Using compute nodes interactively at Prince

<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC Home</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Getting started on Prince</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</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
<tr>
<td>Brooklyn (OpenStack)</td>
</tr>
<tr>
<td>Dalmata (NYU Abu Dhabi)</td>
</tr>
</tbody>
</table>
Transferri
ng data
to/from
the
clusters
Transferri
ng data
to/from
Prince
cluster
using
Globus
Submittin
g jobs
with
sbatch
Available
software
Licensed
Software
Available
on the
HPC
Cluster
Building
Software
Slurm
Tutorial
Tutorials
FAQs
Scratch
Area
Cleanup
Programming
for
Biologists
Acknowledgemen
t
Research
Gallery
HPC
People
Running jobs on the Prince Cluster

Accessing the Prince Cluster
  From Windows workstation
  From Mac workstation

Software and Environment Module

Job script and resource request
  Introduction to job scheduling
  Submitting jobs with sbatch
  Requesting resources
  Using computing nodes interactively

Monitoring batch jobs
  Monitoring batch jobs - squeue
Some things require user interaction - debugging, and many uses of Matlab for example. And sometimes when preparing a job script it is helpful to work through it interactively first. However, the login nodes are not suitable for work requiring large amount of memory or computation. To support interactive use in a batch environment, Slurm allows for interactive batch jobs.

Options for running interactively on the compute nodes:

(Don't just submit the job, but also wait for it to start and connect stdout, stderr and stdin to the current terminal)

- \texttt{-n num}
  Specify the number of tasks to run, e.g. \texttt{-n4}. Default is one CPU core per task.
- \texttt{-t time}
  Request job running duration, e.g. \texttt{-t1:30:00}
- \texttt{--mem=MB}
  Specify the real memory required per node in MegaBytes, e.g. \texttt{--mem=4000}
- \texttt{--pty}
  Execute the first task in pseudo terminal mode, e.g. \texttt{--pty /bin/bash}, to start a bash command shell
- \texttt{--x11}
  Enable X forwarding, so programs using a GUI can be used during the session (provided you have X forwarding to your workstation set up)

To leave an interactive batch session, type \texttt{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 \texttt{ssh}. To end the session, type \texttt{'exit'} again just like the process of logging in and out with \texttt{ssh}.

```
[wds35@log-0 ~]$ srun --pty /bin/bash
[wd35@c17-01 ~]$ hostname
  c17-01
```

To use any GUI-based program within the interactive batch session you will need to extend X forwarding with the \texttt{--x11} option. This of course still relies on you having X forwarding at your login session - try running \texttt{'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:

```
$ srun -n4 -t2:00:00 --mem=4000 --pty /bin/bash
```
If you do not request resources you will get the default settings. If after some directory navigation in your interactive session, you can jump back to the directory you submitted from with:

```
$ cd $SLURM_SUBMIT_DIR
```

**Exercise**

Start an interactive batch session with the \(--x11\) option for X forwarding. NYU has a site license for Matlab, so find and load the Matlab module, and start the GUI. Tinker with it, if you wish.

(Don't forget to quit Matlab and exit the interactive batch session after you're done)