Getting started on Prince

- Logging in to Prince
- Adding Prince to your SSH Tunnel
- Filesystems on Prince
- Transferring files to Prince
- How to migrate from another cluster to Prince
- Compiling Code on Prince
- Submitting batch jobs on Prince
- Interactive sessions
  - GPU Jobs
- Software on Prince

Logging in to Prince

Logging in to Prince is the same 2-stage process as logging into any other NYU HPC cluster. The host name is 'prince.hpc.nyu.edu'. The HPC cluster Prince is not directly visible to the internet (outside the NYU Network). If you are outside NYU's Network (off-campus) you must first login to a \textit{bastion host} named \textit{hpc2.nyu.edu}. The diagram below illustrates the login path.

\textbf{NOTE:} The clusters can still access the internet directly. This may be useful when copying data from servers outside the NYU Network - see: How to copy files to and from the HPC clusters.

\textbf{NOTE:} Alternatively, instead of login to the \textit{bastion host}, you can use \textit{VPN} to get inside NYU's network and access the HPC clusters directly. Instructions on how to install and use the VPN client are available here.

\textbf{NOTE:} You can't do anything on the \textit{bastion host}, except \texttt{ssh} to the HPC clusters.

Adding Prince to your SSH Tunnel

On a Mac:
# first we create the tunnel, with instructions to pass incoming
# packets on ports 8020, 8021 and 8022 through it and to specific
# locations
Host hpctunnel
HostName hpc2.nyu.edu
User NetID

# Add a line like this: if you are already using 8023, choose another number:
LocalForward 8023 prince.hpc.nyu.edu:22

# next we create an alias for incoming packets on port 8023. The
# alias corresponds to where the tunnel forwards these packets
# Add an entry like this. The port number must match the one used above:
Host prince
HostName localhost
Port 8023
ForwardX11 yes
User NetID

For Windows, the text you must add is indicated in bold on this page.

Filesystems on Prince

The table below shows the File Systems available on the Prince Cluster.

File systems listed in green font will be available in the future.

<table>
<thead>
<tr>
<th>Mountpoint</th>
<th>Storage Capacity</th>
<th>FS Type</th>
<th>Backed up?</th>
<th>Flushed?</th>
<th>Availability</th>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home</td>
<td>43 TB</td>
<td>ZFS</td>
<td>Yes</td>
<td>No</td>
<td>All Prince nodes (login, compute)</td>
<td>$HOME</td>
<td>/home/$USER</td>
</tr>
<tr>
<td></td>
<td>(20 GB / user)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/scratch</td>
<td>1.1 PB</td>
<td>Lustre</td>
<td>NO</td>
<td>YES</td>
<td>All Prince nodes (login, compute)</td>
<td>$SCRATCH</td>
<td>/scratch/$USER</td>
</tr>
<tr>
<td></td>
<td>(5 TB / user)</td>
<td></td>
<td></td>
<td>Files unused for 60 days are deleted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/beegfs</td>
<td>500TB</td>
<td>BeeGFS</td>
<td>NO</td>
<td>TBD</td>
<td>All Prince nodes (login, compute)</td>
<td>$BEEGFS</td>
<td>/beegfs/$USER</td>
</tr>
<tr>
<td></td>
<td>(2 TB / user)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/archive</td>
<td>700 TB</td>
<td>ZFS</td>
<td>Yes</td>
<td>No</td>
<td>Only the login nodes</td>
<td>$ARCHIVE</td>
<td>/archive/$USER</td>
</tr>
<tr>
<td></td>
<td>(2 TB / user)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/state</td>
<td>Varies, mostly</td>
<td>ext3</td>
<td>NO</td>
<td>YES</td>
<td>Separate local filesystem on each compute node</td>
<td>$SLURM_JOBID</td>
<td>/state/partition1</td>
</tr>
<tr>
<td>partition1</td>
<td>&gt;100GB</td>
<td></td>
<td></td>
<td>at the end of each job</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Transferring files to Prince

Your files on /scratch are visible on Prince, and will remain such.

Moving many small files takes significantly longer than moving one large file. Therefore, to transfer source code with many small files from /home on another cluster, we recommend tarring up the source directory first.

We recommend rebuilding all models on Prince - this will ensure the best performance and also avoid any problems relating to mismatched OS and library versions between the clusters. Therefore, there is no need to transfer object files or executables.

How to migrate from another cluster to Prince

If you have a workflow on a previous cluster, and you wish to replicate it on Prince:

1. Ensure the files you need are available on Prince
   Files on $SCRATCH are already available on Prince - there is nothing you need to do with these.
2. You will need to rewrite your scripts to convert them from using PBS to using Slurm Job Scheduler. If you are having any questions or issues, contact us for help.

3. Check your `.bashrc` and `.bash_profile` files. If on the older cluster, you loaded certain modules in them, you may wish to add corresponding lines to those files on Prince. You may also wish to replicate your shell aliases.

4. If you are running a model you compiled yourself, you will probably get improved performance by recompiling the model on Prince. See below for more about compiling code.

## Compiling Code on Prince

We recommend using the Intel compiler suite (module load intel), and, for MPI programs, OpenMPI (module load openmpi/intel).

The GNU compiler suite is also available (module load gcc) and over time other compiler versions and MPI implementations will be installed. There will be instances in which an alternative compiler or MPI library performs better or avoids a bug in the recommended compilers, but in the long run the functionality and performance of compilers and MPI libraries is very similar, so for easiest maintenance, only use the alternatives if you really need to.

If you are building software to use for many or long-running jobs, testing for performance with a few MPI libraries and compile options can be very valuable since a specific model might run noticeably faster under one MPI implementation than another.

## Submitting batch jobs on Prince

Prince cluster uses Slurm Job Scheduler, a resource manager designed to allocate compute resources and schedule jobs. Due to the nature of HPC clusters we recommend writing a batch script, rather than running interactive commands.

There are two aspects to a batch job script:

- A set of SBATCH directives describing the resources required and other information about the job
- The script itself, comprised of commands to setup and perform the computations without additional user interaction

Jobs are submitted with the `sbatch` command:

```
$ sbatch options job-script
```

The options tell SLURM information about the job, such as what resources will be needed. These can be specified in the job-script as SBATCH 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 SBATCH directive with the syntax: `#SBATCH option` You can find a list of sbatch options [here](#).

## Interactive sessions

To start an interactive batch session, use:

```
$ srun --pty /bin/bash
```

If you need a GUI, you should use:

```
$ srun --x11 --pty /bin/bash
```

By default single CPU core and 2GB memory are allocated to your job for 1 hour. You can explicitly request more CPUs and more memory with the following options:

```
$ srun -n4 -t2:00:00 --mem=4000 --pty /bin/bash
```

The above command requests 4 compute nodes for 2 hours with 4 Gb of memory.
Most nodes have either 64GB, 128GB, 192GB or 256GB of memory. However, some amount of memory is also needed by the operating system, so nodes have about 62GB, 124GB, 189GB and 250GB available for jobs. If you request 128GB, you restrict the job to the subset of nodes having more than 128GB of memory, and it will probably take longer to schedule.

**GPU Jobs**

In Prince cluster, there are currently 9 GPU nodes. Each of the nodes is equipped with 4 Tesla K80 cards.

<table>
<thead>
<tr>
<th>NVIDIA-SMI 367.48</th>
<th>Driver Version: 367.48</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>GPU Name</td>
<td>Persistence-M</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>0 Tesla K80</td>
<td>On</td>
</tr>
<tr>
<td>N/A 22C</td>
<td>26W / 149W</td>
</tr>
<tr>
<td>1 Tesla K80</td>
<td>On</td>
</tr>
<tr>
<td>N/A 28C</td>
<td>30W / 149W</td>
</tr>
<tr>
<td>2 Tesla K80</td>
<td>On</td>
</tr>
<tr>
<td>N/A 20C</td>
<td>26W / 149W</td>
</tr>
<tr>
<td>3 Tesla K80</td>
<td>On</td>
</tr>
<tr>
<td>N/A 29C</td>
<td>30W / 149W</td>
</tr>
</tbody>
</table>

To request one GPU card, use the Slurm directive:

```
#SBATCH --partition=gpu
#SBATCH --gres=gpu:1
```

**Software on Prince**

Prince uses the same Environment Modules setup as previous clusters. Please check the availability of the module you need with:

```
$ module avail
```

Different versions of software may give slightly different results, especially if your model is numerically sensitive! Please carefully check the results of simulations on Prince and compare them to results on the cluster you used previously!