

Virtual Systems with qemu

Version 0.1 - 2011-02-08

Christian Kulker

Inhaltsverzeichnis

1	Image Creation	2
1.1	Preparations	2
1.2	Creating a Disk Image	2
1.3	Installing the Operating System	2
1.4	Finish the Installation of the Operating System	2
1.5	Starting this in full emulation	3
1.6	Possible problems	3
2	Usage qemu on ampemaster9 as unprivileged user	3
2.1	start qemu as unprivileged user	3
2.2	stop qemu	4
2.3	Performance	4
3	Solutions to performance issues	4
3.1	Checking System Capability	4
3.2	Installation Squeeze (Command line)	4
3.3	Start the default network	5
3.4	Configuration as non root user	5
3.5	Test Configuration	5
3.6	Install machine the kvm way	5
3.7	Test if it is running	6
3.8	Connect to your virtual machine	6
3.9	Install and configure of the virtual machine	6
3.10	Shutdown	6

3.11	Converting images from virtualbox to kvm	6
3.12	Create a development server with COW	7
3.13	Fiddeling around with the configuration	7
3.14	URLs	7
3.15	AUTHOR	7
3.16	LICENSE	7

1 Image Creation

1.1 Preparations

Getting the right ISO installer for your preferred OS

```
export MIRROR=http://ftp.uni-muenster.de/pub/mirrors/ftp.ubuntu.com/  
export ISO=releases/lucid/ubuntu-10.04-server-i386.iso  
wget $MIRROR/$ISO
```

1.2 Creating a Disk Image

Making an disk image (takes a while, approx 320 sec):

```
cd $PATH_TO_A_HD_WITH_SPACE  
dd if=/dev/zero of=ubuntu-10.04-server-i386.img bs=1024k count=10000
```

Alternatively you can do that with a qemu command. This will create fast a self growing image, that grows up to 10G in the following example:

```
cd $PATH_TO_A_HD_WITH_SPACE  
qemu-img create -f qcow2 ubuntu-10.04-server-i386.img 10G
```

1.3 Installing the Operating System

Choose the correct key map for installation.

```
ls /usr/share/qemu/keymaps/
```

Choose en-us

```
qemu -hda /data/ckuelker/qemu/ubuntu-10.04-server-i386.img \  
-cdrom /data/ckuelker/qemu/ubuntu-10.04-server-i386.iso \  
-boot d -m 512 -k en-us
```

1.4 Finish the Installation of the Operating System

The reboot after the package installation to finish the installation fails, because qemu will not boot the hard disk with the above mentioned parameters. Press CTRL-C and continue with this evocation of qemu.

```
qemu -hda /data/ckuelker/qemu/ubuntu-10.04-server-i386.img \  
-cdrom /data/ckuelker/qemu/ubuntu-10.04-server-i386.iso \  
-boot c -m 512 -k en-us
```

1.5 Starting this in full emulation

```
/usr/bin/qemu-system-i386 \  
-hda /Data/qemu/ubuntu-lucid-10.04-server-i386.img \  
-cdrom /Data/ISO/lucid-server-i386.iso -boot c \  
-m 512 -k en-us
```

1.6 Possible problems

Could not initialize SDL - exiting

If you receive this message a qemu process is not able to write to your display. Make sure that you logged into the server with `ssh -X` or `ssh -Y` to enable printing on your display.

2 Usage qemu on apemaster9 as unprivileged user

This section describes how to start and stop the qemu Ubuntu environment on apemaster9.

Pros and Cons:

- very large environment > 1.1G for usage, 10G for backup storage
- very slow
- large main memory foot print
- + can be used as root and as unprivileged user
- + most devices and services working
- + kernel module will load (most likely not needed for compilation)

apemaster9 is considered under dimensioned to run this software efficiently. To less main memory. Consider this as a prove of concept.

2.1 start qemu as unprivileged user

```
cd /data/ckuelker/qemu  
qemu -m 512 -redir tcp:5556::22 -daemonize -nographic \  
-hda /data/ckuelker/qemu/ubuntu-10.04-server-i386.img -boot c -k en-us
```

Wait several minutes to let the system start up. 2 minutes are usual.

```
ssh -p 5556 $USER@localhost
```

2.2 stop qemu

Either log into the machine, get root and issue:

```
poweroff
```

Inside the qemu machine and wait until the machine is down. Or kill the machine via process id. The later will result in file corruption sooner or later.

2.3 Performance

A small binary was compiled in /home/ckuelker/dev

```
time gcc -o tramp3d-v4 tramp3d-v4.cpp
real 8m40.248s
user 7m46.217s
sys 0m23.229s
```

It did not compile until end.

3 Solutions to performance issues

In the past the only way to speed up a qemu instance was to compile the kernel module kqemu. Since kernel 2.6.20 kvm is part of the kernel which together with libvirt provides a generic method to start, stop and execute qemu based virtual machines.

The following section describes a setup under debian squeeze which can be used as a blueprint to setup other computers too.

Other options might be to use virtual-box or xen, which is not covered by this document.

3.1 Checking System Capability

```
egrep '(vmx|svm)' --color=always /proc/cpuinfo
```

3.2 Installation Squeeze (Command line)

```
aptitude install qemu-kvm libvirt-bin
```

If you prefer a GUI, issue additionally:

```
aptitude install virt-manager
```

3.3 Start the default network

```
virsh net-start default
ifconfig
```

```
virbr0    Link encap:Ethernet  HWaddr 3e:ab:a1:5f:cd:c6
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

This make a autostart link:

```
virsh net-autostart default
```

3.4 Configuration as non root user

As root:

```
adduser ckuelker libvirt
exit
#login again as ckuelker
```

3.5 Test Configuration

```
virsh -c qemu:///system list
```

3.6 Install machine the kvm way

```
cd /kvm/iso
wget http://cdimage.debian.org/debian-cd/6.0.0/amd64/iso-cd/debian-6.0.0-amd64-netinst.
virt-install --connect qemu:///system -n vm01 -r 512 --vcpus=2 \
-f /kvm/vm01/vm01.qcow2 -s 12 -c /kvm/iso/debian-6.0.0-amd64-netinst.iso \
--vnc --noautoconsole --os-type linux --os-variant debianSqueeze \
--accelerate --network=network:default --hvm
```

Output:

```
Starting install...
```

```
Creating domain...
```

```
0 B 00:00
```

```
Domain installation still in progress. You can reconnect to  
the console to complete the installation process.
```

3.7 Test if it is running

```
virsh -c qemu:///system list  
Id Name State  
-----  
1 vm01 running
```

3.8 Connect to your virtual machine

```
virt-manager
```

3.9 Install and configure of the virtual machine

Set the IP address if the virtual Machine to on IP of the bridge NAT.

```
vim /etc/network/interfaces
```

```
use 192.168.122.IP
```

```
Remove eth0, eth1 entries in udev:
```

```
vim /etc/udev/rules.d/z25_persistent-net.rules
```

3.10 Shutdown

```
virsh -c qemu:///system shutdown vm01
```

3.11 Converting images from virtualbox to kvm

Needs a lot of space:

```
VBoxManage internalcommands converttoraw file.vdi file.raw  
qemu-img file.raw file.qcow2
```

3.12 Create a development server with COW

”cow” means copy on write. This allows you to set up an image once and use it many times without changing. This is ideal for developing and testing software.

```
qemu-img create -f qcow2 -b linux.img test01.img 3GB
qemu -m 256 -hda test01.img -kernel-kqemu &
```

3.13 Fiddling around with the configuration

The configuration can be changed by hand.

```
vim /etc/libvirt/qemu/100.cow.lenny.xml
```

If a configuration will be cloned, a new unique ID has to be applied. It seems that increasing the ID by one will do it.

```
<uuid>881d1312-2525-c732-7b1b-61ec7453dc7e</uuid>
<uuid>881d1312-2525-c732-7b1b-61ec7453dc7f</uuid>
```

After changes will take effect you have to restart the daemon

```
/etc/init.d/libvirt-bin restart
```

And the bridge has probably to be restarted too. This can be done with `virtsh` or `ifconfig` or `init` script, depending on how you create the bridge device.

The virtual manager needs a restart

```
virt-manager
```

3.14 URLs

```
http://www.howtoforge.com/virtualization-with-kvm-on-a-debian-lenny-server
http://www.howtoforge.com/virtualization-with-kvm-on-ubuntu-9.10
http://wiki.centos.org/HowTos/KVM
http://edin.no-ip.com/content/kvm-debian-mini-howto
```

3.15 AUTHOR

Copyright 2011-02-08 by Christian Kuelker - v0.1

3.16 LICENSE

Latest GFDL with no invariant sections