Xen Grid Site – the Art of Consolidation

Marcus Hardt*, Rüdiger Berlich

Forschungszentrum Karlsruhe, Institut für wissenschaftliches Rechnen,
Postfach 3640, 76021 Karlsruhe, Germany,
{hardt, berlich}@iwr.fzk.de

Abstract. This Paper describes how to set up an LCG-2.4 Training Box that comprises all the Grid Elements required for running a Grid site. The basic central services that are required for running a Grid are covered as well.

1 Introduction

The EDG-based Grid middleware LCG-2.4 was designed to scale from large to very large installations. This is unfortunate in training environments, where only the principles of the software and its installation are of importance, rather than the overall performance of the system. In such a scenario virtualisation software can prove to be useful, as it allows to execute several instances of an operating system on one physical box.

In this paper we describe how this can be accomplished using Xen, a virtualisation environment developed at the University of Cambridge. We will start with a description of how to set up a training site running on one box using Xen. Configuration details for both Xen and LCG are explained whenever required.

This paper is intended to be useful for training organisers as well as their students.

2 A few words about Xen

Xen [3] – often also called a Virtual Machine Monitor – allows to run several instances of an operating system on one physical machine. Since Virtualisation, as known from Mainframes, is unsupported on Xen's main target target, the Intel x86 Architecture, full virtualisation was not the main focus of the Xen team. Instead, the so called "*paravirtualisation*" is used. While only a very limited amount of priviledged calls has to be virtualised in order to run more than one Operating System concurrently, paravirtualisation implies modifications of the guest operating systems. Such modifications are usually limited to just a few thousand lines of code, compared to the approximately 6 million lines of code of the Linux kernel. As one design goal of Xen was to keep the application binary interface (ABI) unchanged, source code modifications are limited to the Operating System core. Application binaries will run inside virtual machines without any modifications.

To the system administrator Xen appears as a kernel modification. Both the kernels of the host and the guest systems need to be patched with code provided by the Xen developers. Precompiled kernel images are available, and Xen has meanwhile been integrated in a number of Linux distributions, with SuSE Linux 9.3 just being an example. The host system is usually referred to as Domain 0 or dom0 – guest instances are called domUs.

A set of userspace tools is provided for controlling the virtual instances. This comprises the basic functions, like creating and destroying virtual instances, connecting consoles to them as well as more advanced functions for stopping/starting, saving and even migrating domains between different physical hosts. The filesystem for guest systems can be placed into raw partitions, on LVM or just inside a file. They are made availabe via the virtual block device (VBD) driver to the guest domain. Booting a system via nfsroot is possible as well. Networking in the guest systems works almost instantly via bridging on the domain 0. MAC addresses are generated by Xen or can be specified in the configuration file holding all information necessary for starting the virtual machines.

2.1 Performance

Past attempts to run a Grid site on User Mode Linux (UML) were rather discouraging. Thus we compared the UML perfomance to that of Xen and to an unmodified reference system [1]. These measurements show that Xen reaches roughly between 95 and 90% of the reference performance, while UML roughly reaches around 75%, and ca even drop to 50% under high load. Also, the responsiveness and the "fairness" in time allocation made a significantly better impression on Xen virtual machines running under high load than on UML machines. For details see [1].

2.2 Migration

Xen allows to migrate domains to different machines while they are running. Network connections *stay active* during the migration. In the case of "Live migration", no noticeable down-time can be observed. First, all domain-specific data is copied. In the next step, only changes between the "pre-copying" stage and the actual migration are sent to the new host. The "ping" command to a domain during a live migration showed just one lost packet. Existing connections (e.g. ssh) were kept alive.

Unfortunately, we were only able to use migration in small test installations. This is because the image file needs to reside on shared network storage, so it does not have to be migrated. Using NFS to share the images between hosts was not possible because of a known issue of the Linux kernel when using loopback files over NFS. The Xen team suggests using GNBD (Network Block Device) or iSCSI (SCSI over IP). Root-mounted NFS may as well be an option.

3 Installation

Our installation took place on a cluster of dual P-III (700MHz), each equipped with a 40GB IDE harddrive and 1GB of RAM. While this system cannot compete with todays hardware, this configuration resulted in an installation of good performance. It thus appears to be possible to use systems with even lower specs. The amount of memory seems to be the most important variable.

3.1 Xen

The base system used with Xen can be any Linux distribution. Unfortunately our hardware seamed to cause trouble when using the 2.4-xen0 kernel and version 3 of Scientific Linux in dom0. We therefore decided to use a more modern distribution that was easier to handle (in our case Debian/Sarge). As is common with new environments such as Xen, the installation instructions were distributed around differend locations. We found the Debian specific-hints in the FAQ section to be very useful. The README found in the binary installation tarball of Xen version 2.0.5 – the version we used for this paper – explains how to run the install script and how to modify the only supported bootloader: Grub. The most useful piece of documentation, however, is the Xen user manual [6].

The Grub configuration is easily done. Just note that `kernel` options are not given to the kernel line in `menu.lst` but in the `module` line. The kerneloption `noreboot` is recommended, if you experience trouble at boot time and cannot see error messages otherwise. The kernel option `max_loop=16` is required to increase the limit of mountable loop file systems from 8 to 16.

Along with this the corresponding device nodes must be created one the host system:

```
for i in ‘seq 8 15‘; do mknod /dev/loop"$i" b 7 "$i"; done
```

It is highly recommended by the Xen team to disable the TLS libraries (e.g. by moving `/lib/tls` to `/lib/tls.disabled` in both dom0 and the domUs). If you don't do this, Xen will run an emulation that is both slow and unstable. TLS is an implementation of threads used by `glibc`, which is incompatible with Xen. After moving the TLS-directory, a Xen-compatible library version will be used. There is hope that this issue will be resolved soon.

3.2 Scientific Linux

The next step was to create a 4GB image with a Scientific Linux distribution that is ready for installation of the LCG software. If you have access to such a basic SL-3.0 installation, it can just be copied over into the images with the following commands. A swap-image is required on machines with less than 2GB Ram as well.

```
# Create an empty 4GB image an mount it
dd if=/dev/zero bs=1M count=4k of=sl-3.0.4.img-1
```

```
mke2fs -j -f sl-3.0.4.img-1
mount -o loop sl-3.0.4.img-1 mnt

# Create a 512MB swap image
dd if=/dev/zero bs=1M count=512 of=swap-1
mkswap swap-1

# Copy existing installation from the sl-3 machine
cd mnt
ssh -c blowfish <machine> "cd /; tar csp <all files except proc,
   lost+found, afs, sys and nfs mounted areas>" | tar xsp

# Install the required kernel modules
cp -af /lib/modules/*xen0 .

# Setup the root password
chroot .
passwd
exit
```

Finally edit `etc/fstab` and change the root partition of your image to be `/dev/hda1`. Similarly, the swap partition could be installed on `/dev/hda2`.

You should be able to boot the image now, using a configuration file similar to the one found on `https://savannah.fzk.de/distribution/internal/xen-grid-site/images` Savannah project [8] in the files area. There you can also download a gzipped image that contains all the above modifications, based on an SL-3.0.4.

Invoking YAIM We will carry out the common parts of the LCG installation still on one image and later create the copies for the individual hosts. The image needs to be booted in order to continue. The command

```
xm create -c sl-3.0.4.conf vmid=1
```

will boot your image. You can detach the console by hitting "`C-]`" a few times, and reallocate it with the `xm console <DOM>` command. It is a good idea to have a thorough look at the functions provided by `xm help` and `xm help <function>`.

From now on we essentially follow the instructions given in [12].

NTP server configuration

```
apt-get update
apt-get dist-upgrade -u
apt-get install ntp libcap libcap-devel
echo 141.52.27.35 > /etc/ntp/step-tickers
ntpdate 141.52.27.35
```

Maybe you want to choose a different ntp server.

NFS Most commonly `$HOME` is shared among all nodes on a site. Now is a good point to set up the mounts for the server and the clients. Useful commands are:

```
# On the NFS server:
mkdir /home/cluster_homes
echo "/home/cluster_homes     141.52.167.0/24(rw,no_root_squash)"
     >> /etc/exports
apt-get install nfs-kernel-server

# Adjust firewall to allow general access from the clients
iptables -A INPUT -s 141.52.167.0/24 -j ACCEPT
iptables -A INPUT -s 127.0.0.0/8 -j ACCEPT
/etc/init.d/firewall restart
# end NFS Server

# On the NFS client
echo -e "nfs.fzk.de:/home/cluster_homes\t/home\tnfs\tdefaults\t0 0"
     >> /etc/fstab
mount -a
```

YAIM installation Install the YAIM rpm and configure it according to the documentation found in [12]. It is handy to place the yaim configuration files into a directory that is shared across all nodes.

Now the images are ready to be multiplied in order to have one image per virtual node we need to start.

```
for i in ‘seq 2 5‘; do
    cp sl-3.0.4.img-1 sl-3.0.4.img-\$i;
    cp swap-1 swap-\$i
done
```

We decided to not focus on sharing other directories than `/home` (e.g. `/usr`, `/lib`) among the virtual instances. This is left as an excercise to the reader. However, please note that upgrading and installing packages might work less straightforward.

Xen specific issues There was only one single area where LCG and Xen did not work together well. This is the EDG version of the openldap services. They are compiled with `libgd4` support, which depends on the libraries used by `glibc`. The openldap server shipped with Scientific Linux does not have this problem. And as long as it carries a version number below `2.0.27` is is still compatible with the information schema currently used.

If you're using a YAIM version newer than `lcg-yaim-2.4.0-3` you can probably skip the patch to `/opt/lcg/yaim/functions/config_bdii` [9]. This patch allows to define a different location for `slapd` and `slapadd` in the yaim configuration file, using

```
BDII_SLAPD_LOCATION=/usr/sbin/slapd
BDII_SLAPADD_LOCATION=/usr/sbin/slapadd
```

LCG installation Finally you can install the LCG distribution using commands similar to the following ones

```
/opt/lcg/yaim/scripts/install_node
       /opt/lcg/yaim/examples/site-info.def lcg-SECLASSIC
/opt/lcg/yaim/scripts/configure_classic_SE
       /opt/lcg/yaim/examples/site-info.def
```

We suggest to follow the instructions given on [12].

4 Acknowledgements

We would like to thank the Xen developers for their consistently short response times for all our questions we had, and of course for developing Xen in the first place.

We would furthermore lile to thank the computer science group of Brian Coghlan at the Trinity College in Dublin for support in the idea of building a Grid in a box.

The EU project EGEE and Forschungszentrum Karlsruhe for funding our work and for providing the hardware we worked on.

References

1. M. Hardt, R. Berlich "Virtualisierung mit Xen", Linuxmagazin, July 2005, Germany
2. Childs, S., Coghlan, B., O'Callaghan, D., Quigley, G., Walsh, J. (2005) "A single-computer Grid gateway using virtual machines" AINA'05, Taiwan, March, 2005
3. Xen project homepage `http://xen.sf.net`
4. XenoServer Project homepage
   `http://www.cl.cam.ac.uk/Research/SRG/netos/xeno`
5. Xen and the Art of Virtualization:
   `http://www.cl.cam.ac.uk/netos/papers/2003-xensosp.pdf`
6. `http://www.cl.cam.ac.uk/Research/SRG/netos/xen/readmes/user.pdf`
7. XenSource, the commercial part of Xen: `http://www.xensource.com`
8. Xen Grid Site Project homepage:
   `http://savanah.fzk.de/projects/xen-grid-site`
9. Patch to the Yaim script to configure lcg_bdii
   `https://savannah.fzk.de/distribution/internal/xen-grid-site/patches`
10. The LHC Computing Grid Project: `http://lcg.web.cern.ch`
11. LCG deployment website `http://grid-deployment.web.cern.ch`
12. LCG install instructions
    `http://web.cern.ch/grid-deployment/documentation/LCG2-Manual-Install`