Intro to HPC and Parallel Computing

Derrick Zwickl
UA Research Technologies

formerly
Computational Evolutionary Biologist

zwickl@arizona.edu
Intro to HPC*

*HPC – high-performance computing

• Why use HPC?
• Anatomy of the HPC cluster
• Basics of working with the Linux shell
• Submitting jobs
Why HPC?

Research is easy!

It’s still running...
Why HPC?

Problems

• Computation takes too long
• Computation is too big
• Too many computations
Why HPC?

Problems

• Computation takes too long
  ➔ Get a more powerful computer

• Computation is too big
  ➔ Link multiple computers

• Too many computations
  ➔ Use a separate one for each job
Why HPC?

• Modern instrument for High-Performance Computing is a cluster, consisting of lots of connected individual computers (nodes).

• Supercomputer is a commonly used nickname.
Why HPC?
Why HPC?
Why HPC?

Laptop: Personal

Supercomputer: Shared
Why HPC?

Laptop : Local

Supercomputer : Remote
Laptop : Interactive

Supercomputer : Batch

Why HPC?
Getting Access to UA HPC

Faculty applies for PI HPC account

User creates HPC account using NetID

User is sponsored by PI

Run analyses using resources allotted to PI

Instructions for PIs and sponsored researchers are here:
https://public.confluence.arizona.edu/display/UAHPC/Account+Creation
Ocelote
The diagram of the UA HPC System

Compute nodes Belong to one of three clusters
- puma
- ocelote
- elgato

Specialized High Memory And GPU nodes also exist on each cluster
Connecting to UA HPC – 2 Methods

```
ssh netid@hpc.arizona.edu
shell
```
Connecting from the command line

Laptop $ ssh chrisreidy@hpc.arizona.edu
This is a bastion host used to access the rest of the RT/HPC environment.
Type "shell" to access the job submission hosts for all environments

[chrirsreidy@gatekeeper ~]$ shell
Last login: Mon Nov 8 20:16:14 2021 from gatekeeper.hpc.arizona.edu
***

(puma) [chrisreidy@junonia 08:35:32 ~]$ ocelote

(ocelote) [chrisreidy@wentletrap Intro_to_HPC]$ interactive

MacOS
- Includes Terminal app

Windows
- Putty
- MinGW
- Git Bash
Connecting with Web Browser

• Open ood.hpc.arizona.edu in your web browser and login with your NetID and password.
• From the “Clusters” drop-down menu “Shell Access"

• Exercise - connect to UA HPC.
Login node

(puma) [chrisreidy@wentletrap 16:03:26 ~]$ ls
The command line

Your NetID
(who are you)

Node name
(where are you)

Prompt
(what are you going to do)

Name of the current directory

Cluster your jobs will be submitted to

(puma) [chrisreidy@wentletrap 16:03:26 ~]$ ls
Login Node aka Submit Node

Login Node

Compute Nodes
Login node

• The login node is a computer intended for users to prepare and manage computations:
  • submit jobs
  • edit files
  • manage files
  • compile codes - NO
  • small-scale testing - NO

• **DO NOT** run any calculations on the login node
The command line

```
(puma) [chrisreidy@junonia 14:16:33 ~]$ whoami
chrisreidy
(puma) [chrisreidy@junonia 14:16:38 ~]$
```

For companion information on the hands-on practice: https://ua-researchcomputing-hpc.github.io/Intro-to-HPC/
The command line

(puma) [chrisreidy@junonia 14:16:38 ~]$ pwd
/home/u13/chrisreidy
(puma) [chrisreidy@junonia 14:21:41 ~]$

- ~ is a shortcut for your /home directory
Command Line

• List all the files and directories

(puma) [chrisreidy@junonia 14:21:41 ~]$ ls

• Make a directory

(puma) [chrisreidy@junonia ~]$ mkdir intro-to-hpc

• List all the files and directories again

(puma) [chrisreidy@junonia 14:21:41 ~]$ ls
Command Line

- Change directory
  
  (puma) [chrisreidy@junonia 14:41:30 ~]$ cd intro-to-hpc/
  
  (puma) [chrisreidy@junonia 14:48:36 intro-to-hpc]$

- Go back a level
  
  (puma) [chrisreidy@junonia 14:49:48 intro-to-hpc]$
  cd ..
  
  (puma) [chrisreidy@junonia 14:49:51 ~]$ $

- Change directory using absolute path
  
  (puma) [chrisreidy@junonia 14:49:51 ~]$ cd ~/intro-to-hpc/
  
  (puma) [chrisreidy@junonia 14:51:21 intro-to-hpc]$


Command Line

- Copy a file  Note the period at the end
  
  $ cp /xdisk/chrisreidy/workshops/LICENSE .

- List all the files and directories again
  
  (puma) [chrisreidy@junonia 14:21:41 ~]$ ls

- View contents of the file on the screen
  
  $ cat LICENSE
Working with a Linux shell

For more information on the Linux command line, follow the links in our documentation:

https://public.confluence.arizona.edu/display/UAHPC/Training#Training-LinuxSelfGuided
• Every user gets two default storage locations:
  
  • /home
    • the default home directory
    • 50GB
    • Not backed up

  • /groups/PI
    • shared by the PI’s group
    • 500GB
    • Not backed up
# Storage

- Command to list all the available storage options – `uquota`

```
$ uquota

<table>
<thead>
<tr>
<th>Path</th>
<th>Used</th>
<th>Soft Limit</th>
<th>Hard Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>/groups/chrisreidy</td>
<td>38.1G</td>
<td>500.0G</td>
<td>500.0G</td>
</tr>
<tr>
<td>/home</td>
<td>12.9G</td>
<td>50.0G</td>
<td>50.0G</td>
</tr>
<tr>
<td>/xdisk/chrisreidy</td>
<td>6.2T</td>
<td>9.9T</td>
<td>9.9T</td>
</tr>
</tbody>
</table>
```
Storage

- Additional Storage:
  - /xdisk/PI
  - Upon request, up to 20TB for limited duration with one renewal

- /tmp
- Every Puma node has about 1.4TB on a NVME SSD
Every PI gets an allocation of free time, and each user shares that allocation:

- **standard**
  - 70,000 cpu hours on Puma
  - 36,000 cpu hours on Ocelote
  - 7,000 cpu hours on Elgato
  - Limits on GPU nodes
  - Not preempted

- **windfall**
  - Unlimited
  - Preemptable
Transferring Files

Users

Bastion host
Login node

File transfer node

Shared data storage

Computes nodes

filexfer.hpc.arizona.edu
Transferring Files

- UA HPC uses a specific node for file transfer
  - hostname – filexfer.hpc.arizona.edu
- Command line options:
  - scp
  - sftp
  - rsync
  - Irods
  - rclone
- GUI options
  - Windows based: WinSCP
  - Cross-platform: Cyberduck
  - Globus
Transferring Files with OnDemand

- Display and manage your files
- Edit text files
- Drag and drop files to/from the file explorer

OnDemand provides an integrated, single access point for all of your HPC resources.
From the login node to compute
From the login node to compute

- How do we know if there are any available nodes?
- How do we decide who gets what and when?
- How do we ensure that a task gets the resources it needs?

- The Scheduler!
- Software that manages the HPC resources and decides which computation runs where and when.
Scheduler - SLURM

Scheduler receives a request for resources and creates a job

Job is put in the queue, where it waits for the resources

Job is assigned to the compute nodes and performs computation

When job is finished output and error files are created
Scheduler

- All clusters use SLURM

- Every computation that requests resources from the scheduler is called a *job*.

- *Submitting a job* means requesting resources from the scheduler and giving it a list of commands to run.
$ cp /xdisk/chrisreidy/workshops/sample_slurm.script .
$ cat sample_slurm.script

#!/bin/bash
#SBATCH --job-name=test
#SBATCH -e test.{e,A}
#SBATCH -o test.{o,A}
#SBATCH --partition=windfall
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00

echo 'This script is running on:'
hostname
sleep 120
#!/bin/bash

#SBATCH --job-name=test
#SBATCH -e test.e%A
#SBATCH -o test.o%A
#SBATCH --partition=windfall
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00

echo 'This script is running on:'
hostname

sleep 120
#!/bin/bash

#SBATCH --job-name=test
#SBATCH -e test.e%A
#SBATCH -o test.o%A
#SBATCH --partition=windfall
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00

echo 'This script is running on:'
hostname

sleep 120
#!/bin/bash
#SBATCH -J job-name=test
#SBATCH -e test.e%A
#SBATCH -o test.o%A
#SBATCH --partition=windfall
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=00:10:00

echo 'This script is running on:'
hostname
sleep 120
• Submitting a batch job

$ sbatch sample_slurm.script

Did you get an error?

If not, you will see:
Submitted batch job 2118950

$ squeue -j 2118950

<table>
<thead>
<tr>
<th>JOBID</th>
<th>PARTITION</th>
<th>NAME</th>
<th>USER</th>
<th>ST</th>
<th>TIME</th>
<th>NODES</th>
<th>NODELIST(REASON)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2118950</td>
<td>windfall</td>
<td>test</td>
<td>chrisrei</td>
<td>PD</td>
<td>0:00</td>
<td>1</td>
<td>(Priority)</td>
</tr>
</tbody>
</table>

$ cat test.o2118950
This script is running on:
r1u11n1.puma.hpc.arizona.edu
Detailed performance metrics for this job will be available at
https://metrics.hpc.arizona.edu/#job_viewer?action=show&realm=SUPREMM&resource_id=73&local_job_id=2118950 by 8am on 2021/09/23.
• Modifying your batch job

1/ Change the script to use 4 cores from one node

$ vi sample_slurm.script

Hint: look at our SLURM documentation

$ sbatch sample_slurm.script
Submitted batch job 2118966

2/ Change the script to use the standard queue

Answer:
#SBATCH --account=<PI GROUP>
#SBATCH --partition=standard

Note: *partition* is the same as *queue*
• Other SLURM commands

$ squeue
You might need to know grep to handle the hundreds of lines of output
$ squeue | wc

$ scancel jobid

$ scontrol show job 2099296
Show details about a running job

$ seff 2099296
Show details about a completed job

$ nodes-busy
$ system-busy
Shows activity of whole cluster at a glance
Interactive jobs

• Batch vs Interactive

Interactive is used when:
• You need to compile code
• You need test runs
• You need access to modules

Modules are not available on the login nodes
bash: module: command not found

interactive takes you to a compute node. It is an alias for:

$ salloc --job-name=interactive --mem-per-cpu=4GB --nodes=1 --ntasks=1 --time=01:00:00 --account=windfall --partition=windfall

Now your prompt includes the compute node hostname:
[chrisreidy@r3u13n1 chrisreidy]$
Job Composer with OnDemand

- Create and monitor your jobs

Jobs

Steps to create a job:
1. Create a new job by copying from an existing job template directory or a previously run job directory.
2. Edit the files in the job via the file explorer.
3. Submit the job and monitor the progress from the "Job Composer" index page.

Showing 0 to 0 of 0 entries
Accessing Software Modules

• Much software is available as “modules”
• 100 Applications from Abaqus to Xcrysden

To see the list:

Either

https://public.confluence.arizona.edu/display/UAHPC/Software+Resources

Or

$ shell
$ ocelote
$ interactive
$ module avail
Accessing Software with OnDemand

Please NOTE: "windfall" jobs will be restarted or terminated without notice if pre-empted by a "standard queue.

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps  A featured subset of all available apps

- ABAQUS GUI
  - System Installed App
- ANSYS Workbench GUI
  - System Installed App
- Mathematica GUI
  - System Installed App
- MATLAB GUI
  - System Installed App
### MATLAB GUI (187278)

- **Host:** `_j0n11.oce1ote.hpc.arizona.edu`
- **Created at:** 2021-09-22 20:34:37 MST
- **Time Remaining:** 59 minutes
- **Session ID:** 747e44e5-83b4-4515-b737-41147f8502fb

**Compression**

0 (low) to 9 (high)

**Image Quality**

0 (low) to 9 (high)

- **Launch MATLAB GUI**
- **View Only (Share-able Link)**

---

**Accessing Software with OnDemand**
Accessing Software with OnDemand
Accessing Software Command Line

- $ ocelote
- $ interactive
- `r3u11n1 $ module avail`
- `r3u11n1 $ module avail python`
- `r3u11n1 $ module load python/3.8`
- `r3u11n2 $ $ python3`
- Python 3.8.2 (default, Mar 16 2021, 17:11:14)
- [GCC 8.3.0] on linux
- Type "help", "copyright", "credits" or "license" for more information.
- `>>> import numpy as np`
- `>>> quit()`
Accessing Software Command Line

What modules do:

- Set your command path:
  - $ echo $PATH

- Set your library path:
  - $ echo $LD_LIBRARY_PATH

Try this from a compute node:
- $ which freesurfer
- $ module load freesurfer
- $ which freesurfer

Now try the two `echo` commands
module command options

Loading / Unloading commands:
- `add | load modulefile [...]`  Load modulefile(s)
- `rm | unload modulefile [...]`  Remove modulefile(s)
- `purge`  Unload all loaded modulefiles
- `reload | refresh`  Unload then load all loaded modulefiles
- `switch | swap [mod1] mod2`  Unload mod1 and load mod2

Listing / Searching commands:
- `list [-t|-l]`  List loaded modules
- `avail [-d|-l] [-t|-l] [mod ...]`  List all or matching available modules
- `aliases`  List all module aliases
- `whatis [modulefile ...]`  Print whatis information of modulefile(s)
- `apropos | keyword | search str`  Search all name and whatis containing str
Getting help

• HPC documentation docs.hpc.arizona.edu

• Support ticket
  https://uarizona.service-now.com/sp?id=sc_cat_item&sys_id=2983102adbd23c109627d90d689619c6&sysparm_category=84d3d1acdbc8f4109627d90d6896191f

• Office Hours
  https://gather.town/app/dVsAprPNBVmI9NpL/hpc-office-hours

• HPC consulting
  hpc-consult@list.arizona.edu

• Visualization consulting
  vislab-consult@list.arizona.edu

• Statistics consulting
  stat-consult@list.arizona.edu
The Research Bazaar is a worldwide festival promoting the digital literacy emerging at the center of modern research. Check us out @resbazaz

Need help with science or computers? Tired of quietly suffering trying to figure it out? Come hang out with us; it’s free!* Just want to hang out? Also free!*

Not into the bar scene? Not into mornings?

**Coffee & Code**

Weekly, Tuesdays 8-10
Catalyst Café (Keating Building)
1657 E Helen St

**HACKY Hour**

Weekly, Thursdays 4-7
Snakes & Lattes (Main Gate Square)
988 E University Blvd

* No beverages are provided. But it’s still fun and fairly cheap one way or the other.*

* Neither fun nor cheap is a guarantee. But if you are still reading this, and it’s funny: we feel more confident about the first part.