



## Tips & Tricks

 [Click To Print](#)

### Tips & Tricks

April 2008 • Vol.8 Issue 4

Page(s) 98-99 in print issue

# Warm Up To Penguins

## The Zen Of Xen

One of the hottest recent topics in the computer industry at this time is virtualization. Products such as VMware and Microsoft's Virtual PC are trying to take advantage of the increasing demand to run virtual computers on top of a physical host. There are several reasons to do this. If, for example, you need to run a number of different versions of Linux from time to time, you can run them as virtual systems on top of a stable host without worrying about constantly reinstalling or corrupting your system.



For businesses, virtual machines let many users share a single physical host. Because a system rarely runs at 100% load all the time, you can usually load-balance and get more bang for your hardware buck with virtual systems.

Linux, of course, has a virtualization application, too: Xen. Xen is an open-source project organized by XenSource ([www.xen-source.com](http://www.xen-source.com)), which is in turn owned by Citrix, known for GotoMyPC ([www.gotomypc.com](http://www.gotomypc.com)), among others. Xen consists of a hypervisor, which is simply a software package that enables virtual hosts to run on Linux, and the virtual hosts themselves. If you've ever used VMware, there are some significant differences between how VMware (and Virtual PC) handles virtualization and how Xen does it. For example, VMware runs as a program (or service) that the virtual hosts run inside, but Xen actually is part of the Linux Kernel. The "host" machine is just another virtual machine (or domain, as Xen calls them) running on the hardware. Although this may seem a bit strange conceptually, it seems to pay off performance-wise.

The good news about this approach is that it is more like the VMware ESX Server strategy, with the virtual machines running "close to the metal." In other words, it minimizes the overhead of managing the virtual machines. Xen enthusiasts claim that it uses as little as one-tenth of the overhead resources as traditional virtualization products. However (You knew there was going to be a however, didn't you?), it's definitely a more challenging product to install and use compared to VMware Server. You know those car ads that include a warning: "Professional driver, closed course"? Well, for the average user, installing Xen should come with the same kind of warning.

### ■ When 32 Is Greater Than 64

Before we really discuss Xen, we need to talk about processors. If you're running a 64-bit processor, you may have been tempted to install a 64-bit version of Linux. Although this has some benefits, it's going to cause nothing but headaches when you try to use Xen.

To understand why, you have to start by knowing that, with the exception of some of the very latest processors, you can't run virtual machines under Xen that are meant for a different architecture type. So, for example, if you install a 64-bit Linux distribution as your base operating system, your dom0 (the base administrative virtual machine) will need to be a 64-bit version of Linux. More to the point, the other virtual machines (referred to as domU images) will also need to be 64-bit images.



So what's the issue? First, you almost certainly don't want to build a domU image from scratch. Unlike VMware, where you just boot from the installation media and install the OS, you need to start with a special version of the domU's OS installation software when installing a Xen image. That's because Xen requires special modifications to the version of the OS that you install as a domU image. For example, if you want to install OpenSolaris, you need to get your hands on the



**Companies are increasingly using virtualization technologies like Xen to increase computing resource flexibility.**

special Xen OpenSolaris installation image, not the generic OpenSolaris media you'd use to install OpenSolaris under typical circumstances.

The very bad news (for 64-bit users) is that there are many more 32-bit domU images out there right now than 64-bit ones. So, you could go through the trouble of installing a 64-bit Linux distribution, getting a Xen dom0 image running on it, only to find that you couldn't find anything interesting to run as domU guests. Now, if you want to use Xen for pure virtualization of the same operating system, this won't be an issue. You can merrily install a 64-bit version of Linux. But if you plan to use Xen to run heterogeneous operating systems (such as FreeBSD and OpenSolaris on top of Linux), you'll find things quite difficult if you started with a 64-bit Linux. In those cases, the solution is to install the 32-bit version of your distribution as the base operating system.

#### ■ Install Xen

Without a doubt, the easiest way to install Xen is when you first install your distribution. Most modern distributions such as OpenSUSE will offer Xen as a package you can add during the initial installation process. If Xen isn't available or you want to add Xen to an existing distribution, the installation process will vary greatly according to your distro. Some, such as those in the Debian family, have a number of useful Xen-related tools that you can also install. Your best bet is to check your distro's forum/wiki. You can also try Xen's Web page for distribution-specific information.

Once you install Xen, you need to reboot your system. If you did things correctly, you'll see a new option in your bootloader, one that starts with Xen. If you select this, you'll boot into Xen with the dom0 (administrative image) attached to your console. It will look a lot like Linux normally would, so the best way to determine that you're really running inside Xen is to use the "xm list" command as shown. (See figure 1.)

We can see domain 0 is running. So now we're ready to install our first guest domain (a domU, in Xen parlance, if you recall). Now, we consider ourselves fairly savvy with Linux, but after a solid week trying to get an example that demonstrated running a cool foreign operating system using Xen and failing miserably, we were forced to fall back on installing a second copy of Debian . . . under Debian. As it turns out, if you install the "xen-tools" package, this is relatively simple to do. You just use a command like:

```
#xen-create-image hostname=test.my.etch \  
--ip=192.168.1.228 --dir=/xen/etch1  
--dist=etch --debootstrap
```

This essentially tells Xen to create a new image running the Debian "Etch" distribution with an IP address of 192.168.1.228 and using the /xen/etch1 directory to hold the image files that contain the root and swap file partitions for the virtual machine.

Now, when we say "relatively easy," we mean it only took us a few hours reading Web pages and forums to find the three things we needed to change to get things to work. (We needed to change the kernel and initrd settings in our xen-tools.conf, install the network bridging package, and configure our Xen installation to set up bridged networks. However, once that was done, we were able to run the xen-create-image command and then start the image with the command: # xm create test.my.etch.cfg. To connect to our new domain, we used the following xm console command: roadwarrior:/xen# xm console test.my.etch.

#### ■ Professional Drivers Only

Reading this article, you might think that we're not big fans of Xen in its current form. And you'd be right. Xen represents what we hate most about some open-source projects: the attitude that making it easy to use is of secondary importance. The frustrating thing is that Xen, at face value, is tremendously powerful and could be a wonderfully useful tool to have on your belt. But right now, it's an adventure into the darkest depths of the jungle to get it working for even the simplest cases. Xen apologists will disagree, but the reality is that Xen is a powerful tool that currently needs a PhD to operate. ■

by James Turner

## Figure 1

```
james@james-laptop:~$ sudo xm list
```

```
Password:
```

```
Name ID Mem(MiB) CPUs State Time(s)
```

```
Domain-0 0 1873 2 r----- 34.3
```

## Infinite Loop: Yellow Submarine Voyages To Outer Space

The rock 'n' roll band that dominated the planet in the '60s may soon dominate the cosmos, if NASA has anything to say about it. On Feb. 4, NASA transmitted the Beatles' single "Across the Universe," well, across the universe. Aimed at Polaris, the transmission commemorates the 40th and 50th anniversaries of the band's recording the song and NASA's founding, respectively. Plus, it'll show aliens our species has some musical talent.

Source: [www.networkworld.com/community/node/24574](http://www.networkworld.com/community/node/24574)