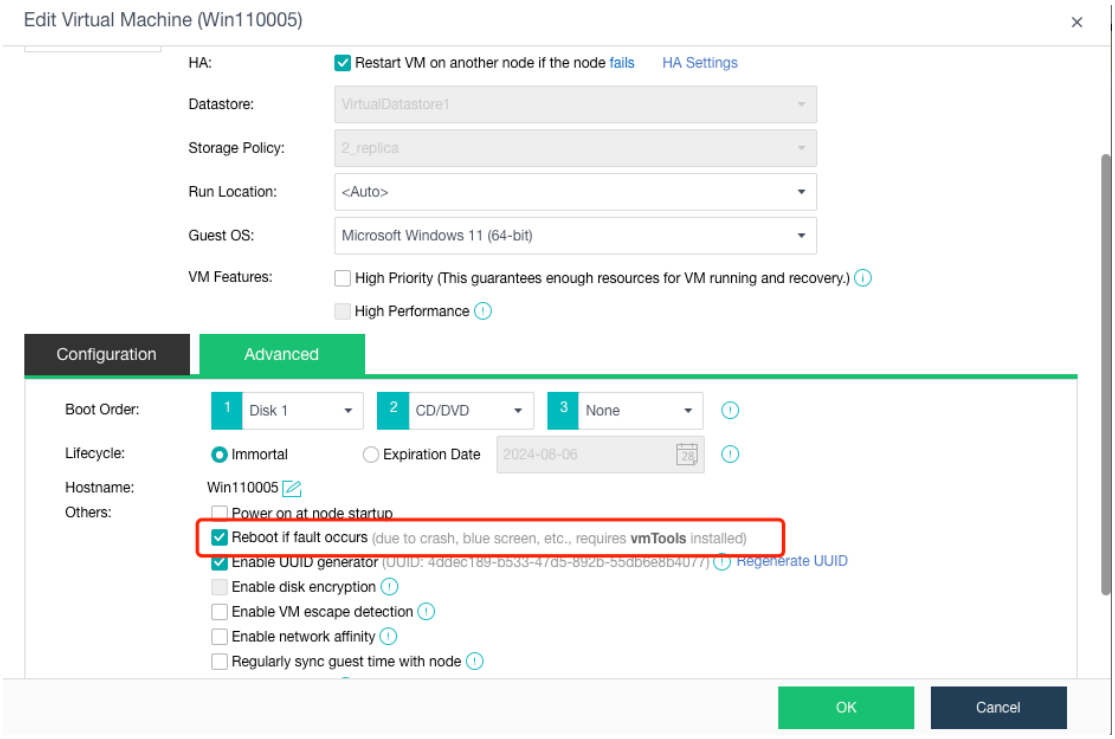


Figure 89: Configuring VM Restart Upon an Error on the Page

Administrators can enable this function by selecting **VM Restart Upon Error** on the VM editing page.

5.1.5.2 HA Mechanism for VMs

1. Provides anomaly detection covering multiple layers such as hardware, platform, and application to comprehensively safeguard business service reliability.
2. Provides proactive HA detection and handling, which can detect risks, analyze the levels of their impacts on running services, and migrate services to healthy nodes in time.
3. Provides live migration and recommends the optimal destination node based on the scheduling engine to minimize the impact of business service interruption.
4. Provides HA failure detection, which can warn users before the HA mechanism fails to ensure the effectiveness of the HA mechanism and business service continuity.

5.1.5.2.1 HA Capabilities

Architecture:

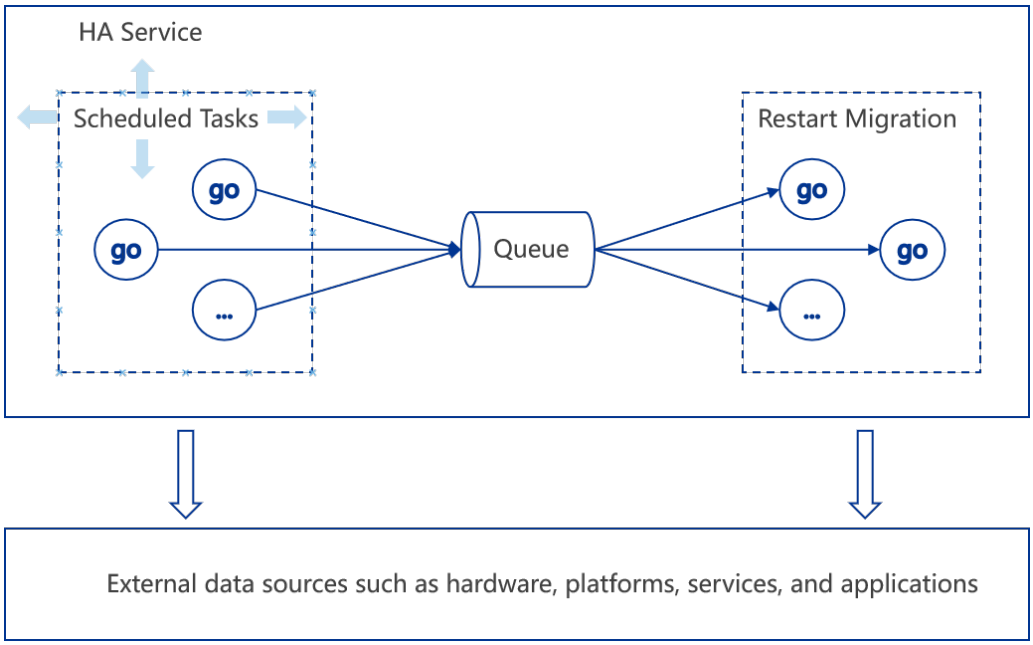


Figure 90: HA Architecture

Principle:

HA 2.0 consists of the detection and handling modules on the whole. The detection module proactively detects the occurred or potential failures based on scheduled tasks and prioritizes the HA services of VMs through internal delay queues and queues with high priority. After passing the decision of a queue, a go coroutine is generated for each VM for anomaly handling. Recovery between VMs does not interfere with each other.

Anomaly Detection and HA Handling:

The administrator can set the detection thresholds and take the handling method corresponding to the cause of the anomaly according to the VM service requirements as recommended below:

- To prioritize business continuity, set the detection thresholds to be as insensitive as possible, and prefer live migration for handling.

- To prioritize data consistency, set the detection thresholds to be as sensitive as possible, shorten PROs and RTOs to quickly recover business services, and prefer aggressive VM restarts for handling.

Anomaly	Cause	Effect	Detection Mechanism/Threshold	Handling Method	Handling Details
Hardware Failure	CPU/Memory/System Disk/Motherboard Failure	VMs are shut down due to the host restart or disconnection.	The physical host is offline and the power statuses of the VMs on it are On in the Status Center, then the VMs are abnormal.	VM Restart	Restart VMs on other hosts.
	Encryption Card/GPU Failure	VMs are running abnormally.	Scheduled hardware traversal for detecting VMs	VM Restart/Ignore	Restart VMs on hosts with redu

			associated with failures.		ndant resources.
Hardware Unhealth	Low CPU Microcode Version	The host may fail.	Hardware detection at power-on or per 30 minutes (Intel only).	Live Migration/Ignore	HCI proactively analyzes impacts and automatically performs live migration.
	High CPU Temperature	The host may be damaged.	The temperature reaches 90 °C for more than 10 minutes (settable).	Live Migration/Ignore	HCI proactively analyzes impacts and

					automatically performs live migration.
	ECC Memory Error (x86/c86)	The available memory is reduced.	The isolated memory exceeds 20% of the total memory by default (settable).	Live Migration/Ignore	HCI proactively analyzes impacts and automatically performs live migration.
	System Disk Lifetime Below Threshold	The host may be offlin	The remaining system disk lifetime is	Live Migration/Ignore	HCI proactively analy

		e.	below 12% by default (settable).		zes impacts and automatically performs live migration.
	Only One Power Module Available	The host may be offline.	Hardware detection at power-on or per 30 minutes (Intel only).	Live Migration/Ignore	HCI proactively analyzes impacts and automatically performs live migration.
Netw	Management Network Disconnection	The	Detection	Ignore	Ignor

ork Disconnection		VM is unaffected by management page failures.	and alert by Status Center.		e.
	Business Network Disconnection	The outbound traffic of the VM business network is interrupted.	The network is disconnected for 5 minutes by default (min: 10 seconds).	Live Migration/VM Restart/Ignore	Quickly restart or live migrate the VM to another host.
	VXLAN Network Disconnection	The east-west traffic is interrupted.	The network is disconnected for 5 minutes by default (min: 10	Live Migration/VM Restart/Ignore	Quickly restart or live migrate the VM

			seconds).		to another host.
	Storage Network Disconnection	The VM system fails.	The network is disconnected for 5 minutes by default (min: 10 seconds).	VM Restart	Restart the VM on the host where the secondary replica is stored.
	aDesk Communication Network Disconnection	The aDesk cannot be accessed.	The network is disconnected for 5 minutes by default (min: 10 seconds).	Live Migration/Ignore	Quickly restart or live migrate the VM to another host.

Host Unhealth	Frequent Host Crashes	The business services fail.	The host is crashed 3 times in a day by default (settable).	Live Migration/Ignore	HCI proactively analyzes impacts and automatically performs live migration.
	Low Host Reliability	The host hardware reliability decreases, with many abnormal hardware	The reliability score is lower than 30, which is deducted based on the frequency and type of memory, disk, and	Live Migration/Ignore	HCI proactively analyzes the level of impacts on running

		indicators detected.	RAID card failures on the host.		business services and migrates them to healthy hosts before they are severely affected.
VM Anomaly	VM Suspension due to Process Error	The VM business services are interrupted.	VM running status detection per 5 seconds.	VM Restart	Restart the VM on the current host.
	VM Suspension due to I/O Error (Non-RDM Disk)	The VM busin	VM I/O status detection	VM Restart	Restart the VM

		ess servic es are interr upted .	per 3 seconds.		on the curre nt host.
	VM Blue Screen	The VM busin ess servic es are interr upted .	VM status detection for 60 seconds per 3 seconds.	VM Restar t	Resta rt the VM on the curre nt host.

Table 8: Anomaly Detection and HA Handling

VM RightSize Prediction:

The free memory of the host that a VM migrates to through the HA mechanism cannot be less than the configuration memory of the VM; otherwise, the VM will fail to restart. For example, a VM with a configuration memory of 16 GB cannot start up on a destination host whose remaining memory is only 12 GB.

HCI 6.10.0 provides the VM RightSize prediction capability, which can analyze the business resource needs by collecting the VM memory information before triggering HA, and recommend the optimal VM specifications to effectively reduce HA failures caused by insufficient cluster resources or high VM specifications.

RightSize = Required Memory + (Configuration Memory - Required Memory) * 0.25

Required Memory is the estimated maximum memory required in six hours.

As in the above example, the available memory of the destination host is less than that required by the VM configuration but more than the RightSize of the

VM which is 10 GB, therefore the VM can be restarted successfully.

NOTICE

The RightSize prediction capability is only available for VM startups triggered by HA, not for ordinary startups and migrations triggered by HA of VMs.

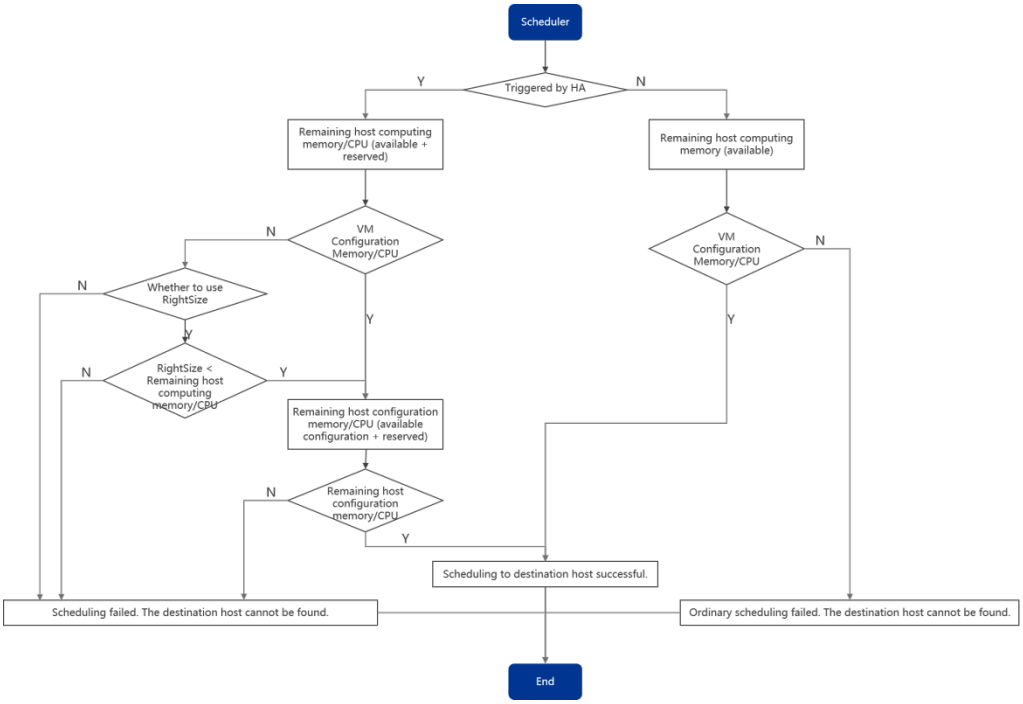


Figure 91: RightSize Prediction Capability

Live Migration for Another VM:

When a VM fails, the free resources (memory and CPU) of the host to which the VM is migrated through the HA mechanism cannot be less than the configuration resources (memory and CPU) of the VM. If the cluster resources are insufficient or the VM configuration resources are too much, the VM may be unable to fail over to a suitable host. HCI 6.10.0 provides intelligent scheduling algorithms, which can organize the resource fragments to perform live migration on another VM to provide a running location for the failed VM.

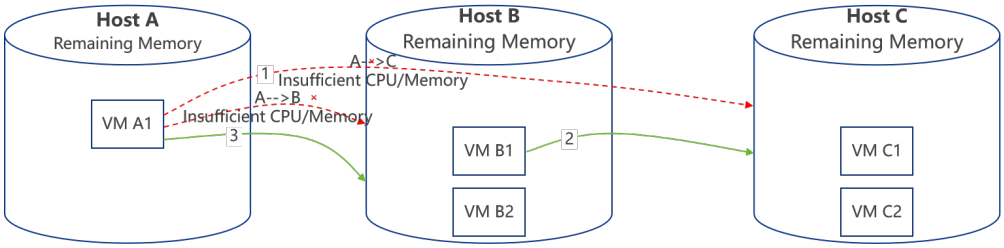


Figure 92: VM Live Migration

Up to two live migrations can be performed on another VM by HA:

- Two Live Migrations in Chain: VM C1: C > D, VM B1: B > C, VM A1: A > B
- One Live Migration: VM B1: B > C, VM A1: A > B

NOTICE

Live migration can only be performed in scenarios with health issues where scheduling fails.

HA Scheduling:

As shown in the following figure, the principle of HA scheduling is to schedule to the optimal host that meets the requirements for running a VM, different from ordinary power-on scheduling only in the following two aspects:

1. HA scheduling uses reserved CPU and memory.
2. HA migration to unhealthy hosts is not possible.

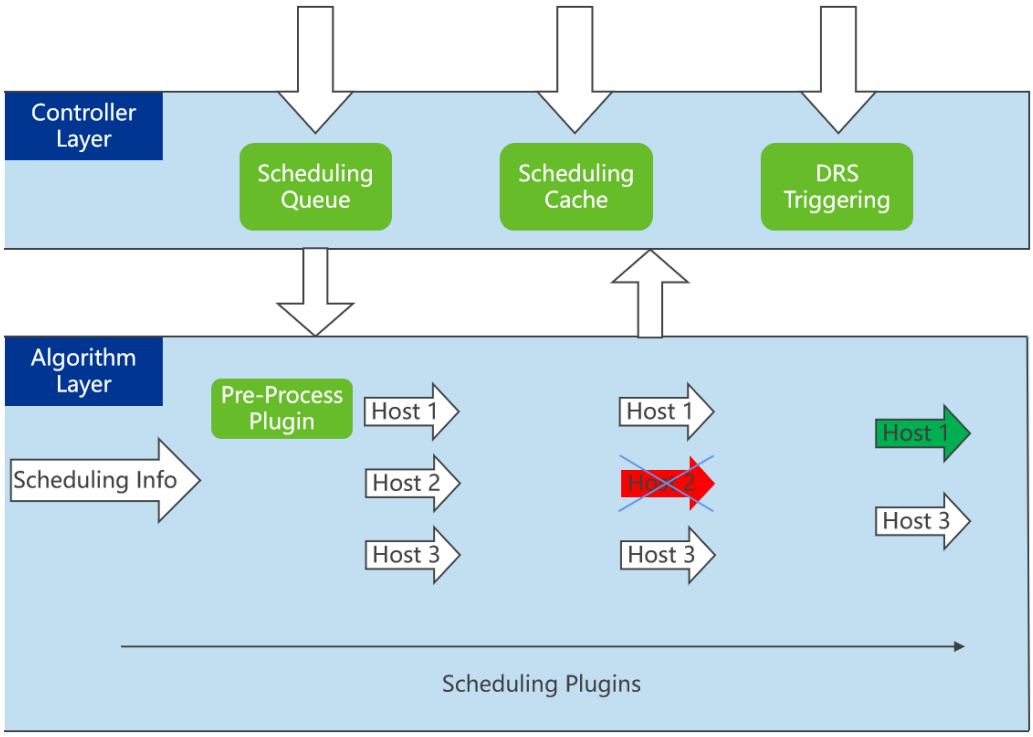


Figure 93: HA Scheduling Principle

5.1.5.2.2 Advanced Features

Live Migration Response:

You can flexibly set the **Migration Schedule** based on the needs of the production environment.

Applicable Scope: HA scenarios configured with **Live Migration for Business Recovery**, such as scenarios with health issues.

UI Path: **Reliability** > **HA** > **Advanced Settings** > **Live Migration Response**.

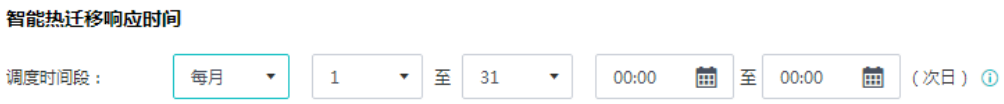


Figure 94: Set Migration Schedule

Resource Reservation:

Resource reservation refers to reserving certain resource space on some hosts, and these resources will not be used by the system or other VMs. When a host fails or a disaster recovery switchover occurs, the reserved resources can be

used to recover VMs immediately, reducing the risk of business recovery failure due to insufficient resources.

Host Unhealth:

HA 2.0 provides the host health check feature to fully cover scenarios such as high CPU temperature, only one power module available, low host reliability, and low CPU microcode version.

- In scenarios with health issues, VMs must be migrated to healthy hosts through the HA mechanism unless none of the hosts are healthy. If HA triggered by a health issue only performs migration from an unhealthy host to another unhealthy host, then it is essentially a proactive defense that does not eliminate the health risks.
- In failure scenarios such as GPU or encryption card failure, or network disconnection, VMs are prioritized to fail over to healthy hosts through the HA mechanism to prevent business services from being impacted. If none of the hosts are healthy, the VMs will fail over to the unhealthy hosts to prioritize business continuity because a failure is more severe than a health issue. After failover, the VM business services on the unhealthy hosts may not be affected.
- Ordinary VM startups or migrations performed by users are prioritized on healthy hosts. If none of the hosts are healthy, the VMs will be started on the unhealthy hosts to ensure that they can be recovered as much as possible.

HA Mechanism Alerts:

1. Common Hardware Failures

Due to the lack of technology, experience, and energy, customers cannot detect hardware risks in time or even handle them in advance.

2. HA Failure

The HA mechanism is often used to ensure business continuity in case of VM failures. However, if the HA fails, VMs will be unable to recover from failures, and services on them will be interrupted. HA 2.0 provides HA failure detection, which enables users to take preventive measures by providing alerts in

advance.

3. Critical HA Service Error

It is recommended to contact a Sangfor technical support representative as soon as possible when the proactive HA system service encounters an error.

4. Insufficient Reserved Resources for HA

When the reserved resources for HA are insufficient to guarantee the VM recovery using the redundant resources on HCI, to ensure the high availability of the business, it is recommended to increase the reserved resources or shut down some unnecessary VMs to release the resources.

5. External Storage Accessed by Only One Host

A single path can lead to VM HA failures that prevent HCI from quickly recovering business services during the host failure. It is recommended to add physical hosts that can access the storage or recover the corresponding paths.

6. Only One Replica Available

Business services on a VM with only one available replica may be unable to be recovered through the HA mechanism:

- In extreme cases, the replica and the VM run on the same physical host, but the physical host is crashed or disconnected from the management network.
- In extreme cases, a single point of failure occurs on the disk where the replica resides, and no more replica is available. It is recommended to increase available replicas of the VM as soon as possible.

Visualized HA Results:

1. The HA records of VMs can be viewed on the **Reliability** > **HA** > **HA History** page for tracking convenience.
2. When a VM fails and is not recovered through the HA mechanism, a corresponding alert log will be generated for ease of troubleshooting.
3. The SCP home page displays HA results, such as **HA Responses**, **Proactive HA Responses**, etc.

5.1.5.2.3 UI Display

1. Log in to HCI and go to **Reliability** > **HA** > **Hardware** to select response methods for hardware failures and hardware health issues.

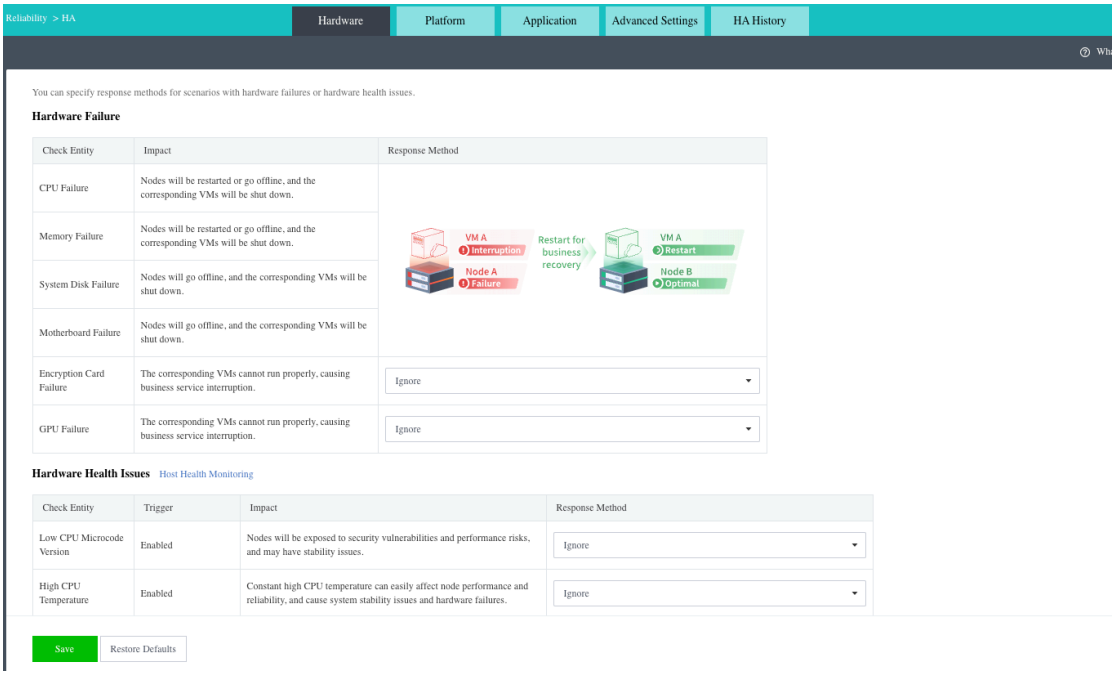


Figure 95: Select Response Method

2. Click the **Platform** tab to select response methods for network disconnection conditions and node health.

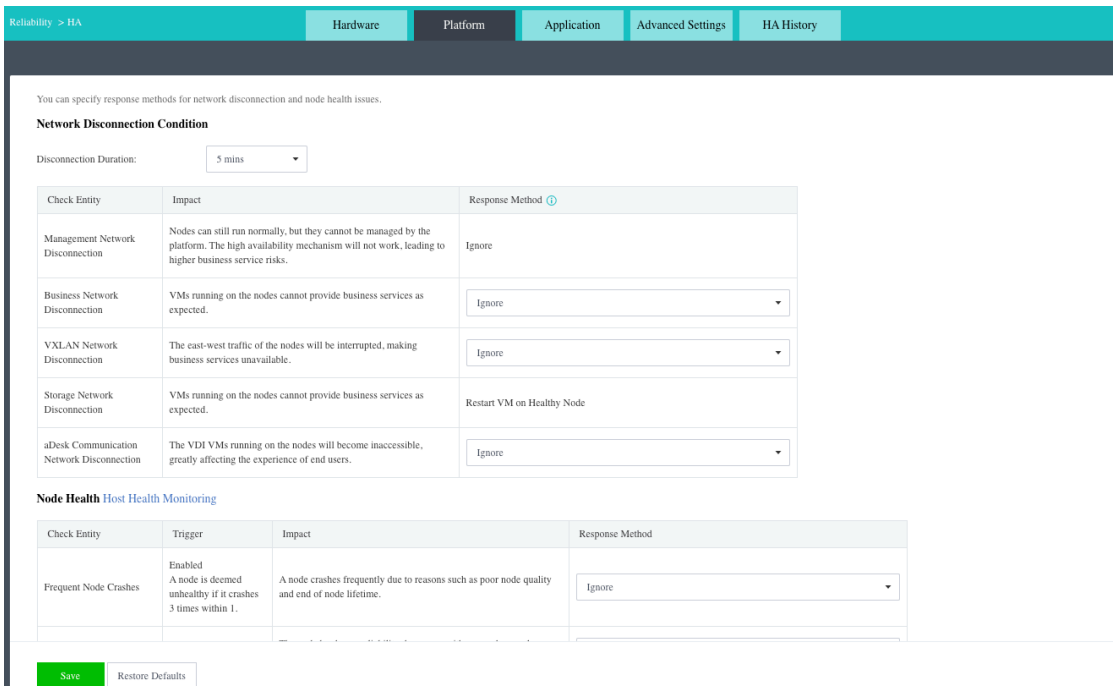


Figure 96: Click Platform Tab

3. Click the **Application** tab to learn that when VMs operate abnormally, HCI can automatically restart them for quick business recovery.

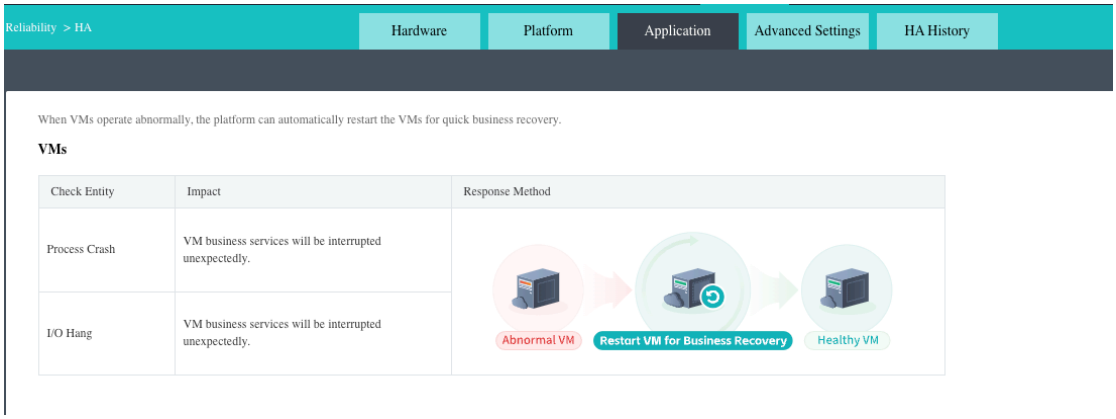


Figure 97: Click Application Tab

4. Click the **Advanced Settings** tab to configure resource reservation, node health monitoring, and live migration response schedule.

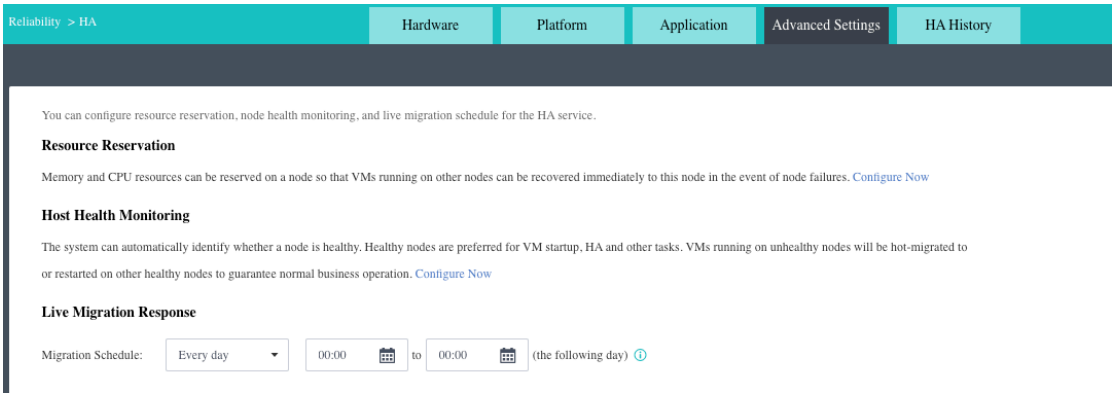


Figure 98: Click Advanced Settings Tab

5. Click the **HA History** tab to view the VM HA records.

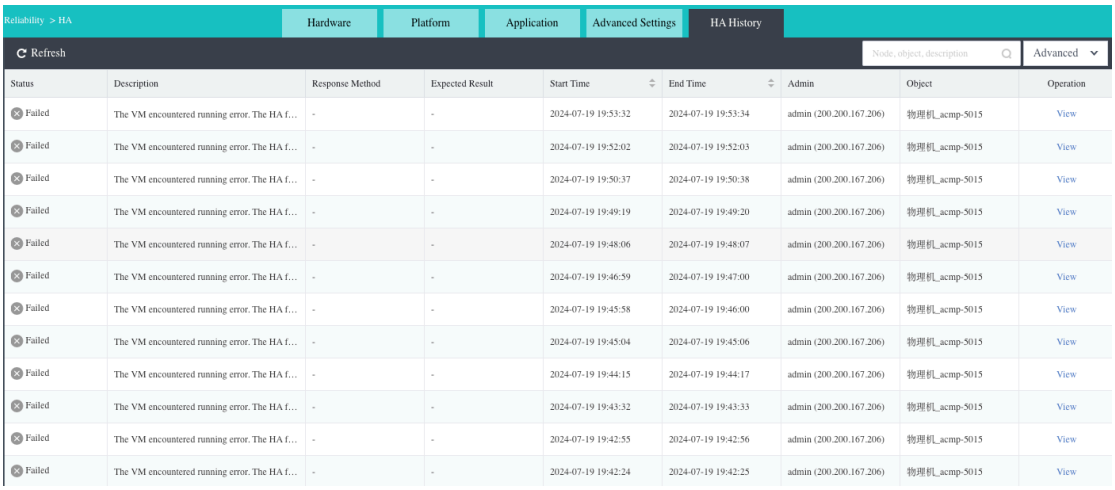


Figure 99: Click HA History Tab

6. Select **Restart VM on another node if the node fails** for **HA** when creating and editing a VM to enable the HA feature.

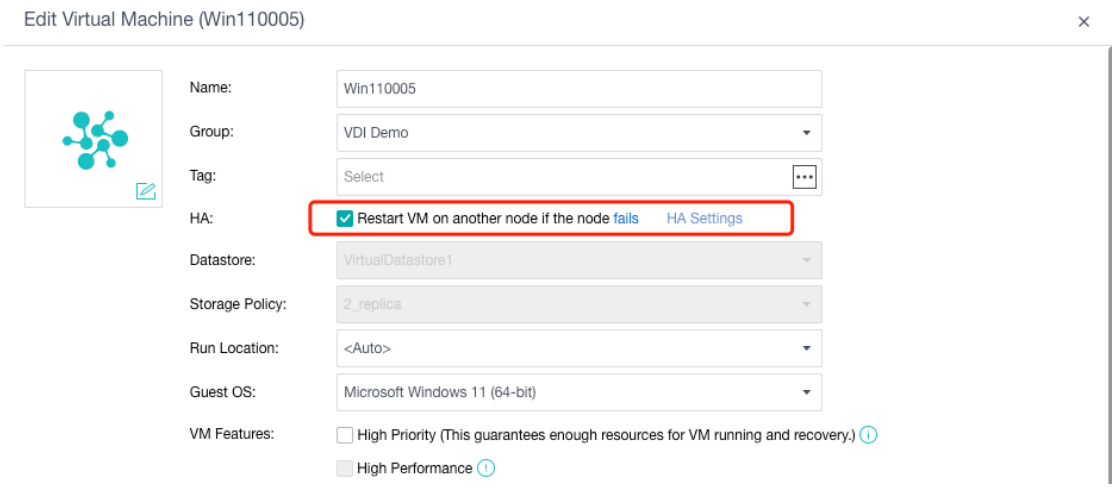


Figure 100: Select Restart VM on another node if the node fails

5.1.5.3 Handling of Subhealthy Hosts

With the handling of subhealthy hosts available, HCI adopts a preventive HA approach to address issues related to subhealthy hosts, which involves checking hosts when they go online. This function is used to identify subhealthy hosts and components by detecting memory failures, system disk failures, and RAID card failures. This prevents faulty hosts or components from entering the cluster and identifies network errors in advance to avoid impacting business operations on the customer side.

The following failure scenarios may occur when a host experiences system suspended animation:

1. Connected driver devices such as keyboards and mice become unresponsive, making it impossible to send commands to the server. However, some processes may appear so active that HCI does not identify the issue as a failure. As a result, no VM HA is implemented.
2. After the host is restarted, the VM HA may fail to meet the requirements for threshold detection, and the VM is rescheduled to the original subhealthy host. This may end up with a cycle of HA.

To address the issues in the above scenarios caused by subhealthy hosts for healthy business operations, HCI adopts the preventive HA function, an optimized edition of HA. This function is used to check the hosts when they go

online and identify network errors in advance. This prevents faulty hardware components from entering the cluster and avoids affecting business operations on the customer side.

Principles and process:

The following figure shows the overall flowchart of preventive HA.

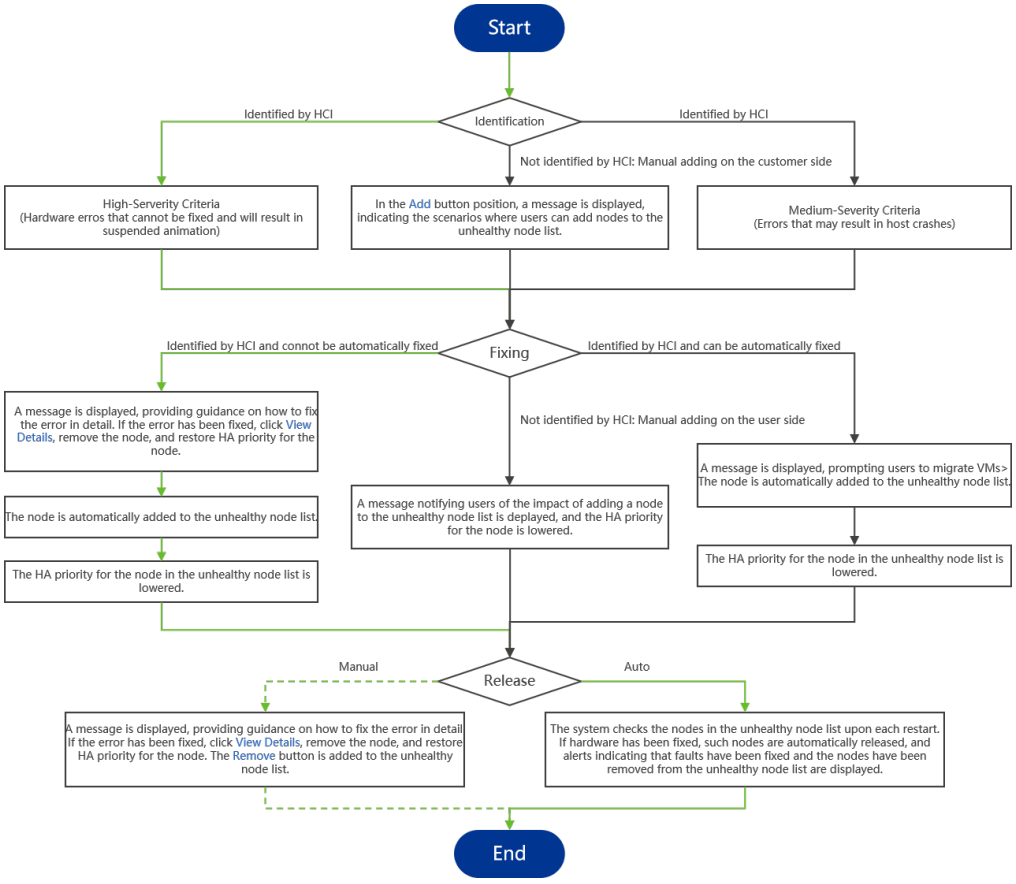


Figure 101: Flow of Preventive HA

Identification of subhealthy hosts:

Memory failure: The correctable errors (CE) and uncorrectable errors (UE) of ECC memory are identified. If such errors are detected 1,024 times per 24 hours, a memory failure occurs.

System disk failure: The system disk is read-only, or the disk lifetime is below 12%.

RAID card failure: The detection of RAID cards is performed through the

detection mechanisms provided by hardware vendors, whose results are collected by HCI after completion.

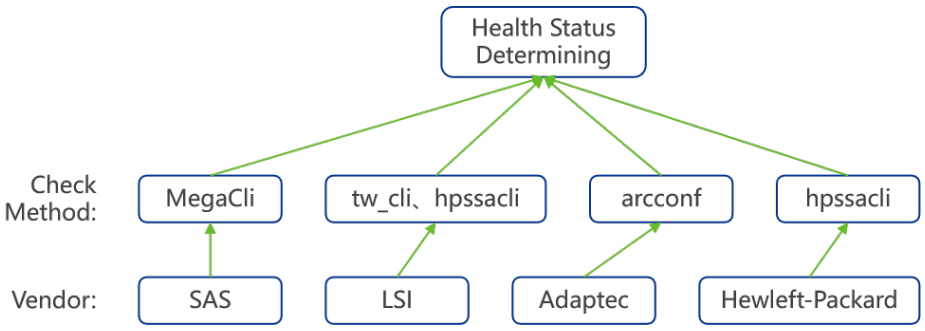


Figure 102: Check Items for RAID Card Failures

Network latency: HCI can detect the network status between hosts and the cluster using the ping command. Bulk detection is supported.

Fixing of subhealthy hosts:

In this phase, faults are divided into the following categories: faults that can be automatically fixed by HCI and faults that cannot. Memory and system disk failures fall into the first category. In response to these faults, a 2+1 approach is provided, which consists of two fixing steps and one pre-message.

The two fixing steps include:

1. Alert: It is recommended that users restart the host and replace the faulty hardware.
2. Automatic adding of the faulty host to the subhealthy host list: The HA priority for the subhealthy host is lowered.

The following describes the logic of priority lowering for subhealthy hosts:

1. Host selection priority by HA: healthy host > subhealthy host
2. Critical VMs are preferentially started on healthy hosts, while non-critical VMs are randomly started based on resource utilization (consistent with the current situation). Host selection for HA and for VM startup share the same scoring logic. As a result, the latter is also influenced by the unhealthy host list, and priority lowering for hosts is performed.

One pre-message

- Alert: Release the host if the error has been fixed.

Release of subhealthy hosts:

- Automatic release

Because neither the system disk nor the memory supports hot swapping, it is necessary to check whether the hosts in the subhealthy host list are fixed during host restart. The hosts are automatically released when they become healthy.

- Manual release

Manual release is adopted in two main scenarios. In the first scenario, it prevents misjudgment during identification and release. In the second scenario, it ensures that hardware status is obtained even in cases where errors such as network interruption occur. The pre-message is provided in the fixing phase.

Precautions:

The handling mechanism for subhealthy hosts only involves migrating the running locations of VMs. After the mechanism is enabled, VMs on unhealthy hosts are migrated to healthy hosts. If the resources required by the corresponding VMs are greater than those available on the healthy hosts, the following approaches are taken:

1. If the VM has a scheduling policy, the VM is preferentially scheduled according to that policy.
2. If the VM has no scheduling policy, the VM is migrated to another relatively healthier subhealthy host.
3. If no healthy host is available in the cluster, the handling mechanism does not take effect.

5.1.5.4 Monitoring and Handling of External Storage I/O Latency

After the external storage is mounted to the HCI cluster, if one of its paths fails due to packet loss or high latency, the IOPS of the LUN where the path is located as well as other normal paths under load balancing will be seriously

affected. The HCI backend can proactively identify and isolate unhealthy links to avoid business impacts caused by their failures. If the isolated path is the primary path in primary/secondary mode, a switchover will be triggered quickly.

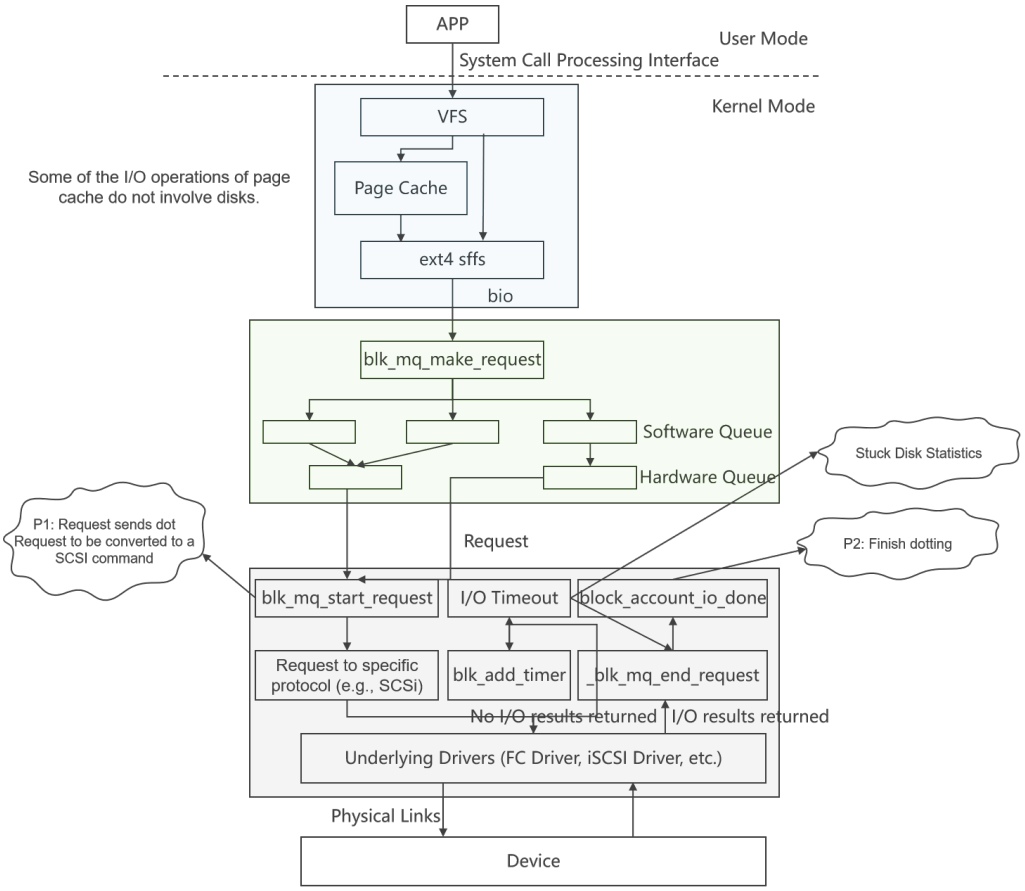


Figure 103: Architecture

5.1.5.4.1 Technical Principles

Count and Alert on Slow Disks:

The time when an I/O (blk_mq_start_request) is submitted to the underlying driver is P1, and the time when the blk_account_io_done function is called to end the I/O (__blk_mq_end_request) is P2. HCI uses the difference calculated from the two timestamps dotted by the kernel in the I/O path to obtain the time spent by this I/O in the underlying driver and device, and then compares that time to the set latency threshold (the value of

/sys/{dev}/queue/io_expected_threshold which can be set individually for disk devices on the **Path Health Settings** page, the default is 512 milliseconds) to obtain the time bucket that this I/O time belongs to, and finally adds one to the count of this bucket.

By adding a three-dimensional array to the hd_struct structure, I/O time buckets of the following threshold multiples are implemented: 0-1/16, 1/16-1/8, 1/8-1/4, 1/4-1/2, 1/2-1, 1-2, 2-4, 4-8, 8-16, 16-32, and 32 times and above.

The distribution of I/O time buckets for each disk partition can be viewed through the /proc/ioslow_stats interface. The service periodically detects the counts of each bucket (The **Duration** of the **Latency** can be selected for 30 seconds, 1, 5, 10, 20, or 30 minutes on the **Storage** > **Other Datastores** > **Storage List** > **Path Health Settings** page, with a default of 10 minutes). If the count of a bucket on an external storage disk is greater than the threshold and increases by more than 50% of the I/O count during the counting period, the disk path will be processed based on the specified policy (**Alert and Deactivate Interface** by default). The maximum number of isolated paths is 50% of the available paths and one available path is always retained. In multi-primary-multi-secondary and load balancing modes, the isolation is performed only when the difference in average latency between all available paths with high latency and those with low latency exceeds 50% of the threshold, while paths in one-primary-multi-secondary mode are isolated once they encounter high latency.

The isolation principle is to first notify the update of the multipath list to set the path unavailable to prevent I/O from being continued to be sent to it and then set the path device to be offline. Disabled paths can be enabled again by clearing and scanning the path device again to initialize it.

Count and Alert on Stuck Disks:

The kernel I/O timeout processing function has been added with logic so that when an I/O does not return within the specified time, the number of stuckness on the corresponding partition will be added by one, and the current time will be recorded. The number of stuckness and the time of the last occurrence for each partition can be viewed through the /proc/iostuck_stats

interface. If an I/O does not return for longer than the kernel I/O timeout (30 seconds by default), it will be set to timeout and synchronized to the slow disk count.

The kernel counts each I/O stuckness. The alert service detects the I/O stuckness number every 10 minutes and reports an alert to remind the user if the number increases. As non-returning I/Os are synchronized to the slow disk count as well, the corresponding handling process will be triggered when a slow disk is identified.

Path Scan:

Paths that encounter I/O errors or are isolated due to unhealthy latency are set to unavailable by the multipathing service and removed by the path scan before the storage is scanned again. The scanning process sends the scsi cmd (report LUNs command (A0)) command to the storage to obtain the LUN ID list mapped to HCI and then sends the scsi cmd (INQUIRY) command to the LUNs in turn to check whether the LUN exists. If it does, it identifies whether the kernel has created the path device. If not, a new one is created.

The path scan can identify and reconnect failed paths on hosts without affecting normal paths, clear abnormal paths that remain from modifying LUN IDs of storage servers or remapping them, and recover paths with high latency that have been isolated due to path health settings.

5.1.5.4.2 UI Display Guide

Log in to HCI, go to **Storage** > **Other Datastores** > **Storage List**, select storage, and click **Path Health Settings** to configure **Path Isolation**.

On the **Storage List** page, click the storage name and then **Nodes and Paths**, and click the number of **Paths (Alerting/Total)** of a node to enable and disable its paths.

Path isolation will trigger an alert on external storage on the **System** > **Alerts** page.

On the **Storage List** page, click the storage name and then **Nodes and Paths**, and click **Path Scan** of a node, or select multiple nodes on the same LUN and click **Path Scan**.

5.1.5.5 Memory Failure Isolation and Recovery

In today's large-scale or small- and medium-sized data centers, the capability to avoid server crashes has always been a critical metric for measuring reliability, availability, and serviceability (RAS), and also a primary consideration for meeting the requirements in the service-level agreement (SLA) between cloud service vendors and customers.

As the foundation of the underlying virtualization resources for the upper cloud layer, HCI boasts the core competitiveness of stability (which is the most fundamental and core requirement) and cost-effectiveness to meet customer needs.

Sangfor HCI OS, based on open-source Linux, features the same level of robustness as the Linux system in terms of RAS.

- Reliability indicates that the system must be reliable as much as possible, without unexpected crashes, restarts, or even physical damage to the system. A reliable system should be capable of performing self-fixing on certain minor errors and isolating errors that cannot be self-fixed as much as possible to ensure that the rest of the system is running properly.
- Availability indicates that the system must ensure that it works for as long as possible without going offline and that minor errors do not affect the normal operation of the entire system. In some cases, even hot swapping is allowed, and faulty components are replaced, thus ensuring that the system downtime is within a certain range.
- Serviceability indicates that the system is capable of providing convenient diagnostic features, such as system logs and dynamic detection, to facilitate system diagnosis and maintenance operations by management personnel. This will lead to early error detection and resolution.

In summary, RAS as a whole is designed to ensure that the entire system operates in a long-term and reliable manner without going offline, and possesses a robust fault tolerance mechanism. RAS is of paramount importance to application environments such as cloud data centers.

The robustness of virtualization platforms represents the core strength of Sangfor HCI, a general solution for cloud data centers. However, a single memory failure often affects the entire host in actual business operations. Memory is a critical component that directly provides cached data for the CPU. The increasing demand for greater memory capacity in standalone mode in customer scenarios results in an extremely higher probability of memory failures. A significant number of system crashes are caused by memory errors on the customer side, which severely affects the quality and reliability of HCI and aServer. Therefore, Sangfor proposes rectification and solutions to eliminate unexpected crashes due to memory errors to enhance system reliability.

Current servers and workstations mostly use the ECC memory. That is, RAM is used for error checking and correction. ECC memory can tolerate and correct memory errors, allowing the continuous normal operation of the system without interruption due to errors.

ECC memory errors are divided into two types: CEs and UEs. A CE in the memory can be automatically corrected by the OS with the consumption of its processing resources. HCI isolates and amplifies the memory address associated with the CE to prevent the CE from escalating into an unrecoverable failure. UE is a type of more severe error that exceeds the error correction capability of the ECC memory. A UE in the memory will directly result in a system crash. Generally, after a UE occurs, HCI reports it directly through the machine check exception (MCE) mechanism triggered when the CPU detects a hardware error. The host is then crashed for restart immediately. This avoids unpredictable risks caused by the UE and reduces the likelihood of a crashed host that cannot restart automatically. However, correcting UEs in this way directly affects customer business operations. In fact, some UEs can be addressed before a system crash.

The new version of HCI uses pre-event, in-event, and post-event processing measures to avoid crashes. In the event of an ECC memory error during system operation, HCI automatically monitors and isolates the memory address associated with the error from the OS. This ensures that the faulty ECC memory is no longer used during system operation, thereby enhancing system stability and reliability.

Pre-event processing: After the device restart, including proactive restart, passive restarts due to hardware failures, and restart after a system crash due to early unrecoverable UEs, hard isolation (hard_offline_early_kill) is performed on identified UE pages before the stable business operation of the host and VMs. This prevents the application layer from reusing the memory regions affected by UEs, thereby preventing potential risks.

In-event processing: When the host and VMs enter the stable operation phase, recovery measures are taken through the OS kernel after the CPU reports a memory failure through the MCE mechanism. This entails killing only the affected application processes or isolating the corresponding faulty pages to kill applications after a UE failure. This will enhance system stability.

Post-event processing involves the following scenario during the stable operation of the host and VMs: A system crash is successfully avoided in the event of an MCE, or the UE addresses are recorded, and soft isolation is performed on idle pages without touching the original data (soft_free_offline) on the application layer service side, and alert notifications are triggered when the kernel ignores noncritical UEs.

5.1.5.6 RAID Card Failure and Recovery

On the live network, a RAID card crash failure in HCI may lead to I/O interrupts and, ultimately, cluster crashes. The business interrupt time can be effectively shortened by monitoring I/O requests, detecting the RAID card crash failure as early as possible, and implementing failure reporting and recovery. HCI of versions earlier than 6.7.0 uses the iostat tool to monitor I/O requests. This way, only the I/O latency and traffic at the block layer can be monitored. However, whether a RAID card failure occurs cannot be determined in zero-

traffic scenarios. In contrast, HCI of the new version uses the extended Berkeley Packet Filter (eBPF) technology to monitor the processing information about I/O requests at the block layer based on the I/O request processing process. This way, RAID card crash failures can be identified. The detected failures are then reported to HCI for appropriate handling.

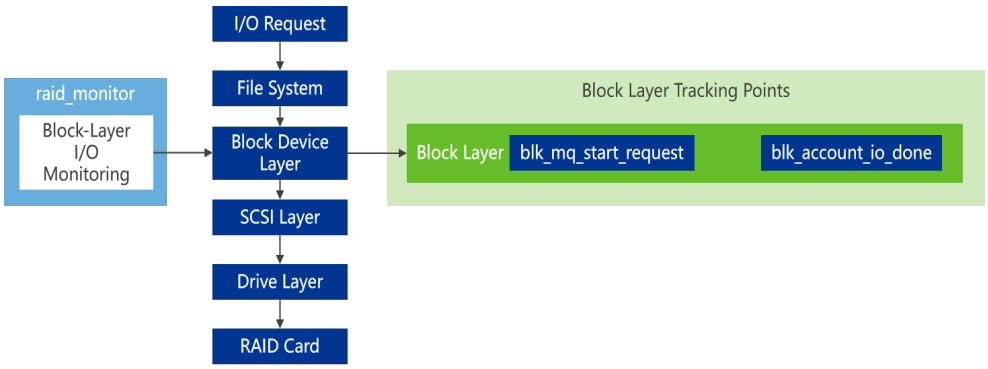


Figure 104: Monitoring Schematic Diagram

The I/O monitoring module can monitor not only I/O but also the reliability of the monitoring program.

I/O monitoring: It is achieved by monitoring the I/O requests at the block layer based on the I/O request process (the above figure shows the monitoring points). The information regarding I/O request execution can be obtained to determine whether a RAID card crash failure occurs.

Monitoring of the reliability of the monitoring program: By using Service_monitor of the monitoring program, HCI monitors the errors in the monitoring program, including the monitoring of memory leaks and the Raid_monitor running process by Supervisor.

Monitoring of memory leaks: Service_monitor periodically collects memory usage data of Raid_monitor. It reports an alert and restarts Raid_monitor if a value exceeding the preset threshold is identified.

Supervisor: It periodically monitors whether the Raid_monitor process exists and restarts the process if it stops running.

5.1.5.7 System Disk Software RAID Group

For high redundancy and reliability of system and data, RAID 1 is usually

configured using a hardware RAID controller, so that even if one hard disk fails, the system can still run stably with secure data. However, reports on the internet indicate that RAID card failures can cause the system disk stuck, triggering business service interruptions. The consequence is so severe that all VMs on the node may be unable to provide services normally, or even lead to anomalies in the entire cluster.

Sangfor enables the system disk software RAID groups under metadata 0.9 to avoid HCI system errors due to the quality problems of low-cost RAID cards, and to match the aServer hosts without RAID cards for cost control.

5.1.5.7.1 Technical Principles

Hardware RAID has all RAID operations performed by the RAID controller chip, which does not occupy any CPU or memory resources. Moreover, there is generally an additional cache on the RAID card to further enhance the performance. A disadvantage is that it is completely hardware-dependent with no way to know its precise internal workings. Software RAID, on the other hand, uses CPU resources for RAID operations and is visible in the operating system.

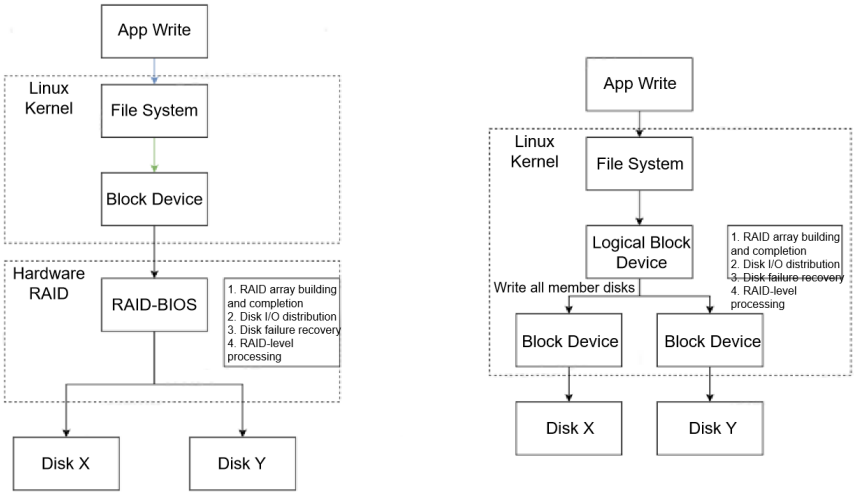


Figure 105: Comparison of Hardware and Software RAID

The software RAID creation process first creates a software RAID 1 array based on physical disks, then partitions the array and makes a boot method on it. The

benefit of this method is to provide full disk image redundancy, including zones outside of the partition tables such as Bootloader, MBR, and GPT.

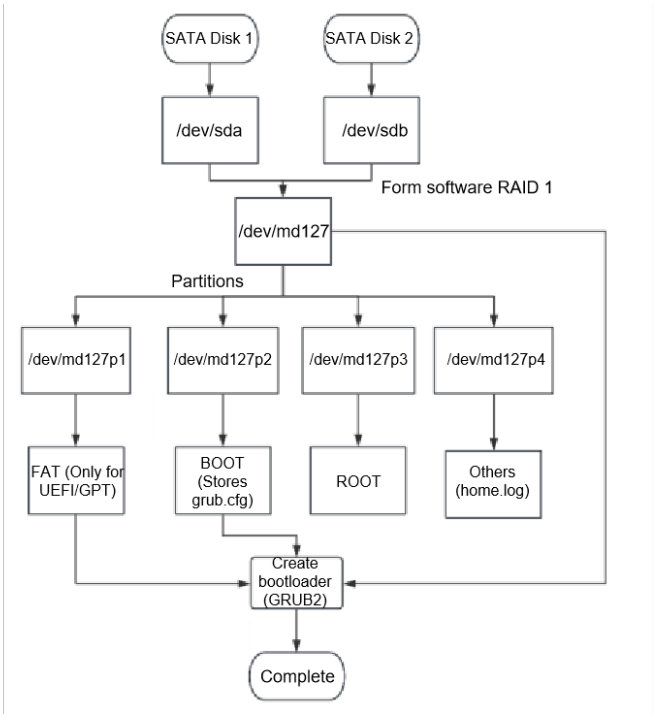


Figure 106: Software RAID Creation Process

The metadata of the software RAID 1 is placed at the end of the disk to prevent it from conflicting with the regular disk layout (such as Bootloader, MBR, and GPT partition tables). Therefore, when the system boots, the UEFI or BIOS will process RAID member disks as if they were non-RAID disks, read their MBR or EFI system partitions, and find the Bootloader to continue the boot process.

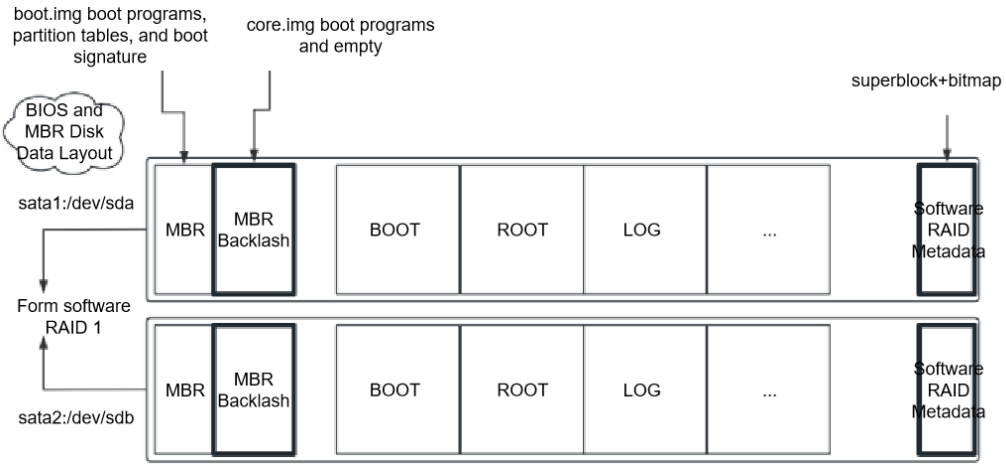


Figure 107: Software RAID Disk Layout

5.1.5.7.2 UI Display Guide

Sign the installation agreement, format the original system disk, select **Software RAID** for the system disk installation method and 1 or 2 disks to form the software RAID group.

The **Nodes** > **System Disks** page will display the **Status** of the installed software RAID group and its member disks (including **Normal**, **Alert**, and **Data sync is in progress.**).

On the **System Disks** page, click **Edit Software RAID Settings** of the software RAID group then **Add Disk**, and select a disk and click **OK** to add it, after which the **Status** of the software RAID group will change to **Data sync is in progress.**

On the **System Disks** page, click **Edit Software RAID Settings** of the software RAID group then **Delete** of a member disk to delete it.

On the **System Disks** page, click **Replace System Disk** of the software RAID group, select a disk to be replaced and click **Next**, select a new disk in the disk drop-down list and click **OK**.

On the **System Disks** page, click **Replace System Disk** of the software RAID group, select both disks and click **Next**, click **Enter Maintenance Mode**, **Back Up**, and **Turn On Disk LED**, unplug the original disk, and select to replace the system disk for ISO image installation.

5.1.5.8 Secure Replacement of System Disks

You can use Sangfor HCI to replace the system disk of a host in a safe and convenient manner when the disk is faulty or subhealthy without affecting business operations.

System Disk Monitoring and Health Check:

The health status of the system disk is obtained based on the information about disks with bad sectors and slow disks described in “Sub-Healthy Disk Detection and Handling” and the information about the lifetime and status obtained from the monitoring of the O&M platform.

Secure System Disk Replacement Process:

The following figure shows the system disk replacement process.

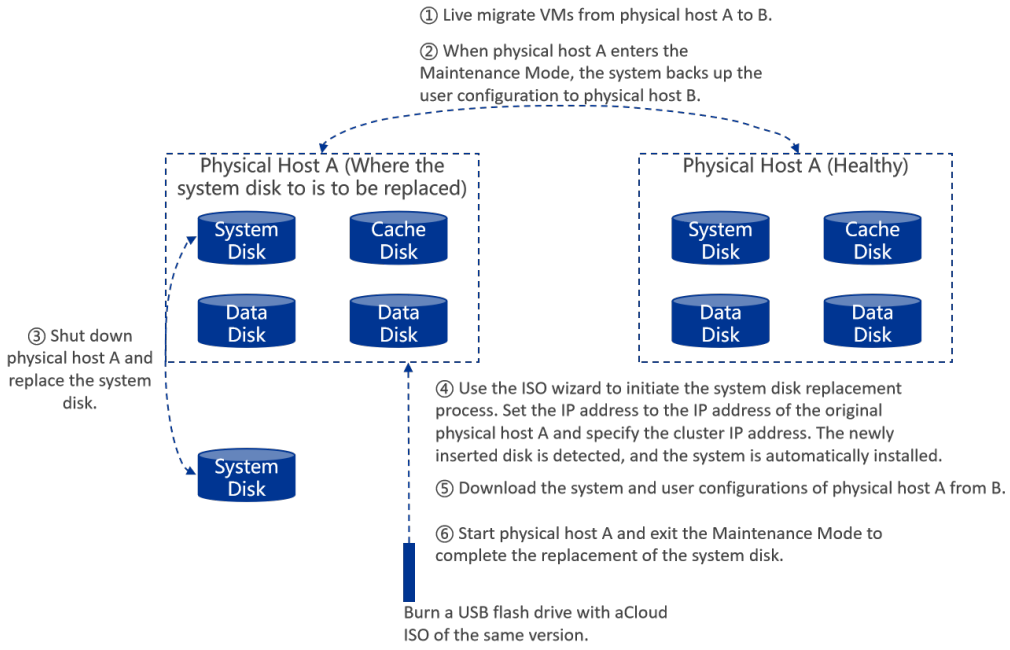


Figure 108: System Disk Replacement Process

The following figure shows the process of system disk configuration backup and restore.

- In cluster scenarios:

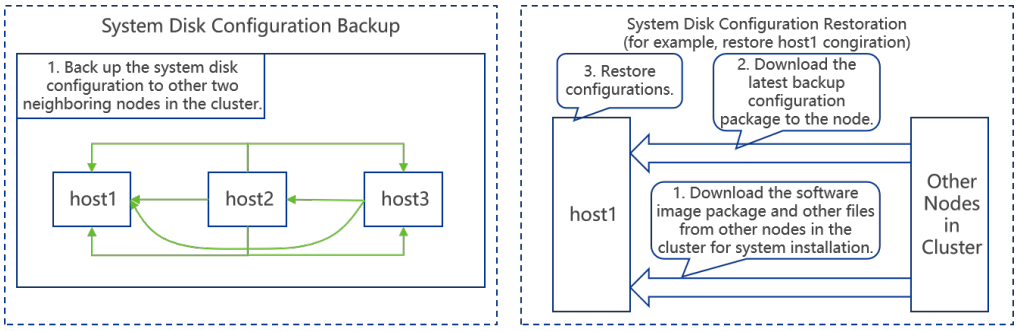


Figure 109: System Disk Replacement Process in Cluster Scenarios

1. System disk configuration backup: System disk configuration backup can be automatically performed periodically (daily). You can also execute such a backup to the other two neighboring nodes in the cluster.
2. System disk configuration restoration: After the replacement of the system disk on the node, the new disk is formatted and partitioned. The VMs, RD files, and service packs are copied from other nodes in the cluster for HCI installation. Then, the previously backed-up system disk configuration is obtained from other nodes in the cluster for restoration, followed by node restart. The restoration is completed.

- In single-host scenarios:

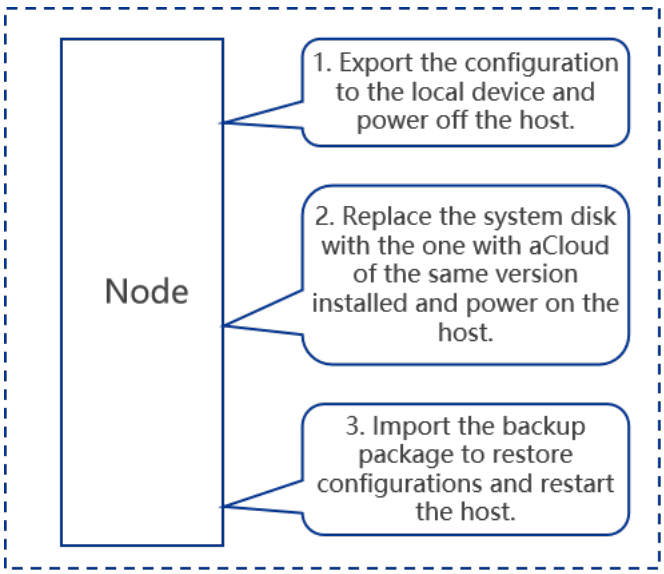


Figure 110: System Disk Replacement Process in Single-Host Scenarios

1. System disk configuration backup: The host system disk configuration is exported to the local storage in HCI.
2. System disk configuration restoration: The old system disk is replaced by a new one with HCI of the same version installed, followed by power-on and startup of the host. The previously exported configuration package is then imported into HCI for configuration restoration. The new configuration takes effect after the host restarts.

Precautions:

1. The single-host maintenance mode must be enabled before the replacement of the system disk.
2. The NIC cannot be changed during the system disk replacement; otherwise, the backed-up system disk configuration package cannot be identified, which may result in failure in configuration restoration.
3. Only one new disk should be used for the system disk replacement. If multiple disks are used, a unique system disk cannot be identified, which will result in system installation failure.
4. The new system disk cannot come from the original host disk; otherwise, no new system disk can be found, which will result in system installation failure.

NOTICE

The secure replacement feature is not applicable to system disks with RAID creation. For further assistance, contact a Sangfor technical support representative.

5.1.5.9 Multi-USB Mapping Between Hosts

In scenarios where a USB key is required by third-party software for application encryption, the USB key inserted in the virtualization platform must be mapped to the VM.

Three common solutions in the industry:

1. Host mapping: Host mapping is directly used for USB key mapping. However, the drawback is that it does not support network mapping and cannot meet the requirements for live migration.
2. USB Anywhere: This solution involves using an intermediate device to which an IP address is assigned. Then, driver installation and peer device configuration are performed on the VM. However, the drawback is that it requires modification and installation of specified software within the guest VM, and cooperation with third-party applications is required.
3. Underlying hardware virtualization + network proxy: This solution supports live migration and network mapping and does not require

modification within the guest VM. Finally, with seamless integration, the original USB devices can be operated and read directly after physical devices are migrated to the virtualization platform. This solution overcomes the defects in the previous two solutions and eliminates obstacles because of peripheral mapping during the promotion of virtualization. Based on the network proxy, during migration to the peer device and VM switching, messages are sent to trigger the USB service program to change the IP address of the destination end connected, and network reconnection is initiated. Once the tunnel is successfully reconnected, USB device communication is normal, which cannot be perceived at the VM layer.

Sangfor adopts the third solution mentioned above while optimizing it by supporting multiple USB devices.

The advantages include:

1. Migration of VMs (plugin installation is not required) between different hosts is supported for mapping between hosts.
2. After VM migration is completed, the original USB devices can be automatically mounted.
3. Automatic remapping is implemented after errors, such as VM restart upon a failure and destination USB device network disconnection, are recovered. This ensures continuous business operations in HA and similar scenarios.

5.1.5.10 UPS Integration

HCI hosts can be associated with UPS, and aSV proactively detects the running status of the UPS. When aSV detects a power failure in the utility power lines and insufficient power supply from the UPS through UPS detection, it proactively reports an alert or initiates soft shutdown of VMs based on user configurations to avoid affecting VM operations due to sudden power failure.

Typical application scenario: Users define VMs running critical businesses as critical VMs by criticality. In the event of a utility power line failure, noncritical

VMs are automatically shut down by default when the UPS battery level drops to 80%, and critical VMs are also shut down when the battery level reaches 30%. Based on their specific business continuity requirements, users can customize the thresholds for layered business protection and reduce the impact of sudden power failure on critical businesses.

The following figure shows the deployment diagram for UPS integration.

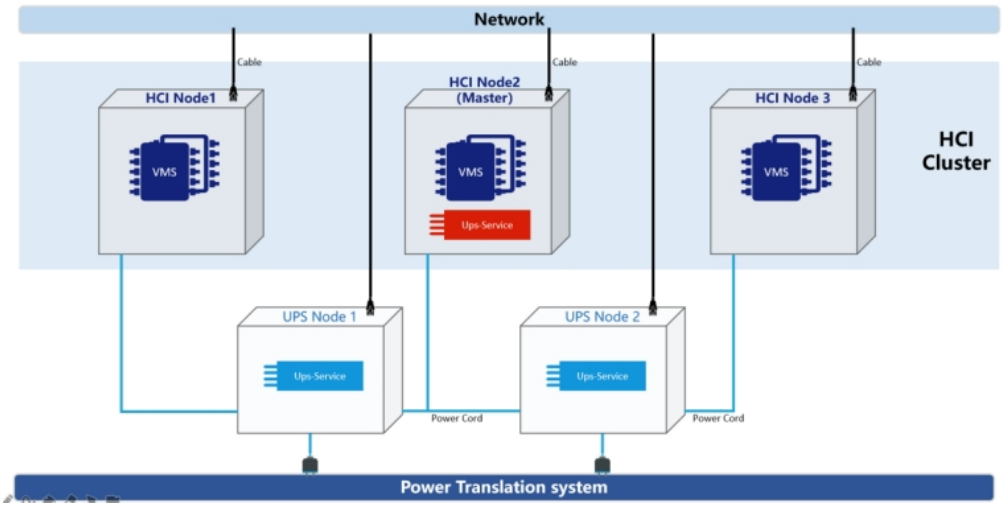


Figure 111: UPS Module Deployment

5.1.5.11 Recycle Bin

When an administrator deletes VMs or other resources by mistake and such resources need to be retrieved, HCI provides a resource recycle bin mechanism. The administrator can use the recycle bin to retrieve VMs and virtual network devices that have not been completely deleted with one click in the recycle bin. This mechanism provides users with a "buffer zone" for deletion by mistake and an opportunity to "undo" their operations, ensuring the reversibility and correctness of user operations as much as possible.

The virtual devices deleted by users are temporarily stored in the recycle bin for a certain period of time, and the disk space occupied by them, as well as relevant data, is not deleted. Devices in such cases can be retrieved. However, when the deleted devices are automatically deleted permanently after being stored in the recycle bin for more than 30 days or permanently deleted by

users, the disk space they occupy is released.

5.1.5.12 Process Watchdog

Unknown errors in system processes may lead to crashes, deadlocks, and other errors. They may prevent system processes from providing services to external systems. In this case, HCI provides a process watchdog mechanism to recover the processes in a timely manner.

The mechanism is implemented by running a separate daemon process at the backend of HCI. The process responsible for monitoring all HCI system processes and has the highest priority. When a system process crashes or undergoes a deadlock, the process watchdog intervenes forcibly to restart the process and recover business operations. It also records the status information about the process at that time in a black box for post-event analysis.

5.1.5.13 Black Box Technology

To ensure business continuity and facilitate fault locating and handling when a system undergoes a crash, process deadlock, or unexpected resetting, HCI prioritizes the recovery of businesses and provides black box technology that is used to back up "last gasp" information to a local directory for further fault analysis and handling.

This technology is used to collect and store kernel logs, diagnostic information from diagnostic tools, and other data before the OS exits unexpectedly on management and compute nodes. This allows system maintenance personnel to export and analyze the data saved by using the black box when the OS crashes.

5.1.5.14 Host Maintenance Mode

The host maintenance mode can be used when administrators need to power off a host. After this mode is enabled, the system first migrates the businesses in HCI to other hosts without affecting the business operations. Self-O&M of hosts can be implemented in this mode. The following figure shows the

principles.

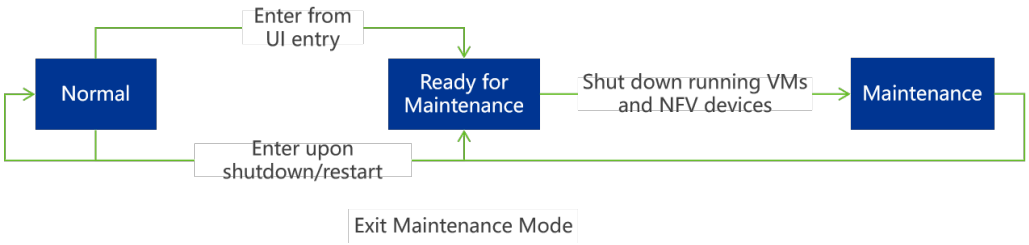


Figure 112: Host Maintenance Mode Process

Each host can be in the following three states: normal, ready for maintenance, and maintenance.

If a host is in the ready for maintenance state, it indicates that the host is in the process of migrating and shutting down VMs, NFV devices, and routers. If a host is in the maintenance state, it indicates that the migration and shutdown processes are completed and that the host no longer accepts any operations on VMs and other devices.

Users can choose to enter the maintenance mode before shutting down or restarting the host. The host is in the ready for maintenance state after the entrance through the UI entry. The VMs, NFV devices, and routers running on the host can be migrated to other hosts or shut down. After the migration or shutdown is completed, the host enters the maintenance state. In this case, the host no longer accepts any requests for operations on VMs or other devices and is no longer involved during the implementation of DRS policies.

Users need to exit the maintenance mode; otherwise, the host will remain in this mode. The following figure shows the flowchart on how a host in a normal state enters the ready for maintenance state.

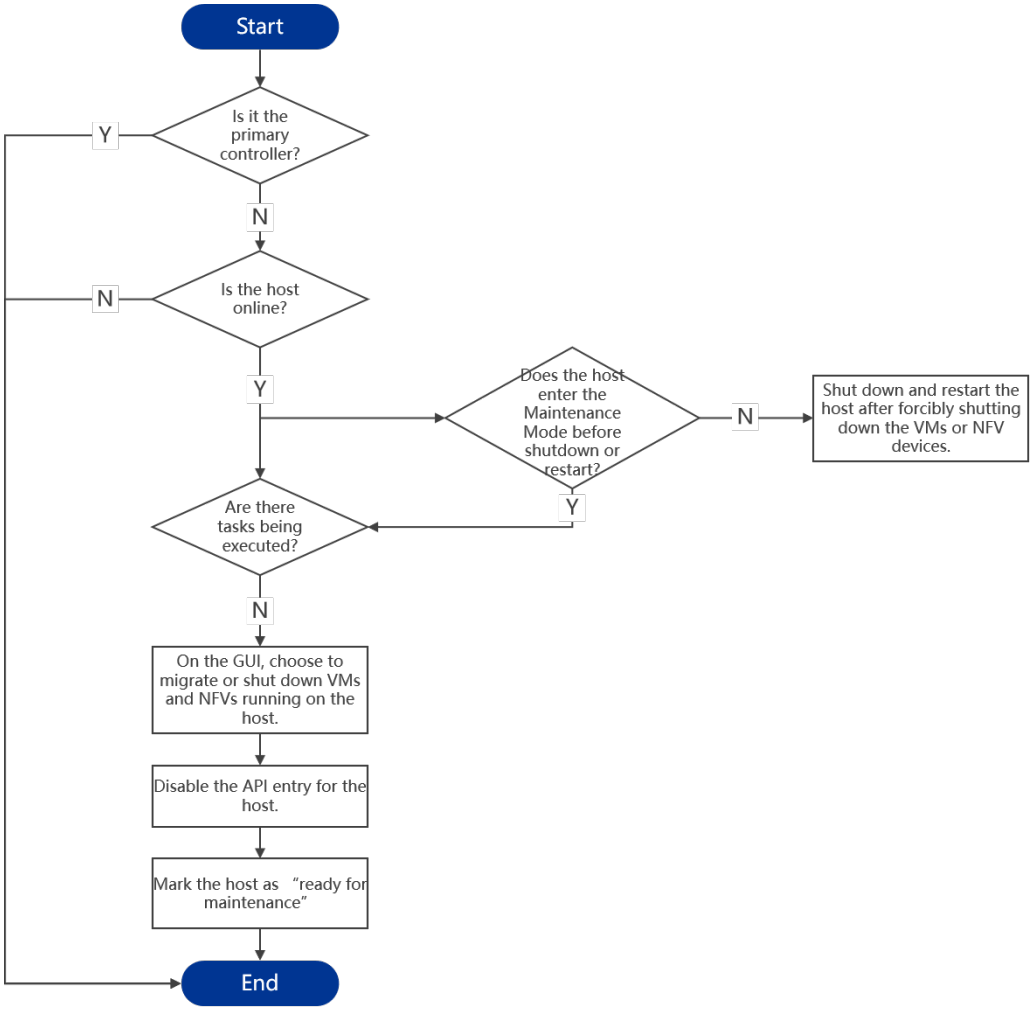


Figure 113: Entering the Ready for Maintenance State

1. A host can enter the maintenance mode through either of the two UI entries: on the host panel or during host shutdown or restart. If a user chooses not to enter the maintenance mode when shutting down or restarting the host, the VMs, NFV devices, and routers running on the host are forcibly shut down.
2. After the host enters the maintenance mode through the UI entry, the system checks for any running tasks, such as migration, clone, and backup. If such tasks are identified, a message is displayed, indicating failure to enter the maintenance mode. The user can either wait or cancel these tasks and try again. Then, the system checks for any routers that cannot be migrated (such as the routers that are disconnected from VMs after

migration to other hosts). The host cannot enter the maintenance mode if any such routers are identified.

3. The user selects the VMs, NFV devices, and routers to be migrated and shut down on the GUI, enters the administrator password, and prohibits all operations on the VMs, NFV devices, and routers.
4. The approaches in step 2 are taken again to check for any running tasks or VMs that cannot be migrated in the current cluster. If so, the APIs disabled in step 3 are enabled, and the maintenance mode is exited.

The following figure shows the flowchart on how a host in the ready for maintenance state enters the maintenance state.

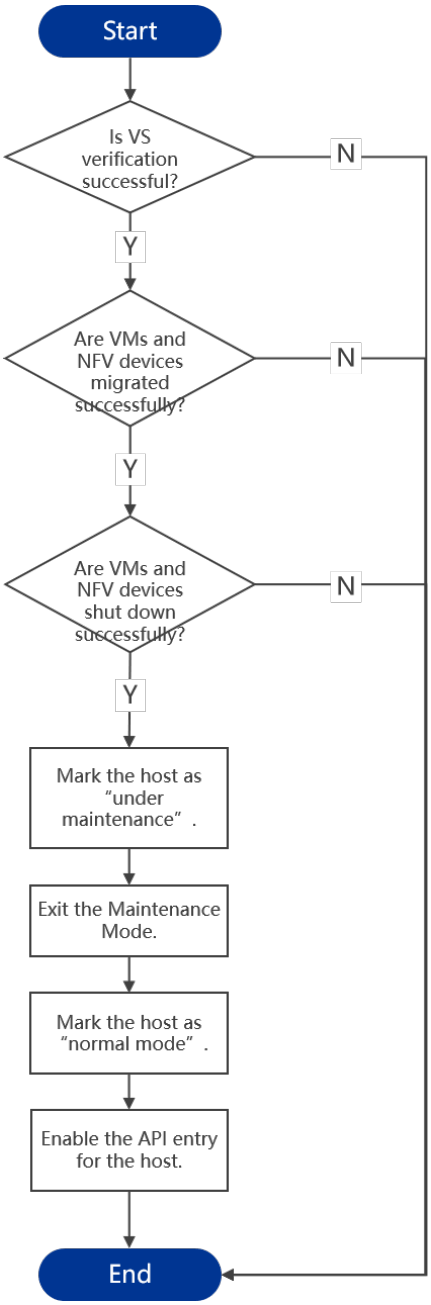


Figure 114: Ready for Maintenance State - Maintenance State

1. The three key steps of the maintenance mode are as follows: 1. The aSAN data API is called for verification. 2. The VMs and NFV devices selected are migrated. 3. The VMs and NFV devices selected are shut down accordingly.
2. After the host enters the maintenance mode, it can be shut down or restarted based on user requirements. Users must exit the maintenance

mode; otherwise, even after the host is restarted, it cannot receive any requests from the frontend.

3. The host is normal after the user exits the maintenance mode.
4. The PUT and POST requests are allowed in the API. This also applies to the scenario where calling the VS API or migrating/shutting down VMs fails.

A host in single-host maintenance mode is subject to the following restrictions:

1. The host is not involved in the generation of migration policies using DRS.
2. The HA startup mechanism does not involve the host.
3. HCI cannot undergo an active upgrade on the host.
4. As all the PUT and POST requests sent to the host are prohibited, operations such as clone, snapshot, migration, and backup and restore cannot be performed on the host. In addition, cloning or backing up and restoring a normal VM to the host is not supported. The host cannot be selected as the VM running location for editing. VMs cannot be migrated to the host.
5. The local storage of the host cannot be added to the backup repository on the **Backup Repository** page. The datastore that has already been added should be displayed as offline.
6. The health check feature is used to check for any offline hosts first. If so, a message is displayed, indicating that information about the offline host cannot be obtained, and the user can continue with the health check as required. However, for the host in single-host maintenance mode, **Network Connection Error** is displayed in the result for the host.
7. Before manual or automatic switching of the primary controller, the system calls the API in single-host maintenance mode to check whether the current host is in single-host maintenance mode. If so, the system does not initiate the switching process.
8. For a host in single-host maintenance mode, local VMs in the recycle bin and routers running on the host cannot be restored or deleted. Files on local storage cannot be deleted.

9. If a host is in single-host maintenance mode, the local storage is offline. As a result, the local storage cannot be deleted, garbage-cleaned, formatted, or edited. Storage space management and adding new local storage to the host also cannot be performed, but the shared datastore is not affected.

5.1.5.15 System File Backup and Restore

HCI provides a one-click backup feature for system files (HCI configuration data). In the event of a systematic failure resulting in the loss of system configuration files, users can quickly recover the related system configurations based on the backup files.

5.1.5.16 Memory Snapshot

VMs can be recovered from memory snapshots to continue running in the status they were in at the moment the snapshots were taken.

5.1.5.16.1 Technical Principles

Memory snapshot creation follows the logic of VM live migration. VM status information (including memory, CPU, and other devices) is saved to a memory snapshot file, from which the information is loaded into memory to recover the VM to continue running in the status it was in at the time the snapshot was created. The process is as follows:

1. Save the memory contents used by the VM to a persistent storage medium (such as disk).
2. Suspend the VM to ensure snapshot consistency.
3. Save status information of the VM's CPU, disks, NICs, and other devices to ensure that the system's running status can be properly reconstructed during recovery.

5.1.5.16.2 UI Display Guide

Select **Include VM memory data** for **Memory Data** when creating a disk-based snapshot for a VM to create a memory snapshot as well. The VM will be automatically recovered to run after the creation.

The **VM Memory Data** of the created memory snapshot will display **Included**. Click **Recover** to recover the VM from it.

5.1.6 Performance and Reliability Optimization

aSV has implemented the following performance optimization measures: VirtIO optimization is performed to shorten the I/O path and reduce context switching times, resulting in improved performance in storage, NICs, and various peripherals. The inclusion of KSM technology helps minimize memory waste by merging shared or identical memory pages, which allows greater resource allocation to support more businesses. NUMA optimization ensures that VMs access local memory as much as possible, thus improving their performance. AI-based performance optimization is performed to further establish business portraits and prioritize the performance supply of critical businesses during resource sharing. VM-level quality of service (QoS) allows users to set performance levels for specific businesses at the VM granularity. Performance levels are determined based on the weight of hot data in the data tier.

5.1.6.1 VirtIO Optimization

As a popular paravirtualization solution for I/O devices, VirtIO aims to provide a unified communication framework and programming interface between VMs and hypervisor virtual devices, reduce compatibility issues across different platforms, and improve driver development efficiency. VirtIO is a frontend and backend architecture that includes frontend drivers, backend devices, and the transfer protocol defined by VirtIO. Using this protocol, VirtIO can be applied to different virtualization solutions, such as QEMU-KVM, and support different implementations of frontend and backend components.

Although featuring excellent compatibility and convenient management and O&M, pure software-based I/O device simulation using native QEMU requires multiple context switching times and data copies because of a relatively long path for each I/O, leading to numerous VM_Entry and VM_Exit events, which translates to poor performance.

Paravirtualized devices are based on the VirtIO framework, which is obtained by installing vmTools on VMs. The tight coupling architecture of HCI makes it easier to optimize VirtIO and unlock greater performance. Various VirtIO drivers exist in HCI, such as Virtio_blk, Virtio_net, and Virtio_balloon.

Virtio_blk is a VirtIO-optimized device driver for virtual disks, Virtio_net is for virtual NICs, and Virtio_balloon is for memory balloons.

VirtIO optimization involves moving the hardware simulated using QEMU to the hypervisor (Linux Kernel) to shorten the I/O path and reduce the number of context switching times, thereby boosting the performance by several times.

VirtIO can be used to optimize the performance of storage, NICs, and various peripherals. Specifically, it can significantly enhance the I/O and throughput performance of virtual disks and virtual NICs, and also facilitate efficient memory utilization and deallocation through memory ballooning. In terms of storage, VirtIO optimizes the performance between the VM and the storage (hypervisor) for a substantial improvement in the I/O capabilities of VMs for virtual disk images.

The following figure shows the difference between traditional QEMU-based simulation and VirtIO-based simulation.

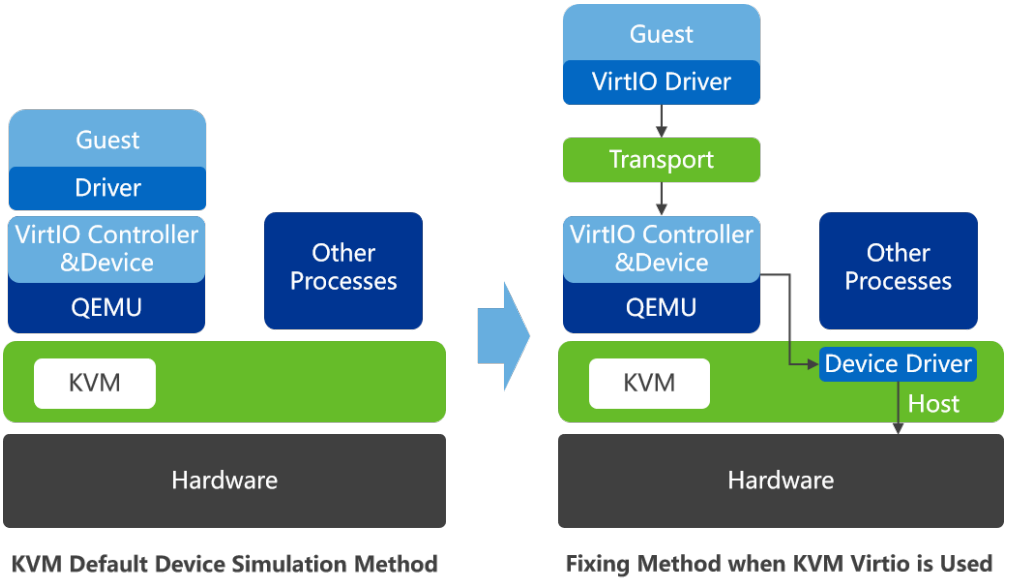


Figure 115: Comparison of QEMU-Based Simulation and VirtIO-Based Simulation

5.1.6.2 NUMA Optimization for Enhanced Computing Performance

Some businesses, such as Oracle and SQL Server databases, need frequent memory access because they rely heavily on a significant amount of memory for caching and frequent memory read and write operations. As a result, these types of businesses are highly sensitive to the memory access rate.

In a server environment, each physical CPU socket has its own directly connected memory, known as local memory, while the memory directly connected to other physical CPU sockets is known as remote memory.

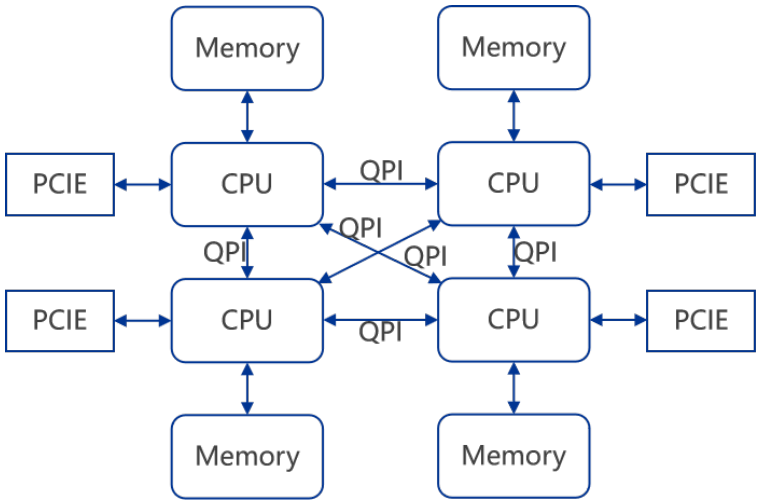


Figure 116: Relationships Between Physical CPUs and Memory

For a physical CPU, local memory can be accessed directly, while accessing remote memory requires passing through the QuickPath Interconnect (QPI) bus, resulting in a slight time delay. Although a single delay is minimal, frequent access to remote memory translates to accumulated delays, leading to significant latency. Obviously, accessing local memory is faster than accessing remote memory. As a result, NUMA optimization aims to maximize CPU access to local memory.

To address the challenges, Sangfor adopts a VM-based approach for NUMA balancing, ensuring that VMs access local memory as much as possible while maintaining a balanced load for physical CPUs.

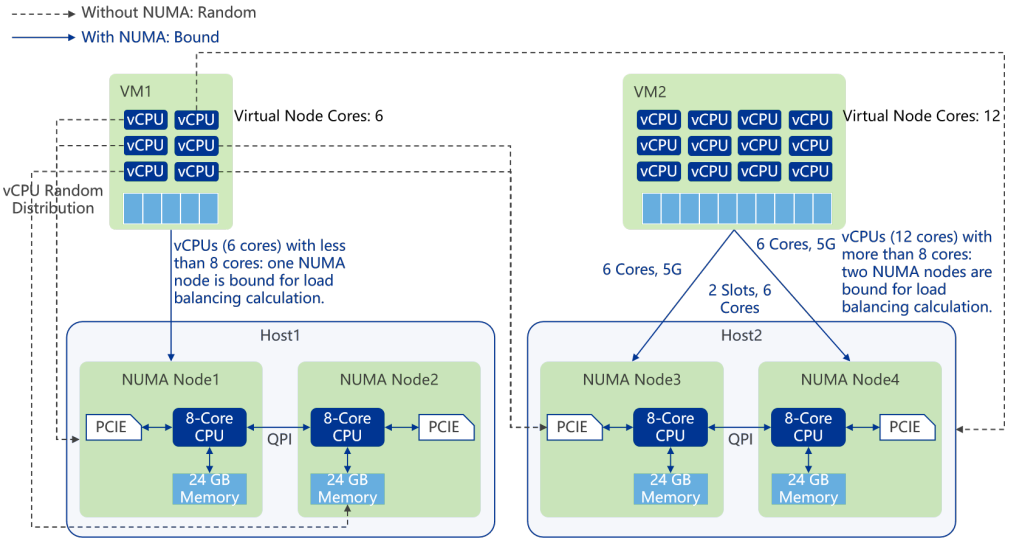


Figure 117: NUMA Optimization Architecture

In the native NUMA implementation of QEMU, the scheduling of vCPUs for VMs on the physical machine is random. This leads to not only excessive access to remote memory in the physical architecture due to uneven allocation but also frequent switching, resulting in unstable performance.

aSV binds the virtual nodes simulated using QEMU to a specific physical CPU. The corresponding memory is also bound to the local memory of that CPU. This way, the virtual architecture is aligned with the physical architecture, and the access of the CPU to local memory is maximized. aSV addresses the following challenges in kernel NUMA scheduling in VM scenarios:

1. Although the original kernel is provided with a CPU load balancing mechanism, it focuses on balancing individual threads (task_struct) instead of VMs. The vCPUs of a VM consist of a group of threads, which may be distributed on different CPUs, leading to more frequent access to remote memory. To address this, Sangfor implements a VM-based NUMA balancing mechanism, aiming to maximize access of each VM to local memory.
2. The original kernel prioritizes CPU load balancing over NUMA remote memory. During the migration of a thread from one CPU to another, the kernel focuses on CPU load without taking the historical local memory hit rate of the destination CPU into account. In contrast, the NUMA mechanism of Sangfor takes both load balancing and maximizing access to local memory into account. Load balancing is prioritized only in the event of overcommitment.
3. To ensure optimal VM performance, aSV proactively identifies the NUMA architecture of the physical server and passes through the VM node information to the physical machine, thereby ensuring NUMA consistency.

For example, in the above figure, the VM NUMA architecture identifies VM2 as a VM with 26 cores. The two virtual nodes (six cores) of VM 2 are passed through to the physical machine, and the NUMA architecture of the physical machine ensures that the two virtual nodes are fixed to either two physical NUMA nodes or one physical NUMA node (if possible).

With NUMA scheduling optimization, aSV significantly boosts its computing

performance by 7% to 30%, catering to businesses with demanding high-performance requirements.

NOTICE

When the number of vCPU cores on a VM exceeds the number of CPU cores on a single NUMA node, it is necessary to distribute the vCPUs evenly to multiple NUMA nodes. In this scenario, the general principle is to minimize the number of NUMA nodes used for a VM while meeting the CPU core requirements for a single NUMA node, thus maximizing access to local memory.

5.1.6.3 VM-Level QoS

The operation of critical businesses should be preferentially guaranteed when the cluster has limited available resources due to insufficient system resources, host failures, or VM HA. To ensure the allocation of resources to critical VMs, Sangfor HCI provides two methods for VM priority classification and marking, that is, critical VM setting and VM performance level setting. The following VM performance levels are available: high performance, default performance, and low performance.

Set business VMs as critical VMs and obtain the following priority for the I/O queue:

I/O of critical VM businesses > I/O of noncritical VMs > other I/O (such as backup and data rebuilding). HCI automatically checks the I/O throughput load and physical space utilization of each physical disk and provides varied scheduling policies according to business requirements, thus effectively maximizing I/O performance within the cluster.

Set VM performance levels and obtain the utilization priority for tier (SSDs) cache space.

- High performance:

High-performance objects occupy more tier storage space to achieve a higher cache hit rate, thus ensuring high performance. When high-performance data blocks get cold and meet the elimination condition, the data blocks will be put

into the hot data queue again instead of being eliminated instantly. The data blocks will be eliminated from the tier only when the elimination condition is met again. This processing method provides privileges for high-priority data, preventing it from being eliminated easily.

- Default performance:

Default-performance objects are processed by the normal elimination algorithm. When data blocks get cold and meet the elimination condition, they are eliminated from the tier. That is, default-performance objects have a half possibility to preempt tier space compared to high-performance objects.

- Low performance:

Low-performance objects are restricted in using the tier storage space, ensuring sufficient space for high-performance and default-performance objects.

aSAN allows a single low-performance disk to occupy at most 10 GB tier space. The rest of the space is available for preemption by high-performance and default-performance objects.

5.1.6.4 CPU, Network and Disk QoS

The QoS upper limits of VM CPU clock speed, disk I/O, and NIC traffic can be set. HCI administrators can set upper limits for the resources used by some VMs based on the criticality of business operations. This ensures that critical VMs have sufficient resources, and abnormal VMs are restricted from consuming excessive resources, thereby maximizing resource availability for normal VMs.

Restriction on the CPU clock speed: The range is between 100 MHz and 1,000 GHz. Note that if the limit of the CPU clock speed is too low, for example, 100 MHz, VM report data may not be updated in a timely manner, and the VM may restart.

NIC traffic restriction: The outbound and inbound traffic limit for the VM NIC is between 1000 Kbps and 20000 Mbps.

Disk I/O restriction: The limit of the maximum VM disk I/O rate is between 128

KB/s and 102,400 MB/s. The limit of the maximum I/O count is between 16 and 2,147,483,647. Note that if the limit of the I/O rate is too low, for example, 128 KB/s, the VM may restart because the too-low system disk I/O rate may not be enough to support the VM operation.

5.1.6.5 Kunpeng BoostKit

Kunpeng BoostKit is based on full-stack tuning of hardware, basic software, and application software. It comprises high-performance open-source components, as well as basic acceleration and application acceleration packages for ultimate application performance.

The basic acceleration library of Kunpeng BoostKit focuses on addressin issues such as low virtualization light-load performance, heavy network loss, severe resource fragments, and open-source ecosystem availability. It provides the tuning guide for enabling open-source virtualization, uses features such as V-Turbo and OVS acceleration to improve system performance, and uses features such as NUMA memory interleaving and Guest NUMA to reduce resource fragments. Kunpeng BoostKit for Virtualization gives full play to the computing power of Kunpeng based on the multi-core architecture and inter-core isolation.

Based on Kunpeng BoostKit and Kunpeng DevKit, Sangfor has conducted in-depth full-stack optimization of HCI, which fully leverages the characteristics of Kunpeng processors and significantly improves its processing performance while shortening data latency.

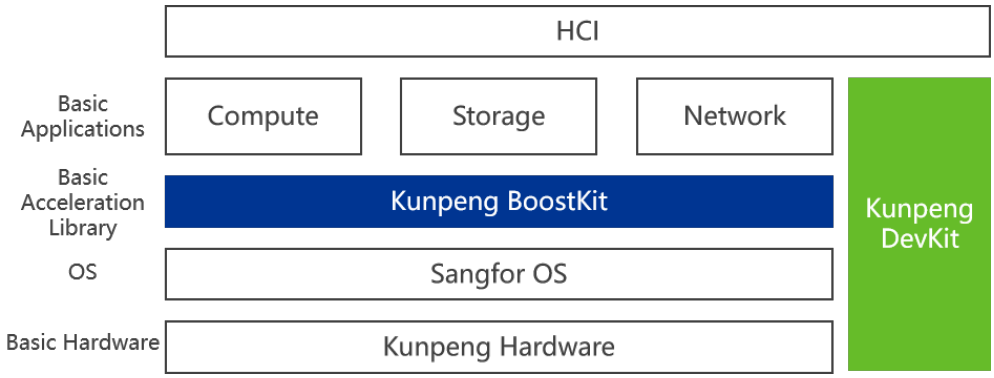


Figure 118: Kunpeng BoostKit Architecture

Kunpeng BoostKit consists of:

- KAE (KAE)

Kunpeng BoostKit Library is optimized based on Arm instructions and developed based on KAE. It covers the system library, compression, encryption and decryption, and other scenarios. Specifically, Glibc-patch optimizes and accelerates the memory, string, and lock based on the microarchitecture of Huawei Kunpeng 920 processors. It makes full use of the advantages of Kunpeng instructions and the Kunpeng processor architecture to improve execution efficiency. KAE, a hardware acceleration solution, includes KAE encryption and decryption as well as KAEzip, which are used to accelerate SSL/TLS applications and data compression, respectively. As a result, it can significantly reduce processor consumption, shorten data compression time, and improve processor efficiency.

- BiSheng Compiler

BiSheng Compiler is a high-performance, trusted, and scalable compiler toolchain designed for general-purpose processor architectures like Kunpeng. It introduces and enhances multiple compilation optimization technologies and supports C/C++/Fortran and other programming languages. In addition to general functions and optimization of LLVM, the toolchain of BiSheng Compiler has the middle- and back-end key technologies optimized and the Auto-tuner feature integrated to support automatic tuning of the compiler.

Based on Kunpeng BoostKit and Kunpeng DevKit, Sangfor has conducted in-depth full-stack optimization of HCI, which fully leverages the advantages of Kunpeng processors and improves the I/O performance by about 20% to 30% after optimization using Kunpeng-recommended Glibc compilation.

5.1.6.6 Large-Core VM Scheduling

Large-core VMs refer to VMs that require a large number of computing resources (VMs with more than 64 vCPUs configured). Generally, core and highly concurrent businesses require the configuration of vCPUs whose quantity exceeds that of the threads on a single physical CPU to enhance computing performance. However, if a VM is configured with excessive vCPUs

(at least 64 vCPUs), a series of resource contention issues that will result in deteriorated performance, such as intensified contention, frequent context switching, and increased scheduling complexity, will occur.

Large-core VMs may encounter vCPU stacking (multiple vCPUs of the same VM run on the same physical CPU). This vCPU stacking may lead to intensified lock contention, affecting the performance. During the host OS operation, the overall performance will deteriorate due to different task workloads on different physical CPUs (as shown by the left diagram in the following figure).

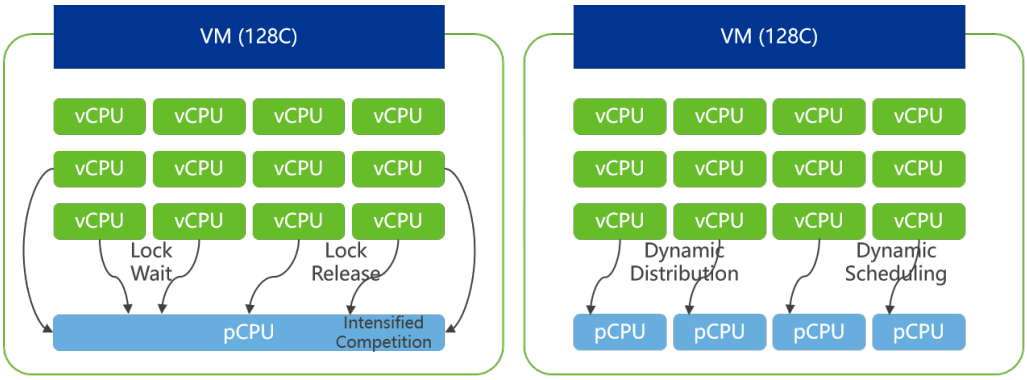


Figure 119: vCPU Scheduling

Sangfor HCI dynamically controls the distribution and scheduling of vCPUs on VMs based on the load of both vCPUs and physical CPUs and schedules the threads requiring resources to multiple idle physical CPUs for processing as much as possible. This prevents vCPU stacking, reduces lock overhead, reduces the synch delay between vCPUs, and enhances the performance of large-core VMs (as shown by the right diagram in the following figure).

5.1.6.7 CPU Scheduling Load Balancing

In business scenarios characterized by heavy network or I/O load, such as web servers, database applications, and log clusters, the interrupt load may be high due to the binding of NIC interrupts to physical cores or disk interrupts to physical cores. As shown in the following figure, the NIC interrupt is bound to CPU core 01, and the I/O processing process is bound to CPU core 02. Interrupts are an essential part of CPU load in actual business scenarios. HCI incorporates the interrupt time into CPU scheduling reference information for

more proper decision-making by the kernel scheduling subsystem and a more balanced CPU load, thereby improving system performance.

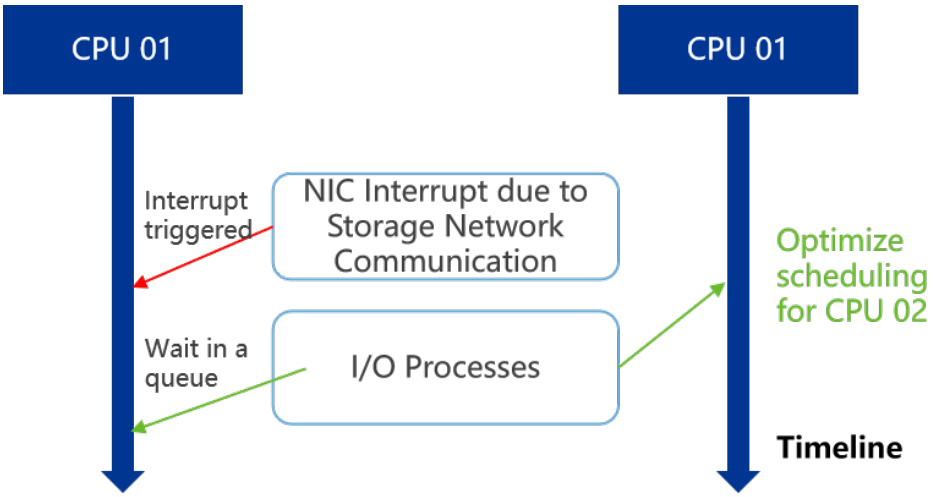


Figure 120: Interrupts and I/O Bound to CPU Cores

By default, Kylin OS does not include CPU usage occupied by interrupts in the CPU load calculation during kernel compilation configuration. However, a large number of NIC interrupts are generated during the actual business operation, consuming CPU resources. If other processes are scheduled to the CPU cores processing these interrupts, it will result in a long wait time because of CPU interrupts, thereby affecting the overall system performance. For example, for I/O-sensitive applications, related services in the I/O path must meet the requirements for low latency. When these services cannot obtain CPU resources due to excessive system load or improper scheduling, the I/O latency increases, resulting in significant I/O performance deterioration. As shown in the following figure, this defect results in uneven distribution of the system load to the two CPUs.

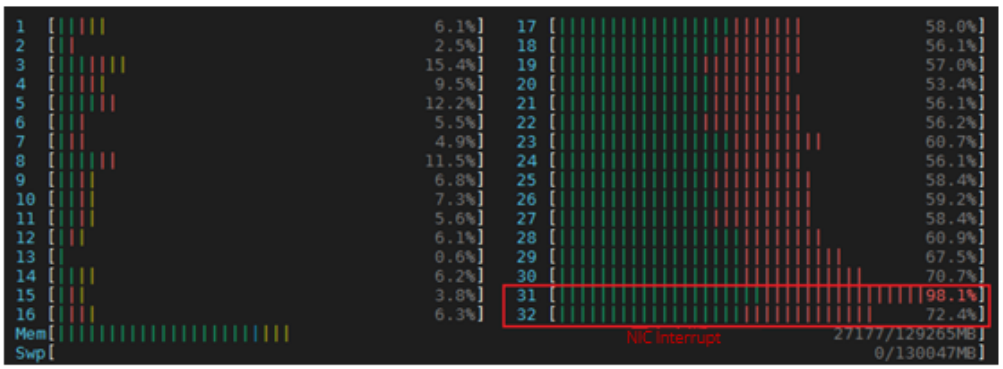


Figure 121: Uneven Distribution of CPU Scheduling Load

Sangfor HCI modifies the kernel compilation configuration on Kylin OS and includes interrupt-occupied CPUs in the CPU load calculation for accurate statistics on CPU load, which avoids uneven CPU load because of the default kernel compilation configuration on Kylin OS. HCI also optimizes the method of finding idle CPUs in the scheduling algorithm, minimizes the CPU interrupt wait time, and balances the CPU load. Through these optimizations, the I/O processing performance is improved by over 20%.

Interrupts are an essential part of CPU load in actual business scenarios. HCI incorporates the interrupt time into CPU scheduling reference information for more proper decision-making by the kernel scheduling subsystem and a more balanced CPU load, thereby improving system performance.

5.1.6.8 CPU Scheduling and Reservation

Through CPU resource allocation and reservation, the CPU scheduling optimization can preferentially guarantee the performance of VMs on which critical customer services are deployed.

5.1.6.8.1 Application Scenarios

- CPU Scheduling

When a host is running both critical and ordinary services with high loads, CPU resources are prioritized for critical services; when a cluster is deployed with both development and testing services with high loads, CPU resources are preferentially allocated to the high-priority VMs. The CPU occupied by VMs with low CPU scheduling priority is identified as idle and used by high-priority VMs with high CPU scheduling priority.

- CPU Reservation

Based on the actual arithmetic demand of VMs, CPU resources are dynamically reserved for VMs to flexibly safeguard the business services. A VM running production services with tidal characteristics can be configured with a CPU reservation schedule in which the VM has the most CPU resources with the

highest CPU scheduling priority.

5.1.6.8.2 Technical Principles

CPU Scheduling:

When a VM has the **High Priority** option enabled, its processes will have high scheduling priority and CPU resources will be preferentially allocated to them.

The scheduling priority of processes is generally varied by adjusting the nice value or time slice shares to only change the allocation ratio of time slices between processes, while the CPU scheduling optimization ensures the performance of high-priority VMs through the following measures:

1. Process Wake-Up Optimization: Enables processes with high priority to directly consume the CPU of processes with low priority when they need, reducing scheduling latency.
2. Process Core Selection Optimization: Enables processes with high priority to preferentially use the CPU of processes with low scheduling priority in the free or running queues, further reducing interference from other processes.
3. Virtualization Accommodation: vCPU threads with natively low scheduling priority can incur significant performance degradation during intense spinlock contention within the VM. In this case, CPU resources are allocated to other vCPU threads to accelerate the release of the target spinlock.

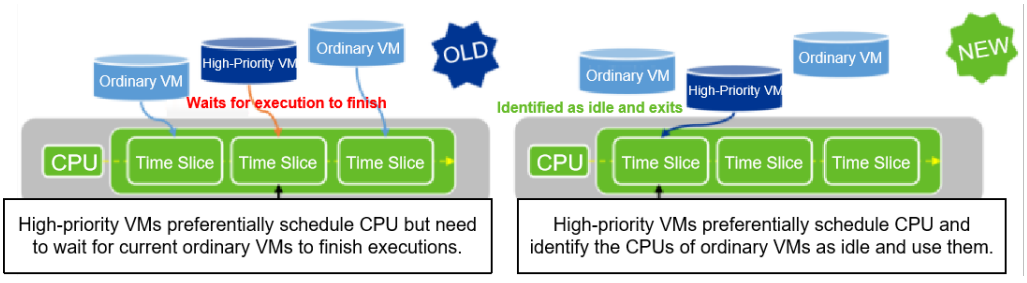


Figure 122: CPU Scheduling Optimization

CPU Reservation:

When a VM has the **Enable CPU reservation** option enabled, the **High Priority**

option is enabled as well. By periodically evaluating the CPU needs of a VM, sufficient physical cores are reserved and allocated to its threads at one time, effectively preventing other high-priority VMs from consuming the CPU resources at the same time, which increases the CPU switching overhead due to time slice preemption in turn. As shown in the figure below:

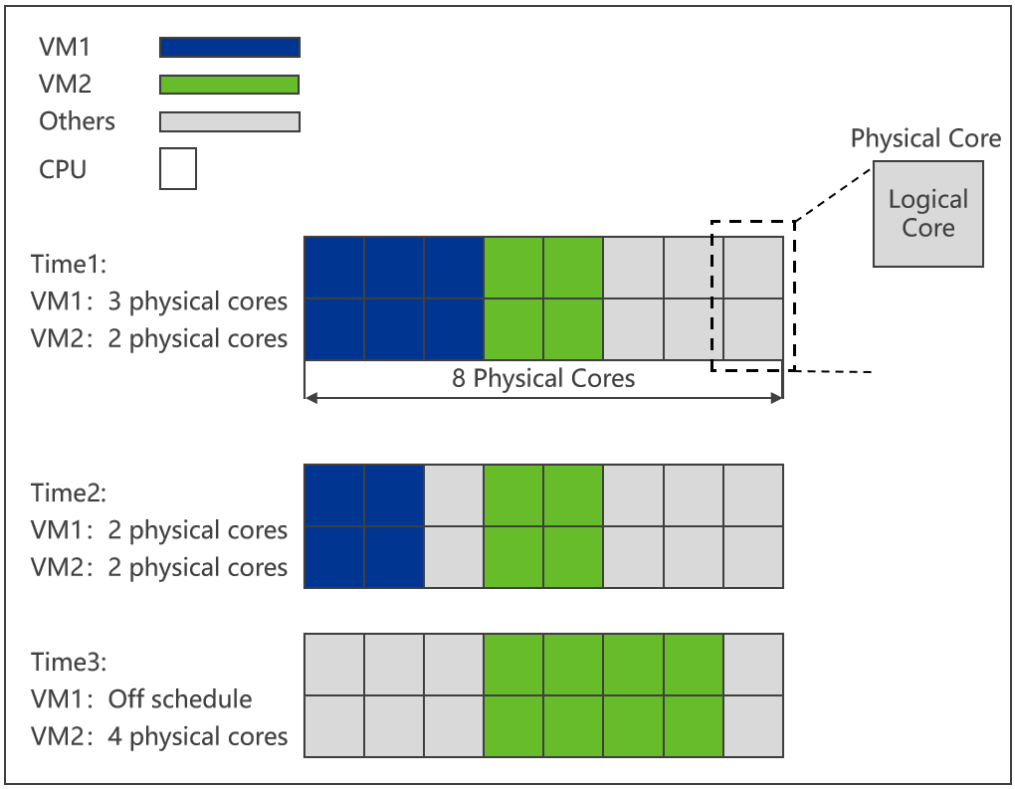


Figure 123: CPU Reservation

1. Time1: 3 physical cores are reserved for VM1 and 2 physical cores are reserved for VM2 to meet their needs, and the remaining physical cores are available for other ordinary VMs and system services.
2. Time2: Physical cores reserved for VM1 are reduced due to the lower load of business services, and the temporarily idle physical cores are allocated to other ordinary VMs.
3. Time3: VM1 has business services running off the reservation schedule and shares the remaining physical cores with other ordinary VMs as a high-priority VM. Physical cores reserved for VM2 are increased to 4 as demand grows.

The following is a comparison of common guarantee methods and dynamic reservation for VM CPU:

	Exclusive Mode	Core Binding	Dynamic Reservation
Mechanism	Exclusive to a VM and cannot be used by other VMs.	Bound to a VM.	Exclusive to a VM and can be used by other VMs when idle.
Advantage	Best performance.	High CPU utilization when idle.	High CPU utilization when idle.
Disadvantage	Low CPU utilization.	The performance cannot be guaranteed due to CPU preemption.	The implementation is complex and the performance is a little lower than that of the exclusive mode.

Table 9: Comparison of Common Guarantee Methods and Dynamic Reservation for VM CPU

In summary, VMs have the best performance in CPU-exclusive mode, but CPU resources are underutilized under low loads, while the CPU core binding method provides high utilization, but cannot guarantee VM performance during intense CPU preemption. Dynamic reservation reserves a number ($\mathrm{CPU_{demand}}$) of physical cores as needed and eliminates the interference of hyper threading, effectively ensuring VM performance and improving CPU utilization.

The formula for calculating the number of physical CPU cores required for a running VM:

$$\mathrm{CPU_{demand}} = \mathrm{CPU_{used}}\left(1 + \frac{\mathrm{CPU_{ready}}}{\mathrm{CPU_{used}} + \mathrm{CPU_{sleep}}}\right)$$

$\mathrm{CPU_{demand}}$: Number of physical CPU cores required for a running VM.

$\mathrm{CPU_{used}}$: Time slices already obtained by vCPUs, which is the actual running time of vCPU threads (counted every 10 seconds).

$\frac{\mathrm{CPU_{ready}}}{\mathrm{CPU_{used}}+\mathrm{CPU_{sleep}}}$: Time slices still wanted by vCPUs.

CPU_{ready}: vCPU thread waiting time for scheduling.

CPU_{sleep}: vCPU thread sleep time.

5.1.6.8.3 UI Display

CPU Scheduling:

1. Select **High Priority** for **VM Features** when creating and editing a VM. After that, the VM will have high scheduling priority.

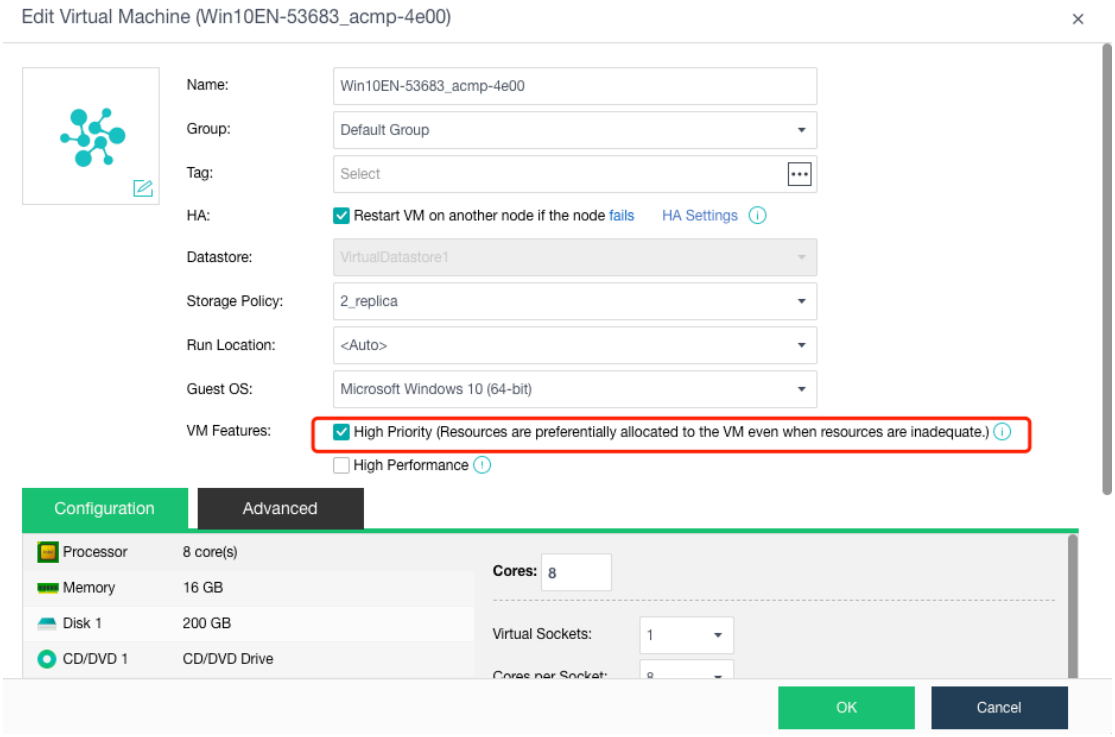


Figure 124: Select High Priority

CPU Reservation:

2. Go to **System** > **Cluster Settings** > **CPU Reservation Ratio Settings** to configure the **Global CPU Reservation Ratio** and **CPU Reservation Ratio** for each node.

Taking “Physical CPU Cores = 100, CPU Reservation Ratio = 80%” as an example, the total number of vCPU cores allocatable to VMs with CPU reservation enabled cannot exceed 80.

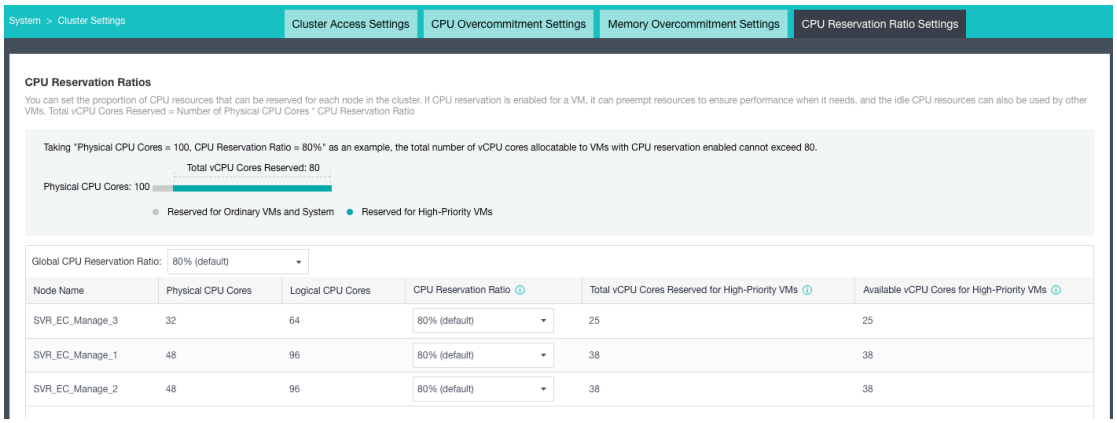


Figure 125: Configure CPU Reservation Ratio

3. Select **Enable CPU reservation** and specify schedules for enabling CPU reservation when creating and editing a VM.

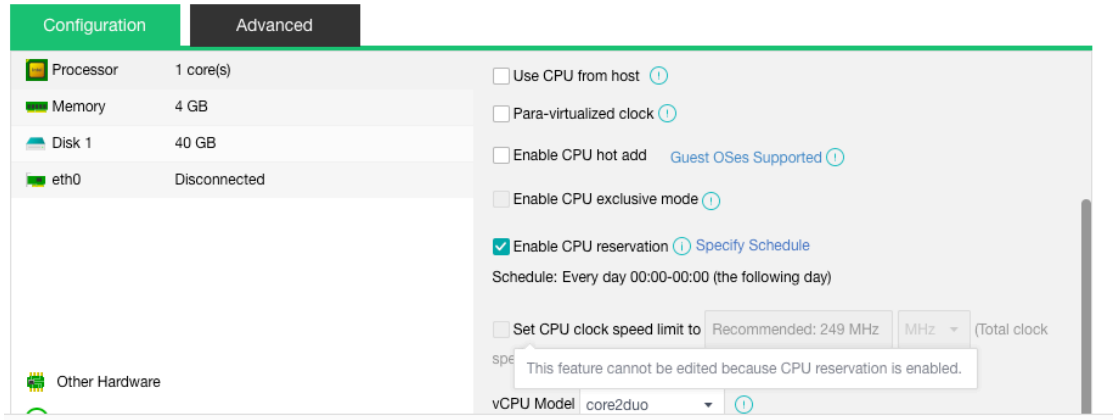


Figure 126: Select Enable CPU Reservation

5.1.6.9 Optimization of Computing Conversion Rate

In some scenarios, customers may worry that their business performance will deteriorate after cloud migration due to the relatively less optimal performance of chips made in China. This can be addressed by reducing virtualization overhead and improving the virtualization conversion rate of made-in-China chips, thus improving customer business performance experience.

Compared to native Kylin OS, Kylin OS used by Sangfor HCI enjoys more paravirtualization features that can effectively reduce virtualization overhead

and bring the following benefits:

1. In a single-core test of UnixBench, the score for a specific sub-item (pipe-based context switching) witnesses an increase of over five times, and the total score witnesses an increase of about 10%.
2. In scenarios of overcommitment and overload (where two VMs with their cores fully utilized are running the same task, occupying all the CPUs), the performance of kernel compilation is improved by over 20%, and the UnixBench score is more than doubled.

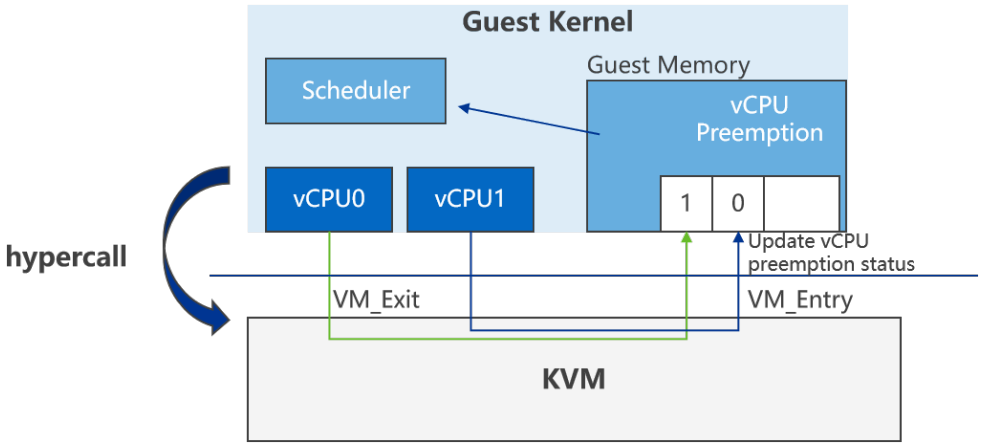


Figure 127: VM_Exit/VM_Entry Diagram

In virtualization scenarios, frequent VM_Entry and VM_Exit operations, which involve switching between the VM and the hypervisor, will result in significant performance overhead. Through the adaption of made-in-China hardware to paravirtualization and Guest-Idle-Haltpoll features, VMs can better perceive the virtualization scenarios and execute optimal policies while reducing the frequency of VM_Exit events. This leads to lower virtualization overhead and an increased virtualization performance conversion rate.

PV-Preempted:

A vCPU of a VM is a thread of the QEMU process on the physical machine, which can be used by the VM only when it is scheduled to a physical CPU and the system enters the VM_Entry state. When the vCPU is idle, VM_Exit is triggered to release the physical CPU.

Before the use of PV-Preempted technology, the VM could not perceive the

preemption status of the vCPU internally. If the vCPU is locked, and the physical CPU is released (VM_Exit is triggered), or if the VM process is scheduled to a vCPU for which the physical CPU is released accordingly (VM_Entry is triggered), it will result in significant virtualization overhead.

By using PV-Preempted technology, the physical machine updates the vCPU status information in the memory shared by the VM and the physical machine when the vCPU is called in or out. This allows the VM to obtain the vCPU status information through a hypercall and select optimal policies to reduce unnecessary performance loss.

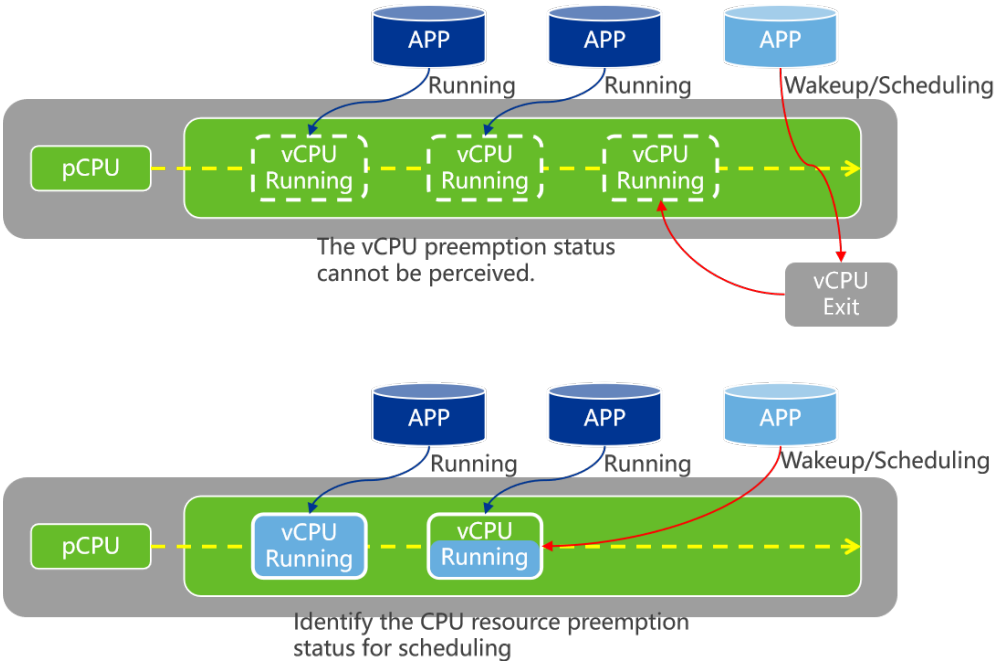


Figure 128: PV-Preempted Technology

PV-qspinlock:

PV-qspinlock optimizes the spin lock in the virtual scenario of CPU overcommitment. It allows the hypervisor to set the vCPU in the lock context to the block state and wake up the corresponding vCPU after the lock is released. In this way, physical CPU resources can be better used in the overcommitment scenario, and the compilation application scenario is optimized to reduce the compilation duration.

Guest-Idle-Haltpoll:

The Guest-Idle-Haltpoll technology indicates that when a vCPU of a VM is idle, WFx/HLT is not executed immediately, and VM-exit it triggered. Instead, polling is performed on the VM for a period of time. Within this period, the tasks of other vCPUs that share the cache on the vCPU are woken up without sending Inter-Processor Interrupts (IPIs). This reduces the overhead of sending and receiving IPIs and the overhead of VM-exit, thereby reducing the task wakeup latency.

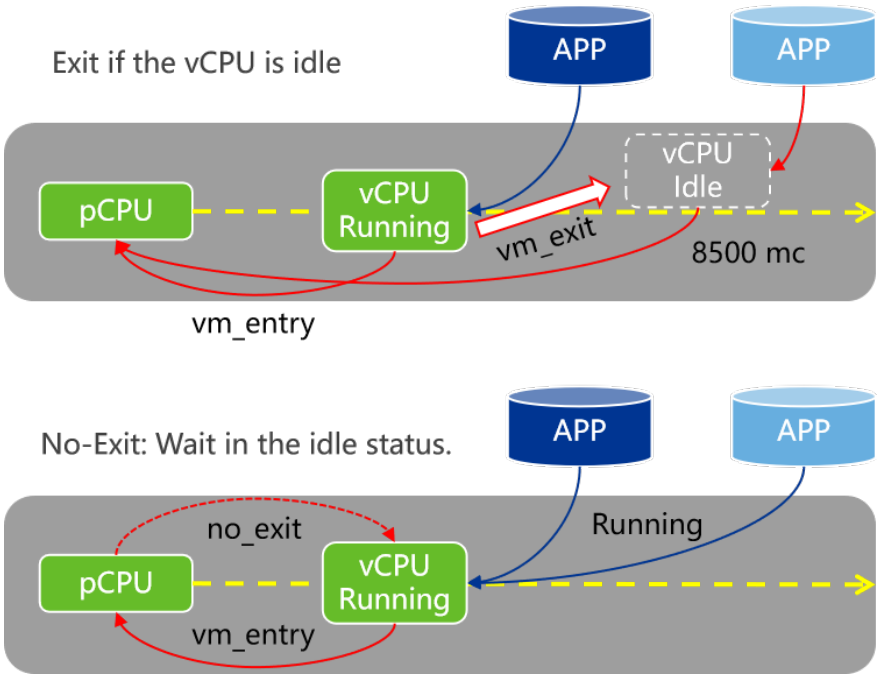


Figure 129: Guest-Idle-Haltpoll Technology

IPI Processing Optimization:

Inter-processor interrupt (IPI) is used for communication between processors in a system and allows one processor to interrupt another to perform a specific task or synchronization. It is commonly used in scenarios such as task scheduling, load balancing, and TLB flush.

In virtualization scenarios, vCPUs need to exit to the host through IPI between them to write MSR registers to notify target vCPUs, and the processing speed of exiting events directly affects IPI performance.

- **Traditional Processing Mechanism**: After exiting to the host, the VM vCPU performs routine processing actions, including enabling host

interrupts and holding SRCU locks, before sending a message to the target vCPU and waiting for a return.

- **Optimized Processing Mechanism**: During exit, a message is sent to the target vCPU before executing routine processing actions to accelerate IPI processing by reducing IPI communication latency by about 10%.

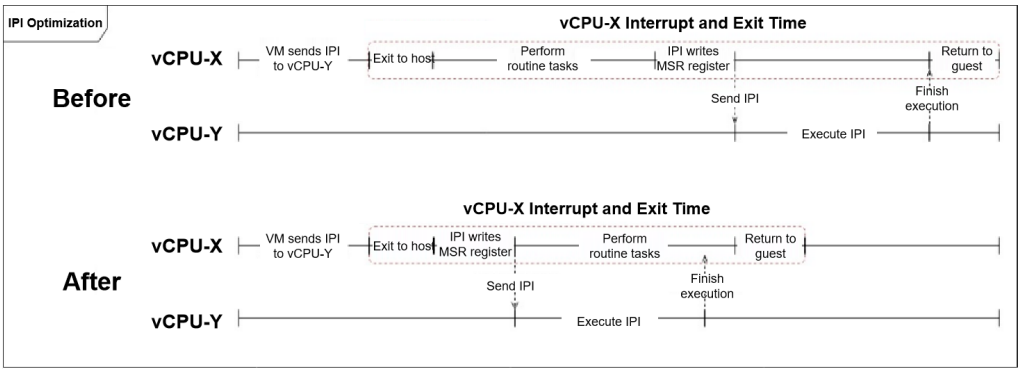


Figure 130: IPI Processing Optimization

5.1.6.10 PAT Optimization for Kylin OS

Page Attribute Table (PAT) is a method of setting various memory attributes at the granularity of page tables in the C86 architecture, which involves write-back, uncached, and write-combined.

The performance when the CPU accesses system memory varies with the memory attribute settings. When the CPU accesses the system memory, uncached provides the poorest performance, write-back provides the best performance, and write-combined falls in between. This discrepancy arises because the hardware cache is used in write-back mode, which allows the CPU to interact solely with the cache without directly accessing the memory for both read and write operations. This leads to better performance and eliminates the delay because of data merging. This leads to better performance and eliminates the delay because of data merging. Therefore, the write-back mode can be better applied to experience-sensitive scenarios in common office environments, such as smooth typing responsiveness in Sogou Input Method and smooth scrolling and page transition on Kylin OS.

After enabling PAT optimization for C86, HCI modifies and sets the memory

page attribute of the virtual display driver cache on Linux VMs in the C86 architecture to write-back, thereby enhancing display performance, reducing latency in operations and display, and ensuring a better user experience for VMs. For example,

HCI increases the VM density of Kylin OS by 10% while maintaining the same experience.

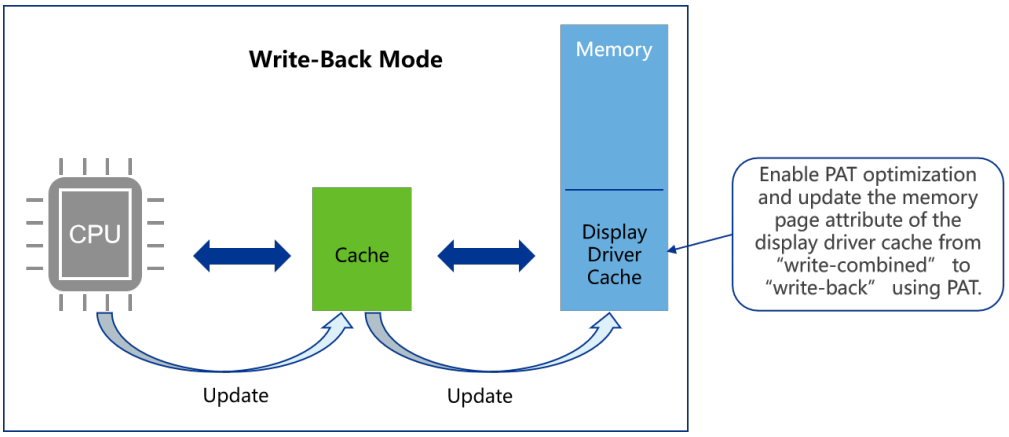
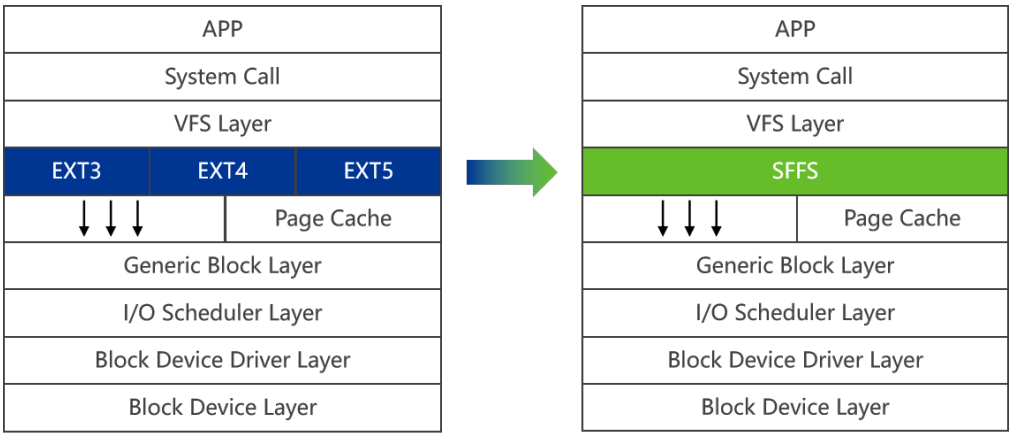


Figure 131: Access of CPU to Memory in Write-back Mode

5.1.6.11 Enhanced File System Reliability for Kylin OS

Kylin OS was mainly used for physical servers with a standalone file system. However, after HCI adopts Kylin OS as the OS of the underlying host, the kernel needs to support distributed cluster file systems. As a result, adaptations and optimizations tailored for the Kylin OS are made to the existing HCI SFFS.



Kylin File System Structure

SFFS File System Structure

Figure 132: File System Optimization and Enhancement

Specific optimizations include:

1. The kernel is adapted and optimized to support not only the distributed cluster file system SFFS but also external storage such as iSCSI and FC.
2. The reliability of the file system is enhanced by addressing issues such as lock contention and system crashes due to out-of-bounds memory access on Kylin OS.
3. Read-only monitoring for system disk I/O is introduced to effectively prevent system crashes during read-only I/O operation.
4. The reliability of the CIFS and SCSI protocol modules is enhanced by fixing system crashes due to out-of-bounds memory access in CIFS and SCSI of the kernel.
5. The reliability of Filesystem in Userspace (FUSE) is enhanced by fixing system crashes due to out-of-bounds memory access in the FUSE driver.

HCI ensures the stability and reliability of file systems with these targeted optimizations.

5.1.6.12 Enhanced Kernel Module for Kylin OS

When Kylin OS is used as a standalone OS, the use of cgroups is not required for the management of underlying resources such as CPU, memory, network, and disk I/O resources. However, the service memory isolation mechanism is designed to isolate the services in HCI, including management services, storage services, network services, and compute services, avoiding affecting each other in memory resources. For example, if the memory usage of compute services in memory overcommitment scenarios exceeds the allocated limit, such services cannot use the memory resources of other services, thus ensuring the stability of other services and HCI.

HCI uses the Linux cgroups mechanism, an underlying foundation for virtualization resource management and control, for memory isolation. This mechanism implements the abstraction of the host resources into the following resource groups (a resource group is a collection of resources

including CPU resources, memory resources, and I/O resources): HCI resource group, storage resource group, network resource group, and compute resource group. Memory resources for different services are managed through these resource groups, thereby restricting and ensuring the supply of memory resources for services accordingly, as shown in the following figure.

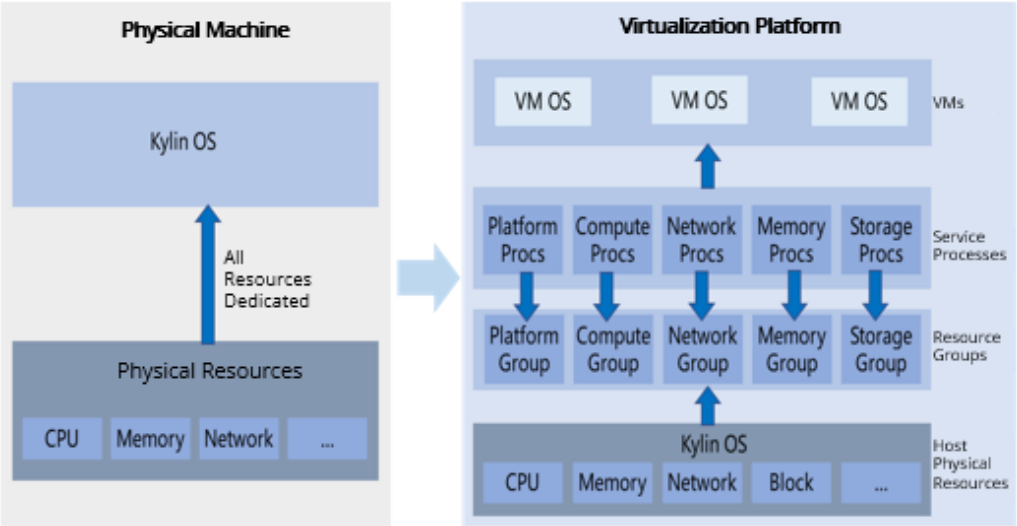


Figure 133: Cgroups for Resource Management

Cgroups play a significant role on virtualization platforms. Therefore, HCI enhances and optimizes the cgroups in the Kylin kernel to support the following features:

1. Memory overcommitment
2. Memory deallocation in groups

For memory scheduling and migration and KSM, the two frequently used features in virtualization scenarios, HCI also enhances and optimizes the corresponding modules in the Kylin kernel. The following describes the specific enhancements and optimizations:

1. System crashes because of memory migration between NUMA nodes are fixed.
2. For KSM, system crashes due to out-of-bounds memory access because of the reuse of released memory and the issue of negative values in KSM memory-saving parameters during memory merging are fixed.

3. For KSM, the memory merging feature in the x86 architecture developed by Sangfor is incorporated, enhancing memory page merging capabilities.

5.1.6.13 Peripheral Driver Adaptation and Enhancement for Kylin OS

In the scenario where Kylin OS is used as the underlying host OS, it imposes higher adaptation requirements for peripheral drivers such as HBA cards, NICs, RAID cards, and SAS cards than in the scenario where Kylin OS is used on a single physical server. For example, a single-core CPU may experience performance bottlenecks when handling packets from multiple NICs, as described above. To address this issue, Sangfor HCI has made extensive peripheral adaptations, adjustments, and optimizations in the x86 architecture. As a result, HCI can optimize and enhance such peripheral drivers for Kylin OS based on such adaptation capabilities.

Specific optimization measures include expanding the kernel compilation modules in HCI to enhance its adaptation capabilities for peripherals and providing a framework to facilitate the upgrade management of the kernel and third-party peripheral drivers.

Furthermore, to ensure the compatibility of HCI with peripherals, HCI fixes the following typical issues identified during its adaptation to commonly used peripherals:

1. System crashes during adaptation to Intel X710 NICs
2. System crashes during use. Solution: Upgrade the negbe driver for Netswift 1 GE NICs.
3. System crashes due to out-of-bounds memory access caused by the HiSilicon HNS3 NIC driver
4. Incompatibility with NICs of new models. Solution: Upgrade the Mellanox NIC driver.
5. System crashes during the use of HBA, RAID, and SAS cards and hardware compatibility issues. Solution: Upgrade the qla2xxx, MegaRAID, and smartpqi drivers.

6. System crashes and incompatibility with new NIC hardware. Solution: Upgrade Intel NIC drivers (i40e, ice, igb, and ixgbe drivers) in the C86 scenario.
7. System crashes due to out-of-bounds memory access caused by the SES driver

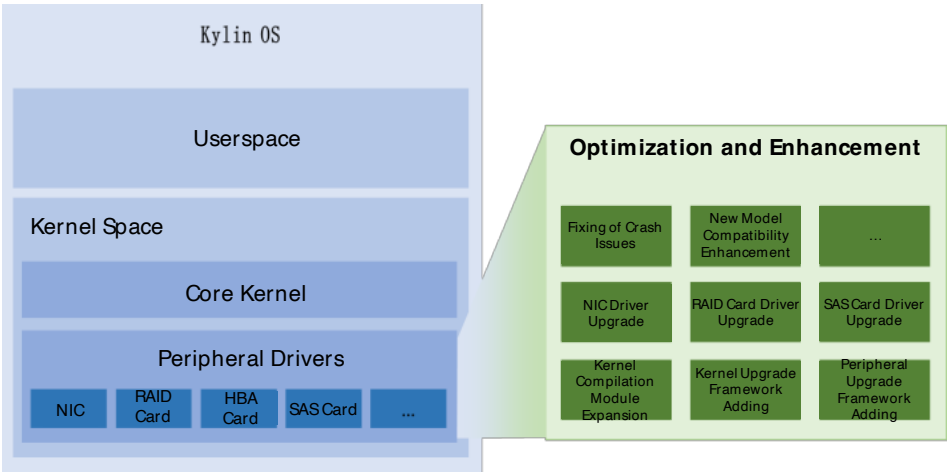


Figure 134: Peripheral Driver Enhancement

5.1.6.14 Advanced CPU Instruction Set Support

5.1.6.14.1 Technical Principles

A vCPU model is the type of virtual processor allocated to a VM in a virtualized environment. Different vCPU models have different processing capabilities and support different instruction sets, and those in recent releases typically offer superior performance and feature sets.

In earlier versions, HCI supported vCPU models such as core2duo, Haswell-noTSX, and phenom, where core2duo (for Intel and AMD) and phenom (for Hygon) were used by default. As technology develops, core2duo and phenom have become relatively obsolete with poor performance. HCI 6.11.1 is added with support for multiple new vCPU models and can adaptively select the default vCPU model for the cluster based on the characteristics of the physical CPU and the vCPU models of the VMs in the following orders, enabling the optimal configuration of the system in terms of performance, security, and

compatibility under the default conditions, providing customers with a comprehensive business guarantee.

- **Intel**: Cascadelake-Server-noTSX > Icelake-Server-v4 > Haswell-noTSX > core2duo.
- **AMD**: EPYC > EPYC-Milan > phenom.
- **Hygon**: EPYC > phenom > Dhyana.

Since vCPU emulation is inherently dependent on the characteristics of the physical CPU, the addition and deletion of physical hosts or the import and migration of VMs may require a change to the default vCPU model of the cluster or the vCPU model of the VMs, and such changes are subject to secondary confirmation.

For customization needs, users can make adjustments on the **Edit Virtual Machine** page or the **Default vCPU Settings** page. Additionally, VMs can be configured to use the CPU from the host, but they may encounter problems such as live migration failures due to different CPU models of the source and destination hosts.

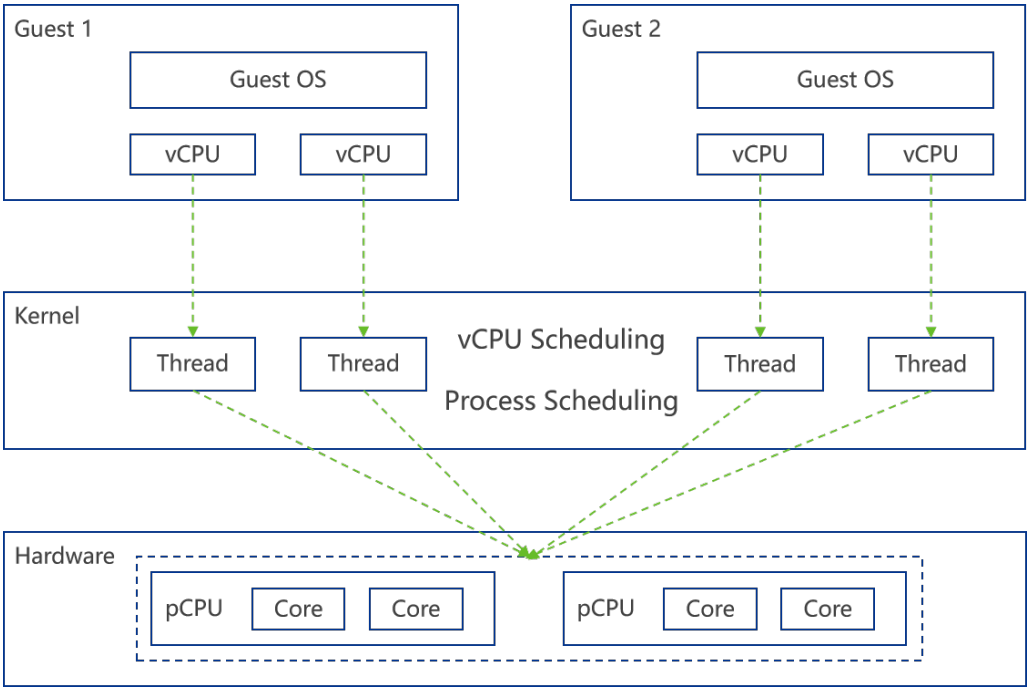


Figure 135: CPU Instruction Set Architecture

5.1.6.14.2 UI Display Guide

Log in to HCI and go to **System** > **Advanced** > **Default vCPU Settings** to select the default vCPU model for the cluster.

Select the vCPU model when creating and editing a VM.

A VM cannot be migrated to a host in the cluster that does not support the vCPU model of the VM.

Cold migrating a VM to a destination cluster that does not support the vCPU model of the VM will change the vCPU model to the default vCPU model of the cluster.

A host with a supported vCPU model that is lower than the default vCPU model of a cluster but higher than or equal to the vCPU models of all the VMs in the cluster can be added to the cluster, but an alert will be generated informing that this operation may lower the default vCPU model of the cluster.

5.1.7 Business Deployment

aSV provides multiple approaches for business deployment. The following describes the specific approaches.

5.1.7.1 VM Import and Export

aSV provides VM import and export features in business deployment and transition scenarios. Currently, it supports import and export either in OVA or VMA format.

- **VMA**: It is the VM import and export format dedicated to aSV, with no disk conversion or virtualization compatibility processing required. It features fast processing and excellent virtualization compatibility for migration between different aSV clusters. However, VMs cannot be imported into other virtualization platforms.
- **OVA**: It is a standard, cross-platform VM image format for import and export.

The following figure shows the processes of importing and exporting VMs in different formats.

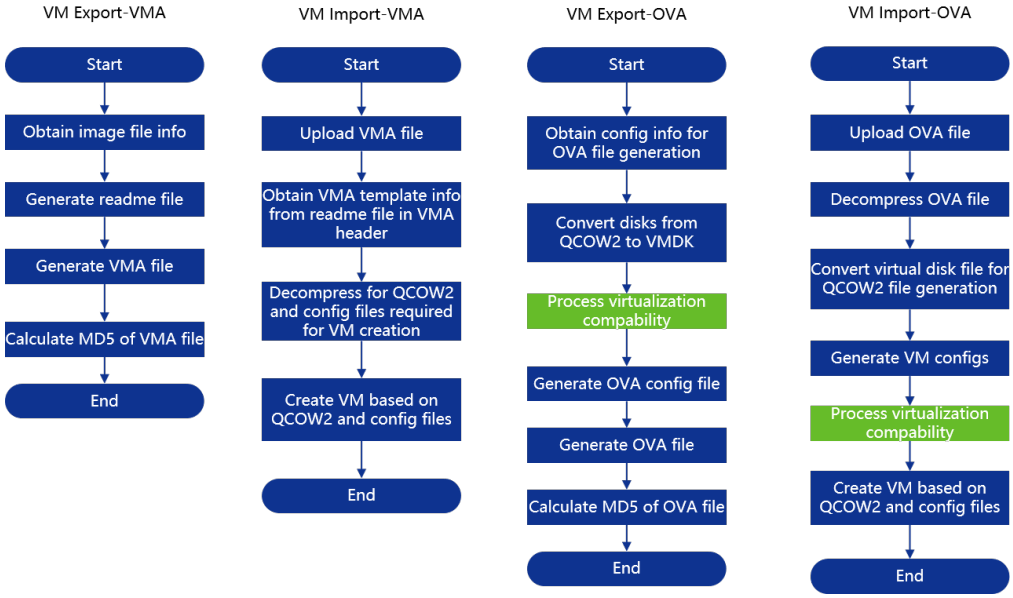


Figure 136: Processes of VM Import and Export in OVA and VMA Formats

During image import between virtualization platforms in OVA format, some pre-processing may be required because the virtualization technology varies with the platform. For example, the disk controller hardware ID internally identified by the Windows OS may vary with the platform. If no processing is performed, the disk cannot be initialized based on the original hardware ID of the disk controller, resulting in errors such as a 7B blue screen.

5.1.7.2 P2V and V2V Deployment

Sangfor provides a set of quick migration solutions tailored for different scenarios to facilitate customers' cloud-based transformation of existing businesses. For businesses currently running in physical environments, the physical to virtual (P2V) approach can be employed for migration to aSV. For businesses running on other VM platforms such as VMware, Citrix XenServer, and Hyper-V, the virtual to virtual (V2V) method is available for migration.

Sangfor cloud migration tool (SCMT) allows customers to perform P2V or V2V migration of their business VMs. To accommodate their specific environment

and business requirements, Sangfor supports the following migration modes:

- **Point-to-point migration** (without proxy): It is specifically designed for VMware, which allows migration without the installation of a migration client on the source VM. In this mode, the VMware VADP interface is used to obtain data. After migration to the destination VM running in HCI, automatic compatibility processing (injection of vmTools) can be performed. This method eliminates the need for backing up the source system because the source data is transmitted to the destination VM through the Sangfor cloud migration platform.
- **Point-to-point migration** (with proxy): In this mode, the source data is directly transmitted to the destination VM by using the migration feature so that the entire source system is migrated to the destination VM. This method eliminates the need for backing up the source system because the source data is transmitted to the destination VM running in HCI.
- **HA migration**: In this mode, the source data is continuously backed up to the Sangfor cloud migration platform by using the CDP backup feature and synced to the destination VM in real time. This minimizes the migration and switching duration.
- **Backup migration**: In this mode, the source data is backed up to the Sangfor cloud migration platform by using the backup function, and the source system is restored to the destination VM through backup files on the server. Compared to point-to-point migration and HA migration, this method has a longer business downtime.

5.1.7.2.1 SCMT Architecture

SCMT consists of the following three main parts. The following describes their functions.

- Sangfor cloud migration platform

1. Facilitates the connection between the source end and destination end and sends migration tasks.

2. Connects to the source virtualization platform during migration without proxy, and obtains data from the source VM.
3. Connects to the HCI management plane and calls the HCI API for automatic creation of the destination VM.

- Migration client

1. Installed on the source host or VM for connecting to the migration platform and obtaining migration tasks sent by the migration platform.
2. Obtains data from the source host or VM and transmits it to the destination VM or Sangfor cloud migration platform.

- Boot media for bare-metal recovery

1. Facilitates the startup of destination VM, which is then connected to the Sangfor cloud migration platform.
2. Receives migration tasks sent by the Sangfor cloud migration platform.
3. Receives data transmitted from the source host or VM, or data restored by the Sangfor cloud migration platform.

The SCMT network architecture varies with the migration mode. The following describes the SCMT network architectures in different migration modes.

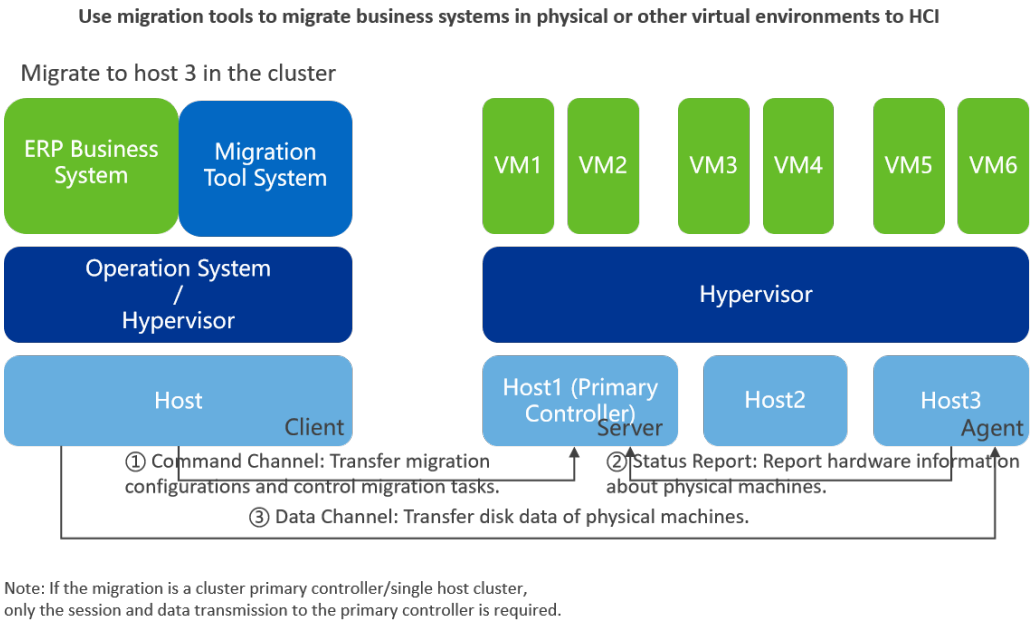


Figure 137: Flowchart of Migration by Using SCMT

1. The OS that loads SCMT allows the physical machine to enter the client mode and communicate with the server (primary controller) and agent (migration destination host).
2. After successful permission verification, the configuration transmission is performed, and the agent collects the hardware information about the physical machine and transmits it to the aSV primary controller through the command channel.
3. Following a series of negotiations, a data channel is established between the client and the agent, and the migration enters the data copy phase, where SCMT optimizes data transmission for copy as required. It analyzes the file system formats of all partitions on the disk, generates a distribution bitmap of user data based on the file distribution in the partitions, and copies data accordingly. Therefore, data is copied at locations where data exists, and the amount of copied data is the size of the actual disk space occupied by the user. Data loss is prevented through technologies such as retransmission and resumption from breakpoints.

Highlights of migration by using SCMT:

1. In a Windows environment, users can directly use the VM generated to take over the businesses of the original system after P2V migration without reconfigurations such as IP address reconfiguration. In addition, the VM is equipped with vmTools for operation in high-performance mode.
2. Through the identification of file system information in the original system, data is copied at locations where data exists, which significantly reduces the time required for disk copy. This mechanism works in a similar way as the cloning and backup software Ghost. After the migration, the VM occupies disk space whose size is equivalent to that of the actual user data.
3. Because of changes to the VM and physical machine hardware, it is not possible to simply clone and import the OS by using software such as Ghost, which may prevent the OS from starting. As a result, during the P2V and V2V migration, SCMT performs compatibility processing on the

new OS after copy to ensure that the system starts properly after migration.

4. Migration by using SCMT is applicable to unattended scenarios. This solution supports retransmission and resumption from breakpoints. In the event of network disconnection, resumption from breakpoints is initiated automatically, eliminating the need to resume.

NOTICE

1. The ISO migration solution can also be used to migrate Windows systems. Its advantage lies in resolving compatibility issues during the migration. However, it has the drawback of losing network configurations and requiring reconfiguration. This solution can be used as an alternative for migrating Windows systems such as Windows 2000 when Sangfor aSV Convertor is not available for Windows system migration.

2. The current version supports analysis and optimized copies of file systems, including FAT, FAT16, FAT32, NTFS, EXT2, EXT3, EXT4, and XFS.

5.1.7.2.2 Point-to-Point Migration (Without Proxy)

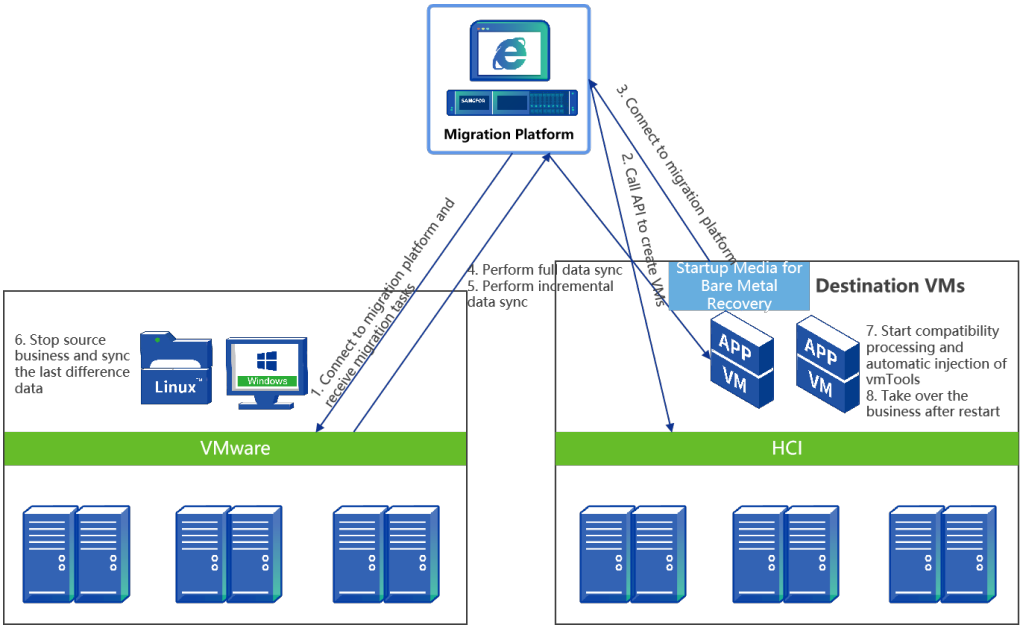


Figure 138: Process of Point-to-Point Migration (Without Proxy)

Process:

1. The Sangfor cloud migration platform connects to the source virtualization platform VMware, obtains the VM list, and sends migration tasks.
2. The Sangfor cloud migration platform calls the HCI API to create the destination VM and automatically loads the boot media for bare-metal recovery.
3. After the IP address of the boot media for bare-metal recovery is changed, the destination VM proactively connects to the Sangfor cloud migration platform for obtaining migration tasks and receiving data from VMware.
4. Full and incremental data syncs between the source end and the destination end are realized through the Sangfor cloud migration platform.
5. The source businesses are stopped, and business switching is performed after syn of the final differential data.
6. The destination VM initiates compatibility processing and is automatically injected with vmTools.
7. The destination VM restarts and takes over the businesses.

5.1.7.2.3 Point-to-Point Migration (with Proxy)

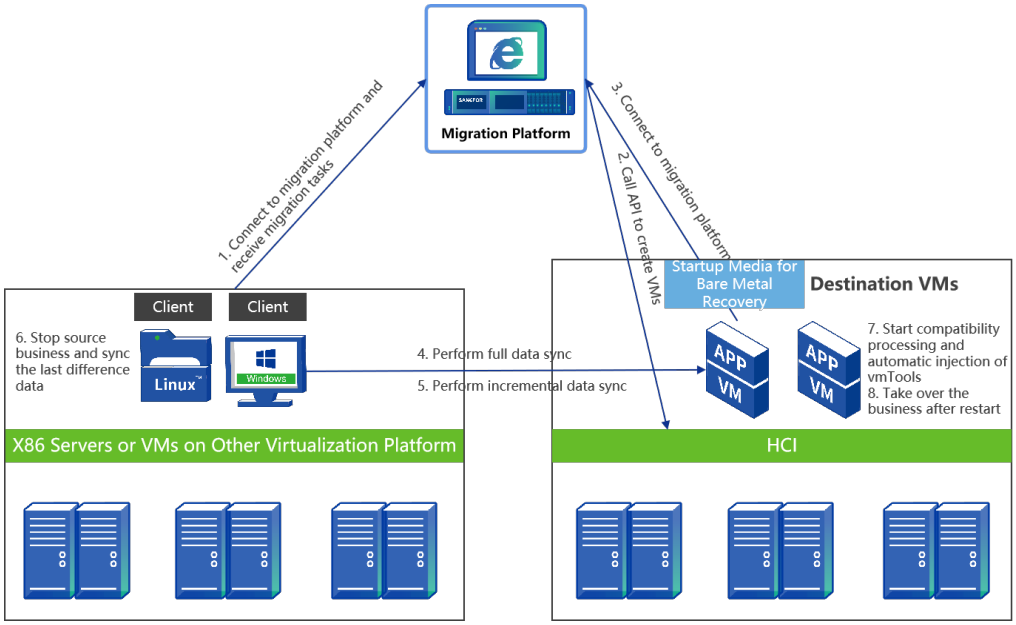


Figure 139: Process of Point-to-Point Migration (with Proxy)

Process:

1. The migration client, which can operate in either proactive or passive mode, is installed on the source host.

- In proactive mode, the migration client on the source host proactively connects to the Sangfor cloud migration platform and receives migration tasks sent from the migration platform.
- In passive mode, the Sangfor cloud migration platform proactively connects to the migration client on the source host and sends migration tasks.

2. The Sangfor cloud migration platform calls the HCI API to create the destination VM and automatically loads the boot media for bare-metal recovery.
3. After the IP address of the boot media for bare-metal recovery is changed, the destination VM proactively connects to the Sangfor cloud migration platform for obtaining migration tasks and receiving data from the migration client on the source host.
4. Full data sync is performed.
5. Incremental data sync is performed.
6. The source businesses are stopped, and business switching is performed after syn of the final differential data.
7. The destination VM initiates compatibility processing and is automatically injected with vmTools.
8. The destination VM restarts and takes over the businesses.

5.1.7.2.4 HA Migration

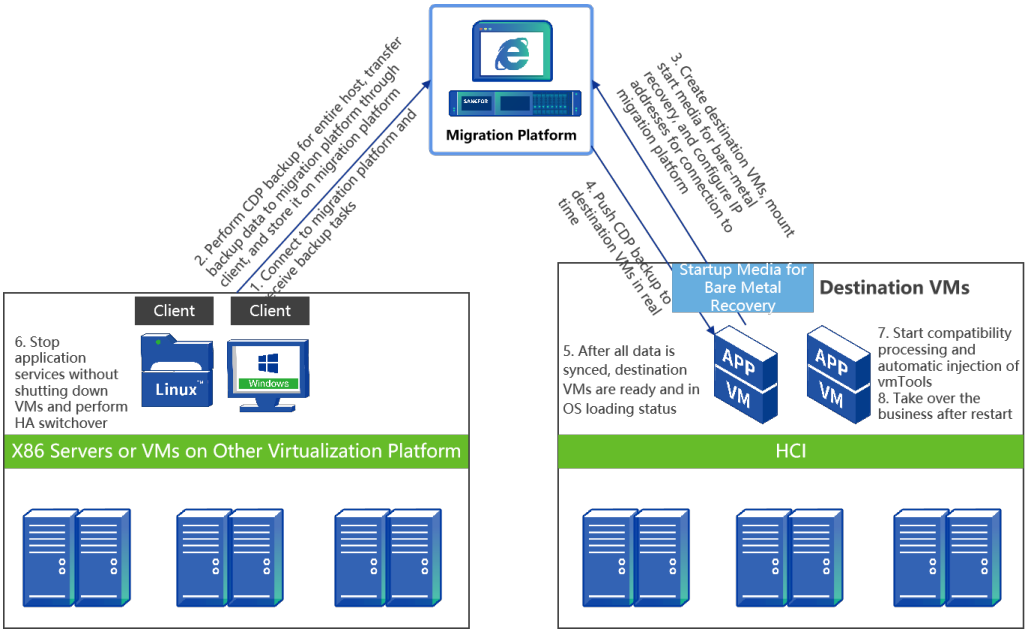


Figure 140: Process of HA Migration

Process:

1. The migration client, which can operate in either proactive or passive mode, is installed on the source host.

- In proactive mode, the migration client on the source host proactively connects to the Sangfor cloud migration platform and receives migration tasks sent from the migration platform.
- In passive mode, the Sangfor cloud migration platform proactively connects to the migration client on the source host and sends migration tasks to the source host.

2. The source host performs a CDP backup of the entire machine and stores the backup data on the Sangfor cloud migration platform.
3. The destination VM is manually created, and the ISO file of the boot media for bare-metal recovery is manually mounted for VM startup. The destination VM is configured with an IP address and connects to the Sangfor cloud migration platform.
4. A HA plan is created, and the Sangfor cloud migration platform pushes the CDP backup of the host to the destination VM in real time.

5. After all data is synced, the destination VM is ready and in a state of loading the OS.
6. The source application services are stopped, but the source host is not shut down, and the HA switching plan is executed.
7. The destination VM initiates compatibility processing and is automatically injected with vmTools.
8. The destination VM restarts and takes over the businesses.

5.1.7.2.5 Backup Migration

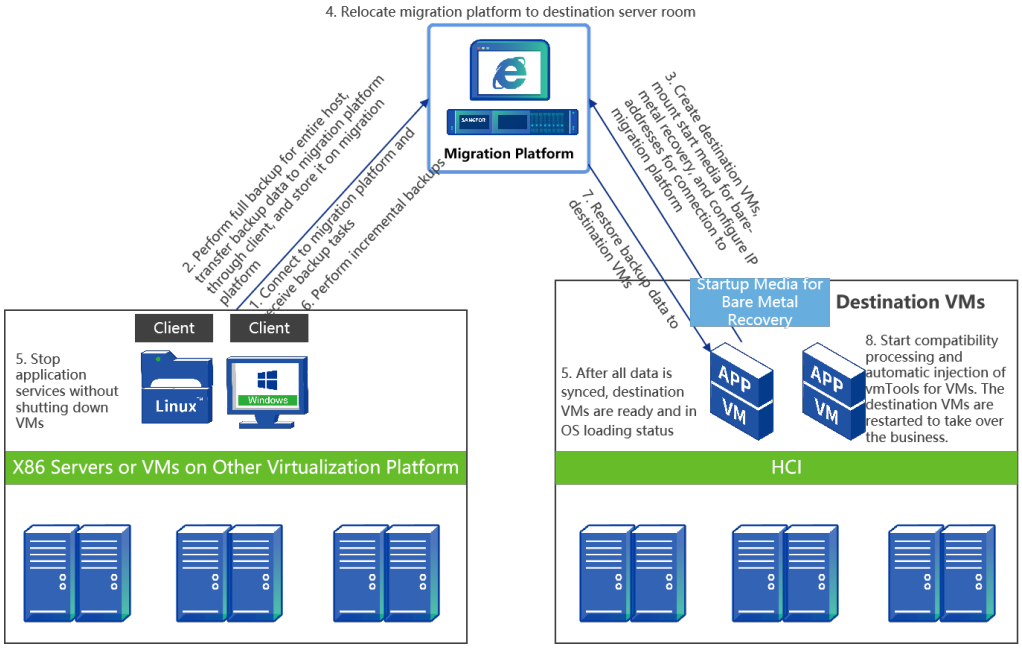


Figure 141: Flowchart of Backup Migration

Process:

1. The migration client, which can operate in either proactive or passive mode, is installed on the source host.

- In proactive mode, the migration client on the source host proactively connects to the Sangfor cloud migration platform and receives migration tasks sent from the migration platform.

- In passive mode, the Sangfor cloud migration platform proactively connects to the migration client on the source host and sends migration tasks to the source host.

2. The source host is fully backed up to the storage on the Sangfor cloud migration platform.
3. The Sangfor cloud migration platform is relocated to the destination server room.
4. The destination VM is manually created, and the ISO file of the boot media for bare-metal recovery is mounted for VM startup. The destination VM is configured with an IP address and connects to the Sangfor cloud migration platform.
5. The source application services are stopped, but the source host is not shut down.
6. The source host performs an incremental backup.
7. The backup data is restored to the destination VM.
8. The destination VM initiates compatibility processing, is automatically injected with vmTools, restarts, and takes over the businesses.

5.1.7.3 Template-Based Deployment

"Templates" are a feature provided by HCI to meet the requirements for bulk operations on VMs in virtual desktop scenarios. This feature allows users to easily deploy or clone multiple VMs based on one VM, thereby alleviating the burden on administrators. In most cases, templates are stored in shared storage to facilitate their use by administrators.

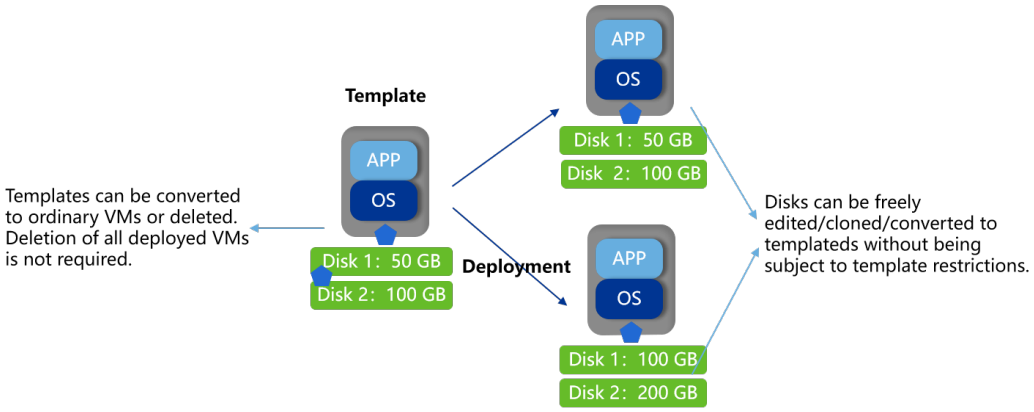


Figure 142: Template-Based Cloning Process

The process of VM creation based on templates is as follows:

1. The user creates a VM.
2. The VM is set as a VM template.
3. The user can perform various operations to update the VM template, including installation, configuration, verification, upgrade, and service pack installation.
4. The VM template is set as the base image.
5. More VMs are created based on the VM template, which is actually a cloning process based on the base image corresponding to the template. Full clone and fast full clone are supported, with aSAN as the backend storage.
6. The user creates the VM template and updates the configuration to support multiple disks, whose size can be reconfigured.
7. The user creates more VMs through the VM template.
8. VMs created through the template are completely independent of the template. More specifically, the deployed VMs do not rely on the original template, and the disks can be edited freely. The original template can be deleted or converted into a common VM for use without the deletion of all deployed VMs.

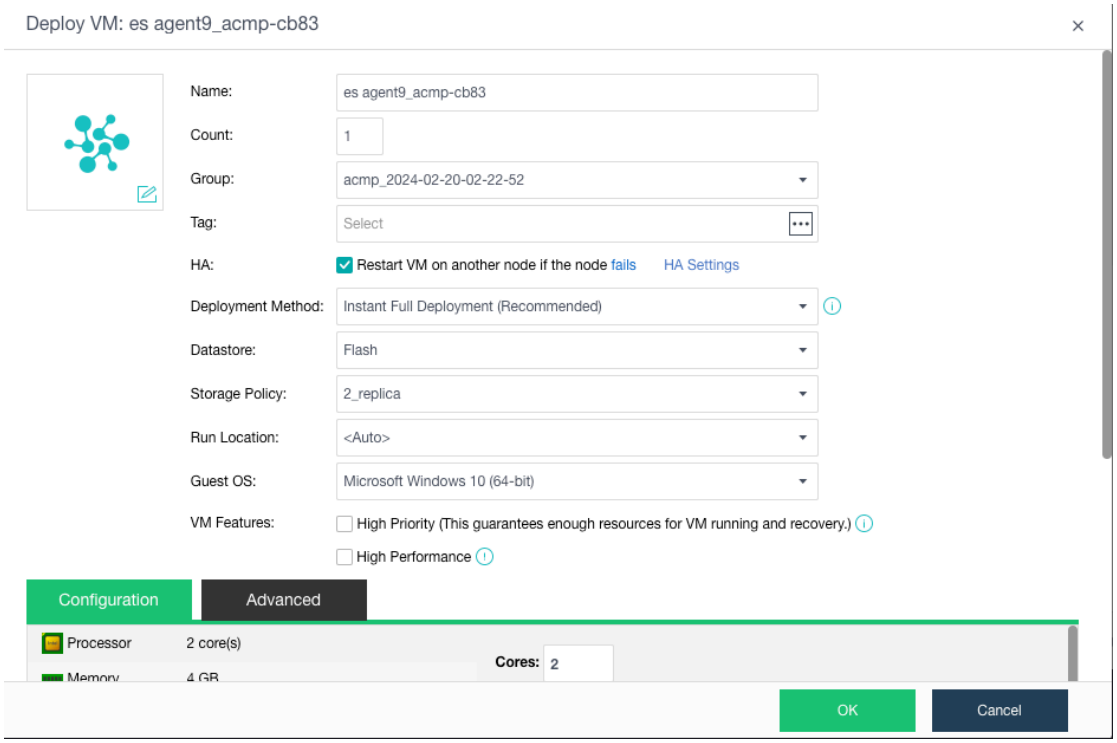


Figure 143: Template-Based Deployment Page

Template-based deployment involves converting a shut down current VM into a VM template and cloning a specified number of VMs whose status is the same as that of the current template. All cloned VMs have a base image identical to that corresponding to the template and a private incremental image. Such VMs can be configured with personal data disks based on the VM template.

Multi-disk VMs can be converted into templates. VMs cloned from templates are independent of the original templates, and the disks can be freely edited. Furthermore, the original templates can be deleted or converted into common VMs for use without the deletion of all cloned VMs.

5.1.7.4 Specifying IP Addresses or Hostnames

HCI supports specifying and editing VM IP addresses and hostnames during VM creation. It also facilitates bulk IP address assignment by entering an IP pool and bulk hostname assignment by entering information in Hostname + No. format during template-based deployment. In addition, the IP addresses assigned by HCI can be changed on the VM.

These features are implemented by passing HCI information through the vmTools predefined on the VM in terms of technology. When the VM starts, HCI transfers the configuration file information to the VM through vmTools. vmTools then modifies the corresponding configuration to complete the parameter settings.

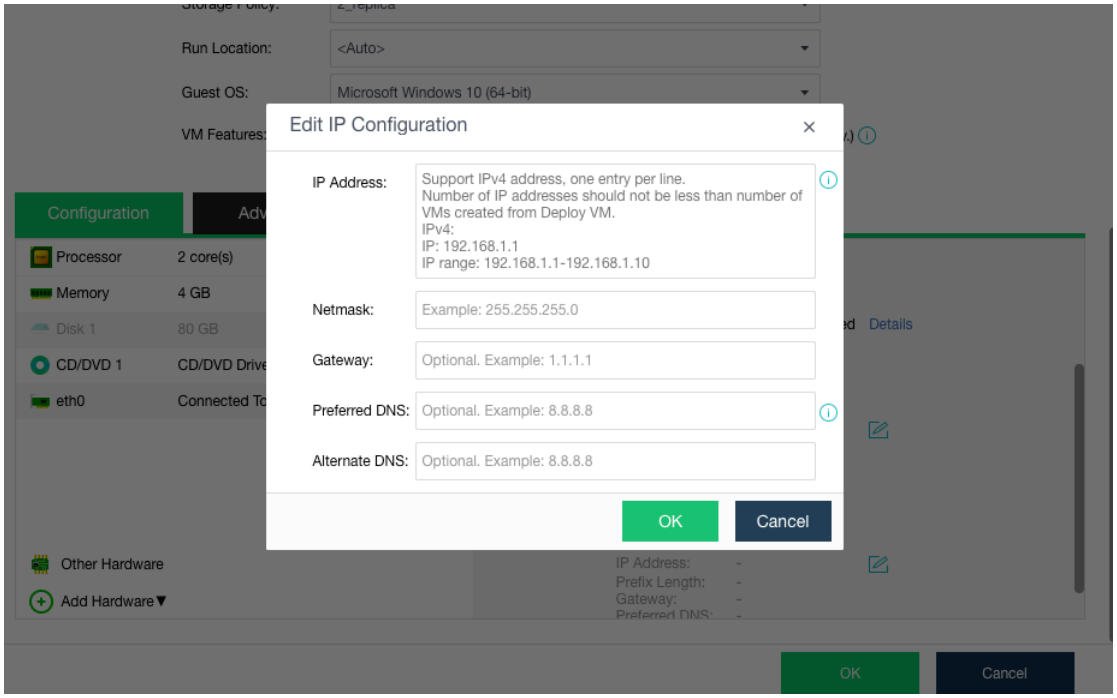


Figure 144: Specifying the VM IP Address

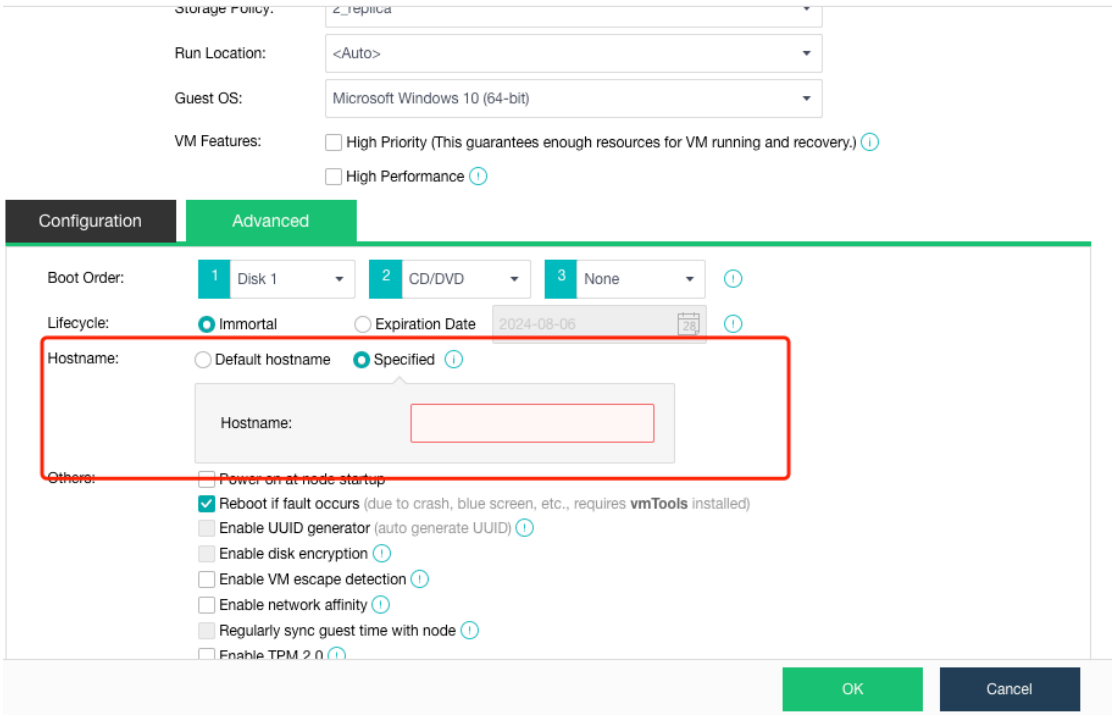


Figure 145: Specifying the VM Hostname

5.1.7.5 VM Scheduling Policy

The running host location of a VM can be set based on scheduling policies. When the running location of a VM is set to be automatically selected, HCI assesses the remaining resources of the hosts and the resources required by the VM and selects an optimal host for running the VM.

5.1.7.5.1 Introduction to Groups

A group is a collection based on various attributes of VMs.

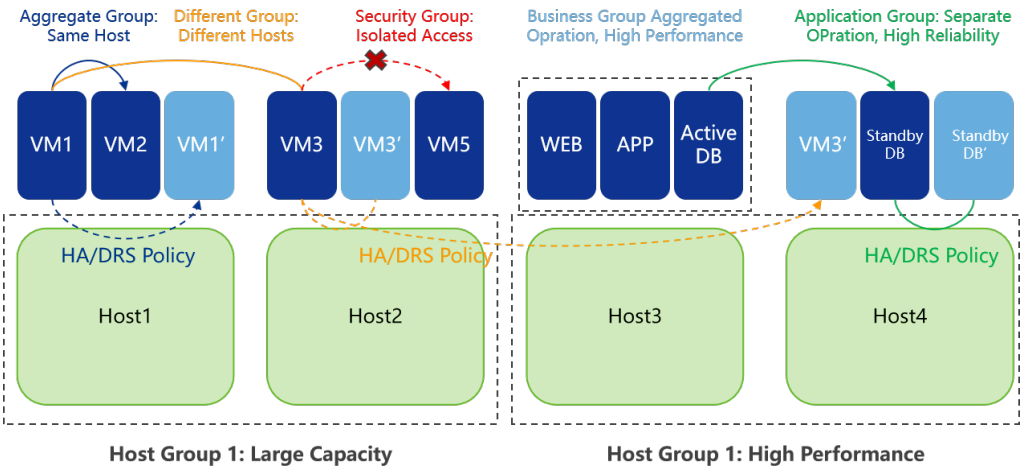


Figure 146: VM Group and Host Group

1. Scheduling group (anti-affinity and affinity)

Scheduling groups are created from the perspective of scheduling the running locations of a group of VMs (1 to N). These groups are classified into affinity (on the same server) and anti-affinity (not on the same server) modes. That is, these groups are created based on the scheduling policies for VMs during startup or DRS.

2. Business group

Business groups are created from the perspective of businesses. For example, a typical three-layer architecture comprising webs, apps, and databases can be considered as a business group. VMs within a business group may be tightly coupled (affinitive) or anti-affinitive. Therefore, VMs may need to run on the same host to improve access efficiency or run on different hosts to ensure high reliability.

3. Application group

An application group consists of specific applications of the same category. For example, the primary database and secondary database can be in the same application group, and a business group may consist of multiple application groups.

4. Host group (fault domain and scheduling domain)

In fault domains, hosts are grouped with the aim of guaranteeing the system

reliability. For example, a fault domain of a stretched cluster comprises 1 to *N* hosts, and each fault domain provides redundancy for another one. In scheduling domains of a cluster, hosts are grouped by function or performance within a cluster.

5. Security group

A security group is a set of VMs that are grouped to implement security features.

5.1.7.5.2 Application Scenarios of Scheduling Groups

Anti-affinity:

Anti-affinity policies should be set for VMs that are deployed in a cluster (including a two-node cluster) to ensure the high availability of these VMs and to avoid a single point of failure caused by VMs running on the same host. Such clusters include Oracle RAC, Spark cluster for big data components, ZooKeeper cluster, and DataNodes that connect to aSAN but do not enable Hadoop Virtualization Extensions (HVE).

To ensure high availability in such service scenarios, the platform will automatically generate anti-affinity conditions by default when these clusters are deployed.

Anti-affinity and affinity:

Take Hadoop DataNode grouping as an example. The following anti-affinity and affinity policies are configured to guarantee the replication mechanism-based high reliability of DataNodes with HVE enabled:

1. DataNodes on the same VM rack must run on the same physical host.
2. DataNodes on different VM racks must run on different physical hosts.

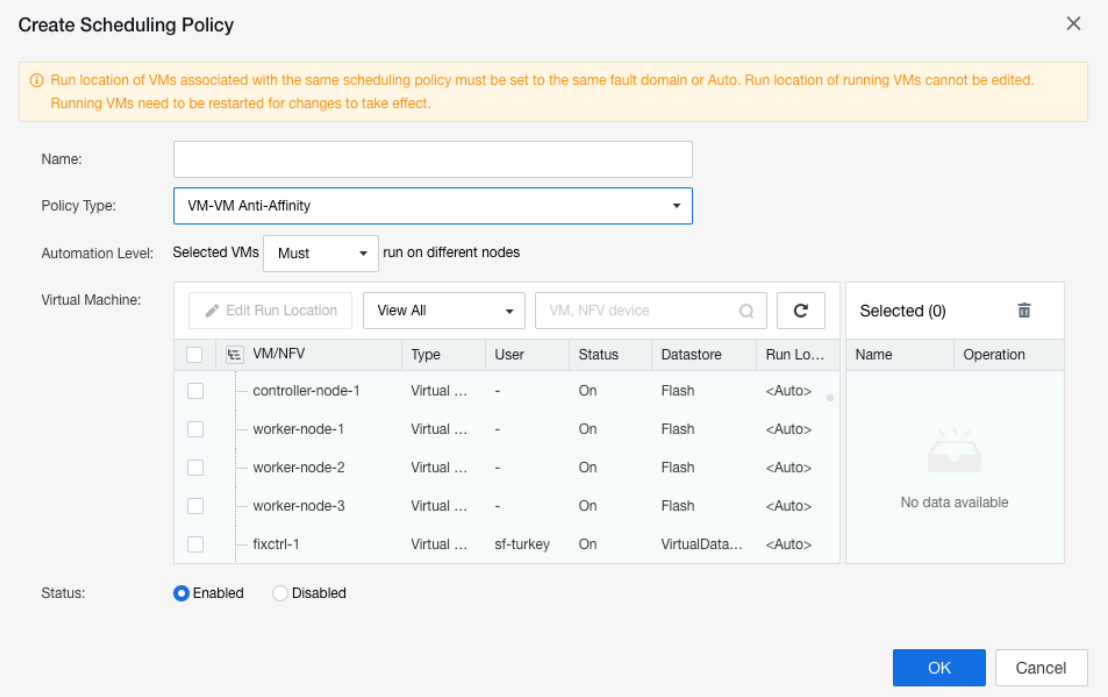


Figure 147: Scheduling Group - Scheduling Policy Settings

5.1.7.5.3 Application Scenarios of Application Groups

Nodes of the same type must be added to the same application group, and the scheduling policy must provide the following capabilities to support HVE:

1. Affinity: VMs in the same application group must run on the same host.
2. Anti-affinity: VMs in different application groups must run on different hosts.

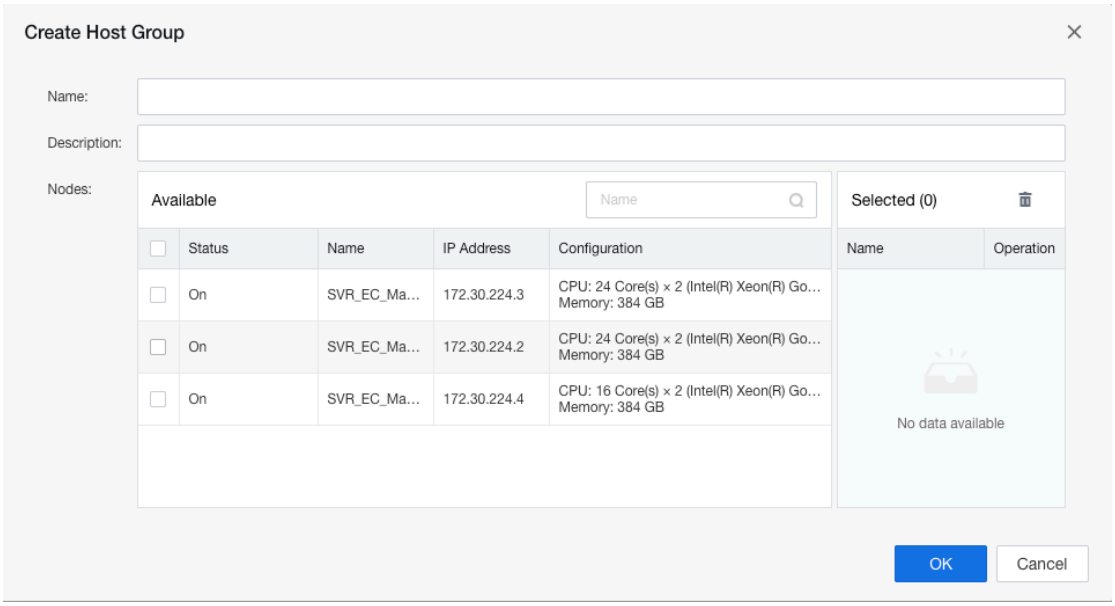


Figure 148: Creating an Application Group

5.1.7.5.4 Application Scenarios of Host Groups

An integrated healthcare platform is deployed in a large cluster, which consists of both high-performance and intermediate-performance hosts. The business VMs are deployed in distributed mode to ensure high availability. Business VMs with high performance requirements must run on high-performance hosts and do not need to be switched to intermediate-performance hosts through the HA mechanism even if host failures occur, preventing VMs running on intermediate-performance hosts from being impacted. Therefore, a high-performance host group and an intermediate-performance host group must be created separately.

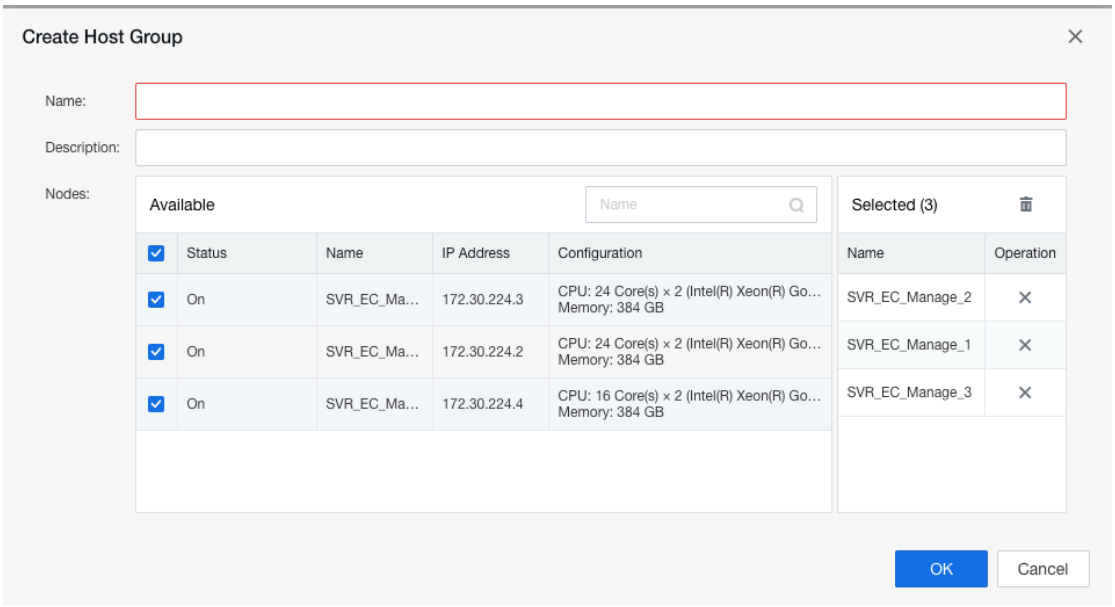


Figure 149: Creating a Host Group

5.1.7.5.5 Application Scenarios of Security Groups

For details, see section 5.3.5.4 "**Error! Reference source not found.**."

5.1.7.5.6 Scheduling Policy Management

When a big data component VM is deployed (including in the scenario of capacity expansion), it is automatically added to the corresponding scheduling policies.

When a DataNode VM is deleted, it is automatically deleted from the corresponding scheduling policies.

NOTICE

Terms:

1. HVE

HVE is provided by Hadoop for distribution policies of data node replicas in virtualization scenarios. The advantages of HVE are as follows:

A. A data block and its replica are not stored in DataNode VMs running on the same physical host.

B. Worker VMs running on the same physical host have higher network throughput and lower latency than those running on different physical hosts, and the scheduling policies can be employed to improve the performance of big data business processing.

2. DataNode

Anti-affinity and affinity are unavailable when pass-through disks are used for DataNodes because the disks are running at fixed locations.

5.1.7.6 Space Release upon Data Deletion

When files are deleted from a VM, the storage space occupied by the image files is released simultaneously, thereby avoiding the problem that storage space remains occupied even after files in the VM are deleted. This feature is available no matter whether internal or external storage is used.

5.1.7.6.1 Internal Storage Space Release

In the environment where the criteria for the discard function to take effect are met, when files are permanently deleted from a VM, the storage space occupied by the VM disk image files is released. The process is as follows:

1. The guest OS of the VM sends a discard I/O request for a data block.
2. When receiving the request from the frontend (guest), the backend (VirtIO device) finds the corresponding driver based on the image file type (qcow2) and uses the discard request processing method in the driver to send the discard I/O request.
3. The image file driver (qcow2) performs alignment based on the cluster size and sends the discard I/O request for the data block to the virtual storage where the files reside so that the space can be deallocated.
4. The storage deallocates the space based on the discard I/O request and returns the processing result to the qcow2 driver.
5. The qcow2 driver returns the processing result to the VirtIO device.
6. The VirtIO device returns the process result to the VM.

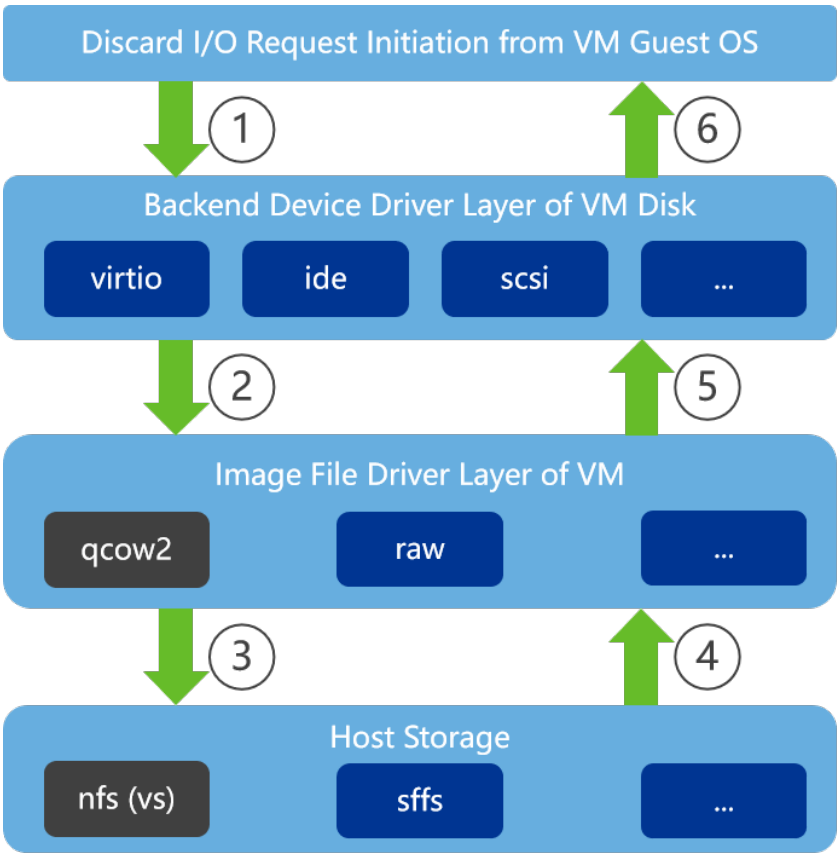


Figure 150: Process of Deallocating Space When Data Is Deleted from a VM

The discard function takes effect when the following criteria are met:

1. Dynamic provisioning or thin provisioning is used for VM disk image files. The discard function is not supported when pre-allocating is used.
2. The discard function is enabled for the guest OS file system. (On Windows OS, this function is enabled by default. On Linux OS, this function needs to be enabled by setting the specified parameter during partition mounting.)
3. The VM stores data in the virtual storage cluster consisting of three or more nodes.
4. Disk space deallocation is enabled by setting **Space Reclamation** on the **Edit Virtual Machine** page of the web console. By default, this function is disabled.

- Disk space deallocation is provided at the disk level.

- The QEMU discard function is enabled as long as disk space deallocation is enabled.
- QEMU supports the function of dynamically enabling and disabling space deallocation for a certain disk when the VM is running.

5.1.7.6.2 External Storage Space Release

LUNs are created in thin provisioning mode for external storage. Then, they are mounted to HCI using iSCSI and FC protocols and formatted as SFFS to provide image storage space for VMs. Customers can create and delete VMs on the external storage through HCI. When a VM or its image files is deleted, the storage space occupied by the image files is released, and the entire space of the storage to which SFFS is mounted is reorganized by deallocating the space of idle data blocks. The following describes the process of releasing external storage space upon file deletion:

1. When receiving a command to change the discard granularity, enable/disable the discard function, or process a discard I/O request, SFFS sends the command to the corresponding subprocess.
2. For the command to process a discard I/O request for space deallocation, SFFS sends the request to the kernel block layer for processing.
3. The kernel block layer sends the request to the storage end for space deallocation.
4. After processing the request, the storage end returns the result to the kernel block layer of HCI.
5. The kernel block I/O stack returns the result to SFFS.
6. SFFS returns the result to the upper layer.

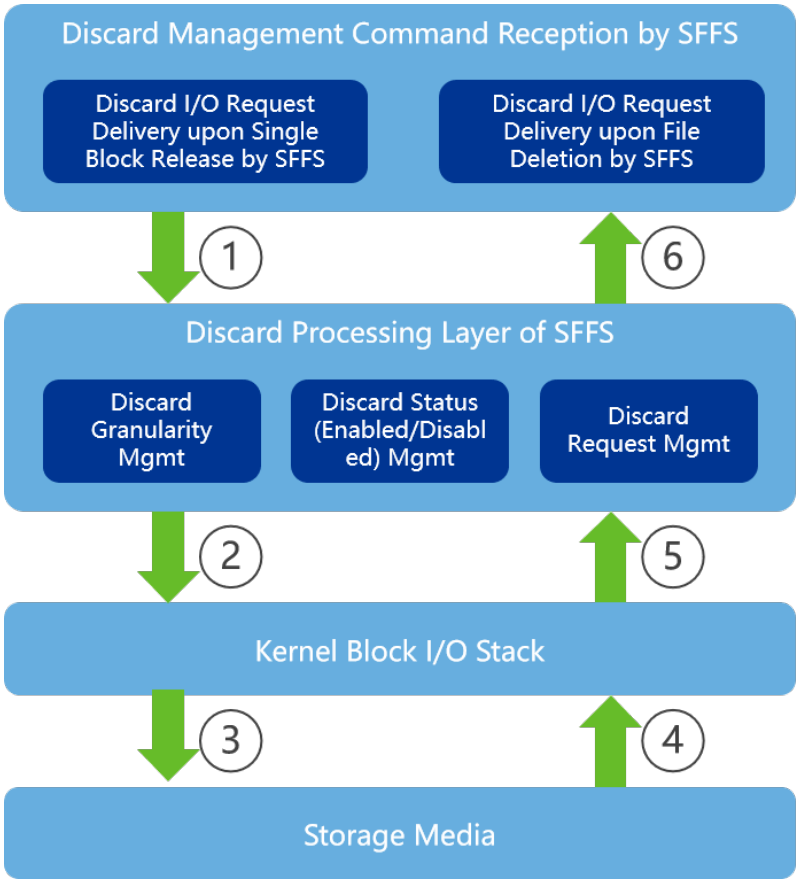


Figure 151: Process of Releasing Storage Space When Files Are Deleted from the External Storage

The discard function of SFFS takes effect when the following criteria are met:

1. External storage uses thin provisioning. Otherwise, space cannot be released even if the discard function is enabled.
2. External storage supports the discard function.

The discard function can be enabled, disabled, and configured on the web console of the HCI as follows:

1. The discard function can be enabled or disabled at the storage granularity. Discard is disabled by default.
2. When storage is added, the **discard** parameter needs to be set to specify whether to enable the discard function.
3. The status (enabled or disabled) and granularity of the discard function can be changed dynamically.

4. The entire space of the storage to which SFFS is mounted can be organized by deallocating the space of the idle data blocks.

5.1.7.7 VM Application Deployment Wizard

Sangfor and ISVs have tested the integration between HCI and applications rigorously from the aspects of compatibility, reliability, high performance, and deployment solutions, and ISV certifications have been issued. Therefore, customers can deploy their business applications in HCI without worries.

Based on the configuration recommended by the ISVs for each application and existing application deployment cases, Sangfor provides the best configurations for application deployment by distinguishing the type and user quantity of industry applications and establishing an industry application deployment wizard to accelerate the application launch process. In addition, Sangfor provides ISV application icons to facilitate VM management for customers. (**Applicable to Chinese Version Only**)



Figure 152: VM Application Deployment Wizard

5.1.7.8 Full Clone

Sangfor supports the VM clone function. Customers can clone VMs to improve the efficiency of business deployment. VMs can be cloned upon power-on or

shut-down. The two clone methods have only one difference. That is, when a VM is cloned upon power-on, a snapshot needs to be taken first, which will be used as a basis to clone configuration files and image files.

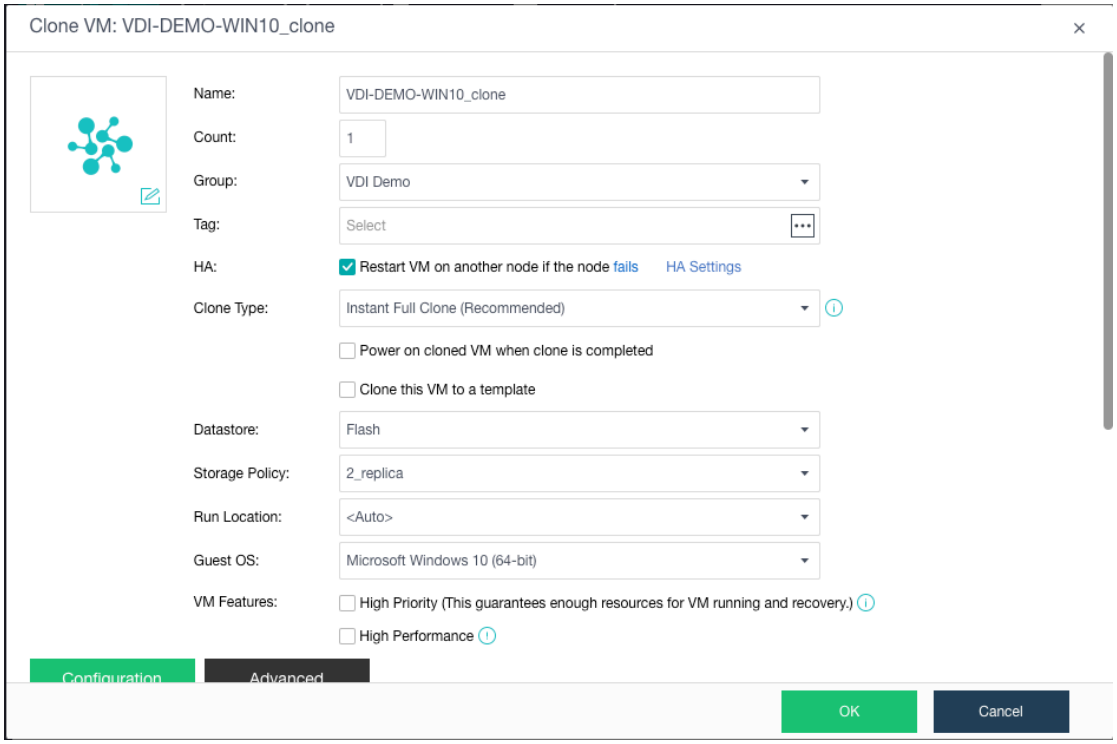


Figure 153: Clone Virtual Machine

5.2 aSAN

5.2.1 Overview

aSAN is a software-defined distributed storage system that can run on universal servers, including x86. It integrates storage resources within a cluster to provide block storage services externally, and external devices can be connected to aSAN by using the standard iSCSI protocol. (HCI guest VMs can use the proprietary storage protocol to connect to aSAN in the cluster to enjoy optimal performance and availability).

5.2.1.1 IOPS

IOPS refers to the number of input/output operations per second. It is mainly

used to evaluate the I/O performance of small data blocks, reflecting the I/O delay and concurrency of the storage system. IOPS is generally accepted as the I/O performance of 4 KB data blocks. A larger value indicates better performance.

5.2.1.2 Throughput

I/O throughput is measured in MB/s. It is mainly used to evaluate the I/O performance of large data blocks, reflecting the I/O bandwidth of the storage system. A larger value indicates better performance.

5.2.1.3 Cache Disk and Data Disk

SSDs have excellent IOPS and throughput. SATA SSDs used for common data centers on the market can achieve an IOPS of higher than 50,000 and a throughput of higher than 400 MB/s. PCIe SSDs feature even higher performance.

HDDs are known for their low price and large capacity, but their performance is relatively low. Especially, its random read/write IOPS is as low as several hundred.

Therefore, Sangfor generally uses hybrid disks for the storage system to realize the optimal balance between performance, capacity and cost. In detail, SSDs are used as cache disks for speeding up I/O reads and writes, and they store hot data that is frequently accessed in customer' businesses, while HDDs serve as the locations where data is persistently stored.

After a large number of statistical analyses on actual business scenarios, Sangfor has derived a normal distribution of hot data over all business data. Based on repeated validation, a referenced configuration of 20% hot data is calculated as the best practice. Therefore, it is recommended that a ratio of 20% be configured for the total cache disk capacity to the actual business data to achieve the best performance of aSAN. If the size and growing trend of business data cannot be accurately assessed, it is recommended that the ratio of the cache disk capacity to HDD capacity be configured as 10% or higher

when setting up the virtual datastore. If a low cache disk capacity is configured, it will result in intense competition among hot data for cache space resources. The cache disks will frequently import and export hot data, which will adversely impact the stability of the cluster and cause performance jitter.

For example, if a physical host is equipped with six 4 TB data disks, we suggest that at least 2.4 TB cache disks be configured. The capacity is calculated as follows: 10% × (6 × 4 TB) = 2.4 TB. Such configuration can satisfy the performance requirements of most businesses.

SSDs have been getting cheaper and larger in capacity recently. Therefore, customers have gradually accepted all-flash storage. A common all-flash storage solution is using SSDs as data disks without configuring cache disks. Another solution is to use extremely high-performance PCIe SSDs as cache disks and SATA SSDs as data disks.

5.2.1.4 Disk Group

If hybrid disks are used, cache disks and data disks are grouped, and each group consists of one SSD and *n* (n ≤ 7) HDDs. The SSD in each disk group only serves as the cache disk to accelerate operations for the HDDs in the disk group.

Such configuration can minimize the impact scope of SSD failures. When an SSD fails, the impact of the failure is limited to the disks in the disk group where the SSD is located, and only data stored in HDDs in the disk group instead of that on the entire host needs to be rebuilt.

As shown in the following figure, there are two disk groups on the 8-disk host, each consisting of one SSD and three HDDs. There are four disk groups on the 12-disk host, each consisting of one SSD and two HDDs.

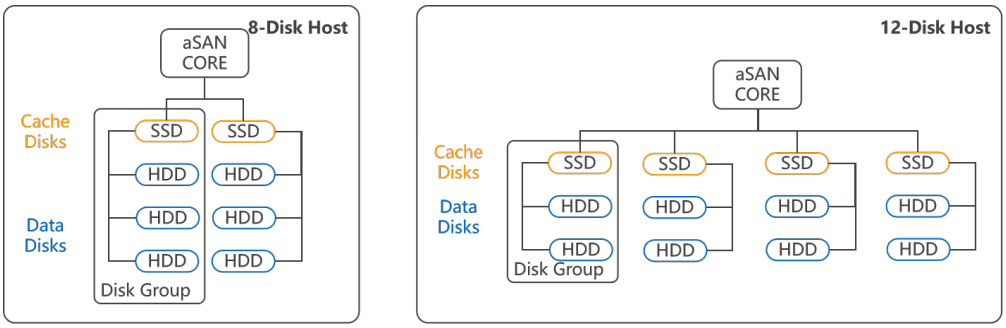


Figure 154: Disk Groups

During the initial installation of virtual storage, the system automatically groups the disks based on the optimal capacity and quantity ratios to ensure that loads of SSDs are balanced and optimal performance can be achieved under the current physical configurations. Generally, customers are not advised to customize the disk grouping.

According to the best performance practice of aSAN, the maximum number of HDDs in a disk group should not exceed seven, which means that the ratio of SSDs to HDDs on a host should not exceed 1:7. A larger number of HDDs in a disk group will result in a lower cache ratio and the optimal performance cannot be achieved when the full striping technology is used.

5.2.2 Functional Architecture

aSAN provides a software-defined storage solution. As a key component of HCI, it is designed for cloud computing and integrates many technologies, including adaptive striping, read/write cache, hot and cold data separation, data redundancy, data balancing, data rebuilding, snapshot, and rolling active upgrade, which can meet the storage requirements of critical businesses in various industries and ensure efficient, stable, and reliable business operations for customers.

Through storage virtualization technology, aSAN performs the pooling operation on local disks on universal x86 servers in the virtual datastore cluster to realize unified integration, management and scheduling of these storage resources, ultimately providing proprietary or iSCSI storage APIs to the upper layers and allowing VMs to use the storage space as needed.

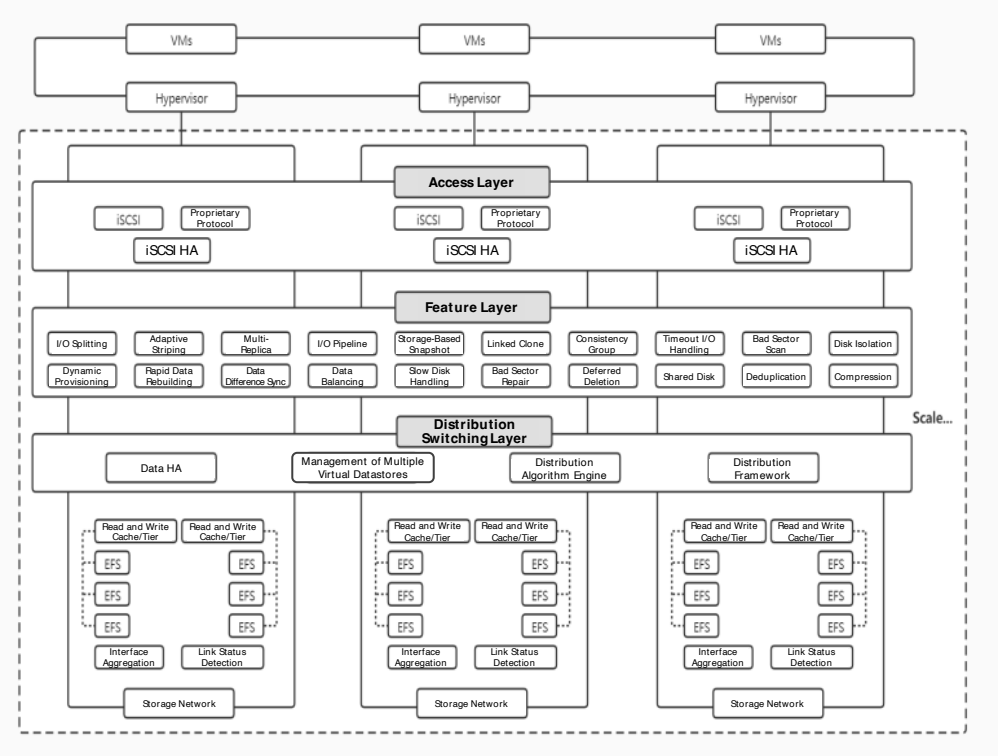


Figure 155: Logical Architecture of aSAN

5.2.2.1 Logical Architecture of aSAN

- Access Layer

The access layer is used to connect to the upper layer applications, supports various protocols, provides advanced feature APIs, and ensures high availability and load balancing of the APIs. It enables external connections through the standard iSCSI protocol and allows internal access to high-performance block services (through proprietary protocols). Storage management and control commands are used to interact with the data plane through RESTful APIs.

- Feature Layer

The feature layer provides data redundancy, performance optimization, and advanced storage features. These features are integrated as modules so that features can be added in the future by simply upgrading the software. The feature layer currently supports advanced features, including the snapshot, adaptive striping, linked clone, and rapid data rebuilding.

- Distribution Switching Layer

The distribution switching layer is the most critical layer, providing globally unified management of data blocks. The layer consists of the data distribution framework, distribution algorithm engine, and other modules. The mechanism and policies are deployed separately. Subsequent updates of the distribution policies can be achieved simply by upgrading the software, flexibly adapting to the business changes. This layer supports EFS that is oriented to the backend storage and the capabilities of flow control, diagnosis, and upgrade.

Using the route distribution algorithm developed by Sangfor, aSAN has the following advantages:

1. Optimal performance: The algorithm can achieve optimal storage performance, and its performance can improve linearly along with the growing cluster size.
2. Data rebuilding: In the event of disk or host failures, aSAN recalculates data distribution, distributes new data across healthy disks, and flexibly rebuilds the data of the faulty disks on multiple healthy disks to quickly recover the data redundancy.
3. Flexible capacity expansion: aSAN allows capacity expansion by adding one or multiple storage nodes or disks and ensures balanced data distribution after expansion.
4. High space usage: Without the restriction of the replication group, aSAN allows heterogeneous hosts and disks, which maximizes storage space usage.

- Backend Storage Layer

The backend storage layer uses the local file system EFS, which was developed by Sangfor, to provide the data disk management function. In addition, functions such as data block indexing and power loss protection are optimized. EFS has no directory structure and has the following advantages over the open-source EXT4 file system:

1. Power loss protection: No data will be lost due to a power loss. After data is written, a success result will not be returned until the bitmap is updated.

2. Faster data write speed: EFS supports the space management of files with 1 MB/128 KB blocks. In the scenario of sequential I/O operations on large blocks, 1 MB of contiguous space can be allocated at a time, thus enhancing the write speed and avoiding disk fragmentation issues after long-time use.
3. Data restore: The block allocation policy and deferred deletion algorithm designed for the mistaken deletion operation can improve the success rate of data restore to a certain extent.
4. Bad sector detection and repair: EFS supports bad sector detection and adopts the technology of repairing bad sectors on multiple replicas, avoiding data losses caused by bad sectors.
5. Read-only issue fixing: Universal file systems tend to turn files into read-only when an index node encounters Express I/O operations, while EFS does not affect the read/write I/O operations on other files but only records the read/write failure of the index node in memory. Afterward, the failed I/O operations trigger automatic restoration, and a request will be sent to rebuild the data if automatic restoration fails.
6. A/B backup mechanism: Critical metadata is stored separately in two different regions and updated alternately, avoiding A/B backups from being damaged simultaneously due to bad sectors. Alternating updating is the key function to guarantee data restore upon failures.

5.2.2.2 Basic aSAN Process

The following figure shows the process in which VM data flows from the virtual storage system client to the virtual storage system server.

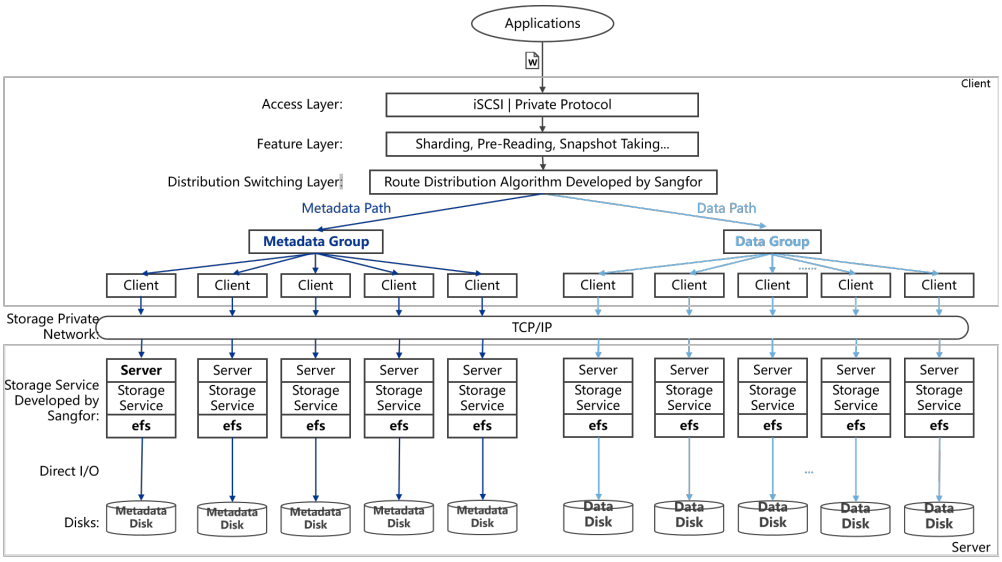


Figure 156: aSAN I/O Path

1. Feature layer (shard): Files on virtual disks are divided into 4 GB shards to achieve a more balanced distribution of data. Based on the sharding and striping algorithms, data is split into small data blocks, and their storage locations are calculated. Then, data is read from or written into the lower layer accordingly.
2. Distribution switching layer (route): During the sharding process, the distribution locations of all data replicas are searched, correlated, and bound in the shard information. This distribution information is referred to as a route.
3. Metadata: It refers to a source data cursor table. It is recommended that metadata be stored in SSDs, either in partitions of cache disks or in partitions of data disks in all-flash storage. Five copies of the metadata should be distributed on different hosts to avoid a single point of failure. The metadata center uses a little data space. A maximum of 10 GB space needs to be reserved for it in the metadata disks. Metadata can be categorized into the following types:

- **Attribute**: Metadata is stored in the replica files as extended attributes. It records the attributes of each replica and ensures the data consistency among replicas.

- **Directory**: Metadata is stored in the metadata center and is primarily used for shard addressing. That is, the metadata records the data disks on which the data replicas and witness replicas of each shard are saved.

4. Storage service developed by Sangfor (route verification): The system checks whether the route information about the shards is consistent. If not, the system returns an error and reselects the route. If so, the process continues.
5. Storage service developed by Sangfor (tier): When receiving a read/write request, the system checks whether the current SSD has cached the data. If all the data is cached, it reads the data from the SSD or writes the data to the SSD and returns the result. If not all data is cached in the SSD, a read request will be sent to EFS for the uncached data.
6. EFS: When receiving a read request, EFS checks whether the data has been written. If not, EFS directly returns an all-zero result. If so, EFS reads the data from the HDD and returns the result.

5.2.2.3 Read I/O Process of VMs

The following figure shows the read I/O path of a VM.

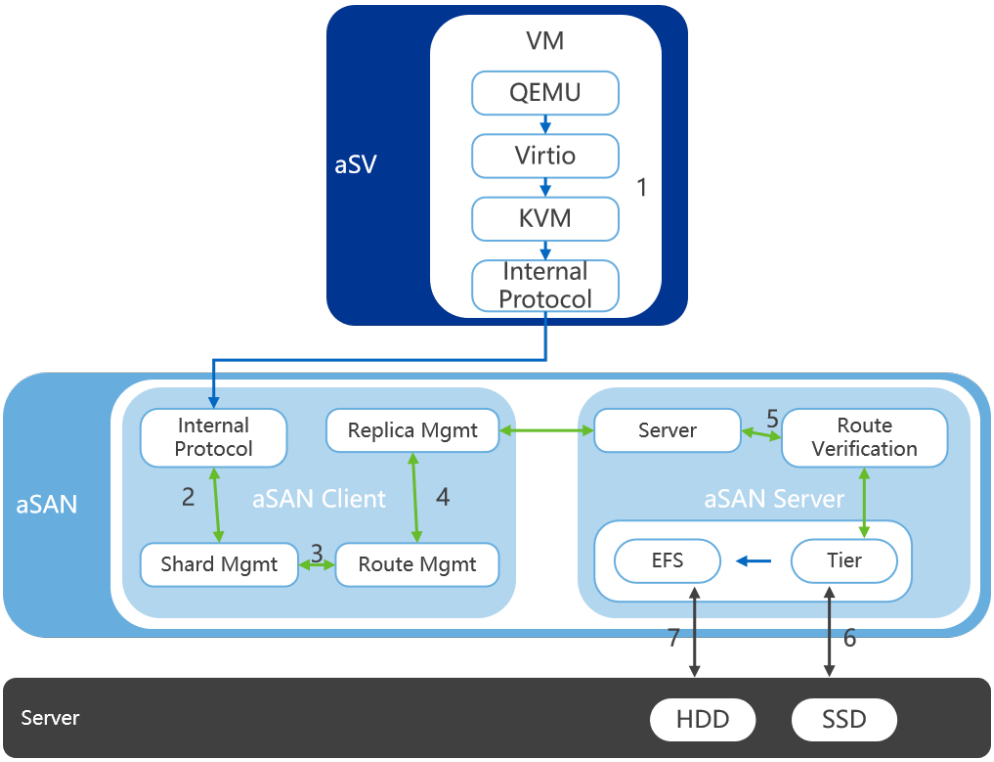


Figure 157: Read I/O Path of a VM

- **QEMU**: Each VM has a corresponding QEMU service instance that implements basic virtualization functions.
- The kernel protocol stack of the VM system sends the I/O requests from the business system to the virtio-blk (virtual block) driver. The virtio-blk driver performs validity checks on the I/O requests and converts them into requests for the underlying image file. These requests are then forwarded to the virtual storage system through the libnfs API.
- **aSAN client**: Client of the virtual storage system.
- **Shard**: Files on virtual disks are divided into 4 GB shards to achieve a more balanced distribution of data. Based on the sharding and striping algorithms, data is split into small data blocks, and their storage locations are calculated. Then, data is read from the lower layer accordingly.
- **Route**: During the sharding process, the distribution locations of all data replicas are searched, correlated, and bound in the shard information. This distribution information is referred to as a route.

Replica: To ensure data redundancy, the system writes multiple data replicas to disks on different hosts. A success result is returned only after all data replicas are written, ensuring consistency among the replicas.

Data server: Data server of the virtual storage system.

The system checks whether the route information about the shards is consistent. If not, the system returns an error and reselects the route. If so, the process continues.

- **Tier**: When receiving a read request, the system checks whether the current SSD has cached the data. If all the data is cached, it reads the data from the SSD and returns the result. If not all data is cached in the SSD, a read request will be sent to EFS for the uncached data.
- **EFS**: When receiving a read request, the system checks whether the data has been written. If not, the system directly returns an all-zero result. If so, the system reads the data from the HDD and returns the result.

5.2.3 High Performance

aSAN uses a high-performance caching system architecture developed by Sangfor, based on which advanced features are developed, including the tier, sharding, adaptive striping, optimal distribution, localization, pipelining, and disk QoS configuration, providing customers with a high-performance storage platform.

5.2.3.1 Data Sharding

As mentioned above, one virtual disk maps one QCOW2 file in aSAN. Virtual disks can be in a small size measured in GB or a large size measured in TB. To provide more flexible storage and processing of data in virtual disks, aSAN divides a single QCOW2 file into 4 GB data shards through the sharding function.

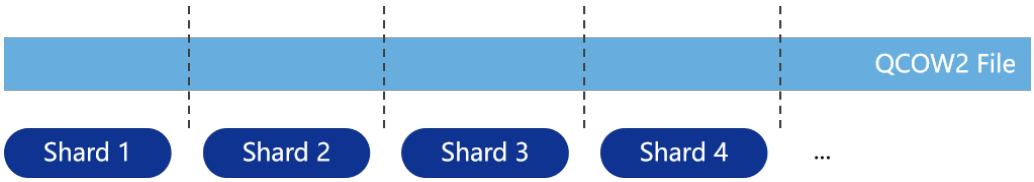


Figure 158: QCOW2 File Sharding

aSAN storage policies are created based on 4 GB data shards instead of files, enabling more balanced data distribution in the virtual datastores and more flexible data management. Taking the data balancing task as an example, when a physical host restarts due to a power loss during the balancing, the task will resume from the last shard processed before the restart rather than balancing the entire QCOW2 file from the beginning. In addition, mechanisms, such as data striping, data rebuilding, and differential data repair, are implemented at the minimum unit of shard.

After sharding, the QCOW2 file size will not be limited to the storage space of one or several physical hosts. The size of a single QCOW2 file (a single virtual disk) can theoretically reach the size of the available capacity of the entire virtual datastore.

5.2.3.2 Data Striping

Striping is a technology also known as RAID 0. This technology divides a continuous piece of data into many small data blocks and then concurrently stores them on different physical disks. This way, the optimal read/write I/O operations can be initiated concurrently, thus achieving excellent performance.

The following figure shows that the striped data can be written to three disks concurrently, while the unstriped data can only be written to one disk at a time. That is, the performance of writing striped data is three times higher than that of writing unstriped data.

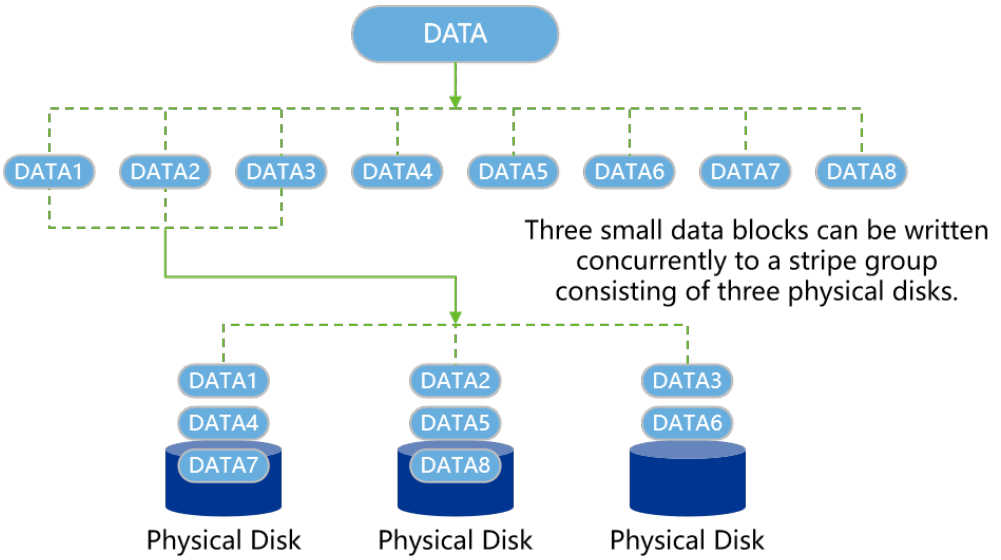


Figure 159: Stripe Group Consisting of Three Physical Disks

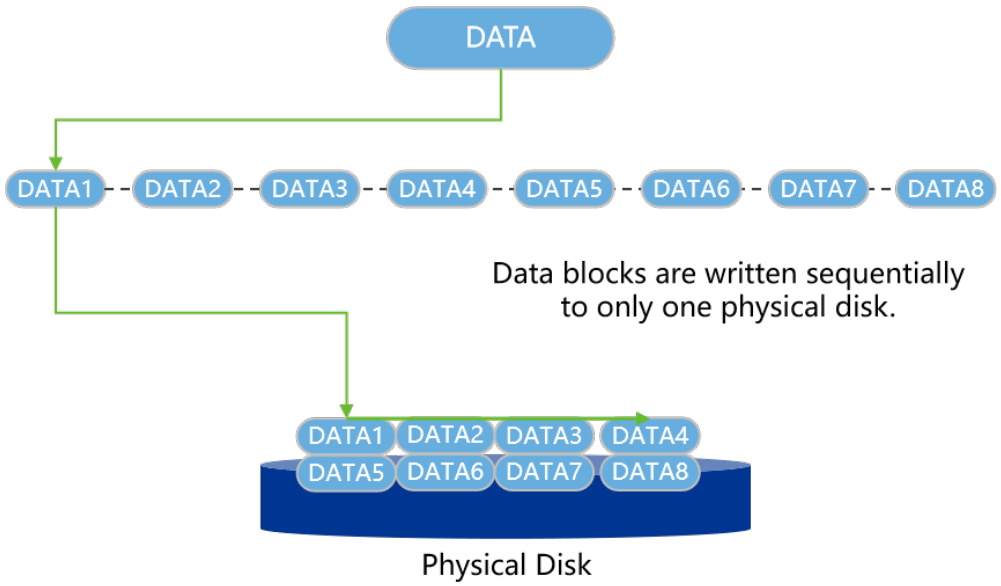


Figure 160: Writing of Unstriped Data

5.2.3.2.1 Adaptive Stripe Width

Stripe width is the number of physical disks that are available for concurrent read/write I/O operations. Typically, an increased stripe width means an increased number of physical disks for concurrent I/O operations, which will bring higher I/O performance.

The stripe width of aSAN is adaptive to the virtual datastore architecture by

default. For optimal I/O performance, the default stripe width cannot exceed the number of data disks on any physical host in the virtual datastore. In this case, the default stripe width of aSAN is 6. When any physical host in the virtual datastore has less than six data disks, the default stripe width of aSAN will be set to the smallest number of data disks among the hosts.

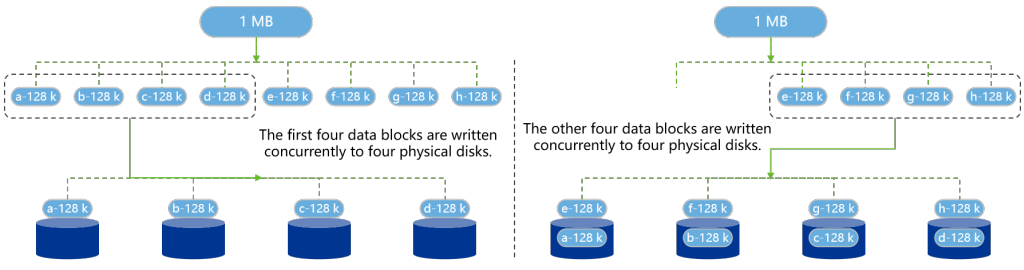
The stripe width of aSAN can be customized to a maximum of 12, provided that it does not exceed the number of data disks on any physical host in the virtual datastore. Changing the default stripe width is recommended only if the current performance issue can be resolved by increasing or decreasing the stripe width.

For example, if a physical host in the cluster has only three data disks, even if all the other physical hosts have six data disks, the stripe width of aSAN is 3 by default and cannot be increased. If each of the physical hosts in the cluster has at least eight data disks, the stripe width of aSAN is 6 by default and can be increased to 8 at most.

5.2.3.2.2 Stripe Size

Stripe size, also referred to as stripe depth, is the size of an incremental data block written during striping. The default stripe size of aSAN is 128 KB.

As shown in the following figure, when consecutive 1 MB data is written, the stripe width is 4, and the stripe size is 128 KB. The 1 MB data will be divided into eight 128 KB data blocks. Because the stripe width is 4, indicating that four physical disks can be written concurrently, the first four 128 KB data blocks are written concurrently to the four physical disks. After the writing is completed, the next four 128 KB data blocks are written.



This figure shows the writing method when the stripe width is 4 and the stripe size is 128 KB.

Figure 161: Striping (Stripe Width: 4; Stripe Size: 128 KB)

5.2.3.3 Tier

Tier is the most important performance layer in the caching system of aSAN. VMs can initiate write operations to write data directly and quickly into the tier and immediately return the writing success result to the upper layer, which can significantly reduce the write I/O latency and increase the write speed. Data in the tier is then written back to the capacity-layer HDDs, thereby releasing the space of the tier for subsequent write operations. The tier functions based on the hybrid elimination algorithm developed by Sangfor. The algorithm combines the advantages of the Least Recently Used (LRU) and Least Frequently Used (LFU) algorithms. Therefore, it has higher query efficiency and a more compact data structure than the traditional Adaptive Replacement Cache (ARC) algorithm and can achieve linear elimination efficiency.

The read operations of VMs can read the required data directly from the tier (which is known as cache hit) instead of the capacity-layer HDDs, thus significantly reducing the read I/O latency.

Sangfor keeps on optimizing the caching system architecture. In the original architecture, each SSD is divided into the read cache, write cache, tier, and redundancy. After optimization, it is divided into tier and redundancy, where the read and write cache ratios of the tier can be dynamically adjusted based on the business requirements. The optimized caching system architecture provides more flexibility and higher performance.

In the original caching system architecture, the read cache is designed for reading the stored data across nodes. However, sometimes, the 10% space reserved for the read cache may be insufficient. In addition, sometimes data may exist in both the read cache and the tier. The optimized caching system architecture abolishes the read cache and write cache. The SSD space occupied by the two caches in the original architecture is allocated to the tier. This way, all space of the tier can be used for read cache, and the data will only be retained in the tier. To address the issue of low performance in writing data of less than 4 KB to SSDs, aSAN aligns the data block sizes of the file system and

SSDs. When data blocks of less than 4 KB are written, the data is expanded to 4 KB in memory before being written to SSDs. The I/O success result is returned only when the data writing to SSDs is completed.

In addition, the tier in Sangfor HCI 6.9.0 has been further reconstructed to reduce the scalable block size (that is, block management granularity of the tier) to 8 KB to 64 KB, which can be adjusted according to the cache size to achieve optimal performance and space usage. Sangfor has developed the simple metadata and optimized the cache index structure and the allocation algorithm for caching metadata. Therefore, the space occupied by cached metadata is reduced, and the speed of reading cached data has been improved. HCI 6.9.0 can store full cached metadata in memory, which improves performance. Also, the memory usage and CPU usage of cached metadata have been minimized to an industry-leading level. With the automated flow control function, the cache system can detect the tier capacity and business traffic in real time, avoiding the dramatic decrease in the business processing capability when the cache space is used up.

5.2.3.3.1 Caching System Architecture

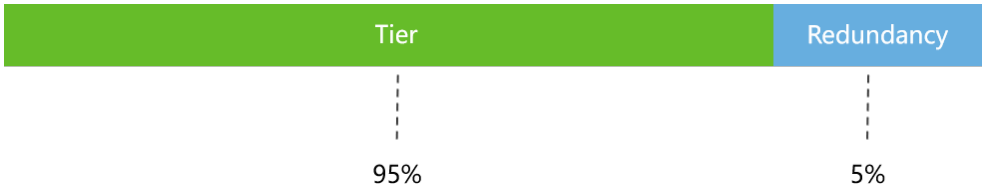


Figure 162: aSAN System Architecture

When aSAN uses hybrid configurations, SSDs are used as the cache layer for speeding up read and write I/O operations but do not actually store data, while HDDs are used to store the data. Each SSD is divided into the following areas in aSAN:

- **Tier**: It is the most important performance layer in the caching system, which can be flexibly configured to provide the read cache and write cache. The cache sizes can be dynamically adjusted to meet the business requirements.

- **Redundancy**: This area occupies about 5% of the space. Based on the redundancy feature of SSDs, this area is used as the garbage collection and swapping space to improve the lifetime of the SSDs and to retain the I/O performance of the SSDs after long-term use.

NOTICE

1. SSDs of the enterprise or data center–class are required. Enterprise-class SSDs provide the power loss protection function and can be supplied by a large capacitor on the SSD circuit board to ensure that the cached data in the SSDs can still be written to the flash memory after a power loss, thus guaranteeing high data reliability. Consumer-grade SSDs do not provide such a feature.

2. In typical hybrid configurations, SSDs are used as cache disks and HDDs as data disks. Similarly, the cache layer architecture in all-flash storage uses PCIe SSDs as cache disks and SATA SSDs as data disks.

5.2.3.3.2 Key Technologies

Tier architecture:

For all business I/O scenarios where hybrid datastores of SSD+HDD or NVMe SSD+SATA SSD, the storage tier has been reconstructed: SSDs are used as the cache layer to support the read and write business traffic and to store hot data. HDDs are used as the capacity layer to store cold data, and the cache layer replaces the cold data in the capacity layer through smart algorithms to achieve automated storage that separates the cold data from the hot data.

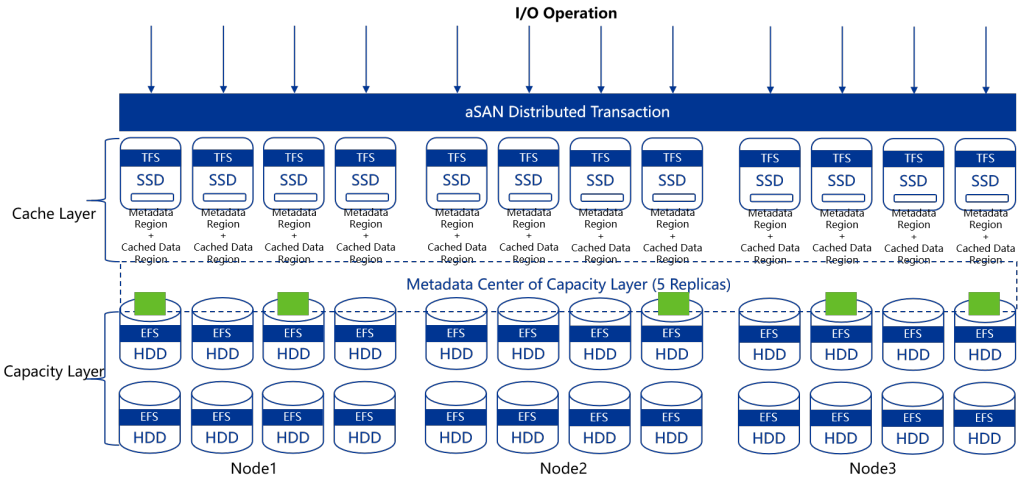


Figure 163: Tier Architecture

To manage the physical space of SSDs and HDDs, the TFS and EFS file systems developed by Sangfor have been deployed. The file systems contain many metadata, which will be read into memory and occupy memory space. The new tier architecture simplifies the cache-layer metadata, which reduces memory usage.

Simple metadata design of the cache layer:

The file system in an OS is used to organize and manage files and directories on a computer. Metadata in a file system describes the attributes, locations, sizes, creation time, access time, modification time, and other information about files and directories. Upon data reading or writing, metadata needs to be cached in memory for processing. Metadata contains information about the file system structure and file attributes, which are required for the proper running of the file system. When reading a file, the OS must check its metadata to determine the location, size, property, and attributes of the file. When writing a file, the OS needs to update its metadata, such as the file size and modification time.

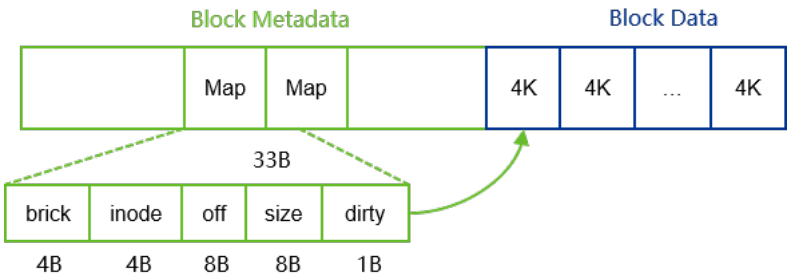


Figure 164: Memory Usage of Cache-Layer Metadata Before Tier Reconstruction

The metadata of the file system consumes some memory resources. Therefore, too much metadata will lead to high memory usage, thus reducing the performance of the system. To address this issue, aRaid 3.0 abandons the original metadata structure and innovatively uses the hash bucket index extent for data block mapping, reducing the space usage of the metadata from 33 bytes to 8 bytes. In actual scenarios, cache metadata contains more than mapping. Based on the test results, the memory usage is reduced by more than 50% if 1.92 TB SSDs are used as cache disks.

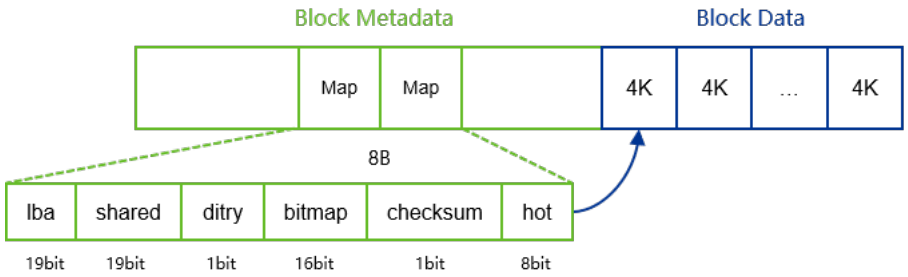


Figure 165: Memory Usage of Cache-Layer Metadata After Tier Reconstruction

Block-based management of the tier:

A data block is the smallest unit for storing data in a file system. The file system divides files into data blocks of equal size to store, and each data block is identified by a unique number (namely, block address). When reading a file, the file system searches for the corresponding data blocks according to the file's metadata (such as the file size and block address) and reads them into memory. When writing a file, the file system also searches for the corresponding data blocks according to the metadata and writes the modified data into them.

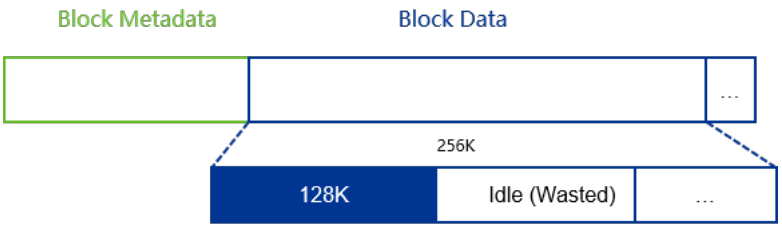


Figure 166: Block-Based Management at the Cache Layer Before Tier Reconstruction

The tier functions as a file system. Reducing the data block size from 256 KB to 8 KB can effectively reduce internal file fragments and ensure more precise space allocation. In the scenario of small block random write, the space usage of the tier can be improved by 40%-50%, and the problem of cache breakdown can also be eliminated. However, reducing the block size can increase the metadata quantity, which will affect the disk space and memory usage. Thanks to the simple metadata design, Sangfor Smart Tier can manage memory usage and avoid a decrease in indexing efficiency.

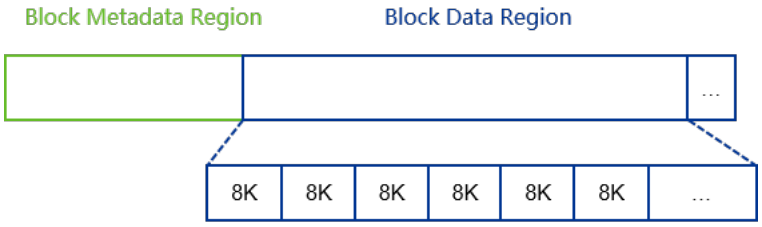


Figure 167: Block-Based Management at the Cache Layer After Tier Reconstruction

Effective block allocation algorithm:

The multi-level index technology of file systems can be implemented for large files, which cannot be achieved by a single-level index. In a file system using a single-level index, all data block addresses of a file are stored in a single index block. If the file is overlarge, a single index block is insufficient for storing all the addresses. In this case, multi-level indices are required.

Sangfor aRaid 3.0 uses bitmap-based multi-level indices to efficiently allocate and index data blocks in the tier. Multi-level indexes function by using multiple index blocks to store the data block address of a file. The first-level index block of a file points to a set of second-level index blocks, each of which, in turn, points to a set of data blocks. Multi-level indices must be used together with

data bitmaps. The first-level index records data with a bitmap. Upon a data read and write, the bitmap index is cached and operated conveniently, which can effectively improve the data retrieval efficiency, reduce memory usage, and improve the performance of the file system.

- During data writing, the operation varies. For a small block random block write, L0 data blocks are allocated directly. For a large block sequential write, the space of L1 and L2 is divided according to the block size and consecutive data blocks are allocated to the same file by region.
- During a data read, third-level index bitmaps are used to record and retrieve data block locations, minimizing caching and retrieval of invalid data blocks and greatly improving the data reading speed.

The tier reconstruction brings the following advantages to HCI 6.9.0:

1. The management granularity of the tier has been minimized to an industry level of 8-64 KB (256 KB-4 MB before reconstruction). The smaller the granularity, the higher the usage ratio and the performance. The usage ratio after tier reconstruction has been improved by about 40%-50%.
2. The memory usage of the cache-layer metadata has reached the industry-leading level of 2.92‰ (8 KB)-0.36‰ (64 KB), and the memory usage of the tier has been reduced by about 50%-64%. (For vSAN, the memory usage is 9.76‰. For Huawei FusionStorage, the memory usage is 7.81‰.)
3. The cache-layer metadata has been minimized to only 24 bytes. Many disruptive and innovative algorithms have been developed. For example, the cache replacement algorithm can be implemented with a resource usage of 5 bytes (1-byte space usage of the algorithm and 4-byte space usage of the shadow linked list, which are much smaller than those required by the ARC algorithm and ARC shadow linked list), and a mapping (LBA + PBA) can be implemented with a space usage of 7 bytes (smaller than the size of a pointer).
4. CPU usage is significantly reduced over the earlier versions and even reduced by about 50% in some scenarios. The CPU usage is collected in benchmark testing scenarios.

5. Oracle RAC scheduled snapshots achieve perfect prevention of cache breakdown and a performance increase of 12%. aSecurity scheduled snapshots can support a business performance increase by 105% in the following scenario: There are 600 users, the last 24 snapshots need to be retained, and a snapshot model needs to be created at an interval of an hour.
6. As a result of the simple tier metadata, the tier system can store the full metadata of a host in memory, which improves performance (reducing one metadata read).
7. The adaptive cache replacement policy is applicable to anti-scanning, episodic, periodic, and other business bulk operations.

5.2.3.3.3 Technical Principles

Write cache:

1. When a VM initiates a write operation, the data is sharded and striped, and then the disk where each shard is to be stored is determined according to the replica storage policy and the optimal data distribution policy. The following figure uses a disk group as an example. The data is written to the tier of SSDs before being written to the capacity-layer HDDs. After the writing is complete, a success result will be returned to the upper layer immediately, and the data will be marked in Dirty state. This state indicates that the data has not been stored in the capacity-layer HDDs.

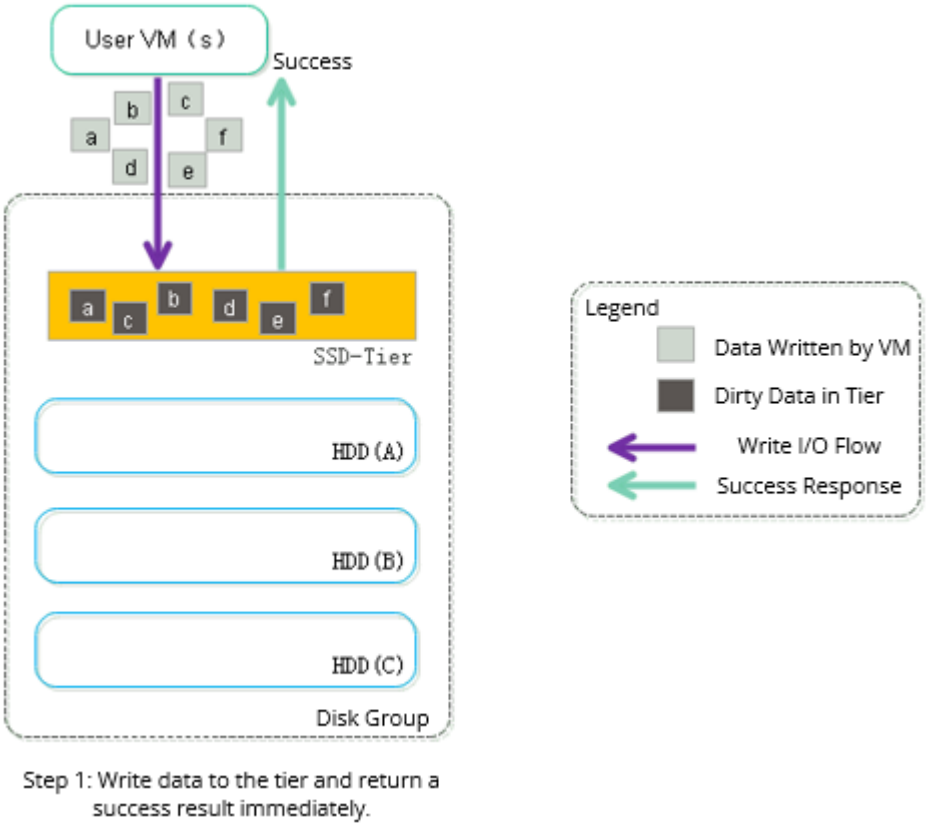


Figure 168: Writing Data to the Tier

2. The data in Dirty state in the tier is consistently written back to the capacity-layer HDDs. When the write-back is completed, the data is marked in the Clean state but not deleted immediately. This state indicates that the data has been stored in the capacity-layer HDDs.

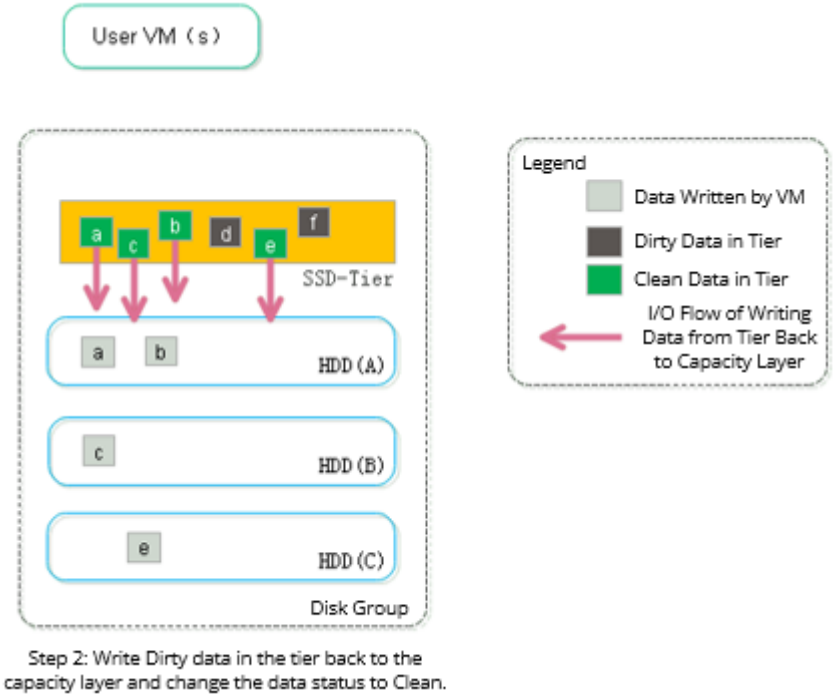


Figure 169: Writing Dirty Data in the Tier to the Capacity Layer

3. If the VM keeps on writing data while the space of the tier is used up, the data in Clean state will be eliminated as it has been stored in the capacity-layer HDDs so that the space can be released for subsequent data to be written. Data elimination takes a very short time.

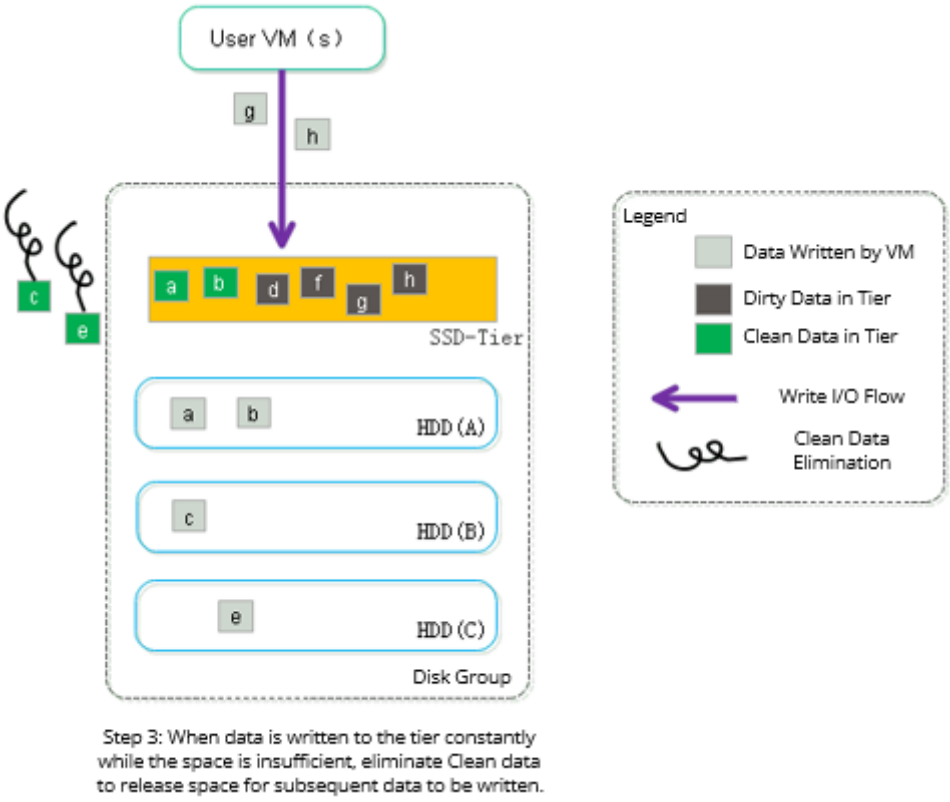


Figure 170: Clean Data Deletion After Data Writing Back to HDDs

Writing back the data in the tier has the following characteristics:

1. Continuous writing: The dirty data in the tier is written back to the capacity-layer HDDs continuously to ensure that the data in the tier is all clean. When the VM writes data next time, the clean data is eliminated immediately. Then, the data is written to the tier space immediately, and a successful writing result is returned to ensure the constant availability of the write cache.
2. Adaptive I/O: The speed of writing dirty data in the tier back to the capacity-layer HDDs is adaptive. The tier applies the low-speed write-back policy in case of high business I/O usage from the upper layer and the full-speed write-back policy in case of low business I/O usage from the upper layer. The adaptive policies allow the data to be written back as soon as possible while ensuring proper business processing.
3. Sequential write: Dirty data in the tier is written back to the capacity-layer HDDs sequentially. Before write-back, the dirty data must be merged,

enabling sequential write-back and improving the write speed. (For HDDs, sequential read/write performance is much higher than random read/write performance.)

NOTICE

Multiple data replicas are written to the tiers of disk groups on different hosts. A success result is returned only after all data replicas are written, ensuring the consistency of the data replicas. Data replicas written back to the capacity-layer HDDs are not required to be consistent. For example, the following scenarios may exist, which are acceptable:

1. One replica of data x has been written back to disk group A and has entered the Clean state in the tier, while the other replica is still Dirty in the tier of disk group B.

2. When data is continuously written to disk A, data x in the clean state is eliminated. In this case, data x does not exist in the SSDs of disk group A but still resides in the SSDs of disk group B.

Read cache:

- **Tier hit**: Assume that a VM initiates a read operation. If the data is located in the tier, the VM reads it (either in Clean or Dirty state) directly. In this case, the tier is hit. As shown in the following figure, data c and data f to be accessed by the VM are both stored in the tier. The result can be returned immediately in this case. The read I/O latency can be significantly reduced when the tier is hit.

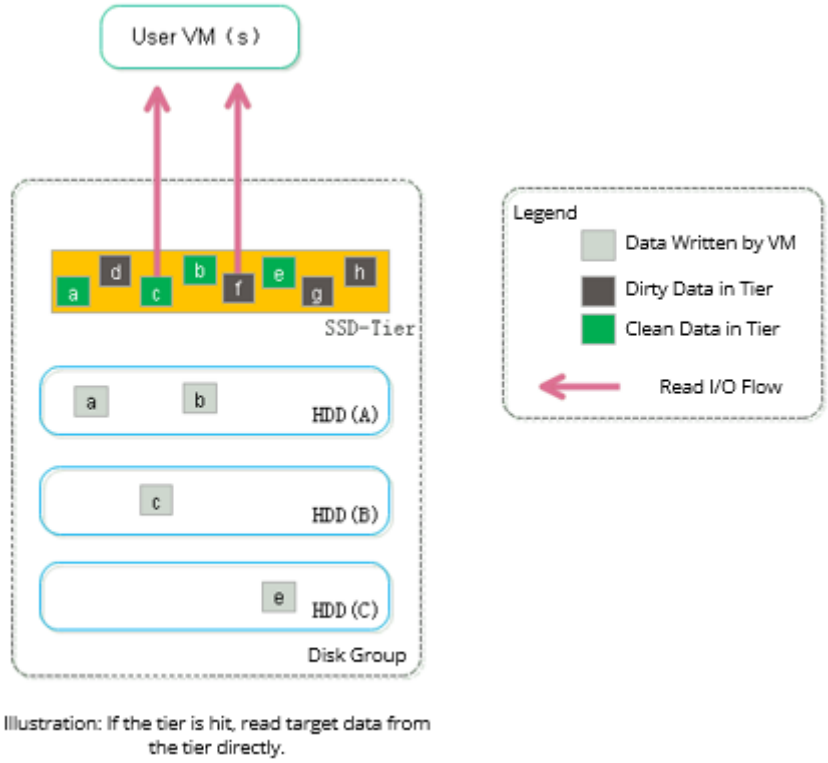


Figure 171: Hot Data Reading from the Tier

- **Tier not hit**: Assume that a VM initiates a read operation. If the data is not located in the tier, the VM reads it from the capacity-layer HDDs. In this case, the tier is not hit. Due to the average performance of HDDs, a latency penalty is incurred. As shown in the following figure, data c and data e to be accessed by the VM are not stored in the tier. In this case, the system reads data from the capacity-layer HDDs. In addition, it determines whether to read data to the tier of the SSDs based on the data access frequency. If so, the tier can be quickly hit when the data is read next time.

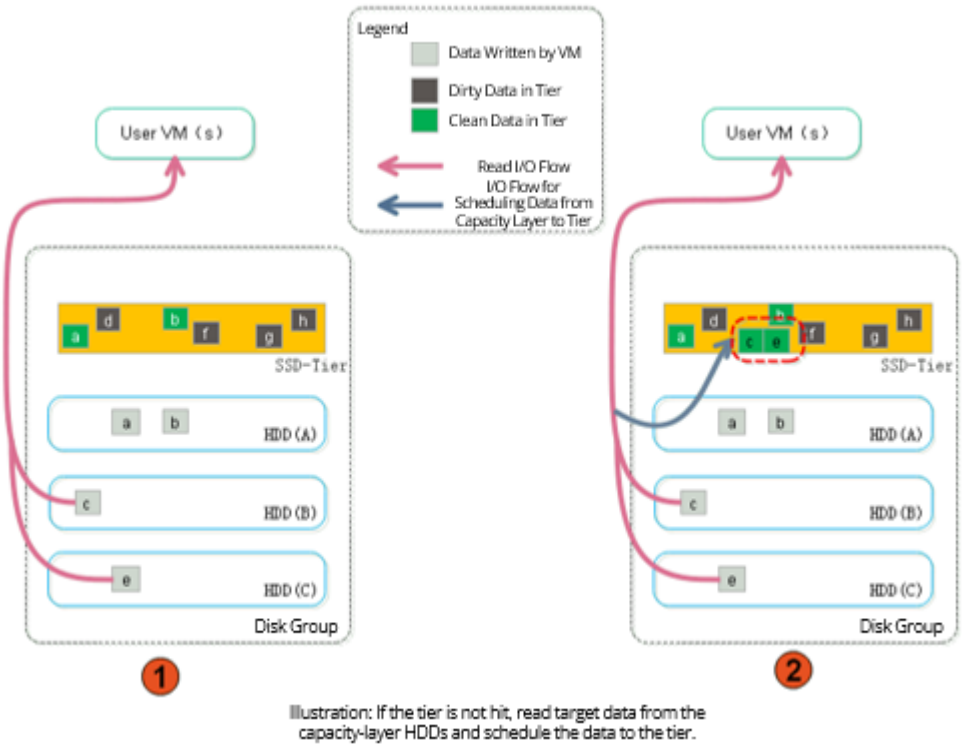


Figure 172: Cold Data Reading from the Capacity Layer

NOTICE

For the write cache, two replicas of data may not exist simultaneously in tiers on two disk groups. In other words, a replica of the data may exist in the tier of one host but not in the tier of the other.

The read cache is similar: Assume that two replicas of data have been written back to the capacity-layer HDDs and eliminated from the tiers. When the data is read, the tiers are not hit. In this case, the system directly reads data from the capacity-layer HDDs, during which either data replica will be read to the tier while the other replica is still in the capacity-layer HDDs.

The subsequent access to the data will be achieved by reading the replica from the tier.

Tier Heat Map:

The tier provides the read cache and write cache to mainly enable VMs to:

1. Directly and quickly write data to the tier, avoiding waiting and reducing write I/O latency.

2. Directly read data from the tier, improving the cache hit rate and reducing read I/O latency.

Only a limited amount of data can reside in the tier due to limited tier space. Therefore, the tier is expected to store the most valuable data to ensure the tier hit upon a data read, while the less valuable data needs to be eliminated from the tier, releasing space for valuable data writing without latency.

The most valuable data is called hot data, which is accessed frequently by the user, while the less valuable data is called cold data, which is not accessed frequently by the user.

The caching system of aSAN identifies hot or cold data based on the frequency and time of data being accessed. As shown in the following figure, the more frequently and more recently the data is accessed, the hotter the data is.

In addition, VM data with **High Priority** selected has a higher priority to enter the tier and is less likely to be eliminated.

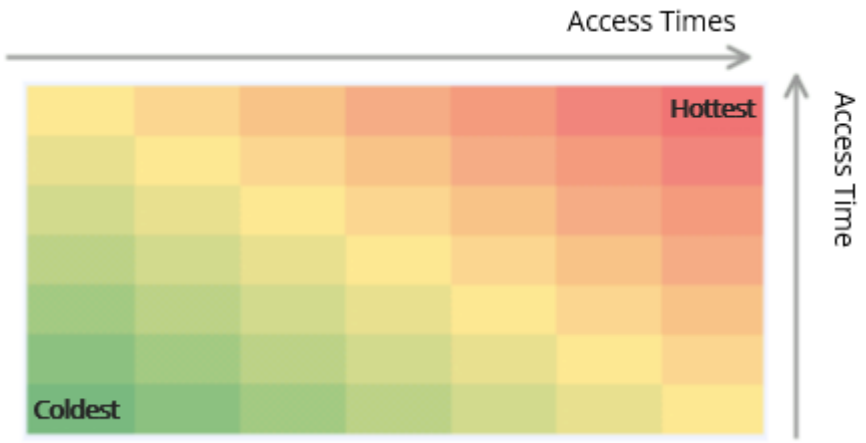


Figure 173: Heat Map of Cold/Hot Data

Data elimination principles of the tier:

1. When data is written to the tier with insufficient space, the clean data that is the least hot is eliminated first, releasing space for fast data writing to the tier and ensuring that the data to be accessed recently will remain in the tier at a high possibility.

2. When the tier is hit upon data reading, the data becomes hotter, further increasing the priority of the data in the tier and reducing the risk of being eliminated.
3. When the tier is not hit upon data reading, the data becomes hotter. If the data and the subsequent 1 MB data are hotter than the data in the tier, they will be read from the capacity-layer HDDs to the tier of the SSDs, eliminating the less hot data in the tier.

Differences Between the Tier Mechanism of aSAN and the Traditional Caching Mechanism:

The traditional caching mechanism also uses high-performance SSDs as the cache layer and low-performance but large-capacity HDDs as the capacity layer. However, SSDs are only partitioned with a fixed portion into the write cache and read cache, which function separately.

Write cache: It occupies 30% of the SSD space. Data is written to the write cache, reducing write I/O latency.

Read cache: It occupies 70% of the SSD space. Data is read from the read cache, reducing read I/O latency.

The traditional caching mechanism has the following disadvantages:

1. High overhead in writing data: When data is written to the write cache, a replica needs to be written to the read cache simultaneously to ensure that the data can be read from SSDs subsequently. That is, the data needs to be written twice, which results in a high overhead and a high lifetime loss of the SSDs.
2. Low SSD performance:

- Low data write performance: When the business data is continuously written, and the write speed is faster than the speed of writing data back to the capacity layer, the data requires a faster write speed than releasing the space from the write cache. However, due to the fixed space size, the write cache will be used up quickly. In this case, the data write speed will decrease to the write-back speed because the space of the write cache can be released only after data is written back to the capacity layer. The

remaining 70% of SSD space is unavailable for the write cache even if it is idle because it is allocated as the read cache.

- Low data read performance: For business data that needs to be read frequently after being written, more read cache space is needed to store hot data. However, only up to 70% of the SSD space is available because 30% of the space is used for the write cache only. That is, the traditional caching architecture cannot adapt to business changes because SSD space cannot be fully used.

3. Vulnerable to data read in scan operations: The traditional caching mechanism uses the LRU algorithm for data elimination. Based on this mechanism, when data is read to the read cache with insufficient space, the data that is less recently accessed will be eliminated. However, the mechanism has a disadvantage in that it is vulnerable to data reads in scan operations. In detail, when a scan operation is performed, valuable data in the cache will be completely crowded out by less valuable data that is read only once. For example, when a user scans for a virus or watches a movie, the data will be read to the read cache and crowd out the valuable data in it.

Advantages of the Tier Mechanism:

A tier provides both read cache and write cache and has the following advantages:

1. Low overhead in writing data: Data can be read quickly after being written to the tier and can still reside in the tier of the SSDs even after being written back to the capacity-layer HDDs.
2. High performance: The tier space allocation can be dynamically adjusted. If a great amount of data needs to be written, all the tier space can be used as the write cache, which can better satisfy the changing requirements of read and write operations.
3. Adaptive to data read in scan operations: The caching system of aSAN identifies hot or cold data based on the frequency and time of data being accessed and defines the elimination priority of the data in the tier. Data read in a scan operations reading was accessed only once despite being

the most recent. Therefore, the data is not the hottest and will not crowd out the hotter data in the tier.

Working Principles of the Write Cache:

Although the caching system of aSAN provides the write cache function through the tier, a small separate area is allocated as the write cache, which is mainly designed to improve the performance of writing tiny blocks of data.

This is because the minimum unit for writing data to the tier is 4 KB. If smaller data needs to be written, 4 KB data needs to be read and then concatenated with the data to be written to form new data, which is finally written to SSDs. This process increases the overhead of data writes. It is performed by the SSD driver or controller and is imperceptible to the upper layer.

The write cache is designed based on oplog. Each write operation is reorganized into an 8 KB log and can be completed in a single operation.

Therefore, data smaller than 4 KB is directly written to the write cache and then written back to the capacity-layer HDDs. Data larger than 4 KB is directly written to the tier. With the combination of tier and write cache, the speed of writing small blocks of data can be improved.

Working Principles of the Read Cache:

Although the caching system of aSAN provides the read cache function through the tier, a small separate area is allocated as the read cache, which is mainly designed to improve the performance of reading data across hosts for VMs.

VMs run preferentially on the host where replicas are aggregated to achieve data localization. This way, the data can be read without crossing the storage network, reducing the network overhead.

However, a VM may run on a different host under special conditions, for example, when a switchover is triggered due to insufficient CPU resources, insufficient memory resources, or a fault on the host. As a result, data localization cannot be achieved, and the VM needs to read data across the storage network.

The following uses the read I/O flow to describe the working principles of the

read cache:

Read cache not hit: As shown in the following figure, when the VM running on host C requests to read data b that is not stored on the local host and fails to be found on the read cache client, the system needs to read the data across the storage network from the tier of another host. If the tier is hit, the data is read directly and returned. Otherwise, the data is read from the capacity-layer HDDs and returned.

When the read data is returned to the VM, a data replica is saved to the SSD read cache on the local host and recorded on the read cache client.

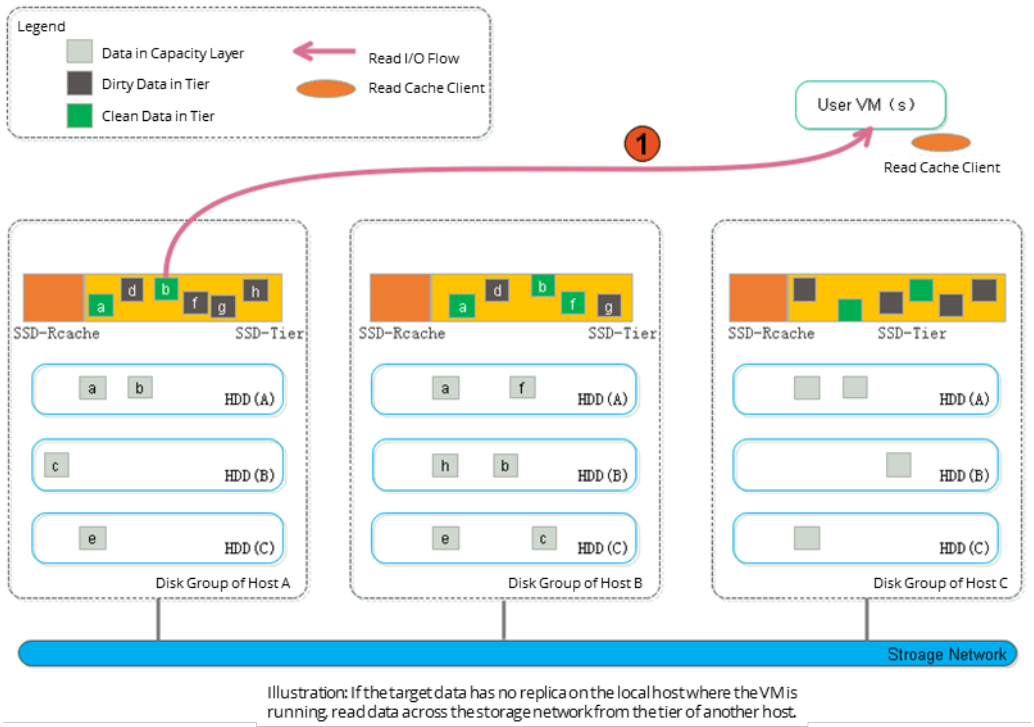


Figure 174: Step 1 of Reading Data by a VM - No Data Replica Exists on the Local Host

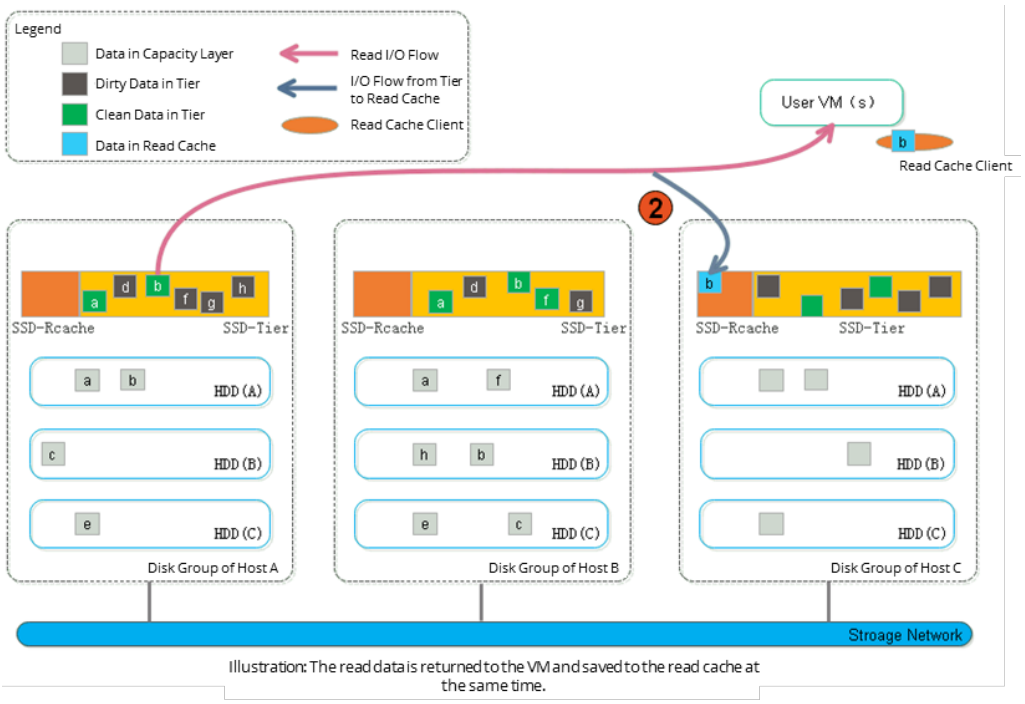


Figure 175: Step 2 of Reading Data by a VM -Data Is Stored to the Read Cache

Read cache hit: As shown in the following figure, when the VM running on host C wants to read data b that is not stored on the local host, the system queries the data on the read cache client. If the data is found, the system reads it directly from the read cache and returns it instead of reading the data across the network.

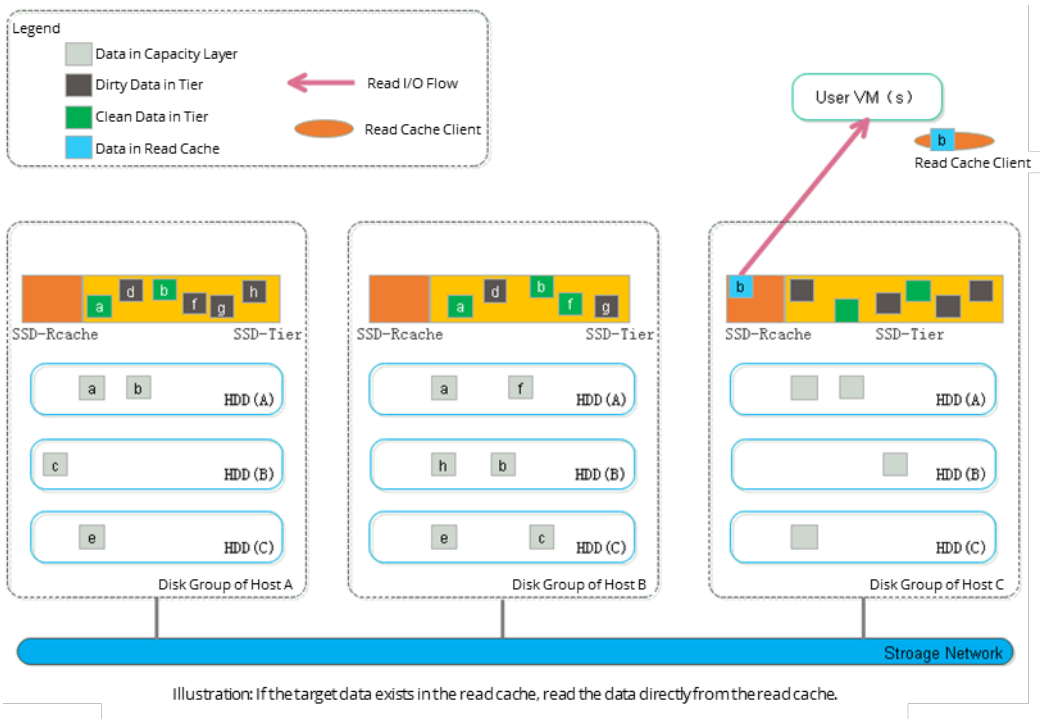


Figure 176: Diagram 3 of Reading Data by a VM - Data Is Read from the Read Cache

With the read cache mechanism, the network overhead will not increase sharply even if the VM cannot read data from the local host. The data elimination mechanism and heat map of the read cache are the same as those of the tier.

Simplified Memory Overhead of the Tier:

To enhance business performance and reduce business I/O operations on disks, the tier stores the metadata information in memory. In addition, to enable hot data to be separated from cold data consistently, the system stores some data eliminated from the tier in the shadow table, which consumes a large amount of memory space.

To minimize memory overhead, the tier:

1. Uses the block metadata merging technology to boost metadata space usage. A block can be a minimum of 8 KB, and a minimum granularity of 4 KB is implemented through bitmaps.
2. Uses the hot data grading technology to optimize the memory usage of hot data and improve the efficiency of data elimination.

3. Optimizes algorithms to greatly reduce the memory overhead of cache metadata.
4. Achieves the effect that less than 1 GB memory is occupied during the running of 480 GB/960 GB/1920 GB cache SSDs.

5.2.3.4 Optimal Data Distribution

Data is distributed after striping, sharding, and multi-replica policies are executed in sequence. If the virtual datastore space is sufficient, host disks are homogeneous, no data rebuilding is triggered by host or disk failures, and no data balancing is triggered by capacity expansion, data distribution is executed based on the following policies to ensure high reliability and performance of data read and write.

5.2.3.4.1 Data Localization

Both aggregated and scattered replicas exist. After striping and sharding, data is divided into data shards. All the shards share the same storage policy and have their own redundant replicas. If each of multiple or all shards has one of their replicas distributed on the same host, an aggregate replica of the data is achieved on the host, and this host is referred to as an aggregate host.

Simply speaking, an aggregated replica indicates the shard replicas that reside on the same host, such as host A shown in the following figure. A scattered replica indicates the shard replicas that are distributed on three hosts at most, such as hosts B, C, and D shown in the following figure.

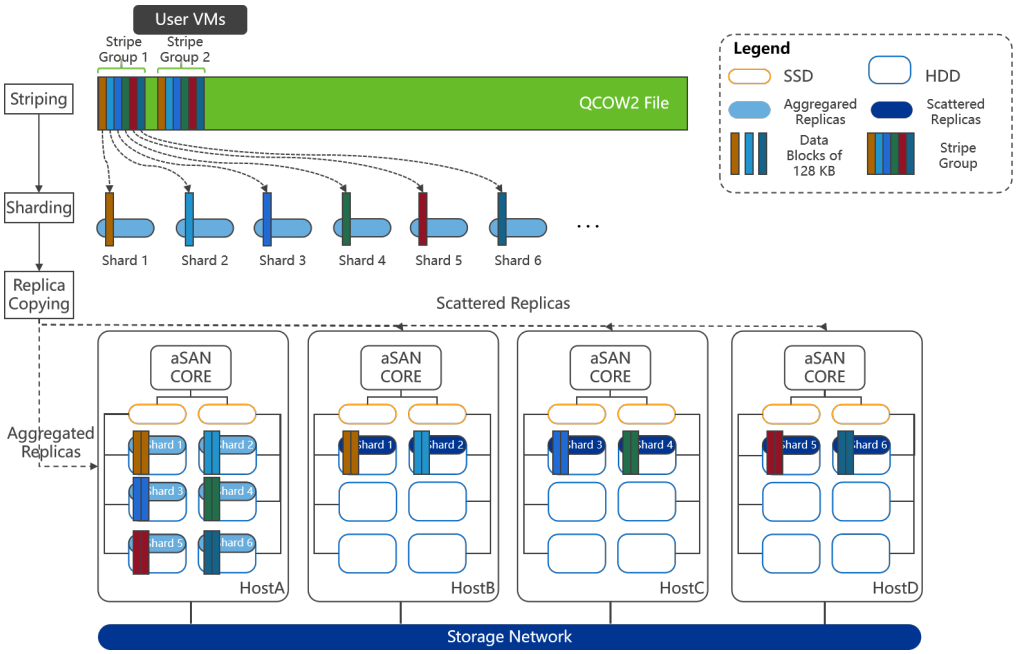


Figure 177: Distribution of Multiple Data Replicas

VMs run preferentially on the host where the aggregated replica resides to achieve data localization. When a VM requests to access data, the system reads the data directly from the aggregated replica rather than across the storage network, improving the read throughput and breaking through network bandwidth bottlenecks, especially in all-flash mode.

HCI has also optimized the distribution policy of scattered replicas to reduce fault domains. In a two-replica stretched cluster, the scattered replica is distributed on three hosts at most other than the host where the aggregated replica resides. In a three-replica stretched cluster, scattered replicas are distributed on five hosts at most other than the host where the aggregated replica resides.

NOTICE

If a virtual datastore consists of only two hosts, the two replicas are both aggregated replicas.

5.2.3.4.2 Utilization of More SSDs

Data is preferentially distributed across disk groups within a host.

Shards within the shard group of either an aggregated or a scattered replica are preferentially distributed across disk groups on the host. That is because each disk group has an SSD available to function as a cache for HDDs in the group. If data is distributed across more disk groups, the caching capability of more SSDs can be used upon concurrent read and write operations on data within a stripe group, improving the disk performance.

5.2.3.4.3 Utilization of More HDDs

Data is preferentially distributed across disks within a disk group.

Shards within the same shard group are distributed across as many disks as possible when they are allocated to the same disk group. As shown in figure 5.2-25, the shards in the same shard group are distributed on different disks on host A. This way, low-performance random I/O on HDDs can be avoided when the data blocks within a stripe group are read and written concurrently.

As introduced in section 5.2.3.2 "Data Striping," aSAN does not allow the stripe width to be greater than the number of data disks on any host in the virtual datastore. This restriction is also set to ensure that sufficient data disks are available for distributing the data blocks within each stripe group of an aggregated replica so that data can be read and written concurrently without compromising performance.

5.2.3.4.4 Balanced Distribution

Data is preferentially distributed on the disk with the largest remaining capacity.

To ensure the balanced data distribution within a virtual datastore and to prevent data on a host or a disk from being more frequently accessed than that on other hosts or disks, the system needs to consider the remaining capacity of the hosts or disks within the virtual datastore when distributing data. Examples are as follows:

1. If the policies of data localization and utilizing more SSDs and HDDs are perfectly executed, shards are preferentially distributed on disks with more remaining capacity and the aggregated replica is preferentially distributed on the host with more remaining capacity.
2. If the space usage of a host has exceeded 85%, the shards to be added to the aggregated replica of the existing QCOW2 file on the host will not be aggregated on the host. In addition, the host will not be selected to distribute the aggregated replica of a new QCOW2 file in the virtual datastore.
3. If the space usage of a disk has exceeded 85%, the disk will be excluded when shards are distributed.

With the above distribution policies, aSAN can effectively improve the concurrent I/O performance while ensuring balanced data distribution.

- A segment of data can be concurrently written to SSDs of multiple disk groups to leverage the performance of multiple SSDs.
- If a segment of data is cached in SSDs, it can be concurrently read from SSDs of multiple disk groups to leverage the performance of multiple SSDs. If the data to be read is not cached on SSDs, it can be concurrently read from multiple HDDs to leverage the performance of multiple HDDs.

5.2.3.5 I/O Path Optimization

Taking the I/O path of HCI as an example, a complete I/O path starts from a VM of aSV. The VM sends an I/O request and communicates with QEMU and libnfs through a proprietary protocol. The request is transferred to the aSAN client through the Unix domain socket and then distributed to the local aSAN server through the shared memory or to a remote aSAN server through TCP, as shown in the following figure. When reaching the server, the I/O request is directly sent to SSDs. Upon successful data read or write, a success message is returned to the upper layer. In addition to the performance, which is the only aspect optimized by other vendors for their storage products, Sangfor optimizes the entire I/O path for HCI aSAN. The I/O performance has been improved by more than 30% over earlier versions, and the performance of

high-depth I/O (such as the I/O depth of 128) has also surpassed that of other vendors. The optimization is based on the following critical technologies.

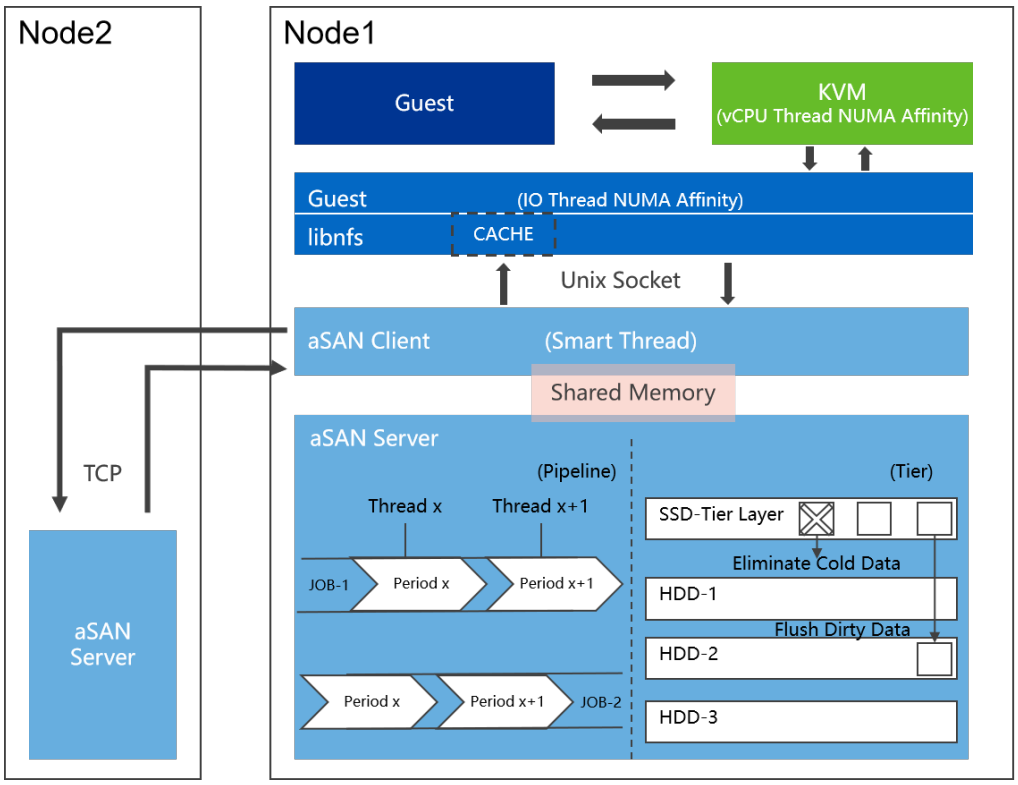


Figure 178: I/O Architecture of aSAN

5.2.3.5.1 Scheduling of VM vCPU and I/O Threads

The I/O path, VM vCPU thread > QEMU main thread > QEMU libnfs thread > aSAN client thread, contains many memory interactions. If access latency occurs, the I/O performance will deteriorate. To improve the memory access performance, aSAN schedules these threads to the same NUMA node and enables huge-page memory.

5.2.3.5.2 Soft Affinity Scheduling

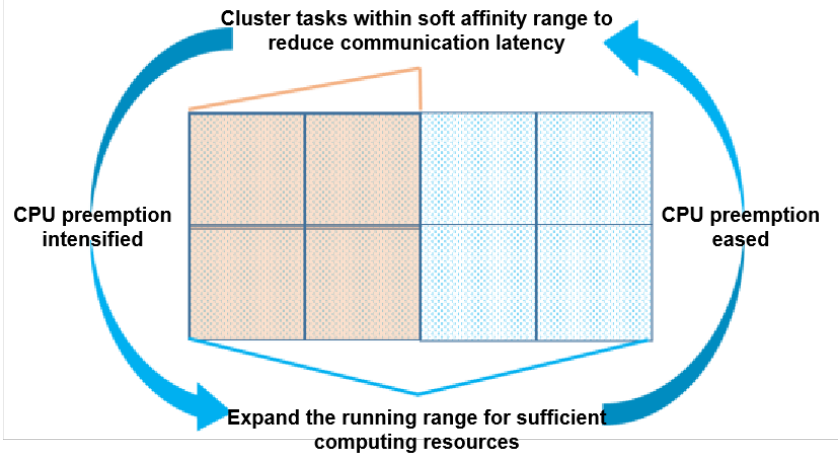


Figure 179: Soft Affinity Scheduling

The soft affinity scheduling mechanism can sense system load changes and dynamically allocate CPU resources to services. When the CPU load is low, services preferentially use CPUs within the soft affinity range to fully utilize the locality of resource access; when the CPU load is high, services are allowed to increase the supply of CPU resources beyond the soft affinity range to ensure the quality of service (QoS) of the business.

In C86 architectures, the communication latency of storage threads across core complexes (CCXs) is significantly higher than that within the same CCX. Based on soft affinity scheduling, frequently interacting storage service threads are scheduled in CCXs to reduce inter-process communication latency and thus reduce I/O latency.

5.2.3.5.3 Read Cache of libnfs

QEMU conducts I/O interactions with aSAN through the libnfs client. When the backend storage reaches a bottleneck (such as poor SSD performance, high pressure in the tier, and fully loaded business threads of aSAN), the I/O performance will encounter a bottleneck. By adding the read cache to each virtual disk, the read I/O performance can be effectively improved. (Upon virtual disk failure, a data loss will occur on the write cache but not on the read cache.) By configuring the read cache for VMs in the libnfs layer of QEMU, read I/O performance can be improved by more than 30% for read-intensive

businesses. The working principle of the read cache is as follows: After enabling the cache mechanism, libnfs caches some read data into local memory, where data can be read directly next time when needed. The detailed process is as follows:

1. A memory area is reserved in the libnfs layer as a cache area.
2. When a read operation is initiated, libnfs queries the hash value based on the disk offset, performs an index operation on the memory area by the hash value, and then checks whether the cache matches the current request based on the disk offset address of the area. If so, libnfs returns the data in the cache. If not, libnfs sends a request to the virtual storage.
3. When receiving data from the virtual storage, libnfs returns the data to QEMU and locally updates the cached data in the associated area and the disk offset of the cached data.
4. When a write operation is initiated, libnfs directly sends a write request to the virtual storage. When receiving data from the virtual storage, libnfs uses the locally written data to update the cache data and cache tags.

An elimination mechanism is configured for data in the read cache of the libnfs layer, and the write-through policy ensures that no data will be lost and all the written data is stored in the virtual storage. Cache data is eliminated at a granularity of 4 KiB, and the granularity for enabling the cache function is 1 MiB.

HCI provides the function of enabling or disabling the read cache of a single disk for VMs and allows the configuration of the read cache space. Enabling the read cache to improve the disk read performance of VMs will consume the corresponding memory space on the physical host. Therefore, it is recommended that the read cache space be set to 1%-10% of the disk space, with a maximum space occupation of 64 GB. The configurations of the read cache on the libnfs layer can be edited only when VMs are shut down. After the VMs are powered on, the read cache will preoccupy the memory space. The restriction configured for memory overcommitment must be compliant after the read cache is added for VMs.

NOTICE

The read cache configuration is disabled when the import and export operations are performed for VMs. It is enabled when snapshot taking, cloning, and migration are performed for VMs. For other operations, the default policy of HCI is followed to determine whether to enable or disable the read cache configuration. By default, it is disabled.

5.2.3.5.4 Smart I/O Pipelining

Traditional storage architectures use an I/O lock mechanism to enable different I/O threads to perform the same operation. The disadvantage of the lock mechanism is that, as the I/O depth grows, indicating an increase in the concurrency, the threads compete for the lock more frequently. When the concurrency is higher than a threshold, the performance deteriorates because too many threads need to compete for the lock and wait for the lock to be released.

The aSAN storage architecture uses the pipelining technology developed by Sangfor to shorten critical I/O paths. In addition, this architecture adopts the coroutine-based and asynchronous I/O technology to perform scheduling freely, which reduces blocking and implements lock-free I/O operations. As the I/O depth grows, indicating an increase in the concurrency, I/O performance can increase proportionally to enhance the performance of the storage architecture.

I/O pipelining is realized by dividing the whole I/O process into several steps, each of which is completed by a fixed thread, excluding the locking operation. This process is similar to the assembly line of industrial production. The production of a product is divided into several steps, and a worker is responsible for only one step. All the steps constitute an assembly line. After a step is completed, the product is handed over to the next worker to complete the next step through the assembly line.

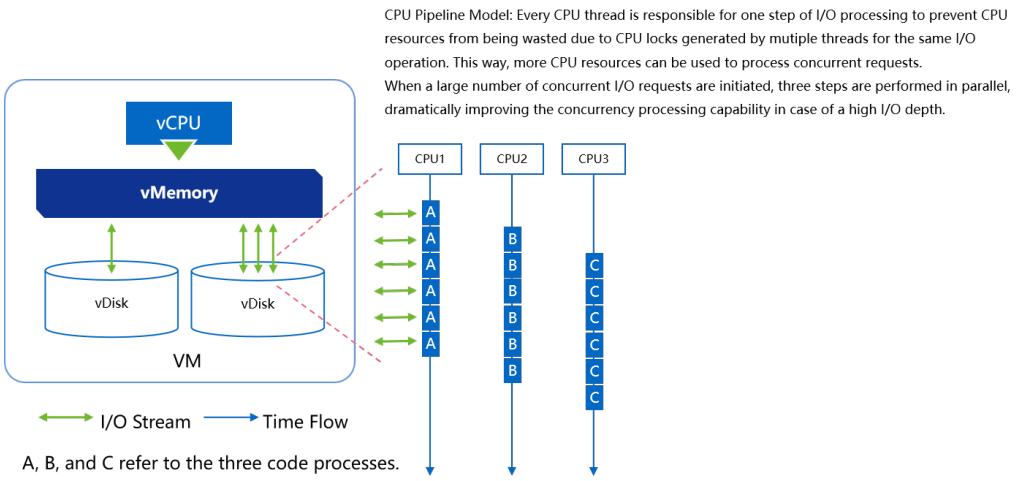


Figure 180: I/O Pipelining Process

The following figure shows the performance optimization after pipelining is implemented.

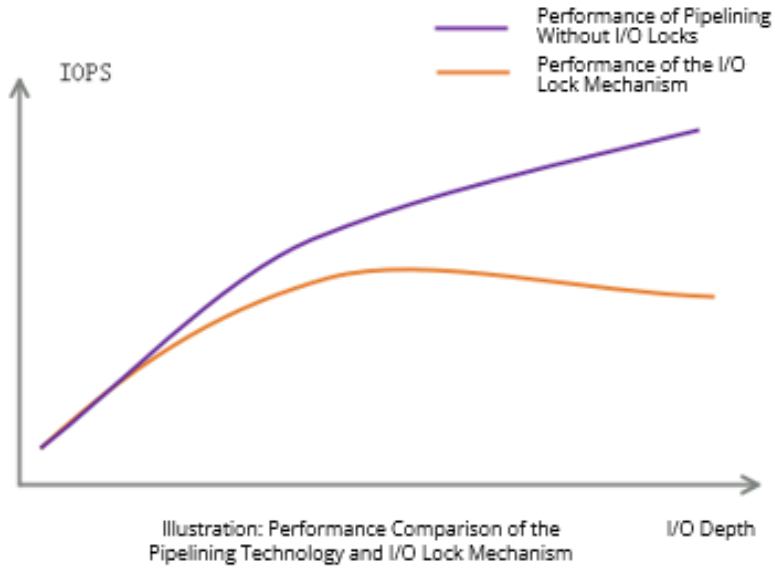


Figure 181: Performance Comparison Between Pipelining and the I/O Lock Mechanism

In addition, the code logic of the entire I/O path is optimized for aSAN to reduce lock competition and improve the concurrency processing capability. Some shared metadata resources in storage backend services need to be accessed frequently with locks. The competition for locks under high concurrency limits the I/O performance, which cannot be further improved. This bottleneck can be resolved as follows:

1. Break large locks into multiple small locks to reduce lock competition.
2. Transform the traditional multi-thread mode into the pipelining mode. In traditional multi-thread mode, one thread completes the whole I/O process, but in pipelining mode, multiple threads cooperate to complete the whole process, with each thread performing only a specified step. That is, turn the vertical processing into horizontal processing.
3. Reconstruct the data structure and business logic based on CPU cache, reducing CPU-level cache misses and improving CPU usage. The optimization by these sophisticated technologies greatly improves the I/O concurrency processing capability.

The test results show that the IOPS is significantly improved when the I/O depth is 128, and no sharp IOPS drop occurs when the I/O depth grows. In addition, the IOPS in case of high I/O depth is improved by more than 50% on average and by almost 100% at its highest over the earlier versions.

In addition, the smart pipelining technology is introduced to the new version of HCI and implemented as follows:

1. Reduce pipelines in case of a low I/O depth. In the pipelining mode used by virtual storage in the earlier version, each step of an I/O request is processed by an independent thread. After a step is completed, the request is handed over to another thread for processing in the next step. In case of a high I/O depth, this mode can effectively prevent lock conflicts and improve the concurrency processing capability. However, it also results in additional CPU overhead at every step switch and increased latency, which may cause reduced performance in low I/O depth scenarios. (Latency in high I/O depth scenarios will also increase but is lower than that in single-thread processing mode because a single thread may take a long time to wait for other threads to complete their processing during peak hours.) In version 6.9.0, HCI can detect the I/O depth, and it will not switch the thread but use the current thread to process the next step when the depth is low. This way, low latency can be achieved in case of a low I/O depth, and a high concurrency processing capability can be implemented in case of a high I/O depth.

2. Increase pipelines in case of a high I/O depth.

5.2.3.5.5 High-Latency Submodule Optimization

Memory allocations and copies are reduced for submodules with high latency on the I/O path to optimize the memory structure, as well as system calls to add fast paths to some of the logic.

5.2.3.5.6 Shared Library Overhead Optimization

The unified management of some common processes in the shared library is implemented, such as saving I/O path parameters to prevent submodules from applying for memory saving independently. The client-server transmission protocol stack is optimized to reduce the time consumption of protocol encoding and decoding, debugging logic is optimized to prevent increasing locks and atomic variables are preferentially used.

5.2.3.5.7 Thread Model Optimization

The self-developed dynamic pipeline scheduling policy can improve the thread performance in low I/O depth by simplifying the pipeline delivery process, and enhance the concurrent processing performance of block storage in high I/O depth by pipeline delivery.

5.2.3.5.8 IOPS Improvement

The mount points of block storage can meet the dd command test performance recommended by Dameng no matter whether RDMA and Turbo mode are enabled.

The IOPS in single I/O depth for small blocks is significantly improved, and the storage latency is greatly reduced. After enabling RDMA and Turbo mode, the IOPS of random reads in single I/O depth for x86 hybrid and all-flash datastores in 4K display increased from 5000 to 7000, and the IOPS of random writes increased from 4400 to 8000. The IOPS of random reads in single I/O depth for ARM hybrid and all-flash datastores in 4K display increased from

4000 to 5700, and the IOPS of random writes increased from 2950 to 4700.

5.2.3.6 Zero-Copy Memory

aSAN is designed with multiple layers, where data needs to be transmitted between different layers in I/O business flows. Multiple memory copies are made during data transmission, and the data must pass through the kernel protocol stack. I/O operations on small blocks consume many CPU resources, which can easily bring a performance bottleneck when random reads are performed in case of a high I/O depth. I/O operations on large blocks consume memory bandwidth, which cannot cause a storage bottleneck. However, if the storage uses too much memory bandwidth, a bottleneck in the business will occur.

The zero-copy memory is a high-performance IPC mechanism developed by the aSAN team of Sangfor based on shared memory. It ensures that processes on a physical host can share data efficiently. Data will not be copied within the memory during processing of local I/O flows in the system, thus reducing resource overhead and improving performance.

Processing of read requests: When a read I/O request of an application reaches the backend storage layer, the layer reads data from the storage media and saves it to the shared memory without transmitting the data to the upper layer. The distribution switching layer, feature layer, and other layers in aSAN can directly access the data in the shared memory. Finally, the access layer returns the data to the application.

Processing of write requests: When a write I/O request of an application reaches the access layer, the layer writes data to the shared memory without transmitting the data to the lower layer. The distribution switching layer, feature layer, and other layers in aSAN can directly access the data in the shared memory. Finally, the backend storage layer obtains the data from the shared memory and writes it.

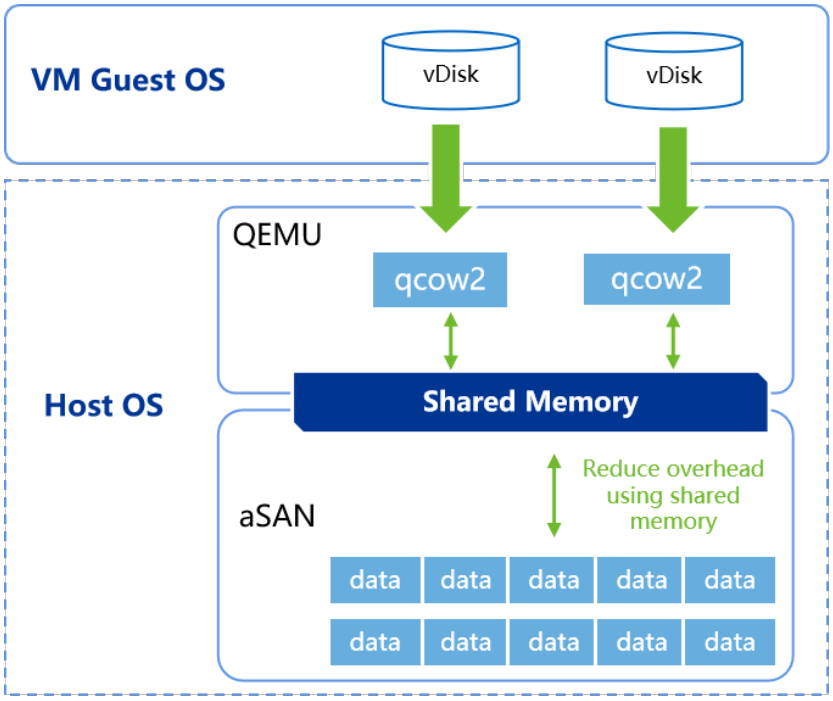


Figure 182: Working Principle of the Shared Memory

Before the shared memory is implemented, aSAN processes on a host using TCP or Unix sockets for data transmission, and the I/O data must pass through the kernel protocol stack, resulting in multiple data copies and consumption of many CPU and memory bandwidth resources. Shared memory can ensure that aSAN processes transmit and access shared memory data through metadata information, achieving zero copies of I/O data, reducing the delay caused by local data transmission and copies, and improving the performance of local data reading and writing.

5.2.3.7 Disk-Level QoS

aSAN provides the automated QoS, which can be configured at the virtual disk level. The following three storage policies are supported:

- **High performance**: High-performance objects occupy more tier space to achieve a higher cache hit rate, thus ensuring high performance. When high-performance data blocks get cold and meet the elimination condition, the data blocks will be put into the hot data queue again instead of being eliminated instantly. The data blocks will be eliminated from the tier only

when the elimination condition is met again. This processing method provides privileges for high-priority data, preventing it from being eliminated easily.

- **Default performance**: Default-performance objects are processed by the normal elimination algorithm. When data blocks get cold and meet the elimination condition, they are eliminated from the tier. That is, default-performance objects have a half possibility to preempt tier space compared to high-performance objects.
- **Low performance**: Low-performance objects are restricted in using the tier space, ensuring sufficient space for high-performance and default-performance objects.
- aSAN allows a single low-performance disk to occupy at most 10 GB tier space. The rest of the space is available for preemption by high-performance and default-performance objects.

5.2.3.8 RDMA

5.2.3.8.1 Technical Background

TCP/IP Network Communication Limitations:

Common network protocols such as TCP/IP are not designed for high-performance application scenarios, and they need to send data in two copies from the local user space to the user spaces of multiple other hosts when performing write I/O, during which several memory copies are made. The process of data transmission between applications on two servers is as follows:

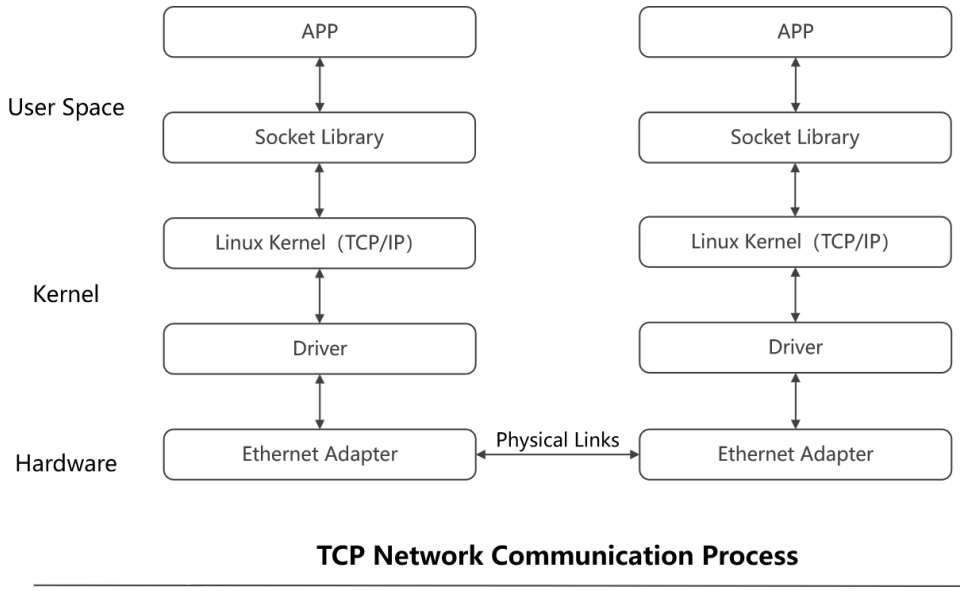


Figure 183: TCP Network Communication Process

The data is first copied from the application cache to the TCP stack cache in the kernel, then to the driver layer, and finally to the adapter cache. Memory copies require CPU interventions, which causes processing latencies to be prolonged by tens of microseconds, and CPU performance to decline dramatically, severely compromising data computation.

In high-speed networks, the huge overhead of using TPC/IP for data movement and copies limit the outbound bandwidth between hosts, making it impossible to provide efficient communication with the growing demand for low-latency networks in various industries.

RDMA Overview:

Remote Direct Memory Access (RDMA) is the first network communication protocol used in high-performance computing, which enables applications to read and write to remote memory in user mode directly without operating system processing, kernel intervention, or memory copies, which saves a lot of CPU resources, improves system throughput, and reduces system network communication latency. The following are some of the features of RDMA:

1. CPU Offload: Applications can access the memory of a remote host without consuming its CPU, whose cache is not filled with the contents of the accessed memory.

2. Kernel Bypass: Applications can perform data transmissions directly in user mode without context switches between kernel and user modes through a dedicated verbs interface instead of a TCP or IP socket interface.
3. Zero-Copy: Applications can directly access the virtual memory of cluster devices, enabling data to be sent to and received from the buffer directly without being copied to the network layer while not involving any network software stack.

Based on kernel bypass and zero-copy, RDMA is characterized by low latency, low CPU overhead, and high memory bandwidth utilization.

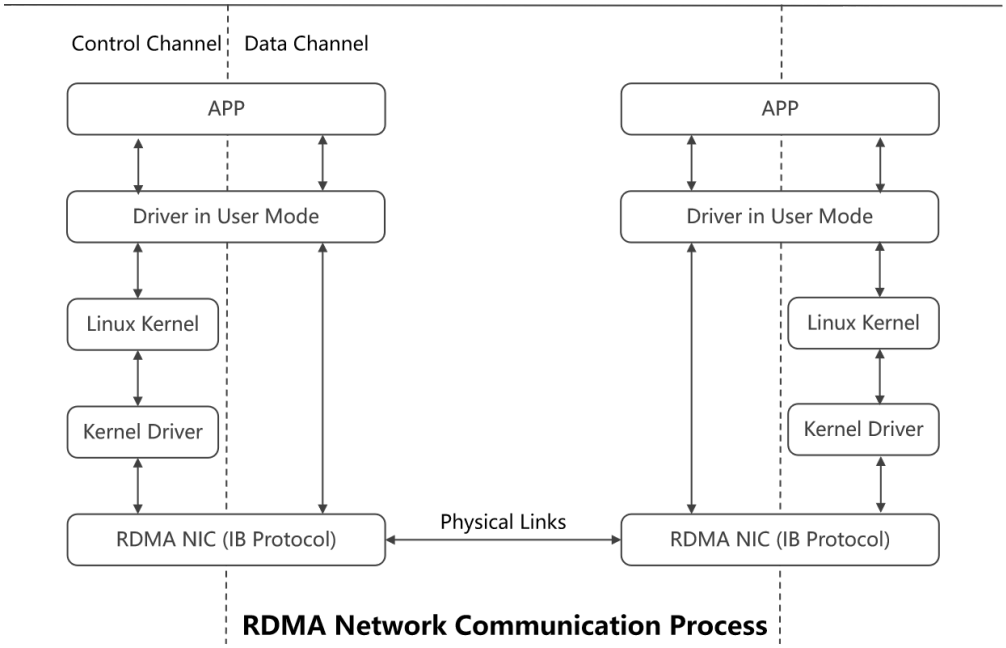


Figure 184: RDMA Network Communication Process

RDMA Implementation:

There are currently three network protocols that implement RDMA: InfiniBand, RDMA over Converged Ethernet (RoCE), and iWARP. Among them, InfiniBand is specially designed for RDMA, which guarantees reliable transmission on the hardware level and has the best performance, but the price of the corresponding adapters and switches is very high. RoCE and iWARP both support the corresponding verbs interfaces based on Ethernet and only require the configuration of specific adapters with low costs. As shown in the following figure, RoCE is available in two versions, RoCEv1 and RoCEv2. RoCEv1

is an Ethernet link layer protocol (which requires switches to be equipped with flow control technologies such as PFC to ensure reliable transmission on the physical layer), while RoCEv2 is implemented on the UDP layer of Ethernet TCP/IP.

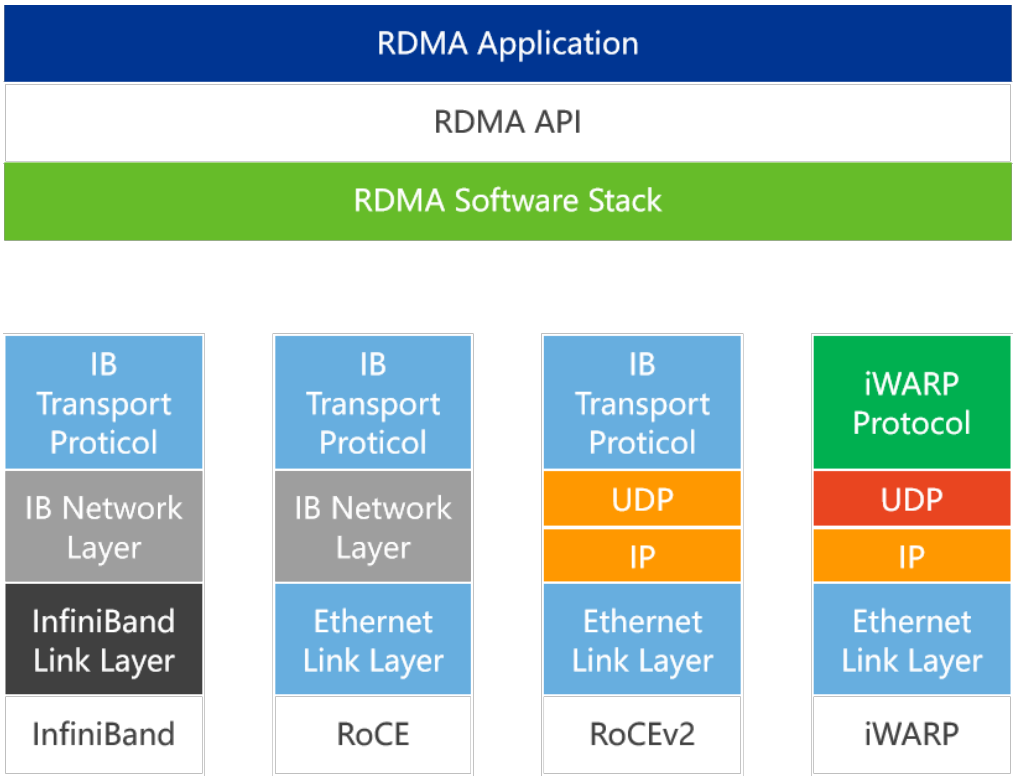


Figure 185: RDMA Implementation

RDMA Advantages and Disadvantages:

- Advantages

1. High Performance and Low Latency: Applications can bypass the kernel processing of the operating system to interact directly with the adapter for communication, which significantly reduces data transmission latency and CPU usage and greatly improves system performance.
2. Efficient Data Transmission: With zero-copy technology, application data can be transmitted directly without being copied to the network layer, which reduces unnecessary copying overhead and enhances transmission efficiency.

3. Massive Concurrent Processing: The communication mechanism enables efficient data exchange and synchronization between compute nodes through multiple independent communication streams, which increases the efficiency of massive concurrent computing.
4. Flexible Memory Access: Remote nodes can directly access local memory, which makes data processing flexible. For example, in a distributed computing system, a node has direct access to the memory data of another node without intermediate network transmissions.
5. Hardware Offload: The network protocol processing can be offloaded to the adapter hardware, which reduces the CPU load and significantly improves network processing performance.

- Disadvantages

1. Hardware Dependency: RDMA requires specific adapters that are typically more expensive than traditional Ethernet adapters, and may need additional hardware and software configurations in cases not supported by the operating system or network devices.
2. Security Risks: Direct access to local memory by remote nodes may pose security risks. For example, an attacker may obtain improper access permissions to a remote system by exploiting vulnerabilities or misconfigurations in the RDMA protocol.
3. Network Size Limitations: Aiming to reach high performance and low latency, RDMA requires high network stability and reliability. In large-scale networks, RDMA performance may suffer due to topology complexity and unpredictability. Furthermore, RDMA has a low tolerance for network packet loss and jitter to prevent significant performance degradation, which may limit its application in certain scenarios. Applications can maximize performance within a certain scale using RDMA and can expect better connectivity and packet loss tolerance using TCP.

5.2.3.8.2 Technical Principles

In HCI 6.10.0, RDMA is applied to the storage area network of the virtual

storage platform and is mainly deployed in the processes of the storage client and server for cross-node network communication. The RDMA feature can replace the TCP network for data transmission, which can reduce the network latency between the storage client and server, and significantly improve the performance of the virtual storage platform and the throughput of the storage client and server.

Currently, the storage client and server communicate with each other through RDMA using the RoCEv2 protocol to ensure data correctness and ordering in reliable connection mode and implement the overlay network interface by integrating with the virtual storage framework.

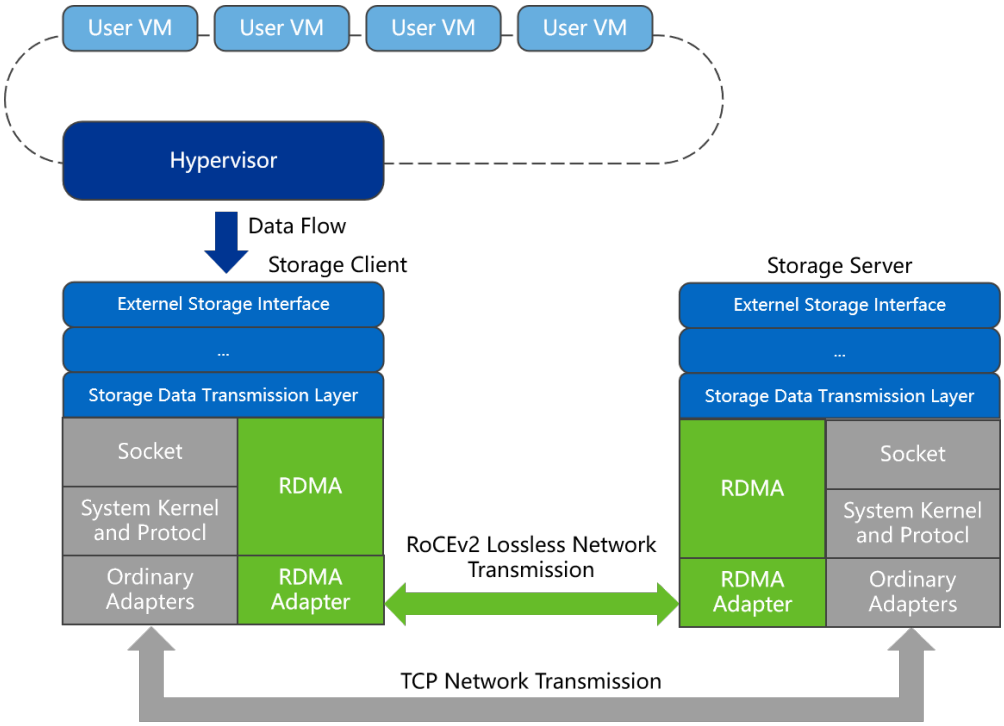


Figure 186: RDMA Principle

RDMA negotiates information such as queue depths, and cache addresses on receiving and sending sides during the connection creation. After this, RDMA performs data transmissions containing both one-way and two-way information. The main mechanisms are as follows:

Data Transmission Management:

There are two modes of RDMA communication, one-way and two-way, each of

which offers advantages and disadvantages. One-way communication allows one side to read or write to memory directly without a response from the other side, while two-way communication requires the participation of both sides. The virtual storage platform quickly transmits data by combining one-way and two-way communication through a well-developed data transmission management mechanism. The following are the principles of the corresponding transmission modes:

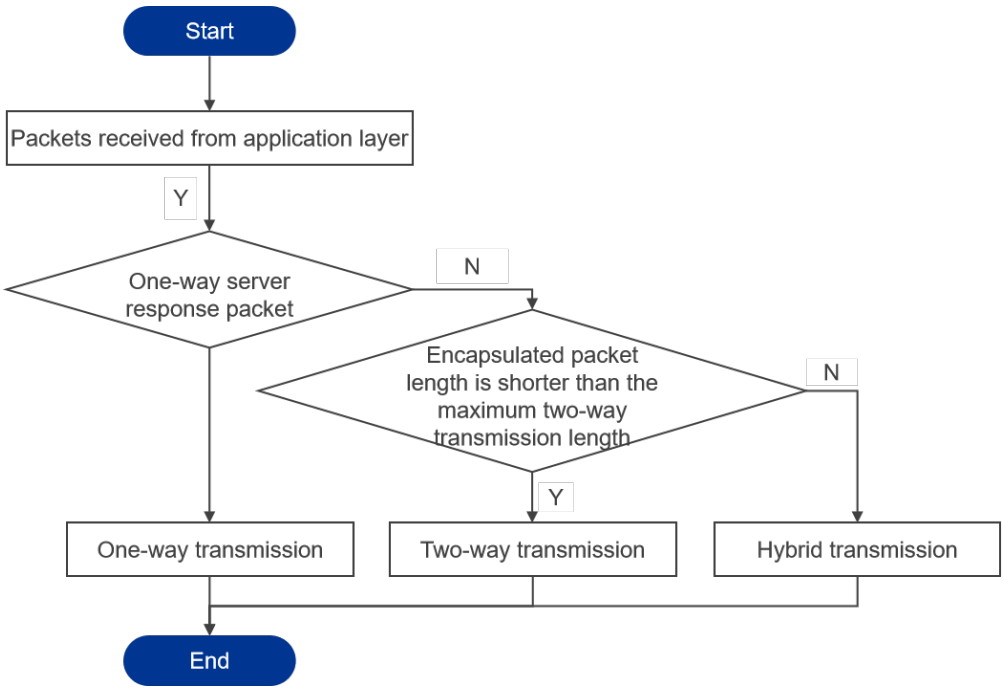


Figure 187: Data Transmission Management

- One-Way Transmission

The storage server obtains the memory created by the storage client by parsing messages received from the client and then writes data directly to the client in one way.

- Two-Way Transmission

The storage client sends data directly to the storage server in a two-way when the length of all data sent by the client is shorter than that of the predefined maximum two-way transmission.

- Hybrid Transmission

The storage client has two types of data write requests: regular write requests and fast write requests. Regular write request workflow: The storage client sends a two-way message encapsulating the header of the data it wants to send to the storage server through two-way communication. The server receives and parses the message, and obtains all data contents from the client through one-way reading. Fast write request workflow: The storage client directly writes the data to be sent to the storage server in one way, and then sends a two-way message to inform the server about the memory where the written data is located, and the server specifies the memory and reads the written data through the message content.

Window Management:

In order to ensure efficient and stable communication between the sending and receiving sides for RDMA, the virtual storage platform provides a set of asymmetric window management mechanisms to effectively perform congestion control on the communication links and prevent RNR errors. This mechanism applies SEQ-ACK mode in the application layer with the following workflow:

1. After the connection is created, the receiving and sending sides exchange their receive queue lengths through private data and set them as the size of their send windows.
2. For each message sent by the sending side, the send windows are reduced by 1. After the windows are reduced to 0, sending stops.
3. For each message received by the receiving side, the count of ACKs is increased by 1. The count of ACKs is carried to the other side as an immediate number for each message sent.
4. When the ACK count of the receiving side reaches half of the size of the receive windows (equal to the send windows of the sending side), it triggers a pure ACK sending (without data) and releases the send windows of the sending side in time.

5.2.3.8.3 UI Display

Log in to HCI and go to **Storage** > **Virtual Storage** > **Advanced** > **RDMA** to enable RDMA for a virtual datastore. You can click **View Compatibility** to view whether the adapter models of the storage network interfaces support the RDMA feature.

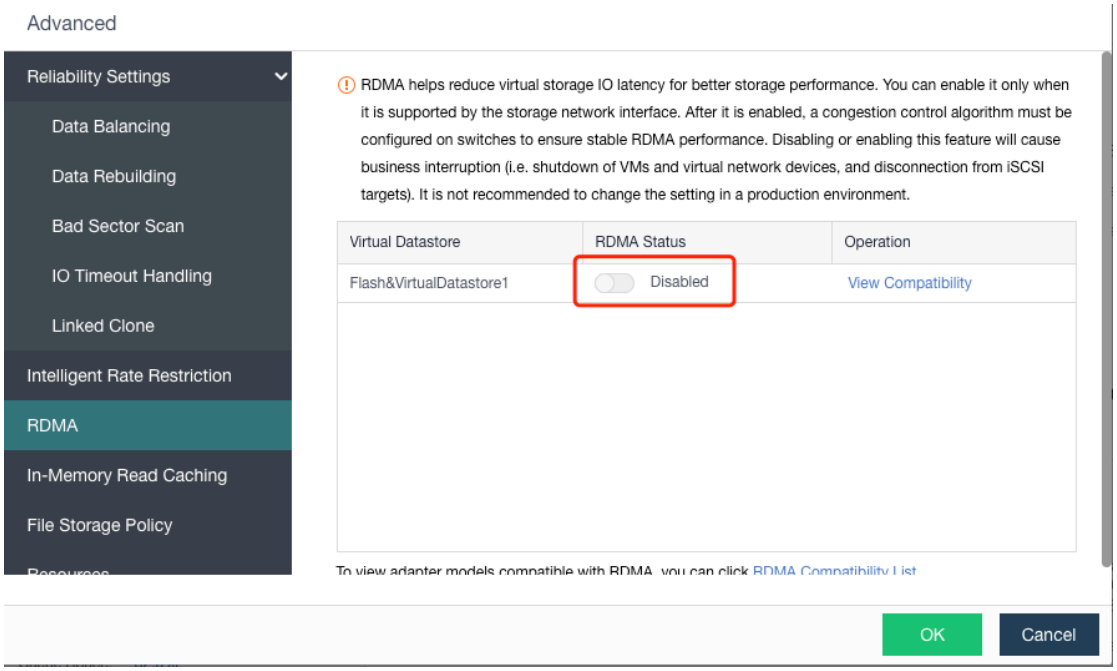


Figure 188: Enable RDMA

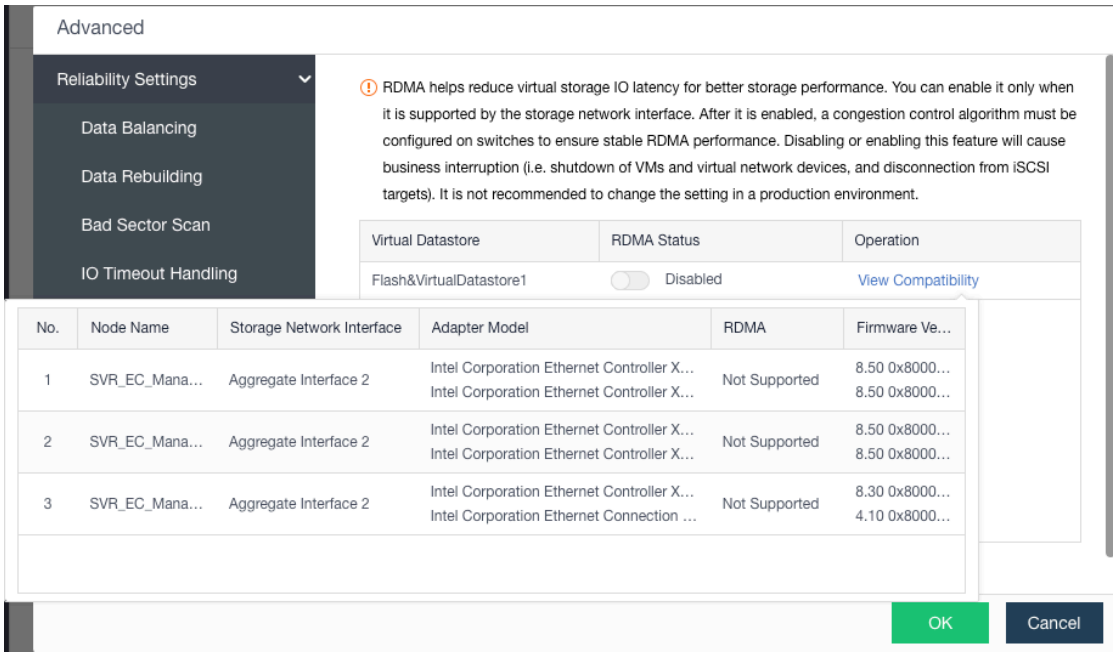


Figure 189: View RDMA Capability

Sangfor HCI integrated virtual storage with RDMA technology using the RoCEv2 transmission protocol, which improves storage write performance by more than 35% with one and two-way hybrid communication.

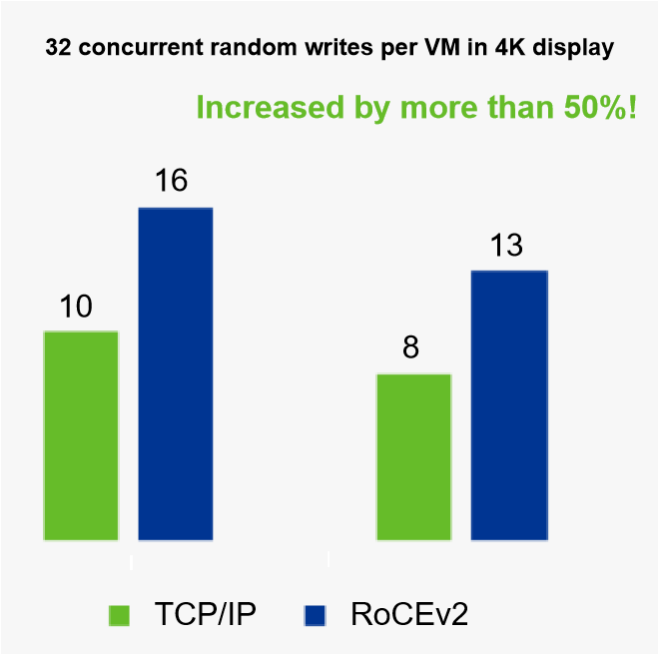


Figure 190: Write Performance Comparison between TCP/IP and RoCEv2

Precautions:

1. To enable RDMA, the storage switch must be configured with a congestion control algorithm; otherwise, business services may encounter network congestion during off-peak hours.
2. Enabling RDMA consumes no memory on nodes with 8 hard disk slots or less, but 2 GB of memory on nodes with 24 hard disk slots.
3. Please make sure that the model of the storage adapter and its firmware version support RDMA. To query the adapter models and firmware versions supported by RDMA (passed the beta test), go to the adapter vendors' official website or Sangfor HCI Compatibility Chart via: https://www.sangfor.com/cloud-and-infrastructure/products/hci-compatibility-chart.

For more information, refer to the ***SANGFOR_HCI_6.10.0_User Manual_EN***.

5.2.3.9 SPDK-Based Turbo Service

HCI carries customers' critical business services, including database applications, big data analytics, artificial intelligence, and machine learning. These services require high storage I/O performance, which is directly associated with service processing speed and user experience. For this reason, Sangfor integrated HCI with the SPDK solution, significantly improving the performance of VMs running I/O tests and I/O-intensive services.

5.2.3.9.1 SPDK Overview

The Storage Performance Development Kit (SPDK) is a set of tools and libraries provided by Intel for building high-performance storage that encapsulates some of the resource management components (including CPU, memory, and PCI devices, etc.) of the Data Plane Development Kit (DPDK). SPDK Application Framework is the core of the SPDK, used for programming and implementated based on user mode, polling, asynchronization, and lock-free. It was originally designed for use in scenarios where NVMe SSDs are used as backend storage to speed up the I/O and optimize the I/O latency of the application software.

SPDK is used to write high-performance and scalable storage applications in user mode. Compared with NVMe drivers in kernel mode, it can significantly reduce the latency of NVMe commands and increase CPU IOPS through the following techniques:

1. User Mode: Moves all necessary drivers to user space and operates the hardware directly in it to prevent numerous system calls and to enable zero-copy based directly on memory location in user mode.
2. Polling: Polls the hardware instead of relying on interrupts to reduce total latency and latency variance.
3. Asynchronization: Provides a number of asynchronous interfaces for event handling to prevent programs from being blocked and to call callback functions after operations.
4. Lock-Free: Uses lock-free queues and programming to prevent performance degradation due to locks in the I/O path.

5.2.3.9.2 Terminologies

Term/Abbreviation	Definition
SPDK	Storage Performance Development kit, a toolkit used for building high-performance storage.
bdve	Block devices compatible with storage of different types. The block device layer is defined by the SPDK.
DPDK	Data Plane Development Kit is the open-source toolkit used by SPDK for building high-performance network storage applications.
SPDK vhost	A component of SPDK that implements high-performance virtualized storage applications. It improves storage performance and efficiency by mapping the storage device of a VM directly to that of the host, eliminating the data replication between the VM and the host.
vhost-user	A virtualization technology that maps the network and storage devices of a VM directly to those of the host. It enables efficient communication between the VM and the host by isolating the control plane and data plane of the storage device from the VM network.
QEMU	An open-source VM monitor and emulator that simulates various hardware architectures and operating systems.
Polling	In SPDK, polling is a common loop-based programming pattern in which an application implements asynchronous I/O operations by continually querying the status of an I/O operation after it is committed until it completes or an error occurs.
Poller	In SPDK, a poller registers a callback function in the main loop of an application and calls it during each polling cycle to periodically perform tasks such as checking status, updating counters, etc.

Table 10: Terminologies

5.2.3.9.3 Technical Principles

SPDK vhost-user is the VM storage backend running as a process on the host. It obtains and processes VM I/O requests from the shared ring buffer by polling, and finally notifies the VM of the processed data through the ring buffer. The whole process is highly efficient without lock protection nor interrupt overhead, and mainly involves the following parties:

VM Storage Frontend Driver: virtio-blk or virtio-scsi driver usually, mainly receives and saves I/O requests from the Guest OS into the ring buffer for backend processing and handles completion notifications from the backend.

Shared Ring Buffer: The memory shared by the VM frontend driver and the host storage backend for the transmission of data, commands, and execution results between them.

SPDK vhost-user Backend: The backend of the VM storage is an independent process that handles the core logic of I/O requests and enables efficient I/O operations through storage drivers in user mode.

SPDK vhost-user interacts with VM data through shared memory. It uses polling to obtain I/O requests from the ring buffer, which eliminates interrupts and multiple switches between user and kernel modes in the I/O path and bypasses QEMU and KVM to implement zero-copy. With the SPDK bdev framework, it is possible to connect to storage of various types in the backend, such as HCI to virtual storage.

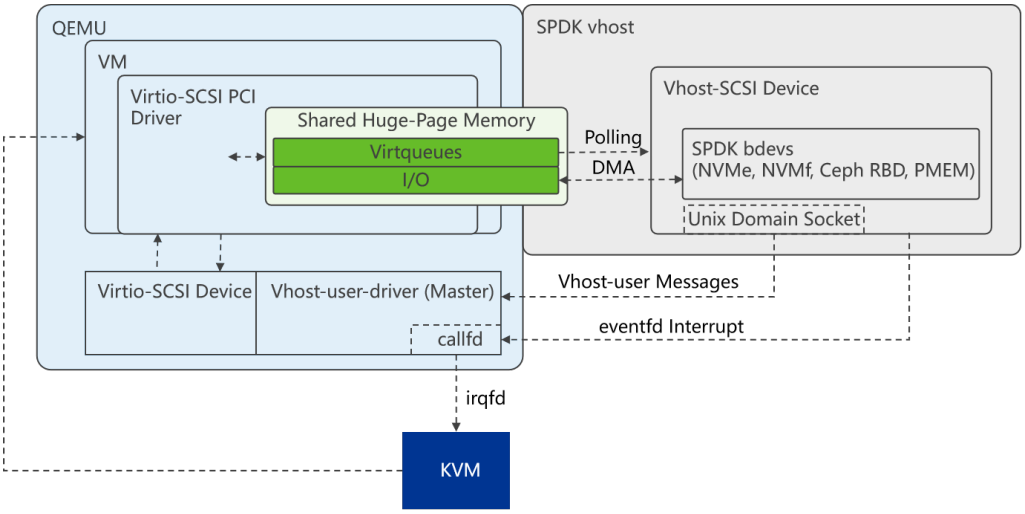


Figure 191: SPDK VM Principle

5.2.3.9.4 Differences in I/O Processing Between SPDK VMs and Ordinary VMs

Ordinary VM:

The following figure shows how the I/O processing of an ordinary VM stored on virtual storage on HCI works:

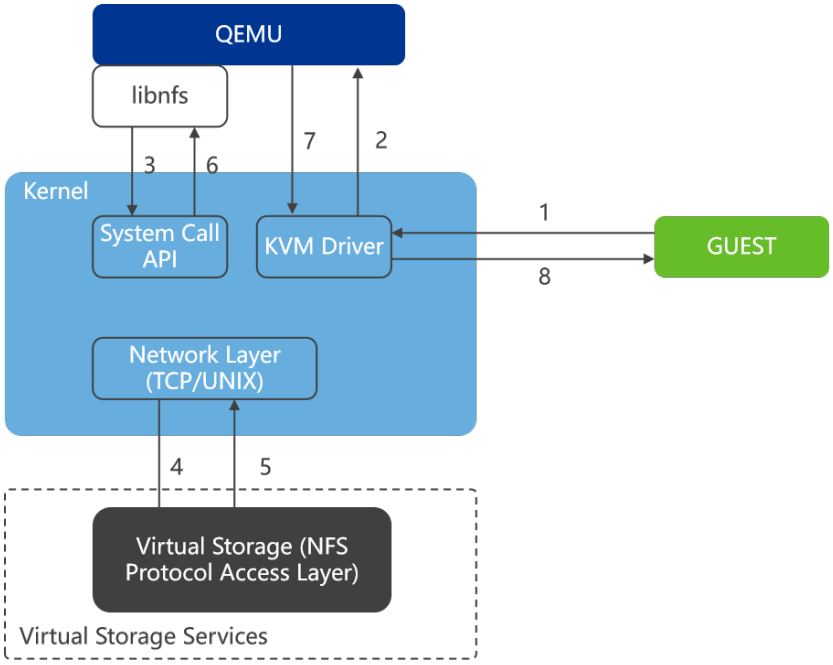


Figure 192: Ordinary VM I/O Processing

1. Guest OS initiates an I/O request to the virtio-blk disk, which will perform a PIO operation, causing a VM-EXIT to the KVM driver.
2. The KVM driver processes the VM-EXIT event and returns it to the QEMU through kvm_ioctl.
3. After the QEMU I/O stack processes the I/O request, QEMU will call the libnfs interface to send the I/O request through the network socket.
4. The I/O request is passed to the virtual storage through the kernel socket.
5. After the virtual storage processes the I/O request, it replies to libnfs through the kernel socket.
6. libnfs notifies QEMU of the processing completion of the I/O request.

7. QEMU injects hardware interrupt notification to Guest OS through kvm_ioctl.
8. At VM-ENTRY, the hardware interrupt for the processing completion of the I/O request is injected into the Guest OS.

The I/O processing of an ordinary VM may experience the following performance bottlenecks:

1. Long I/O Path

- I/O requests from a VM need to successively go through the KVM driver, QEMU, kernel network stack, and virtual storage, and finally return to the VM in the original way, which inevitably increases the I/O latency (by about 30% in total) due to the long processing path.

2. High CPU Consumption

- QEMU has to switch the KVM driver and system calls from kernel mode to user mode for I/O requests, which causes huge CPU consumption when the VM IOPS is very high.
- I/O requests need to go through multiple memory copies and CPU interventions, consuming CPU resources.

3. Performance Limited by Multiple Threads

- All I/O requests are processed by the QEMU process, whose main thread only has up to 200,000 IOPS.
- The libnfs threads, which correspond one-to-one with the virtual disks in QEMU, and the NFS connection processing threads of the aSAN client may become bottlenecks.

Ordinary VMs use the NFS protocol to communicate with virtual storage clients on the same node, and their I/O needs to pass through the kernel layer of the platform system, causing about 70 microseconds of latency. In the NVMe SSD scenario, the hardware I/O latency is about 30 microseconds, while the software I/O latency has a huge impact on the I/O performance cap.

SPDK VM:

In HCI 6.10.0, the software latency of I/O processing for SPDK VMs is

significantly reduced. The I/O processing of an SPDK VM has the following advantages:

1. Short I/O Path

The I/O processing for a VM to access virtual storage requires only two steps. The SPDK vhost module first reads the I/O data within the shared huge-page memory and calls the aSAN API to access the virtual storage. Then the virtual storage processes and writes I/O requests to the shared huge-page memory and notifies the Guest OS.

2. Low CPU Consumption

The I/O path reduces the switches from user mode to kernel mode, process context switches, and memory copies by bypassing the QEMU layer and the kernel network protocol layer, which lowers CPU resource overhead.

3. Fewer I/O Request Thread Stuck Points

I/O requests from ordinary VMs need to go through the QEMU I/O submodule and connect to the virtual storage client through the NFS protocol layer network. I/O requests from SPDK VMs only need to go through the SPDK vhost-user service and virtual storage, bypassing the KVM driver, QEMU, and kernel network stack, which significantly shortens the processing path and reduces the software I/O latency of accessing SSD cache disks from virtual storage.

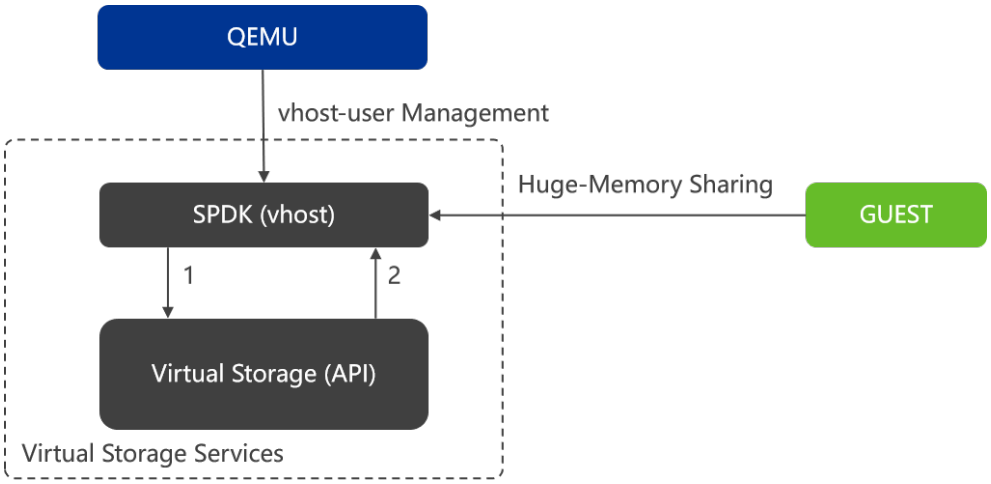


Figure 193: SPDK VM I/O Processing

Further Optimizations Based on SPDK:

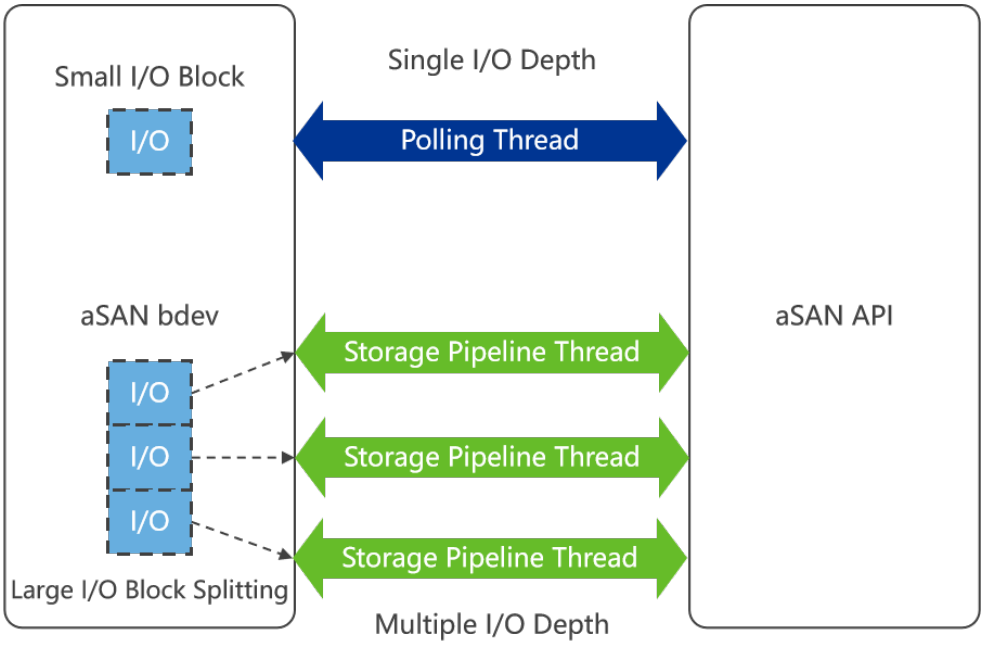


Figure 194: Optimizations Based on SPDK

1. SPDK can dynamically switch to polling mode to improve I/O performance when there is a VM access, and automatically switch to interrupt mode to reduce CPU overhead when there is no VM access.
2. The SPDK vhost module accesses shared huge-page memory by polling to reduce the interruption time of threads responding to events.
3. The process of SPDK access to virtual storage has been deeply optimized: single-depth I/O requests access virtual storage directly in the thread context to reduce thread switches; multi-depth I/O requests are switched to the storage pipeline thread to access virtual storage to prevent the polling thread from reaching a bottleneck; large I/O requests are aligned and divided into multiple I/O requests of 128 KB and distributed to multiple storage pipeline threads for concurrent processing to improve concurrent I/O processing performance.
4. The SPDK framework has been transformed to accommodate advanced features such as double instance checks, HA, and in-place active upgrades for VMs, ensuring a smooth SPDK VM experience.

5. Ordinary VMs and SPDK VMs can coexist and convert to each other on HCI. You can choose whether to use SPDK based on the VM services and on the balance between performance and functionality.

6. After completing an I/O, SPDK will inject an interrupt into the VM to notify the business service. However, frequent interrupt injection at high concurrency will lead to increased VM exit overhead, which slows down internal VM I/O sending. Interrupt merging can adaptively adjust the interrupt injection time range to improve IOPS at high concurrency, without increasing I/O latency at low concurrency.

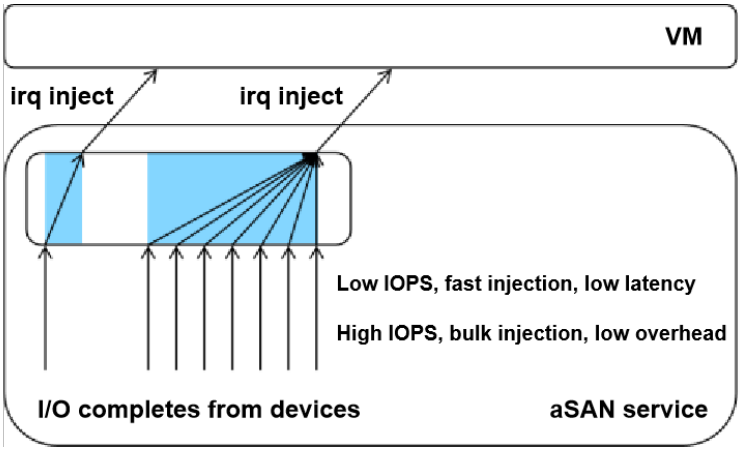


Figure 195: Interrupt Merging

5.2.3.9.5 UI Display

1. Log in to HCI, go to **Systems** > **Advanced** > **Virtual Machine**, and select **Enable** of **Turbo Service**.

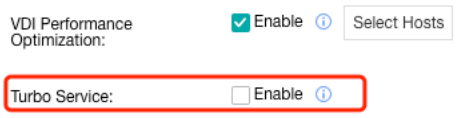


Figure 196: Enable Turbo Service

2. Select **Enable Turbo Mode** when creating and editing a powered-off VM with huge-page memory enabled and vmTools installed.

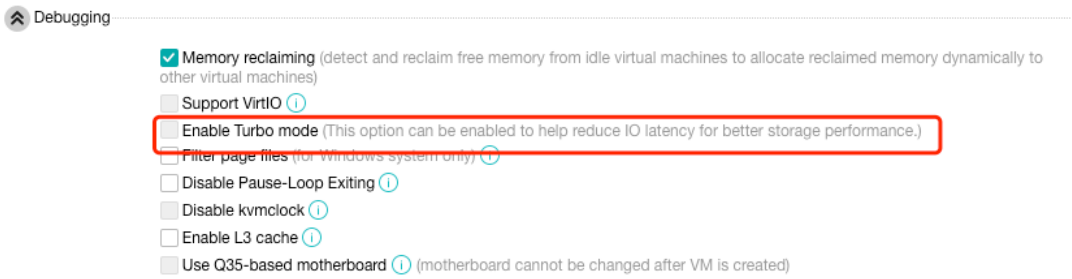


Figure 197: Enable Turbo Mode

5.2.4 High Reliability

With the multi-replica mechanism, Sangfor aSAN provides data redundancy protection and a perfect troubleshooting mechanism. The failure prevention mechanisms include the quorum mechanism for preventing split-brain and the mechanism for detecting and processing unhealthy disks. The troubleshooting mechanisms include the mechanism for detecting and processing silent data corruption, the mechanism for rapid and smart data rebuilding upon a failure, the snapshot storage mechanism for data restore based on backups upon a failure, and the mechanism for data protection by using spare disks.

5.2.4.1 Storage-Based Snapshot

According to the definition provided by the SNIA, a snapshot is a fully usable copy of a specified data set. The copy includes an image of the corresponding data at a certain time point. A snapshot can be a copy or replica of the represented data.

5.2.4.1.1 Scenarios

Scenario 1: online data backup

The storage-based snapshot is a measure to protect business data.

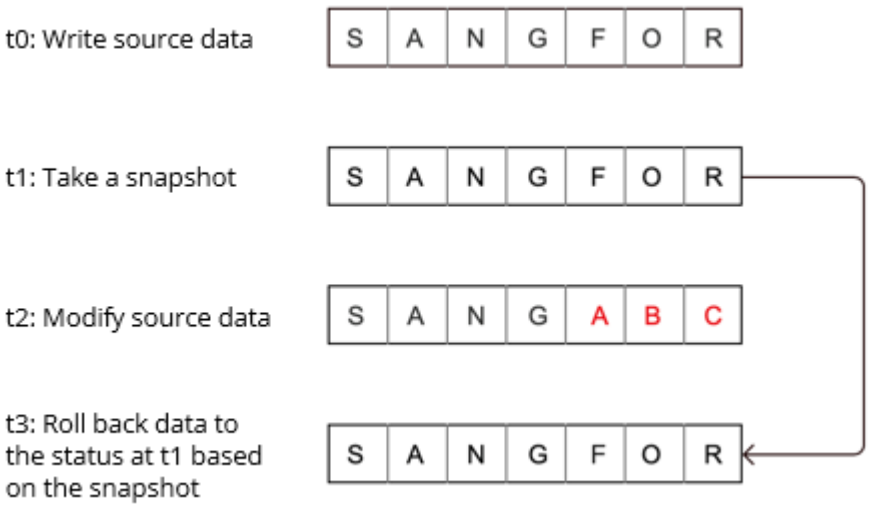


Figure 198: Storage-Based Snapshot Workflow

The figure above supposes that at the time of t0, a VM generates a complete piece of source data. At the time of t1, a snapshot of the data is created for the VM. At the time of t2, the source data is modified or damaged due to misoperation or system error. In this case, the VM data can be restored to the point of snapshot creation (the time of t1) through snapshot-based rollback.

Snapshots are generally created for VMs before software or operating system upgrades and device replacement for the data center to ensure quick rollback in the case of any failure.

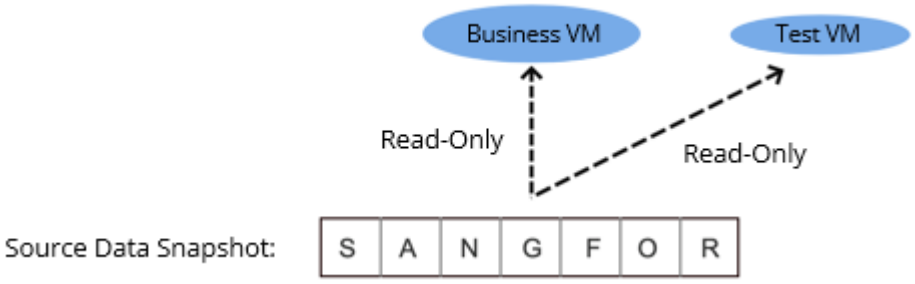


Figure 199: Read-Only Source Data

5.2.4.1.2 Technical Principles

Storage-based snapshots are implemented based on the redirect-on-write (ROW) technology.

Data writing after snapshot creation:

When taking a snapshot of a VM, the system sets all virtual disks as read-only and generates a corresponding quantity of new disk files (snapshot space). After the snapshot is created, all new data and modifications to the source data of the VM will be written to the newly generated snapshot space, and the mapping between the logical addresses of the source virtual disks and the snapshot space will be written to the mapping table. The following figure shows the process of writing data after snapshot creation.

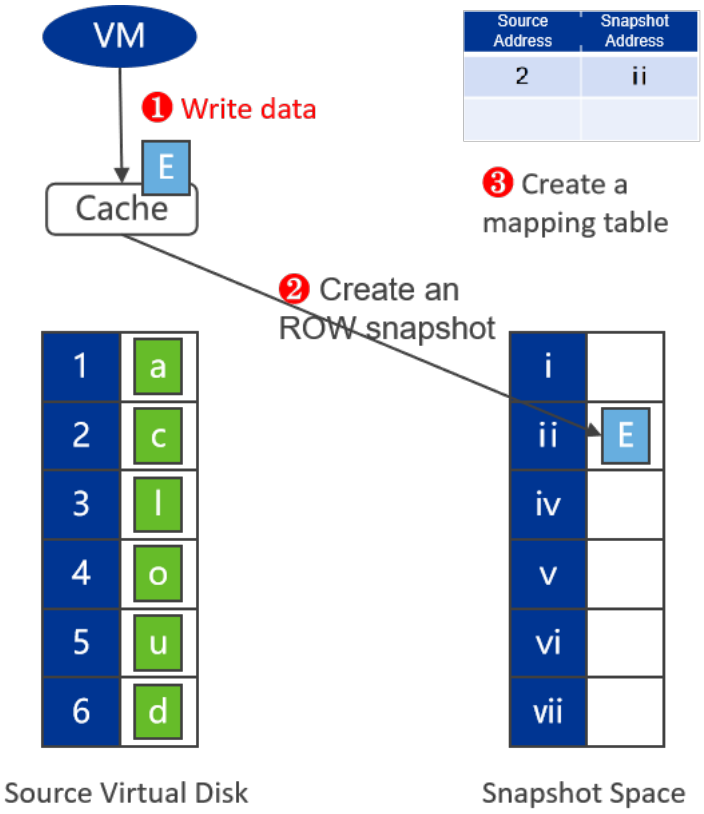


Figure 200: Data Writing After Snapshot Creation

1. A VM writes data, trying to change data c to data E.
2. An ROW snapshot is created for data E in the snapshot space according to the logical address ii.
3. The system writes the mapping between the logical addresses of the source virtual disk and the snapshot space into the mapping table,

mapping data c at the source address to the data at the snapshot address ii.

Data reading after snapshot creation:

There are two scenarios for the data reading by the VM after snapshot creation:

- If the data to be read is written before snapshot creation and is not modified after snapshot creation, the data is read from the source virtual disk.
- If the data to be read is written after snapshot creation or is a modification to the data in the source virtual disk, the data is read from the snapshot space.

The following figure shows the process of reading data after snapshot creation.

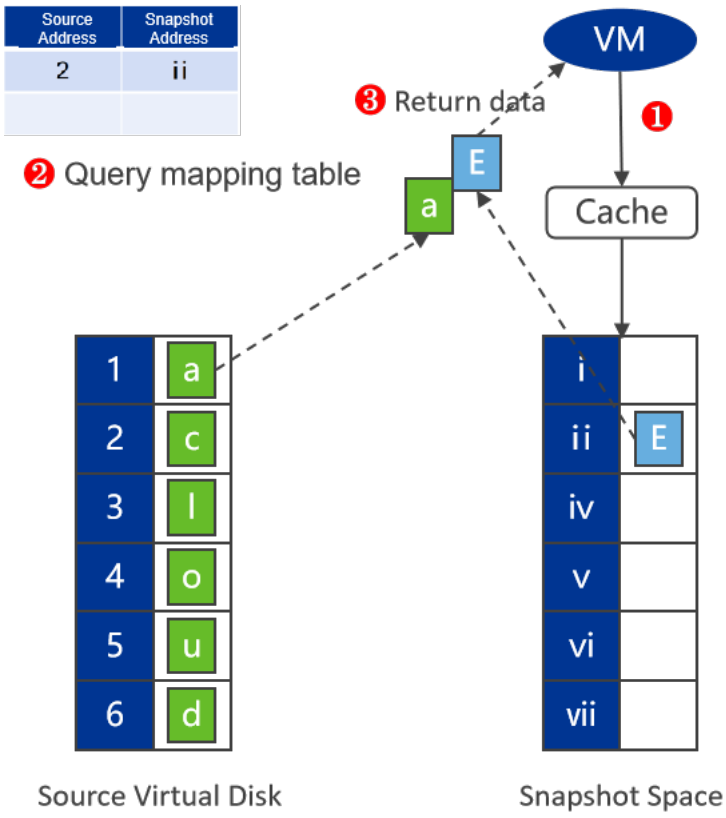


Figure 201: Data Reading After Snapshot Creation

1. The VM issues a data read command.

2. If the VM wants to read the data at source address 1, that is, if the data to be read is written before snapshot creation and address 1 does not exist in the mapping table, the system directly reads data a from the source virtual disk and returns the data.
3. If the VM wants to read the data at source address 2, that is, if the data to be read is written after snapshot creation, the system queries the mapping table, reads data E from the snapshot address ii, and returns the data.

Snapshot-based rollback:

Before the storage-based snapshot creation, data is stored in source virtual disks. After snapshot creation, new business data and modifications to data in the source virtual disks are written to the allocated snapshot space.

When VM data is rolled back based on the snapshot, the data is restored to the status at the time when the snapshot is created, the process of reading data based on the mapping table is canceled, and data is read from and written to the source virtual disks.

5.2.4.1.3 Consistency Group Snapshot

Consistency group snapshot is mainly used in business scenarios where multiple VMs are strongly correlated. The snapshot creation time must be consistent for the VMs. Examples are as follows: an Oracle RAC database consisting of two or more VMs, a distributed application consisting of multiple VMs, and a typical business scenario consisting of an application VM, a middleware VM, and a database VM.

aSAN allows multiple VMs to be added to a consistency group. When a snapshot is created for the consistency group, the snapshot creation time remains consistent for all the VMs in the group, ensuring the business availability for the subsequent snapshot-based rollback and cloning from a snapshot.

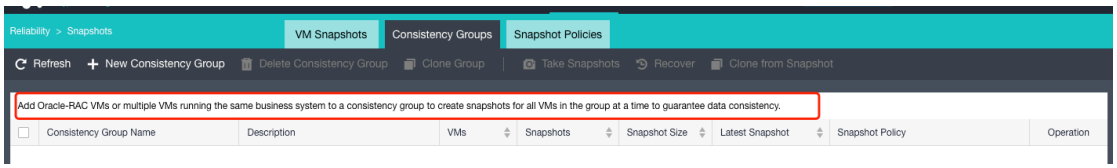


Figure 202: Snapshot Consistency Group

5.2.4.1.4 Scheduled Snapshot Policy

aSAN allows customers to configure scheduled snapshot policies and snapshot retention policies. Expired snapshots will be automatically deleted to release storage space.

With a scheduled snapshot policy, VMs can be protected regularly. In the event of a logical error, such as a virus infection, the VM can be recovered from a snapshot to reduce loss.

Scheduled snapshot-taking and scheduled backup are two ways to protect business data. The following table lists their differences.

	Scheduled Backup	**Scheduled Snapshot**
Initial full data protection	During initial backup, data is backed up fully. The backup data occupies the same size of space as the source VM data occupies.	The source image is set to be read-only. Data does not need to be copied or migrated, and no extra space is occupied.
Incremental data	Only data changes (including addition, modification, and deletion) are backed up.	Data changes (including addition, modification, and deletion) are written to the snapshot space.
Time taken for backup/snapshot creation	The time consists of two parts. One is the time taken to copy all data of the source image during the initial backup, and the other is the time taken to copy incremental data in subsequent backups.	Snapshot creation can be completed in seconds.

Time taken for restore/rollback	After the VM is started in minutes, backup data is copied back to the VM continuously.	The VM data is restored to the state at the snapshot creation time in seconds.
Fault tolerance	The system is tolerant to logical faults (virus infection and data deletion by mistake). If data is backed up to a remote cluster, the system is tolerant to physical faults in the local cluster. If data is backed up to the local cluster, the system is not tolerant of physical faults.	The system is tolerant to logical faults (virus infection and data deletion by mistake). The system is not tolerant of physical faults in the local cluster.
Impact on performance	During the backup, performance is affected to a certain extent. After backup is completed, performance is not affected.	During snapshot creation, performance is not affected. After snapshot creation is completed, performance is slightly reduced and then gradually recovered.

Table 11: Comparison Between Scheduled Backup and Scheduled Snapshot

5.2.4.2 Clone

aSAN supports linked clone, instant full clone, and full clone to meet the differentiated needs for storage space, speed, and performance in different businesses.

5.2.4.2.1 Full Clone

During a full clone, data of the source VM is fully copied. The clone VM can be started only after the data copying is completed.

Characteristics of full clone:

1. The clone VM can be started only after the data is fully migrated.
2. The data on the clone VM is always independent from that on the source VM.
3. The performance of the clone VM is not affected.

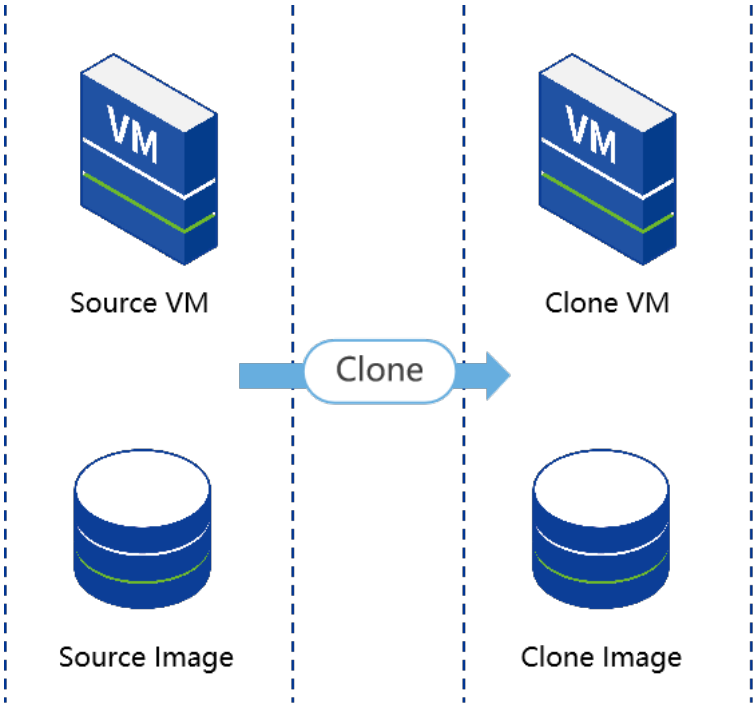


Figure 203: Full Clone

5.2.4.2.2 Linked Clone

It is implemented based on a VM snapshot. The implementation principles are as follows:

1. A storage-based snapshot is created for the source VM, and the source image is set to be read-only.
2. When the clone VM is running, old data is read from the source image, and new or modified data is written to the snapshot space of the clone image.

The clone VM references the source image files of the source VM. Therefore, the performance of the clone VM created in lined clone mode is lower than that created in full clone mode. It is recommended that linked clones be used in development, testing, and other scenarios that do not require high VM performance.

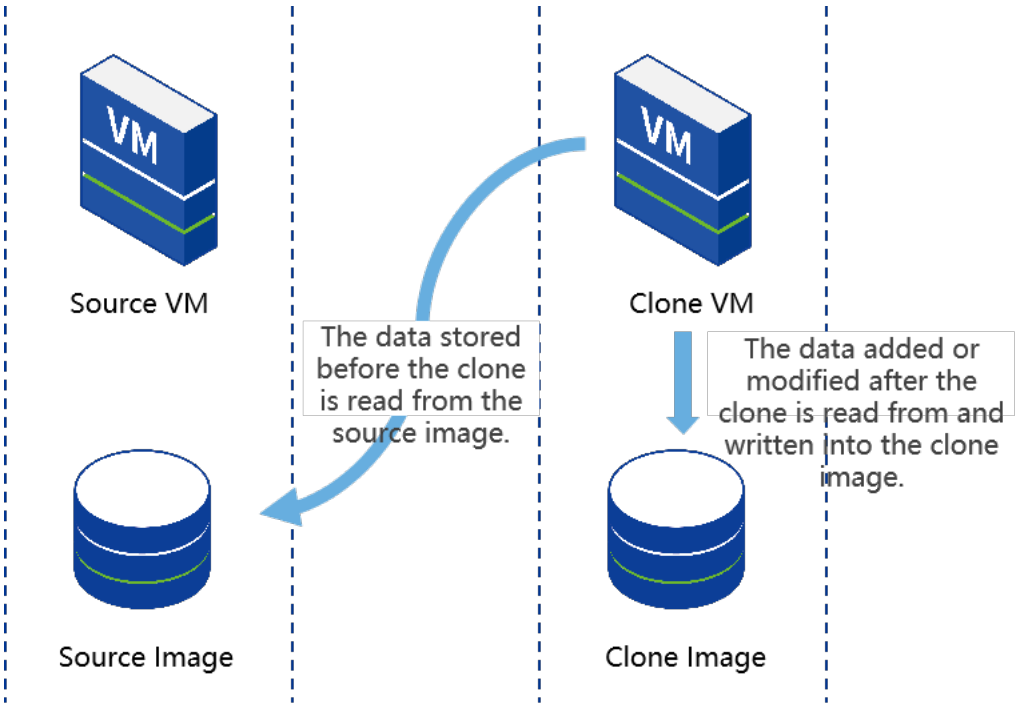


Figure 204: Linked Clone

Characteristics of linked clone:

1. Clone VMs can be started in seconds.
2. Clone VMs occupy a small size of space, saving storage space.
3. Clone VMs reference the source image files of the source VMs, making the performance of the clone VMs deteriorate.

5.2.4.2.3 Instant Full Clone

It is a clone type developed by Sangfor for aSAN by integrating the advantages of full clone and linked clone.

1. As this clone type is implemented based on the linked clone, the clone VM can be started in seconds at the early stage of the clone.
2. During the clone, the HCI backend generates a full clone job, which continuously copies all data from the source image to the image of the clone VM.

3. After the clone is completed, the clone VM is disconnected from the source VM so that data on the clone VM is independent from that on the source VM.

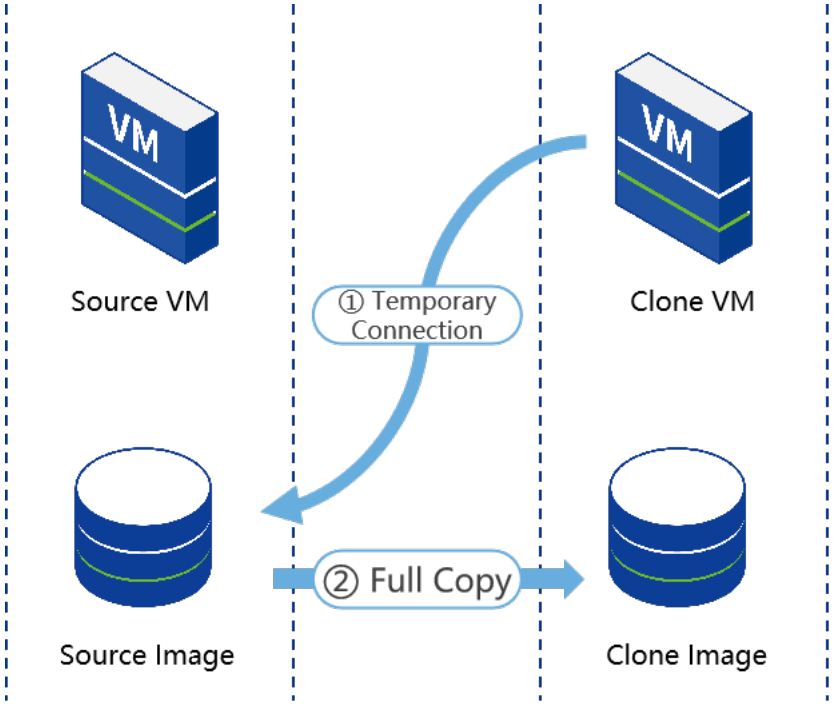


Figure 205: Instant Full Clone

Characteristics of instant full clone:

1. Clone VMs can be started in seconds.
2. The data on the clone VM is always independent from that on the source VM.
3. After the clone is completed, the performance of the clone VM is not affected.

Therefore, the instant full clone is recommended in aSAN by default.

5.2.4.2.4 Comparison Among Three Clone Types

Item	Full Clone	Linked Clone	Instant Full Clone
Clone speed	It depends on the data amount on the source	Only several seconds are taken.	Only several seconds are taken.

	VM. The clone VM can be started only after the data copying is completed.		
Impact on performance	After the clone is completed, performance is not affected.	After the clone is completed, performance is still affected.	After the clone is completed, performance is not affected.
Space usage	After data copying is completed, the space occupied by the clone VM is the same as that occupied by the source VM.	The space is occupied only when data changes are written to the clone VM.	After data copying is completed, the space occupied by the clone VM is similar to that occupied by the source VM.
Restrictions	Storage locations are unlimited for the source and clone VMs.	Both the source and clone VMs need to store data in virtual storage.	Both the source and clone VMs need to store data in virtual storage.

Table 12: Comparison Among Three Clone Types

5.2.4.3 Multi-Replica Data

Multi-replica is a redundancy technology to save multiple replicas of data. With the multi-replica mechanism, the business will not be interrupted due to inaccessible data when a single replica is damaged. The replica copying module of aSAN ensures data consistency and synchronization among replicas. Any data differences between replicas will cause the upper-layer applications to access different data, affecting the business processing results. Replicas are transparent to the upper-layer applications, and therefore, the applications are unaware of the existence of replicas.

5.2.4.3.1 Strong Data Consistency Among Replicas

aSAN uses a strong consistency protocol to ensure data consistency among replicas. A response indicating the completion of the I/O operation is returned to the upper layer only when the data is written to all replicas. If replicas are running properly, data is consistent among them. In this case, when receiving a read request, aSAN reads data locally in priority. If the host where a replica resides encounters an error (such as go-offline, host crash, and network error), aSAN stops initiating read or write requests to the host. Instead, it sends them to a healthy host for processing. When the faulty replica goes online after a short period of interruption, aSAN initiates a request to restore it based on a healthy replica. If the replica remains offline for a long time, it will be rebuilt evenly on another host or disk through the distribution algorithm.

5.2.4.3.2 Replica Distribution Principles

aSAN supports the storage of two or three replicas. The locations to store the data replicas must satisfy the principle of host anti-affinity. In other words, multiple data replicas of the same data file cannot be stored on the same host. This is to ensure that when a host fails, the replicas on other hosts are available.

The principle of host anti-affinity must be followed in data replica distribution and rebuilding. The following figure illustrates the distribution of three replicas on five hosts.

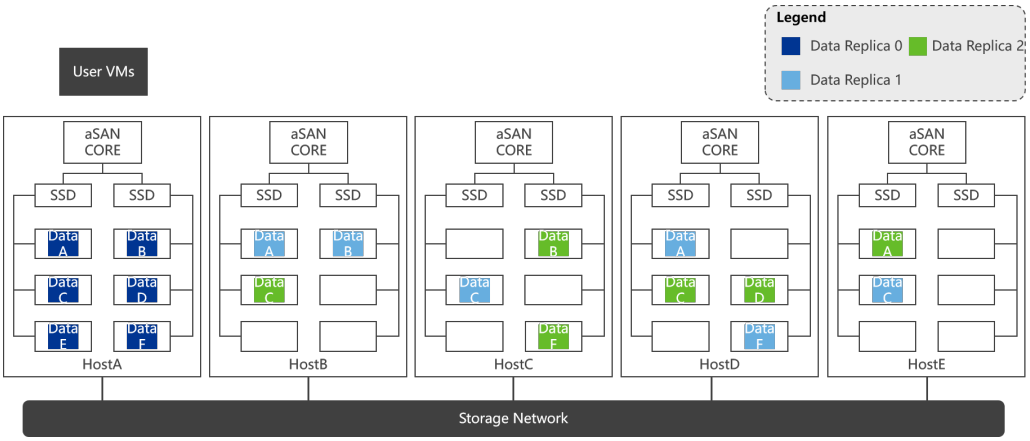


Figure 206: Data Distribution of Three Replicas on Five Hosts (1)

- **Host anti-affinity for data replicas**: For each data file, three replicas are stored on separate hosts. As shown in the figure above, the three replicas marked as Data A are distributed on hosts A, B and E.
- **Replica aggregation and distribution**: One data replica is aggregated on the local host where the VM runs, and the other replicas are distributed on other hosts. As shown in the figure above, host A stores a complete replica of the data. Therefore, the VM will preferentially run on host A.

5.2.4.3.3 Witness Replica Distribution Principles

aSAN provides the witness replica mechanism to ensure higher data reliability. The witness replica stores only attribute information and occupies little space.

- For data with two replicas, one witness replica is available.
- For data with three replicas, two witness replicas are available.

5.2.4.3.4 Data Replica Distribution Principles for Scenarios Where Witness Replicas Exist

Scenario 1: Number of hosts ≥ Number of data replicas + Number of witness replicas

1. Data replicas of the same data file must be stored on separate hosts.
2. Witness replicas of the same data file must be stored on separate hosts.
3. Data replicas and witness replicas of the same data file must be stored on separate hosts.

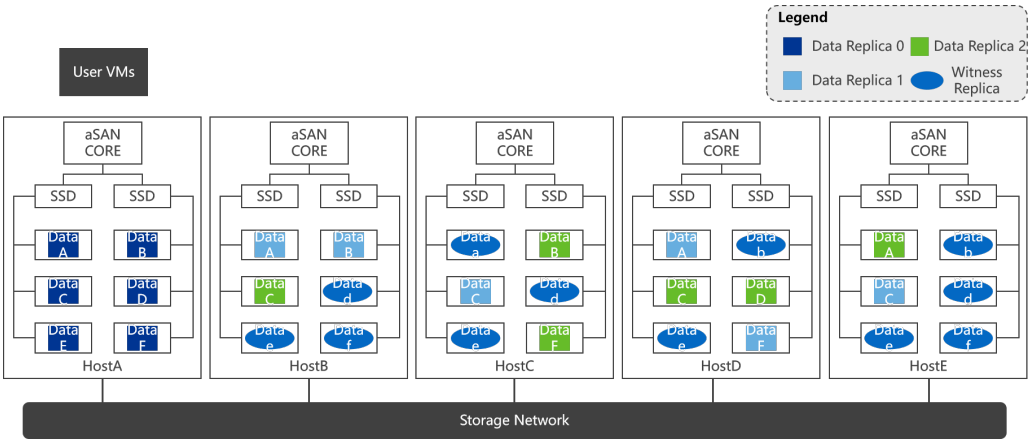


Figure 207: Data Distribution of Three Replicas on Five Hosts (2)

In this scenario, replicas must be distributed based on the following principles:

1. Host anti-affinity for data replicas: Three data replicas marked as Data A are distributed on hosts A, B and E.
2. Host anti-affinity for witness replicas: Two witness replicas marked as Data a are distributed on hosts C and D.
3. Host anti-affinity for data replicas and witness replicas: Data replicas marked as Data A are distributed on hosts A, B and E, while witness replicas marked as Data a are distributed on hosts C and D.

Scenario 2: Number of hosts < Number of data replicas + Number of witness replicas (three replicas distributed on three or four nodes of a cluster)

1. Data replicas of the same data file must be stored on separate hosts.
2. Witness replicas of the same data file must be stored on separate hosts.
3. Data replicas and witness replicas of the same data file must be stored in separate disk groups. However, the disk groups can belong to the same host.

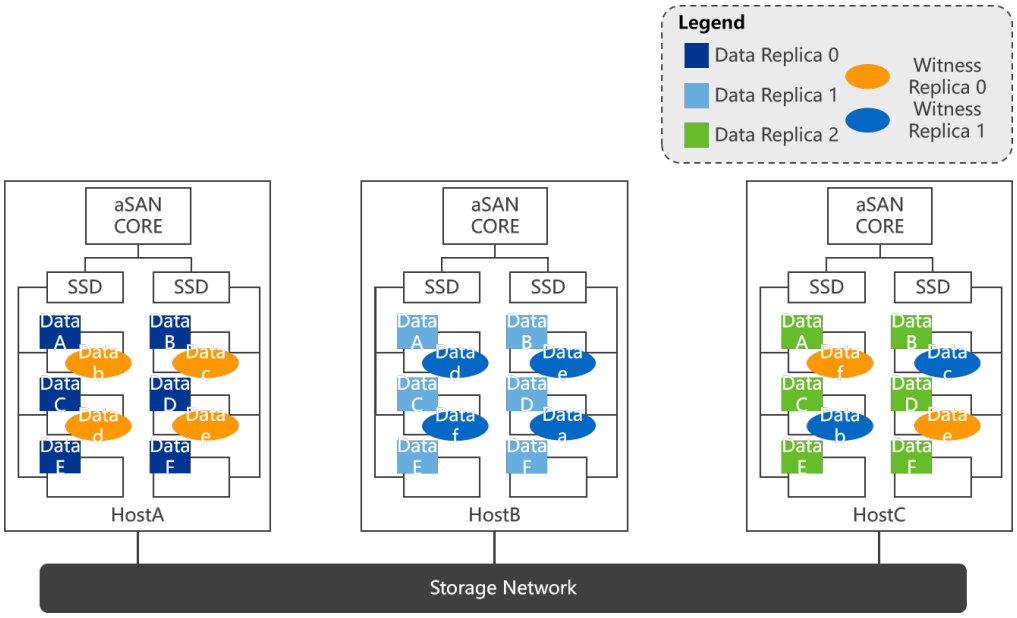


Figure 208: Data Distribution of Three Replicas on Three Hosts

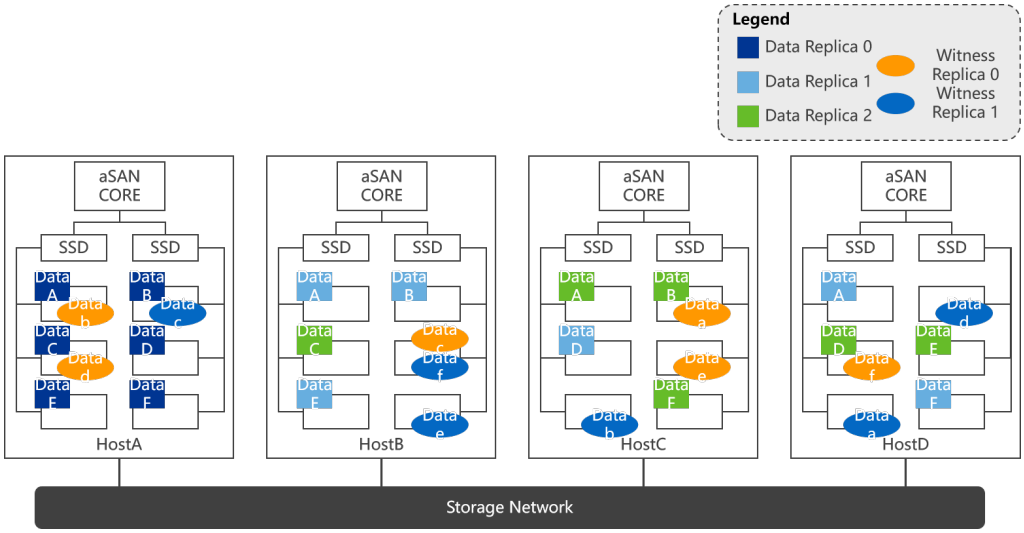


Figure 209: Data Distribution of Three Replicas on Four Hosts

In this scenario (for example, when three replicas are distributed on three hosts), replicas must be distributed based on the following principles:

1. Host anti-affinity for data replicas: Three data replicas marked as Data A are distributed on hosts A, B and C.
2. Host anti-affinity for witness replicas: Two witness replicas marked as Data a are distributed on hosts C and D.

3. Disk group anti-affinity for data replicas and witness replicas (disk groups allowed to belong to the same host): As shown in the above figure (Data Distribution of Three Replicas on Three Hosts), the replica marked as "Data A" and that marked as "Data a" belong to different disk groups on host A.

5.2.4.3.5 Fault Tolerance

The data distribution principle varies depending on the number of replicas and hosts, resulting in different fault tolerance capabilities.

1. The three-replica topology offers superior fault tolerance than the two-replica one.
2. A topology of five or more hosts with three replicas exhibits better fault tolerance than a topology of three or four hosts with three replicas.

Host Count	**Fault Tolerance of Three Replicas**	**Fault Tolerance of Two Replicas**
Three or four hosts	Up to one failed host is allowed. or Up to two failed disks anywhere in the system are allowed.	Up to one failed host is allowed. or Up to one failed disk is allowed.
≥ Five hosts	Up to two failed hosts are allowed. or Up to two failed disks anywhere in the system are allowed.	

Table 13: Fault Tolerance of Different Multi-Replica Topologies

Judgment criterion of fault tolerance: After a disk or host failure occurs in the cluster, the unaffected data replicas of a piece of data can work as long as their number exceeds half of the total replica count.

Fault Tolerance of Two Replicas:

1. A two-replica topology can tolerate up to one host failure or one disk failure.
2. A two-replica topology consists of two data replicas and one witness replica. For example, in the following figure, the three replicas, including two data replicas (marked as "Data A") and one witness replica (marked as "Data a"), are distributed across hosts A, B, and C strictly according to the principle of host anti-affinity.

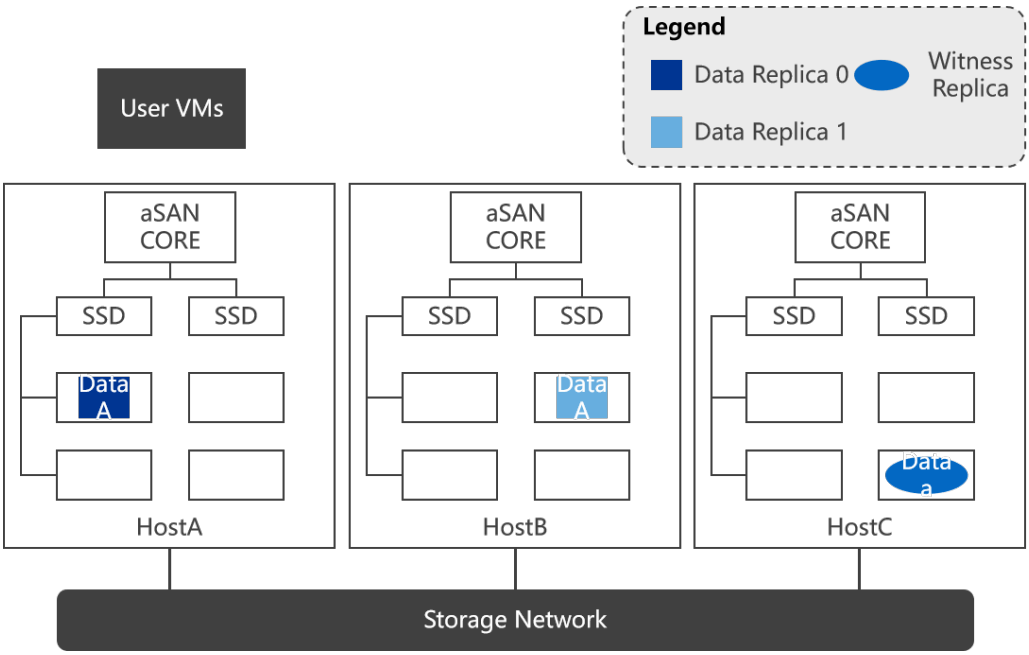


Figure 210: Fault Tolerance of Two-Replica Topology

- Scenario 1: Up to one failed host is allowed.

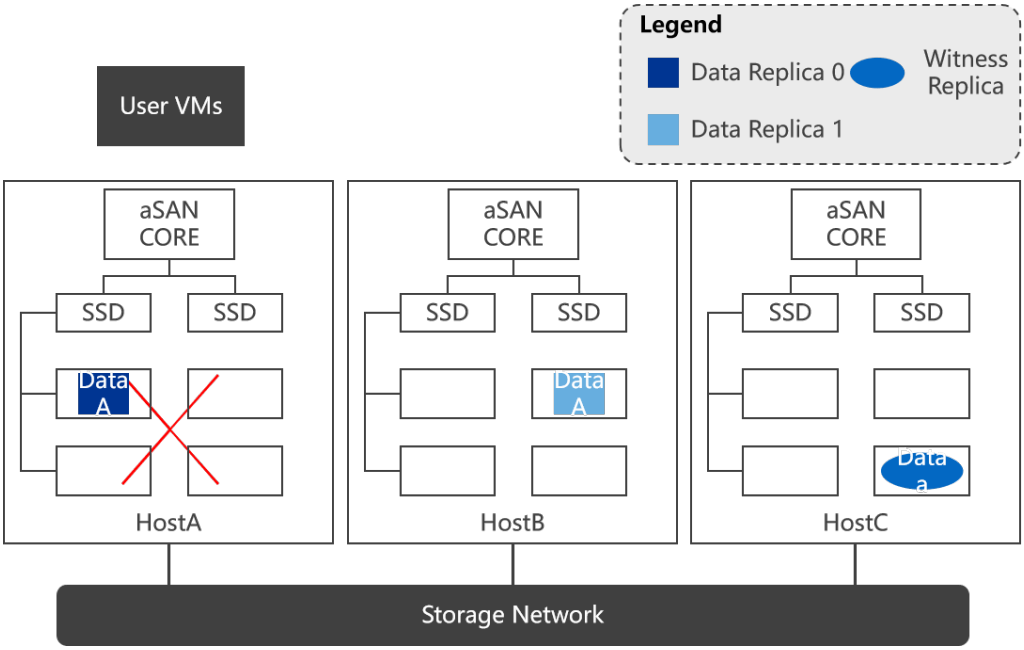


Figure 211: Host Failure in a Two-Replica Topology

As shown in the diagram, host A experiences a failure, and the data replica marked as "Data A" is lost. At this time, according to the judgment criterion of fault tolerance, the number of accessible replicas marked as "Data A" or "Data a" in the cluster is two (one data replica on host B and one arbiter replica on host C), which exceeds half of the total (three). Therefore, the residual data replica remains usable.

If another host fails, the number of accessible Data A/a replicas in the cluster drops to one, a value below half of the total (three). In this case, the residual data replica becomes unusable.

- Scenario 2: Up to one failed disk is allowed across different hosts.

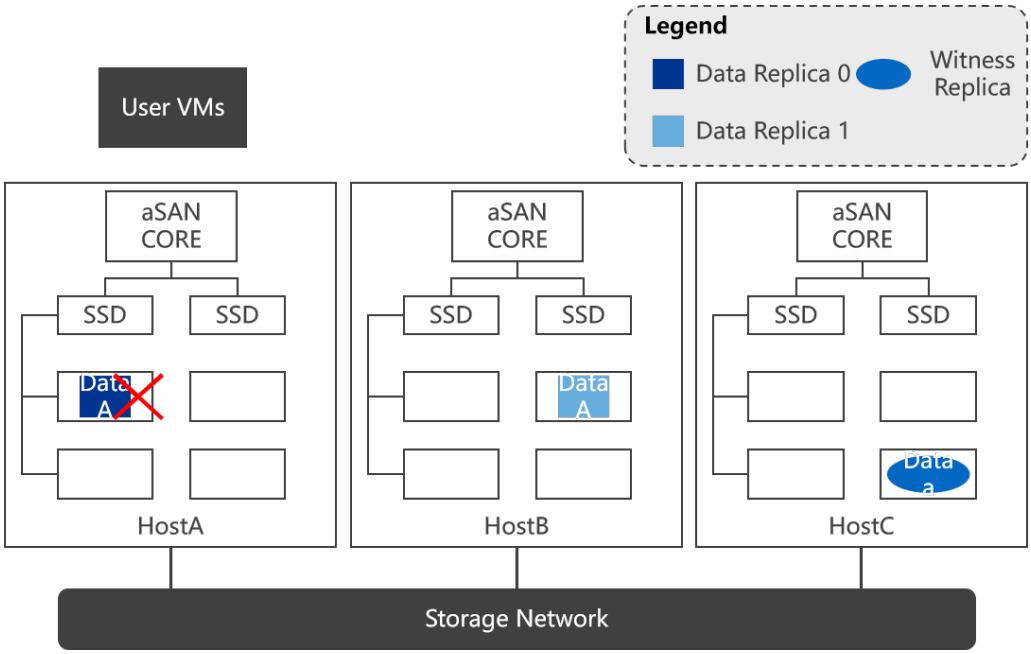


Figure 212: Disk Failure in a Two-Replica Topology

The diagram illustrates a disk failure resulting in the loss of the data replica (marked as "Data A"). According to the judgment criterion of fault tolerance, the number of accessible replicas marked as "Data A" or "Data a" in the cluster is two, which exceeds half of the total (three). Therefore, the residual data replica remains usable.

If another disk fails (for example, the disk of host C containing the "Data a" replica fails), the number of accessible replicas in the cluster drops to one, a value below half of the total (three). In this case, the residual data replica becomes unusable.

NOTICE

For ease of understanding, the above figure only provides an example of one shard. However, in reality, there are numerous data shards on each disk with the same distribution principle. When a single disk storing a shard of data fails, but the subsequent disk failure does not affect the disk storing the corresponding data replica or witness replica, the data replicas marked as "Data A" or "Data a" remain usable, but other shards may not.

Fault Tolerance of Three Replicas Across Three/Four Hosts:

1. Up to two failed disks are allowed at the same time.

2. Up to one failed host is allowed.

The figure below illustrates a topology of three hosts with three replicas:

1. The three data replicas marked as "Data A" are distributed on hosts A, B, and C following the principle of host anti-affinity.
2. The two witness replicas marked as "Data a" are distributed on hosts A and B following the principle of host anti-affinity.
3. On hosts A and B, the data replica "Data A" and witness replica "Data a" are distributed on different disk groups following the principle of disk group anti-affinity.

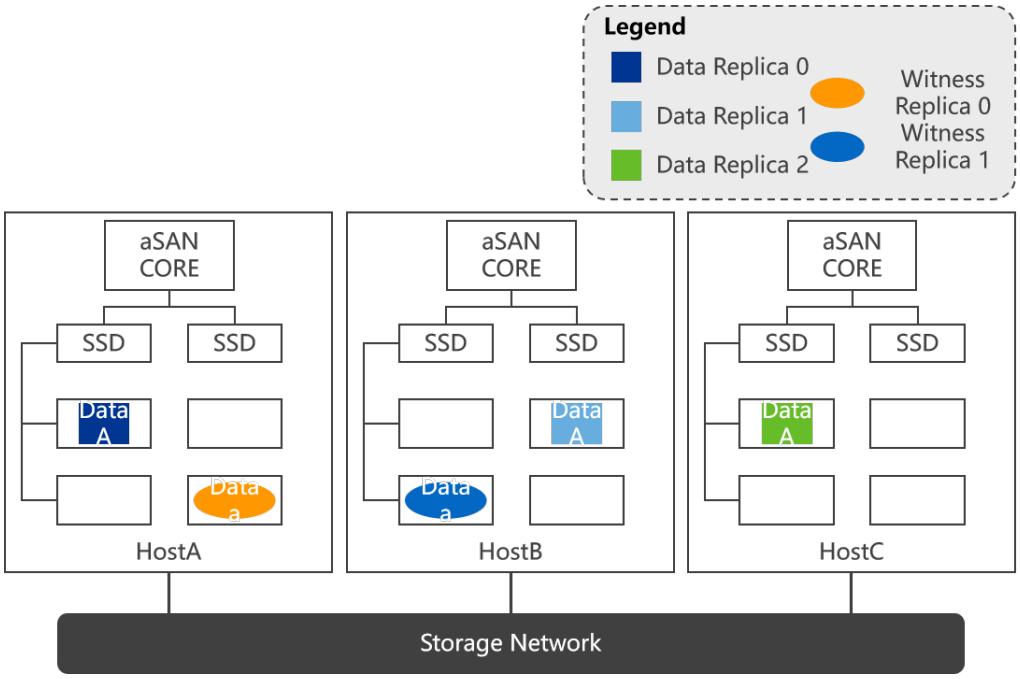


Figure 213: Data Distribution of Three Replicas on Three Hosts

- Scenario 1: Up to one failed host is allowed.

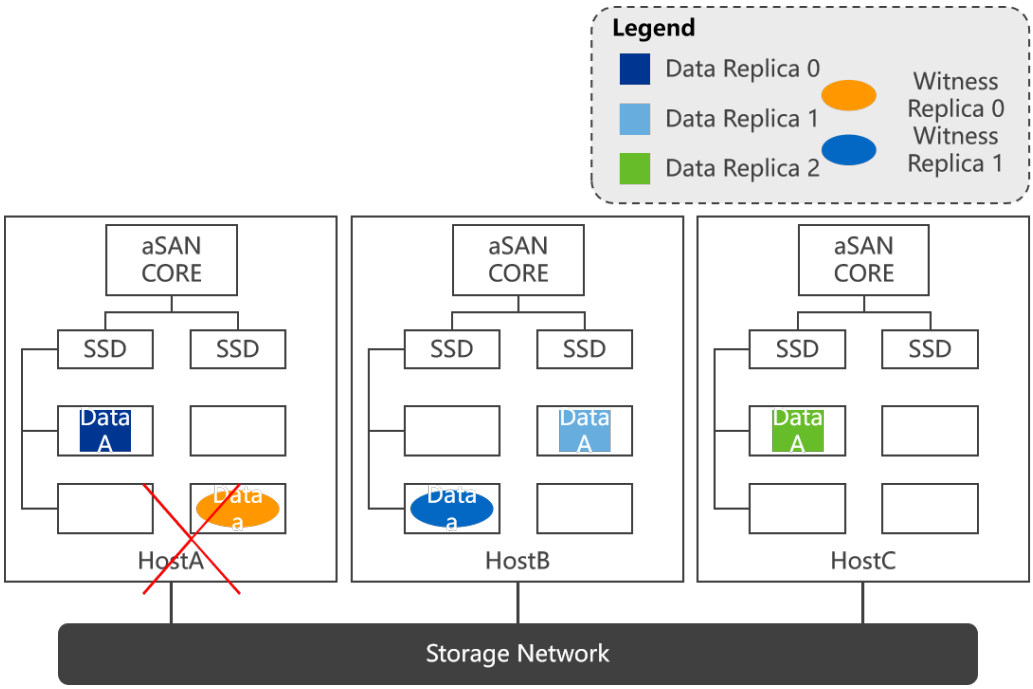


Figure 214: Host Failure in a Topology of Three Hosts with Three Replicas

When host A fails, the data replica and the witness replica stored on host A are lost. At this point, the quorum mechanism determines that there are three accessible replicas of "Data A/a" in the cluster (two on host B and one on host C), which is greater than half of the total number of replicas, five. Therefore, the remaining data replicas are still usable.

However, if another host fails (for example, host C goes down, and the data replica stored on it is lost), the number of accessible replicas in the cluster drops to two (both on host B), a value below half of the total (five). In this case, the remaining data replicas become unusable.

- Scenario 2: Up to two failed disks are allowed across different hosts.

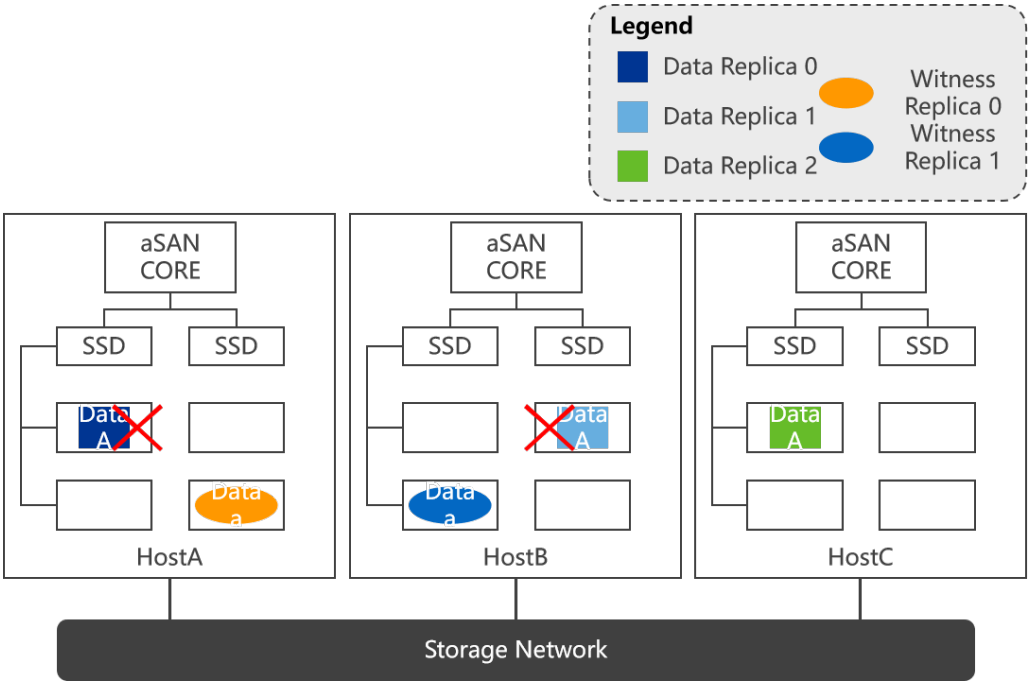


Figure 215: Two Failed Disks in a Topology of Three Hosts with Three Replicas

As shown in the figure, when there is a disk failure on both host A and host B, which leads to the loss of two "Data A" replicas, three "Data A/a" replicas in the cluster remain accessible, exceeding half of the total (five). In this case, the residual data replicas are still usable.

However, if another disk fails (for example, the disk of host C containing the "Data A" replica fails), only two "Data a" replicas in the cluster remain accessible, a value below half of the total (five). In this case, the residual data replicas become unusable.

Fault Tolerance of Three Replicas Across Five or More Hosts:

In a topology of five or more hosts, the three-replica deployment exhibits the highest fault tolerance:

1. Up to two failed hosts are allowed at the same time.
2. Up to two failed disks in the cluster are allowed at the same time.
3. When the number of hosts is five or more, as mentioned earlier, there are actually three data replicas and two arbiter replicas existing for a piece of data.

The figure below illustrates the data distribution in a three-replica deployment:

Three data replicas (marked as "Data A") and two witness replicas (marked as "Data a") are distributed across five hosts in strict accordance with the principle of host anti-affinity.

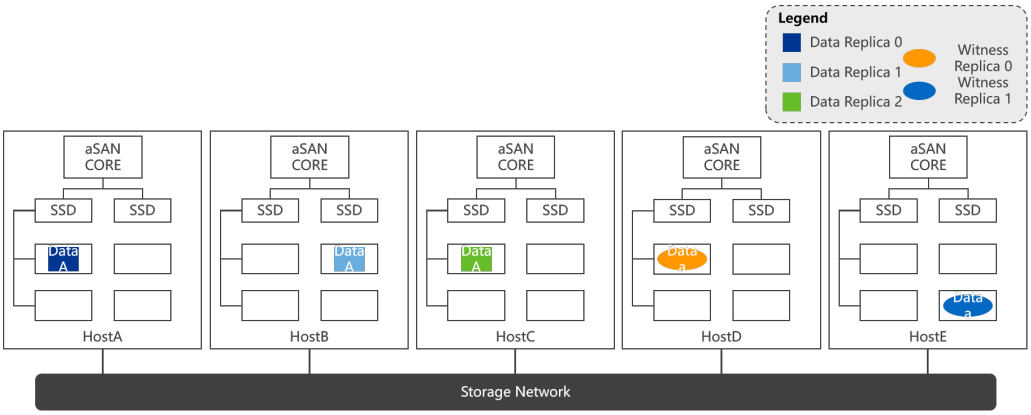


Figure 216: Data Distribution of Three Replicas on Five Hosts

- Scenario 1: Up to two failed hosts in the cluster are allowed.

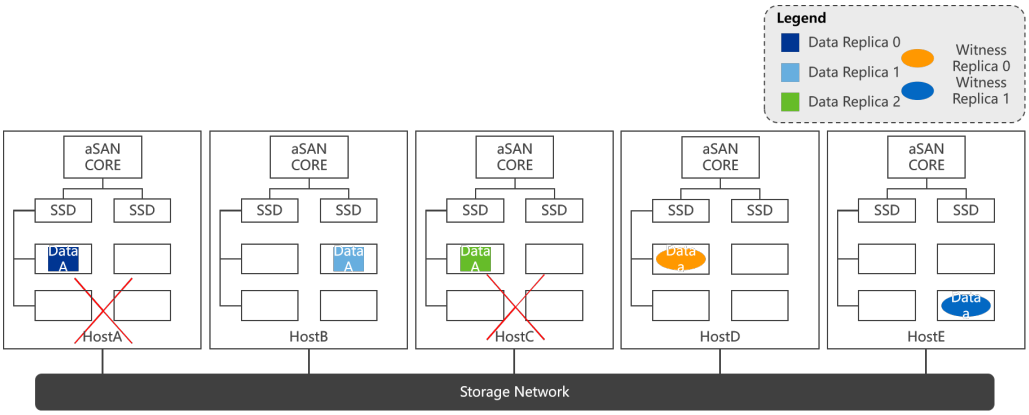


Figure 217: Two Failed Hosts in a Topology of Five Hosts with Three Replicas

When hosts A and C both fail, the quorum mechanism determines that there are three accessible replicas in the cluster, exceeding half of the total number of replicas (five, three data replicas and two arbiter replicas). Therefore, the remaining data replicas are still usable.

However, if another host fails, the number of accessible replicas in the cluster drops to two, a value below half of the total (five). In this case, the residual data replicas become unusable.

- Scenario 2: Up to two failed disks in the cluster are allowed.

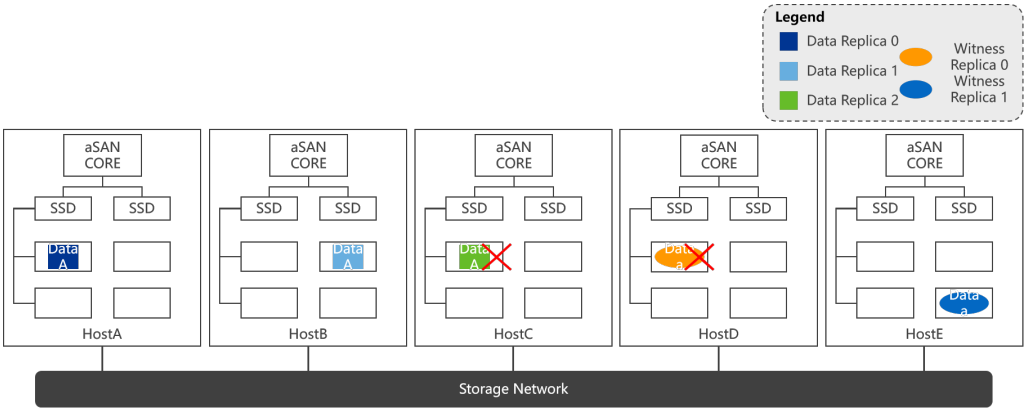


Figure 218: Two Failed Disks in a Topology of Five Hosts with Three Replicas

When both the disks storing the "Data A/a" replicas on host B and host D fail at the same time, there are still three replicas of "Data A/a" in the cluster, exceeding half of the total number of replicas (five). Therefore, the remaining data replicas are still usable.

However, if any disk containing a "Data A/a" replica fails at this point, the number of replicas drops to two, which is less than half of the total. In this case, the residual data replicas become unusable.

5.2.4.4 Quorum Mechanism

VM replicas, as described in the Replica Mechanism section, can be divided into data replicas and witness replicas. This section explains the quorum mechanism, that is, the working principle of a witness replica.

5.2.4.4.1 How Does Split-Brain Occur?

When multiple replicas of data are isolated due to a network failure, they cannot synchronize information with each other and different new data is separately written to each replica, resulting in data inconsistency, as shown in the following figure. This situation is called split-brain.

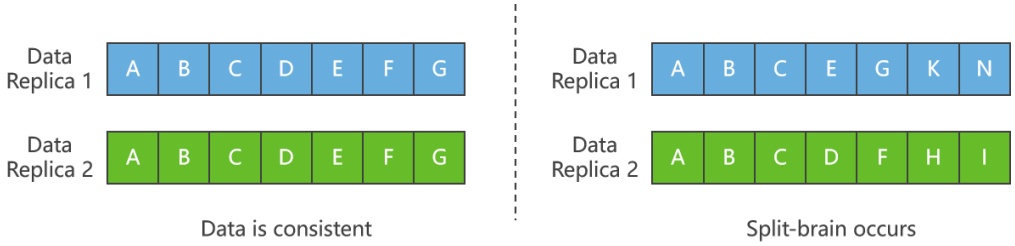


Figure 219: Illustration of Split-Brain

To explain how split-brain occurs, let's consider a scenario with two data replicas but no witness replica:

1. Under normal circumstances, the VM runs on host A, and data is simultaneously written to the two replicas on host A and host B. The data in both replicas is identical.

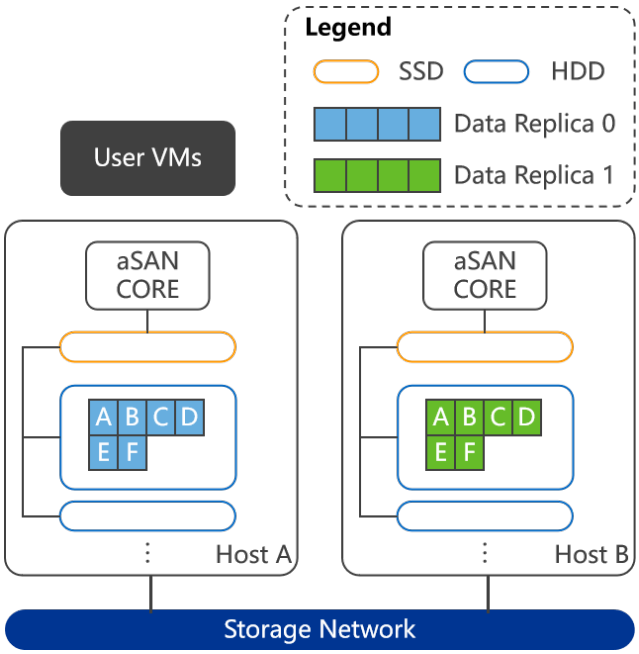


Figure 220: Consistent Data Replicas Under Normal Circumstances

2. When the storage network connection between host A and host B is interrupted, the two hosts become isolated from each other, unable to synchronize data properly. At this point, the VM can continue running on host A, but its newly generated data can only be written to the disks on host A. The data on host B remains outdated, resulting in inconsistency

between the two replicas. However, split-brain has not occurred at this stage.

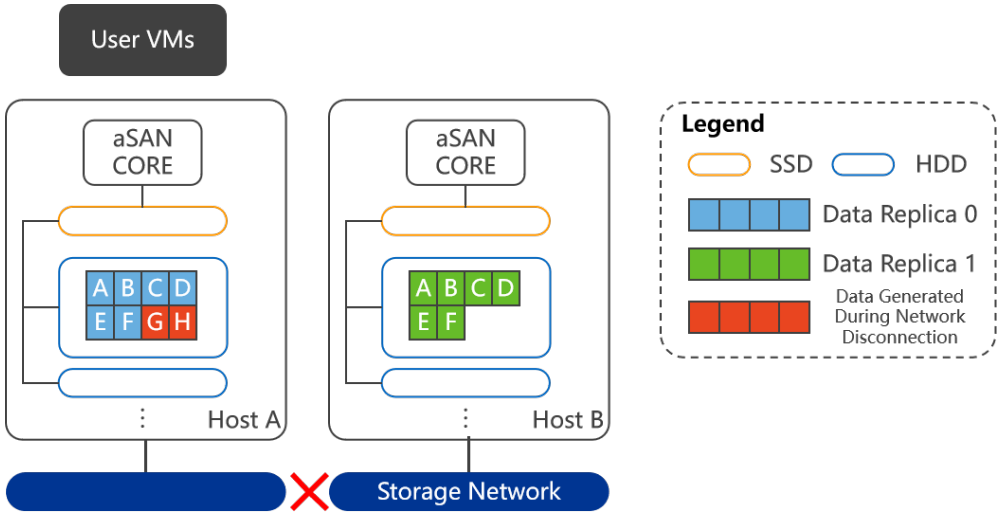


Figure 221: Inconsistent Data Replicas Under Exceptional Conditions Where New Data Is Written Only to Host A

3. Host B fails to receive heartbeat messages from and communicate with host A, considering host A as faulty. At this point, the HA mechanism of the VM takes effect, and the VM is launched on host B. Consequently, the VM runs on both host A and host B, leading to a "dual instance" scenario. Each VM writes new data to the host where it runs. This results in data inconsistency between the two hosts, where each replica holds data that is not present on the other side. This is the situation known as split-brain.

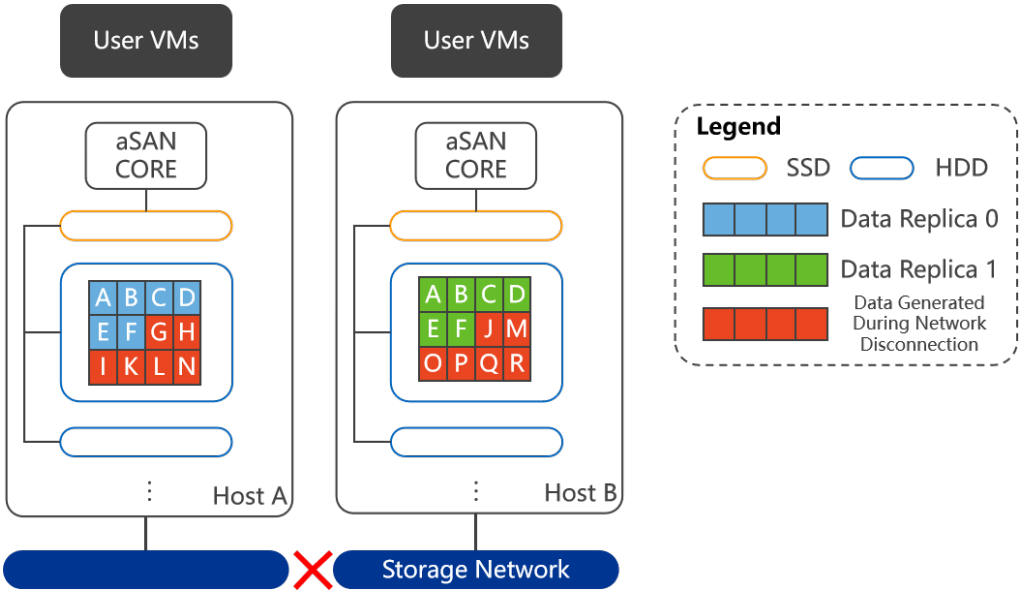


Figure 222: Inconsistent Data Replicas Under Exceptional Conditions Where Different New Data Is Written to Host A and Host B

After split-brain occurs, the data in any replica is actually incomplete and incorrect. When the storage network connection is restored, the system detects the inconsistency between the two data replicas and triggers data repair to restore consistency. In this case, regardless of which replica is used as the data source for repair, partial data loss will occur.

5.2.4.4.2 How Does a Witness Replica Work?

aSAN introduces witness replicas to prevent split-brain. Witness replicas are special replicas that contain only a small amount of verification data and occupy minimal storage space.

The core principle of the quorum mechanism is "minority follows the majority." When the number of data replicas accessible to the VM running on the host where the defrag replica is located is less than half of the total number of replicas (data replicas + witness replicas), the VM is prohibited from running on that host. Otherwise, the VM can run on that host.

To illustrate how the quorum mechanism prevents split-brain situations, consider a scenario with two data replicas and one witness replica:

1. Under normal circumstances, the VM runs on host A, and the data is simultaneously written to hosts A, B, and C. The data replicas are on hosts A and B, while the witness replica is on host C.

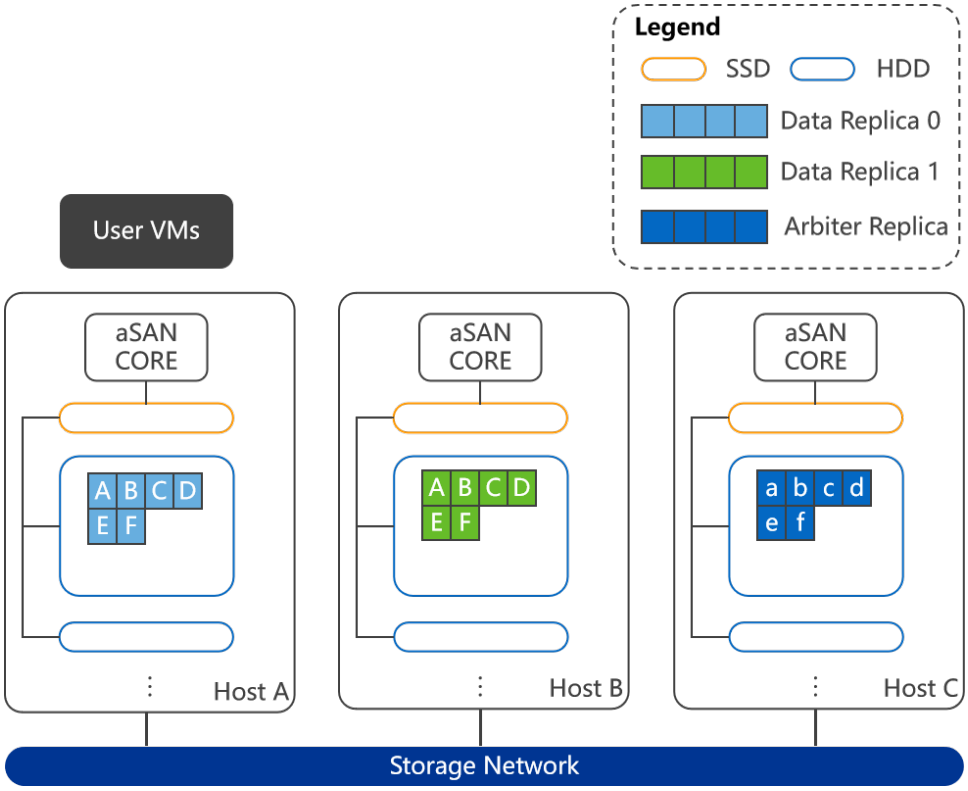


Figure 223: Consistent Data Replicas Under Normal Circumstances

2. When host A is disconnected from the storage network, the host cannot communicate with hosts B and C. The quorum mechanism determines that the VM running on host A can only access a single local data replica. This quantity is less than half of three, the total number of replicas (two data replicas + one witness replica), so the VM process on host A is killed.

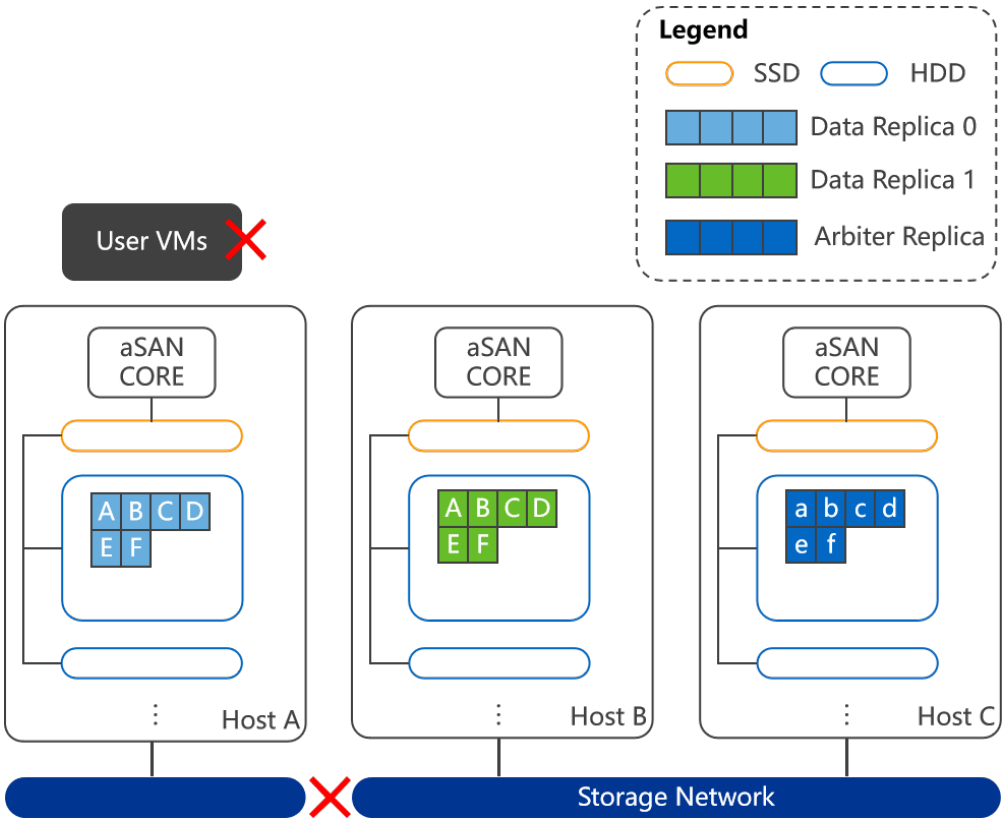


Figure 224: Consistent Data Replicas Despite Exceptional Conditions Where VM Process Is Killed on Host A

3. The quorum mechanism determines that both host B and host C have access to two replicas (the data replica on host B and the witness replica on host C), a quantity that exceeds half of the total number of replicas. Consequently, HA is triggered to start the VM on either host B or host C. The following figure demonstrates that the VM is started on host B. When the VM is running, data can be written normally to the replicas on hosts B and C.

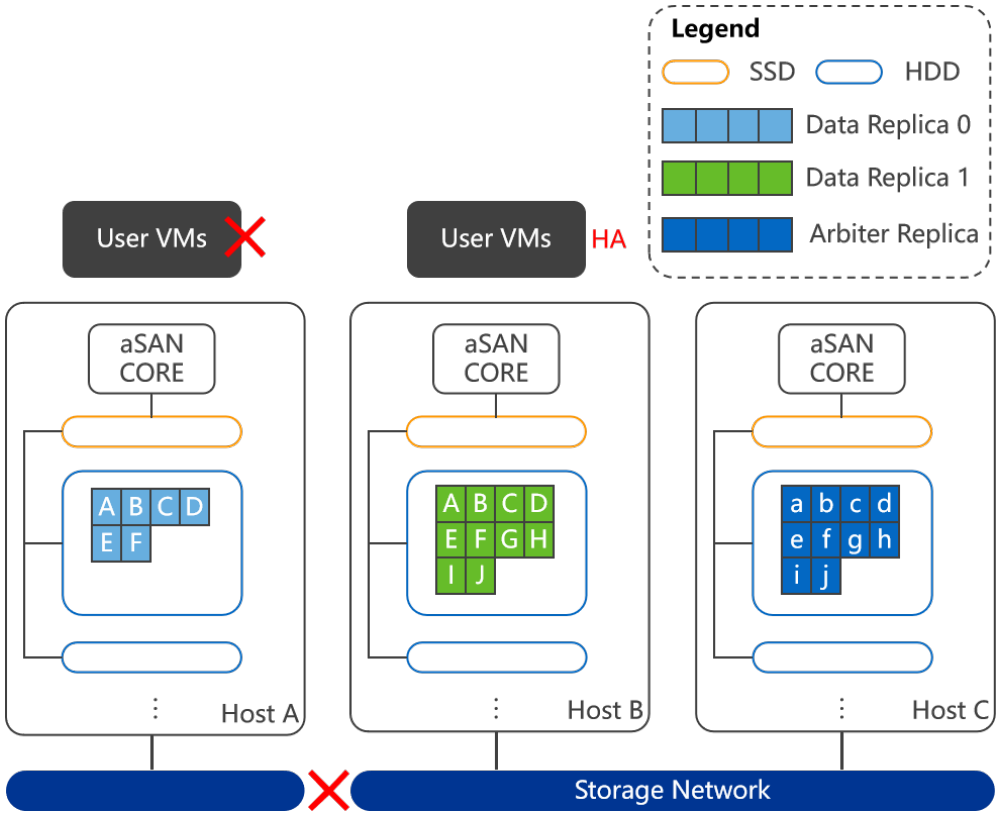


Figure 225: Consistent Data Replicas Despite Exceptional Conditions When VM is Started Based on HA

4. When host A reconnects to the storage network, the system detects data replica inconsistencies between host A and host B and triggers data repair to restore consistency by using the replica on host B as the data source.

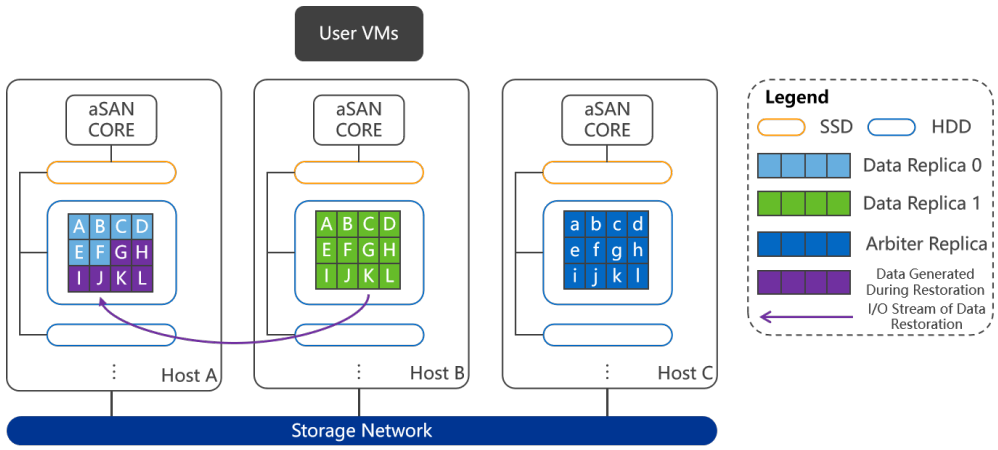


Figure 226: Recovered from Exceptions, Data Synchronized, and Data Replica Count Restored

A virtual datastore composed of two hosts lacks a quorum mechanism, posing risks of data split-brain and potential data loss. To mitigate such risks, aSAN recommends a direct connection between the storage interfaces of the two hosts, bypassing switches to avoid any impact brought by switch failures. Additionally, aSAN implements network leverage in the dual-host scenario, allowing the use of management or data networks as the communication network between host storage when the storage network is completely interrupted. This significantly enhances the reliability of the storage network and greatly reduces the risk of split-brain.



After you add a host to the two-host virtual datastore, the quorum mechanism activates and generates a witness replica. During this process, the VMs on the virtual datastore must be shut down. Therefore, you must shut down all VMs in the cluster before expanding the two-host virtual datastore.

5.2.4.5 Silent Error Detection

During disk usage, unpredictable errors called silent errors may occur. These errors go unnoticed until users need to use the data, at which point they discover that the data has been corrupted and damaged, resulting in irreparable loss. Silent errors have no warning signs and may have lasted a long time, leading to serious issues. NetApp observed over 1.5 million hard disk drives (HDDs) for 41 months and discovered more than 400,000 silent data corruptions, out of which more than 30,000 were not detected by the hardware RAID controller.

To prevent returning incorrect data to users due to silent errors, HCI aSAN supports end-to-end data verification. This is achieved through a Checksum engine, Verify engine, and Checksum management module. A state-of-the-art checksum algorithm is adopted, along with key techniques for checksum storage performance optimization. A checksum is generated for each data block and stored as its fingerprint as soon as the data block enters the system. The checksum is then used for data verification, protecting user data from

silent errors.

The following figure shows the principle.

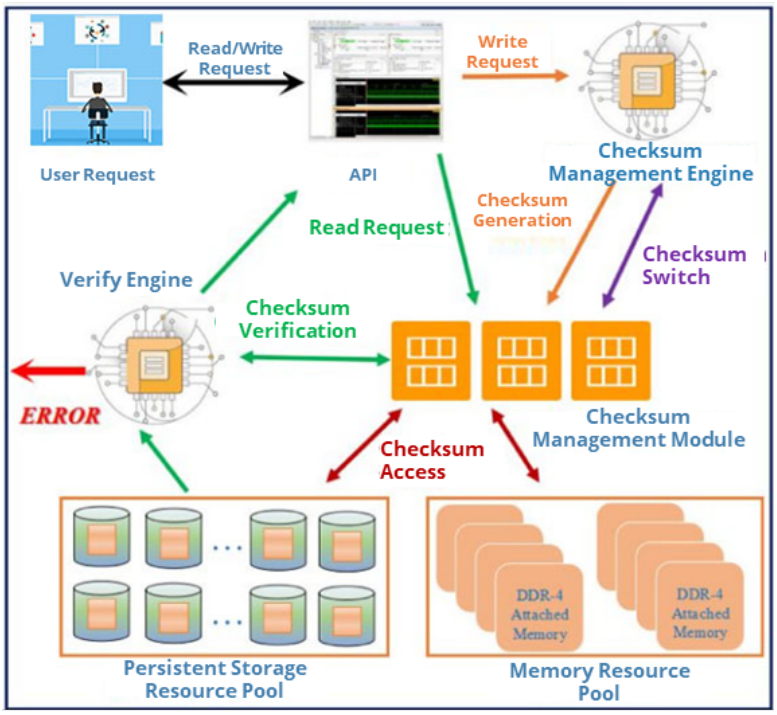


Figure 227: Silent Error Detection Principle

There are two key techniques to implement end-to-end verification: checksum generation algorithm and storage performance optimization during checksum generation. aSAN has industry-leading technical solutions in these two aspects.

1. State-of-the-art checksum algorithm

Checksum algorithms are mainly evaluated from two viewpoints: the speed of generating checksums and the collision probability and uniformity. Collision probability indicates the probability of generating the same checksum for two different inputs.

The data end-to-end verification solution provided by Sangfor HCI aSAN uses the hash64 algorithm, which is faster and has a lower collision probability compared to the widely used CRC-32 and Adler-32 algorithms in the industry.

2. Storage performance optimization during checksum generation

Checksums are generated in memory and can be transmitted and stored with

the data. When the data is stored in non-volatile storage such as disks, the checksums also need to be stored to avoid additional writes and impact on system performance.

Based on a metadata-free centralized architecture, Sangfor HCI fixes the performance issues caused by checksum storage in the aSAN end-to-end verification solution by taking such measures as asynchronous writeback, critical I/O path bypassing, and I/O contention isolation. Additionally, self-verification, collision detection, and time series analysis are implemented to ensure data correctness and consistency.

5.2.4.6 Data Rebuilding

When there is a physical failure in the hardware (disk or host) within a virtual datastore, there are still one or two replicas of the data on other hardware to ensure normal read and write operations for the VM. Nonetheless, the redundancy of the virtual datastore becomes lower at this point. In the event of a subsequent failure in the hardware where a replica is located, the risk of data loss increases, and the system reliability decreases.

After aSAN detects that a disk or host is offline, it reserves time for users to confirm the fault. For example, users can check whether the disk is loose and whether the host is temporarily offline due to reasons like power loss and network failure. If the disk or host goes back online within the preset rebuilding waiting period, data rebuilding will not be performed. Instead, only differential data repair will take place. Data rebuilding is triggered if the disk or host remains offline beyond the preset rebuilding waiting period.

The default rebuilding waiting period is 4 hours for hosts and 30 minutes for disks.

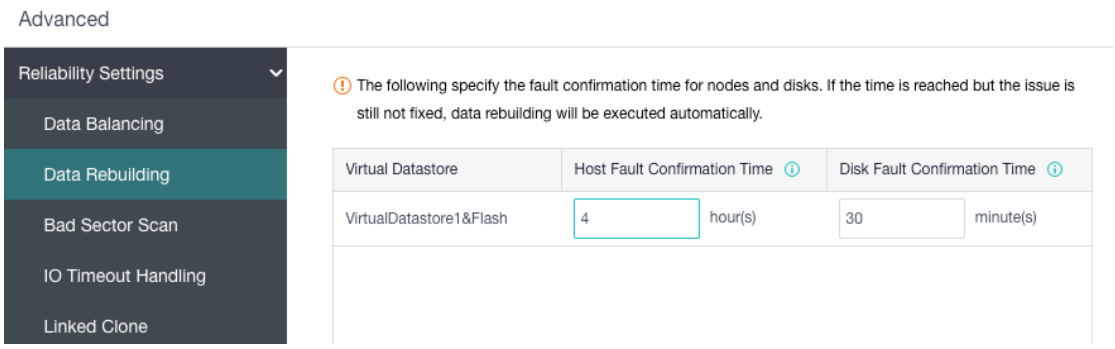


Figure 228: Data Rebuilding Settings

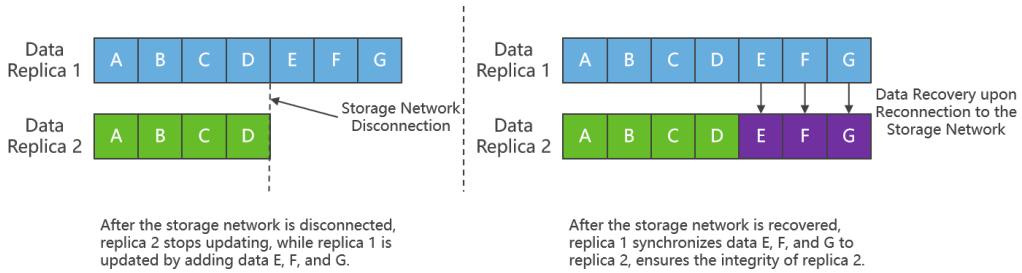


Figure 229: Data Rebuilding

NOTICE

1. Differential data repair is triggered not only by reinserting a loose disk but also under various other circumstances. For example, it is triggered when the storage network is reconnected shortly after an interruption or when a host goes back online after being brought offline for planned maintenance.

2. When a disk or host experiences a failure, a rebuilding task is initiated after the preset rebuilding waiting period elapses. However, if a user requires more time to confirm the fault, the rebuilding task can be postponed by the user.

5.2.4.6.1 Data Rebuilding Scenarios

After hardware failure occurs, the data on the failed hardware will be rebuilt on destination hardware at the shard level by using another replica as the data source. This restores replica integrity, achieves system self-healing, and enhances system reliability.

The data rebuilding mechanism will make maximum efforts to rebuild data and

restore replica redundancy, with the prerequisite that the following data distribution requirements are met:

1. Rebuilt data replicas follow the host anti-affinity principle.
2. Rebuilt witness replicas follow the host anti-affinity principle.

If a host failure occurs and the number of available hosts in the cluster after the failure is greater than or equal to the total number of replicas, the rebuilt data replicas and witness replicas must follow the host anti-affinity principle.

If a host failure occurs and the number of available hosts in the cluster after the failure is less than the total number of replicas, the rebuilt data replicas and witness replicas must follow the principle of disk group anti-affinity.

In three-replica scenarios, data rebuilding will be initiated as long as the four requirements above are met. In two-replica scenarios, if the number of available hosts in the cluster after a failure is less than the total number of replicas, rebuilding will not be initiated.

Below are three typical scenarios used as examples.

Two-Replica Data Rebuilding:

- Data Rebuilding After a Host Failure

1. In a three-host scenario, data rebuilding will not happen when a host fails because the number of available hosts in the cluster is less than the total number of replicas at this time.

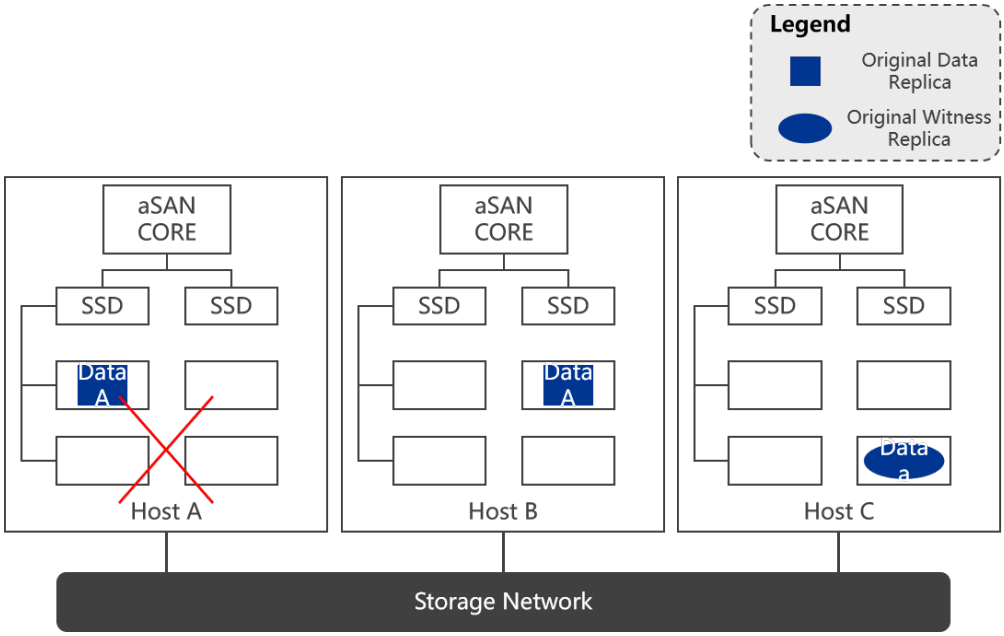


Figure 230: Single Host Failure in a Topology of Three Hosts with Two Replicas

2. In a four-host scenario, there are two data replicas marked as Data A and one witness replica marked as Data a. When host A fails, the Data A replica stored on it is lost.

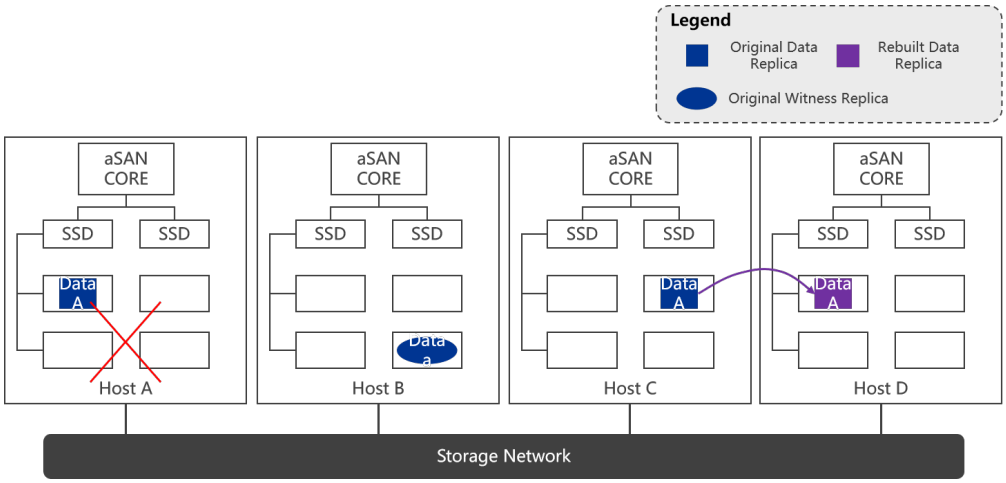


Figure 231: Single Host Failure in a Topology of Four Hosts with Two Replicas

After four hours (the preset rebuilding waiting period for users to confirm a host failure) have passed or when the user chooses to rebuild data immediately, data rebuilding starts by using the Data A replica on host C as the source. The rebuilt replica follows the host anti-affinity principle and is stored

on host D. (Host D did not have the Data A replica before rebuilding.)

- Data Rebuilding After a Disk Failure

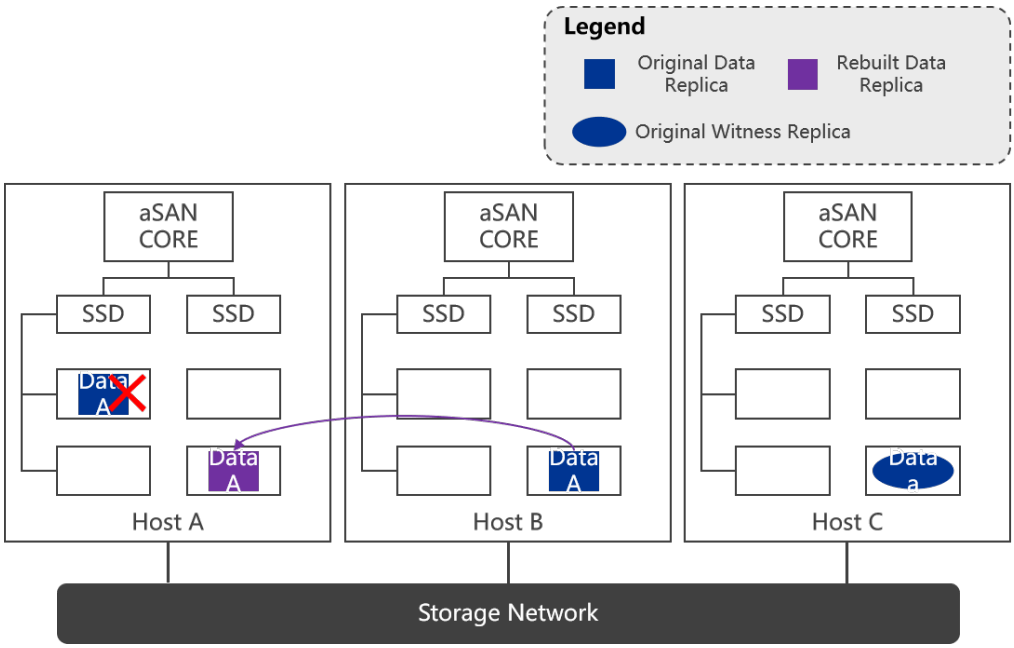


Figure 232: Single Disk Failure in a Topology of Three Hosts with Two Replicas

1. As shown in the diagram, the disk on host A that holds the Data A replica experiences a failure, resulting in the loss of Data A.
2. After 30 minutes (the preset rebuilding waiting period for users to confirm a disk failure) have passed or when the user chooses to rebuild data immediately, data rebuilding starts by using the Data A replica on host B as the source. The rebuilt replica follows the host anti-affinity principle and is stored on host A.
3. In a three-host and two-replica scenario, if a disk failure occurs, data will be rebuilt on another disk of the same host to meet the host anti-affinity requirement. In the figure above, because the disk failure takes place on host A, the rebuilt data is also stored on host A.

Three-Replica Data Rebuilding: Three/Four Hosts:

- Data Rebuilding After a Host Failure - Three Hosts

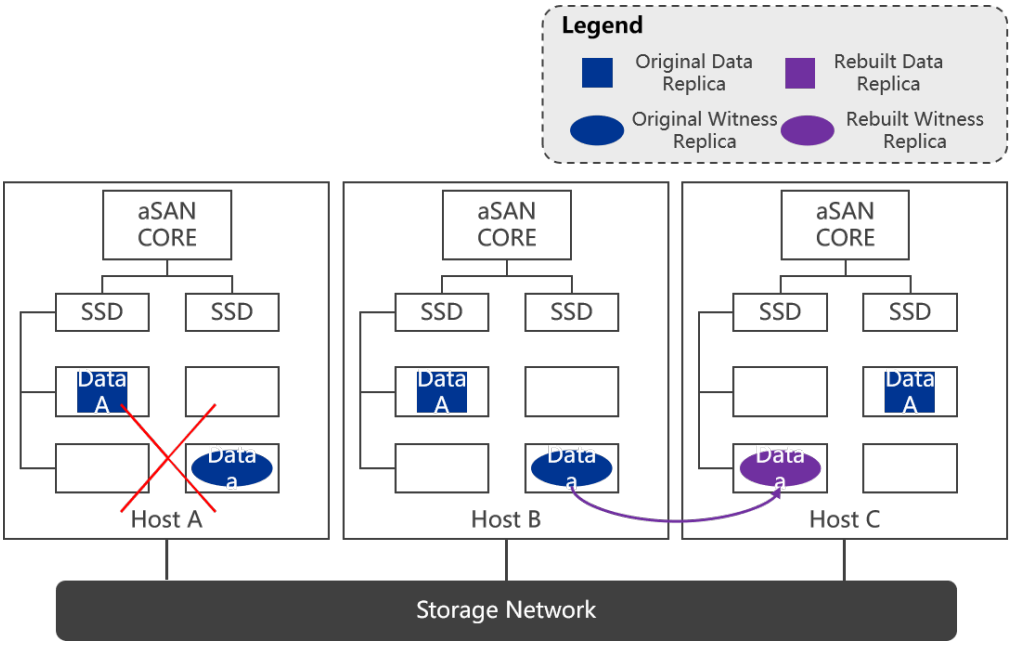


Figure 233: Single Host Failure in a Topology of Three Hosts with Three Replicas

1. In a three-host scenario, if a host fails, only the witness replica on the host is rebuilt. This is because each remaining host holds a data replica. Regardless of which host the new data replica is rebuilt on, the host anti-affinity principle is violated. Therefore, the data replica is not rebuilt.
2. After 4 hours (the preset rebuilding waiting period for users to confirm a host failure) has passed or when the user chooses to rebuild data immediately, rebuilding of the witness replica starts. The rebuilt replica follows the host anti-affinity principle and is stored on a host that did not have the witness replica before rebuilding.
3. In a three-host scenario, when data rebuilding is completed after one host fails, the topology becomes "two hosts + two data replicas + two witness replicas," with each host having one data replica and one witness replica.

- Data Rebuilding After a Host Failure - Four Hosts

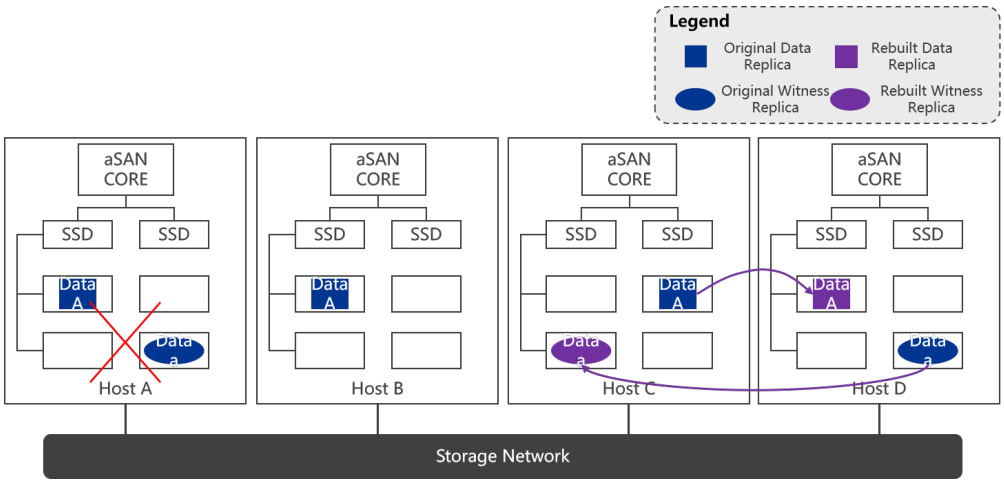


Figure 234: Single Host Failure in a Topology of Four Hosts with Three Replicas

When one out of four hosts fails, both the data replica and witness replica need to be rebuilt.

As shown in the figure above, when host A fails, the Data A and Data a replicas stored on that host are lost. At this point, one Data A replica and one Data a replica need to be rebuilt by using the Data A replica on host B or C and the Data a replica on host D. In a four-host scenario, when data rebuilding is completed after one host fails, the topology becomes "three hosts + three data replicas + two witness replicas."

Three-Replica Data Rebuilding: Five or More Hosts:

- Data Rebuilding After a Host Failure

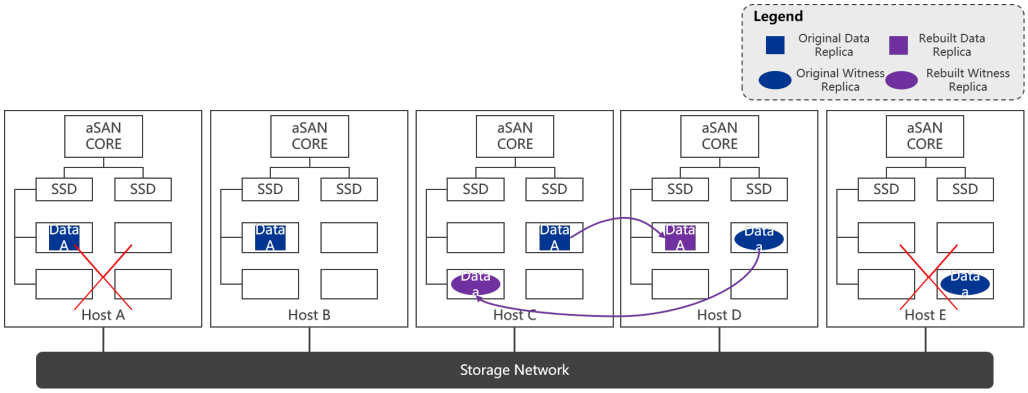


Figure 235: Single Host Failure in a Topology of Five Hosts with Three Replicas

A topology of five hosts with three replicas is used as an example. If host A and

host E fail, one Data A replica and one Data a witness replica are lost. After 4 hours (the preset rebuilding waiting period for users to confirm a host failure) has passed or when the user chooses to rebuild immediately, data rebuilding starts using the available replicas of the lost data. After rebuilding, the configuration becomes three hosts with three data replicas and two witness replicas.

- Data Rebuilding After a Disk Failure

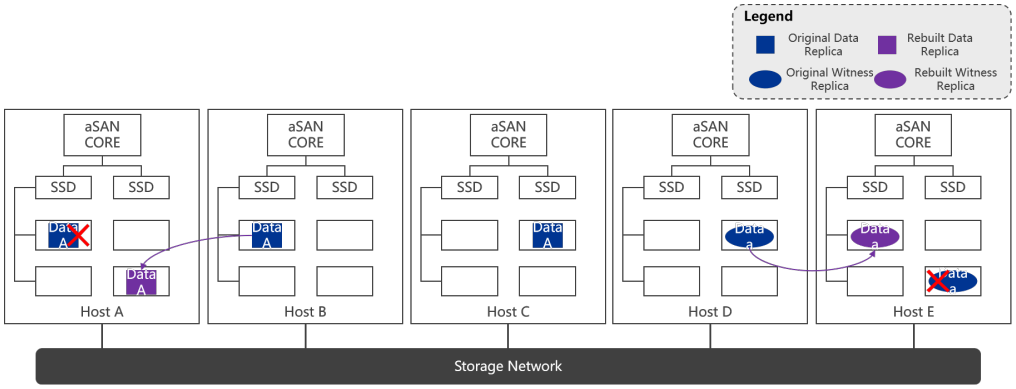


Figure 236: Single Disk Failure in a Topology of Five Hosts with Three Replicas

A topology of five hosts with three replicas is used as an example. If host A and host E experience a disk failure, and one Data A replica and one Data a witness replica are lost. After 30 minutes (the preset rebuilding waiting period for users to confirm a host failure) has passed or when the user chooses to rebuild immediately, data rebuilding starts using the available replicas of the lost data. Although there are disk failures, the cluster is still configured with five hosts, with three data replicas and two witness replicas, and the replica distribution after rebuilding still strictly complies with the host anti-affinity requirements.

5.2.4.6.2 Characteristics of Data Rebuilding

Data rebuilding has the following figures.

Globality and Concurrency:

To implement data rebuilding, concurrent I/O is employed, allowing simultaneous reading from multiple source disks and writing into multiple destination disks. Because the replicas of the shards lost in a failure are

present on multiple disks, the source for data rebuilding consists of multiple disks, and as many disks as possible are selected as the rebuilding destination while it is ensured that the data distribution requirements are met.

Data rebuilding after a host failure is used as an example to describe the globality and concurrency of the rebuilding process.

As shown in the diagram below, host B fails, and the data on the host (shards A/B/D/F/G/S/O/Q/M/U/N) is lost.

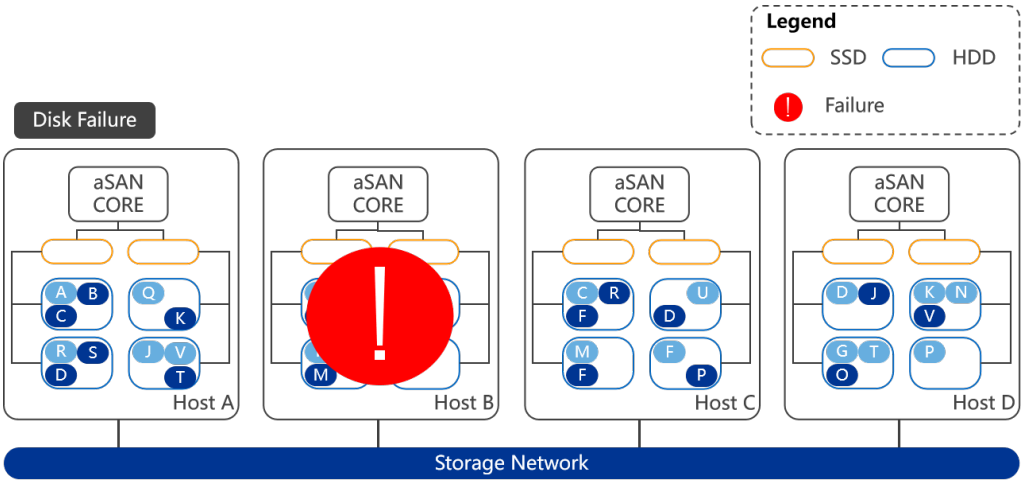


Figure 237: Corruption of One Replica Data for Shards A/B/D/F/G/S/O/Q/M/U/N Due to a Disk Failure

After 4 hours (the preset rebuilding waiting period for users to confirm a host failure) has passed or when the user chooses to rebuild immediately, the system starts data rebuilding using another available replica of the lost data, as shown in the diagram below.

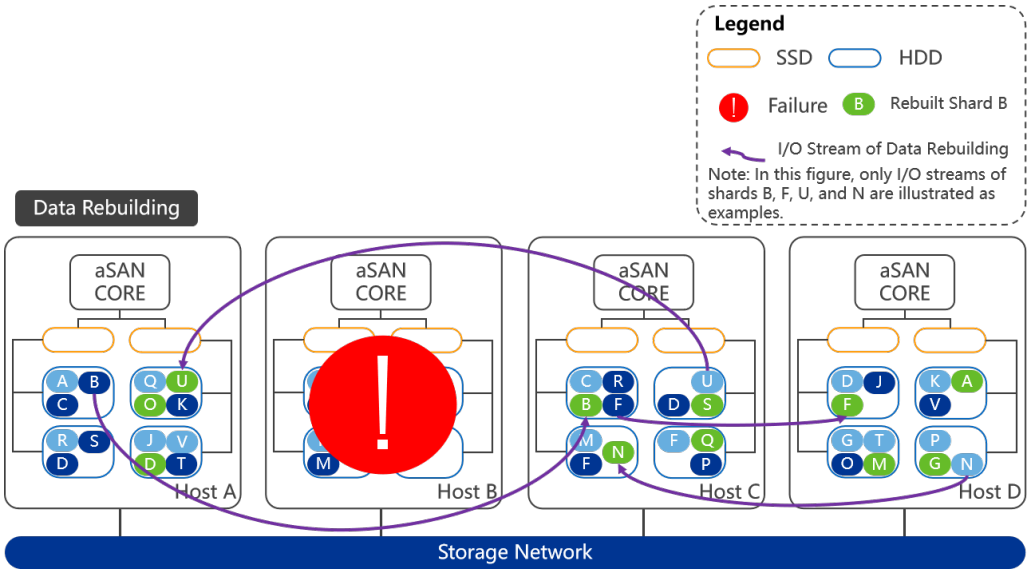


Figure 238: Replica Data Rebuilding for Shards A/B/D/F/G/S/O/Q/M/U/N

From the above diagram, it can be seen that for fast data rebuilding, the I/O of the process is concurrent, that is, reading from multiple source disks and writing into multiple destination disks.

Resumption from the Breakpoint:

Data rebuilding is performed on a shard-by-shard basis. If a network disconnection or another error results in the failure of the rebuilding task, only the shard being rebuilt at the moment of the failure is affected. (the default shard size is 4 GB). Rebuilding of the shards that have been rebuilt is not required, and the rebuilding of the failed shard can be resumed from the failure point.

Priority-Based Processing:

After initiating a rebuilding task, the system prioritizes the rebuilding of VM data that is critical to the user. For example, data of VMs selected as critical VMs, VMs using shared virtual disks, and VMs being started is given priority for data rebuilding. Users can specify VMs prioritized for data rebuilding within the rebuilding task. Priority-based data rebuilding ensures the protection of critical user data when storage volume space resources for rebuilding become insufficient.

Intelligent Throttling:

Data rebuilding consumes storage network bandwidth and disk I/O resources. To address this, aSAN offers intelligent throttling that is used to dynamically adjust the I/O consumption during the rebuilding, given the upper-level business I/O requirements. This ensures efficient data rebuilding without disrupting normal business operations.

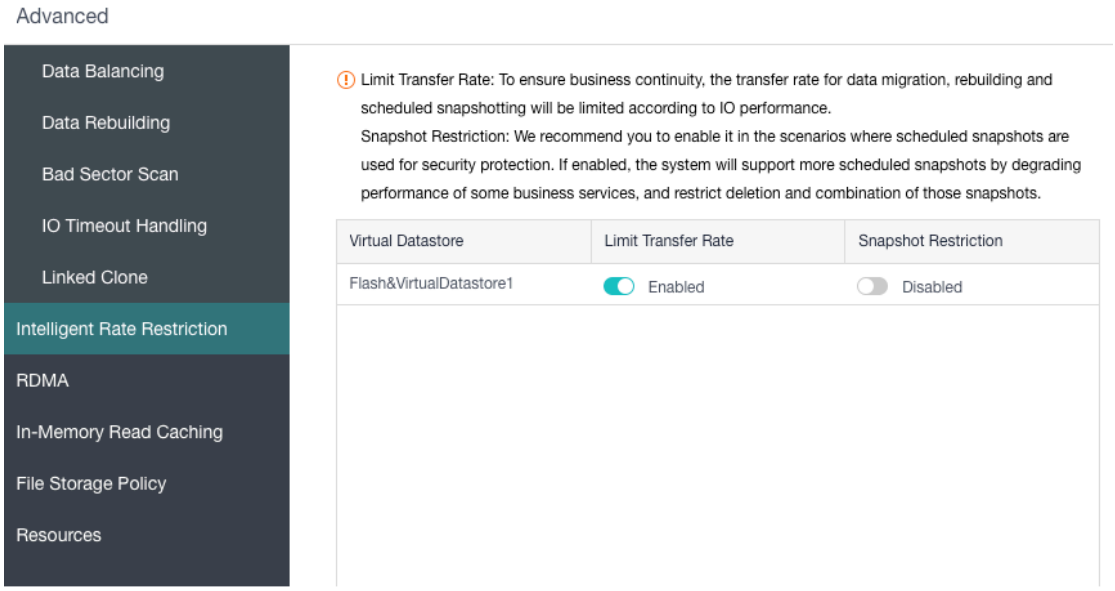


Figure 239: Business Performance Protection Configuration

NOTICE

1. Full-speed data rebuilding is at a speed of 1 TB/30 minutes. That is, when a 2 TB disk fully loaded with data fails, the rebuilding process takes only 1 hour, restoring the replica integrity in a timely manner. (Assume that three or more hosts exist, with each host having at least two cache disks and four data disks).

2. aSAN supports spare disks, but they are not necessary for automatic data rebuilding. Spare disks are mainly used for disk redundancy. When a disk fails, a spare disk automatically goes online to replace the failed disk, restoring the health of the storage volume in a timely manner.

5.2.4.7 Sub-Healthy Disk Detection and Handling

Based on extensive disk issue statistics, it has been found that disks are most

prone to subhealthy issues within the first three months of use and after two years of use.

As described in section 5.2.4.6 "Data Rebuilding," after a disk fails and is offline, aSAN can quickly rebuild data to restore replica redundancy.

In actual data center environments, besides the "healthy/online" and "failed/offline" states, another state is "sub-healthy" for disks, which is characterized by a slowdown, lagging, lifetime about to run out, and excessive bad sectors.

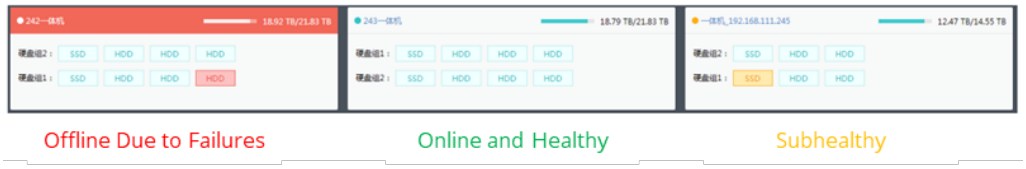


Figure 240: Disk States

Subhealthy disks potentially have a greater impact than those offline due to failures. If the storage system cannot preprocess such disks, it may result in VM latency across the entire cluster and even data loss.

aSAN provides different processing mechanisms for different subhealthy conditions to ensure data and business security.

5.2.4.7.1 Proactive Detection of Bad Sectors

Bad sectors on disks are common. Bad sectors can arise from low disk quality, aging due to prolonged usage, poor power supply, excessive heat, inadequate dust protection, or vibrations.

A disk with a bad sector is considered as in the subhealth state. Though the disk remains online and continues to provide storage service, data in the bad sectors cannot be read and faces a risk of loss.

Timely detecting bad sectors is vital. Typically, bad sectors are identified, marked, and recorded to the disk smartctl information after a failed attempt to read data (such as "h" and "n") from the disk.

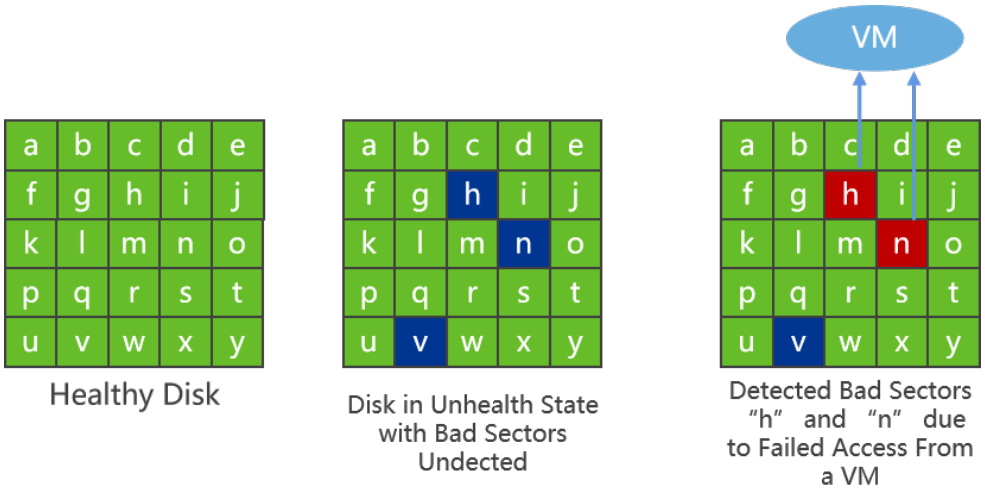


Figure 241: Bad Sector Detection

Therefore, bad sectors on a disk are not discovered until their data is read. Although data on these hidden bad sectors (such as "v") is no longer accessible, the system is unaware and does not take any repair measures. This poses a risk of data loss. For example, once the other replica of data "v" is also abnormal, data "v" will be lost.

aSAN offers the proactive scanning feature to detect hidden bad sectors in time, preventing data from being in a prolonged "single-replica" state. This feature has the following characteristics:

1. You can customize the scan period, which defaults to from 01:00 to 06:00. Bad sectors are immediately repaired once they are detected.
2. Adaptive flow control dynamically adjusts the scan speed based on disk I/O usage, with a performance impact of less than 5%.
3. When the business workload is moderate (a disk usage of around 60%), the disk bad sector scan proceeds at a speed of roughly 1 TB every 3 hours per host. In other words, it takes a total of 30 hours to complete a full scan for 10 TB of data on each host.

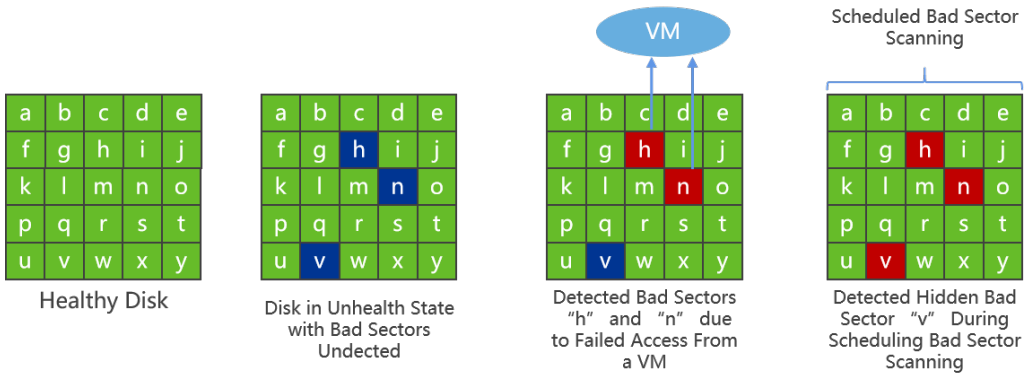


Figure 242: Scheduled Bad Sector Scanning

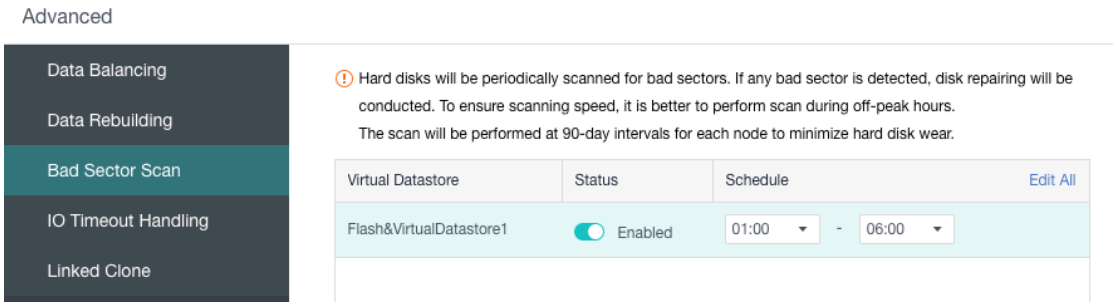


Figure 243: Settings for Scheduled Bad Sector Scanning

5.2.4.7.2 Proactive Bad Sector Repair

aSAN repairs bad sectors immediately after they are detected. It reads data from the healthy replica and restores the data to the disk's reserved sector (each disk has a number of reserved sectors by default) to ensure data redundancy.

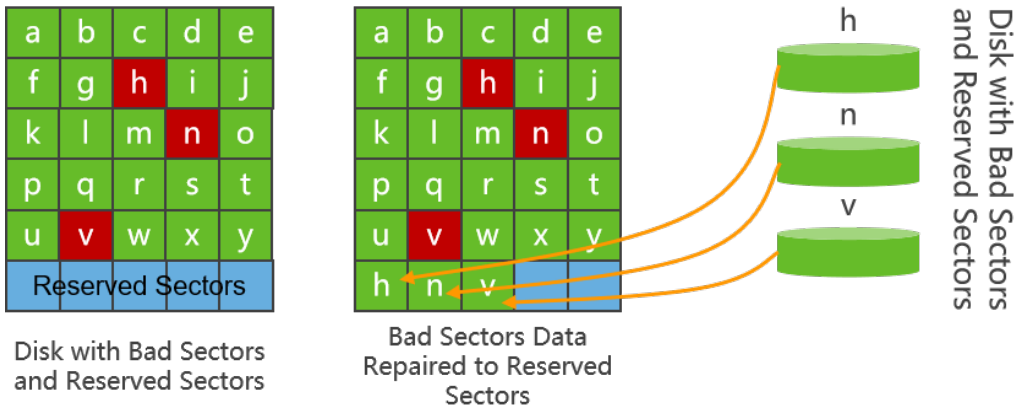


Figure 244: Proactive Bad Sector Repair

5.2.4.7.3 Proactive Data Migration

When a disk has too many bad sectors or an SSD is approaching the end of its lifetime, replace the disk in time.

aSAN proactively migrates data from disks with excessive bad sectors or SSDs approaching the end of their lifetime to other healthy disks to ensure data redundancy.

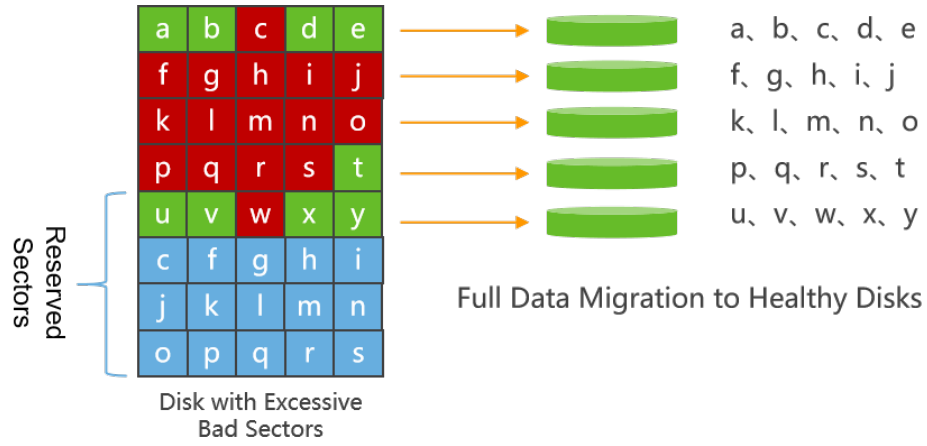


Figure 245: Proactive Data Migration

5.2.4.7.4 Soft Isolation Upon Failure

Mainstream distributed storage employs strong consistency algorithms during multi-replica data writes, so that requests are not returned to business services until writes to all replicas have been completed. When a component (such as a disk or host) is unhealthy, VM write I/O may become slow due to the barrel effect, thus degrading business performance. To solve this problem, Sangfor aSAN provides a set of solutions for soft isolation in case of failure, covering features such as business failure detection, heuristic failure diagnosis, and precise silencing of failed components to effectively guarantee business continuity.

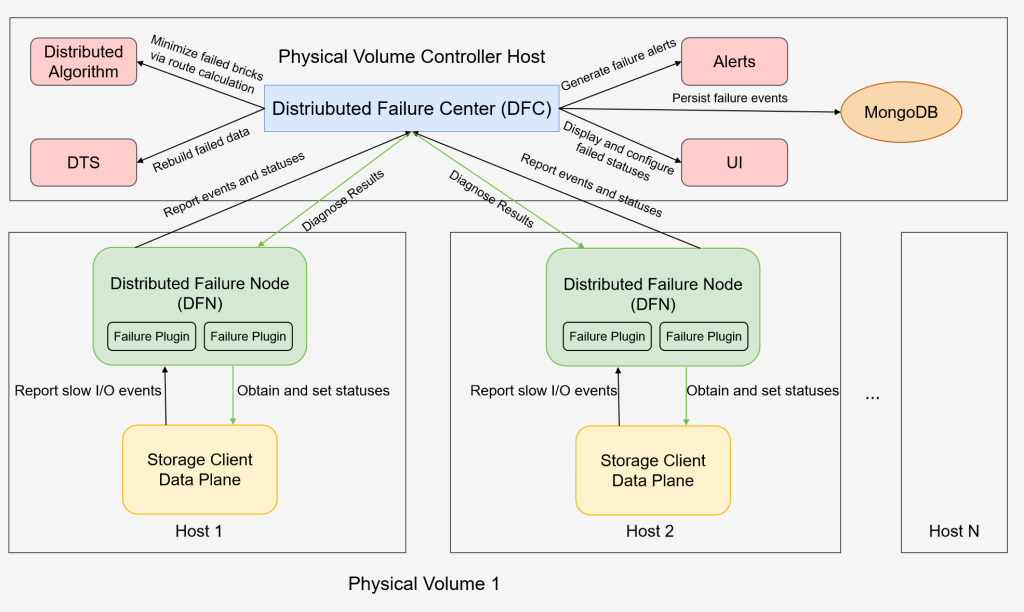


Figure 246: Soft Isolation Framework (1)

The soft isolation framework is divided into two failure handling parts based on the storage volume: storage client data plane and soft isolation control plane. The storage client data plane temporarily isolates unhealthy replicas in seconds through data replica health detection and reports failure information to the control plane to ensure business availability. In the soft isolation control plane, the Distributed Failure Center (DFC) collects the failure information reported by each distributed failure node (DFN) plugin in the storage volume and performs comprehensive and accurate diagnosis based on them to avoid false-positives due to individual failures. After recovering a failure within a short time, a recovery message will be sent to the storage client data plane in time, which will then reuse the original unhealthy replica to prevent the data from being in a few replicas for too long. For failures that cannot be recovered within a long period, complete isolation messages will be sent to the data plane, and failed data replicas will be rebuilt to restore multiple replicas for data reliability.

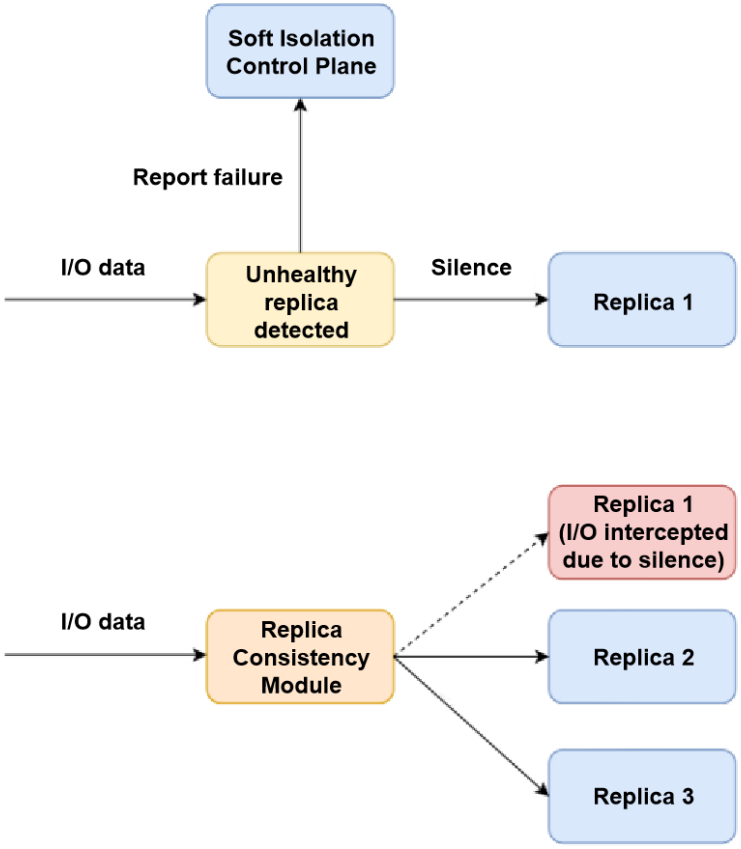


Figure 247: Soft Isolation Framework (2)

The soft isolation framework thoroughly isolates software in the application layer without manipulating the hardware such as unplugging hard disks to avoid compatibility issues caused by different hardware, where DFNs are integrated with the interfaces of the following two failure plugins to extend the failure handling capabilities:

- **Client Data Plane Plugin**: Collects, checks, and reports the failure information of unhealthy replicas in the client data plane to the DFC, responds to the returned diagnosis messages, and takes appropriate actions, such as recovering or permanently isolating failed replicas.
- **Stuck and Slow Disk Detection Plugin**: Identifies stuck and slow disks according to the disk type (SSD and HDD) with a set of highly efficient and accurate algorithms based on massive data and reports them to the DFC. It only collects disk I/O data, while whether a disk needs to be isolated and

rebuilt depends on the failure data in the data plane and the diagnosis results from the DFC.

5.2.4.8 High Reliability of Storage Links

The private storage network is a dedicated Ethernet network for data communication within a virtual datastore. It supports three network deployment modes: link aggregation disabled, link aggregation with one switch, and link aggregation with two switches.

Link aggregation disabled:

This is the simplest deployment mode but with the lowest fault tolerance. Once a link fails, the read and write operations are interrupted due to the disconnection between the host and the virtual datastore on that link.

Link aggregation with one switch:

This mode offers higher fault tolerance than the link aggregation disabled mode. A single link failure does not affect the use of the virtual datastore, whereas a switch failure can cause the entire virtual datastore to go offline.

Link aggregation with two switches:

This deployment mode offers the highest fault tolerance. It prevents single line or switch failures from affecting the use of the virtual datastore and increases storage communication bandwidth between hosts.

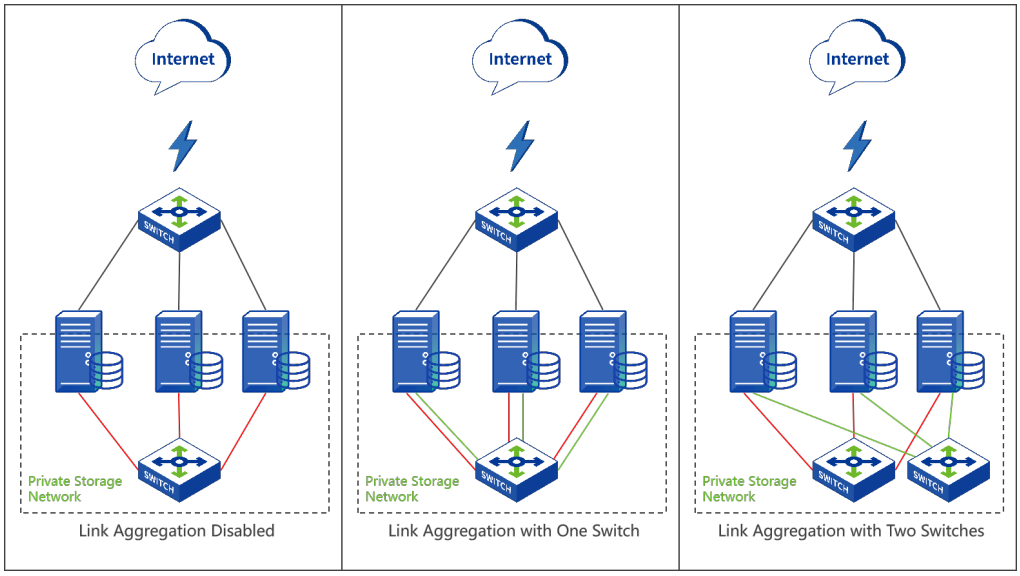


Figure 248: Three Link Aggregation Modes

Link aggregation with one switch and link aggregation with two switches in aSAN both automatically balance loads in active-active mode. This enhances network reliability and increases storage network bandwidth.

- **Link aggregation with one switch**: Loads are balanced based on TCP connections. In versions earlier than aSAN 3.0, fragmentation is not supported, and each virtual disk has one TCP connection that corresponds to one link. As aSAN 3.0 supports fragmentation, a virtual disk is split into multiple 4 GB fragments that are scattered on different physical disks. This creates multiple TCP connections, enabling a single virtual disk to use the bandwidth of the two links.
- **Link aggregation with two switches**: Traffic is allocated to two links through IP packet-based round-robin, which enables a single virtual disk to use the bandwidth of the two links.

NOTICE

1. For a private aSAN storage network in one of the link aggregation modes, no link aggregation configuration is required on the switch. As the private storage network inherently supports link aggregation, you only need to ensure that the network cables of a regular Layer 2 switch are plugged correctly.

2. For link aggregation with two switches, switch stacking is not required. You only need to ensure that the network cables of the two regular Layer 2 switches are plugged correctly.

3. A private aSAN storage network requires 10GE NICs to ensure higher storage performance for the upper layer and faster speed in data rebuilding and data balancing.

5.2.4.9 Spare Disks for Data Protection

aSAN offers spare disks for data protection. When a damaged HDD fails to handle I/O requests and affects the business, the system's spare disk automatically takes over the work of the damaged HDD without the need for manual intervention. Disk failures may occur frequently in large clusters with numerous disks. Preparing spare disks for data protection can prevent data

loss in case damaged disks are not replaced in time.

5.2.4.10 Enhanced Reliability of General I/O Processing Module for Kylin OS

The I/O processing module is a critical communication system in the host OS. It integrates with DAS storage, virtual storage, external storage (such as EDS and iSCSI storage), and peripherals (such as USB). Therefore, the HCI team has thoroughly tested and optimized the general I/O processing module of Kylin OS to ensure its reliability.

The HCI team has tested storage protocol stacks and interfaces involved in general I/O processing, covering more than 160 types of failures in over 300 scenarios, such as file system failures, disk failures, and interface failures. In addition, the team has tested the general I/O processing module by combining kernel stress testing with fault simulation.

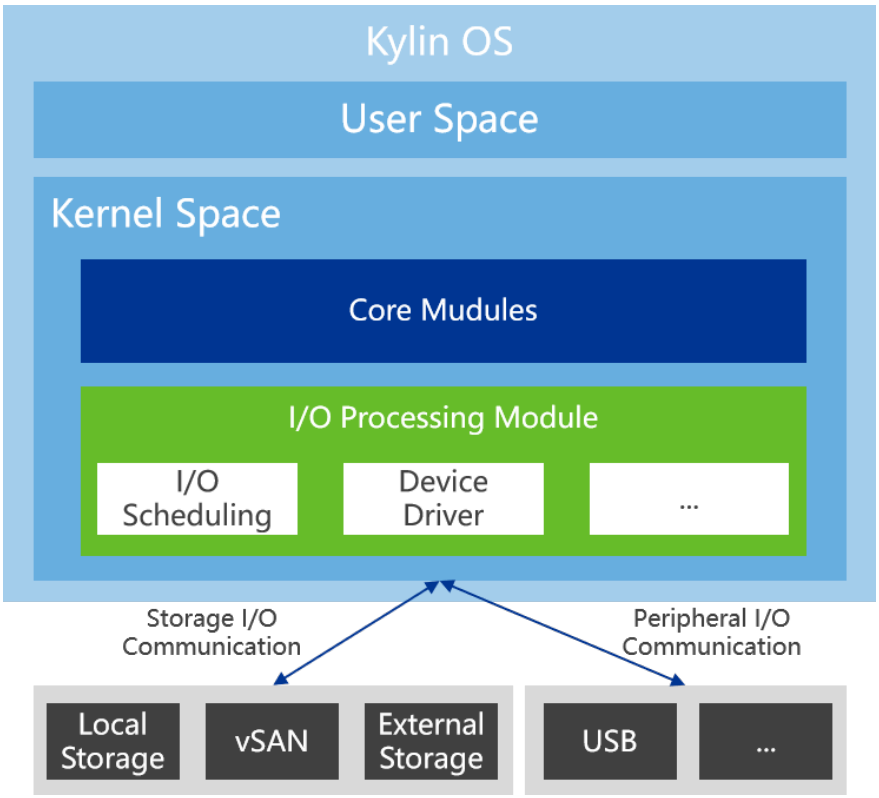


Figure 249: Enhancement in General I/O Processing

During the test, over 10 stability issues were identified in the general I/O processing module. For example, inaccurate stuck/slow disk detection, system crashes due to USB device insertions or removals, inability to bring a disk online after a hot swap, crashes in scheduling a block device's I/O queues, stuck I/O operations in the I/O scheduling module, inability to detect and remove failed NVMe disks, system hangs due to NVMe disk failures, and deadlock issues and system crashes caused by NVMe disks.

To fix the issues uncovered during the test and enhance the stability of the I/O processing module, the HCI team has optimized the kernel code. For example, to address inaccurate stuck/slow disk detection, the HCI team has provided a kernel solution to enable the system to quickly and accurately identify and isolate stuck or slow disks; to address system hangs and crashes due to NVMe disk failures, the HCI team has optimized the kernel and the NVMe driver to effectively isolate faulty NVMe disks and avoid their impact on the system; especially in scenarios where NVMe disks are used as cache disks for virtual storage, faulty NVMe cache disks are effectively isolated to prevent impact on the entire virtual datastore.

5.2.5 Smart O&M

Sangfor provides a self-developed storage forecast feature based on the analysis of extensive real business data and the integration of diverse machine learning and deep learning algorithms. Storage forecast consists of bad sector prediction, capacity prediction, and SSD lifetime prediction.

5.2.5.1 Bad Sector Prediction

Bad sector prediction is a highly accurate feature that Sangfor provides based on the extensive data of disks with bad sectors collected from real use scenarios (SMART data, performance parameters, and disk logs) and advanced training models and algorithms. The following figure shows how bad sector prediction works.

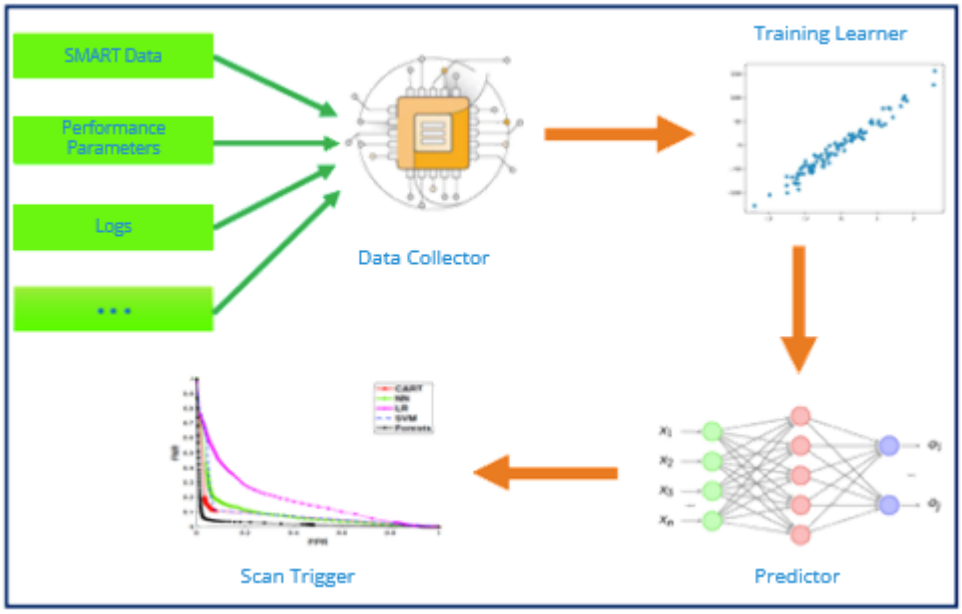


Figure 250: Principle of Bad Sector Prediction

aSAN predicts disks with bad sectors in a cluster according to upper-layer business pressure and disk usage, and preferentially scans these disks for bad sectors at the user-defined time every day. This shortens the time to detect bad sectors on disks from weeks or even months to just a day. Bad sector prediction, in combination with aSAN's data rebuilding, forms a closed-loop mechanism to handle bad sectors on disks, minimizing the risk of single-replica data due to bad sectors.

On the bad sector prediction page, you can view the bad sector prediction, scan, and repair results of each disk, as well as disk risk assessment reports and alerts that prompt you to replace disks with excessive bad sectors in time to prevent data loss.

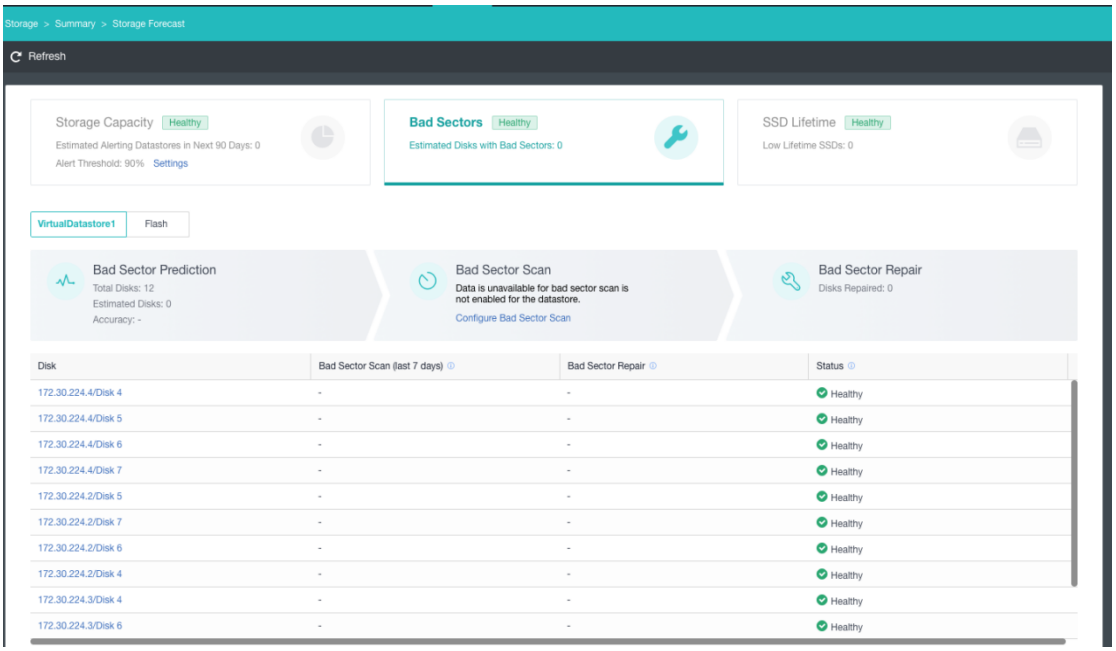


Figure 251: Bad Sector Prediction

5.2.5.2 Capacity Prediction

Online capacity prediction is a highly accurate feature that aSAN provides based on seasonal decomposition, change point detection, and linear models. It dynamically predicts the growth trend in capacity usage of each virtual datastore over the next 90 days for proactive capacity management.

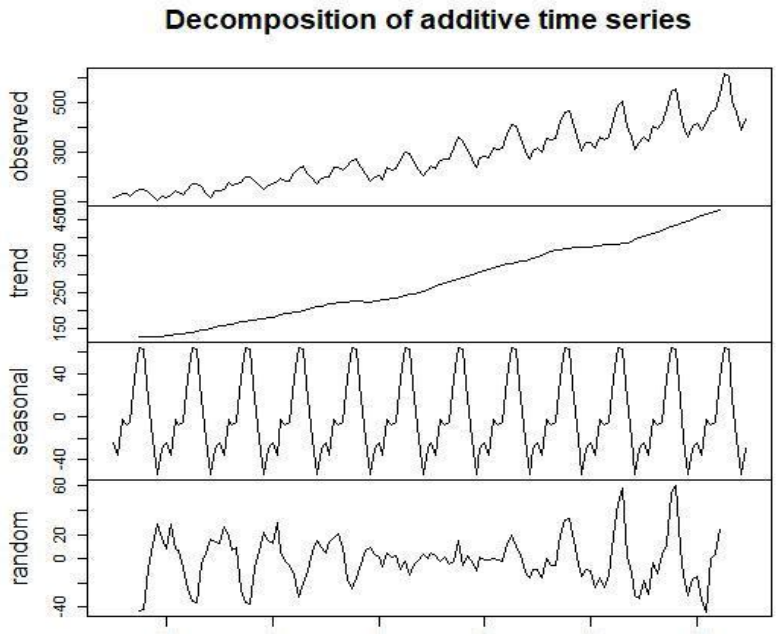


Figure 252: Principle of Virtual Storage Capacity Prediction

On the capacity prediction page, you can view the capacity information of each virtual datastore, including the raw capacity, used capacity, dynamically predicted trend, and risk level assessment based on the user-defined capacity usage threshold. There are three risk levels: high risk (the threshold is reached), risk (the threshold will be reached in the next 90 days), and healthy (the threshold will not be reached in the next 90 days).

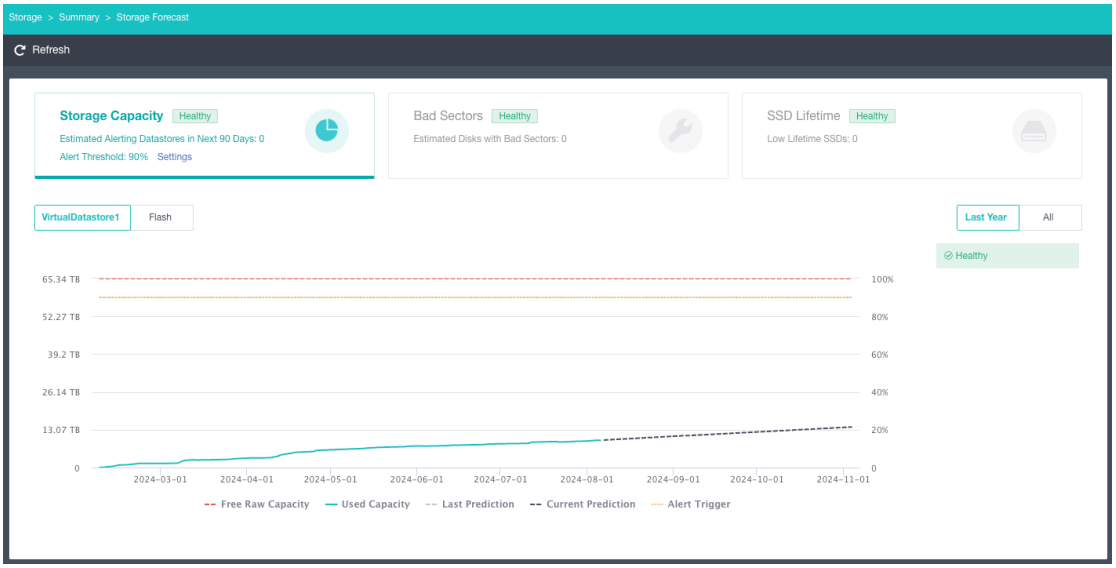


Figure 253: Virtual Storage Capacity Prediction Page

NOTICE

1. Disk lifetime, also referred to as Total Bytes Written (TBW), is calculated based on the following formula: TBW = Diskful Writes Per Day * Disk Capacity * (5 * 365).

2. Diskful Writes Per Day (DWPD) refers to the number of times you can overwrite an SSD's entire capacity each day of its lifetime.

5.2.5.3 SSD Lifetime Prediction

aSAN collects and analyzes I/O data from SSDs in the cluster, calculates their remaining lifetime, and predicts and displays their usable duration according to the resource usage trend of the upper-layer business. In addition, aSAN

provides risk assessment based on the usable duration to prompt you to replace SSDs approaching the end of their lifetime in the cluster in time.

There are three risk levels: high risk (lifetime <= 10% or usable duration <= 90 days), risk (lifetime <= 15% or usable duration <= 180 days), and healthy (lifetime > 15% and usable duration > 180 days).

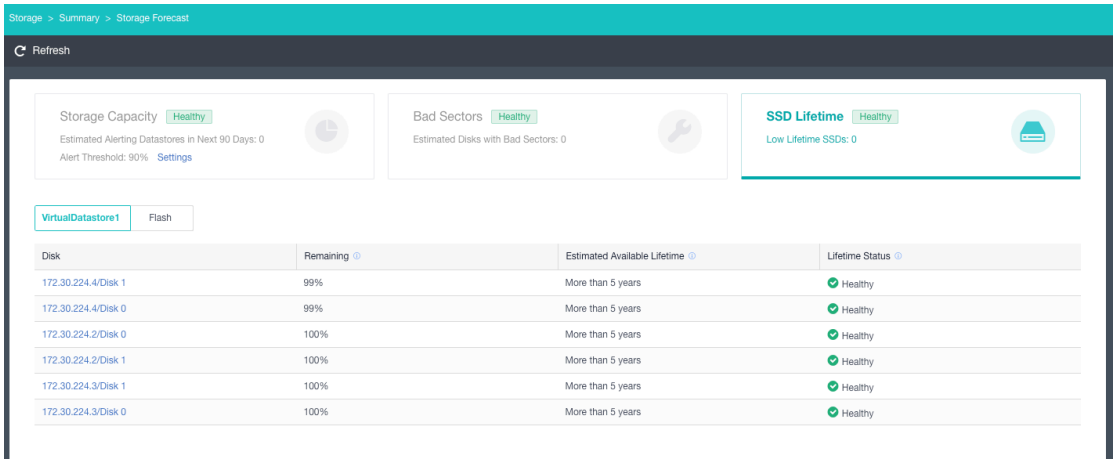


Figure 254: Lifetime Prediction Page

The SSD lifetime prediction process is as follows:

1. aSAN obtains an SSD's remaining lifetime (percentage) from its SMART data.
2. aSAN analyzes the bytes written to the SSD, predicts the bytes to be written to the virtual storage, and calculates the SSD's remaining usable duration based on its remaining lifetime.
3. aSAN assesses the SSD's risk level based on its remaining lifetime and remaining usable duration.

5.2.5.4 Data Self-Balancing

5.2.5.4.1 Data Imbalance

aSAN automatically selects suitable hosts or disks to store data for balanced data distribution. Data imbalance may occur within a virtual datastore in the following scenarios:

1. When you add hosts or disks to a virtual datastore, the newly added hosts or disks are hardware with significant unused capacity in the virtual datastore.
2. When you delete VMs, substantial space is released from the hosts or disks where the VMs reside in the virtual datastore.
3. After a failed host or disk is taken offline, its data is rebuilt on other disks in the cluster (see section 5.2.4.6 "Data Rebuilding"). The next time you bring the failed host or disk online, its capacity is totally unused.
4. When a VM's data surges, the host where the VM's aggregated replica resides experiences a notably faster growth in capacity usage compared to other hosts, even if other hosts also store the aggregated replicas of other VMs.

5.2.5.4.2 Objectives of Data Balancing

aSAN ensures even distribution of data across hosts in a virtual datastore and optimal utilization of resources on each host through its data balancing mechanism:

1. Ensures minimal difference in disk capacity usage across the entire virtual datastore.
2. Prevents situations where a disk in the virtual datastore is full while others still have free space.

aSAN evaluates data balance at the disk level, ensuring data balancing across disks and across hosts.

5.2.5.4.3 Triggers for Data Balancing

Scheduled Data Balancing:

Scheduled data balancing is initiated during the time you specify, as shown in the following figure. When there is a notable capacity usage difference among disks within a virtual datastore, some data is migrated from the disk with higher capacity usage to those with lower capacity usage, accomplishing the first objective of data balancing.

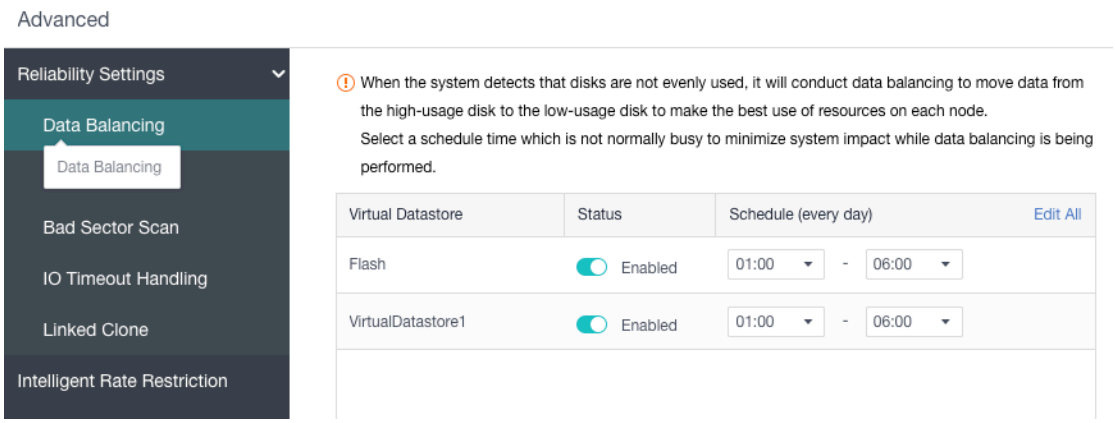


Figure 255: Settings for Scheduled Data Balancing

During the time you specify, aSAN's data balancing module scans all disks in the virtual datastore. If the difference between the highest and the lowest disk capacity usage within the virtual datastore exceeds a specific threshold (defaults to 30%), scheduled data balancing is triggered. The task ends when the difference in capacity usage between any two disks in the virtual datastore is no more than a specific threshold (defaults to 20%).

For example, after you expand the capacity of the virtual datastore, data is migrated to the newly added disks during the time you set for data balancing.

NOTICE

Schedule data balancing for the off-peak hours of your business to minimize its impact on regular I/O operations. If the data balancing task is not completed within the specified time range for the day, the task will pause and then resume during the next scheduled time.

Automatic Data Balancing:

Automatic data balancing is initiated by the system, eliminating the need for manual operations. It helps accomplish the second objective of data balancing by preventing the subhealthy state where a disk in the virtual datastore is full while others still have space.

Automatic data balancing is triggered when the capacity usage of a disk in the virtual datastore exceeds a specific threshold (defaults to 90%). The task ends when the difference between the highest and the lowest disk capacity usage

within the virtual datastore is below a specific threshold (defaults to 3%).



When all disks in the virtual datastore exceed 90% capacity usage and the difference in capacity usage between any two disks is less than 2%, the virtual datastore is almost full. In this case, data balancing serves no purpose and you must increase the virtual datastore's capacity to ensure abundant space on each host for uninterrupted data writing.

Implementation of Data Balancing:

When data balancing is triggered, aSAN, based on calculation, determines the destination disk where each shard from the source disk will be placed. There may be multiple destination disks, and they may be on the same host as the source disk or on another host. The principles for determining destination disks are as follows:

1. Anti-affinity: A data shard and its copies are not located on the same host after migration.
2. Optimal performance: Disks that are eligible for the placement policies described in section 5.2.3.4 "Optimal Data Distribution" are preferred.
3. Optimal capacity: Disks with lower capacity usage are preferred.

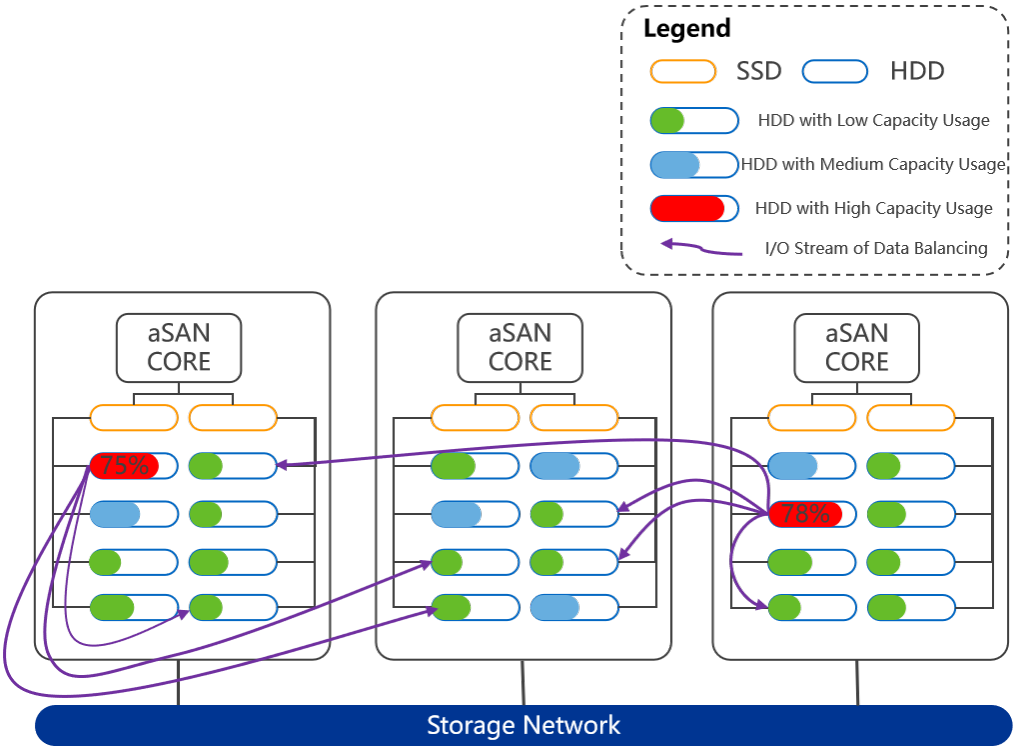


Figure 256: Implementation of Data Balancing

After determining the destination disk, the following data protection measures are taken during data balancing:

1. During data balancing, any new or modified data in the shard being balanced is simultaneously written to both the source and destination to provide an extra replica.
2. Before data balancing is complete, the data balancing module verifies source and destination data to ensure data consistency.
3. After data balancing is complete, the source shard is moved to a temporary directory and retained there for a period of time before it is deleted.

5.2.5.5 SkyOPS

Overview:

Sangfor SkyOPS offers one-stop monitoring SaaS services including proactive remote technical support service, DBA expert service, and security expert service, to guarantee the stability and reliability of resources in on-premises

data centers and the managed cloud. This frees IT staff from tedious maintenance work, enabling them to focus on business innovations.

Features:

1. Risk Monitoring

Offers full-stack risk monitoring capabilities for Sangfor O2O synergized cloud to cover hardware, platform, patches, VMs, applications, security events, and vulnerabilities, and focuses on event remediation for efficient response to data center events.

2. Real-Time Alert Notification

Sends real-time alert notifications through WeCom, DingTalk, email, or text message, and supports aggregating and categorizing alerts to only notify users of critical alerts.

3. Proactive Support Service

The expert team provides 24/7 service. For critical issues, corresponding work orders will be automatically generated to ensure these issues can be fixed promptly. This shift from a reactive to a proactive approach improves response efficiency.

Principle of Communication between HCI and Cloud Agent:

SkyOPS is activated off HCI. By deploying a Cloud Agent VM (a tenant) as a communication agent within the managed cloud platform, a one-way connection and two-way communication between the managed cloud platform and the Cloud Agent is built. The managed cloud platform sends the AK/SK configuration to the Cloud Agent while connecting to it, after which the Cloud Agent accesses the managed cloud platform with this AK/SK for verification. The Cloud Agent communicates with the managed cloud platform through a NIC connected to the internet.

The Cloud Agent calls the key interface of HCI to encrypt SK and then sends the IP address, port, AK/SK, and other configurations to HCI to connect to it. After that, the Cloud Agent collects and periodically reports the data of hardware and services on HCI to the managed cloud platform through the web API. Then the managed cloud platform organizes, filters, and calculates this data to

generate intelligent alerts and display them on HCI.

5.2.6 Advanced Storage Features

5.2.6.1 Storage Partitioning

Sangfor aSAN supports two types of storage partitioning: host-level partitioning and disk-level partitioning.

5.2.6.1.1 Host-Level Partitioning

Host-level partitioning in aSAN allows you to group hosts within an HCI cluster into different storage resource pools. As shown in the following figure, the six hosts in a cluster are grouped into two virtual datastores based on host-level partitioning, with each consisting of three hosts.

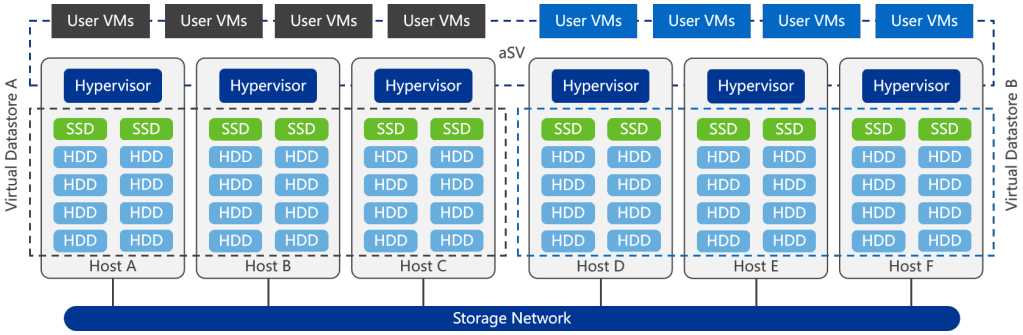


Figure 257: Host-Level Partitioning

Host-level partitioning in aSAN offers the following capabilities:

1. Business-based performance isolation

In actual scenarios, different types of business systems require different storage performance and capacity. As a result, their requirements for server hardware specifications (like CPU, memory, and disks) vary accordingly. With host-level partitioning, you can group servers with different specifications within an HCI cluster into different virtual datastores. This implements performance isolation to meet the needs of the upper-layer business.

Here is an example: You can use a hybrid virtual datastore consisting of SSDs and HDDs for non-critical business, an all-flash virtual datastore or a hybrid

virtual datastore with a higher SSD capacity ratio for high-performance business, and a hybrid virtual datastore with a higher HDD capacity ratio for backup business.

2. Hardware-based elastic scalability

When an HCI cluster's computing and storage resources run low or new businesses are introduced, you need to scale out the cluster by adding hosts. However, as hardware in the market evolves quickly, mainstream hardware often outperforms that of the cluster. With host-level partitioning, you can group new hosts into a virtual datastore during the scale-out. This avoids the bucket effect and maximizes the resources of both the old and new servers. You can also adopt host-level partitioning for reusing existing servers by building clusters of new and old servers.

An HCI cluster consolidates computing and storage resources. As aSAN allows VMs in the cluster to access computing resources across virtual datastores, you can make the most of the cluster's computing resources in host-level partitioning scenarios.

In host-level partitioning scenarios, the default startup policies of VMs in aSAN are as follows:

1. Priority 1: Starts and runs VMs on the hosts where aggregated replicas reside.
2. Priority 2: Runs VMs on other hosts within the virtual datastore that stores the data.
3. Priority 3: Runs VMs across virtual datastores when the CPU and memory resources within the current virtual datastore are insufficient.



Host-level partitioning requires at least six hosts.

5.2.6.1.2 Disk-Level Partitioning

In addition to host-level partitioning, aSAN also provides finer-grained disk-level partitioning.

To use disk-level partitioning in aSAN, your cluster must have three or more hosts. As shown in the following figure, you can divide a 3-host cluster into one all-flash virtual datastore for high-performance business and one hybrid virtual datastore for high-capacity business.

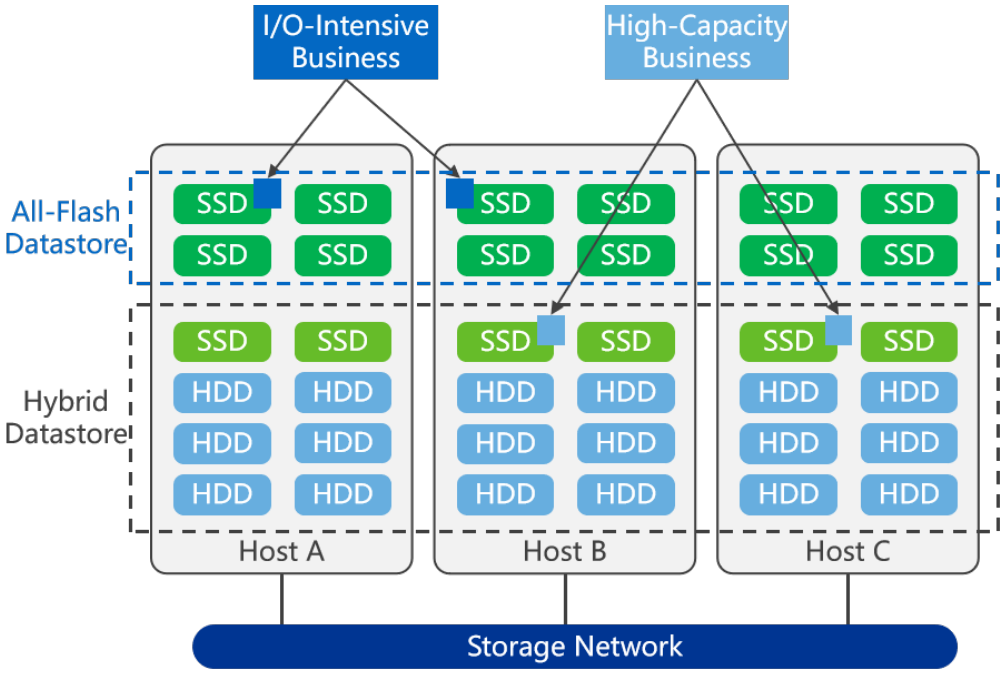


Figure 258: Disk-Level Partitioning

Similar to host-level partitioning, disk-level partitioning in aSAN supports business-based performance isolation and hardware-based elastic scalability. It also offers the following capabilities:

1. Lowers the threshold for multi-virtual datastore deployment

Host-level partitioning requires at least six hosts in a cluster. For clusters with less than six hosts, you can achieve multi-virtual datastore deployment using disk-level partitioning in aSAN. This ensures business-based performance isolation for small and medium-sized enterprises.

2. Offers scalability to meet upper-layer business needs

In actual scenarios, it is difficult for O&M personnel to accurately predict the growth in business concurrency and data volume at the inception of a business system. As business concurrency and data volume increase, the business requires higher storage performance and capacity for the data center. In this

case, you can use the disk-level partitioning feature of aSAN to adapt to the increasing business needs.

Here is an example: A client runs a business system on a hybrid virtual datastore. As the business system requires higher performance, the client can purchase SSDs to create an all-flash virtual datastore and migrate the business system to the new virtual datastore.

3. Maximizes the performance of heterogeneous disks

SAS disks outperform SATA ones. When a cluster's hosts have both SAS and SATA disks, you can group SAS and SATA disks into separate datastores using disk-level partitioning. This avoids the bucket effect and maximizes the performance of SAS disks.

NOTICE

1. Disk-level partitioning requires at least three hosts.

2. There can be up to six virtual datastores in a cluster, with up to one of them being a stretched virtual datastore.

3. A virtual datastore can include partial or all of the disks on the server.

4. An all-SSD virtual datastore does not require cache disks but requires at least two SSDs on each host.

5. A host cannot belong to three virtual datastores at the same time.

6. A host cannot belong to a stretched virtual datastore and a regular virtual datastore at the same time.

7. All-flash datastores for deduplication and compression only support host-level partitioning. All-flash sub-datastores based on disk-level partitioning do not support deduplication and compression.

5.2.6.2 Storage Policies

When creating VMs, you must configure storage policies. Storage policies in aSAN have the following characteristics:

1. aSAN offers common storage policies and allows you to save custom storage policies as templates. This significantly simplifies the workflow and reduces the time for deploying VMs with identical configurations.
2. To suit ever-changing business needs, aSAN allows you to modify storage policies even after they are applied to VMs.
3. To meet diverse business requirements for disk performance and reliability, aSAN allows you to configure storage policies for VMs or virtual disks.

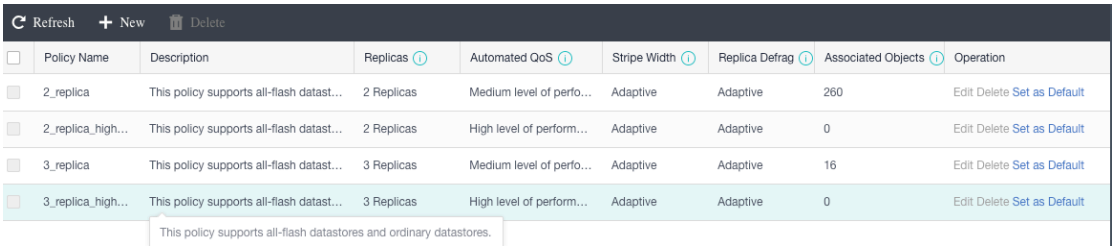


Figure 259: Storage Policy Settings

5.2.6.3 Virtual iSCSI Storage

aSAN supports the standard iSCSI protocol. aSAN storage resources can be used as dedicated external iSCSI storage for physical servers or virtualization platforms like VMware.

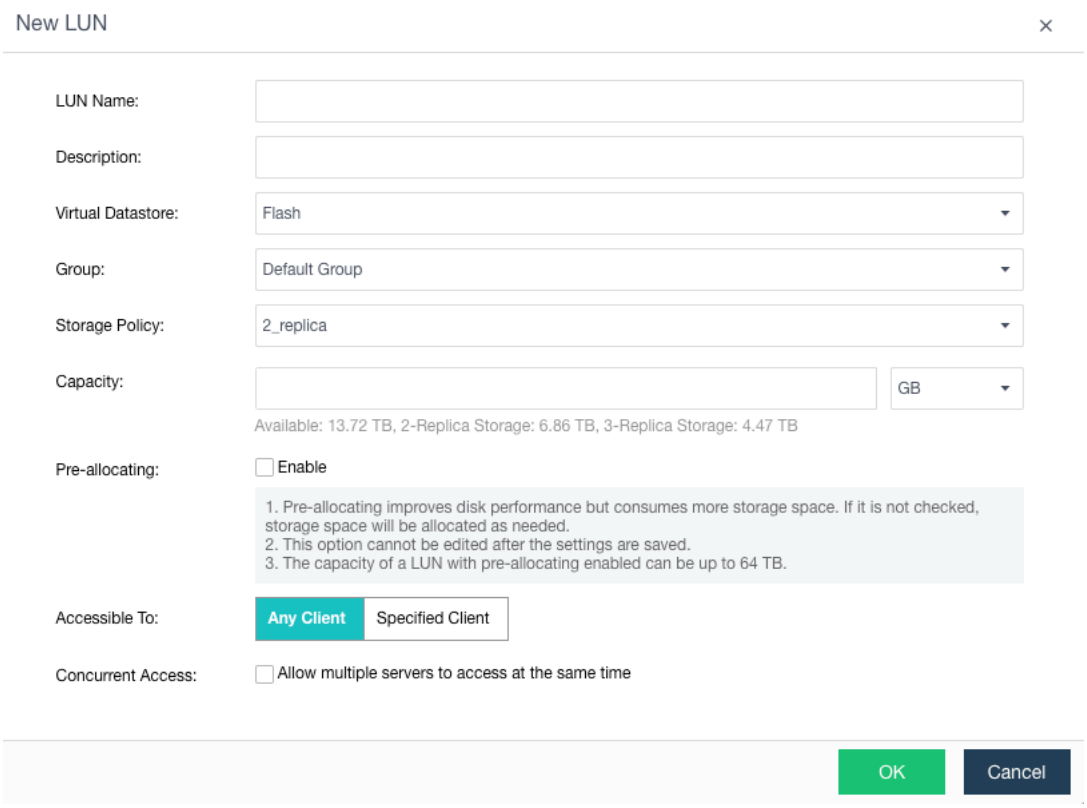


Figure 260: Procedure for Creating Virtual iSCSI Disks

You can use an iSCSI client to discover and connect to the aSAN iSCSI server with a fixed access IP address. The following figure uses the Windows 7 iSCSI Initiator as an example.

The aSAN iSCSI server redirects the requests sent to the access IP address to the virtual IP pool to ensure high availability (no impact from single access point failures) and load balancing.

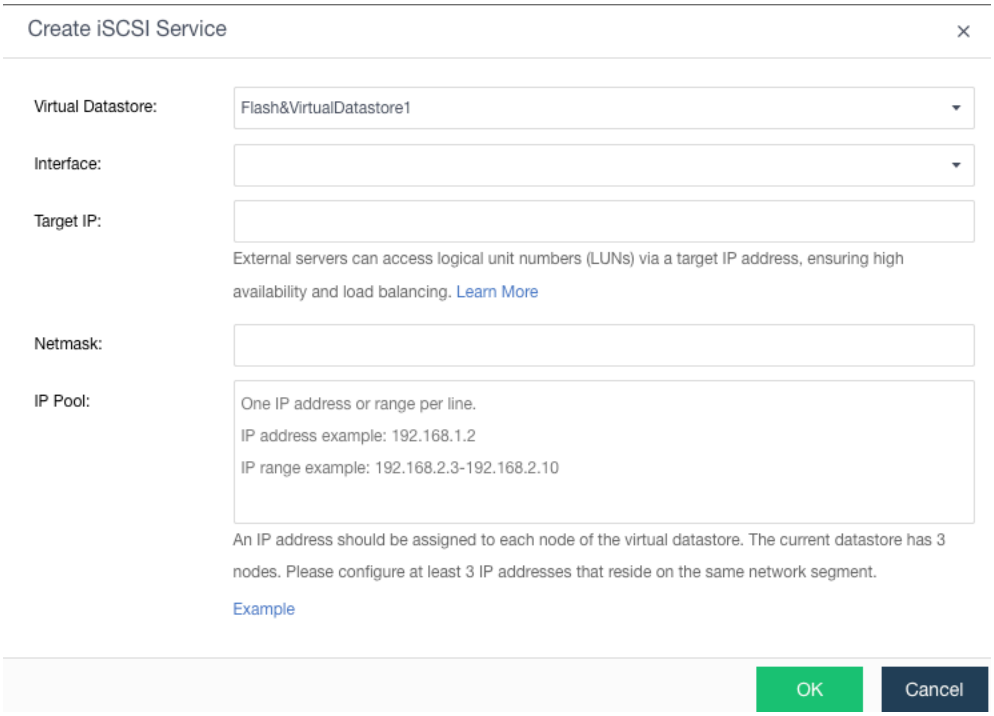


Figure 261: Access IP Settings for the iSCSI Server

You can create multiple virtual iSCSI disks in aSAN for physical servers or other virtualization platforms. The iSCSI clients on the physical servers or virtualization platforms can access the disks via a fixed access IP address provided by the aSAN iSCSI server. When multiple iSCSI clients connect to the aSAN iSCSI server through the fixed access IP address, aSAN automatically assigns a virtual IP address from the virtual IP pool to each host in the virtual datastore and redirects the clients to different hosts. This ensures the following features:

High availability: If any host within the virtual datastore fails, iSCSI connections to that host are redirected to another host to ensure uninterrupted services. As shown in the following figure, when Host C fails, connections from the Linux iSCSI client to Host C are redirected to Host A.

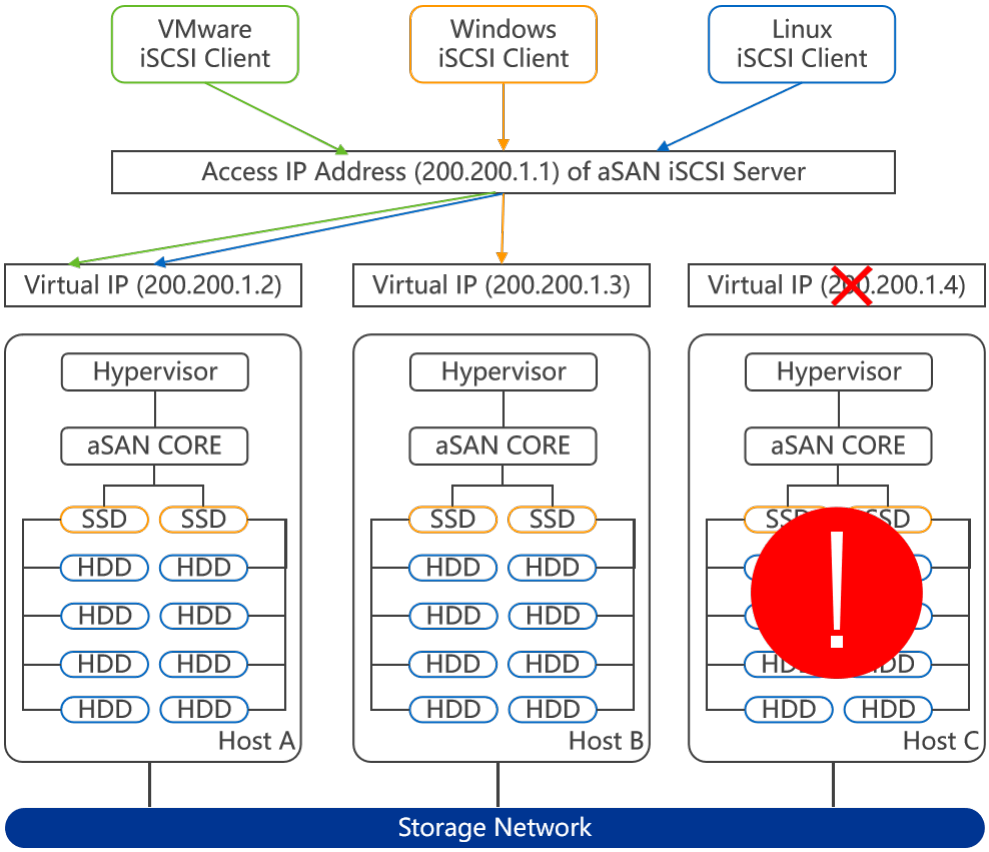


Figure 262: High Availability in iSCSI

Load balancing: When multiple iSCSI clients connect to the aSAN iSCSI server using the access IP address, they are evenly redirected to different hosts. As shown in the following figure, when the three iSCSI clients connect to the access IP address of the aSAN iSCSI server, they are evenly redirected to three different hosts to ensure load balancing.

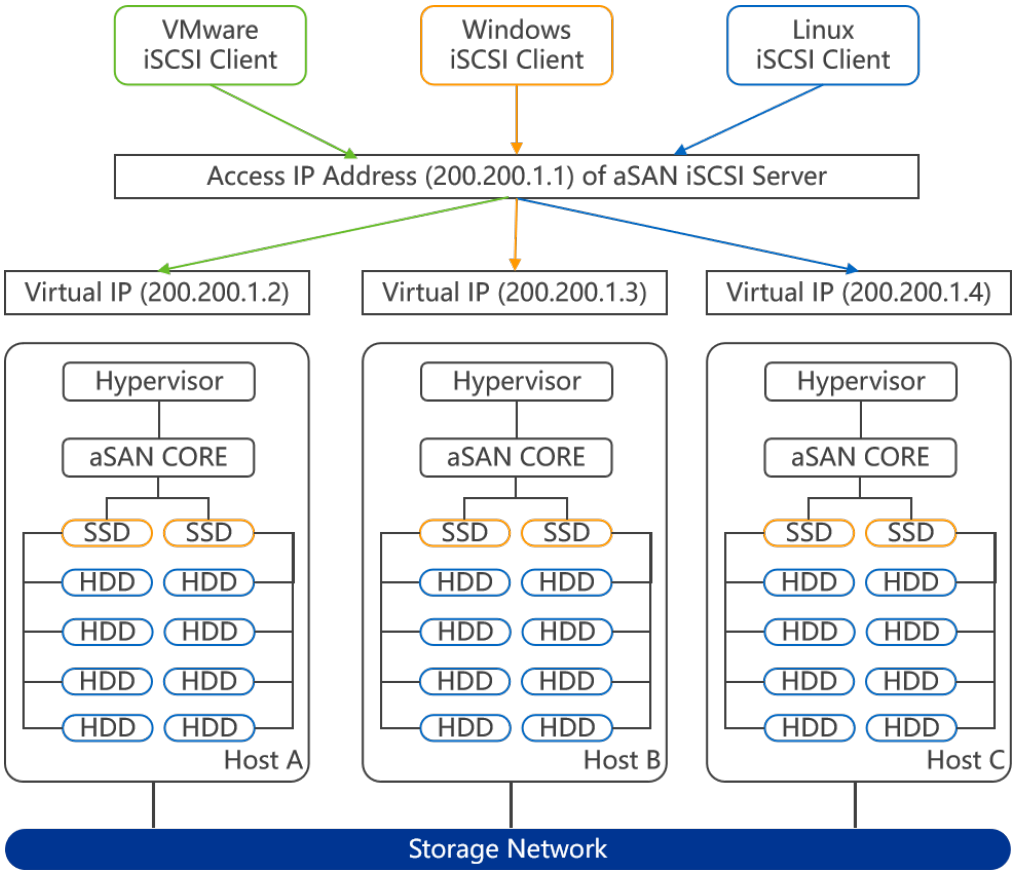


Figure 263: Load Balancing in iSCSI

aSAN's virtual iSCSI disks inherit all the features of aSAN storage, including the multi-replica mechanism, striping, sharding, and cache tier. Put simply, after connecting to aSAN with the access IP address and getting redirected to a host within the virtual datastore, the iSCSI client is like a VM running on that host for read and write access.

NOTICE

You can use aSAN's iSCSI service through the access IP address or the virtual IP pool. Note that the virtual datastore must have at least three hosts.

5.2.6.4 Shared Virtual Disks

Shared virtual disks evolve from aSAN's virtual iSCSI disks and are specifically for Oracle RAC. Using shared disks can significantly reduce the complexity and costs of traditional Oracle RAC deployment.

As shown in the following figure, deploying Oracle RAC in the traditional way requires at least two physical hosts and shared external storage devices such as external iSCSI and FC storage. In HCI, you can replace physical servers with VMs and replace shared external storage devices with shared virtual disks. Mounting the same shared virtual disks on multiple VMs accelerates the deployment of Oracle RAC. In addition, you can efficiently add RAC nodes or increase database storage by adding VMs or shared virtual disks.

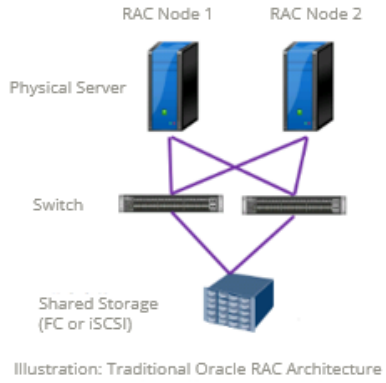


Figure 264: Oracle RAC Deployment

Shared virtual disks and virtual iSCSI disks are essentially the same. The difference is that shared virtual disks are an optimized solution that eliminates the need for configuring an iSCSI server and the access IP address. You can directly mount them on the VMs in the cluster. As virtual disks use the storage network for communication, iSCSI clients on external servers are unable to discover and connect to them.

NOTICE

To use shared virtual disks for deploying Oracle RAC in an HCI cluster, the virtual datastore must have at least three hosts.

5.2.6.5 Virtual Disk Management

HCI supports three virtual disk allocation modes: thin provisioning, pre-allocating, and dynamic provisioning.

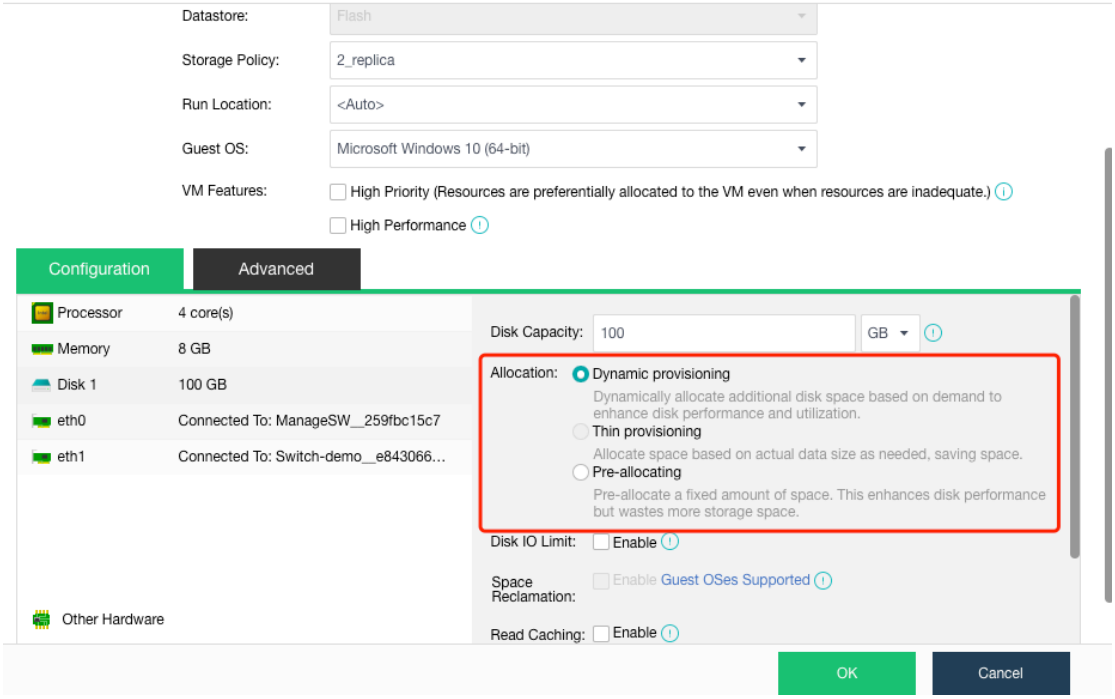


Figure 265: Disk Allocation Modes

- Thin Provisioning

In the thin provisioning mode, space is allocated based on actual usage. This minimizes pre-allocated space, ensuring efficient storage utilization by minimizing pre-allocated space. In this mode, writing data requires writing additional metadata (including index and reference tables), which may lead to I/O amplification and performance decline. Thin provisioning is suited to scenarios with limited storage resources.

- Pre-Allocating

In the pre-allocating mode, space will be allocated based on user-defined disk sizes, which requires more pre-allocated space than thin provisioning. When the allocated disk space is not fully utilized, unused disk resources are wasted. In this mode, write performance is higher as there is no need for writing metadata. Pre-allocating is suited to scenarios with high requirements on I/O performance, such as databases.

- Dynamic Provisioning

As an advanced, intelligent, and efficient capacity allocation and management technology, dynamic provisioning extends storage management by providing

extensive virtual storage for OSes with minimal physical capacity. As the application data volume grows, the actual storage space automatically increases, eliminating the need for manual adjustments.

When you create a disk in the dynamic provisioning mode, the size you specify is a logical virtual capacity rather than a fixed physical space. The system allocates capacity from the physical space based on the predefined policy only when you write data to the logical resource.

In this mode, metadata for data space is pre-allocated. Dynamic provisioning blends the advantages of thin provisioning and pre-allocating, minimizing pre-allocated space while ensuring high write performance.

Summary:

To ensure high performance in the pre-allocating mode, you need to accurately anticipate your current and future business growth and plan space resources in advance. In practice, inaccurate anticipation of the application system scale often results in capacity waste. For example, if you pre-allocate 5 TB of space for an application system that actually only needs 1 TB, you end up wasting 4 TB of space because other application systems are unable to use the space that is already allocated. Even the best system administrators may not be able to allocate storage resources perfectly for applications. According to authoritative industry statistics, about 30% of the total storage is wasted because too much storage space is pre-allocated.

Among the three allocation modes, the dynamic provisioning mode of aSAN effectively tackles the challenge of space allocation in storage resources. It is the optimal mode that ensures both high performance and high resource utilization. This mode is the default mode of HCI.

5.2.7 Stretched Cluster

5.2.7.1 Overview

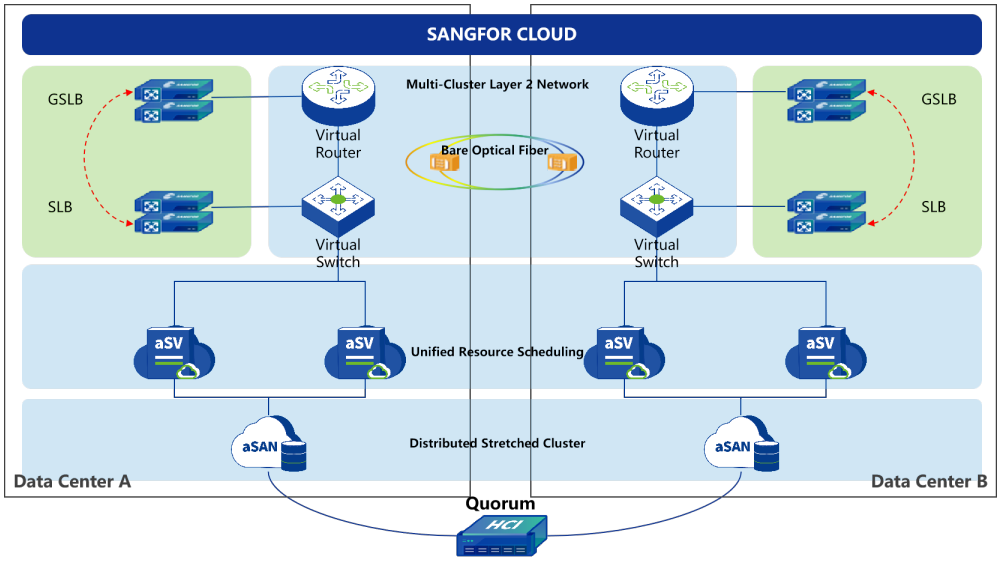


Figure 266: Active-Active Data Center Solution

The active-active data center supports active-active applications, computing, storage, and network connection to ensure high-available business.

1. Active-Active Storage

A distributed storage cluster is built on hyper-converged x86 servers. It supports data replicas to synchronize business data across data centers. The storage cluster ensures no data loss, achieving an RPO of 0 and an RTO of several seconds. Failover is automatically triggered in the event of a data center failure.

2. Active-Active Cloud Platform

The cloud platform enables distributed cluster deployment across data centers based on the Layer 2 network connectivity between the data centers for a large cluster. It offers high availability for the quick recovery of the business in the failed data center.

3. Network Connectivity

The active-active data centers are connected via bare optical fibers, with a

latency of less than 1 ms and a bandwidth of no less than 10 GE between them.

4. Quorum Center

The third-party quorum center allows you to deploy witness nodes for the active-active data centers. A witness node must have at least 8 vCPUs, 16 GB of memory, and 200 GB of SSD storage. The quorum center is connected with the active-active data centers via Layer 2 or 3 network, with a network latency of less than 5 ms and a network bandwidth of no less than 100 MB.

5. Optimized Access

GSLB and SLB work together to flexibly schedule business traffic between the active-active data centers.

5.2.7.2 Technical Principles

Sangfor's distributed storage technology ensures the data security of the active-active data centers by combining redundant replicas with witness nodes. It allows you to configure data protection policies with a minimum of two or three replicas. An HCI stretched cluster requires at least two hyper-converged primary-site nodes, two hyper-converged secondary-site nodes, and one hyper-converged witness node and supports asymmetric scaling of the active-active data centers. The active-active storage solution of Sangfor HCI stretched cluster achieves an RPO of 0 and an RTO of several seconds in the event of data center failures. When a data center fails, businesses running in the stretched cluster can seamlessly use the replicas stored in the other data center. This ensures high availability, live migration, and failover of VMs across sites.

The replicas of business data are written to virtual datastores. When you create an HCI stretched cluster, the replicas of the data from the businesses running in the cluster are synchronously written to both data centers. To ensure the consistency of replicas, a write request is deemed complete only after both data centers confirm that the replicas are successfully written, and the next write request starts only after the current write request is complete. The local replica is accessed when the business works properly, and the replica in the

other data center is accessed when the local replica is unavailable. Therefore, when a data center fails, you can recover the failed VM in the stretched cluster to the other data center using the HA feature and access the second replica to ensure business continuity.

A witness node is a witness replica for storing data addresses. You can deploy a witness node on a VM on VMware vCenter or a physical machine. If possible, do not deploy the witness node in the two fault domains.

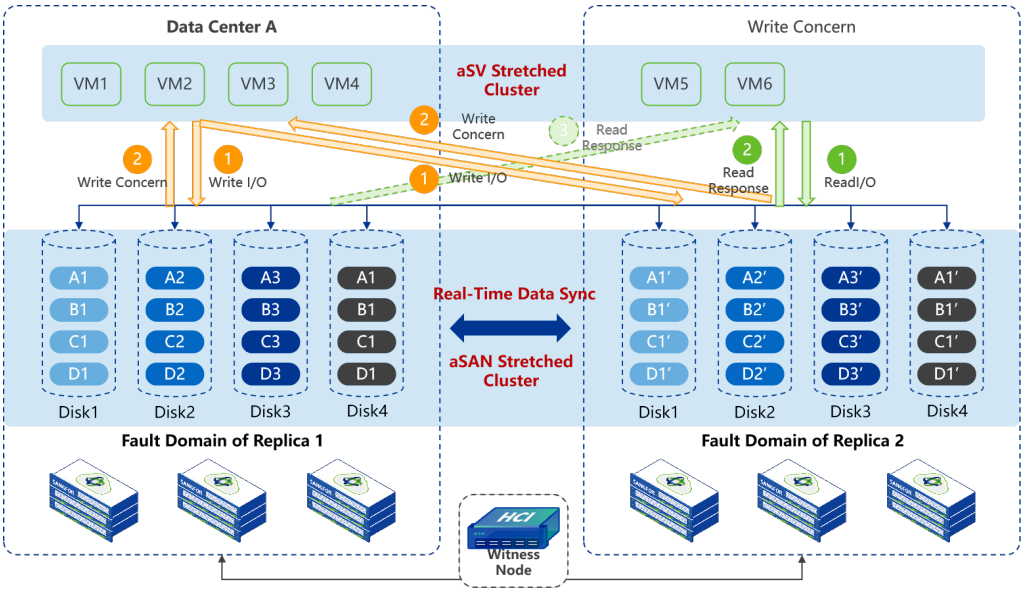


Figure 267: Stretched aSAN Cluster

aSAN supports two-replica and three-replica stretched clusters. Both of them can tolerate fault domain-level failures. When a fault domain fails, the stretched cluster can recover the business to the other fault domain to ensure the high availability of business data.

5.2.7.3 Two-Replica Stretched Cluster

Data in a two-replica stretched cluster has two regular replicas and one witness replica. aSAN employs a data distribution policy to ensure that the two regular replicas are written to two different fault domains while the witness replica is written to the witness node, as shown in the following figure.

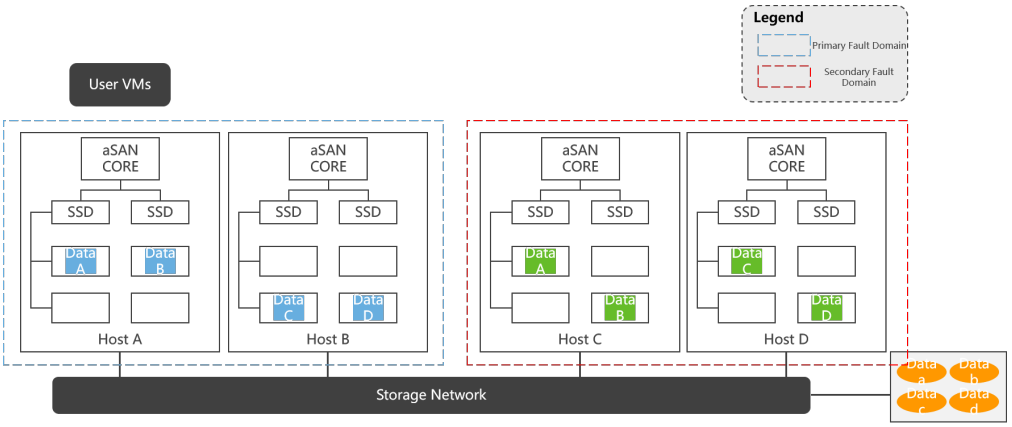


Figure 268: Two-Replica Stretched Cluster

As shown in the following figure, in the event of a failure in the primary fault domain, there are still two replicas available in the cluster (one in Host C and one in the witness node). As the number of the remaining replicas exceeds half of the total of the three, the data is still available. VMs running in the primary fault domain are recovered to the hosts of the secondary fault domain for the high availability of business data.

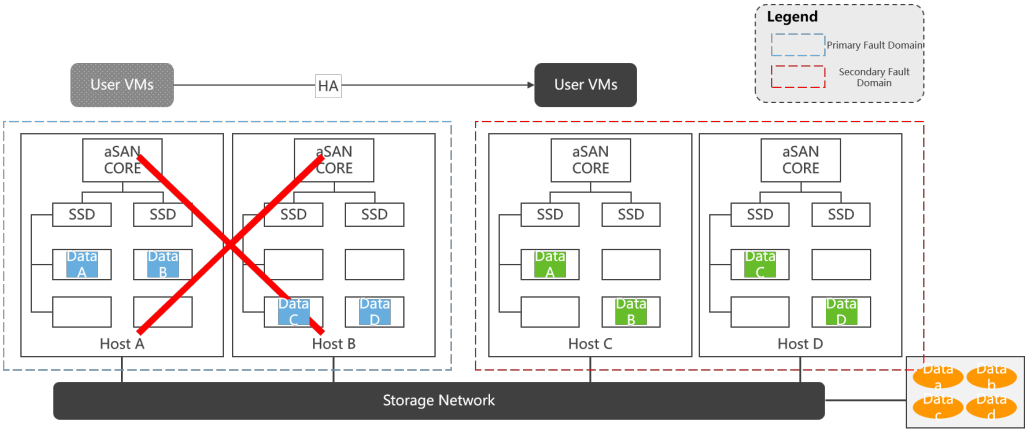


Figure 269: Cluster Failure in a Two-Replica Stretched Cluster

5.2.7.4 Three-Replica Stretched Cluster

Data in a three-replica stretched cluster has three regular replicas and two witness replicas. One fault domain stores two regular replicas, the other fault domain stores one regular replica and one witness replica, and the witness node stores one witness replica.

The data distribution in a three-replica stretched cluster is shown in the following figure.

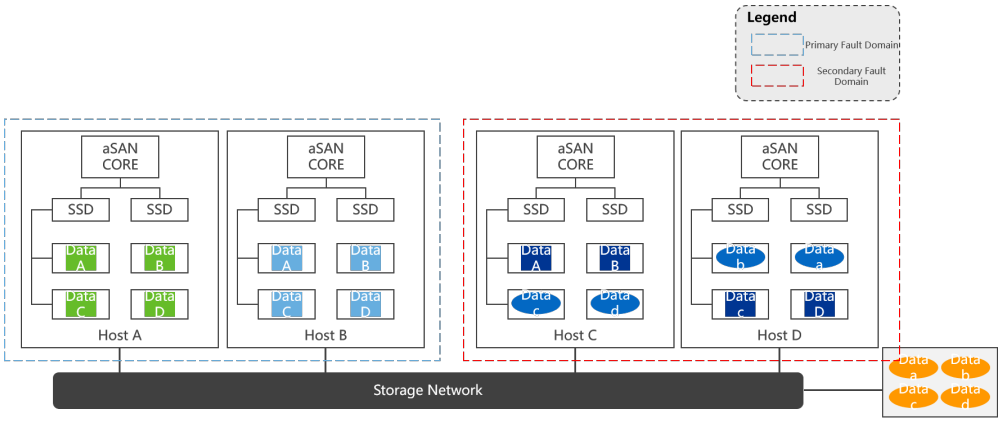


Figure 270: Three-Replica Stretched Cluster

A three-replica stretched cluster can also tolerate fault domain-level failures. As shown in the following figure, in the event of a failure in the primary fault domain, there are still three replicas available in the cluster (Host C, Host D, and the witness node each have one replica). As the number of the remaining replicas exceeds half of the total of five, the data is still available. VMs running in the primary fault domain can be recovered to the hosts of the secondary fault domain to ensure the high availability of business data.

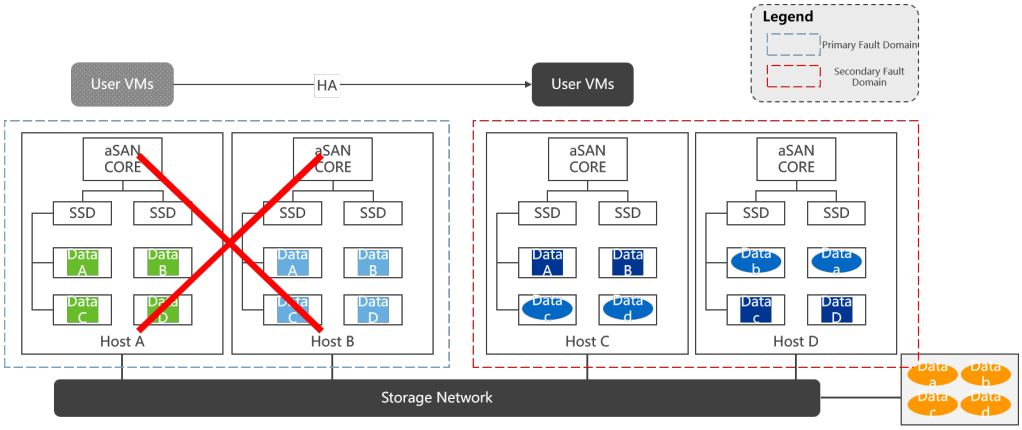


Figure 271: Cluster Failure in a Three-Replica Stretched Cluster

Fault tolerance across fault domains:

A three-replica stretched cluster can tolerate disk or host failures across fault domains.

Scenario	Three-Replica Stretched Cluster	Two-Replica Stretched Cluster
One faulty disk in each fault domain	No data loss and no business interruption.	Business is interrupted due to data loss.
One faulty disk in the fault domain where the VM resides	The VM still runs in and accesses data from the fault domain with two replicas.	The VM must access the remote replica in the other fault domain.

Table 14: Scenarios of Replica Failures Across Fault Domains

In scenarios where the primary and the secondary fault domains both have one faulty disk, data loss occurs in a two-replica stretched cluster because both of the replicas are inaccessible. However, in a three-replica stretched cluster, data will not be lost and business is not interrupted because there is still one replica working in the primary fault domain. As shown in the following figure, although two replicas marked as data A are lost when there is one faulty disk in each fault domain, business is not interrupted because another replica marked as data A in the primary fault domain is still available.

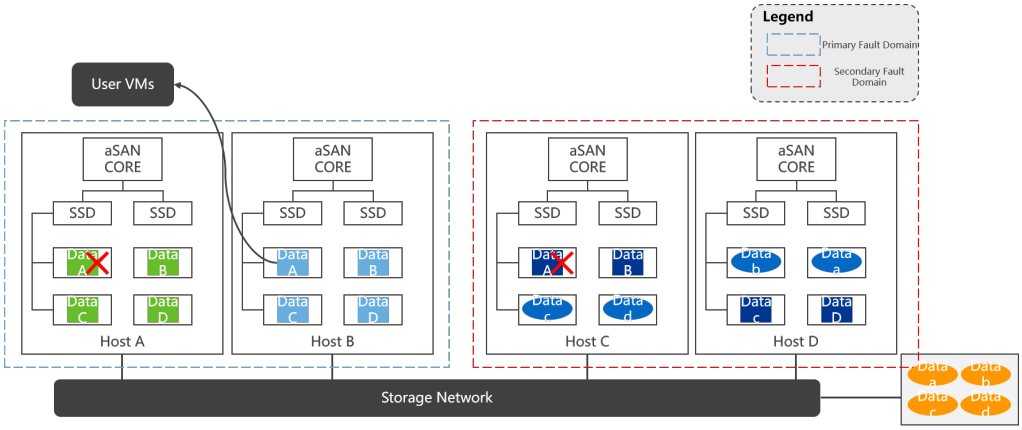


Figure 272: Two-Disk Failure in a Three-Replica Stretched Cluster

When there is one faulty disk in the fault domain where the VM in a two-replica stretched cluster resides, the VM must access data from the remote fault domain. When there is one faulty disk in the primary fault domain where the VM in a three-replica stretched cluster resides, the VM can access the other

replica from the primary fault domain. This improves read performance during disk failures.

5.3 aNET

5.3.1 Overview

aNET manages the configurations of cluster virtual networks and forwards data to ensure communication between virtualization resources. It implements an SDN architecture that focuses on virtual networks based on general SDN principles and host overlay networks.

aNET consists of a management plane, a control plane, and a data plane. This design separates the management plane from the control plane and offers horizontal scalability. aNET ensures high availability based on technologies such as clustering and persistence and provides high performance and scalability based on technologies such as stateless service and external caching.

5.3.1.1 SDN

SDN is an OpenFlow-based network architecture that separates the control plane from the data plane. Compared to the physical network architecture of on-premises data centers, SDN features the separation between the control plane and the data plane, centralized network control, and open APIs. It implements centralized network control, uniformizes protocols, and standardizes operational instructions.

SDN features that are widely acknowledged in the industry include:

1. Separation between the control plane and the data plane
2. Open APIs
3. Centralized network control

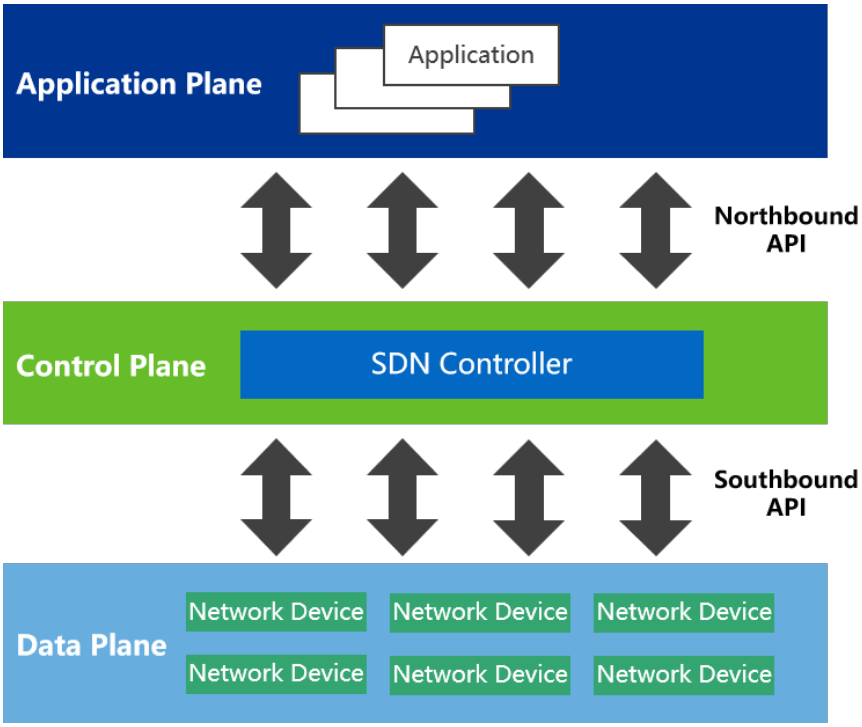


Figure 273: Standard SDN Architecture

The separation between the control plane and the data plane decouples the network element controllers from the data plane. The control plane consisting of network element controllers interprets protocols and generates flow tables, while the data plane only includes network devices. Centralized network control refers to managing and distributing flow tables using controllers. It eliminates the need to operate devices one by one, and you only need to configure controllers. Open APIs allow third-party applications to define new network features and run the features on the controllers.

Generally, aNET is an SDN architecture because it has implemented SDN and has the three core SDN features: separation between the control plane and the data plane, centralized network control, and flexible open APIs.

5.3.1.2 Overlay Technologies

An overlay network is a network built on top of existing network architecture using network virtualization technologies. It allows you to run applications in virtual networks and separate businesses without the need to make great

changes to the underlying network.

An overlay network consists of three elements: edge devices, the control plane, and the data plane.

- **Edge devices**: Devices that are directly connected to VMs
- **Control plane**: Responsible for establishing and maintaining virtual tunnels and broadcasting host reachability information
- **Data plane**: The physical network that processes packets from overlay networks

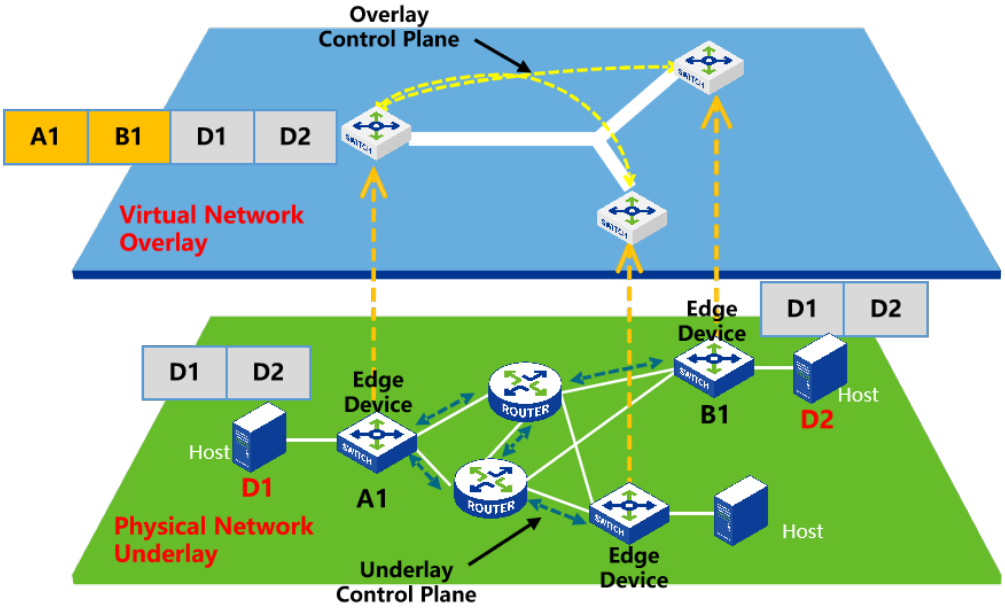


Figure 274: Overlay Network Architecture

There are three common overlay technologies: VXLAN, NVGRE, and STT. Although the three technologies send Ethernet packets to a tunnel plane based on the same underlying IP forwarding, they select and create different tunnels. VXLAN and STT have lower requirements on the load balancing capability of existing network devices. Typical network devices can balance Layer 2-4 traffic through techniques like link aggregation or equal-cost multi-path routing based on the data's content and parameters. However, VXLAN has become the most popular overlay technology because it uses the mature and widely adopted UDP for data transmission.

Overlay technologies bring the following benefits to data centers after server

virtualization:

1. Packets sent from VMs are encapsulated within IP packets, bypassing the size limit on MAC address tables and enabling large-scale VM deployments in data centers.
2. Enhanced isolation capabilities allow you to isolate numerous tenants in the multi-cluster Layer 2 network. Theoretically, you can create up to 16 million VXLANs while using VXLAN network identifiers (VNI).
3. Ethernet packets are encapsulated within IP packets and transmitted through network routing, allowing you to migrate VMs across a broader range without being constrained by the Layer 2 physical network.

Overlay technologies support the following three modes:

- **Network overlay**: Overlay networks are built and extended based on control protocols, with hardware network devices such as routers and switches acting as their edge devices. In this architecture, physical machines and virtualization platforms can access the network.

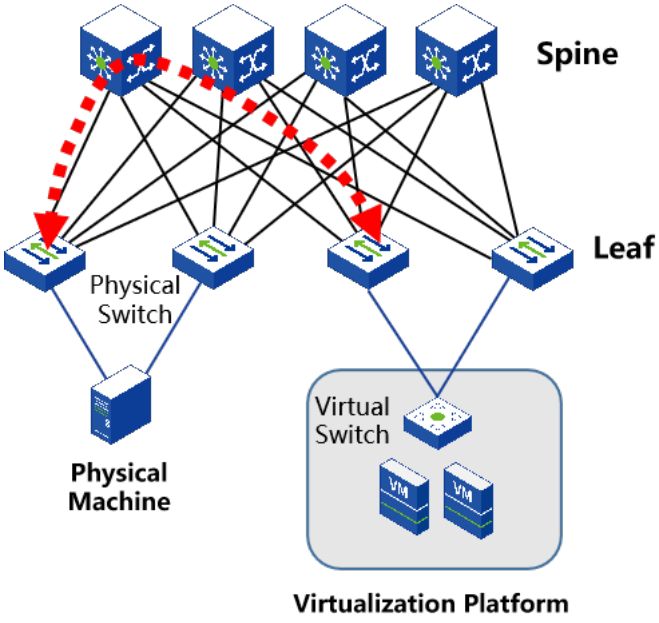


Figure 275: Network Overlay Architecture

- **Host overlay**: Overlay networks are built and extended based on control protocols, with virtual switches on the virtualization platform acting as their edge devices. In this architecture, VMs can access the network.

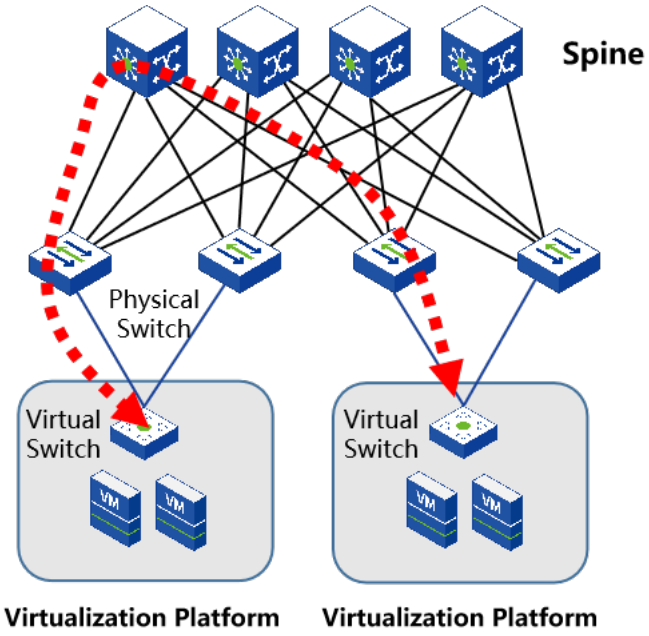


Figure 276: Host Overlay Architecture

- **Hybrid overlay**: Overlay networks are built and extended based on control protocols, with both hardware and virtual network devices acting as their edge devices. In this architecture, both physical machines and VMs can access the network.

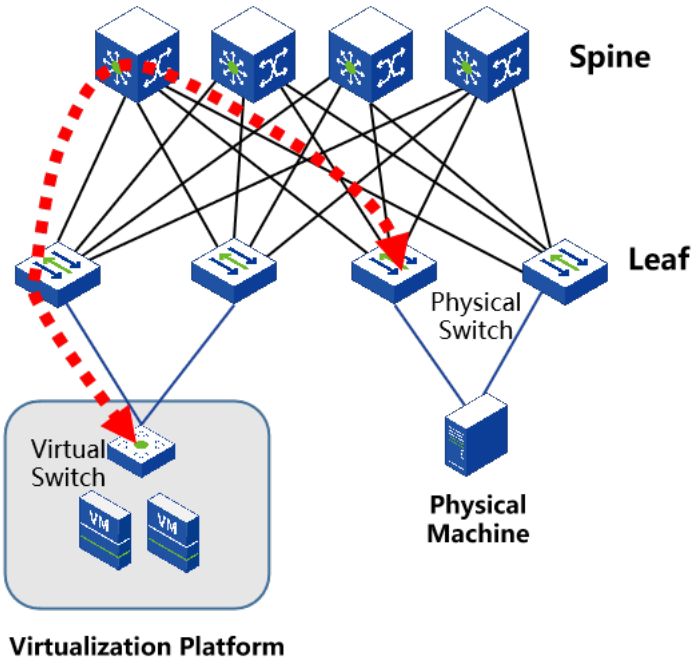


Figure 277: Hybrid Overlay Architecture

5.3.1.3 VXLAN Technology

VXLAN:

IETF defines network virtualization over Layer 3 (NVO3) as a general term for technologies that overlay virtual networks over Layer 3. VXLAN is a standard NVO3 technology. It encapsulates Layer 2 packets with Layer 3 protocols in the Layer 2 over Layer 4 (MAC-in-UDP) mode and forwards the encapsulated packets through tunnels. A typical VXLAN packet consists of, from the outermost to the innermost, outer MAC header, outer IP header, UDP header, VXLAN header, and the original Layer 2 frame.

- The outer MAC header contains 14 bytes in total, including the MAC addresses of the source VXLAN tunnel endpoint (VTEP) and the next-hop device, as well as the VLAN type and VLAN tag for identifying the VLAN type and ID.
- The outer IP header contains 20 bytes in total, mainly containing the IP addresses of the VTEPs to which the source and destination VMs belong. This header is used for identifying the VMs' VTEPs in Layer 3 routing.

- The UDP header contains 8 bytes in total, including the VXLAN port fixed at 4789 and the UDP source port calculated from the original Ethernet frame using a hashing algorithm.
- The VXLAN header contains 8 bytes in total, including a 24-bit VNI for identifying isolated networks.
- The original Layer 2 frame is the original packet that the VM truly observes, mainly including the IP addresses of the source and destination VMs and the payload.

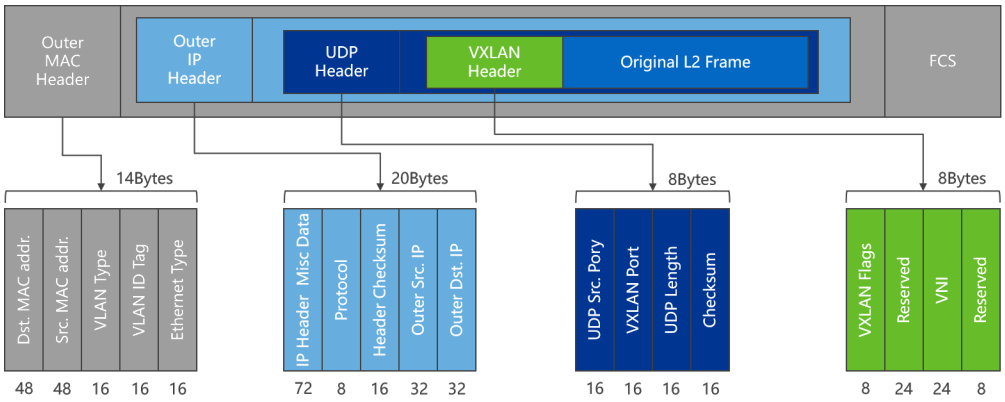


Figure 278: VXLAN Packet Format

VTEP:

VTEPs are the edge devices in a VXLAN that serve as the starting and ending points of VXLAN tunnels for encapsulating and decapsulating VXLAN packets. Both physical and virtual network switches can serve as VTEP devices. The original frames sent from the VMs in a host are forwarded in the IP network after they are encapsulated with the VXLAN header, UDP header, outer IP header, and outer MAC header in order in the virtual network. The destination TOR switch decapsulates the received VXLAN packets and sends the original packets to the destination VMs.

In a host overlay scenario, virtual switches serve as the VTEPs of the VXLAN tunnel for packet encapsulation and decapsulation.

VNI:

A VNI, similar to a VLAN tag, is used to distinguish VXLAN segments. A VNI is 24 bits in size. VMs with the same VNI can communicate directly without a

gateway, while cross-VNI communication requires VXLAN L3 gateways.

5.3.1.4 aNET

aNET manages the configurations of cluster virtual networks and forwards data to control communication between virtualization resources. It implements an SDN architecture that focuses on virtual networks based on general SDN principles and host overlay networks. aNET consists of a management plane, a control plane, and a data plane. This design separates the management plane from the control plane and offers horizontal scalability. aNET ensures high availability based on technologies such as clustering and persistence and provides high performance and scalability based on technologies such as stateless service and external caching.

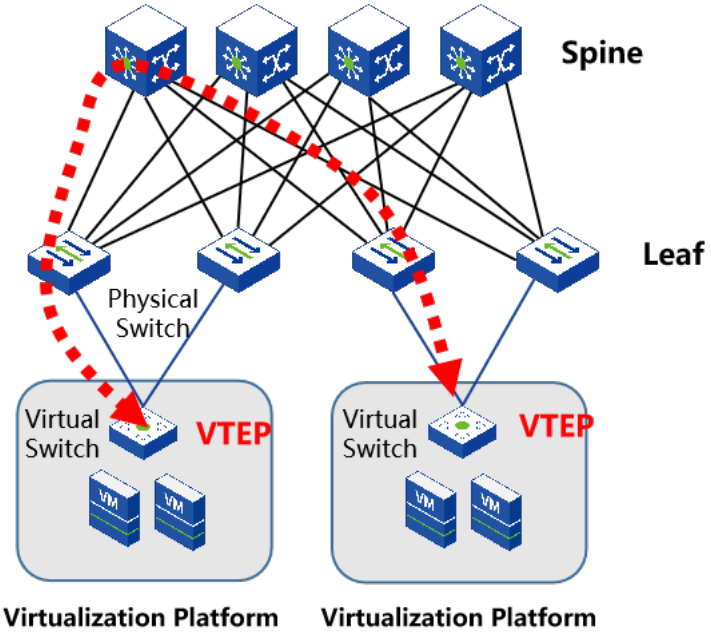


Figure 279: aNET Network

5.3.1.5 Multi-Cluster Layer 2 Network

For medium to large-scale cluster deployments, Sangfor has enhanced HCI's network architecture to support the following specific scenarios:

1. New data centers with 24-100 nodes

2. Geo-redundancy disaster recovery
3. Clusters with heterogeneous servers (Arm and x86)
4. IaaS resource isolation
5. Cluster scalability and cross-cluster network connectivity
6. Large-scale host management via SCP in cluster mode

Sangfor HCI enables peer-to-peer communication between different clusters using VXLAN. It supports cross-cluster network connectivity between VPCs, between VPCs and physical networks, and between physical networks. The implementation is shown in the following figure.

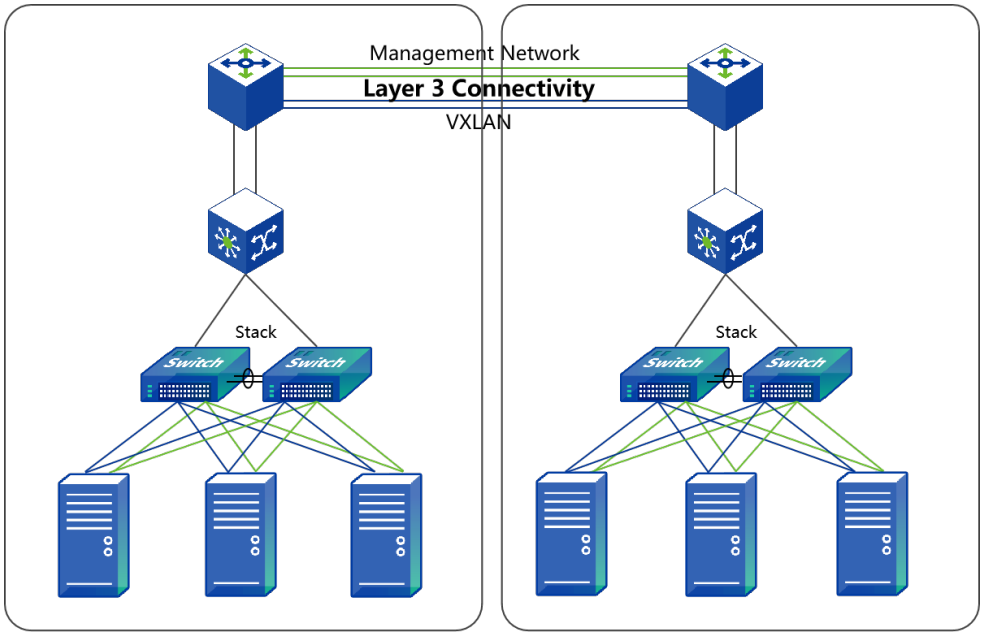


Figure 280: Network Connectivity Between HCI Clusters

Layer 2 Network Connectivity:

Layer 2 networks between clusters or sites are connected via VXLAN. As Layer 2 packets are forwarded after they are encapsulated within Layer 3 packets, you do not need to configure STP or proprietary protocols to prevent loops.

Physical networks are isolated between sites so that network failures (for example, broadcast storms and loops) at a single site do not extend to other sites.

The network management feature allows you to customize classic multi-cluster

networks and create cross-cluster tenant networks on each canvas.

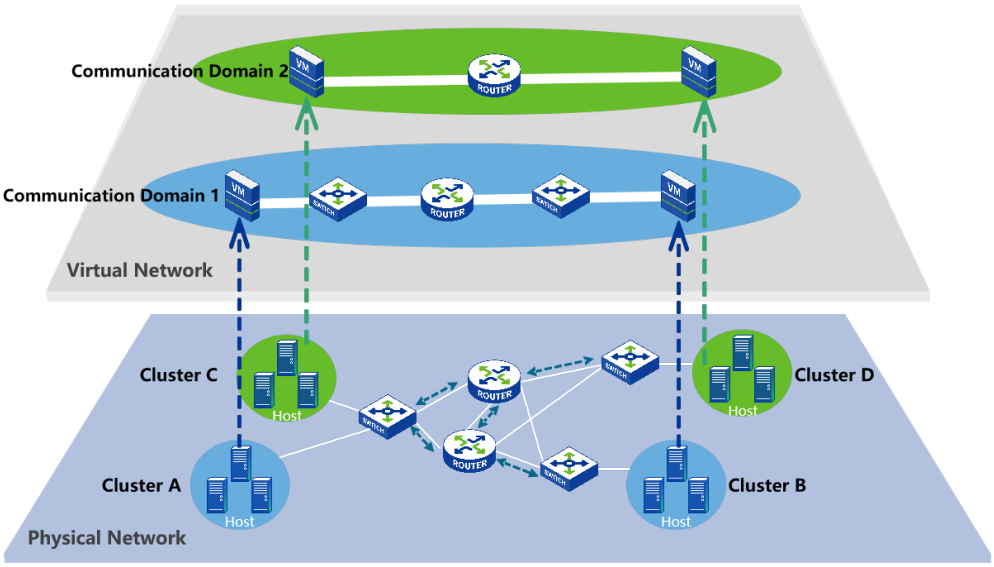


Figure 281: Communication Domains Across HCI Clusters

Communication Domain:

Grouping two clusters into a communication domain requires a connected management network between them. Due to the use of VXLAN gateways, you only need to ensure that the physical network for the VXLAN is reachable at Layer 3.

Communication domains allow you to connect classic networks or tenant networks across clusters. You can also orchestrate networks in the classic network or VPC view on SCP.

5.3.2 Functional Architecture

5.3.2.1 Logical Architecture of aNET

aNET consists of management planes, control planes, and data planes. These planes communicate with each other through standardized and decoupled interfaces. aNET also provides diverse O&M capabilities.

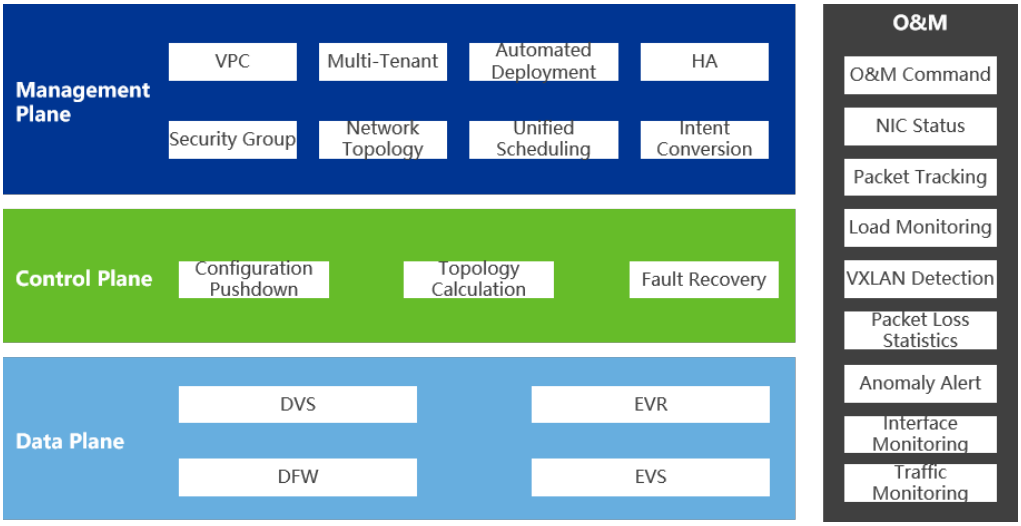


Figure 282: Functional Architecture

To help you understand the architecture model and data flow of management, control, and data planes, the following figure shows an example of a three-node HCI cluster.

HCI provides external network RESTful APIs only in management planes for receiving operational instructions from clients. The control plane distributes the configurations transformed by the management plane, and the data plane executes these instructions.

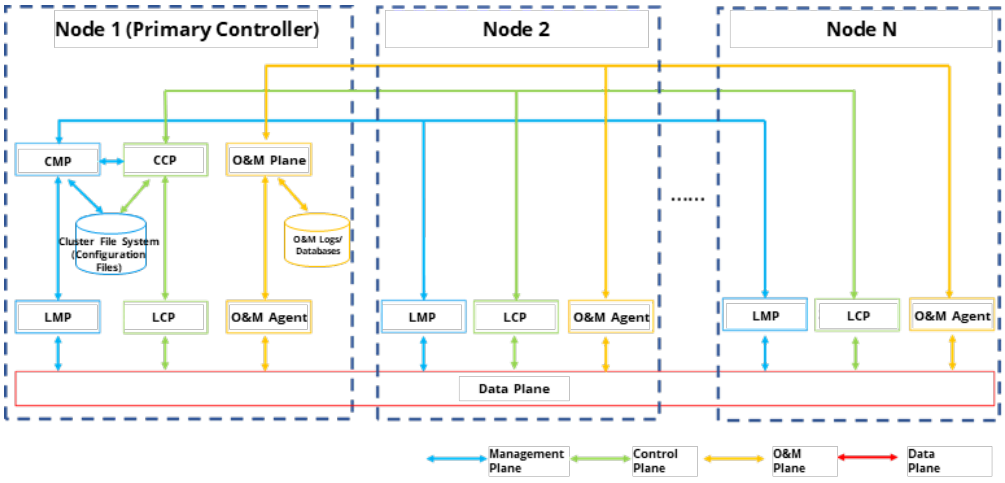


Figure 283: Network Communication

Management Plane:

The external network RESTful APIs of HCI work only in management planes.

There are two types of management planes: central management plane (CMP) and local management plane (LMP). The CMPs run on the primary controller, while LMPs, which are responsible for synchronizing configurations and monitoring status (such as heartbeats), run on all hosts.

1. Simplified O&M

- Management planes allow you to drag and drop components, which is an easy way to operate and maintain virtual network topologies.
- Management planes can manage the full lifecycle of basic virtual network components, including distributed switches, virtual routers, distributed firewalls, and edge virtual switches.
- Management planes can centrally manage Sangfor's NFV security device components based on features such as automated deployment and full lifecycle management.

2. High availability

Management planes employ a high-availability mechanism. When the primary controller fails, the cluster automatically elects a new one and recovers the corresponding management plane on it to continue managing control and data planes. In addition, management planes use the cluster file system, which adopts the multi-replica mechanism to ensure that each node stores one replica. When the primary controller fails and cannot be recovered, the recovered aNET management plane can obtain configurations from the cluster file system.

Control Plane:

There are two types of control planes: the Sangfor-developed central control plane (CCP) and local control planes (LCP). The CCP runs on the primary controller. It distributes the configurations transformed by the CMP and synchronizes configurations to CCPs running on other nodes. The CCP is stateless. If only one cluster exists, the CCP runs on the only primary controller. If multiple clusters exist, the CCP runs on all primary controllers. MySQL and Redis are stateful. If there is only one cluster, MySQL runs in active-standby mode while Redis runs with a single instance; in multi-cluster scenarios, MySQL

runs in active-standby mode while Redis runs in primary-secondary mode.

The Raft consensus protocol is used to guarantee configuration consistency. The controller cluster can still work properly when a controller node fails. In this case, the cluster elects a new controller node to ensure that there are always three controller nodes available in the cluster. If the primary controller fails, the control plane, just like the management plane, elects a new primary controller by reusing the cluster module and recovers the CCP on the new primary controller. Through the proactive reporting feature in network proxy modules on compute and storage nodes, the CCP regains the real-time status of these nodes, guaranteeing the control plane's high availability.

Data Plane:

Generic Linux OSes give network and non-network applications the same priority, which cannot ensure high-performance network I/O throughput. The data plane of aNET is designed based on DPDK. Sangfor has optimized the data plane for network I/O-intensive applications to enable high-performance network packet forwarding.

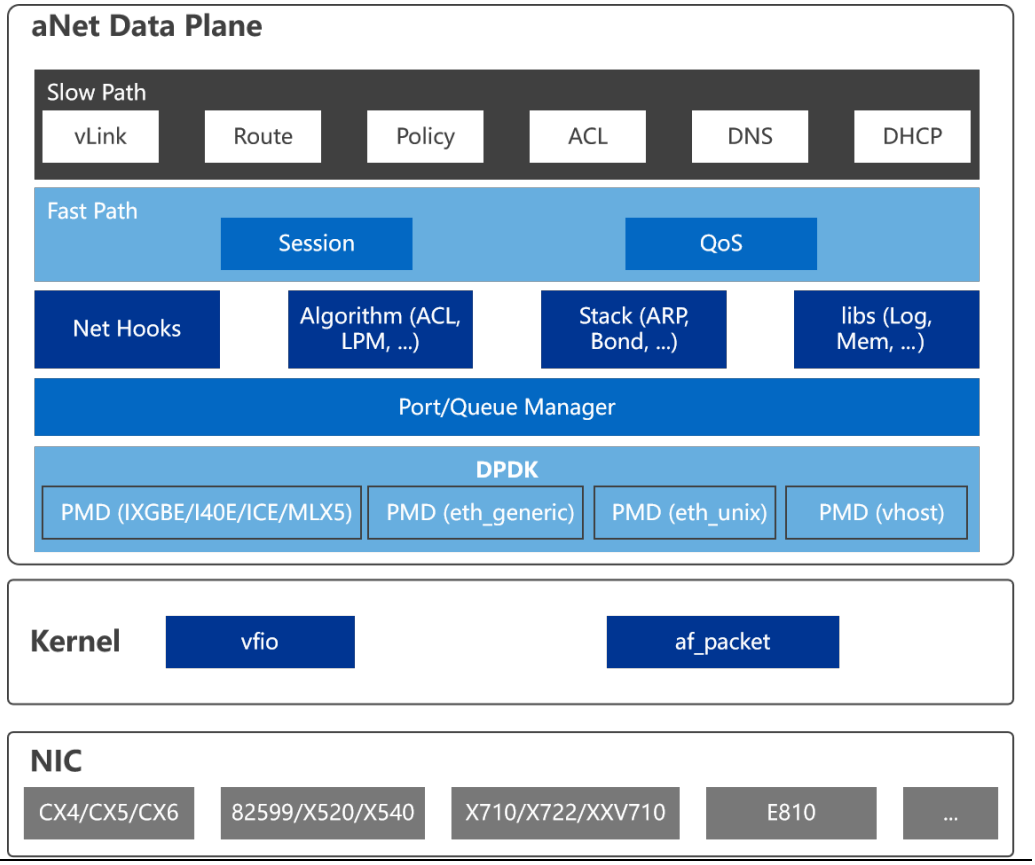


Figure 284: Data Plane Architecture

The data plane of aNET has the following strengths:

Support for low-latency Intel and Mellanox NICs and generic NICs:

For high-performance 10-GB Intel and Mellanox NICs, aNET offers a high-performance DPDK poll mode driver (PMD) solution. For generic NICs, aNET provides a standard solution to ensure hardware compatibility.

The high-performance DPDK PMD solution consists of userspace APIs and PMD. The kernel-mode UIO driver shields the interruption signals sent by NICs, and the userspace PMDs perform polls. The polling mode is suitable for all receiving and sending operations, except for link status notifications, which rely on interrupt signals. This greatly increases the network packet processing capacity of applications.

DPDK is integrated into vector packet processing (VPP, an open-source, high-performance packet processing stack) as a plugin, while the dpdk-input module is integrated into VPP's forwarding graph in the form of a node. The dpdk-input node can select strategies each time it receives a packet.

- If the strategy selection module opts for the interrupt mode, it calls the DPDK API to turn on the interrupt mode and enters epoll sleep, waiting for the next packet.
- If the strategy selection module does not opt for the interrupt mode, packets are forwarded in the default polling mode.

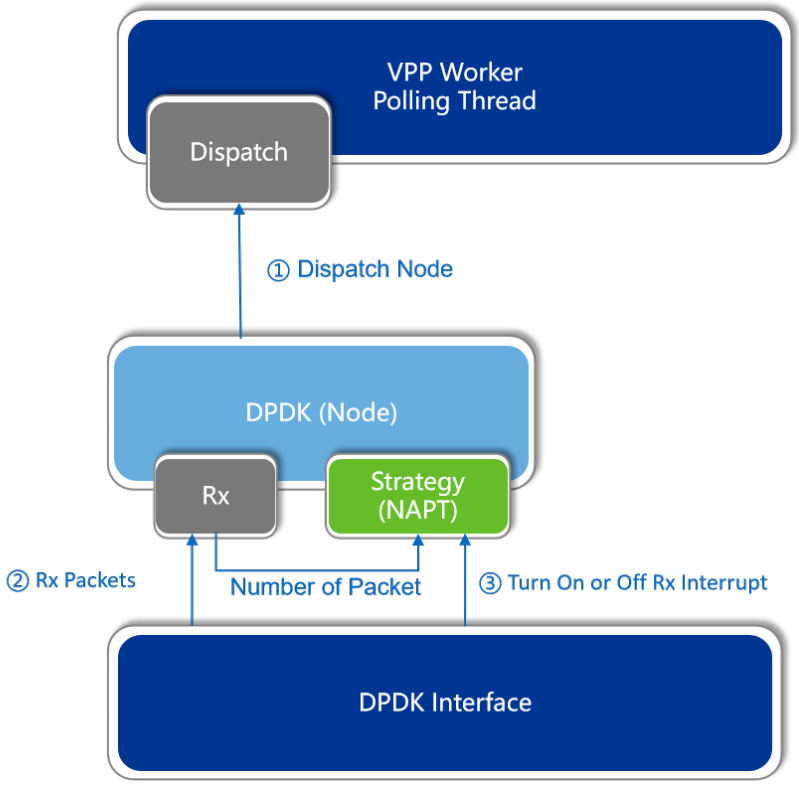


Figure 285: High-Performance DPDK PMD Solution

The scheduling process is as follows:

1. The dpdk-input node is scheduled to receive packets.
2. The dpdk-input node receives packets and sends the number of received packets to the strategy selection module.
3. If the strategy selection module opts for the interrupt mode, it turns on the interrupt mode and enters epoll sleep.

Strategy selection supports both the high-performance PMD solution for high-performance NICs and the standard solution for generic NICs, significantly improving overall compatibility.

Enhanced stability in the data plane by rewriting and simplifying the forwarding flow at the application layer:

1. A minor kernel-mode issue can lead to system crashes, and the worst case at the application layer is unexpected process termination. To address this, aNET introduces a monitoring mechanism that can recover the

network within seconds without affecting VMs even when processes unexpectedly die.

2. The data plane is responsible for packet forwarding and is the core of the entire system. It consists of a control thread and multiple data forwarding threads. The control thread receives configuration messages from the control process, while the data forwarding threads handle packets.
3. The data forwarding threads handle packets by separating fast and slow paths. Packet forwarding is based on sessions, where each flow matches a session. The first packet in a flow looks up table entries, creates a session, and records the lookup results within the session. Subsequent packets in the flow look up the session and are processed and forwarded based on the information from the session.
4. The data forwarding threads receive all packets in the system. Packets to be forwarded do not need to pass through the Linux protocol stack. They are processed by the data forwarding threads and then sent from NICs. However, packets destined for devices (such as SSH, Telnet, OSPF, BGP, or DHCP packets) are routed through the TUN interface to the Linux protocol stack as the data forwarding threads cannot process them directly. After being processed by and sent from the Linux protocol stack, these packets must pass through the data forwarding threads before being sent from NICs.
5. The data plane enables low-level hardware interactions for packet processing, while the application-layer protocol stack offers an optimized implementation of the network stack. Compared to Linux SMP, aNET's data plane achieves superior scalability and stability by reducing its dependency on the Linux kernel.
6. Unlike cloud platforms that run at the kernel level, the data plane runs at the application layer. This prevents data plane anomalies from causing kernel crashes. In the event of failures, services are restarted to rapidly recover the data plane, which greatly reduces the data plane's impact on the platform's reliability. The data plane supports switchover within a host, where the secondary process retains all data plane configurations. When

the primary process exits unexpectedly, the secondary process promptly takes over the data forwarding business as the new primary process, ensuring uninterrupted business and a highly reliable data plane on a single host.

Separation between fast and slow paths:

1. The data forwarding architecture separates fast and slow paths. The lightweight fast path ensures high forwarding speeds with simple business logic, while the slow path handles complex business logic such as routing, DNS, and DHCP.
2. The fast path adopts session-based data forwarding. When a flow (a packet's 5-tuple) matches the table entry in the session, the corresponding action (such as packet forwarding and NAT) is performed. For the first packet in a flow, the data forwarding thread looks up table entries, creates a session, and records the lookup results within the session. For subsequent packets in the flow, the data forwarding thread uses the hash algorithm to find the session and handles the packets based on the information it gets from the session.
3. The data plane consists of a control thread and multiple data forwarding threads. The control thread receives the upper-layer configurations, while the data forwarding threads handle packets. The data plane has also introduced a variety of optimization techniques, including CPU cache prefetching, interrupt moderation, batch processing, TCP segmentation offload, and hardware checksum offload capabilities.
4. The data plane communicates with the Linux protocol stack through high-performance virtio-user or vhost-net. Packets destined for devices (such as SSH or Telnet packets) are redirected through a TAP to the Linux protocol stack. After being processed by and sent from the Linux protocol stack, these packets must pass through the data forwarding threads for processing before being sent from NICs.

O&M:

1. Support for active upgrade. The data plane adopts a primary-secondary DPDK architecture where DPP acts as the primary process and Dataplane

acts as the secondary process (as shown in the following figure). DPP is responsible for the initialization and management of physical interfaces, while Dataplane, the business forwarding process, is responsible for packet transmission and reception on physical interfaces, as well as packet processing and forwarding. During an active upgrade, only the Dataplane process is upgraded to minimize network downtime by avoiding physical interface reinitialization and link negotiation; the vhost connection status is also synchronized to the new Dataplane process simultaneously to avoid vhost disconnection and reconnection; forwarding sessions are shared in the form of memory files to eliminate the extra overhead of synchronizing sessions to the new Dataplane process; the network downtime during an active upgrade is less than one second, which is nearly imperceptible to the business. In scenarios where the active upgrade is not supported, the network downtime may exceed 30 seconds.

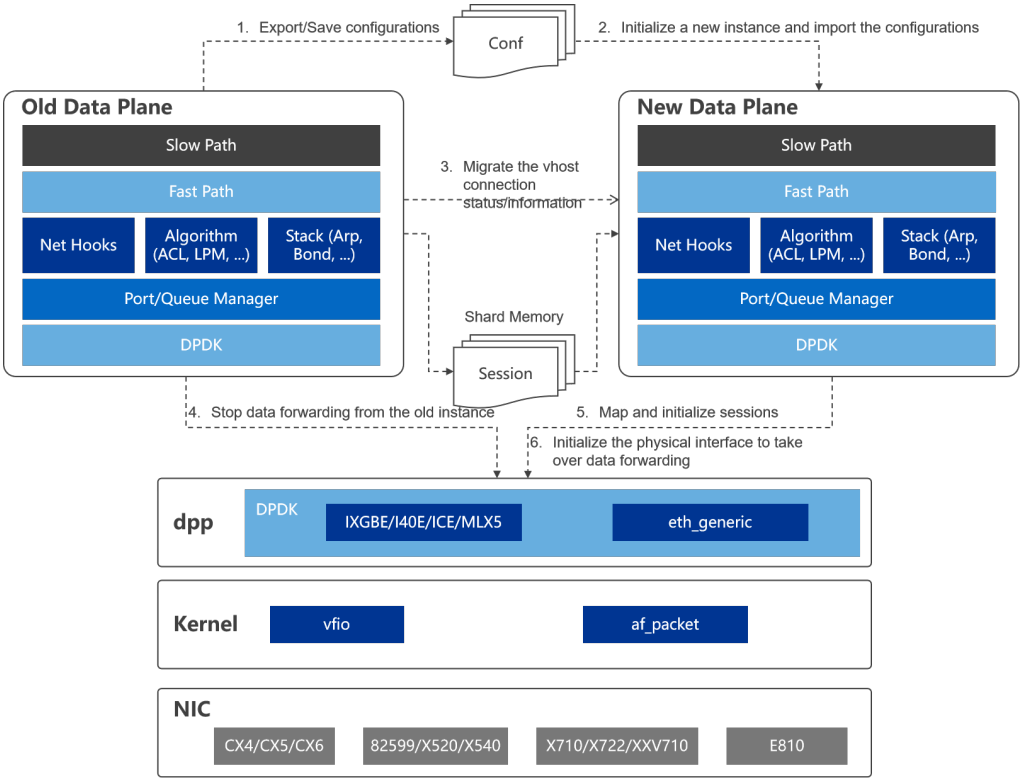


Figure 286: Active Upgrade Architecture

2. Support for the export of IPFIX flow visualization reports. When flow visualization is enabled, the forwarding performance loss is less than 5%.

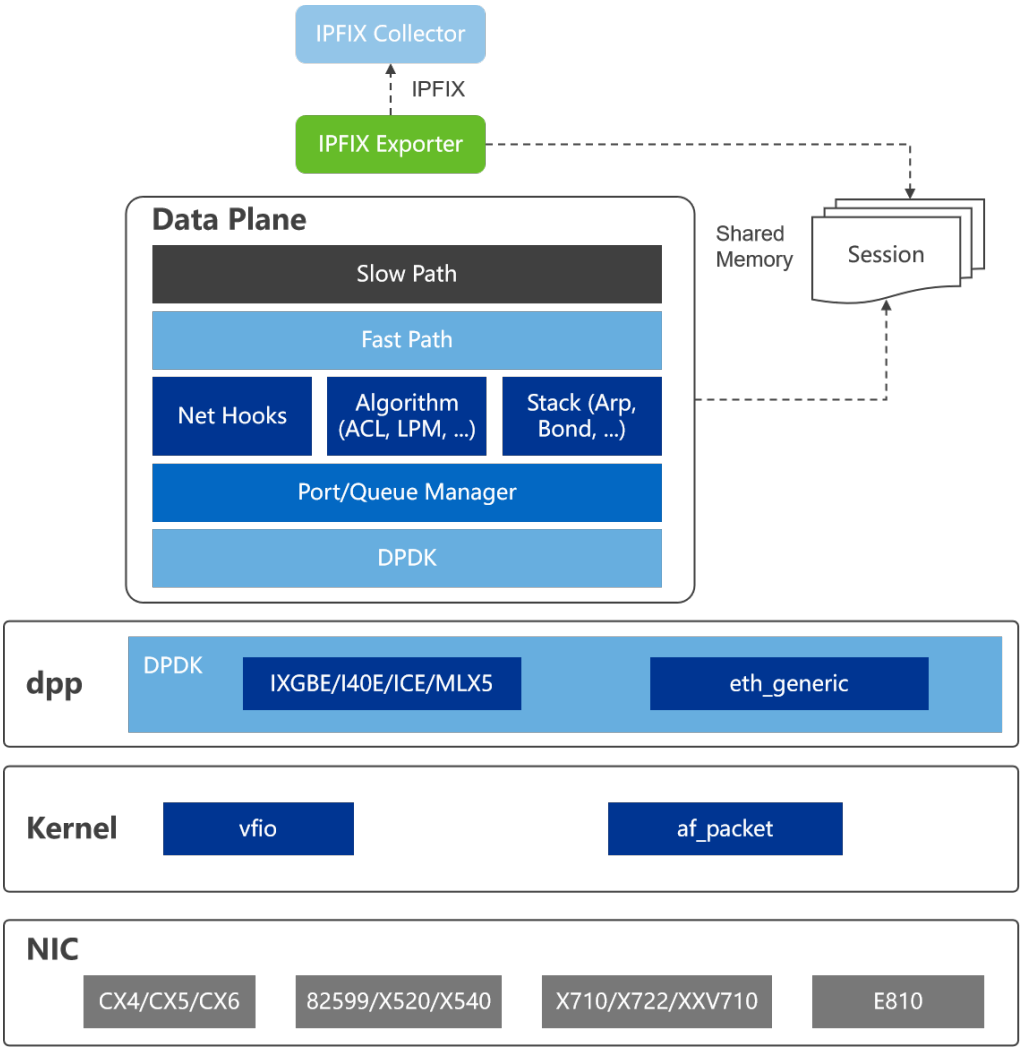


Figure 287: IPFIX Flow Visualization

3. Support for network connectivity test. You can test the IPv4/IPv6 network connectivity between VMs, at Layer 4, or between virtual routers and VMs in single-host or multi-host scenarios. The test report will provide the locations and causes of packet loss.

Reliability:

The data plane runs in userspace, ensuring that anomalies in the data plane do not result in kernel crashes. It also implements a health monitoring mechanism, which detects its health status and swiftly recovers it from

anomalies by restarting services. Additionally, the data plane's containerization significantly reduces its dependence on the host system. These measures have greatly enhanced the reliability of the data plane.

5.3.2.2 Functional Architecture of aNET

5.3.2.2.1 Overview

The functional modules of aNET run on nodes with different roles to ensure horizontal scalability and separation between the management and control planes. The modules collaborate to achieve connectivity, management, and orchestration of the cluster virtual networks.

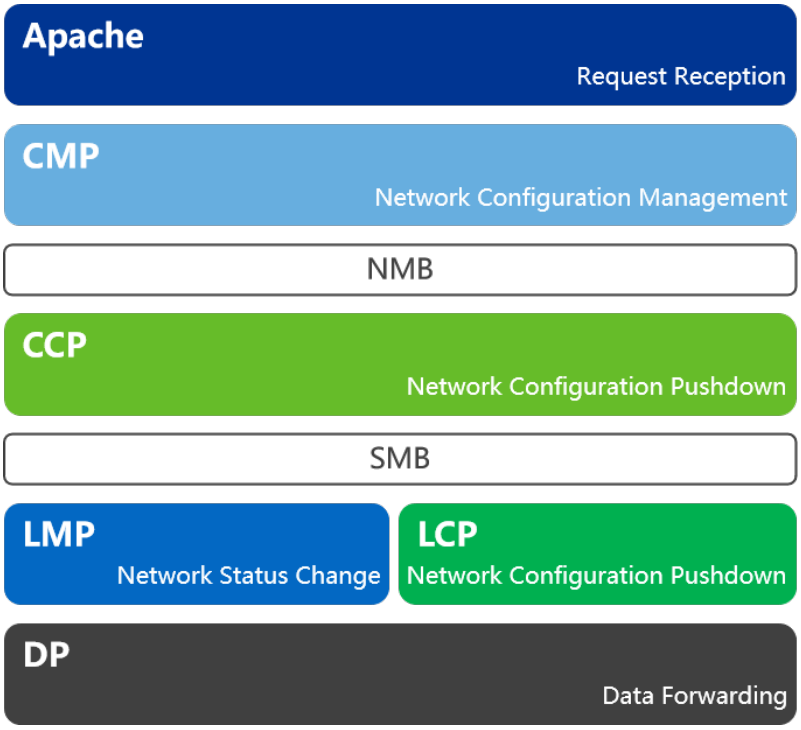


Figure 288: Functional Modules of aNET

- **CMP**: The CMP, which runs on the primary controller, is the only plane where HCI network RESTful APIs can work. It allows you to manage the full lifecycle of all virtual network topologies and basic virtual network components (such as distributed switches, distributed firewalls, and edge

routers). The CMP also provides unified management for all Sangfor NFV devices, such as automated deployment and lifecycle management.

- **LMP**: LMPs, which run on compute nodes, manage the status and events of virtual network resources on these nodes. These planes enable status management (query and modification) and event reporting of virtual network resources. For example, an LMP can provide the status information of the virtual network resources for the reporting and alert services, assist in the live migration or active upgrade of VMs, and report key events generated in the data plane, such as route role switches and network interface disconnections.
- **CCP**: The CCP, which runs on the primary controller, receives configurations from the CMP and then distributes the configurations to specific physical nodes within the cluster after calculation.
- **LCP**: LCPs, which run on compute nodes, receive configurations from the CCP or pull configurations from the cache of the CCP and then distribute the configurations to the data plane after conversion.
- **Data plane**: The data plane, which runs on all nodes, forwards packets from VMs to physical network interfaces or the virtual network interfaces of other VMs.
- **Northbound message bus (NMB)**: NMB is the message bus (MB) from the management plane to the control plane.
- **Southbound message bus (SMB)**: SMB is the message bus from the CCP to LCPs.

Here is an example of updating edge configurations, which can demonstrate how aNET modules collaborate.

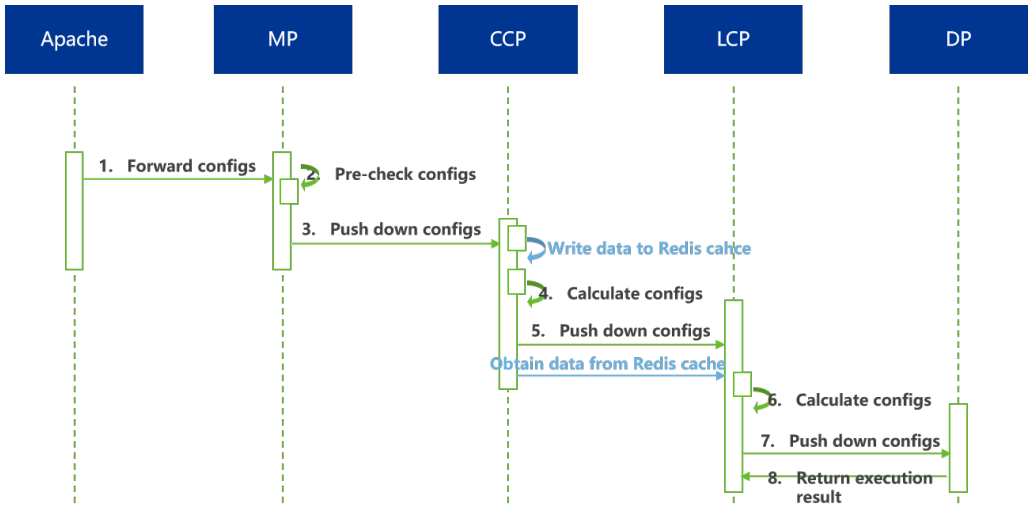


Figure 289: Configuration Distribution Process

1. When you update edge configurations, the Apache module receives, verifies, and responds to your request before sending it to the CMP.
2. The CMP checks the received request before writing the configurations to the database.
3. The CMP distributes the configurations to the CCP through NMB.
4. The CCP stores the configurations in its Redis cache and finds the destination nodes for configuration distribution based on calculation.
5. The CCP distributes the configurations to LCPs through SMB.
6. LCPs receive data from SMB or pull data from the Redis cache and then calculate the configurations.
7. LCPs distribute the configurations to the data plane through the data plane channel.
8. The data plane returns the execution result and forwards packets based on the updated configurations.

5.3.2.2.2 Deployment of Functional Modules

Single-cluster mode:

In an HCI cluster with two nodes, both nodes are controller nodes, with one acting as the primary controller and the other acting as an ordinary controller.

The CMP and the CCP run on the primary controller, Redis runs on the primary controller in single-instance mode, MySQL runs on both nodes in primary-secondary mode, and LMPs, CCPs, and the data plane run on both nodes.

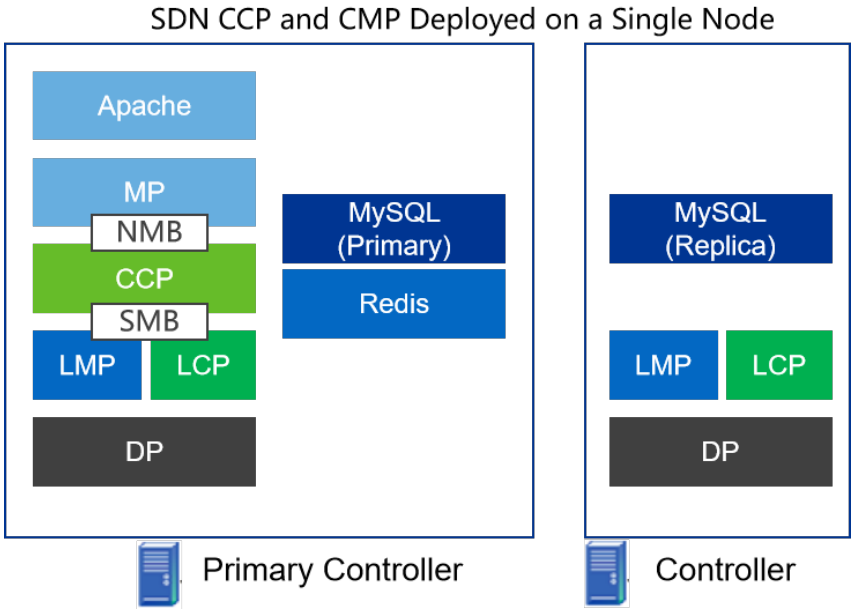


Figure 290: Two-Node HCI Cluster

In an HCI cluster with three or more nodes, there are three controller nodes, with one acting as the primary controller and the other two acting as ordinary controllers. The CMP and the CCP run on the primary controller, Redis runs on the primary controller in single-instance mode, MySQL runs on the primary controller and one ordinary controller in primary-secondary mode, and LMPs, LCPs, and the data plane run on all controller nodes.

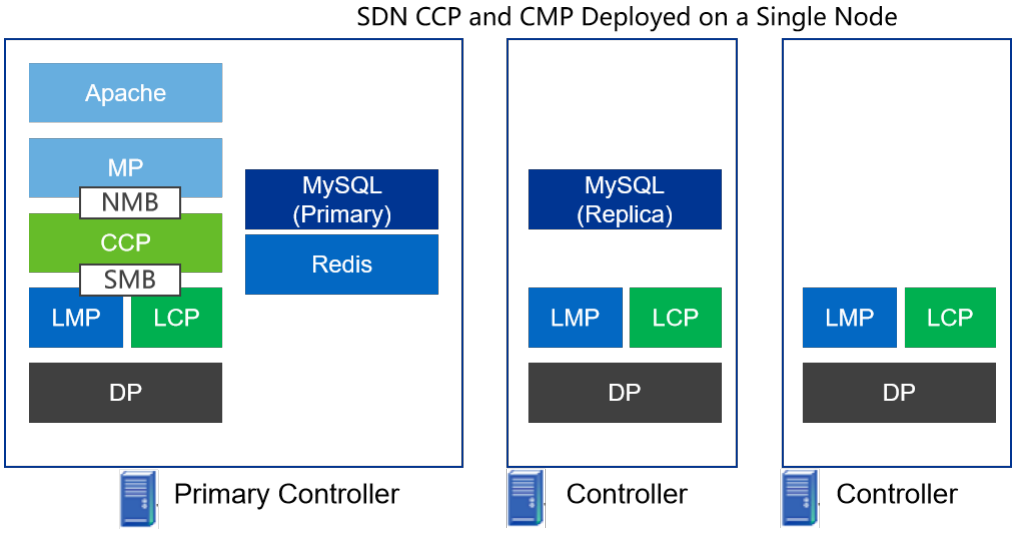


Figure 291: HCI Cluster With Three or More Nodes

Cross-cluster mode:

In scenarios involving multiple HCI clusters, one cluster acts as the master cluster for management and control tasks, while others act as ordinary clusters. There are three controller nodes in the master cluster, with one acting as the primary controller and the other two acting as ordinary controllers.

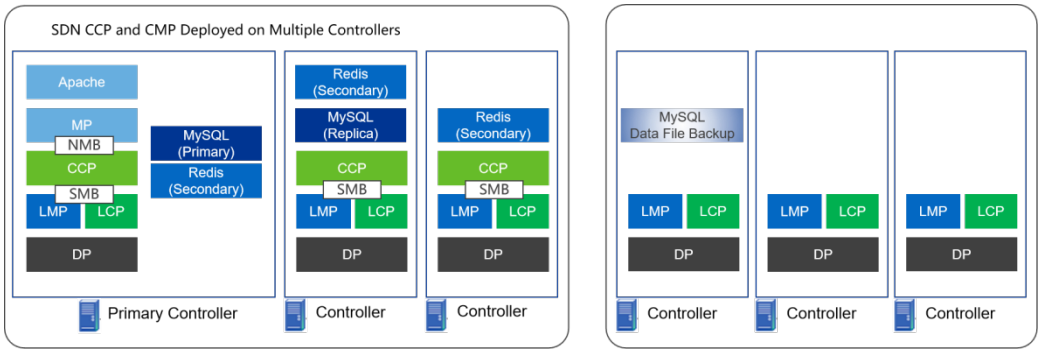


Figure 292: Functional Module Deployment Across Multiple HCI Clusters

The CMP runs on the primary controller, while the CCP runs on the three controller nodes in cluster mode. MySQL runs in primary-secondary mode, with the primary instance running on the primary controller and the secondary instance running on an ordinary controller. Ordinary nodes in ordinary clusters store the backup of the MySQL data to ensure recovery in the ordinary clusters in the event of master cluster failures. LMPs, CCPs, and the data plane run on

the three controller nodes.

Redis adopts a primary-secondary deployment in sentinel mode. The primary Redis instance runs on the primary controller in the master cluster, while the secondary Redis instances run on ordinary controllers in the master cluster.

5.3.2.3 High Availability Design of aNET

5.3.2.3.1 High Availability of CMP

The CMP offers high availability in the same way in single-cluster and cross-cluster scenarios. This plane is stateless. It writes user-provided configurations, such as topology information, to the local MySQL database that is deployed in primary-secondary mode.

1. Under normal conditions, the primary MySQL instance runs on the primary controller for handling read and write requests, while the secondary MySQL instance runs on an ordinary controller as a hot standby. In most cases, it is the newly added controller node that hosts the secondary MySQL instance, as shown in the left diagram in the following figure.
2. If you manually switch the primary controller, the primary and secondary roles in MySQL remain the same, and the CMP starts to run on the new primary controller. Configurations that you send are written to the primary MySQL database by the CMP running on the new primary controller, as shown in the middle diagram in the following figure.
3. When the primary controller is unavailable due to a failure, a MySQL failover is triggered, and the CMP starts to run on the new primary controller. Configurations that you send are written to the new primary MySQL database by the CMP running on the new primary controller, as shown in the right diagram in the following figure.

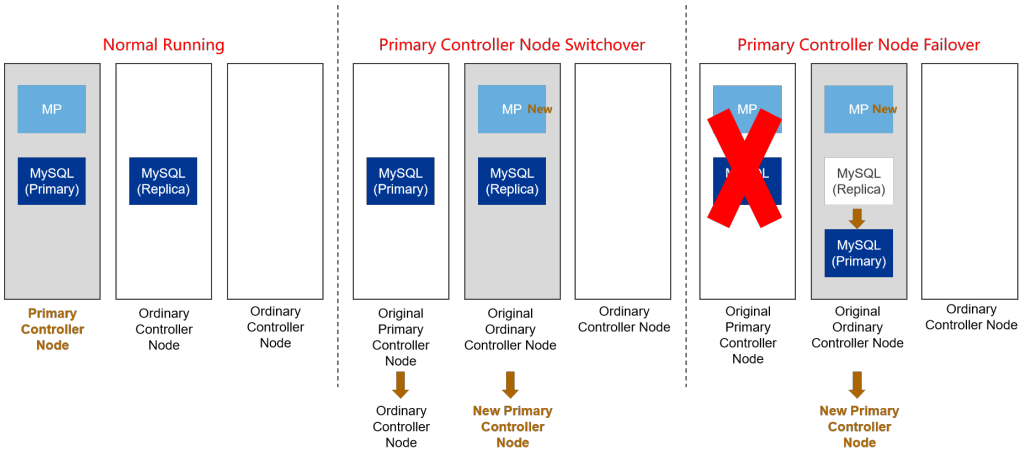


Figure 293: High Availability of CMP

5.3.2.3.2 High Availability of CCP

Single-cluster scenarios:

1. Under normal conditions, the CCP and Redis run on the primary controller, as shown in the left diagram in the following figure.
2. During the switch of the primary controller, the CCP and Redis are started on the new primary controller, while the CCP and Redis on the original primary controller are stopped. The new Redis service synchronizes the data from MySQL before the switch, as shown in the middle diagram in the following figure.
3. In the event of a primary controller failure, the cluster management service elects a new primary controller and starts the CCP and Redis on it. Then, the Redis service synchronizes data from MySQL, as shown in the right diagram in the following figure.

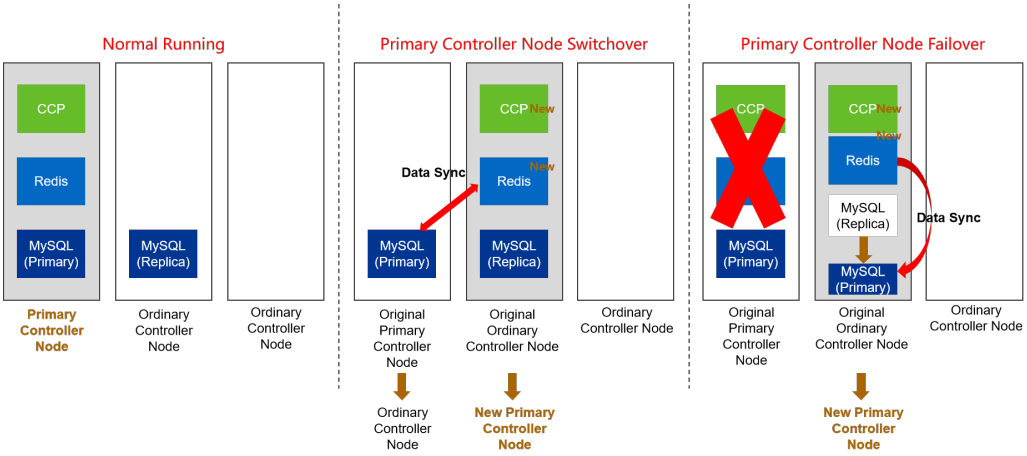


Figure 294: High Availability of CCP in Single-Cluster Scenarios

Cross-cluster scenarios:

In cross-cluster scenarios, the CCP runs on all controller nodes in cluster mode, processing configurations from NMB and writing data to the Redis cache. When the CCP service on a specific node fails, reallocation is triggered for CCP services on other nodes to automatically take over the tasks of CCP on the failed node. Redis adopts a primary-secondary deployment in sentinel mode. The primary Redis instance runs on the primary controller in the master cluster, while the secondary Redis instances run on ordinary controllers in the master cluster.

1. Under normal conditions, the CCP runs on all controller nodes in cluster mode, while Redis adopts a primary-secondary deployment with one primary instance and two secondary instances. The primary Redis instance elected among the three candidates may run on the primary controller or an ordinary controller.
2. During the switch of the primary controller, the CCP and Redis remain the same, while the CMP starts to run on the new primary controller.
3. In the event of a primary controller failure, the cluster management service elects a new primary controller, the Redis instance on the new primary controller becomes the primary if the original primary Redis instance is on the failed primary controller, and a MySQL failover is triggered. As Redis and MySQL are deployed in primary-secondary mode, only failover is triggered, without the need for data synchronization.

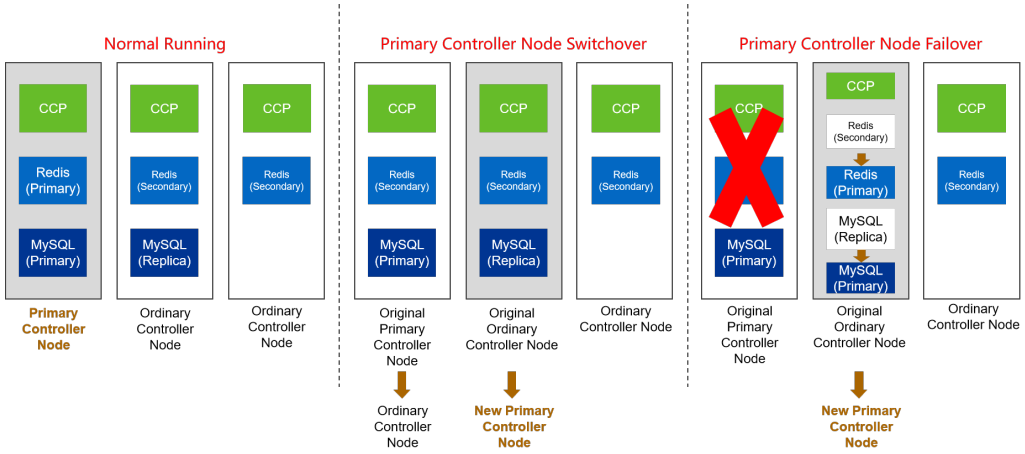


Figure 295: High Availability of CCP in Multi-Cluster Scenarios

5.3.2.3.3 High Availability of MB

MB adopts a Kafka-based deployment in cluster mode. Its high availability is implemented based on Kafka's data replication and leader election mechanisms.

Here is an example of the CCP getting a message from the CMP through NMB, which can demonstrate how high availability is implemented. Firstly, a leader node is elected based on the leader election mechanism; then, the message producer pushes a message to the leader and the follower nodes replicate the message from the leader node; lastly, the CCP consumes the message by pulling it from the leader node. This avoids data loss because the CCP can get messages from follower nodes even if the leader node is unavailable.

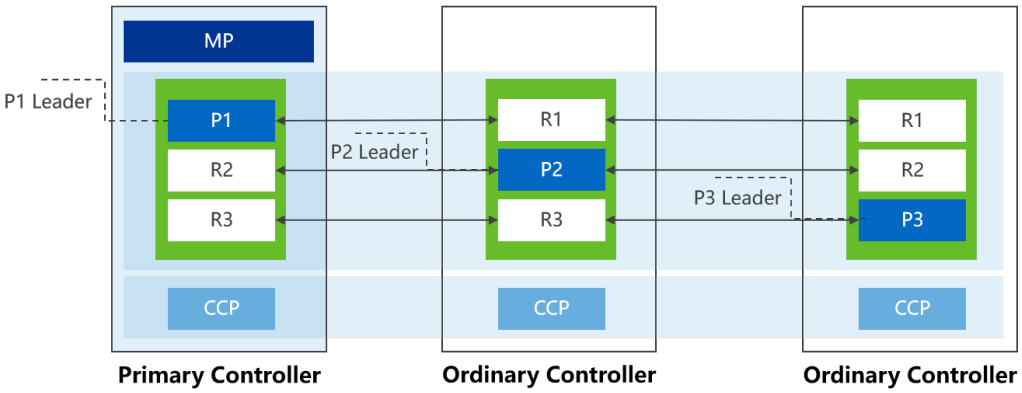


Figure 296: High Availability of MB

5.3.2.4 Reliable Network Planes of aNET

HCI consists of four logical network planes: the management plane, business plane, data plane (VXLAN), and storage plane.

- Management Plane

It ensures network services for the HCI cluster management service and enables administrators to manage HCI clusters. You can achieve redundancy for the management plane through link aggregation, so that the failure of a single switch or link does not affect the stability of HCI.

- Business Plane

It provides external network services for VMs. You can achieve link redundancy for the business plane by aggregating two switches and enable static network interface bonding for outbound business network interfaces. Each server can have multiple physical NICs configured as business network interfaces, which enables VMs in the virtual network to have redundant and reliable business network connections.

- Data Plane (VXLAN)

It handles east-west traffic between VMs and virtual network devices for network communication between them. The data plane achieves link redundancy through physical switch aggregation. HCI features distributed built-in virtual switches and virtual routers, with each host in the cluster having a virtual switch instance. If a host goes offline, traffic originally routed to the virtual switch instance on it is redirected to and handled by another host because the virtual routes and the VM are switched to the new host.

- Storage Plane

It offers a network plane for storage I/O data communication. You can deploy a private storage plane separately to ensure data security. As HCI has implemented software-level link aggregation, link aggregation in the private aSAN network can achieve load balancing based on TCP connections, enabling TCP connections between two hosts to utilize distinct physical links for maximum throughput.

The four network planes are logically isolated, and the failure of one plane does not affect the other planes.

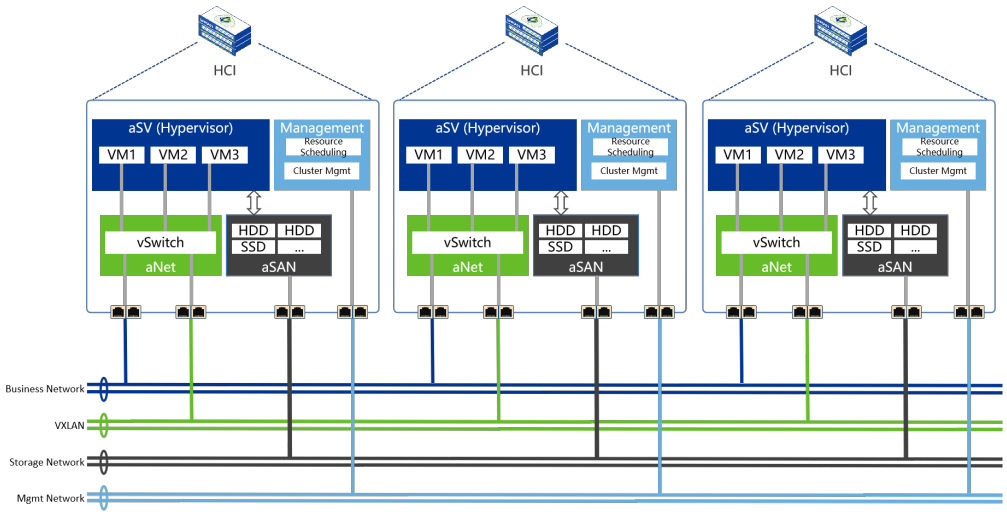


Figure 297: Network Architecture of aNET

5.3.2.5 High-Performance Architecture of aNET

aNET is featured with high performance. It supports high concurrency in large-scale data centers.

1. The CMP is not limited by cookies, as its APIs are stateless. You can scale it out on demand during deployment.
2. MB supports high concurrency by leveraging Kafka's capabilities, such as page cache, zero-copy, and sequential write.
3. The CCP stores its status in the external high-performance Redis.
4. LCPs obtain configurations from Redis through MB, which also guarantees high performance.
5. The SDN controller can manage super-large scale networks, supporting the centralized management of networks with up to 1,024 nodes.
6. The data plane is designed for network I/O-intensive applications. It supports both dedicated and generic NICs. The forwarding flow at the application layer has been rewritten and simplified to ensure enhanced stability in the data plane.

5.3.3 Network Deployment

5.3.3.1 Network Interface Aggregation

HCI allows you to aggregate NICs based on static bonding for redundancy and better performance. It supports the aggregation modes shown in the following figure.

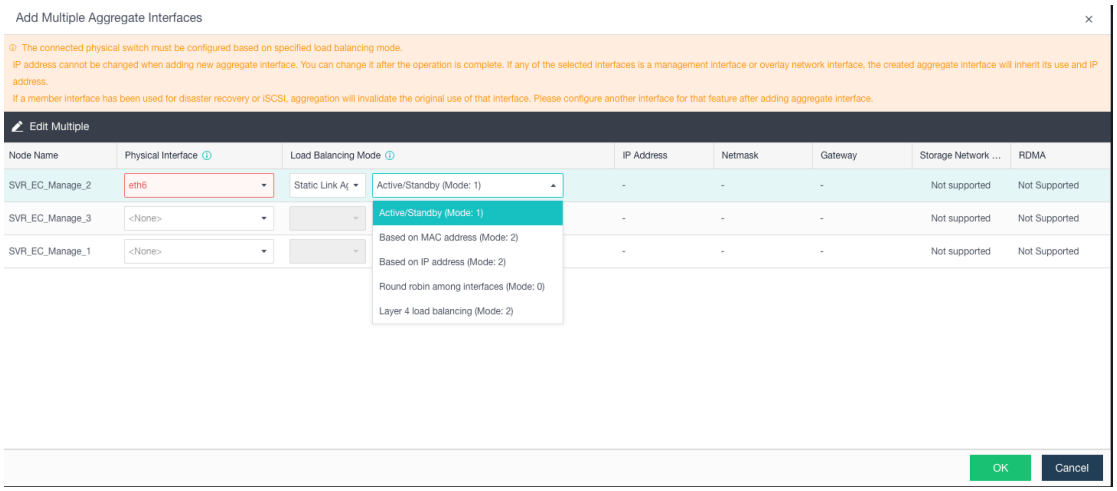


Figure 298: Network Interface Aggregation Modes

1. Active-Backup

In this mode, the two network interfaces act as each other's backup for redundancy, with only one of them forwarding traffic at any given time. When one network interface fails, the other one is automatically used. When an aggregated network interface in this mode communicates with its peer switch, you do not need to enable port aggregation for the two network interfaces of the switch.

2. MAC

In this mode, network interfaces work simultaneously for load balancing and redundancy. Packets are distributed to different network interfaces based on a hash algorithm according to their source and destination MAC addresses.

When an aggregated network interface in this mode communicates with its peer switch, you must enable static port aggregation for the network interfaces of the switch. However, the hash-based load balancing mode of the switch can

differ from that of the HCI NIC. This is because load balancing modes mainly affect the outbound traffic, and inconsistent paths for the request and response packets within an aggregated port group do not affect traffic forwarding.

3. IP

In this mode, network interfaces work simultaneously for load balancing and redundancy. Packets are distributed to different network interfaces based on a hash algorithm according to their source and destination IP addresses.

When an aggregated network interface in this mode communicates with its peer switch, you must enable static port aggregation for the network interfaces of the switch. However, the load balancing mode of the switch can differ from that of the HCI NIC. This is because load balancing modes mainly affect the outbound traffic, and inconsistent paths for the request and response packets within an aggregated port group do not affect traffic forwarding.

4. Round-Robin

In this mode, network interfaces work simultaneously for load balancing and redundancy. Packets are distributed to different network interfaces based on a hash algorithm.

When an aggregated network interface in this mode communicates with its peer switch, you must enable static port aggregation for the network interfaces of the switch. However, the load balancing mode of the switch can differ from that of the HCI NIC (if possible, use the same mode during delivery). This is because load balancing modes mainly affect the outbound traffic, and inconsistent paths for the request and response packets within an aggregated port group do not affect traffic forwarding.

5. L4

In this mode, network interfaces work simultaneously for load balancing and redundancy. Packets are distributed to different network interfaces based on a hash algorithm according to their source IP addresses, destination IP addresses, and ports of the transport-layer protocols.

When an aggregated network interface in this mode communicates with its

peer switch, you must enable static port aggregation for the network interfaces of the switch. However, the load balancing mode of the switch can differ from that of the HCI NIC. This is because load balancing modes mainly affect the outbound traffic, and inconsistent paths for the request and response packets within an aggregated port group do not affect traffic forwarding.

6. LACP

The LACP mode leverages LACP-based parameter negotiation for active link selection. It is also known as M:N mode, where M active links and N inactive backup links are determined in an aggregation group based on LACP. The LACP mode enhances link reliability and enables diverse load balancing methods across active links. In this mode, only some of the links are active for data forwarding. When an active link fails, the link aggregation group automatically selects an inactive link to take its place, ensuring the number of links involved in data forwarding is always the same.

5.3.3.2 Physical Networks

HCI has five physical network planes: management plane, business plane, data plane, storage plane, and out-of-band management (OOBM) plane.

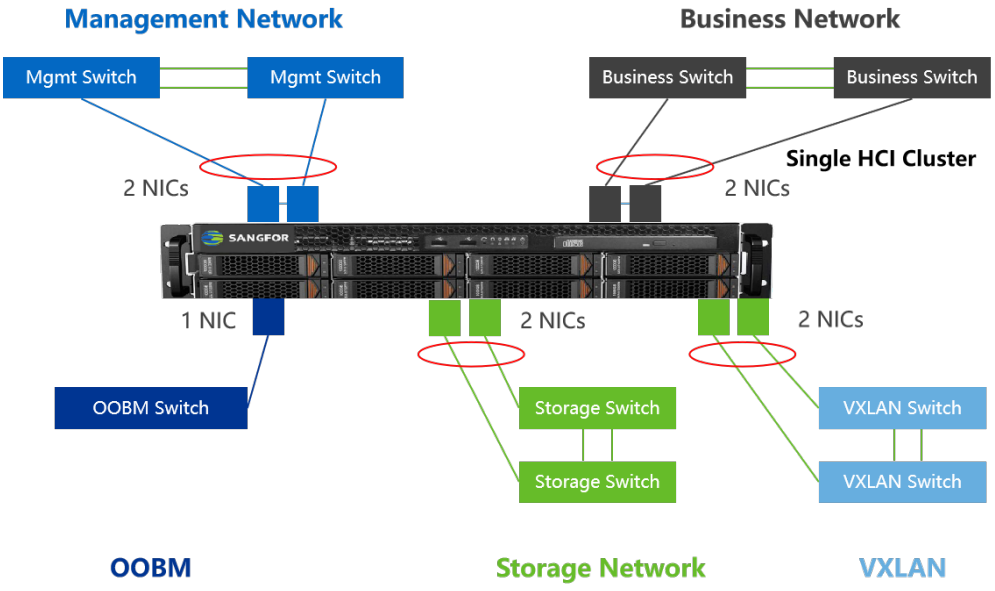


Figure 299: HCI Network Planes

- Management Plane

The management plane manages clusters and live migrates VMs, including distributing creation commands and executing backup restore commands. It ensures cluster reliability through capabilities such as heartbeat mechanism (host status monitoring) and VM migration, and handles no business data. In simple terms, the management network is the "main artery" of the cluster that makes the cluster "alive." This requires that every host in the cluster has a management network interface connected to this main artery.

You can select a 1GE or 10GE NIC based on your business needs and pair it with an access switch of the same speed. For scenarios involving extensive VM migrations, use 10GE NICs. The management plane can work with only one NIC. However, to ensure redundancy, we recommend that you connect two aggregated NICs to two stacked management plane switches for high bandwidth and availability.

- Business Plane

The business plane manages the ingress and egress of north-south traffic. This is optional and not all hosts require a business plane NIC. You can select a 1GE or 10GE NIC based on your business needs and pair it with an access switch of the same speed. The business plane can work with only one NIC. However, to ensure redundancy, we recommend that you connect two aggregated NICs to two stacked business plane switches for high bandwidth and availability.

- Data Plane

The data plane (also known as VXLAN) ensures Layer 2 communication between VMs on different hosts within a cluster. It enables east-west communication across hosts using VXLAN technologies. As the data network offers cross-host access through VXLAN tunnels based on host VTEPs, traditional Ethernet switches (without the need to enable VXLAN) can act as data plane switches to offer underlay network connectivity between host VTEPs.

It is recommended to configure a 10GE VXLAN NIC for each physical host in the cluster and use 10GE access switches. For scenarios involving only minimal east-west traffic, you can also choose 1GE NICs and switches. To achieve high

bandwidth and availability, connect two aggregated data plane NICs to two stacked data plane switches.

- Storage Plane

The storage plane enables storage-level communication between hosts within a cluster. For example, a VM on one host can access storage resources in another host through the storage plane. It is recommended to configure a 10GE storage plane NIC for each physical host in the cluster and use 10GE access switches. The storage plane can work with only one NIC. However, to ensure redundancy, we recommend that you connect two aggregated storage plane NICs to two stacked storage plane switches for high bandwidth and availability.

- OOBM Plane

The OOBM plane manages server hardware. You can connect a server to a single 1GE switch using a single 1GE NIC for out-of-band management.

5.3.3.3 Physical Network Interface Reuse

Physical network interface reuse refers to using the network interface (single or aggregated) of a physical host to handle traffic for multiple HCI network planes. For example, you can use the same 1GE aggregated NIC for the management network and VXLAN.

aNET supports two reuse modes: three-in-one network interface reuse and four-in-one network interface reuse. The following figure demonstrates how network interface reuse is implemented. The figure on the left shows a scenario where no network interface is reused, while the figure on the right shows a scenario where three network planes use the same network interface (three-in-one network interface reuse).

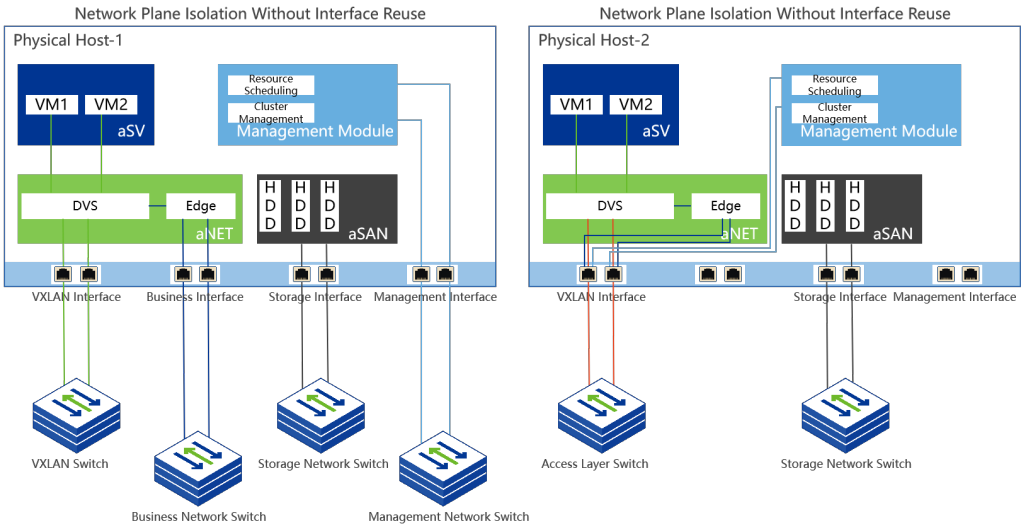


Figure 300: Three-in-One Network Interface Reuse

To ensure an appropriate proportion of traffic from each network plane after network interface reuse, aNET offers QoS policies. This feature allows you to set a traffic limit for each network plane to avoid communication anomalies caused by bandwidth contention. Network interface reuse has the following characteristics:

1. Isolated network planes

Each network plane allows separate VLAN configurations, enabling Layer 2 isolation for different businesses to prevent mutual interference.

2. Simplified deployment

- Enables bulk operations on network configurations, including editing network interfaces, creating aggregate interfaces, adding VLAN subinterfaces, and modifying interfaces for management, data, and business.
- Supports changing management interfaces while retaining the configurations.
- Allows one or multiple VLANs for each aggregate interface.

5.3.3.3.1 Three-in-One Network Interface Reuse

HCI supports three-in-one network interface reuse, where the management,

VXLAN, and business networks use the same aggregate interface. The storage network employs an aggregate interface consisting of two 10GE network interfaces. Compared to the original solution with eight network interfaces and four network planes, this solution simplifies network operations and reduces costs while maintaining the high availability and performance of businesses.

Three-in-one network interface reuse is implemented based on the Layer 2 forwarding of virtual switches in the data plane. In reuse scenarios, the business network employs the physical network interface, while the VXLAN and management networks are connected to a virtual switch through virtual network interfaces. After DPDK forwards packets to the data plane, the virtual switch distributes packets according to their destination MAC addresses. Packets with a matching destination MAC address are distributed to the virtual network interfaces of the VXLAN and management networks accordingly, while other packets are distributed to the business network. This enables a network interface to forward traffic for the business, VXLAN, and management networks simultaneously without causing any mutual interference.

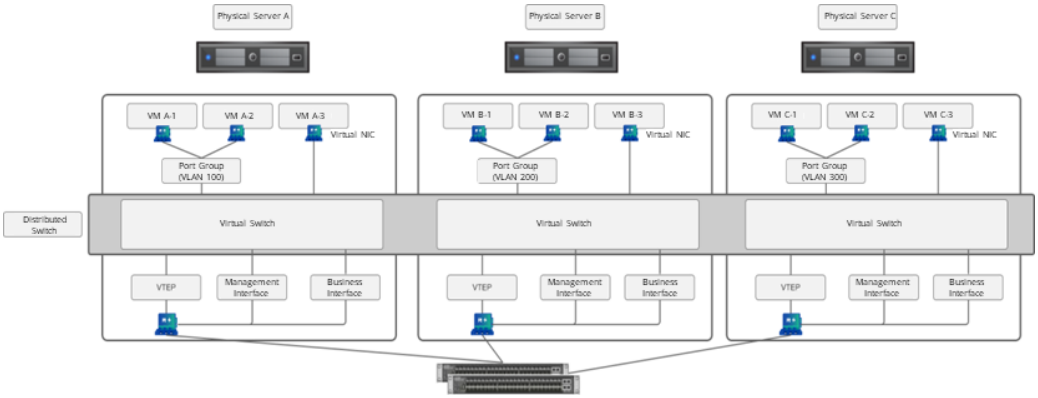


Figure 301: Principle of Three-in-One Network Interface Reuse

5.3.3.3.2 Four-in-One Network Interface Reuse

Key technologies:

Intel has introduced three types of CPU/PCI virtualization technologies to support virtualized environments, including VT-x (processor-based virtualization), VT-d (PCI-based I/O virtualization), and VT-c (network-based virtualization). PCI can be considered as a motherboard slot and is a standard

that Intel introduced in 1991 for defining local buses. SR-IOV is a VT-d technology that offers NIC-level virtualization. It enables direct data movement without going through the VMM layer by offering separate memory space for VMs. SR-IOV has two types of functions: physical functions (PF) and virtual functions (VF). PFs support all PCIe capabilities so that they can be discovered, managed, and configured like regular PCI devices. VFs, which are virtualized from PFs, only support lightweight PCIe capabilities for I/O handling. Once SR-IOV is enabled, the PCI configuration space of each VF can be accessed using the bus, device, and function numbers of PF, and each VF possesses a PCI memory space to map its register set. Compared to traditional bridge mode, the KVM mode provides better performance because it enables physical NICs to directly exchange data with VMs through VFs by bypassing the virtualization stack.

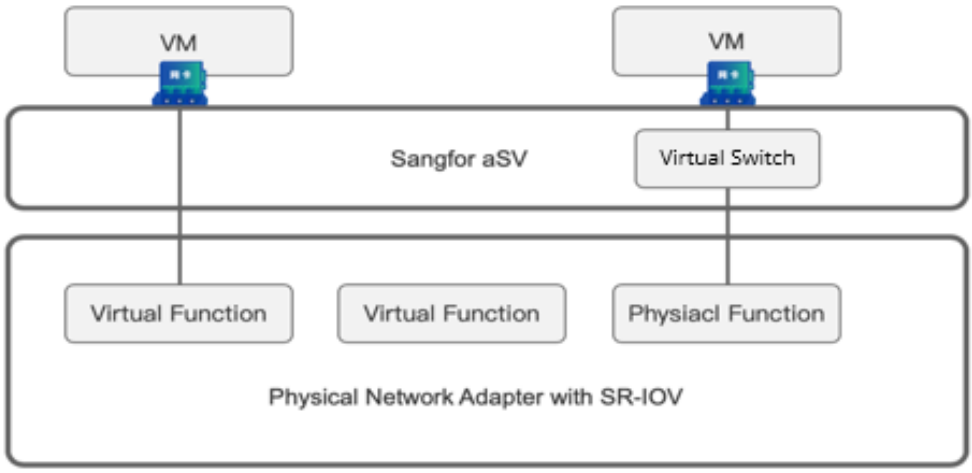


Figure 302: SR-IOV Principle

Principle of four-in-one network interface reuse:

HCI supports four-in-one network interface reuse, where the management, VXLAN, business, and storage networks use the same aggregate interface, which must be 10GE or higher. Compared to three-in-one network interface reuse, four-in-one network interface reuse further simplifies network operations and reduces costs.

They are implemented in different ways. The primary difference is that four-in-

one network interface reuse employs the internal eSwitch of physical NIC to classify and forward traffic. eSwitch refers to the switch inside a NIC that supports SR-IOV. It is used to classify traffic, forward traffic, and connect PFs and VFs to the physical network interface. Once four-in-one network interface reuse is enabled on the aggregate interface and SR-IOV is enabled on the NICs, the following three NICs are created based on the two NICs of the aggregate interface: PF, VF0, and VF1. During eSwitch traffic distribution, PF acts as the network interface for the VXLAN and business networks, VF0 for the management network, and VF1 for the storage network.

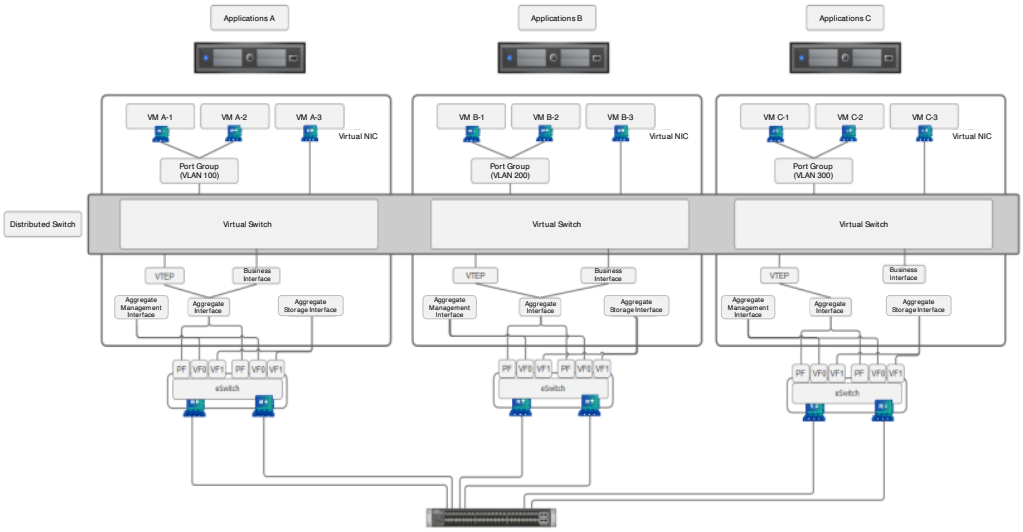


Figure 303: Principle of Four-in-One Network Interface Reuse

NIC models that support four-in-one network interface reuse:

Intel X710, Mellanox CX4, Intel X722, and Intel XL710 (10GE or higher)

5.3.3.4 Cluster Deployment Modes

5.3.3.4.1 Overview

The evolution of cloud data centers has led to two deployment modes. Customers with fewer nodes prefer the single-cluster mode for easy management, while customers with numerous nodes often opt for the multi-cluster mode. aNET's architecture fits well with both modes.

In single-cluster scenarios, the main challenge lies in the different business

requirements for computing, storage, and network resources from the ever-increasing cluster. However, as nodes do not need to be of the same hardware specification, you can avoid this problem by planning ahead and purchasing varied hardware. For example, you can purchase a node with high CPU and NIC specifications but a low disk specification for network forwarding. However, if varied hardware resources are not necessary in a single cluster, you can still opt for the original hyper-converged deployment mode.

Typical scenarios where multi-cluster deployment is required are as follows:

1. The customer requires high reliability and hopes to isolate logical failures by using multiple physical clusters.
2. The customer must achieve physical isolation among clusters to meet regulatory requirements.
3. The customer is accustomed to grouping heterogeneous resources based on physical isolation.

When using and managing multiple clusters, customers prefer a Layer 2 network for unified management. In addition, the management of multiple clusters often involves SCP. Multi-cluster deployment includes four typical types of scenarios: multiple clusters in one data center, multiple clusters across multiple data centers, multiple clusters across headquarters and branches, and single-tenant managed cloud across multiple regions.

All cluster deployment modes rely on the underlying management, business, data, and storage networks of HCI to meet diverse communication requirements.

5.3.3.4.2 Network Communication Domain

In the multi-cluster deployment mode, east-west access between clusters is required. To ensure Layer 2 reachability for VMs across clusters, the overlay networks of the clusters must belong to the same Lay 2 network, while the underlay networks belong to different Layer 2 networks. To facilitate Layer 2 packet forwarding across clusters, aNET for a specific cluster must be shared with other clusters, and a unified control panel must be used to maintain the

network topologies and configurations of the clusters. This enables resource isolation between different clusters while ensuring that their overlay networks belong to the same Layer 2 network. The scope of the multi-cluster Layer 2 network is referred to as the network communication domain. In the network communication domain, HCI clusters use a shared aNET module for network connection and unified management of network configurations across clusters. Creating a network communication domain requires the Layer 2 or 3 connectivity of the management and data networks between clusters.

For example, a tenant's VMs in the same subnet may come from different clusters within a network communication domain. In this case, the corresponding clusters manage the full lifecycle of the VMs, such as creation and scheduling, while SCP centrally orchestrates the virtual networks. This eliminates the need to determine the cluster of each VM, allowing customers to concentrate on unified network orchestration within the communication domain.

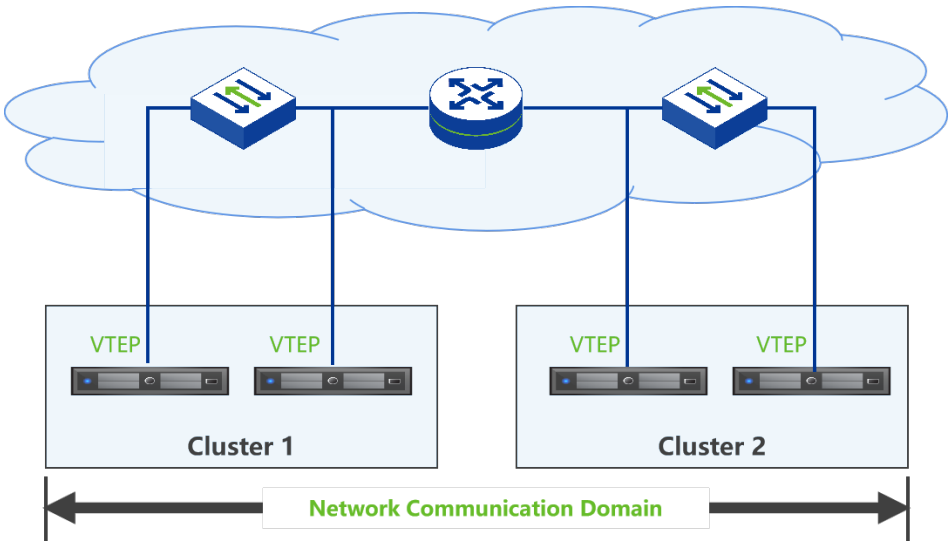


Figure 304: Network Communication Domain

To create a network communication domain across HCI clusters, ensure the Layer 2 or 3 connectivity of the management and data networks between clusters, select cluster resource pools, configure management network IP addresses, and assign data network IP addresses.

5.3.3.4.3 Single-Cluster Deployment

Nodes in a single cluster are deployed in the converged mode. They are not assigned specific roles, and all of them have computing, network, and storage attributes. In a two-node cluster, there are two controller nodes, where one acts as the primary controller for cluster management while the other is an ordinary controller. In a cluster with three or more nodes, there are three controller nodes, where one acts as the primary controller for cluster management while the other two are ordinary controllers.

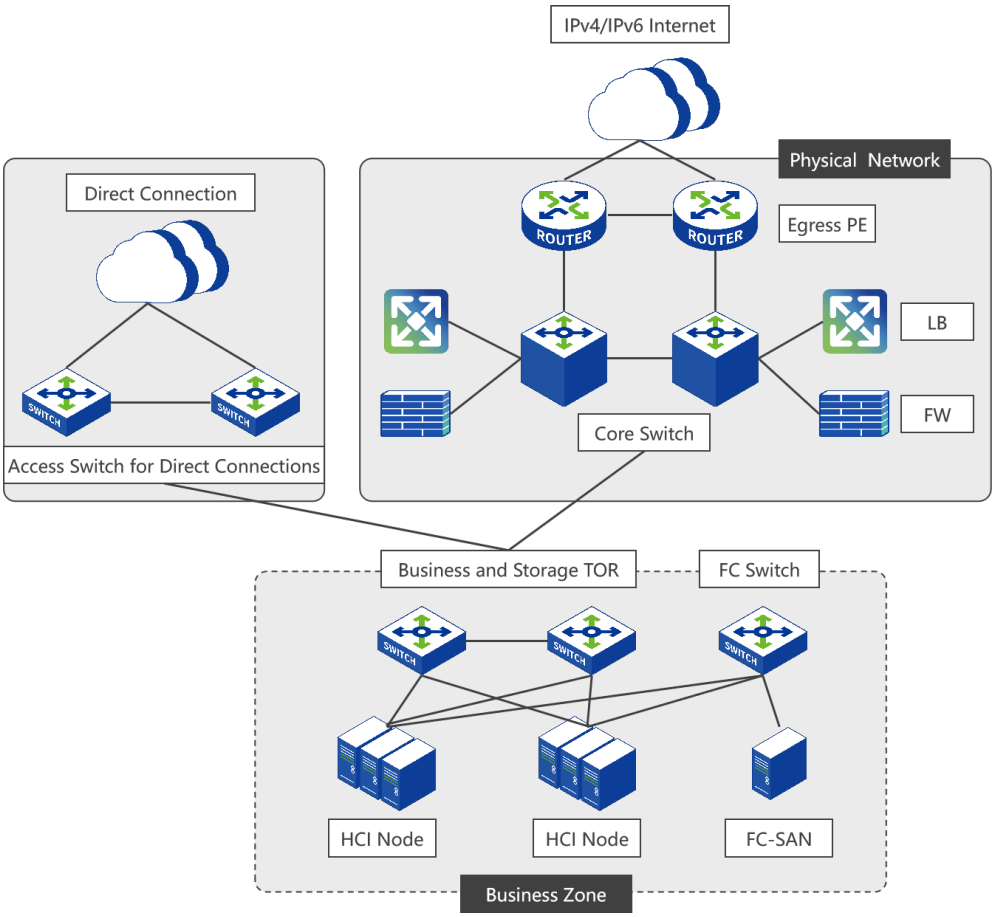


Figure 305: Converged Deployment Architecture

5.3.3.4.4 Multi-Cluster Deployment

Independent deployment of network nodes:

Due to varied demands for computing, network, and storage resources,

customers must purchase and expand different server hardware resources. HCI 6.7.0 has supported the independent deployment of network nodes. Network nodes are mainly used for unified edge traffic forwarding. Edge traffic includes north-south traffic within the data center and cross-subnet tenant traffic within a VPC.

To ensure high reliability, customers must deploy at least two network nodes in a cluster. Virtual edge network devices such as edges and edge routers achieve high availability and traffic load balancing across multiple network node clusters through scheduling policies. Do not run VMs on network nodes because network forwarding is CPU-intensive. For network nodes, select hardware with abundant NICs, high CPU clock speed, and high backplane bandwidth.

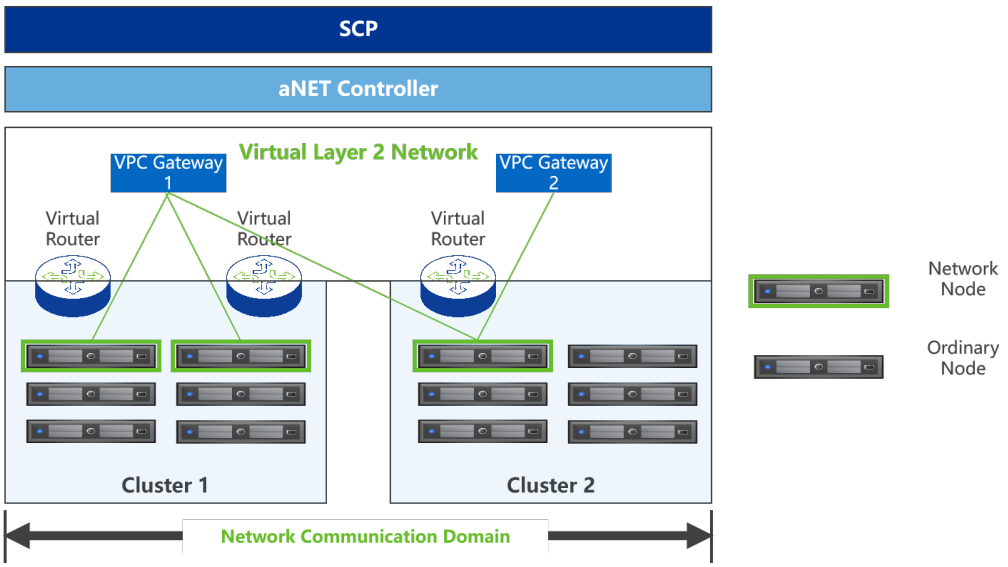


Figure 306: Independent Deployment of Network Nodes

Multiple clusters in one data center:

This deployment mode is well suited for isolating logical failures (such as management plane and upgrade failures), meeting regulatory requirements, and managing heterogeneous resources.

Generally, when there are multiple clusters in a data center, it is necessary to use SCP for unified management. The overlay networks of the clusters are connected through a network communication domain. For physical networks,

the management network and VXLAN of the member clusters must be interconnected, which means that the Layer 2 or 3 connectivity of the management network and VXLAN between hosts must be ensured. In addition, the latency of the VXLAN must be below 100 ms. If the VTEPs in the VXLAN of different clusters belong to different network segments, configure a VTEP gateway in the corresponding VXLAN to ensure Layer 3 connectivity between VTEPs.

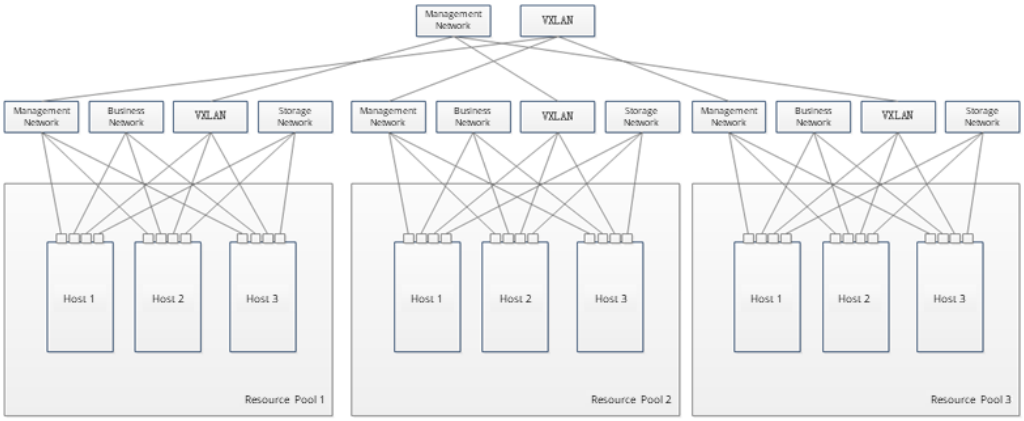


Figure 307: Interconnected Physical Networks of Clusters

Multiple clusters across multiple data centers:

Customers with large-scale businesses often have more than one data center. Each data center may have one or multiple clusters, and all clusters are managed on SCP. Data centers may be far apart due to different physical locations. In this case, you must consider network performance when implementing Layer 2 connectivity between data centers. Specifically, the latency of the cross-data center VXLAN must be below 100 ms. This performance requirement is in place to ensure that modules and applications within the architecture work properly. Here is an example of a typical three-layer application model consisting of frontend, middleware, and database services. In extreme cases, these three services can run on VMs in different clusters but share the same Layer 2 network. When a request comes in, the frontend service sends the data to the middleware service through the VXLAN. Then, the middleware service forwards the data to the database service for returning the data. The three services exchange data across clusters using the

VXLAN. Poor performance in the VXLAN may result in abnormal business access due to possible retransmissions and congestion. As the physical network deployment in multi-cluster and single-cluster modes is the same, customers only need to ensure Layer 3 reachability for the VXLAN and management networks of the member clusters.

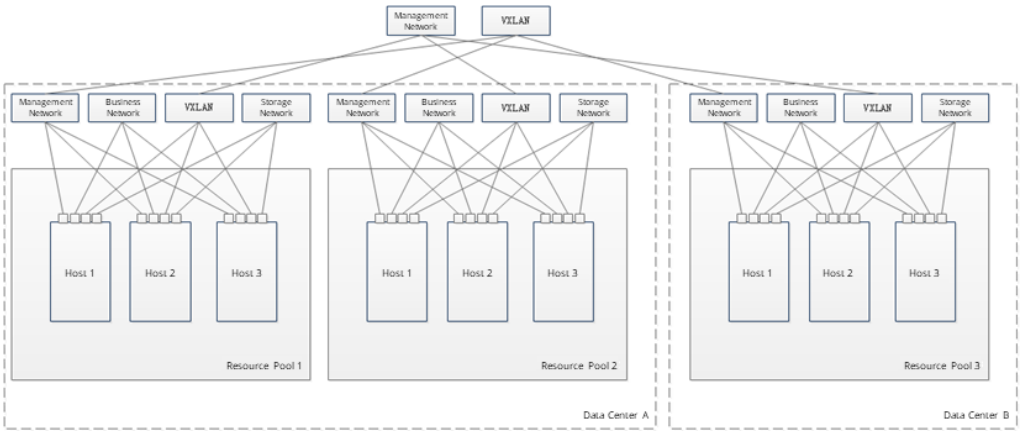


Figure 308: Physical Network of Multiple Data Centers

Multiple clusters across headquarters and branches:

This deployment mode is well suited for deploying SCP and HCI at the headquarters to manage all branch HCI clusters. The physical networks of the headquarters and branches are often interconnected through a VPN or direct connect gateway for secure access. Considering factors like long transmission distances and unpredictable forwarding paths that may affect network performance, it is not recommended to establish cross-cluster Layer 2 connectivity between the headquarters and branches. As the main goal of this deployment mode is enabling the headquarter SCP to manage branch HCI clusters, customers only need to ensure the connectivity between the management networks of the headquarter SCP, headquarter HCI cluster, and branch HCI clusters.

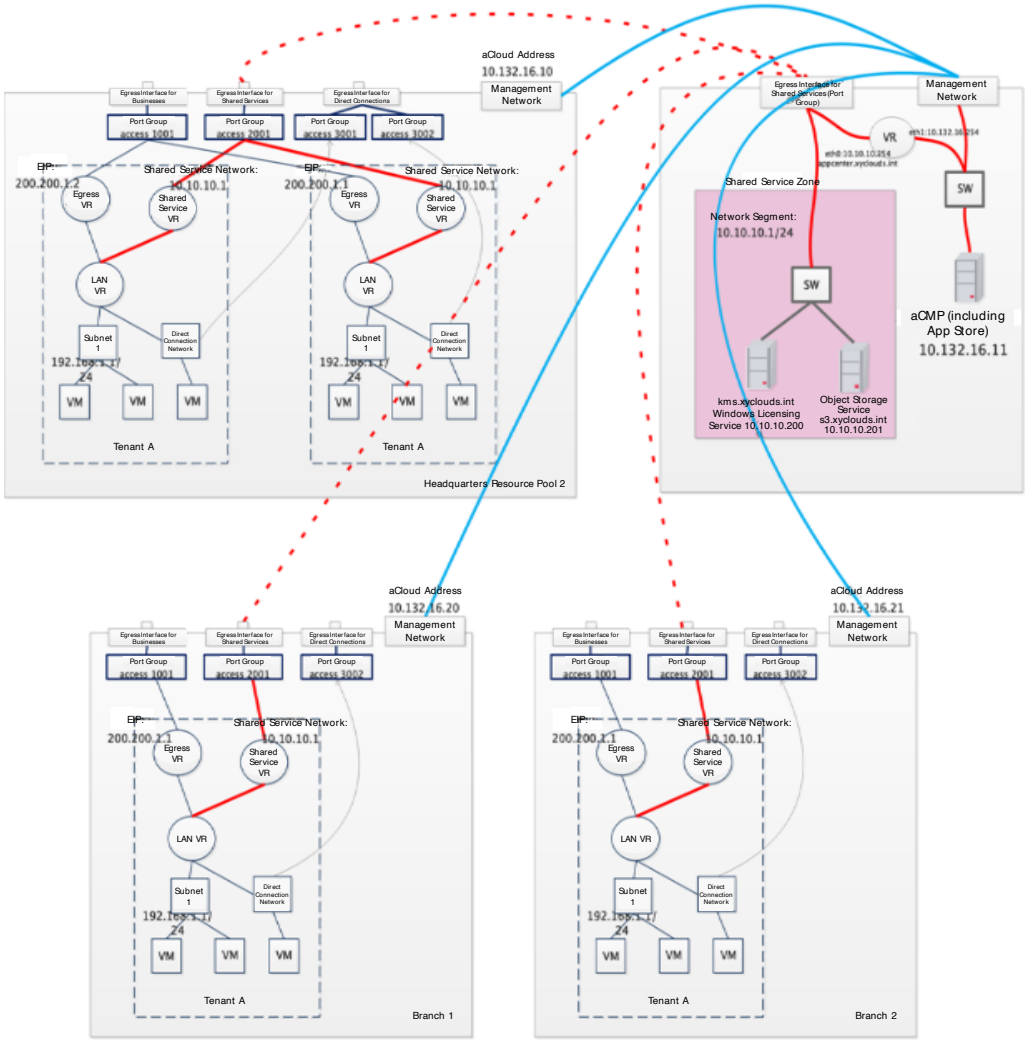


Figure 309: Multiple Clusters Across Headquarters and Branches

5.3.3.5 Virtual Networks

Virtual networks of aNET have two working modes: classic network and VPC.

1. Classic Network

Classic networks are virtual networks used by HCI administrators. In classic network mode, only administrators can plan and maintain the entire network, and there is no longer a logical separation between network management and usage.

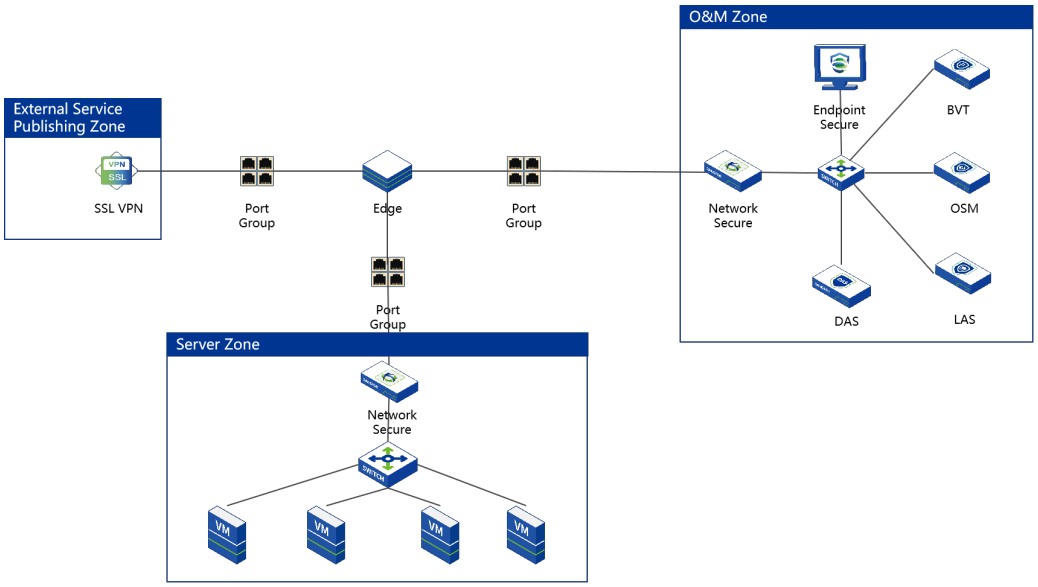


Figure 310: Classic Network Deployment

2. VPC

VPCs are logically isolated network zones for tenants based on classic networks. VPCs have a fixed network topology, where virtual edge routers act as gateways, intranet routers, and subnet switches. Administrators can configure networks for tenants based on their business needs.

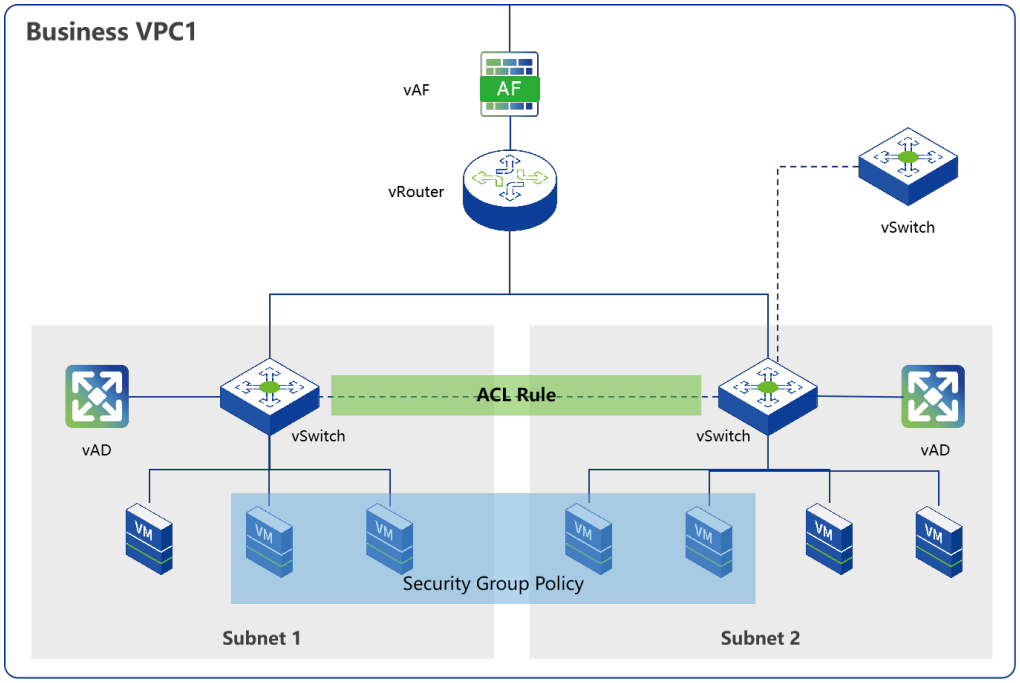


Figure 311: VPC Deployment

5.3.3.6 IPv6 Support

Due to the rise of 5G and IoT technologies and the national policies that promote large-scale IPv6 deployment, customers' demand for IPv6 becomes more urgent. HCI has supported IPv6 to enable IPv6 business and IPv6 transformation.

1. IPv6 support in vRouter

IPv6 support in vRouter includes IPv6 route, DHCPv6, and SLAAC (a method used to automatically configure global IPv6 unicast addresses for devices through ICMPv6 without the need for a DHCPv6 server).

IPv6 route includes static route and policy-based route. The static route allows customers to configure a destination IP address range and the next hop, while the policy-based route allows customers to configure a source IP address range, a destination IP address range, and the next hop. The next hop must be in the same network segment as a specific network interface. Packets with matching destination and source IP addresses are forwarded from the next-hop network interface to the next-hop device.

DHCPv6 allows customers to configure vRouter as a DHCPv6 server. After a VM connects to vRouter, vRouter assigns it an IP address from the DHCP pool and acts as a proxy IPv6 DNS server to provide AAAA domain name resolution for the VM.

HCI also supports IPv6 binding, allowing customers to bind IPv6 addresses to VMs through vmTools static injection.

SLAAC periodically sends RA messages to VMs, notifying them to obtain the global IPv6 unicast addresses and gateway IP addresses. Currently, SLAAC is mainly used for gateway configuration, while DHCPv6 handles IPv6 address assignment.

2. IPv6 support in O&M tools

Sangfor HCI offers built-in network O&M tools to capture IPv6 packets. Customers can use tcpdump to capture packets of specific types (such as ICMPv6 and DHCPv6 packets) from a network interface.

The connectivity test feature supports connectivity tests for IPv6 addresses. The platform pings the destination address on the VM, records the paths of the request and response packets, and displays them on the product interface. If the destination address does belong to the cluster, only virtual devices within the cluster are displayed because the connectivity test feature does not support NAT64 and NAT46.

3. IPv6 support in VMs

IPv6 support in VMs allows customers to configure IPv6 addresses for VMs on the VM management page of HCI. Customers can also view the actual IPv6 address of a VM on the details or list page.

4. IPv6/IPv4 support in the business network and VXLAN

The business network (underlay) and the VXLAN (overlay) support both IPv4 and IPv6.

5. The management and storage networks currently only support IPv4.
6. Before using NAT64 in HCI, configure the vAD NFV component.

5.3.3.7 Low Network Latency (A feature in HCI 6.9.1)

Sangfor has rolled out a low-latency network forwarding solution for the financial industry (such as futures trading), which requires extremely low network latency (below 10 μs) for regular transactions.

5.3.3.7.1 Overview

The low network latency feature of HCI, in combination with low-latency NICs such as Mellanox CX4-LX and Solarflare X2522, allows customers to reduce the platform latency to below 10 μs.

1. Reduced latency

- Supports NIC passthrough to avoid high overhead and high latency at the hypervisor level.

- Introduces low-latency Mellanox NICs and leverages the kernel bypass technology of Mellanox VMA to enable programs to reach NICs without passing through the kernel. This reduces both CPU load and latency.
- Introduces low-latency Solarflare NICs and leverages the kernel bypass technology of Solarflare Onload to enable programs to reach NICs without passing through the kernel. This reduces both CPU load and latency.

2. Enhanced network stability

- Hypervisor CPU cores and caches are divided into groups for scheduling, which prevents network jitter caused by CPU resource contention between HCI services and VMs.
- vCPUs are assigned to run on particular physical CPU cores, which eliminates the resource contention between VMs and the scheduling overhead between vCPU cores, increases CPU cache hit rate, and ensures network latency stability.
- VPP-supported 2 MB huge-page memory, system interruption isolation, KVM instruction optimization, and virtual external interruption optimization are enabled at the hypervisor level. This greatly enhances virtualization performance while maintaining stable network latency for VMs.

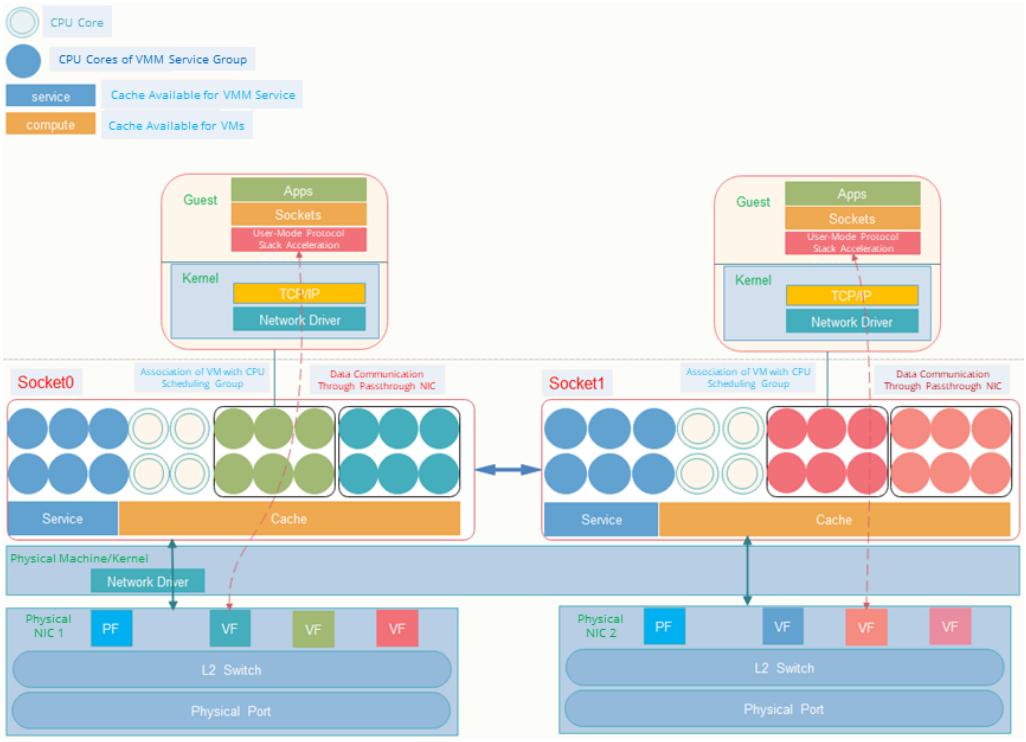


Figure 312: Low Latency Principle

5.3.3.7.2 Key Technologies for Low-Latency Networks

1. Group scheduling of CPU cores and caches

CPU cores and L3 caches are divided into service and compute groups. The computing group focuses on the vCPU threads of VMs, while the service group serves processes and threads other than vCPU threads to avoid network jitter caused by CPU contention between virtualization services and VMs.

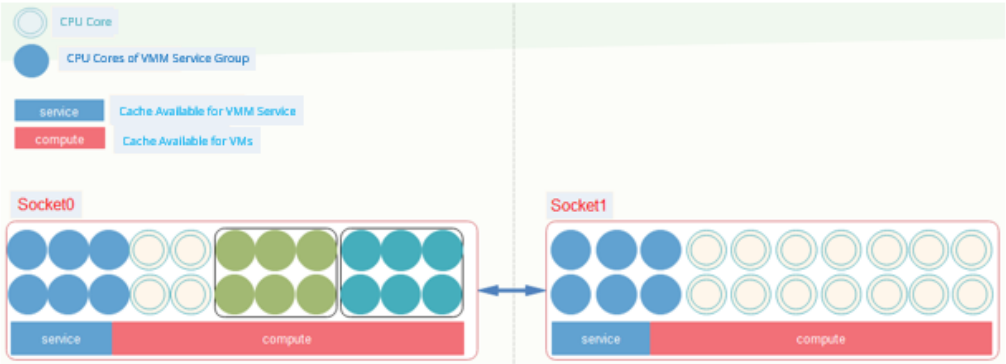


Figure 313: Group Scheduling of CPU Cores and Caches

2. SR-IOV for VMs

Once SR-IOV is enabled, customers can create PFs and VFs based on physical NICs, where each VF possesses an exclusive PCI configuration space. VMs can directly exchange data with VFs of the physical NIC without passing through the hypervisor layer, achieving almost the same performance as that of a physical environment.

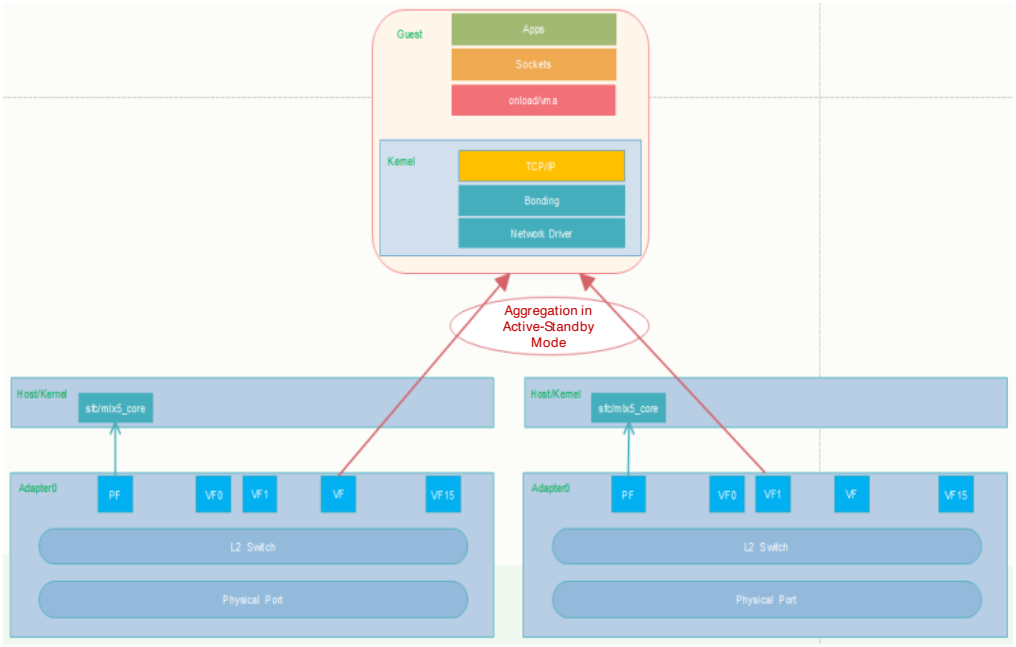


Figure 314: SR-IOV

3. CPU affinity for VMs

CPU affinity for VMs eliminates the CPU contention between VMs and the overhead caused by scheduling vCPU cores across physical CPU cores, increases CPU cache hit rate, and ensures network latency stability.

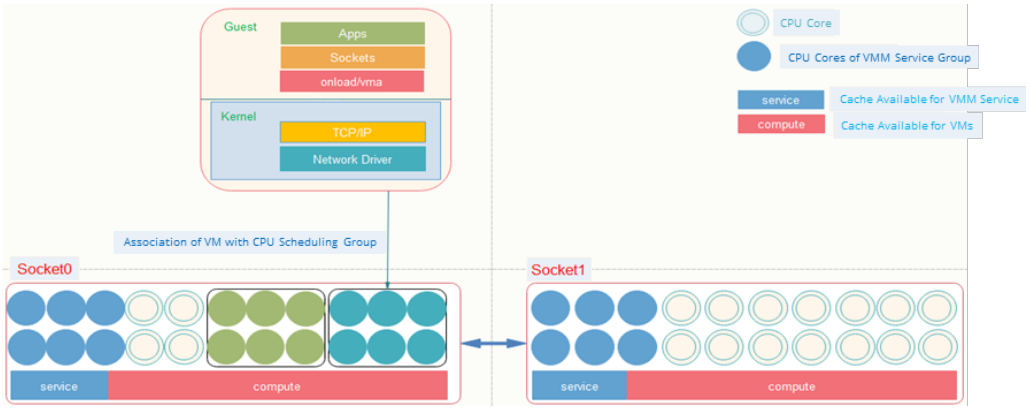


Figure 315: CPU Affinity for VMs

4. Onload/VMA acceleration for direct packet handling on the NIC

Onload/VMA acceleration enables direct packet handling on the NIC by bypassing the kernel protocol stack. This avoids the overhead caused by context switching and data copying between applications and the kernel.

- Onload acceleration: Solareflare's kernel bypass software solution is compatible with socket programming interfaces. It allows customers to bypass the kernel protocol stack for direct packet handling on the NIC.

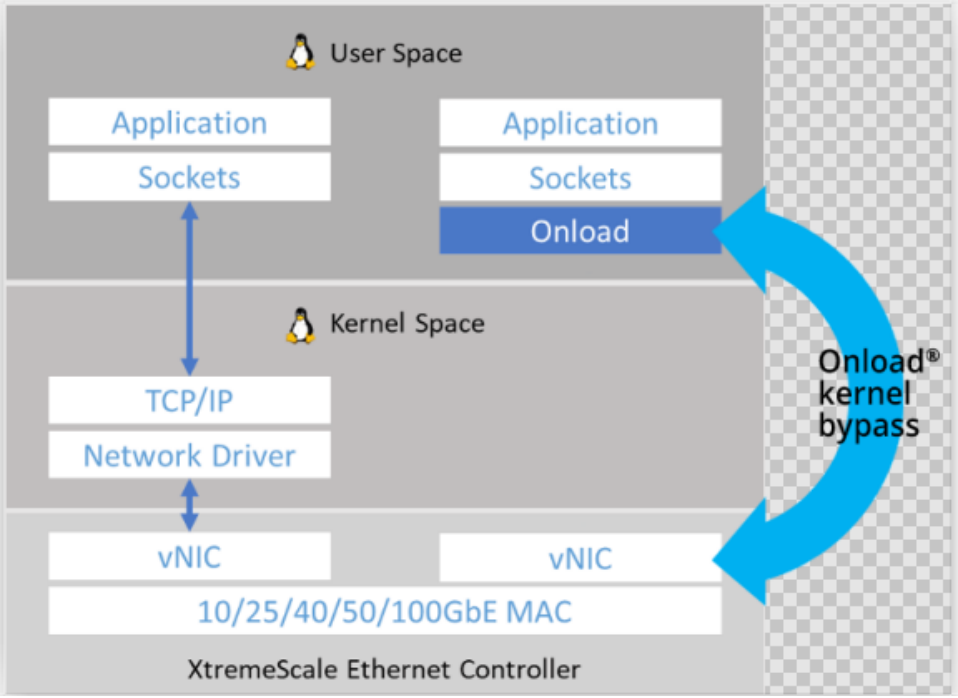


Figure 316: Onload Acceleration Principle

- VMA acceleration: Mellanox Messaging Accelerator (VMA) is a user protocol stack library that is based on the bifurcated driver model and compatible with the POSIX APIs. It allows customers to accelerate TCP/UDP messaging without making any changes to applications. To accelerate network applications coded in C based on Mellanox's physical NIC passthrough technology, customers only need to install the Mellanox NIC driver and VMA on VMs.

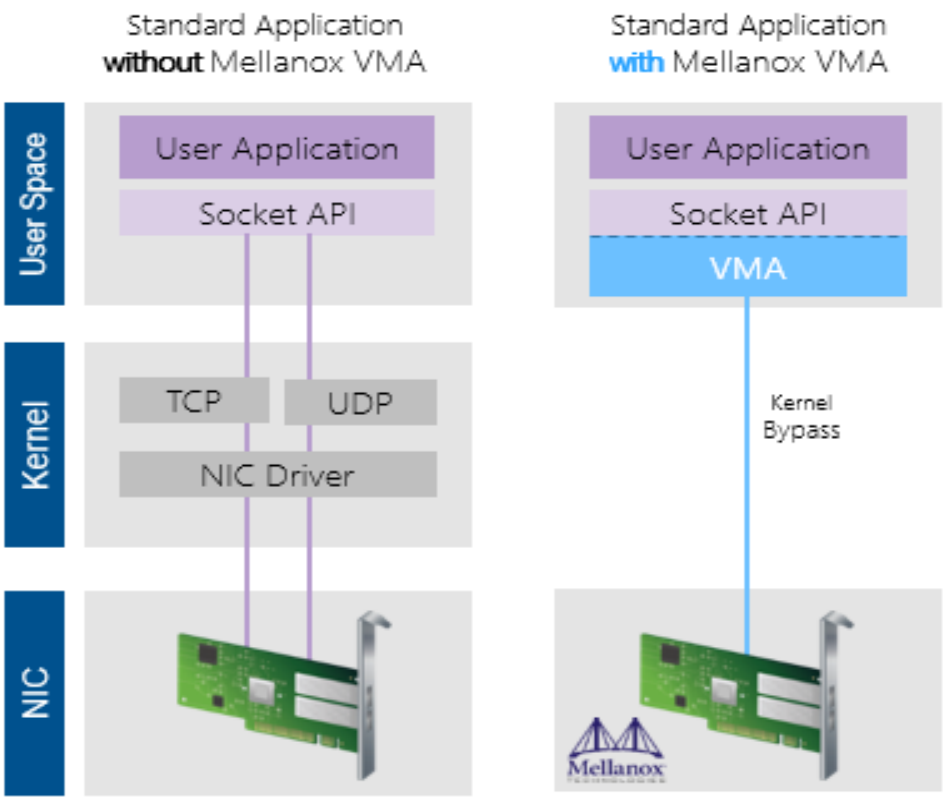


Figure 317: VMA Acceleration Principle

5.3.4 High-Performance Network

5.3.4.1 DPDK-Based High-Performance Network

Intel Data Plane Development Kit (DPDK) is a toolkit that offers library functions and drivers for efficient userspace packet processing on Intel architecture (IA) processors. Unlike Linux's general-purpose design, it is specialized for high-performance packet handling in network applications. The DPDK application runs in userspace and uses its own data plane libraries to handle packets without passing through Linux's kernel protocol stack. For the Linux kernel, the DPDK application is an ordinary userspace process that is compiled, linked, or loaded in the same way as any other standard program. The DPDK application consists of one main thread and multiple child threads that are bound to and run on specified CPU cores. DPDK is now part of the Linux Foundation, offering support for diverse instruction set architectures

(such as Arm, x86, and Power), DMA engines, and hardware and software NICs. The DPDK-based data plane of aNET is designed for network I/O-intensive applications. It supports both dedicated and generic NICs. The forwarding flow at the application layer has been rewritten and simplified to ensure enhanced stability in the data plane.

5.3.4.1.1 Disadvantages of OS Kernel-Based Data Transmission

1. Interrupt handling. An influx of data packets in the network can trigger frequent hardware interrupt requests. These interrupts can disrupt ongoing lower-priority software interrupts or system calls, leading to high performance overhead when they occur frequently.
2. Memory copying. Typically, the flow of a network packet from the NIC to an application is as follows: The packet is first sent from the NIC to a kernel-allocated buffer using methods such as DMA and then copied from the kernel space to the userspace. The copying operation is time-consuming and sometimes even accounts for 57.1% of the total packet processing time.
3. Context switching. Frequent hardware and software interrupts can preempt system call execution, leading to substantial context switching overhead. Additionally, in a multi-threaded server framework, cross-thread scheduling can cause extra overhead due to frequent context switching, as well as lock contention.
4. Local failure. As modern mainstream processors have multiple cores, a packet may be processed across CPU cores. For example, a packet may undergo an interruption on CPU 0, kernel-mode processing on CPU 1, and userspace processing on CPU 2. This cross-core packet processing can lead to CPU cache failure, resulting in local failure. In a NUMA architecture, performance is greatly affected by access across NUMA memory.
5. Memory management. The size of the memory pages in traditional servers is 4 KB. Increasing the number of entries in the cache mapping table speeds up memory access and minimizes cache misses but is at the expense of CPU retrieval efficiency.

5.3.4.1.2 Solutions

To tackle the disadvantages of OS kernel-based data transmission, the industry has promoted the following solutions:

1. Separate the control and data layers. Execute tasks such as packet processing, memory management, and processor scheduling in the userspace, while the kernel only handles some control instructions. This eliminates the previously mentioned problems related to system interrupts, context switching, system calls, and system scheduling.
2. Use multi-core programming instead of multi-threading, and set CPU affinity to bind each thread to a CPU core. This minimizes scheduling and switching between CPU cores.
3. For NUMA systems, use the memory of the NUMA node where the CPU core resides to avoid access across NUMA memory.
4. Replace regular memory with huge-page memory to reduce cache misses.
5. Leverage lock-free technologies to prevent resource contention.

5.3.4.1.3 Implementation of Kernel Bypass

DPDK stands out from numerous high-performance network data processing frameworks. Compared to traditional kernel-based network data processing, DPDK has revolutionized the network data flow from the kernel space to the userspace.

- Network data flow in traditional Linux kernel-based frameworks:

Hardware interrupt > Packet retrieval and distribution to the kernel thread > Software interrupt > Packet processing by the kernel thread in the protocol stack > User-layer notification upon completion

User-layer packet reception > Network layer > Logic layer > Business layer

- Network data flow in DPDK:

Hardware interrupt > Interrupt process termination

User-layer packet retrieval via device mapping > Entry into the user layer

protocol stack > Logic layer > Business layer

5.3.4.1.4 Key DPDK Technologies

1. Userspace I/O

Userspace I/O (UIO) can intercept interrupts and reset interrupt callback behavior to avoid subsequent processing in the kernel protocol stack, which enables DPDK to bypass the kernel protocol stack.

UIO devices are implemented by exposing file interfaces to the userspace. For example, when a UIO device such as uioX is registered, the system generates the file in the /dev/uioX directory. Accessing the file is equivalent to accessing the device's memory. Additionally, devices can be controlled by performing read or write operations on files in the /sys/class/uio directory.

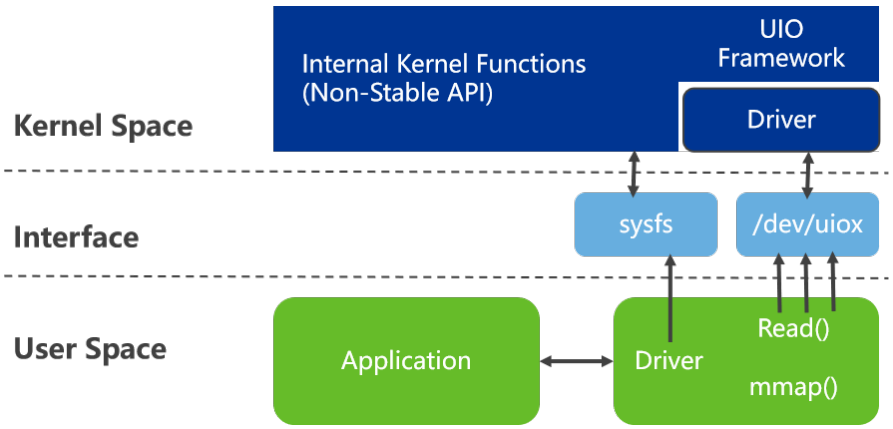


Figure 318: UIO Framework

2. Memory pool

DPDK offers a sophisticated memory pooling technique in the userspace to ensure control transfer between the kernel space and userspace, eliminating the need for memory copying between them. This reduces the overhead of memory copying during packet transmission and reception.

3. Huge-page memory management

DPDK provides APIs for huge-page memory allocation, usage, and release, through which the upper-layer applications can easily request and utilize huge-page memory or request regular memory.

4. Lock-free circular queue

DPDK has implemented a lock-free mechanism based on the lock-free circular buffer of the Linux kernel called kfifo. The lock-free mechanism supports single-producer enqueuing, single-consumer dequeuing, multi-producer enqueuing, and multi-consumer dequeuing operations. It ensures high performance and data synchronization during data transmission.

5. PMD

DPDK discards the interrupt mode for NIC drivers and receives packets in the polling mode to avoid interrupt overhead.

6. NUMA

DPDK leverages memory information from the **proc** file system to allocate memory nearest to the node where the CPU core resides, avoiding performance issues caused by access across NUMA node memory.

7. CPU affinity

DPDK employs CPU affinity to bind threads to specific CPUs to avoid random CPU scheduling during thread execution. This eliminates the overhead from frequent context switches, prevents CPU cache local failures, and increases the CPU cache hit rate.

8. Multi-core scheduling framework

DPDK is based on a multi-core scheduling framework, where cores are assigned the primary or secondary roles. The primary cores initialize modules, while the secondary cores handle specific business tasks.

5.3.4.2 Multiple Queues and Interrupt Binding For NICs

Process of handling a packet from a NIC by CPU: When the NIC receives a packet, it generates an interrupt to notify the CPU of the new packet. The CPU then calls the interrupt handler to copy the packet from the NIC buffer to the memory for response. As the NIC has a limited cache, it discards subsequent packets when the cache is full because packets are not copied out in time. To avoid this, the CPU must swiftly copy packets to the memory. As network I/O bandwidth increases, the CPU resource becomes the performance bottleneck

because a single CPU is unable to handle interrupts from multiple NICs or process interrupts from a high-speed NIC using a single queue.

HCI adjusts interrupt binding and the number of queues according to the interrupt affinity policies for the queues of an NIC. NUMA node affinity is ensured for the interrupt core and forwarding core of the managed network interface to avoid the overhead caused by remote access; the interrupt core of the VS network interface and the forwarding core are separated to prevent CPU contention; two queues are used when the interface transfer rate exceeds the threshold to avoid single-core bottlenecks. HCI binds interrupts to different CPU cores to prevent possible single-CPU core bottlenecks arising from increased network I/O bandwidth. This enhances packets per second (PPS) and bandwidth performance.

Optimization in multiple queues and interrupt binding for NICs can improve the throughput of each host NIC by more than 30%, leading to an increase of more than 30% in the high-depth, large-block write performance of VMs.

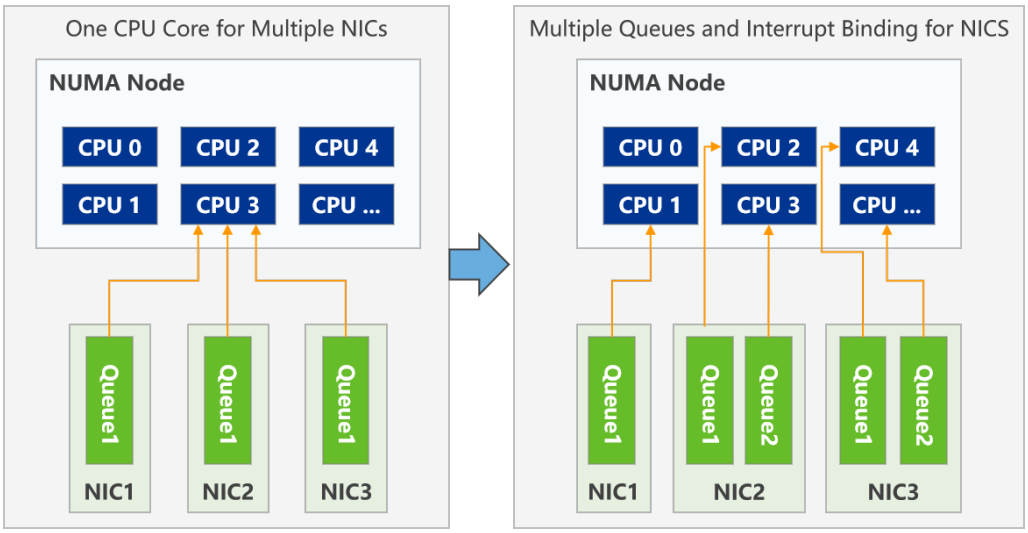


Figure 319: Multiple Queues and Interrupt Binding For NICs

HCI dynamically adjusts the number of NIC queues and the interrupt binding policies to prevent single-core bottlenecks, ensure NUMA node affinity between the interrupt core and HCI network service, and avoid performance loss arising from access across NUMA memory, thus boosting the network performance of HCI.

5.3.4.3 NUMA Node Affinity

Modern mainstream servers adopt a NUMA architecture, where local memory access is faster than remote memory access because NUMA nodes have independent memory controllers. Arm or C86 servers usually have more NUMA nodes. For example, processors like Hygon 5280 and Kunpeng 920 are designed with 4 NUMA nodes each. However, the more NUMA nodes a server has, the more likely a VM and its data plane process run on different NUMA nodes. In this case, communication across NUMA nodes between the VM and the data plane process can cause a decline in network forwarding performance.

To enhance network forwarding performance on Arm and C86 architectures, HCI has integrated NUMA node affinity into the underlying virtualization to forcibly bind a VM and its data plane process to the same NUMA node. This enables local memory access and reduces latency during packet transmission and reception for higher network forwarding performance.

HCI allows customers to configure NUMA node affinity to bind a VM and its data plane process to the same NUMA node for enhanced network performance. When you enable NUMA node affinity for a VM, the VM preferably runs on the same NUMA node as its data plane process. This increases the network performance by more than 20% compared to scenarios where NUMA node affinity is not enabled.

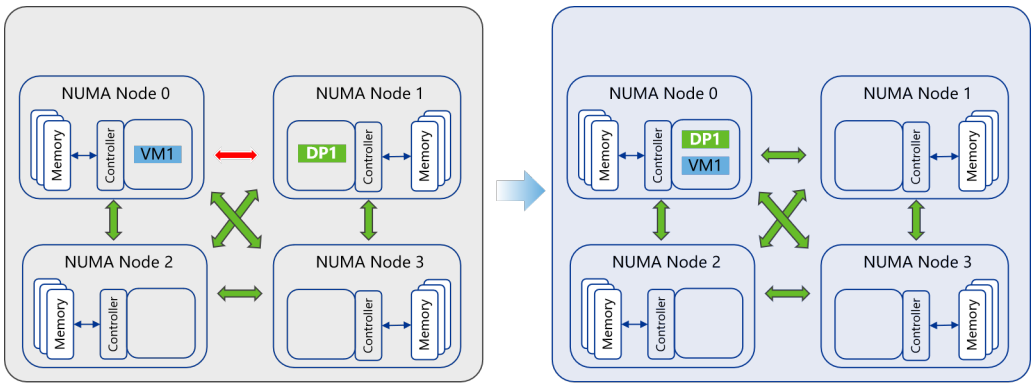


Figure 320: NUMA Node Affinity

As shown in the preceding figure, when NUMA node affinity is enabled for a

VM, the VM preferably runs on the NUMA node where its data plane process runs. When NUMA node affinity is enabled for a VM, VM scheduling may include the following scenarios:

1. When a VM requires two NUMA nodes due to a large number of cores, the data plane process can only run on one of two NUMA nodes.
2. When a NUMA node is overloaded (exceeding 130% of its physical core capacity), VMs with NUMA node affinity disabled are migrated first. However, if all VMs on the overloaded NUMA node have NUMA node affinity enabled, they are migrated like regular VMs.



When exporting, importing, cloning, taking snapshots of, or migrating a VM, default HCI policies are used. When exporting, cloning, taking snapshots of, or migrating a VM, configurations are retained. When importing a VM, only recognized configurations are obtained. When migrating a VM across clusters with different versions, configurations are handled by the destination cluster. If the destination cluster has an earlier version, configurations are not retained.

For servers that have multiple NUMA nodes, NUMA node affinity allows customers to bind a specific VM and the network forwarding process to the same NUMA node for stable performance.

5.3.5 Virtual Network Elements

5.3.5.1 Distributed Virtual Switches

Switches isolate devices in a physical network into separate broadcast domains, where devices connected to the same switch form a Layer 2 network. A distributed virtual switch (DVS) inherits the capabilities of a regular switch. VMs connected to the same DVS are logically within the same Layer 2 network (a subnet).

In real-world cluster environments, VMs are likely to run on different physical nodes, and physical network connections between clusters may be over Layer

3. In this case, to include the VMs in the same Layer 2 network, a VXLAN is introduced based on VXLAN technologies for packet forwarding. In terms of packet encapsulation, physical nodes act as VTEPs to encapsulate or decapsulate Layer 2 packets at Layer 3, ensuring that the packets sent or received by VMs are original Layer 2 frames. VTEPs then forward the packets according to the MAC addresses of the source and destination VMs from the original frames. VTEPs are deployed in a distributed mode, where each physical host in the cluster has a VTEP to complete the ultimate Layer 2 packet forwarding (within the same VNI), as if there were a virtual switch that connects all physical hosts in the cluster forwarding Layer 2 packets for VMs.

In addition to Layer 2 forwarding, DVS also supports advanced features such as ARP broadcast storm prevention, port mirroring, port aggregation, and QoS.

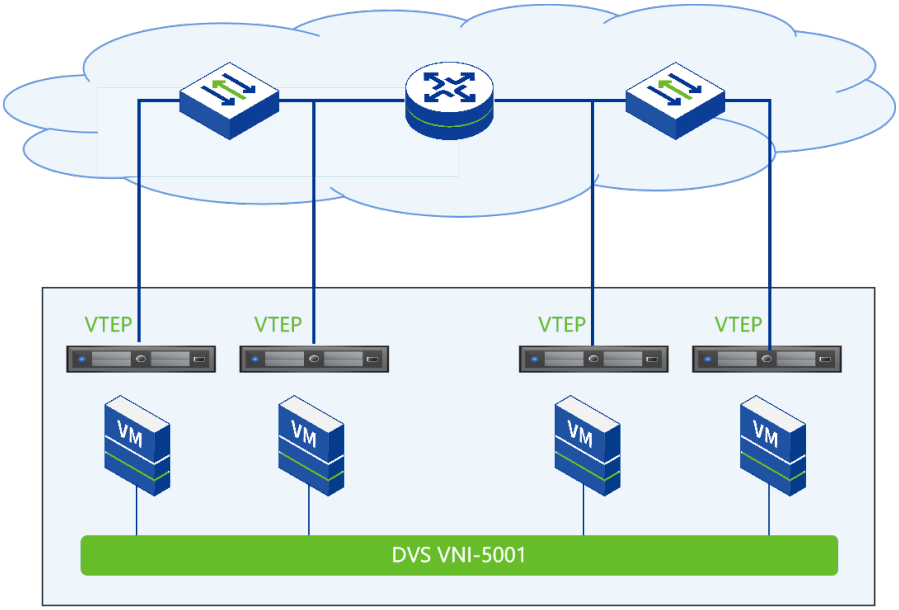


Figure 321: DVS

5.3.5.2 Edge Virtual Routers

An edge virtual router (EVR) is a network element for Layer 3 forwarding within a virtual network, which is deployed on two physical nodes in the form of primary and secondary instances. In addition to routing, EVR also offers the advanced features of physical routers such as NAT, ACL, and DHCP. EVR currently supports IPv4/IPv6 addresses, static IPv4/IPv6 routes, policy-based

routes, and DHCP pools. However, IPv6 ACL is not yet supported in EVR.

Under normal conditions, only one primary EVR is active on a fixed node, while the secondary EVR remains inactive. The primary and secondary EVRs check each other's status by sending heartbeats via the VXLAN interface. When the primary EVR fails, the VXLAN interface is switched to the secondary EVR that becomes the new primary EVR, ensuring the high availability of EVR. EVRs can also be migrated to physical nodes with lower network forwarding overhead for better network performance, which does not cause any business interruption.

Here is an example of PC 1 in the physical network accessing VM 1, as shown in the following figure. 1) PC 1 transmits data to the EVR via the business network interface; 2) the EVR sends data to DVS of Host 2; 3) DVS of Host 2 sends data to DVS of Host 1; 4) DVS of Host 1 sends data to VM 1.

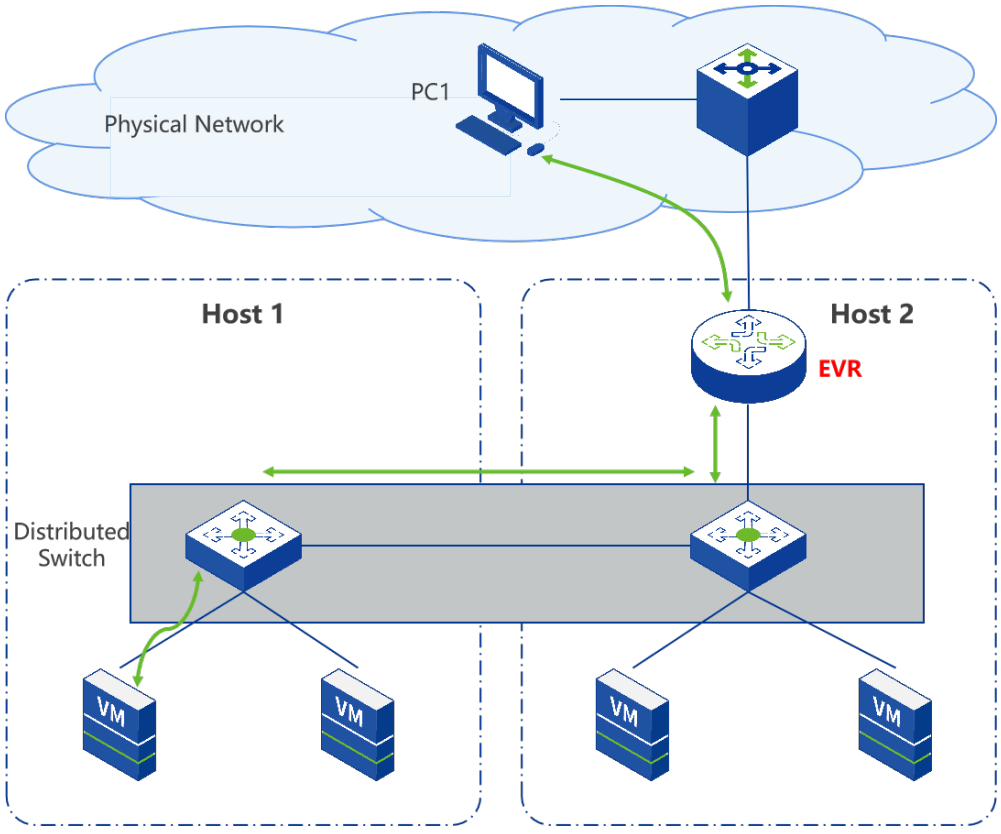


Figure 322: Traffic Forwarding in EVR

5.3.5.3 Edge Virtual Switches

An edge virtual switch (EVS), also known as an edge, is the edge between a virtual network and a physical network that allows the VMs in the cluster to communicate with external devices. After you connect an EVS to the network interface of a host in bridge mode, VMs on that host can communicate with devices in the physical network. EVS supports VLAN for seamless integration with the VLAN configurations of physical switches. EVS can be used in the following scenarios:

1. Neither the physical switch nor the VM within the cluster is configured with a VLAN.

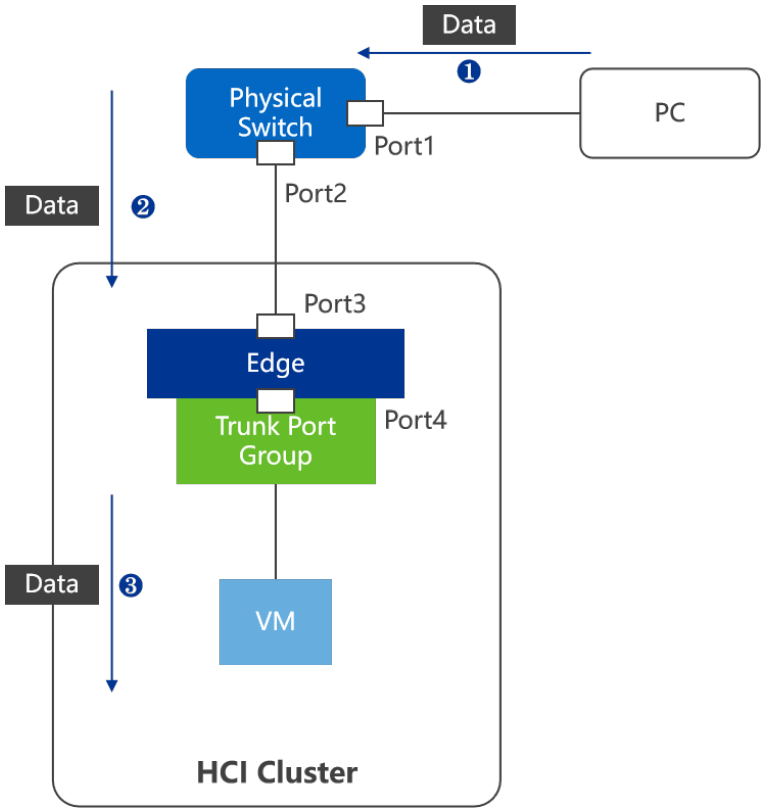


Figure 323: EVS Scenario 1

- Configurations

Port 1: Trunk All (pvid 1)

Port 2: Trunk All (pvid 1)

Port 3: Trunk All (pvid 1)

Port 4: Trunk Port Group

- Forwarding process

The external PC accesses the VM in the cluster via Port 1 (a trunk port). After passing through Port 1, the traffic comes out from Port 2 without any tag. After passing through Port 3 (a trunk port whose pvid is 1) and Port 4 (a trunk port group), the traffic reaches the VM without any tag.

2. The physical switch is configured with an access VLAN, while the VM within the cluster is not configured with a VLAN.

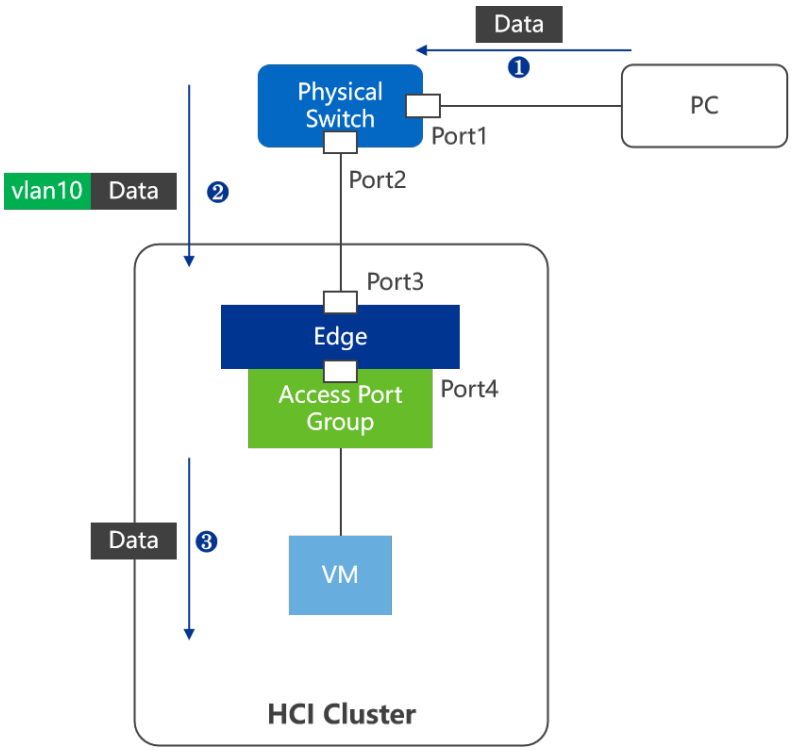


Figure 324: EVS Scenario 2

- Configurations

Port 1: Access VLAN 10

Port 2: Trunk VLAN 1-10

Port 3: Trunk All (pvid 1)

Port 4: Access Port Group (VLAN 10)

- Forwarding process

The external PC accesses the VM in the cluster via Port 1 (access VLAN 10). After passing through Port 1, the traffic comes out from Port 2 with a tag (VLAN 10). After passing through Port 3 (a trunk port whose pvid is 1) and Port 4 (an access port group with VLAN 10 configured), the tag is removed before the traffic reaches the VM.

3. The physical switch is configured with an access VLAN, while the VM (virtual network device) within the cluster is configured with a VLAN.

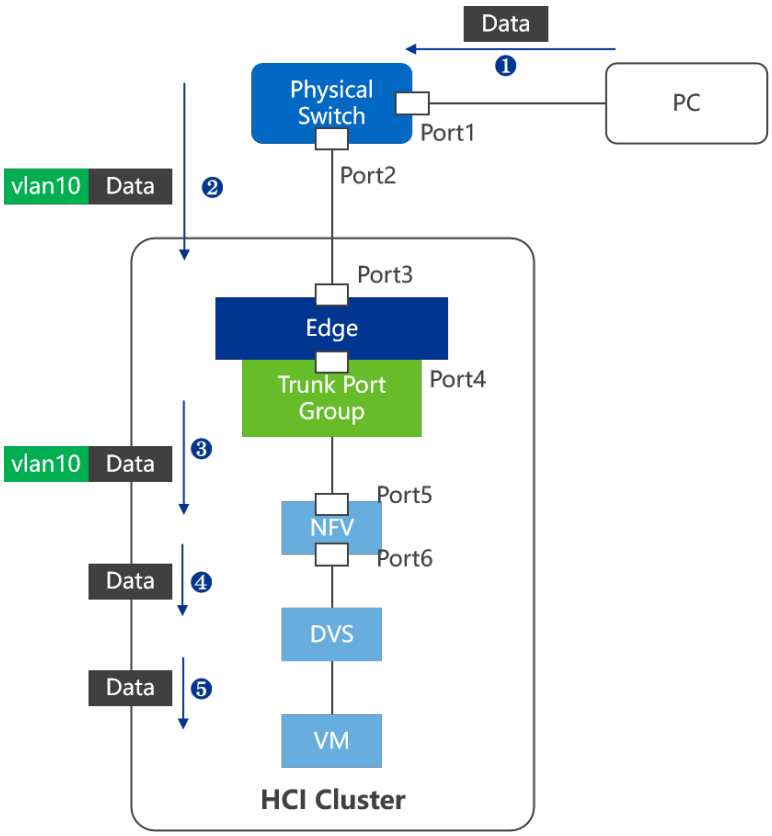


Figure 325: EVS Scenario 3

- Configurations

Port 1: Access VLAN 10

Port 2: Trunk VLAN 1-10

Port 3: Trunk All (pvid 1)

Port 4: Trunk Port Group (VLAN 1-10)

Port 5: Trunk Port (VLAN 1-10)

Port 6: Access Port (VLAN 10)

- Forwarding process

The external PC accesses the VM in the cluster via Port 1 (access VLAN 10). After passing through Port 1, the traffic comes out from Port 2 with a tag (VLAN 10). After passing through Port 3 (a trunk port whose pvid is 1) and Port 4 (a trunk port group with VLAN 1-10 configured), the traffic is forwarded to NFV with the tag. After passing through Port 5 (trunk all) and Port 6 (an access port with VLAN 10 configured), the tag is removed before the traffic is forwarded to DVS. DVS then forwards the traffic to the VM.

5.3.6 Network Traffic Forwarding Models

5.3.6.1 BUM Traffic Forwarding

5.3.6.1.1 Unicast Replication Mode

aNET forwards broadcast, unknown unicast, and multicast traffic (also known as BUM traffic) in unicast replication mode, which does not require the underlay network to support multicast. The BUM traffic is forwarded in unicast replication mode from one VTEP to another VTEP within the same VNI. VTEPs can be in one or multiple network segments and configured with gateways, and all the VTEPs can communicate with each other. In unicast replication mode, all VTEPs in the same VNI can receive BUM traffic.

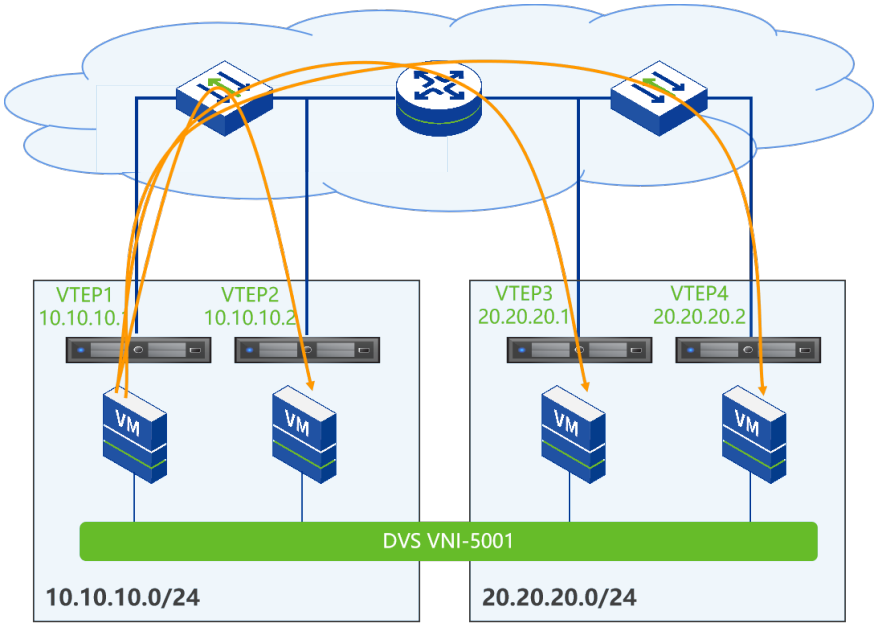


Figure 326: BUM Traffic in Unicast Replication Mode

The following figure is an example of VM 1 sending multicast traffic to VM 3, which can demonstrate the BUM traffic forwarding model.

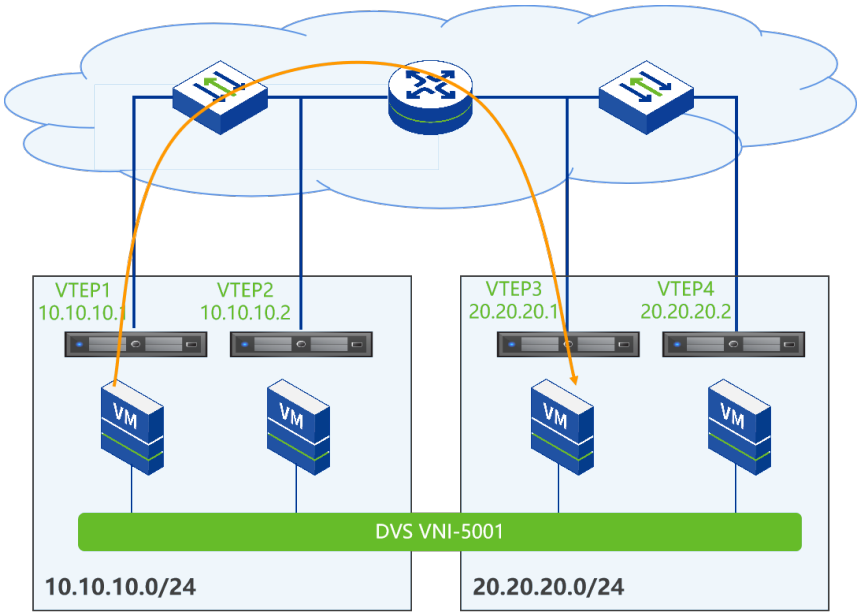


Figure 327: Multicast Traffic From VM 1 to VM 3

1. VM 1 sends a multicast packet to VM 3 through DVS, but DVS does not find any entry from the FDB table.

2. DVS encapsulates the multicast packet in unicast mode and sends the packet to each VTEP within its VXLAN based on the VTEP table. This process is referred to as unicast replication.
3. VM 2, VM 3, and VM 4 all receive this multicast packet, but VM 2 and VM 4 discard the packet because the destination MAC does not match. However, they all learn the FDB table entry of VM 1, as shown in the following table.

MAC	OUT-INTERFACE	VLAN	HOTS-IP
VM 1-MAC	VTEP 3-TUNNEL	VM 1-VLAN	10.10.10.1

Table 15: MAC Entry Learned by DVS on VTEP 3

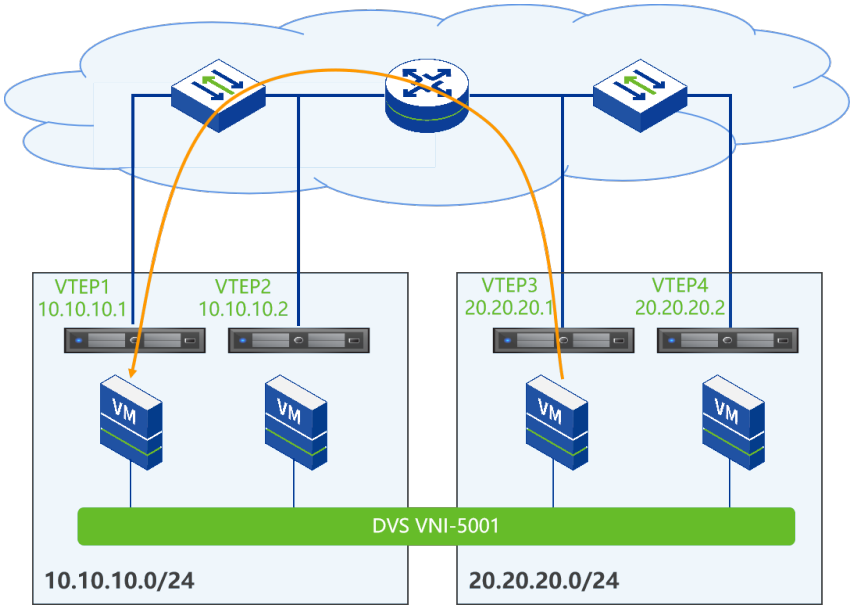


Figure 328: Response Multicast Traffic From VM 3 to VM 1

1. Only VM 3 responds to this multicast packet.
2. VTEP 3 encapsulates the response packet in unicast mode due to the FDB table entry learned in Step 3 and then sends the encapsulated packet to VTEP 1.
3. VTEP 1 decapsulates the packet, learns the MAC entry of VM 3, and forwards the decapsulated packet to VM 1.

5.3.6.1.2 ARP/ND Proxy

aNET data plane supports ARP proxy for ARP traffic. When a VM sends an ARP request, the controller acts as an ARP proxy for querying the ARP information on behalf of the VM and returns the query result to the VM. Only when the controller is unable to retrieve ARP information, unicast replication mentioned in the preceding section is used for ARP traffic forwarding. This typically occurs when a VM comes online and has not updated ARP information to the controller in time. Under normal conditions, ARP traffic is forwarded using an ARP proxy.

aNET also supports Neighbor Discovery (ND) proxy for IPv6 ND traffic. ND proxy is implemented in the same way as ARP proxy, and their functionalities are the same. Here is an example of how an ARP proxy works.

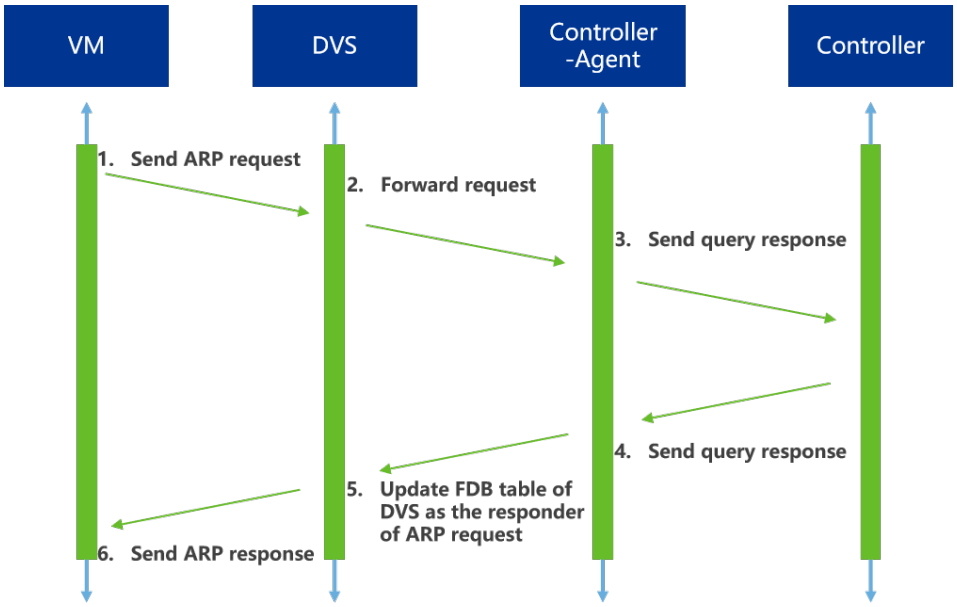


Figure 329: ARP Proxy Principle

1. The VM sends an ARP request to DVS.
2. DVS forwards the request to controller-agent.
3. When receiving the request, controller-agent sends a query message to controller.
4. After processing, controller returns a response to controller-agent

5. When receiving the response, controller-agent updates the FDB table of DVS as the responder of the ARP request.
6. DVS sends an ARP response to the VM.

5.3.6.2 Unicast Traffic Forwarding

5.3.6.2.1 Single-Data Center, Single-Cluster Unicast Traffic Forwarding

There are 14 common network topologies for single-data center and single-cluster deployment, and their traffic forwarding paths are as follows:

Scenario 1: Intra-physical host network communication within a Layer 2 network when DVS is deployed

Traffic forwarding path: Two business VMs running on the same physical host communicate through DVS.

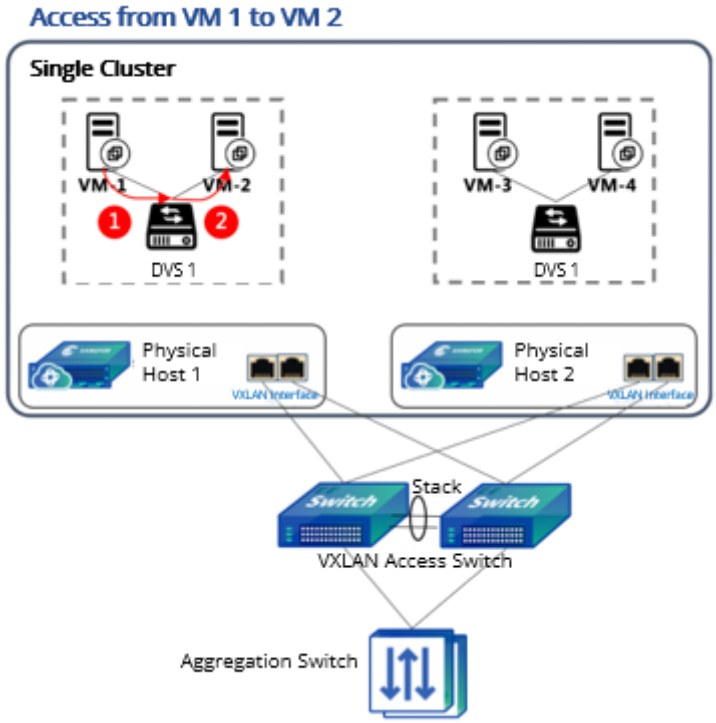


Figure 330: Traffic Forwarding Path for Intra-Host Communication Within a Layer 2 Network

Scenario 2: Cross-physical host network communication within a Layer 2 network when DVS is deployed

Traffic forwarding path: Two business VMs running on different physical hosts communicate through DVS over the VXLAN.

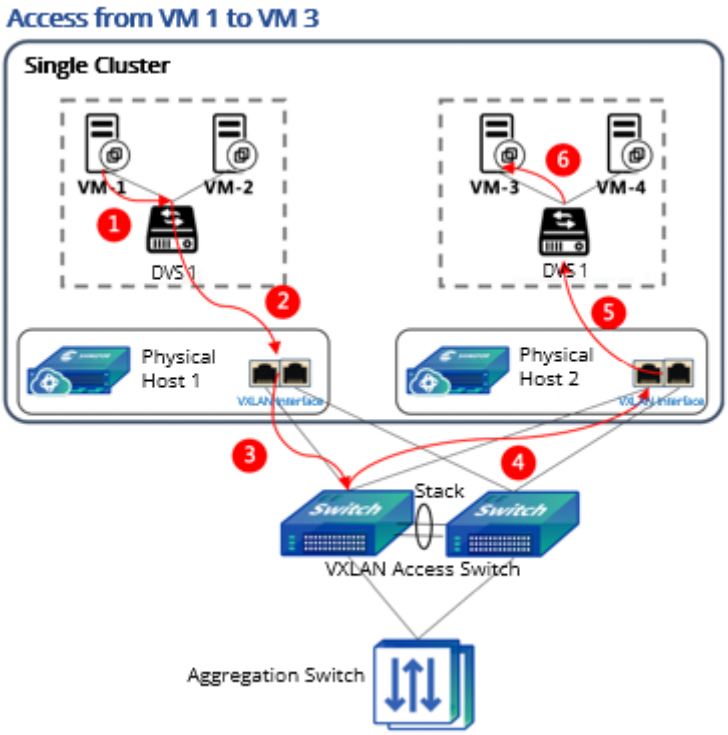


Figure 331: Traffic Forwarding Path for Cross-Host Communication Within a Layer 2 Network

Scenario 3: Intra-physical host network communication within a Layer 2 network when edges are connected in series and DVS is deployed

Traffic forwarding path: Two business VMs running on the same physical host communicate through DVS. In this scenario, the edges are only used for external network communication.

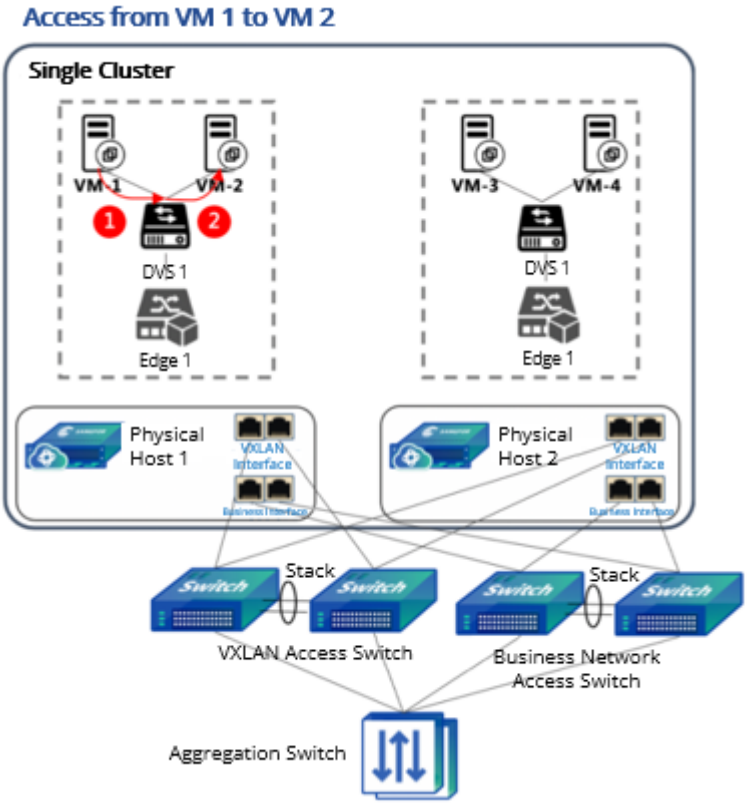


Figure 332: Traffic Forwarding Path for Intra-Host Communication Within a Layer 2 Network with Edges Connected in Series and DVS Deployed

Scenario 4: Cross-physical host network communication within a Layer 2 network when edges are connected in series and DVS is deployed

Traffic forwarding path: Two business VMs running on different physical hosts communicate through DVS and edges over the business network.

The reason why the traffic is not forwarded through the VXLAN interface: DVS is connected to the edge. To avoid loops that may occur when both the VXLAN and business network can forward traffic, the system is designed to convert a virtual switch connected to an edge into a local virtual switch that is not in distributed mode. This prevents loops because only the business network forwards east-west traffic.

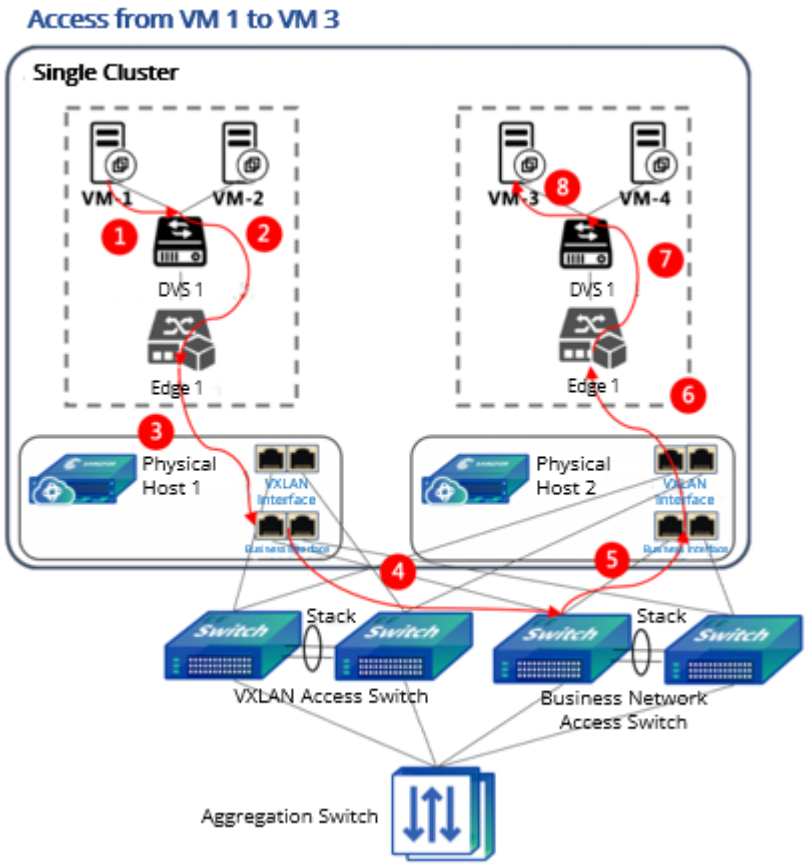


Figure 333: Traffic Forwarding Path for Cross-Host Communication Within a Layer 2 Network with Edges Connected in Series and DVS Deployed

Scenario 5: Intra-physical host network communication when VMs are directly connected to edges

Traffic forwarding path: Two business VMs running on the same physical host communicate through the directly connected edge.

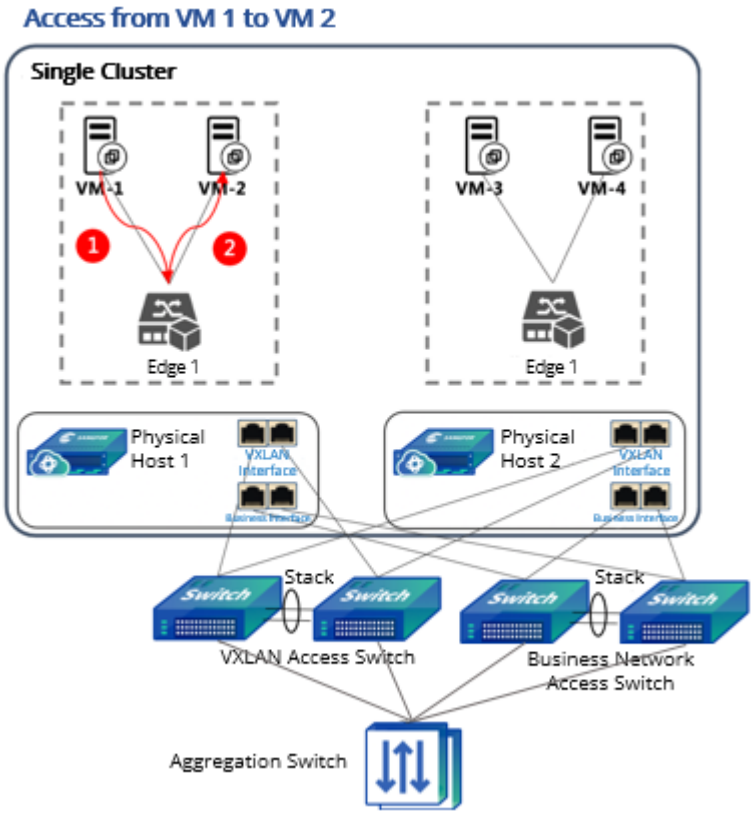


Figure 334: Traffic Forwarding Path for Intra-Host Communication with VMs Directly Connected to Edges

Scenario 6: Cross-physical host network communication when VMs are directly connected to edges

Traffic forwarding path: Two business VMs running on different physical hosts communicate through the directly connected edges over the business network.

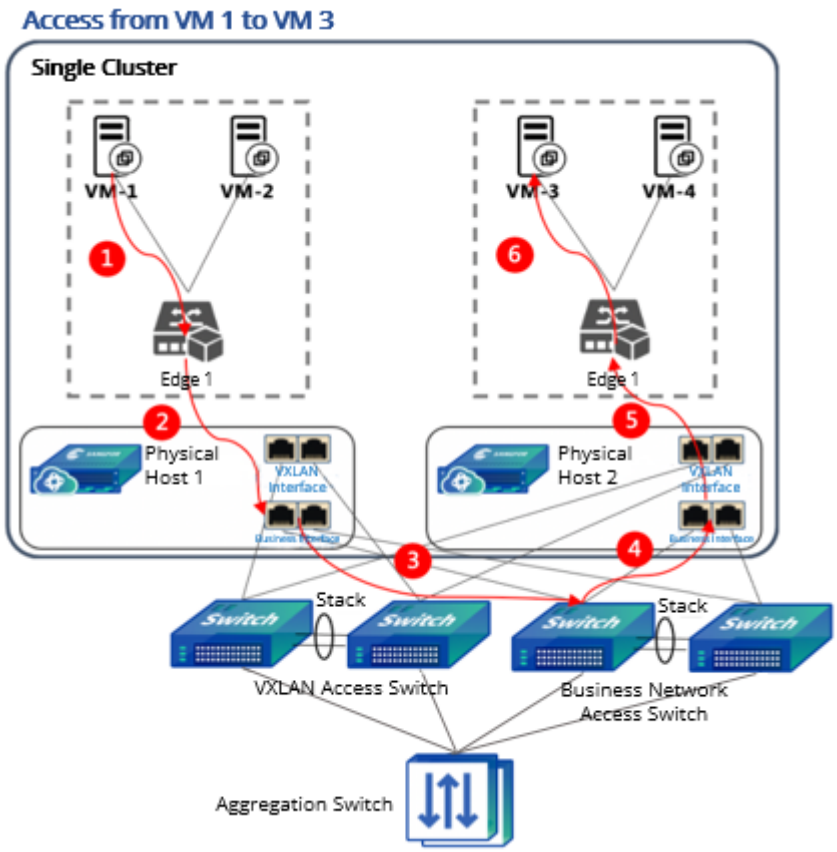


Figure 335: Traffic Forwarding Path for Cross-Host Communication with VMs Directly Connected to Edges

Scenario 7: Intra-physical host network communication across Layer 2 networks when DVS is deployed

Traffic forwarding path: Two business VMs that run on the same physical host with different Layer 2 networks communicate through DVS and the virtual router.

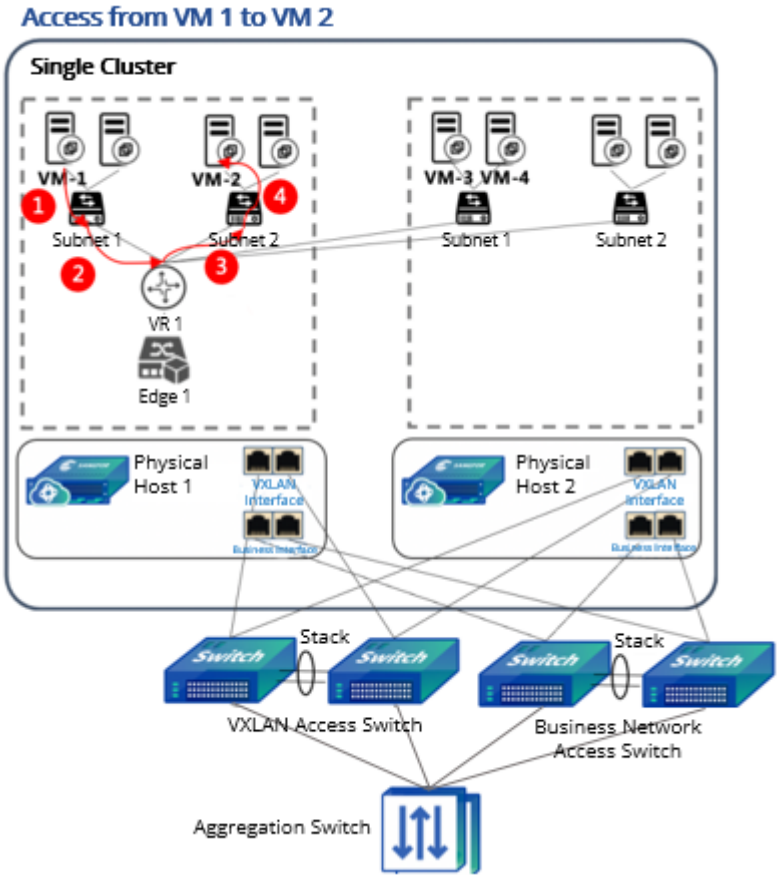


Figure 336: Traffic Forwarding Path for Intra-Host Communication Across Layer 2 Networks

Scenario 8: Cross-physical host network communication across Layer 2 networks when DVS is deployed

Traffic forwarding path: Two business VMs that run on different physical hosts with different Layer 2 networks communicate through DVS and the virtual routers over the VXLAN.

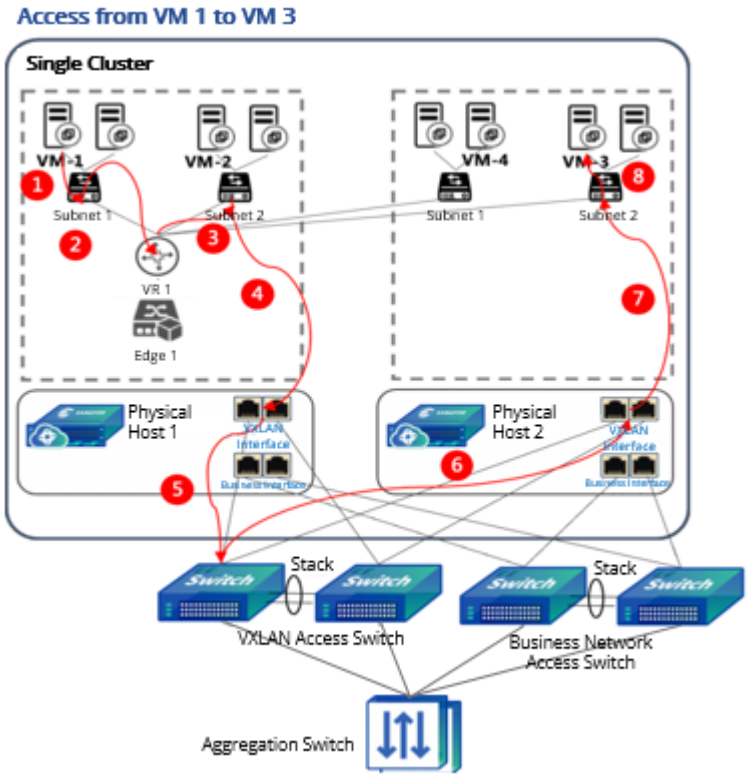


Figure 337: Traffic Forwarding Path for Cross-Host Communication Across Layer 2 Networks

Scenario 9: External communication for VMs when the edge is not on the local host

Traffic forwarding path: When VM 3, a business VM on Host 2 that has no edge, communicates with the external PC 1, the traffic first goes through DVS over the VXLAN network to reach Host 1 where the edge resides. Then, the traffic passes through DVS on Host 1 to reach Edge 1 and finally reaches PC 1 through the cluster business network.

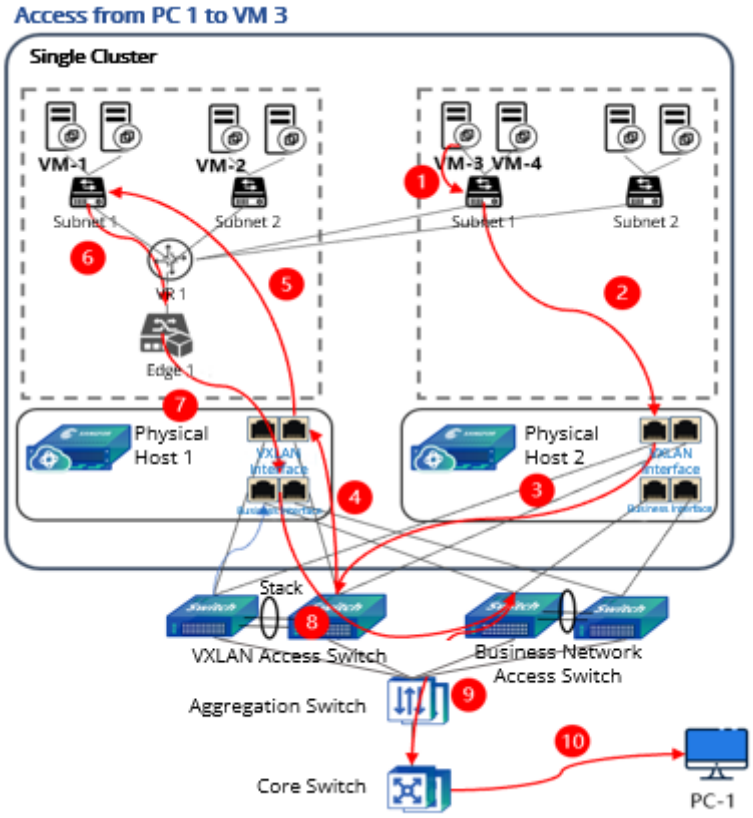


Figure 338: Traffic Forwarding Path for External Communication of VMs

Scenario 10: External communication for VMs when an NFV device (vNGAF) is deployed and the edge is not on the local host

Traffic forwarding path: Similar to Scenario 9, the traffic must reach Host 1 where the edge resides and pass through the NFV device (vNGAF) to reach Edge 1.

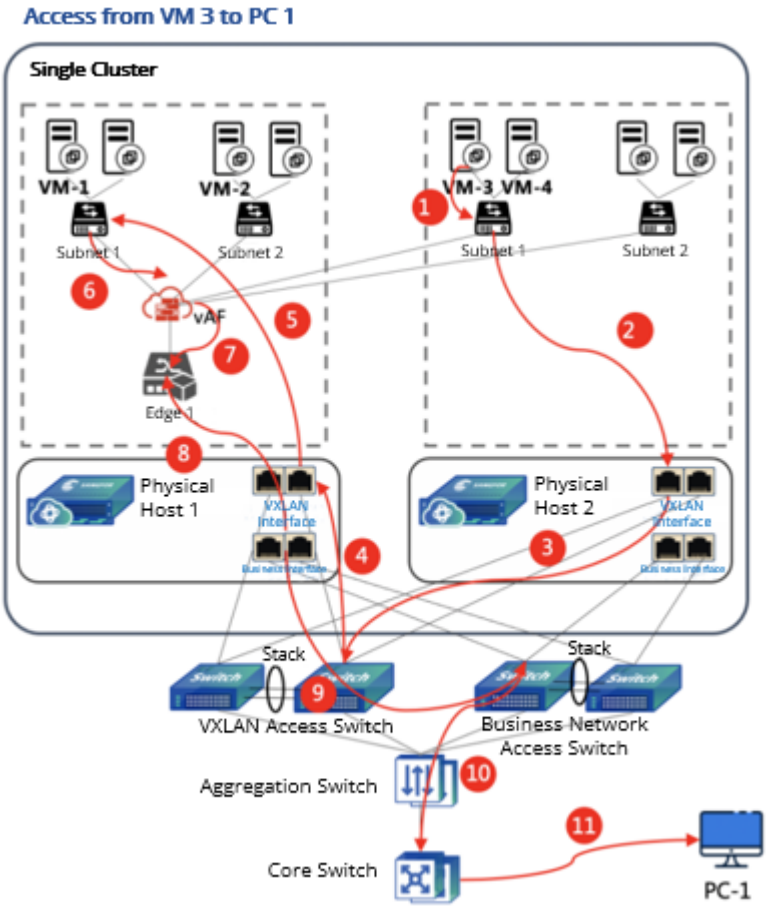


Figure 339: Traffic Forwarding Path for External Communication of VMs with NFV Device (vNGAF) Deployed

Scenario 11: Intra-physical host network communication across the subnets of a tenant VPC

Traffic forwarding path: In multi-tenant VPC scenarios, two business VMs that run on the same physical host with different Layer 2 networks communicate through DVS and virtual routers.

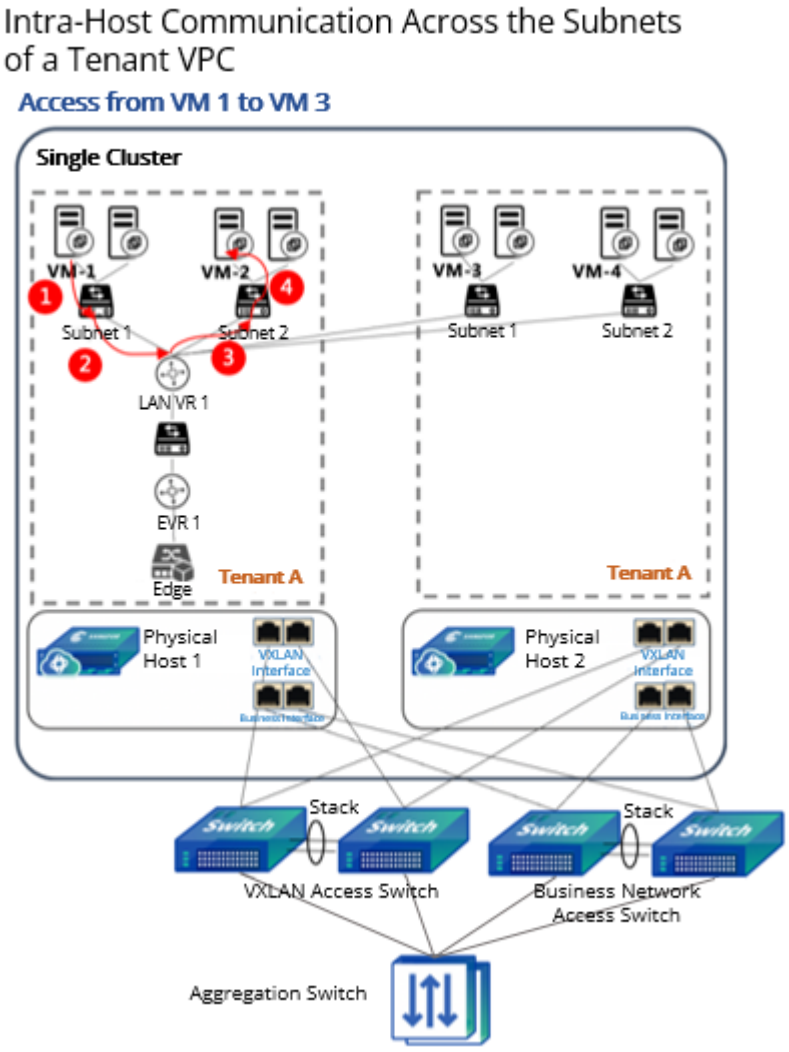


Figure 340: Traffic Forwarding Path for Intra-Host Communication Across Subnets of a Tenant VPC

Scenario 12: Cross-physical host network communication across the subnets of a tenant VPC

Traffic forwarding path: In multi-tenant VPC scenarios, two business VMs that run on different physical hosts with different Layer 2 networks communicate through DVS and virtual routers.

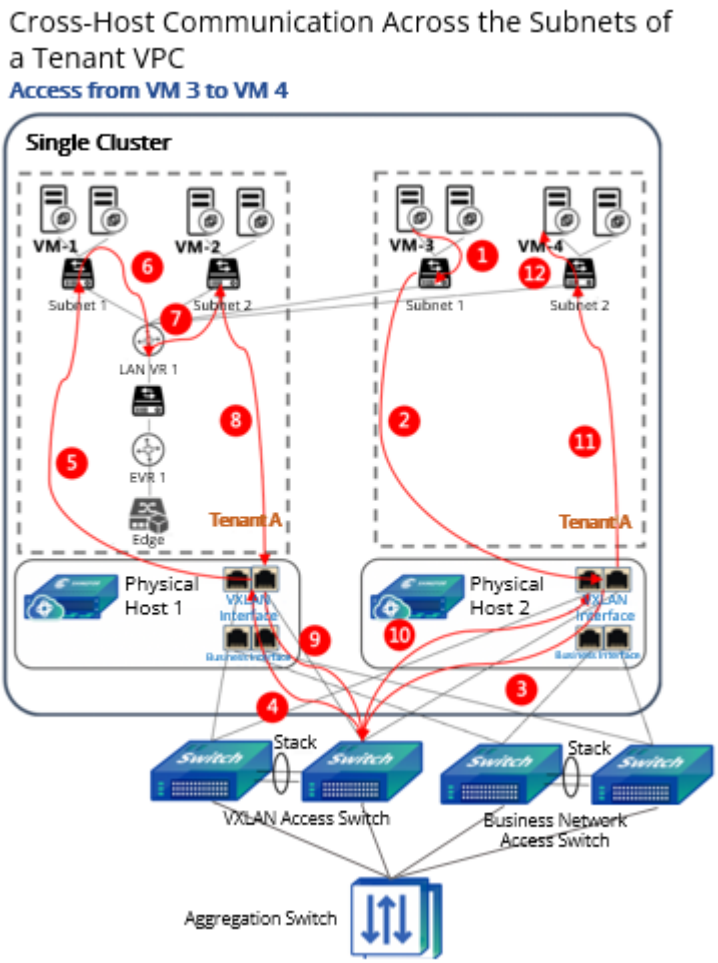


Figure 341: Traffic Forwarding Path for Cross-Host Communication Across Subnets of a Tenant VPC

Scenario 13: External communication for VMs within a tenant VPC when the edge is on the local host

Traffic forwarding path: In multi-tenant VPC scenarios, when a business VM on a host that has an edge communicates with the external PC 1, the traffic goes through DVS, the virtual router, and the edge before it reaches PC 1 through the cluster business network.

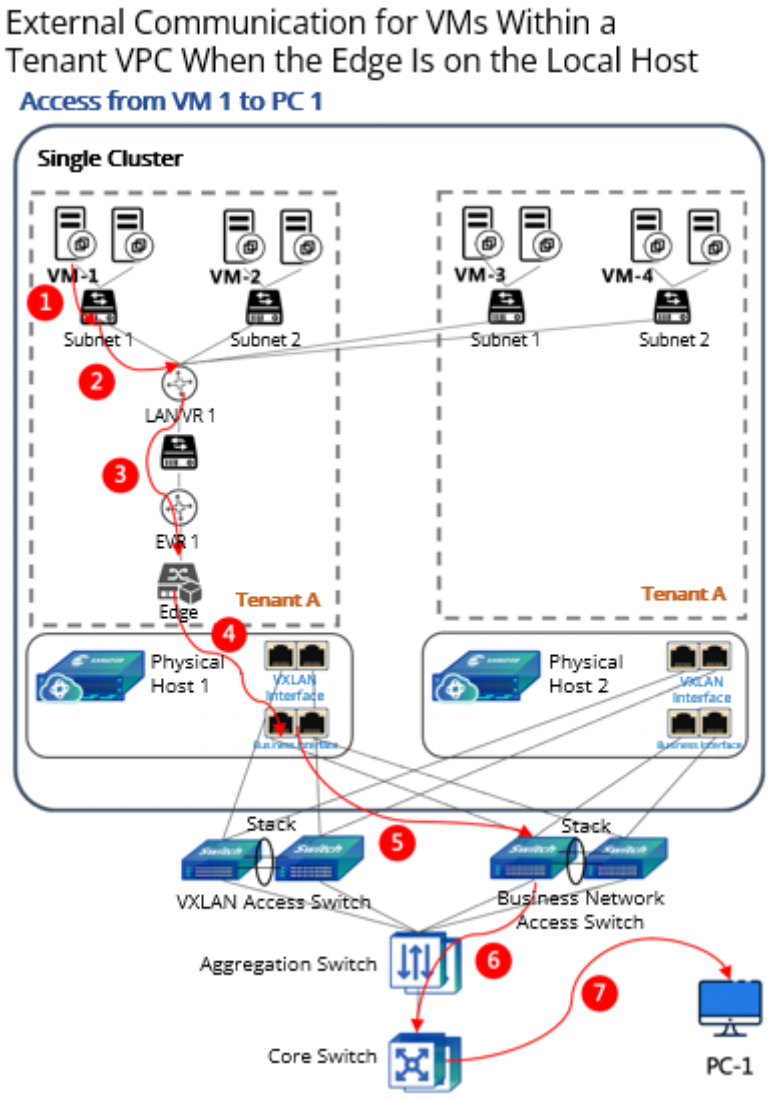


Figure 342: Traffic Forwarding Path for External Communication of VMs Within a Tenant VPC with Edge on Local Host

Scenario 14: External communication for VMs within a tenant VPC when the edge is not on the local host

Traffic forwarding path: In multi-tenant VPC scenarios, when VM 3, a business VM on a host that has no edge, communicates with the external PC 1, the traffic first goes through DVS of VM 3 over the VXLAN to reach Host 1 where the edge resides, then passes through DVS on Host 1 to reach the edge, and finally reaches PC 1 through the cluster business network.

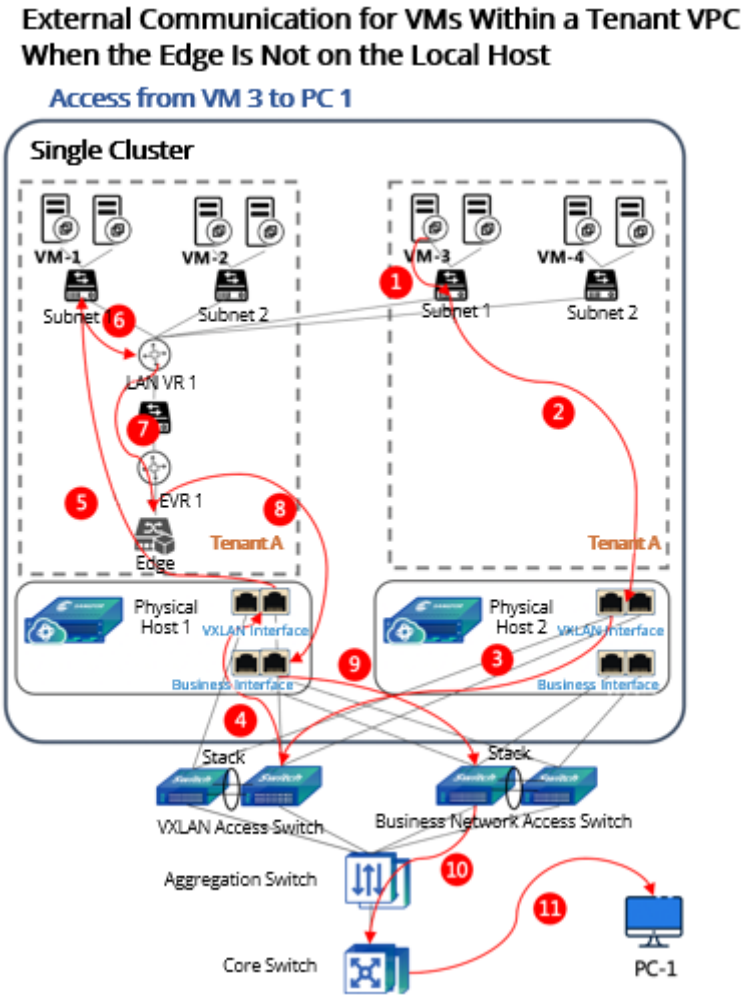


Figure 343: Traffic Forwarding Path for External Communication of VMs Within a Tenant VPC with Edge Not on Local Host

5.3.6.2.2 Single-Data Center, Multi-Cluster Unicast Traffic Forwarding

There are 14 common network topologies for multi-cluster deployment, and their traffic forwarding paths are as follows:

Scenario 1: Intra-cluster network communication within a Layer 2 network when DVS is deployed

Traffic forwarding path: Two business VMs running on the same physical host communicate through DVS.

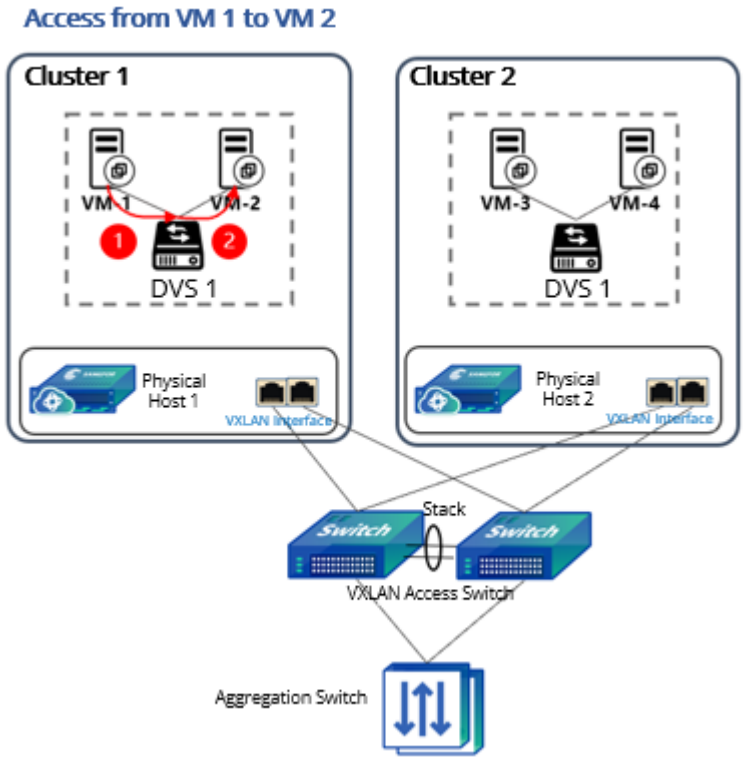


Figure 344: Traffic Forwarding Path for Intra-Cluster Communication Within a Layer 2 Network

Scenario 2: Cross-cluster network communication within a Layer 2 network when DVS is deployed

Traffic forwarding path: Two business VMs running on different physical hosts communicate through DVS over the VXLAN.

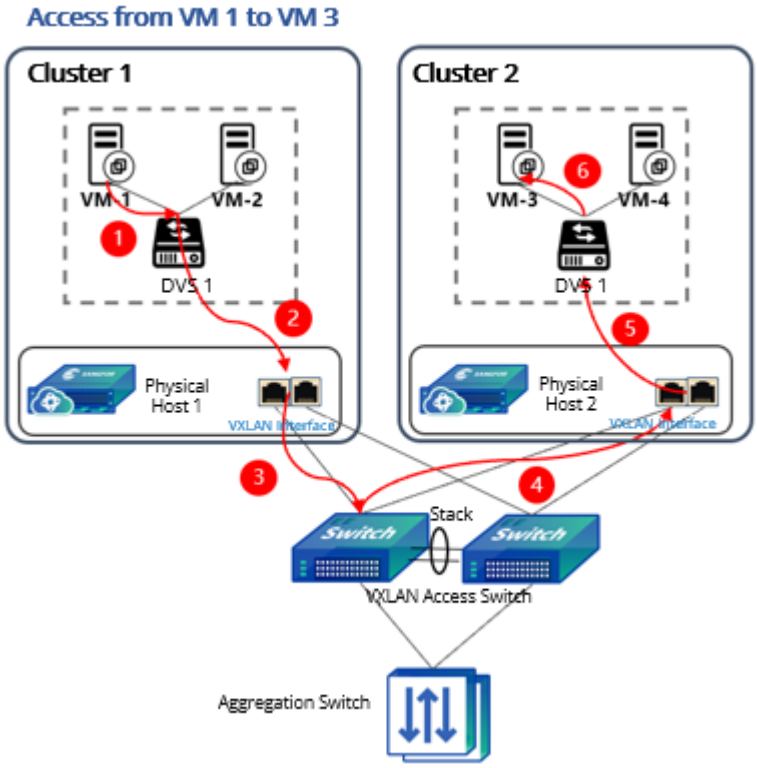


Figure 345: Traffic Forwarding Path for Cross-Cluster Communication Within a Layer 2 Network

Scenario 3: Intra-cluster network communication within a Layer 2 network when edges are connected in series and DVS is deployed

Traffic forwarding path: Two business VMs running on the same physical host communicate through DVS. In this scenario, the edges are only used for external network communication.

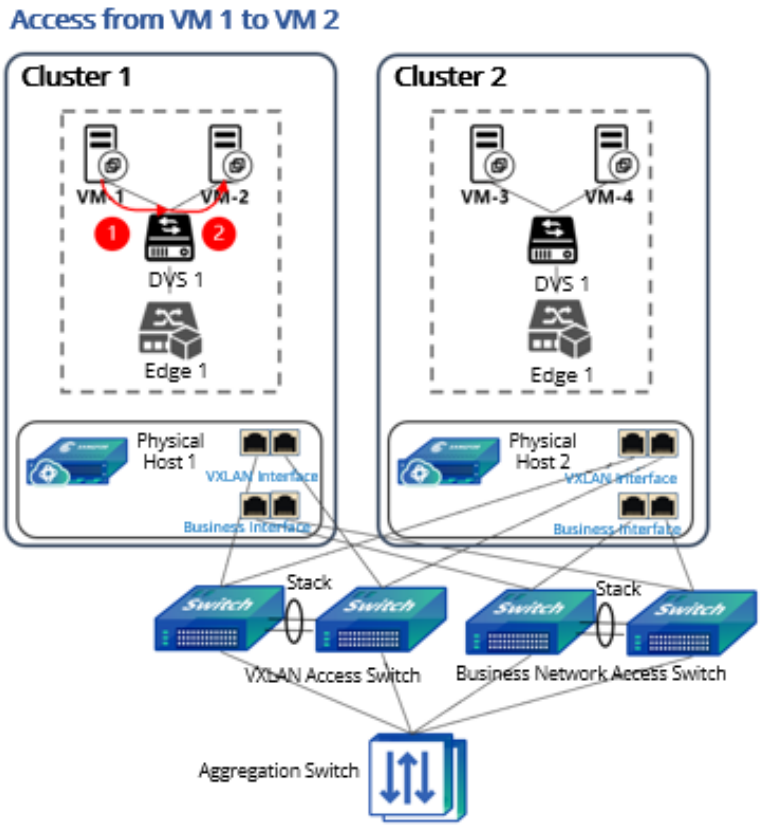


Figure 346: Traffic Forwarding Path for Intra-Cluster Communication Within a Layer 2 Network with Edges Connected in Series and DVS Deployed

Scenario 4: Cross-cluster network communication within a Layer 2 network when edges are connected in series and DVS is deployed

Traffic forwarding path: Two business VMs running on different physical hosts communicate through DVS and edges over the business network.

The reason why the traffic is not forwarded through the VXLAN interface: DVS is connected to the edge. To avoid loops that may occur when both the VXLAN and business network can forward traffic, the system is designed to convert a virtual switch connected to an edge into a local virtual switch that is not in distributed mode. This prevents loops because only the business network forwards east-west traffic.

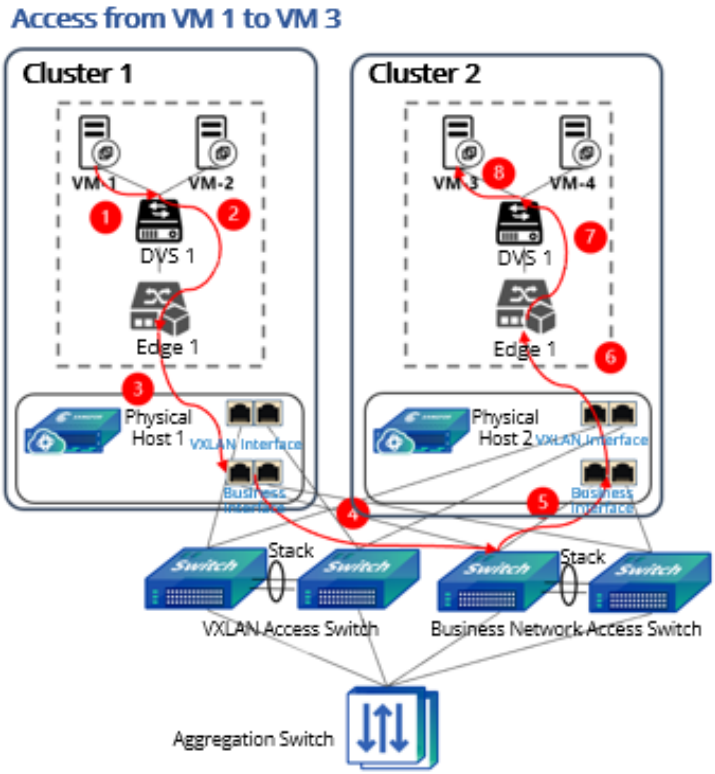


Figure 347: Traffic Forwarding Path for Cross-Cluster Communication Within a Layer 2 Network with Edges Connected in Series and DVS Deployed

Scenario 5: Intra-cluster network communication when VMs are directly connected to edges

Traffic forwarding path: Two business VMs running on the same physical host communicate through the directly connected edge.

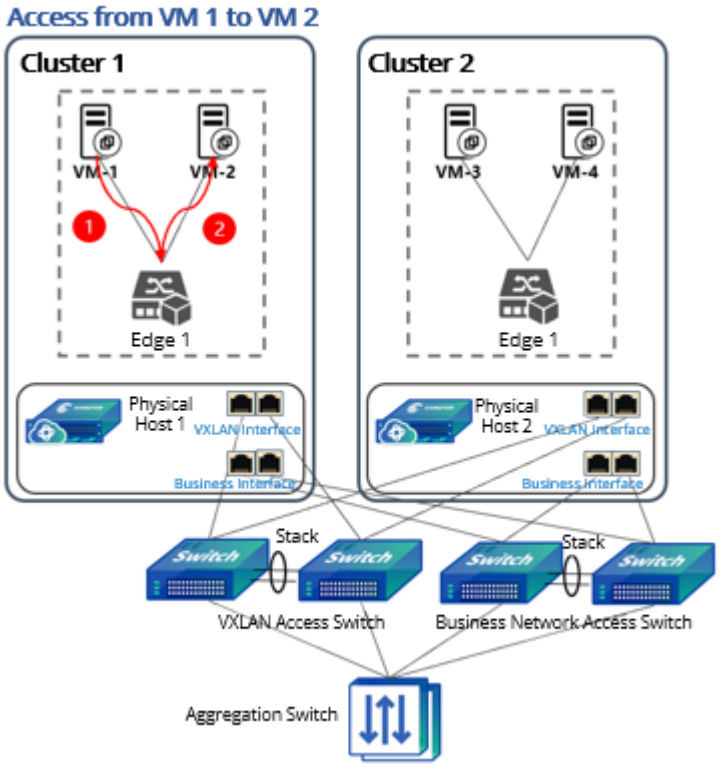


Figure 348: Traffic Forwarding Path for Intra-Cluster Communication with VMs Directly Connected to Edges

Scenario 6: Cross-cluster network communication when VMs are directly connected to edges

Traffic forwarding path: Two business VMs running on different physical hosts communicate through the directly connected edges over the business network.

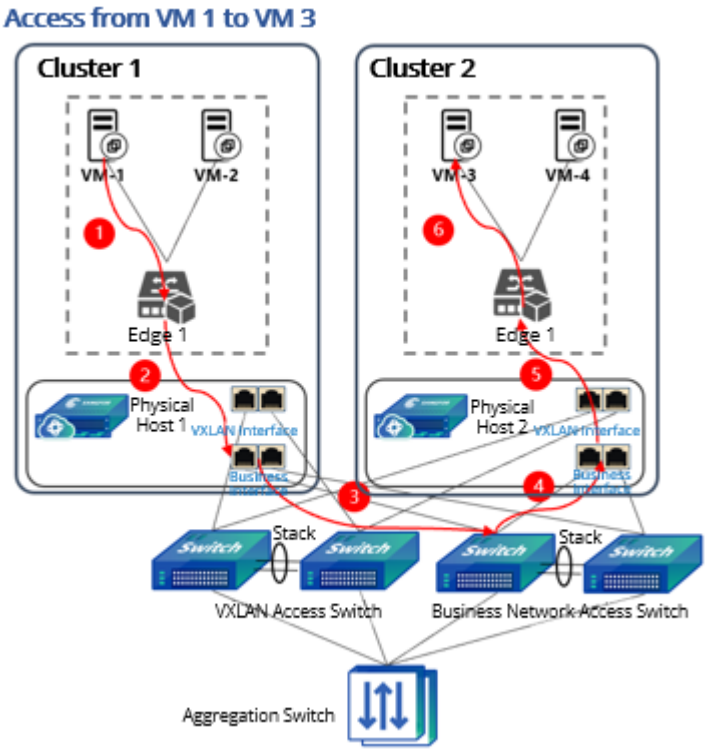


Figure 349: Traffic Forwarding Path for Cross-Cluster Communication with VMs Directly Connected to Edges

Scenario 7: Intra-cluster network communication across Layer 2 networks when DVS is deployed

Traffic forwarding path: Two business VMs that run on the same physical host with different Layer 2 networks communicate through DVS and the virtual router.

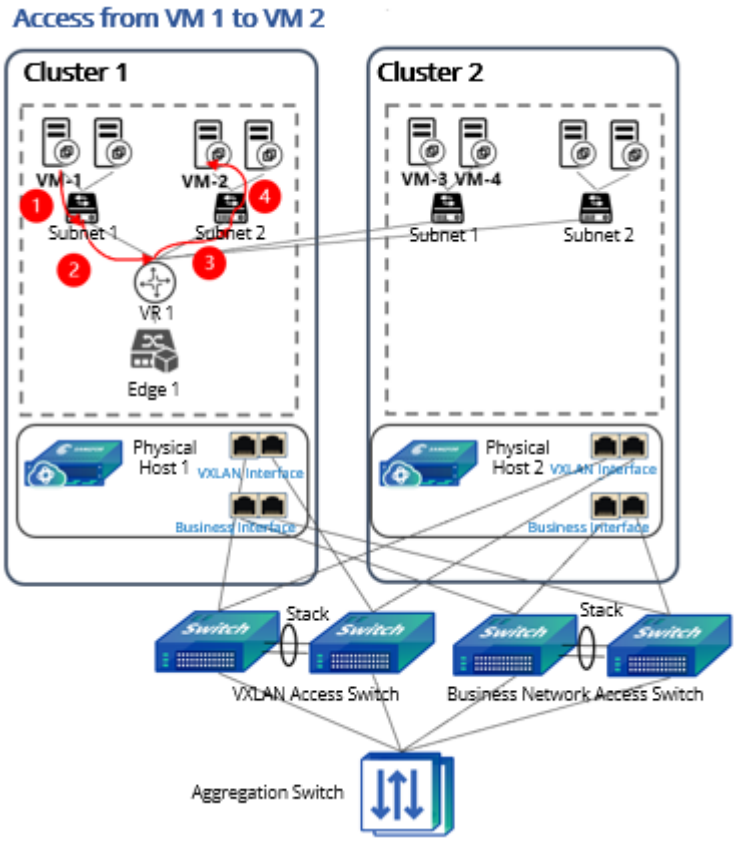


Figure 350: Traffic Forwarding Path for Intra-Cluster Communication Across Layer 2 Networks

Scenario 8: Cross-cluster network communication across Layer 2 networks when DVS is deployed

Traffic forwarding path: Two business VMs that run on different physical hosts with different Layer 2 networks communicate through DVS and the virtual routers over the VXLAN.

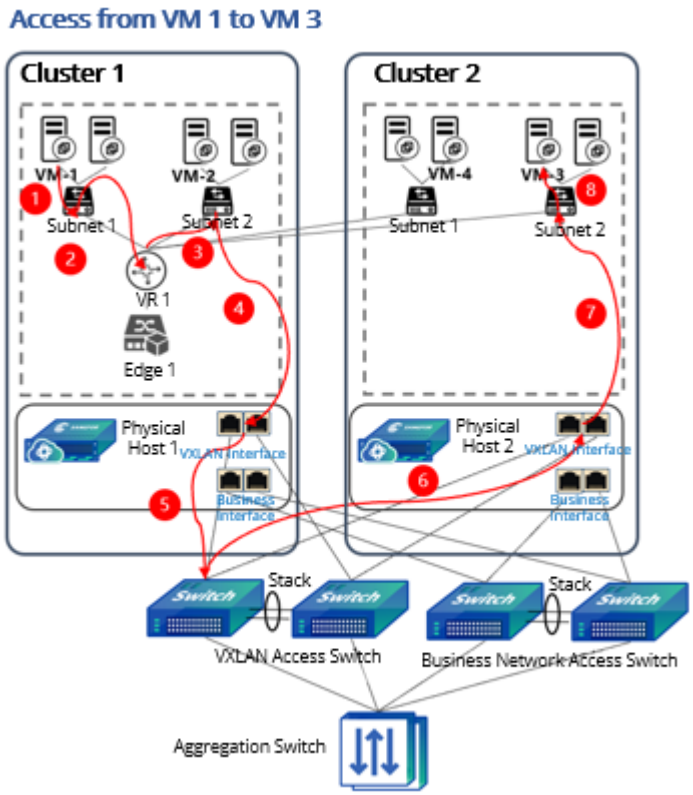


Figure 351: Traffic Forwarding Path for Cross-Cluster Communication Across Layer 2 Networks

Scenario 9: External communication for VMs when the edge is not in the local cluster

Traffic forwarding path: When VM 3, a business VM on Host 2 that has no edge, communicates with the external PC 1, the traffic first goes through DVS over the VXLAN to reach Host 1 where the edge resides. Then, the traffic passes through DVS on Host 1 to reach Edge 1 and finally reaches PC 1 through the cluster business network.

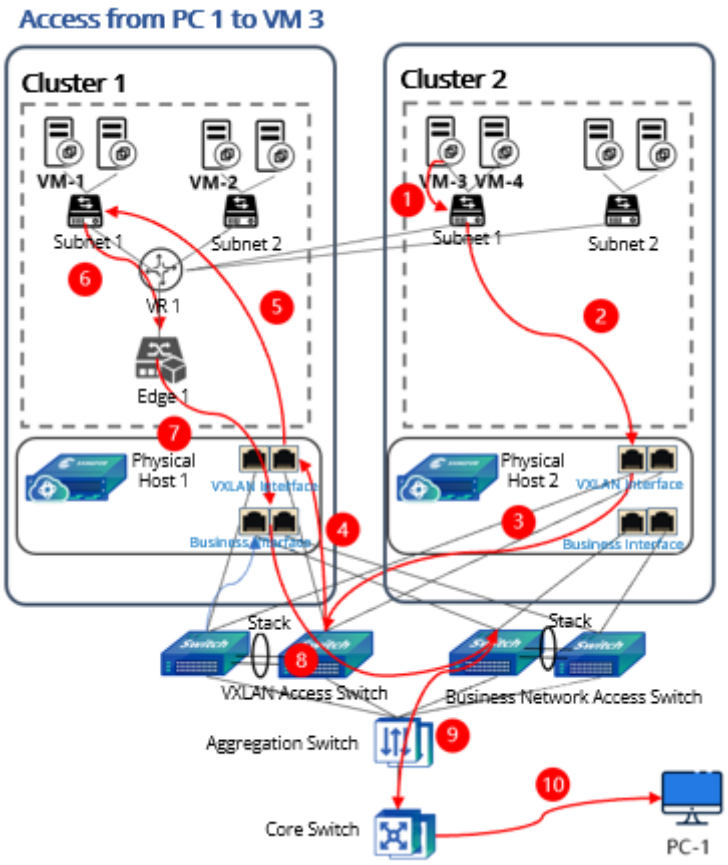


Figure 352: Traffic Forwarding Path for External Communication of VMs

Scenario 10: External communication for VMs when an NFV device (vNGAF) is deployed and the edge is not in the local cluster

Traffic forwarding path: Similar to Scenario 9, the traffic must reach Host 1 where the edge resides and pass through the NFV device (vNGAF) to reach Edge 1.

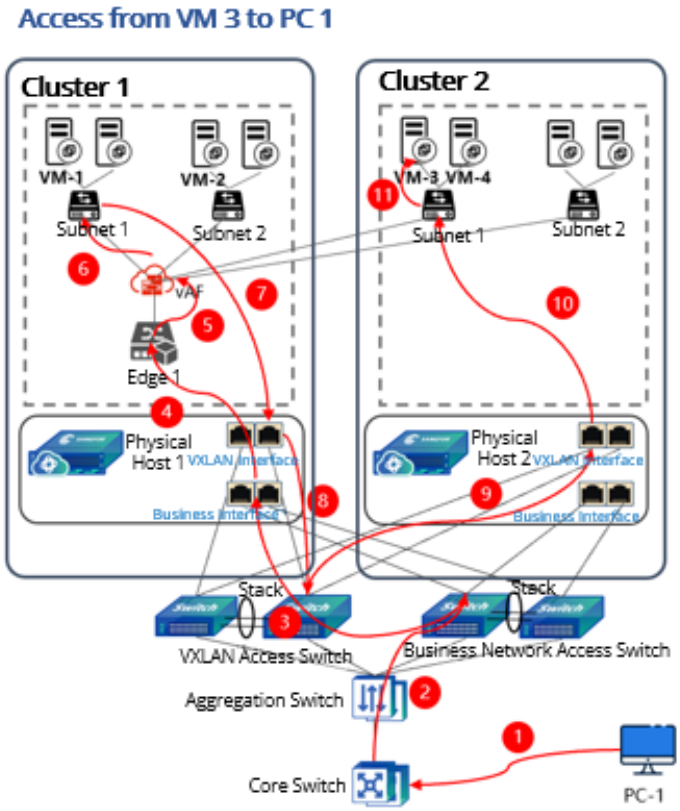


Figure 353: Traffic Forwarding Path for External Communication of VMs with NFV Device (vNGAF) Deployed

Scenario 11: Intra-cluster network communication across the subnets of a tenant VPC

Traffic forwarding path: In multi-tenant VPC scenarios, two business VMs that run on the same physical host with different Layer 2 networks communicate through DVS and virtual routers.

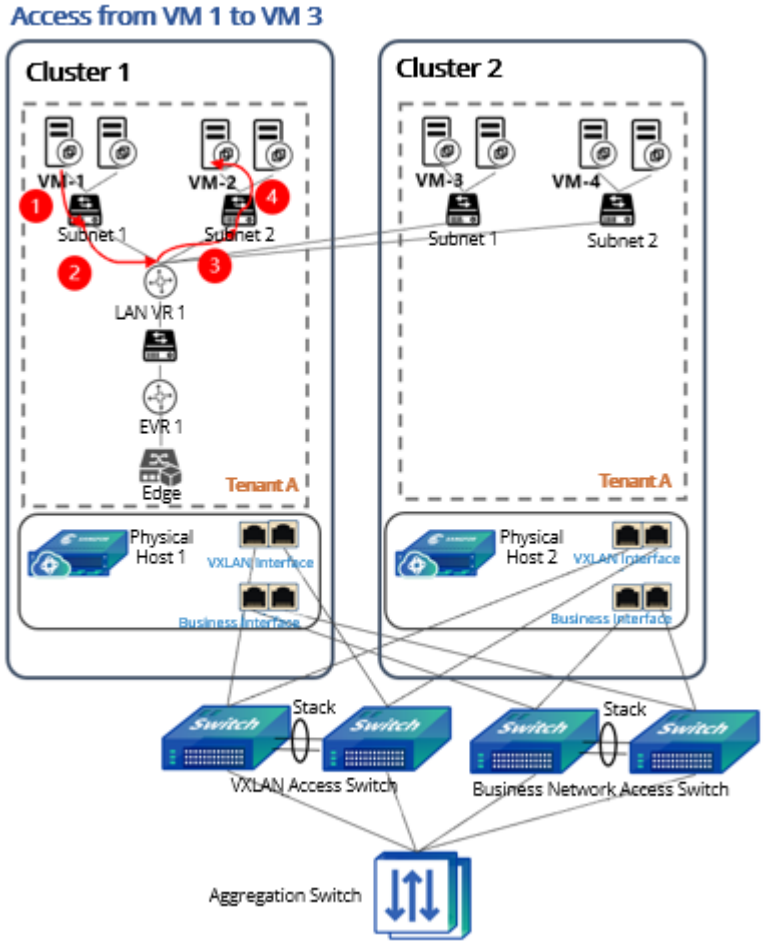


Figure 354: Traffic Forwarding Path for Intra-Cluster Communication Across Subnets of a Tenant VPC

Scenario 12: Cross-cluster network communication across the subnets of a tenant VPC

Traffic forwarding path: In multi-tenant VPC scenarios, two business VMs that run on different physical hosts with different Layer 2 networks communicate through DVS and virtual routers.

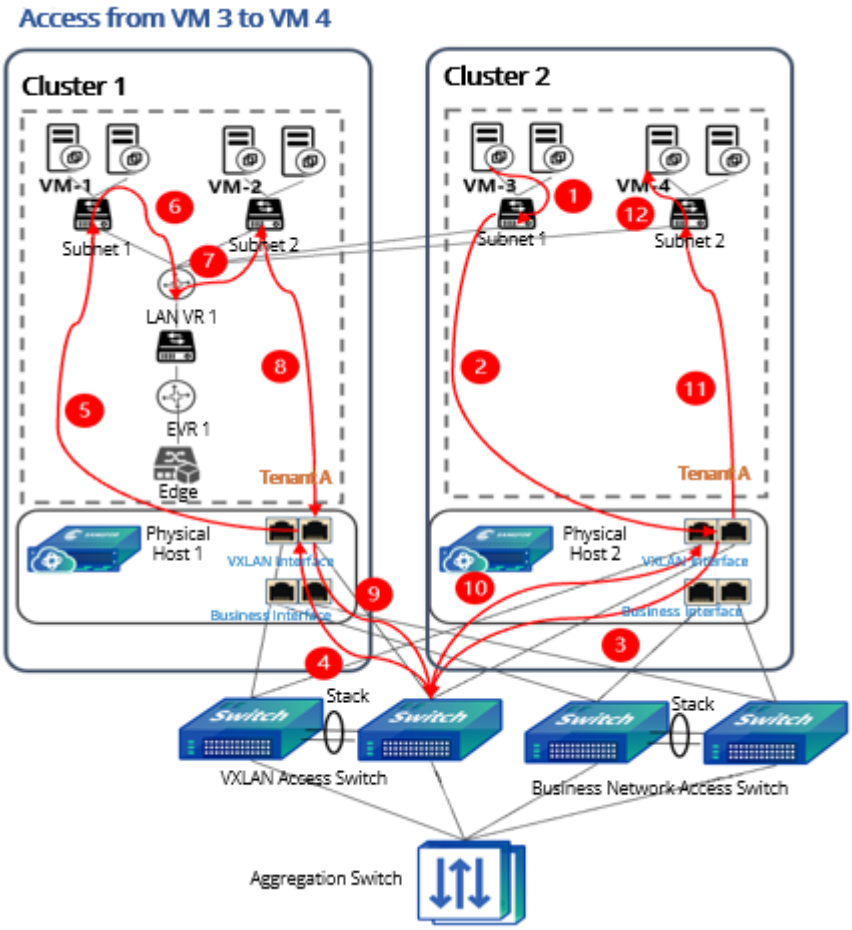


Figure 355: Traffic Forwarding Path for Cross-Cluster Communication Across Subnets of a Tenant VPC

Scenario 13: External communication for VMs within a tenant VPC when the edge is in the local cluster

Traffic forwarding path: In multi-tenant VPC scenarios, when a business VM on a host that has an edge communicates with the external PC 1, the traffic goes through DVS, the virtual router, and the edge before it reaches PC 1 through the cluster business network.

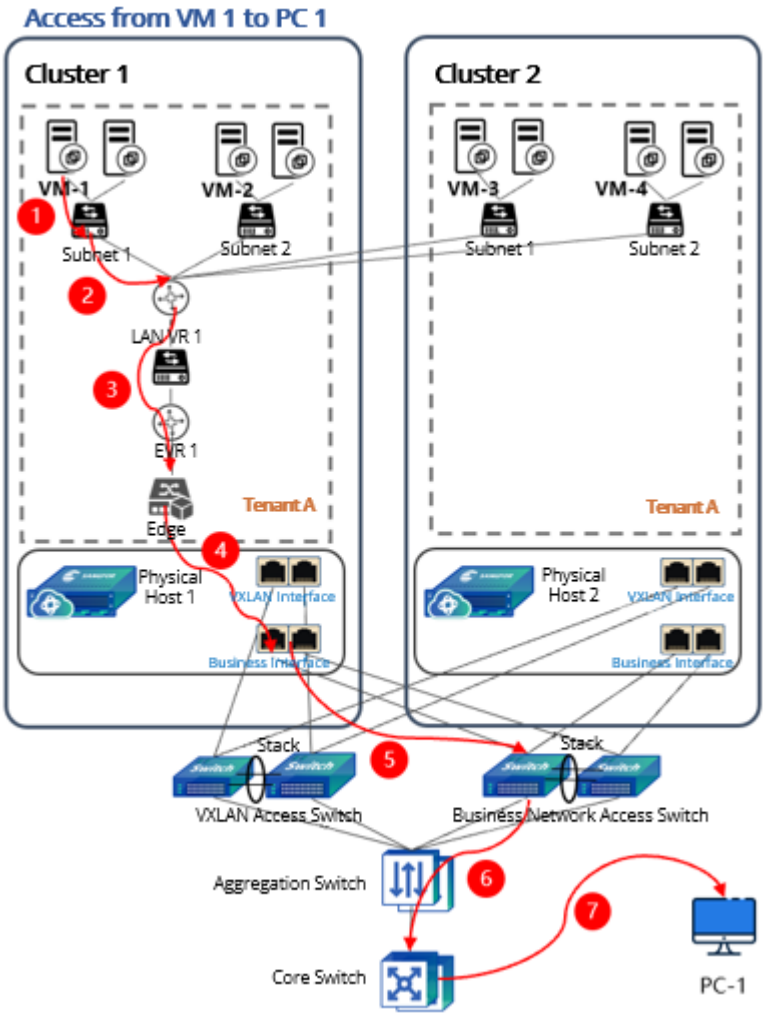


Figure 356: Traffic Forwarding Path for External Communication of VMs Within a Tenant VPC with Edge in Local Cluster

Scenario 14: External communication for VMs within a tenant VPC when the edge is not in the local cluster

Traffic forwarding path: In multi-tenant VPC scenarios, when VM 3, a business VM on a host that has no edge, communicates with the external PC 1, the traffic first goes through DVS of VM 3 over the VXLAN to reach Host 1 where the edge resides, then passes through DVS on Host 1 to reach the edge, and finally reaches PC 1 through the cluster business network.

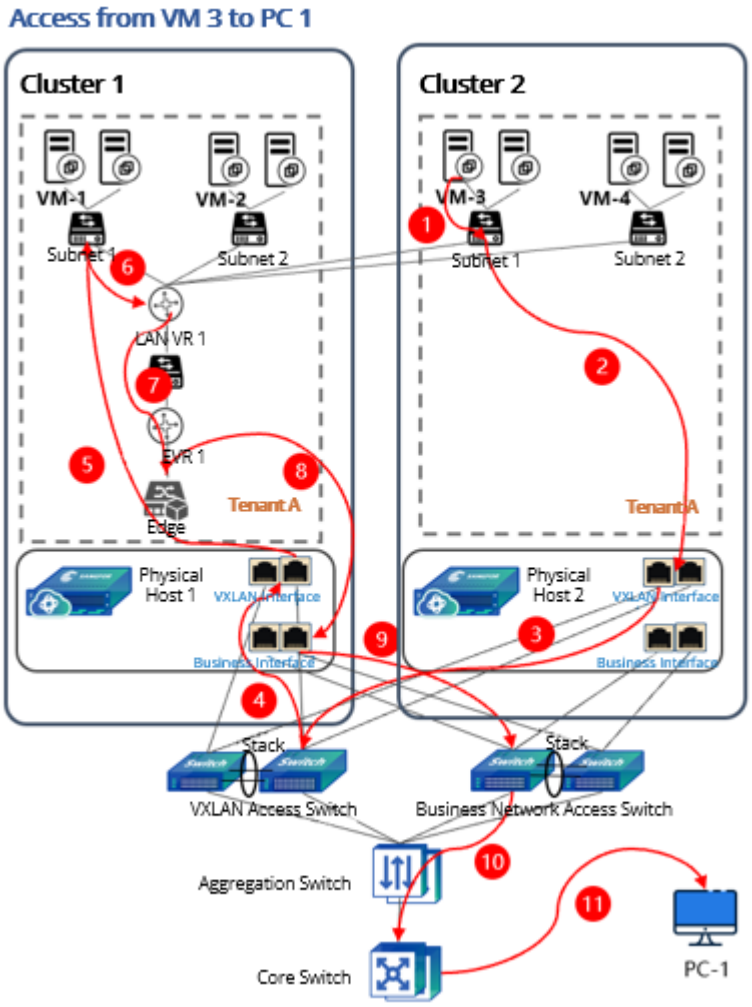


Figure 357: Traffic Forwarding Path for External Communication of VMs Within a Tenant VPC with Edge Not in Local Cluster

5.3.6.2.3 Cross-Site, Multi-Cluster Unicast Traffic Forwarding

There are six common network topologies for cross-site, multi-cluster deployment, and their traffic forwarding paths are as follows:

Scenario 1: Cross-site Layer 2 communication

Traffic forwarding path: Traffic from VM 1 in Site 1 is sent over the VXLAN of Subnet 1 to the VXLAN of Subnet 1 in Site 2 before it is forwarded to VM 5.

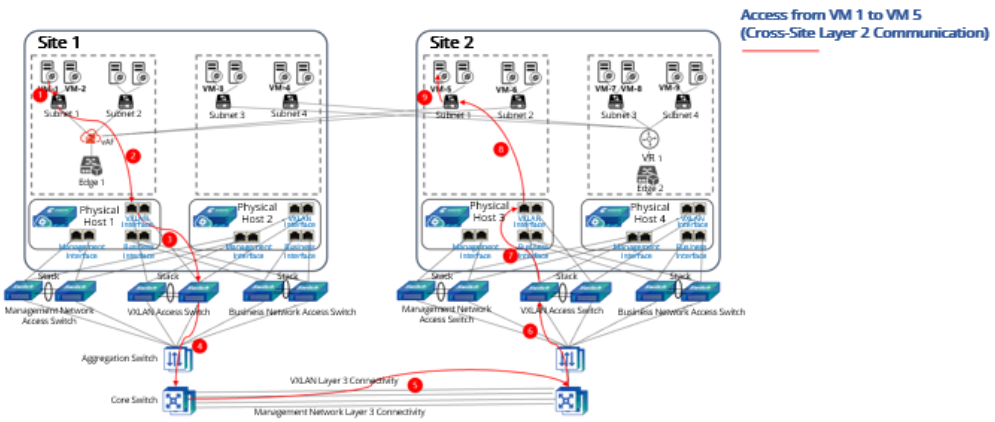


Figure 358: Traffic Forwarding Path for Cross-Site Layer 2 Communication

Scenario 2: Cross-site external communication

Traffic forwarding path: Traffic from VM 3 in Site 1 first goes through the VXLAN of Subnet 3 to reach the VXLAN of Subnet 3 in Site 2, then travels through Virtual Router 1 for Subnet 3, Edge 2, and the business network interface, and finally passes through the business access switch to reach the core switch, where the traffic is forwarded to PC 1.

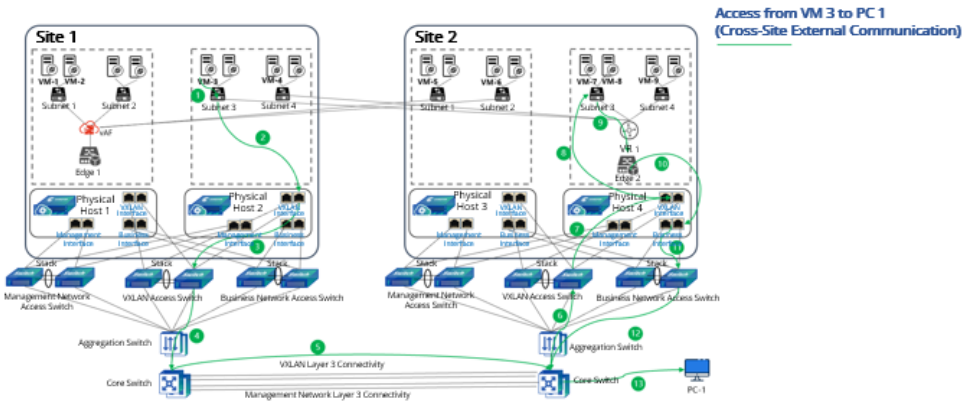


Figure 359: Traffic Forwarding Path for Cross-Site External Communication

Scenario 3: Cross-site Layer 3 communication

Traffic forwarding path: Firstly, traffic from VM 8 in Site 2 reaches the virtual router through the switch for Subnet 3; then, the virtual router finds the switch for Subnet 4 and forwards the traffic to Subnet 4 in Site 1 through the VXLAN; finally, the traffic reaches VM 4.

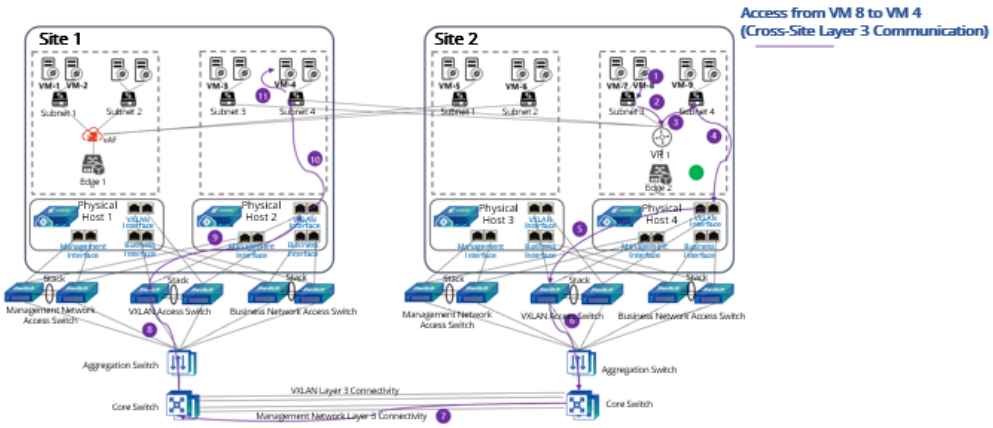


Figure 360: Traffic Forwarding Path for Cross-Site Layer 3 Communication

Scenario 4: Cross-site Layer 2 communication within a tenant VPC

Traffic forwarding path: Traffic from VM 1 of Tenant A in Site 1 is sent over the VXLAN of Subnet 1 to Subnet 1 of Tenant A in Site 2 before it reaches VM 5.

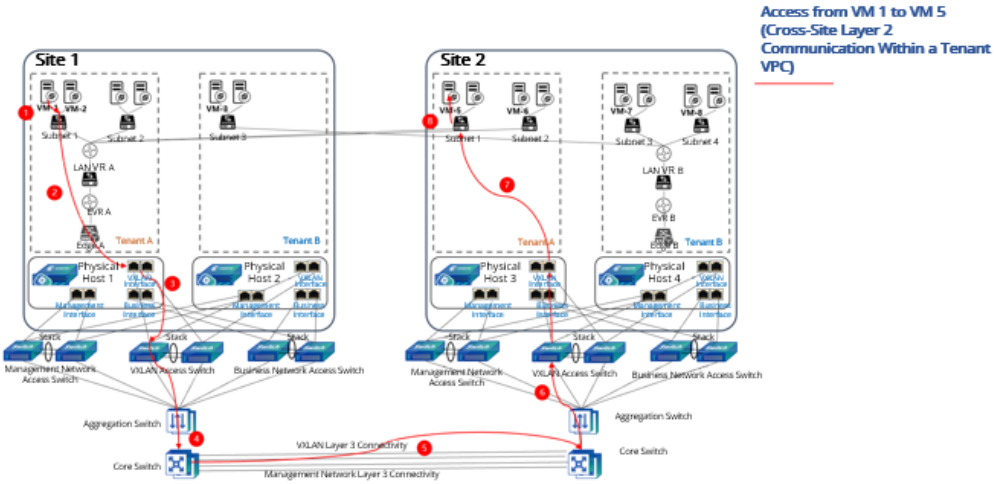


Figure 361: Traffic Forwarding Path for Cross-Site Layer 2 Communication Within a Tenant VPC

Scenario 5: Cross-site Layer 3 communication within a tenant VPC

Traffic forwarding path: Firstly, traffic from VM 8 in Site 2 reaches the virtual router through the switch for Subnet 4; then, the virtual router finds the switch for Subnet 3 and forwards the traffic to Subnet 3 in Site 1 through the VXLAN; finally, the traffic reaches VM 3.

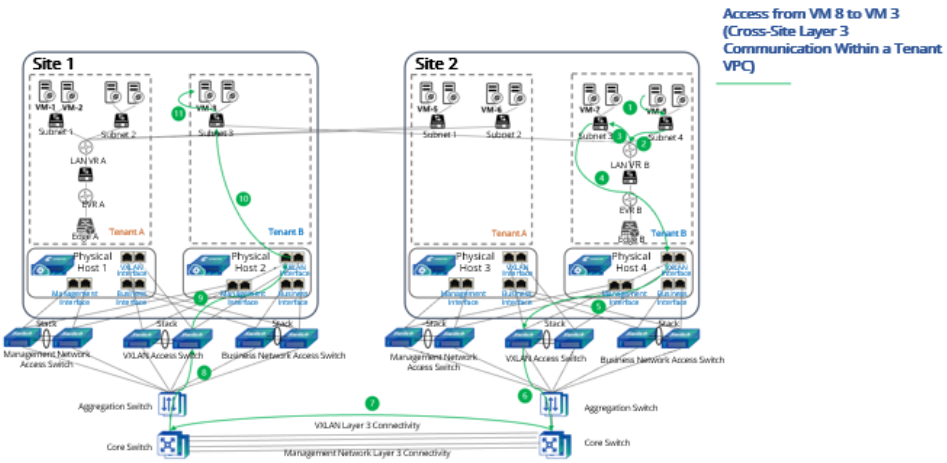


Figure 362: Traffic Forwarding Path for Cross-Site Layer 3 Communication Within a Tenant VPC

Scenario 6: External access from a tenant VPC

Traffic forwarding path: Traffic from VM 6 in Site 2 first goes through the VXLAN of Subnet 2 to reach the VXLAN of Subnet 2 in Site 1, then travels through LAN Virtual Router A for Subnet 2, Edge Router A, Edge A, and the business network interface, and finally passes through the business access switch to reach the core switch, where the traffic is forwarded to PC 1.

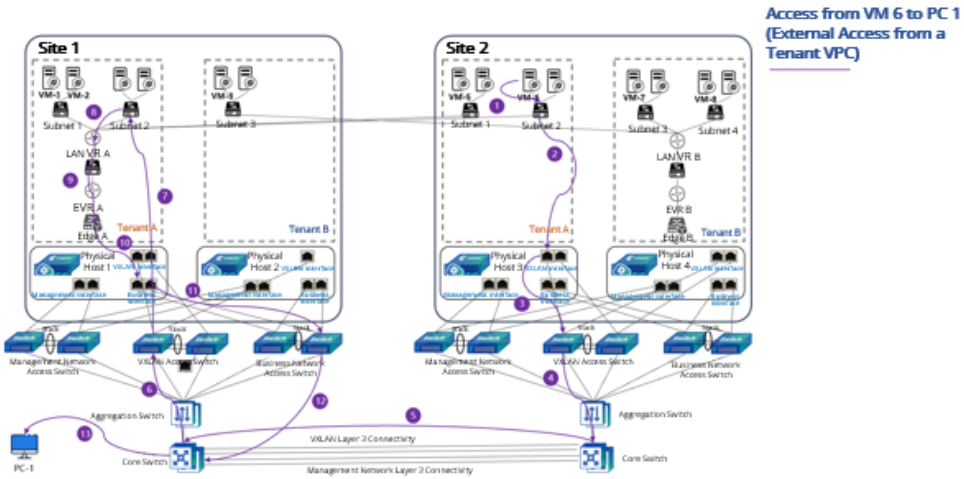


Figure 363: Traffic Forwarding Path for External Access From a Tenant VPC

5.3.7 Network Visualization

Flow visualization enables the collection and analysis of business flows for large-scale and high-complexity businesses. It visualizes bottleneck diagnosis,

fault detection, capacity planning, and service status monitoring in business systems for comprehensive service governance.

1. It visualizes access traffic flows, enabling optimization in business topology.
2. It verifies access traffic based on firewall rules, swiftly isolates abnormal traffic, and helps enhance access policies.
3. It identifies the impact scope of a cyber attack and ensures quick and precise fault domain isolation.
4. It allows querying historical access details, enabling customers to analyze access between specific businesses, facilitating troubleshooting, and improving O&M efficiency.
5. It supports exporting historical access details, allowing customers to analyze business access on internal analysis platforms.
6. It ensures a 100% sampling rate for business access traffic without compromising network forwarding performance, offering complete access traffic flows.

The logical architecture for network traffic access visualization is as follows.

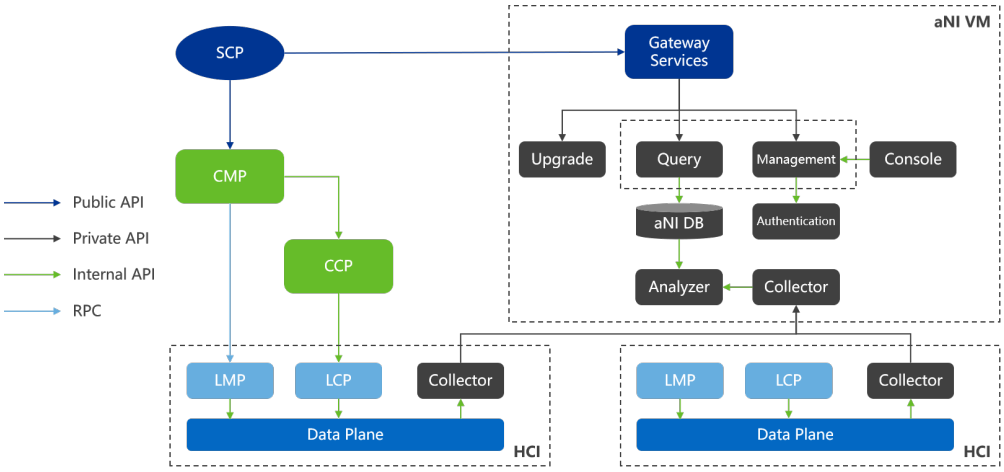


Figure 364: Traffic Access Visualization Logic

Network traffic access visualization is a feature based on Sangfor's extensive security expertise. It accurately analyzes and depicts access traffic flows between business systems.

The collector for network traffic access visualization imports the traffic to the

aNI collector via a bypass. The aNI analyzer analyzes the collected traffic and stores the results in the aNI database for administrators to query. Network traffic access visualization has the following characteristics:

1. The import of access traffic supports high concurrency and high throughput without compromising network forwarding performance because traffic is imported via a bypass.
2. aNI adopts a time-series database for storing analysis data, which minimizes the requirement on storage space and performance by supporting data aggregation and compression. The aNI database also supports querying data of 30 days, real-time data retrieval, and rapid aggregation analysis, with an interface response time of less than 3 seconds.

5.3.8 East-West Distributed Firewall

Attackers can easily target weakly protected servers when they manage to bypass the edge defenses using various methods, enabling them to attack other servers within the data center with ease. Manual SP installation on a large number of servers in the data center is too time-consuming to prevent extensive paralysis of the data center caused by the rapid spread of viruses. Therefore, HCI offers a distributed firewall (DFW) to extend security from the data center edge to its core. This enables micro-segmentation between VMs and ensures L3 and L4 security protection for internal traffic, significantly reducing the impact of attacks on the data center.

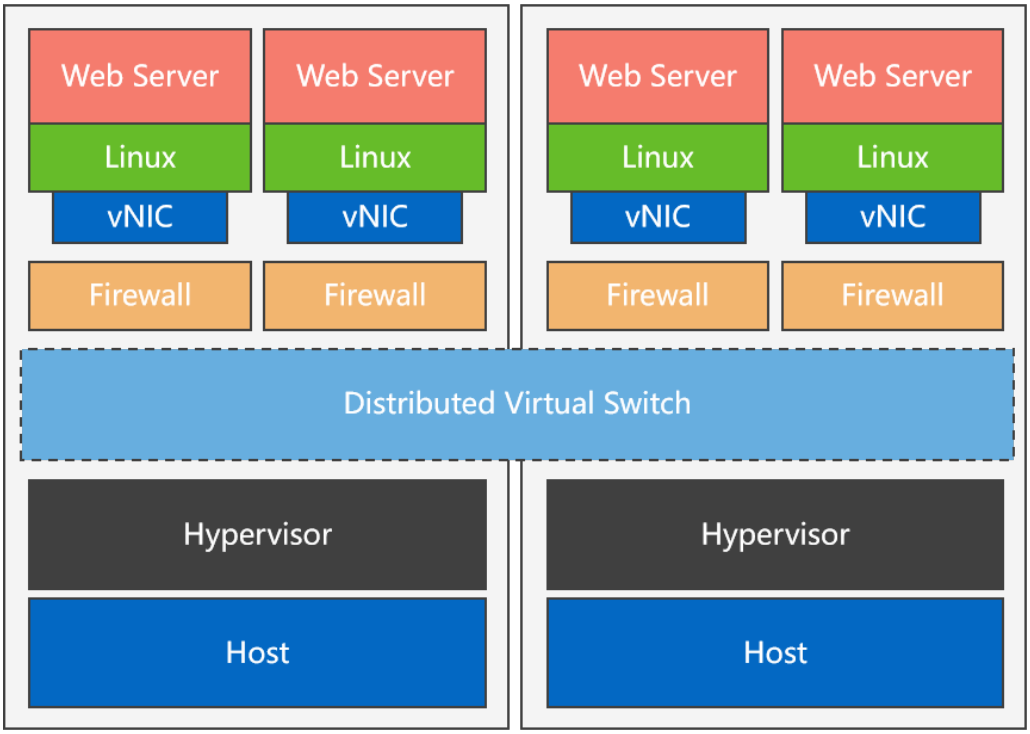


Figure 365: DFW Principle

5.3.8.1 Features

1. Comprehensive protection for all VMs: DFW can be configured on any distributed switch port to protect the inbound and outbound traffic in a region of HCI and ensure micro-segmentation. Corresponding policies are synchronized when a VM is migrated to another node.
2. Rapid enforcement of protection policies: DFW can be directly and swiftly deployed on HCI without installing any plugins due to the simple architecture. The bypass and interception log display features ensure quick verification of rule effectiveness.
3. Simplified network structure for easy troubleshooting: HCI DFW has a simple network structure featuring easy O&M through "What-You-Draw-Is-What-You-Get." The connectivity test enables quick troubleshooting.
4. Collaboration with Sangfor vNGAF for enhanced security: DFW is often used to control internal traffic in virtual networks because its distributed

architecture ensures platform-wide policy effectiveness, while vNGAF is usually deployed at the edge to defend against external traffic attacks.

5. Users can specify the VM, VM group, VM tag, IP address, IP address group, or IP address range for policies.

DFW is available once aNET is licensed. DFW, in combination with vNGAF, can provide all-around security protection for data centers, covering the north-south, east-west, external, and internal traffic.

5.3.8.2 Application Scenarios

In 2016, the Shadow Brokers disclosed multiple remote Windows vulnerability exploit tools, shocking the entire IT community. Windows servers were at severe risk of attacks if any of the following ports were open: 135, 137, 139, 445, and 3389. Bitcoin ransomware had already emerged by the time the vulnerability was discovered.

However, SP installation on all servers for a data center with hundreds of Windows servers is time-consuming. If a hacker penetrates the firewall and gains access to the data center during this process, any Windows server with the corresponding ports open becomes an easy target.

5.3.8.3 Applicable Scope

Sangfor DFW provides the applicable scope feature that supports selecting the VMs for which the distributed firewall rules apply (all VMs by default) as needed for flexible use in various scenarios.

5.3.8.4 Policies and Rules

Sangfor DFW offers east-west network isolation to protect all VMs within a data center using a single rule.

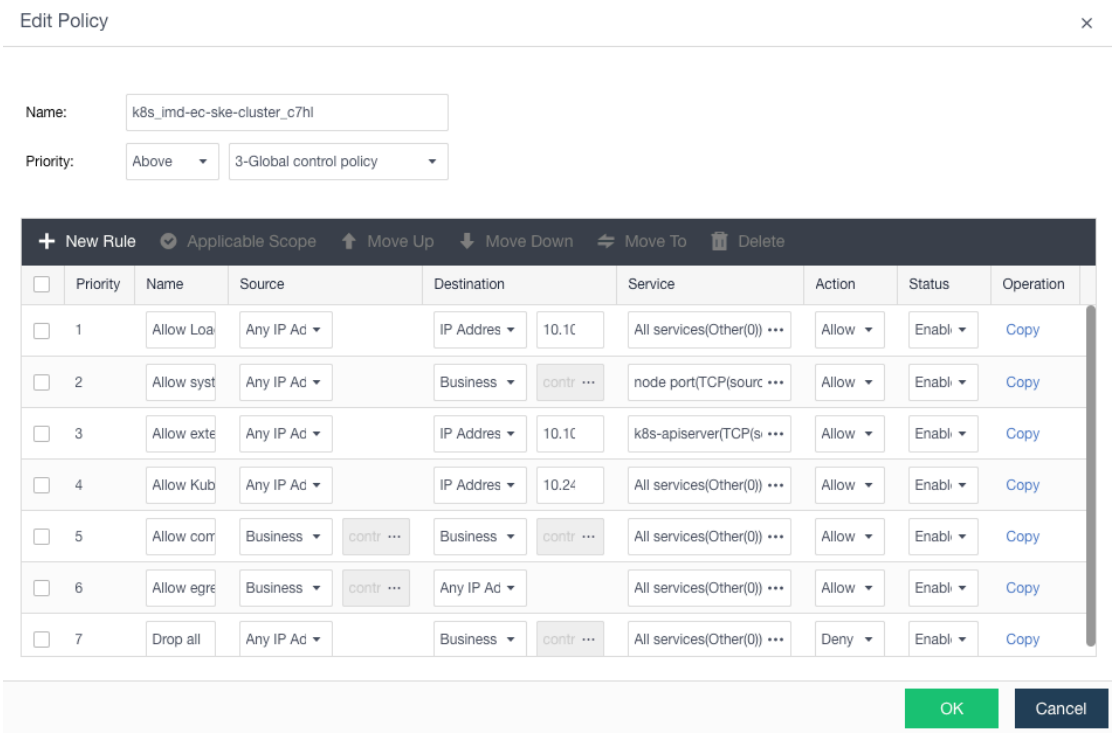


Figure 366: DFW Policy Configuration

- **Policy Name**: Deny inter-VM communication on ports 135, 445, and 3389 within the data center.
- **Source**: All VMs in the data center.
- **Destination**: All VMs in the data center.
- **Service**: Select ports 135, 137, 139, 445, and 3389 for both TCP and UDP.
- **Action**: Deny.

5.3.8.5 Dropped Packet Logging

Sangfor DFW provides the dropped packet logging feature that supports viewing the dropped packet logging information as needed to analyze the traffic information of business services.

5.3.8.6 Rule Anomaly Alert

Sangfor DFW enhances protection by monitoring the rules configured by users for better use. If a rule is abnormal, a real-time alert will be triggered. After

receiving the notification, the user can follow the recommendation to edit the rule to restore it to normal.

5.3.9 Intelligent DFW Policy Recommendation Based on aNI

5.3.9.1 Architecture

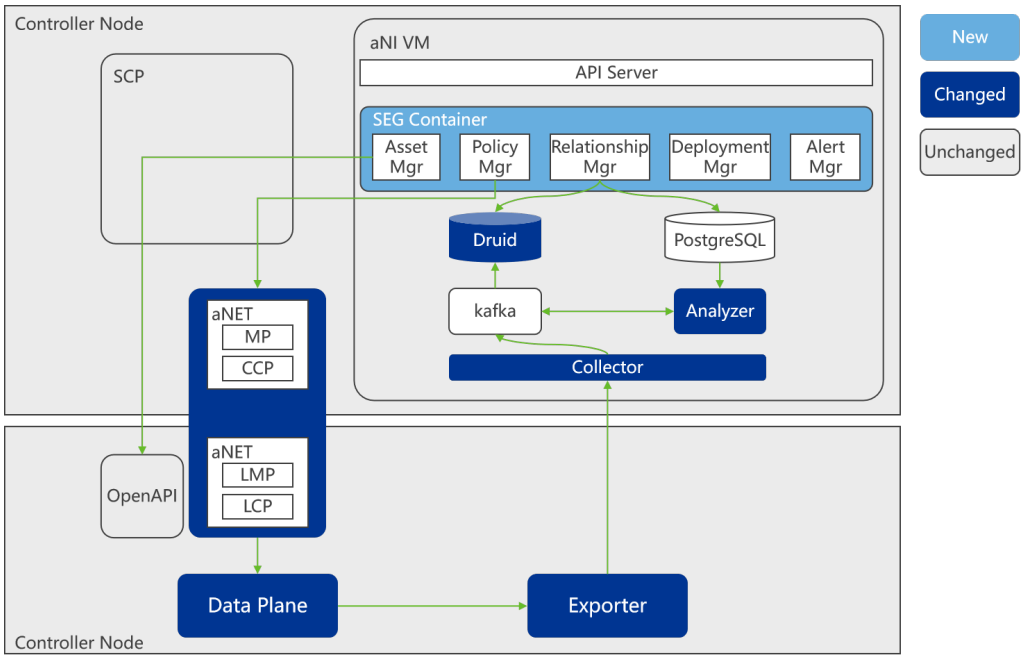


Figure 367: DFW Policy Architecture

5.3.9.2 Technical Principles

1. DFW rules intelligently generated based on Network Insight (aNI) access data

The aNI collector and analyzer collect the access relationships reported by the data plane and log them into the Druid database, from which the SEG container (micro-segmentation) filters the access relationships according to users' configurations (protected object, date, traffic direction, rule granularity, etc.) and generates DFW rules that can be previewed and modified by users.

2. DFW rule pre-release for effect verification

The aNET and data plane provide the DFW rule pre-release capability. After the generated rules are pre-released, they only perform matching and recording, without applying actions, intercepting traffic, or affecting business services. If a request hits a pre-released rule first, it will forward to match an effective rule directly and report the hit pre-released and effective rules to aNI; if a request hits an effective rule first, it will terminate the rule matching and report the hit effective rule to aNI.

3. **Mistakenly-Denied Access** alert

Traffic that may be blocked by the pre-released DFW rules is notified to users as **Mistakenly-Denied Access** and can be quickly unblocked. After observation and modification, users can enforce a pre-released rule and the hit traffic will be denied or allowed at the same time.

5.3.9.3 UI Display

Log in to HCI, go to **Networking** > **Network Insight**, select a business group, and click **Policy Generator** to automatically generate ACL rules between business VMs based on the access relationships of their business applications within 30 days. O&M personnel can quickly and easily sort out access relationships and configure access control policies through Network Insight.

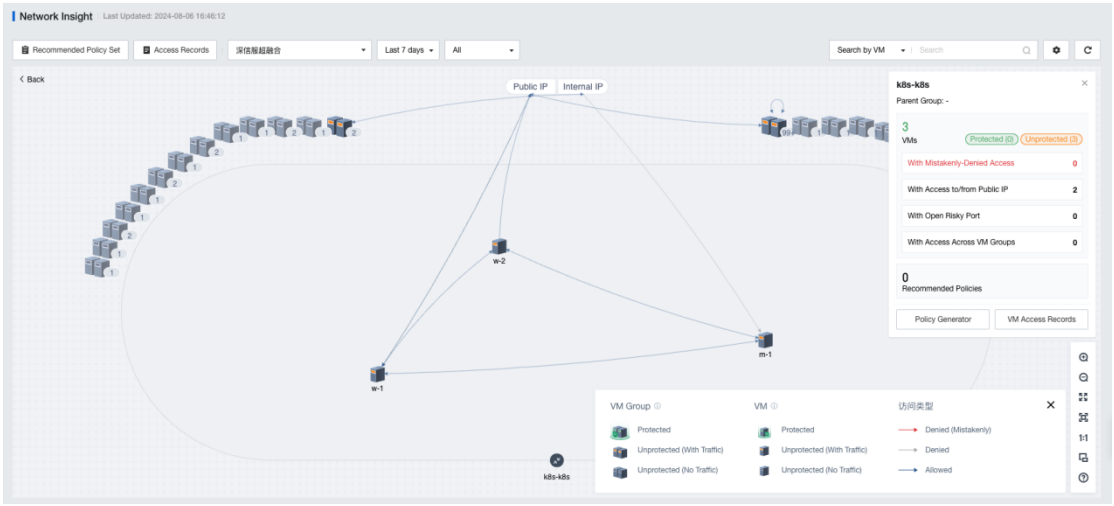


Figure 368: Network Insight

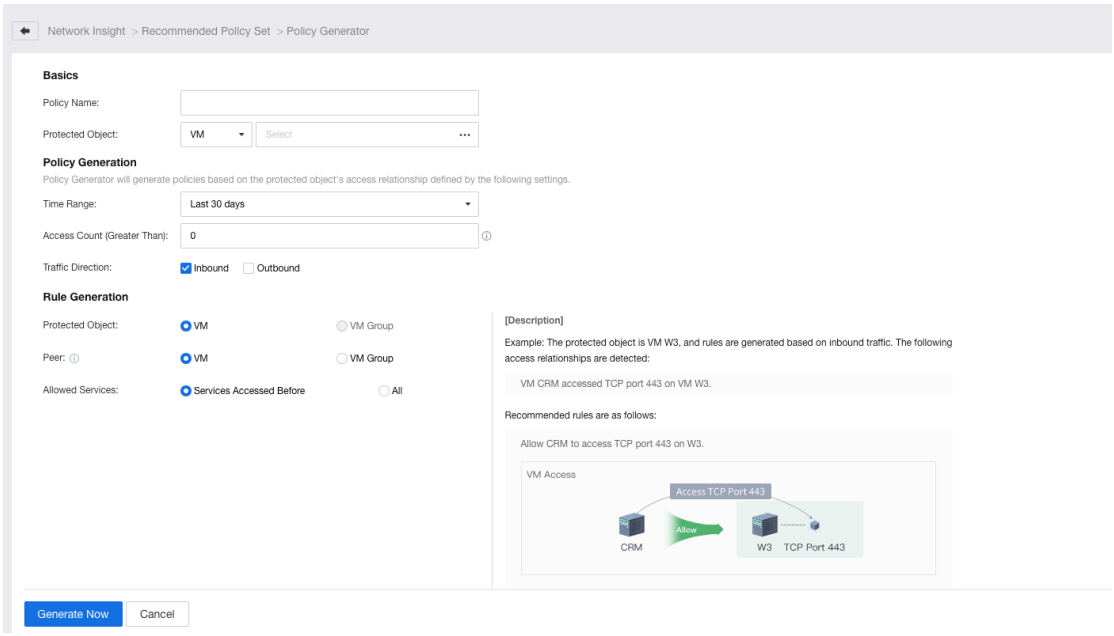


Figure 369: Policy Generator

Go to the details page of the generated recommended policy and click **Pre-Release**. During this period, only the **Mistakenly-Denied Access** alert may be triggered (whose triggers can be set on the **Networking** > **Network Insight** > **Alert Options** page), and traffic will not be denied. The administrator can determine the micro-segmentation policy based on the trial operations of the business services for a period before enforcing the policy.

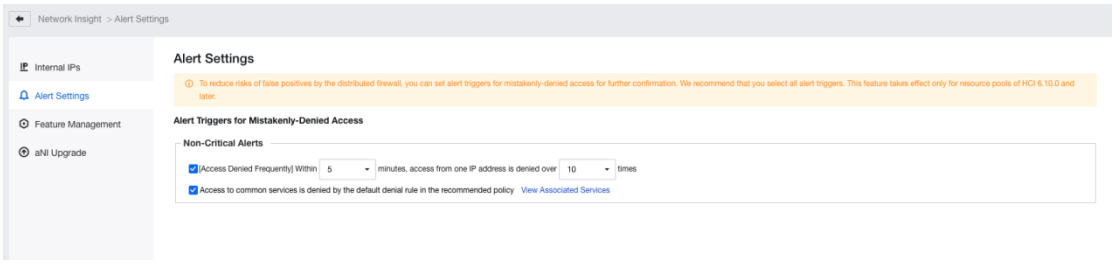


Figure 370: Alert Triggers for Mistakenly-Denied Access

Precautions:

1. Up to 500 Mbps of bandwidth can be occupied by a physical host.
2. Network Insight occupies the bandwidth of the management interface in single NIC mode and the bandwidth of the second NIC in dual NIC mode.

3. VM groups including more than 80 VMs cannot be expanded on the UI.

aNI Size	vCPU	Memory	Disks	VMs	Max Stream
Small	8	16 GB	500 GB	100	100,000/5 mins
Medium	32	64 GB	1 TB	1,000	1,000,000/5 mins
Large	64	128 GB	1.5 TB	3,000	3,000,000/5 mins

Figure 371: Network Insight Sizes

5.3.10 Traffic Mirroring

As SCP supports more and more critical business systems, customers have been concerned about the security and visualization of east-west traffic within the platform, especially in the following scenarios:

1. Audit: Traffic auditing is necessary for governments and financial organizations that have high security and compliance requirements. Traffic mirroring can transparently mirror instance traffic to the unified auditing platform for analysis to meet audit needs.
2. Network O&M: Traffic mirroring allows O&M personnel to troubleshoot network issues by checking the status of network transmission (such as a TCP retransmission), eliminating the need to capture packets within the VM.
3. Security analysis: Self-developed or third-party security software can thoroughly inspect traffic to capture all possible security vulnerabilities and threats for quick responses.

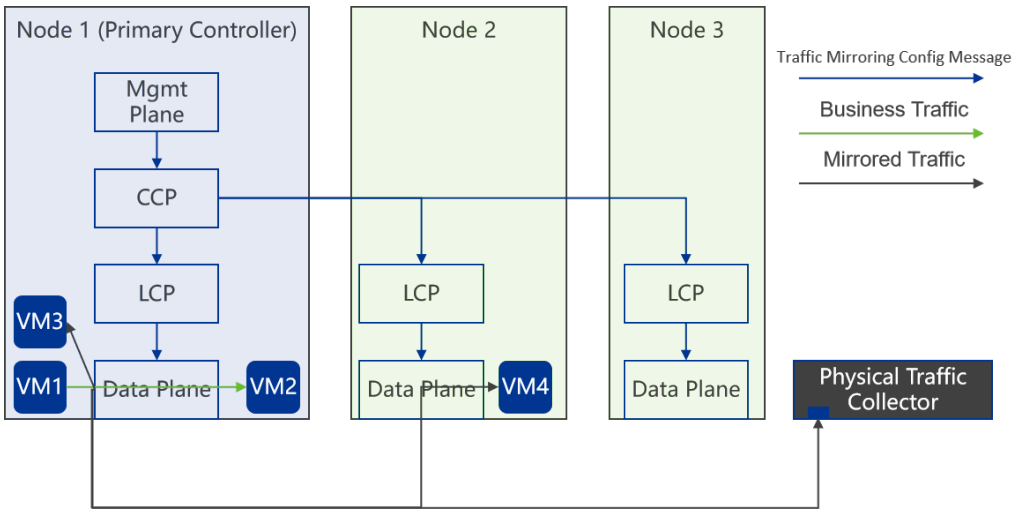


Figure 372: Traffic Mirroring

Traffic mirroring in HCI can be decoupled from the network topology, allowing mirroring of the network interface traffic of any VM to another VM without being restricted by the virtual network topology.

Traffic mirroring involves three network planes: the management plane, the control plane, and the data plane.

1. The management plane receives traffic mirroring rules configured.
2. The control plane converts, computes, and distributes configurations from the management plane. It contains the following two components:

- The CCP saves and delivers the configuration of the management plane and broadcasts the traffic mirroring configuration to the LCPs of different nodes.
- The LCP saves the configuration of the local node, splits, filters, and calculates traffic based on traffic mirroring rules, and delivers the configuration and rules to the data plane.

3. The data plane handles the traffic mirroring function. It delivers the configuration to the virtual switch connected to the mirroring source port and specifies the mirroring source port as the name of the virtual switch port to which the source VM's network port is connected. It also specifies the mirroring destination as the device where the virtual switch port to which the destination port is connected resides. The information includes

the device type, device name, and virtual switch port name. By doing so, the device ID and the virtual switch port ID can be obtained. When a packet passes through the virtual switch port connected to the source port, the data plane matches the traffic direction, checks the sampling rate, copies the original packet, and configures the destination details of the packet copy with the device type, device ID, and virtual switch port ID regarding the destination port. The packet is then transmitted to the network port of the destination VM through the virtual switch port.

Typical Application Scenarios:

The traffic mirroring function can be correlated to Cyber Command to enable fixing and tracking. The process is as follows.

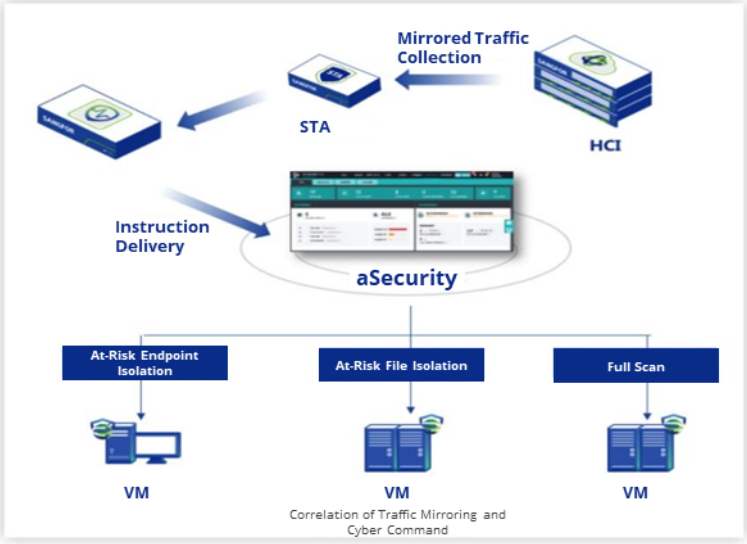


Figure 373: Correlation of Traffic Mirroring and Cyber Command

1. Mirrored traffic is transmitted without the use of proxies, eliminating the need for installing a proxy on each VM of external security devices for traffic audit.
2. Endpoint-side massive data is uploaded to help with the analysis and tracking of external security devices. HCI will directly transmit traffic to the external device, which can utilize the complete traffic information to perform comprehensive analysis and trace potential threats more effectively.

3. Complementary network-endpoint detection mechanisms can reduce false alerts. External security devices carry out detection based on mirrored traffic, while aSecurity detects files and activities on the endpoint side. When a threat is detected, this function sends a fixing instruction to aSecurity. The correlated solution allows network-endpoint security products to work together, enhancing detection accuracy and minimizing false alerts.

5.3.11 Network O&M

5.3.11.1 What-You-Draw-Is-What-You-Get

The network topology, based on business logic, is a unique function of Sangfor HCI, offering a "What-You-Draw-Is-What-You-Get" representation. Pooling of computing and storage resources is achieved through aSV and aSAN, while aNET enables network resource pooling. These components provide a range of elements for building business logic topologies. The management platform facilitates the fast retrieval of resources from the pools, allowing for the presentation of diverse topologies. When building business logic topologies on the management platform, the underlying HCI architecture executes multiple actions and instructions. It also simulates the underlying environment according to the business logic topology, effectively simplifying the platform infrastructure. This enables IT administrators to quickly and intuitively construct the necessary logical topologies for diverse data center services.

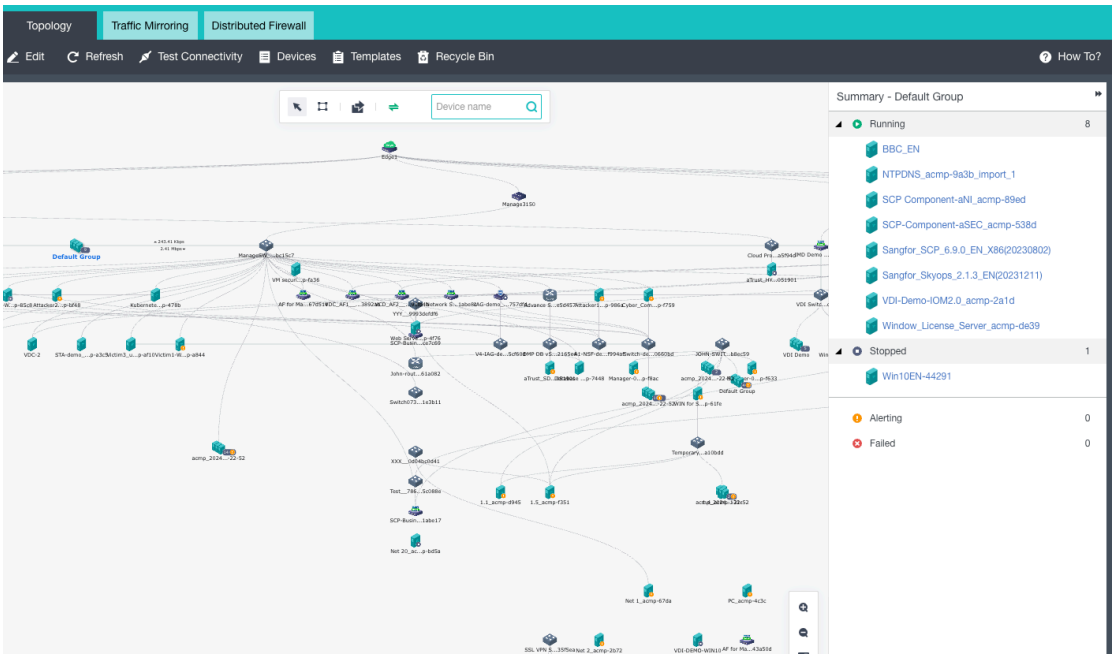


Figure 374: Topology Featuring "What-You-Draw-Is-What-You-Get"

As shown in the figure, administrators can efficiently retrieve resources from the pools by dragging or connecting elements with lines. This allows them to orchestrate businesses and construct diverse topology architectures.

5.3.11.2 LLDP

The Link Layer Discovery Protocol (LLDP) is a link layer protocol specified in IEEE 802.1AB. As a standard Layer 2 topology discovery protocol, LLDP can send the information of a local device, such as management IP address, device ID, and interface ID, to neighbors. Neighbors then save the received information in their Management Information Bases (MIBs) for querying and link status determination by the network management system.

With the continuous expansion of network scales and the increasing variety of network devices, each with its own complex configurations, there is a growing demand for advanced network management capabilities. Traditional network management systems have limitations in their capability to analyze network topology, typically only reaching up to Layer 3. As a result, they are unable to provide detailed information about the topology of network devices or detect configuration conflicts. Therefore, a standard Layer 2 information exchange

protocol is needed. LLDP offers a standard link layer discovery method. By leveraging LLDP, users can quickly obtain Layer 2 information of devices to determine the topology status of connected devices. The paths connecting clients, switches, routers, application servers, and network servers can be presented. Additionally, LLDP can detect configuration conflicts and investigate the causes of network failures. Enterprise network users can utilize network management systems to monitor the link status of devices supporting LLDP, facilitating quick fault locating in case of network failures.

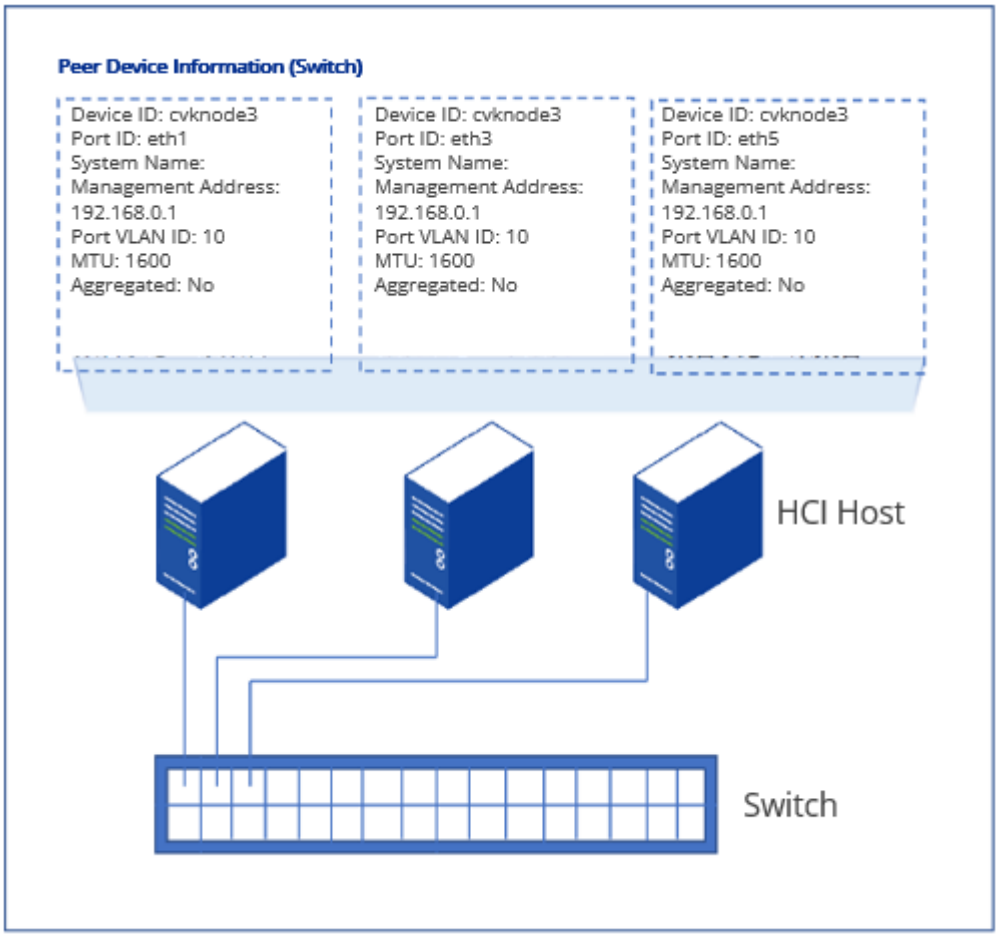


Figure 375: LLDP

1. The LLDP module updates its own LLDP local system MIB and locally defined LLDP extension MIB by interacting with the physical topology MIB,

entity MIB, interface MIB, and other types of MIBs on the device through the LLDP agent.

2. Local device information is encapsulated into LLDP frames and sent to remote devices.
3. The remote device receives the LLDP frames and updates its own LLDP remote system MIB and the custom LLDP extension MIB.
4. By using the LLDP agent to send and receive LLDP frames, the device obtains clear information about the remote device, including the connected interface and MAC address.

The LLDP local system MIB stores local device information such as device ID, interface ID, system name, system description, interface description, and network management address. The LLDP entity can maintain the MIBs of the LLDP local and remote systems. The LLDP local system MIB information is obtained and sent to remote nodes when the status of the local device changes. When there are no changes in the local device status, the LLDP local system MIB information is obtained and sent to remote nodes periodically. Received LLDP frames are identified and processed. When there are changes in the status of the LLDP local system MIB or LLDP remote system MIB, LLDP alerts are sent to the network management system.

5.3.11.3 Fault Prevention and Handling

5.3.11.3.1 Reliability of Layer 3 Forwarding Network

The virtual routers at the aNET network layer are centralized routers. Traffic requiring Layer 3 forwarding needs to be routed through these routers. However, if a node where the router is located experiences a failure, or if the business interface connected to the router cannot communicate, the communication between the devices connected to that router will be affected.

aNET provides the router HA function to ensure the reliability of the Layer 3 forwarding network. The network controller monitors the running status of hosts and the status of business interfaces in the cluster in real time. When a

host failure or a failure in business interface communication is detected, the CCP calculates the affected virtual routers and automatically switches them to other working hosts, ensuring that traffic routed through these routers can be forwarded properly.

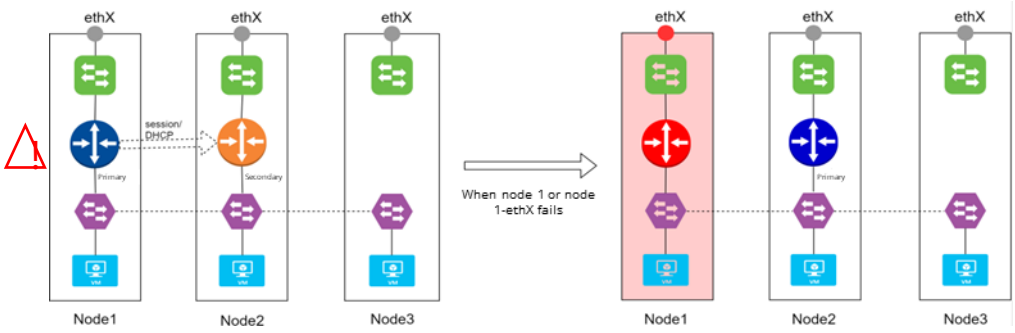


Figure 376: Auto Fixing of Virtual Router Faults

5.3.11.3.2 Network Connectivity Test

The network connectivity test function utilizes tagged ping/TCP/UDP packets to track and record each hop in the data plane. The information of each hop includes the virtual device and forwarding port names. With all recorded information, the packet path to a specific destination can be determined. This method is similar to tracert but offers more detailed detection of outbound and inbound traffic paths, enabling users to locate network issues more intuitively.

This function allows users to customize the construction of detection packets (IPv4/IPv6, TCP, UDP, or ICMP) for connectivity tests on SCP. Connectivity tests on VM network interfaces are supported.

On the network connectivity test page, as shown in the figure, users can specify the IP protocol version, source and destination hosts, and TCP/UDP/ICMP protocol ports. The platform starts the test after users click **Start**, automatically locates the fault, and provides detailed guidance. Administrators can troubleshoot issues based on these prompts, significantly reducing O&M workload.

5.3.11.3.3 VXLAN Connectivity Test

aNET performs regular connectivity tests on the VXLAN by conducting ping tests between the IP addresses of VXLAN interfaces on each host. If there is no response within 5 seconds, it triggers a VXLAN failure alert and displays the connectivity status of the VXLAN of each host, helping users quickly locate the VXLAN link failures of the cluster network. Moreover, for users utilizing high-performance VXLAN, aNET supports the detection of jumbo frames on the VXLAN.

5.3.11.3.4 Network Monitoring and Alert

HCI offers continuous monitoring capabilities for both physical and virtual networks. When the specified alert conditions are met, the platform generates alerts. Custom alert thresholds are supported for certain alert conditions.

- Virtual network alert

Virtual network alerts are generated for virtual network devices, including NFV devices and virtual routers. The connectivity test verifies if the virtual network devices are properly connected to external networks, while network performance monitoring allows customization of packet loss rates and ALG utilization on network device interfaces.

- Physical network alert

Physical network alerts target hardware, covering compatibility checks of NIC firmware, exception detection of NIC optical modules (Tx/Rx power, temperature, and voltage), detection of NIC damage, and packet loss/error detection of network interfaces with customizable thresholds. These alerts allow IT O&M personnel to promptly identify hardware failures and enhance troubleshooting efficiency.

1. Packet error alert: The platform retrieves the interface packet error information, classifies the error type (such as electrical signals or other types), determines the cause of interface faults (such as loose interface connection or excessive optical attenuation), and generates alerts accordingly.

2. Ping monitoring: The platform sends ping packets to other hosts to verify their online status and monitors delays between the host management and virtual networks. Ping monitoring helps pinpoint issues such as network connectivity failures or high ping latency in the event of host packet loss or VXLAN disconnections in the cluster.
3. Abrupt interface disconnection: The platform detects the sudden disappearance of the interface PCI address, indicating an abrupt interface disconnection.
4. Driver compatibility: The platform supports driver or firmware upgrading when there are driver or firmware alert logs in the kernel logs.
5. Session statistics: This function determines if the current virtual network device involves large-number session creations causing performance issues based on the number of created and destroyed sessions per second. Session statistics primarily address packet loss in high-concurrency scenarios. If abrupt session number changes exist and packet loss occurs, the number of sessions created and destroyed can be viewed through session statistics to assist in troubleshooting.
6. Viewing DPDK network interface statistics: This function is used to view statistics about managed interfaces.

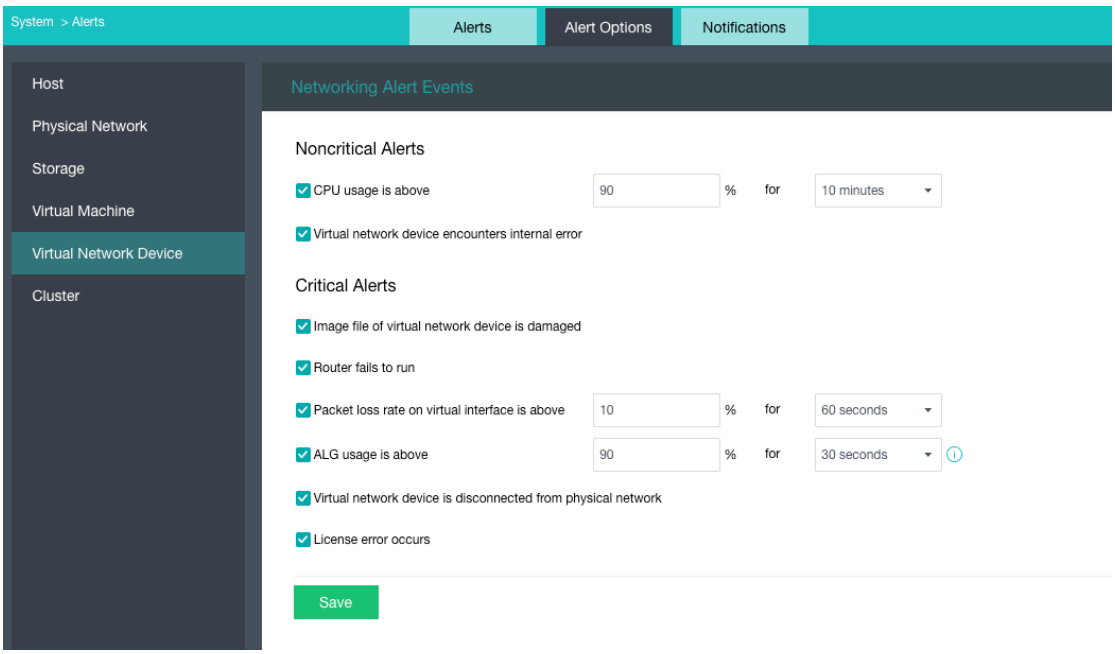


Figure 377: Alert Configurations

Precautions:

1. Session statistics: Session statistics need to be manually enabled and should be disabled after troubleshooting is completed. Once enabled, the virtual network performance may decrease by 15%.
2. Viewing DPDK network interface statistics: Prior notification to the data plane is required to obtain interface statistics data.

5.3.11.4 VM NIC Rate Limiting

aNET provides NIC-level QoS guarantees for connected network devices (VMs). It enables limiting the maximum network speed of the corresponding NIC of a VM, ensuring network bandwidth for critical VMs and preventing abnormal VMs from attacking the aNET platform.

aNET implements rate limiting and shaping for data packets in both the Tx and Rx directions of the VM NIC. This is achieved through the use of the token-based bucket algorithm and a dual-bucket structure consisting of the C bucket and the E bucket. Tc represents the number of tokens in the C bucket, while Te represents the number of tokens in the E bucket. The total capacities of the two buckets are CBS and EBS, respectively. Both buckets are initially full, and the initial Tc and Te values are equal to CBS and EBS, respectively. Tokens are generated at a rate of CIR, typically by adding tokens to the C bucket first. When the C bucket is full, tokens are added to the E bucket. If both buckets are full, any newly generated tokens will be discarded. Tokens are consumed in bytes to perform rate limiting operations on data packets in both directions.

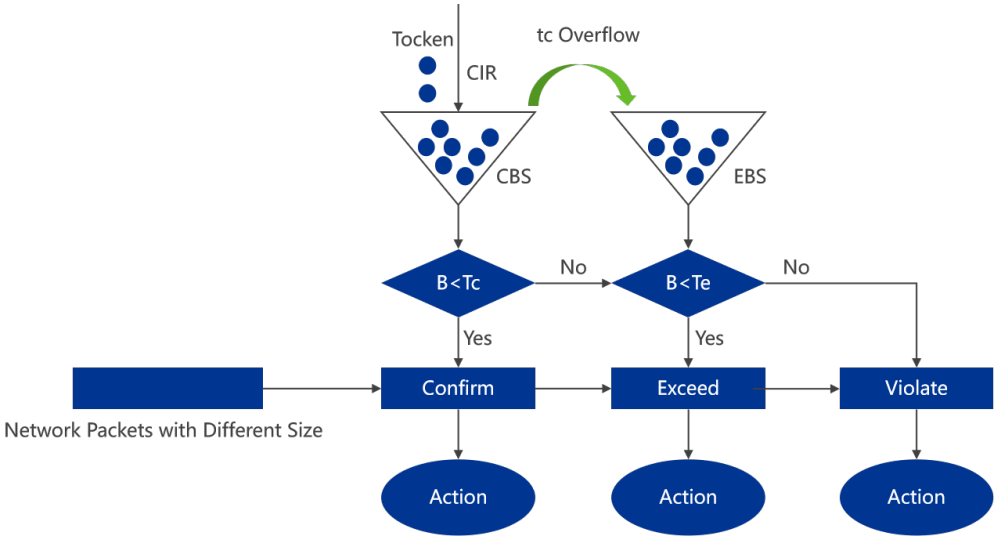


Figure 378: Token-Based Bucket Algorithm

The NIC rate limiting function supports all common deployment scenarios of HCI, including direct VM connection to edges, VM connection to tunnel interfaces, and VM connection to edge routers.

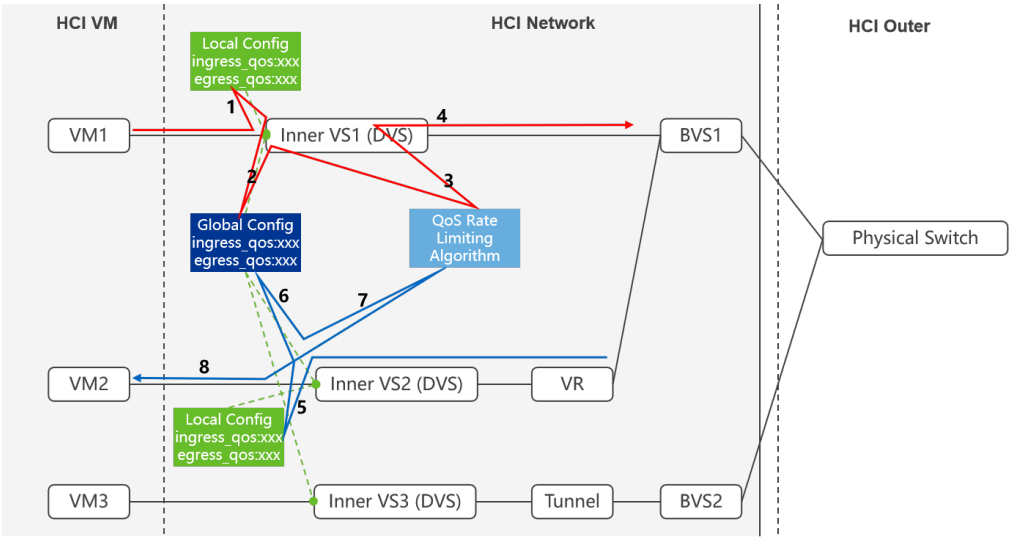


Figure 379: Network Traffic Rate Limiting

As shown in the figure below, users can enable the traffic rate limiting function for a specific NIC on the VM NIC settings page. Both outbound and inbound traffic can be configured with network rate limits ranging from 1 Mbps to 20,000 Mbps.

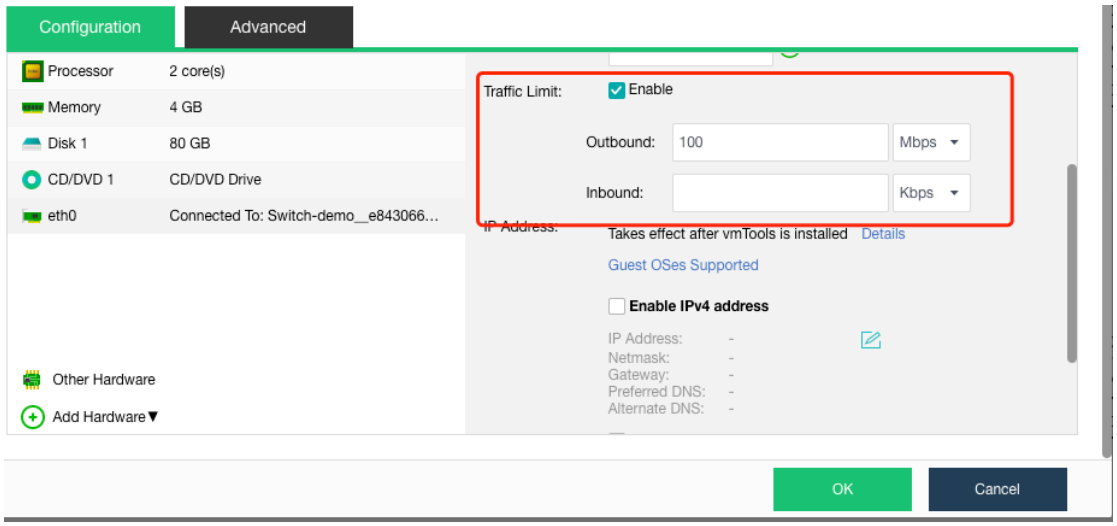


Figure 380: Traffic Rate Limiting Configuration

The actual measurement indicators ensure that the Tx rate of network traffic passing through the corresponding NIC is limited in the range with a 5% error.

5.3.11.5 Network Health Monitoring

A network can be unhealthy due to hardware or software issues (e.g., loosely seated NICs, loosely seated or too hot optical modules), which can greatly degrade performance and affect the system's running (for example, some VM networks encountering latency, or networks between clusters encountering high latency or too many packet losses).

Network health monitoring focuses on physical links from NICs to switch ports. When an unhealthy issue (packet loss, latency) occurs, it can proactively disconnect or isolate the failed link and alert the user to avoid business impact.

As shown in the figure below, the hosts in a cluster are arrayed in rings, with each detecting up to five. The figure shows the host links to be probed by hosts A, B, and C.

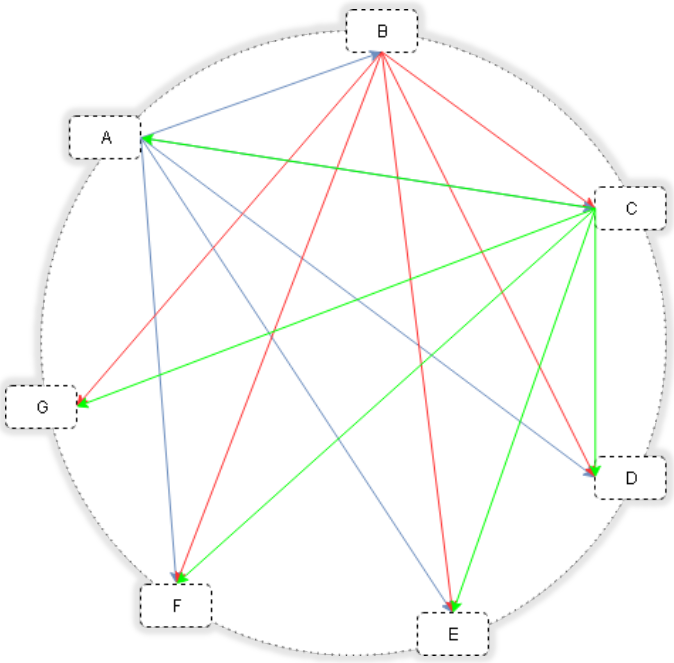


Figure 381: Host Array in Network Health Monitoring

As shown in the following figure, if the active HCI physical interface can receive a response packet from the responding HCI host after initiating a probe request, then the probe is successful. If no response packet is received, a packet loss is recorded. Single probe latency formula: RTT = t4-t1-(t3-t2).

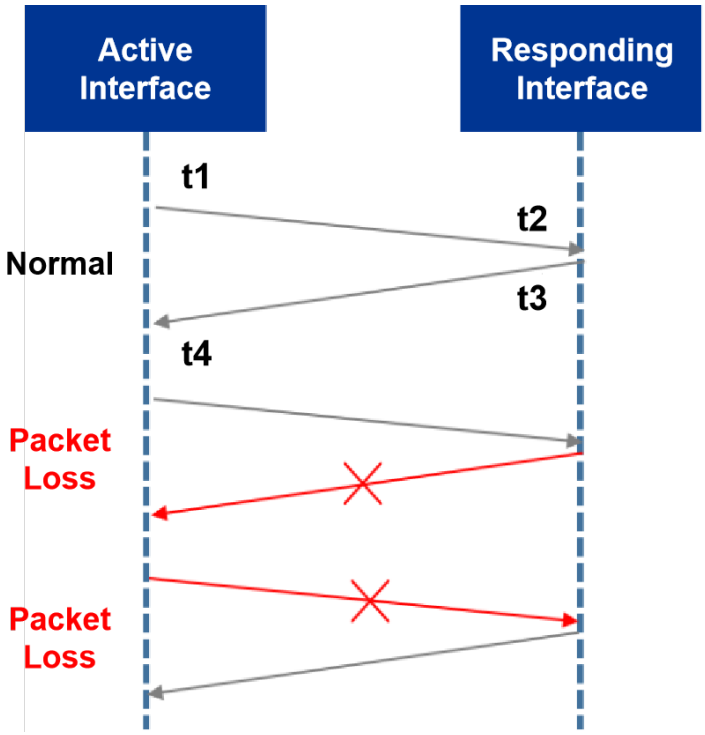


Figure 382: Network Health Monitoring Principle

The network health monitoring detects the packet loss rate and latency of the network link between two server nodes by sending detection packets. When the specified threshold is reached, an alert will be triggered. For different communication networks (management, overlay network, edge-connected, storage network, external storage communication, and aDesk communication networks), the packet loss and latency thresholds and the corresponding actions are provided.

Limitation	Description
Number of Aggregate Interfaces	Storage area network: Network health monitoring is supported only if standard link aggregation is enabled and the number of physical interfaces is less than or equal to 2. Network health monitoring is not supported if the vs-bond mode of earlier versions is used. Non-storage area network: Network health monitoring is supported if the number of member physical interfaces is less than or equal to 2. If network health issues are detected

	in scenarios with link aggregation disabled, alerts are reported, but physical interfaces are not deactivated.
Aggregation Mode	Health issue alerting: Alerts are reported if network health issues are detected in the following aggregation modes: active/standby, manual load balancing, and LACP. Health issue handling: For virtual storage area networks, health issues detected in the following modes are handled: active/standby, manual load balancing, and LACP. For non-virtual storage area networks, health issues detected in manual load balancing mode are not handled.
Switch Configuration	Load balancing policy: Load balancing based on the source address or the source and destination addresses at Layer 2, Layer 2+Layer 3, and Layer 2+Layer 3+Layer 4 are supported. If combined load balancing policies are supported, you need to select the load balancing mode based on the source and destination addresses or source MAC address (e.g., NVIDIA and Mellanox switches).
Virtual Storage Area Network	Network health monitoring is not supported for a datastore with 2 nodes or with 2+1 (2 HCI nodes+1 witness node) and the witness node of a stretched datastore.
Cluster Communication Domain	Network health monitoring is not supported in cluster communication domains.
Interface Multiplexing	Network health monitoring is not supported for a storage network interface with interface multiplexing enabled. The network health monitoring settings (including the thresholds and actions) of a storage area network are synchronized from other network planes.
Network Segments in IP Pool of VXLAN Network	Network health monitoring is not supported in scenarios where different network segments of a VXLAN network communicate over Layer 3.

System Resource Load	Network health monitoring is not supported for an overloaded HCI node (with a CPU idle rate lower than 10% or available memory size less than 2 GB).

Table 16: Network Health Monitoring Limitations

The deactivation of an interface is performed in a manner related to its aggregation mode and network plane, as shown in the following table:

	LACP	Non-LACP
Storage Network Interface	Remove failed interface from aggregate interface	Disable failed interface
Non-Storage Network Interface	Remove failed interface from aggregate interface	Disable failed interface only in active/standby mode

Table 17: Interface Deactivation of Network Health Monitoring

5.4 aSecurity

5.4.1 Overview

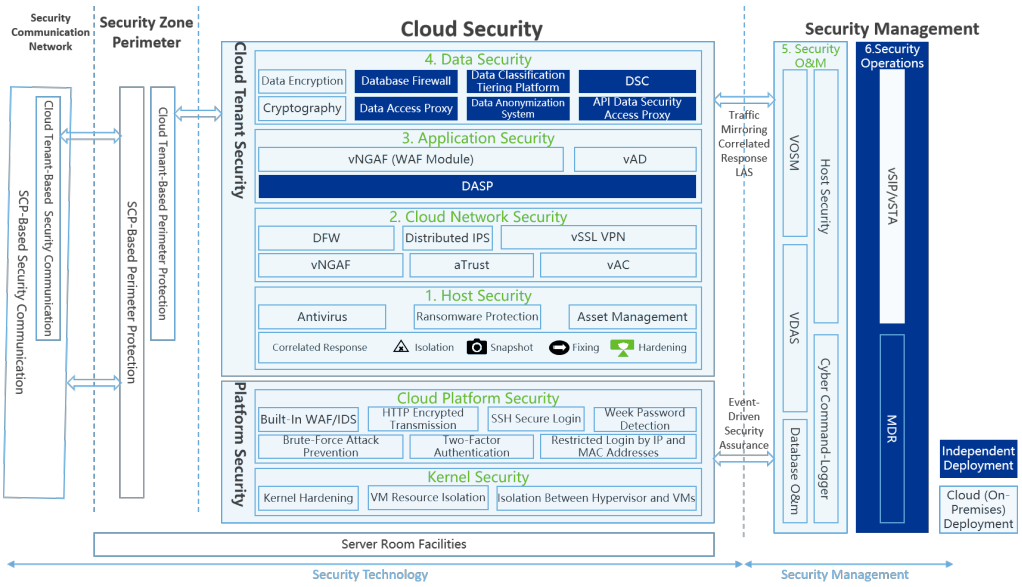


Figure 383: Architecture of Cloud Data Center Security

aSecurity enables the orchestration of security components in the virtual

network topology through virtualization, allowing for rapid deployment and auto-scaling.

Sangfor HCI incorporates built-in security and aSecurity-NFV to provide built-in virtualized security functions. aSecurity offers capabilities such as VM BSI and distributed IPS, enabling functions like antivirus, anti-ransomware, intrusion defense, and application security protection. aSecurity-NFV provides a diverse range of security services through the security resource pool, including vSSL VPN, vNGAF, aTrust, vAC, vOSM, vDAS, Sangfor Host Security, SIP-Logger, and vSIP & vSTA.

Platform security encompasses the basic security capabilities provided by SCP and HCI. Cloud security contains the security capabilities and guarantees offered by SCP to its users (tenants). Some of these capabilities are configured by default, such as tenant isolation, while others, like data encryption, require manual configuration and enabling by users.

NOTICE

1. This section focuses specifically on the implementation of internal security of cloud data centers. Traditional edge security of cloud data centers is not covered here but can be found in the relevant product white papers of the security BG.

2. The cloud security and security management components in gray in the cloud data center security architecture are temporarily not integrated into HCI & SCP 6.9.0. If required, please contact the security BG. Most components are deployed as software platforms.

5.4.2 BSI

BSI is a concept that allows a cloud platform to directly provide security capabilities for cloud-based services, eliminating the need for add-on security components. In HCI, this concept is implemented through the construction of aSecurity that offers security capabilities to cloud-based services. Compared to add-on security components, BSI can better meet business requirements, ensuring security and reliability while reducing security O&M costs. This brings significant benefits to enterprise digital transformation.

Currently, customers' cloud services are facing the following problems:

1. Weak security awareness: lack of business deployment standards and pre-operation checks; relying on events or policies; desire to invest in security without a systematic approach
2. Insufficient professional capabilities and over-reliance on "artificial intelligence": no adequate fixing skills, fixing with poor timeliness, reluctance to fix incidents, and reliance on vendors or outsourcing suppliers
3. Separation of service, O&M, and security: complex and long process covering issue discovery, O&M troubleshooting, security check, issue fixing, and service recovery

To address these problems, Sangfor HCI integrates years of experience with robust security components. By providing aSecurity for unified management and coordinated analysis of security components, it offers classified protection compliance solutions to help users deploy security protection solutions that adhere to defined compliance guidelines. It also provides security analysis and enables the formation of comprehensive security protection solutions for cloud platforms. The aim is to achieve the following customer experience objectives:

1. Early security protection: Establish an automated operation process that ensures security and compliance upon operation, helping users to develop a standard operation process.
2. Transformation from "human intelligence" to "artificial intelligence": Simplify security O&M by creating a security assistant that directly performs actions by itself. Users do not need to rely on documents or phone calls when addressing security issues.
3. Integration: Integrate service, O&M, and security and adopt a unified perspective that prioritizes services.

The current aSecurity covers VM BSI based on Endpoint Secure and network BSI based on IPS.

5.4.3 aSecurity

The technical architecture of aSecurity consists of three layers:

1. Basic Platform Layer

The core technology is Hypervisor, which virtualizes physical resources to maximize resource utilization. HCI provides basic computing, storage, and network functions, while SCP offers cluster, tenant, and permission management capabilities.

2. Core Engine Layer

Based on the capabilities at the basic platform layer, the core engine layer encapsulates interfaces of different platforms (such as Endpoint Secure, CWPP, Neural-X, HCI, and SCP). It integrates the capabilities of various management platforms and security platforms to develop basic component functions (such as multi-version management and AK/SK-based authentication) for aSecurity.

3. Function Presentation Layer

Based on the basic component functions provided by the core engine layer, actual business scenarios, and function characteristics, this layer is divided into multiple functional modules for management, such as MGR management, threat management, and vulnerability management. The business interfaces are called on the UIs that are embedded in HCI and SCP pages. Interface requests are routed from the platforms to Sangfor-WAF of aSecurity and then to the aSecurity services. Then, the UIs receive service responses.

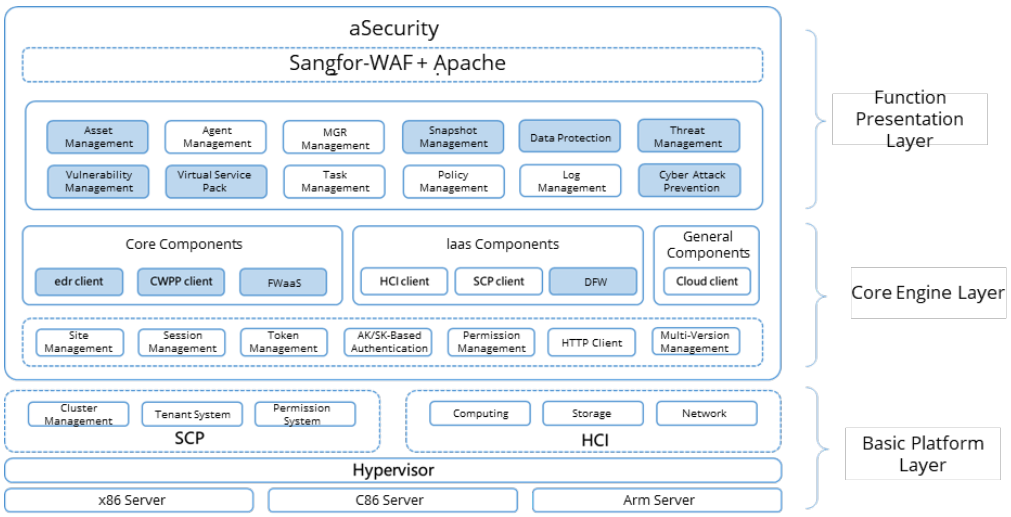


Figure 384: aSecurity Architecture

5.4.4 VM Security

5.4.4.1 Overview

Host security ensures the confidentiality, integrity, and availability of data stored and processed on hosts through various means. It encompasses security measures for hardware, firmware, and system software, as well as additional security technologies and management measures to establish a comprehensive host security environment. Host security requirements include identity authentication, access control, security audit, residual information protection, intrusion prevention, malicious code prevention, and resource control. VM security involves threat detection and management mechanisms that can identify, analyze, and prewarn security threats and vulnerabilities for unified security management. Sangfor HCI protects VMs through VM BSI.

VM security involves threat detection and management mechanisms that can identify, analyze, and prewarn security threats and vulnerabilities for unified security management. Sangfor HCI also protects VMs through aSecurity.

5.4.4.2 Architecture

The technical architecture of VM security is shown in the figure below.

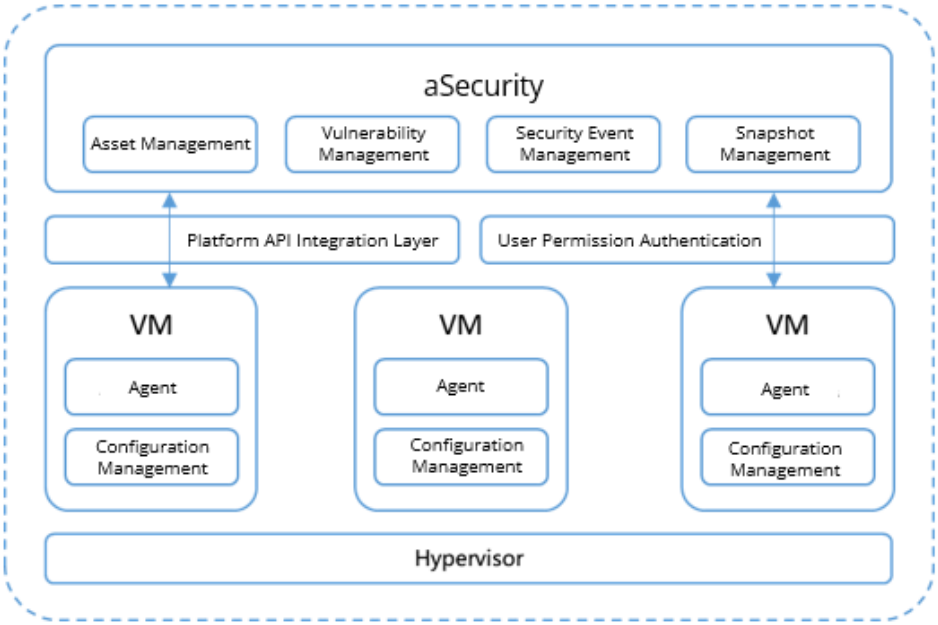


Figure 385: Security Architecture

VM security capabilities include security upon operation, asset identification, one-click threat fixing, and data protection (correlated with snapshots).

5.4.4.3 Security Upon Operation

Easy deployment:

Automatic deployment of endpoint security protection: Once aSecurity operates normally, it automatically installs protection components on assets and enables the protection function for identified assets. This eliminates complex and error-prone manual installation.

Security control:

aSecurity monitors assets, business running status, internal application version, and modifications once it operates normally. It can display all current vulnerabilities and risks (viruses and vulnerabilities) in real time. It also offers convenient entrances for bulk virus and vulnerability scanning and fixing of risky assets, simplifying routine security O&M.

It has integrated the Endpoint Secure capabilities (application identification,

virus scanning, vulnerability scanning, and real-time protection) and uses dedicated security VMs for virus protection. Business hosts only need to automatically install a lightweight agent through vmTools to perform necessary threat fixing and response actions with minimal resource consumption.

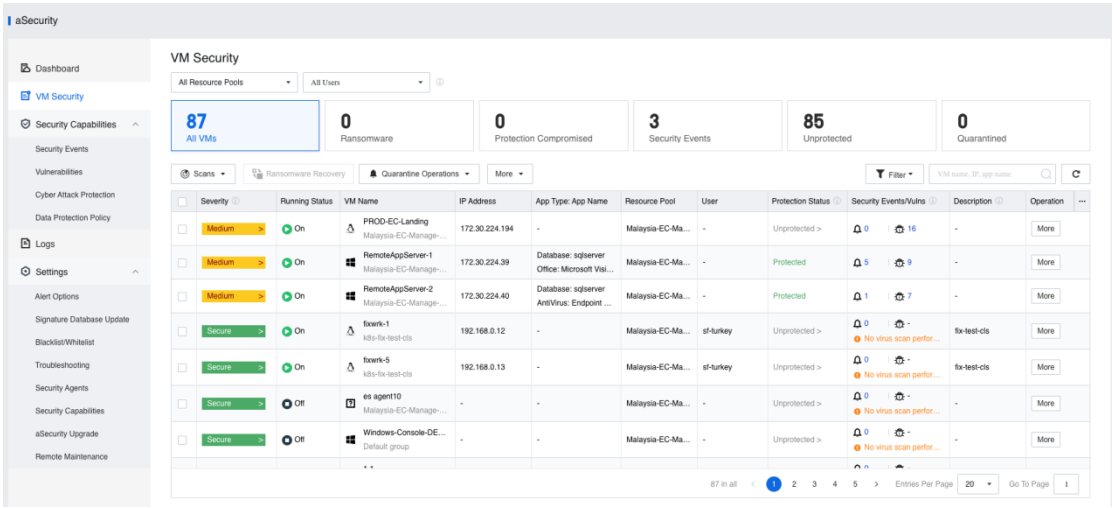


Figure 386: VM Security View

Cloud security lifecycle:

When aSecurity is used, security protection components can be automatically installed using vmTools on VMs after the VMs are created and started. This ensures VM security upon operation. aSecurity can provide security policies by working together with HCI and Endpoint Secure, ensuring real-time VM security protection. In the event of viruses, vulnerabilities, or network attacks during the operation of VMs, convenient and quick fixing methods are provided, achieving close-loop quick fixing. When an incident is detected, snapshots can be taken in real time to retain the information, and email notifications are sent for timely processing. aSecurity ensures the protection of VMs from creation to destruction, mitigating potential security risks.

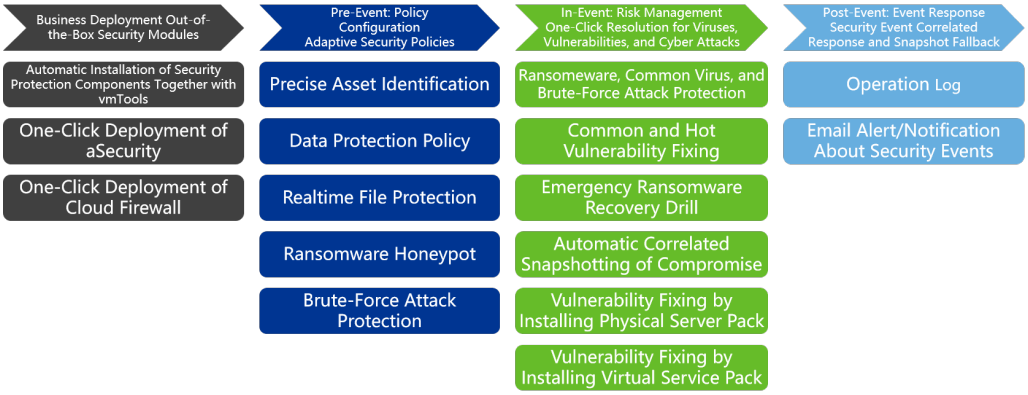


Figure 387: Cloud Security Lifecycle

Agent installation principle:

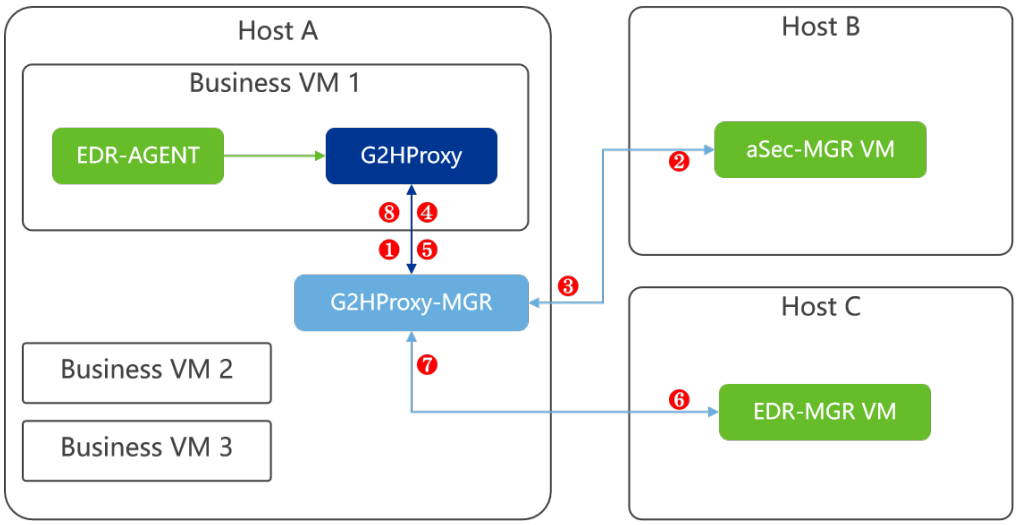


Figure 388: Agent Installation Process

aSecurity issues an agent installation orchestration task to the vmTools on the guest. The vmTools obtains the latest agent installation package from the management platform through the G2Hproxy channel. The guest executes the agent installation task and notifies aSecurity of the installation status.

Network communication between VMs and physical hosts is established using virtio-serial or virtio-vsock. The proxy service on the physical host forwards VM requests to Endpoint Secure and aSecurity. This allows for agent installation on VMs and the execution of virus scanning instructions issued by Endpoint Secure, even when the VMs are not directly connected to Endpoint Secure and aSecurity, achieving security upon operation.

5.4.4.4 Asset Identification

Asset context, launch, and changes can be identified in real time, and relevant information can be collected and displayed in aSecurity, ensuring full-lifecycle asset security management.

1. 1. Leveraging the advantages of cloud platforms, full lifecycle management from resource utilization to business launch is achieved, avoiding complex analysis and asset omissions.
2. 2. Comprehensive asset information, including resource usage, business operation status, internal application versions, and current risks, is displayed in a visualized manner.

Principles:

vmTools and Endpoint Secure capabilities are utilized to automatically collect detailed information such as the OS, web applications, databases, and middleware as well as their versions from VMs through vmTools backend. Endpoint Secure periodically synchronizes the collected asset fingerprint information to aSecurity.

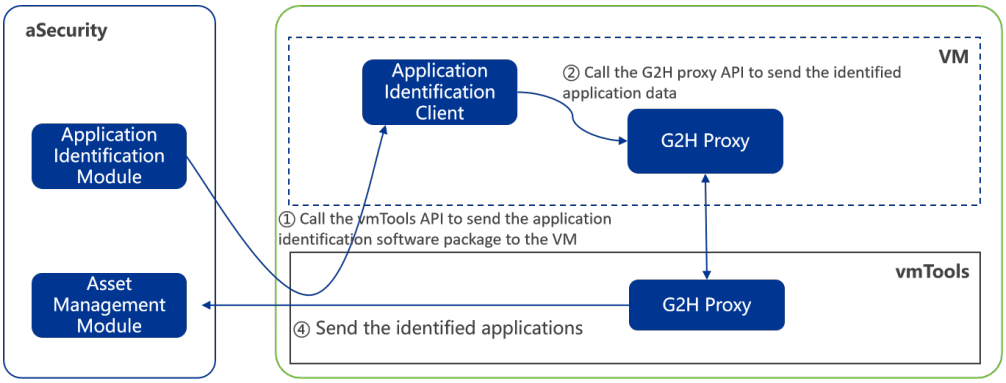


Figure 389: Asset Identification

5.4.4.5 Vulnerability Management

Vulnerability management includes three primary functions: vulnerability scanning, fixing, and overall management, specifically as follows:

- Vulnerability scanning

The EDR-MGR management side regularly reports to aSecurity of asset

information about VMs, applications, and middleware, as well as the corresponding vulnerability information.

By detecting changes in assets, aSecurity automatically assigns scanning tasks to the EDR-MGR. Users can also manually initiate vulnerability scanning on VM endpoints that have installed the agent security component. The agent will return the task progress and vulnerability information to aSecurity in real time.

- Vulnerability fixing

The built-in Endpoint Secure offers interfaces for vulnerability fixing and status querying. aSecurity issues fixing instructions to Endpoint Secure for execution. Results of instruction execution in Endpoint Secure can be queried by aSecurity in real time and displayed on the UI.

Before fixing, snapshots of the VMs to be fixed are taken. If any exceptions occur after the fixing, the VMs can be promptly restored using the snapshots.

aSecurity delivers vulnerability fixing requests through EDR-MGR. Windows system vulnerabilities can be fixed directly, and Linux system vulnerabilities and Windows/Linux application problems can be tagged and fixed.

- Vulnerability management

The EDR-MGR can update its rule database and vulnerability intelligence by correlating with Neural-X. aSecurity provides a visualized UI that classifies the impact of vulnerabilities and actively scans all assets. It highlights high-risk affected assets, helping customers analyze the priorities. This streamlines the fixing of correlated vulnerabilities and VMs.

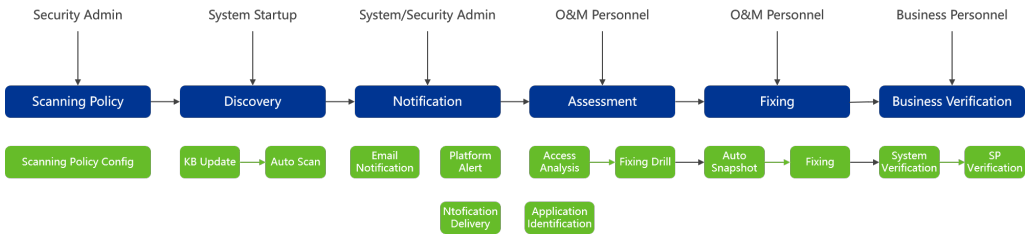


Figure 390: Vulnerability Management Process

5.4.4.6 Ransomware Recovery

aSecurity offers guided fixing of security incidents, network isolation, pre-fixing

snapshots, post-fixing exception recovery, and guided ransomware recovery for encrypted data. This involves network isolation, scene preservation, sandbox verification, security checks, and business launch verification. The built-in EDR-Agent automatically detects VMs infected with ransomware and uploads threat information to Endpoint Secure and aSecurity management side. At the same time, the ransomware recovery process is initiated.

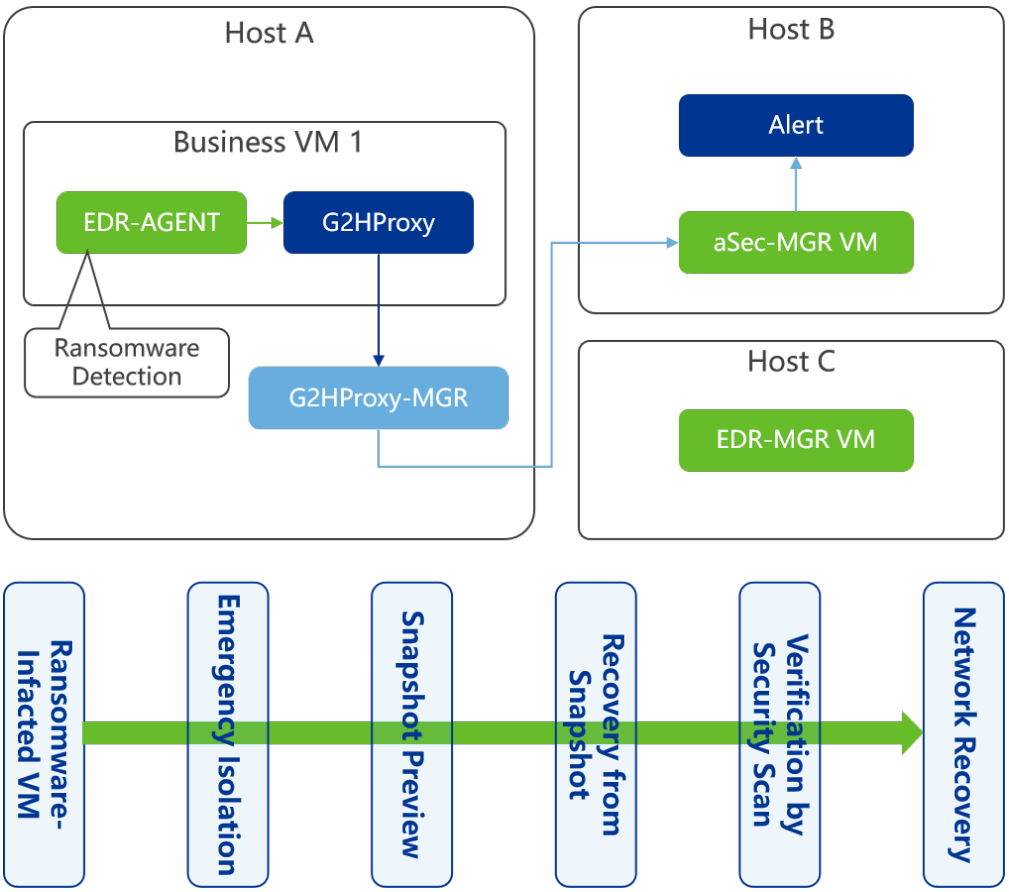


Figure 391: Ransomware Information Reporting

1. Network isolation

aSecurity enables one-click micro-isolation, which can isolate encrypted assets within seconds to prevent virus spread. For assets infected with infectious viruses but requiring business continuity, adaptive isolation is available. This function blocks ports with no traffic during daily operations and high-risk ports, such as shared and remote access ports, commonly used for virus spreading. This minimizes virus spreading paths and increases lateral movement costs

while ensuring business continuity.

2. Data verification

aSecurity utilizes linked clones for business verification, allowing for creation in seconds without additional storage usage. The environment is completely isolated, and restoration to the time point of infection does not lead to virus spread. After determining the recovery point and executing the recovery, encrypted assets are strictly isolated from other assets during the whole process. The assets become online again only after they are confirmed to be safe after a virus scan and isolation is canceled.

3. Business recovery

aSecurity employs a snapshot mechanism to ensure consistency, enabling the recovery of the entire business if any part of assets is infected. This reduces the RTO for business recovery. Before recovery, scene snapshots are preserved for subsequent decryption. After recovery, virus scanning is performed to ensure the security of the environment at the recovery point.

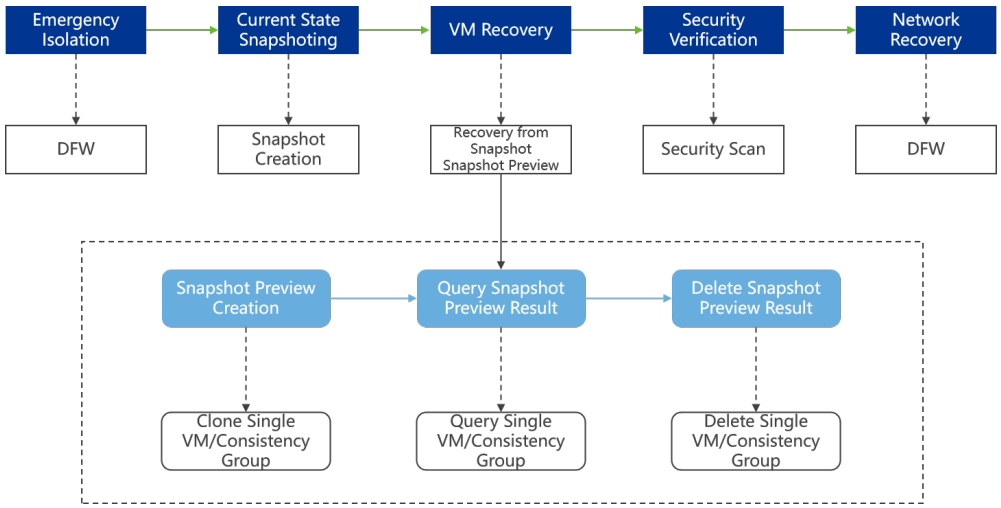


Figure 392: Ransomware-Specific Recovery Process

5.4.4.7 One-Click Threat Fixing

aSecurity integrates the virus scanning capabilities of Endpoint Secure for comprehensive and fast scanning of VM assets and virus fixing.

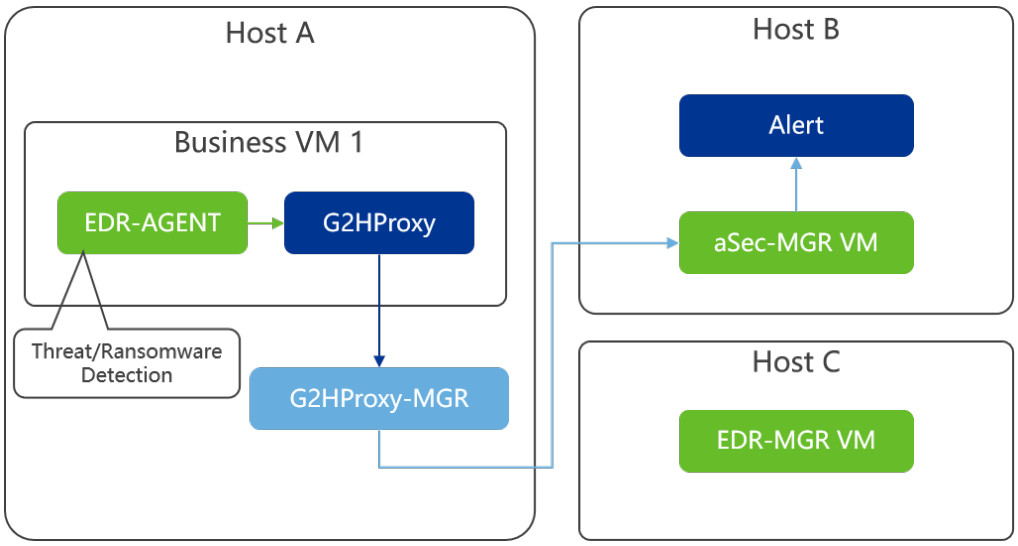


Figure 393: Threat Fixing

1. In scenarios like asset launch and backup recovery, aSecurity obtains status information through internal correlation and intelligently prompts users to perform virus scans. It offers both fast and comprehensive scanning options depending on the scan scope, delivers tasks, and correlates with Endpoint Secure to detect viruses on VMs and obtain results for display.
2. It supports categorizing viruses by risk level, enabling users to analyze priorities with labels. Threat information is presented on a single page, highlighting high-risk VMs. Additionally, it provides convenient fixing for a single VM and one-click bulk fixing for multiple VMs, correlates with Endpoint Secure to fix associated viruses, and quickly restores user assets to the protected state.
3. False positives are inevitable for security software, particularly when AI-based and behavior-based recognition functions are introduced. Some business processes or files may be mistakenly identified as viruses. Moreover, infectious viruses may be injected into critical system files. Improper fixing of these viruses can lead to business or system exceptions. To solve this problem, aSecurity offers file inactivation and isolation before fixing. That is, files are stored in a separate sandbox. Users can take snapshots before fixing to prevent accidental deletion.

Snapshots are used for quick file recovery to enable normal business operations. In the event of a system crash that prevents startup, snapshots can also be used for rapid recovery.

5.4.4.8 Data Protection (Correlated With Snapshots)

1. Daily asset protection is ensured through scheduled snapshots.

Traditional backup solutions are restricted by the backup space and time windows. Generally, core business assets are backed up once a day, general assets are backed up once a week, and non-critical assets are not backed up at all.

aSecurity uses rolling snapshots for backup, eliminating the need for initial full backups of assets. This approach only requires additional storage for incremental data saved every day. Snapshots are taken in seconds without I/O overheads, and the backup process is scheduled during idle hours for efficient snapshot data merging.

2. Snapshots are taken immediately to protect data when security software is uninstalled or suspected ransomware risks exist. The scheduled snapshots are persisted to prevent overwriting with virus-infected data.

Because of the distinctive features of ransomware, most security software provides solid protection against ransomware. Victims of ransomware attacks are often those who have uninstalled security software and then have data encrypted by injected ransomware. Relying solely on security software can only enhance self-protection capabilities but cannot prevent uninstallation.

aSecurity uses assets and multiple behavior detection mechanisms in the cloud. When it detects a suspected uninstallation on an endpoint, it triggers immediate on-site protection through correlated snapshots, ensuring a minimal RPO even in the event of a subsequent ransomware attack.

5.4.5 Cloud Firewall

5.4.5.1 Overview

Traditional cloud security solutions using NFV for network protection face the following challenges:

1. Difficult NFV deployment, involving the creation of VMs, image import, and network topology modification
2. Lack of unified management, often requiring logins and redirection to an independent NFV management UI and reconfiguration of VMs and IP groups on the cloud platform
3. Inability to adapt to dynamic changes in VM businesses, necessitating manual adjustment of security policies after creating or migrating VMs or adjusting networks
4. Lack of elastic scaling, with NFV failures potentially causing network outages

In HCI 6.9.0, Sangfor has implemented servitization of network attack protection to address the issues caused by traditional NFV solutions. Network attack protection can be enabled with just one click, and security policies can be centrally managed in aSecurity, synchronizing asset information such as VMs. Scalability is also provided, allowing specifications adjustment in case of insufficient performance and service degradation in case of excess performance without impacting services. The following benefits are provided to users:

1. Combination of the FWaaS engine and the data plane, enabling deployment-free networks.
2. Policy-based dynamic traffic forwarding technology, providing Layer 7 protection for east-west traffic.
3. Support for virtual patching to fix vulnerabilities existing in VMs.

4. Direct data forwarding at the data plane, eliminating the need for extensive forwarding paths. Compared to traditional NFV security protection methods, the performance is improved by five times.

For classic network scenarios, FWaaS operates in a distributed architecture, protecting all inbound and outbound network traffic of VMs, including east-west and north-south traffic. For VPC network scenarios, FWaaS operates in a centralized architecture, protecting all inbound and outbound traffic at the VPC edge.

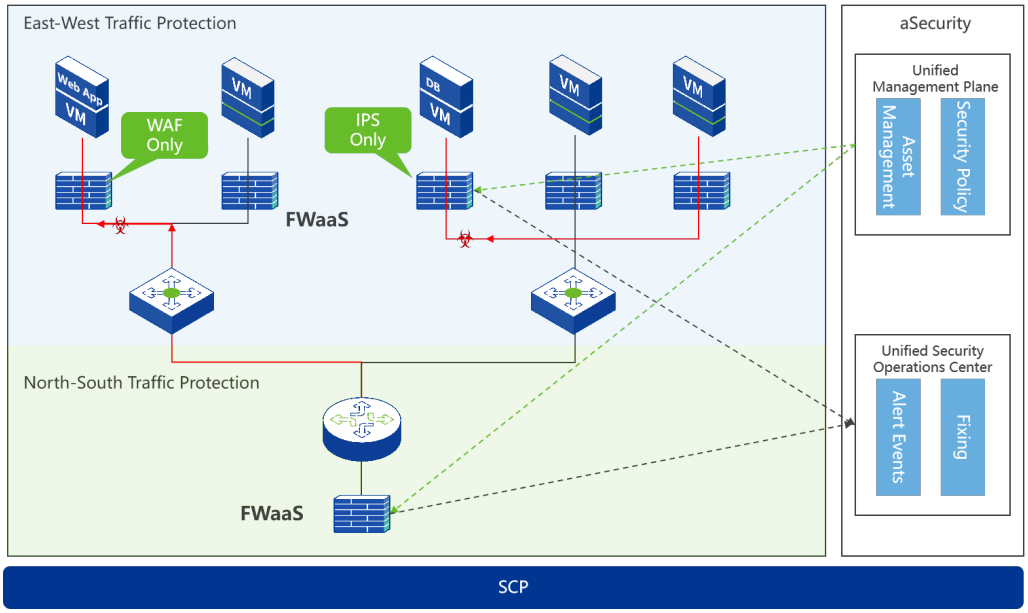


Figure 394: FWaaS Traffic Protection Scenarios

5.4.5.2 Technical Architecture

The technical architecture of the cloud firewall is as follows.

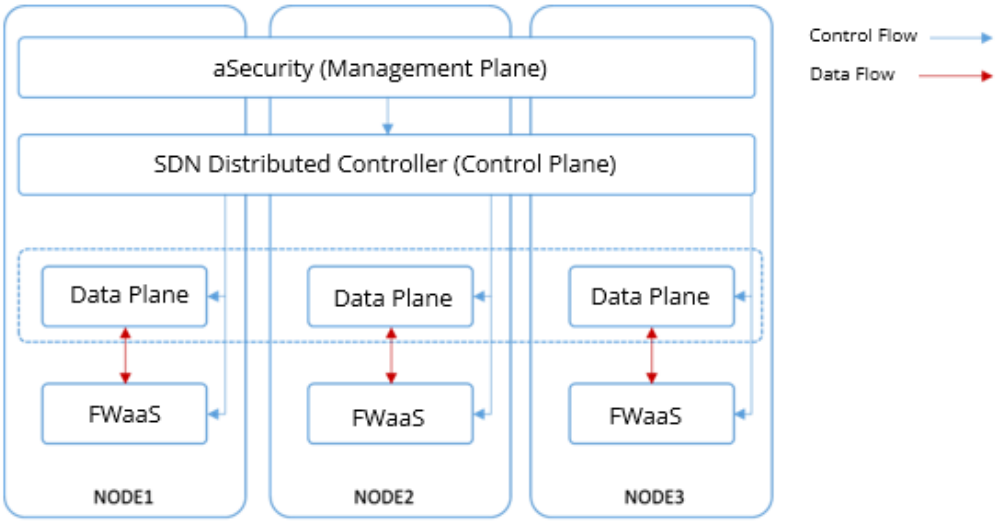


Figure 395: Technical Architecture of Cloud Firewall

- **Management plane**: Provided by aSecurity, it implements functions such as FW instance management and FW policy management. Users can access aSecurity through the Web UI for management operations, and the user-defined FW policy configuration can be persistently saved.
- **Control plane**: The control plane of the cloud firewall reuses the control plane of aNET. The FW policy is distributively delivered by MP-CCP-LCP of aNET. For more information about the control plane technical architecture of aNET, see section 5.3.2.1 "Logical Architecture of aNET."
- **Data plane**: Implemented as independent containers that run the core security detection engine FWaaS. Each HCI node runs an FWaaS container on demand to process the traffic to be protected on that node. The FWaaS engine exchanges data packets with the aNET's data plane through the VirtIO channel and performs real-time detection.

5.4.5.3 Unified Asset Management

Cloud firewall and host BSI are centrally managed in aSecurity, eliminating the need for other management systems or maintaining additional management accounts.

The asset list in aSecurity is synchronized in real time from the VM

management module of the cloud platform. During security policy configuration, VMs can be directly used as objects for configuration. This eliminates the need for asset synchronization based on CMDB and simplifies IP-based management. Even if the IP changes, the security policy remains effective.

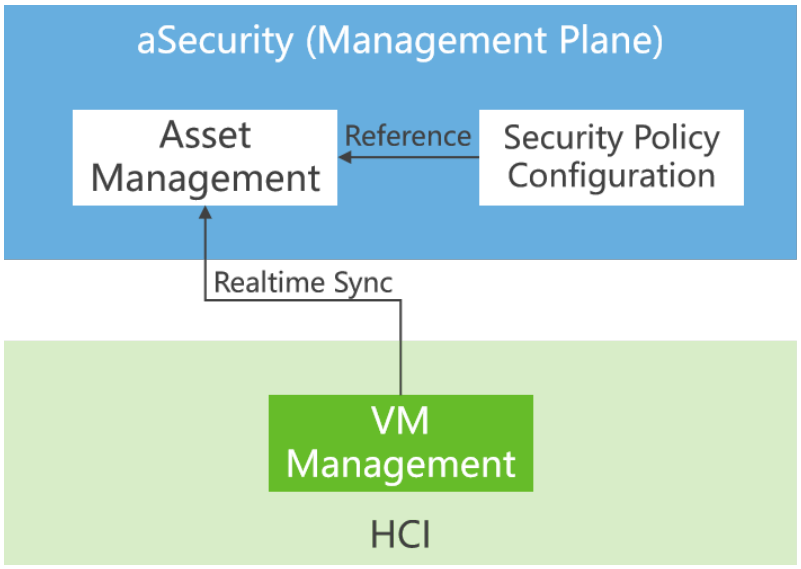


Figure 396: Cloud Asset Management

5.4.5.4 Adaptive Security Policies

The control plane of the cloud firewall is based on the SDN network control framework to schedule and deliver security policies and network policies in a unified manner.

When the network of a business VM changes, the virtual network configuration changes and is delivered again. The unified SDN controller performs scheduling calculations for security policies and re-delivers the correlated security policies to the new cloud firewall node, ensuring adaptive security policies.

Services in the cloud computing environment are highly flexible. The VMs can be rapidly deployed, destroyed, or migrated between hosts and clusters. Therefore, security policies need to adapt accordingly.

5.4.5.5 Improved Data Plane Performance

Traditional firewall virtualization solutions rely on network topology for traffic forwarding and need to forward all network traffic to the firewall VM, which can become a bottleneck. With the implementation of cloud service-oriented architecture, a traffic forwarding tool is introduced at the virtualized network layer. Traffic forwarding policies are automatically generated based on security policies. Only network traffic that matches the security policies is detected by the protection engine. This reduces the impact on user business while enhancing performance.

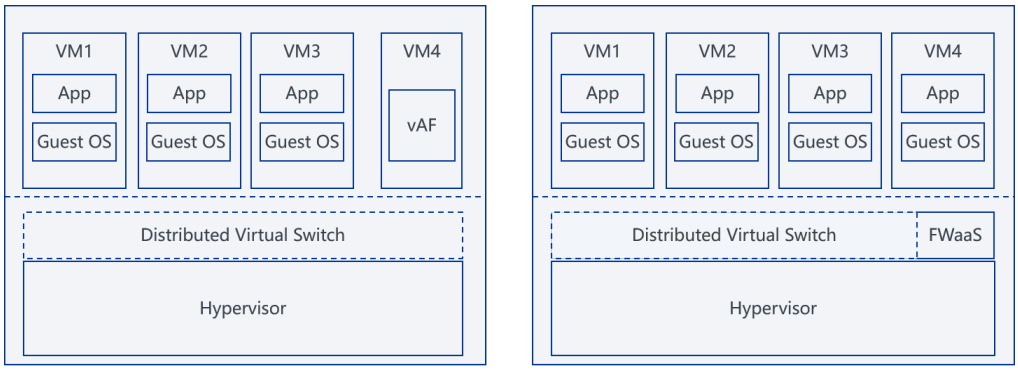


Figure 397: Comparison Between Traditional NFV Firewall and Distributed IPS Architecture

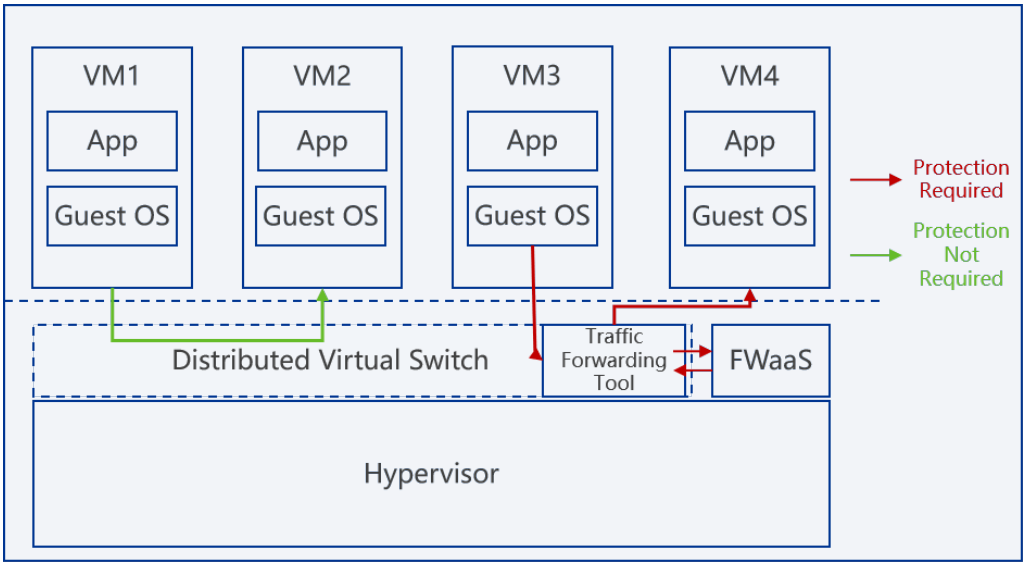


Figure 398: Secure Traffic Forwarding

NOTICE

FWaaS refers to firewall as a service engine.

The control plane delivers traffic forwarding rules to achieve precise forwarding of VM traffic requiring Layer 7 security detection. These rules support configuration options such as the source/destination IP address, source/destination port, application type, and VM ID. Users only need to configure security policies, and the SDN controller automatically generates corresponding traffic forwarding rules. This policy-based traffic forwarding technology enables protection on risky traffic even if it is from the same VM, effectively balancing security effects and business performance.

Service degradation is implemented in the traffic forwarding tool:

The traffic forwarding tool monitors the current traffic load and the performance load of the FWaaS engine in real time. If the traffic exceeds the engine load, the service degradation mechanism triggers a reduction in traffic, ensuring the availability of user business. Additionally, an alert is generated to notify the user of FWaaS engine scaling.

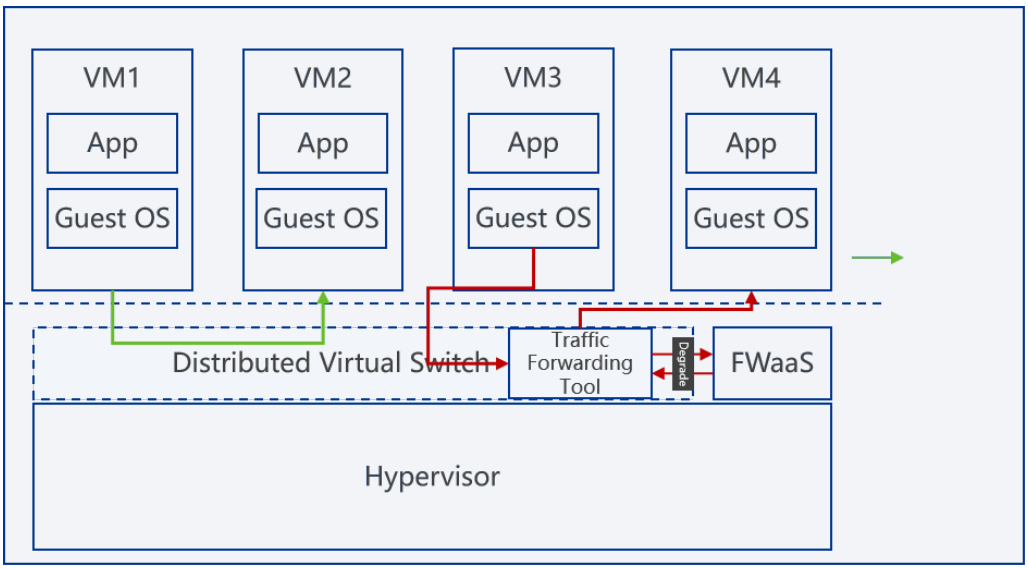


Figure 399: Service Degradation

In traditional VM deployment, the network data plane on the host needs to

transmit traffic to the guest through multiple layers of network protocol stacks. The NFV VM typically undergoes multi-layer processing involving NICs, kernel drivers, and application layer engines, resulting in poor performance. The traffic forwarding tool directly injects traffic into the FWaaS engine on the running host, with both the network data plane and the FWaaS engine operating in userspace. This approach significantly reduces the loss caused by network packet transmission.

5.4.5.6 HA

The cloud firewall service instance automatically scales up based on the configuration to meet the performance requirements of different scenarios. In the case of multiple instances, each instance handles business traffic through load balancing algorithms. If one instance fails, its traffic is redistributed to other instances, ensuring HA.

5.4.5.7 Intelligent Security Engine

The cloud firewall has a built-in security engine with the following features:

1. Continuously refined deep packet inspection (DPI) engine that combines protocol analysis, decoding, machine learning, and regular expression engine matching to detect vulnerability attacks, vulnerability variants, and defense evasion.
2. A library of over 4,000 vulnerability features and a library of over 3,000 web application threat features, capable of comprehensively identifying application-layer and content-level individual threats.
3. Security expert team that regularly provides the threat feature library to ensure timely defense.

5.4.6 vNGAF

Network security edges refer to the edge between the cloud platform and the internet or other external areas (cloud edge) and the edge between internal application systems within the cloud platform (intra-cloud). Deployment of

vNGAF at the cloud edge enables network attack protection, intrusion defense, and application security protection.

5.4.6.1 IPS

IPS focuses on the core concept of "precise identification and effective blocking." It promptly and accurately detects various illegal intrusion activities through in-depth traffic analysis and performs real-time precise blocking to actively and efficiently protect users' network security. Compared to traditional intrusion prevention systems, IPS emphasizes precise identification through multidimensional detection technologies, including AI and sandboxes, and enhances management and operation through simple O&M.

The gray correlation analysis engine of IPS supports over 7,000 attack features, capable of detecting common viruses, worms, backdoors, Trojans, botnet attacks, buffer overflow attacks, and vulnerability exploits. It can block mainstream advanced evasion attacks, detect and defend against abnormal traffic-based attacks, including various flood attacks, and provide the user-defined attack feature code function to specify the comparison content from the network layer to the application layer. In addition, it also provides virtual SPs to ensure the normal operation of network security for users who have not fixed vulnerabilities in a timely manner.

With the increasing number of attacks targeting the application layer, traditional network attack defense technologies are no longer sufficient to meet today's complex security requirements. Sangfor has employed over 10 years of research achievements in the application layer to develop IPS, which can identify and control applications, even encrypted data streams. It can accurately identify over 5,000 applications, including IM applications, P2P download applications, file transfer applications, gaming software, streaming media, and common mobile applications. By utilizing P2P intelligent flow control technology, it accurately identifies tunnels and encrypted P2P applications, ultimately achieving security management and control of applications.

5.4.6.2 WAF

WAF focuses on enhancing the security of websites and web application systems at the application layer. It addresses the limitations of traditional security products like network firewalls and IPS, which often fail to provide deep protection at this critical layer. Sangfor Cloud seamlessly integrates WAF, ensuring that incoming traffic from the access console passes through the WAF device for threat detection by the security detection engine. In this way, the top 10 web security threats identified by OWASP can be effectively defended, and the impact of malicious attackers on web businesses can be quickly mitigated, guaranteeing secure and reliable delivery of web applications.

It provides protection against:

SQL injection attacks: SQL injection attacks occur when web applications fail to validate user input data, allowing attackers to retrieve desired information from databases by submitting a piece of database query code. WAF of Sangfor Cloud filters out SQL injection-related key information using efficient URL filtering technology, effectively preventing such attacks on website servers.

1. Cross-site scripting (XSS): XSS attacks are a type of injection in which malicious HTML code is injected into web pages. WAF can accurately filter out malicious code of XSS attacks using advanced packet regular expression matching principles, ensuring the security of users' web servers.
2. Cross-site request forgery (CSRF): CSRF attacks are similar to XSS attacks but better exploit XSS vulnerabilities to execute complex JavaScript code on the user's browser, allowing hackers to hijack browser sessions and interact with script code running in the user's browser to perform malicious actions with the user's privileges. WAF effectively filters out CSRF attack code through advanced packet regular expression matching principles, preventing web systems from CSRF attacks.
3. Proactive protection: WAF provides proactive protection capabilities by automatically learning the variable types and lengths of URL requests received by protected hosts based on predefined thresholds. Once the

learning process is completed, it is able to defend against various attacks with code variations. Additionally, users can customize parameter rules to match valid URL parameters more precisely, enhancing attack recognition capabilities.

4. Application information hiding: WAF efficiently conceals feedback information from major servers such as web servers, FTP servers, and SMTP servers, preventing hackers from launching targeted attacks using server feedback information.

- **HTTP error page hiding**: Error pages of the web server are shielded, preventing the disclosure of web server version information, database version information, and the absolute path of the website. The return pages can be customized.
- **HTTP(S) response header hiding**: Specific field information in HTTP(S) response headers can be shielded.
- **FTP information hiding**: The FTP server information returned through normal FTP commands can be concealed, preventing hackers from using FTP software version information to launch targeted vulnerability attacks.

5. URL protection: Web application systems often include system administrator management UIs for remote maintenance. However, this convenience can be exploited by hackers to intrude into application systems. WAF provides restricted URL protection, enabling users to selectively expose specific URLs to prevent excessive exposure of application system data to the Internet.
6. HTTP exception detection: WAF analyzes the content of HTTP protocols and performs exception detection, allowing users to define allowed HTTP header request methods for their web business systems. This effectively filters out invalid request information.
7. File upload filtering: Due to the lack of comprehensive security controls during the development of web applications, uploaded files on web servers may be infected with viruses and Trojans and become tools for hackers to launch attacks. WAF prevents files with security risks from being uploaded to the server by strictly controlling the file types and

checking the file headers for their feature codes. It ensures the security of uploaded files through virus protection and plugin filtering, safeguarding web servers.

8. User login permission protection: For sensitive pages or application systems such as administrator login pages, WAF offers SMS authentication for URL login, preventing hackers from gaining access through repeated password login attempts, thereby enhancing access security.
9. Buffer overflow detection: Buffer overflow attacks exploit buffer overflow vulnerabilities to execute unauthorized instructions and gain system privileges to perform unauthorized operations. WAF defends against such attacks by detecting the length of URLs, POST entities, and HTTP headers.

5.4.7 aTrust

5.4.7.1 Overview

Sangfor, as one of the pioneering Chinese enterprises in zero trust applications, is dedicated to researching and developing cutting-edge zero trust security solutions. Sangfor aTrust is an innovative security product that leverages Sangfor’s extensive practical experience and technical expertise in remote office environments. By utilizing core technologies like "traffic identity" and "dynamic adaptive access control," aTrust provides users with a secure, convenient, and high-speed access experience.

aTrust is dedicated to providing government and enterprise clients with essential product components and comprehensive solutions for establishing a zero trust security system. It caters to the security access requirements of clients in diverse business scenarios, including intranet/Internet access, multi-branch connection, cloud migration, and mobile office, facilitating the swift migration of government and enterprise businesses to a zero trust security architecture. Key technologies and features:

1. Exposure reduction

It uses Sangfor's proprietary third-generation SPA single packet authorization technology (TCP+UDP) to effectively minimize service exposure. It addresses the issue of UDP port knocking in SNAT environments and introduces the innovative "one-person-one-code" management mode to combat security code loss and misuse. These measures significantly enhance user access security.

2. Adaptive authentication

Through dynamic adaptive authentication based on access context analysis, it intelligently determines the appropriate authentication level and type for users. When abnormal user identities or access environments are detected, enhanced authentication is automatically triggered to strike a balance between security and user experience.

3. Dynamic permission adjustment

Scenario-oriented dynamic permission adjustment continuously monitors the environment, behavior, and identity of users in real time during access. When a deviation from the security baseline is identified, it promptly adjusts the user's access permissions to minimize service exposure and mitigate security risks. By employing techniques such as secondary authentication and security alerts, it ensures fine-grained control in the access process, departing from the binary approach of coarse management.

5.4.7.2 Architecture

Sangfor aTrust follows the NIST ZTA zero trust architecture standard. It combines the software-defined perimeter (SDP) and unified endpoint management (UEM) technologies to provide enterprise clients with a comprehensive hybrid office solution that includes secure access and data leakage prevention.

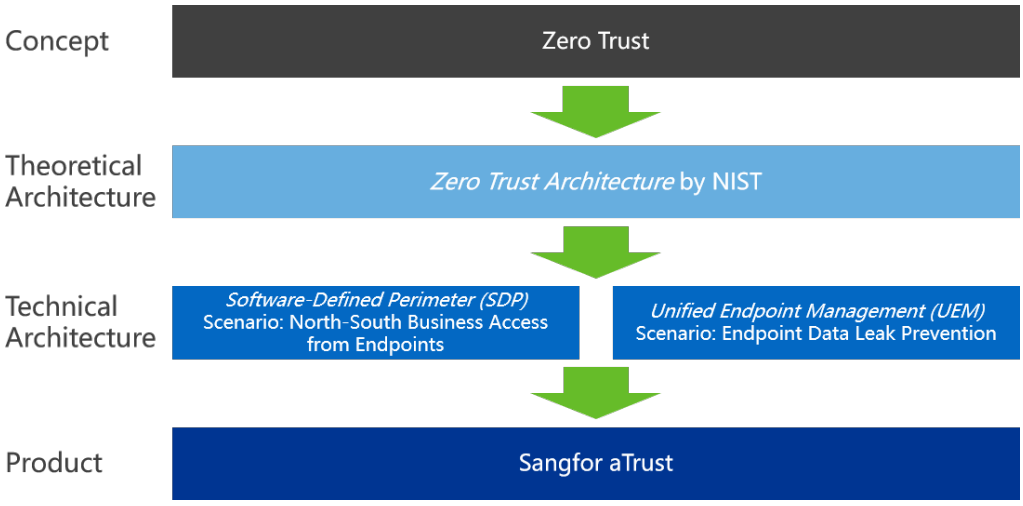


Figure 400: Concept of Sangfor aTrust

The following figure presents the overall architecture of aTrust.

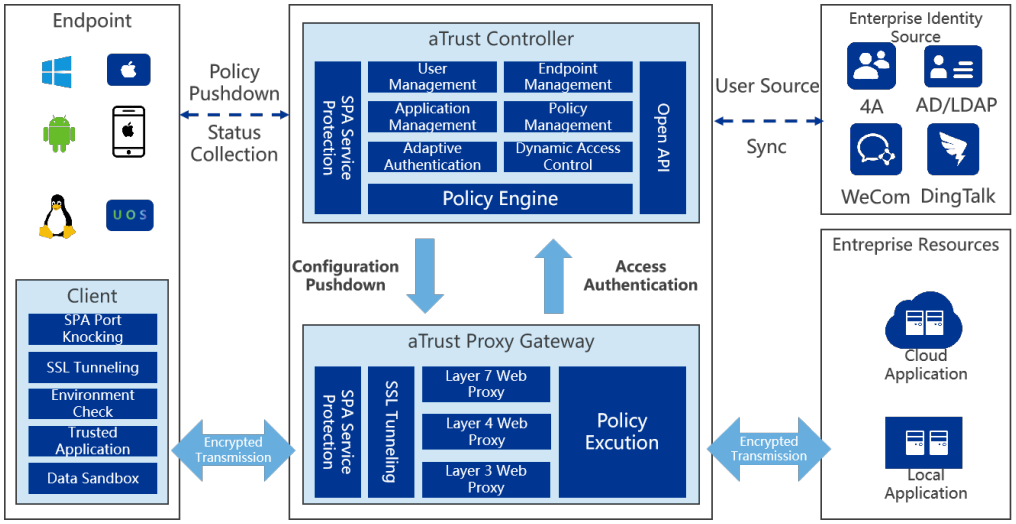


Figure 401: Architecture of Sangfor aTrust

The aTrust client currently supports all mainstream OSes, including Windows, macOS, UOS, Kylin, Ubuntu, Android, iOS, and HarmonyOS. It offers features such as SPA knocking, SSL tunnel access, endpoint environment detection, trusted application identification, and data security sandbox.

Following the design principle of separating the control plane and the data plane on the server side, aTrust consists of two core components: Zero Trust Controller and Zero Trust Proxy Gateway.

Zero Trust Controller serves as the control plane of the entire aTrust solution.

It functions as the policy decision point (PDP) in the NIST ZTA and the controller in the SDP, responsible for making final access authorization decisions. It provides management and control functions such as user management, endpoint management, adaptive authentication, dynamic access control, and policy management. Additionally, it offers open capabilities for integration and synchronization with existing enterprise identity infrastructure (such as 4A and PKI).

Zero Trust Proxy Gateway serves as the data plane of the aTrust solution. It functions as the policy enforcement point (PEP) in the NIST ZTA and the accept host (AH) in the SDP, responsible for executing the authorization policies from the controller, forwarding authorized access requests, and intercepting unauthorized access requests. It is typically deployed in a logical cascading manner at the uplink of protected resources, reducing resource exposure. External users and endpoints cannot directly access the protected resources but need to go through strict identity authentication and authorization decisions by Controller. The access is then proxied by Proxy Gateway. Zero Trust Proxy Gateway supports various access protocol proxies, including Layer 7 web proxy, Layer 4 TCP proxy, and Layer 3 IP proxy.

5.4.7.3 Key Functional Modules

aTrust follows the NIST ZTA standard and the concept of separation of the control plane and the data plane. It has three main functional modules: the aTrust client, the aTrust controller, and the aTrust proxy gateway.

For detailed technical information about aTrust, refer to the ***Sangfor aTrust White Paper***.

For configuration details, please refer to the "[aTrust] Virtualization/NFV" section.

5.4.8 vAC

With the development of Internet technologies, organizations' business models and employees' work patterns and habits are constantly changing.

1. Online business: Organizations are establishing more online business platforms to conduct business over the Internet.
2. Communication bridge: Internal employees increasingly rely on the Internet to communicate and collaborate with external partners, improving work efficiency, obtaining information and knowledge, and maintaining interpersonal connections.
3. Mobile Internet: The consumerization trend of mobile Internet is gradually affecting the internal IT systems of organizations, and employees prefer to use WLAN and mobile devices for work.
4. Network access: The variety of access devices is increasing. For the intranets of governments and enterprises, employees use various devices such as cameras, code scanners, printers, and other IoT devices.
5. Invisibility of business-side access: Many traditional Internet gateways are equipped with a large number of security devices, but abnormal access to and downloads from servers are not audited or analyzed.
6. New regulatory requirements: Take Chinese law as an example. The Cybersecurity Law that was officially issued on June 1, 2017 explicitly requires organizations to perform audits on their Internet access activities and maintain Internet access logs for a minimum of six months.

These changes have introduced new risks to enterprises:

1. Increased intranet security risks due to unidentified identities and devices accessing the network

- Unauthorized access: External individuals can easily connect to the intranet using an Ethernet cable, posing risks to various business systems.
- Non-compliant devices accessing the network: Non-compliant devices, such as those without antivirus software, using non-compliant OSes, or with weak passwords, pose significant threats to the intranet environment, potentially leading to the rapid spread of viruses within the network.

2. Hidden risks behind seemingly normal Internet access activities

- Bandwidth occupancy: Employees using applications (such as P2P applications and streaming media) during work hours occupy great

bandwidths, which severely affect email sending, data downloading, video conferences, and other activities critical to the company's core business.

- External leakage of confidential information: Sensitive data/files, including personal privacy information, organizational confidential information, government documents, and official files, may be sent to external objects unintentionally or casually without monitoring, alerts, or compliance auditing. The operator is unknown.
- Cybercrime: The organization's network may be used to spread rumors, launch personal attacks, and indiscriminately distribute reactionary, gambling, and pornographic content, resulting in significant harm to the company and organization and potential legal consequences.
- Challenges in Internet access supervision: The workplace becomes an unregulated Internet café, leading to decreased work efficiency. Advanced encryption and proxy technologies complicate the control over unauthorized content. Additionally, an application has both "sides," making it challenging to determine whether to implement control measures.

3. Abnormal business access is hard to detect, and it is difficult to manage account misuse, resulting in tracing issues.

- Invisible business behavior: When employees access the business system, comprehensive auditing operations make it difficult to determine the responsibility in cases of operational errors or data leaks.
- Difficulties in detecting abnormal behavior: When employees access the business system, some malicious or unintentional actions may harm the system, such as excessive data download, crawling of all data, and execution of sensitive operations like deletion and clearing.

Sangfor Internet Access Gateway (IAG) effectively addresses these risks and provides control over endpoint access, user Internet access, and endpoint behavior.

Endpoint access control:

Sangfor IAG accurately identifies connected endpoint assets, offers various

identity authentication methods (including 802.1x and Portal), conducts security baseline checks, and controls peripheral device permissions. This helps users establish secure endpoint access standards to reduce security risks and data leakage.

Internet access control:

For endpoints connected to the network, Sangfor IAG utilizes databases containing vast URLs and application rules to achieve precise application control. It ensures comprehensive compliance auditing and supports dynamic flow control and application traffic forwarding to allocate bandwidth properly. This helps users maintain Internet access compliance, enhance user experience, and prevent unauthorized activities.

Endpoint behavior control:

Lightweight desktop control features include port control, application network connection control, software installation management, screen watermarking, external connection control, and peripheral device control. They enable administrators to effectively manage and control endpoint behavior, thereby improving system security. Sangfor IAG integrates network and endpoint control to promote more standardized, efficient, and secure office operations.

For detailed technical information about vAC, refer to the ***Sangfor IAG Technical White Paper***.

For configuration details, please refer to the "[IAG] Virtualization/NFV" section.

5.4.9 vSSL VPN

The vSSL VPN component aims to create value for customers by leveraging innovative solutions tailored to their specific business needs. Based on the comprehensive understanding of the customer's business situation, it effectively resolves challenges encountered during the process of Internet-based transformation. In addition to conventional SSL VPN solutions like mobile office solutions and multi-party access permission allocation schemes, vSSL VPN also addresses security issues of customers' current critical systems by leveraging SSL security features. It ensures the security and confidentiality

of core customer data through SSL encryption and logical isolation.

5.4.10 vAD

The load balancing component offers a server load balancing solution for application release. With performance optimization, single-edge acceleration, and multiple intelligent management schemes, it monitors the status of each cloud server in real time and delivers user access requests to appropriate servers based on predefined rules. This optimizes data flow allocation and maximizes cloud server utilization. It expands the processing capacity of the application system, improves stability, enhances user experience, and reduces IT investment costs. Traditional load balancing accomplishes this by expanding network device and server bandwidth and throughput to ensure stable network data processing.

Sangfor vAD emphasizes stability and incorporates intelligence and optimization features, helping users overcome the challenges of deploying and delivering services in complex application environments. By effectively deploying application delivery devices, users can improve the performance and security of business applications, enhance data center infrastructure efficiency, and adapt to the future trend of deploying virtualized data centers.

Sangfor vAD is a professional device that offers a comprehensive solution for application delivery. It provides features such as multi-data center load balancing, multi-link load balancing, and server load balancing. With performance optimization, single-edge acceleration, and multiple intelligent management schemes, it monitors the status of data centers, links, and servers in real time and delivers user access requests to appropriate data centers, links, and servers based on predefined rules. This optimizes data flow allocation and maximizes utilization of data centers, links, and servers. It expands the processing capacity of the application system, improves stability, enhances user experience, and reduces IT investment costs.

5.4.11 Data Security

Data security encompasses two primary aspects. The first aspect involves

safeguarding the data itself through encryption algorithms, ensuring data confidentiality, integrity, and mutual authentication. The second aspect focuses on protecting data through advanced information storage methods, such as distributed storage featuring high availability and cross-site DR. Addressing data security involves managing the entire data lifecycle, from creation to the point where it no longer holds commercial value or needs to be deleted. Effective data management throughout the lifecycle subject to necessary control measures is crucial to prevent unauthorized internal access.

The data lifecycle, also known as the information lifecycle, typically comprises six stages: creation, protection, access, migration, archiving, and deallocation/destruction. The figure below illustrates the data activity levels and relevant protection measures for each stage. Data security must be upheld at every stage, with measures like physical security, network security, host security, and application security working together to ensure data protection.

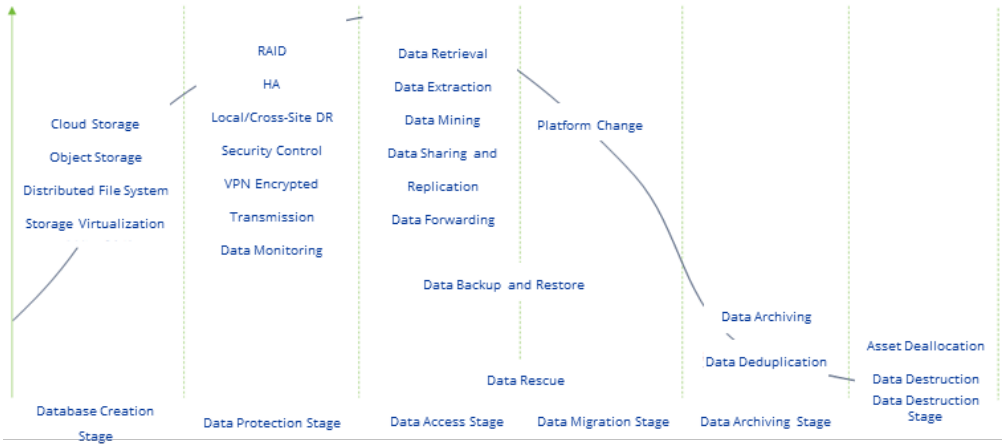


Figure 402: Data Lifecycle

5.4.11.1 Data Encryption

When users need to encrypt and protect data stored in the cloud for security or regulatory compliance, they can utilize the data encryption feature of Sangfor HCI. By integrating with the key management service (KMS) of SCP, users do not need to build, maintain, and protect their own key management infrastructure and encryption implementation mechanism. The data encryption feature of Sangfor HCI ensures the privacy and autonomy of cloud

data. It supports encryption, decryption, and integrity verification of sensitive data in both HCI and SCP. Additionally, it enables tenant hardware-level key management and the encryption of VM and NFV images. The SCP platform supports integration with third-party password service platforms through OpenAPI. Users can use a cryptographic card to provide high-performance data encryption and decryption services.

The cryptographic card is a hardware security module (HSM) specifically designed to store and manage keys, certificates, and other sensitive data. HSM employs physical features and encryption algorithms, such as the anti-counterfeiting identifier, random number generator, symmetric encryption algorithm, and asymmetric encryption algorithm, to ensure the security of keys and data. VMs can use HSM to safeguard sensitive data, such as encryption keys and digital certificates. Virtualization management software (such as VMware vSphere, KVM, and Hyper-V) typically provides interfaces and tools to map HSM to VMs, thereby achieving VM security protection. Cryptographic cards offer two modes of service in HCI:

- Passthrough mode

In this mode, HSM is directly connected to the PCIe bus of the host, allowing the host to directly access the hardware resources of HSM, such as the encryption engine and random number generator. Passthrough mode ensures optimal performance and security by bypassing the virtualization layer, resulting in reduced latency and security risks.

- VF mode

In VF mode, HSM is virtualized into multiple virtual devices, each of which can be assigned to a VM. SR-IOV-VF enhances the performance and efficiency of VMs by directly mapping the data traffic of VMs to the virtual devices of HSM, thereby eliminating the need for virtualization layer involvement.

5.4.11.2 Commercial Cryptography

Passwords are widely used in various aspects of people's lives and are essential to information security. However, the importance of password security is often overlooked. Without proper network security, national security

is at risk. Cryptography is a crucial technology for safeguarding national security and fundamental interests in the realm of network security. However, there are several issues with passwords in China:

1. Limited adoption

Many networks and information systems lack proper password protection or use inadequate password technologies, leaving them vulnerable to security breaches. The use of commercial passwords is particularly low, posing significant risks.

2. Non-standard application

In some regions and departments, there is a lack of emphasis on information security in favor of informatization. This leads to non-standard and incorrect cryptography practices in networks and information systems, resulting in security risks in key management and password system O&M.

3. Insecure application

Many existing networks and information systems still rely on outdated and insecure password algorithms, such as RSA1024, SHA-1, MD5, and DES, as well as insecure password services based on these algorithms.

Commercial cryptography products use cryptography technology to provide encrypted protection or security authentication for information that does not involve state secrets. They are essential tools for protecting the rights and interests of individuals, organizations, and enterprises in the information era. Commercial cryptography is classified as a state secret, and it is the responsibility of all entities and individuals to protect its secrecy.

To promote the application of commercial cryptography, the Chinese government has issued various laws, regulations, and rules.

5.4.11.2.1 Overview

Sangfor HCI and SCP have undergone commercial cryptography transformation to comply with national regulations for commercial password applications. This transformation meets the requirements for commercial passwords in the Multi-level Protection of Information Security Scheme (MLPS);

sensitive data of HCI and SCP are encrypted using the SM4 algorithm, with the keys stored in hardware to ensure that the sensitive data cannot be decrypted if stolen. Additionally, the cryptographic cloud server uses the SM2/3/4 algorithms to provide services such as data encryption/decryption and data integrity verification.

The commercial cryptography transformation follows the interface specification outlined in GM/T 0018-2012 Interface specifications of cryptography device application. This specification ensures data protection by covering the confidentiality and integrity of both data storage and transmission. For data storage, it is necessary to encrypt and add integrity signatures to the data using a cryptographic cloud server that adheres to the above-mentioned specification. For data transmission, confidentiality and integrity are maintained to ensure the secure transmission of data over the network.

5.4.11.2.2 Key Technologies

Both SCP and HCI offer independent commercial cryptography modules to achieve high-performance encryption, decryption, and signature features. These modules ensure the high availability of cryptographic cloud servers in clusters. By utilizing hook technology, which intercepts the original execution flow of a program and adds additional processing logic, the encryption, decryption, and signature processes remain transparent to business operations when the commercial cryptography mode is enabled.

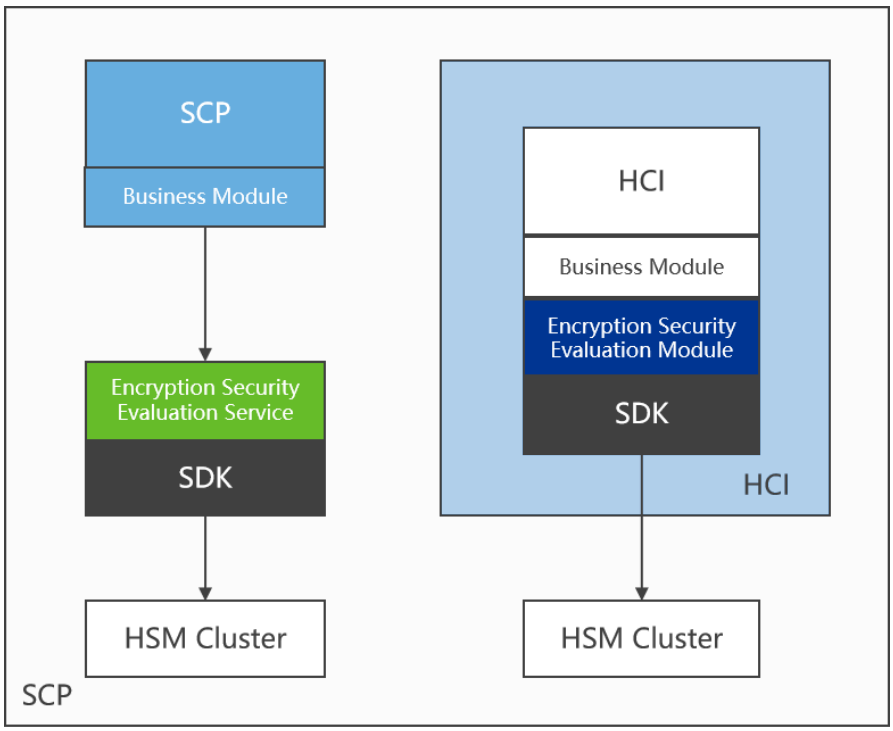


Figure 403: Built-In Password Evaluation Module of SCP and HCI

Main technical points of commercial cryptography:

1. Password evaluation mode: When entering the commercial cryptography mode, the platform activates data integrity and confidentiality protection mechanisms based on the Chinese cryptographic algorithm.
2. Confidentiality protection during data storage: The confidentiality of important data is protected using a signature key. Instead of storing the data in plaintext, it is stored in ciphertext using the SM4 algorithm. Encryption occurs during storage, and decryption takes place during retrieval. For data stored in the MySQL database, encryption is applied before writing the data in ciphertext format, and decryption is performed before returning the data during retrieval.
3. Integrity protection during data storage: Signature processes are implemented for data that may change. Signatures are calculated for different scenarios, and signature data is actively generated. When the data source changes, the updated signature value is immediately called after updating the data in the business process. By combining with

Sangfor's proprietary verification tools, the entire data integrity protection process is completed.

Main evaluation process:

1. Evaluation mode enabling/disabling: When the commercial cryptography mode is enabled, the platform (SCP/HCI) encrypts the existing confidential data using the Chinese cryptography algorithm to prevent the theft of confidential data. The platform also signs the existing integrity data and saves the signature value for verification, preventing data tampering. If the commercial cryptography mode is disabled, the existing confidential data is encrypted, and the signature of the integrity data is cleared.
2. Encryption/decryption of confidential data in business processes: When the commercial cryptography mode is enabled, any addition or modification of confidential data is encrypted using the commercial SM4 algorithm and stored in the MySQL database. During retrieval, the confidential data is decrypted and returned in plaintext.
3. Signature of integrity data in business processes: When the commercial cryptography mode is enabled, the generation or modification of integrity data triggers the signature operation. The commercial cryptography service signs the integrity data using the commercial SM4 algorithm and Digest/Hash and then stores the signature in the signature table. A verification tool is provided to verify the signature of the integrity data, detecting any tampered data.

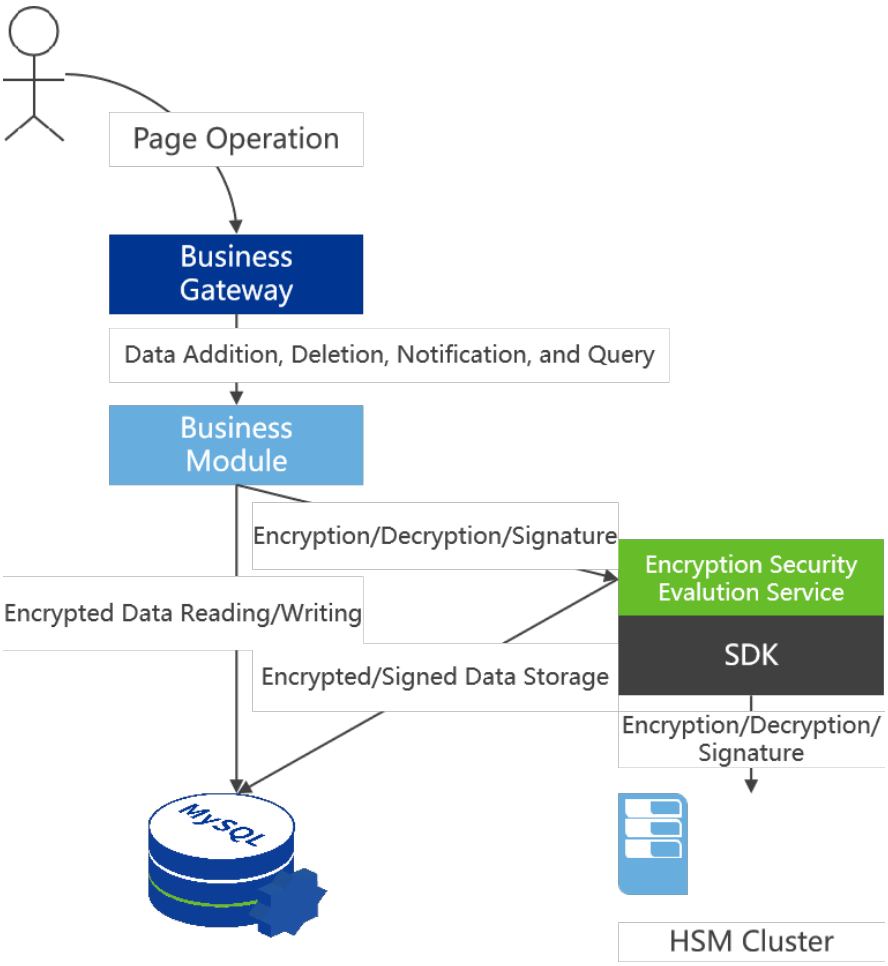


Figure 404: Evaluation Process

5.4.11.2.3 Confidentiality and Integrity of Data Transmission

The confidentiality of data transmission is ensured as follows:

1. Communication data integrity and confidentiality: Deploy a standard IPSec/SSL VPN to protect the confidentiality and integrity of communication data, and authenticate communication entities.
2. Access control information integrity: The network boundary control mechanism is used within IPSec/SSL VPN; signature verification servers, HSM, and other cryptographic products are deployed to perform MAC calculation or signing on other access control information, such as access control lists and then save the signature to ensure its integrity.

3. Secure access authentication: For devices accessing the intranet from outside the network boundary, IPSec/SSL VPN or secure authentication gateways are used for authentication and control. For devices connecting to the intranet from within the network boundary, identity authentication devices (such as identity authentication systems) are used for access control.

5.4.11.2.4 Commercial Cryptography Application Topology

The figure below shows the topology for commercial cryptography transformation in a data center.

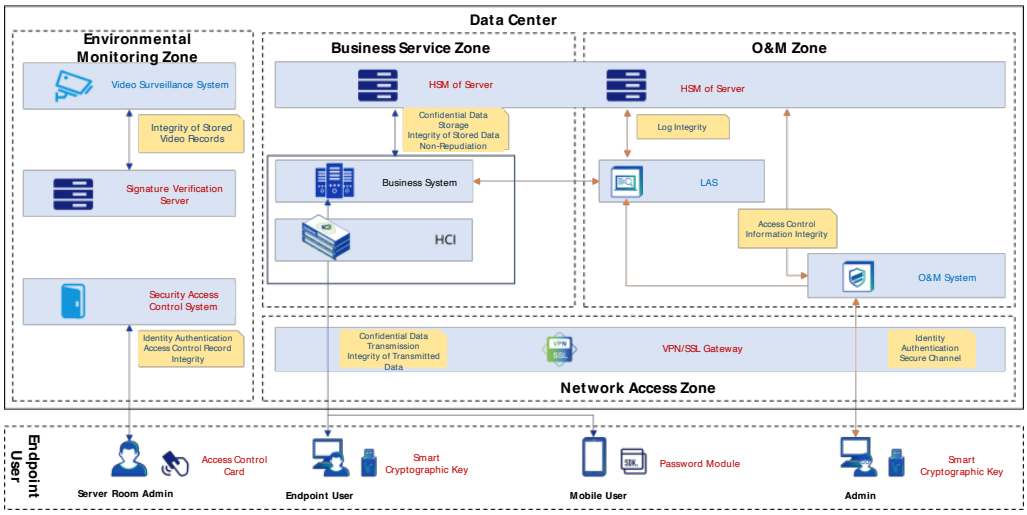


Figure 405: Data Center Topology

For data storage, the data is encrypted with a cryptographic server and applied with an integrity signature to ensure its confidentiality and integrity. For data transmission, a standard IPSec/SSL VPN is deployed to ensure communication data confidentiality and integrity by working together with the network boundary control mechanism within the IPSec/SSL VPN. For other access control information such as access control lists, signature verification servers, cryptographic servers, and other cryptographic products are deployed to perform MAC calculation or signing and then save the signature to ensure information integrity.

5.4.12 Security O&M and Operation

Security O&M involves the routine O&M of IT systems from a security perspective. Traditional O&M focuses on ensuring the normal functioning of systems, while security O&M analyzes, detects, and resolves existing and potential security issues. This establishes a closed-loop security management system that covers protection, monitoring, and resolution, ultimately ensuring the overall normal operation of the enterprise's IT system.

The concept of security operation is broader. In a narrow sense, it revolves around assets, with security event management as a key process. It relies on a security operation center (SOC) to establish a real-time asset risk model and a centralized security management system for event analysis, risk analysis, alert management, and emergency response. In a broader sense, security operation is a complex system that combines technology, processes, and people. It involves effective analysis of data generated by existing security products, tools, and services, continuous value output, and resolution of security risks, achieving the ultimate goal of security.

5.4.12.1 O&M Audit (Bastion Host) (vOSM)

vOSM is a crucial component for MLPS compliance. It provides users with integrated O&M security audit services, including account management, identity authentication, single sign-on, resource authorization, access control, and operation audit. It effectively audits the O&M processes of servers, network devices, security devices, and databases, transforming O&M auditing from event-based to content-based. Using preventive, control, and audit measures implemented through the internal control management platform, it comprehensively addresses O&M security issues.

5.4.12.2 Database Audit (vDAS)

vDAS is a compliance management system that performs detailed auditing of database operations in a business environment. It analyzes, records, and reports on the behavior of business personnel accessing the system, assisting

users with pre-planning, real-time monitoring, response to violations, compliance reporting, and incident tracking to ensure the normal operation of core assets. vDAS can monitor the real-time operation flow of the database server, intelligently analyze various operations, and provide log reports for post-analysis and forensics.

5.4.12.3 Vulnerability Scanning and Baseline Check (Sangfor Host Security)

Sangfor Host Security helps users centrally collect, analyze, and manage security configurations for cloud assets. It is a crucial support for daily cloud information security work. As a unified security configuration check and management system, it accurately and quickly identifies and summarizes security configuration issues and vulnerabilities across different hosts, databases, middleware, networks, and security products. Its main features include:

1. Task formulation: Flexibly formulate security baseline check tasks of various types and cycles, allowing easy and quick setup of check objects and policies.
2. Collection and analysis: Comprehensively and centrally check and analyze local security configuration issues, vulnerabilities, weak passwords, and other issues, reducing redundant work caused by scattered device management.
3. Violation reporting: Provide comprehensive, detailed, and clear management and comparative analysis of scan reports.
4. System reinforcement: Provide detailed and practical system reinforcement solutions to guide users in resolving security issues.

5.4.12.4 Log Audit (SIP-Logger)

SIP-Logger continuously collects and aggregates log information from cloud tenant hosts, OS, and business systems in real time. It assists users in security analysis, compliance audit, and timely detection of security events and

violations.

5.4.12.5 Security Integration (aSI)

Sangfor HCI supports security integration with Cyber Command, enabling intelligent fixing of security events.

5.4.12.5.1 Cyber Command Overview

Cyber Command is a security detection product that utilizes AI and big data security detection technology. It enhances detection accuracy, identifies unknown threats, and takes action by integrating with various security devices. Through STA for hardware, it mirrors the traffic of core switches in data centers. In scenarios where Cyber Command functions as the local security center, it integrates with different security devices to form a comprehensive security protection system.

It relies primarily on STA for detection:

1. STA is an independent hardware device connected to core switches to forward the traffic to Cyber Command.
2. In addition to traffic forwarding, STA can also analyze traffic and provide the result to Cyber Command.

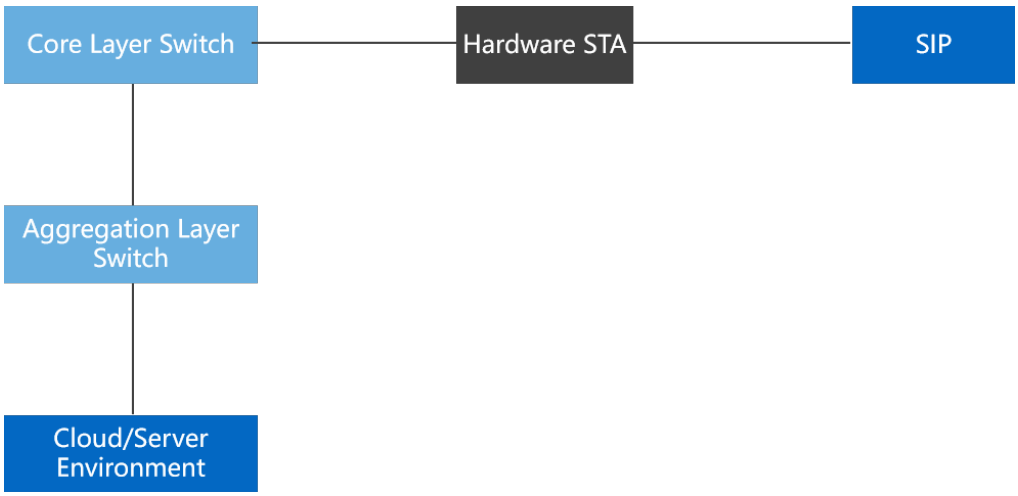


Figure 406: Cyber Command & STA Workflow

5.4.12.5.2 Integration of Cyber Command and HCI

Currently, Cyber Command has integrated with various security devices, including Endpoint Secure, Network Secure, and IAG. It collects security logs from these devices for analysis, statistics, and display and takes correlated fixing measures.

Cyber Command has strong capabilities for detecting and identifying security issues, while HCI has full operational control over business VMs. By combining the detection capabilities of security products with the fixing capabilities of HCI, a more effective and comprehensive security solution can be achieved.

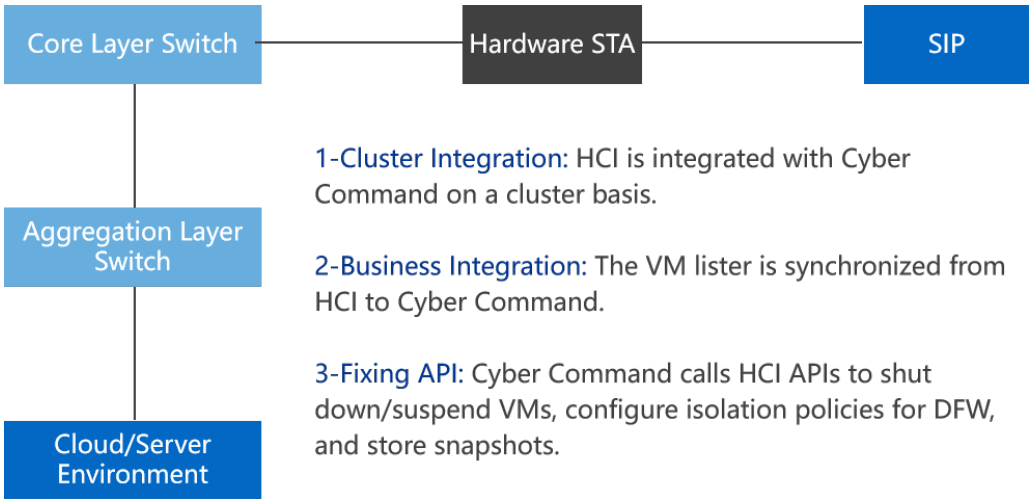


Figure 407: Cyber Command & STA & HCI Workflow

1. HCI is managed by Cyber Command as a cluster with the account and password provided.
2. HCI provides interfaces for synchronizing the VM list, shutting down/suspending VMs, configuring the distributed firewall, and taking storage snapshots. Therefore, the list and on/off status of VMs in the HCI cluster can be viewed in Cyber Command. The HCI cluster can automatically sync the VM list with Cyber Command at an interval of 30 seconds to 1 minute.
3. When a security event is detected in Cyber Command, it supports operations such as shutting down/suspending VMs, taking storage snapshots, and configuring distributed firewall security isolation

strategies for VMs. The processing mechanism based on the integration of Cyber Command and HCI is shown in the following figure.

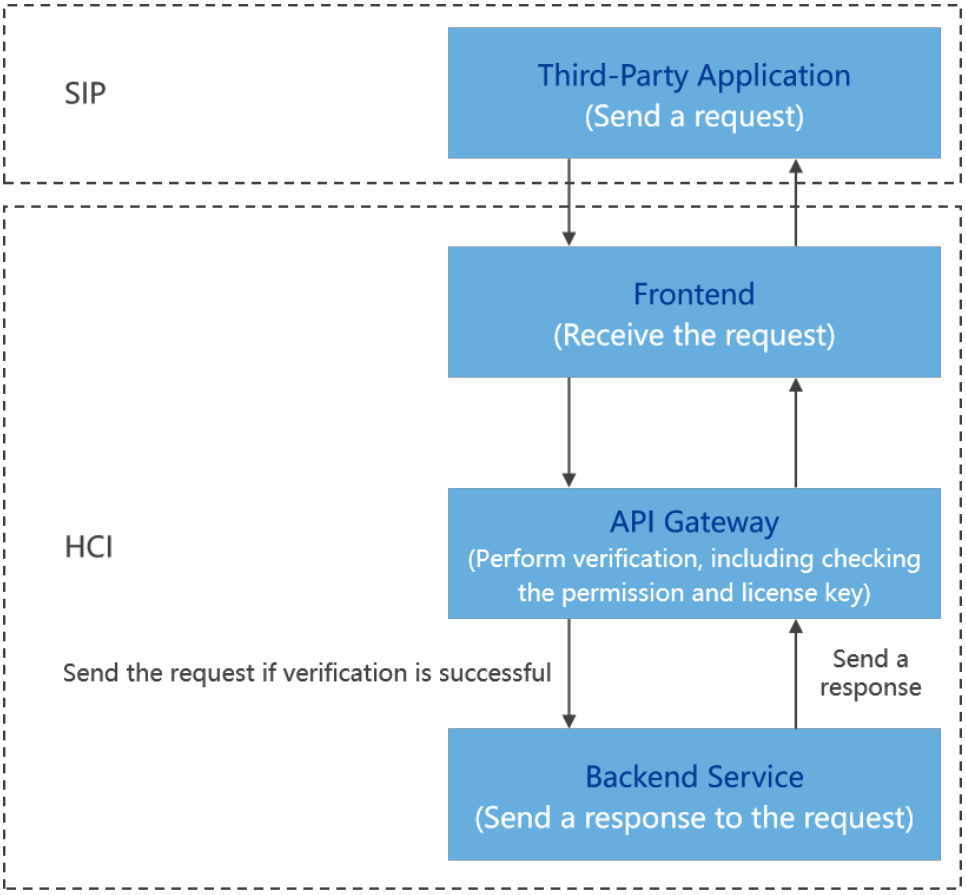


Figure 408: Processing Mechanism Based on Integration of Cyber Command and HCI

The integration between HCI and Cyber Command is based on north-south traffic. East-west traffic of VMs cannot be mirrored externally, so no effective protection is provided for east-west traffic.

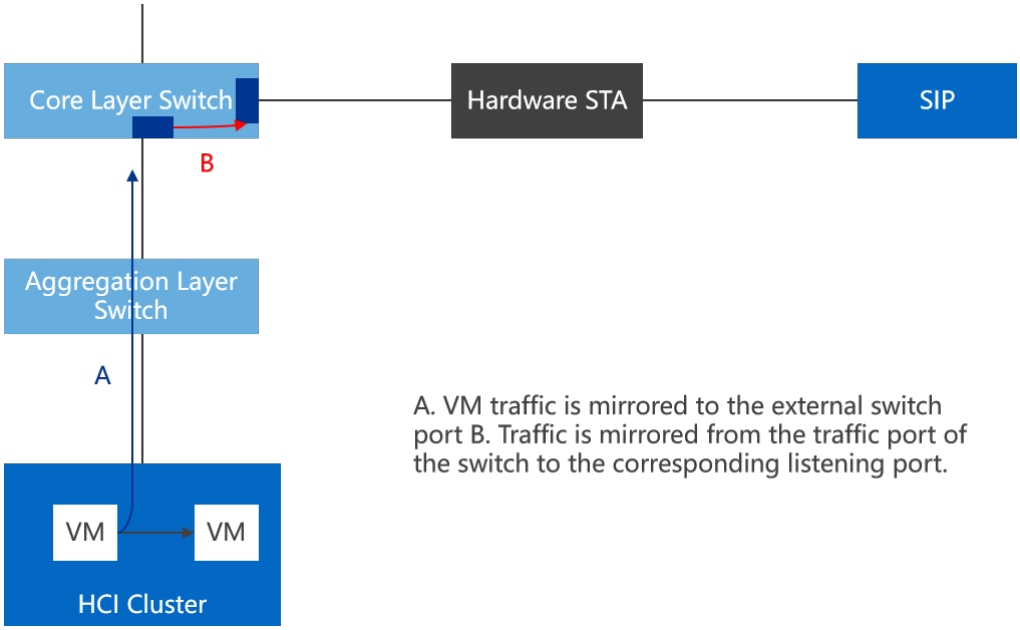


Figure 409: Traffic Acquisition Path of Cyber Command & STA & HCI

5.4.12.5.3 Scenarios

Scenario 1: When Cyber Command detects ransomware, it calls the storage snapshot API of HCI to provide fallback protection.

Processing logic:

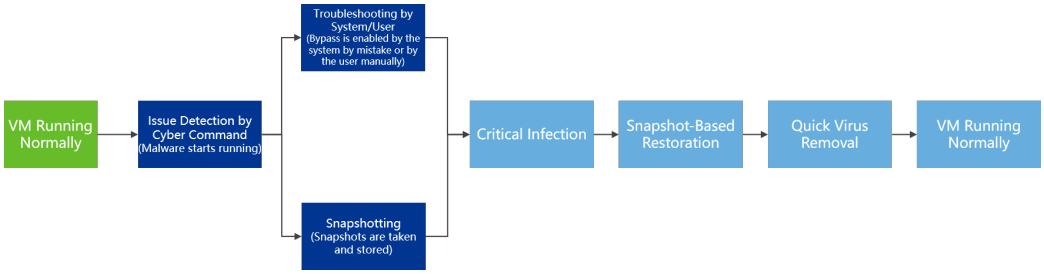


Figure 410: Virus Detection and Processing Process of Cyber Command & HCI

The virus needs time to perform corresponding operations. For ransomware, it typically takes several minutes to an hour to encrypt the disk. Once Cyber Command detects the virus, it promptly takes a snapshot for fallback protection.

Due to the impact on performance and storage space, there are restrictions on snapshot strategies. Only the most recent 15 days of snapshots are retained. A

maximum of two snapshots can be taken for each VM within 24 hours, and a maximum of 10 snapshots can be correlated in the cluster within 24 hours.

Precautions:

1. Cyber Command does not support snapshot rollback operations.
2. Storage snapshots are supported only when storage virtualization is enabled.
3. Storage snapshots are not supported for two-host storage virtualization.
4. Consistency snapshot groups are not supported.
5. When the version of a VM with existing snapshots is upgraded, the existing snapshots must be deleted before using storage snapshots.

Scenario 2: When Cyber Command detects VMs infected with viruses such as crypto mining viruses, it works together with HCI to shut down/suspend the VMs to prevent them from consuming the entire platform's performance.

Scenario 3: When Cyber Command detects VMs in a botnet, it automatically isolates infected VMs using the distributed firewall of HCI.

5.5 HCI Management

5.5.1 Overview

HCI extends beyond the combination of computing, storage, and networking components. It provides unified control and application of resources globally and manages resource pools via clusters. It offers robust security access, scalability, and management capabilities for upgrades and O&M at both global and cluster levels.

5.5.2 O&M Management

Sangfor HCI provides comprehensive monitoring and alert capabilities, covering underlying hardware configurations, the utilization of platform resources, and the operation states of upper-level businesses. It supports in-depth analysis of the entire operating environment through health check,

providing corresponding instructions and processing mechanisms. Additionally, it offers command-line instructions for users to conduct detailed status analysis.

5.5.2.1 Platform Monitoring and Detection

Sangfor HCI offers extensive platform monitoring and detection capabilities, including hardware health monitoring and platform resource monitoring. Users can monitor HCI businesses in a detailed and long-term manner through the monitoring center. The specific implementation is as follows:

5.5.2.1.1 Hardware Health Monitoring

HCI can be delivered in two modes: Integrated delivery of software and hardware, and software-only delivery (with third-party hardware). In both modes, HCI offers hardware-level reliability testing and assurance to prevent serious issues caused by hardware failures.

Hardware reliability monitoring covers the CPUs, memory, NICs, disks, storage, RAID, and other hardware. It promptly detects issues and offers recovery guidance and recommendations for exceptions. The detection results are presented uniformly, allowing users to eliminate risks based on alert information and prompts.

In addition, Sangfor HCI Appliance comes with a BMC diagnostic module, which enables fault diagnosis for key components such as CPUs, memory, disks, NICs, fans, temperature module, and power supply.

CPU monitoring:

- **CPU temperature monitoring**: HCI checks the temperature of each physical core of the CPU every minute. If the CPU temperature remains abnormal for a specified duration (10 minutes), an alert will be reported.
- **CPU clock speed monitoring**: HCI checks the CPU clock speed every hour at the backend. If the CPU clock speed drops, an alert will be reported.

Memory reliability:

ECC monitoring: The memory is monitored in real time using ECC technology.

Memory ECC faults include:

- **UEs**: Uncorrectable errors (UEs) can cause device crashes or reboots.
- **CEs**: Correctable errors (CEs) do not immediately impact the continued use of memory but may eventually become UEs.

Memory ECC issues are primarily caused by factors such as the quality level of memory module chips, the influence of the motherboard signal integrity (SI), environmental disturbances (such as environmental noise, high temperature, and PWM chip high-frequency interference), and overall heat dissipation.

With the reduction in chip processes by various manufacturers and the continuous increase in memory frequency, the ability of capacitors to store charge decreases, making leakage events more likely to occur. In recent years, memory ECC issues have become increasingly apparent. HCI performs statistical monitoring of UC-class uncorrectable ECC errors, reports alerts, and provides solutions to prevent incidents.

Memory throughput monitoring: The HCI platform offers detection of memory throughput. If the memory throughput significantly falls below the nominal value, an alert will be reported.

Disk monitoring:

Hot swapping: Sangfor HCI Appliance supports hot swapping. By default, the Just a Bunch of Disks (JBOD, that is, pass-through) mode is used, is used, in which aSAN ensures high data redundancy and reliability, supporting data reconstruction and balancing after hot swapping of disks. For more details, refer to section 5.2.5.4 "Data Self-Balancing."

The disks on a host undergo comprehensive monitoring to prevent faults and ensure high reliability.

1. Disk status monitoring: HCI continuously monitors the real-time status of disks and reports an alert immediately when any disk goes offline.
2. I/O error monitoring: HCI periodically analyzes Dmesg information to detect I/O errors and reports an alert immediately if any error is detected.

3. SSD lifetime monitoring: HCI periodically checks the lifetime of SSDs using the smartctl command. If the remaining lifetime of an SSD is below 10% of the total lifetime, an alert will be reported immediately.
4. HDD bad sector monitoring: HCI scans all physical disks for bad sectors based on user instructions using the smartctl command. An alert is reported immediately if any bad sector is detected. If the number of bad sectors is below 10, it is recommended the disks be replaced. If the number exceeds 10, the disk is marked as a subhealthy disk and downgraded, and the data is gradually migrated out.
5. I/O latency monitoring: HCI tests the latency of random 4 KiB I/O block reads on raw disks, with a 32-bit queue depth, using the FIO command as per user instructions. An alert is reported if the IOPS performance of a disk reaches a dangerous value. For example, if the IOPS value of a 7,200 RPM disk, 10,000 RPM disk, and 15,000 RPM disk are below 60, 100, and 140, respectively, an alert will be reported.
6. IOPS/throughput monitoring: HCI tests the latency of random 4 KiB I/O block reads on raw disks, with a 32-bit queue depth, using the FIO command as per user instructions. An alert is reported if the IOPS performance of a disk reaches a dangerous value. For example, if the IOPS value of a 7,200 RPM disk, 10,000 RPM disk, and 15,000 RPM disk are below 60, 100, and 140, respectively, an alert will be reported.

NIC monitoring:

1. Interface connection mode detection: HCI continuously checks the operational mode of interfaces for consistency with the negotiated mode, ensuring a proper network environment for business operations. Any inconsistency in modes will trigger an alert.
2. Interface deployment detection: HCI performs deployment detection on all interfaces to ensure the proper functioning of interfaces configured for specific purposes. This prevents faults such as port disconnection or unplugged cables. Improper deployment of interfaces will trigger an alert.
3. Packet loss detection on interfaces: To ensure the stability of the business network, HCI reads NIC information and monitors packet loss. If the

packet loss rate reaches a critical level, an alert will be triggered. For example, if the packet loss rate of an interface exceeds 0.1% in 10 minutes, an alert will be reported.

4. Interface transfer rate detection: HCI checks the transfer rate of interfaces to meet performance requirements. An alert will be reported if the transfer rate falls below a critical level, such as 1 Gbps.
5. Full-duplex mode detection: HCI checks the network mode to ensure high-efficiency full-duplex operation for businesses. If a half-duplex network mode is detected, an alert will be reported.

RAID card monitoring:

Abnormal status check: HCI supports RAID cards for system disks or local storage. The host reads the RAID card status information through RAID card commands and analyzes the health status. If errors or exceptions are detected, an alert is reported to instruct the user to check or replace the RAID card.

5.5.2.1.2 HCI Resource Monitoring

The cluster login homepage allows users to view the current usage and status of basic resources in the environment, including the number and running status of VMs and hosts, the running status of vCenter (if applicable), and the utilization of CPU, memory, and storage.

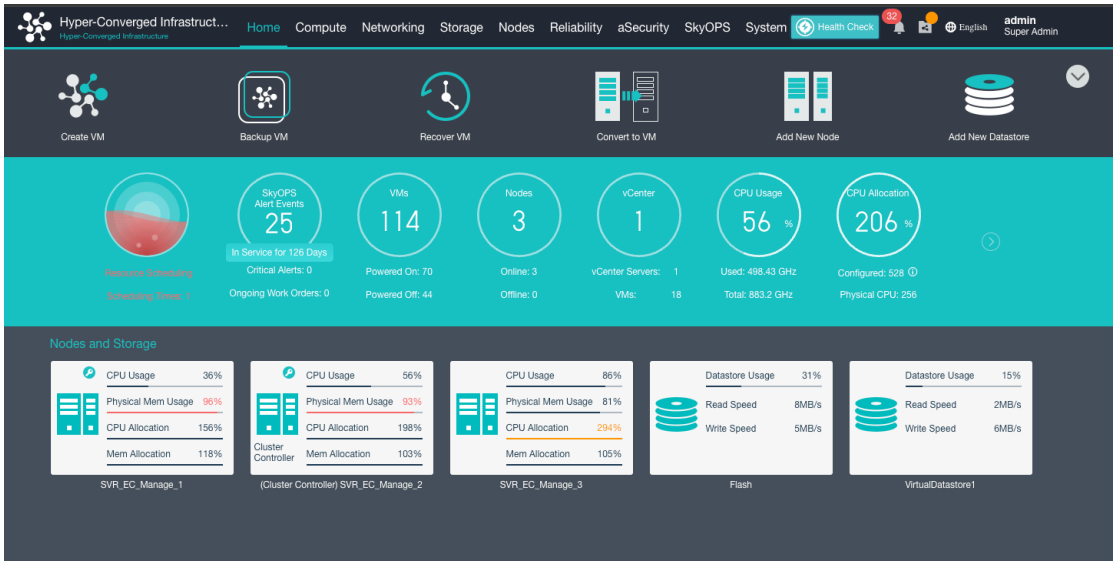


Figure 411: Homepage Resource Monitoring

On the **Nodes** page, users can view the CPU, memory, and disk utilization on the host, as well as CPU, memory, I/O, and throughput within the past 24 hours. This helps trace host performance during business exceptions. Basic host information and hardware configuration can be viewed on the host details page.

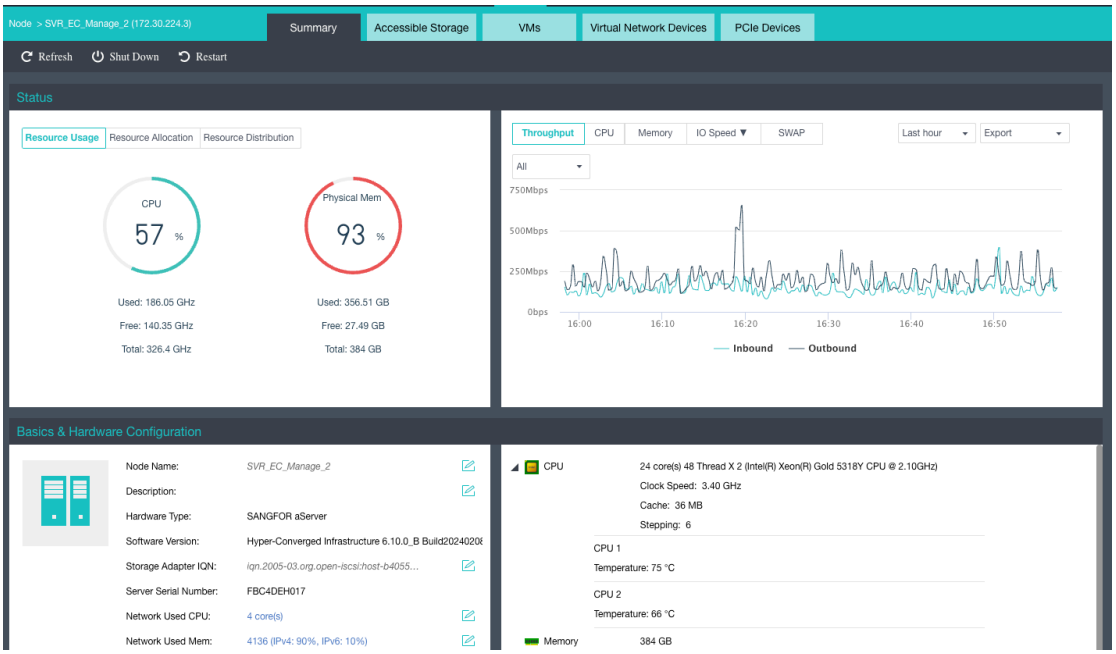


Figure 412: Host Details

On the **Compute** page, users can view the running status, name, IP address, group, resource utilization, running location, I/O, backup, and other basic information of all VMs in the cluster. Search is supported so that O&M personnel can quickly understand the running status of VMs in the cluster.

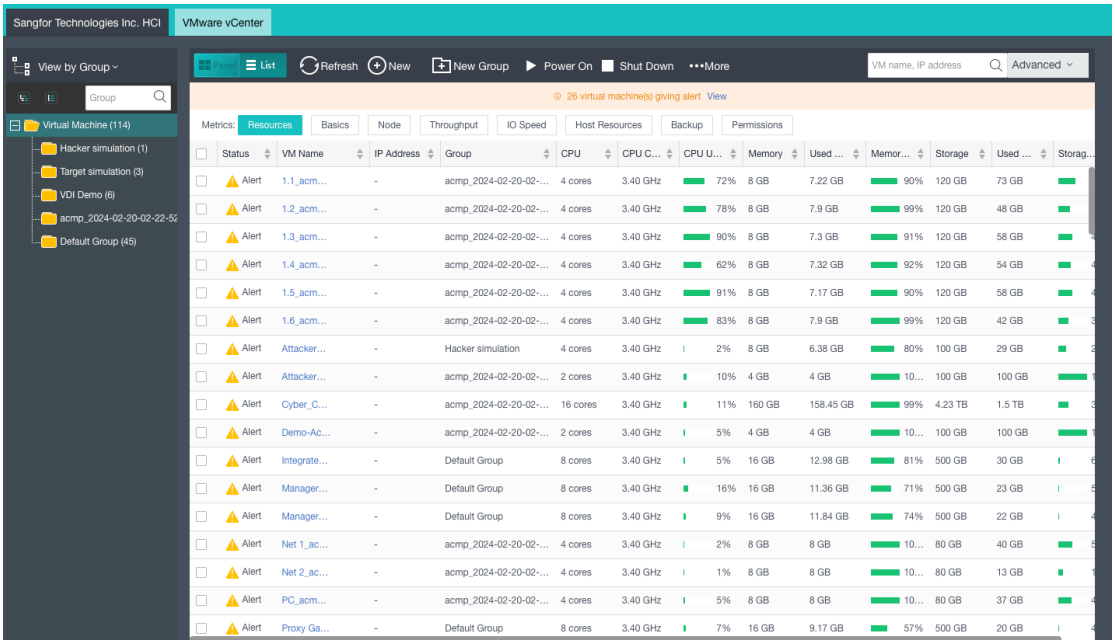


Figure 413: VM List

HCI supports one-click export of resource usage reports, including the resource configuration and usage of various business VMs.

5.5.2.2 Health Check

To simplify O&M, HCI offers a complete risk alert mechanism and a health check function. The upper right corner of the homepage provides access to the detection page that allows convenient and efficient basic checks on the cluster environment, covering cluster status and services, host hardware, VMs, virtual storage, and license keys. This facilitates prompt problem identification and offers relevant recommendations for resolving any issues.

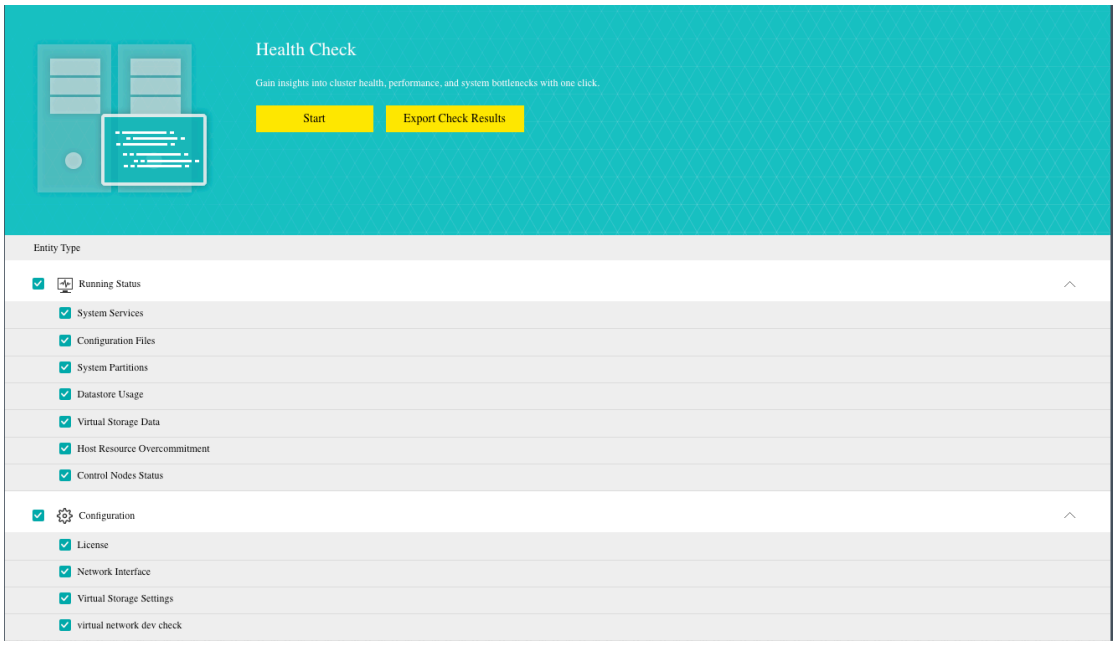


Figure 414: Health Check

5.5.2.3 Alert Service

Sangfor HCI provides a comprehensive alert service that detects exceptions in clusters, hosts, storage, networks, and VMs. When any exception is detected, the alert information is displayed on the page by alert level (currently under development). Multiple alerting methods such as syslog, SNMP server, email, and SMS message are available to ensure timely receipt of key alert information. The alert service also includes alert condition settings and viewing, searching, and exporting of logs.

HCI administrators can customize alert rules. For example, for high host memory usage alerts, HCI administrators can set the CPU clock speed threshold and the duration of time to trigger an alert. In addition to general alert rule settings, HCI administrators can also choose specific projects for customizing alert rules.

HCI logs are classified into operation logs, alert logs, and scheduling logs, allowing different business personnel to focus on specific information. For auditing and reporting purposes, HCI logs can be packaged and exported to the local system. Furthermore, syslog configuration is supported to

automatically upload kernel logs to a specified server.

5.5.2.3.1 Email Alerts

Manual configuration of email alerts is supported, as shown in the figure below.

1. Up to five email addresses can be set, and email alert testing is supported.
2. Alert convergence is supported. The same alert is only reported once within a user-defined period.
3. Different alert levels can be set, allowing for customized alert notification frequencies.

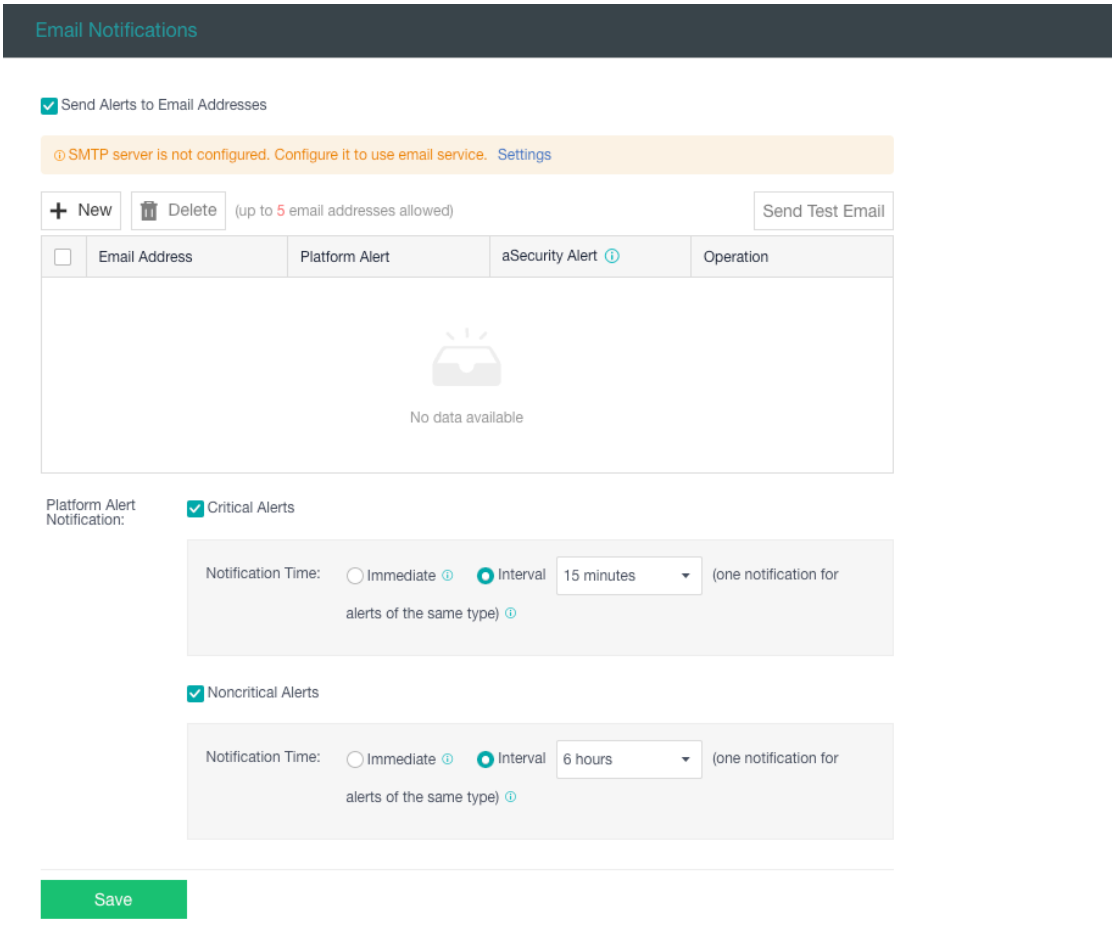


Figure 415: Email Alert Settings

5.5.2.3.2 SMS Alerts

SMS alerts are supported, allowing administrators to receive important alert information via SMS messages and take immediate action to investigate and minimize O&M risks. Mainstream telecom operators are supported.

SMS alert mode:

- **Built-in mode**: Requires HCI hosts to connect to the telecom operator's SMS service platform.
- **External mode**: The SMS message sending module is deployed on other hosts that can connect to the telecom operator's SMS service platform. Currently, only Windows 7 and Windows 10 are supported.

Main methods:

SMS alerts can be sent using three methods: SMS modem, SMS gateway, and SMS interface. SMS alert processes:

1. SMS modem: SMS modem -> Base station -> SMS center -> Base station -> Destination mobile phone
2. SMS gateway: SP -> SMS gateway -> SMS center -> Base station -> Destination mobile phone
3. SMS interface: API -> SMS platform -> Telecom operator facility -> Base station -> Destination mobile phone

Different SMS sending methods require the same mandatory parameters: the recipient's mobile number, alert content, and authentication information provided by the SMS service platform. SMS modem and gateway use standard protocols for sending SMS messages, allowing a unified interface based on the protocols. Sending SMS messages via HTTP lacks a standard protocol and generally involves calling the API provided by the SMS service platform. The SMS service platform sends messages to the specified user and returns the status information after the sending is completed. The implementation is relatively simple, and sending SMS messages across networks is supported. The main process of sending SMS messages using the API provided by the SMS service platform is as follows:

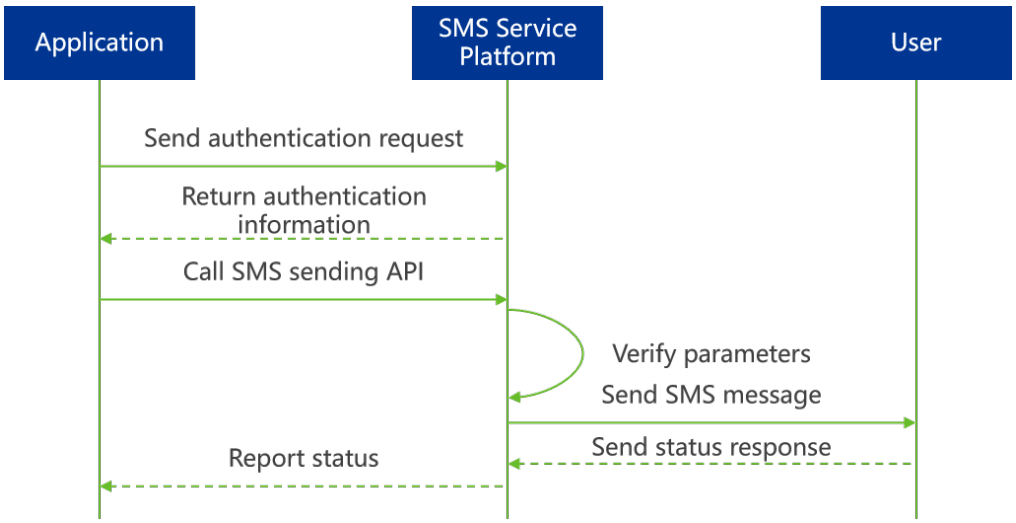


Figure 416: SMS Message Sending Through the SMS Service Platform

The API of the SMS service platform is called using HTTP, which can be classified into common HTTP calls and WebService calls.

- **Common HTTP calls**: HTTP calls are used to send messages directly. Typically, GET/POST methods are employed to execute HTTP requests. The request must include the authentication information, the recipient's mobile number, and the message content.

Example of sending an SMS message using the GET method:

https://smsAPI.ums86.com:9600/sms/API/Send.do?SpCode=000001&LoginName=admin&Password=admin&MessageContent=Alert content &UserNumber=186********

- **WebService calls**: The SMS service platform offers an API for remote calling to send messages. This method also requires parameters of the authentication information, the recipient's mobile number, and the message content for the interface call. WebService uses SOAP for data requests. By contrast, the HTTP request header needs to include attributes indicating the use of SOAP during data transmission.

The two calling methods mentioned above are user-friendly. Users do not need to worry about integrating the SMS service with telecom operators but simply send the messages to the SMS service platform. Research shows that more platforms are offering HTTP APIs for sending SMS messages, while platforms

with WebService API are relatively fewer. This is because WebService uses SOAP, which is more complex than that for common HTTP requests.

When calling the SMS service platform's API to send messages, the following items are required:

1. API call authentication information, provided by the SMS service platform
2. API for sending SMS messages, provided by the SMS service platform
3. Recipient's mobile number, provided by the user
4. Message content, prepared by the user

SMS service platforms typically use direct channels of mainstream telecom operators to provide dedicated SMS services for enterprises. Due to the lack of a unified standard for sending SMS messages using HTTP, different platforms use their own proprietary protocols. The following table describes the example authentication parameters for SMS service APIs provided by the three major telecom operators in China.

Operator	**SMS Gateway Address**	**Login Account**	**Login Password**	**License Key**	**Enterprise No.**	**Message Template**	**Gateway Signature Code**	**app_id**	**app_secret**
China Unicom	√	√	√		√	√			
China Mobile	√						√		
China Telecom	√							√	√

Table 18: Authentication Parameters for SMS Service of China's Three Major Telecom Operators

To call the SMS API provided by China Unicom, an account and password are

required; to call the SMS API provided by China Mobile, a gateway signature code is required; to call the SMS API provided by China Telecom, an application ID and password are required.

To call the SMS API provided by China Unicom, an account and password are required; to call the SMS API provided by China Mobile, a gateway signature code is required; to call the SMS API provided by China Telecom, an application ID and password are required. SMS sending APIs and authentication methods vary with the SMS service platforms. Therefore, no unified API can be used to send SMS messages through different SMS service platforms.

The table below compares different types of SMS alerts.

Type	Advantage	Disadvantage
SMS Modem	Specifically used for sending SMS messages, featuring high reliability and simple configuration parameters.	Good signal quality in the environment is required, but signal quality in server rooms is often poor. Serial port debugging is complex.
SMS Gateway	Uses a standard protocol and fixed configuration parameters.	Requires application for service activation and integration with operator gateway, featuring multiple parameters in the protocol and complex debugging.
HTTP	Simple development, requiring only API calls.	No unified interface standard, which is inconvenient for supporting other message sending methods.

Table 19: Comparison of SMS Alert Types

Research indicates that many customers have not purchased SMS modems. Therefore, HCI currently supports only SMS gateway and HTTP modes.

SMS alert settings:

Up to 10 mobile numbers are supported, requiring configuration of an SMS server for alerting, as shown in the figure below.

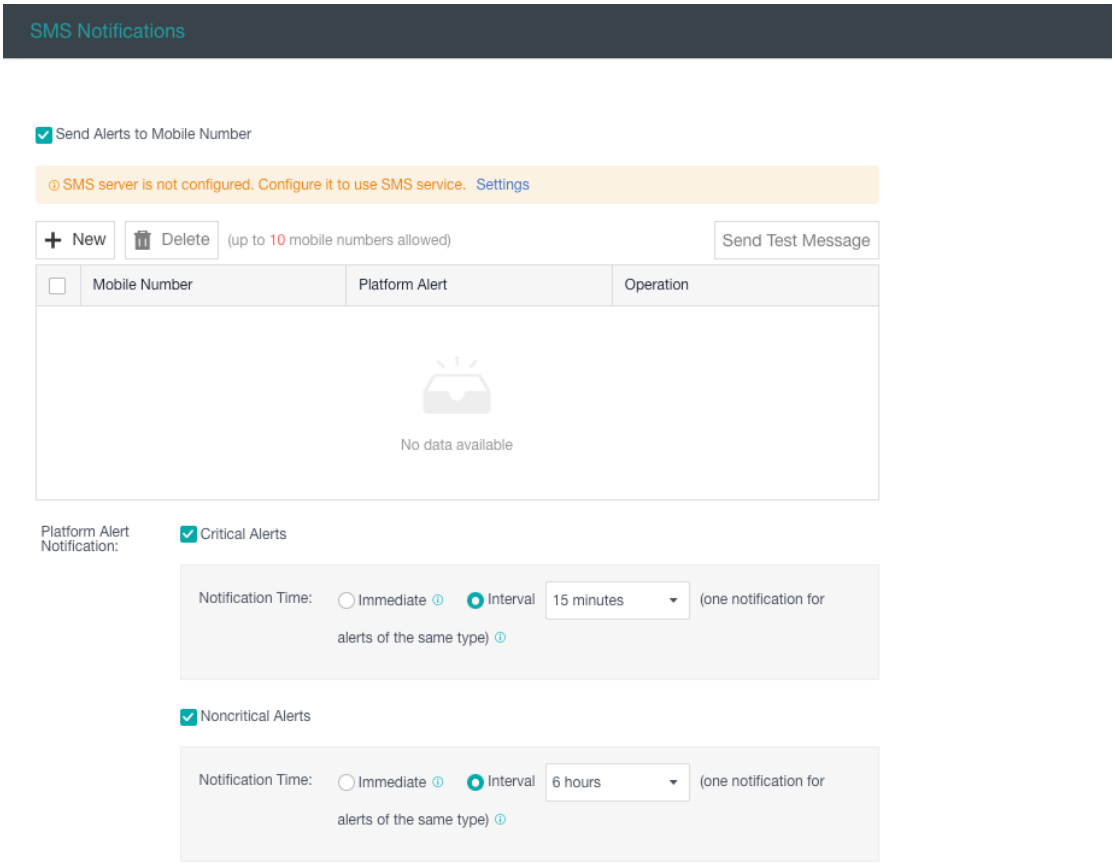


Figure 417: SMS Alert Settings

5.5.2.4 Report Center

Sangfor HCI records logs of various operations for troubleshooting and security auditing, including operation logs (such as console logins, VM creation/modification/deletion, and virtual network adjustments), scheduling logs (scheduling VMs to other physical hosts), and HA logs. Report Center provides a comprehensive record of operations, including completion status, time, operation objects, and content.

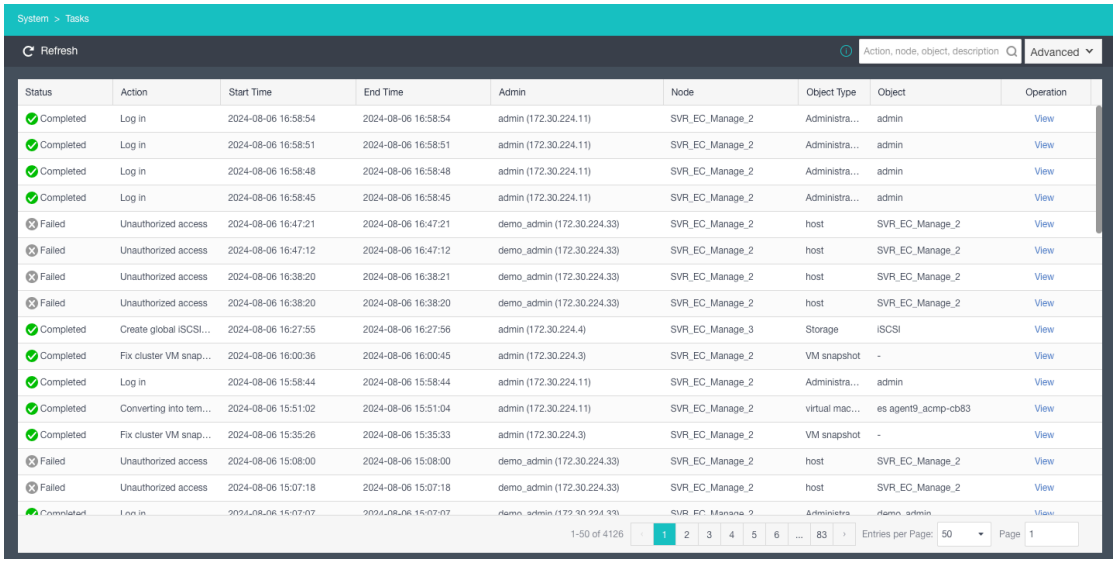


Figure 418: Report Center

5.5.2.5 Command Line Instructions

The HCI frontend offers a command line console interface and 20 visual and operable common maintenance commands. This facilitates basic troubleshooting for O&M personnel and assists channel engineers and customers with technical backgrounds in improving product maintainability.

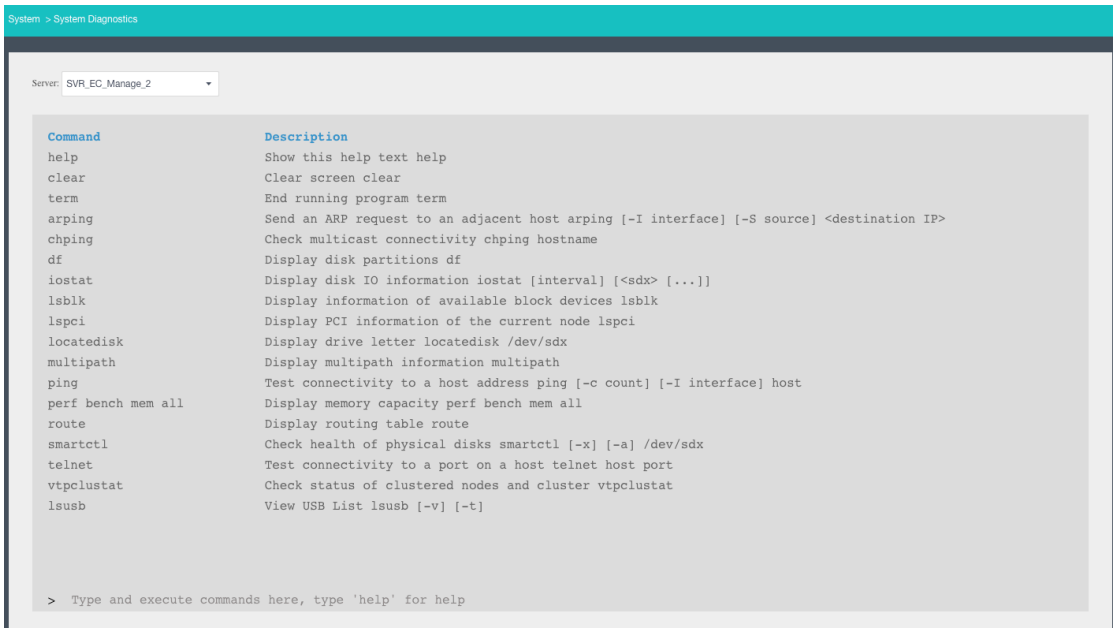


Figure 419: Command Line Instructions

5.5.3 Cluster Management

5.5.3.1 Cluster Communication Mechanism

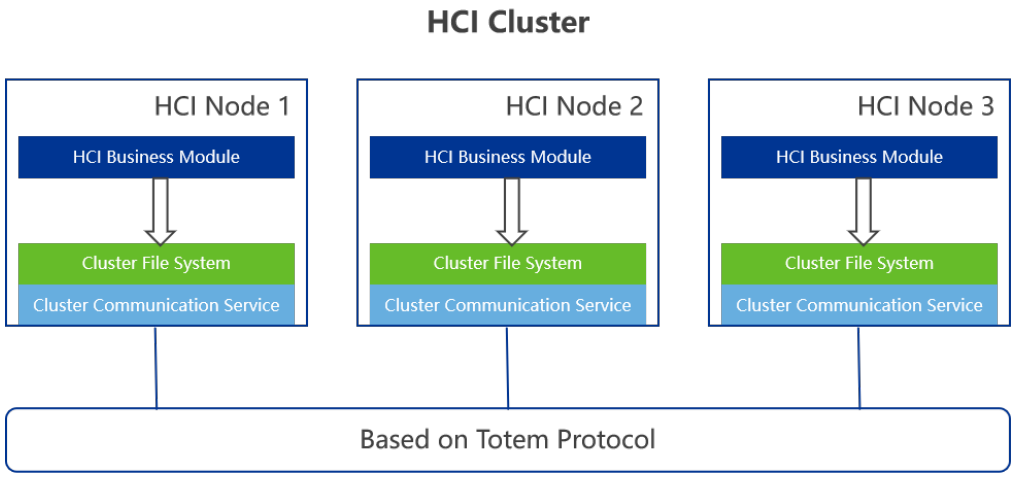


Figure 420: Cluster Communication

HCI uses the corosync service with the Totem protocol to ensure consistent configuration information within a cluster. The service creates a token ring between cluster nodes. During each message sync, all nodes receive the same message queue. Each node maintains a separate replica of the cluster configuration, providing high reliability and availability even in the event of message loss or node crashes.

5.5.3.2 Cluster HA

Cluster HA allows the seamless transfer of services, resources, IP addresses, and other components from one server to another in the event of a failure, ensuring business continuity. Two or more servers form an HA cluster.

HCI ensures HA for nodes, service components, and data. It supports adding nodes, allowing VM service migration to other nodes if the node fails. Compute, network, and storage services are deployed across control nodes and instances, protecting against node failures and service exceptions. In addition, storage nodes use the multi-replica feature to prevent data loss from affecting business access.

100% HA is not possible. However, HCI incorporates operational and recovery procedures to minimize mean time to repair (MTTR) and mean time to failure (MTTF).

5.5.3.3 Dual-Host Quorum

A virtual datastore consisting of two hosts lacks a quorum mechanism, posing risks of data split-brain and potential data loss. To mitigate such risks, aSAN recommends a direct connection between the storage interfaces of the two hosts, bypassing switches to avoid any impact brought by switch failures. Additionally, aSAN implements network leverage in the dual-host scenario, allowing the use of management or data networks as the communication network between host storage when the storage network is completely interrupted. This significantly enhances the reliability of the storage network and greatly reduces the risk of split-brain.



After you add a host to the two-host virtual datastore, the quorum mechanism activates and generates a witness replica. In this process, the VMs on the virtual datastore must be shut down. Therefore, you must shut down all VMs in the cluster before expanding the two-host virtual datastore.

5.5.3.4 Online Expansion

As the IT business continues to grow in various organizations, expanding IT resources becomes inevitable. This brings about the following issues:

1. Hardware resources are insufficient and lead to high CPU, memory, and storage I/O utilization across the HCI cluster, affecting business performance and service capability.
2. Under high load conditions, if a host in the cluster fails and the other hosts lack sufficient resources to bear the VM services from the failed host, part of the businesses may be interrupted and cannot be quickly recovered.

Therefore, it is necessary to expand the cluster resources to ensure business

support. In addition, uninterrupted business operations must be maintained, allowing administrators to smoothly and securely perform expansion.

Sangfor HCI offers online host expansion and online disk expansion functions, allowing administrators to scale out resources without affecting the operation of the original platform. After expanding cluster resources, administrators can perform automatic or manual load balancing on running VMs.

5.5.3.4.1 Host Expansion

The HCI cluster supports the addition of one or more servers, scaling out cluster resources to bear the workload from other hosts. Disks on the new host can also be added to aSAN as required, expanding the capacity and enhancing the cluster performance of the HCI storage resource pool.

HCI provides the one-click host adding feature, allowing quick and convenient online cluster expansion.

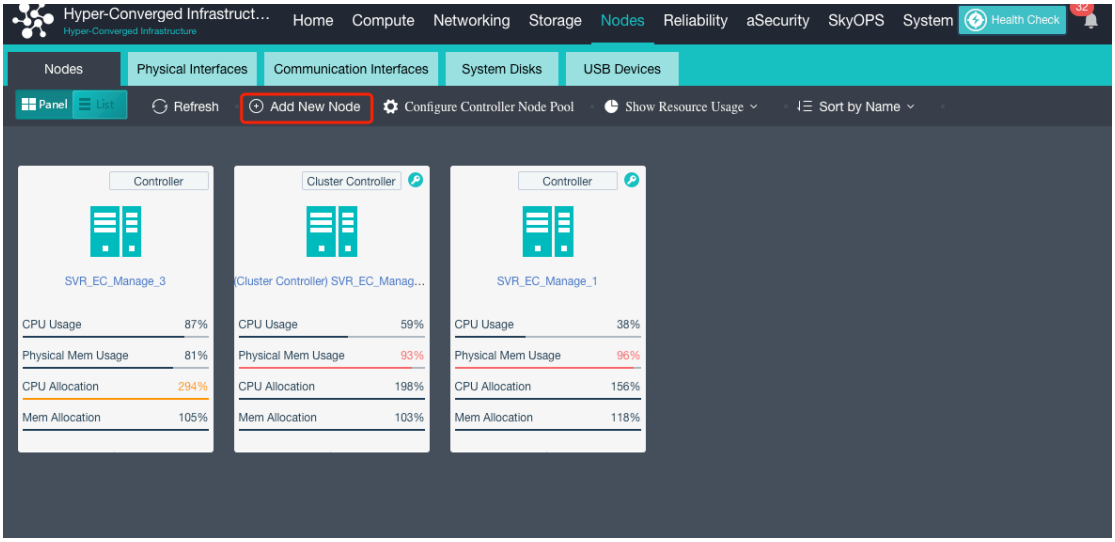


Figure 421: Host Expansion

When virtual datastore resources are expanded in a cluster, one or more disks may have excessive free space. HCI ensures the even distribution of data within the virtual datastore through the data balancing mechanism. This maximizes the space and performance utilization of newly upgraded hardware and performance while ensuring efficient resource allocation.

5.5.3.4.2 Storage Expansion

The distributed storage architecture of aSAN is equipped with the scale-out and scale-up features, enabling seamless adding of storage resources to aSAN virtual datastore. These resources can include disks, disk groups, or entire physical hosts.

Virtual datastore expansion features:

- **Flexibility**: supports expansion of individual disks or physical hosts.
- **Quickness**: Once the expansion components are ready, virtual datastore expansion can be completed in just a few minutes.
- **Smoothness**: Virtual datastore expansion has no impact on business operations.

The process is shown in the figure below.

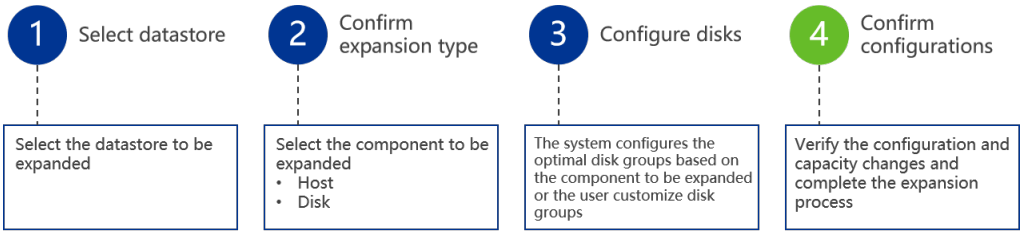


Figure 422: Virtual Datastore Expansion Process

Enhanced fault tolerance capability:

1. HCI ensures the even distribution of data within the virtual datastore through the data balancing mechanism. This maximizes the utilization of newly upgraded hardware's space and performance while ensuring efficient resource allocation.
2. In a three-replica configuration, when three or four hosts are expanded to five or more hosts, aSAN triggers data balancing to distribute the witness replicas to the new hosts. This enhances the fault tolerance capability of the three-replica configuration with five or more hosts, ensuring host anti-affinity for all data replicas and witness replicas.

5.5.4 Cluster Upgrade

The upgrade supports three modes: offline upgrade, in-place active upgrade, and rolling active upgrade. The mode is automatically determined based on the cluster environment. Offline upgrade is the default option, followed by in-place active upgrade, and then rolling active upgrade.

Offline upgrade refers to a scenario where user businesses need to be temporarily stopped for maintenance, resulting in a longer upgrade time. The upgrade will take effect once it is completed in the backend and users are prompted to shut down VMs and restart physical hosts. In-place and rolling active upgrades are both performed while the system is running. For the former, VMs on the nodes do not need to be migrated across hosts, resulting in higher migration efficiency.

For the latter, all VMs on the nodes need to be migrated to other nodes before the upgrade can proceed.

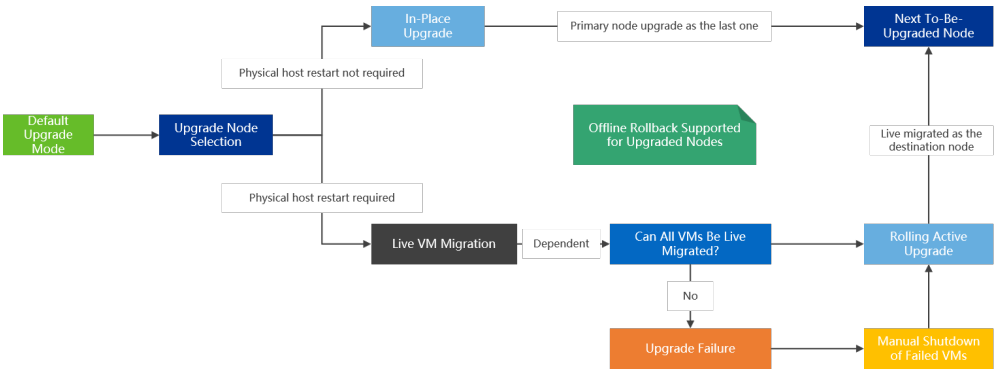


Figure 423: In-Place and Rolling Active Upgrades

The control plane services of HCI cluster are upgraded in parallel first. During the upgrade, the cluster operates in maintenance mode and supports read-only and shutdown operations. Once the control plane upgrade is completed, a rolling upgrade of the node data plane can be performed. In the case of in-place and rolling active upgrades, the former does not require host restart, while the latter does. Both upgrades proceed node by node in active mode, with the node being upgraded in maintenance mode. Tasks and page operations related to the node being upgraded are blocked, while other read and write operations can proceed properly. The active upgrade of the entire

cluster remains incomplete until all nodes complete the data plane upgrade. Both in-place and rolling active upgrades support sequencing and pause point setting, allowing more flexible upgrade management.

In-place active upgrades rely on local live migration of data and live restart of services. This means that new and old services must undergo live switching, live restart, or multi-path replication. Essentially, when the new service starts, the old service does not immediately exit but completes tasks and on-site switching. On the other hand, rolling active upgrades rely on cross-host live migration, requiring stricter migration conditions. Note that during the upgrade process, which may involve both new and old programs, compatibility is achieved through forward compatibility using interfaces. The new server needs to allow access from the old client, and the new client can also access the old server.

5.5.4.1 In-Place Active Upgrade

When customers want to upgrade the cluster version for additional features or increased reliability in HCI, the upgrade may affect running businesses.

The in-place active upgrade technology of Sangfor HCI eliminates the need to restart the physical machine. By limiting the suspension time of business upgrade I/O to 3–5 seconds, the impact on services is minimized, ensuring continuous operation. For example, for a database business, the upgrade process is completed within 3 seconds without causing interruptions.

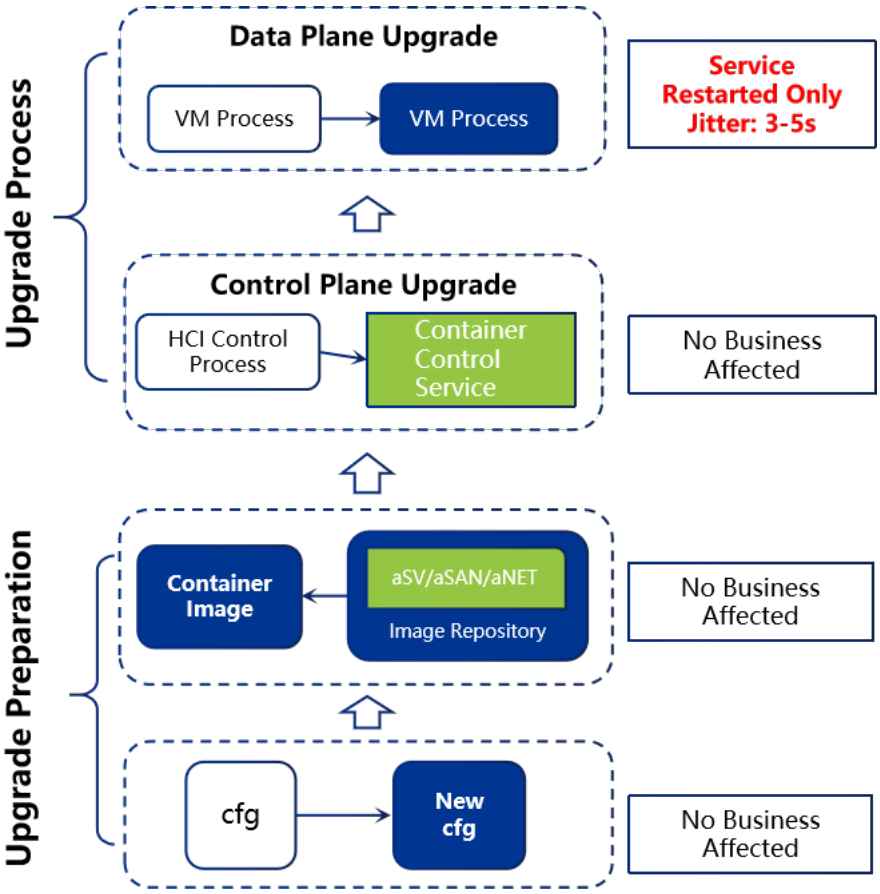


Figure 424: In-Place Active Upgrade

In-place active upgrade does not alter the running location or status of VMs. It directly upgrades the host where the VM is located. The main interruption time comes from the VM upgrade process. By transforming HCI into services and decoupling the components, services are provided through unified APIs. During the underlying file system upgrade on the host, as long as the API of various components is downward compatible and functioning properly, the running status of VMs and migration tasks are not affected.

5.5.4.2 Rolling Active Upgrade

During the upgrade of HCI, if the underlying kernel version differs between the target version and the source version, the upgrade may result in a longer interruption time and possibly require a server restart. Sangfor HCI offers a rolling active upgrade solution that involves live migration of the VMs on the

hosts to be upgraded to other hosts and then in-place active upgrade on the hosts. The rolling upgrade solution ensures business continuity. Rolling active upgrade supports host restarts and provides flexible support for scenarios that require kernel, driver, and OS upgrades.

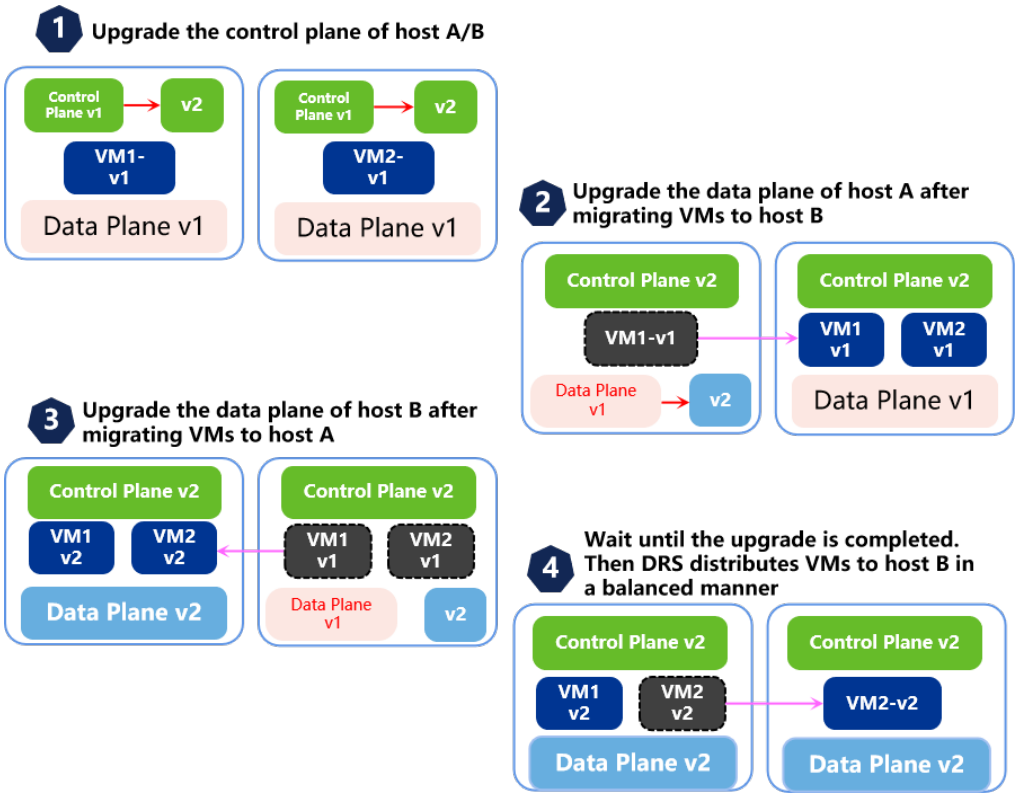


Figure 425: Rolling Active Upgrade

A rolling active upgrade involves live migration of the VMs and an upgrade afterward. The rolling upgrade solution ensures business continuity. Users can specify the scope and sequence of host upgrades based on their business distribution. It allows individual selection for upgrades, supports pausing and resuming, and provides complete control over the upgrade process. Additionally, it offers visual management, ensuring controllability and visibility throughout the upgrade process.

5.5.5 Management Platform Security

Sangfor provides cluster security access protection, including SSH secure login to HCI hosts, as well as HTTPS secure login, anti-brute force measures, two-

factor authentication, and restricted access by IP and MAC address of the HCI management platform. Secure encryption during the login and user permission management are also included.

5.5.5.1 SSH Secure Login to Host

Access to the HCI host console is restricted to the SSH protocol, ensuring the integrity of important data during transmission.

1. The IP address whitelist can be set to allow login to HCI hosts.

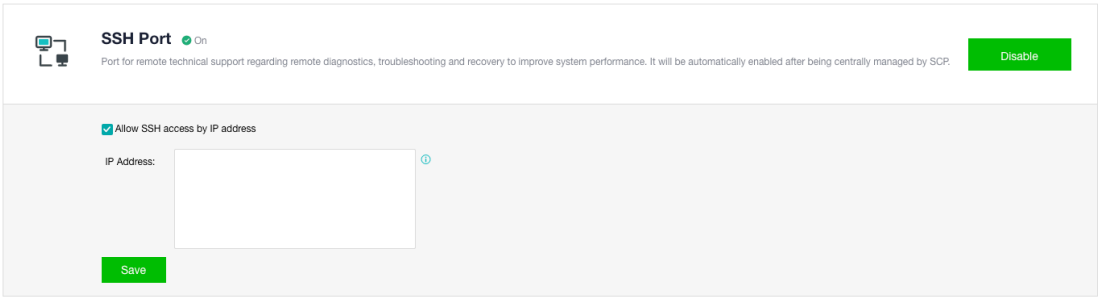


Figure 426: Settings of the SSH Port IP Address Whitelist

2. SSH sessions are automatically disconnected after 10 minutes.

5.5.5.2 HTTPS Secure Login

To prevent attackers from intercepting communication data between the console and devices through network sniffing, HCI encrypts account information using HTTPS. Users must log in to HCI using HTTPS.

5.5.5.3 Anti-Brute Force Secure Login

To protect keys from brute-force attacks, HCI offers customizable anti-brute force mechanisms to ensure secure login.

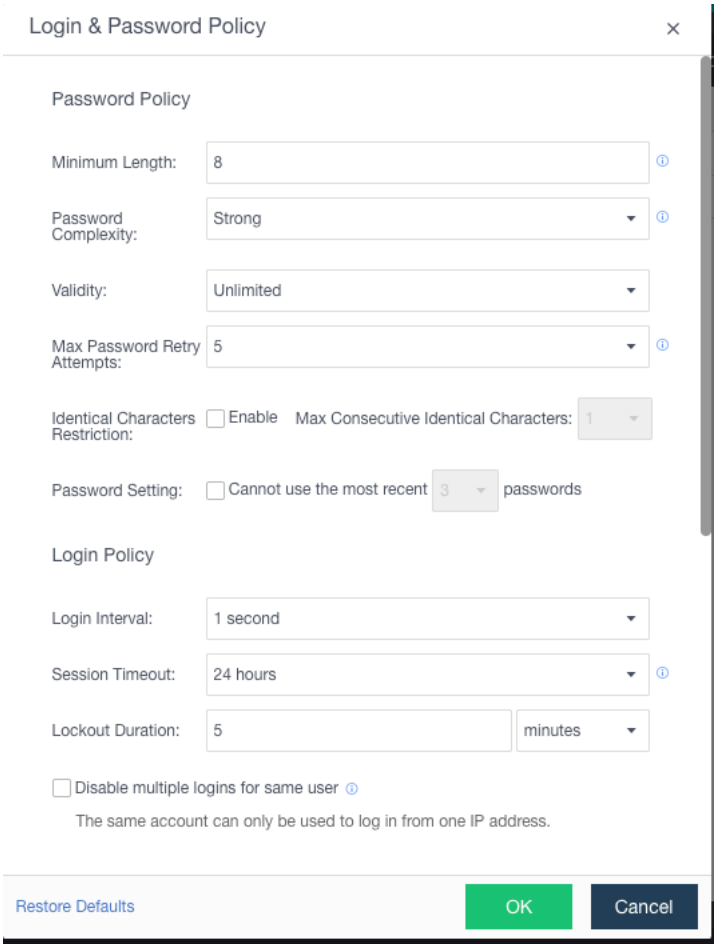


Figure 427: Anti-Brute Force Login Settings

1. Weak password check

By default, HCI utilizes the "pre-shared password" user authentication method. New users are assigned a pre-configured password when being created, which is used along with the account name to log in to HCI. If the user password is weak, it may be subject to brute-force attacks. To solve this problem, HCI performs a weak password check and highlights simple passwords. Additionally, a time lock setting can be enabled to prevent brute-force attacks on web system passwords, ensuring the security of user accounts.

2. User password expiry check

- The list displays the password expiration dates of the current user.
- Expired passwords are dimmed.
- Passwords about to expire are red.

3. IP address lockout for multiple failed login attempts

Users can set a maximum number of login attempts. If this limit is exceeded, access is denied. During the IP lockout period, if the same IP address is used for re-login, regardless of the password correctness, the timer will reset to 300 seconds. During this time, the IP address cannot be used to establish an SSH connection. After 300 seconds, the address will be unlocked, and a connection can be established with the correct password. Successful logins of the IP address reset its previous failure count, while its lockout does not affect operations of other IP addresses.

4. Session encryption and logout due to timeout

Login sessions undergo additional encryption using a random key to prevent information interception. The login interval and session timeout can be set to ensure system security during idle periods. The account is logged out upon session timeout. If the login is stopped on the server, the token will be automatically invalidated.

5. Restriction on account login using an IP address

To enhance security, HCI restricts simultaneous logins of the same account using the same IP address.

5.5.5.4 Two-Factor Authentication for Secure Login

Login using accounts and passwords carries the risk of password leakage. An attacker who obtains the password can directly log in to HCI. To mitigate this risk, HCI uses two-factor login authentication based on the USB key and account password. This further enhances HCI security. Users must correctly register the USB key and provide the correct account password for dual authentication, restricting unauthorized access. Users can customize the two-factor authentication policy, while HCI administrators can set the policy by user.

The USB key-based authentication module operates as a backend service. Through its interface, the browser retrieves USB key-related information from the endpoint and sends commands to burn certificates in the USB key module.

Procedure for enabling the USB key login policy for a user by an administrator:

1. Log in to HCI and locate the user in the user management list.
2. Insert the USB key into the terminal PC device.
3. Bind the USB key.
4. Once the binding is completed, certificate management is enabled for the administrator.

After these settings, the administrator must insert the corresponding USB key and enter the correct account password for successful login.

Procedure for binding USB key for a user:

1. HCI requests the current version number of the service. If no response is available, the user is asked to install the service and provided with a download link. Manual installation is required. If the service responds with a version number that is not the latest, the user is asked to download the latest installation package. Otherwise, the next step cannot be performed.
2. HCI requests the number of USB keys currently inserted in the endpoint. Only one USB key can be burned at a time. If no USB key is available or more than one key is available, the user is notified of improper operation.
3. HCI requests the owner of the USB key. If the currently inserted USB Key does not belong to the logged-in user, login is not allowed.
4. HCI burns the USB key. The service generates a public-private key pair, with the private key stored in the USB key and the public key at the backend.
5. The user is successfully bound to the key.

USB key authentication process for user login:

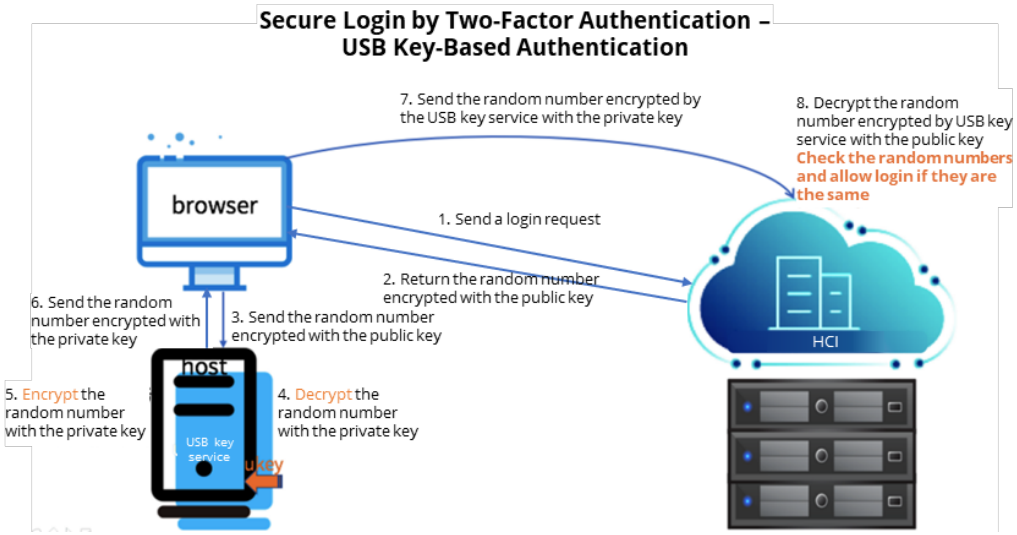


Figure 428: USB Key Authentication Process

1. The browser sends a login request to the backend.
2. The backend returns a randomly generated number encrypted by using the public key.
3. The browser sends the number to the service.
4. The service decrypts the number by using the private key.
5. The service encrypts the number again by using the private key and returns the result to the browser.
6. The browser sends the number to the backend.
7. The backend decrypts the number by using the public key, verifies it, and allows login if the number is the same as the one it returns.

5.5.5.5 Restricted Secure Login by IP Address and MAC Address

To further mitigate the risk of unauthorized access to the web console, HCI offers a secure login control policy based on IP and MAC addresses. Users can enable this policy and specify the IP and MAC addresses for secure login. Only users with allowed IP and MAC addresses can successfully log in, providing additional protection against unauthorized access.

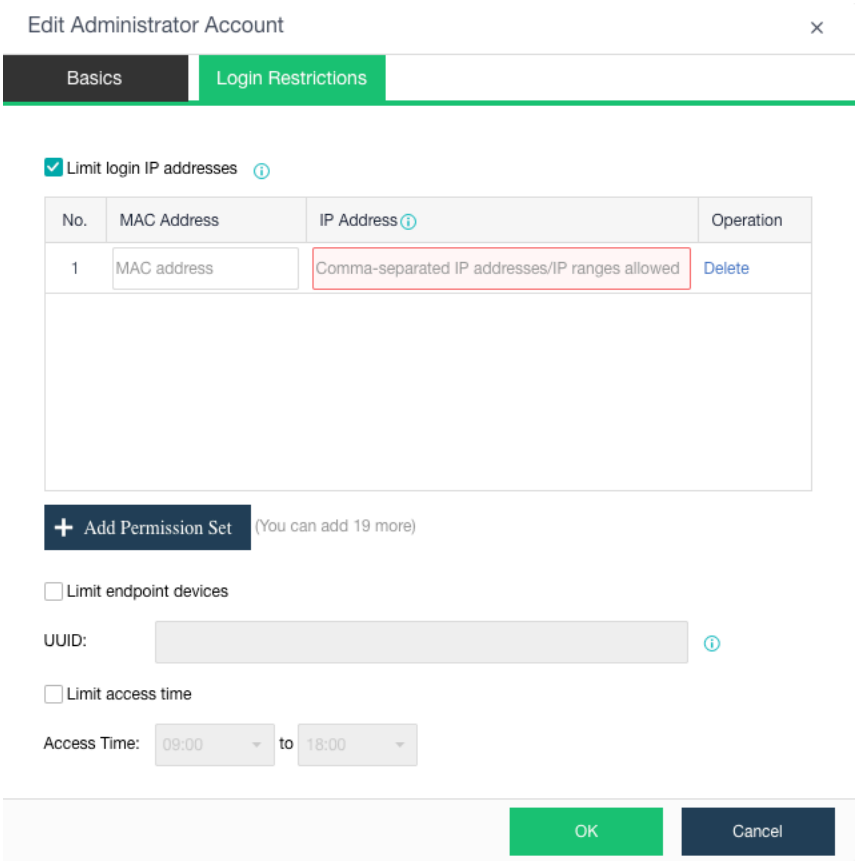


Figure 429: Secure Login Configuration Based on IP/MAC Addresses

Implementation process:

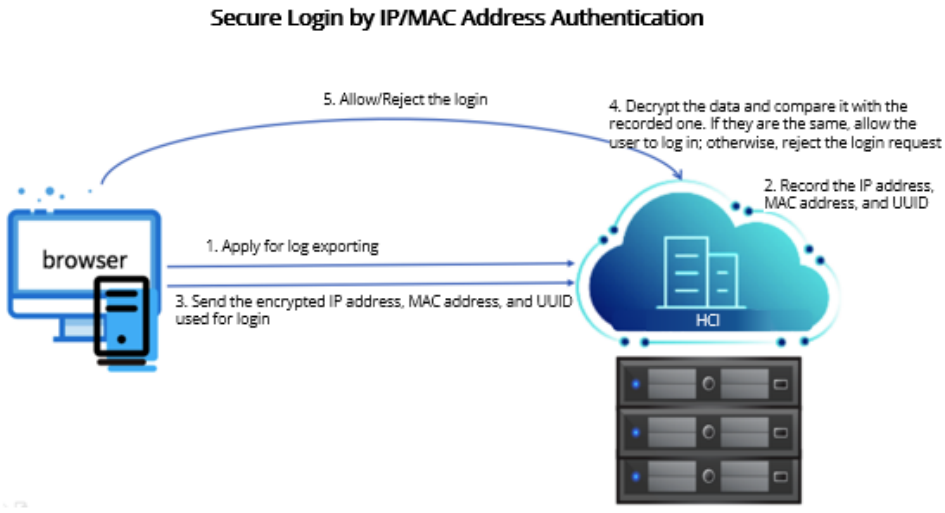


Figure 430: Secure Login Process Based on IP/MAC Addresses

Implementation process description:

The administrator logs in to HCI and sets the allowed IP/MAC addresses for login and endpoint device restrictions on the corresponding pages for the user. The browser sends the setting information to the backend for storage, as shown in the figure below.

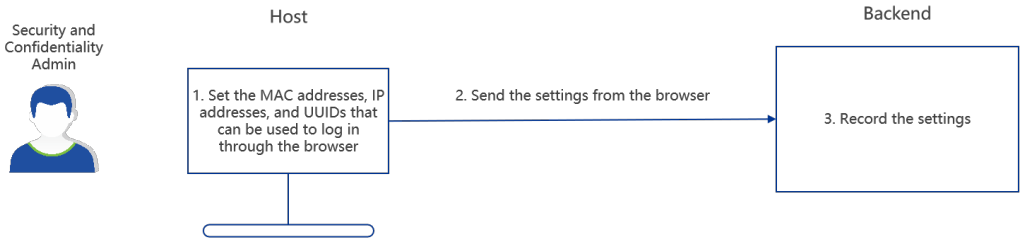


Figure 431: IP/MAC Address Settings

If the administrator sets login restrictions during the login of a user, the user will not be logged out, and the restrictions will take effect upon the next login. The following figure shows the diagram.

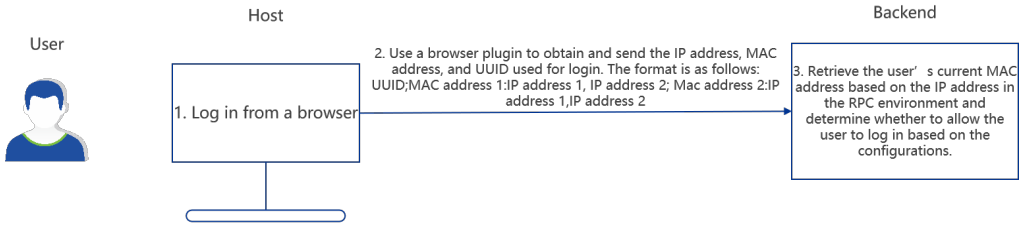


Figure 432: Login by IP/MAC Addresses

During login, the UUID of the current host and the IP/MAC address of the NIC are obtained, concatenated with random numbers, encrypted, and transmitted to the backend.

Thus, each login generates different data for the backend, preventing users from using a fixed string for login on any host.

The backend decrypts the data, extracts the random number, and compares the data with the previous data it has sent. If the random numbers do not match, login is prohibited. The backend verifies the user's login host UUID and the IP/MAC addresses of all NICs based on the administrator's settings. If the data matches, login is allowed; otherwise, it is prohibited.

5.5.5.6 Security Encryption Technology

1. HCI login password:

The transmission process uses HTTPS encryption. The password is stored in a one-way hash form. The salt value for hashing is generated using the /dev/urandom pseudo-random number generator.

2. VM disk encryption

The AES256 encryption algorithm is used.

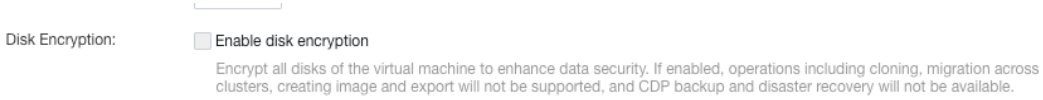


Figure 433: Disk Encryption

5.5.5.7 User Permissions Management

To prevent unauthorized access from attackers, HCI provides a user permissions management feature that supports the configuration of resource usage permissions at the granularity of individual VMs, virtual networks, storage, and physical disks. It supports setting resource quota limits for users, covering CPU, memory, and storage. Additionally, HCI supports system permission settings, including permissions of physical resources, system configuration and maintenance, business maintenance configurations, and read-only and invisible permissions.

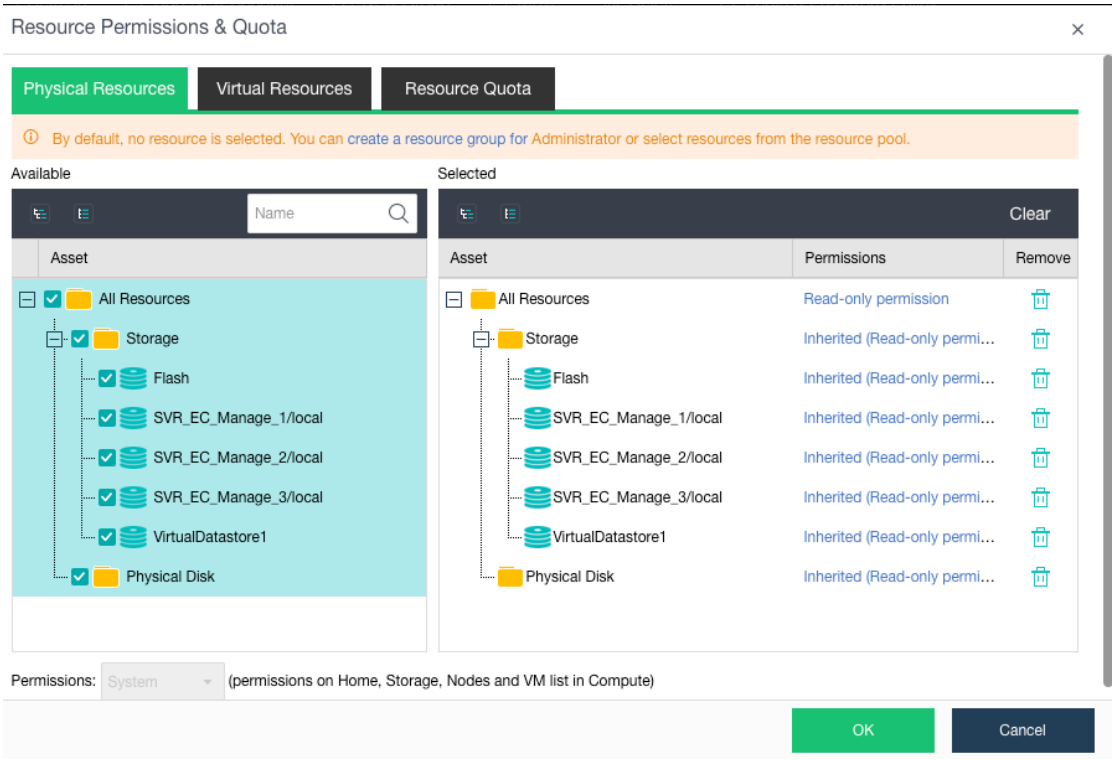


Figure 434: User Permissions Management

5.5.5.8 Permission Separation

Sangfor HCI provides three types of security management personnel: system administrators, security and confidentiality administrators, and security auditors. Role-based access control is implemented, and relevant permissions are assigned to users based on their respective roles. System administrators are responsible for system operation, security and confidentiality administrators oversee security and confidentiality matters, while security auditors handle the security auditing work. The system imposes strict restrictions on the accessible data and operations for each role. It incorporates permission division in API design, ensuring that each user possesses specific permissions, thereby preventing unauthorized access.

HCI does not feature a super administrator role in its administrator mode. Instead, it adopts a permission separation approach, where the role of the super user is subdivided into system administrator, security and confidentiality administrator, and security auditor. Each role carries its own set of

responsibilities, and user permissions are relatively independent. This arrangement facilitates mutual supervision and restraint, strengthening the management of sensitive information systems and mitigating the risk of data leakage.

	System Administrator	Security and Confidentiality Administrator	Security Auditor
Responsibilities	Management of resources, including VMs, hosts, storage, backup, and migration	Permissions management, including distributed firewalls, user permissions, VM classification, and VM security policies	Log auditing

Table 20: Roles and Responsibilities

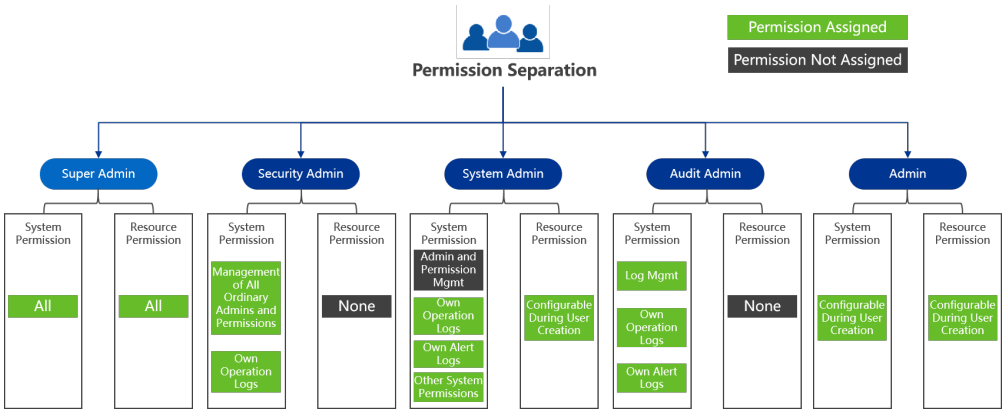


Figure 435: General Principles of Permission Separation

5.6 Typical HCI Business Processes

5.6.1 VM Network Traffic Forwarding

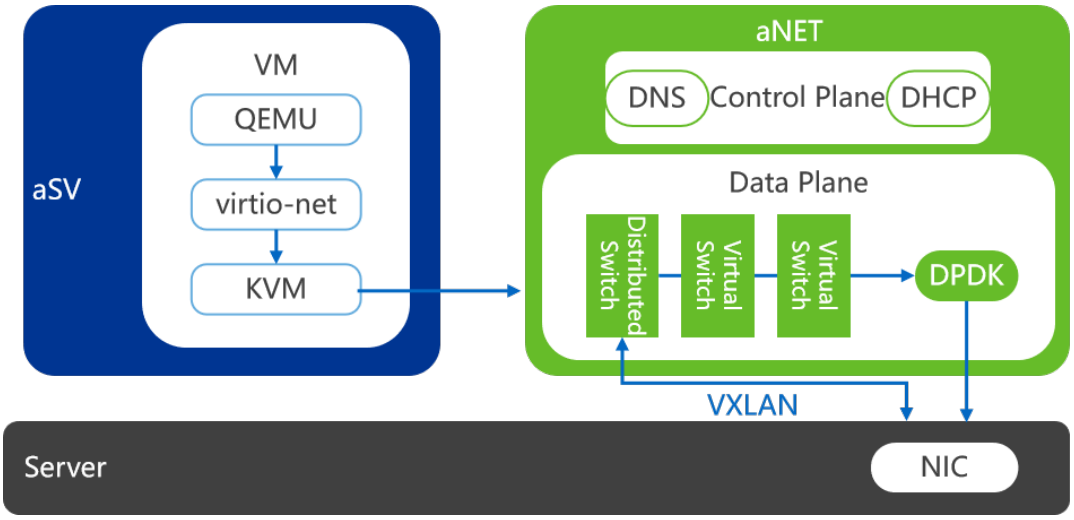


Figure 436: VM I/O Forwarding Path

- **QEMU**: Each VM has a corresponding QEMU process instance that is used to implement basic virtualization functions.
- **Dataplane**: Data plane forwarding service of the network virtualization system.
- **DPDK**: toolset of functions and drivers for data packet processing.

Virtual NIC packet sending process:

1. The kernel protocol stack of the VM OS (Windows/Linux) sends packets from the business system to the virtio-net driver and shared memory. These packets are then forwarded for logic processing by the distributed switch.
2. The distributed switch checks the destination MAC address of the packet. If the outgoing interface is a VXLAN interface, it bypasses the virtual router and virtual switch logic and sends the packet directly to the physical NIC. If the outgoing interface is connected to a virtual router, the packet is forwarded to the virtual router for processing.

3. The virtual router goes through a series of network services, such as DNS or DHCP, and determines whether the destination is a virtual switch based on the outgoing interface. It then forwards the packets to the virtual switch for logic processing.
4. The virtual switch obtains the outgoing interface information, calls the DPDK API to send the packets, and sends them to the physical NIC.

5.6.2 VM I/O Reading Process

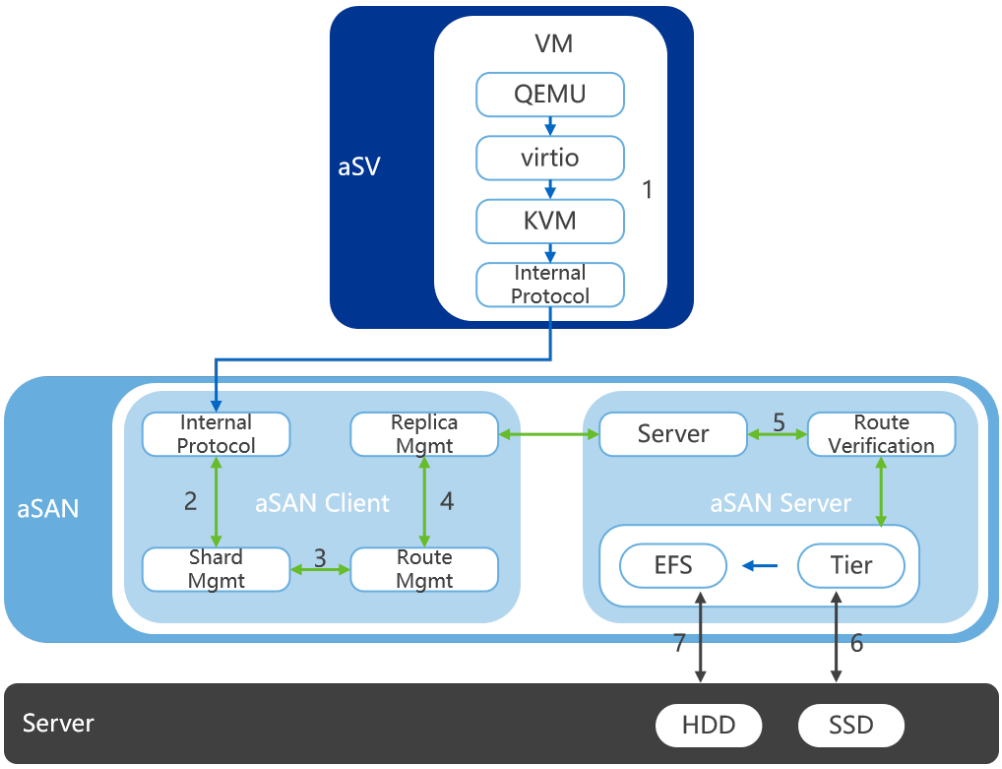


Figure 437: VM I/O Reading Path

1. QEMU: Each VM has a corresponding QEMU service instance that implements basic virtualization functions.
2. The kernel protocol stack of the VM system sends the I/O requests from the business system to the virtio-blk (virtual block) driver. The virtio-blk driver performs validity checks on the I/O requests and converts them into requests for the underlying image file. These requests are then forwarded to the virtual storage system through the libnfs API.

3. aSAN client: Client of the virtual storage system.
4. Shard: Files on virtual disks are divided into 4 GB shards to achieve a more balanced distribution of data. Based on the sharding and striping algorithms, data is split into small data blocks and their storage locations are calculated. Then, data is read from the lower layer accordingly.
5. Route: During the sharding, the distribution locations of all data replicas are searched, correlated, and bound in the shard information. This distribution information is referred to as a route.
6. Replica: To ensure data redundancy, the system writes multiple data replicas into disks on different hosts. A success result is returned only after all data replicas are written, ensuring consistency among the replicas.
7. Data server: data server of the virtual storage system.
8. The system checks whether the route information about the shards is consistent. If not, the system returns an error and reselects the route. If so, the process continues.
9. Tier: When receiving a read request, the system checks whether the current SSD has cached the data. If all the data is cached, it reads the data from the SSD and returns the result. If not all data is cached in the SSD, a read request will be sent to EFS for the uncached data.
10. EFS: When receiving a read request, the system checks whether the data has been written. If the data has never been written, it directly returns 0. Otherwise, it reads from the HDD and returns the result.

5.7 Open Support

Sangfor offers HCI APIs and a command-line (CLI) tool.

5.7.1 HCI APIs

To meet customer needs for accessing HCI APIs for product development and management, HCI provides API documentation. For details, refer to the ***Sangfor HCI API Documentation***.

5.7.2 CLI Support

Traditionally, server virtualization O&M management involves GUI-based operations on clients of virtualization software products for basic O&M. These operations are intuitive and convenient. However, as the virtual pool grows, the traditional management mode requires repetitive manual operations, resulting in high manpower and time costs, risks of misoperation, difficulty in tracking operation records, and reduced O&M efficiency.

To address these problems, Sangfor provides a CLI tool that primarily handles HCI O&M. It allows users to perform VM management, disk management, device O&M, and other operations without logging in to the web page. It also meets the requirements of users for bulk configuration, O&M, change, and inspection, improving operational efficiency and reducing related risks.

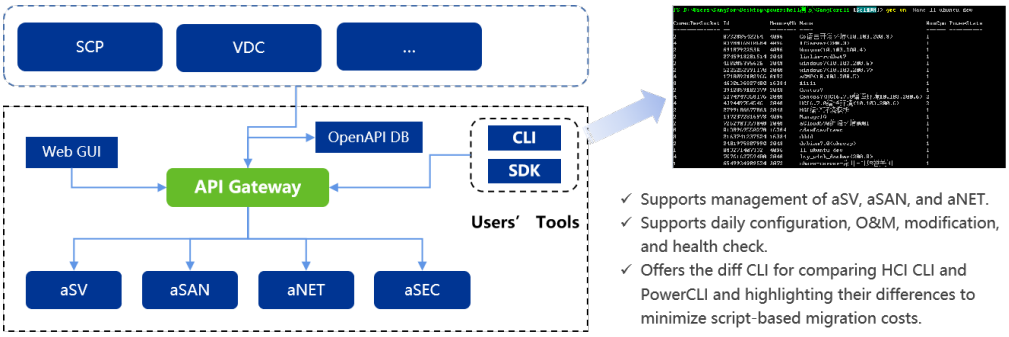


Figure 438: CLI Tool Released

5.8 Configuration Information and Product Compatibility

To provide customers with an integrated service experience, Sangfor HCI offers various series of cloud servers, including 2 U/4 U servers, low-/medium-/high-end servers, and all-flash and GPU servers that cater to different business requirements.

Sangfor HCI is widely compatible with mainstream server hardware devices and OS. It supports mainstream x86, C86, and Arm CPUs, with compatibility for over 600 types of servers. It also supports more than 20 types of GPUs, such as Tesla, Quadro, GeForce, Titan, and Ampere. HCI can host VMs running on over

10 types of OS.

For specific compatibility information regarding HCI, log in to the Sangfor Cloud Compatibility Platform for details.

5.8.1 HCI Underlying Configuration Information

Development languages: C, C++, Perl, Python, and Golang

Databases/middleware: SQLite, Redis, ZooKeeper, MySQL, and Kafka

Kernels:

1. Linux-4.18.193.19.1 (x86);
2. Linux-4.19.90-25.17.v2101.osc.sfc.6.9.0.0028.ky10.x86_64 (C86);
3. Linux-4.19.90-25.17.v2101.osc.sfc.6.9.0.0028.ky10.aarch64 (Arm)

5.8.2 Server Compatibility

HCI optimizes products at the system software level and is compatible with a wide range of third-party servers from IBM, Dell, Huawei, HP, Inspur, Lenovo, Sugon, H3C, Cisco, PowerLeader, Nettrix, Supermicro, Fujitsu, and Greatwall.

To provide an integrated service experience, Sangfor also offers various cloud servers, including 2 U/4 U servers, low-/medium-/high-end servers, and all-flash and GPU servers that cater to different business requirements.

5.8.3 CPU Compatibility

Sangfor HCI supports various CPUs from Intel, AMD, Hygon, Zhaoxin, Kunpeng, and Phytium.

5.8.4 GPU Compatibility

NVIDIA GPUs: M10, P40, M60, Tesla T4, V100, V100S, Quadro RTX 6000, P100, GeForce 2080Ti, and GeForce 3090

Made-in-China GPUs: Moore Thread S2000, Huawei Atlas 300v Pro, Cambricon MLU-270-S4, and Cambricon MLU-370-S4

5.8.5 Host OS Compatibility

Microsoft, SUSE Linux, openSUSE, RedFlag Asianux, RedHat Linux, CentOS, Ubuntu, Debian, Fedora, Oracle Linux, Red Flag, Kylin, and UOS

NOTICE

1. For compatibility with more software and hardware, see the HCI Compatibility List.

2. The above part describes compatibility for brands/categories. For compatibility with specific models and versions, refer to the HCI Compatibility List.

6 HCI

6.1 Overview

Sangfor HCI functions as a software-defined data center IT infrastructure that seamlessly integrates computing, storage, network, and security resources through virtualization technology. It provides advanced features, including disaster recovery and backup, O&M management, and intelligent monitoring, featuring high stability, high performance, solid security, intelligence, and convenience. Sangfor HCI helps users build a simple, stable, secure, and high-performance IT base to support their digital transformation.

Sangfor HCI is widely adopted across various industries, such as government affairs, finance, healthcare, education, energy, transportation, manufacturing, cultural tourism, and high-tech enterprises. With over 13,000 customers worldwide, more than 60% of them rely on Sangfor HCI to support their core businesses.

Sangfor HCI, with its services, is applicable to different scenarios in IT infrastructure construction, including new data centers, virtualization replacement, private cloud, branch cloud, DR cloud, and managed cloud. Additionally, it offers a range of industry-specific solution packages to better support customers in building the IT service capabilities for their digital transformation.

6.2 Private Cloud Solution

6.2.1 Business Scenarios and Challenges

The digital transformation of enterprises has led to diverse business systems and increasing demands for business experience. This poses challenges to IT infrastructure in terms of computing power, storage capacity, and storage performance. Traditional data centers with complex and fragile infrastructure cannot adapt themselves to rapid changes and updates, hindering new business deployments and changes. Key challenges include:

1. Low Resource Utilization

Businesses deployed on individual physical servers result in low resource utilization, significant waste, and high investment costs.

2. Insufficient Architectural Elasticity

Complex architectures with multiple device types lead to long construction cycles and slow business launch speeds. In times of business surges, IT resources lack the capability to scale elastically.

3. Fragmented Security Construction

Coexistence of security devices from multiple vendors hampers overall security construction and global security state awareness. Cloud platform security and tenant security construction are fragmented.

4. Challenging O&M Management

Multiple types of IT devices with inconsistent models lack a unified O&M management platform. Different applications have their own IT resources, requiring onerous manual O&M.

6.2.2 Solution Architecture and Benefits

A private cloud data center addresses the challenges faced during the customer's digital transformation. It integrates intelligent infrastructure to deliver efficient and agile application architecture, supporting the digital business of enterprises. Sangfor Data Center Solution is based on HCI, with a focus on business application systems. It ensures high reliability and security for traditional, agile, and innovative businesses while offering convenient and efficient O&M management capabilities. The architecture of Sangfor Data Center Solution is shown in the figure below.

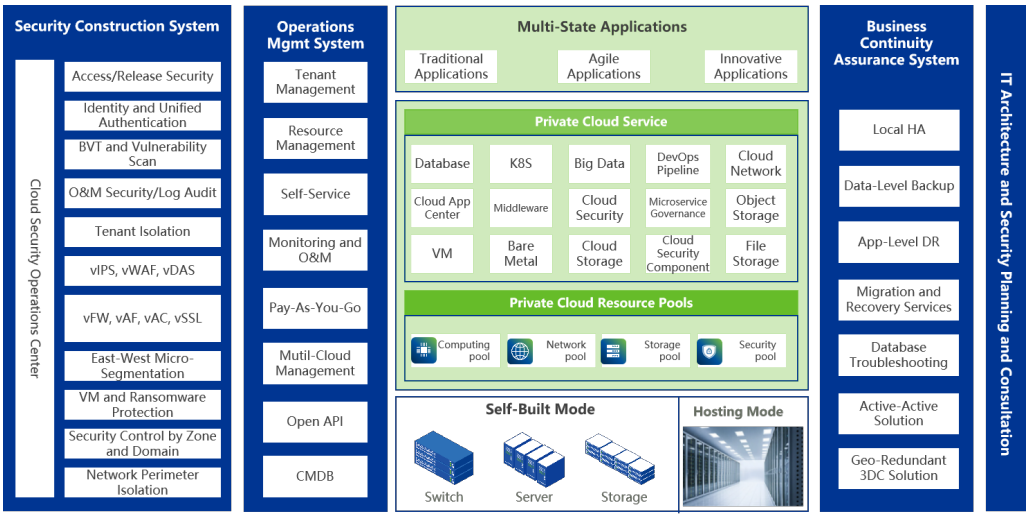


Figure 439: Data Center Solution Architecture

The key benefits and strengths of the Sangfor private cloud data center solution based on HCI are as follows:

1. Security and Reliability

The solution establishes a comprehensive security and reliability framework covering hardware, platform, cloud infrastructure, and network boundaries. It integrates security resource pools to deliver a robust and integrated business and security environment. The cloud platform guarantees the reliability of the hardware, platform, virtualization, and disaster recovery, ensuring uninterrupted business operations.

2. Smoothness and Elasticity

Utilizing a distributed architecture, the solution enables the creation of resource pools with just three cloud servers. Additional resources can be easily added as required by incorporating x86 servers, facilitating seamless and hassle-free scalability. The backup and disaster recovery modules of HCI allow for rapid expansion into a multi-data center disaster recovery and backup mode (geo-redundant 3-DC) through licensing.

3. Saving in Time and Effort

The simplified IT infrastructure, consisting of only switches and x86 servers, streamlines deployment and expansion processes. The solution provides a user-friendly UI with business-centric display, visual O&M, and one-click fault

locating, simplifying O&M tasks. Precise O&M is implemented without the need for training. Additionally, innovative features such as one-click application deployment and What-You-Draw-Is-What-You-Get improve deployment and resource orchestration efficiency.

4. Various Business Capabilities

As the underlying IT infrastructure for business operations, HCI offers a wide range of business capabilities to support both traditional and new applications during the digital transformation. These capabilities span various domains, including industry-specific applications, database services, as well as GPU support, data lakes, big data platforms, and cloud-native capabilities catering to the needs of innovative businesses.

6.2.3 Customer Value

A private cloud data center using Sangfor HCI provides customers with a digital business management platform that facilitates data sharing, business collaboration, and agile innovation. The key benefits include:

1. Simplified Data Center IT Architecture

The cloud infrastructure comprises only switches and x86 servers, making the IT architecture highly streamlined and easy to manage. Scaling is also straightforward, as expanding performance and capacity only requires adding more x86 servers without the need for business or architectural changes.

2. Ensured Stable Business Operations

The solution offers predefined virtualization features such as HA and DRS to minimize the risk of business interruptions caused by physical node failures. Additionally, it provides CDP, load balancing, and disaster recovery technologies for business continuity and accessibility. Data reliability is achieved through distributed cross-node replication and scheduled backups.

3. Enhanced Business Security

The solution provides comprehensive and multi-level cloud security protection, encompassing the cloud platform, tenants, and applications, as well as security protection between VMs and tenants. It ensures the protection of businesses

and customers' data assets.

4. Simplified O&M Management

Sangfor HCI simplifies usage and O&M, allowing limited manpower to focus more on business innovation and deliver better services to customers.

6.3 Disaster Recovery Solution

6.3.1 Business Scenarios and Challenges

As information technology construction becomes more prevalent and advanced, enterprise data centers face escalating risks and threats. Unexpected system interruptions can result in significant economic and reputational losses. Previously, IT resources have transitioned from decentralization to centralization to meet management requirements. Today, informatization typically requires a highly integrated system. However, the high integration means that any accidents can lead to business interruptions and subsequent economic losses. Data loss can also cause irreversible damage to enterprises. To ensure uninterrupted 24/7 operations, a comprehensive business continuity plan is required, providing dual guarantees of high business availability and data reliability.

Nevertheless, this path of informatization construction presents challenges for enterprises, including:

1. Business Interruptions Due to Accidents in Centralized Data Centers

According to IDC statistics, among the companies that experienced disasters in the United States in the 10 years before 2000, 55% went bankrupt at that time. Among the remaining 45%, 29% went bankrupt within two years due to data loss, and only 16% survived.

Gartner Group's statistics indicate that among companies that experienced system shutdowns due to major disasters, two-fifths never recovered, and among the remaining companies, one-third went bankrupt within two years.

2. Business Policy Compliance Requirements

Companies are required to consider business continuity against accidents

during the information construction process, as mandated by national standards, industry standards, and enterprise standards.

3. Business Security Against Threat of Ransomware Attacks

In light of frequent ransomware incidents in the industry, companies need to address the issue of data retrieval after ransomware attacks on their information platforms.

6.3.2 Solution Architecture and Benefits

To support customer business continuity, ensure stable operations, and meet industrial disaster recovery compatibility requirements, Sangfor offers various HCI-based disaster recovery solutions. These solutions include active-standby, active-active, and geo-redundant 3-DC modes, catering to different scenarios. The typical architecture of a disaster recovery and backup solution is shown in the figure below.

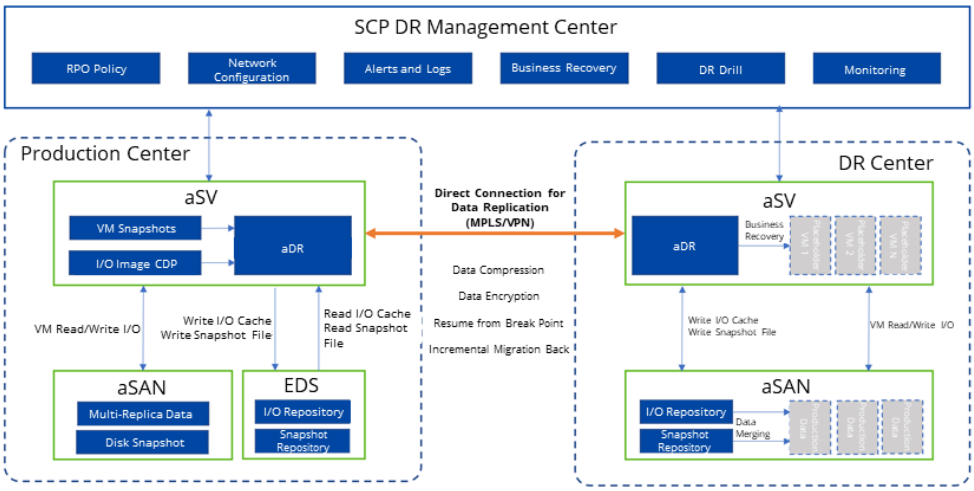


Figure 440: Architecture of the Disaster Recovery Solution

The HCI-based disaster recovery and backup solutions leverage the extensive experience of Sangfor in this field, providing a transparent and visualized disaster recovery process. Customers can easily perform business switching through a user-friendly UI.

The benefits of HCI-based disaster recovery solutions are as follows:

1. Strong Compatibility

They support a wide range of third-party private clouds, public clouds, virtualization platforms, and hyper-converged platforms for disaster recovery and backup in Sangfor HCI.

2. Various Policies

Multiple disaster recovery methods, such as VM syn disaster recovery and CDP, are available. You can select policies that meet specific requirements and can select policies with different time granularity options like second-level, minute-level, and hour-level.

3. Ease of Use

The solutions simplify the disaster recovery process through one-click switching to the target device and recovering the business system in the event of a source device failure. Disaster recovery is performed on HCI for unified O&M management. One-click disaster recovery and the backup and proxy-less mode are supported, eliminating the need for software installation during disaster recovery. The disaster recovery drill does not affect the services of the source device or require configuration changes. It facilitates easy configuration and operation through a wizard-style UI, with automatic network configuration during the recovery. All disaster recovery management and operations can be performed on one platform, eliminating the need to leave the workstation.

6.3.3 Customer Value

1. One-Stop Service in the Entire Lifecycle

The solutions encompass business consulting, solution planning, deployment, drills, business continuity management (BCM), and disaster response. Cloud service providers take on the responsibility of disaster recovery and O&M, allowing users to access disaster recovery services as required.

2. Saving in Time and Effort

The disaster recovery module is integrated with the O&M management platform. The operation steps are simple and clear, requiring no expertise.

3. High Compatibility and a Wide Range of Support

Disaster recovery of any platform, including physical machines, virtualization platforms, and public cloud environments, is supported in Sangfor HCI.

4. Various Disaster Recovery Mechanisms

The disaster recovery management platform provides a range of RTO options, such as 1 second, 1 minute, 5 minutes, and 15 minutes, allowing flexible configuration based on business requirements.

6.4 Database Management Solution

6.4.1 Business Scenarios and Challenges

In the process of enterprise digital transformation, data has become the core asset, highlighting the crucial role of databases. However, traditional self-built databases face various challenges in terms of usage and O&M, including:

1. Complex Database Deployment and Slow Business Launch

The deployment of traditional databases involves an intricate process covering the establishment of hosts, storage, and networks of hardware devices, software installation and parameter configuration. The whole process often takes several days, significantly delaying business launch.

2. Difficult Database O&M Requiring Professional Knowledge

The increasing number and types of databases in digital business demand a wide range of professional knowledge. Hiring or inviting senior database administration (DBA) experts through outsourcing is costly. Additionally, fragmented DBA, repetitive tasks, and manual operation errors make it challenging to ensure the quality of business operations.

3. "Firefighting" Reactive Fault Handling Leading to Poor Real-Time Effectiveness

Without real-time monitoring and alerts, database administrators can only react to issues as they arise. Resolving faults can take a long time, causing significant business interruptions. Moreover, this reactive approach, lacking pre-event prevention and in-event fixing and recovery capabilities, can only be used to handle incidents after they occur.

4. Difficulty in Building a Comprehensive Data Security Protection System and High Data Security Risks

Most intrusion attacks aim to steal or destroy data in databases. General security protection technologies are not enough to provide comprehensive protection, especially against risks such as database permission leakage and data deletion resulting from poor management.

6.4.2 Solution Architecture and Benefits

Sangfor Database Management Platform (DMP) offers solutions for database deployment, usage, and O&M in the cloud computing era. Based on SCP, DMP focuses on enhancing database O&M capabilities. It provides various database services through SCP, including RDS, database migration, database backup, database monitoring, database autonomous O&M, and operation services. These services address the reliability and O&M challenges of databases, meeting the demand for servitization and centralized management.

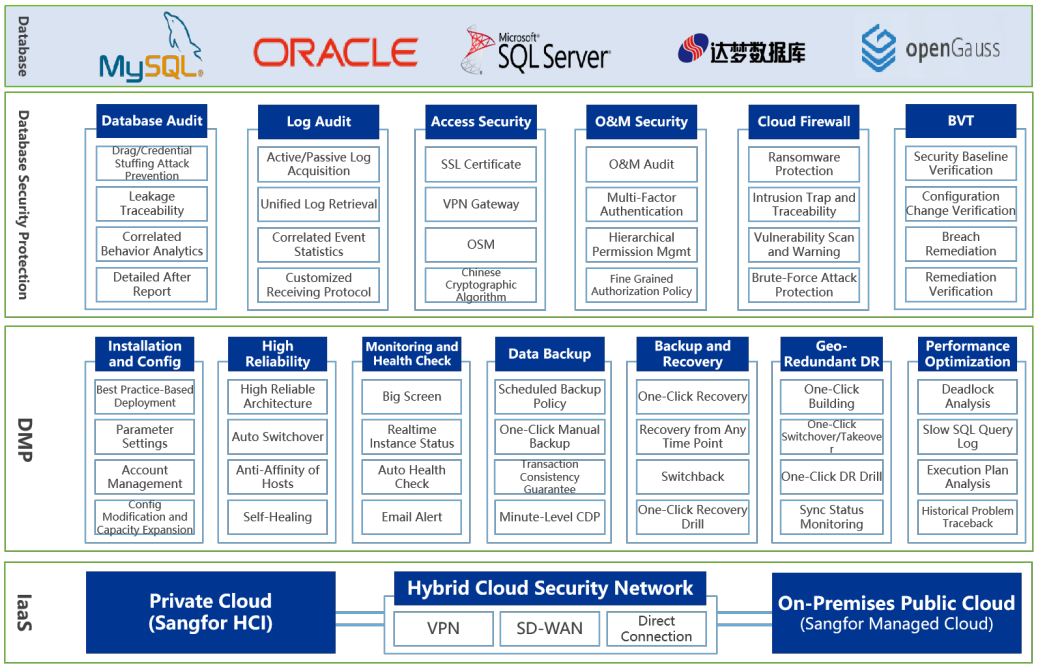


Figure 441: DMP System Architecture

Sangfor DMP offers the following benefits:

1. High Reliability: Proactively Prevents Unexpected Failures

- Provides over 50 monitoring metrics and over 30 inspection rules to prevent failures in advance.
- Realizes end-to-end fault fixing by integrating the database and platform APIs at all layers.
- Offers automatic product O&M combined with real-time analysis and response from cloud experts.

2. High Performance: Locates Root Causes of Performance Issues Effectively

- Utilizes a scalable distributed storage base that allows low-latency access with IOPS of over 500,000.
- Incorporates optimized database best practices to deliver processing performance with million-level transactions per minute (TPM).
- Provides a comprehensive performance monitoring dashboard for visual analysis of resource bottlenecks and root causes of response issues.

3. Easy Management: Features High Cost-Effectiveness and Ease of Use

- Implements integrated delivery and rapid deployment, accelerating the launch by 50 times.
- Delivers 20% higher performance and 40% lower cost compared to traditional storage solutions.

6.4.3 Customer Value

DMP offers full lifecycle management capabilities through services, effectively shielding customers from the complexity of the underlying database. This allows customers to prioritize valuable business innovations instead of database O&M. Additionally, DMP employs tool-based and intelligent full-stack processing of database issues to implement automated O&M policies, reduce manual intervention, and lower O&M costs. The value of the solution includes:

1. Flexible Architecture for Various Scenarios

Supports single-device deployment for development and testing scenarios; enables primary-secondary architecture deployment with host anti-affinity by default to meet HA requirements.

2. Proactive Monitoring and Health Check

Provides monitoring at multiple levels of the database and combines with corresponding health check capabilities to detect problems and risks in a timely manner; gradually shifts from reactive post-event response to proactive pre-event prevention.

3. Reliable Platform with Built-In Cluster Fault Self-Recovery

Deploys database clusters based on best practices, incorporating built-in fault self-recovery capabilities; covers most common fault scenarios of standalone/primary-secondary databases and VMs, reducing O&M manpower and saving costs.

4. Database Backup for Data Security

Achieves database backup based on transaction consistency without relying on third-party software; effectively mitigates the risk of data loss and supports database recovery from any BP; supports immediate verification of backup effectiveness.

6.5 Server Virtualization Solution

6.5.1 Business Scenarios and Challenges

Digital transformation is an inevitable trend in today's business development. As enterprises progress in their digital transformation, the existing virtualized IT infrastructure cannot meet the requirements of business agility and elasticity and faces challenges in maintaining stable business operations and ensuring data asset security. The typical challenges faced by the current virtualized IT infrastructure in users' data centers are as follows:

1. Diversified New Businesses

The digital transformation of businesses has led to diverse demands on information systems, including the need for virtualization to ensure reliable business operations and the growing proportion of storing unstructured data from audio and video media.

2. Scalability of IT Resources

New technologies have resulted in a significant increase in the demand for IT resources. Data set volume has grown from terabyte level to petabyte level, and IT systems now frequently require expansion, reuse, and replacement. This necessitates efficient scalability to implement seamless expansion, extensive reuse, and efficient replacement.

3. System Security and Reliability

Ensuring secure and stable business operations is the primary objective of information system development. Core information systems must provide comprehensive security solutions at the business, data, and infrastructure levels to ensure reliable and stable data center operation.

6.5.2 Solution Architecture

Sangfor server virtualization solution is designed with a modular and scenario-based approach, separating computing and storage. This architecture allows for convenient expansion and agile launch of information systems by achieving loose coupling between software and the underlying physical infrastructure. By utilizing Sangfor server virtualization as the computing platform and integrating computing, network, and security virtualization components, Sangfor provides users with a reliable, stable, secure, and convenient cloud computing platform in terms of business support, cloud-network construction, and security protection. The solution supports Sangfor enterprise-class distributed storage or integration with third-party centralized storage, providing diverse storage capabilities for businesses.

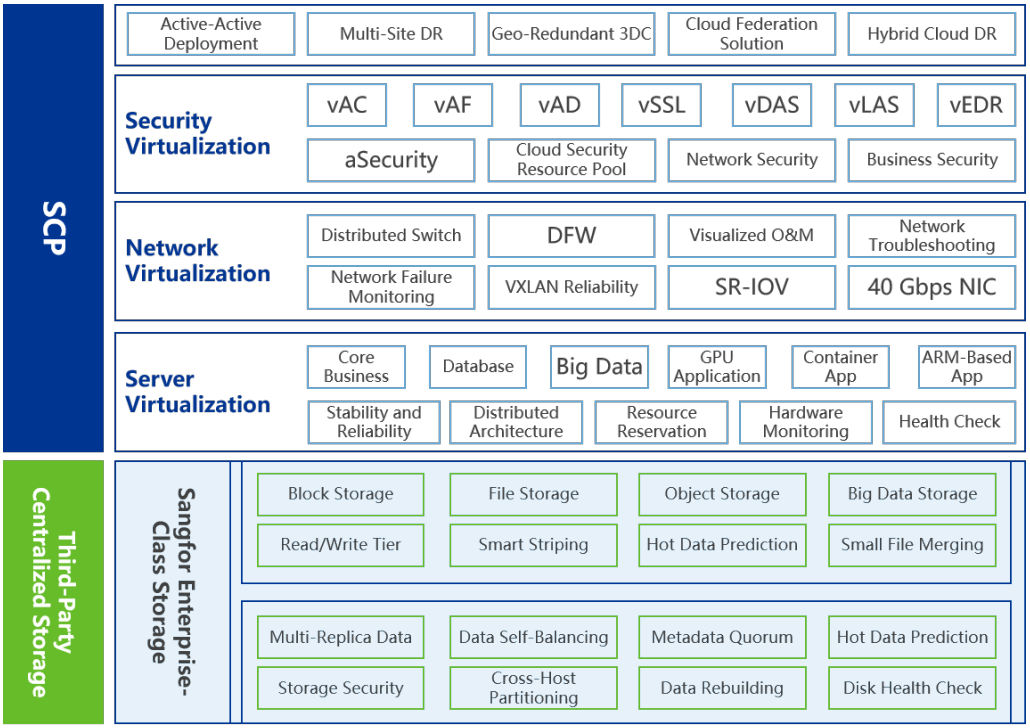


Figure 442: Virtualization Solution Architecture

Highlights of the Sangfor HCI server virtualization solution:

1. Construction of Large-Scale Data Center

Through network connectivity domain technology, the solution implements multi-cluster layer 2 interconnection of multiple resource pools. Different types of resource pools are managed through multiple domains, allowing the creation of hundreds of interconnected domains within a single cloud. Additionally, within the same domain, the solution supports 100 resource pools and 1,024 physical hosts.

2. Low Construction Costs

The integration of computing, network, and security components, together with block, object, and file storage methods supported in a single solution, reduces hardware expenses and maintenance costs for security and storage devices. The solution supports smooth expansion and replacement without interrupting business operations, thereby reducing time costs.

3. Security and Stability

The virtualization platform embeds security capabilities and performs

intelligent correlated analysis and security fixing based on business relationships and load. This ensures secure and stable operations. The platform offers various reliability policies, including hardware isolation, data balancing, resource scheduling, and active-active disaster recovery, enhancing business continuity.

4. Unified Management

The platform provides a unified management portal for IT resources, seamlessly integrating HCI, VMware, and physical machine resources. After consolidation, it delivers services externally, offers simplified operations to meet the unified management requirements for various resources, and implements smooth management transition and visualization of resource indicators.

6.5.3 Customer Value

1. Low Initial Investment Cost

The HCI architecture allows the creation of a lightweight virtualization platform with just three servers, which reduces initial investment costs. By using software-only and hardware decoupling solutions, it allows users to select the IT infrastructure and maximizes the utilization of existing IT equipment.

2. Support for Diverse Businesses

HCI offers not only computing, storage, and network resources at the IaaS layer but also containers, microservices, middleware, GPUs, and data based on the specific IT requirements of business systems. It supports traditional, agile, and innovative businesses throughout enterprises' digital transformation.

3. Saving in Time and Effort

In addition to unified management of computing, storage, network, and security resources, Sangfor HCI provides unified multi-cloud management, What-You-Draw-Is-What-You-Get, network traffic visualization, and business monitoring modules to streamline O&M, freeing up manpower to focus on business innovation.

4. Flexible Scalability and Upgrades

The HCI architecture allows online, seamless scale-out and scale-up, only requiring the purchase of general servers and relevant virtualization software licenses for rapid resource expansion. Additionally, it can easily adapt to the evolving needs of business development, such as multi-cluster deployment, active-active data center, and geo-redundant 3-DC.

6.6 GPU Scenario Solution

6.6.1 Business Scenarios and Challenges

With the rise of the AI era, GPUs are being widely used in various applications, including deep learning, scientific research, medical imaging, video rendering, engineering manufacturing, remote sensing mapping, financial analysis, and weather forecasting. However, due to the high cost, customers often choose to optimize GPU utilization through effective management instead of purchasing a large number of GPUs. Key challenges faced by users during GPU utilization include:

1. Inefficient GPU Server Management

Enterprises often purchase a large number of GPU servers without a unified O&M management platform, lacking cloud platform support for GPU server application, allocation, and deallocation.

2. Lack of Visual Management for GPU Resources

A lack of visual management tools exists in GPU resource management scenarios. Instead, command-line interaction is used for GPU resource management.

3. Low GPU Resource Utilization

Traditional graphics workstation architectures are siloed, with limited sharing capabilities, resulting in inefficient scheduling and allocation of GPU resources and low utilization of device resources.

4. Complex Management and Maintenance

Managing and performing O&M on graphics workstations with multiple heterogeneous devices can be challenging. Different design software requires

different GPU resources, and the underlying architecture lacks flexibility, making it difficult to adapt efficiently. In addition, system expansion suffers from complexity and limited scalability.

6.6.2 Solution Architecture and Benefits

Sangfor HCI is deeply integrated with GPUs, providing an alternative to traditional GPU demand scenarios. It allows customers to quickly build GPU cloud platforms and offers features such as GPU resource quota management, user self-service, and billing. Through extensive customization and development, HCI effectively manages the full lifecycle of GPU resources, significantly improves management efficiency, and supports diverse business operations in various scenarios.

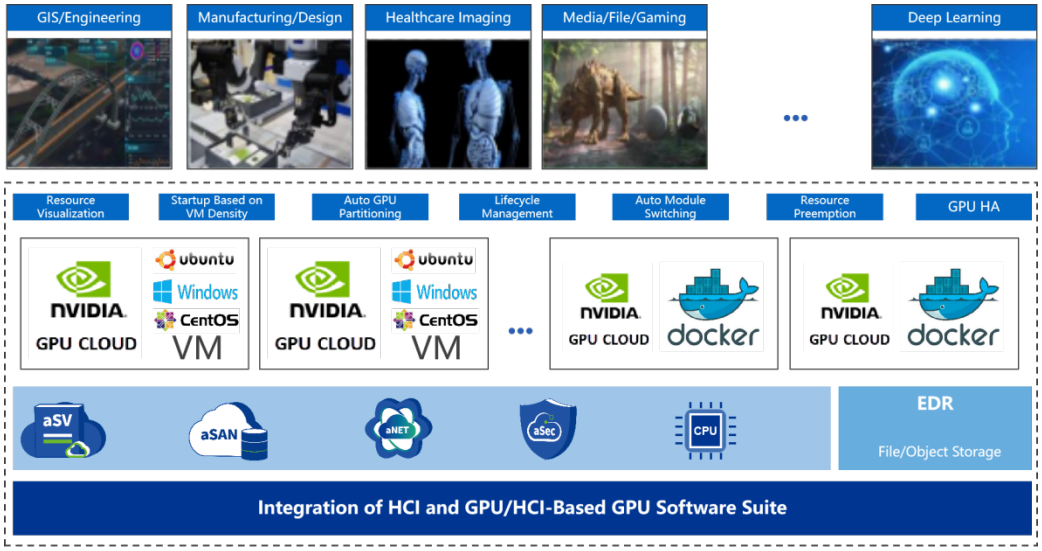


Figure 443: GPU Scenario Solution Architecture

Highlights of the GPU scenario solution based on Sangfor HCI:

1. GPU Resource Pooling

In vGPU mode, GPU resources across the entire cluster can be flexibly scheduled as a unified resource pool. License and hardware resources are automatically released to the resource pool when users shut down.

2. GPU Resource Visualization

Administrators can easily monitor the GPU resource utilization, available GPU

resources, partitioning of each GPU, and VM usage through the control panel.

3. Full-Lifecycle Management of GPU/vGPU Resources

The solution supports full-lifecycle management of GPU/vGPU resources, including resource application, switching, monitoring, and deallocation. All operations can be performed through the HCI web UI.

4. Hybrid Cluster in HA Mode

The solution allows the combination of common and GPU servers to form a cluster, providing HA rules for VMs on the clustered servers to ensure business continuity.

6.6.3 Customer Value

1. Higher Resource Utilization

Through intensive deployment, the solution fully utilizes and schedules GPU resources and allows business departments to quickly build VM and GPU resources that meet the requirements of business applications, improving the overall resource utilization.

2. Various Businesses

GPU pass-through and vGPU modes, as well as dynamic switching between GPU and vGPU modes, are supported. The solution can quickly adapt to the GPU resource usage requirements of different businesses and effectively support various types of customer businesses.

3. Improved Management Efficiency

Administrators can easily view the utilization and availability of GPU resources, which facilitates proper planning and allocation of resources, as well as assessment of resource expansion requirements.

4. Reduced O&M Complexity

Using the GPU cloud platform instead of traditional physical graphics workstations can eliminate the management of scattered and heterogeneous hardware resources and different software versions, effectively reducing O&M complexity and workload.

6.7 VDI Solution

6.7.1 Business Scenarios and Challenges

Virtual Desktop Infrastructure (VDI) is a strategic choice for many enterprises in terms of digital transformation. By leveraging VDI to eliminate reliance on hardware devices, the desktops, applications, and data running on PCs can be centralized and migrated to servers in data centers. This not only solves the challenges of unified startup management of desktops and data security but also provides users with the flexibility and accessibility of desktops required for work, allowing access anytime and anywhere.

However, enterprises also face the following challenges when adopting VDI to achieve goals such as digital office, secure R&D, and data protection:

1. Resource Isolation

VDI is isolated from other businesses in terms of IT infrastructure resources, requiring independent planning, procurement, deployment, and O&M. This leads to resource silos and increases the complexity of architecture management.

2. Resource Waste

During idle time, VDI-specific compute resources, such as GPUs, CPUs, and memory, cannot be efficiently utilized, resulting in lower average resource utilization.

3. Complex O&M

VDI and other business clusters require separate O&M, expansion, upgrade, monitoring, alerting, and problem handling with different tool platforms and teams, making it difficult to improve O&M efficiency.

6.7.2 Solution Architecture and Benefits

Sangfor VDI uses HCI as the underlying architecture, fully leveraging the highlights of HCI in terms of stability, performance, security, and intelligence. It provides a flexible and elastic cloud infrastructure for VDI scenarios, catering to

various business requirements.

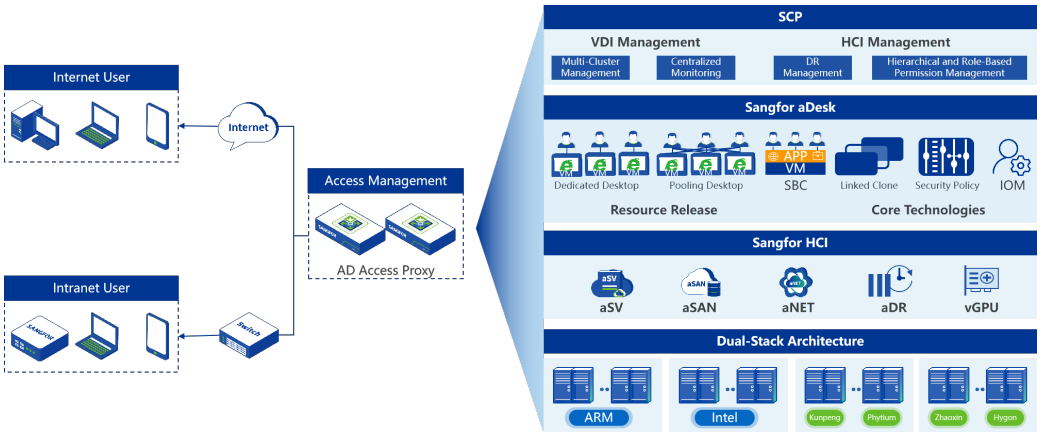


Figure 444: Architecture of VDI and HCI Integration

Highlights of the VDI scenario solution based on Sangfor HCI:

1. Simplified O&M

Cluster resources of VDI and general HCI business VMs can be easily deployed, managed, and operated all through SCP, significantly improving the efficiency of deployment, O&M, and disaster recovery of VDI businesses.

2. Excellent User Experience

- **vGPU**: The first vGPU solution based on the KVM technology in China, which improves the rendering effect of graphic design, provides a better design experience, and delivers performance comparable to graphic workstations. Professional and consumer-class GPU models are supported.
- **Video redirection + hardware chip decoding**: allows users to achieve smooth playback similar to that on a PC while saving server resources. It allows controlled CPU utilization within 10% when multiple videos are played concurrently.
- **SSD cache acceleration**: automatically utilizes SSDs for I/O acceleration of frequently accessed or duplicated data. Sangfor has achieved a cache hit rate of over 60%, resulting in improved desktop startup speed and smoother operations.

3. Stability and Reliability

- **Multi-level HA design mechanism**: utilizes multiple HA mechanisms, including HCI clusters, live migration, dynamic expansion, and port aggregation, to build an HA desktop architecture with enhanced fault tolerance capability.
- **Multi-replica data storage**: real-time storage of data replicas on two or three hosts, ensuring maximum redundancy and eliminating data loss issues due to host or disk failures.

4. Security and Effectiveness

Virtualization security: The industrial first BSI architecture in China, providing VDI businesses with comprehensive capabilities such as WAF, anti-ransomware, and automatic isolation.

6.7.3 Customer Value

Sangfor VDI solution based on HCI helps small and medium-sized enterprises in various industries achieve cost reduction, efficiency improvement, and unified management of data centers. The main benefits include:

1. Flexible Business Support

The VDI business shares an HCI cluster with general business and uses resources isolated through software licensing, allowing resource allocation and scaling as required. This enhances the flexibility and agility of IT infrastructure resource planning and construction.

2. More Reliable Operational Assurance

Key technologies such as ECC, DRS/DRX, and 3-replica mode ensure stability and reliability. Additionally, aDR supports local/geo-redundancy disaster recovery, improving the reliability and continuity of VDI business.

3. More Efficient Operation and Management

SCP centrally manages VDI resources and general clusters, implementing integrated management of resource allocation, VDI license, monitoring, and O&M. It also facilitates hot patch installation, active upgrade, and live migration of clusters, making upgrades and capacity expansion faster and more efficient.

7 Cloud Computing Concepts and Terminology

7.1 General Concepts

1. Cloud Computing

According to the definition by the National Institute of Standards and Technology (NIST), cloud computing is a pay-per-use model that provides convenient, on-demand network access to a shared pool of configurable computing resources, including networks, servers, storage, applications, and services. These resources can be rapidly provided, with minimal management effort or interaction with the service provider.

2. Resource Pool

A resource pool is a logical abstraction of a certain number of flexible and configurable resources, typically referring to the total sum of allocatable computing, storage, network, and other resources. Resource pools can have a hierarchical structure and subordination.

Sangfor resource pool refers to a virtualized environment that combines computing, storage, network, and security virtualization. It emphasizes the supply of infrastructure resources. Based on the resource pools, upper-layer application components such as cloud management and containers can be deployed. Sangfor resource pool encompasses HCI software, various HCI cloud servers, or HCI + active-active disaster recovery for software-defined data centers (SDDCs).

3. Private Cloud

A private cloud refers to cloud computing services provided exclusively to specific users instead of the general public. In addition to features such as network accessibility, elastic and scalable resource allocation, and self-service, the computer system of the private cloud is generally deployed in the customer's own data center for enhanced security. Unlike a public cloud, customers of a private cloud purchase and own the entire computer system.

Sangfor's private cloud solution typically includes HCI and SCP, which encompasses PaaS, service catalog, security system, and aOPS.

4. Public Cloud

A public cloud refers to cloud computing services available to the general public. It is accessible via the Internet, and the computer system is owned by the service provider. Users purchase the specific resources they require, such as compute, storage, and network resources.

5. Managed Cloud

In a managed cloud, the cloud service provider manages the user's compute resources. Unlike a public cloud, a managed cloud offers dedicated compute resources to users, with physical resources being isolated from one another. This approach reduces hardware costs during the informatization and facilitates the development of enterprise users. The service system is tailored for enterprises, allowing them to save one-time expenses and provide value-added services such as live backup and migration, disaster recovery, and fault tolerance.

6. Hybrid Cloud

A hybrid cloud combines both public and private clouds that are tightly integrated through a unified management system. Data can be shared through public Internet or dedicated links, and users can allocate compute, storage, and network resources and other resources between the two clouds based on their requirements.

The cloud is referred to as a hybrid cloud when a customer's IT environment simultaneously incorporates two of the private cloud, managed cloud, and public cloud. Resources are managed and scheduled across these clouds using a unified cloud management platform. A hybrid cloud allows users to leverage the advantages of both public and private clouds. For instance, sensitive data can be processed in the private cloud to ensure data security, while the low-latency feature of the private cloud can provide resources for computing and storage of frequently accessed data. The cost-effectiveness of the public cloud can be utilized for storing non-sensitive cold data, and the scalability of the public cloud can provide temporary resource expansion during peak loads.

7. Branch Cloud

A distributed cloud structure with local clouds built for multiple remote departments that are centrally managed by the headquarters cloud.

8. Edge Cloud

Edge computing, as defined by ISO/IEC JTC 1/SC 38, is a form of distributed computing that places primary processing and data storage on the edge nodes of a network. The Edge Computing Consortium defines it as an open platform that integrates network, computing, storage, and application capabilities at the network edge, close to the physical location or the source of the data. It provides nearby edge intelligence services to meet the key requirements of industry digitization, such as agile connection, real-time business, data optimization, application intelligence, security, and privacy protection.

The definition by the European Telecommunications Standards Institute (ETSI) defines it as providing an IT service environment and computing capabilities at the edge of the mobile network, focusing on proximity to mobile users to reduce latency in network operation and service delivery to improve user experience. With the gradual maturity of the 5G technology, Multi-Access Edge Computing (MEC), also known as Mobile Edge Computing, has become a common concern for upstream and downstream ecosystem partners.

9. Informatization

Informatization is the process of generating new information resources by processing various business processes of an enterprise, such as production processes, material movement, transaction processing, cash flow, and customer transactions, through various information systems and networks. It allows personnel at all levels within the enterprise to have a clear understanding of dynamic business information, such as the current situation of the business and the progress of processes, in order to make decisions that optimize the combination of production factors, allocate resources properly, enhance the enterprise's adaptability, and maximize economic benefits.

10. Digitization

Digitization is the process of using intelligent products, intelligent devices,

sensors, and visual recognition to automatically collect holographic real-time data, forming a digital world in computers. Any change in the real world or the digital world triggers a sync reaction in the other. For enterprises, digitization is based on a large amount of operational data (data recorded by information systems) to mathematically model and optimize the operational logic (management experience) of the enterprise to guide daily operations. This is actually a process of "machine learning," where the system repeatedly learns from the enterprise's data and operational patterns, becoming more professional, better understanding the enterprise, and guiding the enterprise's operations.

11. Smartization

Smartization refers to the capability of things to dynamically meet various requirements by making system decisions and commanding corresponding departments to execute them, with the support of technologies such as the network, big data, IoT, and AI. It typically exhibits one or more of the following characteristics: (1) intelligent OS; (2) intelligent sensing and networking of things; (3) visual recognition and voice interaction; (4) deep learning technology for optimizing the scheduling of social resources such as talent, materials, equipment, warehouses, transportation vehicles, and funds.

12. Digital Transformation

Digital transformation is a strategic approach that utilizes digital technologies (such as big data, cloud computing, and AI) to drive organizations to transform their business models, organizational structures, and corporate culture. It involves building a digital world through digital thinking and the application of digital technologies to innovate and reshape existing business models, operational modes, and corporate culture in order to realize business value.

13. Cyberspace

Cyberspace is a new type of space that is based on computer technology, communication networks, IoT, and other information technologies. Knowledge and information are the carriers. Cyberspace is carried by naturally existing electromagnetic energy. It uses artificial networks as the platform and takes information control as the purpose. It relies on electromagnetic signals to

transmit intangible information and control the behavior of entities, thereby constituting a complex space where the entity, electromagnetic, and virtual layers are intertwined and integrated across domains.

7.2 Cloud Computing Concepts

1. Compute Virtualization

Compute virtualization involves decoupling hardware resources of physical servers from the upper-layer applications through the hypervisor. This creates a unified pool of computing resources that can be elastically allocated to logical VMs for shared use.

Compute virtualization can be understood as the utilization of a VMM to virtualize and run one or more virtual systems on a physical machine. It primarily refers to server virtualization, which involves virtualizing CPU, memory, and other compute resources to improve the utilization of computer hardware resources and enhance IT support efficiency. In the new business environment, GPU virtualization is also becoming increasingly relevant to compute virtualization, related to a Sangfor product called aSV.

2. Storage Virtualization

Storage virtualization is the process of abstracting storage hardware resources and involves abstracting, hiding, or isolating the internal functions of the storage (sub)system or storage services. This separation allows the independent management of storage or data from applications, servers, and network resources. By virtualizing SSD and HDD resources on servers, storage virtualization creates storage resource pools and provides various storage services such as block storage, file storage, and object storage to businesses. Sangfor offers two storage virtualization products: aSAN and EDS.

3. Network Virtualization

It refers to the SDN technology in a broad sense, which allows the creation of logical networks on top of physical or virtual networks. This allows logical network addressing to customize network topology that meets specific user requirements.

Broadly, it encompasses the hardware network (with the OpenFlow protocol primarily), NFV, and Overlay (with the VXLAN protocol). In Sangfor, it primarily refers to the SDN technology implemented using VXLAN. Distributed virtual switches, routers, and other related capabilities rely on this technology. Relevant products of Sangfor include aNET and aSecurity.

4. Security Resource Virtualization

Security resource virtualization involves the use of virtualization technology to implement software-defined hardware security devices. This implements the configuration, management, and monitoring of virtual security devices through a unified management plane. It can transfer security functions from dedicated hardware devices to software that can be flexibly deployed in commercial hardware or cloud networks.

Sangfor offers virtualized secure resource components such as vNGAF, vAD, and vAC.

5. GPU Virtualization

GPU virtualization involves creating a unified pool of physical GPU resources through the hypervisor, which can then be dynamically allocated to logical VMs for shared usage.

Sangfor provides the GPU virtualization solution.

6. Hyper-Convergence

Gartner views hyper-convergence as a software-integrated architecture that allows scale-out. It leverages centrally managed distributed components to apply a modular approach to computing, network, and storage processes on standard hardware.

Hyper-convergence typically integrates computing, network, storage, and virtualization into a single system. The hyper-convergence platform combines hypervisors for virtualized computing, software-defined storage, and virtualized network on standard servers.

Sangfor HCI provides infrastructure services that integrate computing, storage, network, and security virtualization technologies. HCI can function as the underlying platform for private and managed clouds, offering a resource

pooling solution that simplifies installation, deployment, optimization, and O&M processes.

7. Desktop Virtualization

VDI is a method of virtualizing desktops, where the desktop OS, typically Microsoft Windows, runs and is managed in a data center. The desktop image is delivered over the network to endpoint devices, allowing users to interact with the OS and its applications as if they were running locally. Endpoint devices can include traditional PCs, thin clients, or mobile devices.

Sangfor offers a related product called aDesk.

8. IaaS

Infrastructure as a Service (IaaS) is defined by the NIST as providing customers with processing, storage, and network resources together with fundamental computing resources. Customers can run any software, including the OS and applications, on these resources. Customers do not manage or control the underlying cloud infrastructure, but they have control over the OS, storage, and deployed applications and may have limited control over selected networking components.

IaaS is a practice that provides a complete computing stack, including servers, storage, and network resources and operational software. It is a virtualization structure. By eliminating the fixed relationship between hardware and operating software, IaaS architectures can flexibly and quickly scale the data environment to meet workload requirements. Integrated management is often used in IaaS to provide server computing, storage, and network resources, avoiding performance bottlenecks and conflicts. This allows users to simplify their hardware infrastructure and allocate resources according to different business requirements.

9. PaaS

Platform as a Service (PaaS) is defined by the NIST as providing customers with development languages and tools provided by cloud vendors to develop applications and deploy them onto cloud infrastructure. Customers do not manage or control the underlying cloud infrastructure (networks, servers, OS,

or storage devices) but have control over application deployment and runtime environment configurations.

PaaS is a cloud computing model that provides customers with a complete platform for developing, running, and managing applications. It includes the OS, middleware, database, and applications, and customers do not need to build and maintain these platforms locally. PaaS platforms offer virtualized servers and related services to meet the requirements of application design, development, testing, deployment, and management.

10. SaaS

Software as a Service (SaaS) is defined by the NIST as providing customers with applications that are provided by service providers and run them on the cloud infrastructure. These applications can be accessed on various client devices. For example, web emails can be accessed through web browsers using thin clients. Customers do not manage or control the underlying cloud infrastructure (networks, servers, OS, storage devices, or independent applications). In some cases, they have limited control over application configurations.

SaaS is a service model based on cloud computing, which allows users to access and use software over the internet. In SaaS, the service providers manage the applications in their data centers, and customers access these applications through standard web browsers.

11. DRaaS

DRaaS is defined by the NIST as providing productized services to allow vendors to manage the following items:

- Replication of server images and production data to the cloud
- Creation of disaster recovery operation manuals
- Automatic failover and recovery between on-premises and cloud environments
- Network components and functions required during the recovery operation

- Disaster recovery performed using cloud computing resources in the form of services

12. DaaS

Desktop as a Service (DaaS) is a cloud computing service model that abstracts desktop services using virtualization technology. Users can access virtual desktop services through the cloud, which provides an experience similar to a traditional OS desktop.

13. Service Catalog

A service catalog is a document that lists the IT services that your organization provides to its customers. It allows users to easily understand the various capabilities and relationships of a series of products. This allows users to quickly discover the services they need and facilitates product marketing.

14. Subscribe Mode

It is also known as the publish-subscribe mode and is commonly used to handle one-to-many dependencies. When the status of a publisher object changes, an intermediate platform schedules and notifies multiple subscribers. The specific subscription relationship can be implemented by the intermediate platform based on specific scenarios. For example, subscribers may only partially subscribe to the content of a particular publisher, or publishers may only allow certain content to be subscribed to.

15. Message Queue

Message queues are a mechanism used to improve the performance of distributed systems. It is often used to implement event-driven architectures. By buffering message events using a queue data structure and decoupling processing relationships between modules through a publishing-subscription mechanism, message queues can implement efficient asynchronous message processing.

7.3 Database Concepts

1. RDS

Relational Database Service (RDS) is a highly available, reliable, secure, and

scalable online database service provided through cloud platforms. It can help realize database lifecycle management, backup and restore, monitoring, disaster recovery, and rapid scaling, simplifying database operations and allowing users to focus on business development.

Sangfor RDS provides online relational database services through the cloud platform and supports mainstream databases such as Oracle, SQL Server, MySQL, MariaDB, and PostgreSQL. The services include automated deployment, lifecycle management, monitoring and alerting, backup and restore, and online scaling.

2. DBaaS

Database as a Service (DBaaS) is a cloud computing service model that focuses on self-service and convenient management of databases. It allows customers to deploy and manage databases on their own, with pre-configured options, clear metering and charging, and reduced costs for database system deployment and O&M.

3. Cloud-Native Database

A cloud-native database is built, deployed, and distributed through a cloud platform. It is typically installed on cloud infrastructure and provides users with direct access, runtime scalability, and enhanced reliability.

4. Database Services

Database services encompass various O&M management capabilities for databases, including installation, deployment, backup and restore, monitoring and diagnostics, data restoration and migration, troubleshooting, and performance tuning. These services ensure the smooth O&M and management of databases.

5. Database Autonomy

Database autonomy refers to the self-awareness, self-repair, self-optimization, self-O&M, and self-security capabilities that are achieved through machine learning, big data, and expert experience. Database autonomy eliminates complexity and human errors, ensuring the stability, security, and efficiency of database services and improving operational efficiency.

6. Database Migration

Database migration involves transferring data, relational structures, and user permissions from a local database to a cloud-based database. It is crucial to ensure a seamless migration with no data loss and normal business connection to the cloud database after migration.

7. Relational Database

A relational database is a type of database that organizes data using a relational model. The relational model can be understood as a two-dimensional table model consisting of rows and columns. Users can manipulate relational databases using common SQL statements.

8. NoSQL

NoSQL refers to non-relational databases that do not adhere to the atomicity, consistency, isolation, durability (ACID) principles of transactional data storage systems. NoSQL databases have simpler data models and are easily scalable, making them suitable for storing large-scale data without fixed schemas.

9. IMDB

An in-memory database (IMDB) is a type of database where data access and operations are performed throughout the memory. The main database resides in the memory, featuring faster read and write speeds compared to disk-based databases. IMDBs are commonly used for specific data types or analytical applications.

10. Distributed Database

A distributed database system consists of smaller computer systems located in different places. Each computer may have a complete or partial copy of the database management system (DBMS) and its own local database. These computers are connected through a network, forming a complete, globally centralized, and physically distributed large-scale database.

7.4 Cloud Native/Cloud Security/O&M Concepts

1. Cloud Security

External definitions:

1: Refers to a set of programs and technologies used to protect cloud computing environments from internal and external security threats.

2: Involves using cloud-based security capabilities to safeguard end users from security threats.

2. Cloud Native

As defined by the Cloud Native Computing Foundation (CNCF), cloud native technologies empower organizations to build and run scalable applications in public, private, and hybrid clouds. Representative technologies include containers, service meshes, microservices, immutable infrastructure, and declarative APIs. These technologies allow the creation of loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, cloud native technologies allow engineers to easily make high-impact and predictable changes frequently.

3. Cloud Native Security

Cloud native security refers to the inherent security provided by cloud platforms and cloud security products. As an emerging security concept, cloud native security not only addresses the security challenges brought by the widespread adoption of cloud computing but also emphasizes the mindset of building, deploying, and applying security on the cloud. It promotes deep integration between security and cloud computing.

For enterprises, fully integrating security capabilities into cloud migration or selecting a cloud platform with robust security features helps overcome the limitations of traditional security approaches. Cloud-native security incorporates security capabilities into the cloud platform, implementing cloud deployment, data connectivity, and product correlation. It maximizes the utilization of security resources, reduces the cost of security solutions, and

achieves inclusive security.

4. Container

The use of containers is an efficient way to divide the resources of a single OS into isolated groups for better balancing of conflicting resource requirements. Docker is a leading container technology.

5. Docker

Docker is an open-source application container engine that allows developers to package their applications and dependencies into portable images. These images can be deployed on popular Linux or Windows servers and can also be virtualized. Containers are provided with a sandbox environment and have no interfaces for mutual connections.

6. DevOps

DevOps, a combination of Development and Operations, encompasses a set of processes, methods, and systems that facilitate communication, collaboration, and integration of development (applications/software engineering), technical operations, and quality assurance (QA) departments. By automating software delivery and architectural changes, DevOps aims to accelerate software development, testing, and release, making software delivery faster, more frequent, and more reliable.

7. Microservice

Microservice is a software development technique that is a variant of Service-Oriented Architecture (SOA), where applications are built as a set of loosely coupled services. In this architecture, services are fine-grained, and protocols are lightweight. Microservice (or microservice architecture) is a cloud-native architectural approach where individual applications consist of many loosely coupled, independently deployable small components or services. These services have their own stacks, including databases and data models. They communicate through RESTful APIs, event streams, and message brokers. They are organized by business capabilities, with service boundaries referred to as bounded contexts.

Microservice is a software development architecture and organizational

approach that breaks down software into small, independent services communicating through lightweight APIs. Each service can be individually built and deployed, facilitating faster and more scalable software development.

8. Microservice Governance

Microservice is a distributed architecture pattern based on SOA, consisting of lightweight and loosely coupled services. Each service implements a single business function. Microservice governance is part of service governance, which involves adopting and implementing SOA through recognized practices, principles, and government regulations. SOA governance ensures optimal service quality, consistency, predictability, and performance. It enforces prescribed policies and fixes system issues or policy violations. SOA governance comprises registry, policies, and testing processes.

9. OPS

OPS is short for operations. OPS engineers are technical operations and maintenance personnel. They are responsible for planning information, networks, and services as required. They ensure long-term stability and availability of services through activities such as network monitoring, event alerts, business scheduling, troubleshooting, and upgrades.

Sangfor aOPS simplifies O&M in SCP.

10. CMDB

A configuration management database (CMDB) is a standardized database that contains relevant information about the hardware and software components used in IT services, along with their relationships. It provides an organized view of configuration data and allows examination from different perspectives.

11. Service Mesh

A service mesh is a configurable, low-latency infrastructure layer that handles a high volume of network-based interprocess communication among application services using APIs. It ensures fast, reliable, and secure communication among containerized and ephemeral application infrastructure services. Key functions it has provided include service discovery, load balancing, encryption, observability, traceability, authentication, licensing, and support for circuit

breaker patterns.

A service mesh is typically implemented by using a proxy instance for each service instance. The proxy instance handles communication, monitoring, and security. This allows developers to focus on the development, support, and O&M of application code, while the O&M team maintains the service mesh and runs applications.

7.5 Emerging Technology Concepts

1. AI

AI broadly refers to the theory and development of computer systems that are able to simulate human cognitive processes and intelligent behaviors, such as learning, reasoning, thinking, and planning. It is a field of computer science focused on developing intelligent machines that can mimic human intelligence, enabling computers to perform a variety of advanced functions. It encompasses fields like machine learning and computer vision.

2. AI Platform

An AI platform is a collection of software tools that supports AI algorithm modeling and development and data analysis. It includes components such as deep learning frameworks (TensorFlow, PyTorch, MXNet), machine learning frameworks (Scikit-Learn, Dask, Pandas, NumPy), resource and task scheduling software (Kubernetes, Docker), and service deployment management software (Kubeflow, SpringBoot).

3. AI Application

AI has numerous applications in various fields, including computer science, finance, medicine, diagnostics, heavy industry, transportation, remote communication, online and telephone services, law, scientific discovery, toys and games, music, etc.

4. Big Data

In the *Big Data: A Revolution That Will Transform How We Live, Work, and Think* by Viktor Mayer-Schonberger and Kenneth Cukier, big data refers to the analysis and processing of all data instead of using random sampling methods (such as

surveys) as a shortcut. IBM mentions that the five V's of big data (velocity, volume, value, variety, and veracity) are the five main characteristics of big data.

5. Big Data Platform

A big data platform is an infrastructure designed to handle tasks such as mass data storage and computing and real-time computing of uninterrupted data flow. It refers to a platform capable of conducting big data analysis.

Sangfor aBDI is a development and operating platform that allows big data application analysis.

6. Big Data Application

Advanced big data techniques can be used to collect, store, and analyze massive and diverse data sets to uncover new insights, drive innovation, and optimize capabilities. It also refers to analytical software developed for specific business scenarios or industries and built on top of a big data platform. It independently models or shares models, generates and presents analysis results, and provides data-driven decisions.

7. Data Governance

According to DAMA International, data governance is the exercise of authority and control over the management of data assets.

The Data Governance Institute (DGI) defines data governance as a system of decision rights and accountabilities for information-related processes, executed according to agreed-upon models that describe who can take what actions with what information, and when, under what circumstances, using what methods.

8. BI

BI, short for Business Intelligence, is a comprehensive solution that integrates existing enterprise data, implements fast and accurate reporting, and provides decision-making support for business operations.

9. IoT

The Internet of Things (IoT) connects various objects to the Internet through sensing and communication technologies and enables remote monitoring,

automated alerts, control, diagnostics, and maintenance. It realizes the integration of management, control, and operations in a networked environment.

Sangfor has established Sundray as its brand company to offer IoT solutions.

10. Cloud-Data Integration

Big data and cloud computing are integrated, with a complete technology stack. The integrated platform features full-process delivery, security, and reliability.

Sangfor integrates the aBDI big data analysis platform with SCP to provide delivery capabilities through services based on cloud-data integration.

7.6 Sangfor Cloud Terms

Name	Term	Description
Sangfor cloud	Sangfor Cloud	Sangfor Cloud is the cloud computing brand of Sangfor, encompassing all of Sangfor's cloud computing services. It offers a comprehensive range of services for building cloud-based data centers, including virtualization, HCI, and SCP. Sangfor Cloud provides a complete set of services, such as IaaS, PaaS, security, and O&M. Additionally, it offers a comprehensive range of tailored SDDC solutions, covering scenarios such as VMware vSphere replacement, GPU business cloud, business continuity, Arm cloud construction, cloud data center, managed cloud, and container cloud construction. These solutions empower the evolution of user data centers to cloud data centers.
Sangfor Cloud Platform	SCP	SCP is capable of managing HCI, VMware, Alibaba Cloud, and physical servers. It provides functions such as reliability center, security center, and monitoring center.
Sangfor hyper-converged infrastructure	HCI	Sangfor HCI running on commercial server hardware is a foundational infrastructure service that integrates compute virtualization, storage virtualization, network virtualization, and security virtualization in a software-defined and distributed architecture. HCI can function as the underlying platform for private and managed clouds, offering a resource pooling solution that simplifies installation, deployment, optimization, and O&M processes. Sangfor HCI includes compute virtualization (aSV), storage virtualization (aSAN), network virtualization (aNET), and security virtualization (aSecurity).
Sangfor	aSV	Sangfor aSV virtualizes CPUs, memory, and I/O devices

compute virtualization		through the VMM.
Sangfor storage virtualization	aSAN	aSAN is an exclusive distributed storage system that integrates distributed caching, SSD caching acceleration, global application delivery, and multiple data protection technologies. It ensures efficient and stable operation of critical businesses.
Sangfor network virtualization	aNET	aNET is an exclusive network virtualization layer that provides a visual representation (What-You-Draw-Is-What-You-Get) of business logic. It supports the deployment of virtual routers, virtual switches, and virtual network security devices.
Sangfor security virtualization	aSecurity	aSecurity includes virtual security components such as vNGAF, vAC, vAD, and SSL VPN.

Table 21: Sangfor Cloud Terms

