Research Computing (RC) Club Training Session

Luke Gibson

Oct. 4th, 2018

Outline

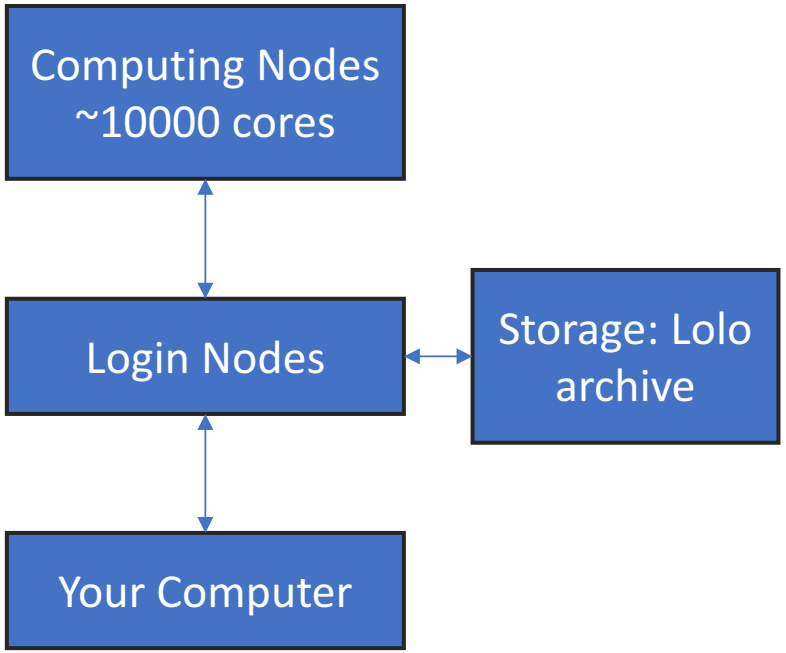
- Hyak and RCC
- Logging into Hyak
- Basic Linux Commands
- Transferring Files Between Your PC and Hyak
- Submitting Your Jobs

Hyak Overview

- ~10000 nodes in total
- STF has access to 3,600 cores
- Ikt.Hyak
  - 16 core processors
  - 64 GB memory
  - CentOS 6 Linux
- Mox.Hyak
  - 28 – 32 core processors
  - 128 – 256 GB memory
  - 10 GPU nodes (28 core processors)
  - 1 interactive GPU node
  - CentOS 7 Linux

Hyak Overview

- Node type:
  - Computing nodes
    - Production jobs
    - E.g., `n0123`
  - Login nodes
    - Job submission
    - File transfers
    - E.g., `mox1`
- **Never run jobs on login nodes!**
- Hyak wiki:
  - http://wiki.hyak.uw.edu/



RC Club

- Research Computing Club (RCC)
  - Formally known as High Performance Computing Club (HPCC)
  - http://students.washington.edu/hpcc/
- As a club member, you can have access to thousands of CPU cores on UW Hyak supercomputer.
- Become a member:
  - http://students.washington.edu/hpcc/getting-started/

Logging into Hyak

- This is essentially remote-accessing a Linux system via SSH protocol
- Linux & Mac
  - Mac: Applications > Utilities > Terminal
  - Ubuntu: Search for “Terminal” in your applications
- Windows:
  - SSH client alternatives
    - Putty (http://www.putty.org/)
    - cmder (http://cmder.net/)
    - xshell
    - GitBash
- More info here:
  - http://wiki.cac.washington.edu/display/hyakusers/Logging+In

Logging into Hyak

- To connect to Ikt.Hyak
  - `ssh -X <yourUWnetid>@ikt.hyak.uw.edu`
- To connect to Mox.Hyak
  - `ssh -X <yourUWnetid>@mox.hyak.uw.edu`
- Enter your UWID password

```
ldgibson:~ $ ssh -X ldgibson@mox.hyak.uw.edu
Password:
Enter passcode or select one of the following options:

 1. Duo Push to Android (XXX-XXX-4626)
 2. Phone call to Android (XXX-XXX-4626)

Duo passcode or option [1-2]:
```

- Confirm with DUO Mobile on your phone
  - https://itconnect.uw.edu/security/uw-netids/2fa/

Basic Linux Commands

- Show current directory: `pwd`
  - `$HOME` directory
- Show contents in current directory: `ls [options]`
- Change current directory: `cd [+ path]`
  - `cd /absolute/path/to/directory`
  - `cd relative/path/to/directory`
- Create a new folder: `mkdir folder_name`
- Remove a file/folder: `rm [options]`

Transfer Files

- Linux & Mac: SCP
  - Usage: scp [options] <source> <target>
- Windows: WinSCP
  - Download: https://winscp.net/eng/download.php
- Upload files to Hyak:

```
$ scp filename user@ikt.hyak.uw.edu:path/to/folder
```

- Download files from Hyak:

```
$ scp user@ikt.hyak.uw.edu:path/to/file.txt .
```

- Note that all these commands should be executed on the remote host (your laptop).
- More info:

  http://wiki.cac.washington.edu/display/hyakusers/Managing+your+Files#Managi

STF Workspace

- Ikt
  - `/suppscr/stf`
  - Lolo: `/lolo/archive/hyak/stf`
- Mox
  - `/gscratch/stf`
- File scrubber:
  - Any files that are older than 30 days will be scrubbed (a.k.a. deleted permanently)

Interactive Node Usage

- To get an interactive node in STF group for 2 hours:

```
srun -p stf -A stf --mem=100G --time=2:00:00 --pty bash –l
```

- To get 2 nodes for interactive use for 2 hours:

```
srun -p stf -A stf --nodes=2 --mem=100G --time=2:00:00
--pty bash –l
```

  - When the above command completes, you will have been allocated 2 nodes and you will be on one of the two nodes.

- Interactive nodes are computing nodes. They cannot transfer files between Ikt/Mox/Lolo. You ***can*** submit jobs from interactive nodes.

Submitting Batch Jobs

- sbatch:
  - `sbatch [options] command-for-running-your-job`
  - e.g., `sbatch -p stf -A stf --time=2:00:00 --mem=120G test.sh`
- Use SLURM jobscripts, which:
  - Include instructions for the scheduler
  - Set up the work environment
  - Execute your program
- More info:
  - http://wiki.cac.washington.edu/display/hyakusers/Hyak+mox+Overview

Checkpoint (`ckpt`) Queue

- Checkpoint queue lets jobs run on all nodes (even nodes that the user is not allowed to access directly).
  - Jobs can be cancelled at any time
  - Jobs will run for 4 hours max and then be resubmitted
- Why use it? Consider this hypothetical situation:
  - # of idle nodes on stf: 5
  - # of idle nodes total: 655

It requires that your code is checkpointed!

```
sbatch -p ckpt -A <my short group>-ckpt test-job.sh
```

Manage Jobs

- Check the status of your job
  - `scontrol show job <job_id>`
- Queue status
  - `squeue -p <your_group>`
  - `squeue -u <your_netid>`
  - E.g., `squeue -p stf`
- To cancel a job
  - `scancel <job_id>`
- To see all nodes in a group
  - `hyakalloc <your_group>`
  - E.g., `hyakalloc stf`
- More info:
  - http://wiki.cac.washington.edu/display/hyakusers/Hyak+mox+Overview

Transfer between Ikt and Mox

- From Ikt to Mox

```
hyakbbcp myfile user@mox1.hyak.uw.edu:/gscratch/MY_GROUP/
hyakbbcp -r mydirectory
user@mox1.hyak.uw.edu:/gscratch/MY_GROUP/
```

- From Mox to Ikt

```
hyakbbcp myfile ikt1.hyak.uw.edu:/gscratch/MY_GROUP/
hyakbbcp -r mydirectory ikt1.hyak.uw.edu:/gscratch/MY_GROUP/
```

- File scrubber
  - There is a scrubbed temporary filesystem available at /gscratch/scrubbed. Files can be removed at any time, but they will be removed on a periodic basis based on creation date (files older than 30 days will be scrubbed).

Available partitions to STF on Mox

- Regular compute nodes
  - `sbatch -p stf -A stf …`
- Batch GPU nodes
  - `sbatch -p stf-gpu -A stf …`
- Interactive GPU node
  - `srun -p stf-int-gpu -A stf …`

- Build nodes (can connect to internet)
  - `srun -p build …` (any allocation works)