Submitting a Scheduler Job on Nero

Jobs are submitted by creating a script file and passing this to the qsub command. The script file should contain PBS directives, then any variable declarations, then the command to be run. (PBS is the scheduler upon which Torque is based and is used in a very similar way).

You must create your script in the cluster folder in your home folder, as only this folder is shared with the execution nodes. If you specify any relative file paths in your job then these will be relative to this folder. Any output files your job creates will appear in this folder.

This folder is actually a link to the real folder at /cluster/*loginname*. This is the path at which it is mounted on the execution hosts, and so within your job it can be referred to by this path. It is actually your home folder on these machines and so can also be referred to as /home/*loginname*, or by the environment variable $HOME.

Creating a job script file

Directives

PBS directives are lines beginning #PBS and have a number of uses. Other lines beginning with a # are comments

```
# specify the shell to run the job
#PBS –S /bin/bash

# name the job
#PBS –N myjob

# use the submission environment
#PBS –V

# redirect job’s standard output
#PBS –o /path/to/stdout.txt

# redirect job’s standard error
#PBS –e /path/to/stderr.txt
```

Requesting Resources

Jobs must request cluster resources. This is done via the #PBS –l directive, or the –l option to qsub. If both are used then the qsub option takes precedence.

The table below shows common resource requests and their default values if not specified by the user.

Resource	Description	Default Value
`walltime`	The maximum amount of real time that a job can be in the running state. If this is exceeded then the job will be aborted. Can be specified as a number of seconds or [[HH:]MM:]SS `#PBS -l walltime=1:00:00`	1 hour
`nodes`	The number of nodes the job can span. If specifying nodes then you can further specify the number of processors per node (`ppn`), separated by a colon. `#PBS -l nodes=2:ppn=16` You can request up to 2 nodes and up to a total of 64 processors.	1 node and 1 processor
`ncpus`	The number of CPU cores requested on a single node where a job cannot span more than one node. `#PBS -l ncpus=10` You can request up to 64 processors. You cannot specify both `nodes` and `ncpus`.	1 core
`mem`	The maximum amount of physical memory that the job can use. Use the suffixes kb, mb, gb. `#PBS -l mem=1gb` This is ignored for jobs spanning more than one node. In this case use `pmem` instead.	1 Gbyte
`pmem`	The maximum amount of physical memory used by a single process in a job. Use this for job arrays and jobs that request more than one node. Use the suffixes kb, mb, gb. `#PBS -l pmem=256mb`	1 Gbyte

The Job Environment

If you specify the –V directive, each running job as access to a number of environment variables. These are some examples.

Resource	Description
PBS_O_JOBNAME	User specified job name
PBS_JOBID	Unique ID number assigned to the job by Torque
PBS_ARRAYID	Array index of a job that is part of a job array
PBS_O_HOST	Host running the job
PBS_O_HOME	User's home folder, /home/*loginname* on the front end. Using this variable on the execution nodes will refer to the folder /home/*loginname*/cluster/
PBS_O_WORKDIR	Job's submission directory, that is the folder from which qsub was run. This will be somewhere in /home/*loginname*/cluster/ but this variable will contain the real path within /cluster/*loginname* and so can be used on the execution nodes
PBS_O_LOGNAME	Name of user

An example of the use of these variables might be to ensure that a job is run on an execution host from the same directory from which it was submitted on the front end in order to find its input files. Suppose that your job was submitted from a folder called /home/*loginname*/cluster/jobdir, ie you ran qsub from this folder. This is actually /cluster/*loginname*/jobdir, and this is the value that would be held in $PBS_O_WORKDIR. On the execution nodes running your job this path refers to the same folder. Therefore including the statement

```
cd $PBS_O_WORKDIR
```

after the directives and before the command to launch the job would ensure that all relative file paths refer to the same files on all nodes as on the front end.

The variable $PBS_O_HOME will have the value /home/*loginname*, which on the execution nodes is the same as /cluster/*loginname* on the front end, ie your job's top level parent directory.

Another useful example might be to use $PBS_JOBID or $PBS_ARRAYID to construct the name of an output file.

Job Arrays

Job arrays allow the same script to be submitted multiple times via a single command. This is specified by the –t option either to qsub or as a PBS directive, followed by series of array indices. These are specified as a comma separated list of single values and ranges. For example

```
#PBS -t 0-5,10
```

or

```
qsub -t 0,1,2,3,4,5,10 …..
```

When running a job array you should specify the resource request `pmem` which will request an amount of memory per process, rather than `mem` which requests a total for the job.

When running, each job has available to it its array index as the environment variable PBS_ARRAYID. This can be used to make each job behave differently, for example by reading different input files or creating different output files.

Monitoring Jobs

The qsub command will output the scheduler ID of your new job. Use qstat to see information about your jobs. You will see columns representing job ID, user defined job name, run time so far and status (R=running, Q=waiting, E=exiting, H=held, S=suspended, C =complete).

Use the –f parameter to see all information about a job. To see information about a job array via qstat, use the –t option.

The PBS directive –m tells the scheduler to email the user on specified job events. Define this using any combination of b (beginning), e (end) and a (abort). The user's email address must also be specified using the –M directive.

```
# email when job completes
#PBS -m e
#PBS -M user@domain.com
```

Each job will produce standard output and standard error files that, unless redirected, will appear in the job's working directory and be named *jobname*.o*jobID* and *jobname*.e*jobID*.

Cancelling a job

This is done via the qdel command, with the only parameter being the job ID.

More Information

For more information about creating job scripts and using the scheduler commands, see the man pages, eg

```
man qsub
man qstat
man qdel
```

The official Torque documentation can be found here

http://docs.adaptivecomputing.com/suite/8-0/basic/help.htm#topics/torque/2-jobs/submittingManagingJobs.htm