Supercomputing: An Overview

Shelley Knuth
shelley.knuth@colorado.edu

Peter Ruprecht
peter.ruprecht@colorado.edu

www.rc.colorado.edu

Link to survey on this topic: http://goo.gl/forms/8VidcwOhRT

Slides: https://github.com/ResearchComputing/Final_Tutorials

Purpose

- The purpose of this talk is to inform you about Research Computing resources on campus
  - Use for teaching
  - Use for your research
  - Expose your students to new skills and knowledge

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

HPC as a Teaching Resource

- Janus is available for CU courses
  - Can run large compute jobs as part of courses you might teach
  - Visualization
  - Software you do not have on a current system
  - Can accommodate large groups
- XSEDE resources
- RC members can give technical talks to catch your students up on subjects they need to know
  - Linux, Python, Matlab, etc

Examples

- Terrestrial carbon cycle model to estimate permafrost carbon feedback
- Electronic structure and molecular dynamics relevant to production of renewable hydrogen
- Machine learning to mine next-generation sequencing data
- Geocomputational approaches to improving U.S. Census data
- Fluid dynamics applied to aerodynamic and cardiovascular flows

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

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

World's Fastest Supercomputers

www.top500.org June 2015

Rank	Site	Name	TeraFlops
1	National Super Computer Center (Guangzhou, China)	Tianhe-2	54902.4
2	Oak Ridge National Laboratory (United States)	Titan	27112.5
3	DOE/NNSA/LLNL (United States)	Sequoia	20132.7
4	RIKEN Advanced Institute for Computational Science (Japan)	K	11280.4
5	DOE/Argonne National Lab (United States)	Mira	10066.3
6	Swiss National Supercomputing Centre (Switzerland)	Piz Daint	7788.9
7	King Abdullah University of Science and Technology (Saudi Arabia)	Shaheen II	7235.2
8	Texas Advanced Computing Center (United States)	Stampede	8520.1
9	Forschungszentrum Juelich (Germany)	JUQUEEN	5872.0
10	DOE/NNSA/LLNL (United States)	Vulcan	5033.2

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)

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)

Initial Steps to Use RC Systems

- Apply for an RC account
  - https://portals.rc.colorado.edu/account/request
- Get a One-Time Password device
- Apply for a computing allocation
  - Startup allocation of 50K SU granted immediately
  - Additional SU require a proposal
  - You may be able to use an existing allocation

What is Job Scheduling

- Supercomputers usually consist of many nodes
- Users submit jobs that may run on one or multiple nodes
- Sometimes these jobs run for a long time; sometimes there are many short 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
  - 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” to receive higher priority
  - Generally though based on job size and time of entry
- Might have different queues based on different job needs
- Can earmark space for a job by creating a reservation

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!

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

Storage Spaces

- **Home Directories**
  - Not high performance; not for direct computational output
  - 2 GB quota
  - /home/user1234
- **Project Spaces**
  - Not high performance; can be used to store or share programs, input files, maybe small data files
  - 250 GB quota
  - /projects/user1234
- **Lustre Parallel Scratch Filesystem**
  - No hard quotas
  - Files created more than 180 days in the past may be purged at any time
  - /lustre/janus_scratch/user1234

Research Data Storage: PetaLibrary

- NSF Major Research Instrumentation grant
- Long term storage option
- Keep data on spinning disk or tape
- Provide expertise and services around this storage
  - Data management
  - Consulting
- No HIPAA, FERPA data
- Infrastructure guaranteed for 5 years

Training

- Weekly tutorials on computational science and engineering topics
- Meetup group
  - http://www.meetup.com/University-of-Colorado-Computational-Science-and-Engineering
- All materials are online
  - https://github.com/ResearchComputing/Final_Tutorials
  - Various boot camps/tutorials

Consulting

- Support in building software
- Workflow efficiency
- Parallel performance debugging and profiling
- Data management in collaboration with the Libraries
- Getting started with XSEDE

Research Computing - Contact

- rc-help@colorado.edu
  - Main contact email
  - Will go into our ticket system
- http://www.rc.colorado.edu - main web site
- http://data.colorado.edu - data management resource
- Mailing Lists (announcements and collaboration)
  https://lists.rc.colorado.edu/mailman/listinfo/rc-announce
- Twitter @CUBoulderRC
- Facebookhttps://www.facebook.com/CuBoulderResearchComputing
- Meetuphttp://www.meetup.com/University-of-Colorado-Computational-Science-and-Engineering/

Questions?

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

- Link to survey on this topic:
  http://goo.gl/forms/8VidcwOhRT

- Slides:
  https://github.com/ResearchComputing/Final_Tutorials