



VirginiaTech
Invent the Future



Getting Started on HokieSpeed

Advanced Research Computing

Outline

- Guest wireless access: ConnectToVT-Wireless
- Windows users: download putty
- Log in to HokieSpeed
- Overview of HokieSpeed user environment and module structure
- Fundamentals of job submission and monitoring
- NVIDIA Training

Important Login Information

- Account sheets provide login information – username, password and hostname
- Windows Users: Download PuTTY
 - Google PuTTY
 - First result
 - Save putty.exe to Desktop

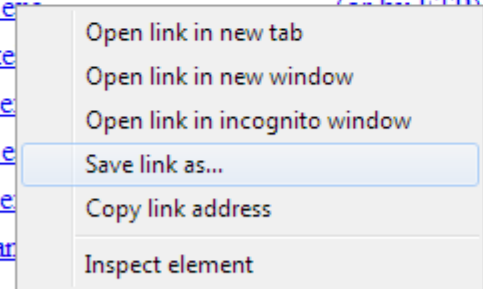
Binaries

The latest release version (beta 0.62). This will generally be a version I t (below) to see if I've already fixed the bug, before reporting it to me.

For Windows on Intel x86

PuTTY: [putty.exe](#) (or by FTP)
PuTTYtel: [puttytel.exe](#)
PSCP: [pscp.exe](#)
PSFTP: [psftp.exe](#)
Plink: [plink.exe](#)
Pageant: [pageant.exe](#)
PuTTYgen: [puttygen.exe](#) (or by FTP)

A ZIP file containing all the binaries (except PuTTYtel) and also t



Getting Started Steps

1. Sheet distributed provides your training account username, password and hostname
2. Log in to HokieSpeed via ssh
3. System examples
 - a. Edit job submission script
 - b. Submit to job script scheduler
4. Compile and submit your own programs

Sample Unix Commands

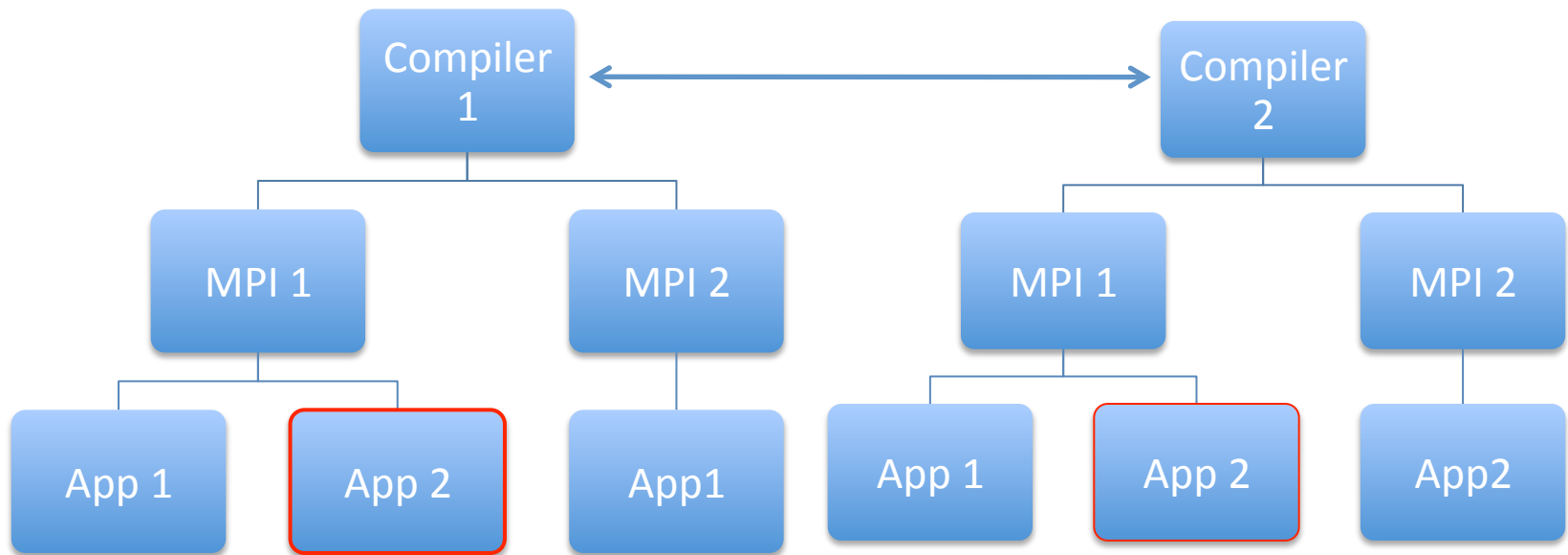
Command	Meaning
cd	Change directories
ls	View contents of a directory
mkdir	Make directory
rm, rmdir	Remove a file or directory
mv	Move or rename a file
cp	Copy a file
cat	Create a file or concatenate files
more	View a text file page by page
vi	Text editor
man	Search for or learn about a command
exit	Log out of system

USER ENVIRONMENT

Modules

- Modules are used to manage your environment by setting the \$PATH and other environment variables
- Modules can be loaded at the command line and/or in job scripts
- Each application is built with a certain compiler (PGI, gcc, intel) and a certain software stack (openmpi, mvapich2, etc.)

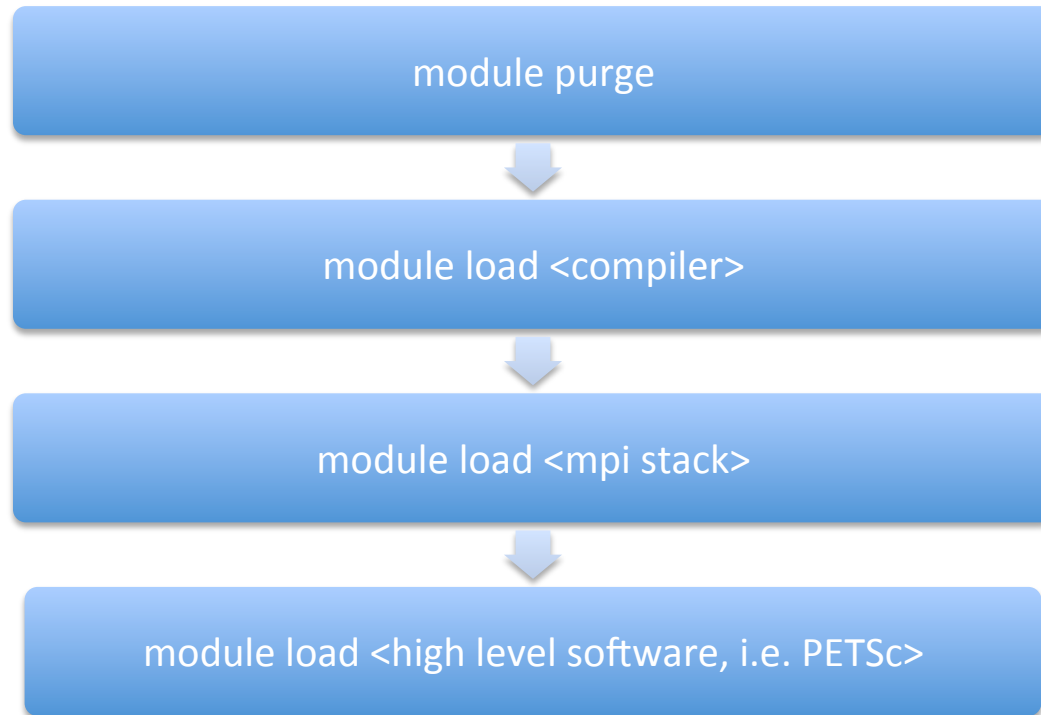
Hierarchical Module Structure



Module commands

<code>module</code>	list options
<code>module list</code>	list loaded modules
<code>module avail</code>	list available modules
<code>module load <module></code>	add a module
<code>module unload <module></code>	remove a module
<code>module swap <mod1> <mod2></code>	swap two modules
<code>module help <module></code>	module environment
<code>module show <module></code>	module description
<code>module reset</code>	reset to default
<code>module purge</code>	unload all module

Modules



- The default modules are provided for minimum functionality.
- Module dependencies against choice of compiler and MPI stack are automatically taken care of.

Modules: Things to Try Right Now

List currently loaded modules

```
> module list
```

Unload all modules from your environment

```
> module purge
```

Load the module for the PGI compiler

```
> module load pgi
```

```
> module show pgi
```

JOB SUBMISSION & MONITORING

Job Submission

- Submission via a shell script
 - Job description: Nodes, processes, run time
 - Modules & dependencies
 - Execution statements
- Submit job script via the qsub command:
`qsub <job_script>`
- Scheduler will place your job on compute nodes subject to availability

Prepare Submission Script

1. Copy sample script: `cp /home/TRAINING/Job.sh`
2. Edit sample script as desired:
 - a. Walltime
 - b. Resource request (nodes/ppn)
 - c. Module commands (add PGI module)
 - d. Command to run your job
3. Save it with a new name if you like

Typical Submission Script

```
#!/bin/bash

#PBS -l walltime=00:10:00
#PBS -l nodes=1:ppn=12
#PBS -q normal_q
#PBS -W group_list=training
#PBS -A hokiespeed

Module purge
module load pgi

cd $PBS_O_WORKDIR
echo "Running job on compute node!"
./runme

exit;
```

Text Editors

Use `'vi'` to edit text files from the command line

- <http://www.arc.vt.edu/resources/software/unix/editors.php>
- <http://www.unix-manuals.com/tutorials/vi/vi-in-10-1.html>

Submit the job

1. Edit your job submission script

2. Submit the job:

```
qsub ./Job.sh
```

3. HokieSpeed returns job number:

```
1234.master.cluster
```

Wait for job to complete

1. Check job status:

```
qstat -f 1234 or qstat -u <username>  
checkjob -v 1234
```

2. When complete:

1. Job output: `Job.sh.o1234`
2. Errors: `Job.sh.e1234`

Job Monitoring

- Determine job status, and if pending when it will run

Command	Meaning
<code>checkjob -v JOBID</code>	Get the status and resource of a job
<code>qstat -f JOBID</code>	Get status of a job
<code>showstart JOBID</code>	Get expected job start time
<code>qdel JOBID</code>	Delete a job
<code>pbsnodes</code>	Show status of cluster nodes

Thank you.

Questions?