Office of Research Computing, George Mason University

Hopper Quick Start Guide

January 08, 2025

This document provides a brief introduction to High-Performance Computing (HPC) at Mason. For a more in-depth guide please see our wiki page at https://wiki.orc.gmu.edu/. This guide focuses primarily on the Hopper cluster.

1 WHAT IS A COMPUTER CLUSTER, AND WHAT DOES IT DO?

The High-Performance Computing (HPC) clusters operated by the GMU Office of Research Computing (ORC) are each comprised of multiple computers interconnected via a network. Conceptually, a cluster functions similarly to a personal laptop or desktop computer. Much like a personal computer, a cluster can run programs, send and receive information, and manage data input and output (I/O). However, each computer node in a cluster typically possesses more powerful processors, greater memory capacity, and increased data storage compared to a personal computer. Additionally, when using software designed for parallel computing, a cluster can use many computer nodes simultaneously to run a single program. This enables a cluster to run cases which might require months of processing if run on a single computer, in a fraction of that time.

The Hopper cluster consists of two types of nodes corresponding to two different CPU architectures: Intel and AMD. The cluster has CPU and GPU nodes available. Each GPU node has multiple GPU's and CPU's. Jobs are run through a task manager and can be launched from cluster login nodes. There are 4 login nodes available to access the cluster.

2 BASIC CLUSTER OPERATIONS

The cluster may be accessed using either a command line interface or the Open OnDemand web interface.

2.1 Logging in to the Hopper cluster using a terminal

The Hopper cluster is Linux-based and accessed largely via command line from a 'terminal'. Access to Hopper cluster is done via SSH (Secure Shell) from a terminal. The procedure of accessing the cluster via a terminal is the same on Windows, Linux, or Mac. Once logged in, you are assigned to a 'login node', or a 'head node'. The terms – 'login node' and 'head node' are used interchangeably. In the rest of this document, we will use the term 'login node'. The login node allows the user to manage files in their home directory, load modules, compile and test code, and submit and manage jobs.

To log into the Hopper cluster, start a terminal session on your local computer. Then use ssh (secure shell) to log in by entering:

```
ssh netID@hopper.orc.gmu.edu
```

Where 'netID' is your GMU netID. If your GMU email address is jsmith@gmu.edu, then your netID is "jsmith".

If a login node with a specific architecture (AMD or intel) is necessary, the type of login node can be specified as follows:

To log in to an AMD node

```
ssh netID@hopper-amd.orc.gmu.edu
```

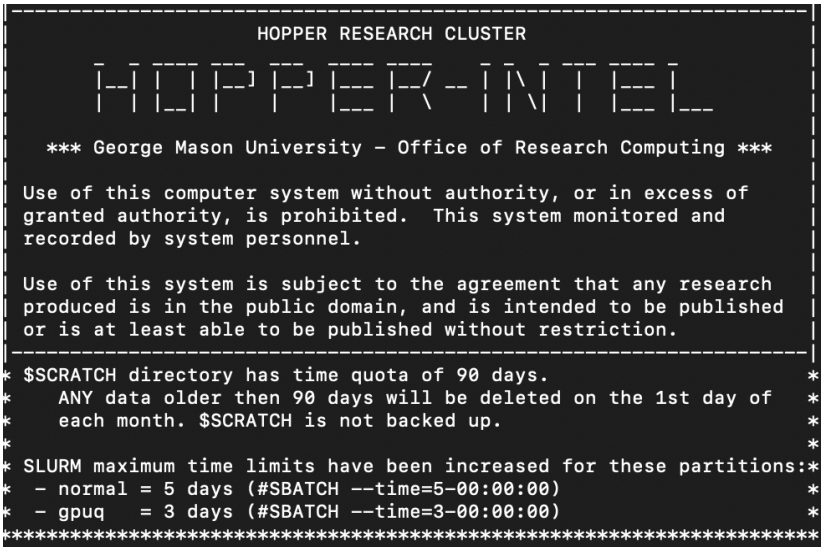
To log in to an intel node

```
ssh netID@hopper-intel.orc.gmu.edu
```

NOTE: If you use “hopper.orc.gmu.edu”, you will be put on any of the four login nodes, while “hopper-intel.orc.gmu.edu” will put you on either of the 2 Intel login nodes which are named “hopper1” and “hopper2” and the “hopper-amd.orc.gmu.edu” login nodes will put you on either of the 2 AMD nodes which are named “hop-amd-1” and “hop-amd-2”. The “hopper-intel” login nodes do not explicitly use the word ‘intel’ in the command line prompt of the terminal session. Please pay attention to the screen prompt you get when you log in, as it will tell you whether you are on an Intel node or an AMD node. The name of the login node is shown after you login. This information is helpful if you have to file a help ticket and have to specify the login node name.

Here is an example of what you will see upon successful logging into the “hopper-intel”:

```
$ ssh jsmith@hopper.orc.gmu.edu
jsmith@hopper-intel.orc.gmu.edu's password:
Last login: Tue Oct 11 07:15:56 2022 from 72.205.52.16
```




```
[jsmith@hopper-1 ~]$
```

Here is an example of what you will see upon successful logging into the “hopper-amd”:

```
$ ssh jsmith@hopper.orc.gmu.edu
jsmith@hopper-amd.orc.gmu.edu's password:
Last login: Tue Oct 11 07:15:56 2022 from 72.205.52.16
```

```
|---------------------------------------------------------------------|
|                     HOPPER RESEARCH CLUSTER                         |
|          _  _ ____ ___  ___  ____ ____    ____ _  _ ___            |
|          |__| |  | |__] |__] |___ |__/ __ |__| |\/| |  \           |
|          |  | |__| |    |    |___ |  \    |  | |  | |__/           |
|                                                                     |
|   *** George Mason University - Office of Research Computing ***    |
|                                                                     |
| Use of this computer system without authority, or in excess of      |
| granted authority, is prohibited.  This system monitored and        |
| recorded by system personnel.                                       |
|                                                                     |
| Use of this system is subject to the agreement that any research    |
| produced is in the public domain, and is intended to be published   |
| or is at least able to be published without restriction.            |
|---------------------------------------------------------------------|
* $SCRATCH directory has time quota of 90  days.                      *
*    ANY data older then 90  days will be deleted on the 1st day of  *
*    each month. $SCRATCH is not backed up.                          *
*                                                                    *
* SLURM maximum time limits have been increased for these partitions:*
*  - normal = 5 days (#SBATCH --time=5-00:00:00)                     *
*  - gpuq   = 3 days (#SBATCH --time=3-00:00:00)                     *
**********************************************************************
```

```
[jsmith@hop-amd-1 ~]$
```

2.2 Using Open OnDemand

The Open OnDemand (OOD) Server on Hopper is a web portal accessible through any browser. It allows users to access cluster resources through a graphical user interface (GUI). Users can view, edit, upload and download files, create, edit, submit and check jobs, and run GUI applications (such as MATLAB, Mathematica, Jupyter Lab, etc.). It also allows for command line shell access. Interactive sessions using Open OnDemand have a maximum length of 172 hours. If runtimes longer than 172 hours are required, the program must be run using the Slurm workload manager.

Currently the following GUI applications are available via Open OnDemand:

- Hopper Desktop – an X11 login session.
- MATLAB
- Mathematica
- HOPPER Shell Access
- Jupyter Lab
- RStudio
- Hopper GPU Desktop
- Job Composer
- QuantumATK
- SAS
- STATA
- VSCode Code-Server

If you have any specific GUI applications you would like to be able to access via Open OnDemand, please contact the ORC helpdesk. More details about the OnDemand service can be found on the Open OnDemand Wiki page.

To access the OOD web server directly, go to https://ondemand.orc.gmu.edu. You will have to authenticate the credentials with your GMU NetID and Password. Please note that you will not be able to submit jobs before attending the new users' tutorial.

3 DATA TRANSFER AND STORAGE

Once you have logged into the cluster, your home directory is your current directory. The path of your home directory is /home/<userID> (referred as $HOME in the rest of the document). The home directory has a limit of 60GB. Files in the home directory are backed up. A much larger amount of disk space is available in the /scratch/<userID> directory (referred as $SCRATCH in the rest of the document), which is created when you log in for the first time. However, storage on $SCRATCH is temporary, and files older than 90 days are deleted on the first of each month. Files in $SCRATCH are not backed up.

For long-term storage needs and/or for large files, faculty members can request up to 1TB of storage in the /projects space for use by themselves and their group members. Group members are limited to faculty, doctoral students, and postdocs. More information can be found on the ORC wiki at Storage_on_the_Cluster. For storage needs beyond 1TB, faculty and departments can purchase storage from the ORC at an annual cost. The cost for storage is currently $60/TB/year.

To transfer data, files in $SCRATCH or $HOME directory can be copied elsewhere using SCP (secure copy protocol). The process is the same from a terminal on Mac/Linux/Windows10.

The format for uploading a file from your laptop/workstation to the cluster is:

```
$ scp <local_file_path> netID@hopper.orc.gmu.edu:~/<cluster_folder_path>
```

For downloading, you need to switch the source and destination paths. You can use the "-r" option to copy an entire directory. To avoid entering your password each time, you can set up password-less SSH login to the cluster. Details can be found in the ORC wiki at the Uploading Data.

3.1 Globus

GMU has a Globus site subscription which is a research data management service. Globus provides tools and an easy-to-use interface to facilitate parallel, load-balanced, fault-tolerant data transfer of large data securely. The Globus connect portal can be used to transfer data across different clusters such as local cluster Hopper, national supercomputers, NCAR, etc. It can also be used to transfer data between a personal desktop/laptop and Hopper cluster. More information about these tools appears in the ORC wiki under Globus.

4 ENVIRONMENT MANAGEMENT

Unlike laptops/workstations, the cluster can have many versions of the same software – for example, Python3.6, Python3.7, Python3.8. Additionally, software packages can be compiled using different compilers – like GCC, Intel Compiler etc. Modules are useful in managing different versions of applications. We use "modules" software to ensure that your environment variables like, $PATH, $LD_LIBRARY_PATH etc., are set up for the software you want to use. The modules system is a tool that allows you to control your access to the different pieces of software that are currently installed on the cluster. It can be used to dynamically change your environment (i.e., add/remove directories in the $PATH, $LD_LIBRARY_PATH etc.) without having to log out and log back into the cluster to change environments.

The Hopper cluster uses an environment module system called “LMod” (a Lua based module system) which uses a hierarchical model which displays only modules that are compatible with the compiler and MPI library you have loaded at a given time to avoid incompatibilities. More details can be found here.

You must load the module for the software or library before using it inside your application. Some commonly used commands allow you to perform basic functions involving modules. module – provides usage options available with module commands

- module avail – query what modules are available
- module list – list what modules are already loaded in your workspace
- module load <module_name> – upload a module that is available.

To get a feel for the usage of modules in your own programs, a few commands are given below:

```
[jsmith@hop-amd-1 ~]$ module
[jsmith@hop-amd-1 ~]$ module avail
```

The output of the above two commands is quite large, hence is not shown here; you can try it yourself on your terminal session. Below is an example of using the “module list” command before and after using the load command to upload the module openblas:

```
[jsmith@hop-amd-1 ~]$ module list
Currently Loaded Modules:
  1) use.own     4) gnu9/9.3.0          7) hwloc/2.1.0
  2) autotools   5) ucx/1.12.1-4g       8) openmpi4/4.0.4
  3) prun/2.0    6) libfabric/1.10.1    9) hosts/hopper
[jsmith@hop-amd-1 ~]$ module load openblas
[jsmith@hop-amd-1 ~]$ module list
Currently Loaded Modules:
  1) use.own      5) ucx/1.8.0           9) hosts/hopper
  2) autotools    6) libfabric/1.10.1   10) openblas/0.3.13-sp
  3) prun/2.0     7) hwloc/2.1.0
  4) gnu9/9.3.0   8) openmpi4/4.0.4
```

5 PARTITIONS

Partitions are queues on a cluster to which jobs are submitted. A simple way to get basic information on the available partitions is to use the command 'sinfo' in a terminal session to get basic information about node and partition configurations on the cluster as shown below:

Partition	Default Time Limit (Days-Hours:Min)	Description
normal	5-00:00	access to cpu nodes
bigmem	5-00:00	access to large memory nodes
contrib	5-00:00	users can submit jobs to this partition however they will be subject to preemption
gpuq	3-00:00	gpu node access
gpuq-contrib	3-00:00	similar to the CPU contrib, but for GPU nodes
interactive	0-12:00	default queue, interactive jobs e.g Open OnDemand

6 RUNNING JOBS

The Hopper and Argo clusters use the Slurm (Simple Linux Utility for Resource Management) workload manager to manage the cluster resources like partitions, number and types of nodes, CPU cores required, number of GPU needed, runtime limit, memory required etc. Job submission script will specify the resources needed which is used by Slurm to determine the cluster resources available for the submitted jobs and use cluster resources efficiently for multiple jobs.

To run jobs in batch mode on the cluster, the "sbatch" command is used to submit jobs from a login node. The various sbatch options for the batch job, should be put inside a single bash script along with the program to be executed and passed to a Slurm queue as shown below:

```
$ sbatch <name_script>.slurm
```

Some of the more commonly used Slurm commands are: sbatch, squeue, scancel, sacct.

- sbatch Submit a job to a partition
- squeue Check job status
- scancel Delete jobs
- sacct Check status of jobs

More details on usage with advanced job options such as running parallel jobs (MPI, Pthreads, etc.), example scripts that show how to set common parameters for a batch job, and more can be found on the ORC wiki at the Getting_Started_with_SLURM page.

6.1 Runtime Options:

Running a job with SLURM can be done either via command-line or through a script file. A template script to serve as a model for writing scripts can be found here: Slurm Template

In addition to running a 'standard' single-core job similar to a job on a laptop/desktop, there are other modes of operation that are available on a cluster. Some of these are as follows:

6.2 Running Array Jobs:

It is possible to run multiple instances of a job using an Array Job. This might be useful for doing a parameter sweep, where you run the same program or script with small variations between each run. The basic form of the job array command is as follows:

```
#SBATCH --array= <start>-<end>
```

More information to run array jobs can be found at the ORC wiki under running-job-arrays link.

6.3 Running Parallel Jobs:

This job type is used to execute several loads or jobs simultaneously. The 3 types of multi-threaded or multi-processor jobs that can be run are:

- **MPI Jobs**: MPI stands for Message Passing Interface. The MPI specification is an API that provides useful routines for communication between the nodes. For more information on key tasks like compiling your program, linking against a dynamic library and running your MPI job with "sbatch" refer to the Running an MPI job.
- **Threaded Jobs**: A thread job is a type of background job that runs a command or expression in a separate thread within the current session process. Threading can be used in conjunction with MPI in Slurm. In this case both "--ntasks-per-node" and "--cpus-per-task" must be specified simultaneously. To know more about this, please refer to the Hybrid Parallelism.
- **GPU/CUDA Jobs**: GPUs are treated as generic resources in Slurm. To run a CUDA or GPU based application you have to first allocate the desired number of GPUs to your job using the "--gres=gpu:" parameter and request the GPU partition. To know more about these tasks, refer to the GPU Parallelism.

7 SUMMARY

This is a startup guide. For detailed information, the main web page of the ORC clusters is: https://wiki.orc.gmu.edu/. This page is updated frequently with information about the HPC clusters, and we recommend that you refer to it often as you continue to use HPC for your projects, research, and tasks.

8 GETTING HELP

If you cannot find the answers you are looking for on the main web page, you can send email to the support staff at orchelp@gmu.edu.