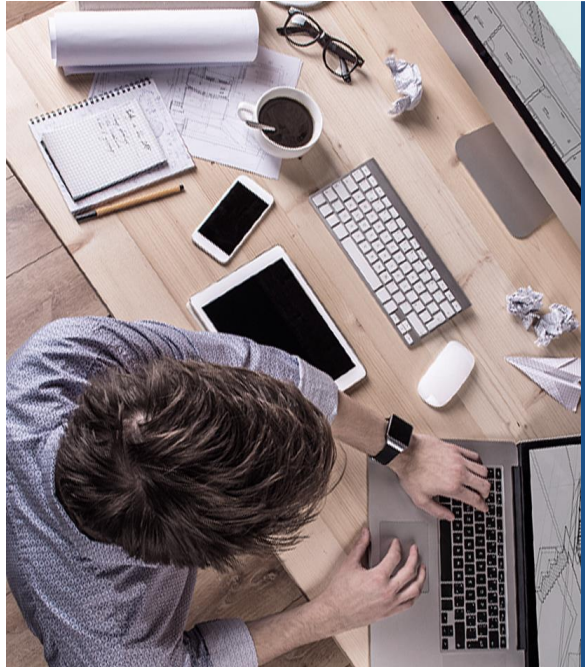




Sangfor HCI 6.3.0

Principle of vmtools





- 1 Uses of vmtools
- 2 Operating Systems supported by vmtools
- 3 Principle of vmtools
- 4 Installation failure case

1. Uses of vmtools



SANGFOR
深信服科技

- **Improve the read/write IO of the network port and disk of the virtual machine**

After installing the vmtools tool, the IO capability of the virtual machine to the virtual disk image can be greatly improved, and reduce the performance loss of virtual machines accessing storage through the virtualization layer.

- **Report virtual machine information to the HCI platform, such as CPU, memory, IP address and other information**

After the vmtools tool is installed, information such as CPU, memory, disk, and IP of the virtual machine can be transmitted. For virtual machines without vmtool installed, the information displayed on the page is mainly displayed on the virtual machine details page, which is not transmitted through the virtual machine, but transmitted by qemu.

- **Special optimization functions: hot add resource, network connectivity detection, etc**

After installing the vmtools tool, you can directly verify the communication between virtual machines or between virtual machines and external networks on the virtual network page. Hot-adding of virtual machine resources can only be supported after the vmtools tool is installed.

2. Systems supported by vmtools



SANGFOR
深信服科技

Systems supported by vmtools

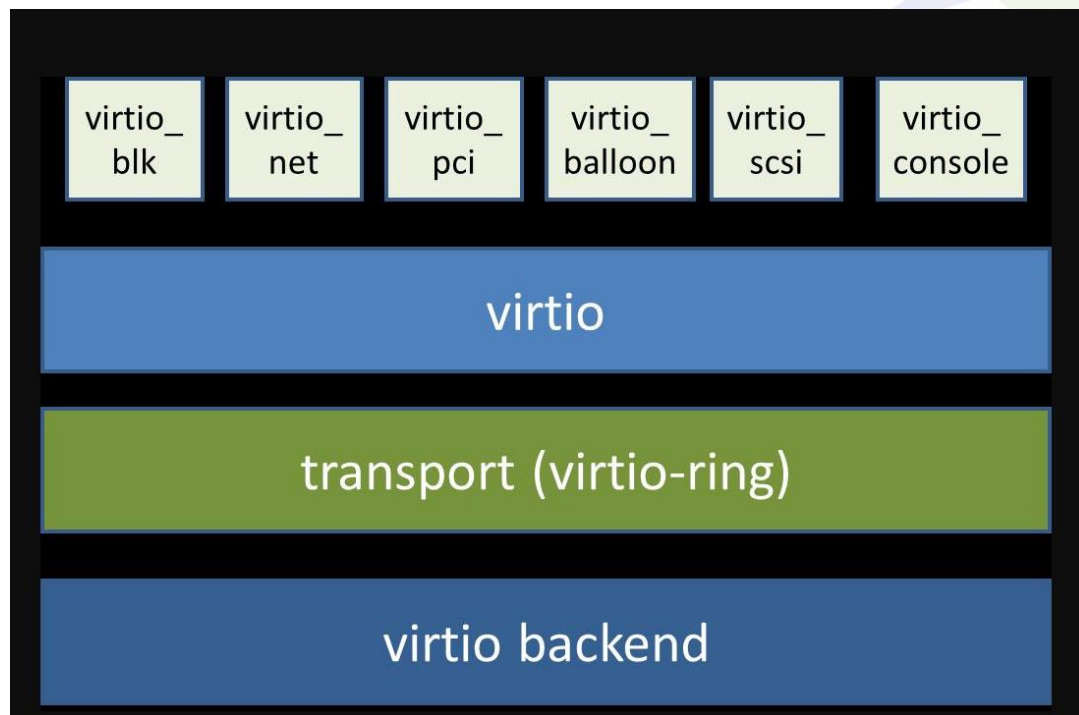
OS Type	Windows	Linux	Unix
Supported OS	Windows Server: Windows2003SP until Windows2019 Windows Personal: XPsp3 until Windows10	Kernel greater than 2.6.25	Unix system installation is not supported, such as: freeBSD

3. Principle of vmtools



SANGFOR
深信服科技

Principle of vmtools



- vmtool works based on a paravirtualized driver (virtio). The way to get it is by installing the vmtools on the virtual machine.
- Virtio exists in the HCI with various types of device drivers: virtio_blk, virtio_net, virtio_balloon, etc. For example:
 - virtio_blk optimizes device drivers for virtio for virtual disks.
 - virtio_net optimizes device drivers for virtio for virtual network cards.
 - virtio_balloon optimizes device drivers for virtio with memory bubbles.
- The optimization idea for Virtio is to move some of the hardware simulated by Qemu down to the Hypervisor, that is the Linux Kernel, to shorten its IO path, reduce the number of context switches, and improve performance several times.
- Through optimization, the IO and throughput performance of virtual disks and virtual network cards can be greatly improved, and the memory bubble function can be used for rational utilization and recycling of memory.

4. Installation failure case



SANGFOR
深信服科技

Installation failure case



■ Problem Description

The migrated Linux virtual machine, after mounting the vmtools, executes `sudo ./install.sh` to install and reports an error:

initrd file not find!

Update rd file failed!

Current system is not support virtio

At the same time, the interface prompts that the current operating system does not support the installation of performance vmtools.

■ Handling process

1. The overall method is to first check whether there are paravirtualized drivers in the virtual machine, mainly virtio_net (network card) virtio_blk (disk) virtio_console (for reporting).

2. Check whether the virtual machine has a paravirtual driver virtio

`find /usr/lib/ -name *virtio*` Normally it is in the /usr/lib/ directory

`find /lib/ -name *virtio*`

If there are no drivers mentioned in 1, it is not supported.

```
root@CR4eb ~# cat /proc/version
Linux version 3.10.0-693.el7.x86_64 (builder@builder.dev.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)) #1
SMP Tue Aug 22 21:09:27 UTC 2017
root@CR4eb ~# find /usr/lib -name *virtio*
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/block/virtio_blk.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/char/hu_random/virtio_rng.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/char/virtio_console.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/gpu/drm/virtio
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/gpu/drm/virtio/virtio_gpu.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/net/virtio_net.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/scsi/virtio_scsi.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio/virtio.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio/virtio_balloon.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio/virtio_input.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio/virtio_pci.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/drivers/virtio/virtio_ring.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/net/vmw_vsock/vsock_virtio_transport.ko.xz
/usr/lib/modules/3.10.0-693.el7.x86_64/kernel/net/vmw_vsock/vsock_virtio_transport_common.ko.xz
root@CR4eb ~#
```

Hardware Requirements



3. Confirm that the driver viewed in 2 exists, and then check whether the semi-virtual driver is loaded in the rd file of the kernel.

Execute `lsinitrd /boot/initramfs-$(uname -r).img | grep virtio`

`$ (uname -r)` It is to add the result of the `uname -r` command execution to the above position.

4. If the driver is not loaded in rd, make a new rd file with driver, first backup the existing rd file `initramfs-$(uname -r).img`,

`cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak`

Then execute the following command:

`mkinitrd -f --with=virtio_pci --with=virtio_blk --with=virtio_console /boot/initramfs-$(uname -r).img $(uname -r)`

Recreate the rd file, restart and install `vmtool` after successful execution.

Thank you !

tech.support@sangfor.com

community.sangfor.com



SANGFOR

Your Future-Proof IT Enabler

Sangfor Technologies (Headquarters)

Block A1, Nanshan iPark, No.1001
Xueyuan Road, Nanshan District,
Shenzhen, Guangdong Province,
P. R. China (518055)

