

Hopper Cluster

Hopper is our group's high-performance computing cluster, which we share with the Reed, Steinschneider, and Anderson groups.

Zach Brodeur wrote [a user guide for Hopper](#) and [a more detailed guide](#) which you should read.

Access

To get access to Hopper, message Vivek (this should be part of your onboarding process).

You can access Hopper for file transfers through [Globus](#) (see the Cornell CAC wiki for information on [how to use Globus](#)).

Hardware

- 22 compute nodes (c0001-c0022) with dual 20-core Intel Xeon Gold 5218R CPUs @ 2.1 GHz, 192 GB of RAM
- 225 TB of total usable storage. This should be a healthy amount per person. We'll figure out a more formal way to partition storage resources over time.
- Hyperthreading is available on all nodes and enabled by default (so you can treat each physical core as two logical cores, if the software allows; in other words, think of each node as consisting of 80 CPUs). If you don't want to use hyperthreading, including the following in your submission script: `#SBATCH --hint=nomultithread`

Software

Filesystem

Use your home directory (`~`) for data you want to keep. These directories are not backed up by default.

Scheduler

Hopper uses the [SLURM scheduler](#). There is only one queue (`normal`) with no resource or time limits.

Modules

- To see the list of available modules, use `module avail`.
- To see loaded modules, use `module list`.

Usage Guidelines

When To Use Hopper

If you aren't going to be using a lot of cores (10+ cores), it might be better to use your workstation, especially if Hopper's resources are highly in demand when you want to run your job. Hopper's cores are only 2.1 GHz (compare that to 3-4 GHz on your workstation), so you're only going to get an advantage if you can exploit parallelization. This is great for things like Monte Carlo simulations, multi-objective optimization, or really large-scale modeling experiments where you can have each core run a different member of the ensemble. If you're doing model calibration, think carefully about whether your method can actually use that many cores. Most MCMC procedures, for example, cannot, but Approximate Bayesian Computation, pre-calibration or particle filtering can be embarrassingly parallel.

Run Jobs using SLURM

Do not run jobs from the command line in your home directory.* Either set up a SLURM script and run it or

Benchmark Your Code

Always benchmark your parallel code on your workstation to get a sense for how it scales (though note that your workstation has faster cores than Hopper does!). You don't want to just give SLURM an unrealistic amount of time, since we need to coordinate Hopper as a shared resource. Dave Gold has a good [post](#) about this at Water Programming. Dave also has a good post about how to make a plan and organize an HPC experiment [here](#). In general, Water Programming has a lot of good resources about [HPC](#) and [SLURM](#), among other topics.

Other Tips

- If you are going to request a large fraction of Hopper's resources (more than 5 nodes), make sure that you give the other lab groups plenty of notice so they can coordinate their use.

- Right now, the only filesystem is the user home directories, `~`. This is not a parallel file system, so file I/O with lots of files might slow things down a bit. Try to read files once and minimize file I/O as much as possible.
- The default shell is `bash`. If you want to use a different shell, contact CAC using the instructions on the Wiki page.
- I recommend using the following instructions to colorize `git` output for easier reading:

Colorize Git

```
1 git config --global color.ui auto
2 git config --global color.branch auto
3 git config --global color.status auto
```

You can also modify your `~/.gitconfig` to customize the colors, like in this example from [StackOverflow](#):

Modify .gitconfig for Color

```
1 [color "status"]
2   added = green
3   changed = red bold
4   untracked = magenta bold
5
6 [color "branch"]
7   remote = yellow
```

- Make sure you set up your [SSH key](#) for Github. This isn't essential (you can use https instead), but this is a bit more secure.

More Information

For more information, see the [Cornell CAC Wiki entry](#).

Last update: June 26, 2023