
CLUSTER FAQ

1. What is a Queue?

Queue is a collection of jobs (or job-related tasks) within the batch queuing system. Each queue has a set of associated attributes which determine what actions are performed upon each job within the queue. Typical attributes include queue name, queue priority, resource limits, destination(s) and job count limits. Selection/scheduling of job implementation is defined. The use of the term "queue" does not imply the ordering "first in, first out."

2. How will you create a job script in PBS?

PBS batch script, is a text file containing job setup information for the batch system followed by commands to be executed. It can be created using any text editor and may be given any name. A job script is simply a shell script. It consists of PBS directives, comments, and executable statements.

3. What are all the common queue statuses in PBS?

All PBS jobs require a script to specify the resource requirements and other parameters associated with a parallel job.

The common cases include:

- Q - signifies that the job is queued and awaiting its turn to be executed.
- R - indicates that the job is currently running.
- C - means that the job has been completed.

4. How to check running job status in detail?

Job status

The command for viewing job status is `qstat`. Without arguments, it displays status for all jobs currently in the system. For example: **`#qstat -an`**

5. What is the basic parameter required for running parallel script in PBS?

The short answer is just to use mpirun as normal.

When properly configured, Open MPI obtains both the list of hosts and how many processes to start on each host from PBS directly. Hence, it is unnecessary to specify the --hostfile, --host, or -np options to mpirun. Open MPI will use PBS mechanisms to launch and kill processes (rsh and/or ssh are not required). At run time, the variable PBS_NODEFILE is set to the location of a file that is created for your job that includes a list of the assigned nodes.

6. What happens if \$PBS_NODEFILE is modified?

Modifying the \$PBS_NODEFILE in each job according to local policies will cause Open MPI to behave in an unpredictable fashion. As long as no new hosts are added to the hostfile, it usually means that Open MPI will incorrectly map processes to hosts, but in some cases, it can cause Open MPI to fail to launch processes altogether.

The best course of action is to not modify the \$PBS_NODEFILE.

7. How do I determine if Open MPI is configured for PBS Pro?

If you are configuring and installing Open MPI yourself, and you want to ensure that you are building the components of Open MPI required for Torque/PBS Pro support, include the **--with-tm** option on the configure command line. Run **./configure --help** for further information about this configure option.

The **ompi_info** command can be used to determine whether or not an installed Open MPI includes Torque/PBS Pro support:

shell\$ ompi_info | grep ras If the Open MPI installation includes support for Torque/PBS Pro, you should see a line similar to that below. Note the MCA version information varies depending on which version of Open MPI is installed. **MCA ras: tm (MCA v2.1.0, API v2.0.0, Component v3.0.0)**

8. What are all the basic PBS Commands?

The following is a list of the basic PBS commands. They are organized according to their functionality.

Job Control

- qsub: Submit a job
- qdel: Delete a batch job
- qsig: Send a signal to batch job
- qhold: Hold a batch job
- qrls: Release held jobs
- qrerun: Rerun a batch job
- qmove: Move a batch job to another queue

Job Monitoring

- qstat: Show 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.

Miscellaneous

- qalter: Alter a batch job
- qmsg: Send a message to a batch job

9. What are the basic PBS Resources available?

PBS resources are requested by using the -l job attribute (see the Setting PBS Job Attributes section).

The following is a list of some common PBS resources. A complete list can be found in the pbs_resources

manual pages.

Resource	Values	Description
nodes	-	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 the job. By default, the integer is interpreted in units of bytes.
ncpus	positive integer	Declares the number of CPUs requested.

10. What are the basic PBS Environment Variables?

The following is a list of the environment variables set by PBS for every job.

Variable	Description
PBS_ENVIRONMENT	set to PBS_BATCH to indicate that the job is a batch job; otherwise, set to PBS_INTERACTIVE to indicate that the job is a PBS interactive job

PBS_JOBID	the job identifier assigned to the job by the batch system
PBS_JOBNAME	the job name supplied by the user
PBS_NODEFILE	the name of the file that contains the list of the nodes assigned to the job
PBS_QUEUE	the name of the queue from which the job is executed
PBS_O_HOME	value of the HOME variable in the environment in which qsub was executed
PBS_O_LANG	value of the LANG variable in the environment in which qsub was executed
PBS_O_LOGNAME	value of the LOGNAME variable in the environment in which qsub was executed
PBS_O_PATH	value of the PATH variable in the environment in which qsub was executed
PBS_O_MAIL	value of the MAIL variable in the environment in which qsub was executed
PBS_O_SHELL	value of the SHELL variable in the environment in which qsub was executed
PBS_O_TZ	value of the TZ variable in the environment in which qsub was executed
PBS_O_HOST	the name of the host upon which the qsub command is running
PBS_O_QUEUE	the name of the original queue to which the job was submitted
PBS_O_WORKDIR	the absolute path of the current working directory of the qsub command

11. How to select a Queue?

PBS is often set up with multiple job queues which assist the PBS scheduler to determine the order in which to run jobs. These are typically organized to sort jobs by size. A user should select a queue based on an estimate for the size of his/her job. A list of the available queues can be found by using the, **qmgr** utility with the **ps** command.

- If the resource requirements for a job exceed the maximum values for the queue it is submitted to, the job will automatically be terminated by PBS.
- Unless otherwise specified (see the Setting PBS Jobs Attributes section), jobs are submitted to the, which is the most resource constrained queue.

12. How to run a batch job?

To run a job in batch mode, a control script is submitted to the PBS system using the **qsub** command:

qsub [options] <control script>

PBS will then queue the job and schedule it to run based on the job's priority and the availability of computing resources.

The control script is essentially a shell script that executes the set commands that a user would manually enter at the command-line to run a program. The script may also contain PBS directives that are used to set attributes for the job. Here is a simple script that prints some environment information and runs the program simultaneously on four nodes.

Special features of this control script should be noted:

- The lines beginning with #PBS are the PBS directives that set job attributes. Actually, the job attributes are just the options that can be passed to **qsub** as command-line arguments. A few of the more common attributes are listed in the Setting PBS Job Attributes section. For a complete list, see the OPTIONS section of **qsub** manual pages.

- PBS automatically defines several environment variables that start with the PBS_ prefix that contain information about the environment in which the control script is executing. A complete list of the PBS environment variables can be found in the PBS Environment Variables section or in the qsub manual pages. The wait command after the for-loop is essential to make sure that the PBS jobs waits for the background jobs to complete before exiting.

13. How to stop or cancel a running job?

- Find the job ID using the qstat command. The job ID is the number listed under the "Job ID" field (NOT including the server name). For example, if the job ID listed by qstat is 394.master.amcl, the job ID is 394.
- Send the running job the KILL signal by using the command

qsig -s KILL <job ID>

14. How to use an interactive session?

For debugging purposes, it is often useful to be able to use an "interactive" session. From a user's perspective, an interactive session is not very different from logging directly into and working on a compute node. However, there are a few essential differences:

- PBS will automatically attempt to place you on a node with a minimal load
- an interactive session may give the user control of multiple nodes/processors
- if sufficient resources are available, a user may request exclusive use of nodes/processors.

A user requests an interactive session by using the command

qsub -I

To request a specific node/processor configuration for the session, the optional -l command-line argument may be used. For example, the command

qsub -I -l nodes=2:compute #shared

will start an interactive session consisting of 2 nodes from the pool of ``compute'' nodes that are not exclusively allocated. More information on various node configurations can be found in the Specifying a Node Configuration section.

As for batch jobs, PBS sets several environment variables for interactive sessions. When multiple nodes are requested, the `$PBS_NODEFILE` variable is particularly important because it contains a list of the nodes that have been allocated to the session.

15. How to monitor a job?

Users can monitor jobs in the PBS system using the `qstat` command. Most commonly, it is used to examine the current load on the system. However, it can also be used to extract detailed information (e.g. wallclock time, cpu time, memory use) about specific jobs in progress.

By default, `qstat` returns information about all jobs in the system. Information about specific jobs is retrieved by providing `qstat` a list of job id's. More information can be found in the `qstat` manual pages.

16. How to troubleshoot a batch problem?

My job is running slower than it should!

Here are a few of the reasons your job may be running slowly:

- Your job has exceeded available physical memory and is swapping to disk. This is always a bad thing in an HPC environment as it can slow down your job dramatically. Either cut down on memory usage, request more memory, or spread a parallel job out over more nodes.
- Your job isn't using all the nodes and/or cores you intended it to use. This is usually a problem with your batch script.
- Your job is spawning more threads than the number of cores you requested. Context switching involves enough overhead to slow your job.
- You are doing too much I/O to the network file servers (home and project directories), or you are doing an excessive number of small I/O operations to the

parallel file server. An I/O-bound program will suffer severe slowdowns with improperly configured I/O.

- You didn't optimize your program sufficiently.
- You got unlucky and are being hurt by someone else's misbehaving job. As much as we try to isolate jobs from each other, sometimes a job can cause system-level problems. If you have run your job before and know that it usually runs faster, OSC staff can check for problems.

Someone deleted my job!

If your job is misbehaving, it may be necessary for HPC staff to delete it. Common problems are using up all the virtual memory on a node or performing excessive I/O to a network file server. If this happens you will be contacted by HPC Help with an explanation of the problem and suggestions for fixing it. We appreciate your cooperation in this situation because, much as we try to prevent it, one user's jobs can interfere with the operation of the system.

Occasionally a problem not caused by your job will cause an unrecoverable situation and your job will have to be deleted. You will be contacted if this happens.

Why can't I delete my job?

If you can't delete your job, it usually means a node your job was running on has crashed and the job is no longer running. HPC staff will delete the job.

My job is stuck!

There are multiple reasons that your job may appear to be stuck. If a node that your job is running on crashes, your job may remain in the running job queue long after it should have finished. In this case you will be contacted by AQUA and will probably have to resubmit your job.

If you conclude that your job is stuck based on what you see in qpeek, it's possible that the problem is an illusion. This comment applies primarily to code you develop yourself. If you print progress information, for example, "Input complete" and "Setup complete", the output may be buffered for efficiency, meaning it's not written to disk immediately, so it won't show up in qpeek. To have it written immediately, you'll have to flush the buffer; most programming languages provide a way to do this.

My job crashed. Can I recover my data?

If your job failed due to a hardware failure or system problem, it may be possible to recover your data from \$TMPDIR. If the failure was due to hitting the walltime limit, the data in \$TMPDIR would have been deleted immediately. Contact AQUA Help for more information.

The trap command can be used in your script to save your data in case your job terminates abnormally.

17. Why isn't my job running?

There are many reasons that your job may have to wait in the queue longer than you would like. Here are some of them.

- System load is high. It's frustrating for everyone!
- A system downtime has been scheduled and jobs are being held. Check the message of the day, which is displayed every time you login, or the system notices posted on AQUA webpage.
- You or your group have used a lot of resources in the last few days, causing your job priority to be lowered ("fairness policy").
- You or your group are at the maximum processor count or running job count and your job is being held.
- Your job is requesting specialized resources, such as large memory or certain software licences, that are in high demand.
- Your job is requesting a lot of resources. It takes time for the resources to become available.
- Your job is requesting incompatible or non-existent resources and can never run.
- Jobs are unnecessarily stuck in batch hold because of system problems (very rare!).

18. What are the available Scheduling Policies and Limits?

Fair-share limits

To keep any one user or group/project from monopolizing the system when others need the same resources, the scheduler imposes what are known as fair-share limits. If a user or group/project uses large amounts of computing resources over a period of a few days, any new jobs they submit during that period will have reduced priority.

Short jobs for debugging

A small number of nodes are set aside during the day for jobs with a walltime limit of 1 hour or less. Please remember to exit debug jobs when you are done using the resources, to free them up for other users.

GPU jobs

All GPU nodes are reserved for jobs that request gpus. Short non-GPU jobs are allowed to backfill on these nodes to allow for better utilization of cluster resources.

19. How do I change my cluster password?

Once you are logged in to the cluster, type the command:

passwd

and respond to the prompts by entering your old password and your new password. Please remember that this will change your password. Passwords must meet the following criteria:

- Must be 6 to 16 characters long.
- Must contain at least one alphabetic and one non-alphabetic character, or one number and one non-alphanumeric character.
- Cannot contain the user's user ID
- Is case-sensitive.
- Allowable characters are: a-z, A-Z, 0-9, and the symbols !@#\$%&*().

20. My text contains ^M characters and I get an error when I use the text. What should I do?

An example of such an error is:

```
/bin/ksh^M: bad interpreter
```

You have created a text file in the Windows text format. You must either store your file in the UNIX text format, convert the text file to the UNIX text format during transmission to server or later on convert the text file on server.

Use the dos2unix command to convert the text file on server, e.g.:

```
$ dos2unix myrun.job
```