xCAT 2.0 Cookbook

7/7/2008

Table of Contents

1.0 Introduction

xCAT 2.0 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. This example cluster is built with Fedora 8, but the same concepts apply to RHEL 5.

For a quick overview of the functions that xCAT 2.0 provides, see the summary of all the xCAT commands: http://xcat.sf.net/man1/xcat.1.html . For a summary of using the xCAT database, see http://xcat.sf.net/man5/xcatdb.5.html .

1.1 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

Before beginning, ensure that your networks are setup correctly.

2.1 Download Fedora 8 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:

```
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
```

3. Install createrepo:

```
yum install createrepo
```

2.2 Downloading and Installing xCAT 2.0

2.2.1 If Your Management Node Has Internet Access:

2.2.1.1 Download Repo Files

YUM can be pointed directly to the xCAT download site.

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

2.2.1.2 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 2.2.3, Install xCAT 2.0 software & Its Dependencies.

2.2.2 If Your Management Node Does Not Have Internet Access:

2.2.2.1 Download xCAT2.0 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://xcat.sf.net/yum/dep-rpms-snap.tar.bz2
tar jxvf core-rpms-snap.tar.bz2
tar jxvf dep-rpms-snap.tar.bz2
```

2.2.2.2 Get Fedora 8 OSS dependencies

```
cd /root/xcat2/dep-snap/rh/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
```

2.2.2.3 Setup YUM repositories for xCAT and Dependencies

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

2.2.3 Install xCAT 2.0 software & Its Dependencies

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

2.2.4 Test xCAT installation

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

2.2.5 Update xCAT 2.0 software

If you need to update the xCAT 2.0 rpms later, download the new version of http://xcat.sf.net/yum/core-rpms-snap.tar.bz2 (if the management node does not have access to the internet) and then run:

```
yum 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, Install or Stateless Boot the Service Node):

```
export BASEIMAGE=/install/netboot/fedora8/x86_64/service/rootimg
rm -f $BASEIMAGE/etc/yum.repos.d/*
cp -pf /etc/yum.repos.d/*.repo $BASEIMAGE/etc/yum.repos.d
yum --installroot=$BASEIMAGE update '*xCAT*'
packimage -o fedora8 -p service -a x86_64
```

2.2.6 Setup Yum 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 of the compute nodes. We call these additional nodes service nodes. You can have one or more service nodes setup to install groups of compute nodes.

The service nodes need to communicate with the xCAT2.0 database on the Management Node and run xCAT command to install the nodes. The service node will be installed with the xCAT code and required the PostgreSQL Database be setup instead of SQLite Default database. PostgreSQL allows a client to be setup on the service node such that the service node can access (read/write) the database on the Management Node (Master Node) from the service node.

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

3.1 Switching to PostgreSQL Database

To setup the postgresql database on the Management Node follow these steps.

This example assumes:

- 192.168.0.1: ip of master
- xcatdb: database name
- xcatadmin: database role (aka user)
- cluster: database password
- 192.168.0.10 & 192.168.0.11: service nodes

Substitute your address 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`

2. Initialize the database :
   ```
   service postgresql initdb
   ```

3. `service postgresql start`

4. `su - postgres`

5. ```
   createuser -P xcatadmin
   Enter password for new role: cluster
   Enter it again: cluster
   Shall the new role be a superuser? (y/n) n
   Shall the new role be allowed to create databases? (y/n) n
   Shall the new role be allowed to create more new roles? (y/n) n
   ```

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 192.168.0.1/32 md5
host all all 192.168.0.10/32 md5
host all all 192.168.0.11/32 md5
```

where 192.168.0.10 and 11 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 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, again substituting your info. This points the xCAT database access code to the new database.

    ```
    Pg:dbname=xcatdb;host=192.168.0.1|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:

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

17. Start the xcatd daemon using the postgresql database

    ```
    service xcatd restart
    ```

18. Run this command to get correct Master 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 the following settings (using tabdump, tabedit, chtab):

```
#key,value,comments,disable
"xcatiport","3002",,
"xcatdport","3001",,
"master","mn20",,
```
where 11.16.0.1 and mn20 are the ip address and hostname of the management node as known by the service nodes.

21. Verify the policy table contains:
```
#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 to add and update the service nodes' attributes in the site, nodelist and noderes tables. Note: service nodes are required to be defined with group “service”. The commands below are using the group “service” to update all service nodes.

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.

3.2.1 Define Service Nodes in nodelist Table

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

3.2.2 Define Service Nodes in the servicenode table

The service node table defines all service nodes and the services that will be setup on those servicenodes. To setup dns, dhcp, tftp, conserver,ntp and vsftp on the service nodes:

```
chtab node=service servicenode.nameserver=1 servicenode.dhcpserver=1
   servicenode.tftpserver=1 servicenode.conserver=1 servicenode.ntpserver=1
   servicenode.ftpserver=1
```

3.2.2.1 Define Service Nodes in noderes Table

```
chtab node=service noderes.netboot=pxe noderes.installnic=eth0
   noderes.primarynic=eth0
```

3.2.2.2 Define Service Nodes in ipmi Table

```
chtab node=service ipmi.bmc='|^(.+)$|($1)bmc|' ipmi.username=USERID
   ipmi.password=PASSW0RD
```

3.2.2.3 Define Service Nodes in switch Table

If using IPMI node discovery to collect service node MAC addresses, add the service nodes to the switch table:

```
chtab node=rra000-m switch.switch=11.16.255.254 switch.port=1/0/26
```

3.2.2.4 Define the chain Table for Service Node Discovery

If using IPMI node discovery to collect service node MAC addresses, add the discover action to the chain table:

```
chtab node=ipmi chain.chain=standby chain.ondiscover=nodediscover
```

If you first need to have the BMC for the service node set up correctly, use the following chain value instead:

```
chtab node=ipmi chain.chain="runcmd=bmcsetup,standby"
   chain.ondiscover=nodediscover
```

3.2.2.5 Define Service Nodes in nodehm Table

```
chtab node=service nodehm.cons=ipmi nodehm.mgt=ipmi  nodehm.serialspeed=19200
   nodehm.serialflow=hard nodehm.serialport=0
```

3.2.2.6 Define Service Nodes in nodetype Table

```
chtab node=service nodetype.arch=x86_64 nodetype.os=fedora8 nodetype.nodetype=osi
```

3.2.2.7 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
```

3.2.2.8 Define Service Node OS and Profile attributes

```
chtab node=service nodetype.os=fedora8 nodetype.profile=service
```

4.0 Setup Services

4.1 Setup networks Table

All networks in the cluster must be defined in the networks table. When xCAT was installed, makenetworks ran 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 for the management node 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
```

4.2 Setup 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 (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 setup 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.

```
chtab node=xcatdefaults postscripts.postscripts=syslog,remoteshell,setupntp
```

If using Service Nodes, set the servicenode table ntpservers attribute to setup the Service Node as a NTP server to the compute nodes. See section 3.2.2, Define Service Nodes in the servicenode table.

```
chtab node=service  servicenode.ntpserver=1
```

4.3 Setup 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, forwarders and domain 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
chtab key=domain site.value=cluster.net   # domain part of the node hostnames
```

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
```

Setup /etc/resolv.conf:

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

Start DNS:

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

4.4 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 definitionss as you would like to and setup convenient nodegroups 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 nodegroup definitions you need. See man mkrrbc.

For example, running these mkrrbc commands will create the following definitions in the nodelist table. (These nodegroups 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)|'
```

4.5 Setup AMM

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.

```
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 to update firmware:

Put CNETCMUS.pkt in /tftpboot

```
telnet AMM
env -T mm[1]
update -v -i TFTP_SERVER_IP -l CNETCMUS.pkt
```

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 Startup TFTP

```
mknb x86_64
service tftpd restart
```

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 Setup the nodelist Table

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

/opt/xcat/share/xcat/tools/mkrrnodes will allow you to automatically define as many nodes as you would like to and setup nodegroups 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 nodegroups 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 nodegroups 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

```
chtab node=opteron-cua nodehm.cons=blade nodehm.mgt=blade nodehm.conserver=rra000-
   m  nodehm.serialspeed=19200 nodehm.serialflow=hard nodehm.serialport=1
chtab node=cell-cua nodehm.cons=blade nodehm.mgt=blade nodehm.conserver=rra000-m
   nodehm.serialspeed=19200 nodehm.serialflow=hard nodehm.serialport=0
chtab node=opteron-cub nodehm.cons=blade nodehm.mgt=blade nodehm.conserver=rrb000-
   m  nodehm.serialspeed=19200 nodehm.serialflow=hard nodehm.serialport=1
chtab node=cell-cub nodehm.cons=blade nodehm.mgt=blade nodehm.conserver=rrb000-m
   nodehm.serialspeed=19200 nodehm.serialflow=hard nodehm.serialport=0
```

When Table.pm supports where strings on rows that contain regex's (or I think when the servicenode table is used to determine if conserver should be set up??), the above lines can be replaced with:

```
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
```

5.3 Set Up the mp Table

```
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 Setup Conserver

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

Test a few nodes with rpower and rcons.

5.5 Setup the noderes Table

The noderes table will define for the node or nodegroup, the service node used to service the node or group, the type of network booting supported, the node which is the tftpserver, dhcpserver,etc as known by the node.

If you <u>are</u> using Service Nodes:
For each node or nodegroup defined in the noderes table change the service node attribute in the noderes table to point to the name or ip address of it's service node.

```
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
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
```

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

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

5.5.1 Sample noderes table

Your noderes table will end up looking like this (if you use service nodes):

```
#node,servicenode,netboot,tftpserver,nfsserver,monserver,kernel,initrd,kcmdline,nf
   sdir,serialport,installnic,primarynic,xcatmaster,current_osimage,next_osimage,co
   mments,disable
"opteron","|rr(.).*|rr($1)000-m|","pxe","|rr(.).*|
   rr($1)000|",,,,,,,"1","eth0","eth0","|rr(.).*|rr($1)000|",,,,
"cell","|rr(.).*|rr($1)000-m|","yaboot","|rr(.).*|
   rr($1)000|",,,,,,,"1","eth0","eth0","|rr(.).*|rr($1)000|",,,,
"service",,"pxe",,,,,,,,"1","eth0","eth0",,,,,
```

5.5.2 Setting up which services run on the Service Nodes

Note: if in the noderes table you have an assigned servicenode for a node, you must make sure you have that service started on that service node by having an entry for it set to 1 or yes in the servicenode tables . To do this, run:

```
chtab node=service servicenode.nameserver=1 servicenode.dhcpserver=1
   servicenode.tftpserver=1 servicenode.nfsserver=1 servicenode.conserver=1
   servicenode.monserver=1 servicenode.ldapserver=1 servicenode.ntpserver=1
   servicenode.ftpserver=1  servicenode.comments='Starts all services on all
   service nodes'
```

5.6 Setup nodetype table

Define the OS and profile type for building the stateless image.

```
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
```

5.6.1.1 Sample nodetype table

Your nodetype table will look something like this:

```
#node,os,arch,profile,nodetype,comments,disable
"service","fedora8","x86_64","service","osi",,
"opteron","fedora8","x86_64","compute","osi",,
"cell","fedora8","ppc64","compute","osi",,
```

5.7 Setup 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 Setup deps Table for proper boot sequence

The following is an example of how you can setup 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

Add names of postscripts that should be run for all nodes by using the xcatdefaults row of the postscripts table. (xCAT automatically fills in this table with defaults during a new install of the xCAT software on the management node, so you may not have to do this step.)

```
chtab node=xcatdefaults postscripts.postscripts=syslog,remoteshell
```

To setup NTP on the nodes add the setupntp postinstall script to the list.

```
   chtab node=xcatdefaults postscripts.postscripts=syslog,remoteshell,setupntp
```

Also add postscripts that should be run on the service nodes:

```
chtab node=service
   postscripts.postscripts=configeth,servicenode,xcatserver,xcatclient
```

5.11 Get MAC addresses

```
getmacs tb
```

To verify mac addresses in table:

```
tabdump mac
```

5.12 Setup DHCP

The dynamic ranges for the networks were set up already in chapter 4.

Define dhcp interfaces in site table. 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'
```

Ensure dhcpd is running:

```
service dhcpd start
```

Configure DHCP:

```
makedhcp -n
service dhcpd restart
```

6.0 Install or Stateless Boot the Service Node

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 Configure the Service Node BMCs and Discover MACs

- Create the initrd:

```
rm /tftpboot/pxelinux.cfg/*
mknb x86_64
```

(creates the netboot image and writes out the master parameter to the /tftpboot/pxelinux.cfg/* files).
If you have conserver set up and would like to watch the discovery process in a console window, make sure the correct console information is set in /tftpboot/pxelinux.cfg/*:

```
APPEND initrd=xcat/nbfs.x86_64.gz quiet xcatd=11.16.0.2:3001
 console=ttyS0,19200 console=ttyS1
```

- Make sure your boot sequence is set to boot from network before harddrive:

```
rbootseq <noderange> list
```

If not, change it:

```
rbootseq <noderange> f,c,n,h
```

- Power up the service nodes:

```
rpower <noderange> on.
```

- Within a few seconds of booting to the network, any BMCs should be configured and be setup to allow ssh. All nodes will be network booted (you can watch /var/log/messages for DHCP and TFTP traffic, and run rcons if configured).
- Verify the results:

```
nodels <noderange>  vpd.serial vpd.mtm mac.mac
```

should show interesting data after discovery.

6.2 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 6.3, 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.

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 setup 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.

6.3 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-2.0-*.rpm
- xCAT-client-2.0-*.rpm
- xCAT-nbkernel-x86_64-2.6.18_8-*.noarch.rpm
- xCAT-nbroot-core-x86_64-2.0-*.noarch.rpm
- xCAT-nbroot-oss-x86_64-2.0-*.noarch.rpm
- xCAT-server-2.0-*.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-2.0-*.x86_64.rpm

6.4 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
```

6.5 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

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 if 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. Make sure you have latest xCAT installed (later than Thu Apr 24 17:34:48 UTC 2008)

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

3. 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!

4. Remove old aufs.ko:

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

5. 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.)

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
```

Notice helpful message:

```
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 Installing OpenLDAP

12.1 Setup LDAP Master Server

On the management node:

1. export /home (rw) for testing
   ```
   echo '/home *(rw,no_root_squash,sync)' >> /etc/exports
   exportfs -a
   ```

2. add a test userid “ibm”
   ```
   useradd ibm
   mkdir ~ibm/.ssh
   mkdir ~ibm/.pbs_spool
   ```
3. Assign a password
   ```
   passwd ibm
   ```
4. Generate root ssh keys for mn20 and give ibm id root ssh authority

   ```
   ssh-keygen -t rsa -q -N "" -f ~ibm/.ssh/id_rsa
   cp ~ibm/.ssh/id_rsa.pub ~ibm/.ssh/authorized_keys

   vi  ~ibm/.ssh/config
   ```

   Add the following lines:

   ```
   ForwardX11 yes
   StrictHostKeyChecking no
   FallBackToRsh no
   ```

```
BatchMode yes
ConnectionAttempts 5
UsePrivilegedPort no
Compression no
Cipher blowfish
UserKnownHostsFile /dev/null
CheckHostIP no
```

5. Set permissions :

```
chown -R ibm.ibm ~ibm
chmod 700 ~ibm/.ssh
chmod 600 ~ibm/.ssh/*
```

12.1.1 Install the LDAP rpms

```
yum install openldap-servers
```

or download from:

http://download.fedora.redhat.com/pub/fedora/linux/releases/8/Everything/x86_64/os/Packages/

The following rpms should be installed:

```
openldap-*
openldap-devel-*
openldap-clients-*
openldap-servers-*
```

12.1.2 Configure LDAP

```
cd /etc/openldap
```

6. edit slapd.conf

Change the following the two lines that start with suffix and rootdn:

database bdb

Update the following information in the file:

```
suffix          "dc=cluster,dc=net"

#root access
rootdn          "cn=root,dc=cluster,dc=net"
```

```
#passwd generated with: slappasswd  cluster  - you should generate your own
   password and use  in the ldap commands below in place of cluster.

rootpw       {SSHA}ChnyPaCrc2uUZ0O6iRHy3zk4rEH7HNpL



# password hash algorithm
password-hash {SSHA}

# The userPassword by default can be changed by the entry owning it if they
# are authenticated. Others should not be able to see it, except the admin.
access to attrs=userPassword
        by dn="uid=admin,ou=People,dc=cluster,dc=net" write
        by anonymous auth
        by self write
        by * none

#
##password aging
access to attrs=shadowLastChange
        by dn="uid=admin,ou=People,dc=cluster,dc=net" write
        by self write
        by * read
# for debug add the following.  Make sure the syslog.conf is configured to
# capture local4 for ldap errors.  For example  local4.*  /var/log/ldap.log
# and touch ldap.log and system syslog restart.  Uncomment when needed.
#
#loglevel -1
```

7. Check the syntax of your slapd.conf file

```
slaptest -v -f /etc/openldap/slapd.conf
config file testing succeeded
```

8. Make ldap userid id the owner and group for the /var/lib/ldap directory
```
cd /var/lib
chown ldap.ldap ldap
```

9. Copy config file:
```
cp /etc/openldap/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
```

10. start ldap to make sure “OK”
```
service ldap start
```

12.1.3 Migrate Users

```
cd /usr/share/openldap/migration
cp migrate_common.ph migrate_common.ph.save
```

Edit migrate_common.ph and change the following lines to be:

```
vi migrate_common.ph
$DEFAULT_MAIL_DOMAIN = "cluster.net";
$DEFAULT_BASE = "dc=cluster,dc=net";
$EXTENDED_SCHEMA = 1;
```

Run:

```
./migrate_base.pl >/tmp/base.ldif
./migrate_passwd.pl /etc/passwd >>/tmp/base.ldif
./migrate_group.pl /etc/group >>/tmp/base.ldif
cd /var/lib/ldap
service ldap stop
slapadd -l /tmp/base.ldif
chown ldap.ldap *
service ldap start
```

Test the database by searching for a the user ibm:

```
ldapsearch -x -v -D  "cn=root,dc=cluster,dc=net" -w cluster  -b
   "ou=People,dc=cluster,dc=net" "uid=ibm"
```

Output should be as follows:

```
ldap_initialize( <DEFAULT> )
filter: uid=ibm
requesting: All userApplication attributes
# extended LDIF
#
# LDAPv3
# base <ou=People,dc=cluster,dc=net> with scope subtree
# filter: uid=ibm
# requesting: ALL
#

# ibm, People, cluster.net
dn: uid=ibm,ou=People,dc=cluster,dc=net
uid: ibm
cn: ibm
sn: ibm
mail: ibm@cluster.net
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword:: e2NyeXB0fSEh
shadowLastChange: 13998
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 501
gidNumber: 501
homeDirectory: /home/ibm
```

```
# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

12.1.4 Add a User to LDAP

Setup a new user adduser.ldif file with the following contents:

```
dn: uid=ibm4,ou=People,dc=cluster,dc=net
uid: ibm4
cn: ibm4
sn: ibm4
mail: ibm4@cluster.net
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 13998
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 504
gidNumber: 504
homeDirectory: /home/ibm4
```

Run:

```
   ldapadd -x -c -D  "cn=root,dc=cluster,dc=net" -w cluster -f adduser.ldif

adding new entry "uid=ibm4,ou=People,dc=cluster,dc=net"
```

Verify:

```
ldapsearch -x -v -D  "cn=root,dc=cluster,dc=net" -w cluster  -b
   "ou=People,dc=cluster,dc=net" "uid=ibm4"
```

1.1.1 Modify a User in LDAP

Setup a new user moduser.ldif file with the following contents:

```
dn: uid=ibm4,ou=People,dc=cluster,dc=net
uid: ibm4
cn: ibm4
sn: ibm4
mail: ibm4@cluster.net
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowLastChange: 13998
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 504
gidNumber: 504
homeDirectory: /home/ibm44 <---- modification
```

Run:

```
ldapmodify -x -c -D  "cn=root,dc=cluster,dc=net" -w cluster -f moduser.ldif

modifying entry "uid=ibm4,ou=People,dc=cluster,dc=net"
```

Verify:

```
ldapsearch -x -v -D  "cn=root,dc=cluster,dc=net" -w cluster  -b
   "ou=People,dc=cluster,dc=net" "uid=ibm4"
```

12.1.5 Set LDAP Userid Password

Root sets the password to ibm4

```
ldappasswd -x -w cluster -D 'cn=root,dc=cluster,dc=net' -s ibm4
   'uid=ibm4,ou=People,dc=cluster,dc=net'
Result: Success (0)
```

User sets the password to ibm4

```
ldappasswd -x -w ibm4 -D 'uid=ibm4,ou=People,dc=cluster,dc=net' -s ibm4
   'uid=ibm4,ou=People,dc=cluster,dc=net'
Result: Success (0)
```

12.2 Setup LDAP Client

12.2.1 Install LDAP into the Compute Node Image

```
yum --installroot=/install/netboot/fedora8/x86_64/compute/rootimg \
        install openldap-clients nss_ldap nfs-utils vi
```

12.2.2 Update the LDAP Configuration in the Image

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

Edit /etc/ldap.conf with these changes:

```
uri ldap://mn20/
base dc=cluster,dc=net
nss_base_passwd ou=People,dc=cluster,dc=net
nss_base_shadow ou=People,dc=cluster,dc=net
nss_base_group  ou=Group,dc=cluster,dc=net
tls_cacertdir /etc/openldap/cacerts
pam_password md5
ssl no
```

Edit /etc/openldap/ldap.conf with these changes:

```
URI  ldap://mn20
BASE dc=cluster,dc=net
```

Add ldap to the following three lines in /etc/nsswitch.conf

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

Edit /etc/pam.d/system-auth, change (order important!):

change:

```
account     required      pam_unix.so
```

to:

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

Add to fstab to Mount /home for testing:

```
mn20:/home /home nfs timeo=14,intr 1 2
```

12.2.3 Build the Image and Install

Add the following rpms to the image for testing. Note: the order of modules in the geninitrd command is important!

```
cd /opt/xcat/share/xcat/netboot/fedora
./geninitrd -i eth0 -n tg3,bnx2,sunrpc,lockd,nfs,nfs_acl -o fedora8 -p compute
packimage -o fedora8 -p compute -a x86_64
```

```
nodeset rra001a netboot
rpower rra001a boot
```

12.2.4 Test LDAP Client

```
ssh to rra001a
```

Run:

```
ldapsearch -x -v -D  "cn=root,dc=cluster,dc=net" -w cluster  -b
   "ou=People,dc=cluster,dc=net" "uid=ibm"
```

Check to see if you get output from the LDAP server as in section 12.1.3, Migrate Users.

Now authenticate the ibm users from LDAP by changing it's password and su to ibm.

```
passwd ibm
su – ibm
```

13.0 Setup Hierarchical LDAP

For Hierarchical LDAP the Service Node(s) will be set as a replica server(s) to the LDAP Master server on the Master Node. The implementation uses slurpd to push updates from the Master LDAP database to the replica server.

13.1 Configure the Master Server

First setup the Master Server Configuration as defined in the non-hierarchical setup. See section 12.1, Setup LDAP Master Server.

Make ldap userid id is the owner and group for the /var/lib/ldap directory

```
cd /var/lib
chown ldap.ldap ldap
```

Edit the /etc/openldap/slapd.conf file and add the additional hierarchical configuration:

```
cd /etc/openldap
vi /etc/openldap/slapd.conf
```

The following information in the slapd.conf file defines the name of the servicenode(s) that will be the replica server. You can add multiple replica lines for mutilple servicenodes, but only one replogfile for all the replica servers. Note replogfile must come first in the file. Also, replogfile must appear in the path as defined by “directory” in the slapd.conf file.

```
replogfile /var/lib/ldap/openldap-master-replog
replica host=rra000:389 suffix="dc=cluster,dc=net"
   binddn="cn=root,dc=cluster,dc=net" bindmethod=simple credentials=cluster


service ldap start
Starting slapd:                                            [  OK  ]
```

```
Starting slurpd:                                           [  OK  ]
```

13.1.1 Install LDAP into the Service Node image

yum --installroot=/install/netboot/fedora8/x86_64/service/rootimg \
```
    install openldap-clients nss_ldap nfs-utils vi openldap-devel openldap-
servers
```

13.1.1.1 Update the LDAP Configuration in Service Node image

On the master, shutdown the database :

```
service ldap stop
```

Export the database to a file:

```
slapcat -l /tmp/master.ldif
```

Copy the exported database into the service node install image:

```
export SNIMAGE=/install/netboot/fedora8/x86_64/service/rootimg
cp /tmp/master.ldif  $SNIMAGE/tmp/master.ldif
```

Edit the $SNIMAGE/etc/openldap/slapd.conf file:

```
database        bdb
suffix          "dc=cluster,dc=net"
rootdn          "cn=root,dc=cluster,dc=net"
updatedn        "cn=root,dc=cluster,dc=net"
updateref       ldap://mn20
```

Make ldap userid id the owner and group for the /var/lib/ldap directory

```
cd $SNIMAGE/var/lib
chown ldap.ldap ldap
```

Copy the sample config file in the image:

```
cp $SNIMAGE/etc/openldap/DB_CONFIG.example $SNIMAGE/var/lib/ldap/DB_CONFIG
```

13.1.2 Build the Service Node image and install

```
cd /opt/xcat/share/xcat/netboot/fedora
./geninitrd -i eth0 -n tg3,bnx2,sunrpc,lockd,nfs,nfs_acl -o fedora8 -p service

nodeset rra000 netboot
rpower rra000 boot
```

13.1.3 Import the Master Database on the Service Node

On the master, shutdown the database :

```
service ldap stop
```

On the Service Node import the database and start ldap:

```
cd /var/lib/ldap
slapadd -c -l /tmp/master.ldif
chown ldap.ldap *
service ldap start
```

On the master, start the database:

```
service ldap start
```

13.1.4 Test the Replica Server

Add a userid (e.g. Ibm6) to the database on the Master Node:
See section 12.1.4, Add a User to LDAP.

Go to the Service Node and search the database for the new user id.
Run:

```
ldapsearch -x -v -D  "cn=root,dc=cluster,dc=net" -w cluster  -b
   "ou=People,dc=cluster,dc=net" "uid=ibm6"
```

13.1.5 Setup LDAP client on the compute nodes

To setup LDAP on the compute nodes, install the same rpms as indicated in 12.2, Setup LDAP Client for the non-hierarchical LDAP support, but you will not have to manually setup the configuration files in the image. To setup the configuration files for the compute nodes, you can Optionally Use Light

Weight Postscript, to setup the LDAP client. For diskfull installs add the setupLDAP postscript to the default section of the postscripts table, see Set Up Postscripts to be Run on the Nodes.

14.0 Install Torque

14.1 Setup 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
```

14.2 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
```

14.3 Define Nodes

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

14.4 Setup 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
```

14.5 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
```

14.6 Install Perl Curses for pbstop

```
yum install perl-Curses
```

14.7 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
```

14.8 Setup Torque Client (x86_64 only)

14.8.1 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
```

14.8.2 Configure Torque

14.8.2.1 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
```

14.8.2.2 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
```

Setup the config file:

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

14.8.3 Pack and Install image

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

15.0 Setup Moab

15.1 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
```

15.2 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
```

15.2.1 Start Moab

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

16.0 References

- xCAT man pages: http://xcat.sf.net/man1/xcat.1.html
- xCAT DB table descriptions: http://xcat.sf.net/man5/xcatdb.5.html
- 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
- xCAT wiki: http://xcat.wiki.sourceforge.net/