

Getting going using the RHIT cluster

March 20, 2015

- ▶ We have a 32 node machine each with 6 cores. We will have multiple users each trying to use multiple cores.
- ▶ It is not reasonable to expect users to coordinate with each other to allocate resources appropriately.
- ▶ So we use a job-scheduler to take care of resource management
- ▶ Our job scheduler is called TORQUE and you can find all kinds of information about it on the internet.

- ▶ TORQUE actively keeps track of the load on each machine in the cluster.
- ▶ TORQUE keeps a job queue, each job lists the resources it would like to use
- ▶ When the cluster has resources available for the job, the job runs ... this can be some time after the job was submitted.
- ▶ Output from the job is written to file.
- ▶ Not surprisingly, there are lots of possible options here ...

You may submit two types of jobs

- ▶ *interactive* - You request some resources from TORQUE. As soon as they become available a prompt appears, and you control those resources in real time from a command prompt on one of the machines.
- ▶ *batch* - This is the standard way to use TORQUE. You write a script defining what resources your job needs, and what to do when the resources become available. You submit the job to a queue, and the job runs when resources become available. Output is written to a file. You do not need to be logged in for the job to run.

To begin an interactive session:

- ▶ Log in to the head node
- ▶ Type:

```
$ qsub -I -V
```

to start an interactive job. This will give you a new prompt on one of the nodes.

- ▶ You have now reserved some resources from the job scheduler, and you have control of those resources from this prompt.
- ▶ To exit the job just type exit.

- ▶ `qsub -I` will default to allocating 1 node and 1 core to your job.
- ▶ You can request more nodes
`$ qsub -I -V -l nodes=4`
- ▶ You can request more cores per node
`$ qsub -I -V -l nodes=4:ppn=2`

- ▶ TORQUE lets you submit jobs interactively with `qsub -I`, but it also lets you submit jobs to its batch queue which run non-interactively.
- ▶ Nice if you have a big job (or medium size) and you are going to have to wait for resources before you can run.
- ▶ TORQUE job specification files usually end with the `.pbs` extension.

An example.pbs file

```
#!/bin/sh
```

```
#Lines that begin with #PBS
```

```
#are options for PBS.
```

```
#Lines that don't begin with a # are interpreted by the shell
```

```
#Set the name of the job.
```

```
#It will default for you to something ugly.
```

```
#PBS -N eichholz
```

```
#You can set the name of the output files.
```

```
#I don't recommend it
```

```
##PBS -o foobs
```

```
##PBS -e foobs.e
```

```
#Setup the environment variables correctly
```

```
#PBS -V
```


#Mail me. Options are a,b,e a means email on abort,
#b means email me when the job begins, e means email
#when the job ends.

#PBS -m be

#My email address

#PBS -M eichholz@rose-hulman.edu

#Easiest to just give the full

#path to the executable you want run.

mpirun \${PBS_O_HOME}/sayhi

- ▶ Now, submit the job to the queue using
`qsub example1.pbs`
- ▶ The options in the file are requesting resources for you
- ▶ They can be quickly overridden from the command line though.'

- ▶ Examine the state of the batch queue using

```
$qstat
```

Job id	Name	User
533.mpi-master	STDIN	oliverr
534.mpi-master	STDIN	oliverr
535.mpi-master	STDIN	oliverr
536.mpi-master	eichholz	eichholz

- ▶ Delete a job with
`$qdel 536.mpimaster`

- ▶ http://www.csc.fi/english/pages/louhi_guide/batch_jobs/commands/qsub
- ▶ <https://wikis.nyu.edu/display/NYUHPC/Tutorial+-+Submitting+a+job+using+qsub>