Copy of Submitting a job - Job arrays

This page is retained from an earlier version of the HPC wiki only for reference, and the equivalent up-to-date page is at Running jobs on the NYU HPC clusters.

Running many small tasks with Job Arrays or pbsdsh

Options for many similar jobs (array jobs and pbsdsh):

- \(-t 1,10,50-100\)
  - Submit an array of jobs with array ids as specified. Array ids can be specified as a numerical range, a comma-separated list of numbers, or as some combination of the two. Each job instance will have an environment variable \$PBS_ARRAYID
- \(-t 1,10,50-100\%5\)
  - As above, but the appended \"\%5\" specifies the maximum number of array items (in this case, 5) which should be running at one time
- Submit a single "shepherd" job requesting multiple processes and from it start individual jobs with pbsdsh.

The naive approach to running a large set of jobs based on the same script is to repeatedly \texttt{qsub} the script at the command line, perhaps changing a few environment variables, directories or input files each time.

A slightly less naive approach is to parameterize the script with some variables and \texttt{qsub} it in a shell loop.

Torque offers two methods, both more elegant than either of the above, for managing such workflows:

1. A job array groups a set of jobs under the same \texttt{PBS\_JOBID}, each with a unique \texttt{PBS\_ARRAYID}. Batch system commands such as \texttt{qstat}, \texttt{qdel}, etc can be called on individual jobs or on the job array as a whole.
2. If the individual jobs are small, the queuing overhead is relatively large. In this circumstance it is better to launch a single parallel job which uses pbsdsh to run the set of small jobs.

Job Arrays

See Using an Array Job to run a set of experiments

Using pbsdsh for many small jobs

(Still to come. A solution for when you have a huge number of jobs each needing only a few minutes, however it performs badly if the run times of the jobs are not uniform)