Copy of qsub resource limit options

Options to request compute resources:

- **-l walltime=walltime**
  Maximum wallclock time the job will need. Default is 1 hour. Walltime is specified in seconds or as hh:mm:ss.

- **-l mem=memory**
  Maximum memory per node the job will need. Default depends on queue, normally 2GB for serial jobs and the full node for parallel jobs. Memory should be specified with units, eg 500MB or 8GB.

- **-l nodes=num:ppn=num**
  Number of nodes and number of processors per node required. Default is 1 node and 1 processor per node. The :ppn=num can be omitted, in which case (at NYU HPC) you will get full nodes. When using multiple nodes the job script will be executed on the first allocated node.

- **-q queue**
  Submit to a specific queue. If not specified, Torque will choose a queue based on the resources requested.

A job submitted without requesting a specific queue or resources will go to the default serial queue (s48 on Mercer) with the default resource limits for that queue.

Resources can be requested with multiple -l options, or as a comma-separated list of options. Both of the following examples are correct:

```
$ qsub -l walltime=2:00 -l mem=4GB myjob.q

$ qsub -l walltime=2:00,mem=4GB myjob.q
```

Most nodes on Mercer have 48GB or 64GB memory. Requesting a large portion of the memory on a node will cause Moab to reserve an entire node for your job even if you only request 1 CPU, since there will be insufficient remaining memory to run other jobs.

A small amount of memory on each node is needed by the operating system, so for example on a 64GB node, only about 62GB is available to jobs. A job requesting 64GB of memory will therefore be too big for a 64GB node, and Moab will schedule it on a 96GB or 192GB node instead. We have fewer nodes with so much memory, so the job is likely to spend longer waiting in the queue. **Tip: try requesting 62GB instead.**

The serial queues on NYU HPC clusters are limited to a single node, but allow multiple processors on that node to be used. Therefore, parallel jobs using only one node, such as OpenMP or multithreaded jobs can be submitted to a serial queue.

**When using more than one node, the job script is executed only on the first node.** To make use of the other nodes you must use MPI or pbsdsh.