QSUB Manual

This is a tutorial for submitting Jobs on DPC environment.
Job Submission and SchedulingActual Queue System QUEUENodes/CPUs Arch Max Available Time limit per CPU Total Time limit planck 14/28 PIII
 28 36:00:0072:00:00
 fast 04/16
 AMD64 16 24:00:00
36:00:00
 grid 14/28
 PIII 08
 24:00:00
36:00:00
 gridats 14/28 PIII 08
 24:00:00 36:00:00 serial 18/40
 PIII 01
 24:00:00
36:00:00

 Quick start

In general submit a PBS jobscript with

qsub script.pbs

It is possible to use the following queues:

 planck 18 node x 2 processors (18 Intel Pentium III, 1100Mhz, 2Gb RAM) fast 4 node x 4 processors (4 Opteron Dual Core 1800Mhz, 4Gb RAM).

Our system has a normal ethernet connection, 2xGigabit from frontend to disk storage and between nodes of queue fast and the frontend, 10/100 between nodes of the queue planck and the frontend.

An example PBS job script can be found in:

/lfi_dpc_test/Pipelines/test_signal/pbsfiles

Maybe that's useful to consider - this example uses "mpiBatch" to distribute a list of commands onto the available nodes:

```
#####################BEGIN EXAMPLE PBS SCRIPT####################
#!/bin/bash
#PBS -N logfiles/multimod
#PBS -l nodes=4:ppn=2
#PBS -q fast
#PBS -r n
#PBS -u <username>
```

You should substitute <username> with your actual username !!!nodes number of nodes ppnnumber of cpus per node

To monitor jobs you can use the command line

qstat -a

To make an interactive session (say with IDL), just run

qsub -I
 The Long Story Introduction

Jobs are submitted using the TORQUE/PBS resource manager. TORQUE takes care of the low-level mechanics of job submission, monitoring node activity and running or terminating jobs on nodes. When and where jobs are run is decided by the Maui scheduler, based on information provided by TORQUE and by policies specified within the Maui configuration. Some functionality such as monitoring job-status or deleting jobs can be achieved using either TORQUE or Maui commands. On occasions, when Maui has difficulty communicating with TORQUE because TORQUE is busy, then it may be more effective to use the TORQUE commands. Documentation for TORQUE commands is provided by the extensive online man pages, eg. man qsub. Documentation for Maui user commands is available on the Web.

The batch system has the following components.
- Job Server - also called pbs_server provides the basic batch services such as receiving/creating a batch job, modifying the job, protecting the job against system crashes, and running the job.
- Job Executor - a daemon (pbs_mom) that actually places the job into execution when it receives a copy of the job from the Job Server. Mom creates a new session as identical to a user login session as is possible and returns the job's output to the user.
- Job Scheduler - TORQUE allows each site to create its own Scheduler. We are using the Maui Scheduler. The Maui Scheduler can communicate with various Moms to learn about the state of a system's resources and with the Server to learn about the availability of jobs to execute. Submitting Jobs

Jobs are submitted to the queue using the Torque command qsub to submit jobs. The commands to be run must be placed in a script file - qsub doesn't accept commands as direct arguments . There are extensive man pages for qsub (use man qsub to view the man pages) and other Torque commands, but a few examples should help you get started.TORQUE Environment Variables

Before entering in some details on how to submit a job on the cluster we list some useful environment variables used by TORQUE.

There are a number of predefined environment variables. These include the following:
- Variables defined on the execution hosts
- Variables exported from the submission host to the execution host;
- Variables defined by TORQUE.

The following environment variables relate to the submission machine: NameDescription
PBS_O_HOST The host machine on which the qsub command was run.
PBS_O_LOGNAME The login name on the machine on which the qsub was run.
PBS_O_HOMEThe home directory from which the qsub was run.
PBS_O_WORKDIR The working directory from which the qsub was run.

The following variables relate to the environment where the job is executing:Name Description
PBS_ENVIRONMENT
This is set to PBS_BATCH for batch jobs and to PBS_INTERACTIVE for interactive jobs.PBS_O_QUEUEThe original queue to which the job was submitted.
PBS_JOBIDThe identifier that PBS assigns to the job.
PBS_JOBNAMEThe name of the job.
PBS_NODEFILEThe file containing the list of nodes assigned to a parallel job.

Submitting Sequential Jobs

To submit a sequential job use the command:

%qsub my_script

This submits the commands in the file my_script to a single node. The file my_script can be as simple as the following two lines

#!/bin/sh

my_prog < input_file > output_file

Note however, that by default, jobs run in the directory in which they were submitted. So you might want to add a line to the script which cd's to the working directory before my_prog is executed

Submitting Parallel Jobs

For a multi-node job you need something like:%qsub -l nodes=4:ppn=2 my_script

To get 4 nodes with 2 processors per node (ie 8 processors in total). See the example MPI script for details of how to construct a parallel job script.

```
#!/bin/sh
#PBS -l nodes=4:ppn=2
#PBS -l walltime=04:00:00
#Change to directory from which job was submitted
cd $PBS_O_WORKDIR
# set number of processors to run on (list of node names is in file $PBSNODES)

NUM_NODES=`wc -l $PBS_NODEFILE | awk '{ print $1 }'`
/usr/bin/mpirun -np $NUM_NODES -machinefile $PBS_NODEFILE mpi_prog
```

Specifying Job Resources

The resources which may specified for a job include the the total runtime, number of nodes, amount of memory per node or other resources listed by man pbs_resources. If these are not specifed then the defaults are: 1 node for 72 hours requiring upto 1977 MB of memory per node (see note on memory specification). Non-default resource requirements are specified by including them in the -l option argument on the qsub command or in the PBS job script. eg.

%qsub -l walltime=10:30:00 my_script

submits a job with a maximum runtime of 10 hours 30 minutes. While:

%qsub -l walltime=1:12:00:00,mem=3000MB my_script

submits a job with a maximum runtime of 1 day 12 hours to a node with at least 3000 MB of Memory.

To get 4 nodes with 2 processors per node for upto 12 hours use

%qsub -l walltime=12:00:00,nodes=4:ppn=2 my_script
Configuring TORQUE Scripts

Default resource requirements can be defined within a job script by including a line begining with "#PBS" near the beginning - before the first executable line (these are called PBS "Directives". If my_script contained the two lines

#PBS -l walltime=30:00:00
#PBS-l nodes=2:ppn=2

it would request 2 nodes for 30 hours when submitted with:

qsub my_script

but any resource specified by a directive can be overridden by the resources specified on the qsub command line.

Below are some of the commonly used TORQUE options in a job script file. Option Description
#PBS -N myJob
Assigns a job name. The default is the name of PBS job script.
#PBS -l nodes=4:ppn=2
The number of nodes and processors per node.
#PBS -q queuename
Assigns the queue your job will use.
#PBS -l walltime=01:00:0 The maximum wall-clock time during which this job can run.
#PBS -o mypath/my.out
The path and file name for standard output.#PBS -e mypath/my.err

The path and file name for standard error.#PBS -j oe
Join option that merges the standard error stream with the standard output stream of the job.
#PBS -W stagein=file_list Copies the file onto the execution host before the job starts. (*)#PBS -W stageout=file_list
Copies the file from the execution host after the job completes. (*)
#PBS -m b
Sends mail to the user when the job begins.
#PBS -m e
Sends mail to the user when the job ends.
#PBS -m a
Sends mail to the user when job aborts (with an error).
#PBS -m ba
Allows a user to have more than one command with the same flag by grouping the messages together on one line, else only the last command gets executed. #PBS -r n
Indicates that a job should not rerun if it fails.
#PBS -V

Exports all environment variables to the job.
Monitoring and Deleting jobs

The control of all your jobs can be checked with either the TORQUE or the Maui commands.Monitoring Jobs

The status of all your jobs can be checked with either the TORQUE qstat or the Maui showq commands. The status of an individual job can be determined either with the qstat command or the Maui checkjob command, with the jobid (as returned by qsub) as an argument. e.g.

qstat 42789

The amount of information displayed can be varied with specifying various flags, e.g.. to show the nodes on which a job is running:

qstat -n 42789

To get an estimate of the time at which a job will start use the Maui command showstart with the jobid as argument (note that this may change as either jobs finish early or other jobs acquire more priority. It is only likely to be reasonably accurate as a job nears the top of the queue.) Once a job has finished the standard output and standard error for the job are placed by default in files in the directory from which the job was submitted, e.g. in my_script.o42789 and my_script.e42789. If you want to change this default behaviour see the qsub man page. As these files are only placed in your filestore after the job has terminated, you may need to redirect the standard output to a file (do this within your job script) if you need to observe the output while the job is running. Output redirection is also advisable if your output files are very large.

Deleting Jobs

Use the TORQUE qdel or Maui canceljob commands (with the jobid as argument) to delete a job (either queued or running).Summary TablesCommand Descriptionqstat -aCheck status of jobs, queues, and the PBS server
qstat -f
Get all the information about a job, i.e. resources requested, resource limits, owner, source, destination, queue, etc.qdel job.ID
Delete a job from the queueqhold job.ID Hold a job if it is in the queue
qrls job.ID Release a job from hold

Command
Description
showqShow a detailed list of submitted jobs

showbfShow the free resources (time and processors available) at the momen
checkjob job.ID Show a detailed description of the job job.ID
showstart job.ID Gives an estimate of the expected started time of the job job.ID
Interactive Jobs

On the master node no production is allowed and any serial execution program lasting more than 10 minutes is automatically deleted. Execution of serial application on computational nodes can be only done the through the queuing system, even for interactive runs.

On occasions, for instance when needing for debugging, it can be useful to start jobs directly from a compute node. To do this this the -I flag can be used with qsub. By default this will give a single node for 36 hours, but this can be changed with the normal flags to qsub. If sufficient resources are available, the interactive job will start immediately, otherwise it will still need to queue to start and the qsub command will hang until the request can be satisfied. Once the job has started the user is logged in to the node with the normal PBS environment (so a script that runs in batch mode can also be run in interactive mode, in particular, for multi-node jobs, the variable $PBS_NODEFILE contains the name of a file with the nodes allocated to the job).
As resources may not be available immediately to satisfy the requirements of an interactive job, it is normally only practical to use interactive jobs for short jobs of a few hours or less, running on a handful of nodes. Some estimate of what resources are available at any given time to run an interactive job can be gained with the Maui command showbf - This will show any nodes free, or nodes that are reserved for another job which is not able to start until a later date.
Simple examples

Suppose for instance you want to run your test.x interactivel on one node with one CPU and that you need to make tests for 30 minutes:%qsub -l walltime=0:30:00 -I

You will be prompted to the login session of one of the nodes of the cluster.

If test.x is a parallel program that requires 4 CPUs, you may want to run:

%qsub -l nodes=2:ppn=2,walltime=0:30:00 -I
Tips and Tricks

How to specify the nodes. Suppose one of the nodes of the cluster is not able to run a code (ex. ssh key broken) , it is possible to avoid the use of this node by specifing the list of nodes required.%qsub -l nodes=2:ppn=2 -W x="HOSTLIST:palantir10:palantir10:palantir11,palantir11"