qsub arrays and 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 "%n" 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 `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 `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 `$PBS_JOBID`, each with a unique `$PBS_ARRAYID`. Batch system commands such as `qs`, `tat`, `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

(describe basics of monitoring them too, ie the syntax for `qstat`, `qdel` etc)

Using pbsdsh for many small jobs

(cover that it is best for many small jobs, mention unevenness in job run times)