What is a Supercomputer?

Shelley Knuth
shelley.knuth@colorado.edu

www.rc.colorado.edu Questions? #RC_BasicSC

Link to survey on this topic: http://tinyurl.com/rcpresurvey

Slides:
https://github.com/ResearchComputing/Final_Tutorials/tree/master/Basics_Supercomputing

Outline

- About RC and the workshop
- General information about supercomputers
- Supercomputer detail
- Running jobs
- Accessing RC resources

Survey!

- http://tinyurl.com/rcpresurvey

What does Research Computing do?

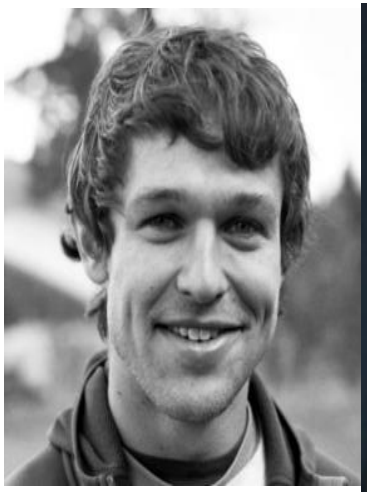
- We manage
  - Shared large scale compute resources
  - Large scale storage
  - High-speed network without firewalls – ScienceDMZ
  - Software and tools
- We provide
  - Consulting support for building scientific workflows on the RC platform
  - Training
  - Data management support in collaboration with the Libraries

Who Are We?

- Meet your instructors!
  - Shelley Knuth
  - Max Joseph
  - Tim Dunn
  - Pete Ruprecht
  - Aaron Holt
- Meet your assistants!
  - Sarah Papich
  - Matt Oakley
  - Zach Schira

Tim Dunn

Shelley Knuth



Max Joseph

Pete Ruprecht

Agenda – Day 1

09:00-10:15	What is a Supercomputer?	Shelley Knuth
10:15-10:30	Break	
10:30-11:45	Getting to Know the Command Line	Max Joseph
11:45-13:00	Lunch	
13:00-14:15	Submitting Jobs to the Supercomputer	Shelley Knuth
14:15-14:30	Break	
14:30-15:45	Installing and Building Software	Tim Dunn
15:45-16:00	Break	
16:00-17:15	How Can I Move and Store All that Data?	Max Joseph

Agenda – Day 2

09:00-10:15	What's Different About Summit?	Pete Ruprecht
10:15-10:30	Break	
10:30-11:45	Using the Supercomputer from a Website	Tim Dunn
11:45-13:00	Lunch	
13:00-14:15	What is this Parallel Computing Thing?	Shelley Knuth
14:15-14:30	Break	
14:30-15:45	Efficient Submission of Serial Jobs	Aaron Holt
15:45-16:00	Break	
16:00-17:15	How to Parallel Program	Shelley Knuth

General Information

What Is a Supercomputer?

- A supercomputer is one large computer made up of many smaller computers and processors
- Each different computer is called a node
- Each node has processors/cores
  - Carry out the instructions of the computer
- With a supercomputer, all these different computers talk to each other through a communications network
  - Example - InfiniBand

Computers and Cars - Analogy

Computers and Cars - Analogy

Why Use a Supercomputer?

- Supercomputers give you the opportunity to solve problems that are too complex for the desktop
  - Might take hours, days, weeks, months, years
  - If you use a supercomputer, might only take minutes, hours, days, or weeks

- Useful for problems that require large amounts of memory

World's Fastest Supercomputers

www.top500.org June 2016

Rank	Site	Name	TeraFlops
1	National Supercomputing Center (Wuxi, China)	Sunway	125435.9
2	National Super Computer Center (Guangzhou, China)	Tianhe-2	54902.4
3	Oak Ridge National Laboratory (United States)	Titan	27112.5
4	DOE/NNSA/LLNL (United States)	Sequoia	20132.7
5	RIKEN Advanced Institute for Computational Science (Japan)	K	11280.4
6	DOE/Argonne National Lab (United States)	Mira	10066.3
7	DOE/NNSA/LANL/SNL (United States)	Trinity	11078.9
8	Swiss National Supercomputing Centre (Switzerland)	Piz Daint	7788.9
9	HLRS - Höchstleistungsrechenzentrum Stuttgart (Germany)	Hazel Hen	7403.5
10	King Abdullah University of Science and Technology (Saudi Arabia)	Shaheen II	7235.2

What Does It Mean to Be Fast?

- Titan can do 27 trillion calculations per second
- A regular PC can perform 17 billion per second
- Researchers can get access to some of these systems through XSEDE (The Extreme Science and Engineering Discovery Environment)

Supercomputer Details

Hardware - Janus Supercomputer

- 1368 compute nodes (Dell C6100)
- 16,428 total cores
- No battery backup of the compute nodes
- Fully non-blocking QDR Infiniband network
- 960 TB of usable Lustre based scratch storage
  - 16-20 GB/s max throughput

Additional Compute Resources

- 2 Graphics Processing Unit (GPU) Nodes
  - Visualization of data
  - Exploring GPUs for computing
- 4 High Memory Nodes
  - 1 TB of memory, 60-80 cores per node
- 16 Blades for long running jobs
  - 2-week walltimes allowed
  - 96 GB of memory (4 times more compared to a Janus node)

Different Node Types

- Login nodes
  - This is where you are when you log in
  - No heavy computation, interactive jobs, or long running processes
  - Script or code editing, minor compiling
  - Job submission
- Compile nodes
- Compute/batch nodes
  - This is where jobs that are submitted through the scheduler run
  - Intended for heavy computation

Storage Spaces

- System variations
- **Home Directories**
  - Store source code
  - Not for direct computation
  - Small quota (2 GB)
  - Backed up
- **$PROJECT Space**
  - Mid level quota (250 GB)
  - Large file storage
  - Not backed up

- **Scratch Directory**
  - Much larger – depends on system
  - Output from running jobs should go here
  - Files generally purged at some point

Jobs

What is Job Scheduling

- Supercomputers usually consist of many nodes
- Users submit jobs that may run on one or multiple nodes
- Sometimes these jobs are very large; sometimes there are many small jobs
- Need software that will distribute the jobs appropriately
  - Make sure the job requirements are met
    - Reserve nodes until enough are available to run a job
    - Account for offline nodes
- Also need software to manage the resources
- Integrated with scheduler

Job Scheduling

- On a supercomputer, jobs are scheduled rather than just run instantly at the command line
  - People “buy” time to use the resources (allocation)
  - Shared system
  - Request the amount of resources needed and for how long
  - Jobs are put in a queue until resources are available
  - Once the job is run they are “charged” for the time they used

Job Scheduling - Priority

- What jobs receive priority?
  - Can depend on the center
  - Can arrange for certain people who “pay more” receive priority
  - Generally though based on job size and time of entry
- Might have different queues based on different job needs
- Can receive priority on a job by creating a reservation

Wall Times

- The maximum amount of time your job will be allowed to run
- How do I know how much time that will be?
- What happens if I select too much time?
- What happens if I select too little time?

Job Schedulers - Slurm

- Jobs on supercomputers are managed and run by different software
- Simple Linux Utility for Resource Management (Slurm)
  - Open source software package
- Slurm is a resource manager
  - Keeps track of what nodes are busy/available, and what jobs are queued or running
- Slurm is a scheduler
  - Tells the resource manager when to run which job on the available resources

Running Jobs

- What is a “job”?
- Interactive jobs
  - Work interactively at the command line of a compute node
- Batch jobs
  - Submit job that will be executed when resources are available
  - Create a text file containing information about the job
  - Submit the job file to a queue
- Load the Slurm module!

Useful Slurm Commands

- **sbatch**: submit a batch script to slurm
  - Standard input (keyboard)
  - File name
    - Options preceded with #SBATCH
- sbatch exits immediately after receiving a slurm job ID
- By default, standard output and errors go to file named slurm-%j.out (job allocation number)
- Slurm runs a single copy of the script on the first node in the set of allocated nodes

http://slurm.schedmd.com/sbatch.html

http://slurm.schedmd.com/sbatch.html

SBATCH Options

- In batch script put:
  #SBATCH <options> OR sbatch <options>

- Allocation: `--A=<account_no>`
- Checkpoints: `--checkpoint=<interval>`
- Sending emails: `--mail-type=<type>`
- Email address: `--mail-user=<user>`
- Number of nodes: `-N <nodes>` or `--nodes=<nodes>`
- Queues: `--qos=<qos>`
- Reservation: `--reservation=<name>`
- Wall time: `--time=<wall time>`
- Job Name: `-J <jobname>` or `-job-name=<jobname>`

Queues

- There are several ways to define a “queue”
- Clusters may have different queues set up to run different types of jobs
  - Certain queues might exist on certain clusters/resources
  - Other queues might be limited by maximum wall time
- Slurm can use a “quality of service” for each queue
  - aka “QOS”
- Also can use a “partition” (or set of nodes) that corresponds to a queue

Quality of Service = Queues

- janus-debug
  - Only debugging – no production work
  - Maximum wall time 1 hour, 2 jobs per user
    - (The maximum amount of time your job is allowed to run)
- janus
  - Default
  - Maximum wall time of 24 hours, 480 nodes/job
- janus-long
  - Maximum wall time of 7 days; 40 nodes/user
- himem
- crestone
- gpu

Software

- Common software is available to everyone on the systems
- To find out what software is available, you can type **ml avail**
- To set up your environment to use a software package, type **ml <package>/<version>**
- Can install your own software
  - But you are responsible for support
  - We are happy to assist

Initial Steps to Use RC Systems

- Apply for an RC account
  - https://portals.rc.colorado.edu/account/request
- Get registered with Duo
  - Duo invitation
  - Smart phone app
  - Push notifications
- Apply for a computing allocation
  - Startup allocation of 50K SU granted immediately
  - Additional SU require a proposal

Janus/Summit Access

- To access RC's computing, in general:

```
ssh login.rc.colorado.edu –l <username>
Password:  duo:<identikey>
```

- Need the Duo application

What's Next?

- So far we've introduced you to the basics of supercomputing
- Next, learn to:
  - Use the command line
  - Submit jobs!
  - Transfer data!
  - Load up some software!

Questions?

- Email rc-help@colorado.edu
- Twitter: CUBoulderRC

- Link to survey on this topic:

http://tinyurl.com/curc-survey16

- Slides: https://github.com/ResearchComputing/Final_Tutorials/tree/master/Basics_Supercomputing