09
Oct
2009

Xen Virtualization: Migrating 32-bit domUs to 64-bit dom0s

Virtualization provides the facility to run multiple isolated computer operating systems on one piece of computing hardware. There has been a huge increase of interest in virtualization technology because recent advances in multi-core technology provide significantly more computing power in each machine with ever decreasing costs. Virtualization is one of the best ways to take advantage of these big changes in hardware.

Currently, Xen is the most mature FOSS (Free and Open Source Software) virtualization technology. Although we love the idea of KVM, since it requires a special processor extension on X86 systems, it cannot work on older hardware. So for at least another few years, we think Xen is the more flexible choice for FOSS virtualization projects.

The Xen infrastructure consists of the Xen hypervisor which "runs the show", a domain 0 (dom0) which runs a special, privileged version of the operating system (typically Linux, but NetBSD and Solaris are also supported), and one or more domain U (domU) "guest" (or "User") operating systems. We have found that Xen is easy to configure in many situations, but we encountered some complications in running a domU on a dom0 with a different architecture.

We recently migrated some 32-bit domUs running Debian Etch (4.0) from a 32-bit dom0 to a newer 64-bit dom0 running Debian Lenny (5.0). We did a direct move (using rsync) of the Logical Volume Manager (LVM) slices from the 32-bit dom0 to the 64-bit dom0. This means we'd now be running our 32-bit Etch domUs on a 64-bit Lenny dom0.

The first question was whether this would be possible. Absolutely! 32-bit domUs have no trouble running on 64-bit dom0s, we could even use the 64-bit Xen kernel in these 32-bit systems to avoid additional kernel installations we'd need to maintain on the dom0. The second question was whether we could properly load the 64-bit kernel modules inside our domU. Again, yes! But with a caveat: the domUs were 32-bit Etch, so the 64-bit Lenny kernel modules were not simply installable via apt. We realized that copying over the .deb package for the kernel modules and running `dpkg -i --force-architecture linux-modules-2.6.26...deb` would not be a maintainable way to handle the kernel module updates moving forward. So we weighed our options:

1. Serve these modules via a network file system (such as NFS) to each domU on bootup.

2. Deploy a script that would notify the domU and copy the new kernel modules .deb to it for installation. We could then install the new module package at our discretion.

We decided that the first option violated our strict security policy which calls for running as few services on the dom0 as possible. Since the second solution is scriptable and therefore automatable, it fit our vision of having easily maintainable systems regardless of the underlying complexity. So we installed the 64-bit modules prior to migration so that all the proper modules would be loaded as soon as we brought up the domU on the new Dom0. The result was a flawless migration of our 32-bit domUs to the new 64-bit dom0.

Posted by Elizabeth Krumbach *in* Tech Notes, Virtualization, *0 comments*