Qemu

Table des matières

1 Installation

Cf : Site officiel.

1.1 Installation de base

```
# apt-get install qemu
# chmod og+wxr /opt
$ qemu-img create /opt/virtualDebian.img 7G
```

- Boot via le CDROM :

```
$ qemu -hda /opt/virtualDebian.img -cdrom /dev/cdrom -boot d
```

- Boot via le fichier (pseudo partition *hda*) :

```
$ qemu -hda /opt/virtualDebian.img -cdrom /dev/cdrom -boot c
```

1.2 Emulation d'un sous réseau

- Fichier */etc/network/interface* côté machine physique :

```
iface tap0 inet static
        address 192.168.2.1
        netmask 255.255.255.0
```

- Création de l'interface :

```
# modprobe tun
# apt-get install uml-utilities
# tunctl -t tap0
Set 'tap0' persistent and owned by uid 0

# tunctl -u nroche -t tap0
Set 'tap0' persistent and owned by uid 1000

# cat /proc/net/dev | grep tap0
# ifup tap0
# ifconfig tap0
```

- Lancement de la machine virtuelle avec une carte réseau :

```
qemu -hda /opt/virtualDebian.img -boot c -net nic,vlan=0 -net tap,vlan=0,ifname=tap0,script=no
```

Détail des options :

- `-net nic,vlan=n` : ajoute côté machine virtuelle une carte **ne2k_pci** sur le vlan 'n'
- `-net tap,vlan=0,ifname=tap0,script=no` : utilise côté machine physique l'interface *tap0* sans utiliser le script */etc/qemu-ifup*

- Fichier */etc/network/interface* côté machine virtuelle :

```
iface tap0 inet static
        address 192.168.2.2
        netmask 255.255.255.0
        gateway 192.168.2.1
```

- Ouverture du réseau via masquerade :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
# iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -j MASQUERADE
```

Essayez de joindre le serveur de noms...

1.3 QEMU acceleration layer

When used as a virtualizer, QEMU achieves near native performances by executing the guest code directly on the host CPU. A host driver called the QEMU accelerator (also known as KQEMU) is needed in this case. The virtualizer mode requires that both the host and guest machine use x86 compatible processors.

- Compilation du module *kqemu*.

```
# apt-get install kqemu-common kqemu-source linux-headers-2.6.26-1-486 gcc libc6-dev
# cd /usr/src/
# tar -jxf kqemu.tar.bz2
# chown ...
$ cd /usr/src/modules/kqemu/
$ ./configure
$ make
# ./install.sh
```

- Test.

```
# modprobe kqemu
qemu-img create /opt/virtualTest.img 1G
qemu -hda /opt/virtualTest.img -boot c -no-kqemu
qemu -hda /opt/virtualTest.img -boot c
```

- Création manuelle du device (non requis à priori)

```
# mknod /dev/kqemu c 250 0
# chmod 666 /dev/kqemu
# modprobe kqemu
```

Attention toutefois à */etc/modprobe.d/kqemu* qui doit faire un truc comme :

```
#  /sbin/modprobe kqemu major=250
```

2 Script de lancement

```
#!/bin/bash

if [ "$(id --user)" -eq 0 ] then
   echo $0 "only root can run this script"
   exit
fi

# acceleration
modprobe tun

# Emuler le sous réseau
modprobe kqemu
ifup tap0
iptables -t nat -D POSTROUTING -o eth1 -s 192.168.1.120 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.1.120 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward

# creation de l'instance
sudo -nroche qemu \
        -hda /opt/virtualDebian \
        -cdrom /dev/cdrom \
        -net nic,vlan=0 \
        -net tap,vlan=0,ifname=tap0,script=no \
        -boot c &
```