Introduction To HPCC Faculty Seminars in Research and Instructional Technology Dec 16, 2014

https://wiki.hpcc.msu.edu/x/JwJiAQ

Dirk Colbry

colbrydi@msu.edu

Director, High Performance Computing Center

Institute for Cyber-Enabled Research





All Day Agenda

- Introduction to HPCC
 - Introduction to iCER
 - The Seven Steps to using the HPCC
- Advanced HPCC, Doing more faster
 - Powertools
 - Pleasantly Parallel
 - Shared Memory Parallelization
 - Shared Network Parallelization





How this workshop works

- We are going to cover some basics. Hands on experience is the best for learning.
- When you get tired of listening to me talk, skip ahead to an exercise and give it a try.
- Exercises are denoted by the following icon in your notes:





Red and Green Flags

- Use the provided sticky notes to help me help you.
 - -NO Sticky = I am working
 - -Green = I am done and ready to move on
 - -Red = I am stuck and need more time and/or I could use some help





What is Advanced Computing Hardware?

- Anything more advanced than your desktop
- Local resources
 - Lab, Department, Institution (HPCC)
- National resources
 - NSF (XSEDE, Blue Waters), DOE (Jaguar),
 Others
- Commercial Resources (cloud computing)
 - Amazon, Azure, Liquid Web, Others





Why use Advanced Computing Hardware?

- Science takes too long
- Computation runs out of memory
- Needs licensed software
- Needs advanced interface (visualization/database)
- Lots of file i/o





Institute for Cyber Enabled Research

The Institute for Cyber-Enabled Research (iCER) at Michigan State University (MSU) was established to coordinate and support multidisciplinary resource for computation and computational sciences. The Center's goal is to enhance Michigan's national and international presence and competitive edge in disciplines and research thrusts that rely on advanced computing.



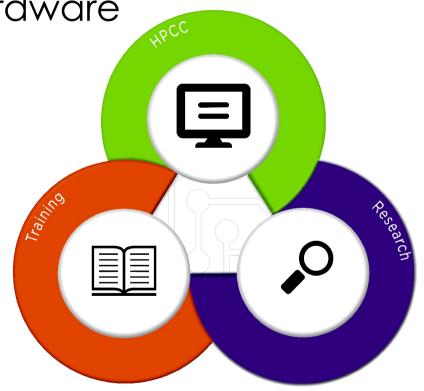


Research Resource

iCER is a research unit at MSU. We provide:

Advanced computing hardware

- •Software-as-a-service
- Training
- Consulting
- Proposal writing support







Upcoming Training

For course and registration details visit: http://icer.msu.edu/events





January

- Vim-R Lecture
- Software Carpentry
 - Introduction to Command Line (automating tasks using the Unix shell)
 - Introduction to Python: The Basics
 - Introduction to HPCC





February

- Vim-R Tutorial (for developers)
- Version Control & Git (how to track and share work efficiently)
- Compiling programs on the HPCC & MakeFiles
- Using Python for Data Analysis
- Unit Testing using xUnit-style framework.
- Advanced Topics in HPCC

<u>Domain Specific</u>

- Introduction to HPCC for Economists
- Bioinformatics Workshop





March

- Introduction to Command Line
- Advanced Shell Scripting
- Introduction to Python
- Introduction to HPCC
- Designing with Testing in Mind





April

- Introduction to Command Line
- Job Scheduling & Monitoring Tricks
- Debugging serial and parallel programs
- Visualizing your data
- Bioinformatics Workshop





+++ More Workshops

For course and registration details visit:

http://icer.msu.edu/events





Bigger Science

- The goal of iCER is <u>NOT</u>:
 - Kflops(floating point operations per second)
- Instead, the goal of iCER <u>IS</u>:
 - KSciences / second
- Doing More Science, Faster
 - Reducing the "Mean time to Science"
- iCER is designed to help researchers do their science and when appropriate scale them up to one of the national labs



HPC Systems

- Large Memory Nodes (up to 6TB!`)
- GPU Accelerated cluster (K20, M1060)
- PHI Accelerated cluster (5110p)
- Over 600 nodes, 7000 computing cores
- Access to high throughput condor cluster
- 363TB high speed parallel scratch file space
- 50GB replicated file spaces
- Access to large open-source software stack and specialized bioinformatics VMs







Software Stack

- Compiled open-source software stack
 - Close to 2000 titles!
- Optimized Math/Communications libraries
- Some commercial software available
 - E.g. Ansys, MATLAB (+many toolboxes), Stata,Gaussian, SAS







Buy-In Opportunities

- We will maintain your computers for you
- Researchers get exclusive use of their nodes within 4 hours of submitting a job
- Buy-in jobs will automatically overflow into the general resources.





2014 Cluster Buy-in

- intel14 Option 1 (20 core 64gb base node): \$3805.82
- intel14 Option 2 (20 core 256gb large memory node): \$5338.46
- intel14 Option 3 (20 core 128gb K20 GPU node): \$7899.15
- intel14 Option 4 (20 core 128gb Phi Node): \$9042.91
- intel14 Chassis: \$1216.44

More information: https://wiki.hpcc.msu.edu/x/dwH3





What if I want more?

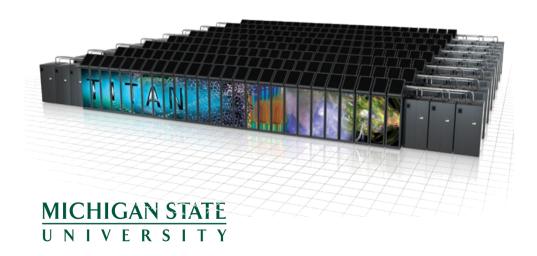


Extreme Science and Engineering Discovery Environment













Online Resources

- icer.msu.edu iCER Home
 - hpcc.msu.edu HPCC Home

• wiki.hpcc.msu.edu – HPCC User Wiki





Seven Steps to using the HPCC (The Basics)

http://www.softwarecarpentry.org/

https://www.youtube.com/user/icermsu





Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!





Accounts

- Pls must request accounts for students:
 - http://www.hpcc.msu.edu/request
- Each user has 50Gigs of backed-up personal hard drive space.
 - /mnt/home/username/
- Users have access to 363TB of high speed parallel scratch space.
 - /mnt/scratch/username/
- Shared group space is also available upon request.





Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!





Apple



- Run Terminal program
 - ssh already installedssh –X userid@hpcc.msu.edu
 - scp already installed
 scp ./mylocalfile userid@hpcc.msu.edu:~/mylocalfile
- May need to install Xquarts (mac X11 Server)
 - Installer should be on USB drive







MobaXterm

- Complete toolbox for remote computing:
 - Multi-tab terminal
 - X11 server
 - SSH
 - File transfer
 - More
- Open source



http://mobaxterm.mobatek.net/





Windows Software

- PuTTY:
 - http://www.chiark.greenend.org.uk/~sgtatham/putty/
- Xming:
 - http://www.straightrunning.com/XmingNotes/
- Xming install:
 - https://wiki.hpcc.msu.edu/x/swAk
- WinSCP:
 - http://winscp.net





Exercise: Portable HPCC



- Plug in your USB thumb drive
- Open the thumb drive folder and select
 - PortableApps
- You should see a new menu in your system tray for navigating



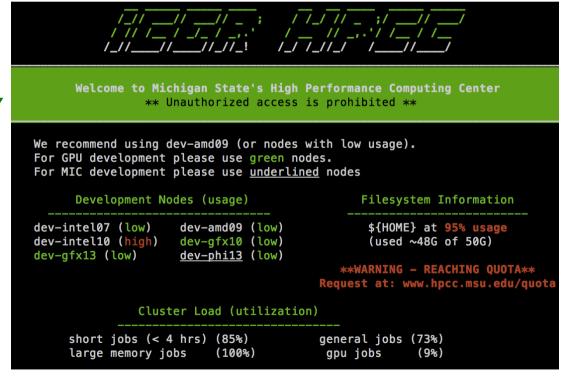




Exercise: Connect to HPCC

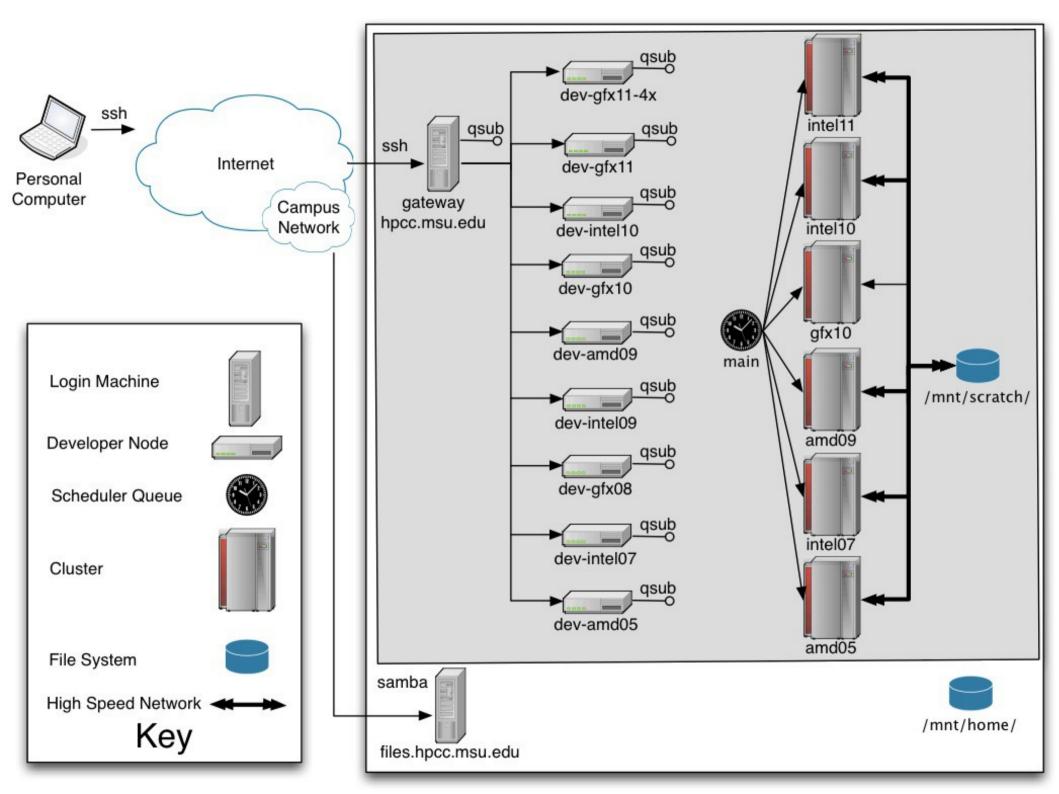


- Step 1: Log into gateway.hpcc.msu.edu
- Step 2: ssh into a dev node (developer node)
 - > ssh dev-intel10
- Step 3: execute a command
 - > echo "Hello world"









Command Line Interface

- Command Line Interface (CLI)
- Shell
 - Program to run Programs
- Bash (Bourne Again Shell)
- Use it because:
 - many tools only have command-line interfaces
 - allows you to combine tools in powerful new ways





Shell Navigation

Basic Navigation commands:

pwd	print working directory
cd	change working directory
Is	list directory

Use the following symbols to indicate special directories:

•	current directory
	parent directory
~	home directory
-	previous directory





Exercise – Shell Navigation



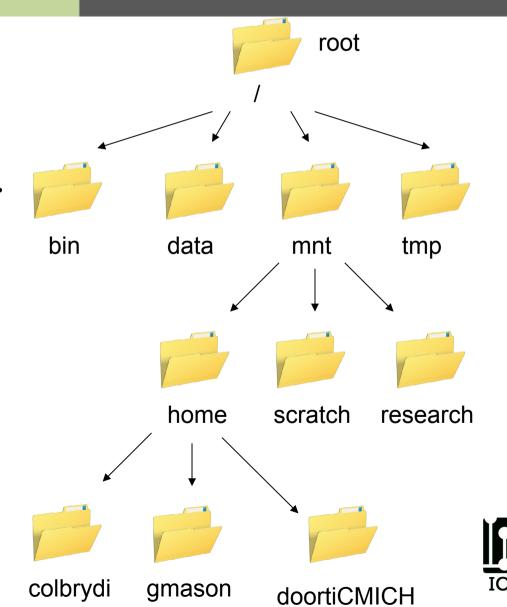
- Show the path to the current directory
 - > pwd
- Change to the scratch directory
 - >cd /mnt/scratch/
- List the contents of the current directory:
 - >1s
- Change back to home
 - >cd ~





Directories

/mnt/home/colbrydi





Man Pages

- "Manual" pages.
- Type "man" and then the command name
- Example:

>man qsub

Use "q" key to quick out of the man program





Example: File Manipulation



Try Commands

mkdi r	make directory
ср	copy file
cat	display contents of text file
r m	remove file

- See the contents of your ".bashrc" file
 - > cat .bashrc
- Make a directory called "hpccworkshop", change to that directory and list the contents.
 - > mkdir hpccworkshop
 - > cd ./hpccworkshop





Available Software

- Center Supported Development Software
 - Intel compilers, openmp, openmpi, mvapich, totalview, mkl, pathscale, gnu, ...
- Center Supported Research Software
 - MATLAB, R, fluent, abaqus, HEEDS, amber, blast, ls-dyna, starp...
- Customer Software
 - gromacs, cmake, cuda, imagemagick, java, openmm, siesta...
 - For a more up to date list, see the documentation wiki:
 - http://wiki.hpcc.msu.edu/





Module System

- To maximize the different types of software and system configurations that are available to the users, HPCC uses a Module system
- Key Commands
 - module avail show available modules
 - module list list currently loaded modules
 - module load modulename load a module
 - module unload modulename unload a module
 - module spider keyword Search modules for a keyword





Exercise – Module



- List loaded modules
 - >module list
- Show available modules:
 - >module avail
- Try an example (Shouldn't work):
 - >powertools





Exercise: getexample



- Load a newly available module:
 - >module load powertools
- Show powertools (should work now):
 - >powertools
- Run the "getexample" powertool
 - >getexample
- Download the helloMPI example
 - >getexample helloworld





Standard in/out/err and piping

- You can redirect the output of a program to a file using ">" greater than character:
 - myprogram > output.txt
- You can also cause the output of the program to be the input of another program using the "|" pipe character:
 - myprogram | myotherprogram





Exercise: Redirection and Piping



Change to the helloworld directory:

```
> cd ~/hpccworkshop/helloworld
> ls -la
```

Redirect the output of the Is command:

```
> ls -la > numOfLines
> cat numOfLines
```

Pi pe Commands toget her

```
> wc -1 * | sort -n
```

Easy command to calculate the number of lines of code in your programs





Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!





SCP/SFTP –Secure File transfer

- WinSCP for Windows
- Command-line "scp" and "sftp" on Linux

scp ./mylocalfile username@hpcc.msu.edu:~/folder/

- Many other scp and sftp clients out there as well
- Functions over SSHv2 protocol, very secure





Exercise: Transfer a file (1/3)

- Make a file called minlines using notepad++ on your thumb drive
- Put in the following lines:

```
#!/bin/bash
echo "smallest file"
wc -1 * | sort -n | head -1
```

Save the file as minlines (or minlines.txt)







Exercise: Transfer a file (2/3)

- On Windows, Open WinSCP on your thumb drive and Drag and drop using the interface.
- On Apple (Linux):
 - Open the "terminal" program and cd to your working directory
 - Use the following transfer command scp ./minlines username@hpcc.msu.edu:~/hpccworkshop





Exercise: Transfer a file (3/3)

- Go back to the hpcc
- Change to the helloMPI directory
 - cd ~/hpccworkshop/helloworld
- and view the file
 - cat minlines.txt
- Try to run the file as a script
 - ./minlines.txt
- Should get a permission denied error. Why?



File Permissions









_		user	group	all
	read			X
	write		X	X
-	execute	X	X	X





Permissions

Common Commands

chmod	Change permissions	
	(change mode)	
Is -a -I	List all long	
	(including permissions)	





Example: permissions



Show current file permissions

```
> ls -la
```

- Make the minlines file executable to the user
 - > chmod u+x minlines
- Check permissions again
 - > ls -la
- Now you can run minlines as a command
 - > ./minlines





Environment Variables

- Scripts also let you use environment variables
- These variables can be used by your script or program
- Use "export" and = to set a variable
- Use the \$ and {} to display the contents of a variable





Example: Environment Variables



- Display all environment variables
 - >env
- Display specific environment variable
 - >echo \${MACHTYPE}
- Make a new variable
 - > export MYVAR="Hello World"
- Use your variable
 - >echo \${MYVAR}



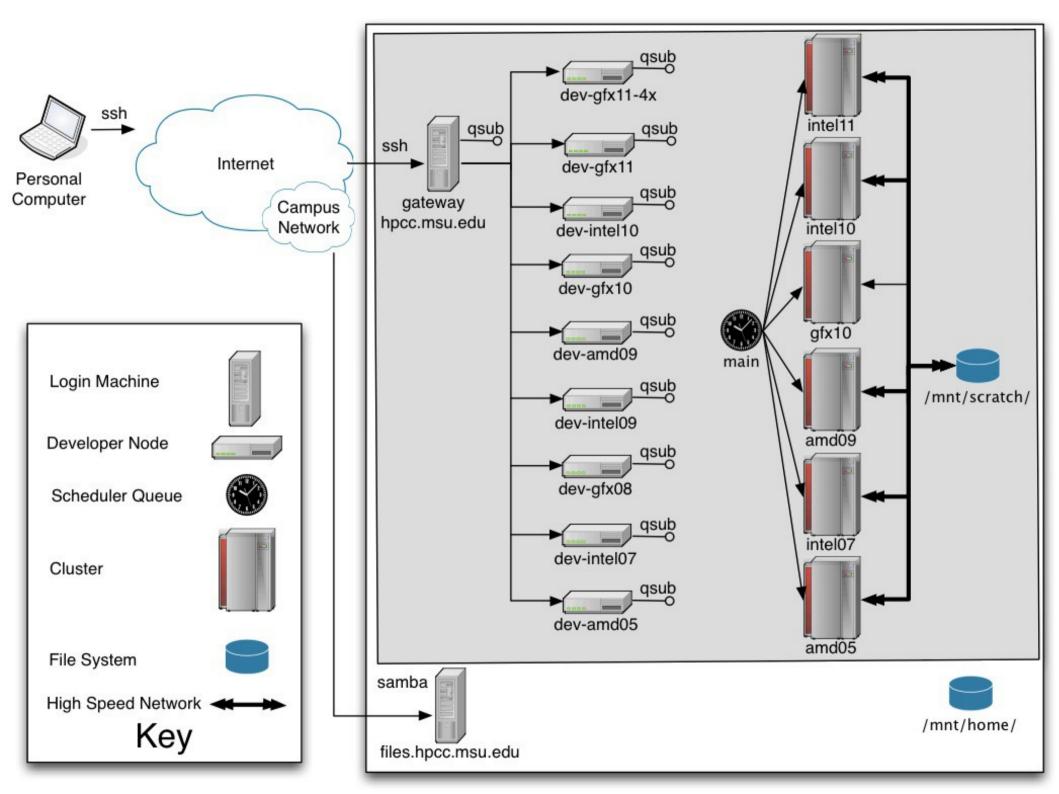


Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!







Running Jobs on the HPC

- Submission scripts are used to run jobs on the cluster
- The developer (dev) nodes are used to compile, test and debug programs





Advantages of running Interactively

- You do not need to write a submission script
- You do not need to wait in the queue
- You can provide input to and get feedback from your programs as they are running





Disadvantages of running Interactively

- All the resources on developer nodes are shared between all users.
- Any single process is limited to 2 hours of cpu time.
 If a process runs longer than 2 hours it will be killed.
- Programs that overutilize the resources on a developer node (preventing other to use the system) can be killed without warning.





Developer Nodes

Name	Cores	Memory	Accelerators	Notes
dev-intel07	8	8GB	-	
dev-gfx10	4	18GB	2 x M1060	Nvidia Graphics Node
dev-intel10	8	24GB	-	
dev-intel14	20	64GB	-	
dev-intel14-phi	20	128GB	2 x Phi	Xeon Phi Node
dev-intel14-k20	20	128GB	2 x K20	Nvidia Graphics Node





Compilers

- By default we use the gnu compilers. However, lots of other compilers are available including Intel and Portland compilers.
- The module system always sets environment variables such that you can easily test with other compilers.
 - -\${CC}
 - \${FC}
 - Etc.





Exercise: Compile Code



 Make sure you are in the helloworld directory:

```
>pwd
```

Run the gcc compilers:

```
>${CC} -03 -o hello hello.c
```

Run the program:





Running in the background

- You can run a program in the background by typing an "&" after the command.
- You can make a program keep running even after you log out of your ssh session by using "nohup command"
- You can run an entire session in the background even if you log in and out of your ssh session by using the "screen" or "tmux" commands
- All three of these options are common to linux and tutorials can be found online





CLI vs GUI

CLI – Command Line Interface

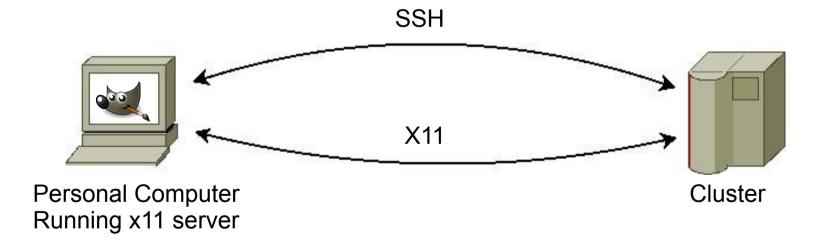
• GUI – Graphical User Interface





What is X11?

 Method for running Graphical User Interface (GUI) across a network connection.







What is needed for X11

- X11 server running on your personal computer
- SSH connection with X11 enabled
- Fast network connection
 - Preferably on campus





Graphical User Interface

- X11 Windows: Install Xming
 - Installation instructions at: https://wiki.hpcc.msu.edu/x/swAk
- ssh -X username@hpc.msu.edu
- Turn on x11 forwarding

Note: Mac Lion Users should use XQuart





Remote Desktop Gateway

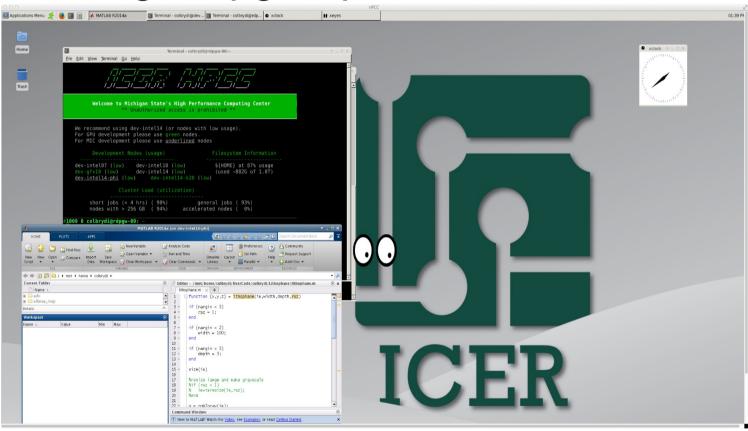
- New service
- Microsoft protocol
- Alternative to using X11
- Works off campus
- Windows (built in)
- Apple (download windows RDP client from app store)





Remote Desktop Protocal

Connect using: rdpgw.hpcc.msu.edu











Try one of the following Commands

xeyes	Test X11
firefo x	Web browser

- >xeyes
- >firefox &
- >ps <- Find the process ID #### for firefox
- >kill ####





Programs that can use X11

- R statistical computing and graphics
- firefox Web browser
- totalview C/C++/fortran debugger
- gedit, gvim, emacs Text editors
- And others...



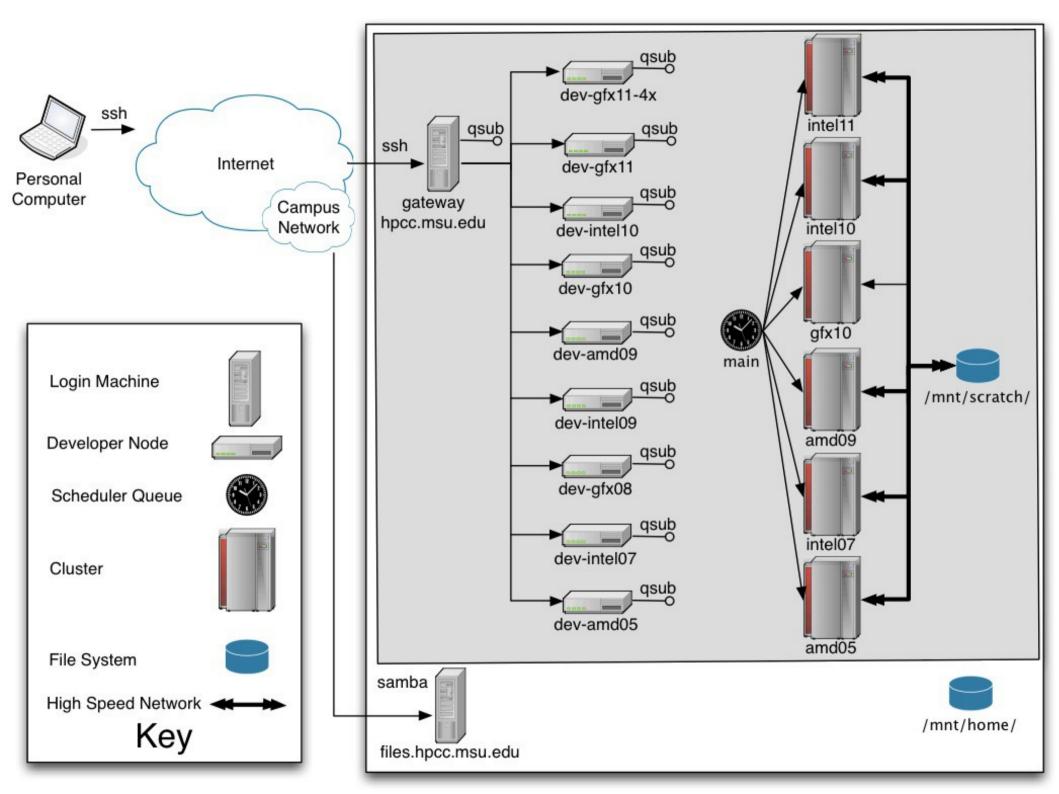


Steps in Using the HPCC

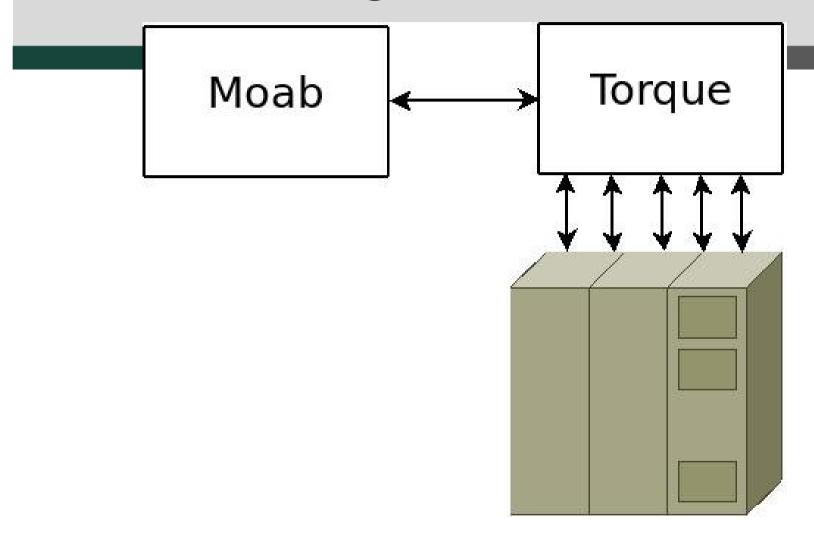
- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!







Resource Manager and scheduler



Not First In First Out!!





Schedulers vs. Resource Managers

- Scheduler (Moab)
 - Tracks and assigns
 - Memory
 - CPUs
 - Disk space
 - Software Licenses
 - Power / environment
 - Network

- Resource Manager
 (PBS/Torque)
 - Hold jobs for execution
 - Put the jobs on the nodes
 - Monitor the jobs and nodes





Common Commands

- qsub <Submission script>
 - Submit a job to the queue
- qdel <JOB ID>
 - Delete a job from the queue
- showq -u <USERNAME>
 - Show the current job queue
- checkjob <JOB ID>
 - Check the status of the current job
- showstart –e all <JOB ID>
 - Show the estimated start time of the job





Submission Script

- 1.List of required resources
- 2.All command line instructions needed to run the computation





Typical Submission Script

Define Shell Shell Comment #!/bin/bash -login -l walltime=10:00:00, mem=3Gb, nodes=10:ppn=1 #PBS -j oe cd \${PBS_O_WORKDIR} ./myprogram -my input arguments qstat -f \${PBS_JOBID} Resource Requests Shell Commands Special Environment Variables

Example: Submit a job



- Go to the top helloworld directory
 >cd ~/hpccworkshop/helloworld
- Create a simple submission script
 >nano hello.qsub
- See next slide for what to type...









```
#!/bin/bash -login
#PBS -1 walltime=00:01:00
#PBS -1 nodes=1:ppn=1, feature=gbe
cd ${PBS O WORKDIR}
./hello
qstat -f ${PBS JOBID}
```

Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!





Submitting a job

- qsub –arguments <Submission Script>
 - Returns the job ID. Typically looks like the following:
 - 5945571.cmgr01

Time to job completion

UNIVERSITY





Example: Submit a job, cont.



- Submit the file to the queue
 - >qsub hello.qsub
- Record jobid number (######) and wait at most 30 seconds
- Check the status of the queue
 - >showq





Example: Monitor a job



- Submit the file to the queue:
 - >qstat -f ######
- When will a job start:
 - >showstart -e all ######





Scheduling Priorities

- Jobs that use more resources get higher priority (because these are hard to schedule)
- Smaller jobs are backfilled to fit in the holes created by the bigger jobs
- Eligible jobs acquire more priority as they sit in the queue
- Jobs can be in three basic states:
- Blocked, eligible or running

NIVERSITY



Current Cluster Resources

Year	Name	Description	ppn	Memory	Nodes	Total Cores
2007	intel07	Quad-core 2.3GHz Intel Xeon E5345	8	8GB	126	1008
2010	gfx10	NVIDIA CUDA Node (no IB)	8	18GB	32	256
2010	intel10	Intel Xeon E5620 (2.40 GHz)	8	24GB	191	1528
2011	intel11	Intel Xeon 2.66 GHz E7-8837	32	512GB	2	64
			32	1TB	1	32
			64	2TB	2	128
2014	intel14	Intel Xeon E5-2670 v2 (2.6 GHz)	20	64GB	128	2560
			20	256GB	24	480
		2 NVIDIA K20 GPUs	20	128GB	40	800
		2 Xeon Phi 5110P	20	128GB	28	560
2014	Intel14-XL	Intel Xeon E7-8857 v2 (3 GHz)	48	1-3TB	5	240
			96	6 TB	1	96
	Total				580	7752

System Limitations

- Scheduling
 - 5 eligible jobs at a time
 - 520 running cores
 - 1000 submitted jobs
- Resources
 - 1 week of walltime
 - 520 cores (nodes * ppn)
 - ppn=64
 - 2TB memory on a single core
 - -~200 GB Hard Drive





Job completion

- By default the job will automatically generate two files when it completes:
 - Standard Output:
 - Ex: jobname.o5945571
 - Standard Error:
 - Ex: jobname.e5945571
- You can combine these files if you add the join option in your submission script:
 - "#PBS -j oe"

UNIVERSITY

- You can change the output file name
 - #PBS -o /mnt/home/netid/myoutputfile.txt



Other Job Properties

- resources (-I)
 - Walltime, memory, nodes, processor, network, etc.
- #PBS –I feature=gpgpu,gbe
- #PBS –l nodes=2:ppn=8:gpu=2
- #PBS –I mem=16gb
- Email address (-M)
 - Ex: #PBS -M colbrydi@msu.edu
- Email Options (-m)
 - Ex: #PBS –m abe

Many others, see the wiki: http://wiki.hpcc.msu.edu/





Advanced Environment Variables

- The scheduler adds a number of environment variables that you can use in your script:
 - PBS_JOBID
 - The job number for the current job.
 - PBS_O_WORKDIR
 - The original working directory which the job was submitted

Ex:

```
mkdir ${PBS_O_WORKDIR}/${PBS_JOBID}
```



Steps in Using the HPCC

- 1. Get an account
- 2. Install needed software (SSH, SCP, X11)
- 3. Transfer input files and source code
- 4. Compile/Test programs on a developer node
- 5. Write a submission script
- 6. Submit the job
- 7. Get your results and write a paper!!





Getting Help

- Documentation and User Manual wiki.hpcc.msu.edu
- Contact HPCC and iCER Staff for:
 - Reporting System Problems
 - HPC Program writing/debugging Consultation
 - Help with HPC grant writing
 - System Requests
 - Other General Questions
- Primary form of contact http://contact.icer.msu.edu/
- HPCC Request tracking system rt.hpcc.msu.edu
- HPCC Phone (517) 353-9309
- HPCC Office 1400 PBS
- Open Office Hours 1pm Monday (BPS1440)



