xCAT 2 Cookbook for Linux

10/07/2008

(Valid for both xCAT 2.0.x and pre-release 2.1)

Table of Contents

1.0 Introduction

xCAT 2 is a complete rewrite of xCAT 1.2/1.3, implementing a new architecture. All commands are client/server, authenticated, logged and policy driven. The clients can be run on any OS with Perl, including Windows. The code has been completely rewritten in Perl, and table data is now stored in a relational database.

This cookbook provides step-by-step instructions on setting up an example stateless cluster. For completeness, some advanced topics are covered, like hierarchical management (for extremely large clusters), compute nodes with large pages, NFS hybrid mode, mixed node architectures, and accelerator nodes. If you do not intend to use some of these features, skip those sections. (Section 1.2 will tell you which sections to skip.) The example cluster in this document is built with Fedora 8, but the same concepts apply to Fedora 9, RHEL 5, and (to a lesser extent) SLES 10.

1.1 Stateless and Stateful Choices

Stateless nodes are an important concept in xCAT 2. A stateless node is defined as one that has no "state" (configuration changes, software updates, etc.) stored permanently on it. This is extremely useful in a cluster for the following reasons:

- All nodes will have a much greater likelihood of staying consistent. And if the administrator does suspect that a node is out of sync with the rest of the cluster, they can simply reboot it and know that it is back in its original, pristine state.
- If a node experiences a hardware problem, the hardware can be pulled from the rack and replaced with new hardware and the node booted again and it will come up with the same state as before.
- In a provisioning environment, new nodes can be provisioned or moved without the worry of them losing state.

xCAT 2 provides the choice of either stateless or stateful nodes. A stateful node is one that has the OS installed on its local hard disk and therefore, changes to the node (configuration changes, software updates, etc.) can be made over time and those changes will persist.

Stateless nodes in xCAT 2 are implemented by not putting the OS on the local disk of the node. There are 3 choices for stateless:

1. **RAM-root** – The entire OS image is contained in a RAM file system that is sent to the node when it boots. Typical size for a minimal compute node for Linux is 75-160 MB of memory.
2. **Compressed RAM-root** – The OS image is in a compressed tar file. Individual files are extracted and cached when read. File writes are done to the cached copy. Typical size for a minimal compute node for Linux is 30-64 MB of memory.
3. **NFS Hybrid** – This is more accurately called NFS-root with copy-on-write. A minimal boot kernel is sent to the node, which readonly NFS mounts the OS image from the server. Files read are cached in memory. File writes are done to the cached copy. Typical size for a minimal compute node for Linux is 5 MB of memory.

1.2 Scenarios

The following scenarios are meant to help you navigate through this document and know which sections to follow and which to ignore for an environment that is similar to yours.

1.2.1 Simple Cluster of Rack-Mounted Servers – Stateful Nodes

- Use the xCAT iDataPlex cookbook: http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT-iDpx.pdf

1.2.2 Simple Cluster of Rack-Mounted Servers – Stateless Nodes

- Follow chapter 2 to install the xCAT software
- Use chapter 3 as an example for defining your BMC-base stateless nodes (skip the postgreSQL section and the servicenode table section)
- Follow chapter 4 to configure the management node
- Follow chapter 6 to boot the stateless nodes
- Optionally follow chapters 12 and 13 to install Torque and Moab

1.2.3 Simple BladeCenter Cluster – Stateful or Stateless Nodes

- Follow chapter 2 to install the xCAT software
- Follow chapter 4 to configure the management node

- Follow chapter 5 to define the compute nodes in the xCAT database, except that instead of using the service node as the conserver and xcatmaster, use the management node hostname.
- If you want the nodes to be stateful (full operating system on its local disk) follow chapter 7
- If you want the nodes to be stateless (diskless) follow the example of booting the LS21 blades in chapter 9
- Optionally follow chapters 12 and 13 to install Torque and Moab

1.2.4 Hierarchical Cluster - Stateless Nodes

- Follow essentially the whole document, but the following sections are optional, depending on your environment:
  - If you do not have QS22 blades, skip chapter 8 and chapter 10 and the parts of chapter 9 that refer to QS22.
  - If you do not want to use NFS Hybrid mode, skip chapter 11.

1.3 Other Documentation Available

- xCAT web site: http://xcat.sf.net/
- xCAT man pages: http://xcat.sf.net/man1/xcat.1.html
- xCAT DB table descriptions: http://xcat.sf.net/man5/xcatdb.5.html
- Installing xCAT on iDataPlex: http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT-iDpx.pdf
- Using LDAP for user authentication in your cluster: http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2.ldap.pdf
- Monitoring Your Cluster with xCAT: http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2-Monitoring.pdf
- xCAT on AIX Cookbook: http://xcat.svn.sourceforge.net/svnroot/xcat/xcat-core/trunk/xCAT-client/share/doc/xCAT2onAIX.pdf
- xCAT wiki: http://xcat.wiki.sourceforge.net/
- xCAT mailing list: http://xcat.org/mailman/listinfo/xcat-user
- xCAT bugs: https://sourceforge.net/tracker/?group_id=208749&atid=1006945
- xCAT feature requests: https://sourceforge.net/tracker/?group_id=208749&atid=1006948

1.4 Cluster Naming Conventions Used in This Document

Throughout this doc, an example node naming convention is used to demonstrate how to use your naming patterns to reduce the amount of input you need to give to xCAT. The example name convention is:

- All node names begin with “rr”.

- The cluster is divided into management sub-domains called connected units (CU). Each CU has its own subnet (and broadcast domain) and is designated by a single letter. So the 1st CU is rra, the 2nd rrb, etc.
- Within each CU, the nodes are grouped into threes (designated by a, b, c) and then the groups are numbered sequentially: rra001a, rra001b, rra001c, rra002a, etc. In this particular example, the "a" node is an opteron node, and the "b" and "c" nodes are accelerator Cell nodes for the opteron node.
- Each CU has a service node that acts as an assistant management node on behalf of the main management node. The service node has 2 ethernet adapters: the adapter on the management node side is named, for example, rra000-m, and the adapter on the CU compute node side is named, for example, rra000.
- The BladeCenter chassis within each CU are numbered sequentially, e.g. bca01, bca02, etc.

2.0 Installing the Management Node

2.1 Prepare the Management Node

2.1.1 Set Up Your Networks

xCAT install process will scan and populate certain settings from the running configuration. Having the networks configured ahead of time will aid in correct configuration.

2.1.2 Install the Management Node OS

It is recommended to ensure that dhcp, bind (not bind-chroot), expect, httpd, nfs-utils, vsftpd, and perl-XML-Parser are installed. If the management server will be on the network and RHN activated or yum is pointed to the Fedora repositories, these installs will happen automatically later if not done now.

2.1.3 Ensure That SELinux is Disabled

/etc/sysconfig/selinux should contain:

```
SELINUX=disabled
```

If this change had to be made, reboot the system.

2.1.4 Prevent DHCP client from overwriting DNS configuration

Find the /etc/sysconfig/network-scripts/ifcfg-* files relevant to any NICs that are DHCP configured, and put "PEERDNS=no" into them.

2.1.5 Configure Cluster-Facing NICs

Configure the cluster facing NICs. An example /etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
```

```
IPADDR=11.16.0.1
NETMASK=255.255.0.0
```

2.1.6 Configure Hostname

/etc/sysconfig/network should have HOSTNAME=(desired hostname).

2.1.7 Configure DNS Resolution

/etc/resolv.conf should at least point to its own DNS (which will get set up later). For example:

```
search cluster
nameserver 11.16.0.1
```

2.1.8 Set Up basic hosts file

Ensure a line like the following is in /etc/hosts:

```
11.16.0.1 mn20.cluster mn20
```

2.1.9 Restart Management Node

Though it is possible to restart the correct services for all settings except SELinux, the simplest step would be to reboot the management server at this point.

2.1.10 Configure Ethernet Switches

xCAT can use the ethernet switches for discovery. In general, this requires that the user in advance set up an ip address and basic snmp functionality. Allowing the snmp version 1 community string "public" read access will allow xCAT to communicate without further customization. It is also recommended that spanning tree be set to portfast or edge-port for faster boot performance. Please see the relevant switch documentation as to how to configure these items.

2.2 Download Linux Distro ISOs and Create Repository

1. Get Fedora ISOs and place in a directory, for example /root/xcat2:

```
mkdir /root/xcat2
cd /root/xcat2
export BASEURL=ftp://download.fedora.redhat.com/pub/fedora/linux/releases/8
wget $BASEURL/Fedora/x86_64/iso/Fedora-8-x86_64-DVD.iso
wget $BASEURL/Fedora/ppc/iso/Fedora-8-ppc-DVD.iso
```

2. Create YUM repository for Fedora RPMs (not needed on SLES):

```
mkdir /root/xcat2/fedora8
mount -r -o loop /root/xcat2/Fedora-8-x86_64-DVD.iso /root/xcat2/fedora8

cd /etc/yum.repos.d
mkdir ORIG
mv fedora*.repo ORIG
```

```
Create fedora.repo with contents:

[fedora]
name=Fedora $releasever - $basearch
baseurl=file:///root/xcat2/fedora8
enabled=1
gpgcheck=0
```

On SLES, get access to the SLES RPMs and run "zypper sa <url>" to point to them.

3. Install createrepo (not needed on SLES):

```
yum install createrepo
```

2.3 Downloading and Installing xCAT 2

2.3.1 Choosing the Version of xCAT You Want to Use

You can choose between 3 different versions of xCAT for Linux:

- The latest **official release** of the **stable branch**, 2.0.x:
  - YUM directory: http://xcat.sourceforge.net/yum/xcat-core
  - Tarball: http://xcat.sourceforge.net/yum/xcat-core-2.0.2.tar.bz2 (or pick the latest 2.0.x version)
- The latest **snapshot** of the **stable branch**:
  - YUM directory: http://xcat.sourceforge.net/yum/core-snap
  - Tarball: http://xcat.sourceforge.net/yum/core-rpms-snap.tar.bz2
- The latest **snapshot** of the **development branch**, 2.1 pre-release:
  - YUM directory: http://xcat.sourceforge.net/yum/devel/core-snap
  - Tarball: http://xcat.sourceforge.net/yum/devel/core-rpms-snap.tar.bz2

In all cases, use the same package of dependencies:

- YUM directory: http://xcat.sourceforge.net/yum/xcat-dep
- Tarball: http://sourceforge.net/project/showfiles.php?group_id=208749&package_id=258529 (choose latest version number)

Now use the appropriate links you've chosen above in section 2.3.2 or 2.3.3.

2.3.2 If Your Management Node Has Internet Access:

2.3.2.1 Download Repo Files

YUM can be pointed directly to the xCAT download site. For example:

```
cd /etc/yum.repos.d
wget http://xcat.sf.net/yum/core-snap/xCAT-core-snap.repo
wget http://xcat.sf.net/yum/xcat-dep/rh5/x86_64/xCAT-dep.repo
```

Or on SLES, also do:

```
zypper sa http://xcat.sf.net/yum/core-snap
zypper sa http://xcat.sf.net/yum/xcat-dep/sles10/x86_64
```

2.3.2.2 Set Up Repo File for Fedora Site

Create fedora-internet.repo:

```
[fedora-everything]
name=Fedora $releasever - $basearch
failovermethod=priority
#baseurl=http://download.fedora.redhat.com/pub/fedora/linux/releases/
   $releasever/Everything/$basearch/os/
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
   $releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/RPM-GPG-
   KEY
```

Continue now at step 2.3.4, Install xCAT 2 software & Its Dependencies.

2.3.3 If Your Management Node Does Not Have Internet Access:

2.3.3.1 Download xCAT 2 and Its Dependencies

Note: do the wget's on a machine with internet access and copy the files to the management node.

```
cd /root/xcat2
wget http://xcat.sf.net/yum/core-rpms-snap.tar.bz2
wget http://downloads.sourceforge.net/xcat/xcat-dep-2.0.1.tar.bz2?use_mirror=osdn
   # choose latest version available by browsing
   https://sourceforge.net/project/showfiles.php?group_id=208749&package_id=258529
tar jxvf core-rpms-snap.tar.bz2
tar jxvf xcat-dep-2*.tar.bz2
```

2.3.3.2 Get Distro Open Source Dependencies from Fedora Site

```
cd /root/xcat2/xcat-dep/rh5/x86_64
export
   BASEURL=http://download.fedora.redhat.com/pub/fedora/linux/releases/8/Everything
   /x86_64/os/Packages/

wget $BASEURL/perl-Net-SNMP-5.2.0-1.fc8.1.noarch.rpm
wget $BASEURL/perl-XML-Simple-2.17-1.fc8.noarch.rpm
wget $BASEURL/perl-Crypt-DES-2.05-4.fc7.x86_64.rpm
wget $BASEURL/net-snmp-perl-5.4.1-4.fc8.x86_64.rpm
wget $BASEURL/ksh-20070628-1.1.fc8.x86_64.rpm
wget $BASEURL/perl-IO-Socket-INET6-2.51-2.fc8.1.noarch.rpm
wget $BASEURL/dhcp-3.0.6-10.fc8.x86_64.rpm
wget $BASEURL/syslinux-3.36-7.fc8.x86_64.rpm
wget $BASEURL/mtools-3.9.11-2.fc8.x86_64.rpm
```

```
wget $BASEURL/expect-5.43.0-9.fc8.x86_64.rpm
wget $BASEURL/perl-DBD-SQLite-1.12-2.fc8.1.x86_64.rpm
wget $BASEURL/perl-Expect-1.20-1.fc8.1.noarch.rpm
wget $BASEURL/perl-IO-Tty-1.07-2.fc8.1.x86_64.rpm
wget $BASEURL/scsi-target-utils-0.0-1.20070803snap.fc8.x86_64.rpm
wget $BASEURL/perl-Net-Telnet-3.03-5.1.noarch.rpm

createrepo .
```

2.3.3.3 Set Up YUM repositories for xCAT and Dependencies

```
cd /root/xcat2/xcat-dep/rh5/x86_64
./mklocalrepo.sh
cd /root/xcat2/core-snap
./mklocalrepo.sh
```

Or on SLES, also do:

```
zypper sa file:///root/xcat2/core-snap
zypper sa file:///root/xcat2/xcat-dep/sles10/x86_64
```

2.3.4 Install xCAT 2 software & Its Dependencies

```
yum clean metadata
yum install xCAT.x86_64
```

Or on SLES, do:

```
zypper install xCAT
```

2.3.5 Test xCAT installation

```
source /etc/profile.d/xcat.sh
tabdump site
```

2.3.6 Update xCAT 2 software

If you need to update the xCAT 2 rpms later:

- If the management node does not have access to the internet: download the new version of http://xcat.sf.net/yum/core-rpms-snap.tar.bz2 and untar it in the same place as before
- (If the management node has access to the internet, the yum command below will pull the updates directly from the xCAT site.)

Then run:

```
yum update '*xCAT*'
```

Or on SLES, do:

```
zypper update '*xCAT*'
```

If you have a service node stateless image, don't forget to update the image with the new xCAT rpms (see chapter 6.1, Build the Service Node Stateless Image):

2.3.7 Set Up the Install Directory for Fedora8 Node Installs

```
umount /root/xcat2/fedora8
cd /root/xcat2
copycds Fedora-8-x86_64-DVD.iso
copycds Fedora-8-ppc-DVD.iso
```

The copycds commands will copy the contents of the DVDs to /install/fedora8/<arch>.

Edit /etc/yum.repos.d/fedora.repo and change:

```
baseurl=file:///root/xcat2/fedora8
```

to

```
baseurl=file:///install/fedora8/x86_64
```

3.0 xCAT Hierarchy Using Service nodes

In large clusters it is desirable to have more than one node (the Management Node) handle the installation and management of the compute nodes. We call these additional nodes service nodes. You can have one or more service nodes set up to install & manage groups of compute nodes. With xCAT, you have the choice of either having each service node install a distinct set of compute nodes, or having a pool of service nodes, any of which can respond to an installation request from a compute node. This document will cover the former case (distinct sets).

The service nodes need to communicate with the xCAT 2 database on the Management Node and run xCAT commands to install the nodes. The service node will be installed with the xCAT code and requires that the PostgreSQL Database be set up instead of the SQLite Default database. PostgreSQL allows a remote client to be set up on the service node such that the service node can access (read/write) the database on the Management Node.

If you do not plan on using service nodes, you can skip this chapter 3 and continue to use the SQLite Default database.

3.1 Switching to PostgreSQL Database

To set up the postgresql database on the Management Node follow these steps.

This example assumes:

- 11.16.0.1: IP of management node (cluster-facing NIC)
- xcatdb: database name
- xcatadmin: database role (aka user)
- cluster: database password
- 11.16.1.230 & 11.16.2.230: service nodes (mgmt node facing NIC)

Substitute your addresses and desired userid , password and database name as appropriate.

The following rpms should be installed from the Fedora8 media on the Management Node (and service node when installed). These are required for postgresql.

1. `yum install perl-DBD-Pg postgresql-server postgresql  # or use zypper for SLES`

2. Initialize the database :
   `service postgresql initdb`
3. `service postgresql start`
4. `su - postgres`
5. `createuser -SDRP xcatadmin`
   `Enter password for new role: cluster`
   `Enter it again: cluster`
6. `createdb -O xcatadmin xcatdb`
7. `exit`
8. `cd /var/lib/pgsql/data/`
9. `vi pg_hba.conf`

   Lines should look like this (with your IP addresses substituted). This allows the service nodes to access the DB.

   ```
   local all all ident sameuser
   # IPv4 local connections:
   host all all 127.0.0.1/32 md5
   host all all 11.16.0.1/32 md5
   host all all 11.16.1.230/32 md5
   host all all 11.16.2.230/32 md5
   ```

   where 11.16.0.1 is the MN and 11.16.1.230 and 11.16.2.230 are service nodes.

10. `vi postgresql.conf`
    set listen_addresses to '*':

    ```
    listen_addresses = '*'   # This allows remote access.
    ```

    Note: Be sure to uncomment the line

11. `service postgresql restart`
12. `chkconfig postgresql on`

13. Backup your data to migrate to the new database. (This is required even if you have not added anything to your xCAT database yet. Required default entries were created when the xCAT RPMs were installed on the management node which, and they must be migrated to the new postgresql database.)

```
mkdir -p ~/xcat-dbback
dumpxCATdb -p ~/xcat-dbback
```

14. /etc/xcat/cfgloc should contain the following line, substituting your specific info. This points the xCAT database access code to the new database.

```
Pg:dbname=xcatdb;host=11.16.0.1|xcatadmin|cluster
```

15. Copy /etc/xcat/cfgloc to /install/postscripts/etc/xcat/cfgloc for installation on the service nodes.

```
mkdir -p /install/postscripts/etc/xcat
cp /etc/xcat/cfgloc /install/postscripts/etc/xcat/cfgloc
chmod 700 /etc/xcat/cfgloc
```

16. Restore your database to postgresql (bypass mode runs the command without xcatd):

```
XCATBYPASS=1 restorexCATdb -p ~/xcat-dbback
```

17. Start the xcatd daemon using the postgresql database

```
service xcatd restart
```

18. Run this command to get the correct management node name known by ssl:

```
openssl x509 -text -in /etc/xcat/cert/server-cert.pem -noout|grep Subject:
```

this will display something like:

```
Subject: CN=mgt.cluster
```

19. Update the policy table with mgt.cluster output from the command:

```
chtab priority=5 policy.name=<mgt.cluster> policy.rule=allow
```

Note: this name must be an MN name that is known by the service nodes.

20. Make sure the site table has at least the following settings (using tabdump, tabedit, chtab):

```
#key,value,comments,disable
"xcatiport","3002",,
"xcatdport","3001",,
"master","mn20",,
```

where mn20 is the hostname of the management node as known by the service nodes.

21. Verify the policy table contains at least:

```
#priority,name,host,commands,noderange,parameters,time,rule,comments,disable
"1","root",,,,,,"allow",,
"2",,,"getbmcconfig",,,,"allow",,
"3",,,"nextdestiny",,,,"allow",,
"4",,,"getdestiny",,,,"allow",,
"5","mn20",,,,,,"allow",,
```

3.2 Define the service nodes in the database

For this example, we have two service nodes: rra000 and rrb000. (The adapters on the service nodes that the management node will use to manage them are rra000-m and rrb000-m, respectively. The bonded adapters on the service nodes that will communicate with their respective compute nodes are rra000 and rrb000, respectively.) To add the service nodes to the database run the following commands. Note: service nodes are required to be defined with group "service". Some of the commands in this document use the group "service" to update all service nodes.

Note: For table attribute descriptions, run "tabdump -d <table name>". Also, in some of the following table commands, regular expressions are used so that a single row in the table can represent many nodes. See http://xcat.sf.net/man5/xcatdb.5.html for a description of how to use regular expressions in xCAT tables, and see http://www.perl.com/doc/manual/html/pod/perlre.html for an explanation of perl regular expression syntax.

3.2.1 Add Service Nodes to the nodelist Table

```
nodeadd rra000-m,rrb000-m groups=service,ipmi,all
```

3.2.2 Set Attributes of the Service Nodes

You can use the chdef command to set many attributes of the service nodes without having to know which database tables the attributes need to go in. To do this, create a file called service.attributes with the following contents:

```
service:
 objtype=group
# nodehm attributes (for hw control)
 mgt=ipmi
 cons=ipmi
 serialport=0
 serialspeed=19200
 serialflow=hard
# ipmi attributes (the reg expression means remove "-m" and add "-bmc")
 bmc="|^(.+)-m$|($1)-bmc|"
 bmcpassword=PASSW0RD
 bmcusername=USERID
# nodetype attributes (what OS image to use for deployment)
 os=fedora8
 arch=x86_64
 profile=service
 nodetype=osi
# noderes attributes (controls how deployment is done)
```

```
 netboot=pxe
 installnic=eth0
 primarynic=eth0
# chain attributes (controls what happens when a new node is discovered)
 chain="runcmd=bmcsetup,standby"
 ondiscover=nodediscover
# servicenode attributes (what services get started/configured on the Sns)
# turn off any you don't need, just make sure your compute nodes don't refer
# to them in their noderes attributes
 setupnameserver=1
 setupdhcp=1
 setuptftp=1
 setupnfs=1
 setupconserver=1
 setupldap=1
 setupntp=1
 setupftp=1
# postscript attributes (customization scripts to run after deployment)
# configeth is a sample script to configure the 2nd ethernet NIC on the service
# node.  It should be modified to fit your specific environment.
 postscripts=configeth,servicenode,xcatserver,xcatclient
```

Then run:

```
cat service.attributes | chdef -z
```

You can also provide attribute values directly as command line arguments to chdef, if you are only changing a few. To list the attributes of the service group, run:

```
lsdef -t group -l service
```

To add your own postscripts to further customize the service nodes, see 14 Appendix: Customizing Your Nodes by Creating Your Own Postscripts.

3.2.3 Configure the Service Node BMCs and Discover MACs

1. Set up the switch table so xCAT knows what nodename to associate with each port on the switch. For example:
   ```
   chtab node=rra000-m switch.switch=11.16.255.254 switch.port=1/0/26
   ```
2. (The chain table for service node discovery was already defined above.)
3. Manually power up the service nodes. All nodes will be network booted (you can watch /var/log/messages for DHCP and TFTP traffic). Within a few seconds of booting to the network, any BMCs should be configured and be set up to allow ssh.
4. Verify the results. This command should show interesting data after discovery:
   ```
   nodels <noderange> vpd.serial vpd.mtm mac.mac
   ```

3.2.4 Set Necessary Attributes in site Table

```
chtab key=defserialport site.value=0
chtab key=defserialspeed site.value=19200
```

If you are **not** using the NFS-hybrid method of stateless booting you compute nodes, set the installloc attribute to "/install". This instructs the service node to mount /install from the management node. (If you don't do this, you have to manually sync /install between the management node and the service nodes.)

```
chtab key=installloc site.value=/install
```

4.0 Set Up Services on the Management Node

4.1 Set Up networks Table

All networks in the cluster must be defined in the networks table. When xCAT was installed, it ran makenetworks, which created an entry in this table for each of the networks the management node is on. We will update the entry for the network the management node uses to communicate to the service nodes, and create one for each CU.

```
chtab net=11.16.0.0 networks.netname=mvnet networks.mask=255.255.0.0
   networks.mgtifname=eth4 networks.gateway=9.114.88.190
   networks.dhcpserver=11.16.0.1 networks.tftpserver=11.16.0.1
   networks.nameservers=11.16.0.1  networks.dynamicrange=11.16.1.210-11.16.1.250
chtab net=11.17.0.0 networks.netname=cuanet networks.mask=255.255.0.0
   networks.mgtifname=eth1 networks.gateway=11.17.255.254
   networks.dhcpserver=11.17.0.1 networks.tftpserver=11.17.0.1
   networks.nameservers=11.17.0.1  networks.dynamicrange=11.17.1.200-11.17.1.250
chtab net=11.18.0.0 networks.netname=cubnet networks.mask=255.255.0.0
   networks.mgtifname=eth1 networks.gateway=11.18.255.254
   networks.dhcpserver=11.18.0.1 networks.tftpserver=11.18.0.1
   networks.nameservers=11.18.0.1  networks.dynamicrange=11.18.1.200-11.18.1.250
```

Disable the entry for the public network (connected to the outside world):

```
chtab net=9.114.88.160 networks.netname=public networks.disable=1
```

Set domain in the site table:

```
chtab key=domain site.value=cluster.net   # domain part of the node hostnames
```

Add the relevant networks to DHCP:

```
makedhcp -n         # will automatically restart dhcpd
```

If you defined service nodes, add them to DHCP:

```
makedhcp -a
```

4.2 Set Up DHCP

The dynamic ranges for the networks were set up already in section 4.1 Set Up networks Table. Now you should define the dhcp interfaces in site table if you want to limit which NICs dhcpd will listen on. We use this weird value because our MN uses eth4 to communicate with the service nodes, and the service nodes use eth1 to communicate with the compute nodes.

```
chtab key=dhcpinterfaces site.value='mn20|eth4;service|eth1'
```

4.3 Set Up NTP

To enable the NTP services on the cluster, first configure NTP on the management node and start ntpd.

Next set the ntpservers attribute in the site table. Whatever time servers are listed in this attribute will be used by all the nodes that boot directly from the management node (i.e. service nodes and compute nodes not being managed by a service node).

If your nodes have access to the internet you can use the global servers:

```
chtab key=ntpservers site.value=0.north-america.pool.ntp.org,          1.north-
   america.pool.ntp.org,2.north-america.pool.ntp.org,                3.north-
   america.pool.ntp.org
```

If the nodes do not have a connection to the internet (or you just want them to get their time from the management node for another reason), you can use your Management Node as the NTP server.

```
chtab key=ntpservers site.value=mn20  # IP of mgmt node
```

To set up NTP on the nodes, add the setupntp postinstall script to the postscripts table. See section 5.10, Set Up Postscripts to be Run on the Nodes. Assuming you have a group named compute:

```
chtab node=compute postscripts.postscripts=setupntp
```

If using Service Nodes, ensure that the NTP server will be set up on the Service Nodes (see section 3.2.2, Set Attributes of the Service Nodes), and add the setupntp postscript to the service nodes:

```
chdef -t group -p service postscripts=setupntp
```

4.4 Set Up DNS

Note: The DNS setup here is done using the non-chroot DNS configuration. This requires that you first remove the bind-chroot rpm (if installed) before proceeding:

```
rpm -e bind-chroot-9.5.0-16.a6.fc8
```

Set nameserver, and forwarders in the site table:

```
chtab key=nameservers site.value=11.16.0.1 # IP of mgmt node
chtab key=forwarders site.value=9.114.8.1,9.114.8.2 # site DNS servers
```

Edit /etc/hosts to be similar to:

```
127.0.0.1       localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
192.168.2.100   b7-eth0
```

```
192.168.100.1   b7
192.168.100.10  blade1
192.168.100.11  blade2
192.168.100.12  blade3
172.30.101.133  amm3
```

Run:

```
makedns
```

Set up /etc/resolv.conf:

```
search cluster.net
nameserver 11.16.0.1
```

Start DNS:

```
service named start
chkconfig --level 345 named on
```

4.5 Define AMMs as Nodes

The nodelist table contains a node definition for each management module and switch in the cluster. We have provided a sample script to automate these definitions for the RR cluster.

/opt/xcat/share/xcat/tools/mkrrbc will allow you to automatically define as many BladeCenter management module and switch node definitions as you would like to and set up convenient node groups needed to manage them. You can first run mkrrbc with the --`test` option to verify that the nodeadd commands that will be run will create the node and node group definitions you need. See man mkrrbc.

For example, running these mkrrbc commands will create the following definitions in the nodelist table. (These node groups will be used in additional xCAT Table setup so that an entry does not have to be made for every management module or switch.)

```
/opt/xcat/share/xcat/tools/mkrrbc -C a -L 2 -R 1,4
/opt/xcat/share/xcat/tools/mkrrbc -C b -L 2 -R 1,4
```

adds to the nodelist table entries like:

```
"bca01","mm,cud,rack02",,,
"swa01","nortel,switch,cud,rack02",,,
```

After running mkrrbc, define the hardware control attributes for the management modules:

```
chtab node=mm nodehm.mgt=blade
chtab node=mm mp.mpa='|(.*)|($1)|'
```

4.6 Set Up AMMs

Note: currently the network settings on the MM (both for the MM itself and for the switch module) need to be set up with your own customized script. (Eventually, this will be done by xCAT through lsslp, finding it on the switch, looking in the switch table, and then setting it in the MM. But for now, you must do it yourself.) After setting the network settings of the MM and switch module, then:

```
rspconfig mm snmpcfg=enable sshcfg=enable
rspconfig mm pd1=redwoperf pd2=redwoperf
rpower mm reset
```

Test the ssh set up with:

```
psh -l USERID mm info -T mm[1]
```

TIP for SOL to work best telnet to nortel switch (default pw is "admin") and type:

```
/cfg/port int1/gig/auto off
```

Do this for each port (I.e. int2, int3, etc.)

4.6.1 Update the AMM Firmware, If Necessary

Updating AMM Firmware can be done through the web GUI or can be done in parallel with ssh. To do it in parallel using psh:

Download Firmware from http://www-304.ibm.com/systems/support/supportsite.wss/docdisplay?brandind=5000008&lndocid=MIGR-5073383

```
cd /tftpboot/
unzip ibm_fw_amm_bpet36k_anyos_noarch.zip
# Perform update
psh -l USERID mm "update -i 11.16.0.1 -l CNETCMUS.pkt -v -T mm[1]"
# Reset the AMM, they will take a few minutes to come back online
psh -l USERID mm "reset -T mm[1]"
```

You can display the current version of firmware with:

```
psh -l USERID mm "info -T mm[1]" | grep "Build ID"
```

4.7 Start Up TFTP

```
service tftpd restart
```

4.8 Other Services

An HTTP server is needed for node installation (diskful), and an FTP server is needed for the nodes to access the postscripts and credentials. Both of these services should be set up automatically when xCAT is installed.

5.0 Define Compute Nodes in the Database

Note: For table attribute definitions run "tabdump -d <table name>". In some of the following table commands, regular expressions are used so that a single row in the table can represent many nodes. See http://xcat.sf.net/man5/xcatdb.5.html for a description of how to use regular expressions in xCAT tables, and see http://www.perl.com/doc/manual/html/pod/perlre.html for an explanation of perl regular expressions.

5.1 Set Up the nodelist Table

The nodelist table contains a node definition for each node in the cluster. For simple clusters, nodes can be added to the nodelist table using nodeadd and a node range. For example:

```
nodeadd blade01-blade40 groups=all,blade
```

For more complicated clusters, in which you want subsets of nodes assigned to different groups, we have provided a sample script to automate these definitions.

/opt/xcat/share/xcat/tools/mkrrnodes will allow you to automatically define as many nodes as you would like to and set up node groups needed to manage those nodes. You can first run mkrrnodes with the `--test` option to verify that the nodeadd commands that will be run will create the nodes and node groups you need. See man mkrrnodes.

For example, running these mkrrnodes commands will define the following nodes with the assigned groups in the nodelist table. (These node groups will be used in additional xCAT Table setup so that an entry does not have to be made for every node.)

```
/opt/xcat/share/xcat/tools/mkrrnodes -C a -L 1 -R 1,12  ( see man mkrrnodes)
```

adds to the nodelist table entries like the following:

```
"rra001a","rra001,ls21,cua,opteron,compute,tb,all,rack01",,,
"rra001b","rra001,qs22,cua,cell,cell-b,compute,all,tb,rack01",,,
"rra001c","rra001,qs22,cua,cell,cell-c,compute,all,tb,rack01",,,
"rra002a","rra002,ls21,cua,opteron,compute,tb,all,rack01",,,
"rra002b","rra002,qs22,cua,cell,cell-b,compute,all,tb,rack01",,,
"rra002c","rra002,qs22,cua,cell,cell-c,compute,all,tb,rack01",,,
```

5.2 Set Up the nodehm table

Specify that the BladeCenter management module should be used for hardware management. Also specify (via a regular expression), that the service node assigned to each blade should run the conserver daemon for that blade. (For example, rra000-m should run conserver for rra001a.)

```
chtab node=cell nodehm.cons=blade nodehm.mgt=blade nodehm.conserver='|rr(.).*|
   rr($1)000-m|' nodehm.serialspeed=19200 nodehm.serialflow=hard
   nodehm.serialport=0
chtab node=opteron nodehm.cons=blade nodehm.mgt=blade nodehm.conserver='|rr(.).*|
   rr($1)000-m|' nodehm.serialspeed=19200 nodehm.serialflow=hard
   nodehm.serialport=1
```

Note: if you are using JS blades, do not set serialspeed or serialport.

5.3 Set Up the mp Table

Specify (via regular expressions) the BladeCenter management module (mpa) that controls each blade and the slot (id) that each blade is in. (For example, the regular expression in the 1st line below would calculate for node rrd032a an mpa of bcd11 and an id of 5.)

```
chtab node=opteron mp.mpa="|rr(.)(\d+)\D|bc(\$1)(sprintf('%02d',((\$2-1)/3+1)))|"
   mp.id='|rr.(\d+)\D|((($1-1)%3)*4+1)|'
chtab node=cell-b mp.mpa="|rr(.)(\d+)\D|bc(\$1)(sprintf('%02d',((\$2-1)/3+1)))|"
   mp.id='|rr.(\d+)\D|((($1-1)%3)*4+3)|'
chtab node=cell-c mp.mpa="|rr(.)(\d+)\D|bc(\$1)(sprintf('%02d',((\$2-1)/3+1)))|"
   mp.id='|rr.(\d+)\D|((($1-1)%3)*4+4)|'
```

5.4 Set Up Conserver

Now that the nodehm and mpa tables are set up, hardware management should work.

```
makeconservercf
service conserver stop
service conserver start
```

Test a few nodes with rpower and rcons.

5.5 Set Up the noderes Table

The noderes table defines where each node should boot from (xcatmaster), where commands should be sent that are meant for this node (servicenode), and the type of network booting supported (among other things).

If you <u>are</u> using Service Nodes:
The servicenode attribute should be set to the hostname of the service node that the management node knows it by (in our case rr*000-m). The xcatmaster attribute should be set to the hostname of the service node that the compute node knows it by (in our case rr*000).

```
chtab node=opteron noderes.netboot=pxe noderes.servicenode='|rr(.).*|rr($1)000-m|'
   noderes.xcatmaster='|rr(.).*|rr($1)000|' noderes.installnic=eth0
   noderes.primarynic=eth0 noderes.nfsserver='|rr(.).*|rr($1)000|'
chtab node=cell noderes.netboot=yaboot noderes.servicenode='|rr(.).*|rr($1)000-m|'
   noderes.xcatmaster='|rr(.).*|rr($1)000|' noderes.installnic=eth0
   noderes.primarynic=eth0
```

Note: for each service you refer to here, you must ensure you have that service started on that service node in section3.2.2, Set Attributes of the Service Nodes.

If you are <u>not</u> using Service Nodes:

In this case, the management node hostname (as known by the compute node) should be used for xcatmaster (servicenode will default to the MN).

```
chtab node=opteron noderes.netboot=pxe noderes.xcatmaster=mn20 nodehm.serialport=1
   noderes.installnic=eth0 noderes.primarynic=eth0 noderes.nfsserver=mn20
chtab node=cell noderes.netboot=yaboot noderes.xcatmaster=mn20
   nodehm.serialport=0 noderes.installnic=eth0 noderes.primarynic=eth0
```

5.6 Set Up nodetype Table

Define the OS version and the specific set of packages (profile) that should be used for each node. The profile refers to a pkglist and exlist in /opt/xcat/share/xcat/netboot/<os> or /opt/xcat/share/xcat/install/<os>.

```
chtab node=opteron nodetype.os=fedora8 nodetype.arch=x86_64
   nodetype.profile=compute nodetype.nodetype=osi
chtab node=cell nodetype.os=fedora8 nodetype.arch=ppc64 nodetype.profile=compute
   nodetype.nodetype=osi
```

5.7 Set Up Passwords in passwd Table

Add needed passwords to the passwd table to support installs.

```
chtab key=system passwd.username=root passwd.password=cluster
chtab key=blade passwd.username=USERID passwd.password=PASSW0RD
chtab key=ipmi passwd.username=USERID passwd.password=PASSW0RD
```

5.8 Verify the Tables

To verify that the tables are set correctly, run lsdef on a service node, opteron blade, and cell blade:

```
lsdef rra000-m,rra001a,rra001b
```

5.9 Set Up deps Table for Proper Boot Sequence of Triblades

Note: A triblade is a special hardware grouping of 1 LS21 blade and 2 QS22 blades. If you are not using triblades, skip this section.

The following is an example of how you can set up the deps table to ensure the triblades boot up in the proper sequence. The 1st row tells xCAT the opteron blades should not be powered on until the corresponding cell blades are powered on. The 2nd row tells xCAT the cell blades should not be powered off until the corresponding opteron blades are powered off.

```
chtab node=opteron deps.nodedep='|rr(.\d+)a|rr($1)b,rr($1)c|' deps.msdelay=10000
   deps.cmd=on
chtab node=cell deps.nodedep='|rr(.\d+).|rr($1)a|' deps.msdelay=10000 deps.cmd=off
```

Verify the dependencies are correct:

```
nodels rra001a deps.nodedep
nodels rra001b deps.nodedep
```

5.10 Set Up Postscripts to be Run on the Nodes

xCAT automatically adds the syslog and remoteshell postscripts to the xcatdefaults row of the table. If you want additional postscripts run that are shipped with xCAT, for example the ntp setup script:

```
chtab node=compute postscripts.postscripts=setupntp
```

To add your own postscripts to further customize the nodes, see 14 Appendix: Customizing Your Nodes by Creating Your Own Postscripts.

5.11 Get MAC Addresses for the Blades

For blades, MACs can either be collected through the boot discovery process (like used for the service nodes in section 3.2.3 Configure the Service Node BMCs and Discover MACs) or by using the getmacs command:

```
getmacs tb
```

("tb" is the group of all the blades.) To verify mac addresses in table:

```
tabdump mac
```

5.12 Add Compute Nodes to DHCP

Ensure dhcpd is running:

```
service dhcpd status
```

Configure DHCP:

```
makedhcp -a
```

6.0 Install or Stateless Boot the Service Nodes

The service node must contain not only the OS, but also the xCAT software. In addition, a number of files are added to the service node to support the postgresql database access from the service node to the Management node, and ssh access to the nodes that the service nodes services. The following sections explain how to accomplish this.

6.1 Build the Service Node Stateless Image

We recommend that you use stateless service nodes, but if you want to have diskfull, statefull service nodes instead, skip this section and follow section 6.2, Set Up the Service Nodes for Installation.

Note: this section assumes you can build the stateless image on the management node because the service nodes are the same OS and architecture as the management node. If this is not the case, you need to build the image on a machine that matches the service node's OS/architecture.

1. Check the service node packaging to see if it has all the rpms required:

   ```
   cd /opt/xcat/share/xcat/netboot/fedora/
   vi service.pkglist service.exlist
   ```

   Make sure service.pkglist has the following packages (these packages should all be there by default).

   ```
   bash
   stunnel
   dhclient
   kernel
   openssh-server
   openssh-clients
   busybox-anaconda
   vim-minimal
   rpm
   bind
   bind-utils
   ksh
   nfs-utils
   dhcp
   bzip2
   rootfiles
   vixie-cron
   wget
   vsftpd
   rsync
   ```

   Edit service.exlist and verify that nothing is excluded that you want on the service nodes.

   While you are here, edit compute.pkglist and compute.exlist, adding and removing as necessary. Ensure that the pkglist contains bind-utils so that name resolution will work during boot.

2. Run image generation:

```
rm -rf /install/netboot/fedora8/x86_64/service
cd /opt/xcat/share/xcat/netboot/fedora/
./genimage -i eth0 -n tg3,bnx2 -o fedora8 -p service
```

3. Install xCAT code into the service node image:

```
rm -f /install/netboot/fedora8/x86_64/service/rootimg/etc/yum.repos.d/*
cp -pf  /etc/yum.repos.d/*.repo
   /install/netboot/fedora8/x86_64/service/rootimg/etc/yum.repos.d
yum --installroot=/install/netboot/fedora8/x86_64/service/rootimg install
   xCATsn
```

4. Prevent DHCP from starting up until xcatd has had a chance to configure it:

```
chroot /install/netboot/fedora8/x86_64/service/rootimg chkconfig dhcpd off
chroot /install/netboot/fedora8/x86_64/service/rootimg chkconfig dhcrelay off
```

5. Edit fstab:

```
cd /install/netboot/fedora8/x86_64/service/rootimg/etc/
cp fstab fstab.ORIG
```

   Put in fstab:

```
proc             /proc           proc         rw 0 0
sysfs            /sys            sysfs        rw 0 0
devpts           /dev/pts        devpts       rw,gid=5,mode=620 0 0
service_x86_64   /               tmpfs        rw 0 1
```

6. (Because we do not set site.installloc to anything, the service nodes will NOT mount /install. This is what you want if the compute nodes are going to mount /install from the service nodes using the NFS-hybrid mode. If you are going to use RAM-root mode for the compute nodes, you can set site.installloc to "/install". This will cause the service nodes to mount /install from the management node, and then you won't have to manually sync /install to the service nodes.)

7. Export /install read-only in service node image:

```
cd /install/netboot/fedora8/x86_64/service/rootimg/etc
echo '/install *(ro,no_root_squash,sync,fsid=13)' >exports
```

8. Pack the image

```
packimage -o fedora8 -p service -a x86_64
```

9. To update the xCAT software in the image at a later time:

```
yum --installroot=/install/netboot/fedora8/x86_64/service/rootimg update '*xCAT*'
packimage -o fedora8 -p service -a x86_64
```

Note: The service nodes are set up as NFS-root servers for the compute nodes. Any time changes are made to any compute image on the mgmt node it will be necessary to sync all changes to all service nodes. After any service node reboot a sync must also be done. This is covered in chapter 11, Using NFS Hybrid for the Diskless Images.

6.2 Set Up the Service Nodes for Installation

Note: If you are using stateless service nodes, skip this section.

To prepare for installing the service nodes, you must copy the xCAT software and necessary prereqs into /install/postscripts, so it can be installed during node installation by the servicenode postscript.

```
mkdir -p /install/postscripts/xcat/RPMS/noarch
mkdir -p /install/postscripts/xcat/RPMS/x86_64
```

The following rpms should be copied to /install/postscripts/xcat/RPMS/noarch:

- perl-Expect-1.20-1.noarch.rpm
- perl-xCAT-2*-*.rpm
- xCAT-client-2*-*.rpm
- xCAT-nbkernel-x86_64-2.6.18_8-*.noarch.rpm
- xCAT-nbroot-core-x86_64-2*-*.noarch.rpm
- xCAT-nbroot-oss-x86_64-2*-*.noarch.rpm
- xCAT-server-2*-*.noarch.rpm

The following rpms should be copied to /install/postscripts/xcat/RPMS/x86_64:

- atftp-0.7-1.x86_64.rpm
- atftp-client-0.7-1.x86_64.rpm
- atftp-debuginfo-0.7-1.x86_64.rpm
- conserver-8.1.16-2.x86_64.rpm
- conserver-debuginfo-8.1.16-2.x86_64.rpm
- fping-2.4b2_to-2.x86_64.rpm
- ipmitool-1.8.9-2.x86_64.rpm
- ipmitool-debuginfo-1.8.9-2.x86_64.rpm
- perl-IO-Tty-1.07-1.x86_64.rpm
- xCATsn-2*-*.x86_64.rpm

6.3 Boot or Install the Service Nodes

To diskless boot the service nodes:

```
nodeset service netboot
```

To install the service nodes:

```
nodeset service install
```

Then:

```
rpower service boot
wcons service      # make sure DISPLAY is set to your X server/VNC or
     rcons <one-node-at-a-time>    # or do rcons for each node
tail -f /var/log/messages
```

6.4 Test Service Node installation

- ssh to the service nodes.
- Check to see that the xcat daemon xcatd is running.
- Run some database command on the service node, e.g tabdump site, or nodels, and see that the database can be accessed from the service node.
- Check that /install and /tftpboot are mounted on the service node from the Management Node.

7.0 Install the LS21 Blades

If you want to boot the LS21 blades stateless, skip this chapter. If you want to run the LS21 blades diskfull, statefull, then at this point, simply run:

```
nodeset <nodename> install
rpower <nodename> boot
rcons <nodename>
tail -f /var/log/messages
```

Now that you have installed your LS21 blades, you don't need to follow chapter 9, Build and Boot the LS21 and QS22 Stateless Images for your LS21 blades. (Although, if you have QS22 blades, you will still need to follow that chapter to diskless boot them.)

8.0 iSCSI Install a QS22 Blade

Before you can build a stateless image for a node, you need a sample node installed with the same OS and architecture. When your nodes are the same OS/architecture as your management node, then you can build the stateless image directly on your management node. If not, you must first full-disk install a node with the correct OS/architecture. In the case of QS22 blades, this is a little more challenging,

since they don't have disks. Fortunately, xCAT provides a relatively easy way to boot the blade with an iSCSI (virtual, remote) disk and install Linux into that.

Note: in these instructions, substitute your management node hostname for mn20.

NOTE: Edit kickstart file and make sure /boot has at least 200MB of space for kernel installs.

```
yum install yaboot-xcat scsi-target-utils
chtab key=iscsidir site.value=/install/iscsi
```

Pick a QS22 blade for the iSCSI install that can access the management node. Add it as a node (and its management module, if necessary). In our example, the blade is called mvqs21b and the management module of the chassis it is in is called bca2:

```
nodeadd mvqs21b groups=compute,iscsi
nodeadd bca2 groups=mm2
```

Make sure the root userid and password are in the iscsi table

```
chtab node=mvqs21b iscsi.userid=root iscsi.passwd=cluster iscsi.server=mn20
```

Other table settings:

```
chtab node=mvqs21b noderes.nfsserver=mn20 nodehm.serialport=0
   noderes.netboot=yaboot noderes.installnic=eth0 noderes.primarynic=eth0
chtab node=mvqs21b nodetype.os=fedora8 nodetype.arch=ppc64 nodetype.profile=iscsi
   nodetype.nodetype=osi iscsi.server=mn20
chtab node=mvqs21b nodehm.mgt=blade nodehm.cons=blade nodehm.serialspeed=19200
   nodehm.serialflow=hard
chtab node=bca2 nodehm.mgt=blade
chtab node=mvqs21b mp.mpa=bca2 id=2
chtab node=bca2 mp.mpa=bca2

getmacs mvqs21b
```

Put mvqs21b and bca2 in /etc/hosts, then:

```
makedns
makedhcp -n

service tgtd restart
nodech mvqs21b iscsi.file=
setupiscsidev -s8192 mvqs21b

nodeset mvqs21b install
rpower mvqs21b boot
```

If at some point you want to reinstall this blade:

```
nodech mvqs21b nodetype.profile=iscsi
nodeset mvqs21b install
rpower mvqs21b boot
```

If you want to just boot it to its already installed iSCSI disk (maybe to add a few packages):

```
nodech mvqs21b nodetype.profile=iscsi
nodeset mvqs21b iscsiboot
rpower mvqs21b boot
```

9.0 Build and Boot the LS21 and QS22 Stateless Images

You are now ready to build the stateless images and then boot nodes with them. In our example, we have 2 types of compute nodes: qs22 (ppc64) blades and ls21 (x86_64) blades. The steps for each are very similar, so we have combined them. Go through these instructions once for each type.

9.1 Build the Stateless Image

1. On the management node, check the compute node package list to see if it has all the rpms required.

   ```
   cd /opt/xcat/share/xcat/netboot/fedora/
   vi compute.pkglist compute.exlist    # for ppc64, edit compute.ppc64.pkglist
   ```

   For example to add vi to be installed on the node, add the name of the vi rpm to compute.pkglist. Make sure nothing is excluded in compute.exlist that you need. For example, if you require perl on your nodes, remove ./usr/lib/perl5 from compute.exlist . Ensure that the pkglist contains bind-utils so that name resolution will work during boot.

2. If the stateless image you are building doesn't match the OS/architecture of the management node, logon to the node you installed in the previous chapter and do the following. (If you are building your stateless image on the management node, skip this step.)

   ```
   ssh mvqs21b
   mkdir /install
   mount mn20:/install /install
   ```

   Create fedora.repo:

   ```
   cd /etc/yum.repos.d
   rm -f *.repo
   ```

   Put the following lines in /etc/yum.repos.d/fedora.repo:

   ```
   [fedora]
   name=Fedora $releasever - $basearch
   baseurl=file:///install/fedora8/ppc64
   enabled=1
   gpgcheck=0
   ```

   Test with: `yum search gcc`

Copy the executables and files needed from the Management Node:

```
mkdir /root/netboot
cd /root/netboot
scp mn20:/opt/xcat/share/xcat/netboot/fedora/genimage .
scp mn20:/opt/xcat/share/xcat/netboot/fedora/geninitrd .
scp mn20:/opt/xcat/share/xcat/netboot/fedora/compute.ppc64.pkglist .
scp mn20:/opt/xcat/share/xcat/netboot/fedora/compute.exlist .
```

3. Generate the image:

   If you are building the image on a sample, continue the steps above by running:

   ```
   ./genimage -i eth0 -n tg3 -o fedora8 -p compute
   ```

   Note: iSCSI, QS22, tg3, all slow - take a nap

   If you are building the image on the management node:

   ```
   cd /opt/xcat/share/xcat/netboot/fedora/
   ./genimage -i eth0 -n tg3,bnx2 -o fedora8 -p compute
   ```

4. On the management node, edit fstab in the image:

   ```
   export ARCH=x86_64      # set ARCH to the type of image you are building
   export ARCH=ppc64       # choose one or the other
   cd /install/netboot/fedora8/$ARCH/compute/rootimg/etc
   cp fstab fstab.ORIG
   ```

   Edit fstab. **Change:**

   ```
   devpts  /dev/pts devpts   gid=5,mode=620 0 0
   tmpfs   /dev/shm tmpfs    defaults       0 0
   proc    /proc    proc     defaults       0 0
   sysfs   /sys     sysfs    defaults       0 0
   ```

   to (replace $ARCH with the actual value):

   ```
   proc             /proc                 proc         rw 0 0
   sysfs            /sys                  sysfs        rw 0 0
   devpts           /dev/pts              devpts       rw,gid=5,mode=620 0 0
   #tmpfs           /dev/shm              tmpfs        rw 0 0
   compute_$ARCH    /                     tmpfs        rw 0 1
   none             /tmp                  tmpfs        defaults,size=10m 0 2
   none             /var/tmp              tmpfs        defaults,size=10m 0 2
   ```

5. Pack the image:

   ```
   packimage -o fedora8 -p compute -a $ARCH
   ```

9.2 Test Boot the Stateless Image

Even though we aren't done yet customizing the image, you can boot a node with the image, just for fun:

```
nodeset <nodename> netboot
rpower <nodename> boot
```

9.3 To Update QS22 Stateless Image

If you need to update the image at any point with additional packages:

1. Set $ARCH:
   ```
   export ARCH=x86_64      # or...
   export ARCH=ppc64
   export ROOTIMG=/install/netboot/fedora8/$ARCH/compute/rootimg
   ```

2. Before running genimage, yum, or rpm against the image:
   ```
   rm $ROOTIMG/var/lib/rpm/__db.00*
   ```

3. To update the image by running genimage, add packages to compute.ppc64.pkglist and rerun genimage as described in the previous section.

4. To update the image using YUM:
   ```
   rm -f /$ROOTIMG/etc/yum.repos.d/*
   cp /etc/yum.repos.d/*.repo $ROOTIMG/etc/yum.repos.d
   yum --installroot=$ROOTIMG install <rpms>
   ```

5. To update image using RPM:
   ```
   rpm --root /$ROOTIMG -Uvh <rpms>
   ```

6. Re-pack the image
   ```
   packimage -o fedora8 -p compute -a $ARCH
   ```

9.4 Build the Compressed Image

9.4.1 Build aufs on Your Sample Node

Do this on the same node you generated the image on. Note: if this is a node other than the management node, we assume you still have /install mounted from the MN, the genimage stuff in /root/netboot, etc..

```
yum install kernel-devel gcc squashfs-tools
```

```
mkdir /tmp/aufs
cd /tmp/aufs
svn co http://xcat.svn.sf.net/svnroot/xcat/xcat-dep/trunk/aufs
# if your node does not have internet acccess, do that elsewhere and copy

tar jxvf aufs-2-6-2008.tar.bz2
cd aufs
mv include/linux/aufs_type.h fs/aufs/
cd fs/aufs/
patch -p1 < ../../../aufs-standalone.patch
chmod +x build.sh
./build.sh
strip -g aufs.ko
```

9.4.2 Generate the Compressed Image

If you are building on a sample qs node:

```
cp aufs.ko /root/netboot
cd /opt/xcat/share/xcat/netboot/fedora
./geninitrd -i eth0 -n tg3,squashfs,aufs,loop -o fedora8 -p compute -l $(expr
   100 \* 1024 \* 1024)
```

If you are building on the management node:

```
cp aufs.ko /opt/xcat/share/xcat/netboot/fedora/
cd /opt/xcat/share/xcat/netboot/fedora
./geninitrd -i eth0 -n tg3,bnx2,squashfs,aufs,loop -o fedora8 -p service -l $(expr
   100 \* 1024 \* 1024)
```

Note: the order of the modules in the -n option is important.

Note: The -l is the size of the / file system in RAM

9.4.3 Optionally Use Light Weight Postscript

In extremely large clusters, the flexible postscript infrastructure that xCAT provides can increase the time it takes to boot all the nodes at once. You can optionally use a single, light weight, script that can be customized to do all your node post boot configuration. The sample provided assumes that all services come from the same service node that responded to the DHCP broadcast. To use this light weight postscript:

```
export ARCH=x86_64      # or...
export ARCH=ppc64
export ROOTIMG=/install/netboot/fedora8/$ARCH/compute/rootimg
cd $ROOTIMG
cp -r /root/.ssh ./root
cp /opt/xcat/share/xcat/netboot/add-on/stateless/stateless etc/init.d
chroot .
chkconfig xcatpostinit off
chkconfig --add stateless
```

9.4.4 Pack and Install the Compressed Image

On the Management Node:

```
yum install squashfs-tools       # if you did not do this earlier
packimage -a $ARCH -o fedora8 -p compute -m squashfs

chtab node=blade nodetype.profile=compute nodetype.os=fedora8
nodeset blade netboot
rpower blade boot
```

Note: If you have a need to unsquash the image:

```
cd  /install/netboot/fedora8/x86_64/compute
rm -f rootimg.sfs
packimage -a x86_64 -o fedora8 -p compute -m cpio
```

9.4.5 Check Memory Usage

```
# ssh <node> "echo 3 > /proc/sys/vm/drop_caches;free -m;df -h"
             total       used       free     shared    buffers     cached
Mem:          3961         99       3861          0          0         61
-/+ buffers/cache:         38       3922
Swap:            0          0          0
Filesystem            Size  Used Avail Use% Mounted on
compute_ppc64         100M  220K  100M   1% /
none                   10M     0   10M   0% /tmp
none                   10M     0   10M   0% /var/tmp
```

Max for / is 100M, but only 220K being used (down from 225M). But wheres the OS?

Look at cached. 61M compress OS image. 3.5x smaller

As files change in hidden OS they get copied to tmpfs (compute_ppc64) with a copy on write. To reclaim space reboot. The /tmp and /var/tmp is for MPI and other Torque and user related stuff. if 10M is too small you can fix it. To reclaim this space put in epilogue:

```
umount /tmp /var/tmp; mount -a
```

10.0 Building QS22 Image for 64K pages

Note: consider merging 9/10 if building kernel for 64K pages and NFS-hybrid boot.

On Management Node:

```
cd /opt/xcat/share/xcat/netboot/fedora
cp compute.exlist compute.exlist.4k
echo "./lib/modules/2.6.23.1-42.fc8/*" >>compute.exlist
```

```
cd /tmp
wget
   http://download.fedora.redhat.com/pub/fedora/linux/releases/8/Fedora/source/SRPM
   S/kernel-2.6.23.1-42.fc8.src.rpm
scp kernel-2.6.23.1-42.fc8.src.rpm mvqs21b:/tmp
nodech mvqs21b nodetype.profile=iscsi
nodeset mvqs21b iscsiboot
rpower mvqs21b boot
```

On the sample blade:

```
ssh mvqs21b
mkdir /install
mount mgmt:/install /install
yum install rpm-build redhat-rpm-config ncurses ncurses-devel kernel-devel gcc
   squashfs-tools
cd /tmp
rpm -Uivh kernel-2.6.23.1-42.fc8.src.rpm
rpmbuild -bp --target ppc64 /usr/src/redhat/SPECS/kernel.spec
cd /usr/src/redhat/BUILD/kernel-2.6.23
cp -r linux-2.6.23.ppc64 /usr/src/
cd /usr/src/kernels/$(uname -r)-$(uname -m)
find . -print | cpio -dump /usr/src/linux-2.6.23.ppc64/
cd /usr/src/linux-2.6.23.ppc64
make mrproper
cp configs/kernel-2.6.23.1-ppc64.config .config
```

STOP: Do step 10.3 if NFS-hybrid required.

```
make menuconfig

Kernel options  --->
[*] 64k page size
Platform support  --->
[ ] Sony PS3
<exit><exit><save>

Edit Makefile suffix:
EXTRAVERSION = .1-42.fc8-64k

make -j4
make modules_install
strip vmlinux
mv vmlinux /boot/vmlinuz-2.6.23.1-42.fc8-64k
cd /lib/modules/2.6.23.1-42.fc8-64k/kernel
find . -name "*.ko" -type f -exec strip -g {} \;
```

10.1 Rebuild aufs

Skip this section if using NFS-hybrid.

Rebuild aufs.so:

```
rm -rf aufs
tar jxvf aufs-2-6-2008.tar.bz2
cd aufs
mv include/linux/aufs_type.h fs/aufs/
cd fs/aufs/
patch -p1 < ../../../aufs-standalone.patch
chmod +x build.sh
./build.sh 2.6.23.1-42.fc8-64k
strip -g aufs.ko
cp aufs.ko /root
```

On sample blade:

```
cd /root
./genimage -i eth0 -n tg3 -o fedora8 -p compute -k 2.6.23.1-42.fc8-64k
```

10.2 Test unsquashed:

On sample blade:

```
cd /root
./geninitrd -i eth0 -n tg3 -o fedora8 -p compute -k 2.6.23.1-42.fc8-64k
```

On Management Node:

```
rm -f /install/netboot/fedora8/ppc64/compute/rootimg.sfs
packimage -a ppc64 -o fedora8 -p compute -m cpio
nodech mvqs21b nodetype.profile=compute nodetype.os=fedora8
gnodeset mvqs21b netboot
rpower mvqs21b boot
```

10.2.1 Check memory

```
# ssh left "echo 3 > /proc/sys/vm/drop_caches;free -m;df -h"
             total       used       free     shared    buffers     cached
Mem:          4012        495       3517          0          0        429
-/+ buffers/cache:         66       3946
Swap:            0          0          0
Filesystem            Size  Used Avail Use% Mounted on
compute_ppc64         2.0G  432M  1.6G  22% /
none                   10M     0   10M   0% /tmp
none                   10M     0   10M   0% /var/tmp
```

10.3 Test squash

On sample blade:

```
cd /root
./geninitrd -i eth0 -n tg3,squashfs,aufs,loop -o fedora8 -p compute -k
   2.6.23.1-42.fc8-64k -l $(expr 100 \* 1024 \* 1024)
```

Note: the order of the modules in the -n option is important.

On Management Node:

```
rm -f /install/netboot/fedora8/ppc64/compute/rootimg.sfs
packimage -a ppc64 -o fedora8 -p compute -m squashfs #bug, must remove sfs first
nodech left nodetype.profile=compute nodetype.os=fedora8
nodeset left netboot
rpower left boot
```

10.3.1 Check memory

```
# ssh left "echo 3 > /proc/sys/vm/drop_caches;free -m;df -h"
             total       used       free     shared    buffers     cached
Mem:          4012        127       3885          0          0         65
-/+ buffers/cache:         61       3951
Swap:            0          0          0
Filesystem            Size  Used Avail Use% Mounted on
compute_ppc64         100M  1.7M   99M   2% /
none                   10M     0   10M   0% /tmp
none                   10M     0   10M   0% /var/tmp
```

./lib/modules/* in compute.exlist: (??)

10.4 To Switch Back to 4K Pages

On sample blade:

```
cd /root
./geninitrd -i eth0 -n tg3 -o fedora8 -p compute
```

OR

```
./geninitrd -i eth0 -n tg3,squashfs,aufs,loop -o fedora8 -p compute -l $(expr
   100 \* 1024 \* 1024)
```

From Management Node:

```
rm -f /install/netboot/fedora8/ppc64/compute/rootimg.sfs
packimage -a ppc64 -o fedora8 -p compute -m cpio
```

OR

```
packimage -a ppc64 -o fedora8 -p compute -m squashfs
nodech mvqs21b nodetype.profile=compute nodetype.os=fedora8
nodeset mvqs21b netboot
rpower mvqs21b boot
```

11.0 Using NFS Hybrid for the Diskless Images

NOTE: NFS Hybrid will increase the NFS load on the management and/or service nodes. The number of NFS daemons should be increased.

1. Get stateless cpio or squashfs set up and test (see previous notes).

2. Patch kernel and build new aufs.ko:

   Get AUFS from CVS:

```
cd /tmp
mkdir aufs
cd /tmp/aufs
cvs -d:pserver:anonymous@aufs.cvs.sourceforge.net:/cvsroot/aufs login #CVS
   password is empty
cvs -z3 -d:pserver:anonymous@aufs.cvs.sourceforge.net:/cvsroot/aufs co aufs
cd /tmp/aufs/aufs
cvs update
```

   Install stuff

```
yum install rpm-build redhat-rpm-config ncurses ncurses-devel kernel-devel gcc
   squashfs-tools
```

   Kernel notes (x86_64 and ppc64):

```
cd /tmp
wget
   http://download.fedora.redhat.com/pub/fedora/linux/releases/8/Fedora/source/
   SRPMS/kernel-2.6.23.1-42.fc8.src.rpm
rpm -Uivh kernel-2.6.23.1-42.fc8.src.rpm
yum install redhat-rpm-config
rpmbuild -bp --target $(uname -m) /usr/src/redhat/SPECS/kernel.spec
cd /usr/src/redhat/BUILD/kernel-2.6.23
cp -r linux-2.6.23.$(uname -m) /usr/src/
cd /usr/src/kernels/$(uname -r)-$(uname -m)
find . -print | cpio -dump /usr/src/linux-2.6.23.$(uname -m)/
cd /usr/src/linux-2.6.23.$(uname -m)
make mrproper
cp configs/kernel-2.6.23.1-$(uname -m).config .config
patch -p0 < /tmp/aufs/aufs/patch/put_filp.patch
cd /tmp/aufs/aufs
make -f local.mk kconfig
cp -r include /usr/src/linux-2.6.23.$(uname -m)
cp -r fs/aufs /usr/src/linux-2.6.23.$(uname -m)/fs
cd /usr/src/linux-2.6.23.$(uname -m)
```

   Edit fs/Kconfig and change (at end):

```
source "fs/nls/Kconfig"
source "fs/dlm/Kconfig"
```

To:

```
source "fs/nls/Kconfig"
source "fs/dlm/Kconfig"
source "fs/aufs/Kconfig"
```

Append to: fs/Makefile

```
obj-$(CONFIG_AUFS) += aufs/
```

```
make menuconfig

File system --->
  <M> Another unionfs
  ---   These options are for 2.6.23.1-42.fc8
  [ ]   Use simplified (fake) nameidata
        Maximum number of branches (127)  --->
  [*]   Use <sysfs>/fs/aufs
  [ ]   Use inotify to detect actions on a branch
  [ ]   NFS-exportable aufs
  [ ]   Aufs as an readonly branch of another aufs
  [ ]   Delegate the internal branch access the kernel thread
  [ ]   show whiteouts
  [*]   Make squashfs branch RR (real readonly) by default
  [ ]   splice.patch for sendfile(2) and splice(2)
  [*]   put_filp.patch for NFS branch
  [ ]   lhash.patch for NFS branch
  [ ]   fsync_super-2.6.xx.patch was applied or not
  [ ]   deny_write_access.patch was applied or not
  [ ]   Special handling for FUSE-based filesystem
  [*]   Debug aufs
  [ ]   Compatibility with Unionfs (obsolete)
Exit, Exit, Save
```

Edit Makefile line 4: `EXTRAVERSION = .1-42.fc8-aufs`

```
make -j4
make modules_install
make install
cd /lib/modules/2.6.23.1-42.fc8-aufs/kernel
find . -name "*.ko" -type f -exec strip -g {} \;
```

Whew!

3. Remove old aufs.ko:

```
cd /opt/xcat/share/xcat/netboot/fedora
rm -f aufs.ko
```

4. Boot NFS:

Create ifcfg-eth0:

```
cd /install/netboot/fedora8/x86_64/compute/rootimg/etc/sysconfig/networks-
   scripts
```

Put in ifcfg-eth0:

```
ONBOOT=yes
BOOTPROTO=none
DEVICE=eth0
```

(This solves an intermittent problem where DHCP hoses IP long enough to hose NFS and then nothing works. It's also one less DHCP and it boots faster.)

Note: for Fedora 9 only, there is a bug that appears to need the following work-around: in /sbin/dhclient-script change "if [x$keep_old_ip = xyes]; then" to "if true; then". (This has been submitted as a bug: https://bugzilla.redhat.com/show_bug.cgi?id=453982 .)

Append to fstab:

```
cd /install/netboot/fedora8/x86_64/compute/rootimg/etc
```

add this line:

```
sunrpc            /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
```

```
yum --installroot=/install/netboot/fedora8/x86_64/compute/rootimg install nfs-
   utils
cd /opt/xcat/share/xcat/netboot/fedora
./geninitrd -i eth0 -n tg3,bnx2,aufs,loop,sunrpc,lockd,nfs_acl,nfs -o fedora8 -
   p compute -k 2.6.23.1-42.fc8-aufs   # (or -64k for PPC64)
packimage -a x86_64 -o fedora8 -p compute -m nfs
```

Note: Contents of /install/netboot/fedora8/x86_64/compute/rootimg **must** be available on all service and management nodes and NFS exported.

Note: the order of the modules in the -n option above is important.

```
nodeset <noderange> netboot
rpower <noderange> boot
```

10. 9 Updating images.

To update image use yum/rpm/vi/chroot from the mgmt node for x86_64 or yum/rpm /vi/chroot from the QS22 iSCSI image as if for a cpio or squashfs system.

To propagate the changes to all service nodes (if applicable) after rebooting the service nodes:

```
xdcp service -f 4 -r /usr/bin/rsync -o '-e ssh -craz' /install/netboot/*/*/compute
   /install/postscripts /install
```

To propagate the changes to all service nodes (if applicable) after changing any of the images:

```
xdcp service -f 20 -r /usr/bin/rsync -o '-e ssh -crazv --delete' /install/netboot/
   */*/compute /install/postscripts /install
```

No need to reboot compute nodes after updates.

12.0 Install Torque

12.1 Set Up Torque Server

```
cd /tmp
wget http://www.clusterresources.com/downloads/torque/torque-2.3.0.tar.gz
tar zxvf torque-2.3.0.tar.gz
cd torque-2.3.0
CFLAGS=-D__TRR ./configure \
        --prefix=/opt/torque \
        --exec-prefix=/opt/torque/x86_64 \
        --enable-docs \
        --disable-gui \
        --with-server-home=/var/spool/pbs \
        --enable-syslog \
        --with-scp \
        --disable-rpp \
        --disable-spool
make
make install
```

12.2 Configure Torque

```
cd /opt/torque/x86_64/lib
ln -s libtorque.so.2.0.0 libtorque.so.0
echo "/opt/torque/x86_64/lib" >>/etc/ld.so.conf.d/torque.conf
ldconfig
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/xpbsnodes /opt/torque/x86_64/bin/
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbsnodestat
   /opt/torque/x86_64/bin/
```

Create /etc/profile.d/torque.sh:

```
export PBS_DEFAULT=mn20
export PATH=/opt/torque/x86_64/bin:$PATH
chmod 755 /etc/profile.d/torque.sh
source /etc/profile.d/torque.sh
```

12.3 Define Nodes

```
cd /var/spool/pbs/server_priv
nodels '/rr.*a' groups | sed 's/: groups://' | sed 's/,/ /g' | sed 's/$/ np=4/'
   >nodes
```

12.4 Set Up and Start Service

```
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbs /etc/init.d/
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbs_mom /etc/init.d/
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbs_sched /etc/init.d/
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbs_server /etc/init.d/
chkconfig --del pbs
chkconfig --del pbs_mom
chkconfig --del pbs_sched
chkconfig --level 345 pbs_server on
service pbs_server start
```

12.5 Install pbstop

```
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/pbstop /opt/torque/x86_64/bin/
chmod 755 /opt/torque/x86_64/bin/pbstop
```

12.6 Install Perl Curses for pbstop

```
yum install perl-Curses
```

12.7 Create a Torque Default Queue

```
echo "create queue dque
set queue dque queue_type = Execution
set queue dque enabled = True
set queue dque started = True
set server scheduling = True
set server default_queue = dque
set server log_events = 127
set server mail_from = adm
set server query_other_jobs = True
set server resources_default.walltime = 00:01:00
set server scheduler_iteration = 60
set server node_pack = False
set server keep_completed=300" | qmgr
```

12.8 Set Up Torque Client (x86_64 only)

12.8.1 Install Torque

```
cd /opt/xcat/share/xcat/netboot/add-on/torque
./add_torque /install/netboot/fedora8/x86_64/compute/rootimg mn20 /opt/torque
   x86_64 local
```

12.8.2 Configure Torque

12.8.2.1 Set Up Access

```
cd /install/netboot/fedora8/x86_64/compute/rootimg/etc/security
echo "-:ALL EXCEPT root:ALL" >>access.conf
cp access.conf access.conf.BOOT
cd /install/netboot/fedora8/x86_64/compute/rootimg/etc/pam.d
```

Edit system-auth and replace:

```
account     sufficient    pam_ldap.so
account     required      pam_unix.so
```

with:

```
account     required      pam_access.so
account     sufficient    pam_ldap.so
account     required      pam_unix.so
```

12.8.2.2 Set Up Node to Node ssh for Root

This is needed for cleanup:

```
cp /root/.ssh/* /install/netboot/fedora8/x86_64/compute/rootimg/root/.ssh/
cd /install/netboot/fedora8/x86_64/compute/rootimg/root/.ssh/
rm known_hosts
```

Set up the config file:

```
echo "StrictHostKeyChecking no
FallBackToRsh no
BatchMode yes
ConnectionAttempts 5
UsePrivilegedPort no
Compression no
Cipher blowfish
CheckHostIP no" >config
```

12.8.3 Pack and Install image

```
packimage -o fedora8 -p compute -a x86_64
nodeset  opteron netboot
rpower  opteron boot
```

13.0 Set Up Moab

13.1 Install Moab

```
cd /tmp
wget http://www.clusterresources.com/downloads/mwm/moab-5.2.1-linux-x86_64-
   torque.tar.gz
tar zxvf /tmp/moab-5.2.1-linux-x86_64-torque.tar.gz
```

```
cd moab-5.2.1
./configure --prefix=/opt/moab
make install
```

13.2 Configure Moab

```
mkdir -p /var/spool/moab/log
mkdir -p /var/spool/moab/stats
```

Create /etc/profile.d/moab.sh:

```
export PATH=/opt/moab/bin:$PATH
```

```
chmod 755 /etc/profile.d/moab.sh
source /etc/profile.d/moab.sh
```

Edit moab.cfg and **change:**

```
RMCFG[mn20]      TYPE=NONE
```

to:

```
RMCFG[mn20]      TYPE=pbs
```

Append to moab.cfg :

```
NODEAVAILABILITYPOLICY      DEDICATED:SWAP
JOBNODEMATCHPOLICY          EXACTNODE
NODEACCESSPOLICY            SINGLEJOB
NODEMAXLOAD                 .5
JOBMAXSTARTTIME             00:05:00
DEFERTIME                   0
JOBMAXOVERRUN               0
LOGDIR                      /var/spool/moab/log
LOGFILEMAXSIZE              10000000
LOGFILEROLLDEPTH            10
STATDIR                     /var/spool/moab/stats
```

13.2.1 Start Moab

```
cp -f /opt/xcat/share/xcat/netboot/add-on/torque/moab /etc/init.d/
chkconfig --level 345 moab on
service moab start
```

14.0 Appendix: Customizing Your Nodes by Creating Your Own Postscripts

xCAT automatically runs a few postscripts that are delivered with xCAT to set up the nodes. You can also add your own postscripts to further customize the nodes. To add your own postscript, place it in

/install/postscripts on the management node. Then add it to the postscripts table for the group of nodes you want it to be run on (or the "all" group if you want it run on all nodes):

```
chtab node=mygroup postscripts.postscripts=mypostscript
```

On each node, 1st the scripts listed in the xcatdefaults row of the table will be run and then the scripts for the group that this node belongs to. If the node is being installed, the postscripts will be run after the packages are installed, but before the node is rebooted. If the node is being diskless booted, the postscripts are run near the end of the boot process. Best practice is to write the script so that it can be used in either environment.

When your postscript is executed on the node, several variables will be set in the environment, which your script can use to control its actions:

- MASTER – the management node or service node that this node is booting from
- NODE – the hostname of this node
- OSVER, ARCH, PROFILE – this node's attributes from the nodetype table
- NODESETSTATE – the argument given to nodeset for this node
- NTYPE - "service" or "compute"
- all the site table attributes

Note that some compute node profiles exclude perl to keep the image as small as possible. If this is your case, your postscripts should obviously be written in another shell language, e.g. bash.