

创建虚拟机



概述

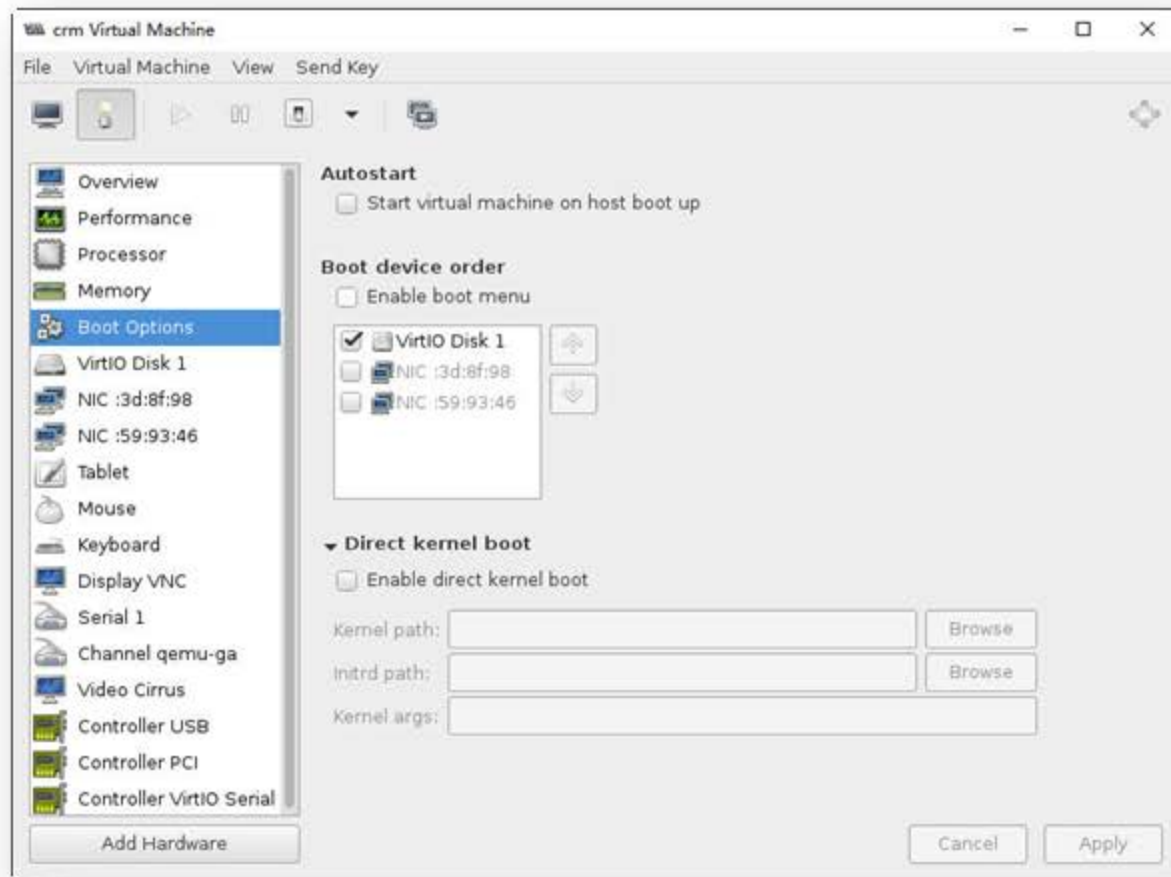
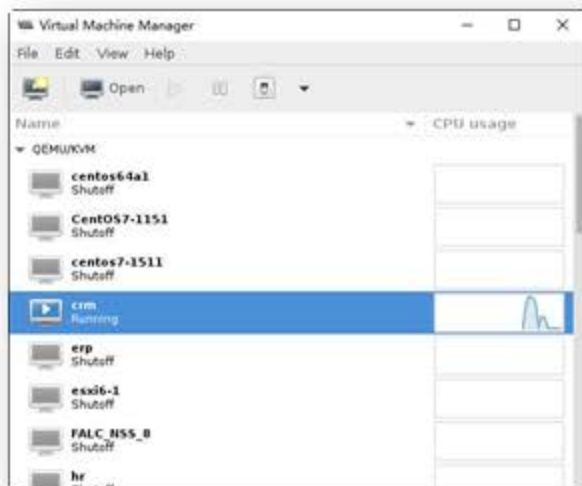
- ▶ 使用virt-manager创建虚拟机
- ▶ 使用virt-install创建虚拟机
- ▶ 半虚拟化驱动virtio
- ▶ QEMU Guest Agent和SPICE Agent

◆ 使用virt-manager创建虚拟机

▶ virt-manager基本使用

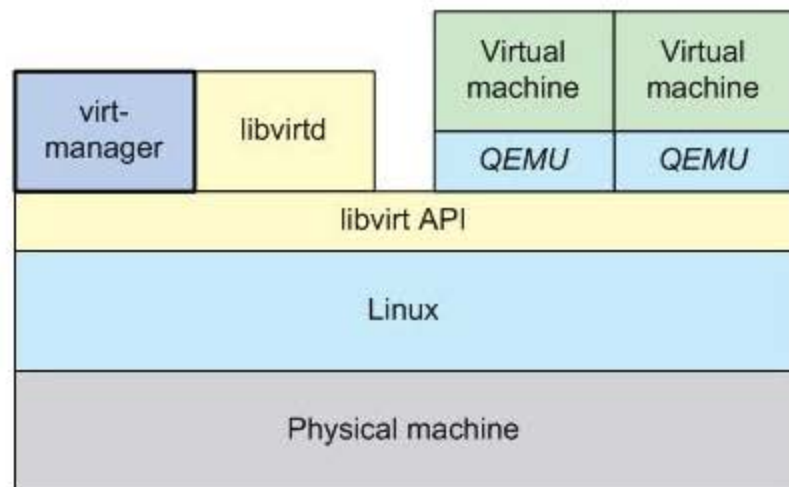
▶ 实验

- ▶ 环境准备
- ▶ 创建Windows 虚拟机
- ▶ 创建Linux虚拟机



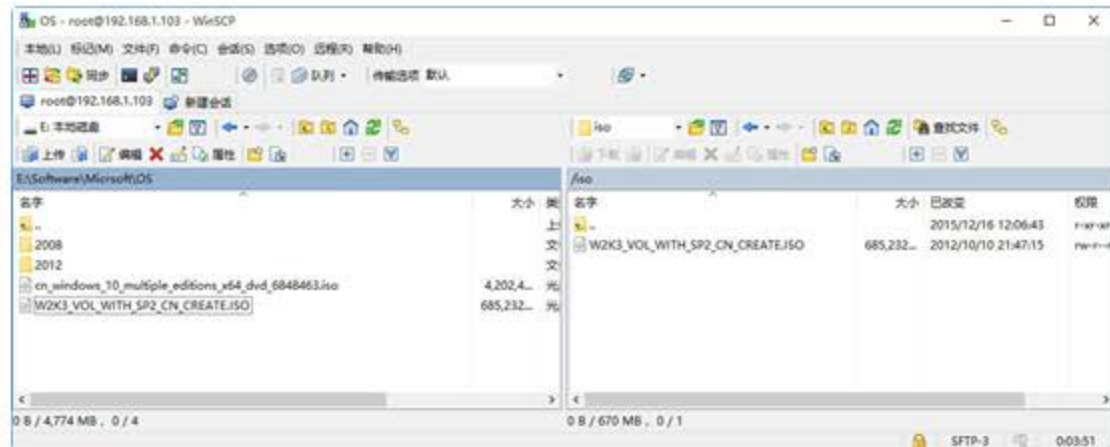
virt-manager基本使用

- ▶ 启用virt-manager
- ▶ 虚拟机管理主窗口
- ▶ 硬件细节窗口
 - ▶ 配置虚拟机启动选项
 - ▶ 附加USB设备给虚拟机
 - ▶ USB重定向
- ▶ 虚拟机图形控制台
- ▶ 添加远程连接
- ▶ 显示虚拟机细节
- ▶ 性能监视



实验的环境准备

▶ 操作系统安装介质、ISO文件



▶ 磁盘空间准备

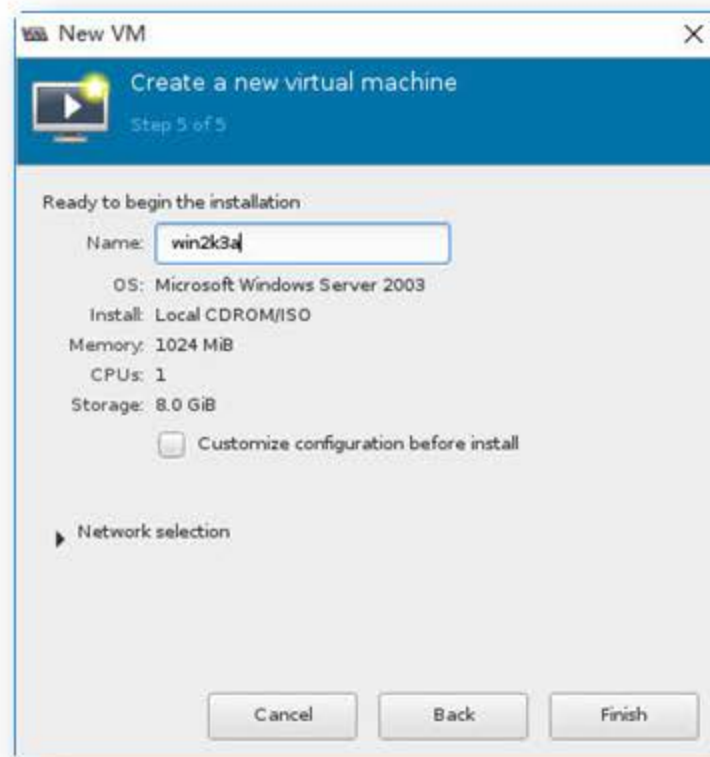
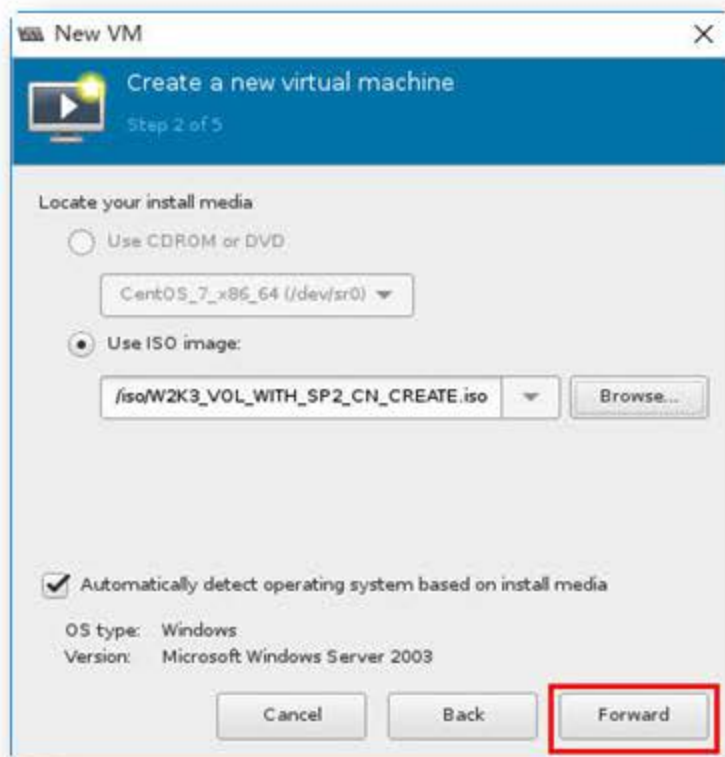
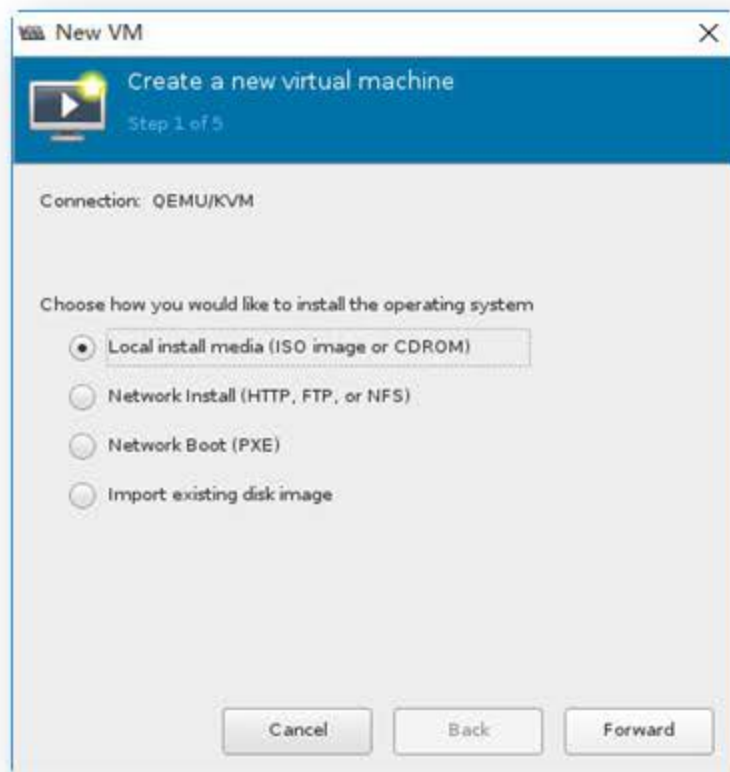
- ▶ 添加新磁盘
- ▶ 创建LV
- ▶ 创建文件系统及挂载点

```
# pvcreate /dev/sdb
# vgcreate vmvg /dev/sdb
# lvcreate -n lvvm1 -l 20479 vmvg
# mkfs.ext4 /dev/vmvg/lvvm1
```

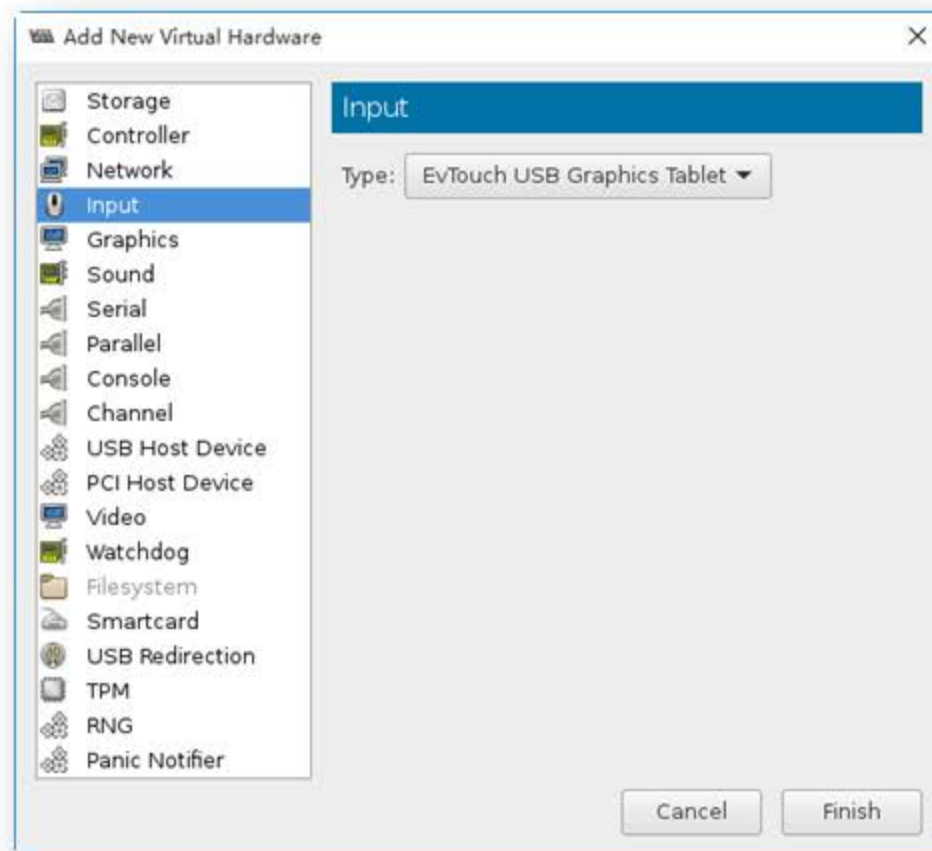
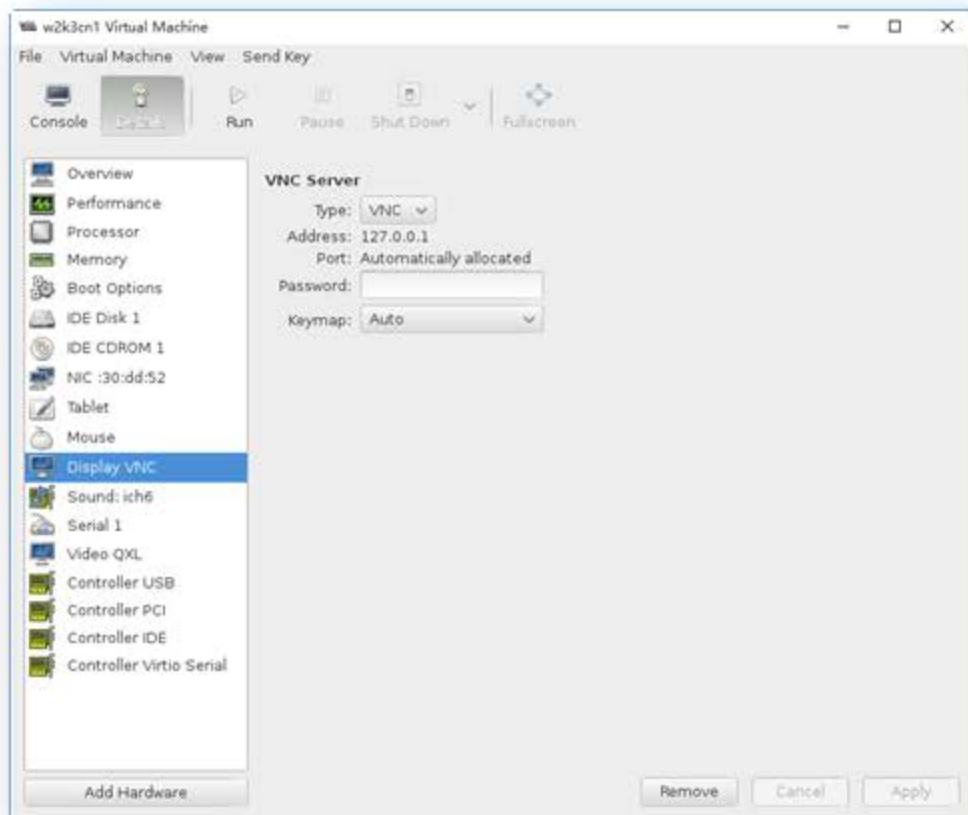
◆ 创建Windows 虚拟机

- ▶ 使用向导的默认配置来创建虚拟机
- ▶ 最佳策略：Spice与VNC、Tablet设备
- ▶ 安装操作系统
- ▶ 考察虚拟机的配置

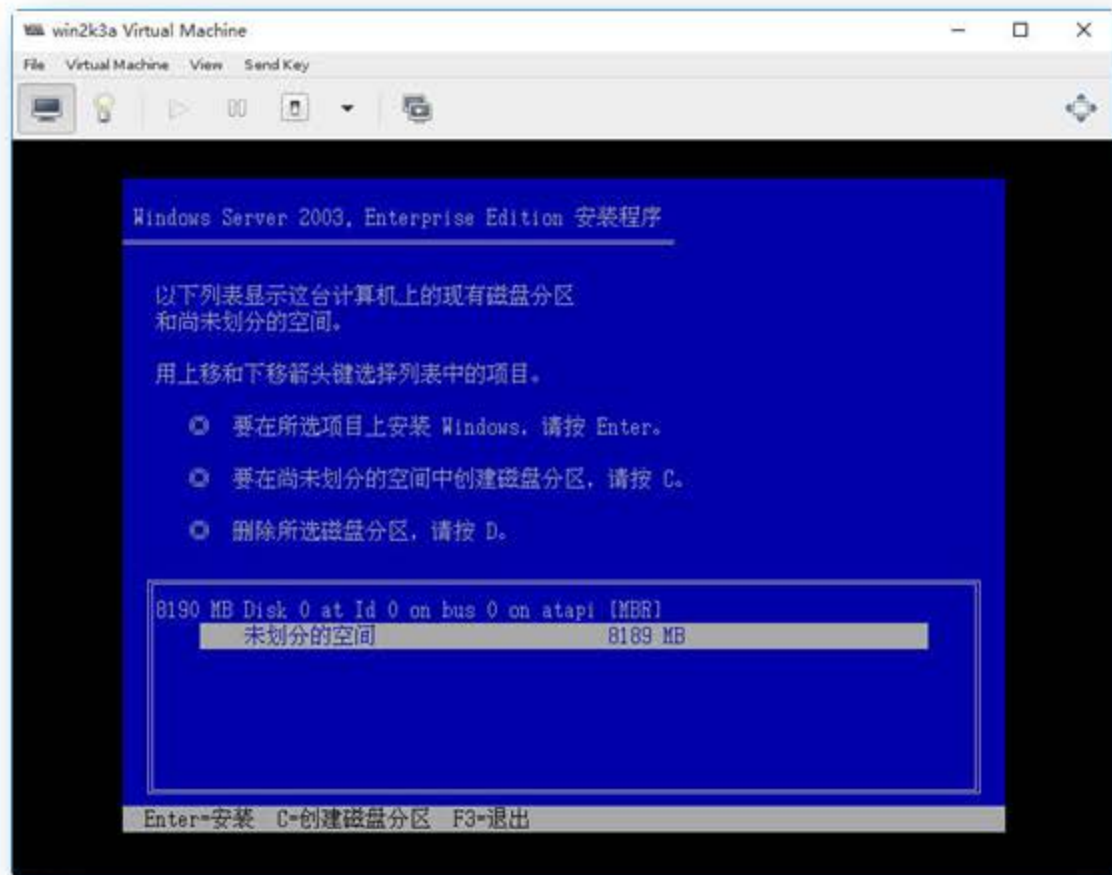
使用向导的默认配置来创建虚拟机



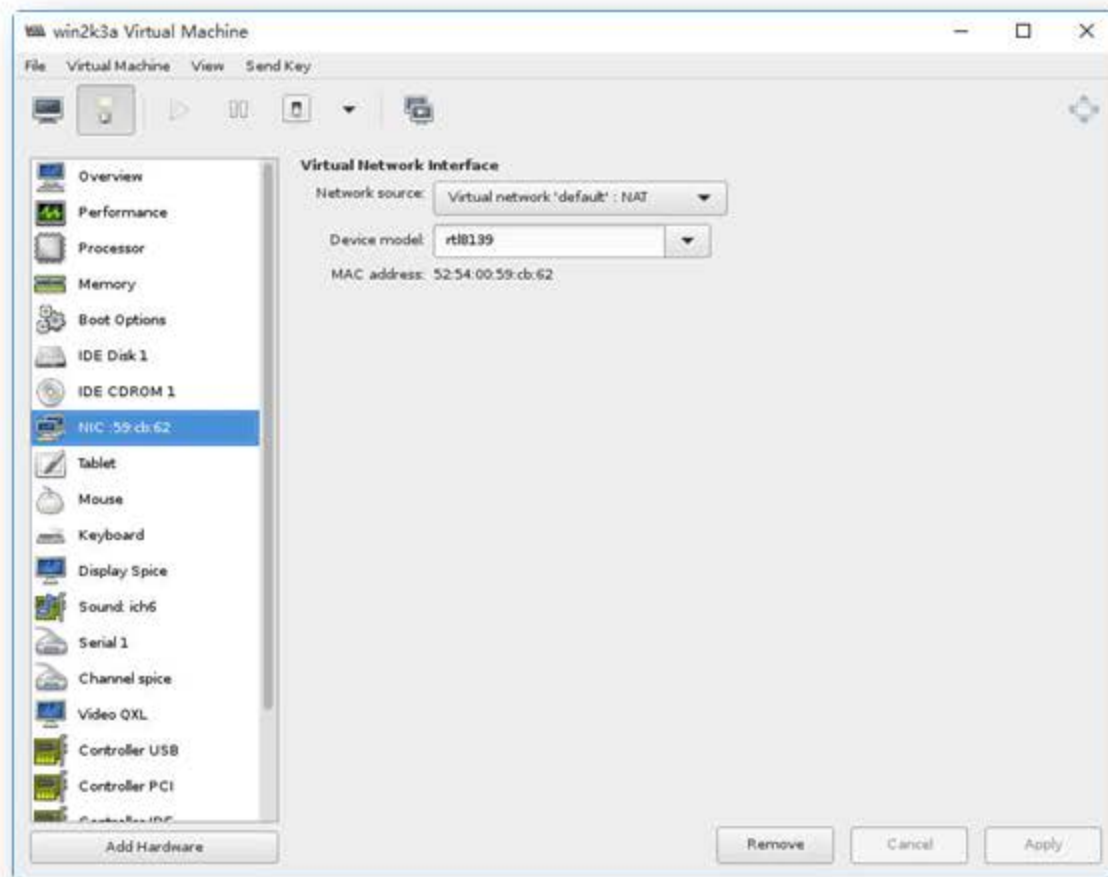
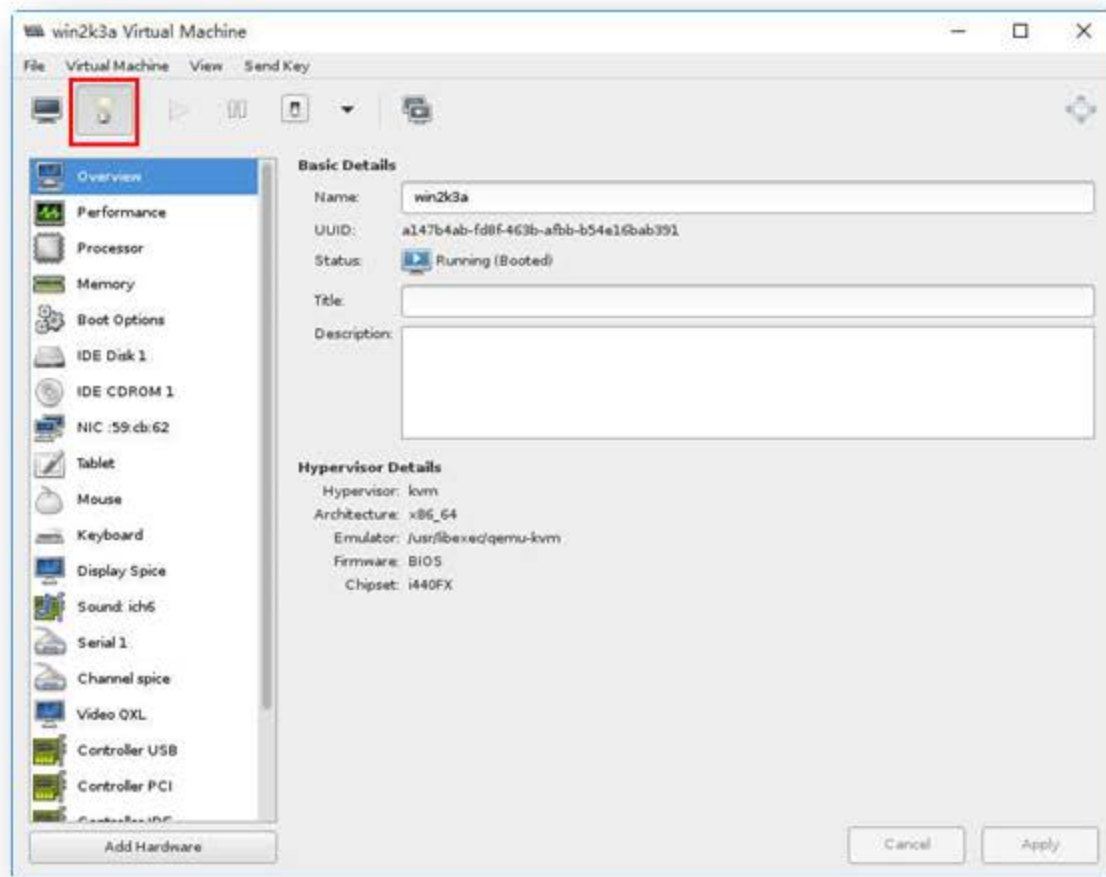
最佳策略：Spice与VNC、Tablet设备



安装操作系统



考察虚拟机的配置



创建Linux虚拟机



```
# 最小配置的VM, 用于测试
#
install
cdrom
text
lang en_US.UTF-8
keyboard us
network --onboot yes --device eth0 --bootproto dhcp --noipv6
rootpw 123456
firewall --disabled
authconfig --enablesshadow --passalgo=sha512
selinux --disabled
timezone --utc Asia/Shanghai
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb quiet"
reboot

$packages
@base
@core
$end
```

◆ 使用virt-install创建虚拟机

- ▶ 命令参数参考
- ▶ 通过本地ISO文件来进行安装
- ▶ Kickstart安装
- ▶ 网络安装
- ▶ PXE安装

```
[root@zzkvm1 ~]# whatis virt-install
virt-install (1)      - provision new virtual machines
[root@zzkvm1 ~]# virt-install --help
usage: virt-install --name NAME --ram RAM STORAGE INSTALL [options]

Create a new virtual machine from specified install media.

optional arguments:
  -h, --help                show this help message and exit
  --version                  show program's version number and exit
  --connect URI              Connect to hypervisor with libvirt URI

General options:
  -n NAME, --name NAME      Name of the guest instance
  --memory MEMORY            Configure guest memory allocation. Ex:
                              --memory 1024 (in MiB)
                              --memory 512,maxmemory=1024
  --vcpus VCPUS              Number of vcpus to configure for your guest. Ex:
                              --vcpus 5
                              --vcpus 5,maxcpus=10,cpuset=1-4,6,8
                              --vcpus sockets=2,cores=4,threads=2,
  --cpu CPU                  CPU model and features. Ex:
                              --cpu coreduo,+x2apic
                              --cpu host
  --metadata METADATA        Configure guest metadata. Ex:
                              --metadata name=foo,title="My pretty title",uuid=...
                              --metadata description="My nice long description"

Installation Method options:
  --cdrom CDROM              CD-ROM installation media
  -l LOCATION, --location LOCATION
                              Installation source (eg, nfs:host:/path,
                              http://host/path, ftp://host/path)
  --pxe                      Boot from the network using the PXE protocol
  --import                   Build guest around an existing disk image
  --livecd                   Treat the CD-ROM media as a Live CD
  -x EXTRA_ARGS, --extra-args EXTRA_ARGS
                              Additional arguments to pass to the install kernel
                              booted from --location
```


命令参数参考 1/3

- name name: 设定虚拟机名称;
- M machine: 指定要模拟的主机类型, 如Standard PC、ISA-only PC或Intel-Mac等
- m megs: 设定虚拟机的RAM大小;
- cpu model: 设定CPU模型, 如coreduo、qemu64等
- smp n[,cores=cores][,threads=threads][,sockets=sockets] [,maxcpus=maxcpus]: 设定模拟的SMP架构中CPU的个数等、每个CPU的核心数及CPU的socket数目等; PC机上最多可以模拟255颗CPU; maxcpus用于指定热插入的CPU个数上限;
- numa opts: 指定模拟多节点的numa设备;
- fda file:
- fdb file: 使用指定文件(file)作为软盘镜像, file为/dev/fd0表示使用物理软驱;
- hda file -hdb file -hdc file
- hdd file: 使用指定file作为硬盘镜像;
- cdrom file: 使用指定file作为CD-ROM镜像, 需要注意的是-cdrom和-hdc不能同时使用; 将file指定为/dev/cdrom可以直接使用物理光驱;

命令参数参考 2/3

-drive option[,option[,option[...]]]: 定义一个硬盘设备；可用子选项有很多。

file=/path/to/somefile: 硬件映像文件路径；

if=interface: 指定硬盘设备所连接的接口类型，即控制器类型，如ide、scsi、sd、mtd、floppy、pflash及virtio等；

index=index: 设定同一种控制器类型中不同设备的索引号，即标识号；

media=media: 定义介质类型为硬盘(disk)还是光盘(cdrom)；

snapshot=snapshot: 指定当前硬盘设备是否支持快照功能：on或off；

cache=cache: 定义如何使用物理机缓存来访问块数据，其可用值有none、writeback、unsafe和writethrough四个；

format=format: 指定映像文件的格式，具体格式可参见qemu-img命令；

-boot [order=drives][,once=drives][,menu=on|off]: 定义启动设备的引导次序，每种设备使用一个字符表示；不同的架构所支持的设备及其表示字符不尽相同，在x86 PC架构上，a、b表示软驱、c表示第一块硬盘，d表示第一个光驱设备，n-p表示网络适配器；默认为硬盘设备；

-boot order=dc,once=d

命令参数参考 3/3

`-net nic[,vlan=n][,macaddr=mac][,model=type][,name=name][,addr=addr][,vectors=v]`: 创建一个新的网卡设备并连接至vlan n中; `macaddr`用于为其指定MAC地址, `name`用于指定一个在监控时显示的网上设备名称; `emu`可以模拟多个类型的网卡设备

`-net tap[,vlan=n][,name=name][,fd=h][,ifname=name][,script=file][,downscript=dfile]`: 通过物理机的TAP网络接口连接至vlan n中, 使用`script=file`指定的脚本(默认为`/etc/qemu-ifup`)来配置当前网络接口, 并使用`downscript=file`指定的脚本(默认为`/etc/qemu-ifdown`)来撤消接口配置; 使用`script=no`和`downscript=no`可分别用来禁止执行脚本;

`-net user[,option][,option][,...]`: 在用户模式配置网络栈, 其不依赖于管理权限; 有效选项有:

`vlan=n`: 连接至vlan n, 默认n=0;

`name=name`: 指定接口的显示名称, 常用于监控模式中;

`net=addr[/mask]`: 设定GuestOS可见的IP网络, 掩码可选, 默认为10.0.2.0/8;

`host=addr`: 指定GuestOS中看到的物理机的IP地址, 默认为指定网络中的第二个, 即x.x.x.2;

`dhcpstart=addr`: 指定DHCP服务地址池中16个地址的起始IP, 默认为第16个至第31个, 即x.x.x.16-x.x.x.31;

`dns=addr`: 指定GuestOS可见的dns服务器地址; 默认为GuestOS网络中的第三个地址, 即x.x.x.3;

`tftp=dir`: 激活内置的tftp服务器, 并使用指定的dir作为tftp服务器的默认根目录;

`bootfile=file`: BOOTP文件名称, 用于实现网络引导GuestOS; 如: `qemu -hda linux.img -boot n -net user,tftp=/tftpserver/pub,bootfile=/pxelinux.0`

通过本地ISO文件来进行安装

```
# qemu-img create -f qcow2 vm1-disk1.qcow2 10G

# virt-install \
  --name=vm1 \
  --disk path=/vm/vm1-disk1.qcow2\
  --vcpus=1 --ram=1024 \
  --cdrom=/iso/CentOS-6.4-i386-bin-DVD1.iso \
  --network network=default \
  --graphics vnc,listen=0.0.0.0 \
  --os-type=linux \
  --os-variant=rhel6
```

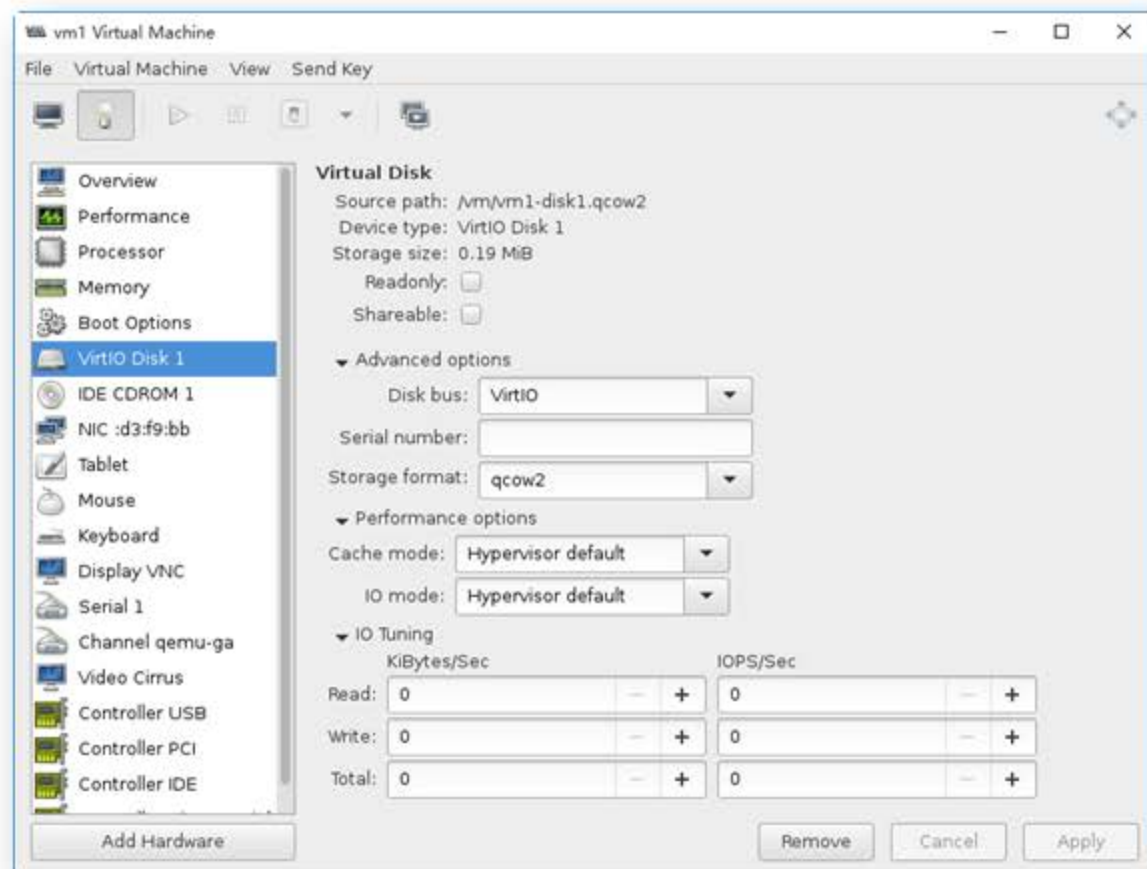
考察新创建虚拟机

```
virsh # list --all
```

Id	Name	State
-	vm1	shut off

```
virsh # dominfo vm1
```

```
virsh # edit vm1
```



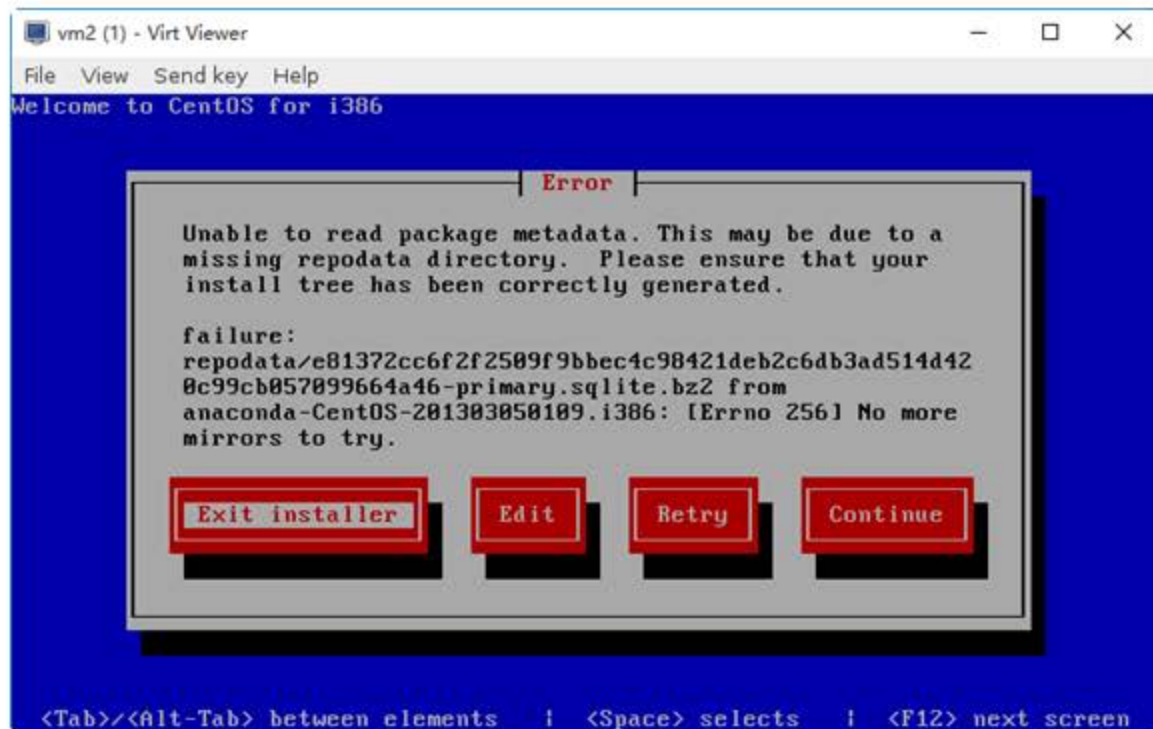
Kickstart安装

- ▶ Kickstart是一种无人值守的安装方式
- ▶ 如果在安装过程中出现要填写参数的情况，安装程序首先会去查找Kickstart生成的文件，如果找到合适的参数，就采用所找到的参数；如果没有找到合适的参数，便需要安装者手工干预了

```
# virt-install \  
  --name=vm1 \  
  --disk path=/vm/vm1-disk1.qcow2\  
  --vcpus=1 --ram=1024 \  
  --network network=default \  
  --graphics vnc,listen=0.0.0.0 \  
  --os-type=linux \  
  --os-variant=rhel6 \  
  --location /iso/CentOS-6.4-i386-bin-DVD1.iso \  
  --extra-args="ks=http://192.168.1.11/mini.txt"
```

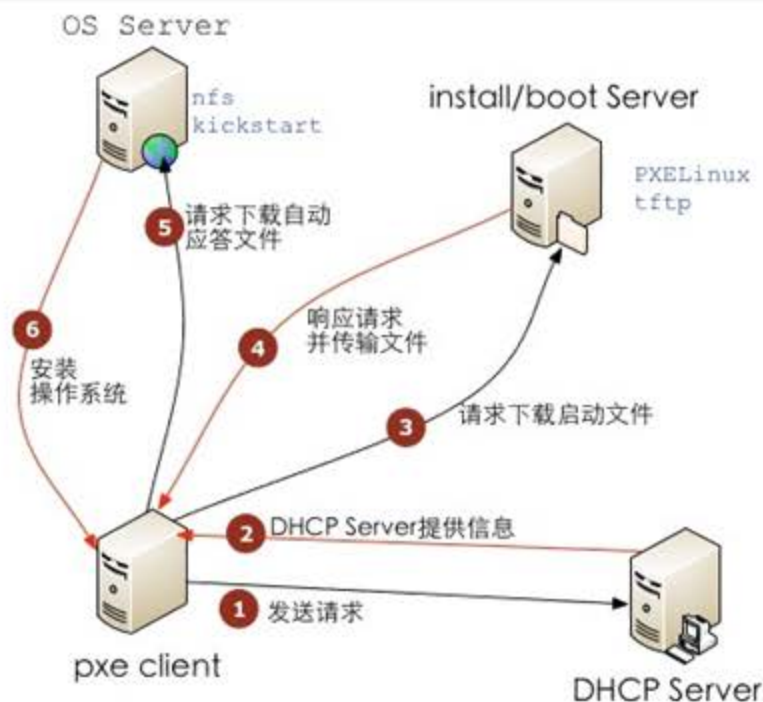
网络安装

- ▶ 我们也可以指定安装介质文件位置网络上
`--location=http://192.168.1.18/os/centos64i386/`
- ▶ 如果Web服务器是Windows，有可能出现以下错误



PXE安装

```
# virt-install --hvm --connect qemu:///system \  
--network=bridge:br0 --pxe --graphics spice \  
--name=rhel6-machine --ram=756 --vcpus=4 \  
--os-type=linux --os-variant=rhel6 \  
--disk path=/var/lib/libvirt/images/rhel6-machine.img,size=10
```



◆ 半虚拟化驱动virtio

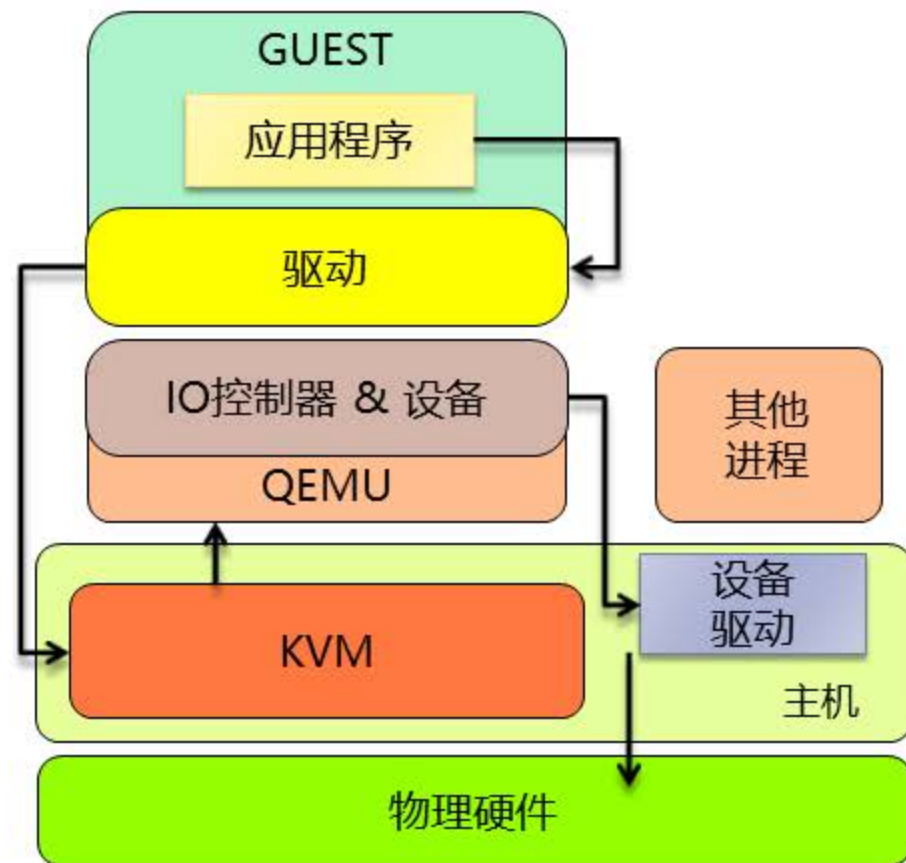
- ▶ 半虚拟化驱动virtio原理概述
- ▶ 性能对比
- ▶ 半虚拟化设备统一接口

使用半虚拟化驱动virtio的目的

- 为了提高内存、硬盘、网络的性能，需要支持半虚拟化

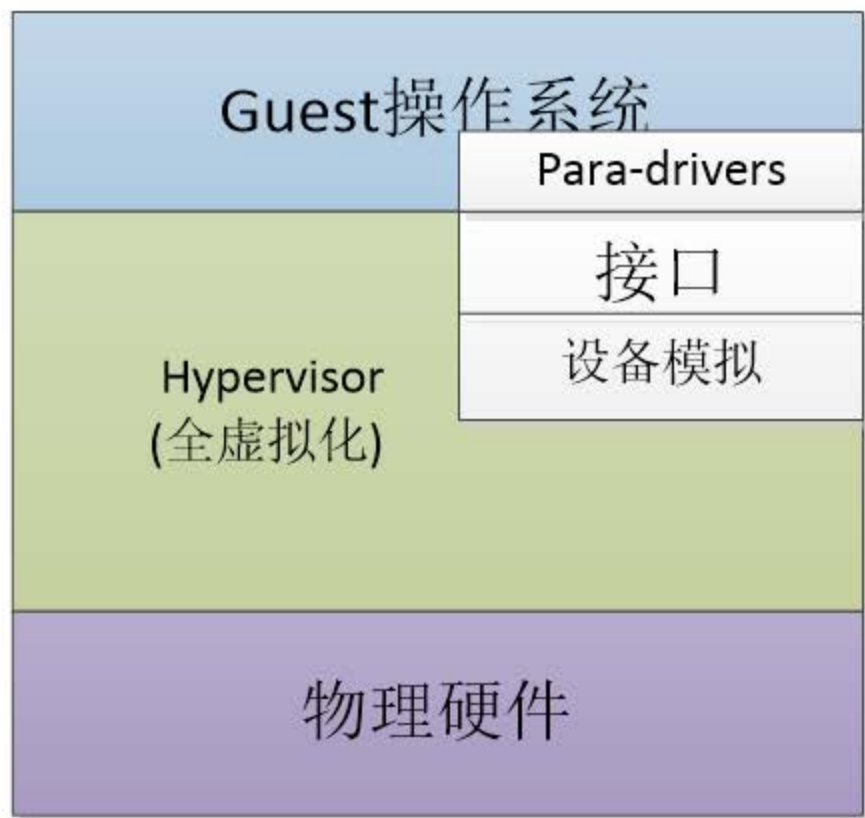


没有virtio的全虚拟化

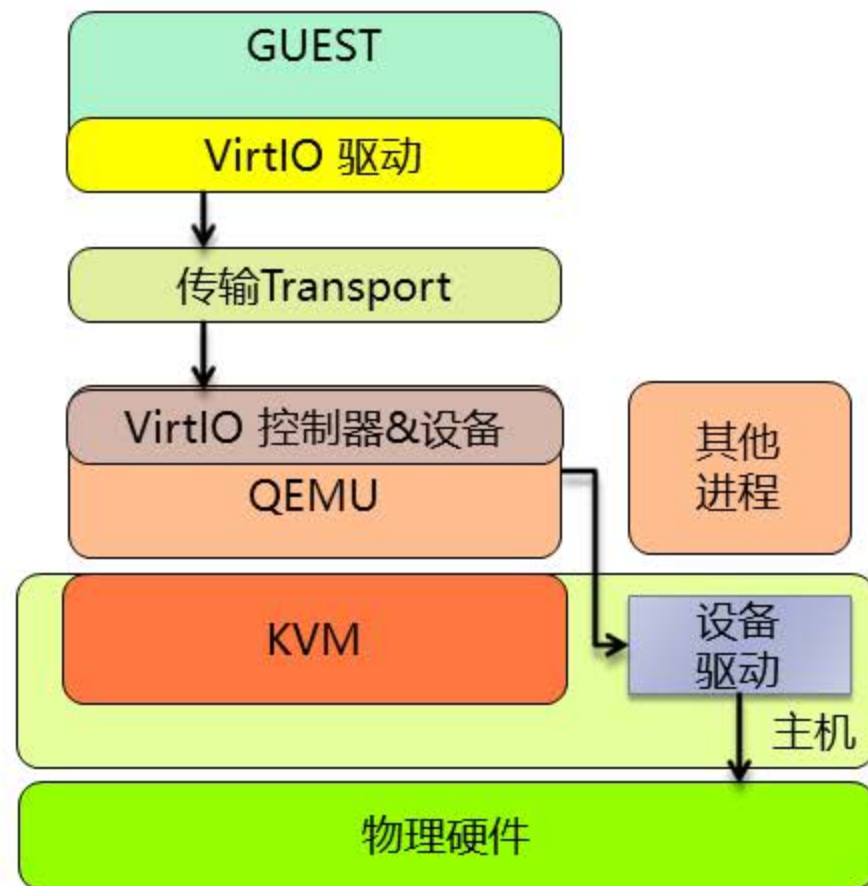


使用半虚拟化驱动virtio的目的

- 为了提高内存、硬盘、网络的性能，需要支持半虚拟化



带有virtio

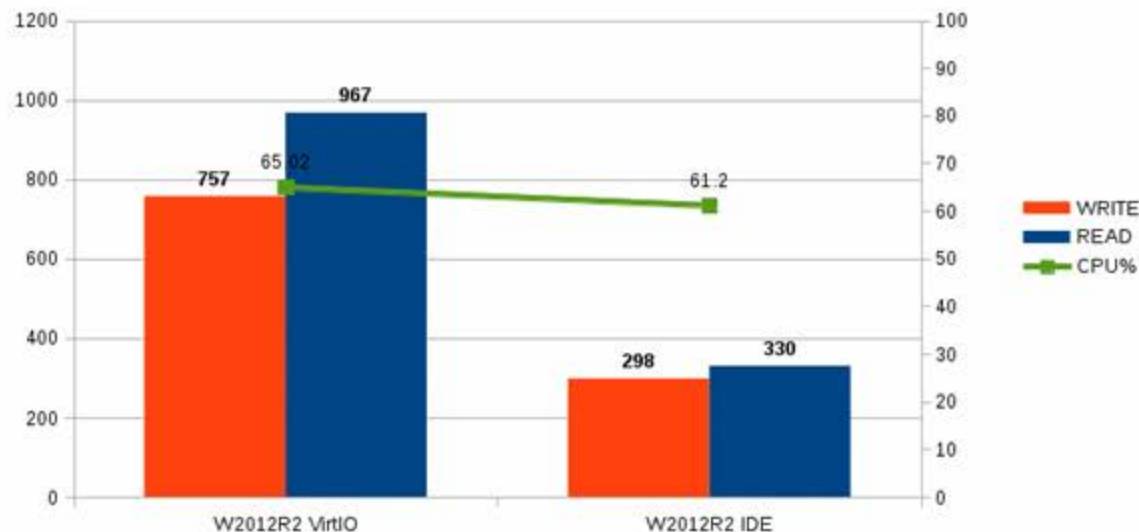


性能比较

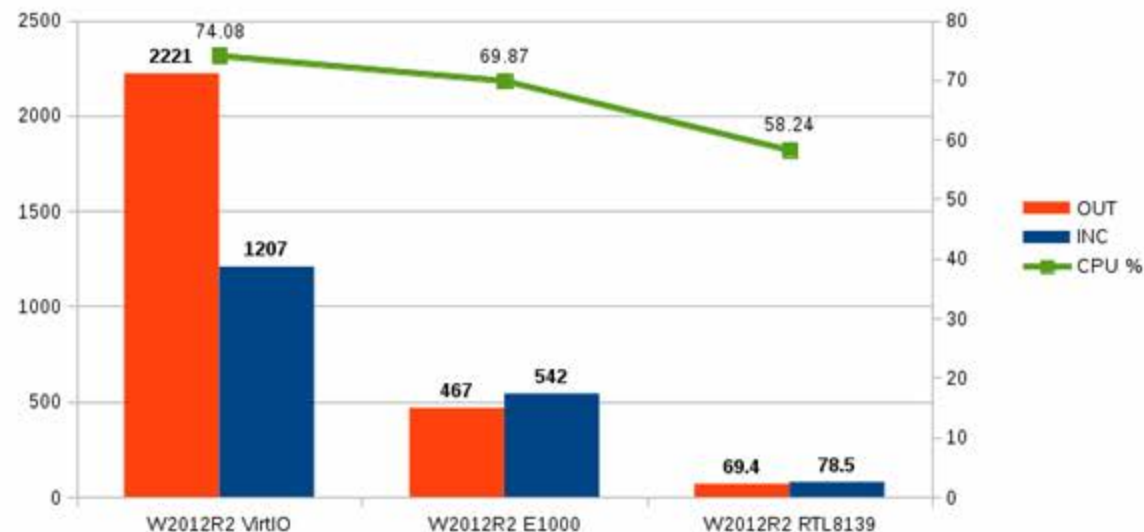
► KVM VirtIO paravirtualized drivers: why they matter

<http://www.ilsistemista.net/index.php/virtualization/42-kvm-virtio-paravirtualized-drivers-why-they-matter.html>

Disk sequential speed, MB/s and CPU load

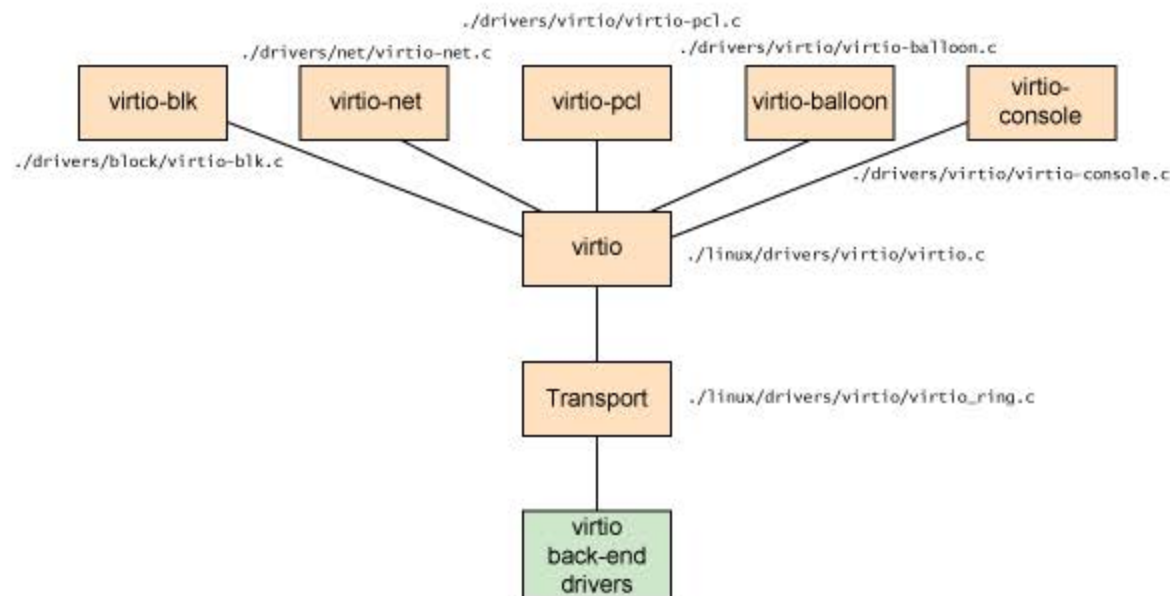
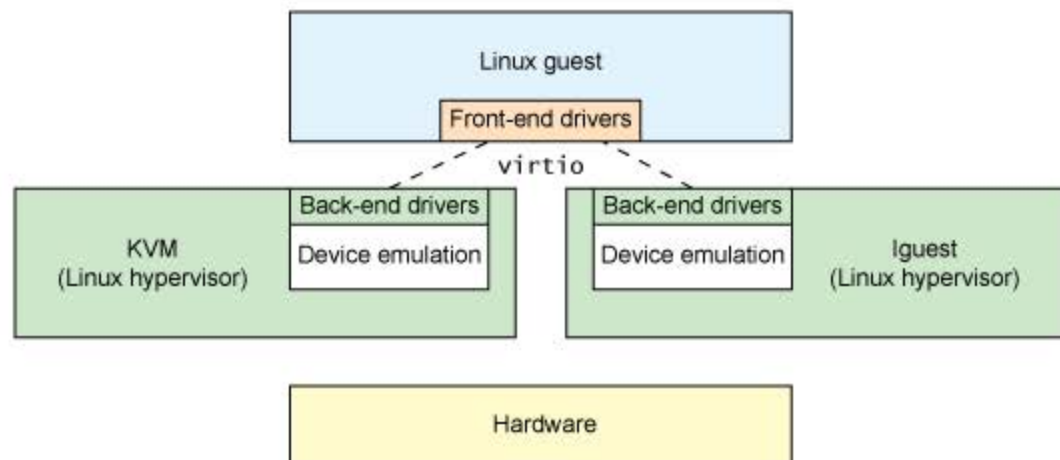


TCP transfer speed, Mbit/sec and CPU load



半虚拟化设备统一接口

- ▶ 通过统一的接口virtio以支持的多种硬件设备
 - ▶ 不同的虚拟设备和不同的虚拟机可以有不同的前端驱动
 - ▶ 不同的硬件设备可以有不同的后端驱动
 - ▶ 两者之间的交互遵循virtio的标准



获得virtio驱动程序

▶ 红帽RHEL 4.8之后自动加载和安装virtio驱动

```
[root@guest-centos1 ~]# lsmod | grep virtio
virtio_balloon 3866 0
virtio_net     13237 0
virtio_blk      6091 3
virtio_pci      5392 0
virtio_ring     7020 4 virtio_balloon,virtio_net,virtio_blk,virtio_pci
virtio          3760 4 virtio_balloon,virtio_net,virtio_blk,virtio_pci
```

▶ Windows操作系统需要额外安装virtio的驱动。

▶ 方法1：从RHN上下载

▶ 如果您有RHN订阅，可以从以下位置下载virtio-win包：

▶ <https://rhn.redhat.com/rhn/software/packages/details/Overview.do?pid=868414>

▶ 方法2：从社区获得

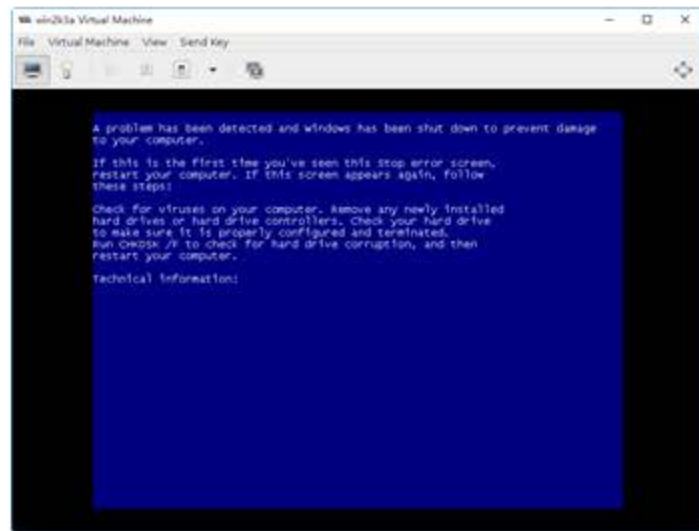
▶ <http://www.linux-kvm.org/page/Downloads>

实验1：在已经安装好的Windows虚拟机上安装virtio驱动

- ▶ 通过向导创建的VM，默认磁盘为IDE接口、网卡为rtl8139，即全虚拟化驱动。
- ▶ 一定要按以下步骤来进行操作，否则会出现蓝屏错误
 - ▶ 关闭虚拟机。编辑虚拟机的配置，添加一个很小的virtio硬盘、一块virtio网卡
 - ▶ 虚拟光驱加载上传的ISO文件
 - ▶ 启动虚拟机。登录后，Windows会发现新的硬件。
 - ▶ 安装时使用手动搜索驱动，不同的操作系统路径如下。

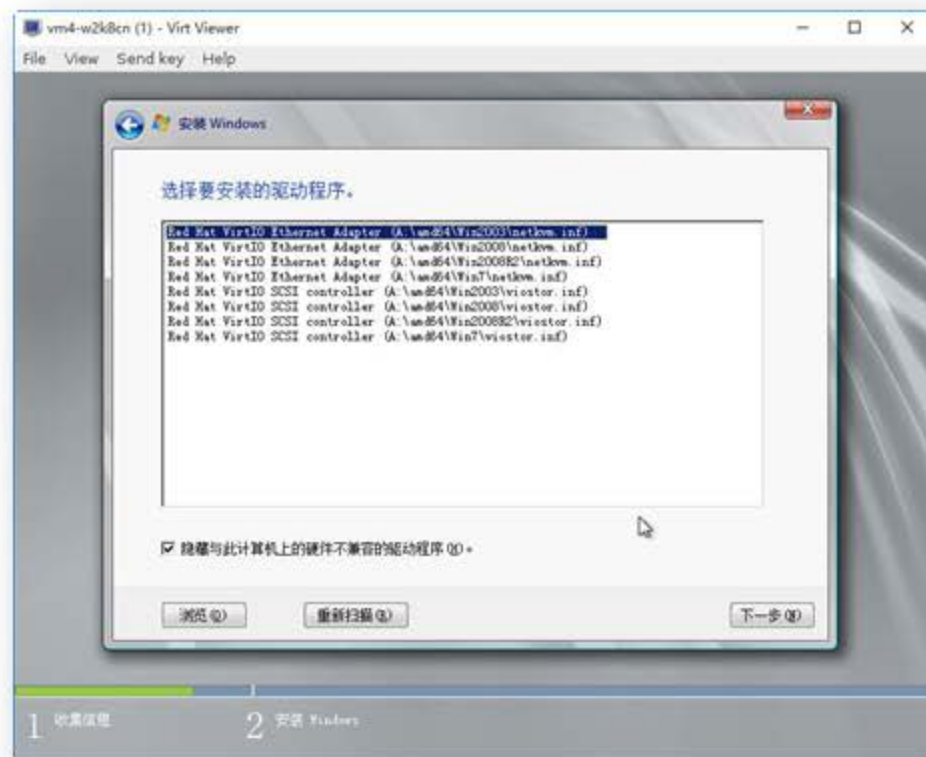
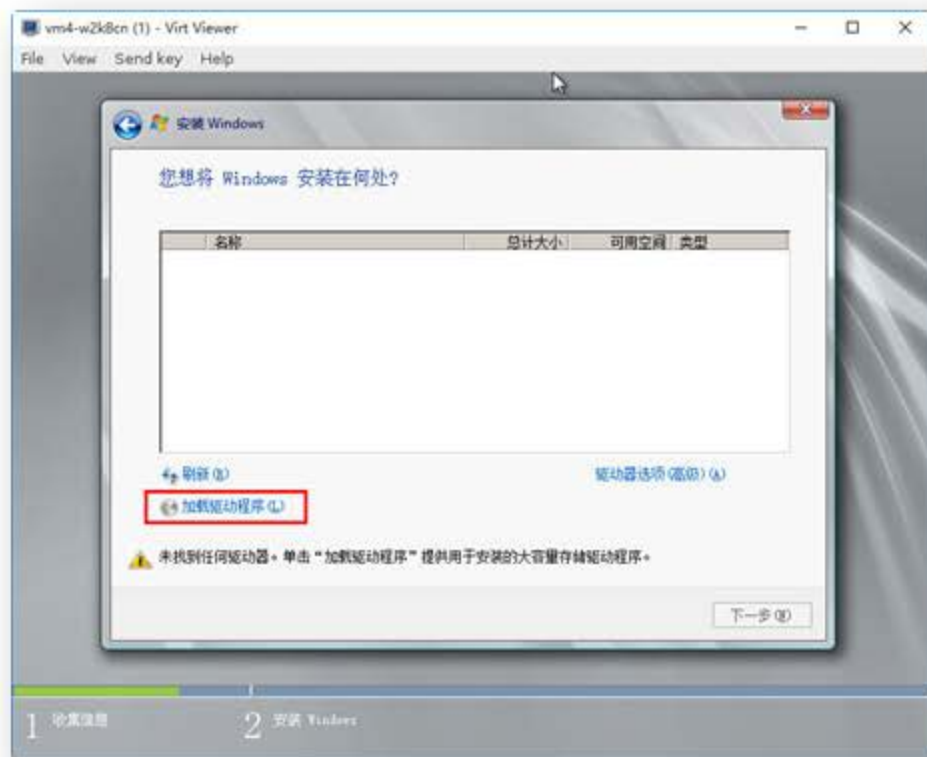
	磁盘	网卡
XP	viostor\xp\x86\viostor.inf	NetKVM\xp\x86\netkvm.inf
2k3	viostor\2k3\x86\viostor.inf	NetKVM\2k3\x86\netkvm.inf
win7	viostor\w7\x86\viostor.inf	NetKVM\w7\x86\netkvm.inf
.....

- ▶ 最后，安装其他驱动程序如Balloon、串口等



实验2：安装Windows操作系统时安装virtio驱动程序

- Windows 2008及之后操作系统，在安装时指出驱动程序的位置

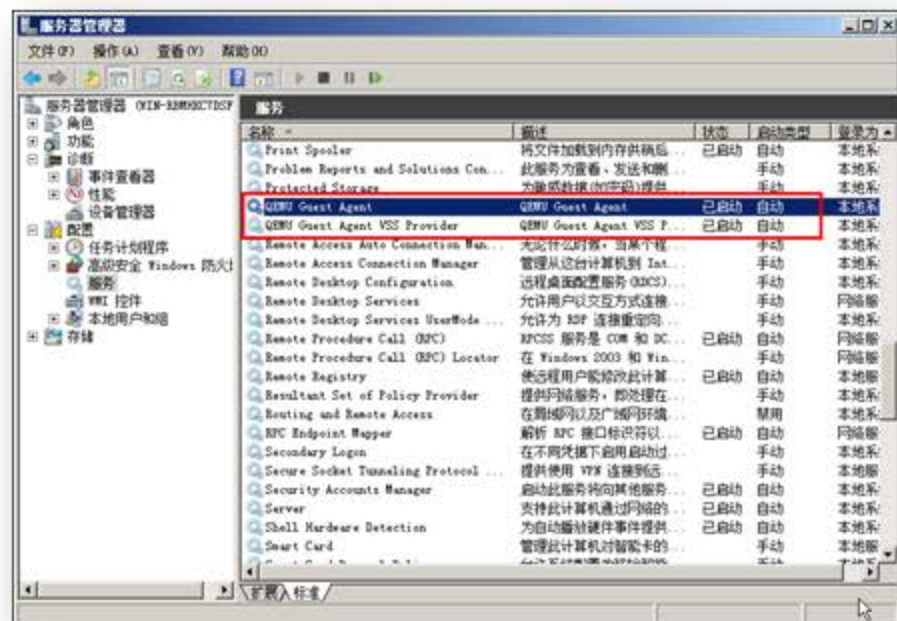
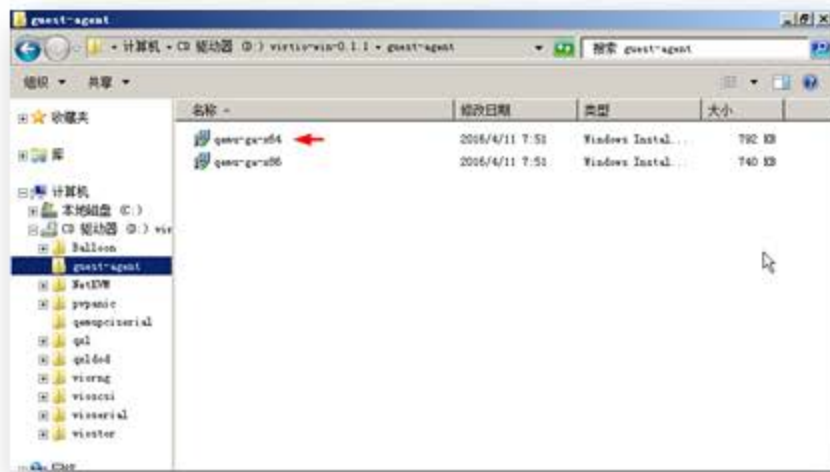


◆ QEMU Guest Agent和SPICE Agent

- ▶ QEMU guest agent安装与配置
- ▶ 通过libvirt来使用QEMU guest agent
- ▶ SPICE agent安装与配置

QEMU guest agent安装与配置

- ▶ 如果VM中安装了QEMU guest agent，Host就可以使用libivrt向VM发送命令，例如“冻结”、“释放”文件系统，虚拟CPU的热添加及移除等。
- ▶ RHEL/CentOS 7中有相应的安装包。qemu-guest-agent-xxx.rpm
- ▶ Windows需要手工安装



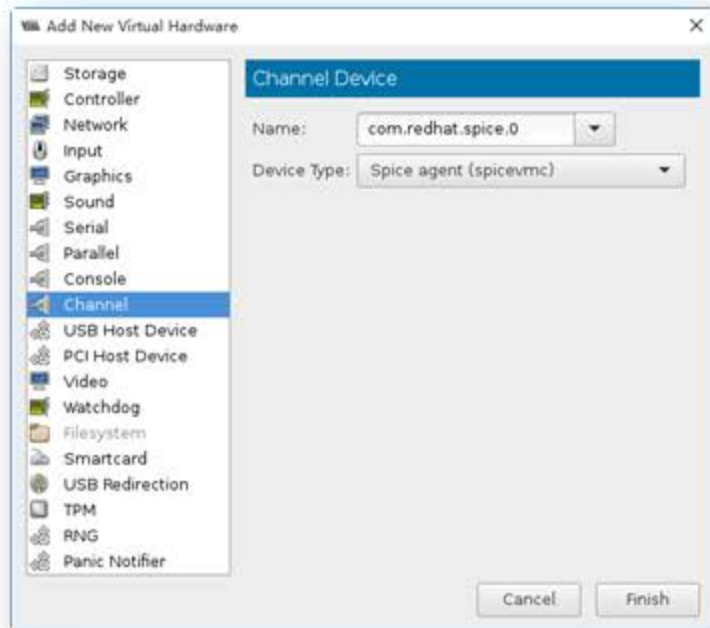
通过libvirt来使用QEMU guest agent

安装了QEMU guest agent后，对libvirt命令有如下的增强

- | | |
|---|--|
| ▶ <code>virsh shutdown --mode=agent</code> | 比 <code>--mode=acpi</code> 更加安全地关闭操作系统 |
| ▶ <code>virsh snapshot-create -quiesce</code> | 在创建快照之前面，将缓存的内容刷入到磁盘 |
| ▶ <code>virsh domfsfreeze</code> | 静默文件系统 |
| ▶ <code>virsh domfsthaw</code> | 恢复静默的文件系统 |
| ▶ <code>virsh domfstrim</code> | 让虚拟机trim文件系统 |
| ▶ <code>virsh domtime</code> | 获得虚拟机的时间 |
| ▶ <code>virsh setvcpus</code> | 配置虚拟机的vCPU |
| ▶ <code>virsh domifaddr --source agent</code> | 查询虚拟机的IP地址 |
| ▶ <code>virsh domfsinfo</code> | 显示虚拟机的文件系统列表 |
| ▶ <code>virsh set-user-password</code> | 设置虚拟机用户的密码 |

SPICE agent安装与配置

- ▶ 通过在VM操作系统中安装SPICE client，SPICE agent使virt-manager等图形应用程序更加流畅。例如：
 - ▶ 在virt-manager中调整窗口尺寸，SPICE agent自动调整X会话的分辨率
 - ▶ 在Host与Guest之间复制与粘贴
 - ▶ 防止鼠标拖尾等。
- ▶ 下载：
 - ▶ <http://www.spice-space.org/download/>



总结

- ▶ 使用virt-manager创建虚拟机
- ▶ 使用virt-install创建虚拟机
- ▶ 半虚拟化驱动virtio
- ▶ QEMU Guest Agent和SPICE Agent