Introduction to HPC on CURC

Introduction to HPC on Blanca

- Me: Andrew Monaghan (*Andrew.Monaghan@Colorado.edu*)
- *RC Homepage: https://www.colorado.edu/rc*
- *RC Docs: https://www.colorado.edu/rc*
- *Courses and Office Hours:* https://www.colorado.edu/crdds/events
- *For this lecture, slides and examples available for download at:* https://github.com/ResearchComputing/CHEM4555_Spring_2022/

Outline for this presentation

- Intro to CURC resources
- Linux basics and navigating the CURC system
- Software
- Scheduling batch and interactive jobs on Summit
- CURC JupyterHub, CURC EnginFrame (Demo)
- Data transfers (Globus, scp) (Demo)

What is Research Computing?

- Provide services for researchers that include:
  - Large scale computing
  - Data storage
  - High speed data transfer
  - Data management support
  - Consulting
  - Training

- We are likely best known for:
  - Summit Supercomputer (~12,000 cores)
  - Blanca ”condo” cluster (~4,000 cores)
  - PetaLibrary storage

Hardware - Summit Supercomputer

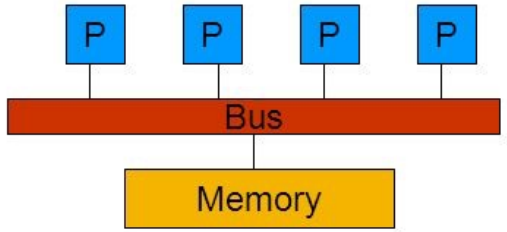
- 450+ compute nodes (mostly Intel Xeon Haswell)
- 24 cores per “shas” (general compute) node, different core counts for other node types
- High-speed network fabric between nodes
- 1.2 PB scratch storage
- 67% CU, 23% CSU, 10% RMACC

What Would I Use Summit For?

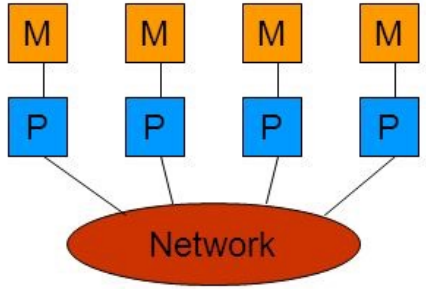
Solving large problems that require more:

- Memory than you have on your personal computer
- Cores/nodes/power than you have on your personal computer

- Blanca is set up for both shared memory (single node) and distributed memory (multi-node) parallelization.
  - Can also use Summit for"big" distributed memory parallelization



- **Shared memory**



- **Distributed memory**

Source: https://images.slideplayer.com/25/7599921/slides/slide_4.jpg

Typical Summit Workflow

shas0101	shas0201	shas0301
shas0102	shas0202	shas0302
shas0103	shas0203	shas0303
shas0104	shas0204	shas0304
…	…	…
shas0401	shas0501	shas0601
shas0402	shas0502	shas0602
shas0403	shas0503	shas0603
shas0404	shas0404	shas0604
…	…	…

Logging In

- `ssh <identikey>@login.rc.colorado.edu`
- Enter your `identikey_password`
- Authenticate by accepting the Duo push to your smartphone
  - Can also authenticate by text message, phone call, or token
- More info here: https://curc.readthedocs.io/en/latest/access/logging-in.html

Basic Linux commands

- **ls** – lists the contents of a directory (`ls -l` gives detailed listing)
- **pwd** – prints full path to current directory
- **cd** – changes directory; can use full or relative path as target
- **mkdir** – creates a subdirectory in the current directory
- **rm** – removes a file (`rm -r` removes a directory and all of its contents)
- **cp** – copies a file
- **mv** – moves (or renames) a file or directory
- **more** – displays a file one screen at a time
- **head** – prints the first few lines of a file
- **tail** – prints the last few lines of a file

File editing with nano

- To edit a file:
  - nano myfile.txt
- From within Nano:
  - Ctrl+o save (need to confirm filename)
  - Ctrl+x exit
  - Ctrl+k cut
  - Ctrl+u paste
- Other population Linux editors: vi, emacs

Your personal directories on CURC

- /home/<username>
  - Very small: 2GB.
  - Backed up daily.
  - Good for ‘can’t lose’ files
- /projects/<username>
  - 250 GB
  - Backed up regularly
  - Good for storing scripts, self-installed software, some data
- /scratch/summit/<username>
  - 10 TB (we can expand if you need)
  - Good for jobs with lots of I/O
  - Not backed up
  - Temporary: data deleted 90 days from creation.

Software

- Common software is available to everyone on the systems

- Research Computing uses modules to manage software
  - You load modules to prepare your environment for using software
    - Modules set any environment variables, paths, etc.
    - Set environment so application can find appropriate libraries, etc.

- You can also install your own software
  - It is best if you are responsible for support
  - We are happy to assist

Using Modules

- Must be on a compute node to browse the modules (e.g, bnode0501)
- To set up your environment to use a software package, type `module load <package>/<version>`
- Some modules might require a specific hierarchy to load
  - For some modules, you may need to specify a specific version
    - For example, `module load intel/17.4 impi lammps/29Oct20`
- To find dependencies for a module, type `module spider <package>`
- To find out what software is available, you can type `module avail`

Using Anaconda for Python, R

- Many users prefer to use conda to manage python (and R, etc) packages and environments.

- We have Anaconda python installed on our system!
  - (but it isn't a module…)

- Documentation on use: https://curc.readthedocs.io/en/latest/software/python.html

Next topic: Job scheduling

- **Job** – a system allotment of resources that run a particular application.
- **Slurm** -- resource manager
- Because Summit and Blanca are shared resources among a variety of groups on campus, users must run their applications through jobs.
  - Ensures everyone can utilize the system
  - No one person taking up too much of the system!

Two types of jobs

- **Interactive Jobs**
  - Interactive allotments or resources where user run their applications manually
  - Useful for:
    - Debugging Applications
    - Running GUI Applications
- **Batch Jobs**
  - Non-interactive allotment of resources that run applications in the background
  - Think “baking a batch of cookies”
  - Useful for:
    - Applications that take a substantial amount of time
    - Non-interactive Applications

Scheduling an interactive job

- To work with a program interactively, we request time from Summit
- When the resources become available the job starts

- Commands to run:

```
sinteractive --reservation=chem4555 --time=00:10:00
```

*note “--reservation=chem4555” is only for this workshop

- Once we receive a prompt, then:

```
module load gaussian
g16
```

- Once we finish we must exit! (job will time out eventually)

Scheduling Batch Jobs

- `sbatch` command: schedule a batch job with Slurm
- The sbatch command usually takes in 1 parameter: A job script.
  - Job scripts provide all information on what is needed for the job.
  - Parameters can be overwritten or added externally by specifying the parameter as a flag.
- Example:

```
sbatch test.sh
```

or

```
sbatch test.sh --time=02:00:00
```

Anatomy of a job script

Job parameters

Your job commands

```
#!/bin/bash

#SBATCH --ntasks=1                          # Number of requested tasks
#SBATCH --time=0:01:00                      # Max wall time
#SBATCH --partition=shas            # Specify APPM nodes
#SBATCH --qos=normal          # Specify you are student
#SBATCH --output=test_%j.out                # Rename standard output file

# Now run job (commands below here)

# Purge and load needed modules
module purge
module load R

# Run commands
Rscript myscript.R
```

Job Parameters

- Allocation: `--account=<account-name>`
- Partition: `--partition=<partition-name>`
- Number of nodes: `--nodes=<nodes>`
- Number of Tasks: `--ntasks=<number-of-tasks>`
- Quality of service: `--qos=<qos>`
- Reservation: `--reservation=<name>`
- Wall time: `--time=<wall-time>`
- Job Name: `--job-name=<jobname>`
- Output File: `--output=<outputname>`

More on slurm commands: https://slurm.schedmd.com/quickstart.html

FYI: You do NOT actually type <> above – this designates something specific you as a user must enter about your job

Available Partitions

Partition	Description	# of nodes	cores/node	GPUs/node
shas	General Compute (Haswell)	380	24	0
sgpu	GPU-enabled nodes	10	24	effectively 4
smem	High-memory nodes	5	48	0
sknl	Phi (Knights Landing) nodes	20	68	0

Quality of Service

QoS	Description	Maxwall	Max jobs/user	Max nodes/user
normal	Default QoS	Derived from partition	n/a	256
testing	For quick turnaround when testing	30 Min	1	2/user; max 12 cores/node
interactive	For interactive jobs (command or GUI)	4 Hours	1	1 core
long	For jobs needing longer wall times	7 D	n/a	20
condo	For groups who have contributed to the Summit condo	7 D	n/a	n/a

Let’s schedule a batch job

```
cd /projects/$USER
git clone https://github.com/researchcomputing/CHEM4555_Spring_2022
cd CHEM4555_Spring_2022/example_scripts/lammps
sbatch --reservation=chem4555 testscript_lammps.sh
```

Other topics

CURC JupyterHub

Website: https://jupyter.rc.colorado.edu

Docs: https://curc.readthedocs.io/en/latest/gateways/jupyterhub.html

CURC EnginFrame Remote Desktop:

Website: https://viz.rc.colorado.edu

Docs: https://curc.readthedocs.io/en/latest/gateways/enginframe.html

Data Transfers to/from CURC:

https://curc.readthedocs.io/en/latest/compute/data-transfer.html

Thanks!

- Please fill out the survey: http://tinyurl.com/curc-survey18
- Contact: rc-help@Colorado.edu, *Andrew.Monaghan@Colorado.edu*
- Course materials for today:
  - https://github.com/ResearchComputing/CHEM4555_Spring_2022
- CURC documentation: https://curc.readthedocs.io
- Learning materials: https://www.colorado.edu/crdds/learning-materials
- Courses and Office Hours: https://www.colorado.edu/crdds/events