Intro to HPC

Dima Shyshlov
HPC consultant
University of Arizona
Intro to HPC*

*HPC – high-performance computing

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

Research is easy!

It’s still running...
Why HPC?

Problem

• Computation takes too long

• Computation is too big

• Too many computations
Why HPC?

Problem

• Computation takes too long

• Computation is too big

• Too many calculations

Solution

Get a specific computer just for the computation

Divide the task between many computers

Use lots of computers simultaneously
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?

Laptop

Supercomputer
Why HPC?

Laptop

Supercomputer
Why HPC?

Laptop
Personal

Supercomputer
Shared
Why HPC?

Laptop

Local

Supercomputer

Remote
Why HPC?

Laptop

Interactive

Supercomputer

Batch
Ocelote
The diagram of the UA HPC cluster
The diagram of the UA HPC cluster

- Users
- Bastion host
- Login node
- File transfer node
- Shared data storage
- Computes nodes
- High memory node
- GPU nodes
Connecting to Ocelote

ssh hpc.arizona.edu
Connecting to Ocelote (web browser)

• Open ood.hpc.arizona.edu in your web browser and login with your NetID and password.
• From the “Clusters” drop-down menu choose which HPC cluster you would like to access:
  • Exercise - connect to Ocelote.
Command line

Your NetID
(who are you)

Name of the current
directory

Node name
(where are you)

Prompt
(what are you going to do)
Login node

[dshyshlov@login2 ~]$
Login node

- Login node is a computer intended for users to prepare and manage computations:
  - submit jobs
  - edit files
  - compile codes
  - manage files
  - small-scale testing

- **DO NOT** run any calculations on the login node

I’ll just open MATLAB real quick
Working with a Linux shell

[dshyshlov@login2 ~]$ whoami

Output:
dshyshlov

Command: whoami
Working with a Linux shell

[dshyshlov@login2 ~]$ pwd
/home/u1/dshyshlov
[dshyshlov@login2 ~]$

• ~ is a shortcut for your /home directory
Working with a Linux shell

• List all the files and directories
  ```
  [dshyshlov@login2 ~]$ ls
  ```

• Make a directory
  ```
  [dshyshlov@login2 ~]$ mkdir Intro_to_HPC
  ```

• List all the files and directories again
  ```
  [dshyshlov@login2 ~]$ ls
  ```
Working with a Linux shell

- Change directory

```
[dshyshlov@login2 ~]$ cd Intro_to_HPC
[dshyshlov@login2 Intro_to_HPC]$  
```

- Go back a level

```
[dshyshlov@login2 Intro_to_HPC]$ cd ..
[dshyshlov@login2 ~]$  
```

- Change directory using absolute path

```
[dshyshlov@login2 ~]$ cd ~/Intro_to_HPC/
[dshyshlov@login2 Intro_to_HPC]$  
```
Working with a Linux shell

• Copy a file

```bash
$ cp /tmp/first_script.pbs .
```

• List all the files and directories again

```bash
[dshyshlov@login2 ~]$ ls
```

• View contents of the file on the screen

```bash
$ cat first_script.pbs
```
Storage

- Users
- Bastion host
- Login node
- File transfer node
- Shared data storage

/home /extra

Computes nodes
Storage

• Every user gets two default storage locations:
  
  • /home
    • the default home directory
    • 15GB
    • the only backed up storage on UA HPC/extra
  
  • /extra
    • full path: /extra/NetID
    • 200GB
    • not backed up
    • has file count limit 600 files/GB
Storage

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

<table>
<thead>
<tr>
<th>Storage</th>
<th>Back-up</th>
<th>File limits</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home</td>
<td>15 GB</td>
<td>Nightly</td>
<td>None</td>
</tr>
<tr>
<td>/extra</td>
<td>200 GB</td>
<td>None</td>
<td>600 files / GB</td>
</tr>
<tr>
<td>/xdisk</td>
<td>200 – 1000 GB (45 day limit)</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>/rsgrps</td>
<td>Rented space</td>
<td>None</td>
<td>600 files / GB</td>
</tr>
<tr>
<td>/tmp</td>
<td>Varies ~ 800 GB (Ocelote)</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>
Transferring Files

filexfer.hpc.arizona.edu
Transferring Files

- Ocelote has two specific nodes for file transfer
  - hostname – filexfer.hpc.arizona.edu

- Command line options:
  - scp, sftp, rsync, irods

- GUI options
  - Windows based: WinSCP
  - Cross-platform: Cyberduck

- Parallel data transfer
  - Globus
  - best option for large files
Transferring Files (web browser)

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

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

• Ocelote uses the scheduler PBS Pro.

• 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.
Scheduler

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
Queues

- **Standard queue**
  - 36,000 CPU-hours/month per group
  - higher priority

- **Windfall queue**
  - No time limit
  - Preemption

- **Debug queue**
  - 10 min limit
PBS script

• View the contents of a file `first_script.pbs` with a command

```
$ cat first_script.pbs
```

```
#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
```
PBS script

It’s a bash shell script

#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

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

#PBS  -W  group_list=hpcteam
#PBS  -q  standard
#PBS  -l  select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS  -l  walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
PBS script

Name of your HPC group (usually it is your sponsor’s NetID). You can find your groups name with a `vap` command.

```bash
#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
```
#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
PBS script

“Select statement”.
Create a custom computer from Ocelote’s resources for your job.

#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
PBS script

How long your custom computer will exist.

#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
PBS script

List of commands to run on your custom computer

#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
Exercise – running a batch job

• Submit your first job on Ocelote.

```
$ qsub first_script.pbs
qsub: Bad GID for job execution
```

• Edit first_script.pbs to correct the group name.

```
$ va find the group name
$ qsub first_script.pbs
1827586.head1.cm.cluster
Exercise – running a batch job

• Check on the status of the job.

  $ qstat -u NetID

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessID</th>
<th>NDS</th>
<th>TSK</th>
<th>Req'd Memory</th>
<th>Req'd Time</th>
<th>S</th>
<th>Elap Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1827579.head1.c</td>
<td>dshyshlo</td>
<td>oc_stand</td>
<td>first scri</td>
<td>--</td>
<td>1</td>
<td>1</td>
<td>6gb</td>
<td>00:05</td>
<td>Q</td>
<td>--</td>
</tr>
</tbody>
</table>

• Look in the standard output and error files (use `cat` command).
Exercise – customizing a job

• Submit the job that uses 4 cores from 1 node.

• What parameters do you need to change in the PBS script?
Exercise – deleting a job

• You can delete the job from the queue or when it’s running with a `qdel` command.

```
$ qdel 1827579.head1.cm.cluster
```

• You can find a full job ID with a command

```
$ qstat -wa -u NetID
```
Interactive job

• Request compute node resources for interactive work
  • interactive data analysis or compute
  • long-running debugging
  • copying large files

```
$ qsub -I first_script.pbs
```

• Close the interactive session with the command `exit`
or `logout`. 
Interactive job

#!/bin/bash
#PBS -W group_list=hpcteam
#PBS -q standard
#PBS -l select=1:ncpus=1:mem=6gb:pcmem=6gb
#PBS -l walltime=00:05:00

echo 'This script is running on:'
hostname
sleep 120
Accessing Software

GUI

Command line

$ matlab

Or more often...

$ matlab -noscreen < input.m > output.txt
Accessing Software

$ matlab

-bash: matlab: command not found

• To run any software in Linux the system must know where to look for appropriate binaries and libraries.

• Check your environment:

$ echo $PATH
Accessing Software

• Environment modules are a convenient way to customize your environment to use software

• Manage your modules with a command

  $ module

• Running module command without any options will open a help page.
Accessing Software: hand-on activity

• List your current modules
  
  `$ module list`

• List all the available modules
  
  `$ module avail`

• List all the available MATLAB modules
  
  `$ module avail matlab`

• Load a MATLAB module
  
  `$ module load matlab`

• List your current modules
Accessing Software: hands-on activity

• Check your environment again:

```
$ echo $PATH
```

• Anything new?

• When you run a batch job *module load* commands must be in the PBS script.
Being a good HPC user

• Things that can “break” the system:
  • heavy use of the login node
  • too many jobs
  • too many files
  • heavy I/O jobs
  • copying GB of data
  • *for* loops in PBS scripts
Getting help

• HPC documentation – docs.hpc.arizona.edu

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

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

• Statistics consulting – stat-consult@list.arizona.edu