How do I schedule a job with Torque?

Tell Me

1. Torque is an Open Source scheduler based on the old PBS scheduler code. The following is a set of directions to assist a user in learning to use Torque to submit jobs to the URC cluster(s). It is tailored specifically to the URC environment and is by no means comprehensive.
2. Details not found in here can be found online at http://docs.adaptivecomputing.com/torque/6-0-1/help.htm

Note:
Some of the sample scripts displayed in the text are not complete so that the reader can focus specifically on the item being discussed. Full, working examples of scripts and commands are provided in the Examples section at the end of this document.

Submitting a Job

To submit a job to the Copperhead cluster, you must first SSH into the Research Computing submit host, hpc.uncc.edu. Scheduling a job in Torque requires creating a file that describes the job (in this case a shell script) and then that file is given as an argument to the Torque command "qsub" to execute the job.

First of all, here is a sample shell script (myjob.sh) describing a simple job to be submitted:

```bash
#!/bin/bash
#
# ==== Main ======
/bin/date
```

This script simply runs the 'date' command. To submit it to the scheduler for execution, we use the Torque qsub command:

```
$ qsub -N "MyJob" -q "copperhead" -l procs=1 my_script.sh
```

This will cause the script (and hence the date command) to be scheduled on the cluster. In this example, the "-N" switch gives the job a name, the "-q" switch is used to route the job to the "copperhead" queue, and the "-l" switch is used to tell Torque (PBS) how many processors your job requests.

Many of the command line options to qsub can also be specified in the shell script itself using Torque (PBS) directives. Using the previous example, our script (my_script.sh) could look like the following:

```bash
#!/bin/sh
#
# ===== PBS OPTIONS =====
### Set the job name
#PBS -N "MyJob"
### Specify queue to run in
#PBS -q "copperhead"
### Specify number of CPUs for job
#PBS -l procs=1
#
# ==== Main ======
/bin/date
```

This reduces the number of command line options needed to pass to qsub. Running the command is now simply:

```
$ qsub my_script.sh
```

For the entire list of options, see the man page qsub i.e.

```
$ man qsub
```

Standard Output and Standard Error

In Torque, any output that would normally print to stdout or stderr is collected into two files. By default these files are placed in the initial working directory where you submitted the job from and are named:

```
scriptname.{o}jobid for stdout
scriptname.{e}jobid for stderr
```

In our previous example (if we did not specify a job name with -n) that would translate to:

```
My_script.sh.oNNN
My_script.sh.eNNN
```

Where NNN is the job ID number returned by qsub. If I named the job with -N (as above) and it was assigned job id 801, the files would be:

```
MyJob.o801
MyJob.e801
```

Logs are written to the job’s working directory ($PBS_O_WORKDIR) unless the user specifies otherwise.
**Monitoring a Job**

Monitoring a Torque job is done primarily using the Torque command “qstat.” For instance, to see a list of available queues:

```
$ qstat -q
```

To see the status of a specific queue:

```
$ qstat "queuename"
```

To see the full status of a specific job:

```
$ qstat -f jobid
```

where jobid is the unique identifier for the job returned by the qsub command.

**Deleting a Job**

To delete a Torque job after it has been submitted, use the qdel command:

```
$ qdel jobid
```

where jobid is the unique identifier for the job returned by the qsub command.

**Monitoring Compute Nodes**

To see the status of the nodes associated with a specific queue, use the torque command pbs_nodes(1) (also referred to as qnodes):

```
$ pbsnodes :queue_name
```

where queue_name is the name of the queue prefixed by a colon (:). For example:

```
$ pbsnodes :copperhead
```

would display information about all of the nodes associated with the “copperhead” queue. The output includes (for each node) the number of cores available (np=). If there are jobs running on the node, each one is listed in the (jobs=) field. This shows how many of the available cores are actually in use.

**Parallel (MPI) Jobs**

Parallel jobs are submitted to Torque in the manner described above except that you must first ask Torque to reserve the number of processors (cores) you are requesting in your job. This is accomplished using the -l switch to the qsub command:

For example:

```
$ qsub -q copperhead -l procs=16 my_script.sh
```

would submit my script requesting 16 processors (cores) from the “copperhead” queue. The script (my_script.sh) would look something like the following:

```bash
#!/bin/bash
module load openmpi
mpirun -hostfile $PBS_NODEFILE my_mpi_program
```

If you need to specify a specific number of processors (cores) per compute host, you can append a colon (:) to the number of specified nodes and then append the number of processors per host. For example, to request 16 total processors (cores) with only 4 per compute host, the syntax would be:

```
$ qsub -q copperhead -l nodes=4:ppn=4 my_script.sh
```

As described previously, options to qsub can be specified directly in the script file. For the example above, my_script.sh would look similar to the following:

```bash
#!/bin/bash
### Set the job name
#PBS -N MyJob
### Run in the queue named "copperhead"
#PBS -q copperhead
### Specify the number of cpus for your job.
#PBS -l nodes=4:ppn=4
### Load OpenMPI environment module.
module load openmpi
### execute mpirun
mpirun my_mpi_program
```

**Examples of Torque Submit Scripts**

NOTE: Additional sample scripts can be found online in /apps/torque/examples.
[1] Simple Job (1 CPU)

```bash
#!/bin/bash
#PBS -N MyJob
#PBS -q copperhead
#PBS -l procs=1

# Run program
/bin/date
```

[2] Parallel Job – 16 Processors (Using OpenMPI)

```bash
#!/bin/bash
#PBS -N MyJob
#PBS -q copperhead
#PBS -l procs=16

### load env for Infiniband OpenMPI
module load openmpi/1.10.0-ib

# Run the program "simplempi" with an argument of "30"
mpirun /users/joe/simplempi 30
```

Related FAQs

- High-Performance Computing (HPC)
- What are Environment Modules?
- How do I schedule a job with Torque?
- Why did I receive a "MOAB job resource violation" email?
- Can I transfer files using the URC batch scheduler (Torque/Moab)?