By Sidhartha Mandal

Published: 2008-06-04 18:17

XEN On An Ubuntu Hardy Heron (8.04) Server System (amd64) - High Performance

This tutorial provides step-by-step instructions on how to install XEN on an Ubuntu Hardy Heron (8.04) Server System (amd64) without compromising on disk I/O and network throughput. You can find all the software used here in the Ubuntu repositories, so no external files or source compilation are required.

Xen is an open-source para-virtualizing virtual machine monitor(VMM), or 'hypervisor', for the x86 processor architecture. Xen cansecurely execute multiple virtual machines on a single physical systemwith close-to-native performance. Xen facilitates enterprise-gradefunctionality, including:

- Virtual machines with performance close to native hardware.
- Live migration of running virtual machines between physical hosts.
- Up to 32 virtual CPUs per guest virtual machine, with VCPU hotplug.
- x86/32, x86/32 with PAE, and x86/64 platform support.
- Intel Virtualization Technology (VT-x) for unmodified guest operating systems (including Microsoft Windows).
- AMD Virtualization Technology (SVM aka Pacifica) on AM2 and F stepping Opterons (2006H2)
- Excellent hardware support (supports almost all Linux device drivers).

1. Partition And Primary OS Installation

I will use Ubuntu Hardy Heron (x86_64) for both the host OS (`dom0`) and the guest operating systems (`domU`). Prior to this you should have a target system ready with partitions like in the following example:

For example we have 120GB disk space and we want to use 2 guest operating systems on it:

- Take as minimal as required for host OS ***(dom0),*** say 10-12GB => so create the first partition accordingly.
- We will divide the rest of the free space in 2 partitions, each for 2 guest OS's (domU's) => so create the next two partitions accordingly.

Install the Ubuntu Hardy Heron (x86_64) Server Edition (***http://www.ubuntu.com/products/whatisubuntu/serveredition***) on the first partition only. Afterwards, on this we will be install the Xen stuff and use this as the host OS (dom0). As per better practice, dom0 should not be used for any production use except for controlling guest OS's. So only choose "ssh" in package selection during OS installation.

After the OS installation the first job is to update/upgrade the OS. Login to the system as root and type following commands:

```
# apt-get update


# apt-get upgrade
```

2. Host OS And Xen Installation

To install Xen and all needed dependencies, all we have to do is run the following command:

```
# apt-get install ubuntu-xen-server
```

The *ubuntu-xen-server* package installs the Ubuntu kernel *2.6.24-17-xen* and other dependent xen packages.

Also upgrade the vi editor as follows:

```
# apt-get install vim
```

We also need to add the *loop* module to the kernel every time we boot our system, so we edit */etc/modules*. If you already have a *loop* line in there, make it look like this, otherwise add it at the bottom of the file:

```
# vi /etc/modules
```

```
loop max_loop=64
```

That's all. Now reboot the system. And after the reboot check the OS and xen service as follows:

```
# uname -a


# xm list
```

3 Creating Filesystem For Guest OS (*domU*)

For the creation of domU we will be using "***debootstrap***". Here we will be using direct physical disk partitions and no filesystem images for better disk I/O for the guest OS. So format the partition for your filesystem choice. If you are interested in the xfs filesystem then you must install the "xfsprogs" package on dom0 like this:

```
# apt-get install xfsprogs
```

To create a filesystem on the second partition we use the following command:

If you are choosing ext3 then:

```
# mkfs.ext3 /dev/sda2
```

If you are choosing xfs then:

```
# mkfs.xfs /dev/sda2
```

Post creation of filesystem in target partition mount it to */mnt*:

```
# mount /dev/sda2 /mnt
```

4. Installing The Guest OS

Install the base OS stuff in `/mnt` using "debootstrap":

```
# debootstrap --arch amd64 hardy /mnt http://archive.ubuntu.com/ubuntu
```

This is going to take sometime depending on your Internet connection speed.

Once done, prepare the chroot environment for `/mnt`:

```
# mount --bind /dev /mnt/dev
```

```
# mount proc /mnt/proc -t proc
```

```
# chroot /mnt /bin/bash
```

Open another terminal and copy a couple of files from dom0 to `/mnt`'s respective folder:

```
# cp /etc/resolv.conf /mnt/etc/resolv.conf
```

```
# cp /etc/network/interface /mnt/etc/network/interface
```

```
# cp /etc/apt/source.list /mnt/etc/apt/sources.list
```

```
# cp -R /lib/modules/2.6.24-17-xen/* /mnt/lib/modules/2.6.24-17-xen/
```

(If the respective folder is not present, then please create it before copying.)

Now again from the domU chroot environment use the following commands:

```
# apt-get update
```

```
# apt-get upgrade
```

```
# apt-get install vim ssh
```

If you are using the xfs filesystem then also install *xfsprogs* as follows:

```
# apt-get install xfsprogs
```

Update */etc/fstab* as follows:

If you are using the ext3 filesystem then:

```
/dev/hda1        /        ext3 defaults    1    2
```

If you are using the xfs filesystem then:

```
/dev/hda1        /        xfs defaults    1    2
```

This is the minimal setup of the domU environment. Not set the root password as:

```
# passwd
```

It's going to prompt you for root's password, so provide some secret for it.

Now exit from the chroot environment:

```
# exit
```

Now umount all partitions:

```
# umount  /mnt/dev /mnt/proc /mnt
```

5. Creation Of Xen Config File

Now create the first guest OS's xen config file named `domu1.cfg` in the `/etc/xen/` folder like this:

```
vi /etc/xen/domu1.cfg
```

```
kernel      = '/boot/vmlinuz-2.6.24-17-xen'
ramdisk     = '/boot/initrd.img-2.6.24-17-xen'
memory      = '512'
#
#  Disk device(s).
#
root        = '/dev/hda1 ro'
disk        = [
                  'phy:/dev/sda2,hda1,w',
      ]
#
#  Hostname
#
name        = 'domu1'
#
#  Networking
```

```
#
vif         = [ 'ip=192.168.1.102,mac=00:16:3E:62:DA:BB' ]
#
#  Behaviour
#
on_poweroff = 'destroy'
on_reboot   = 'restart'
on_crash    = 'restart'
vcpus     = '2'
extra = 'xencons=tty1'
```

You must use a free and different IP for each guest OS.

Then start the first domU like this:

```
# xm create /etc/xen/domu1.cfg -c
```

Log in using root and that's all... your first guest OS (domu1) is ready now.

In dom0, to list running OS's, type:

```
# xm list
```

The steps to create second virtual system on */dev/sda3* are similar...

The same steps can be followed for i386 also...

Thanks.