<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC at NYU</td>
</tr>
<tr>
<td>HPC Newsletter</td>
</tr>
<tr>
<td>Getting started on Prince</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Prince How-to Articles</td>
</tr>
<tr>
<td>Logging in</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Mac / Linux</td>
</tr>
<tr>
<td>Clusters and Storage July 2017</td>
</tr>
<tr>
<td>Mercer (HPC, retired)</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
</tbody>
</table>
Dalmata (NYU Abu Dhabi)

Transfering data to/from the clusters

Transfering data to/from Prince cluster using Globus

Submittin g jobs with sbatch

Available software

Building Software Tutorials

Programming for Biologists

Research Gallery

FAQs

NYU HPC Cheat Sheet

Wiki Map

HPC Policies
Running jobs on the NYU HPC clusters

Job Scheduling Overview

Login and compute nodes

Queues

Writing a job script

Job stdout and stderr

Which filesystems should I use?

Submitting a Job

Basic qsub options and directives

Setting resource limits

Running MPI jobs

Running GPU jobs

Running R jobs

Running Matlab, Gaussian, etc jobs

Working interactively
Jobs are submitted with the `qsub` command:

```
$qsub options job-script
```

The options tell Torque information about the job, such as what resources will be needed. These can be specified in the job-script as PBS directives, or on the command line as options, or both (in which case the command line options take precedence should the two contradict each other). For each option there is a corresponding PBS directive with the syntax:

```
#PBS option
```

For example, you can specify that a job needs 2 nodes and 8 cores on each node by adding to the script the directive:
#!/bin/bash

#PBS -l nodes=2:ppn=8

or as a command-line option to qsub when you submit the job:

```bash
$ qsub -l nodes=2:ppn=8 my_script.q
```

### PBS directives and qsub options for working interactively

#### Options for running interactively on the compute nodes:

- **-I**
  Don’t just submit the job, but also wait for it to start and connect stdout, stderr and stdin to the current terminal.
- **-X**
  Enable X forwarding, so programs using a GUI can be used during the session (provided you have X forwarding to your workstation set up)
- **-q interactive**
  Run specifically in the interactive queue. At NYU HPC, this queue has smaller job limits (maximum of two nodes and 4 hours walltime) but very high priority.
- **-V**
  Pass the current environment to the interactive batch job
- **exit**
  To leave an interactive batch session, type `exit` at the command prompt.

Certain tasks need user interaction - such as debugging and some GUI-based applications. However the HPC clusters rely on batch job scheduling to efficiently allocate resources. Interactive batch jobs allow these apparently conflicting requirements to be met.

When you start an interactive batch job the command prompt is not immediately returned. Instead, you wait until the resource is available when the prompt is returned and you are on a compute node and in a batch job - much like the process of logging in to a host with `ssh`. **To end the session, type `exit`**, again just like the process of logging in and out with `ssh`.

```bash
$ qsub -I -X
qsub: waiting for job 3707318.crunch.local to start
qsub: job 3707318.crunch.local ready

$ hostname
compute-12-3
```

**To use any GUI-based program** within the interactive batch session you will need to extend X forwarding with the `-X` option or directive. This of course still relies on you having X forwarding at your login session - try running `xterm` before starting the interactive to verify that this is working correctly.

You can request resources for an interactive batch session just as you would for any other job, for example to request 4 processors with 4GB memory for 2 hours:

```bash
$ qsub -I -X -l nodes=1:ppn=4 -l mem=4GB -l walltime=2:00:00
```

By default your job will be placed in the s48 queue for serial jobs. For higher priority (thus less time waiting in the queue for the session to start) or to access more than one node, submit the job to the interactive queue with `-q interactive`.

**Your interactive session will start in your home directory**, you can jump to the directory you submitted from with:

```bash
$ cd $PBS_O_WORKDIR
```