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

Outline

- Presentation on Research Computing resources
- General information on large scale computing
- Access to CU resources
- Demonstration of how to access systems
- Education/training opportunities
- Office hours

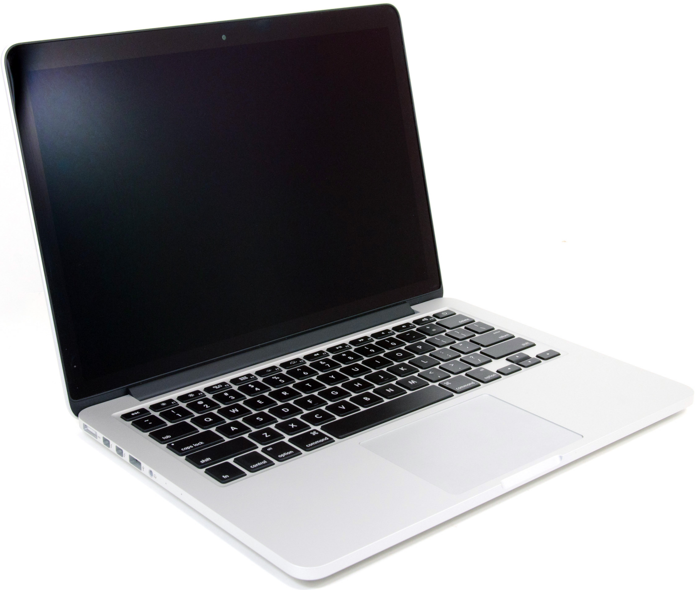
What does Research Computing do?

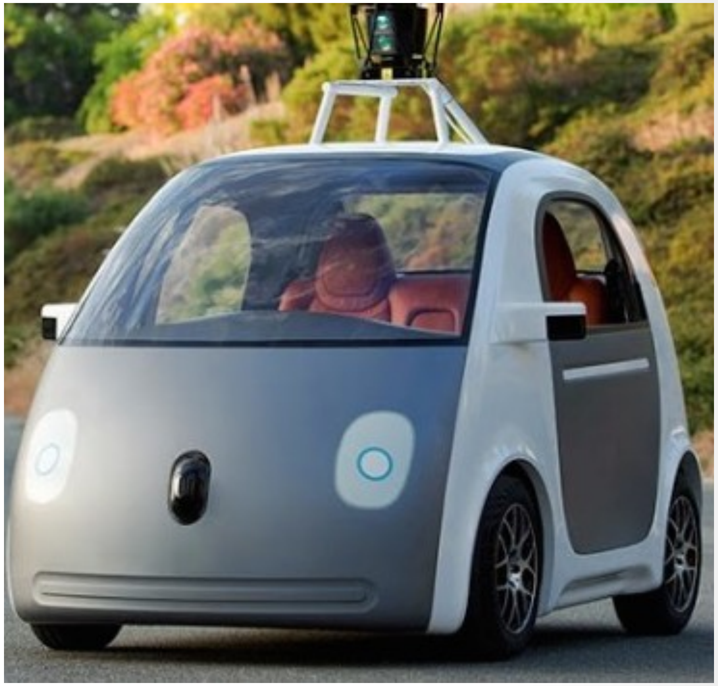
- We manage
  - Shared large scale compute resources
  - Large scale storage
  - High-speed network without firewalls – ScienceDMZ
  - Software and tools
- We provide
  - Consulting support for building scientific workflows on the RC platform
  - Training
  - Data management support in collaboration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

What Is a Supercomputer?

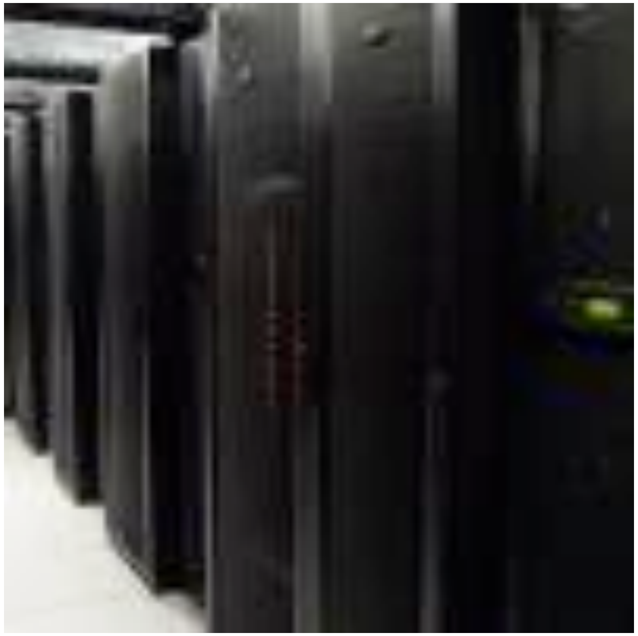
- A supercomputer is one large computer made up of many smaller computers and processors
- Each different computer is called a node
- Each node has processors/cores
  - Carry out the instructions of the computer
- With a supercomputer, all these different computers talk to each other through a communications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

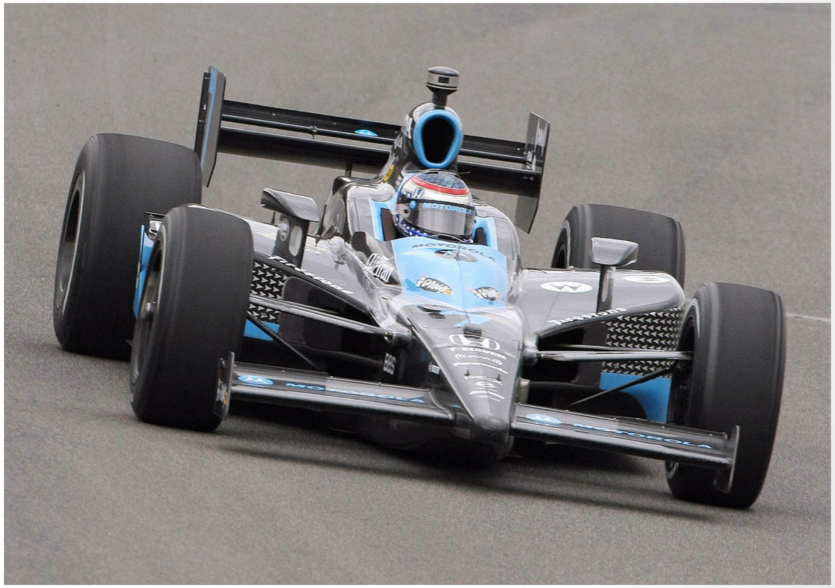
Computers and Cars - Analogy





Computers and Cars - Analogy





World’s Fastest Supercomputers

www.top500.org   June 2015

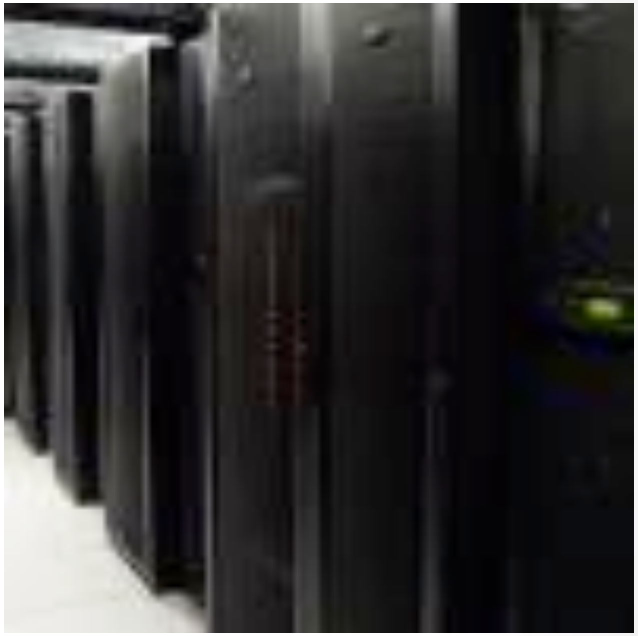
Rank	Site	Name	TeraFlops
1	National Super Computer Center (Guangzhou, China)	Tianhe-2	54902.4
2	Oak Ridge National Laboratory (United States)	Titan	27112.5
3	DOE/NNSA/LLNL (United States)	Sequoia	20132.7
4	RIKEN Advanced Institute for Computational Science (Japan)	K	11280.4
5	DOE/Argonne National Lab (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)

Next-Generation Supercomputer at CU-Boulder

- Funded via an NSF MRI grant awarded jointly to CU-Boulder and CSU
- $2M to CU and $700K to CSU … with matching funds the hardware budget is about $3.5M
- RFP has been published, vendor award by early December
- Installed and running late spring 2016

Next-Generation Supercomputer

- Expected performance about 450 TFLOPS (compared to about 170 for Janus)
- Compute nodes
  - Expect 24 real cores and 128 GB RAM
- 10 GPU/visualization nodes
  - 2x NVIDIA K80 GPUs
- 5 High-memory nodes
- 20 Xeon Phi (“Knight’s Landing”) nodes
- “Omni-Path” high-performance interconnect
- 1 PB of high-performance scratch storage

Next-Generation Supercomputer

- HPC is moving toward multi-core and many-core processors
- In order to see performance improvements, good parallel programming is important
- SIMD or vectorization can be even more important

Initial Steps to Use RC Systems

- Apply for an RC account
  - https://www.rc.colorado.edu/support/getting-started.html#account

- 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

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

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 3 more years

Submit Batch Job example

```
#!/bin/bash
#SBATCH –N 2                            #No. nodes
#SBATCH --ntasks-per-node=12            #No. cores
#SBATCH --time=1:00:00                  #Max walltime
#SBATCH --job-name=SLURMDemo            #Job name
#SBATCH --output=SLURMDemo.out          #Output file name
###SBATCH -A <account>                  #Allocation
###SBATCH --mail-type=end               #Send Email completion
###SBATCH --mail-user=<your@email>      #Email address

ml intel
ml openmpi/1.8.5

mpirun ./hello
```

Submit Batch Job example

- Have to make sure the slurm module is loaded!
- Submit the job, and specify the queue:

```
sbatch --qos janus-debug slurmSub.sh
```

  - Demonstrates that you can add slurm functions at the command line or in the bash script

- Check job status in the janus-debug queue:

```
squeue –q janus-debug
```

- Check output:

```
cat SLURMDemo.out
```

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

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

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