Computing with ICS-ACI

Ben Seiyon Lee

Pennsylvania State University

Department of Statistics

Graduate Workshop

April 15, 2019

Outline

1. High Performance Computing
2. Accessing ACI + Details
3. Transferring Files to/from Local Machines
4. Submitting Jobs on ACI
5. Monitoring Jobs
6. Parallelization
7. Exercises

ACI-ICS High Performance Computing Environment

The Institute for CyberScience Advanced CyberInfrastructure (ICS-ACI) is Penn States's high-performance computing (HPC) infrastructure.

High Performance Computing

- Provides researchers with the hardware, software, and technical expertise needed to solve problems.
- 1000+ servers with 23,000+ cores and 6 Petabytes of storage
- Appropriate for computationally burdensome tasks (parallelization, storage, memory, long run times)
- Access provided through allocations (department or research lab) or the Open queue

"Getting Started with ACI" is a fantastic resource!

Accessing ACI-B for Batch Jobs

Sign up for an account:

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

Mac

- Open Terminal
- ssh into ACI: ssh <username>@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

How to run R scripts on ACI

For code development and testing (simple tasks), use the interactive shell (rstudio.aci.ics.psu.edu).
*I prefer testing on my laptop and syncing files via Github.

For computationally demanding tasks, send batch jobs to ACI-B.
ACI-B Server Types:

- **Basic:** 2.2 GHz Intel Xeon Processor, 24 CPU/node, 128 GB RAM
- **Standard:** 2.8 GHz Intel Xeon Processor, 20 CPU/node, 256 GB RAM
- **High:** 2.2 GHz Intel Xeon Processor, 40 CPU/node, 1 TB RAM
- **GPU:** 2.5 GHz Intel Xeon Processor, 2 Nvidia Tesla K80 computing modules/server, 24 CPU/server

Transferring Files - SCP Clients

2 Options: SCP client or the command line

SCP Client

- Cyberduck (Mac) or WinSCP (Windows) or FileZilla (All OS)
- **Host name: datamgr.aci.ics.psu.edu**
  Dedicated hostname for file transfers.
- File Protocol: SFTP
- PSU username and password
- Port 22
- Go through 2 Factor Authentication
- Easy-to-use interface similar to Windows Explorer or Finder.

Transferring Files - Command Line

Command line

- Helpful for transferring many files
- Use the dedicated data manager hostname **datamgr.aci.ics.psu.edu**
- Other option: git for small files and the command line for large files

Examples:

- **Copy a file from local machine to ACI-B:**

  scp <filename> <username>@datamgr.aci.ics.psu.edu:<ACI Directory>

  scp test.txt skl5261@datamgr.aci.ics.psu.edu:∼/work/Workshop/

- **Copy a file from ACI-B to local machine**

  scp <username>@datamgr.aci.ics.psu.edu:<filename> <Local Directory>

  scp skl5261@datamgr.aci.ics.psu.edu:∼/work/Workshop/test.txt ∼/home/Workshop/

Sending Jobs via PBS (Portable Batch System)

Batch jobs

- Sends jobs to batch scheduler using a **.PBS** file
- Scheduler allocates memory, nodes, and processors as necessary

Example PBS File:

```
#!/bin/bash
#PBS -A open Allocation
#PBS -l nodes=1:ppn=3 Nodes and processors per node
#PBS -l walltime=72:00:00 Requested wall time
#PBS -l pmem=5GB Memory per processor
#PBS -N ProjName Job Name
#PBS -j oe Prints log file for completion and errors
#PBS -m abe Sends email when job aborts, begins, and ends
#PBS -M psuid@psu.edu Email Address
Rscript test.R Runs the R file
```

Running and Monitoring Jobs

Use the command line to submit, monitor, or cancel your jobs.

Submit a Job

- qsub <PBSfile>
- Details should be provided in the PBS file

Monitor a Job

- Show queued or running jobs: qstat
- Jobs by userid: qstat -u <userid>
- Jobs by jobid: qstat -i <jobid>
- Show all jobs: showq

Cancel a Job

- qdel jobid

Conditional Execution

- Run a job after another job has finished
  qsub -W depend=afterok:<jobid> <second PBS file>

Parallelizing Jobs

Types of Clusters:

- PSOCK: Small-scale computing environments (1 node/<20 cores)
- MPI: Larger-scale computing environments (2+ nodes/20+ cores)

USE MPI FOR JOBS REQUIRING 20+ CORES!

- Open allocation: Limited to 100 processors (5 nodes) and 48 hours of walltime.

R packages for parallelization:

- Snowfall
- Rmpi
- doParallel

Try it out!

Instructions:

1. Download and unzip the folder **workshop**
2. Login into ACI
3. Copy the “Workshop” folder to your ACI work directory (∼ /work/.)
4. Follow instructions in the Readme file

Exercises:

1. **Simple Example run on command line**
2. **Simple Script using PBS Scheduler + 1 Jobs**
3. **Parallel Script using PBS Scheduler**
4. **Parallel Script using PBS Scheduler + MPI**
5. **Simple Script using PBS Scheduler + 10 Jobs**

Results

Exercises:

1. **Simple Example run on command line**
   ∼100 seconds (1.7 minutes)
2. **Simple Script using PBS Scheduler + 1 Jobs**
   – This takes ∼2 minutes
3. **Parallel Script using PBS Scheduler**
   – This takes ∼15 seconds. Major speedup!
4. **Parallel Script using PBS Scheduler + MPI**
   – Limited to 100 processors (5 nodes) on the Open queue.
   – Requires installing Rmpi which can be tricky. Installation script included in exercise files.
   – This takes ∼49 seconds. Longer times due to communication costs.
5. **Simple Script using PBS Scheduler + 10 Jobs**
   – This takes ∼15 seconds per run. But jobs start at different times.

The End

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 (piping): |
  - Write command output to a file: ls > log.txt
- Create Directory: mkdir

```
cd ~/work
mkdir Workshop
```

- Remove Directory: rm -r <directory>

```
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 directoy:

```
pwd
```

- Past commands:

```
history
```

- Manage permissions for a file:

```
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 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).