Scheduler Installation and Configuration

In this workshop we'll be utilizing a job scheduler to control resources on the nodes such that many jobs can run at once without impacting one another. The choice for the workshop is to use the open source version of PBS Professional (PBS Pro). A basic PBS Pro configuration is fairly simple to set up.

The scheduler packages will be installed on all nodes, however only the head node will actually facilitate the scheduling of jobs. We'll utilize the Puppet configuration management system to install the RPM for the scheduler on all nodes, then distribute an appropriate configuration file.

Pull in the latest changes to the Git repo:

```
$ cd ~/build-a-cluster && git pull
```

1. Install the RPM build utilities on master node if not already done

```
$ sudo yum install rpmdevtools rpm-build createrepo

$ rpmdev-setuptree
```

2. Install PBS Dependencies on master node (Done as part of the "scheduler" Puppet module on compute nodes)

```
$ sudo yum install -y gcc make libtool hwloc-devel \

    libX11-devel libXt-devel libedit-devel libical-devel \

    ncurses-devel perl postgresql-devel python-devel tcl-devel \

    tk-devel swig libXext-devel expat-devel openssl-devel \

    libXft-devel fontconfig autoconf automake vim-enhanced expat libedit \

    postgresql-server python sendmail sudo tcl tk
```

3. Download the PBS Pro scheduler

```
$ cd ~

$ git clone https://github.com/PBSPro/pbspro.git

$ cd pbspro

$ git checkout afdc22994999b948c29b7e4758d68233ceeabb79

$ git reset --hard; git clean -f -d -x -f
```

4. Rebuild the RPM since PBS Pro only worked with the first version of CentOS 7.

```
$ bash ./autogen.sh

$ ./configure --prefix=/opt/pbs --libexecdir=/opt/pbs/libexec

$ make -j4 dist

$ cp -fv pbspro-*.tar.gz ~/rpmbuild/SOURCES/

$ rpmbuild -bb pbspro.spec
```

5. Place the newly created RPM in the yum repo that was created in the head node setup

```
$ sudo cp ~/rpmbuild/RPMS/x86_64/pbspro-* /var/www/html/hpc-repo/Packages
```

6. Build the yum repo

```
$ cd /var/www/html/hpc-repo/

$ sudo createrepo $(pwd)
```

7. Install the ‘pbspro-server’ and associated dependencies on the head node

```
$ sudo yum makecache fast

$ sudo yum install pbspro-server

$ sudo systemctl start pbs
```

8. Install the ‘pbspro-execution’ package on the compute nodes (can be done via Puppet)

```
$ sudo clush -w @compute yum makecache fast

$ sudo vim /etc/puppet/modules/scheduler/manifests/install.pp

$ sudo vim /etc/puppet/modules/scheduler/manifests/config.pp

$ sudo vim /etc/puppet/modules/scheduler/manifests/service.pp

$ sudo clush -w @compute puppet agent -t

$ # POSSIBLE DEBUGGING HERE!!

$ sudo clush -w @compute rm -rf /var/spool/pbs/

$ sudo clush -w @compute /opt/pbs/libexec/pbs_postinstall

$ sudo setfacl -m u:puppet:r /etc/shadow

$ sudo clush -w @compute puppet agent -t

$ sudo clush -w @compute systemctl is-active pbs
```

9. Configure the PBS server with the compute nodes

```
$ for i in $(cluset -e @compute); do sudo /opt/pbs/bin/qmgr -c "c n $i"; done
```

10. Check that PBS is active on compute nodes and get the environment configured to use PBS.

```
$ sudo clush -bw @compute systemctl is-active pbs

$ . /etc/profile.d/pbs.sh
```

11. Verify nodes are up and seen by the PBS server.

```
$ /opt/pbs/bin/pbsnodes -a | grep state | sort | uniq -c
```

12. Launch basic job using an interactive qsub session.

```
$ /opt/pbs/bin/qsub -I

$ # If successful, release job.

$ exit
```

13. Build OpenMPI against PBS (See ~/build-a-cluster/scripts/openmpi_build.sh). Also play with `screen` and describe the use of a multiplexer.

```
$ cd ~/build-a-cluster && git pull && cd ~/sw

$ wget https://www.open-mpi.org//software/ompi/v2.1/downloads/openmpi-2.1.0.tar.bz2

$ tar xvjf openmpi-2.1.0.tar.bz2

$ cd openmpi-2.1.0

$ LD_LIBRARY_PATH=/opt/pbs/lib:$LD_LIBRARY_PATH LDFLAGS="-L/opt/pbs/lib -lpbs -lpthread -lcrypto" ./configure --with-tm=/opt/pbs --prefix=/apps/gcc/4.8/openmpi/2.1.0

$ make

$ sudo make install

$ sudo mkdir -p /apps/opt/lmod/mf/gcc/4.8/openmpi/

$ sudo cp ~/build-a-cluster/apps/opt/lmod/mf/gcc/4.8/openmpi/2.1.0.lua /apps/opt/lmod/mf/gcc/4.8/openmpi/

$ module avail

$ module load openmpi

$ which mpicc
```

14. Tell PBS about shared filesystems [Optional -- only important at scale]

    a. Open /var/spool/pbs/mom_priv/config from one of your compute nodes in a text editor

```
$ cd ~

$ clush -bw node02 --rcopy /var/spool/pbs/mom_priv/config --dest ./
```

b. ...and add the following content to config.node02:

```
$usecp *:/dhome/ /dhome/

$usecp *:/apps/ /apps/
```

c. Copy the edited mom_config to the rest of the compute nodes

```
$ sudo clush -bw @compute --copy config.node02 --dest
/var/spool/pbs/mom_priv/config
```

d. Restart MoM

```
$ sudo clush -bw @compute systemctl restart pbs
```

15. Run a test job

```
$ sudo su - lcistudent

$ git clone https://github.com/WyoARCC/build-a-cluster.git

$ cd build-a-cluster/code/03_intro_to_mpi

$ module load openmpi

$ mpicc mpi_pi.c

$ qsub -I -l select=4:ncpus=1

$ cd $PBS_O_WORKDIR

$ module load openmpi

$ mpirun hostname

$ mpirun ./a.out 1000000
```