Queues

High Performance Computing Centers use scheduling systems to run and monitor jobs submitted via various batch systems. NYU HPC uses MOAB/Torque for job scheduling. The queues are distributed in a way that favors shorter jobs. A simple MOAB fair share baseline is implemented to even out usage between users. Should you need more resources than the fair share allocations because of critical deadlines such as a grant application, a publication deadline, or class use, please email hpc@nyu.edu to make special arrangements. To see queues offered by a particular cluster, use the Torque/PBS command:

```
$ qstat -q
```

To request a specific queue, use the Torque/PBS operative below in your pbs script:

```
#PBS --q <queue name>
```

Same can be done from command line:

```
$ qsub --q <queue name>
```
This section outlines the queues associated with each of the clusters.

- **USQ Queues**
- **Bowery Queues**
- **Cardiac Queues**

### USQ Queues

<table>
<thead>
<tr>
<th>Type</th>
<th>Name of Queue</th>
<th>Maximum Walltime</th>
<th>Max Jobs Per User</th>
<th>Max CPU Core/User *</th>
<th>Maximum Nodes</th>
<th>Node Allocation Type</th>
<th>Active</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial</td>
<td>ser2</td>
<td>48 hours</td>
<td>N/A</td>
<td>64,128</td>
<td></td>
<td>Shared</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Serial</td>
<td>serlong</td>
<td>96 hours</td>
<td>N/A</td>
<td>32,64</td>
<td></td>
<td>Shared</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>interactive</td>
<td>4 hours</td>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>Shared</td>
<td>Yes</td>
<td>highest</td>
</tr>
</tbody>
</table>

**Notes:**

* Max CPU Core/User defines the largest processor count available to any one user. The first number represents a soft limit and the second number a hard limit. These flexible dual limits are set to ensure efficient utilization of cluster resources.

** Exclusive nodes versus shared nodes. Due to the complexity of message passing used by parallel jobs, all NYU HPC parallel queues are setup for "exclusive" node use, which means only one job can run on a node at the same time. Serial jobs using serial queues on the other hand can share the same node, up to matching the CPU/core count.

USQ is for running serial jobs. Please use Bowery for parallel jobs.

### Serial Queues

- **ser2**: Generic serial queue for jobs up to 48 hours.
- **serlong**: Generic serial queue for up to 96 hours.

### Interactive Queue

- **interactive**: This queue allows code testing and debugging in the compute node environment. Using the interactive queue for code testing alleviates extra stress on the login nodes. Compute node-based debugging is a highly recommended practice. The priority of the interactive queue is set to be the highest for easy access.

### Bowery Queues

<table>
<thead>
<tr>
<th>Type</th>
<th>Name of Queue</th>
<th>Maximum Walltime</th>
<th>Default Walltime</th>
<th>Max Jobs Per User</th>
<th>Max CPU Core/User</th>
<th>Maximum Nodes</th>
<th>Node Allocation Type</th>
<th>Active</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>p12</td>
<td>12 hours</td>
<td>1 hour</td>
<td>6</td>
<td>288,576</td>
<td>N/A</td>
<td>Exclusive</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Parallel</td>
<td>p48</td>
<td>48 hours</td>
<td>1 hour</td>
<td>N/A</td>
<td>64,128</td>
<td>N/A</td>
<td>Exclusive</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Serial</td>
<td>s48</td>
<td>48 hours</td>
<td>1 hour</td>
<td>500</td>
<td>36,72</td>
<td>1</td>
<td>Shared</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>bigmem</td>
<td>48 hours</td>
<td>1 hour</td>
<td>N/A</td>
<td>96,192</td>
<td>N/A</td>
<td>Shared</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>GPU</td>
<td>cuda</td>
<td>48 hours</td>
<td>1 hour</td>
<td>N/A</td>
<td>N/A</td>
<td>2</td>
<td>Exclusive</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

**Parallel Queues**

p12 and p48: Soft and hard limits are set to control CPU core totals per user (288,576 and 64,128, respectively). The scheduler will allocate jobs based on cluster traffic to maximize usage and will allow override of soft limits for individual users when resources are available.

All the p48 jobs can only be submitted to the 64 nodes from chassis 0 to 3 (compute-0-0 to compute-3-15), while p12 jobs can make use of all the compute nodes (compute-0-0 to compute-9-15). 64 out of 96 nodes from chassis 3 to 9 are owned by a private group but these nodes are available to the NYU public community when not being used by the group. The group’s jobs have the highest priority on such nodes.

Please specify "ppn=8" in your PBS script if you submit your jobs to the p48 queue. For the p12 queue, it is recommended to specify "ppn=12" for node balance purposes.

**Serial Queues**

The serial queue s48 on Bowery has 12 nodes on chassis 12 and all 32 nodes on chassis 13 for serial jobs. Each node has 12 cores with 48GB
memory. If you need memory more than 48GB, please use bigmem queue.

**Interactive Queue**

interactive: This queue allows code testing and debugging in the compute node environment. Using the interactive queue for code testing alleviates extra stress on the login nodes. Compute node-based debugging is a highly recommended practice. The priority of the interactive queue is set to be the highest for easy access.

**Bigmem Queue**

The bigmem queue has 16 nodes, each with 12 CPU cores and 96 GB memory, and one node with 16 CPU cores in total with 256 GB memory. It has been created for jobs with heavy memory usage requiring more than 24 GB of memory. If your memory requirement is 24 GB or less, please use other queues. Bigmem is one node with maximum ppn of 16.

**GPU Queue**

The cuda queue has 4 GPU nodes, each with 12 CPU cores, Nvidia Tesla M2090 GPU card and 24GB memory. It has been created for jobs that require GPU cores for computation (programs written in CUDA programming language). Please check this page for more information on using this queue to run cuda jobs on GPU nodes.

**Cardiac Queues**

Cardiac is a shared resource between a private owner and HPC/ITS ("Public"). MOAB FS, or "fair share" policy, is used to allocate resources between the private and public researchers to ensure priorities and to maximize utilization. The fair share policy aims to balance the 75/25% ownership ratio to flexibly increase the usage targets when private nodes are idle.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name of Queue</th>
<th>Maximum Walltime</th>
<th>Max CPU Core/User Per User</th>
<th>Max Queued Nodes</th>
<th>Maximum Nodes</th>
<th>Node Allocation Type</th>
<th>Active</th>
<th>Ownership</th>
<th>User Share</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel</td>
<td>p12</td>
<td>12 hours</td>
<td>192,384</td>
<td>Exclusive</td>
<td>Yes</td>
<td>Public</td>
<td>25% +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parallel</td>
<td>p48</td>
<td>48 hours</td>
<td>96,192</td>
<td>Exclusive</td>
<td>Yes</td>
<td>Public</td>
<td>25% +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial</td>
<td>ser2</td>
<td>48 hours</td>
<td>72,144</td>
<td>Shared</td>
<td>Yes</td>
<td>Public</td>
<td>25% +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Serial</td>
<td>serlong</td>
<td>96 hours</td>
<td>32/64</td>
<td>Shared</td>
<td>Yes</td>
<td>Public</td>
<td>25% +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive</td>
<td>interactive</td>
<td>4 hours</td>
<td>2</td>
<td>2</td>
<td>Shared</td>
<td>Yes</td>
<td>Public</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Parallal</td>
<td>card</td>
<td>no limit</td>
<td>no limit</td>
<td>no limit</td>
<td>Exclusive</td>
<td>Yes</td>
<td>Private</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>Parallal</td>
<td>s-card</td>
<td>no limit</td>
<td>no limit</td>
<td>no limit</td>
<td>Shared</td>
<td>Yes</td>
<td>Private</td>
<td>75%</td>
<td></td>
</tr>
</tbody>
</table>

**Public Queues**

Public queues on Cardiac are set to reflect general NYU HPC cluster usage policies.

p12: Max walltime = 12 hours

p48: Max walltime = 48 hours

ser2: Max walltime = 48 hours

serlong: Max walltime = 96 hours

**Interactive Queue**

interactive: This queue allows code testing and debugging in the compute node environment. Using the interactive queue for code testing alleviates extra stress on the login nodes. Compute node-based debugging is a highly recommended practice. The priority of the interactive queue is set to be the highest for easy access.

**Private Queues**

card: Parallel queue

s-card: Serial queue
PBS Script Generator
An interactive tool that generates PBS script based on user's input. Check this page for more details.

Front-Line HPC Consulting
HPC consultations are available once a week, Monday 1-3 PM. Appointments are required. Please make an appointment at hpc@nyu.edu.