

PBS arrays

Using **PBS Arrays** allows users to submit a large number of jobs based on the same PBS command file. Additionally, array job naming convention allows users to reference the entire set of jobs as a unit, or to reference one particular job from the set. Job arrays are submitted through the use of the `-t` option to `qsub`, or by using `#PBS -t` in your job file.

How to submit array of job via PBS

There are various methods to use arrays to submit jobs to PBS. The first method using the PBS command file below runs four individual Matlab scripts called `matlabdemo0`, `matlabdemo1`, `matlabdemo2` and `matlabdemo3`, and places the output from each script into `out.dat[0-3]`. The use of arrays significantly reduces the load on the queue manager, compared to the same number of individual jobs.

```
#!/bin/bash -l

#### job name & output files
#PBS -N demo_array
#### select resources
#PBS -l nodes=1:ppn=1
#PBS -l walltime=0:10:00
#PBS -l mem=4g
#### request that regular output (stdout) and
#### terminal output (stderr) go to the same file
#PBS -j oe
#### mail options
#PBS -m ae
#PBS -M ruser@yourinstitution.edu
#PBS -q standard

#### array size
#PBS -t 0-3

#### load matlab module
module load matlab/2011b

#### change directory to working directory
echo "Changing directory to work area"
cd $PBS_O_WORKDIR
```



```
#### start computation
matlab -nodisplay -nosplash < matlabdemo$PBS_ARRAYID.m > out.dat$PBS_ARRAYID
```

Four individual jobs will then appear in the output from `qstat`, e.g. `11-0.demo_array` for job one and `11-1.demo_array` for job two and so on. Each individual job can then be referenced using commands such as `qdel` and `qalter`, and are allocated separate resources for each job.

PBS multiply jobs

Another method of submitting multiply jobs via one PBS command file is to list each individual job on separate lines within the one command file. Below is an example of submitting four individual ***Matlab** jobs called `matlabdemo0`, `matlabdemo1`, `matlabdemo2` and `matlabdemo3`, and placing the output from each job into `out.dat[0-3]`.

```
#!/bin/bash -l

#### job name & output files
#PBS -N array_multi_test
#PBS -l nodes=1:ppn=1
#PBS -l walltime=0:10:00
#### set restart level
#PBS -c s
#### request that regular output (stdout) and
#### terminal output (stderr) go to the same file
#PBS -j oe
#### mail options
#PBS -m ae
#PBS -M ruser@yourinstitution.edu
#### select queue
#PBS -q standard

#### load matlab module
module load matlab/2011b

### change directory to working directory
echo "Changing directory to work area"
cd $PBS_O_WORKDIR
pwd
/bin/cat $PBS_NODEFILE

#### start computations
matlab -nodisplay -nosplash < matlabdemo0.m > out.dat0 &
matlab -nodisplay -nosplash < matlabdemo1.m > out.dat1 &
matlab -nodisplay -nosplash < matlabdemo2.m > out.dat2 &
matlab -nodisplay -nosplash < matlabdemo3.m > out.dat3 &

#### wait for background jobs to finish
wait
```

The four individual jobs will only appear as one job in the output from `qstat`, e.g. `11.demo_multy` for all individual jobs. Each individual job can not be referenced using commands such as `qdel` and `qalter`. There are some legitimate reasons for using this method. For example, as each individual job is run on the same node only one Matlab license will be used, whereas using PBS arrays could use four Matlab licenses to run the same individual jobs, as they will be run on different nodes. Note that this type of job submission runs all individual jobs on the one node, no matter how many nodes are specified in the PBS command file.