Introduction to Supercomputing with Janus

Shelley Knuth
shelley.knuth@colorado.edu

Peter Ruprecht
peter.ruprecht@colorado.edu

www.rc.colorado.edu

Outline

- Who is CU Research Computing?
- What is a supercomputer?
- Accessing resources
- General supercomputing information
  - Job submission
  - Storage types
  - Keeping the system happy
  - Using and installing software
  - With examples!

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

JANUS SUPERCOMPUTER AND OTHER COMPUTE RESOURCES

What Is a Supercomputer?

- CU's supercomputer is one large computer made up of many smaller computers and processors
- Each computer is called a node
- Each node has processors/cores
  - Carry out the instructions of the computer
- Within a supercomputer, all these different computers talk to each other through a communications network
  - On Janus – InfiniBand

Why Use a Supercomputer?

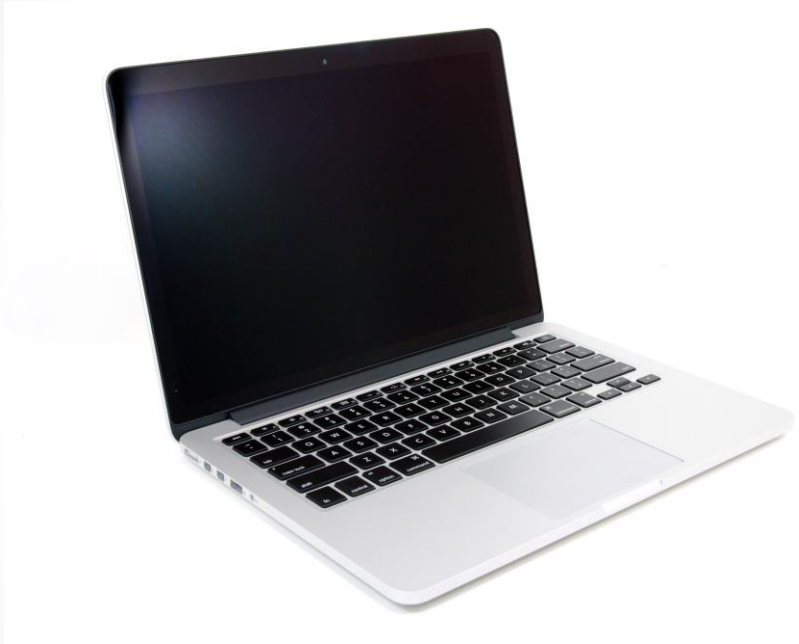
- Supercomputers give you the opportunity to solve problems that are too complex for the desktop
  - Might take hours, days, weeks, months, years
  - If you use a supercomputer, might only take minutes, hours, days, or weeks
- Useful for problems that require large amounts of memory

World's Fastest Supercomputers

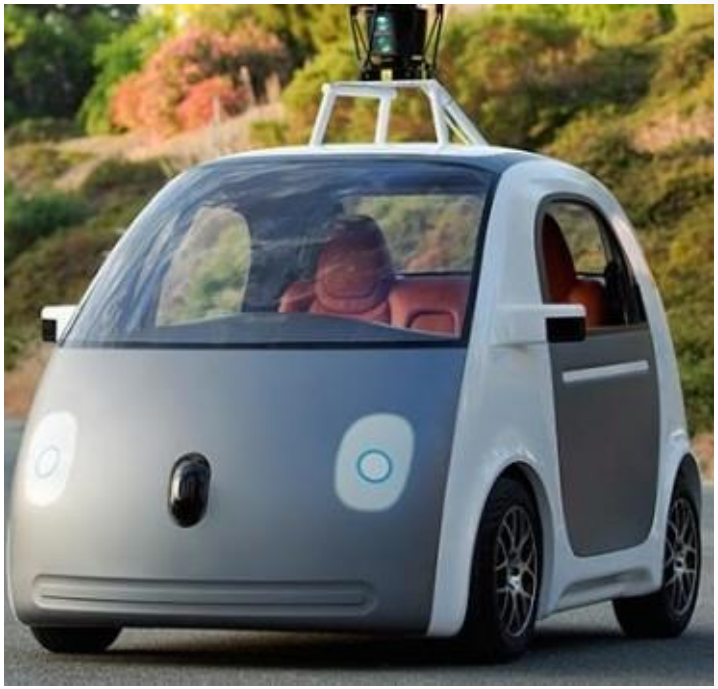
www.top500.org

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	Texas Advanced Computing Center (United States)	Stampede	8520.1
8	Forschungszentrum Juelich (Germany)	JUQUEEN	5872.0
9	DOE/NNSA/LLNL (United States)	Vulcan	5033.2
10	Government (Undisclosed) (United States)	Undisclosed	3143.5

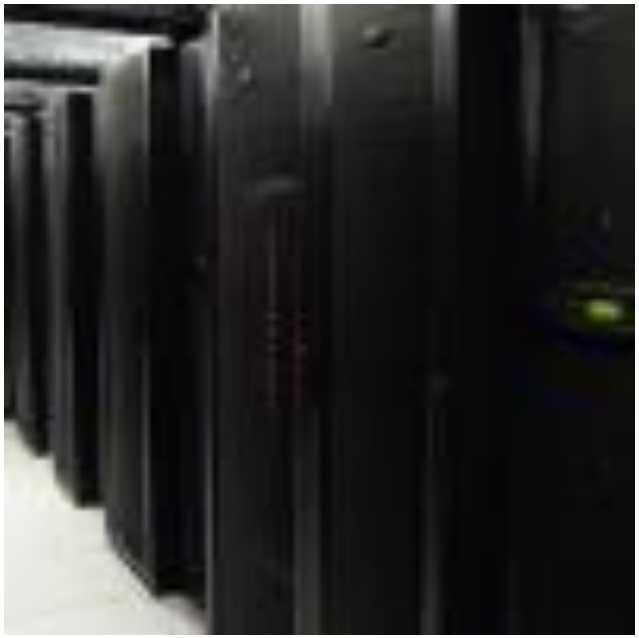
Computers and Cars - Analogy



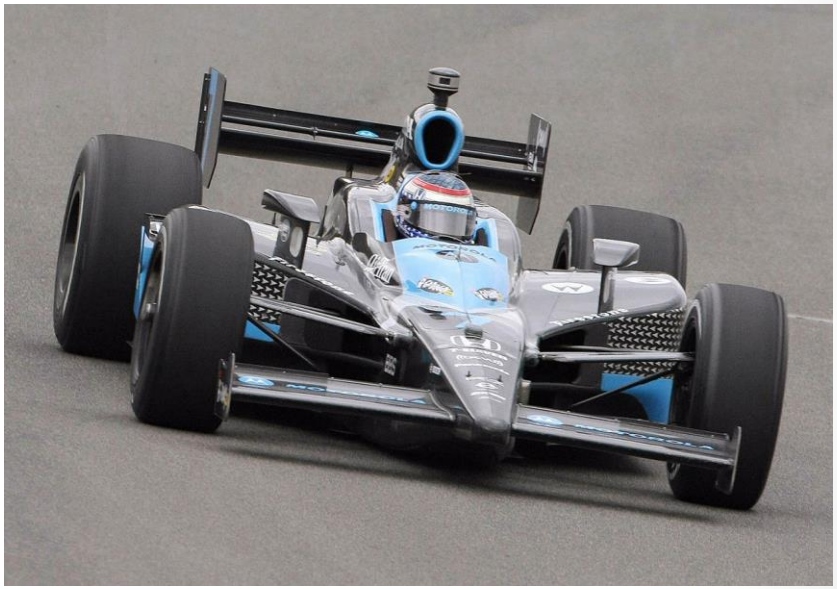
≈



Computers and Cars - Analogy

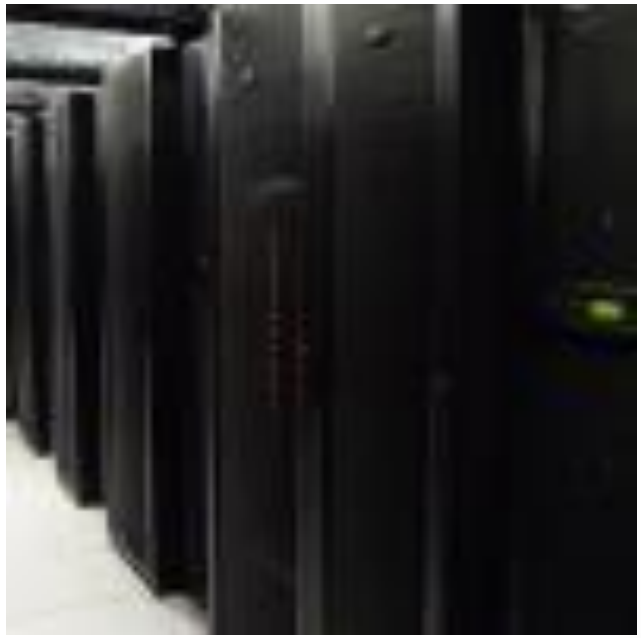


≈



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 to login.rc.colorado.edu
  - No heavy computation, interactive jobs, or long running processes
  - Script or code editing, minor compiling
  - Job submission
- Compile nodes
  - Compiling code
  - Job submission
- Compute/batch nodes
  - This is where jobs that are submitted through the scheduler run
  - Intended for heavy computation

ACCESSING AND UTILIZING RC RESOURCES

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

Logging In

- SSH to login.rc.colorado.edu
  - Use your RC username (same as IdentiKey) and One-Time Password
- From a login node, can SSH to a compile node - janus-compile1(2,3,4) - to build your programs
- All RC nodes use RedHat Enterprise Linux 6

Job Scheduling

- On supercomputer, jobs are scheduled rather than just run instantly at the command line
  - Several different scheduling software packages are in common use
  - Licensing, performance, and functionality issues have led us to choose Slurm
  - SLURM = Simple Linux Utility for Resource Management
    - Open source
    - Increasingly popular at other sites

More at https://www.rc.colorado.edu/support/examples/slurmtestjob

Running Jobs

- Do NOT compute on login or compile nodes
- Interactive jobs
  - Work interactively at the command line of a compute node
  - Command: salloc --qos=janus-debug
- Batch jobs
  - Submit a 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
  - sbatch --qos=<queue> file

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)
- normal
  - Default
  - Maximum wall time of 24 hours
- janus-long
  - Maximum wall time of 168 hours; limited number of nodes
- himem
- serial
- gpu

Parallel Computation

- Special software is required to run a calculation across more than one processor core or more than one node
- Sometimes the compiler can parallelize an algorithm, and frequently special libraries and functions can be used to abstract the parallelism
- OpenMP
  - For multiprocessing across cores in a single node
  - Most compilers can auto-parallelize with OpenMP
- MPI – Message Passing Interface
  - Can parallelize across multiple nodes
  - Libraries/functions compiled into executable
  - Several implementations (OpenMPI, mvapich, mpich, …)
  - Usually requires a fast interconnect (like InfiniBand)

Software

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

EXAMPLES

Login and Modules example

- **Log in:**
  - `ssh username@login.rc.colorado.edu`
  - `ssh janus-compile2`
- **List and load modules**
  - `module list`
  - `module avail`
  - `module load openmpi/openmpi-1.8.0_intel-13.0.0`
  - `module load slurm`
- **Compile parallel program**
  - `mpicc -o hello hello.c`

Submit Batch Job example

- **Batch Script:**

```
#!/bin/bash
#SBATCH -N 2                          # Number of requested nodes
#SBATCH --ntasks-per-node=12          # number of cores per node
#SBATCH --time=1:00:00                # Max walltime
#SBATCH --job-name=SLURMDemo          # Job submission name
#SBATCH --output=SLURMDemo.out        # Output file name
###SBATCH -A <account>                # Allocation
###SBATCH --mail-type=end             # Send Email on completion
###SBATCH --mail-user=<your@email> # Email address
module load openmpi/openmpi-1.8.0_intel-13.0.0
mpirun ./hello
```

- **Submit the job:**
  - `sbatch --qos janus-debug slurmSub.sh`

- **Check job status:**
  - `squeue -q janus-debug`
  - `cat SLURMDemo.out`

DATA STORAGE AND TRANSFER

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

Keeping Lustre Happy

- Janus Lustre is tuned for large parallel I/O operations.
- Creating, reading, writing, or removing many small files simultaneously can cause performance problems.
- Don’t put more than 10,000 files in a single directory.
- Avoid “ls –l” in a large directory.
- Avoid shell wildcard expansions (*) in large directories.

Data Sharing and Transfers

- ***Globus tools***: Globus Online and gridftp
  - https://www.globus.org
  - Easier external access without CU-RC account, especially for external collaborators
  - Endpoint: colorado#gridftp

- ***SSH***: scp, sftp, rsync
  - Adequate for smaller transfers

TRAINING, CONSULTING AND PARTNERSHIPS

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
  - http://researchcomputing.github.io
- 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

Thank you!

Questions?

www.rc.colorado.edu