Unify Context › Apply Policy › Act Intelligently

Installation Notes for Moab and TORQUE on the Cray XT

Moab Unified Intelligent Automation

Installation Notes for Moab and Torque on the Cray XT

This document provides information on the steps to install Moab and Torque on a Cray XT system.

Overview

Moab and Torque can be used to manage the batch system for a Cray XT4, XT5 or later supercomputers. This document describes how Moab can be configured to use Torque and Moab's native resource manager interface to bring Moab's unmatched scheduling capabilities to the Cray XT4/XT5.

Note: For clarity this document assumes that your SDB node is mounting a persistent /var filesystem from the bootnode. If you have chosen not to use persistent /var filesystems please be aware that the instructions below would have to be modified for your situation.

Torque Installation Notes

Perform the following steps from the *boot node* as root:

Download the latest Torque release

Download the latest Torque release.

Example 1. Download Torque

```
# cd /rr/current/software
# wget http://www.adaptivecomputing.com/downloads/torque/torque-2.2.0.tar.gz
```

Unpack the Torque tarball in an xtopview session

Using xtopview, unpack the Torque tarball into the software directory in the shared root.

Example 2. Unpack Torque

```
# xtopview
default/:/ # cd /software
default/:/software # tar -zxvf torque-2.2.0.tar.gz
```

Configure Torque

While still in xtopview, run configure with the options set appropriately for your installation. Run ./configure —help to see a list of configure options. Adaptive Computing recommends installing the Torque binaries into /opt/torque/$version and establishing a symbolic link to it from /opt/torque/default. At a minimum, you will need to specify the hostname where the Torque server will run (--with-default-server) if it is different from the host it is being compiled on. The Torque server host will normally be the SDB node for XT installations.

Example 3. Run configure

```
default/:/software # cd torque-2.2.0
default/:/software/torque-2.2.0 # ./configure --prefix=/opt/torque/2.2.0 --with-
server-home=/var/spool/torque --with-default-server=sdb --enable-syslog --disable-
gcc-warnings --enable-maxdefault --with-modulefiles=/opt/modulefiles
```

Note: The —enable-maxdefault is a change from Torque 2.4.5 onwards. This will enforce max_default queue and server settings the same way previous versions of Torque did by default. Your site may choose not to follow this configuration setting and get the new behavior. See Job Submission for more information.

Compile and Install Torque

While still in xtopview, compile and install Torque into the shared root. You may also need to link /opt/torque/default to this installation. Exit xtopview.

Example 4. Make and Make Install

```
default/:/software/torque-2.2.0 # make
default/:/software/torque-2.2.0 # make packages
default/:/software/torque-2.2.0 # make install
default/:/software/torque-2.2.0 # ln -sf /opt/torque/2.2.0/ /opt/torque/default
default/:/software/torque-2.2.0 # exit
```

Copy your Torque server directory to your Moab server host

In this example we assume the Torque server will be running on the SDB node. Torque's home directory on the SDB will be /var/spool/torque which is mounted from the bootnode (persistent var). The SDB is usually nid00003 but you will need to confirm this by logging into the SDB and running 'cat /proc/cray_xt/nid'. Use the numeric nodeid from this command in the following example.

Example 5. On the boot node, copy the Torque home directory to the SDB node's persistant /var filesystem (as exported from the bootnode)

```
# cd /rr/current/var/spool
# cp -pr torque /snv/3/var/spool
```

Stage out MOM dirs to login nodes

Stage out the MOM dirs and client server info on all login nodes. This example assumes you are using a persistent /var filesystems mounted from /snv on the boot node. Alternatively, a ram var filesystem must be populated by a skeleton tarball on the bootnode (/rr/current/.shared/var-skel.tgz) into which these files must be added. The example below assumes that you have 3 login nodes with nids of 4, 64 and 68. Place the hostname of the SDB node in the server_name file.

Example 6. Copy out MOM dirs and client server info

```
# cd /rr/current/software/torque-2.2.0/tpackages/mom/var/spool
# for i in 4 64 68

     > do cp -pr torque /snv/$i/var/spool
     > echo nid00003 > /snv/$i/var/spool/torque/server_name
     > # Uncomment the following if userids are not resolvable from the pbs_server
host
     > # echo "QSUBSENDUID true" > /snv/$i/var/spool/torque/torque.cfg
     > done
```

Perform the following steps from the Torque server node (sdb) as root:

Setup the Torque server on the sdb node

Configure the Torque server by informing it of its hostname and running the Torque.setup script.

Example 7. Set the server name and run torque.setup

```
# hostname > /var/spool/torque/server_name
# export PATH=/opt/torque/default/sbin:/opt/torque/default/bin:$PATH
# cd /software/torque-2.2.0
# ./torque.setup root
```

Customize the server parameters

Add access and submit permission from your login nodes. You will need to enable host access by setting acl_host_enable to true and adding the nid hostnames of your login nodes to acl_hosts. In order to be able to submit from these same login nodes, you need to add them as submit_hosts and this time use their hostnames as returned from the hostname command.

Example 8. Customize server settings

Enable scheduling to allow Torque events to be sent to Moab. Note: If this is not set, Moab will automatically set it on startup.

```
# qmgr -c "set server scheduling = true"
```

Keep information about completed jobs around for a time so that Moab can detect and record their completion status. Note: If this is not set, Moab will automatically set it on startup.

```
# qmgr -c "set server keep_completed = 300"
```

Remove the default nodes setting

```
# qmgr -c "unset queue batch resources_default.nodes"
```

Set resources_available.nodes equal to the maximum number of procs that can be requested in a job.

```
# qmgr -c "set server resources_available.nodes = 1250"
```

Do this for each queue individually as well.

```
# qmgr -c "set queue batch resources_available.nodes = 1250"
```

Only allow jobs submitted from hosts specified by the acl_hosts parameter.

```
# qmgr -c "set server acl_host_enable = true"
# qmgr -c "set server acl_hosts += nid00004"
# qmgr -c "set server acl_hosts += nid00064"
# qmgr -c "set server acl_hosts += nid00068"
# qmgr -c "set server submit_hosts += login1"
# qmgr -c "set server submit_hosts += login2"
# qmgr -c "set server submit_hosts += login3"
```

Define your login nodes to Torque.

Define your login nodes to Torque. You should set np to the maximum number of concurrent jobs for your system. A value of 128 is suggested as a typical setting.

Example 9. Populate the nodes file

In this example we have defined three execution hosts in Torque. Additionally, we assigned specific properties to a couple of the nodes so that particular workload can be directed to these hosts (moms).

```
# vi /var/spool/torque/server_priv/nodes
```

```
    login1 np=128 login2 np=128 mom_himem login3 np=128 mom_netpipe
```

Install the pbs_server init.d script on the server (Optional)

Torque provides an init.d script for starting pbs_server as a service.

Example 10. Copy in init.d script

```
# cd /rr/current/software/torque-2.2.0
# cp contrib/init.d/pbs_server /etc/init.d
# chmod +x /etc/init.d/pbs_server
```

Edit the init.d file as necessary -- i.e. change PBS_DAEMON and PBS_HOME as appropriate.

```
# vi /etc/init.d/pbs_server
```

```
    PBS_DAEMON=/opt/torque/default/sbin/pbs_server PBS_HOME=/var/spool/torque
```

Uncomment the following line to retain core dump files:

ulimit -c unlimited # Uncomment this to preserve core files

Install the pbs_mom init.d script on the login nodes (Optional)

Torque provides an init.d script for starting pbs_mom as a service.

Example 11. Copy in init.d script

```
# cd /rr/current/software/torque-2.2.0
```

Edit the init.d file as necessary -- i.e. change PBS_DAEMON and PBS_HOME as appropriate, retain core files, etc.

```
# vi contrib/init.d/pbs_mom
    PBS_DAEMON=/opt/torque/default/sbin/pbs_mom PBS_HOME=/var/spool/torque
```

Uncomment the following line to retain core dump files:

ulimit -c unlimited # Uncomment this to preserve core files

Stop the Torque server

Example 12. Stop Torque

```
# /opt/torque/default/bin/qterm
```

Alternatively, if you installed the init.d script, you may run:

```
# service pbs_server stop
```

Update the Torque MOM config file on each MOM node

Edit the MOM config file so job output is copied to locally mounted directories.

Example 13. Edit the MOM config file

```
# vi var/spool/torque/mom_priv/config
    $usecp *:/home/users /home/users $usecp *:/scratch /scratch
```

Note: It may be acceptable to use a $usecp *:/ / in place of the sample above. Consult with the site.

Startup the Torque Mom Daemons

On the boot node as root:

Example 14. Start up the pbs_moms on the login nodes.

```
# pdsh -w login1,login2,login3 /opt/torque/default/sbin/pbs_mom
```

Alternatively, if you installed the init.d script, you may run:

```
# pdsh -w login1,login2,login3 /sbin/service pbs_mom start
```

Startup the Torque Server

On the Torque server host as root:

Example 15. Start pbs_server

```
# /opt/torque/default/sbin/pbs_server
```

Alternatively, if you installed the init.d script, you may run:

```
# service pbs_server start
```

Moab Install Notes

Install Torque

If Torque is not already installed on your system, follow the Torque-XT Installation Notes to install Torque on the SDB node.

Perform the following steps from the boot node as root:

Download the latest Moab release

Download the latest Moab release from Cluster Resources, Inc.

Note: The correct tarball type can be recognized by the xt4 tag in its name. The xt4 tarball will be used when it is a Cray XT4, XT5 or later.

Example 16. Download Moab

```
# cd /rr/current/software
# wget --http-user=user --http-passwd=passwd
http://www.adaptivecomputing.com/download/mwm/moab-5.4.1-linux-x86_64-torque2-
xt4.tar.gz
```

Unpack the Moab tarball

Using xtopview, unpack the Moab tarball into the software directory in the shared root.

Example 17. Unpack Moab

```
# xtopview
default/:/ # cd /software
default/:/software # tar -zxvf moab-5.4.1-linux-x86_64-torque2-xt4.tar.gz
```

Configure Moab

While still in xtopview, run configure with the options set appropriately for your installation. Run ./configure —help to see a list of configure options. Adaptive Computing recommends installing the Moab binaries into /opt/moab/$version and establishing a symbolic link to it from /opt/moab/default. Since the Moab home directory must be read-write by root, Adaptive Computing recommends you specify the homedir in a location such as /var/spool/moab.

Moab no longer installs XT4 scripts by default. Use `--with-xt4` when running ./configure to install them.

Example 18. Run configure

```
default/:/software # cd moab-5.4.1
default/:/software/moab-5.4.1 # ./configure --prefix=/opt/moab/5.4.1 --with-
homedir=/var/spool/moab --with-torque=/opt/torque/default --with-
modulefiles=/opt/modulefiles --with-xt4
```

Compile and Install Moab

While still in xtopview, install Moab into the shared root. You may also need to link /opt/moab/default to this installation.

Example 19. Make Install

```
default/:/software/moab-5.4.1 # make install
default/:/software/moab-5.4.1 # ln -sf /opt/moab/5.4.1/ /opt/moab/default
```

Install the module files (Optional)

Moab provides a module file that can be used to establish the proper Moab environment. You may also want to install these module files onto the login nodes.

Example 20. make modulefiles

```
default/:/software/moab-5.4.1 # make modulefiles
```

Install the Perl XML Modules and exit xtopview

Moab's native resource manager interface scripts require a Perl XML Module to communicate via the basil interface. The Perl XML::LibXML module should be installed. The default method is to use the perldeps make target to install a bundled version of the module into a local Moab lib directory. This module may also be downloaded and installed from Perl's CPAN directory. Exit xtopview.

Example 21. make perldeps

```
default/:/software/moab-5.4.1 # make perldeps
default/:/software/moab-5.4.1 # exit
```

Customize the Moab configuration file for your Moab server host

The moab.cfg file should be customized for your scheduling environment. We will use /rr/current/var/spool/moab as a temporary staging area before copying them out to their final destinations. See the Moab Admin Guide for more details about Moab configuration parameters.

Example 22. Edit the Moab configuration file

```
# cd /rr/current/var/spool/moab
# vi moab.cfg
    SCHEDCFG[moab] SERVER=sdb:42559 TOOLSDIR /opt/moab/default/tools
RMCFG[clustername] TYPE=NATIVE:XT4 NODECFG[DEFAULT] OS=linux ARCH=XT
NODEACCESSPOLICY SINGLEJOB JOBMIGRATEPOLICY IMMEDIATE CLIENTCFG[msub]
FLAGS=AllowUnknownResource
```

Customize the XT4 native resource manager interface configuration file

Edit the configuration file ($MOABHOMEDIR/etc/config.xt4.pl) used by the xt tools.

Example 23. Edit the XT4 configuration file

```
# cd /rr/current/var/spool/moab/etc
# vi config.xt4.pl

$ENV{PATH} = "/opt/torque/default/bin:/usr/bin:$ENV{PATH}";
$batchPattern = "^login|xt1|xt2|nid00008\b|nid00011\b"; # Non-interactive jobs run
here only
# The following two lines may also modified or uncommented to support
# interactive job launch. This allows the jobs to roam in the event
# the local MOM on the login node is down.
%loginReplaceTable = (nid00008 => login1, nid00011 => login2);
$allowInteractiveJobsToRoam = "True"
```

Copy your Moab home directory to your Moab server host

In this example we assume the Moab server will be running on the SDB node. If you are installing Moab with its server home in /var as in this example and assuming that your var filesystem is being served from your boot node under /snv, you will need to login to SDB and determine the nid with 'cat /proc/cray_xt/nid'.

Example 24. Copy out Moab home directory

```
# cd /rr/current/var/spool
# cp -pr moab /snv/3/var/spool
```

Copy the Moab configuration files to all of the login nodes

Both the Moab configuration file (moab.cfg) and the configuration file for the xt4 scripts (config.xt4.pl) must be copied out to the /var filesystem on the login nodes. The only essential parameter that must be in the moab.cfg on the login nodes is the SCHEDCFG line so the clients can find the server.

Example 25. Copy out the configuration files

```
# cd /rr/current/var/spool/moab
# for i in 4 64 68; do mkdir -p /snv/$i/var/spool/moab/etc
/snv/$i/var/spool/moab/log; cp moab.cfg /snv/$i/var/spool/moab; cp
etc/config.xt4.pl /snv/$i/var/spool/moab/etc; done
```

Install the Moab init.d script (Optional)

Moab provides an init.d script for starting Moab as a service. Using xtopview into the SDB node, copy the init script into /etc/init.d.

Example 26. Copy in init.d script to the SDB node from the shared root.

```
# xtopview -n 3
node/3:/ # cp /software/moab/moab-5.1.0/contrib/init.d/moab /etc/init.d/
```

Edit the init.d file as necessary -- i.e. retain core files, etc.

Uncomment the following line to retain core dump files

ulimit -c unlimited # Uncomment to preserve core files

```
node/3:/ # xtspec /etc/init.d/moab
node/3:/ # exit
```

Perform the following steps from the Moab server node (sdb) as root:

Set the proper environment

The MOABHOMEDIR environment variable must be set in your environment when starting Moab or using Moab commands. If you are on a system with a large number of nodes (thousands), you will need to increase your stack limit to unlimited. You will also want to adjust your path to include the Moab and Torque bin and sbin directories. The proper environment can be established by loading the appropriate Moab module, by sourcing properly edited login files, or by directly modifying your environment variables.

Example 27. Loading the Moab module

```
# module load moab
```

Example 28. Exporting the environment variables by hand (in bash)

```
# export MOABHOMEDIR=/var/spool/moab
# export
PATH=$PATH:/opt/moab/default/bin:/opt/moab/default/sbin:/opt/torque/default/bin:/opt/
```

Example 29. Setting the stack limit to unlimited

If you are running on a system with large numbers of nodes (thousands), you may need to increase the stack size user limit to unlimited. This should be set in the shell from which Moab is launched. If you start Moab via an init script, this should be set in the script, otherwise it would be recommended to put this in the appropriate shell startup file for root.

```
# ulimit -s unlimited
```

Apply an orphan cleanup policy

Occasionally, Moab can encounter an orphaned ALPS partition -- that is a partition which is no longer associated with an active job. These orphans can occur under different circumstances, such as manually created alps partitions, partitions created by a different resource manager, or as a result of jobs that have been lost to Moab's memory by a catastrophic outage. By setting the MOABPARCLEANUP environment variable, you can set Moab's policy for handling orphaned ALPS partitions. If MOABPARCLEANUP is unset, Moab will not attempt to cleanup orphaned ALPS partitions. If MOABPARCLEANUP is set to Full, Moab will aggressively clean up any orphan it encounters, whether it was the creator of the partition or not. If MOABPARCLEANUP is set to anything else (such as 1, yes, TRUE, etc.), Moab will attempt to clean up only those orphans that it knows that it had a hand in creating. This environment variable must be set in the environment when starting Moab to take effect. This can be accomplished by including it in the appropriate module, init script, or via a manual setenv or export command.

Example 30. Activate aggressive ALPS partition cleanup in the optional Moab startup script

```
# vi /etc/init.d/moab
export MOABPARCLEANUP=Full
```

Customize Moab to use alps topology ordering (Optional)

Communication performance within parallel jobs may be improved by customizing Moab to allocate nodes according to a serialized alps XYZ topology ordering. There are two main methods for doing this -- by presenting the nodes to Moab in the serialized topology order, or by prioritizing the nodes in the serialized topology order. By default, Moab will allocate nodes according to an lexicographical (alphanumeric) ordering.

Option A -- Prioritizing the nodes in serialized topology order. This approach requires that you tell Moab to allocate its nodes according to a priority function based on an explicit priority for each node, which we set based on the alps XYZ ordering. An advantage of this method is that the mdiag -n output will remain in lexicographical ordering. If the apstat -no command is not supported in your version, you may build up the priority list by hand by using the XYZ topology information in the alps database. The example will show how to do this by running a script that uses apstat -no to populate a Moab configuration include file according to the XYZ topology ordering. If your current version of alps does not support the XYZ topology ordering, you may build up the nodeprio.cfg file yourself based on XYZ topology information obtained from alps.

Example 31. Populate a Moab configuration node priority file

```
# /opt/moab/default/tools/node.prioritize.xt.pl >/var/spool/moab/nodeprio.cfg
# echo "#INCLUDE /var/spool/moab/nodeprio.cfg" >> /var/spool/moab/moab.cfg
```

Option B -- Presenting the nodes in serialized topology order. This approach requires that the nodes are reported to Moab in the alps XYZ ordering. Moab will, by default, allocate nodes in the reverse order from which they are reported. This method requires alps support for the XYZ ordering. Its implementation is simple and dynamic but will cause mdiag -n to report the nodes in the serialized topology order. The example will show how to do this by setting a configuration option in the config.xt4.pl file (which was discussed in the previous section).

Example 32. Uncomment the topologyOrdering parameter

```
# vi /var/spool/moab/etc/config.xt4.pl
```

```
    $topologyOrdering = 1;
```

Enable steering of jobs to designated execution hosts (Optional)

It is possible to direct a job to launch from an execution host having a job-specified feature. Assigning features to the MOM nodes and declaring these features to be momFeatures allows you to indicate which job features will effect the steering of a job's master task to certain moms as opposed to steering the job's parallel tasks to certain compute nodes.

Example 33. Declaring MOM features

Indicates that when a feature of mom_himem or mom_netpipe are specified for a job, this will be used to steer the job to an execution_host (mom) having this feature as opposed to scheduling the parallel tasks on compute nodes having this feature.

```
# vi /var/spool/moab/etc/config.xt4.pl
```

```
# Setting momFeatures allows you to indicate which job features will effect
# the steering of jobs to certain moms as opposed to steering to compute nodes
@momFeatures = ("mom_himem", "mom_netpipe");
```

Startup the Moab Workload Manager

Start up the Moab daemon.

Example 34. Start Moab

```
# /opt/moab/default/sbin/moab
```

Alternatively, if you installed the init.d script, you may run:

```
# service moab start
```

Torque Upgrade Notes

Quiesce the system.

It is preferable to have no running jobs during the upgrade. This can be done by closing all queues in Torque or setting a system reservation in Moab and waiting for all jobs to complete. Often, it is possible to upgrade Torque with running jobs in the system, but you may risk problems associated with Torque being down when the jobs complete and incompatibilities between the new and old file formats and job states.

Perform the following steps from the torque server node (sdb) as root:

Shutdown the Torque Mom Daemons

On the boot node as root:

Example 35. Shut down the pbs_moms on the login nodes.

```
# pdsh -w login1,login2,login3 /opt/torque/default/sbin/momctl -s
```

Alternatively, if you installed the init.d script, you may run:

```
# pdsh -w login1,login2,login3 /sbin/service pbs_mom stop
```

Stop the Torque server

Example 36. Stop Torque

```
# /opt/torque/default/bin/qterm
```

Alternatively, if you installed the init.d script, you may run:

```
# service pbs_server stop
```

Perform the following steps from the boot node as root:

Download the latest Torque release.

Download the latest Torque release from Cluster Resources, Inc.

Example 37. Download Torque

```
# cd /rr/current/software
# wget http://www.adaptivecomputing.com/downloads/torque/torque-2.2.0.tar.gz
```

Unpack the Torque tarball

Using xtopview, unpack the Torque tarball into the software directory in the shared root.

Example 38. Unpack Torque

```
# xtopview
default/:/ # cd /software
default/:/software # tar -zxvf torque-2.2.0.tar.gz
```

Configure Torque

While still in xtopview, run configure with the options set appropriately for your installation. Run ./configure —help to see a list of configure options. Adaptive Computing recommends installing the torque binaries into /opt/torque/$version and establishing a symbolic link to it from /opt/torque/default. At a minimum, you will need to specify the hostname where the torque server will run (--with-default-server) if it is different from the host it is being compiled on. The torque server host will normally be the sdb node for XT installations.

Example 39. Run configure

```
default/:/software # cd torque-2.2.0
default/:/software/torque-2.2.0 # ./configure --prefix=/opt/torque/2.2.0 --with-
server-home=/var/spool/torque --with-default-server=nid00003 --enable-syslog
```

Compile and Install Torque

While still in xtopview, compile and install torque into the shared root. You may also need to link /opt/torque/default to this installation. Exit xtopview.

Example 40. Make and Make Install

```
default/:/software/torque-2.2.0 # make
default/:/software/torque-2.2.0 # make packages
default/:/software/torque-2.2.0 # make install
default/:/software/torque-2.2.0 # rm /opt/torque/default
default/:/software/torque-2.2.0 # ln -sf /opt/torque/2.2.0/ /opt/torque/default
default/:/software/torque-2.2.0 # exit
```

Startup the Torque Mom Daemons

Note: If you have still have running jobs, you will want to start pbs_mom with the -p flag to preserve running jobs. By default, the init.d startup script will not preserve running jobs unless altered to start pbs_mom with the -p flag.

On the boot node as root:

Example 41. Start up the pbs_moms on the login nodes.

```
# pdsh -w login1,login2,login3 /opt/torque/default/sbin/pbs_mom -p
```

Startup the Torque Server

On the torque server host as root:

Example 42. Start pbs_server

```
# /opt/torque/default/sbin/pbs_server
```

Alternatively, if you installed the init.d script, you may run:

```
# service pbs_server start
```

Moab Upgrade Notes

Quiesce the system.

It is preferable to have no running jobs during the upgrade. This can be done by setting a system reservation in Moab and waiting for all jobs to complete. Often, it is possible to upgrade Moab with running jobs in the system, but you may risk problems associated with Moab being down when the jobs complete.

Shutdown the Moab Workload Manager

Shut down the Moab daemon.

Example 43. Stop Moab

```
# /opt/moab/default/sbin/mschedctl -k
```

Alternatively, if you installed the init.d script, you may run:

```
# service moab stop
```

Perform the following steps from the boot node as root:

Download the latest Moab release

Download the latest Moab release from Cluster Resources, Inc.

Note: The correct tarball type can be recognized by the xt4 tag in its name.

Example 44. Download Moab

```
# cd /rr/current/software
# wget --http-user=user --http-passwd=passwd
http://www.adaptivecomputing.com/downloads/mwm/temp/moab-5.2.2.s10021-linux-x86_64-
torque2-xt4.tar.gz
```

Unpack the Moab tarball

Using xtopview, unpack the Moab tarball into the software directory in the shared root.

Example 45. Unpack Moab

```
# xtopview
default/:/ # cd /software
default/:/software # tar -zxvf moab-5.2.2.s10021-linux-x86_64-torque2-xt4.tar.gz
```

Configure Moab

While still in xtopview, run configure with the options set appropriately for your installation. Run ./configure —help to see a list of configure options. Adaptive Computing recommends installing the Moab binaries into /opt/moab/$version and establishing a symbolic link to it from /opt/moab/default. Since the Moab home directory must be read-write by root, Adaptive Computing recommends you specify the homedir in a location such as /var/spool/moab.

Example 46. Run configure

```
default/:/software # cd moab-5.2.2.s10021
default/:/software/moab-5.2.2.s10021 # autoconf
default/:/software/moab-5.2.2.s10021 # ./configure --prefix=/opt/moab/5.2.2.s10021
--with-homedir=/var/spool/moab --with-torque
```

Compile and Install Moab

While still in xtopview, install Moab into the shared root. You may also need to link /opt/moab/default to this installation.

Example 47. Make Install

```
default/:/software/moab-5.2.2.s10021 # make install
default/:/software/moab-5.2.2.s10021 # ln -sf /opt/moab/5.2.2.s10021/
/opt/moab/default
```

Install the Perl XML Modules and exit xtopview

If you have previously installed the perl modules in the perl site directories (configure --with-perl-libs=site), you should not need to remake the perl modules. However, the default is to install the perl modules local to the Moab install directory and since it is normal practice to configure the Moab upgrade to use a new install directory (configure --prefix), it will generally be necessary to reinstall the perl modules. Exit xtopview when done with this step.

Example 48. make perldeps

```
default/:/software/moab-5.2.2.s10021 # make perldeps
default/:/software/moab-5.2.2.s10021 # exit
```

Manually merge any changes from the new XT4 native resource manager interface configuration file

If the upgrade brings in new changes to the config.xt4.pl file, you will need to edit the file and manually merge in the changes from the config.xt4.pl.dist file. One way to discover if new changes have been introduced is to diff the config.xt4.pl.dist from the old and new etc directories. This is rare, but does happen on occasion. One will generally discover quite quickly if necessary changes were not made because the xt4 scripts will usually fail if the config file has not been updated.

Example 49. Merge any updates into the XT4 configuration file

```
# diff /snv/3/var/spool/moab/etc/config.xt4.pl.dist /rr/current/software/moab-
5.2.2.s10021/etc/config.xt4.pl.dist
# vi /snv/3/var/spool/moab/etc/config.xt4.pl
```

Reload the new environment

Example 50. Swapping in the new Moab module

```
# module swap moab/5.2.2.s10021
```

Startup the Moab Workload Manager

Start up the Moab daemon.

Example 51. Start Moab

```
# /opt/moab/default/sbin/moab
```

Alternatively, if you installed the init.d script, you may run:

```
# service moab start
```