Install and run Xen-unstable (2.0)

By Rune Johan Andresen

Overview

This is a HOWTO for installing Xen-unstable (2.0), a virtual machine monitor for the IA32 platform. The tutorial will go trough three main steps: Get the Xen source code, build Xen and finally get it up and running. You need a preinstalled Linux distribution on a x86 computer, preferable 512+ MB ram, in order to install and run Xen. In a more technical view the tutorial will go trough the following steps:

1. Get the Xen source code
2. Build Xen
3. Install Xen
4. Boot Xen
5. Configure XenoLinux - guest OS
6. Run XenoLinux - guest OS
7. List of necessary preinstalled packages
8. Get more information

All necessary console commands are written in `this font with a # to represent bash`. You can validate your results with the images.

PS All information which can be informative but might not be necessary for a read-trough and install is written in this style.

1 Get Xeno Source Code:

1.1 Fetch a local copy of BitKeeper tool:

Go to: *http://www.bitmover.com/download*

They will ask for a username and password. Use username BITKEEPER and password GET BITKEEPER in order to download the file. Download BitKeeper for *glibc23 (bk-3.2.2c-x86-glib23-linux.bin)*

1.2 Install BitKeeper:

If X[1] is not running on your system, make sure to add the install directory after the install command. In the directory you download bk-3.2.2c-x86-glib23-linux.bin, type the following in the console to install bk:

```
# ./bk-3.2.2.c-x86-glib23-linux.bin
```

[1] X window system - a client/server interface between display hardware and the desktop environment

PS The public master BK repository for the unstable release can be found at *bk://xen.bkbits.net/xeno-unstable.bk*

1.3 Get Xen-Unstable(2.0) source code

After Bitkeeper is installed we need to get the Xen source code. Type the following in the console to 1. go to the root for installing under / and 2. get the source code.

1. `# cd /`
2. `# bk clone bk://xen.bkbits.net/xeno-unstable.bk`

PS Under the current directory(/), a new directory named xeno-unstable.bk has been created, which contains all the source code for Xen and XenoLinux.

If you want to get the newest Xen-Unstable changes to the repository (It's updated continuously) write in the console:

1. `# cd /xeno-unstable.bk`
2. `# bk pull`

2. Build Xen:

After getting the source code it is time for building Xen-unstable. First we need to build the whole tree, then we have to configure it for the hardware and install it on the system.

2.1 Build the kernels

PS You need several packages installed in order to do this. Your Linux install might have the necessary packages. See the required package list in section 7.

First (1) we go the the repository top directory. Then (2) we build it for the first time:

1. `# cd /xeno-unstable/`
2. `# make world` (Builds Xeno-Unstable under the current directory)

Figure1 *Screen after successfully build xen*

2.2 Configure the kernel config file

After the build is finished you may need to edit the Xeno kernel configuration to make it work on your hardware. In order to configure the Xen domain0[2] kernel. Type the following commands:

1. `# cd /xeno-unstable.bk/install/boot/`
2. `# cp a config-2.4.26-xen0 configxenold` (backup configfile)
3. `# emacs config-2.4.26-xen0` (or pico/vi config..)

PS The Xen team changes the default settings for this file continuously, so make sure that the recent file is properly configured for your system. For example, I had to change the ethernet config from *CONFIG_E100 = y* to *#CONFIG_E100* is not set AND *#CONFIG_EEPRO100 is not set* to: *CONFIG_EEPRO100=y*

After the file is modified go back to the xeno-unstable.bk directory:

```
# cd /xeno-unstable.bk/
```

To submit the changes, if any, for the Xen0 kernel in the config-2.4.26-xen0 file, type :

```
# make -j4 world
```

3. Install Xen

PS You need several packages installed in order to do this. Your Linux install might have the necessary packages. See the package required in section 7 in order to install and run Xen-unstable.

For installing Xen on the system we first need to copy the kernel for domain 0 to the /boot/ directory. Then we need to install the xen tool software.

[2]The main domain for the virtual machine monitor

3.1 Copy the xen kernel to /boot/

Copy the kernel images to the /boot/ directory:

1. `# cd /xeno-unstable.bk/install/boot/`
2. `# cp xen.gz /boot/`
3. `# cp vmlinuz-2.4.26-xen0 /boot/`

3.2 Install the Xen tools

PS The xen tools are for management of the virtual machine monitor, creating new domains, list domains, connect to a domain e.g.

Type in the console:

```
# cd /xeno-unstable.bk/tools/
```

```
# make install
```

Fig2. Screen after successfully installing xen-tools

4. Boot Xen:

For booting the Xen you need a multiboot boot loader[3], which means that Lilo won't do the work. If you dont have grub installed as a boot loader go to section *4.1 Install grub*. If you have grub installed, go to *4.2 configure grub.conf*

[3] http://www.gnu.org/software/grub

4.1 Install Grub

PS You need to download the grub package if is not on your system!

To install Grub on the MBR of the first IDE disk:

```
# grub-install /dev/hda
```

4.2 Configure Grub

PS Depending on the Linux distribution, there are two files which can configure the Grub boot loader: grub.config and menu.lst. You can edit any of them if both exists.

Edit the grub.config or menu.lst file under /boot/grub/:

1. `# cd /boot/grub`
2. `# emacs menu.lst` (vi/pico menu.lst)

PS Some distributions have a automagic kernel lists. Make sure you add text AFTER the automagic section in the file.

Add the following in the file:

#start input#

title Xen-unstable (2.4.26)

root(0,0)

kernel /boot/xen.gz dom0_mem=131072 com1=115200,8n1 noht

module /boot/vmlinuz-2.4.26-xen0 root=/dev/hda1 ro console=tty0

#end input#

PS If your root is on another partition than hda1, please edit root=/dev/

4.3 Debugging (optional)

If the Xen kernel hangs on bootup it is easier to debugg with a serial output to another computer. You can use a Windows computer with Hyperterminal and a serial cable. To get the serial output edit the following in the grub.conf or menu.lst file:

1. Change *console=tty0* to *console = ttyS0 (console=ttyS0)*

2. Connect a serial cable to another computer. In this way you get the dmesg output in for example Hyperterminal (Win XP). Remember to use the same settings as under com1= in Hyperterminal.

3. Add a line to /etc/inittab: *c:2345:respawn:/sbin/mingetty ttyS0*

PS If you boot over a network you can edit the default option on the top of the menu.lst file. Default is 0, which means the first title. If Xen is title number two, change default to: default 1. If you boot local just choose kernel on bootup.

5. Configure XenoLinux:

If Xen boots up it is time to start the Xend deomon and create new domains. Each domain should have its own configuration file for dedicating CPU, disk, ip address e.g, so we need to do some configuration before we can create a new domain.

PS See lists of packages in order to install and run Xen in section 7.

5.1 Start Xend

First we need to run xend in order to use the xen tools. At any directory location type:

```
# xend start
```

PS You can also type *'xend restart'* or *'xend stop*'. Now we can use *xm*, the Xen domain management tool, in order to create, destroy, check status of the domains e.g.

5.2 Use Xen domain management tool - xm

For all options type:

```
# xm help
```

The most important switches are:

- xm list: Lists all domains running.
- xm consoles: Gives information about the domain consoles.
- xm console: open a console to a domain.
- xm create: Creates a new domain:

The following command is just an Example, don't type this before you finish section 5.3:

`# xm create -f xmdconfig vmid=1` xmdconfig is the domains config file and vmid is the domain id

Before we can type this we need to make a config file for each domain we want to create! xm need to know which config file to use. In this config file you define where to find HD space, which CPU to use, ipsetup e.g. Lets make a config file for our first domain (1)

5.3 Edit config files for the domains

First we need to go to /etc/xen, the location of the default config file to make a copy and edit it for domain 1.

PS The default xmdefaults is not able to run any domain. You have to edit this file to create a domain!

1. `# cd /etc/xen`
2. `# cp -a xmdefaults xmd1` the name xmd1 is just an example
3. `# pico xmd1` or emacs, vi e.g.

After running pico,vi,emacs e.g. we can edit the xmd1 file. We need to set up things properly in order to make a domain running.

1. Go to Kernel image file and edit the path to the guests kernel image. In our example the image is in the /xeno-unstable.bk/install/boot/ path. The line should look like this: *kernel = /xeno-unstable.bk/install/boot/vmlinuz-2.4.26-xenU*
2. (Optional) The domain gets 64 MB by default. You might want to edit this under Initial memory allocation for new domain.
3. (Optional) If you are running Xen on a multi processor system you might want to edit how Xen dedicates CPUs to the domains under Which CPU to start domain on. For most purposes I recommend to change this to *cpu = -1* (leaves up to Xen to pick CPU). If you let the mod of the id number decide you might end up with one domain running on one CPU and three on another!
4. Dedicate ip address under define network interfaces: Either configure the networks DHCP server to respond to the new domainss MAC address or edit xmd1 to configure an IP address, netmask and gateway manually. If you want it to take the config from whatever you have in dom0 and add 1 to the IP address, add under *#dhcp=dhcp* in the xmd1 file:

#dhcp=dhcp

ip = add_offset_to_ip(get_current_ipaddr(),vmid)

netmask = get_current_ipmask()

gateway = get_current_ipgw()

If you want to use DHCP, either go within the generated MAC address, or set your own with:

vif = [mac=00:06:AA:F6:BB:B3]

dhcp=dhcp

Finally (5) we need to define the disk devices you want the domain to have access to. Go to *"Define the disk devices... "* There are several options, you only need to choose one of them! The author of this document has only tested option 3, and is recommended for this tutorial.

PS Choose only one of these options, number three is recommended! In the final release for Xen 2.0 you can ignore this step. It will do step 3 automatical.

1. Give each domain its own physical partition. In order to do this you need to make disk = [phy:hda2,sda1,w]
2. Use LVM[4] to dynamically chop the partition up into volumes, each of which you can assign to a domain e.g disk = [phy:/dev/vg01/vm%d,sda1,w%(vmid)]
3. Put a file system on the partition and export files to domains using the loopback device (losetup). This enables you to use sparse files, allocating disk space on demand.

How to put a file system on the partition and export files, a howto for point number 3 above:

1. `# dd if=/dev/zero of=vm1disk bs=1k seek=2048k count=1` Creates a 2GB sparse file (actually only consumes 1KB of disk)
2. `# losetup /dev/loop0 vm1disk` Choose a free loop back device and attach file
3. `# mkfs -t ext3 /dev/loop0` Make a file system on the loop back device
4. `# mount /dev/loop0 /mnt` and `# cp -ax / /mnt` Populate the file system e.g. by copying from the current root:

[4] Logical Volume Management - allows you to create logical volumes out of the physical storages

5. Tailor the file system by editing /etc/fstab /etc/hostname etc. PS! In the mounted file system /mnt/etc/fstab e.g. For this example put /dev/sda1 to root in fstab.
6. `# umount /dev/loop0` Unmount!

If you go for option number 3 (loopback devices) set under *"disk devices"* in the xmd1 config file:

disk = [phy:loop0,sda1,w]

PS Two domains SHOULD ONLY share a disk if set to read only e.g. disk = [phy:loop0,sda1,r] Never let two domains share the same config file or disk without setting disk to read only!

PS As you write to the disk, the sparse file will become filled in and consume more space up to the original 2GB. In near future xend can track which loop devices are currently free and can do the allocating itself.

Save the xmd1 file!

6 Ready to run XenoLinux:

Remember to start xend:

```
# xend start
```

To start a new domain (1):

```
# xm create -f xmd1 vmid=1
```

To open a console to domain 1:

```
# xm console 1
```

PS The domain boots up and you should see a "normal" bootup screen.

7.Required packages:

ip-route

make

python-dev

gcc

zlib1g-dev

libcurl12

python2.3-pycurl

python2.3-twisted (2.2 not supported)

bridge-utils

8 More information

It‘s probably a good idea to join the xen-devel mailing list at http://lists.sourceforge.net/lists/listinfo/xen-devel