Introduction To HPCC
Jan 3, 2013

https://wiki.hpcc.msu.edu/x/65Ph

Dirk Colbry
colbrydi@msu.edu
Research Specialist
Institute for Cyber-Enabled Research

CMU / MSU Partnership

- CMU users are HPCC Buy-in User

cmichhelp@hpcc.msu.edu

Agenda

- Introduction to HPCC
  - Introduction to iCER
  - Connecting to HPC
  - Interactive Computing
  - Using the Batch Scheduler
- Advanced HPCC, Doing more faster
  - Pleasantly Parallel
  - Shared Memory Parallelization
  - Shared Network Parallelization

How this workshop works

- We are going to cover some basics. Lots of hands on examples.
- 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:

- Raise your hand if you have questions. I have helpers.

Science and Computation

- Pillars of Science:
  - Theory
  - Experimentation
  - Simulation
  - (Big) Data

The End of Science?

The quest for knowledge used to begin with grand theories. Now it begins with massive amounts of data. Welcome to the Petabyte Age.

Wired Magazine, 2009

MICHIGAN STATE UNIVERSITY

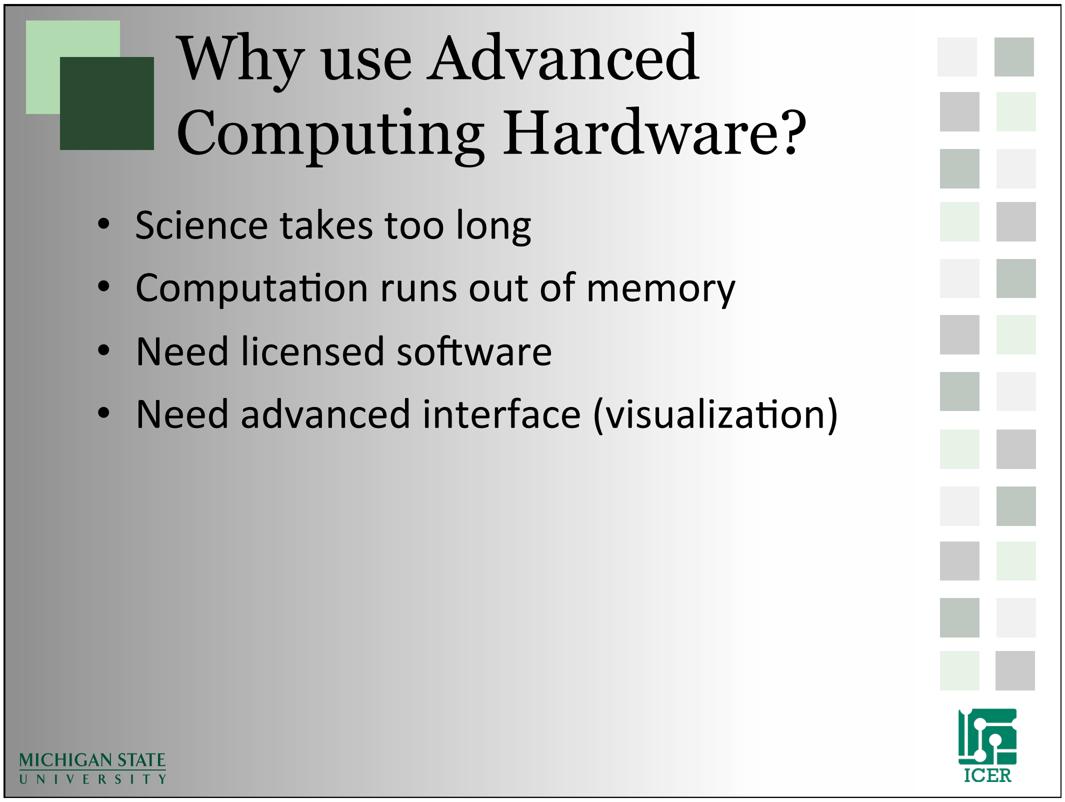
ICER

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

MICHIGAN STATE UNIVERSITY

ICER

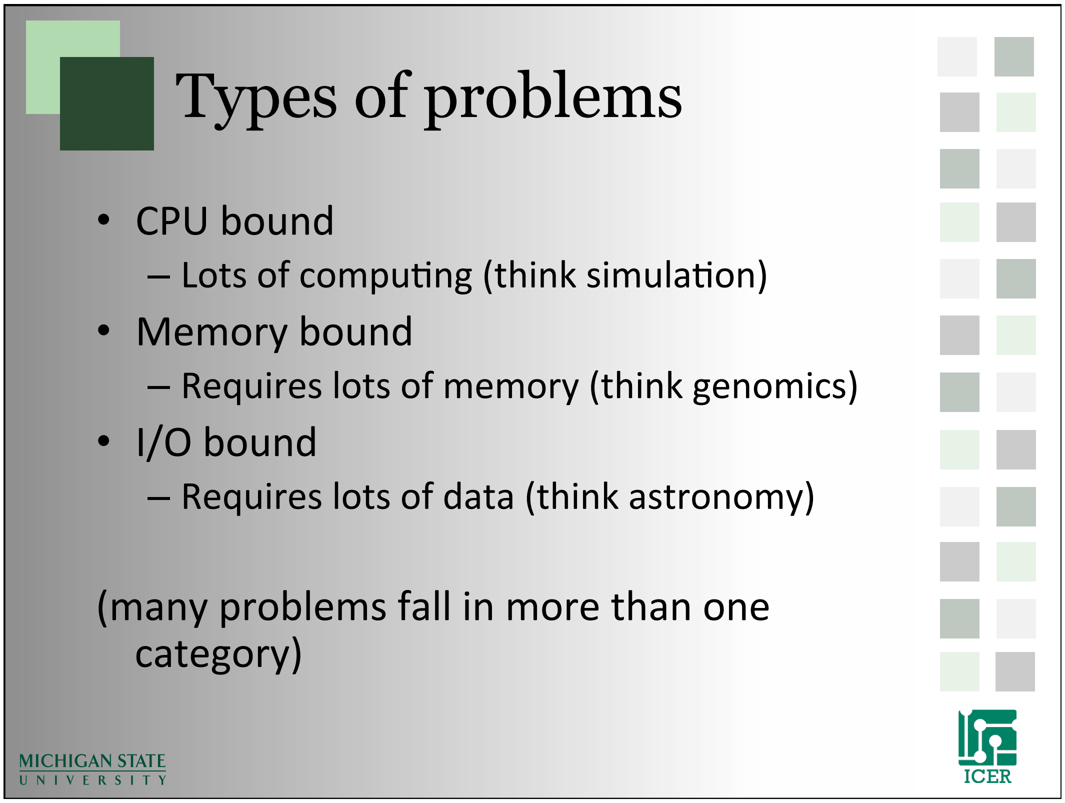


Why use Advanced Computing Hardware?

- Science takes too long
- Computation runs out of memory
- Need licensed software
- Need advanced interface (visualization)

MICHIGAN STATE UNIVERSITY

ICER



Types of problems

- CPU bound
  – Lots of computing (think simulation)
- Memory bound
  – Requires lots of memory (think genomics)
- I/O bound
  – Requires lots of data (think astronomy)

(many problems fall in more than one category)

MICHIGAN STATE UNIVERSITY

ICER

Types of Systems

Local Resources

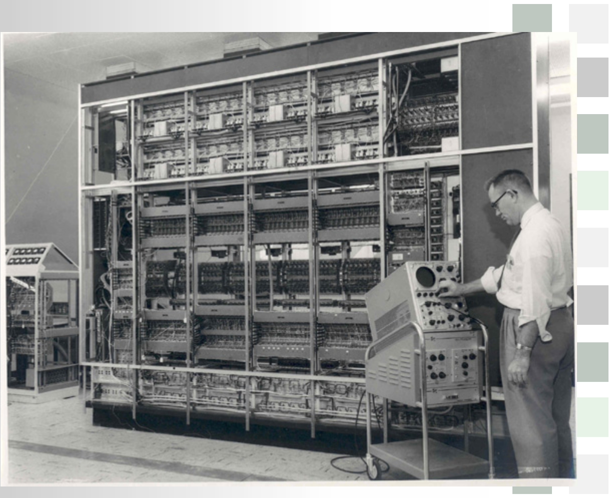
- Special Use Systems
  - SMP – Symmetric Multiprocessor (fat node)
  - GPGPU (General Purpose Graphics Processing Unit)
- Typical HPC cluster
  - Commodity computers
  - High speed backbone
  - High speed network storage

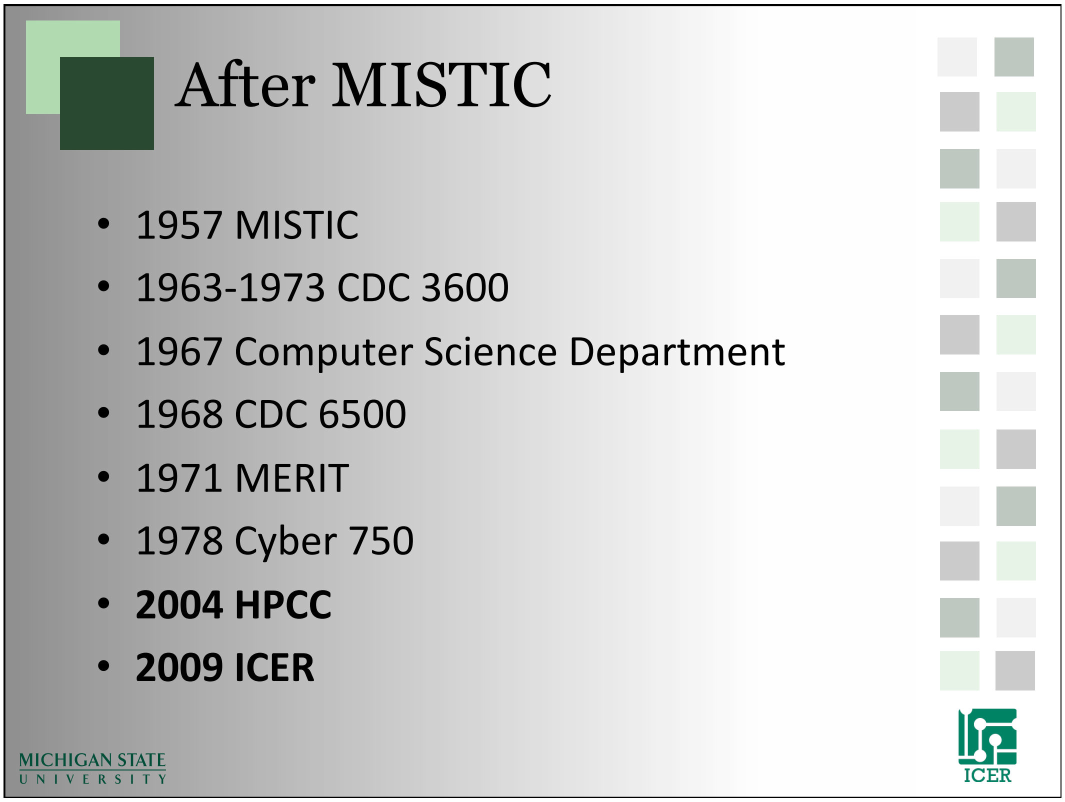
National and Commercial

- Advanced HPC
  - Specialty hardware
  - High speed backbone
  - High speed storage
- Grid
  - Many HPC systems linked together by high speed network
- Cloud
  - Lots of definitions
  - Typically refers to computing as a service using highly flexible virtual machines

1957 MISTIC Mainframe

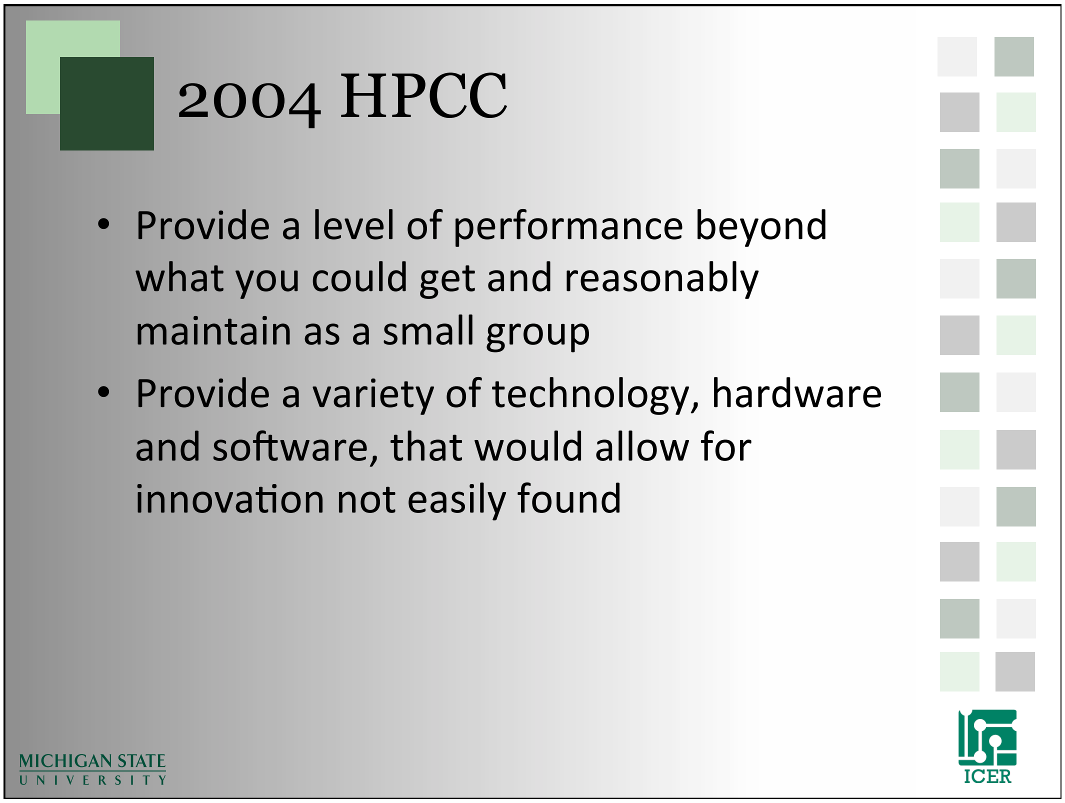
- MSU's first mainframe
- Hand built by grad students
  - Dick Reid
  - Glen Keeney





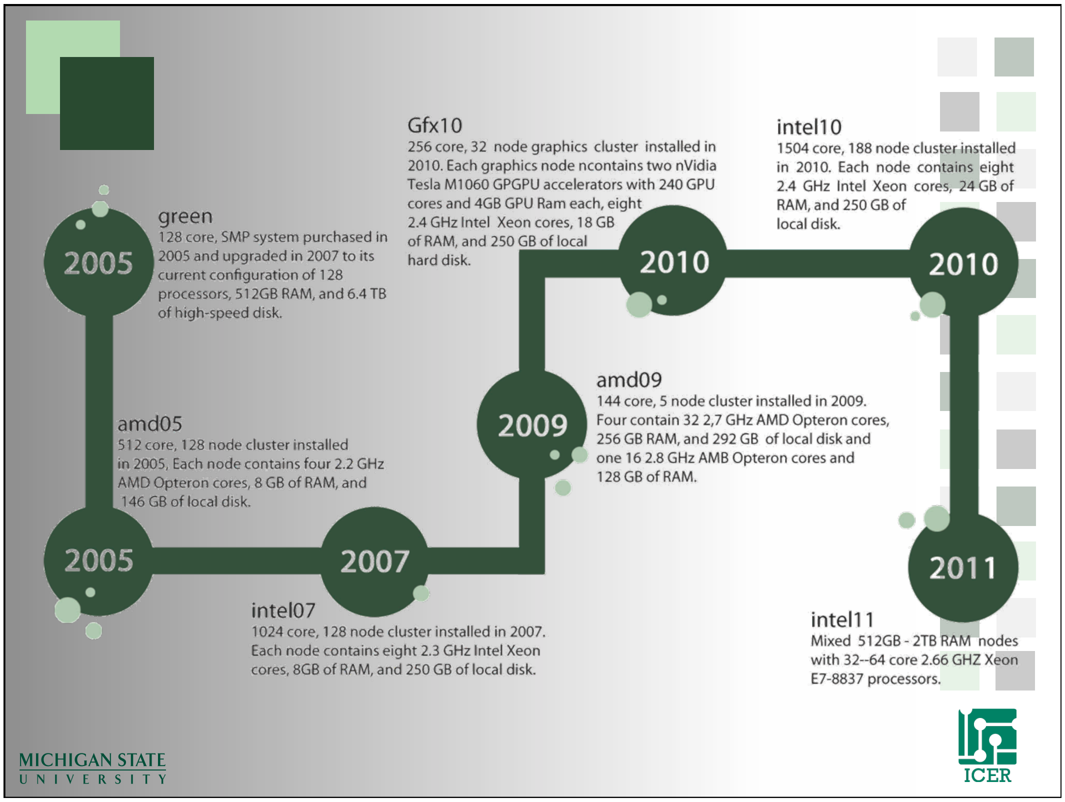
After MISTIC

- 1957 MISTIC
- 1963-1973 CDC 3600
- 1967 Computer Science Department
- 1968 CDC 6500
- 1971 MERIT
- 1978 Cyber 750
- **2004 HPCC**
- **2009 ICER**



2004 HPCC

- Provide a level of performance beyond what you could get and reasonably maintain as a small group
- Provide a variety of technology, hardware and software, that would allow for innovation not easily found



green
128 core, SMP system purchased in 2005 and upgraded in 2007 to its current configuration of 128 processors, 512GB RAM, and 6.4 TB of high-speed disk.

amd05
512 core, 128 node cluster installed in 2005, Each node contains four 2.2 GHz AMD Opteron cores, 8 GB of RAM, and 146 GB of local disk.

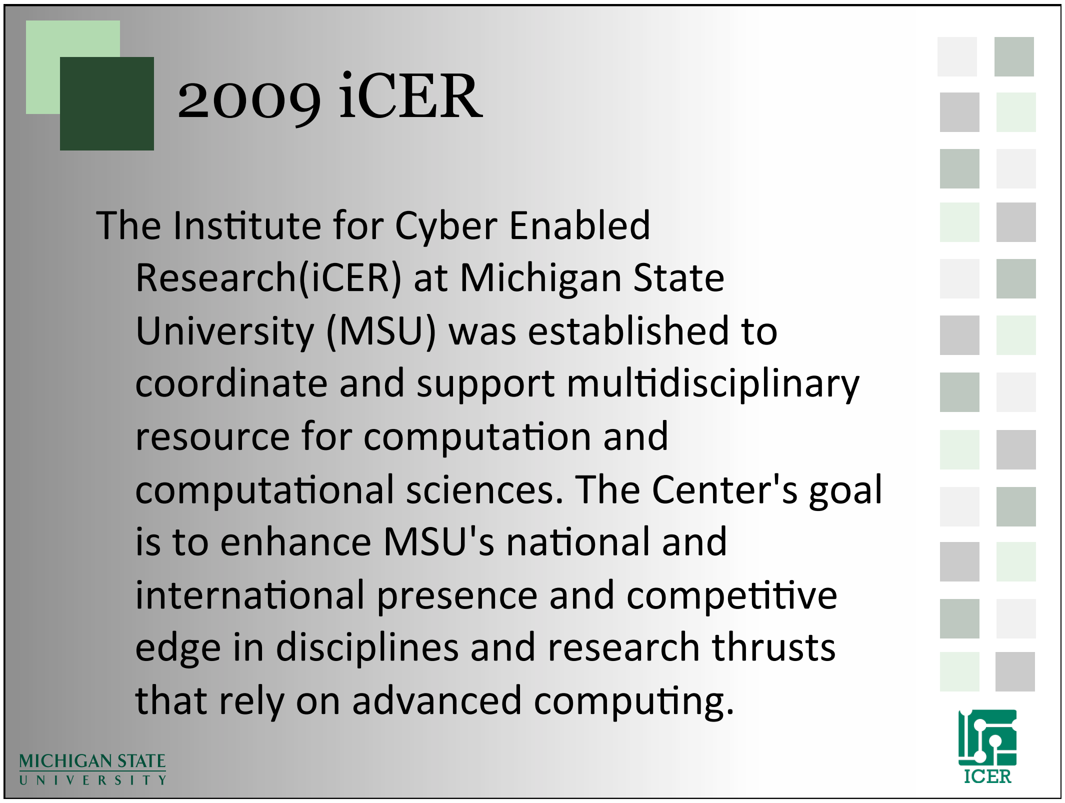
intel07
1024 core, 128 node cluster installed in 2007. Each node contains eight 2.3 GHz Intel Xeon cores, 8GB of RAM, and 250 GB of local disk.

amd09
144 core, 5 node cluster installed in 2009. Four contain 32 2,7 GHz AMD Opteron cores, 256 GB RAM, and 292 GB of local disk and one 16 2.8 GHz AMB Opteron cores and 128 GB of RAM.

Gfx10
256 core, 32 node graphics cluster installed in 2010. Each graphics node ncontains two nVidia Tesla M1060 GPGPU accelerators with 240 GPU cores and 4GB GPU Ram each, eight 2.4 GHz Intel Xeon cores, 18 GB of RAM, and 250 GB of local hard disk.

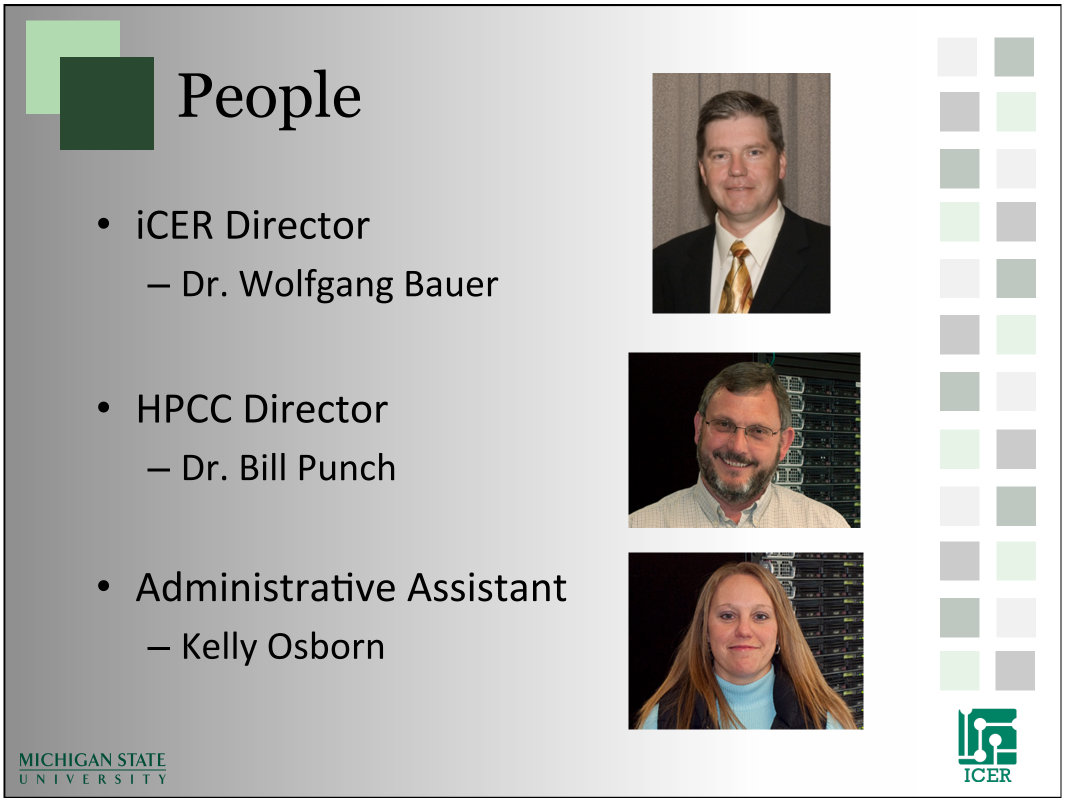
intel10
1504 core, 188 node cluster installed in 2010. Each node contains eight 2.4 GHz Intel Xeon cores, 24 GB of RAM, and 250 GB of local disk.

intel11
Mixed 512GB - 2TB RAM nodes with 32--64 core 2.66 GHZ Xeon E7-8837 processors.



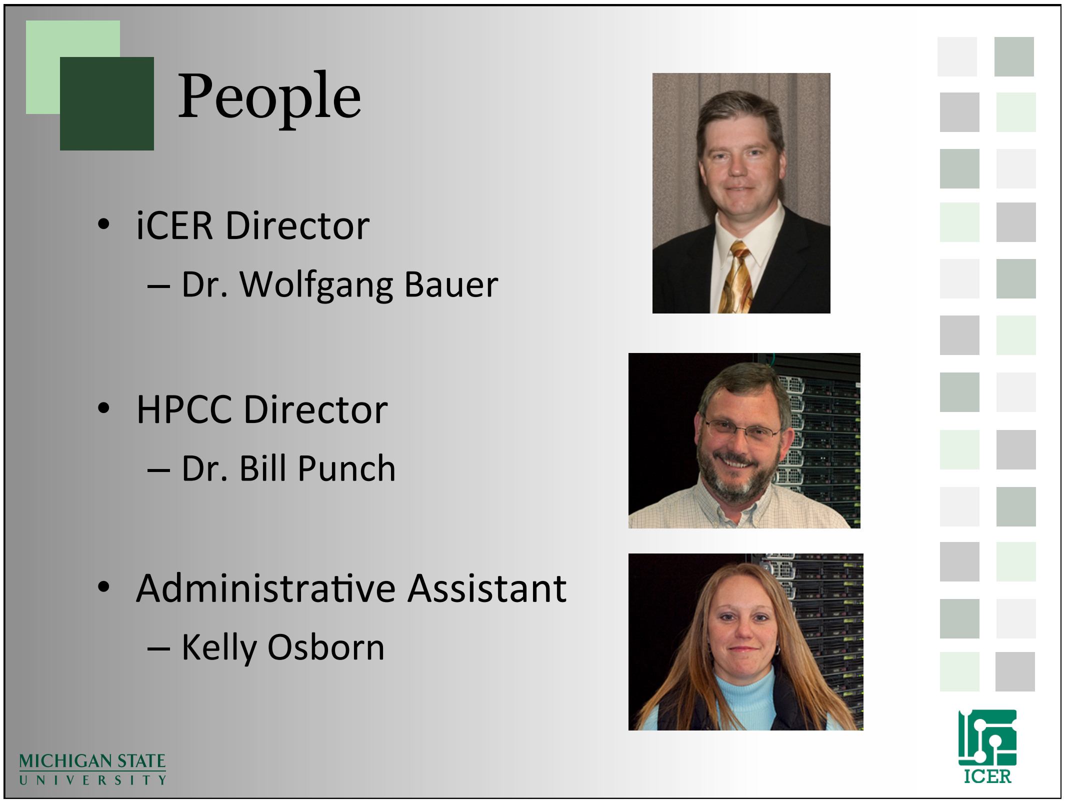
2009 iCER

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 MSU's national and international presence and competitive edge in disciplines and research thrusts that rely on advanced computing.



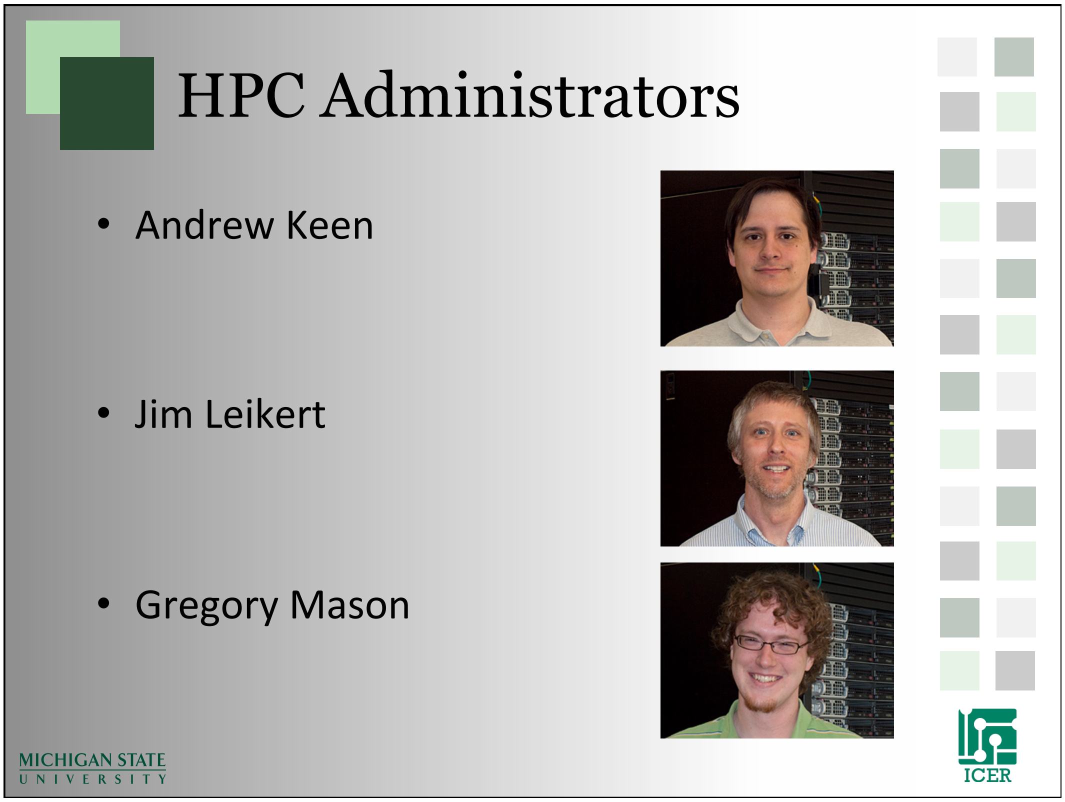
People

- iCER Director
  - Dr. Wolfgang Bauer
- HPCC Director
  - Dr. Bill Punch
- Administrative Assistant
  - Kelly Osborn



People

- iCER Director
  - Dr. Wolfgang Bauer
- HPCC Director
  - Dr. Bill Punch
- Administrative Assistant
  - Kelly Osborn

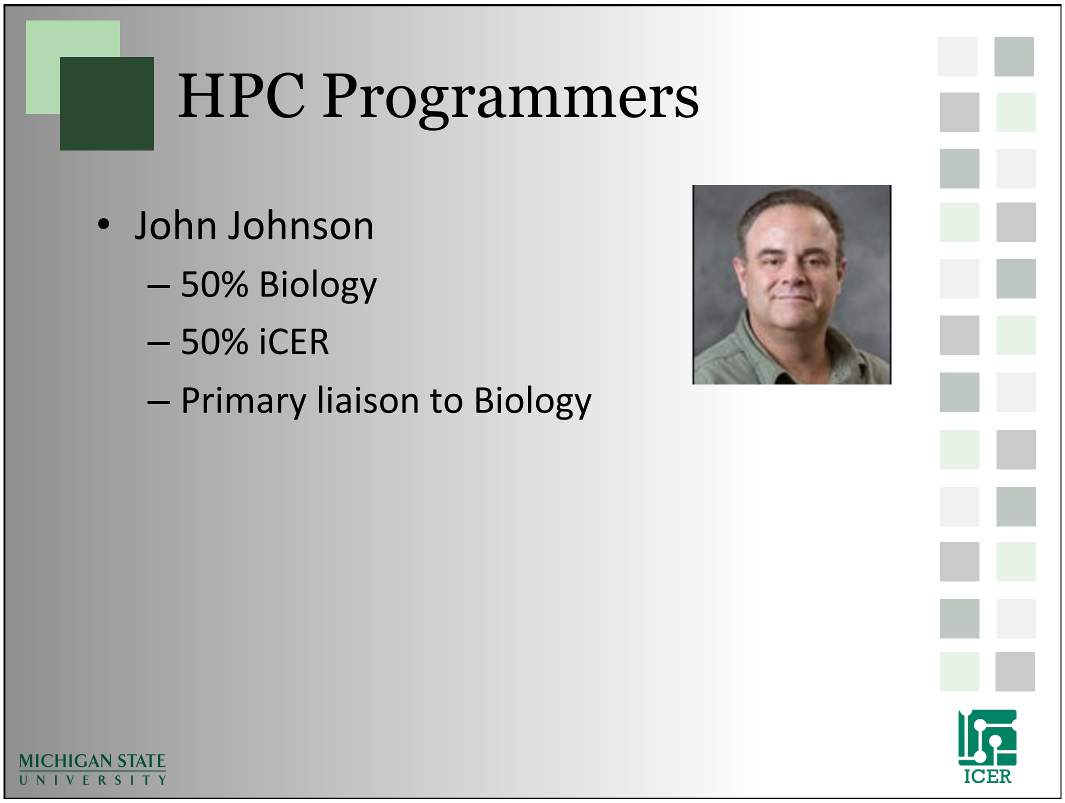


HPC Administrators

- Andrew Keen
- Jim Leikert
- Gregory Mason

MICHIGAN STATE UNIVERSITY

ICER



HPC Programmers

- John Johnson
  - 50% Biology
  - 50% iCER
  - Primary liaison to Biology

MICHIGAN STATE UNIVERSITY

ICER

iCER Research Specialist

Ben Ong and Me

- Research Consulting
- HPCC Programming
- Proposal Writing
- Training and Education
- Outreach

MICHIGAN STATE UNIVERSITY

ICER

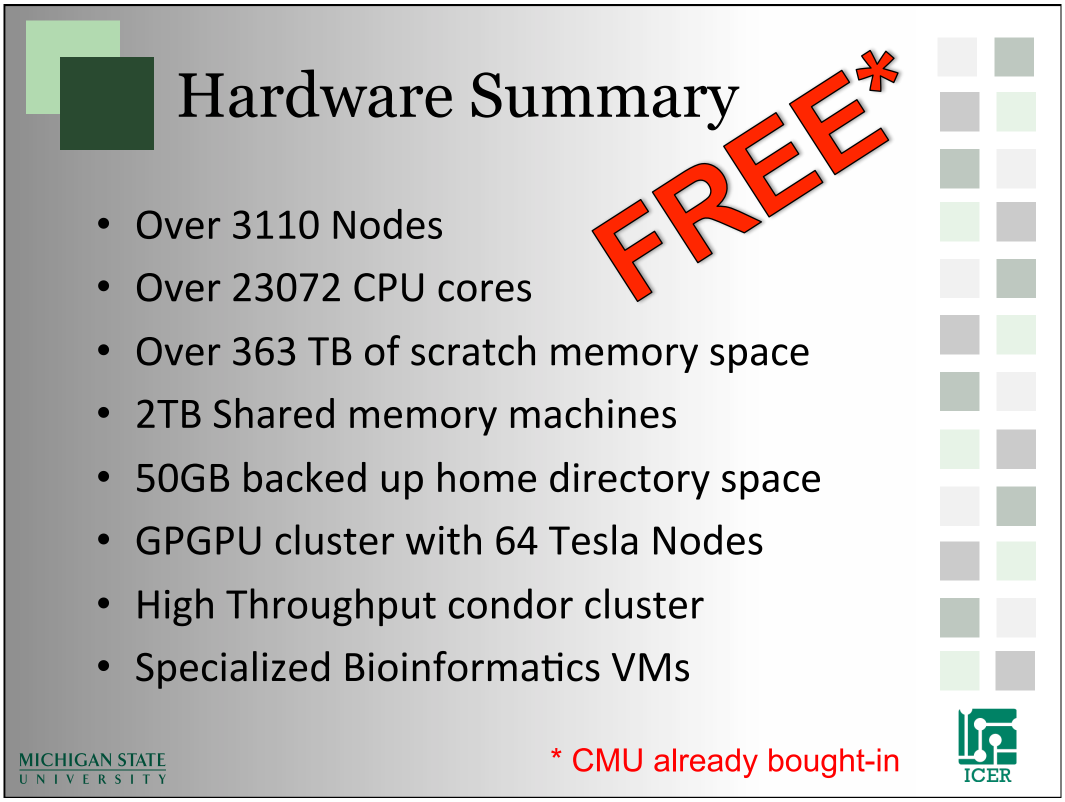
Bigger Science

- The goal of iCER is NOT:
  - Kflops

    (floating point operations per second)
- Instead, the goal of iCER IS:
  - 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

MICHIGAN STATE UNIVERSITY

ICER



Hardware Summary

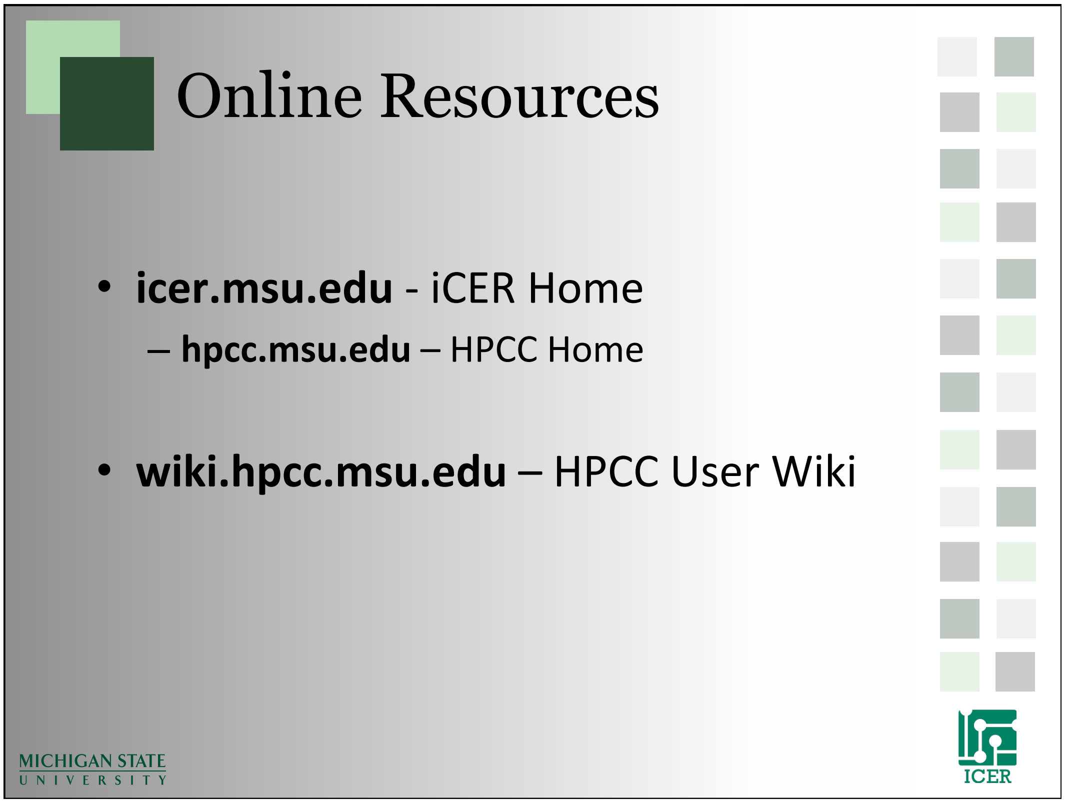
FREE*

- Over 3110 Nodes
- Over 23072 CPU cores
- Over 363 TB of scratch memory space
- 2TB Shared memory machines
- 50GB backed up home directory space
- GPGPU cluster with 64 Tesla Nodes
- High Throughput condor cluster
- Specialized Bioinformatics VMs

* CMU already bought-in

MICHIGAN STATE UNIVERSITY

ICER

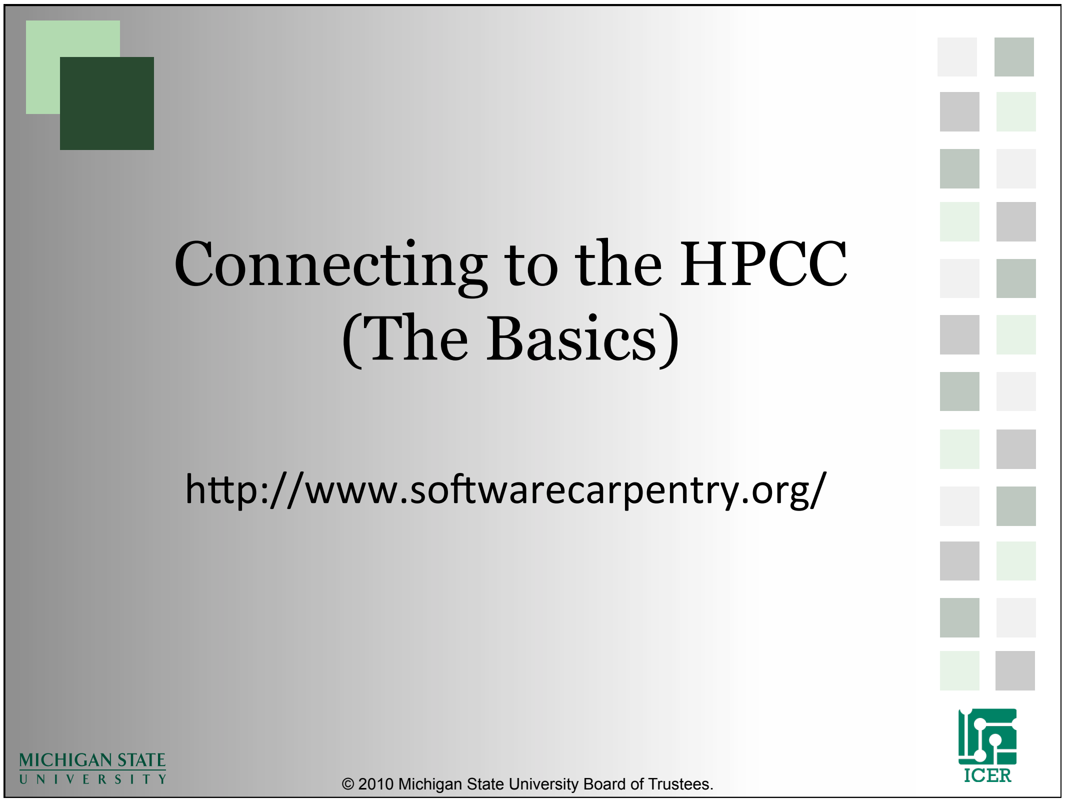


Online Resources

- **icer.msu.edu** - iCER Home
  - **hpcc.msu.edu** – HPCC Home
- **wiki.hpcc.msu.edu** – HPCC User Wiki

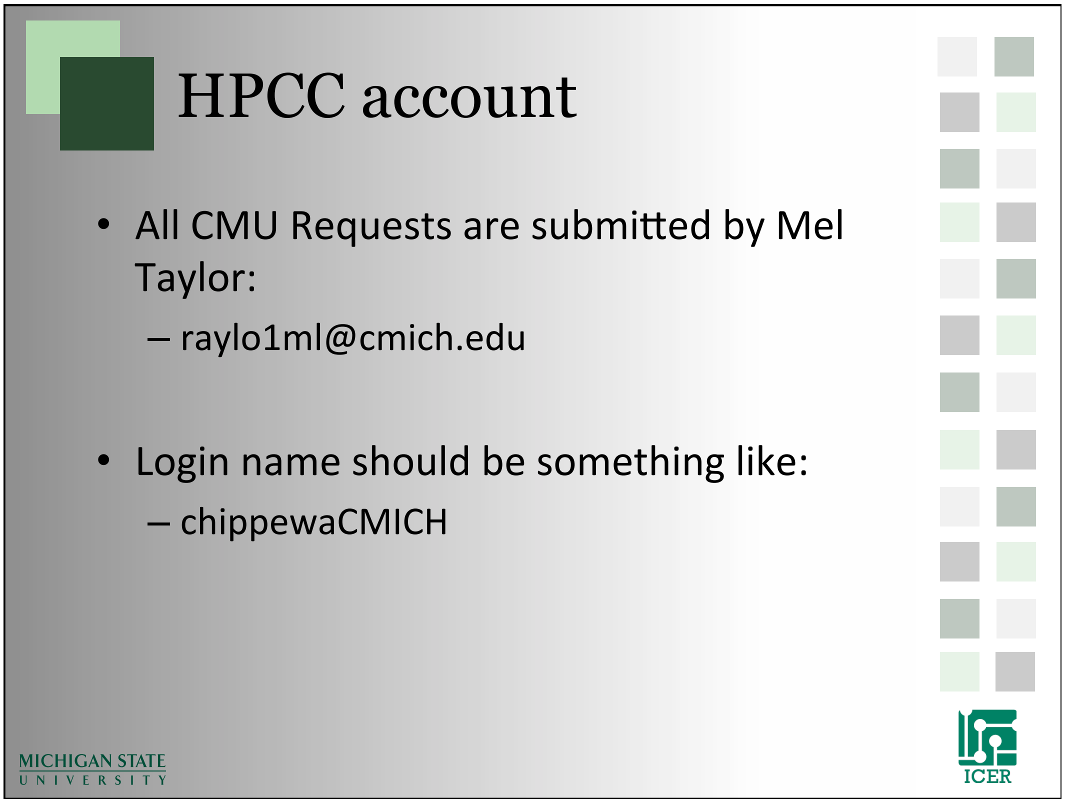
MICHIGAN STATE UNIVERSITY

ICER



Connecting to the HPCC (The Basics)

http://www.softwarecarpentry.org/



HPCC account

- All CMU Requests are submitted by Mel Taylor:
  - raylo1ml@cmich.edu
- Login name should be something like:
  - chippewaCMICH

URLs for Required 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

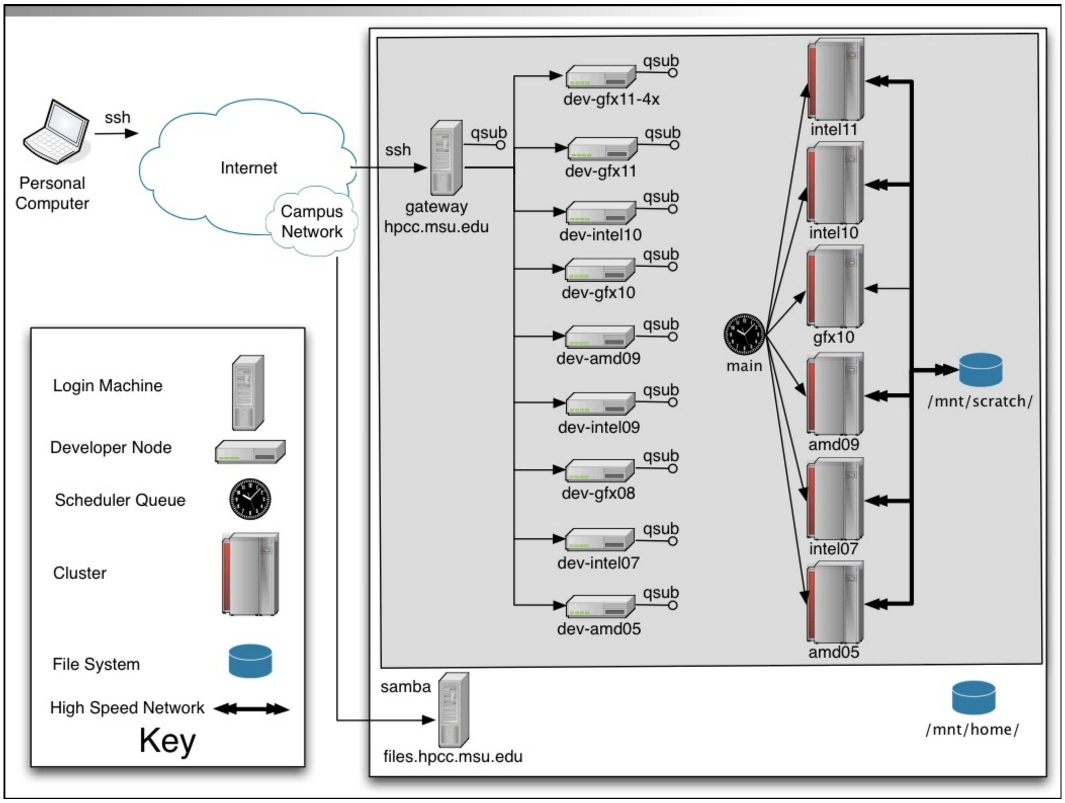
- Plug in your USB thumb drive
- Open the thumb drive folder and select
  - PortableApps
- You should see a new menu in your system tray for navigating
- Start the pchat program

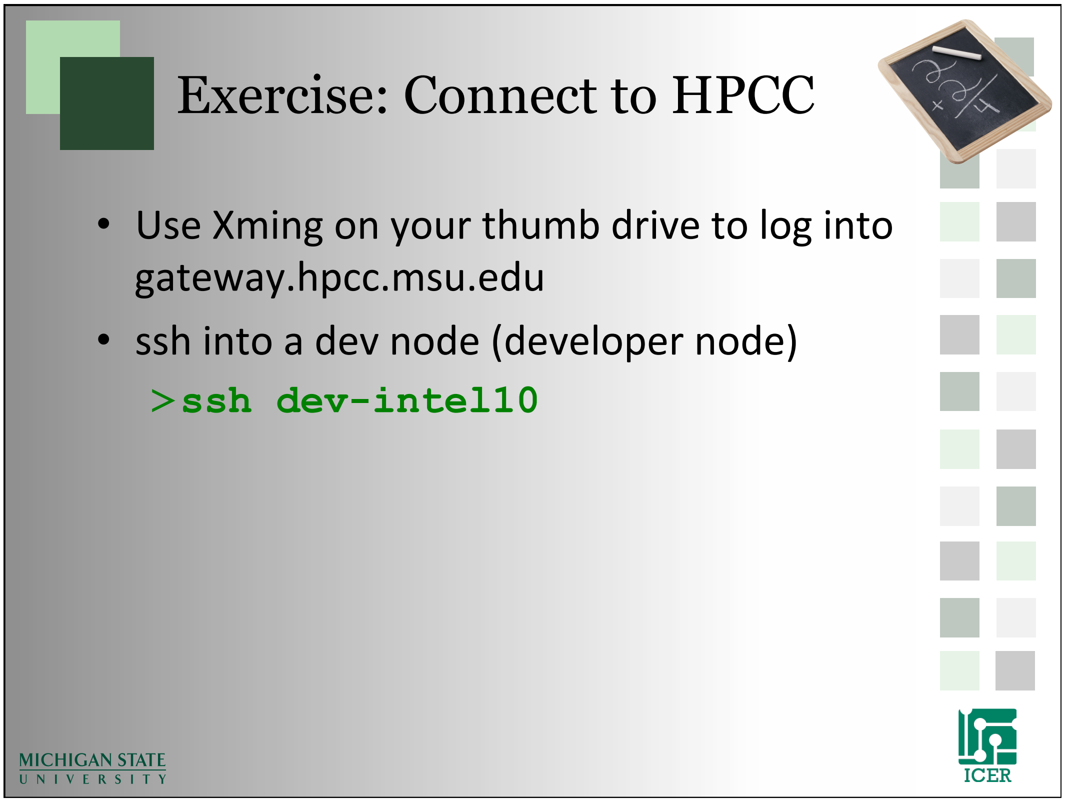
HPCC Cluster Overview

- Linux operating system
- Primary interface is text based though Secure Shell (ssh)
- All machines in the main cluster are binary compatible (compile once run anywhere)
- A scheduler is used to manage jobs running on the cluster
- A submission script is used to tell the scheduler the resources required and how to run a job
- A module system is used to manage the loading and unloading of software configurations

MICHIGAN STATE UNIVERSITY

ICER

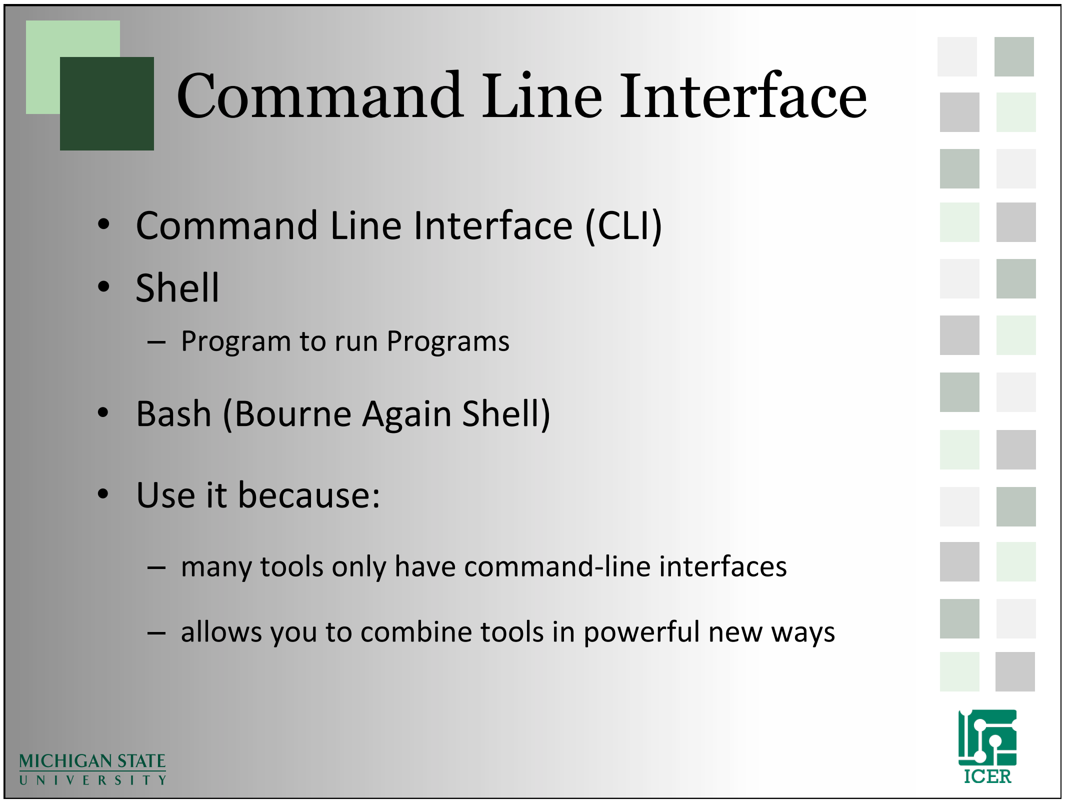




Exercise: Connect to HPCC

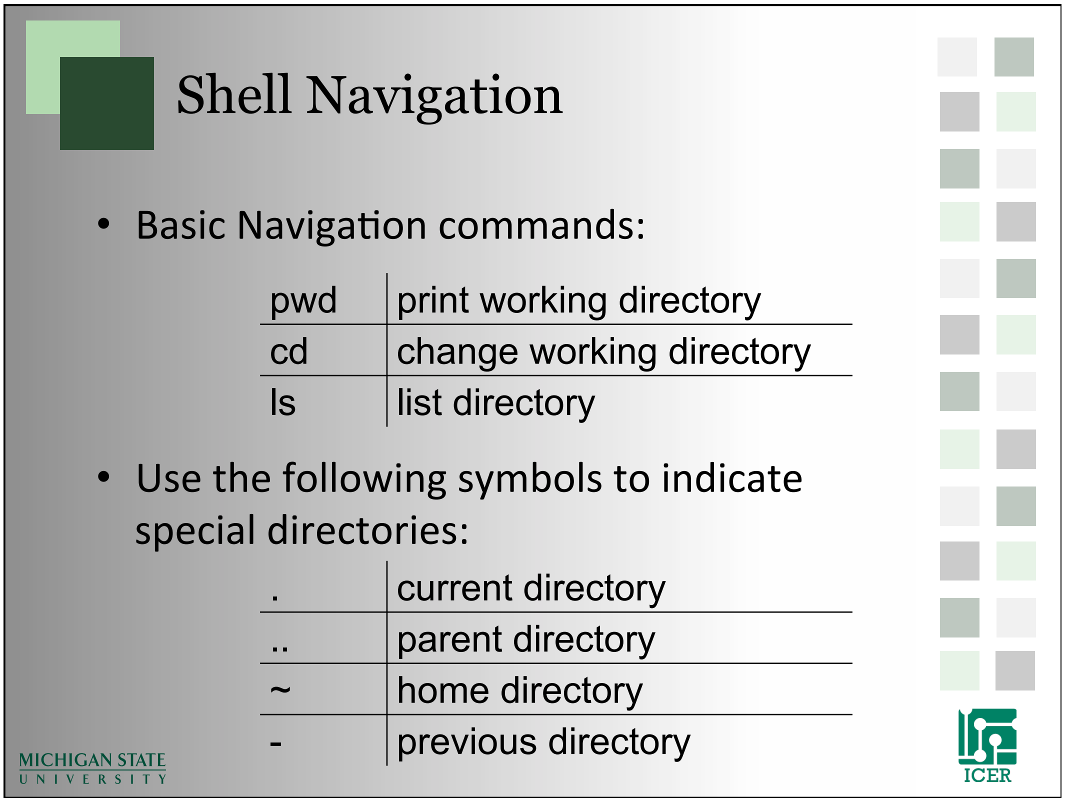
- Use Xming on your thumb drive to log into gateway.hpcc.msu.edu
- ssh into a dev node (developer node)

```
>ssh dev-intel10
```



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



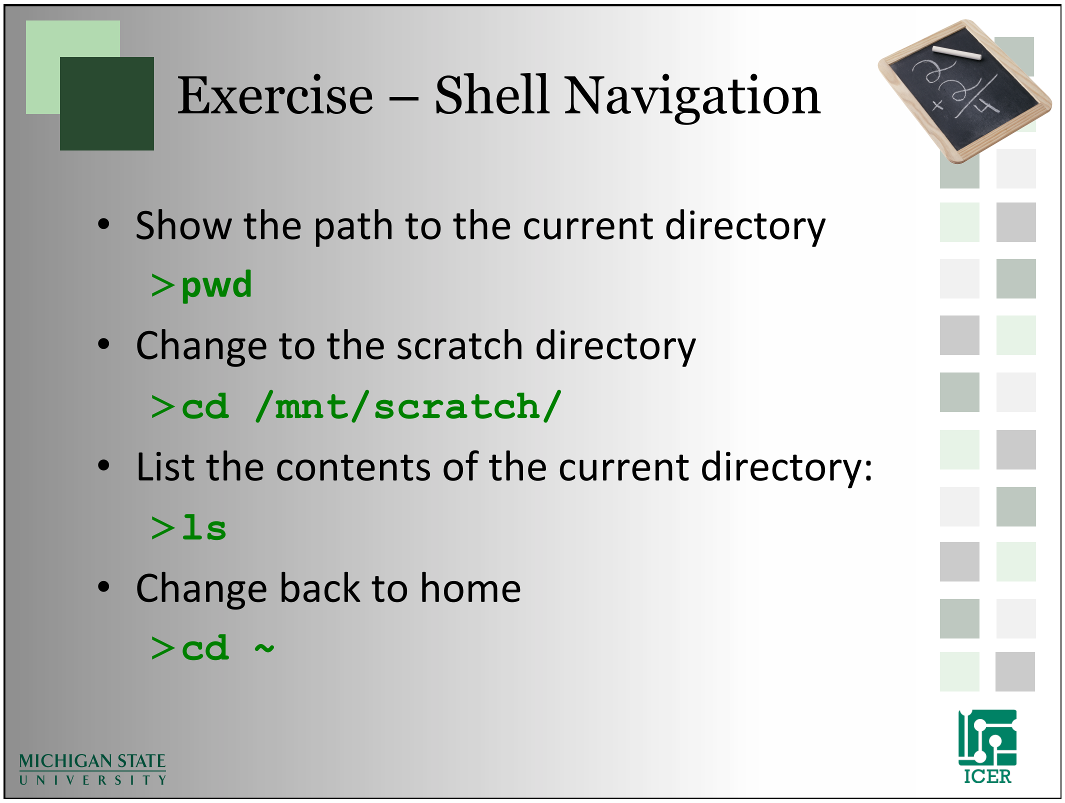
Shell Navigation

- Basic Navigation commands:

pwd	print working directory
cd	change working directory
ls	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

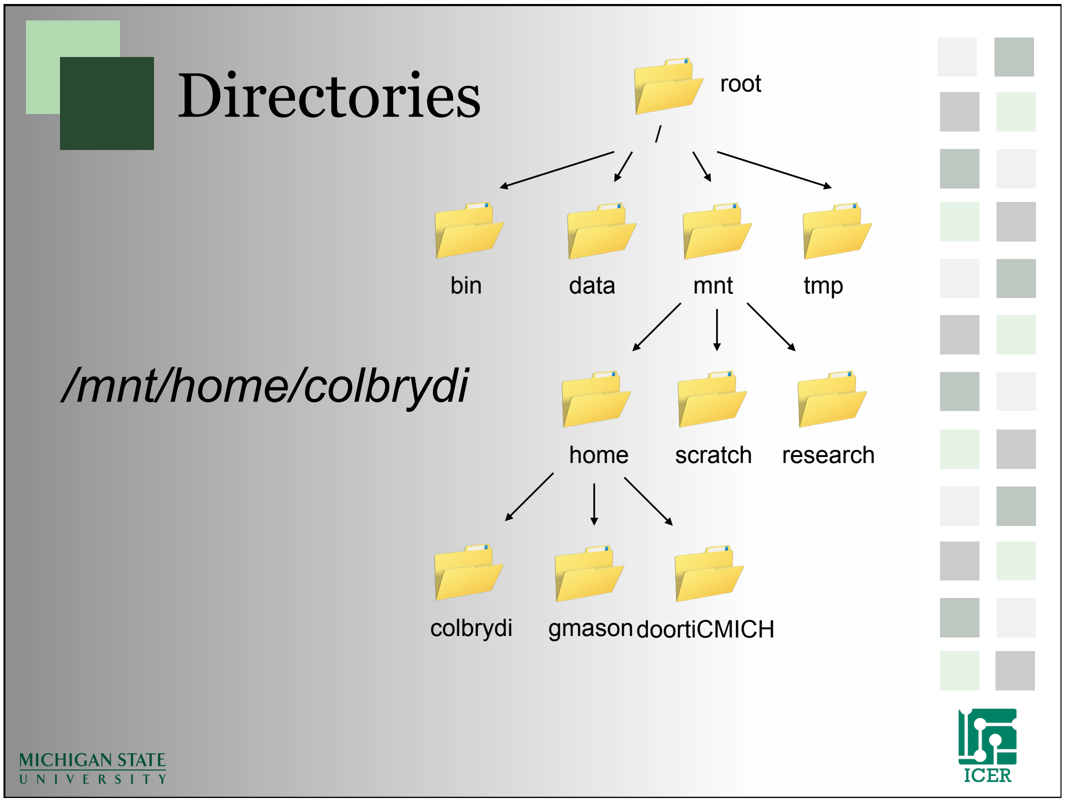
```
>cd /mnt/scratch/
```

- List the contents of the current directory:

```
>ls
```

- Change back to home

```
>cd ~
```



Directories

/mnt/home/colbrydi

root

/

bin

data

mnt

tmp

home

scratch

research

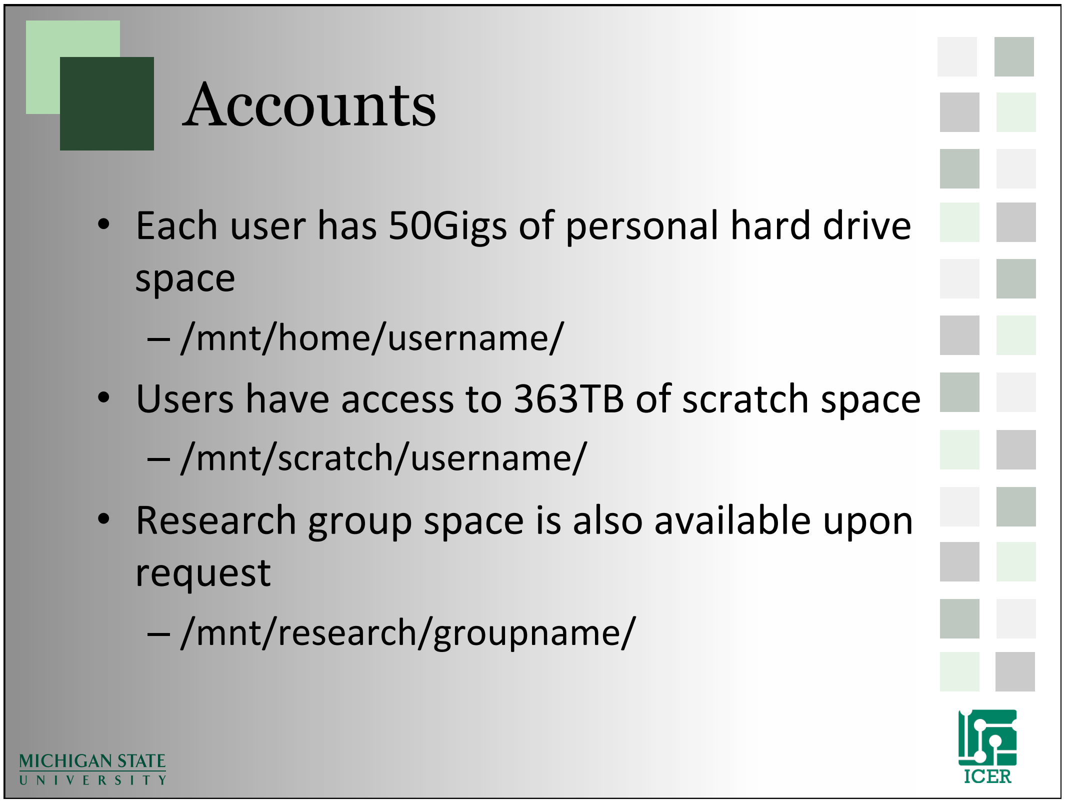
colbrydi

gmason

doortiCMICH

MICHIGAN STATE UNIVERSITY

ICER

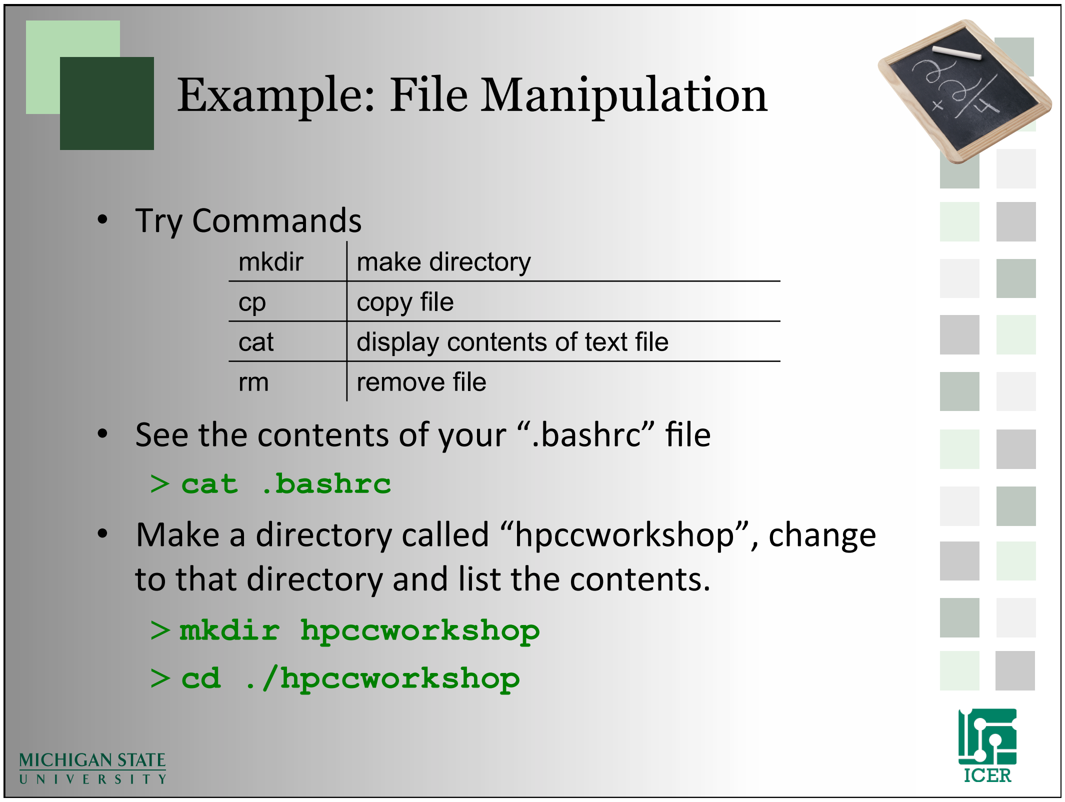


Accounts

- Each user has 50Gigs of personal hard drive space
  - /mnt/home/username/
- Users have access to 363TB of scratch space
  - /mnt/scratch/username/
- Research group space is also available upon request
  - /mnt/research/groupname/

MICHIGAN STATE UNIVERSITY

ICER



Example: File Manipulation

- Try Commands

mkdir	make directory
cp	copy file
cat	display contents of text file
rm	remove file

- See the contents of your ".bashrc" file

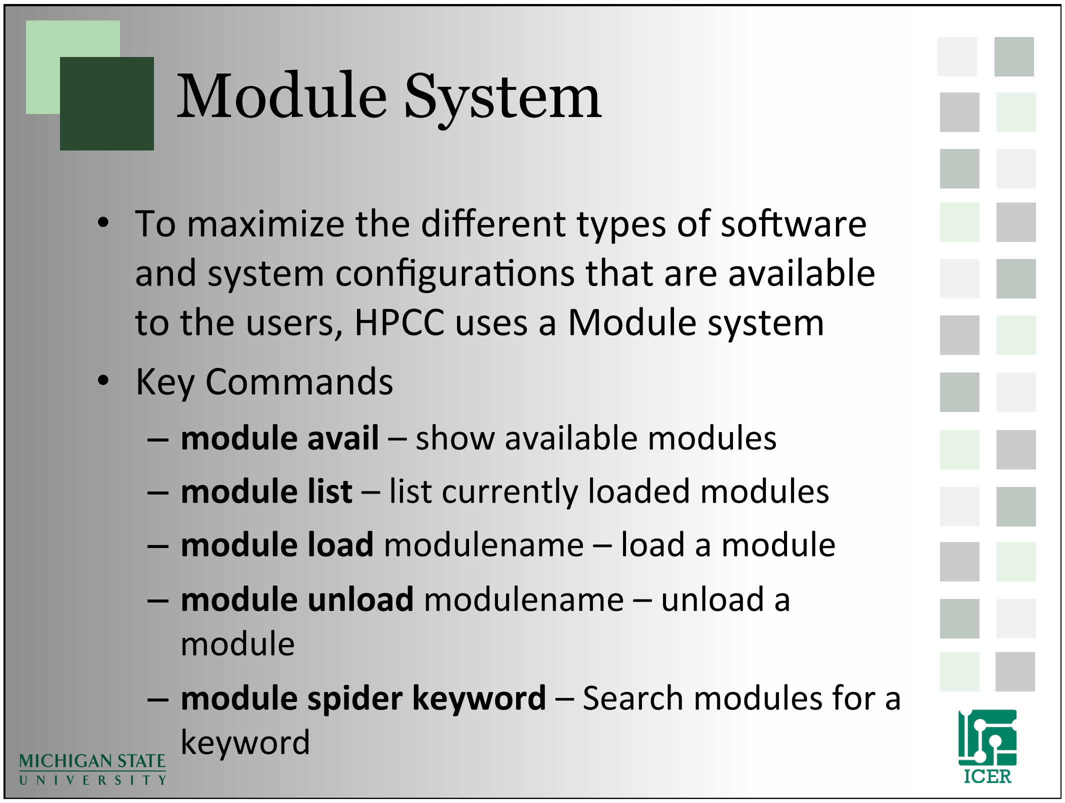
```
> cat .bashrc
```

- Make a directory called "hpccworkshop", change to that directory and list the contents.

```
> mkdir hpccworkshop
> cd ./hpccworkshop
```

MICHIGAN STATE UNIVERSITY

ICER

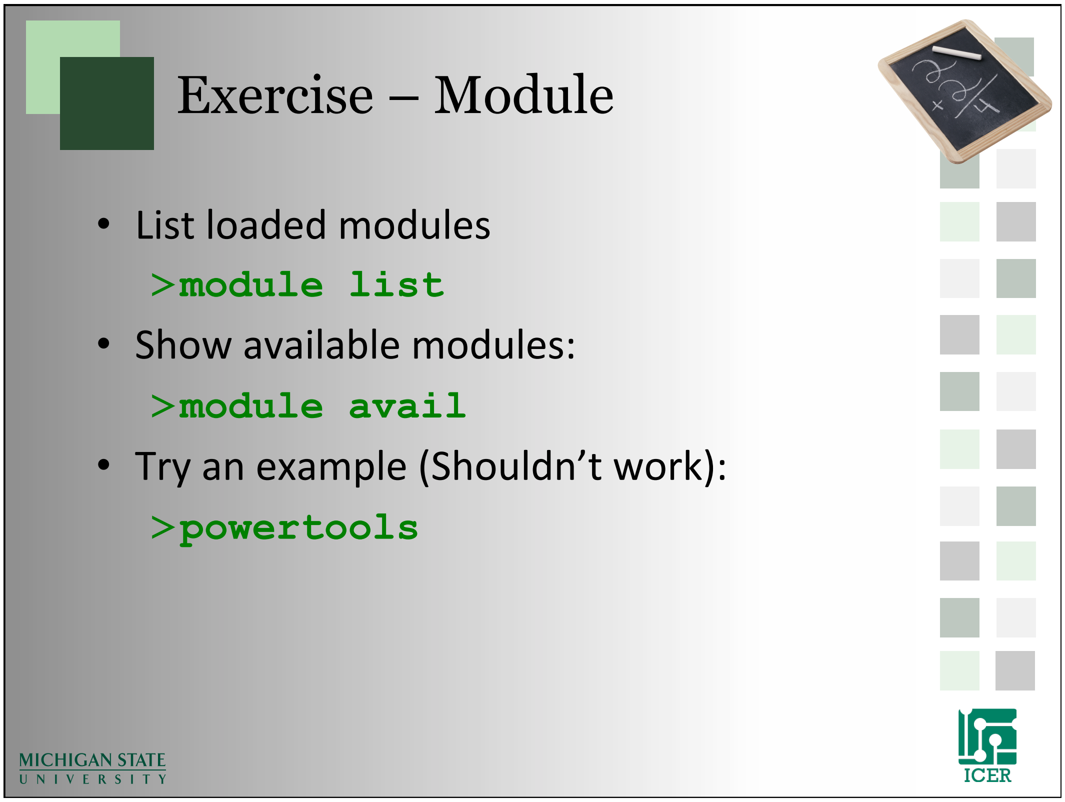


Module System

- 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

MICHIGAN STATE UNIVERSITY

ICER



Exercise – Module

- List loaded modules

```
>module list
```

- Show available modules:

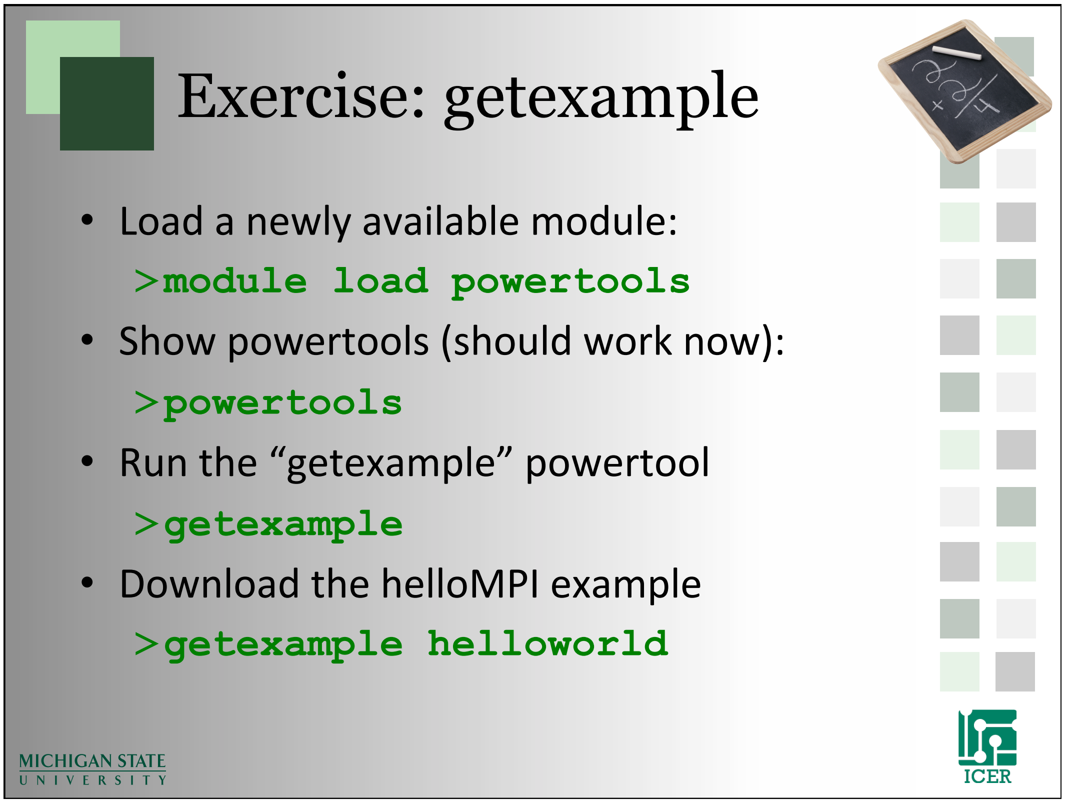
```
>module avail
```

- Try an example (Shouldn’t work):

```
>powertools
```

MICHIGAN STATE
UNIVERSITY

ICER



Exercise: getexample

- Load a newly available module:

```
>module load powertools
```

- Show powertools (should work now):

```
>powertools
```

- Run the “getexample” powertool

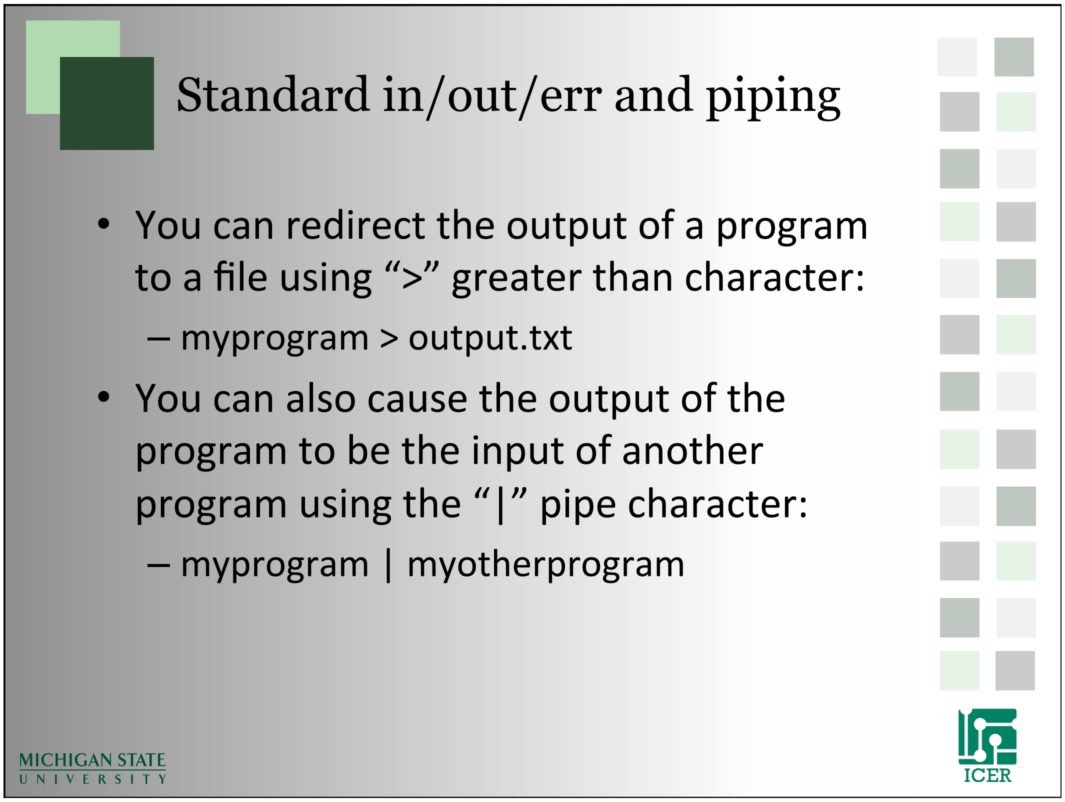
```
>getexample
```

- Download the helloMPI example

```
>getexample helloworld
```

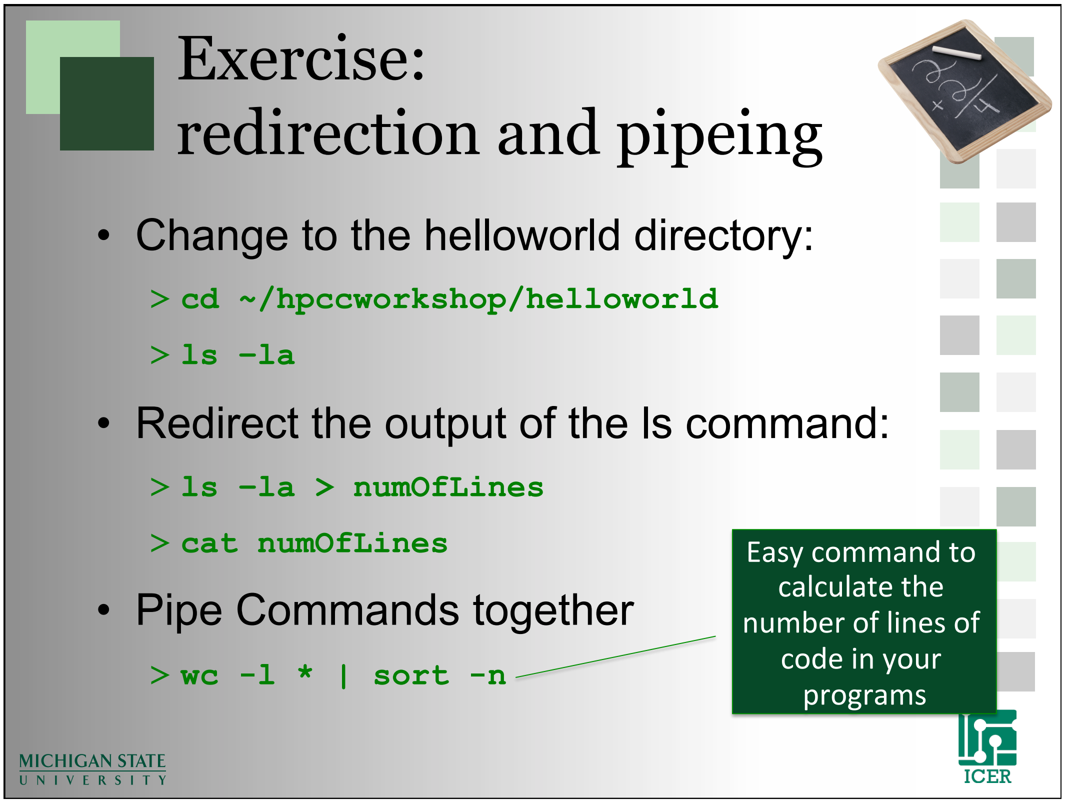
MICHIGAN STATE
UNIVERSITY

ICER



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 pipeing

- Change to the helloworld directory:

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

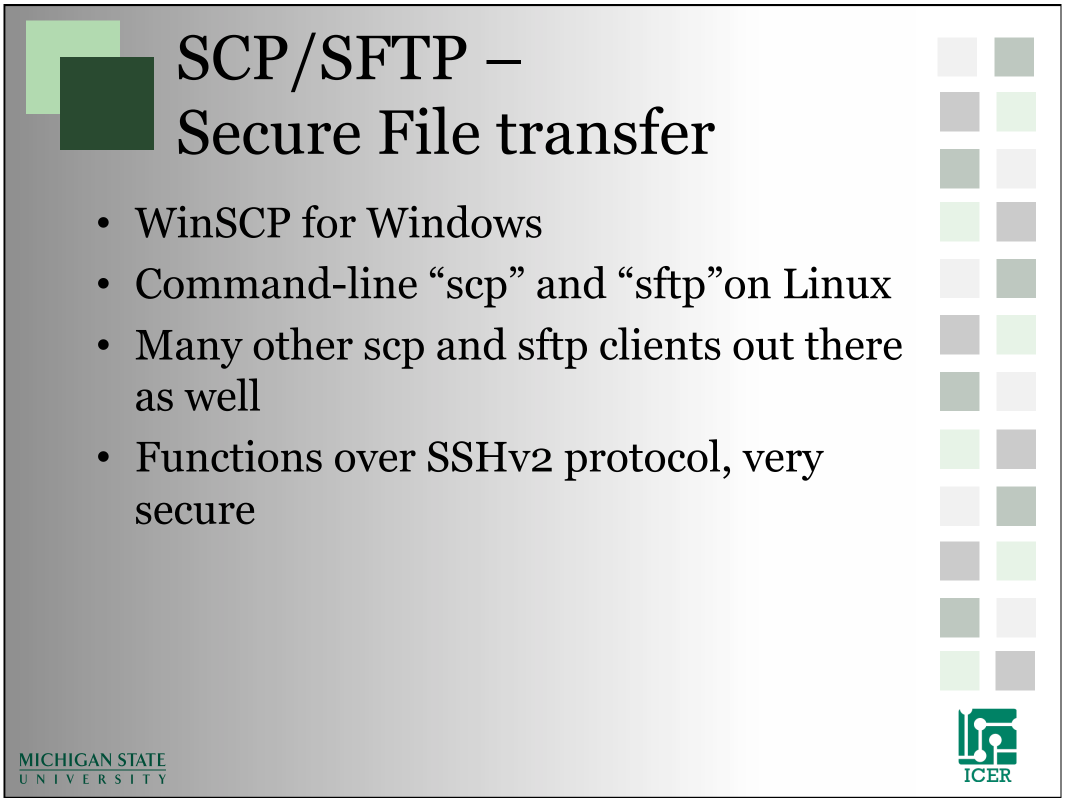
- Redirect the output of the ls command:

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

- Pipe Commands together

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

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

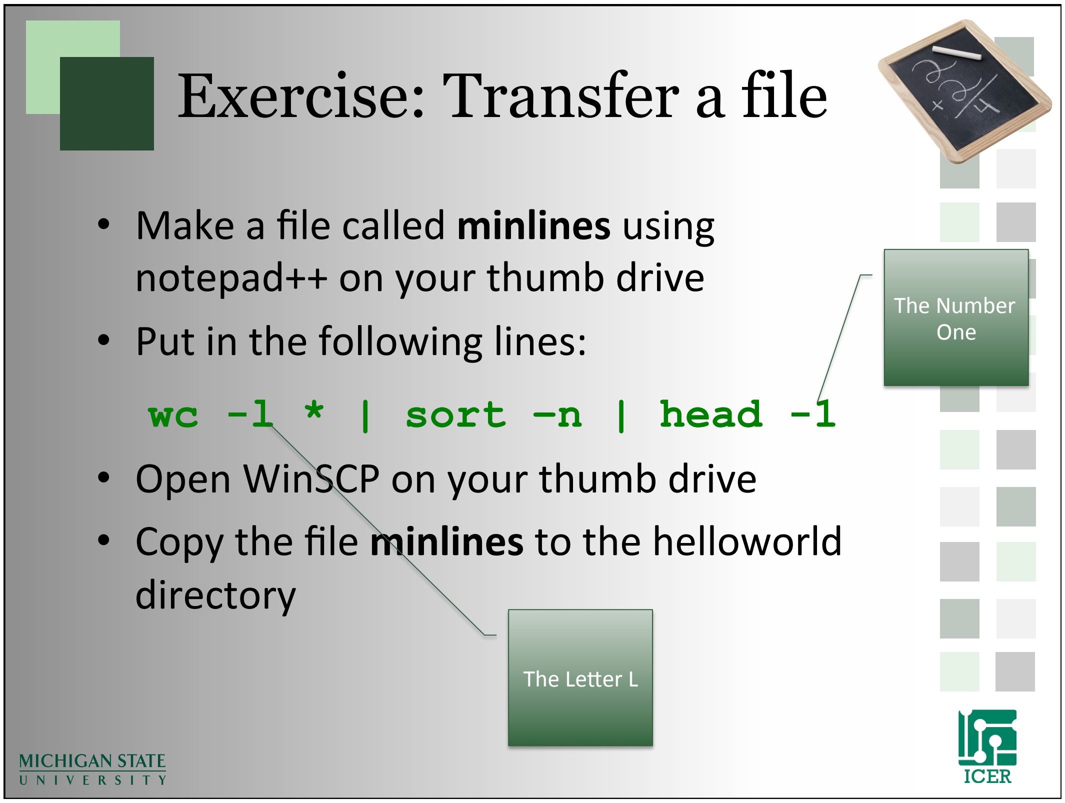


SCP/SFTP – Secure File transfer

- WinSCP for Windows
- Command-line “scp” and “sftp”on Linux
- Many other scp and sftp clients out there as well
- Functions over SSHv2 protocol, very secure

MICHIGAN STATE UNIVERSITY

ICER



Exercise: Transfer a file

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

```
wc -l * | sort -n | head -1
```

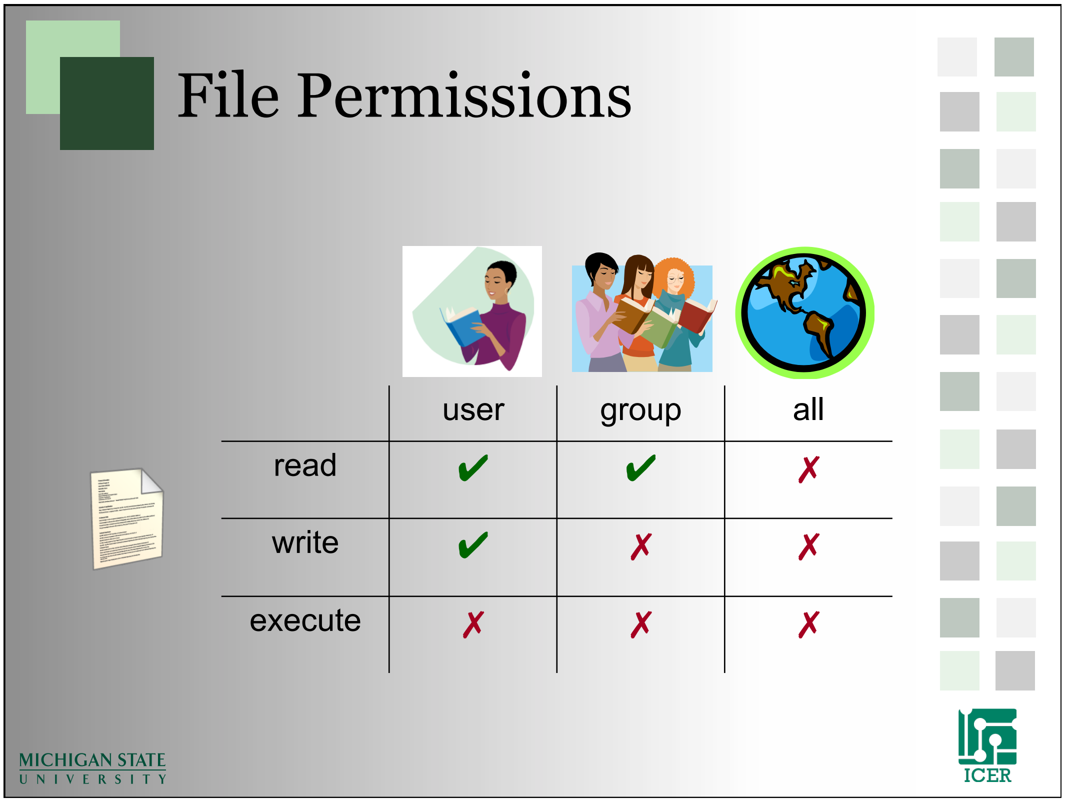
- Open WinSCP on your thumb drive
- Copy the file **minlines** to the helloworld directory

The Number One

The Letter L

MICHIGAN STATE UNIVERSITY

ICER

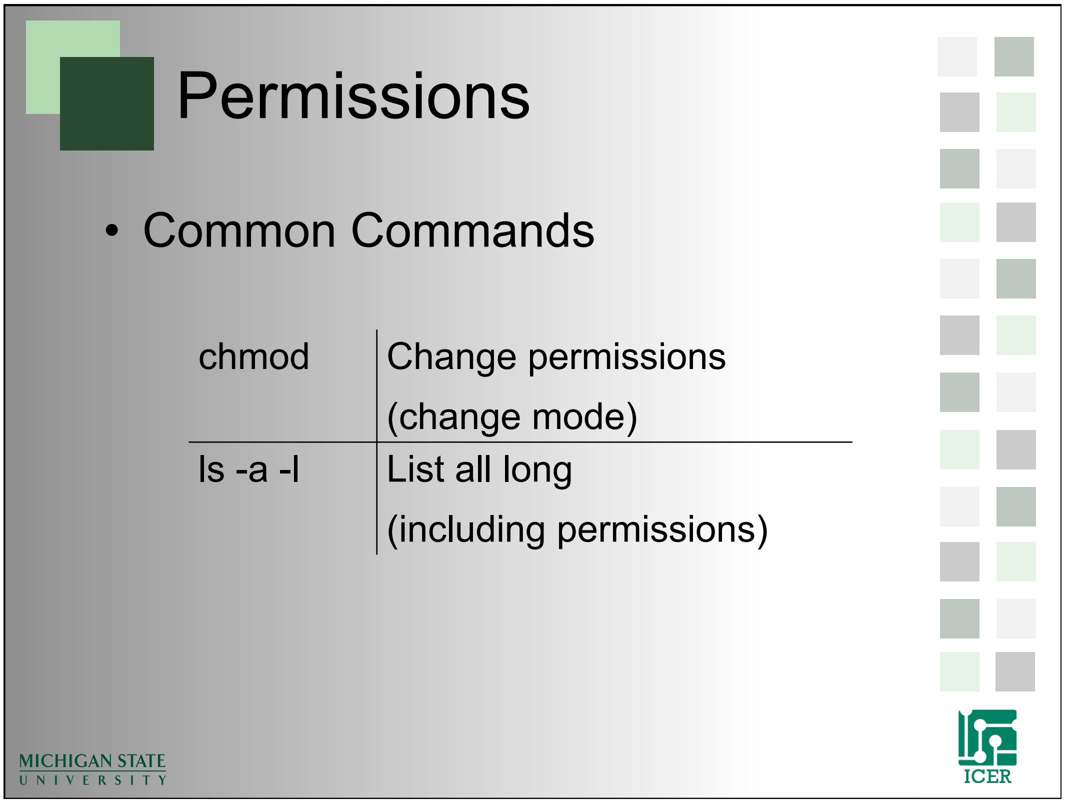


File Permissions

	user	group	all
read	✓	✓	✗
write	✓	✗	✗
execute	✗	✗	✗

MICHIGAN STATE UNIVERSITY

ICER



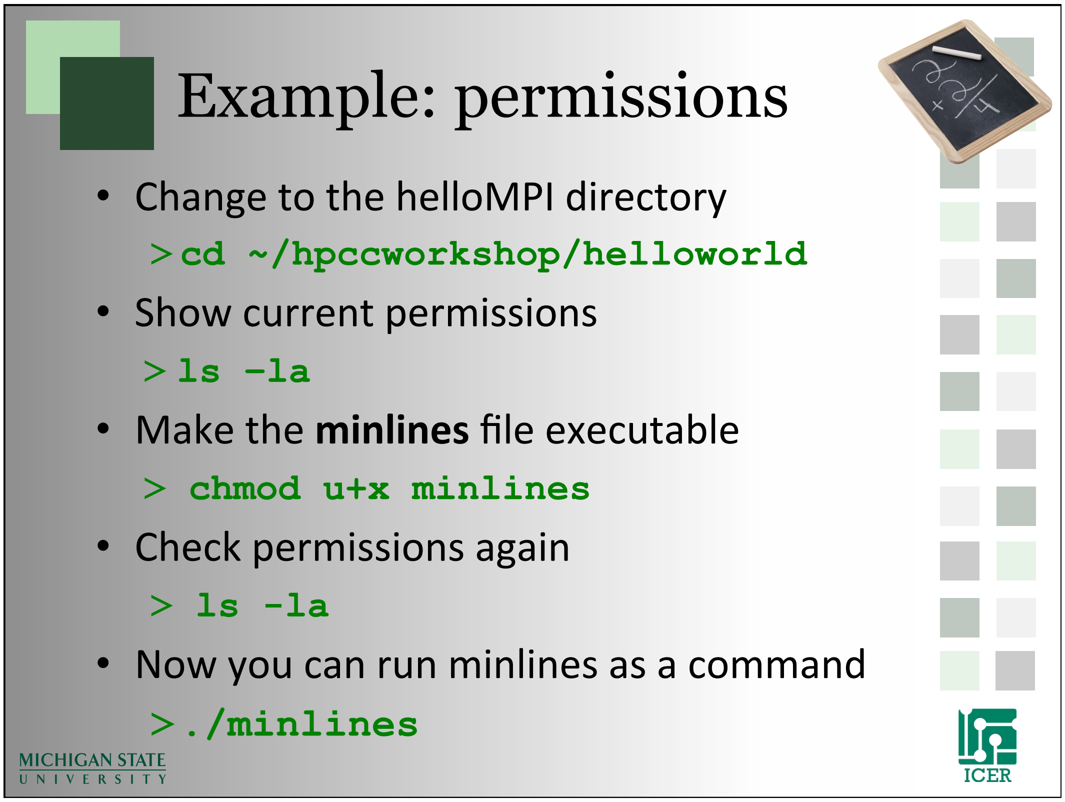
Permissions

- Common Commands

chmod	Change permissions (change mode)
ls -a -l	List all long (including permissions)

MICHIGAN STATE UNIVERSITY

ICER



Example: permissions

- Change to the helloMPI directory

```
>cd ~/hpccworkshop/helloworld
```

- Show current permissions

```
>ls -la
```

- Make the **minlines** file executable

```
> chmod u+x minlines
```

- Check permissions again

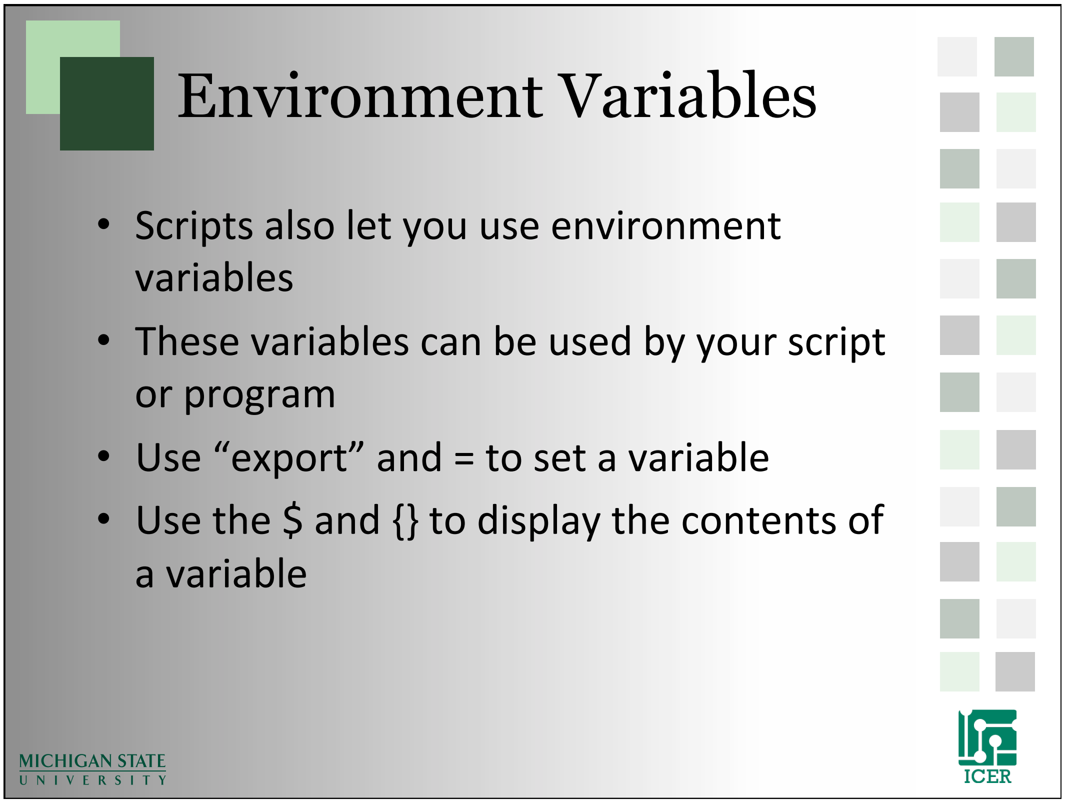
```
> ls -la
```

- Now you can run minlines as a command

```
>./minlines
```

MICHIGAN STATE UNIVERSITY

ICER

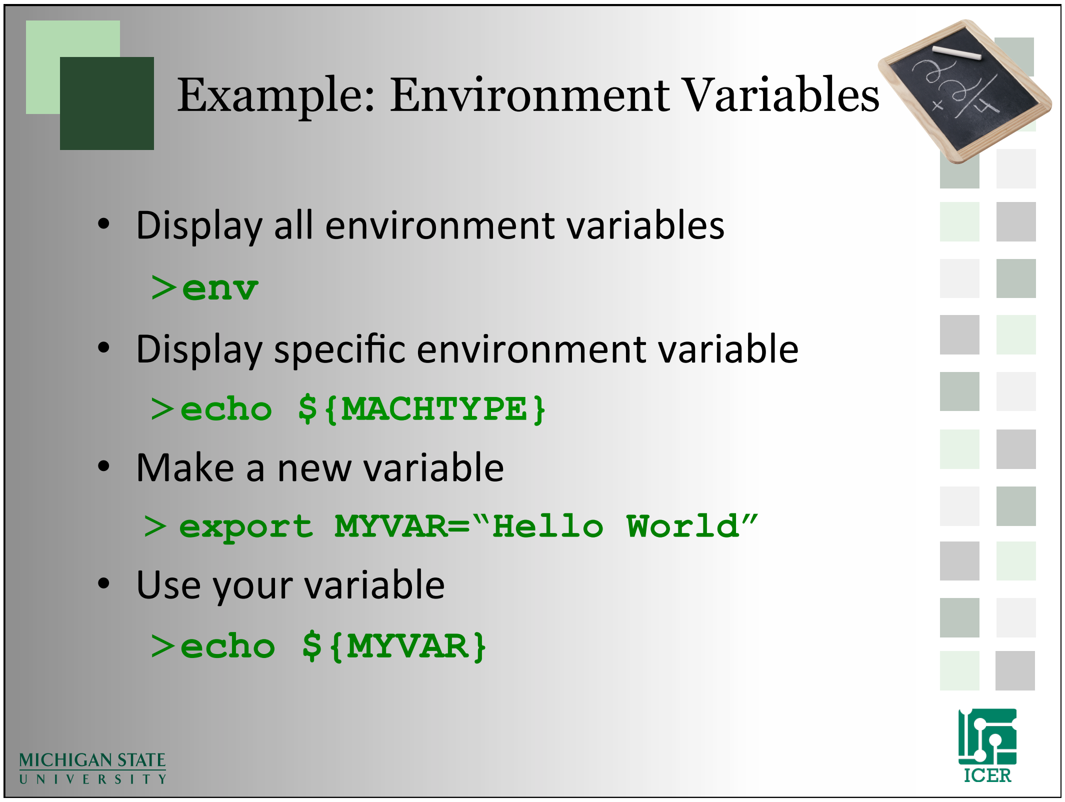


Environment Variables

- 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

MICHIGAN STATE UNIVERSITY

ICER



Example: Environment Variables

- Display all environment variables

```
>env
```

- Display specific environment variable

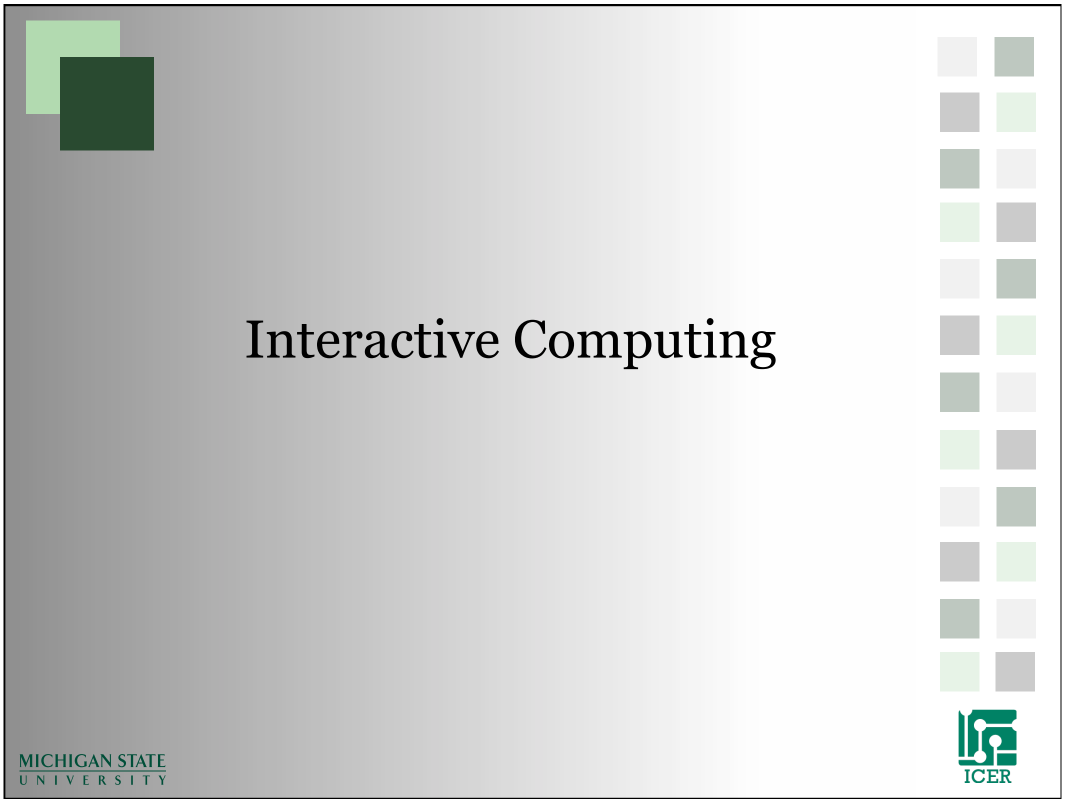
```
>echo ${MACHTYPE}
```

- Make a new variable

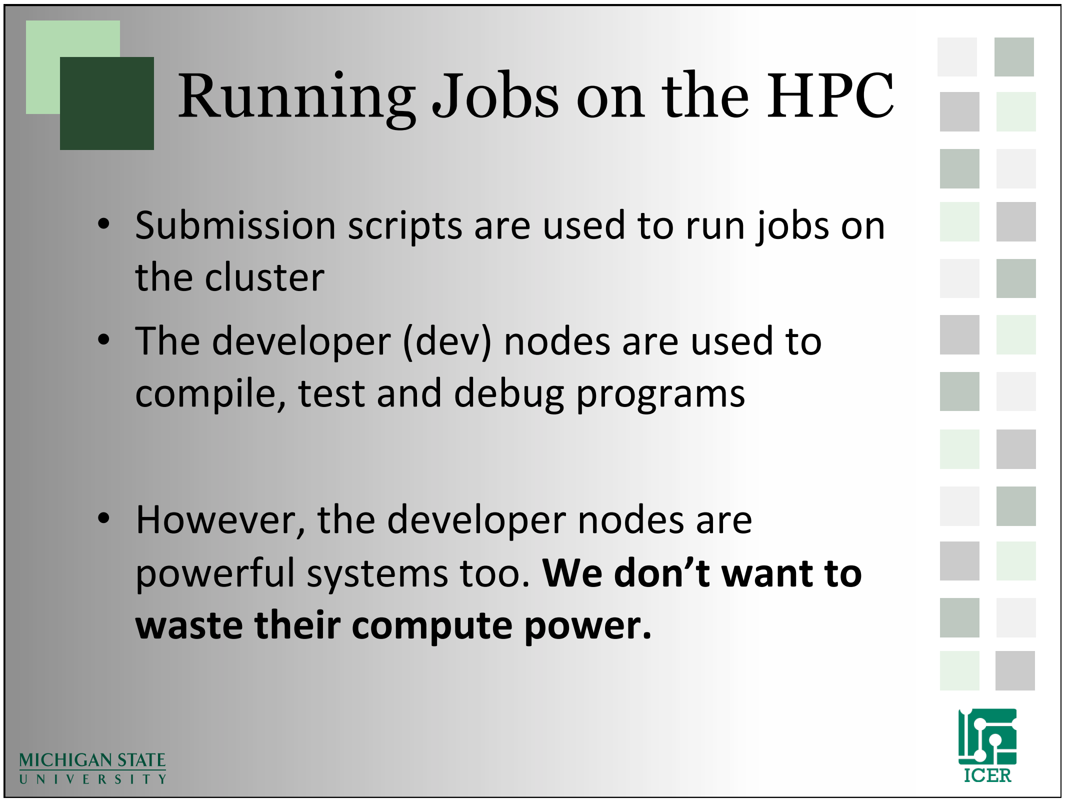
```
> export MYVAR=“Hello World”
```

- Use your variable

```
>echo ${MYVAR}
```



Interactive Computing



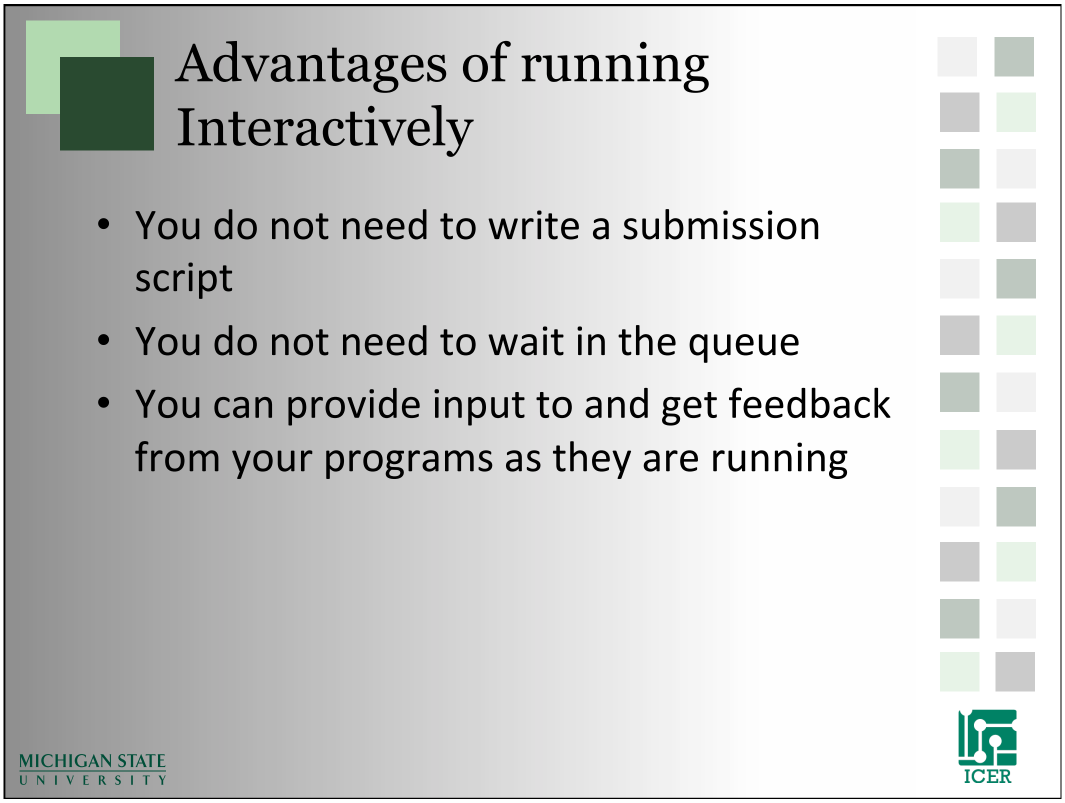
Running Jobs on the HPC

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

- However, the developer nodes are powerful systems too. **We don't want to waste their compute power.**

MICHIGAN STATE UNIVERSITY

ICER



Advantages of running Interactively

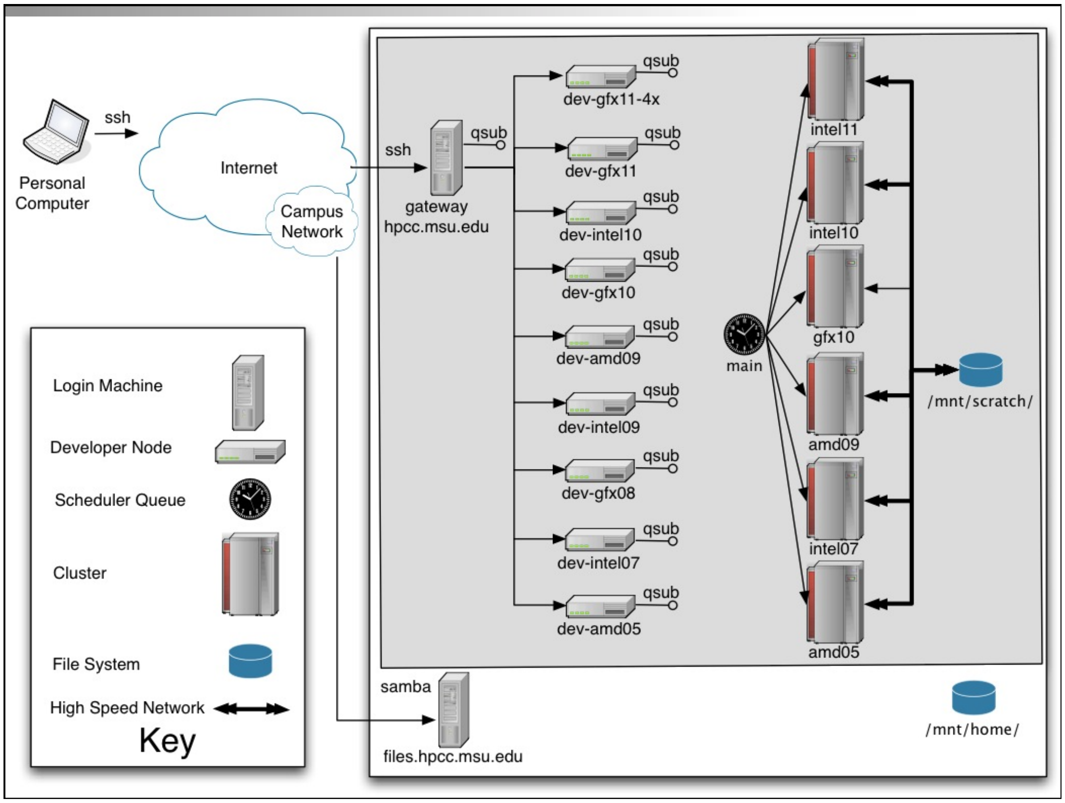
- You do not need to write a submission script
- You do not need to wait in the queue
- You can provide input to and get feedback from your programs as they are running

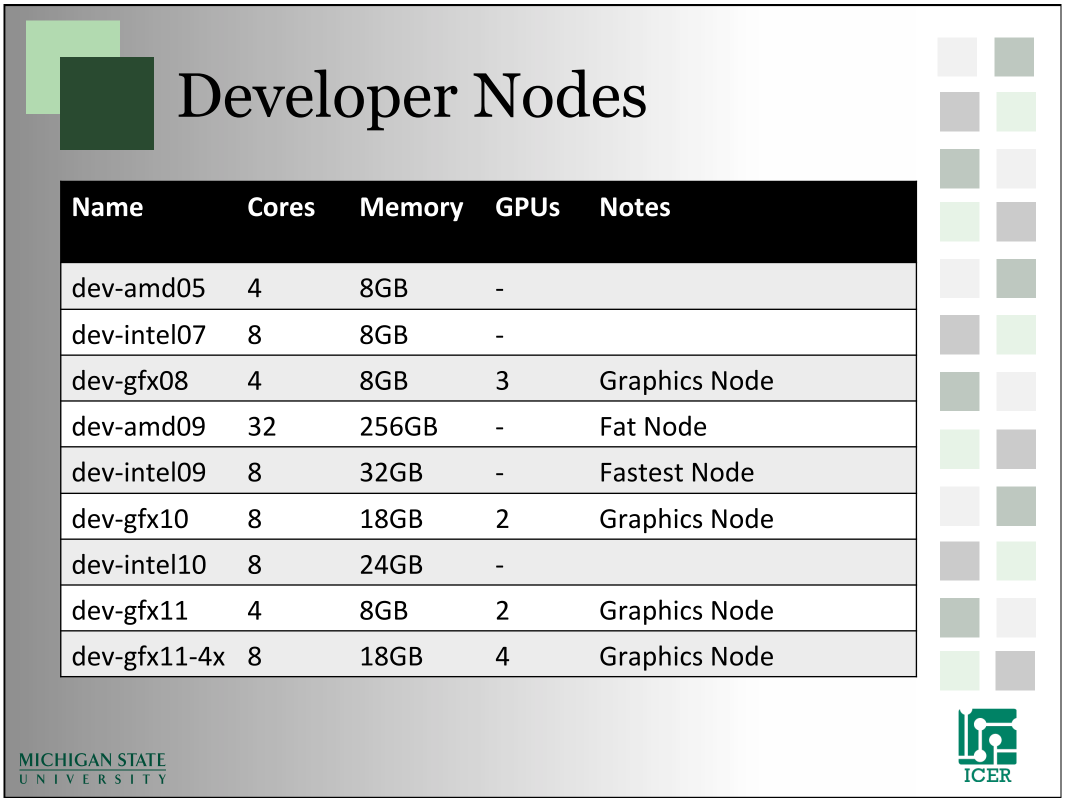
MICHIGAN STATE UNIVERSITY

ICER

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.



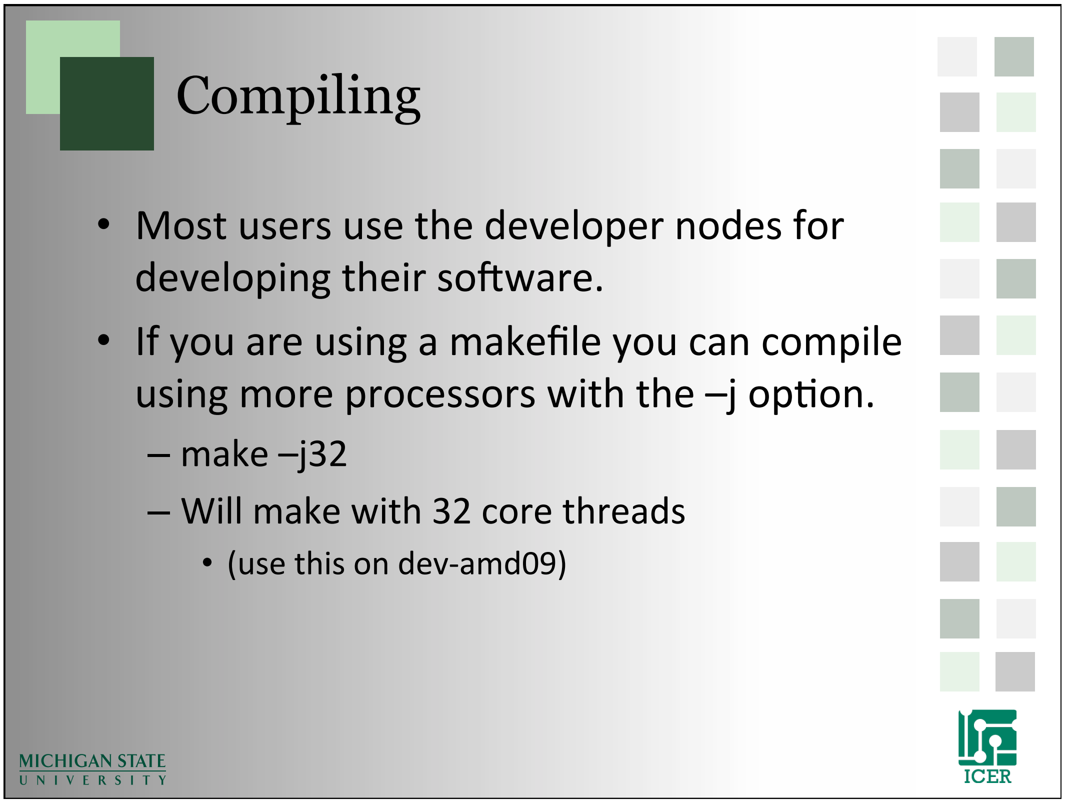


Developer Nodes

Name	Cores	Memory	GPUs	Notes
dev-amd05	4	8GB	-	
dev-intel07	8	8GB	-	
dev-gfx08	4	8GB	3	Graphics Node
dev-amd09	32	256GB	-	Fat Node
dev-intel09	8	32GB	-	Fastest Node
dev-gfx10	8	18GB	2	Graphics Node
dev-intel10	8	24GB	-	
dev-gfx11	4	8GB	2	Graphics Node
dev-gfx11-4x	8	18GB	4	Graphics Node

MICHIGAN STATE UNIVERSITY

ICER

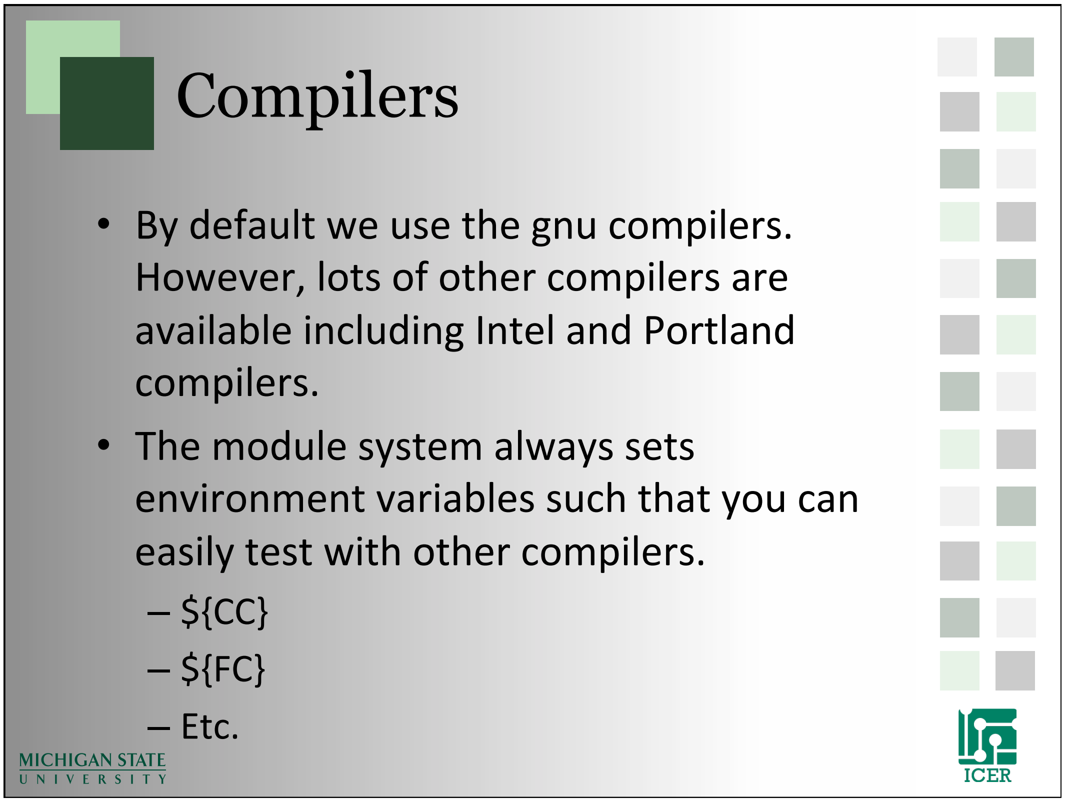


Compiling

- Most users use the developer nodes for developing their software.
- If you are using a makefile you can compile using more processors with the –j option.
  - make –j32
  - Will make with 32 core threads
    - (use this on dev-amd09)

MICHIGAN STATE UNIVERSITY

ICER

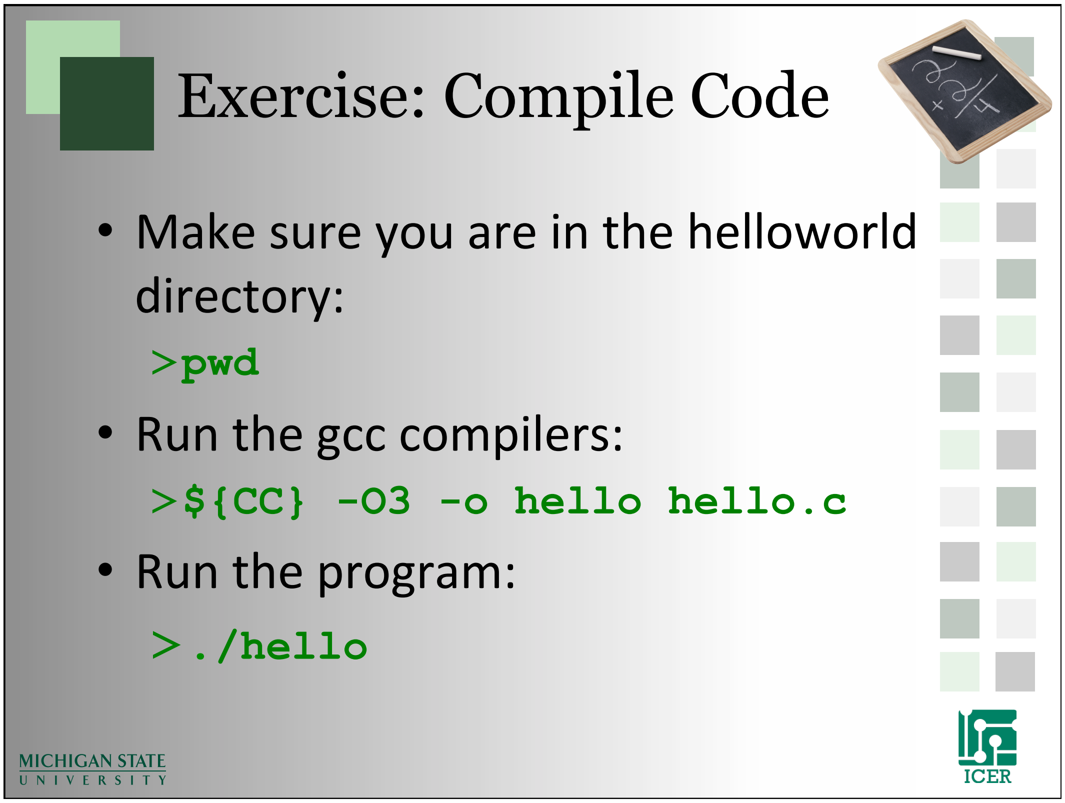


Compilers

- 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.

MICHIGAN STATE
UNIVERSITY

ICER



Exercise: Compile Code

- Make sure you are in the helloworld directory:

```
>pwd
```

- Run the gcc compilers:

```
>${CC} -O3 -o hello hello.c
```

- Run the program:

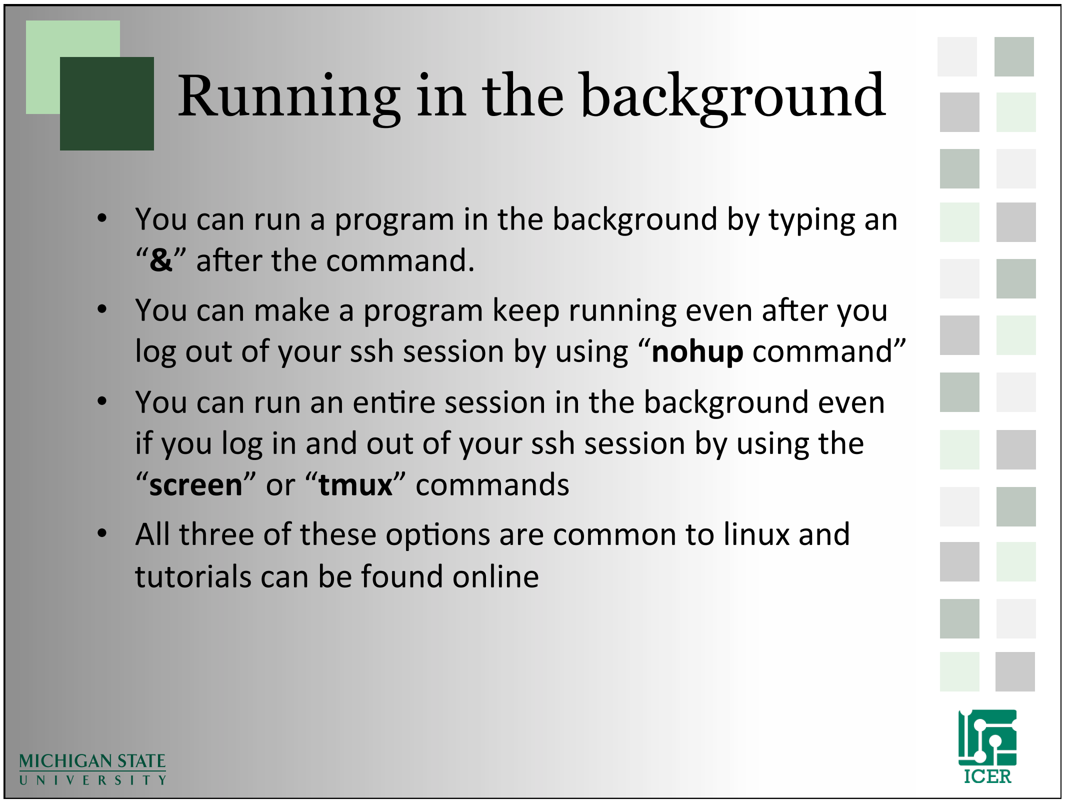
```
>./hello
```

MICHIGAN STATE
UNIVERSITY

ICER

Available Software

- Center Supported Development Software
  - Intel compilers, openmp, openmpi, mvapich, totalview, mkl, pathscale, gnu...
- Center Supported Research Software
  - R, HEEDS, amber, blast ...
- 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/



Running in the background

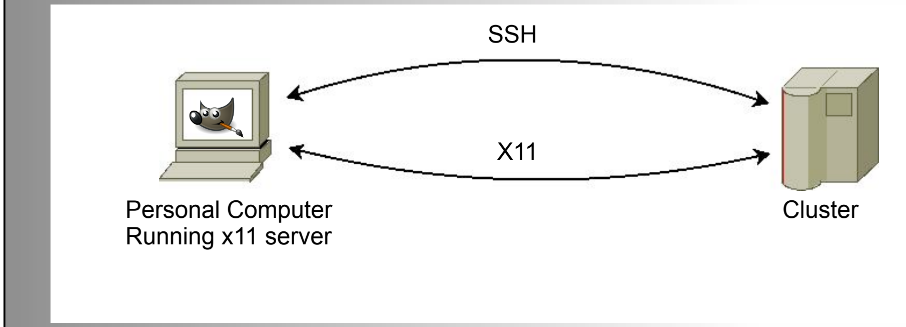
- You can run a program in the background by typing an “**&**” after the command.
- You can make a program keep running even after you log out of your ssh session by using “**nohup** command”
- You can run an entire session in the background even if you log in and out of your ssh session by using the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

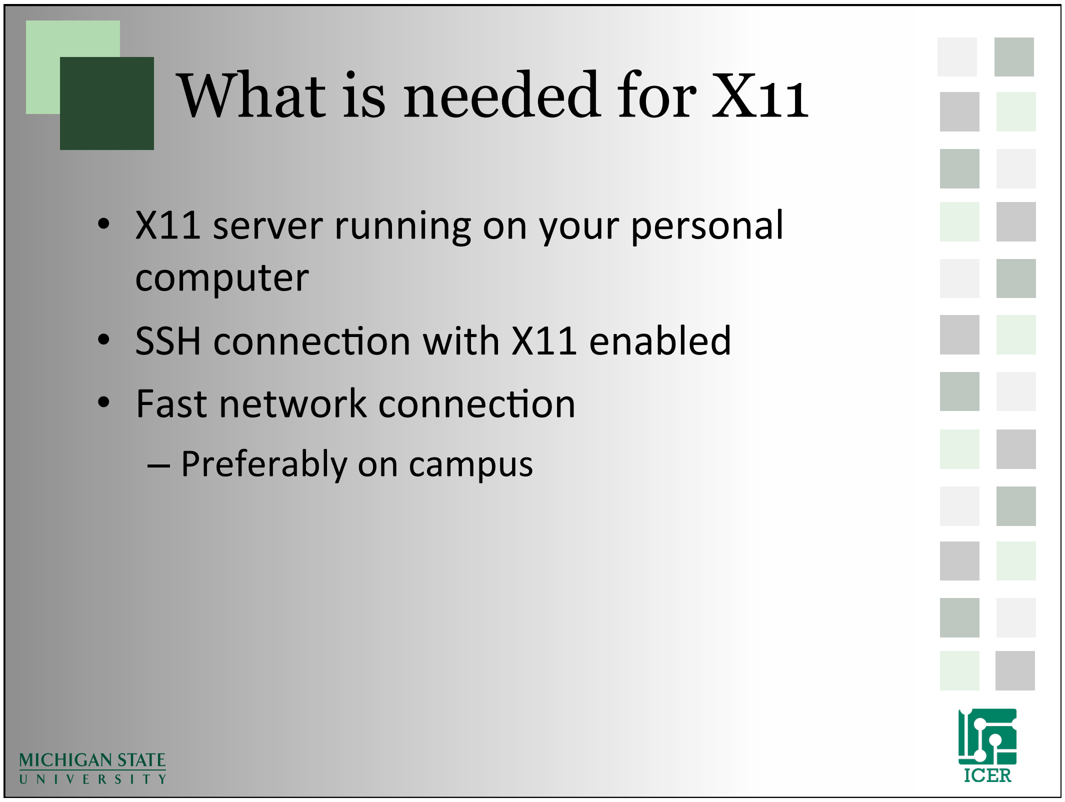
CLI vs GUI

- CLI – Command Line Interface
- GUI – Graphical User Interface

What is X11?

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



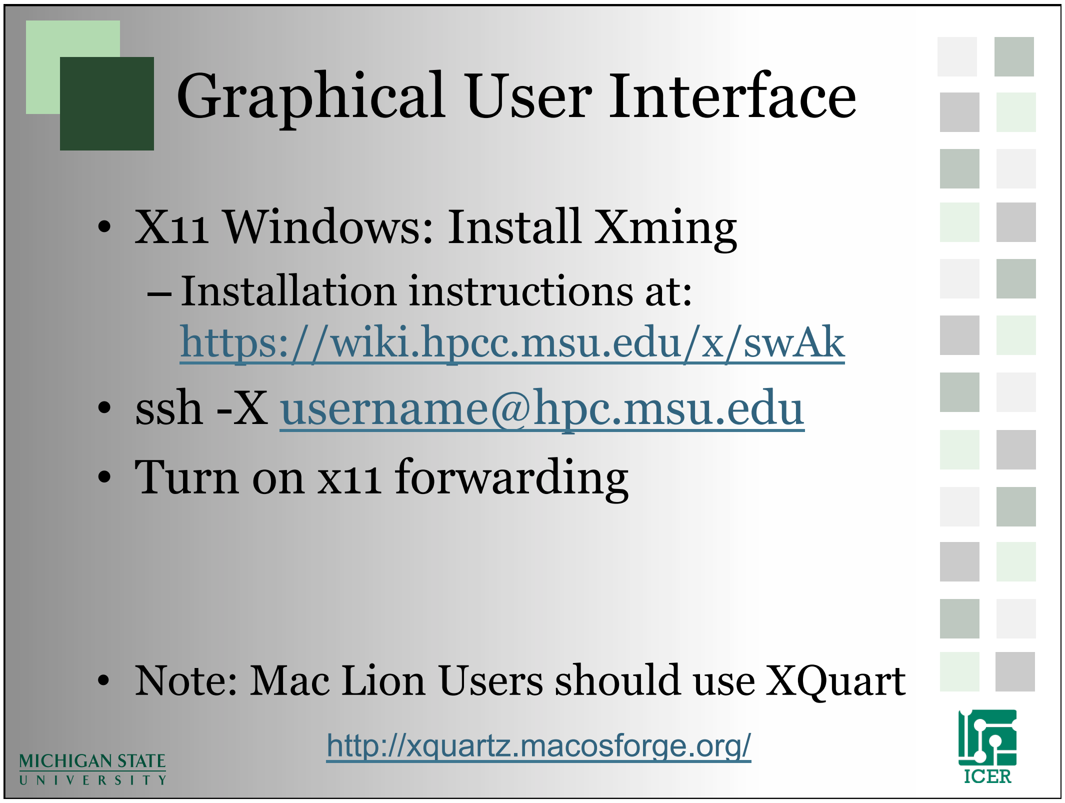


What is needed for X11

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

MICHIGAN STATE UNIVERSITY

ICER



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

http://xquartz.macosforge.org/

MICHIGAN STATE UNIVERSITY

ICER

Exercise: Transfer a file

- Try one of the following Commands

xeyes	Test X11
firefox	Run web browser

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

MICHIGAN STATE UNIVERSITY

ICER

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…

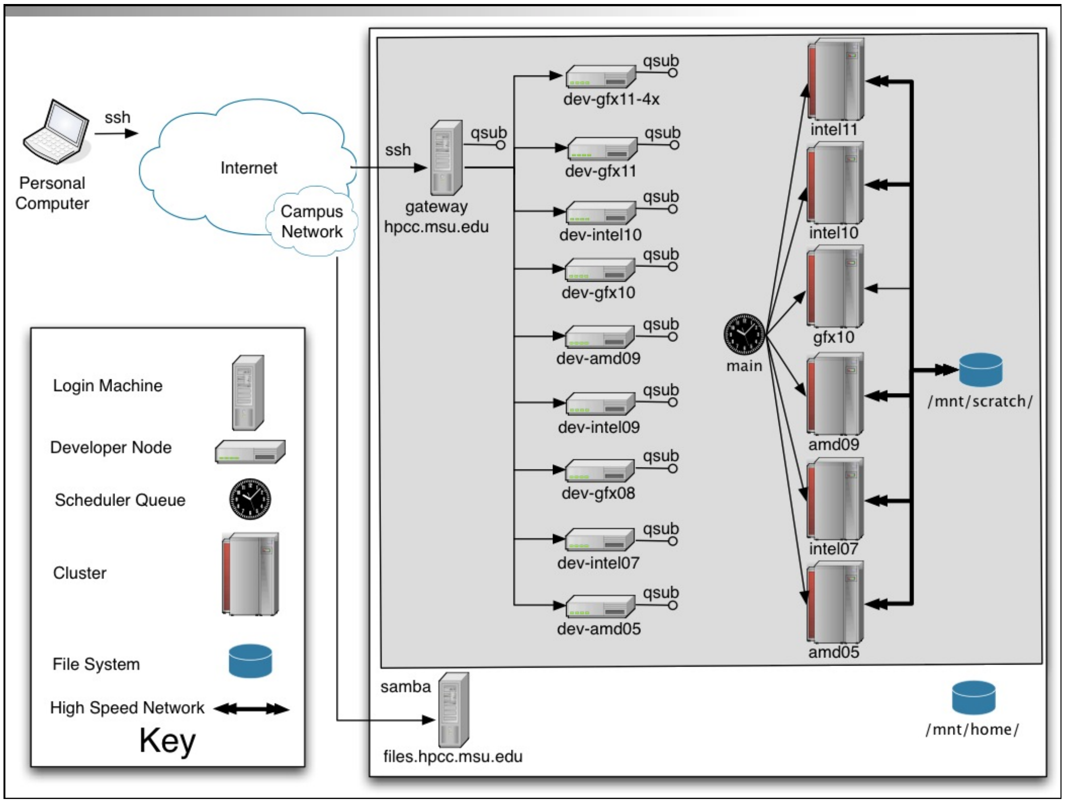
MICHIGAN STATE UNIVERSITY

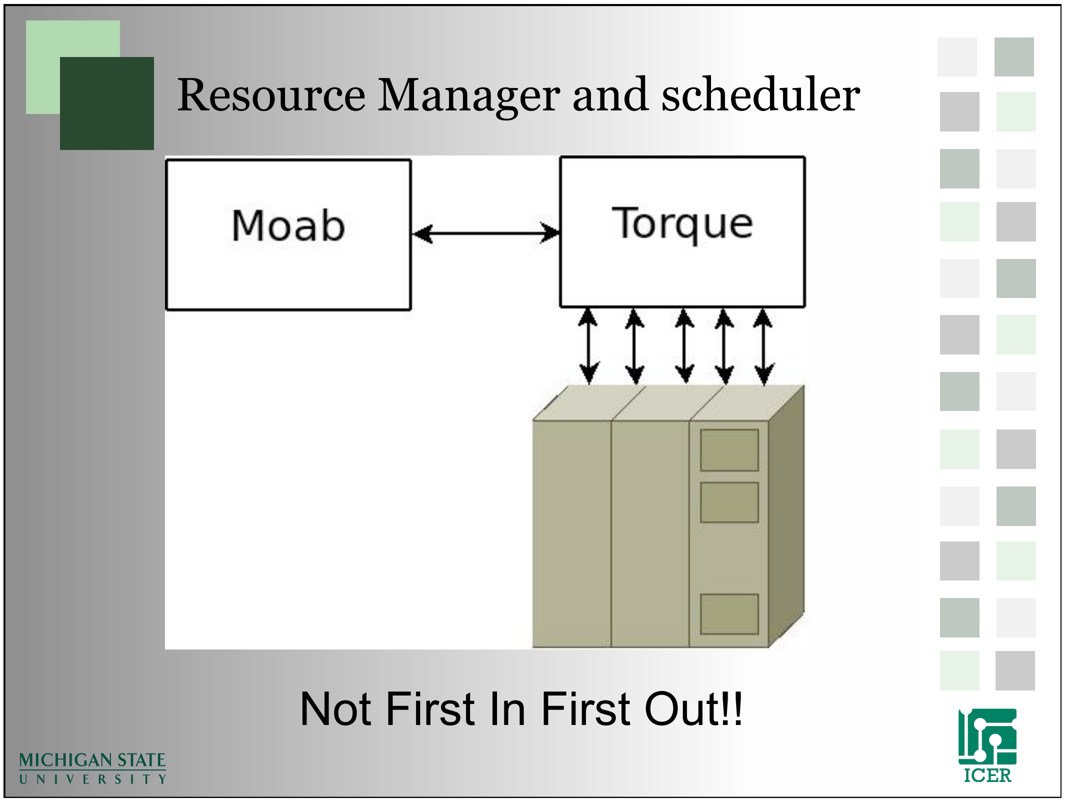
ICER

Using The Batch Scheduler

MICHIGAN STATE
UNIVERSITY

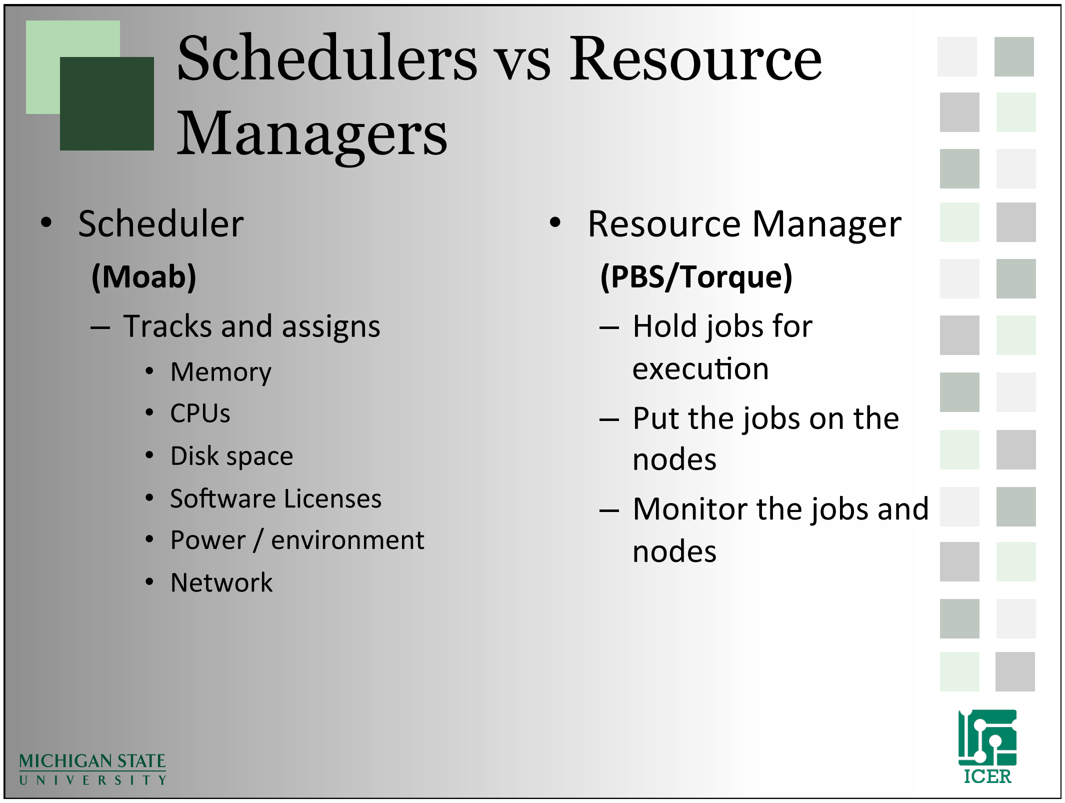
ICER





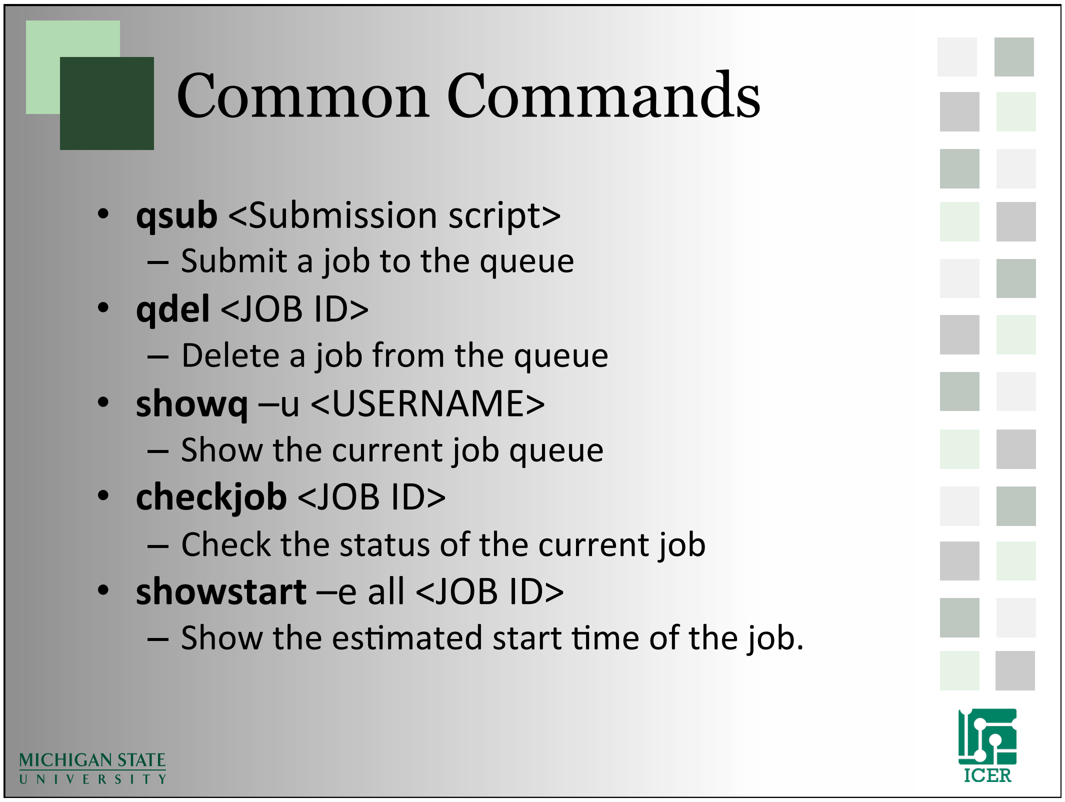
Resource Manager and scheduler

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

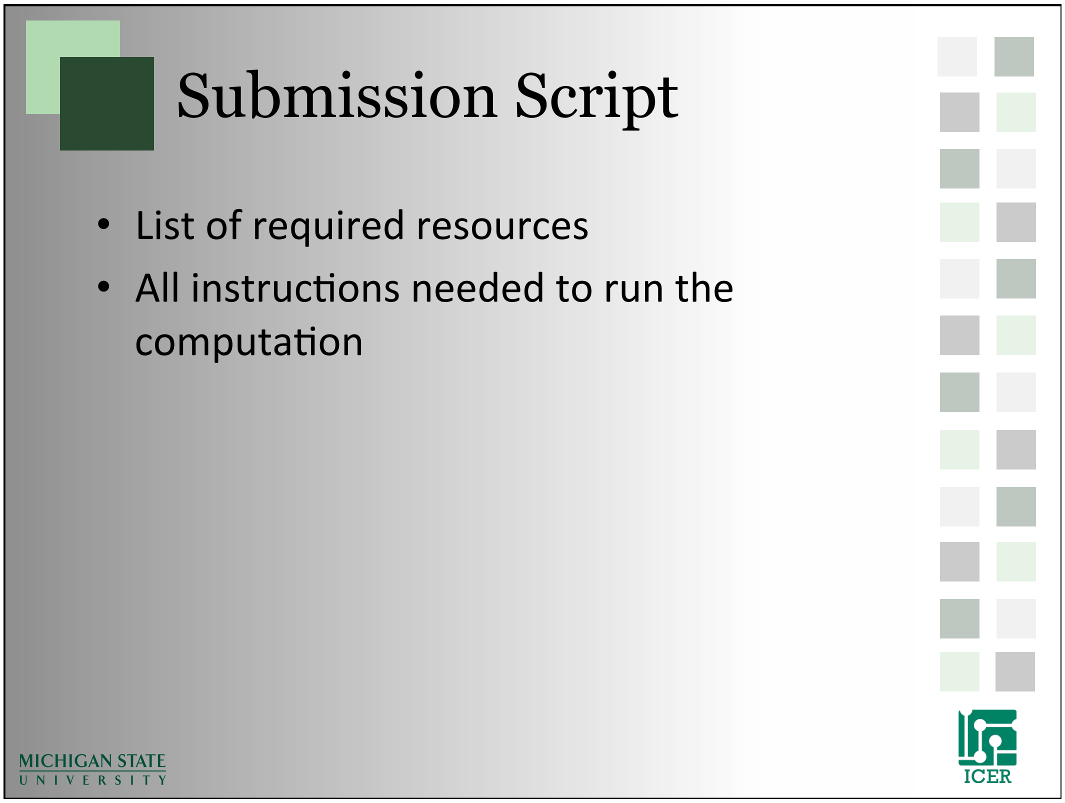


Common Commands

- **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.

MICHIGAN STATE UNIVERSITY

ICER



Submission Script

- List of required resources
- All instructions needed to run the computation

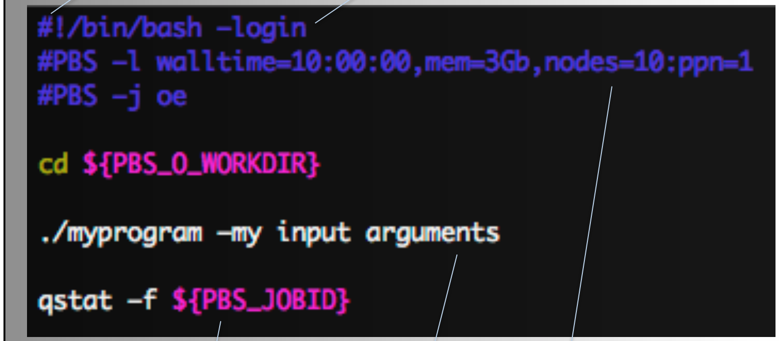
MICHIGAN STATE UNIVERSITY

ICER

Typical Submission Script

Shell Comment

Define Shell



```
#!/bin/bash -login
#PBS -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

MICHIGAN STATE
UNIVERSITY

ICER

Example: Submit a job

- Go to the top helloworld directory

  `>cd ~/hpccworkshop/helloworld`

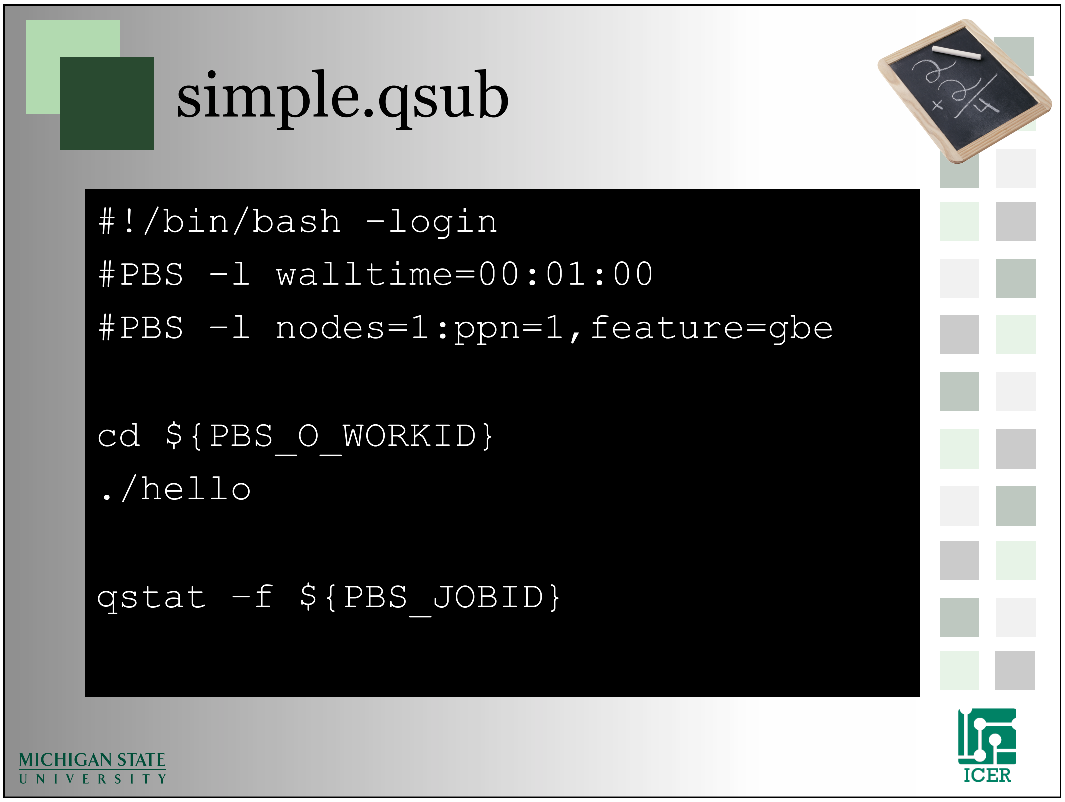
- Create a simple submission script

  `>nano simple.qsub`

- See next slide for what to type…

MICHIGAN STATE
UNIVERSITY

ICER



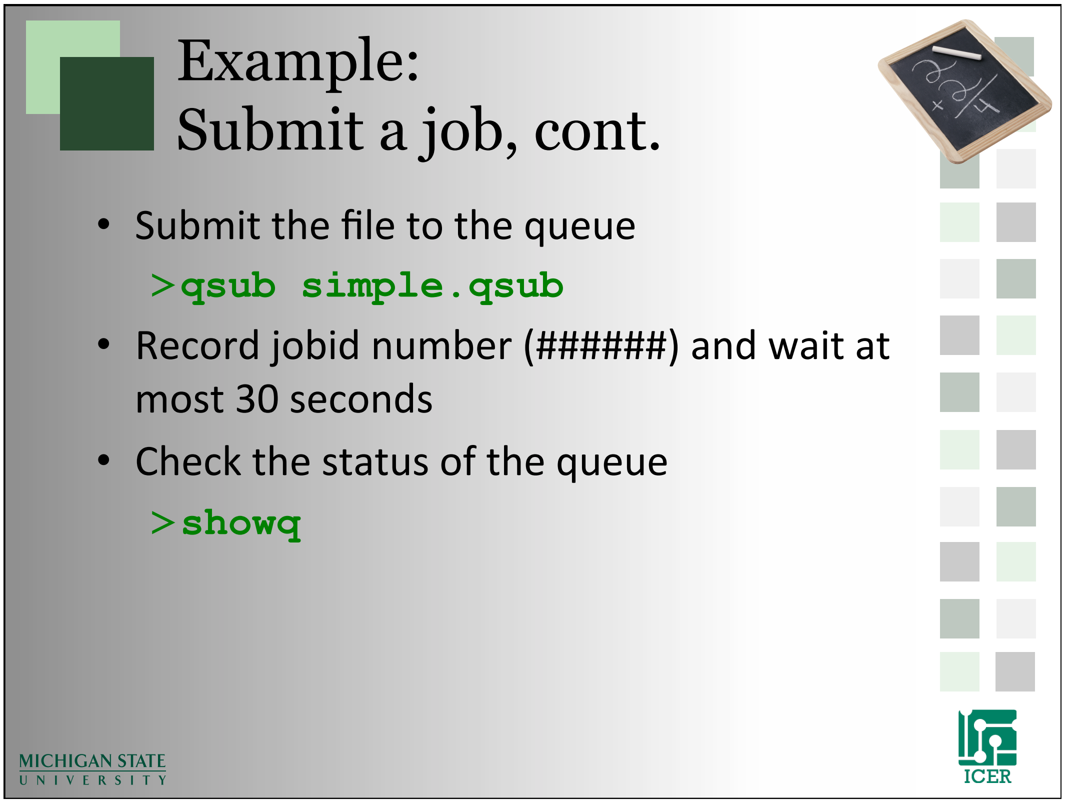
simple.qsub

```
#!/bin/bash -login
#PBS -l walltime=00:01:00
#PBS -l nodes=1:ppn=1,feature=gbe

cd ${PBS_O_WORKID}
./hello

qstat -f ${PBS_JOBID}
```

MICHIGAN STATE
UNIVERSITY
ICER



Example:
Submit a job, cont.

- Submit the file to the queue

  `>qsub simple.qsub`

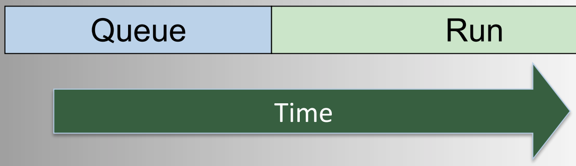
- Record jobid number (######) and wait at most 30 seconds
- Check the status of the queue

  `>showq`

MICHIGAN STATE
UNIVERSITY
ICER

Submitting a job

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



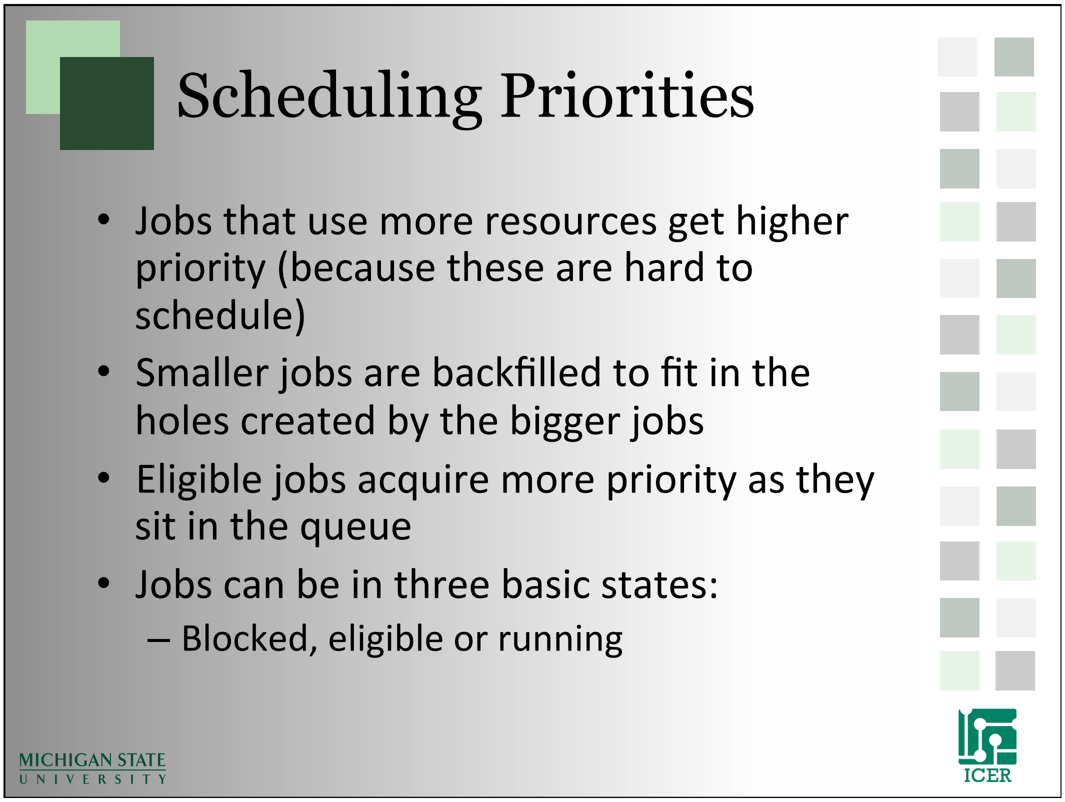
Example: Monitor a job

- Submit the file to the queue:

```
>qstat -f ######
```

- When will a job start:

```
>showstart -e all ######
```

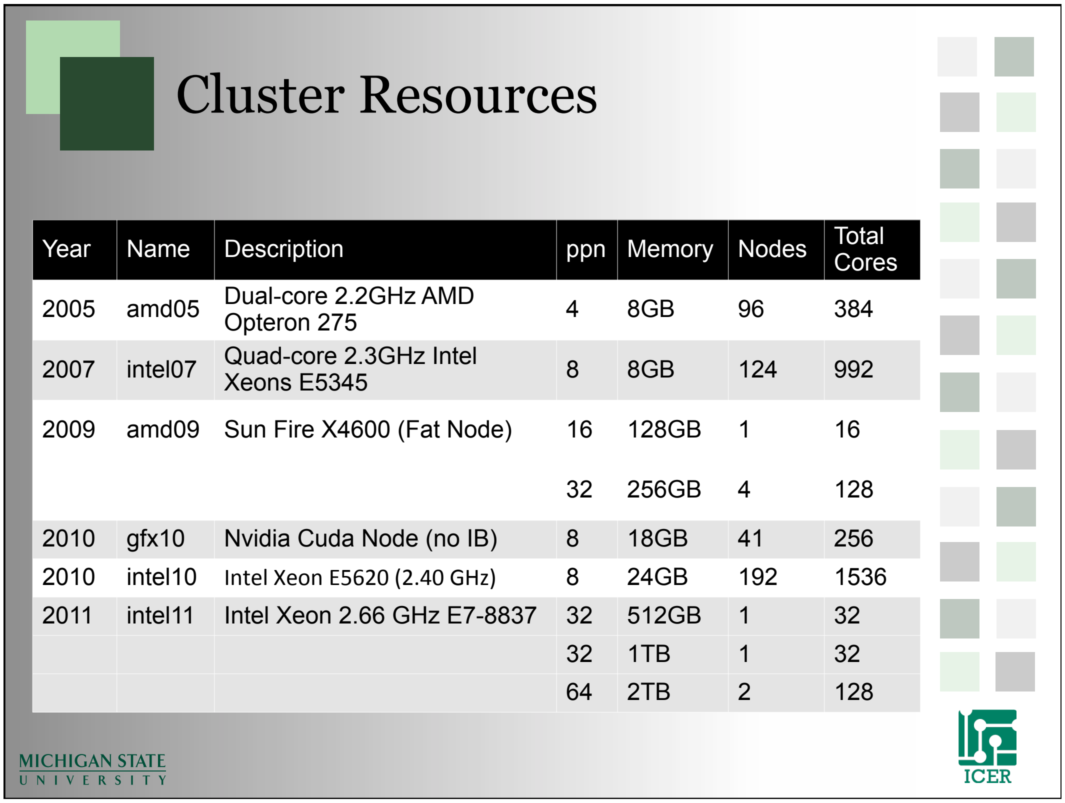


Scheduling Priorities

- 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

MICHIGAN STATE UNIVERSITY

ICER

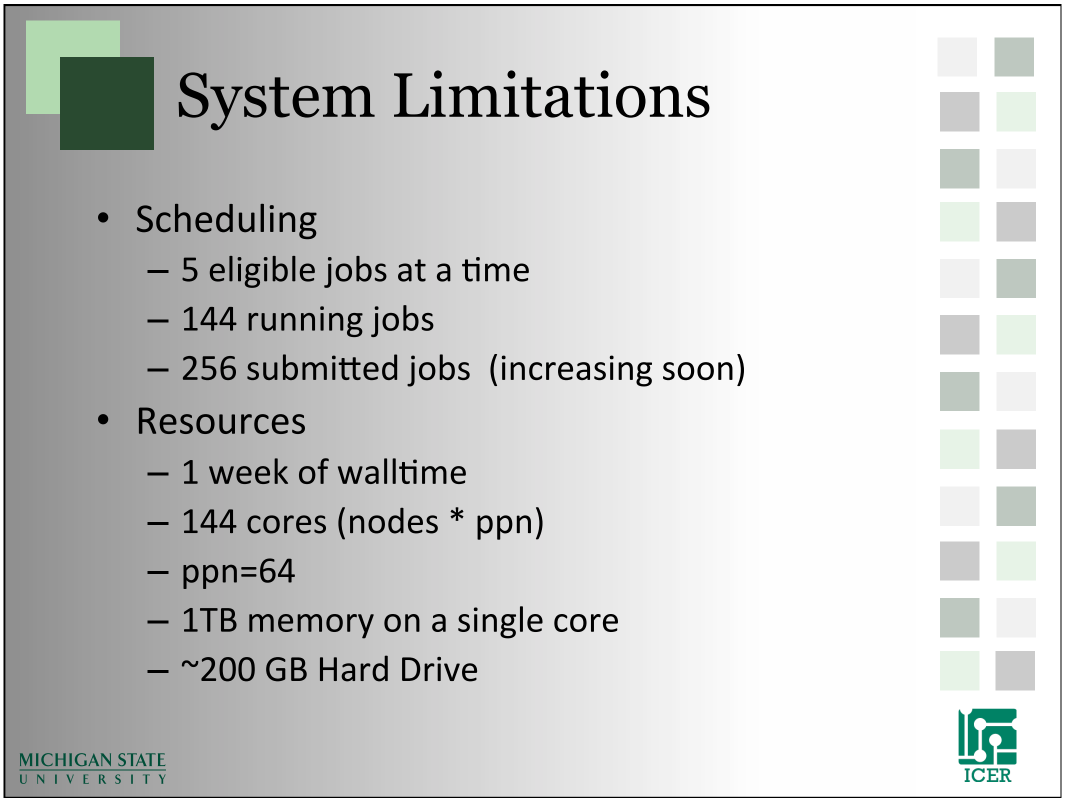


Cluster Resources

Year	Name	Description	ppn	Memory	Nodes	Total Cores
2005	amd05	Dual-core 2.2GHz AMD Opteron 275	4	8GB	96	384
2007	intel07	Quad-core 2.3GHz Intel Xeons E5345	8	8GB	124	992
2009	amd09	Sun Fire X4600 (Fat Node)	16	128GB	1	16
			32	256GB	4	128
2010	gfx10	Nvidia Cuda Node (no IB)	8	18GB	41	256
2010	intel10	Intel Xeon E5620 (2.40 GHz)	8	24GB	192	1536
2011	intel11	Intel Xeon 2.66 GHz E7-8837	32	512GB	1	32
			32	1TB	1	32
			64	2TB	2	128

MICHIGAN STATE UNIVERSITY

ICER

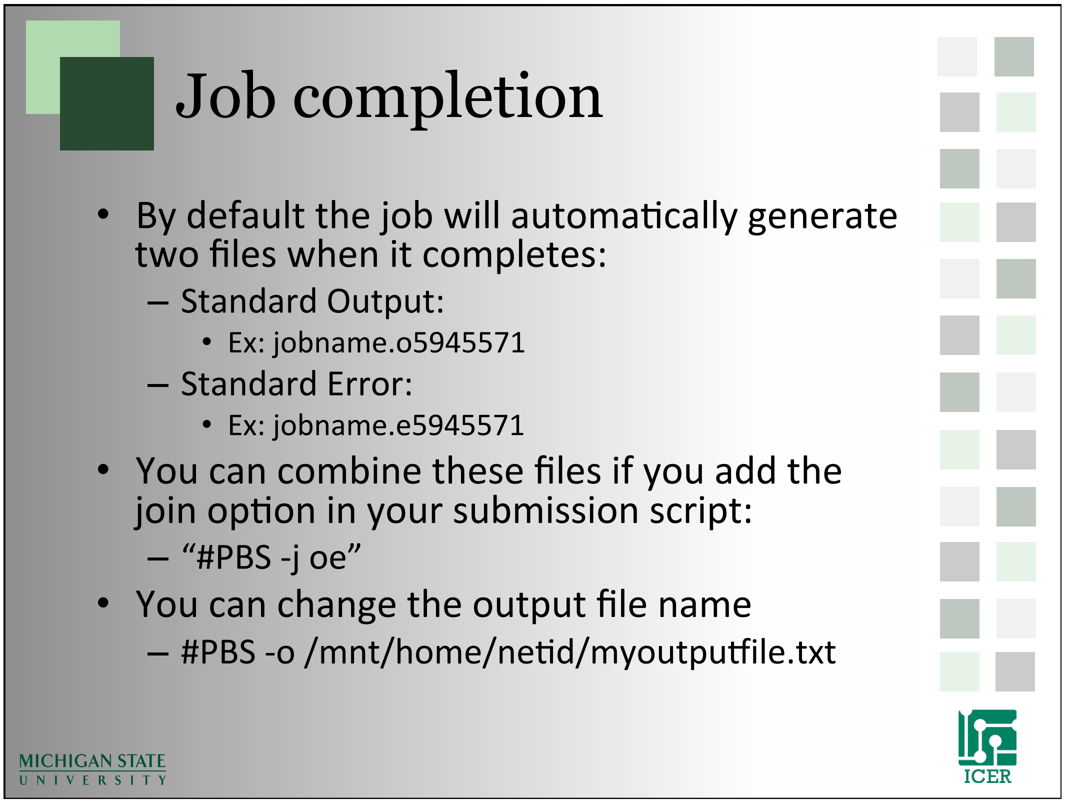


System Limitations

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

MICHIGAN STATE UNIVERSITY

ICER

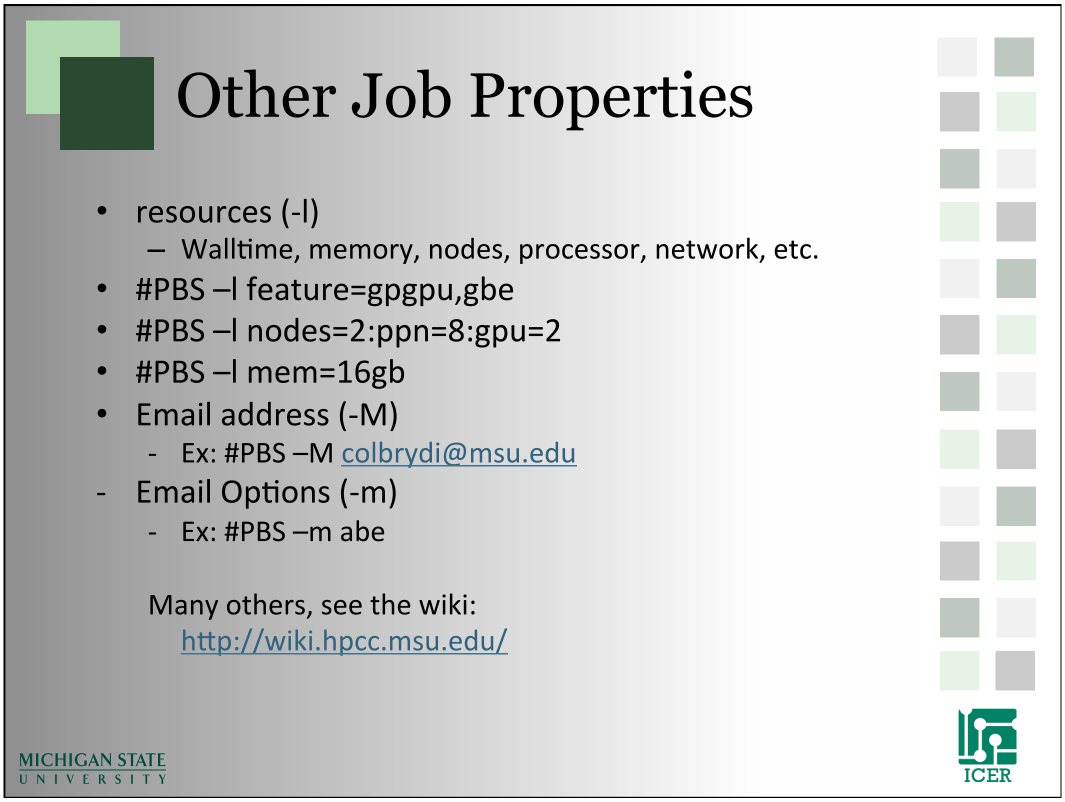


Job completion

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

MICHIGAN STATE UNIVERSITY

ICER



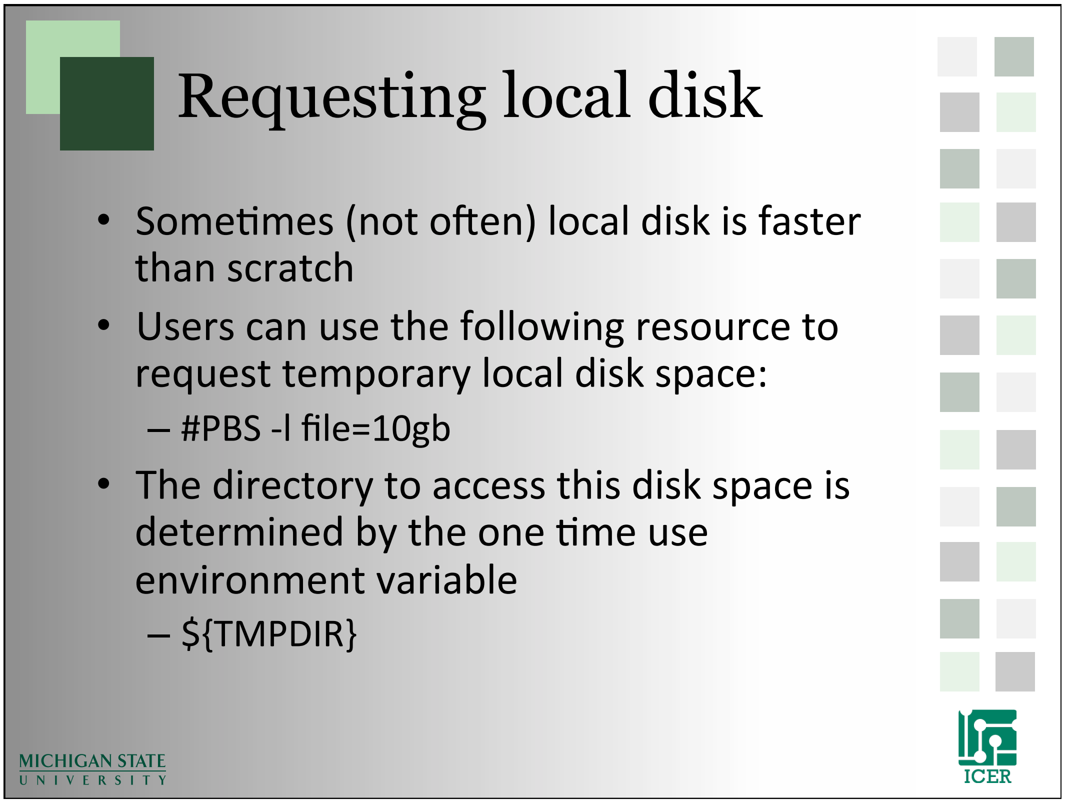
Other Job Properties

- resources (-l)
  - Walltime, memory, nodes, processor, network, etc.
- #PBS –l feature=gpgpu,gbe
- #PBS –l nodes=2:ppn=8:gpu=2
- #PBS –l 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/

MICHIGAN STATE
UNIVERSITY

ICER

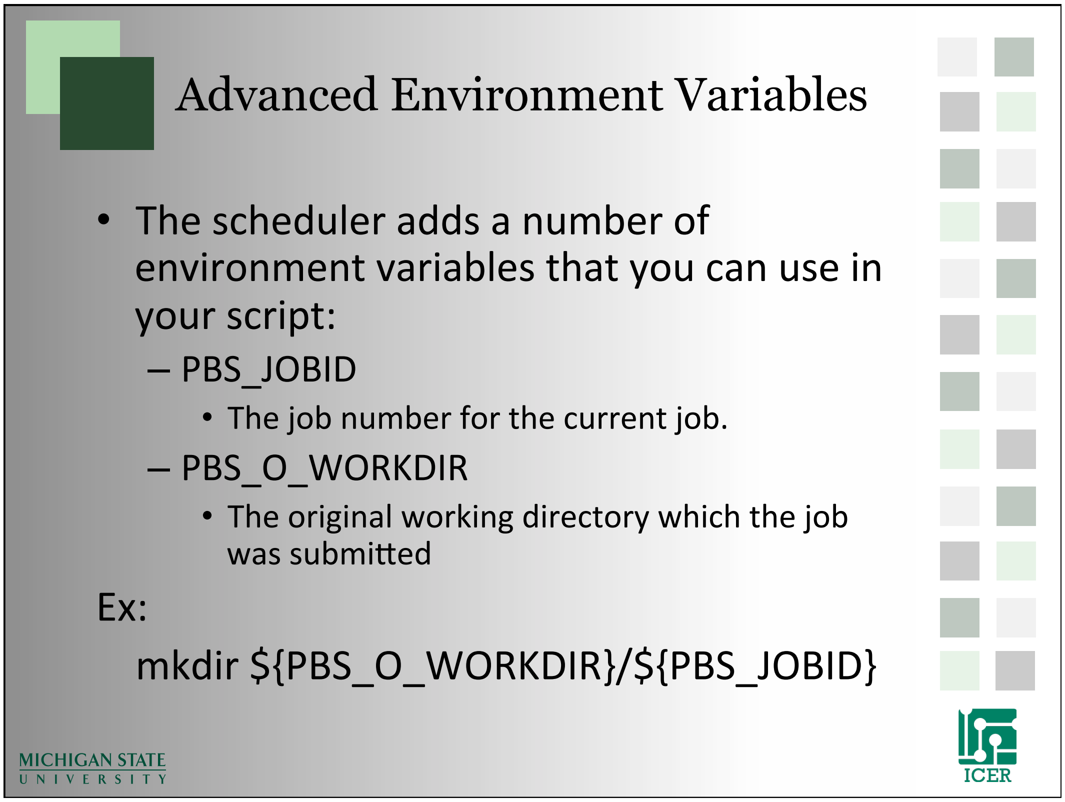


Requesting local disk

- Sometimes (not often) local disk is faster than scratch
- Users can use the following resource to request temporary local disk space:
  - #PBS -l file=10gb
- The directory to access this disk space is determined by the one time use environment variable
  - ${TMPDIR}

MICHIGAN STATE
UNIVERSITY

ICER



Advanced Environment Variables

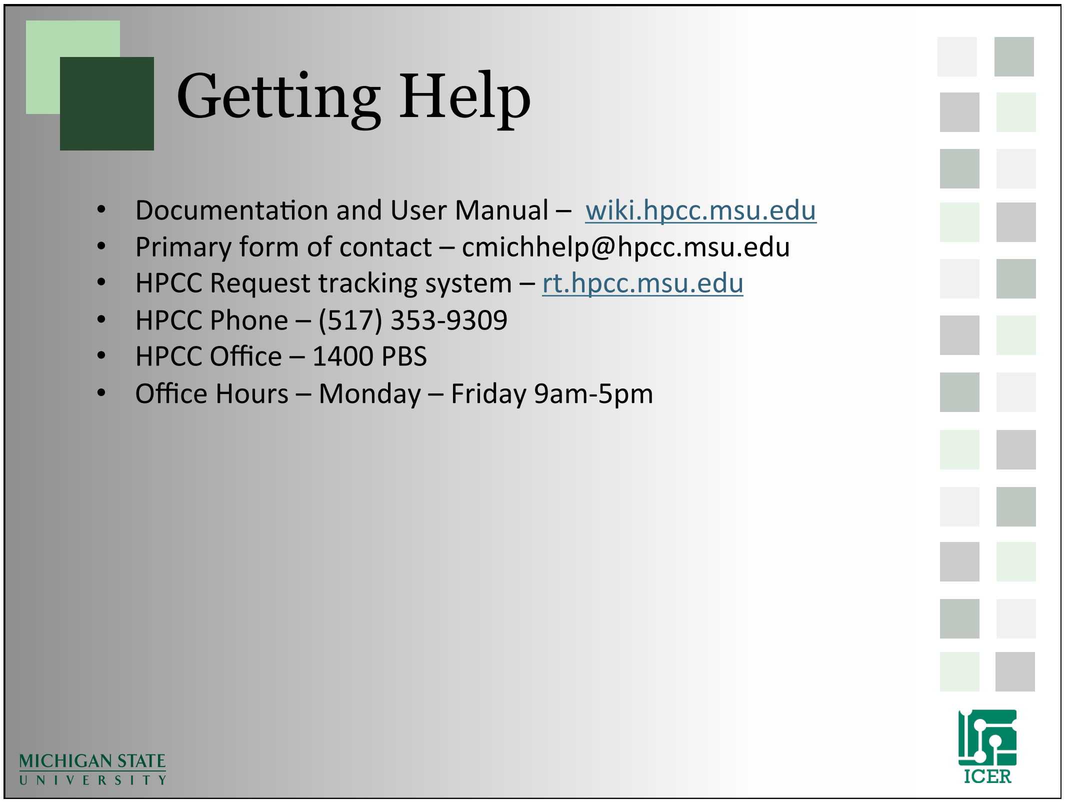
- The scheduler adds a number of environment variables that you can use in your script:
  - PBS_JOBID
    - The job number for the current job.
  - PBS_O_WORKDIR
    - The original working directory which the job was submitted

Ex:

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

MICHIGAN STATE UNIVERSITY

ICER



Getting Help

- Documentation and User Manual – wiki.hpcc.msu.edu
- Primary form of contact – cmichhelp@hpcc.msu.edu
- HPCC Request tracking system – rt.hpcc.msu.edu
- HPCC Phone – (517) 353-9309
- HPCC Office – 1400 PBS
- Office Hours – Monday – Friday 9am-5pm

MICHIGAN STATE UNIVERSITY

ICER