

PCSE

Intro to HPC Systems

1/17/2013

Instructors:

Kent Milfeld, TACC

Victor Eijkhout, TACC

ACCESS to TACC HPC Systems

- One username, one password work everywhere.
- Use SSH to access systems: (See SSH in User Guides)
`ssh <username>@lonestar.tacc.utexas.edu`
`ssh <username>@stampede.tacc.utexas.edu`

User Guides: www.tacc.utexas.edu
(User Services → User Guides)

Need to know: Basics of a system.

Getting Started: login in, get files, edit, compile code, run code, submit jobs.

Scientific Computing Terminology

Terms

Definition

- HPC System —————>• “High Performance Computing” Computer– Computers Connected through high speed interconnect and configured for scientific computing.
- Interconnect —————>• The wiring, chips, and software that connects computing components.
- Node (blade, sled, etc.) —————>• An independent computing unit of an HPC System. Unit has its own Operating System (and memory). The physical cases of a node are often called blades and sleds.
- Chassis —————>• Nodes are often aggregated into a chassis (with a backplane) for sharing electrical power, cooling and sharing a local interconnect.

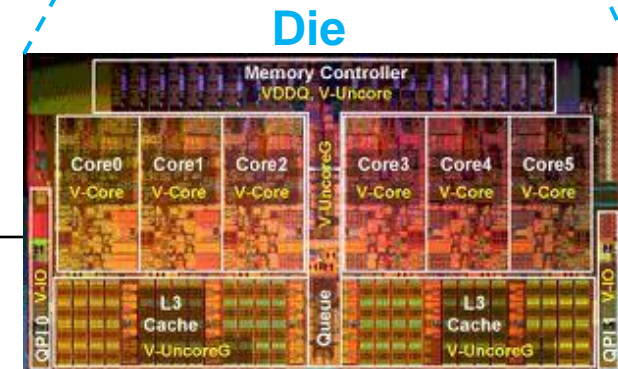
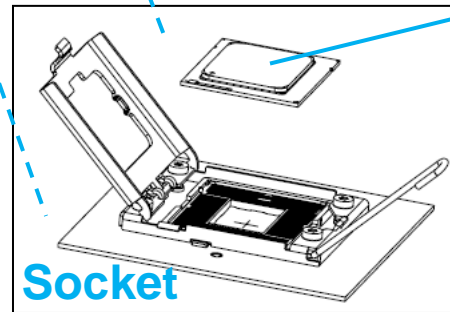
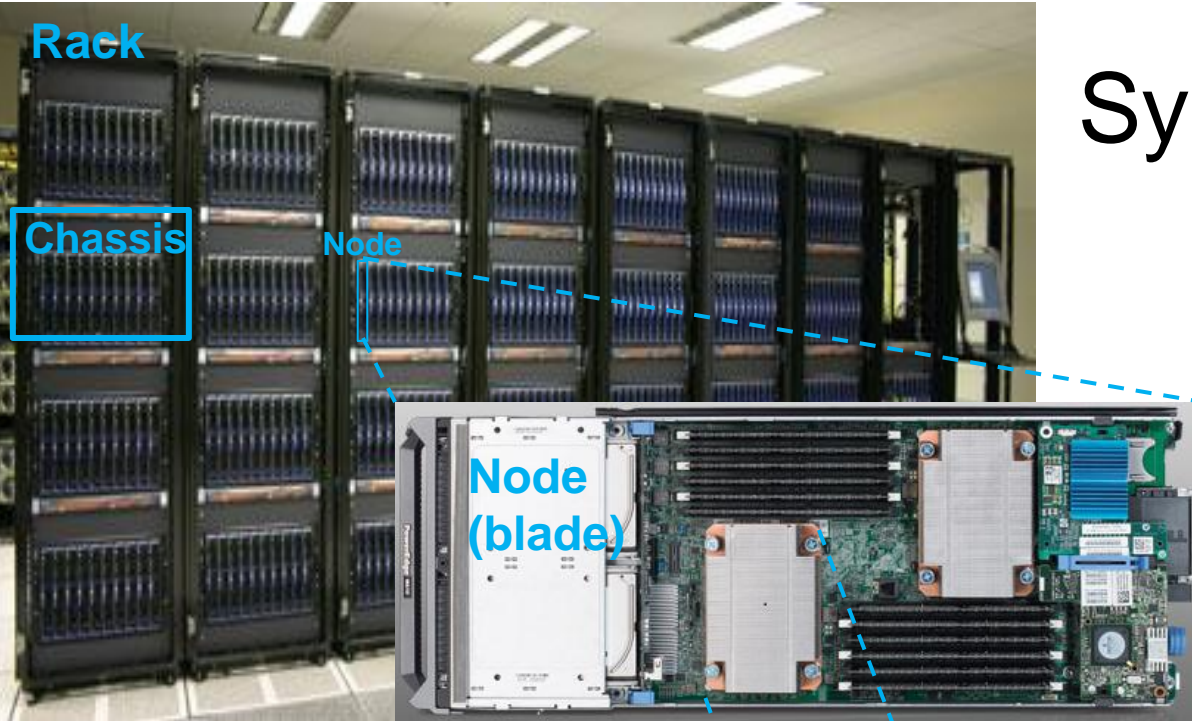
Scientific Computing Terminology

Terms

Definition

- Chip or Die → • A set of self-contained circuits on a single media of size ~20mm x 20mm, containing up to ~1 billion transistors.
- Socket → • A hardware contraption that provides a connection between and chip and a motherboard.
- CPU (or processor?) → • A Central Processor Unit, consisting of a Chip or Die. (often called a processor)
- Core → • Modern CPUs contain multiple cores. A core is an execution unit in a CPU that can execute a code's instructions independently (simultaneous while other cores execute a different code's instructions).
- Hyper-Threading → • A single core can have additional circuitry that allows two or more instruction streams (threads) to proceed through a single core "simultaneously". Hyper-Thread is an Intel trademark for 2 threads. Xeon Phi Coprocessor supports 4 threads.

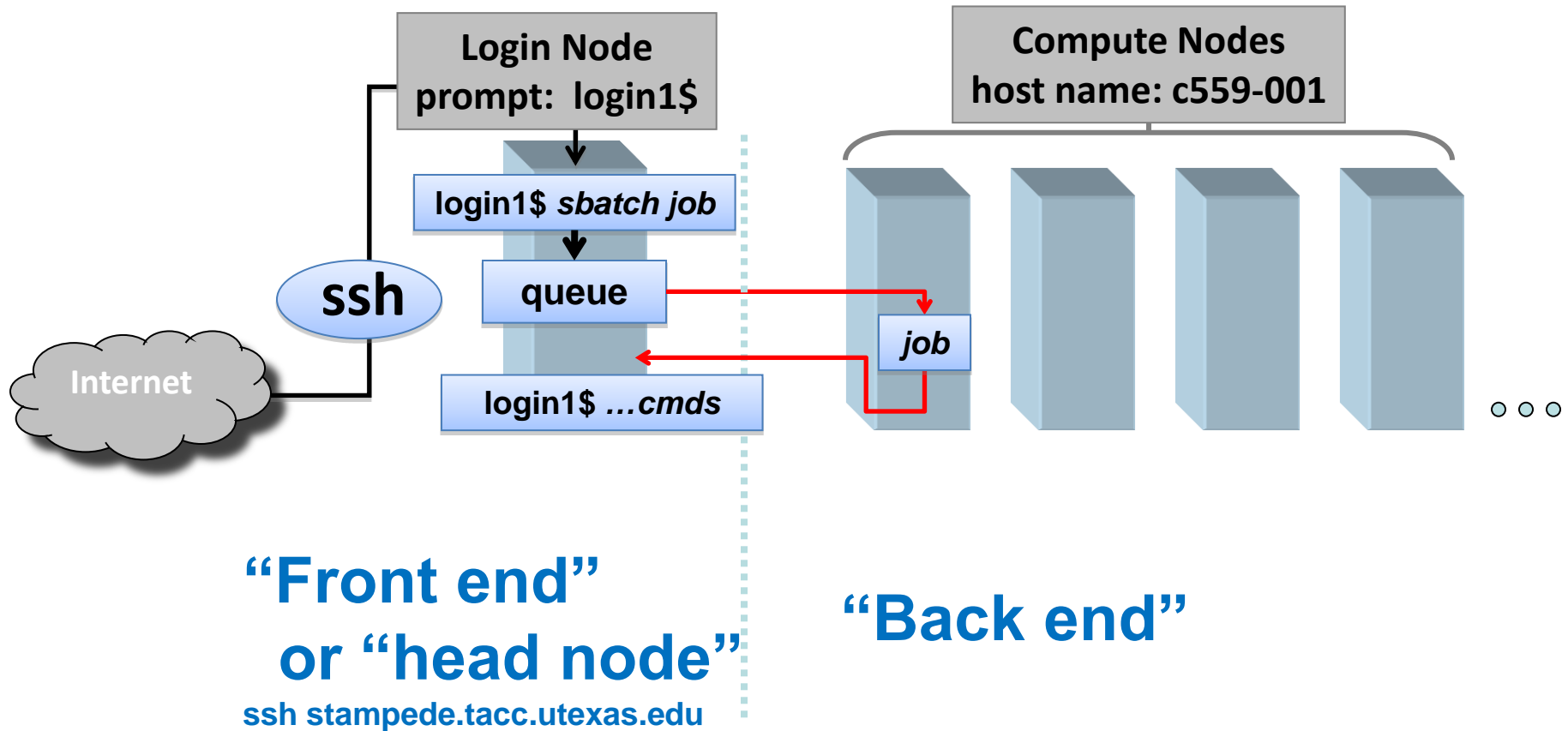
System Hierarchy



- Start with a Linux-like terminal or equivalent connected to internet
 - Linux command line
 - Mac terminal app
 - PuTTY, Secure Shell Client, GSI-SSH on XSEDE portal,...
- Pick a favorite editor; become proficient
 - nano – simple
 - vi (vim) – terse
 - emacs – powerful

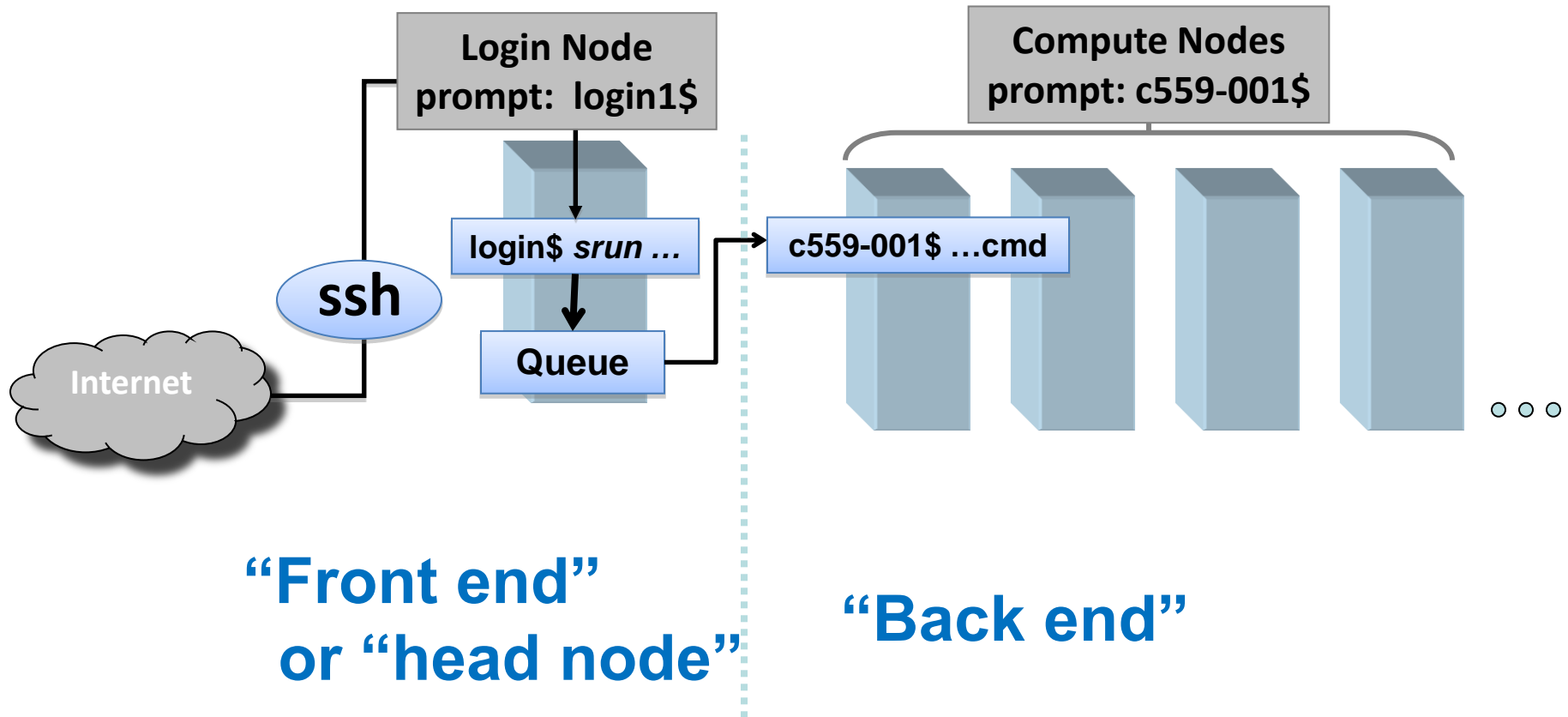
Nodes Have Personalities and Purposes

Stampede

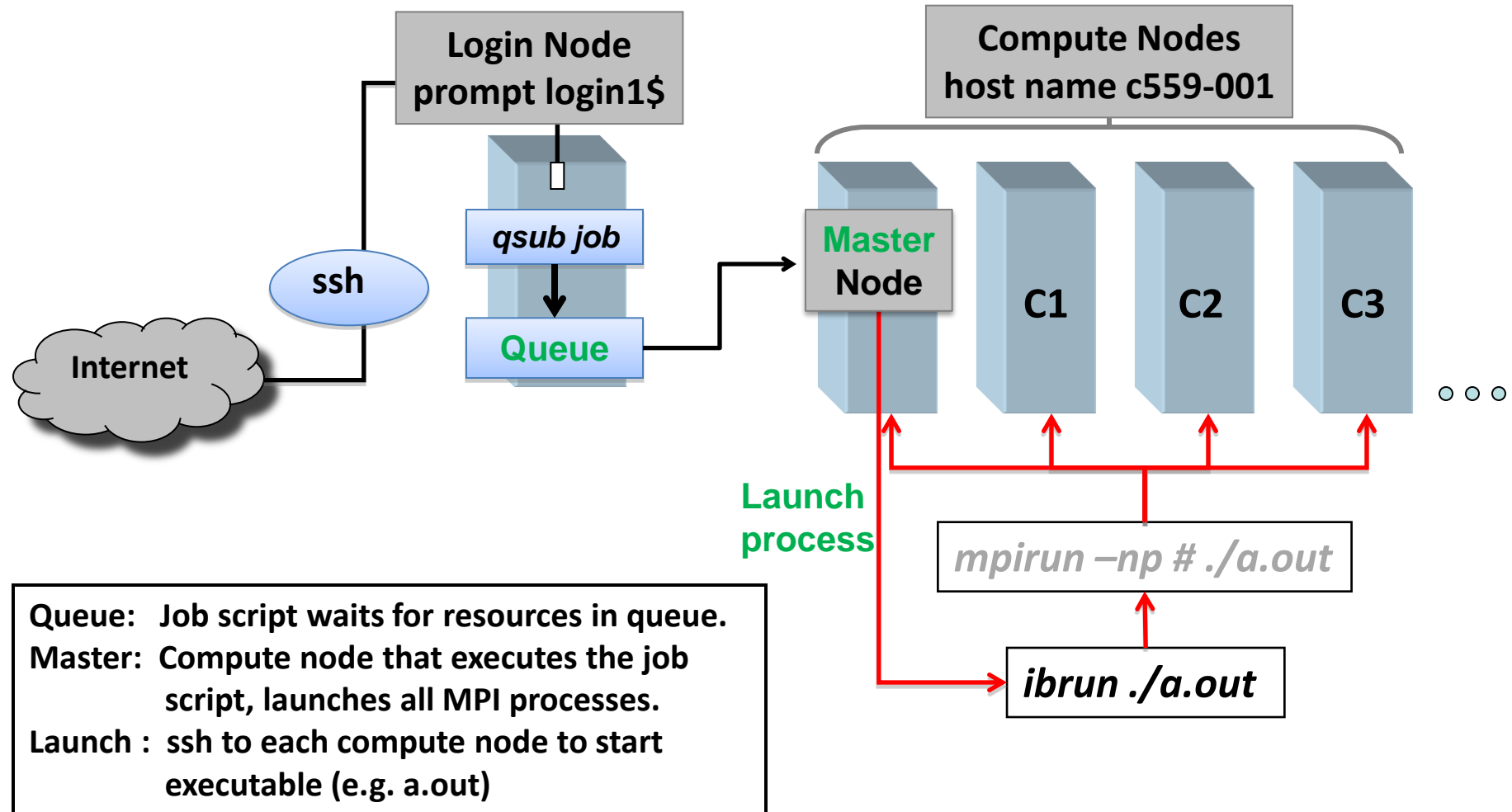


Nodes Have Personalities and Purposes

Stampede



Batch Submission Process



Preferred Resource

- Which system should I work on?

Stampede

6,400 nodes (“PCs”)
Interconnect **FDR**
2 CPUs + **MIC Accelerator**
(8+8 cores) + **61 cores**
(32 GB Memory) + **8GB**

SGE Batch System

Intel compilers

Lonestar

1,888 nodes (“PCs”)
Interconnect **QDR**
2 CPUs + **MIC Accelerator**
(6 + 6 cores)
(24 GB Memory)

SLURM Batch System

Intel compilers

Getting Familiar with the System

1. `mkdir tests` (make a directory called tests)
2. `cd tests` (go to the tests directory)
3. `cp ~pcse00/examples/job_submission/* .`
(get example files)
("*" means all file)
(".") means present directory)

`mkdir` = make directory command

`cd` = change to directory command

`cp` = copy command

`~pcse00` evaluates to `/home1/00092/pcse00`

Getting Familiar with the System

4. `mpif90 mpihello.f90` (compile Fortran code)
`mpicc mpihello.c` (compile C code)
5. `cat job` (display job file)
6. Look over the job script.
 - a) 1st line sets the shell to use
 - b) “#” lines are comments about resources— this is read by the batch system
 - c) Other statements are Unix commands (information)
 - d) `ibrun ./a.out` executes `a.out` on all nodes.

`mpif90` and `mpicc` are Fortran and C MPI compilers
`cat` is the Unix “concatenate” (print to screen) cmd.

(Magic) Job Scripts

Lonestar SGE

```
#!/bin/bash
#$ -N test1
#$ -j y
#$ -o $JOB_NAME.o$JOB_ID
#$ -q development
#$ -l h_rt=00:10:00
#$ -pe 12way 24
```

```
#$ -A TACC-PCSE
#$ -V
#$ -cwd
```

```
echo "master: `hostname`"
echo "PWD_DIR: `pwd`"
```

```
ibrun ./a.out
```

shell for job script
job name
merge stderr & stdout
job output
job queue
time hh:mm:ss
Total Cores

Account
Inherit submission env.
Start in submission
director

Stampede SLURM

```
#!/bin/bash
#SBATCH -J test1
#SBATCH -o test1.o%j
#SBATCH -p development
#SBATCH -t 01:30:00
#SBATCH -n 32
```

```
#SBATCH -A TACC-PCSE
```

```
echo "master: `hostname`"
echo "PWD_DIR: `pwd`"
```

```
ibrun ./a.out
```

Getting Familiar with the System

Lonestar

Stampede

- | | | |
|---------------------------------|---------------------------|----------------|
| 7. qsub job | SBATCH job | (submit job) |
| 8. qstat | squeue -u <i>username</i> | (monitor jobs) |
| 9. showq -u | showq -u | (monitor jobs) |
| 10. Check output in test1.o#### | | |

Interactive Access

Use idev to access a Lonestar compute node:

```
login1% idev
```

```
Idev needs to insert a few startup commands in your /home/.../milfeld/.login file.  
(To see the commands, set the IDEV_LOGIN_CMS to yes, and rerun idev.)
```

```
May I insert the commands? (yes,no, default is yes) >yes
```

```
...
```

```
Your job 44996 ("id27657") has been submitted
```

```
Environment job status: qw
```

```
Environment job status: qw
```

```
...
```

```
C205-199%
```

Use srun to access a Stampede compute node:

```
login1% srun -A TACC-PCSE -p development -t 60 -n16 --pty /bin/bash -l
```

```
...
```

```
--> Checking available allocation...(TACC-PCSE)...OK
```

```
C559-001%
```



OR do this once per login session: `login1% source ~pcse00/sourceme`
for each interactive session : `login1% srun1`

End of Environment

- More to come later: ...
- Read the System Overview, System Access, Computing Environment and Transferring Files of the Stampede User Guide.

Questions?

UNIX/LINUX

- The Internet provides tutorials/help at all levels
 - <http://www.ee.surrey.ac.uk/Teaching/Unix/>
 - <http://wiki.linuxquestions.org/wiki/LinuxIntro>
 - <http://www.linux-tutorial.info/>
 - Use google, bing, etc.

Editors

New to Unix/Linux

- Use emacs (<http://www.linuxhelp.net/guides/emacs>)
 - Invoke with X11 window: `emacs <file>`
 - Use the GUI to edit text
 - Requires ssh -X
 - Invoke without X11 window: `emacs -nw <file>`
 - Use commands to edit text
 - Save file: CTRL-X, CTRL-S
 - Exit file: CTRL-X, CTRL-C
 - etc.

Familiar with Unix/Linux

- Use what you are familiar with, if it is available
 - emacs, vi, etc.

UNIX Shells

- Your account uses the `bash` as the login shell
- You can change to the `tcsh` shell, if you are more familiar with “C-type shells”
 - How to: Use `chsh` on Lonestar. Submit a ticket to change your shell on Stampede (for now).