



UA SCIENCE

**LUNAR & PLANETARY  
LABORATORY**

## How do I run jobs on the LPL HPC?

PACMAN and HiPAS use the Torque system for queueing and Maui for scheduling batch jobs. The goal is to allocate our limited computing resources to users, on demand, as fairly as possible.

**Run your jobs on the compute nodes using qsub.**

**Jobs running on the head node will be killed by the systems administrators.**

**Run your jobs from your cdata directory, not from your hipas home directory.**

### Using Torque

To use Torque, simply put the commands you would normally use to run your job into a job script, and submit the job script to the cluster using `qsub`. You should refer to `man qsub` for more detailed information as you read the following overview.

Also, the man page for `qsub` is available online [here](#).

The `qsub` program has a lot of options which may be supplied on the command line, or as special directives inside the PBS job script.

### Example Job Script

The following job script declares a job having the name `myjob` and requiring one node. It then changes to the work directory, and sends the execution host name, current date, and working directory to standard output.

---

```
#!/bin/sh
## Set the job name
#PBS -N demo_job
#PBS -l nodes=1
# Run my Job
beorun --nolocal --np 1 /path/to/my/job
echo Host: $HOSTNAME
echo Date: $(date)
echo Dir: $PWD
```

Assuming the above job script is in a file called `myjob`, you would submit it as follows:

---

```
[bjosh@hipas]$ qsub myjob
15.hipas
```

Note that `qsub` returns the Job ID immediately, although the job is simply queued to run at some future time to be decided by the scheduler. The Job ID is an incrementing integer followed by the name of the submit host.

## Equivalent Job Started From Command Line

You are not required to use job scripts. You could instead type all the options and commands at the command line. However, job scripts make it easier to manage your actions and their results. Following is the equivalent command line version of the above job script.

---

```
[bjosh@hipas]$ qsub -N myjob -l nodes=1:ppn=1 -j oe
cd $PBS_O_WORKDIR
echo Host: $HOSTNAME
echo Date: $(date)
echo Dir: $PWD
^D
15.master
```

We entered all of the `qsub` options on the initial command line. The `qsub` read our job commands line by line until we typed Control-D, the end of file character. At that point, `qsub` queued the job and returned the Job ID to us.

## A More Complex Job Script Using MPICH

TODO

## Checking Job Status

Check the status of your job using `qstat`. Here's an example with output:

```
$ qsub myjob && watch qstat -n

master:

Job ID          Username Queue      Jobname      SessID NDS  TSK  Req'd  Req'd  Elap
-----  -----  -----  -----  -----  ---  ---  ---    ---    ---
15.hipas        bjosh  default  myjob        --     1   --     --    00:01 Q    --
  --
```

The `watch` command is used to execute the `qstat -n` command every 2 seconds by default. This will help you see the progression of events. Press Control-C to interrupt `watch`.

Some Helpful commands

Command	Purpose
<code>ps -ef   bpstat -P</code>	Display all running jobs, with node number for each.
<code>qstat -Q</code>	Display status of all queues.
<code>qstat -n</code>	Display status of queued jobs.
<code>qstat -f JOBID</code>	Display very detailed information about JOBID.
<code>qstat -Q -f</code>	Display status of all queues in more detail.
<code>pbsnodes -a</code>	Display status of all nodes.

## How to Find Which Nodes Your Job is Using

`qstat -an`

Note your jobid(s).

`qstat -f jobid`

Note the process id(s) of your job(s).

```
ps -ef | bpstat -P | grep yourname
```

The number of the node running your job will be displayed in the first column of output.

## Where To Find Job Output

When your job terminates, Torque will store its output and error streams in files in the script's work directory.

The output file is `[JOBNAME].o[JOBID]` by default. You can override that using the `qsub -o PATH` option.

The error file is `[JOBNAME].e[JOBID]` by default. You can override that using the `qsub -e PATH` option.

The `qsub -j oe` option can be used to join the output and error streams into a single file.