Accelerate Your Science: An Introduction to High Performance Computing
Lab 1: Accessing IU HPC resources, using software modules, running jobs

Guido Juckeland
Visiting Scholar
Technische Universität Dresden, Germany

July 7, 2014
Goal

• Find out how to access IU HPC resources
• Login to Big Red II
• Try out basic Linux commands
• Work with software modules
• Working with the batch system
Access to IU HPC resources

• Access available to everybody with an IU Account

• Account management online
  https://itaccounts.iu.edu/

• Use “Mange Accounts” to view your existing accounts or create more accounts

• If you do not have an account for Big Red II, create one
• (do the same for additional resources you plan to use)
Login to Big Red II

- Grab the SSH-program of your choice
  - Windows: PuTTY (https://iuware.iu.edu/Windows/Package/781)
  - Windows: Cygwin (https://www.cygwin.com/)
  - Linux/Mac OS X: Just open a terminal

- Server address: bigred2.uits.iu.edu
- Username: Your IU username

- e.g.: ssh iu-network-id@bigred2.uits.iu.edu
Optional: Login without a passphrase (a.k.a public key authentication)

- Works with a keypair (public + private key)
- Public key goes onto Big Red II
- Private key always stays with you!

- For Linux/Mac OS X/Cygwin see here: https://kb.iu.edu/d/aews
- For PuTTY see here: http://www.ualberta.ca/CNS/RESEARCH/LinuxClusters/pka-putty.html
Ways to copy data onto Big Red II

• SCP/SFTP
  – For Windows use WinSCP
    (https://iuware.iu.edu/Windows/Package/785)
  – For Linux/Mac OS X use the `scp` command
    `scp myfile.zip username@bigred2.uits.iu.edu:`

• Pull the data from the WWW on BRII
  `wget http://www.some.url/go/myfile.zip`
Get the lab material

- Pull the lab material from the IU website
  
  `wget ...`

- Unpack the material
  
  `tar xvzf lab1.tar.gz`

- Use WinSCP/scp to copy the file “README.txt” onto your computer
  
  `scp iu-network-id@bigred2.uits.iu.edu:README.txt .`
Basic Linux shell commands

- The prompt (on Big Red II):
  `username@login1:~>`
  - `username`: will be your IU username
  - `login1`: tells you which system you are logged into (in this case login node #1 of bigred2)
  - `~`: tells you where in the directory structure you currently are (~ means your personal home directory)
  - `>`: after this the cursor will blink waiting (patiently) for your actions (sometimes also shown as $ or %)

- General command structure
  `command [options] <filename>`
Basic Linux Shell commands (2)

• Data Handling
  – List files in the current directory:  `ls`
  – Change into a subdirectory:  `cd <dirname>`
    (cd .. for going one directory back)
  – Create a subdirectory:  `mkdir <dirname>`
  – Remove an empty subdirectory:  `rmdir <dirname>`
  – Copy a file:  `cp <srcfile> <dstfile>`
  – Rename/Move a file:  `mv <srcfile> <dstfile>`
  – Remove a file:  `rm <filename>`

• Note: Separator between directories is `/`
• Note: Wildcard `*` can be anything (e.g. `rm * .txt`)
Accomplish basic tasks

- List the contents of your home directory
- Copy the file “foo” to a new file called “bar”
- Delete the file “foo”
- Rename the file “bar” to “foo”
- Change into the subdirectory lab1
- List its contents
Software modules

• are an easy way to control the available software
• Allow you to use exactly the software package and version you need, a.k.a. a module
• Typical actions:
  – Show installed modules module avail
  – Show currently loaded modules module list
  – Load a module module load <name>
  – Unload a module module unload <name>
  – Swap module 2 for 1 module switch <1> <2>
• More at: https://kb.iu.edu/d/bcwy
Introducing the swiss army knife on Linux systems

• **Usage:**
  module load midnightcommander *(only needed on Big Red II)*
  mc

• **Use it for**
  – File operations
  – Archive handling
  – Text viewing
  – Text editing
Using midnight commander

- Copy the file “foo” to a new file called “bar”
- Delete the file “foo”
- Rename the file “bar” to “foo”
- Change into the subdirectory lab1
- Go back to your home directory
- View the file “README.txt”
- Edit the file “changeme.txt”
The batch system – your personal assistant

Login-Node

Task 1
Task 2
Task 3
Task 4
...

Compute Nodes

1

2

3

4
A batch job

- **Batch script describes the requirements of your task**

```bash
#!/bin/bash
#PBS -l nodes=1:ppn=32
#PBS -l walltime=00:10:00
#PBS -N my_job
#PBS -q cpu
#PBS -V
aprun -n 32 my_binary
```

- **Available options here:**
  - [https://kb.iu.edu/d/avmy](https://kb.iu.edu/d/avmy)
  - [https://kb.iu.edu/d/avgl](https://kb.iu.edu/d/avgl)
  - [https://kb.iu.edu/d/bdkt](https://kb.iu.edu/d/bdkt)
  - [https://kb.iu.edu/d/bcqt#queues](https://kb.iu.edu/d/bcqt#queues)
Controlling batch jobs

• Submission
  – qsub batchscript
  – Will print the ID for your job

• Checking on jobs
  – qstat -u <username>
  – checkjob [-v] <jobid>
  – showstart <jobid>

• Removing bad jobs
  – qdel [-W] <jobid>
Your own batch job

- Run the sample job
  `qsub sample.job`

- Check its output by looking at `sample.out`

- Change the job to run on the GPU nodes of BRII
Your personal supercomputer: Interactive batch jobs

- `qsub` option `-I`
- e.g.:
  
  `qsub -I -l walltime=00:30:00 -l nodes=1:ppn=32 -q debug_cpu`

- If you are on Big Red II, do not forget the `aprun`!

- More information here: [https://kb.iu.edu/d/bdsi](https://kb.iu.edu/d/bdsi)
Run a sample program

- Grab an interactive session
  ```bash
  qsub -I -l nodes=1:ppn=16 -l walltime=00:30:00 -q debug_gpu
  ```
- Load the CUDA toolkit
  ```bash
  module load cudatoolkit
  ```
- Go to lab 1
  ```bash
  cd lab1
  ```
- Compile the sample program
  ```bash
  nvcc sample.cu
  ```
- Run the sample program
  ```bash
  aprun ./a.out
  ```