

PBS Tutorial

Fangrui Ma*
Universit of Nebraska-Lincoln

October 26th, 2007

Abstract

In this tutorial we gave a brief introduction to using PBS Pro. We gave examples on how to write control script, and submit a PBS job to a queue. Some commonly used command examples also given here for user's convenience.

1 Introduction

The Portable Batch System (PBS) is a resource management system, which handles the management and monitoring of the computational workload on the Bioinformatics Core Reaserch Facility (BCRF) clusters. Users submit “jobs” to the resource management system where they are queued up until the system is ready to run them. PBS selects which job to run, and decides when and where to run the job in order to balance competing user needs and to maximize efficient use of the cluster resources.

To use PBS, you create a batch job command file which you submit to the PBS server to run on the BCRF clusters. A batch job file, normally called control script, is simply a shell script containing the set of commands you want run on some set of cluster computer nodes. It also contains directives which specify the characteristics (attributes), and resource requirements (e.g. number of computer nodes and maximum runtime) that your job needs. Once you create your PBS job file, you can reuse it if you wish or modify it for subsequent runs.

2 Submission host and job queue

In BCRF we have a submission host, `bioinfocore.unl.edu`, a default queue, `workq`, and a set of computation nodes. The submission host is the host machine where you can submit your jobs to the PBS system. The jobs submitted will be first queued in `workq` and then sent to computation nodes.

There are two sets of computation nodes. One has 32-bit CPU cores, and the other 64-bit CPU cores. The 32-bit CPU core computers can run only 32 bit applications. The 64-bit CPU core computer can run both 64- and 32-bit applications. With the default setting PBS sends your applications to some of the 32-bit or 64-bit computers. If you want definite 64-bit computers to run your application you have to specify it in your control script. There are examples in Section 7.

In order to use PBS job management system in BCRF, you need a Linux account. If you already have one you can login to the submission host and begin to use it. The web site for account application is given below. The link “Apply for an account” is on the right column of the page.

<http://biocore.unl.edu>

*Bioinformatics Core Research Facility.

3 Quick start

Suppose you have a program called `my_exe` and want to submit it to our cluster for running. Your Linux account user name is `jsmith`, and `my_exe` is in the `programs` folder under your home directory. First write a control script named `test_qsub` and save it in the `programs` directory.

```
# qsub control script: test_qsub.  
# Author: John Smith  
  
#PBS -j oe  
#PBS -o out.log  
  
#cd to your execution directory  
cd /home/jsmith/programs  
  
./my_exec
```

Then submit it to the cluster.

```
qsub test_qsub
```

There are more examples in the Example section.

4 PBS commands

The commonly used PBS commands are listed here according to their functionality. Each command with a brief explanation. For more information on each of the commands, you can use manual pages in the submission host (`bioinfocore.unl.edu`).

- Job control
 - qsub** submit a job
 - qdel** delete a batch job
 - qsig** send a signal to a batch job
 - qhold** hold a batch job
 - qrerun** rerun a batch job
 - qmove** move a batch job to another queue
- Job monitoring
 - qstat** show the status of batch jobs
 - qselect** select a specific subset of jobs
- Node status
 - pbsnodes** list the status and attributes of all nodes in the cluster.
- Others
 - qalter** alter a batch job
 - qmsg** send a message to a batch job

Table 1: qsub attributes

Attribute	Values	Description
-l	comma separated list of required resources (e.g. nodes=2, cput=00:10:00)	Defines the resources that are required by the job and establishes a limit to the amount of resources that can be consumed. If it is not set for a generally available resource, the PBS scheduler uses default values set by the system administrator. Some common resources are listed in the Requesting PBS Resources section.
-N	name for the job	Declares a name for the job
-o	[hostname:]pathname	Defines the path to be used for the standard output (STDOUT) stream of the batch job.
-e	[hostname:]pathname	Defines the path to be used for the standard error (STDERR) stream of the batch job.
-q	name of destination queue, server, or queue at a server	Defines the destination of the job. The default setting is sufficient for most purposes.
-M	email list	specify email recipient list.
-m	MailOptions	specify email notification options. It could be begin, end, or abort.

Table 2: PBS resources

Resources	Values	Description
nodes	details in Example section	declares the node configuration for the job.
walltime	hh:mm:ss	specifies the estimated maximum wallclock time for the job.
cput	hh:mm:ss	specifies the estimated maximum CPU time for the job.
mem	positive integer optionally followed by b, kb, mb, or gb	specifies the estimated maximum amount of RAM used by job. By default, the integer is interpreted in units of bytes.
ncpus	positive integer	declared the number of CPUS requested.

5 Set PBS job attributes

There are two ways to set a PBS job attributes. Table 1 listed some attributes.

1. using `qsub` arguments, or
2. using PBS directives in a batch script for `qsub`.

6 PBS resources

In the batch script, you can specify the resources needed by your job. This does not mean that you can just specify the maximum amount of resources possible at all times. PBS tries to optimize the utilization of resources and to balance the job load. If current resources does not meet your requirement, PBS will let your job wait in the queue and pick a job in the queue, whose job requirement can be satisfied by the current resources, to run. In general, it is better to ask for the amount of resources you need. Table 2 listed some resources you can use.

7 Examples

To run a job in batch mode, use `qsub` to a PBS control script in the following syntax.

```
qsub [options] <control script>
```

Draft

The job will be queue in the queue specified and schedule it to run based on the job priority and the availability of computing resources.

7.1 Batch script

The control script os essentially a shell script that executed a set of commands. The script also contain PBS directives that are used to set attributes for a job.

```
#PBS -N jobname
#PBS -q workq
#PBS -l nodes=3,walltime=0:12:30
#PBS -M johnsmith@unl.edu
#PBS -m abe
#PBS -j oe
#PBS -o out.log

#cd to your execution directory
cd /path/to/programs/code

./my_executable parameter list
```

In the script, #PBS prefix is PBS directives. You can specify the computing resources and how PBS handles your job.

The following example showed the way to run a mpi job.

```
#PBS -N jobname
#PBS -q workq
#PBS -l nodes=3,walltime=0:12:30
#PBS -l select=mpi_available=1
#PBS -M johnsmith@unl.edu
#PBS -m abe
#PBS -j oe
#PBS -o out.log

#cd to your execution directory
cd /path/to/programs/mpicode

mpirun -np 4 my_executable parameter list > mpi.out
```

If you want run your application only on the 64-bit computers, use the resource flag `select=arch64=1` in your control script like the following example.

```
#PBS -N jobname
#PBS -q workq
#PBS -l nodes=3,walltime=0:12:30
#PBS -l select=arch64=1
#PBS -M johnsmith@unl.edu
#PBS -m abe
#PBS -j oe
#PBS -o out.log

#cd to your execution directory
cd /path/to/programs/code

./my_executable parameter list
```

Draft

You can also put the 64-bit and mpi flags together in your control script as follows.

```
#PBS -l select=arch64=1;mpi_available=1
```

In this case PBS will run your mpi application on 64-bit CPU core computers.

The node configuration may also have one or more global modifiers of the form #<property> appended to the end of it which is equivalent to appending <property> to each node specification individually. That is,

```
4+3:fast+2:compute#large
```

is completely equivalent to

```
4:large+3:fast:large+2:compute:large
```

The shared property is a common global modifier. The following are some common node configurations. For each configuration, both the exclusive and shared versions are shown.

1. specified number of nodes:

```
nodes=<num nodes>
nodes=<num nodes>#shared
```

2. specified number of nodes with a certain number of processors per node:

```
nodes=<num nodes>:ppn=<num procs per node>
nodes=<num nodes>:ppn=<num procs per node>#shared
```

3. specific nodes:

```
nodes=<list of node names separated by '+'>
nodes=<list of node names separated by '+'>#shared
```

4. specified number of nodes with particular properties

```
nodes=<num nodes>:<property 1>:...
nodes=<num nodes>:<property 1>:...#shared
```

7.2 Interactive jobs

If your job need user intervention during the execution, you can submit your job using the “-I” flag as in the following syntax.

```
qsub -I <control_script>
```

7.3 Command examples

qsub -I submit an interactive job

qsub -q queue submit a job directly to a pacified queue

qstat list information about queues and jobs

qstat -q list all queues on system

qstat -Q list queue limits for all queue

Draft

qstat -a list all jobs on system

qstat -au userid list all jobs owned by userid

qstat -s list all jobs with status comments

qstat -r list all running jobs

qstat -f jobid list all information known about specified job

qstat -Qf queue list all information known about specified queue

qstat -B list summary information about the PBS server

qdel jobid delete specified job

qalter jobid modify the attributes of the job or jobs specified by jobid

pbsnodes -a list all nodes are currently available and some node characteristics.

pbsnodes -l list all nodes are currently offline

8 Comments

You are welcome to comment on this tutorial. My email address is fma2@unl.edu.