

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/2017_July

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 or Omni-Path

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 2017

Rank	Site	Name	TeraFlops
1	National Supercomputing Center (Wuxi, China)	Sunway	93,014.6
2	National Super Computer Center (Guangzhou, China)	Tianhe-2	33,862.7
3	Swiss National Supercomputing Centre (Switzerland)	Piz Daint	19,590.0
4	DOE/SC/Oak Ridge National Laboratory (United States)	Titan	17,590.0
5	DOE/NNSA/LLNL (United States)	Sequoia	17,173.2
6	DOE/SC/LBNL/NERSC (United States)	Cori	14,014.7
7	Joint Center for Advanced High Performance Computing (Japan)	Oakforest-PACS	13,554.6
8	RIKEN Advanced Institute for Computational Science (Japan)	K	10,510.0
9	DOE/SC/Argonne National Lab (United States)	Mira	8,586.6
10	DOE/NNSA/LANL/SNL (United States)	Trinity	8,100.9

What Does It Mean to Be Fast?

- Titan can do 17 thousand 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 - Summit Supercomputer

- 475 compute nodes (Intel Xeon Haswell)
- 24 cores per node
- 11,400 total cores
- Omni-Path network
- 1.2 PB scratch storage
- GPFS File system



Additional Compute Resources

- 10 Graphics Processing Unit (GPU) Nodes
 - NVIDIA Tesla K80 (2/node)
- 5 High Memory Nodes
 - 2 TB of memory/node, 48 cores/node
- Phi Nodes (planned summer 2017)
 - 20 nodes
 - Intel Xeon Phi

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
 - Where you compile code
- Compute/batch nodes
 - This is where jobs that are submitted through the scheduler run
 - Intended for heavy computation

Storage Spaces

- **Home Directories**

- /home/\$USER
- Not for direct computation
- Small quota (RC: 2 GB)
- Backed up

- **\$PROJECT Space**

- /projects/\$USER
- Mid level quota (RC: 250 GB)
- Large file storage
- Backed up

- **Scratch Directory**

- /scratch/summit/\$USER
- 10 TB
 - Can ask for more if needed
- Files purged around 90 days (RC)

Jobs

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

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
 - Shared system
 - Jobs are put in a queue until resources are available
- Need software that will distribute the jobs appropriately and manage the resources
 - Simple Linux Utility for Resource Management (Slurm)
 - Keeps track of what nodes are busy/available, and what jobs are queued or running
 - Tells the resource manager when to run which job on the available resources

Partitions and 'Quality of Services'

- There are several ways to define where your job will run
- Partitions (basically a queue):
 - Resources/hardware
- QoS:
 - Tells what the limits or characteristics of a job should be
 - Maximum wall time
 - Number of nodes
- One partition might have multiple QoS
- A QoS might exist on multiple partitions

Partitions

- By default, jobs will run on the general compute (Haswell) nodes on RC systems
- Recommend specifying a partition
- Do this using the `-p` partition flag

Available Partitions (RC)

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 - [not currently available]	20	64	0

Quality of Service (RC)

QoS	Description	Maxwall	Max jobs/user	Max nodes/user
normal	Default QoS	Derived from partition	n/a	256
debug	For quick turnaround when testing	1 H	1	32
long	For jobs needing longer wall times	7 D	n/a	20
condo	For groups who have purchased Summit nodes	7 D	n/a	n/a

Allocations

- You will need an allocation to use resources
- I have an account – why do I need an allocation?
 - An account validates you are eligible to use resources
 - An allocation allows us to keep track of your use of the system
 - This is important because:
 - Making sure enough resources to accommodate all users
 - Helps for reporting

What is Fair Share?

- RC is moving to fair share
- Fair share scheduling uses a complex formula to determine priority in queue
- Looks at load for each user and each QOS and balances utilization to fairly share resources
 - Involves historical use by user plus how long job has been in the queue
- System will first look at weighted average utilization of user over last 4 weeks
- Then compare it to the fair share target percentage of a user

Fair Share Target Percentage

- The target percentage depends on your priority based on your project proposal
- Everyone not associated with a project shares a target percentage of 13% (20% of the CU fraction)
 - No guaranteed level per user
- If you are under (over) your target percentage (based on a 4 week average) your priority is increased (decreased)
- Reminder this all only impacts pending jobs
- If no other pending jobs and enough resources are available then your job will run regardless of your previous usage

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?

Software

- Common software is available to everyone on the systems
- Many groups use modules to manage software
 - You can load modules to prepare your environment for using software
 - Set any environment variables
 - Set environment so application can find appropriate libraries, etc.

Important Things to Know About Modules

- Some modules might require a specific hierarchy to load
 - For some modules, you may need to specify a specific version
 - For example, **module load R/3.3.0**
 - For other modules, you may be able to be more generic
 - For example, **module load matlab**
- Some modules may require you to first load other modules that they depend on
- To find dependencies for a module, type `module spider <package>`
- To find out what software is available, you can type **module load avail**
- To set up your environment to use a software package, type **module load <package>/<version>**

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

Summit Access

- For this tutorial, we will be using accounts on Summit
- In a terminal or Git Bash window, type the following:

```
ssh <username>@tutorial-login.rc.colorado.edu
```

Password:

What's Next?

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

Questions?

- Email rc-help@colorado.edu
- Link to survey on this topic:
<http://tinyurl.com/curc-survey16>

Speaker: Shelley Knuth

Title: What is a Supercomputer? July 2017 BSW

- Slides:
https://github.com/ResearchComputing/Final_Tutorials/tree/master/Basics_Supercomputing/2017_July