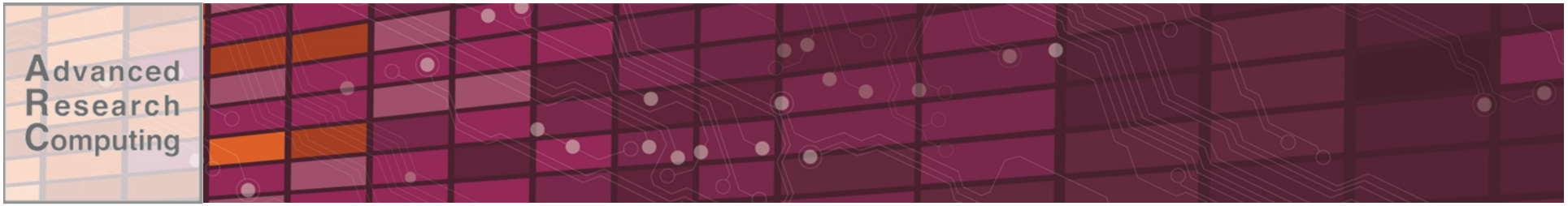


Introduction to ARC Resources and User Environment

Advanced Research Computing

September 9, 2015

ARC Resources

Advanced Research Computing

Compute Resources

System	Usage	Nodes	Node Description	Special Features
Ithaca	Beginners, MATLAB	79	8 cores, 24GB (2× Intel Nehalem)	10 double-memory nodes
HokieOne	Shared, Large Memory	82	6 cores, 32GB (Intel Westmere)	2.6TB shared-memory
HokieSpeed	GPGPU	201	12 cores, 24 GB (2× Intel Westmere)	402 Tesla C2050 GPU
BlueRidge	Large-scale CPU, MIC	408	16 cores, 64 GB (2× Intel Sandy Bridge)	260 Intel Xeon Phi 4 K40 GPU 18 128GB nodes
NewRiver	Large-scale, Data Intensive	134	24 cores, 128 GB (2× Intel Haswell)	8 K80 GPGPU 16 “big data” nodes 24 512GB nodes 2 3TB nodes

Advanced Research Computing

Storage Resources

Name	Intent	File System	Environment Variable	Per User Maximum	Data Lifespan	Available On
Home	Long-term storage of files	GPFS (NewRiver) NFS (Other)	$HOME	500 GB (NewRiver) 100 GB (Other)	Unlimited	Login and Compute Nodes
Group	Shared Data Storage for Research Groups	GPFS (NewRiver)	$WORK	TBD	Unlimited	Login and Compute Nodes
Work	Fast I/O, Temporary storage	GPFS (NewRiver) Lustre (BlueRidge) GPFS (Other)	$WORK	20 TB (NewRiver) 14 TB (Other) 3 million files	120 days	Login and Compute Nodes

Advanced Research Computing

Storage Resources (continued)

Name	Intent	File System	Environment Variable	Per User Maximum	Data Lifespan	Available On
Archive	Long-term storage for infrequently-accessed files	CXFS	$ARCHIVE	-	Unlimited	Login Nodes
Local Scratch	Local disk (hard drives)		$TMPDIR	Size of node hard drive	Length of Job	Compute Nodes
Memory (tmpfs)	Very fast I/O	Memory (RAM)	$TMPFS	Size of node memory	Length of Job	Compute Nodes

Advanced Research Computing

Visualization Resources

- VisCube: 3D immersion environment with three 10’ by 10’ walls and a floor of 1920×1920 stereo projection screens
- DeepSix: Six tiled monitors with combined resolution of 7680×3200
- ROVR Stereo Wall
- AISB Stereo Wall

GETTING STARTED ON ARC'S SYSTEMS

Getting Started Steps

- Apply for an account
- Log in (SSH) into the system
- System examples
  - Compile
  - Test (interactive job)
  - Submit to scheduler
- Compile and submit your own programs

Advanced Research Computing

ARC Accounts

- Review ARC's system specifications and choose the right system(s) for you
  - Specialty software
- Apply for an account online
- When your account is ready, you will receive confirmation from ARC's system administrators within a few days

Log In

- Log in via SSH
  - Mac/Linux have built-in client
  - Windows need to download client (e.g. PuTTY)

System	Login Address (xxx.arc.vt.edu)
NewRiver	newriver1 to newriver8
BlueRidge	blueridge1 or blueridge2
HokieSpeed	hokiespeed1 or hokiespeed2
HokieOne	hokieone
Ithaca	ithaca1 or ithaca2

Advanced Research Computing

Browser-based Access

- Browse to http://newriver.arc.vt.edu
- Xterm: Opens an SSH session with X11 forwarding (but faster)
- Other profiles: VisIt, ParaView, Matlab, Allinea
- Create your own!

ALLOCATION SYSTEM

Advanced Research Computing

Allocations

- "System unit" (roughly, core-hour) account that tracks system usage
- Applies only to NewRiver and BlueRidge

http://www.arc.vt.edu/allocations

Advanced Research Computing

Allocation System: Goals

- Track projects that use ARC systems and document how resources are being used
- Ensure that computational resources are allocated appropriately based on needs
  - Research: Provide computational resources for your research lab
  - Instructional: System access for courses or other training events

Advanced Research Computing

Allocation Eligibility

To qualify for an allocation, you must meet at least one of the following:

- Be a Ph.D. level researcher (post-docs qualify)
- Be an employee of Virginia Tech and the PI for research computing
- Be an employee of Virginia Tech and the co-PI for a research project led by non-VT PI

Advanced Research Computing

Allocation Application Process

- Create a research project in ARC database
- Add grants and publications associated with project
- Create an allocation request using the web-based interface
- Allocation review may take several days
- Users may be added to run jobs against your allocation once it has been approved

Advanced Research Computing

Allocation Tiers

- Research allocations fall into three tiers:
- Less than 200,000 system units (SUs)
  - 200 word abstract
- 200,000 to 1 million SUs
  - 1-2 page justification
- More than 1 million SUs
  - 3-5 page justification

Allocation Management

- Web-based:
  - –User Dashboard -> Projects -> Allocations
  - –Systems units allocated/remaining
  - –Add/remove users
- Command line:
  - –Allocation name and membership: `glsaccount`
  - –Allocation size and amount remaining:
    `gbalance -h -a <name>`
  - –Usage (by job): `gstatement -h -a <name>`

USER ENVIRONMENT

Advanced Research Computing

Consistent Environment

- Operating system (CentOS)
- Storage locations
- Scheduler
- Hierarchical module tree for system tools and applications

Modules

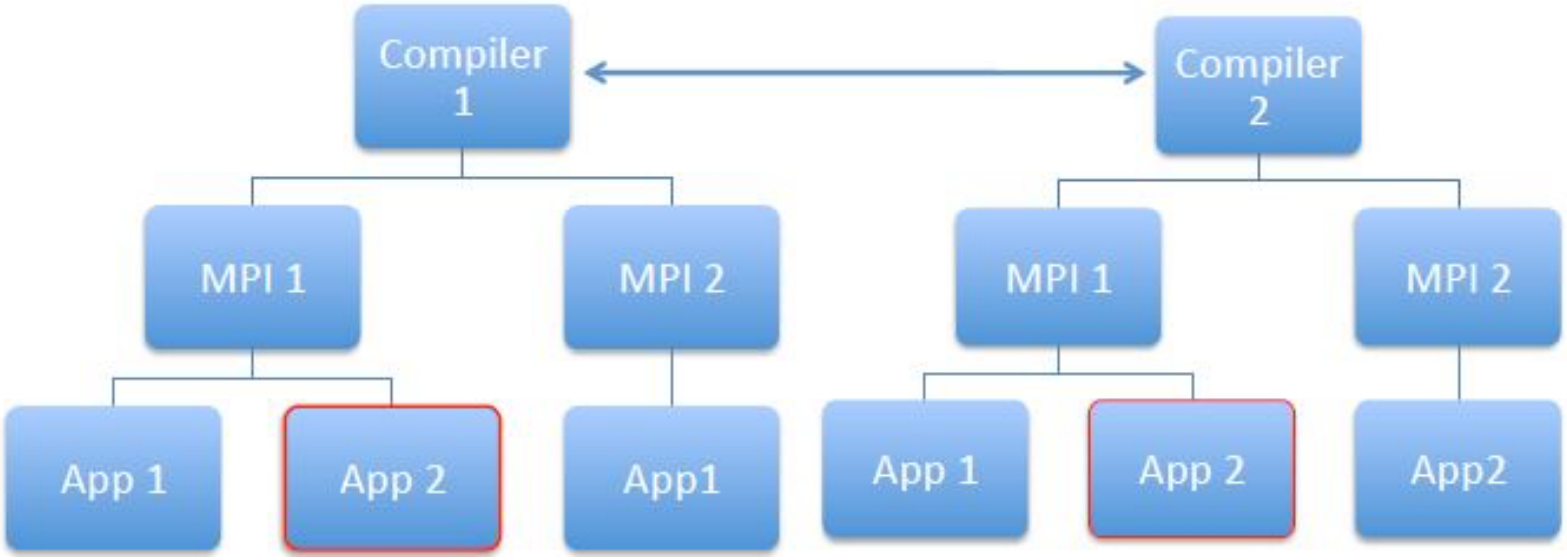
- Modules are used to set the PATH and other environment variables
- Modules provide the environment for building and running applications
  - Multiple compiler vendors (Intel vs GCC) and versions
  - Multiple software stacks: MPI implementations and versions
  - Multiple applications and their versions
- An application is built with a certain compiler and a certain software stack (MPI, CUDA)
  - Modules for software stack, compiler, applications
- User loads the modules associated with an application, compiler, or software stack
  - modules can be loaded in job scripts

Advanced Research Computing

Module commands

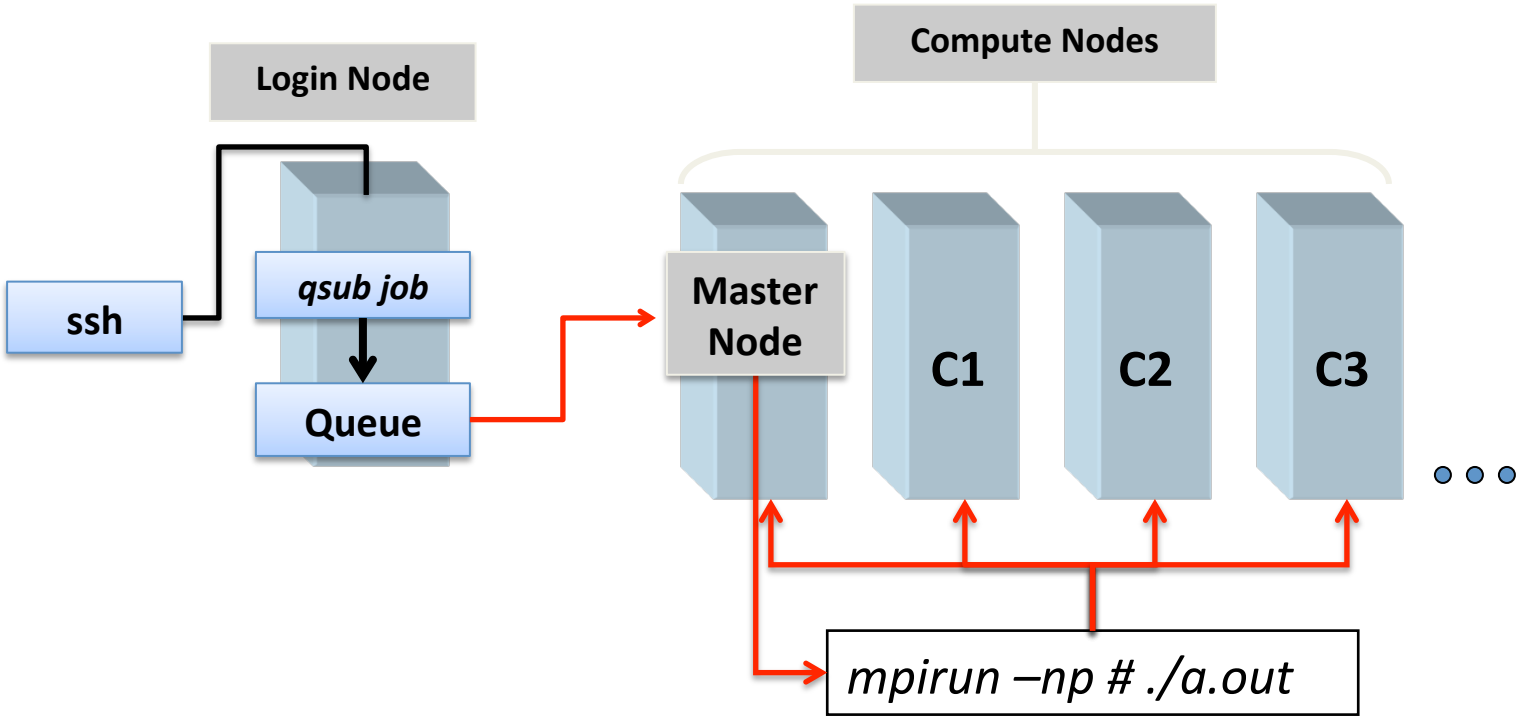
Command	Result
`module`	List options
`module list`	List loaded modules
`module avail`	List available modules
`module load <module>`	Add a module
`module unload <module>`	Remove a module
`module swap <mod1> <mod2>`	Swap two modules
`module help <module>`	Module environment
`module show <module>`	Module description
`module spider <module>`	Search modules
`module reset`	Reset to default
`module purge`	Unload all modules

Hierarchical Module Structure



JOB SUBMISSION & MONITORING

What is Job Submission?



Submitting a Job

- Submission via a shell script
  - Job description: Resources required, run time, allocation
  - Modules & dependencies
  - Execution statements
- Submit job script: `qsub <job_script>`

- Interactive options:
  - Interactive job: `qsub -I …`
  - Interactive job with X11 forwarding: `qsub -I -X …`

Job Monitoring

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

Command	Meaning
`checkjob -v JOBID`	Get the status and resources of a job
`showq`	See what jobs are running and cluster utilization
`showstart JOBID`	Get expected job start time
`qdel JOBID`	Delete a job

Job Execution

- Order of job execution depends on a variety of parameters:
  - Submission Time
  - Queue Priority
  - Backfill Opportunities
  - Fairshare Priority
  - Advanced Reservations
  - Number of Actively Scheduled Jobs per User

Advanced Research Computing

Examples: ARC Website

- See the Examples section of each system page for sample submission scripts and step-by-step examples:
  - http://www.arc.vt.edu/newriver
  - http://www.arc.vt.edu/blueridge
  - http://www.arc.vt.edu/hokiespeed
  - http://www.arc.vt.edu/hokieone
  - http://www.arc.vt.edu/ithaca

A Step-by-Step Example

Getting Started

- Find your training account (hpcXX)
- Log into Ithaca
  - Mac: `ssh hpcXX@ithaca2.arc.vt.edu`
  - Windows: Use PuTTY
    - http://www.chiark.greenend.org.uk/~sgtatham/putty/
    - Host Name: `ithaca2.arc.vt.edu`

Example: Running MPI_Quad

- Source file:

  http://www.arc.vt.edu/wp-content/uploads/2015/04/mpi_quad.c

- Copy the file to Ithaca
  - –wget command
  - –Could also use scp or sftp

- Build the code

Compile the Code

- Intel compiler is already loaded

```
module list
```

- Compile command (executable is mpiqd)

```
mpicc -o mpiqd mpi_quad.c
```

- To use GCC instead, swap it out:

```
module swap intel gcc
```

Prepare Submission Script

- Copy sample script:

```
cp /home/TRAINING/ARC_Intro/it.qsub  .
```

- Edit sample script:
  - Walltime
  - Resource request (nodes/ppn)
  - Module commands (add Intel & mvapich2)
  - Command to run your job
- Save it (e.g., mpiqd.qsub)

Submission Script (Typical)

```
#!/bin/bash
#PBS -l walltime=00:10:00
#PBS -l nodes=2:ppn=8
#PBS -q normal_q
#PBS -W group_list=ithaca
#PBS -A AllocationName          <--Only for NewRiver/BlueRidge

module load intel mvapich2

cd $PBS_O_WORKDIR
echo "MPI Quadrature!"
mpirun -np $PBS_NP ./mpiqd

exit;
```

Submission Script (Today)

```
#!/bin/bash
#PBS -l walltime=00:10:00
#PBS -l nodes=2:ppn=8
#PBS -q normal_q
#PBS -W group_list=training
#PBS -A training
#PBS -l advres=NLI_ARC_Intro.13

module load intel mvapich2

cd $PBS_O_WORKDIR
echo "MPI Quadrature!"
mpirun -np $PBS_NP ./mpiqd

exit;
```

Submit the job

- Copy the files to $WORK:

```
cp mpiqd $WORK
cp mpiqd.qsub $WORK
```

- Navigate to $WORK

```
cd $WORK
```

- Submit the job:

```
qsub mpiqd.qsub
```

- Scheduler returns job number:

```
25770.master.cluster
```

Wait for job to complete

- Check job status:

```
checkjob -v 25770
showq -u hpcXX
```

- When complete:
  - Job output: `mpiqd.qsub.o25770`
  - Errors: `mpiqd.qsub.e25770`
- Copy results back to $HOME:

```
cp mpiqd.qsub.o25770 $HOME
```

Advanced Research Computing

Resources

- ARC Website: http://www.arc.vt.edu
- Compute Resources & Documentation: http://www.arc.vt.edu/hpc
- Storage Documentation: http://www.arc.vt.edu/storage
- New Users Guide: http://www.arc.vt.edu/newusers
- Frequently Asked Questions: http://www.arc.vt.edu/faq
- Linux Introduction: http://www.arc.vt.edu/unix
- Module Tutorial: http://www.arc.vt.edu/modules
- Scheduler Tutorial: http://www.arc.vt.edu/scheduler