

Computing in the ACI Cluster

Ben Seiyon Lee

Pennsylvania State University

Department of Statistics

October 5, 2017

Outline

- 1 High Performance Computing
- 2 Accessing ACI
- 3 Basic Unix Commands
- 4 Navigation and File Creation
- 5 SCP clients
- 6 PBS scripts
- 7 Running PBS scripts
- 8 Parallelization
- 9 Best Practices

High Performance Computing

- Large number of processors
- Large memory requirements
- Large storage requirements
- Long runtimes

ACI-B: Batch

- Log in to a head node and submit jobs to compute nodes
- Groups can purchase allocations or use open queue
- Intel Xeon E5-2680 v2 2.8 GHz, 256 Gb RAM, 20 cores per node
- Statistics Department has 5 nodes (20 processors per node)

Sign up for an account:

- ICS-ACI Account Sign-up
- 2-Factor Authentication

Mac

- Open Terminal
- ssh into ACI: `ssh jusernamej@aci-b.aci.ics.psu.edu`
- Complete 2 Factor Authentication

Windows

- Open Putty
- Enter `aci-b.aci.ics.psu.edu` in the Host Name field
- Select SSH then X11 and Enable X11 forwarding
- Select Connection then Data and enter your username in the Auto-login username field

Unix Commands

- Change directories: `cd`
 - Home Directory: `cd`
 - Here: `cd .`
 - Up one directory: `cd ..`
 - All files in the directory: `ls *`
 - Wildcards: `Test*` . `*.png`
 - Send output to another command: `—`
 - Write command output to a file: `ls > log.txt`

- Create Directory: `mkdir`

```
cd ~/work
mkdir Workshop
mkdir WorkshopB
ls
```

- Remove Directory: `rmdir`

```
rmdir WorkshopB
ls
mkdir WorkshopB
```

Unix Commands

- Move Files: mv

```
mv file1.txt ./WorkshopB/
```

```
mv ../WorkshopB/file1.txt ./WorkshopB/file2.txt
```

- Copy Files: cp

```
cp ../WorkshopB/file1.txt ../WorkshopB/file2.txt
```

- Remove Files: rm

```
rm file1.txt
```

```
rm -r WorkshopB
```

- Access Manual for commands: man

```
man rm
```

```
q
```

- List files: ls

```
ls
```

```
ls ~/work/Workshop
```

Unix Commands

- Print the current directory: `pwd`

```
pwd
```

- Past commands: `history`

```
history
```

- Manage permissions for a file: `chmod`

```
chmod u=rwx,g=rwx,o=rwx file1.txt
```

```
chmod 777 file1.txt
```

The digits represent the permissions for the user, group, and others, in that order. Each digit is a combination of the numbers 4, 2, 1, and 0: 4 stands for "read", 2 stands for "write", 1 stands for "execute", and 0 stands for "no permission." 7 is the combination of permissions 4+2+1 (read, write, and execute), 5 is 4+0+1 (read, no write, and execute), and 4 is 4+0+0 (read, no write, and no execute).

Transfer files using a local SCP client or directly from the command line

SCP Client

- Cyberduck (Mac) or WinSCP (Windows)
- Host name: `datamgr.aci.ics.psu.edu`
- PSU username and password
- Port 22
- Go through 2 Factor Authentication

Transferring Files to/from ACI

Directly from the command line

- Copy a file from local machine to ACI-B
scp `jfilenamej jusernamej@aci-b.aci.ics.psu.edu:jDirectory on ACIj`
scp `/test.txt skl5261@aci-b.aci.ics.psu.edu: /work/Workshop/`
- Copy a file from ACI-B to local machine
scp `jfilenamej jusernamej@aci-b.aci.ics.psu.edu:jDirectory on local machinej`
scp `skl5261@aci-b.aci.ics.psu.edu: /work/Workshop/test.txt /Downloads/`

Batch jobs

- Sends jobs to batch scheduler
- Scheduler allocates memory, nodes, and processors as necessary

Running and Monitoring Jobs

Running a Job

- `qsub -A groupname filename`
- Open Queue: `qsub -A open filename`

Monitoring a Job

- `qstat filename`
- `qstat -u userid`
- `qstat -p`
- `qstat -i jobid`

Deleting a Job

- `qdel jobid`

Conditional Execution

- Run a job after another job has finished
`qsub -W depend=afterok:ijobidi jsecond PBS filej`

R packages for parallelization

- Snowfall
- Rmpi
- doParallel

Possible Topics

- Unix Commands
- SCP clients
- PBS Scripts
- Parallelization
- Programming in other languages (C++, Python, Julia, Matlab, etc...)