PBS FAQ

(updated on Jan/'09 by EM)

What is PBS?

Portable Batch System (PBS) is a workload management system in which jobs are distributed across multiple timeshared hosts to even out the workload on each host.

What does it do?

From system administrator‘s perspective, it allows us to utilize many computing resources. From the user's perspective, a workload management system allows you to make more efficient use of your time. You specify the tasks you need to execute. The system takes care of running these tasks and returning the results back to you. If the available computers are full, then the workload management system holds your work and runs it when the resources are available.

How does it work?

You create a *batch job,* which you then submit to PBS. A batch job is simply a shell script containing the set of commands you want to run on some sets of execution machines. It also contains directives that specify the characteristics (attributes) of the job and resource requirements (*e.g.,* memory, CPU time, number of processors) for your job. Once you create your PBS job, you can reuse it or modify it for subsequent runs.

PBS also provides a special kind of batch job called *interactive-batch*. An interactive batch job is treated just like a regular batch job (it is queued up and waited for resources to become available before it can be run). Once it is started, however, the user's terminal input and output are connected to the job. So it appears that the user is logged into one of the available execution machines, and the resources requested are reserved for the user during the session.

How do you submit jobs?

[By interactive mode]

This is the easiest method to start using PBS. It logs the user into one of the available nodes in the cluster and the resources requested by the job are reserved for that job. This is useful for debugging/testing applications or if you want to run small jobs without preparing a script.
Simply type:

```
>qsub -I
```

on the command line, then you would get a message similar to:

```
qsub: waiting for job 1259.head.cluster to start
qsub: job 1259.head.cluster ready
```

Then you can start using it as a regular terminal session.

Remember that your job will be terminated when you logout from your session (just like the terminal session). ***Remember also that you are occupying the queue until you logout the interactive session. Associated resource with your queue (node, processors, etc.) will not be released for other users until you exit. Please be kind to other users!***

How do you submit jobs?

[By submitting a batch job]

An example script is as follows:

```
#!/bin/sh
#PBS -N jobname
#PBS -j oe
cd test
clustalw -infile=test -outfile=test.clw > log
```

The first line is standard for any shell script, which specifies which shell to use to execute the script. The Bourne shell (sh) is the default, but you can change this to your favorite shell. **"#PBS"** prefix is not a comment. Lines starting with **#PBS** are the PBS directives, which request certain resources and modify the behavior for how the queue handles the job. PBS reads down the shell script until it finds the first line that is not a valid PBS directive (without **#PBS** prefix), and then stops. It assumes the rest of the script is the list of commands or tasks that the user wishes to run.

In the above example, **-N** indicates the job name, **-j oe** requests PBS to combine the standard output and standard error to one file, which is written at the completion of your job. The remaining part of the file is the shell command to run your job.

If your script is called "mypbsscript", submit this script using the **qsub** command:

```
>qsub mypbsscript
1275.head.cluster
```

Upon successful submission of a job, PBS returns a job identifier (1275.head.cluster in the example above). You will need the job identifier for any actions involving the job, such as checking job status, modifying the job, tracking the job, or deleting the job.

You can override resource attributes contained in your job script by specifying them on the command line. In fact, any job submission option or directive that you can specify inside the job script can be specified on the **qsub** command line. This is particularly useful if you just want to submit a single instance of your job, but you don't want to edit the script.

Read **man** page for **qsub** to find out many options to control your job and the resource usage.

How can I check the status of my jobs?

To show your job status, simply type **qstat**. To see which node(s) your job is running on, use **qstat -n** command. To kill a job, use **qdel** command:

```
>qdel 1275.head.cluster
```

or simply

```
>qdel 1275
```