Using xCSM with CSM for High Performance Computing Software Stack Installation and Configuration

August 9, 2004

vallard@us.ibm.com
mccombsk@us.ibm.com

Introduction

Cluster Systems Management (CSM) provides a distributed system management solution that allows a system administrator to set up and maintain a cluster of nodes that run the Linux operating system. CSM simplifies cluster administration tasks by providing management from a single point-of-control (a Management Server). CSM has the ability to ease the set up of High Performance Computing (HPC) clusters. This document will give instructions on how to install, configure, and test several common open source high performance computing applications using tools available in xCAT

xCAT is an additional product provided by IBM that contains additional cluster tools that can be used with CSM. In the latest xCAT (1.2.0-pre6) there are such tools as hardware setup, HPC stack configuration, helpful lsnode aliases and many other tools that are used by IBM Linux cluster professionals. Many scripts in xCAT may eventually be merged into the official CSM product.

Torque is a portable batch scheduler(PBS)/resource manager package that includes a client, a server, and a scheduler. It was spun off of Open PBS to provide a free PBS version.

Maui is a batch scheduler that works with Torque or other batch schedulers. It is the scheduler part of the operation that runs jobs and maintains the queue with the PBS server.

MPICH and LAM are MPI libraries used for parallel programming in a cluster environment. In general LAM produces better performance on Ethernet connections, while MPICH has given better performance on Myrinet.

If you are going to install these components then it is highly recommended that you subscribe to the mailing lists as things change very quick and this document can be outdated at anytime. (and probably already has been by the time you read this)

The only way to keep up is to be on the mailing list and see what is happening in the community. See a full collection of mailing lists at the end of this paper.

*Note: Previously additional tools to CSM were provided by Enhanced Cluster Tools. These tools have been merged in to xCAT to simplify cluster management and to not have another package. We hope that this will cause less confusion.

To briefly describe our method we will first install CSM and xCAT on the Management Server. Next we will define nodes to CSM but not install them. We will then use scripts and commands to set up the HPC components on the Management Server and build node install images. Next, using CSM, we will deploy the compute nodes. Finally, we will run a few other scripts to finalize setup.

Quick and Dirty Overview of Default CSM Torque, MPICH and Maui setup:

- install CSM, definenodes
- install xCAT core
- run /opt/xcat/csm/sbin/setupxcsm
- build torque using torquemaker, build maui using mauimaker, build mpich using mpimaker
- export /home and /usr/local
- run /opt/xcat/csm/hpc/torque/setuptorqueinstall
- run csmsetup[ks|yast]
- run installnodes
- when nodes have been installed run: /opt/xcat/csm/hpc/torque/gencsmpbs
- configure root ssh between management server and all nodes of the cluster.
- just to be sure, run cfmupdatenode on the nodes that installed
- run /opt/xcat/csm/bin/addclstrusr <–n noderange> userid1 userid2 …
- set up mpich environment
- go to work!

Install CSM and xCAT on the Management Server

CSM is available here:
http://techsupport.services.ibm.com/server/cluster/fixes/csmfixhome.html

Install CSM on the management server by following the instructions in the

CSM Planning and Installation Guide.
You should define all the nodes with definenode and verify that all entries are correct.

Next install the xCAT core tarball There is a link from http://www.xcat.org to this tarball. Make sure you have at least xcat-dist-core-1.2.0-pre6.

cd /opt
tar zxvf xcat-dist-core-1.2.0-pre6

Now run /opt/xcat/csm/sbin/setupxcsm.

(Note: there currently is not a mechanism to keep xCAT data files and CSM database files in synch. Therefore, if you change something in your CSM database (i.e: add a node, remove a node, change a nodegroup) you will then need to run: /opt/xcat/csm/sbin/csm2xcat to keep files in synch.)

At this point, CSM should be installed and all nodes defined. You may want to install the nodes now, but just hold your horses and don't be so impatient, there's still some additional management server setup to do:

Build/Install Torque

Prerequisites
For SLES you will need xdevel and tcl-devel installed on the MS. For RedHat make sure you have XFree86-devel, tcl-devel, tk-devel, compilers, and libcompat rpms installed. As a good rule of thumb and as suggested in other xCAT documents, you might as well save yourself a headache and install everything on the Management Server.

You need to download Torque from
http://www.supercluster.org/downloads/torque/

Copy the rpm into the /tmp directory, then run:

 /opt/xcat/build/torque/torquemaker /tmp/<torque tarball name> [rcp | scp]
example:

This will build torque with the linux binaries. If it doesn't work, you probably didn't run setupxcsm

Note with torquemaker:

- The server home is in /var/spool/pbs. The default PBS installation uses /usr/spool, however, since logs fill up quick and ruin machine functionality, it is best to keep these all in /var. In addition, all commands are put in /usr/local/pbs so as to distinguish pbs commands from other commands from maui and mpich.
- we enable syslog so that we can see errors from pbs in /var/log/messages
- we build pbs with scp in the conf.cmd. While some people prefer rcp, there have been known scaling problems with rsh. In general, if it is not a significant amount of nodes and security is not a concern, rsh will probably give you faster performance. (not sure if that is true or not, but people think it is)
- we don't enable the GUI (with torquemaker)
- If you don't like this, then copy the conf command in /opt/xcat/csm/hpc/torque into the build directory. Modify it and run it.

If everything was successful then you should have files in /usr/local/pbs/$ARCH/bin and /usr/local/pbs/$ARCH/sbin.

Build/Install Maui

Download Maui from:
http://66.237.84.51/cgi-bin/login.cgi

You have to log in to get it.

Copy the maui tarball it into the /tmp directory and run:
/opt/xcat/build/maui/mauimaker /tmp/<tarball>

There are several assumptions:

- The compiler is gcc.
- All Maui components will be installed in /usr/local/maui
- You are using pbs that is based in /usr/local/pbs
- In the post configuration after the install we move the logs to /var/spool/maui to keep in sync with pbs.

If any of those assumptions are incorrect, you need to build by hand, or modify the scripts.

Build/Install Lam-mpi

To be able to pass compute messages across your cluster you will either need mpich or lam-mpi, you shouldn't need both unless you want to test for performance. In general for clusters where the interconnect is Ethernet connections you should use lam/mpi.

Download lam to the /tmp directory
http://www.lam-mpi.org/7.0/download.php

Next copy lammaker to the /tmp file as well and build it. lammaker was created by Matt Bonsack and is more flexible then the other programs. Read the source to see what it can do. It also makes several assumptions and does some configuring:

- builds with gnu or pgi compilers
- builds in /usr/local/…
- puts lam in the path in /usr/local/{LAM_HOME}/etc/conf.[csh|sh]

```
# ./opt/xcat/build/lam/lammaker lam-7.0.5.tar.gz gnu ssh
See lam-7.0.5/make.log.
./lammaker: lam-7.0.5.tar.gz gnu ssh build successful
```

Watch the build:

```
# cd lam-7.0.5
# tail -f configure.log
# tail -f make.log
```

Build/Install mpich

The most commonly used implementations of MPI are mpich and mpich-gm. For the purpose of this document we will discuss mpich. If you are using the GM protocol over a Myrinet network, you will want to use mpich-gm.

Download mpich to /tmp
http://www-unix.mcs.anl.gov/mpi/mpich/
(version 1.2.5.2 is about 12.4 MB)

The download only says mpich.tar.gz. Move this to mpich-<version>.tar.gz so mpimaker can use it.

```
# cp mpich.tar.gz /opt/xcat/build/mpi/mpich-1.2.5.2.tar.gz
```

Run:

#./opt/xcat/build/mpi/mpimaker 1.2.5.2 smp gnu ssh

mpimaker: 1.2.5.2 smp gnu ssh build start
mpimaker: 1.2.5.2 smp gnu ssh make
mpimaker: 1.2.5.2 smp gnu ssh build successful

MPICH installed in /usr/local/mpich/1.2.5.2/ip/smp/gnu/ssh

Server and Compute Node Setup

Now that you've got the nodes ready to install and mpich, torque, and maui installed on the management server you're finally ready to install the nodes.
First, make sure you export:
/usr/local
and
/home.

Now, make sure that cfmroot is configured so that /etc/hosts, /etc/passwd/, /etc/group, and /etc/shadow are distributed to the nodes.
Since CSM doesn't set up ssh across the whole cluster, I recommend you do the following:
- ssh localhost (this will add the ms key into its own known_hosts file)
- link /root/.ssh to /cfmroot/root/.ssh so that the nodes will be able to ssh to each other with out a password.

Now run:
/opt/xcat/csm/hpc/torque/setuptorqueinstall

This will setup a post install script so that the nodes will be all set to do PBS when they boot up.

Now, the moment you've been waiting for: Install the nodes!
csmsetup[yast | ks] –n noderange
installnode –n noderange

(watch the progress by opening rconsole's to them as well as running:
tail –f /var/log/messages (in one window
watch monitorinstall –rn noderange (in another window)

Once they are installed and up and ready to go, run cfmupdatenode just to make sure.
Then, try ssh'ing to some of them to make sure it doesn't prompt you for a password.
Then run:

/opt/xcat/csm/hpc/torque/gencsmpbs <noderange>

This will setup PBS across the cluster. Log out and log back in (to use the updated path) and try running showq to see that the nodes are being seen.

Add a user by running:
/opt/xcat/csm/bin/addclstrusr –n noderange userid

Now your user is added you can do some benchmarks, etc. Have a lot of fun.

.