FIU IRCC HPC QuickStart Guide

The purpose of this document is to provide members of the FIU community with the basic tools necessary to access the High-Performance Computing (HPC) resources at the Instructional and Research Computing Center (IRCC). This guide is designed as a starting point for cluster use and is intended to act as a reference to supplement the on-boarding training required for all users.

Obtaining a FIU HPC account

In order to have an account created on the HPC, please e-mail hpcadmin@fiu.edu. Students must be associated with a faculty member and have their approval for joining their group on the HPC.

Login and File Transfer

The HPC is located behind the protection of the FIU campus firewall. If you are not connecting from a secure campus network, you will need to be connected to the FIU network via the FIU VPN in order to access the HPC. For more details, see https://network.fiu.edu/vpn/.

Users typically access the cluster via command-line interface, using our login nodes to connect. This requires a UNIX-like terminal (such as the Terminal app included with MacOS) or a secure shell client (such as PuTTY in Windows).

To log in to hpclogin01 from a UNIX-like terminal, input the following command, where “username” is your FIU username, enter the following in terminal:

```
$ ssh username@hpclogin01.fiu.edu
```

File transfer to the cluster can be accomplished via a secure copy client (such as WinSCP in Windows) or via the secure copy (scp) command in a UNIX-like terminal. To copy a file from your local disk to your home directory on the cluster using a UNIX terminal, issue the following command:

```
$ scp path_to_file username@hpclogin01.fiu.edu:~
```

Note that when logging in, it does not matter whether you are connected to and uploading data to hpclogin01 or hpclogin02, since your home directory on the cluster exists in a logical volume on a parallel file system. This directory is mounted on both login nodes and every compute node, any file you copy using the above command will be visible to every computer in the cluster. As an alternate to command line transfer using secure copy, you may also use SFTP either via command line or GUI-driven client software such as such as WinSCP or CyberDuck. FTP access is disabled on the HPC for security.

In addition to the storage provided for users within their home directories (up to a maximum of 100 GB/user and 300 GB/group), additional storage is available for research use. The HPC has 47 TB of scratch space consisting of high-speed solid-state drives. This is intended for temporary storage and should also be utilized by jobs that require extensive input and output operations. Each lab group has a folder under the PI's username, and users have read/write permissions within their respective lab folder.

Loading Software as Modules

Hundreds of software applications are pre-installed on the HPC and available as modules, which allow users to readily utilize these tools. In the context of SLURM, modules are indispensable tools that grant access to various software packages, libraries, and environments. Any software beyond the basic operating system is incorporated through the module utility.

You may browse all installed modules by running the following command on the login node:

```
$ module avail
```

Note that you can tailor the list by entering the beginning of the module name you are seeking, so that you do not have to browse all installed modules. For instance, if you wanted to load a particular version of miniconda, you can type

```
$ module avail miniconda
```

and all versions installed as modules will be listed:

```
-------------------------------------------------
/home/share/Modules/4.1.3/modulefiles/linux-centos7-x86_64 ------------
-----------------------------------
```

```
miniconda2-4.5.11-gcc-8.2.0-aavxm3w  miniconda2.7-4.8.3-testing
miniconda3-4.5.11-gcc-8.2.0-oqs2mbg  miniconda3-23.5.2
```

You can then load the appropriate module as follows:

```
$ module load miniconda3-4.5.11-gcc-8.2.0-oqs2mbg
```

This temporarily modifies your PATH variable for the current session, a list of directories that the shell searches through when you enter a command. By loading a module, you temporarily update the PATH to include the directories containing the executables associated with the chosen software package. This allows you to run the commands provided by the loaded module without needing to type out the full path to the executable.

To view the modules you've loaded in your current session, use the command

```
$ module list
```

Unloading modules is accomplished with:

```
$ module load miniconda3-4.5.11-gcc-8.2.0-oqs2mbg
```

Unloading modules may be necessary when switching versions of a program, for instance, if you want to switch to a different version of Miniconda. Note that modules are automatically unloaded upon session logout. Therefore, each user needs to load the desired software in every session. For a more streamlined experience, you can add commonly-used module commands to your ~/.bashrc file.

Users have a number of options for the installation and execution of software tools for their research, including compiling their own tools from source using, the use of Conda environments to manage packages and dependencies, as well as using Singularity container. Keep in mind that dependencies and basic utilities like make and gcc also require loading as modules before use.

Job Submission using the Slurm Workload Manager

In order to efficiently balance the load from all of our users, the HPC utilizes a resource management system called Slurm. Slurm is a highly configurable workload manager commonly used in high-performance computing (HPC) clusters. (Full documentation for

slurm is available here: https://slurm.schedmd.com/documentation.html) It efficiently manages and schedules jobs and tasks across a cluster of computers, helping users and administrators allocate and utilize computing resources effectively. Slurm provides a command-line interface and robust job scheduling capabilities, making it an essential tool for managing large-scale computational workloads. (Full documentation for Slurm is available here: https://slurm.schedmd.com/documentation.html.)

A Slurm submission script serves as a blueprint for users to prepare and submit jobs for execution on a cluster. It is essentially a BASH script with specific syntax that Slurm interprets to understand and manage the job's resource requirements and priority. An example script follows:

```
#!/bin/bash
#SBATCH --account iacc_jdoe
# Node location
#SBATCH -p IB_40C_1.5T
# Number of nodes
#SBATCH -N 1
# Number of tasks
#SBATCH -n 16
#SBATCH --mail-type=ALL
#SBATCH --output=%x.%j.log

############################################################
############ Setup environment ############################
############################################################

##module load

module load spades-3.12.0-gcc-8.2.0-e65rkhy

############################################################
############ Executable commands ##########################
############################################################

spades.py -1 /home/data/path/to/R_1.fq.gz -2
/home/data/path/to/R_2.fq.gz -o /home/data/output/Assembly
```

The first block of this text file, the header, specifies key job attributes for a Slurm submission script. In this particular example, it assigns the job to the "iacc_jdoe" account, requests one node from the "IB_40C_1.5T" partition, allocates 16 tasks, sets up email notifications for all job events, and designates the output file name. These parameters should be modified to meet your compute requirements and available resources on the HPC.

The second block in the example above sets up the environment with required software and dependencies for subsequent analyses, in this case loading a genome assembler with the **module load** function.

The third block of text contains the primary computational task of the job. This particular example runs the SPAdes genome assembly tool, specifying the input data locations as well as the desired output path.

Submit a Job: To submit a job for execution on the cluster, use the **sbatch** command followed by the name of your submission script. For example:

```
$ sbatch ExampleAssembly.sub
```

Check Job Status: To monitor the status of your submitted jobs, utilize the **squeue** command with the **--me** flag. This will display information about your jobs only:

```
$ squeue –me
```

Cancel a Job: If you need to cancel a job, you can do so using the **scancel** command followed by the job ID:

```
$ scancel <jobid>
```

Monitoring Cluster Resources and Usage

Check Storage and Slurm Usage: To review your group's storage and Slurm resource usage for the month, run the **hpcusage** command. This provides insights into resource consumption.

Check Node Information: To get information about the nodes in the cluster, use the **sinfo** command. This command displays node status, partitions, and other node-related details:

Check Storage Quota: For monitoring your individual storage usage, employ the **myquota** command. It helps you keep track of your allocated storage and usage.