Introduction to HPC Using the New Cluster at GACRC

Georgia Advanced Computing Resource Center
University of Georgia
Zhuofei Hou, HPC Trainer
zhuofei@uga.edu

Outline

- What is GACRC?
- What is the new cluster at GACRC?
- How does it operate?
- How to work with it?

What is GACRC?

Who Are We?

- **G**eorgia **A**dvanced **C**omputing **R**esource **C**enter
- Collaboration between the Office of Vice President for Research (**OVPR**) and the Office of the Vice President for Information Technology (**OVPIT**)
- Guided by a faculty advisory committee (GACRC-AC)

Why Are We Here?

- To provide computing hardware and network infrastructure in support of high-performance computing (**HPC**) at UGA

Where Are We?

- http://gacrc.uga.edu (Web) http://wiki.gacrc.uga.edu (Wiki)
- https://wiki.gacrc.uga.edu/wiki/Getting_Help (Support)
- https://blog.gacrc.uga.edu (Blog) http://forums.gacrc.uga.edu (Forums)

GACRC Users September 2015

Colleges & Schools	Depts	PIs	Users
Franklin College of Arts and Sciences	14	117	661
College of Agricultural & Environmental Sciences	9	29	128
College of Engineering	1	12	33
School of Forestry & Natural Resources	1	12	31
College of Veterinary Medicine	4	12	29
College of Public Health	2	8	28
College of Education	2	5	20
Terry College of Business	3	5	10
School of Ecology	1	8	22
School of Public and International Affairs	1	3	3
College of Pharmacy	2	3	5
	40	214	970
Centers & Institutes	9	19	59
TOTALS:	49	233	1029

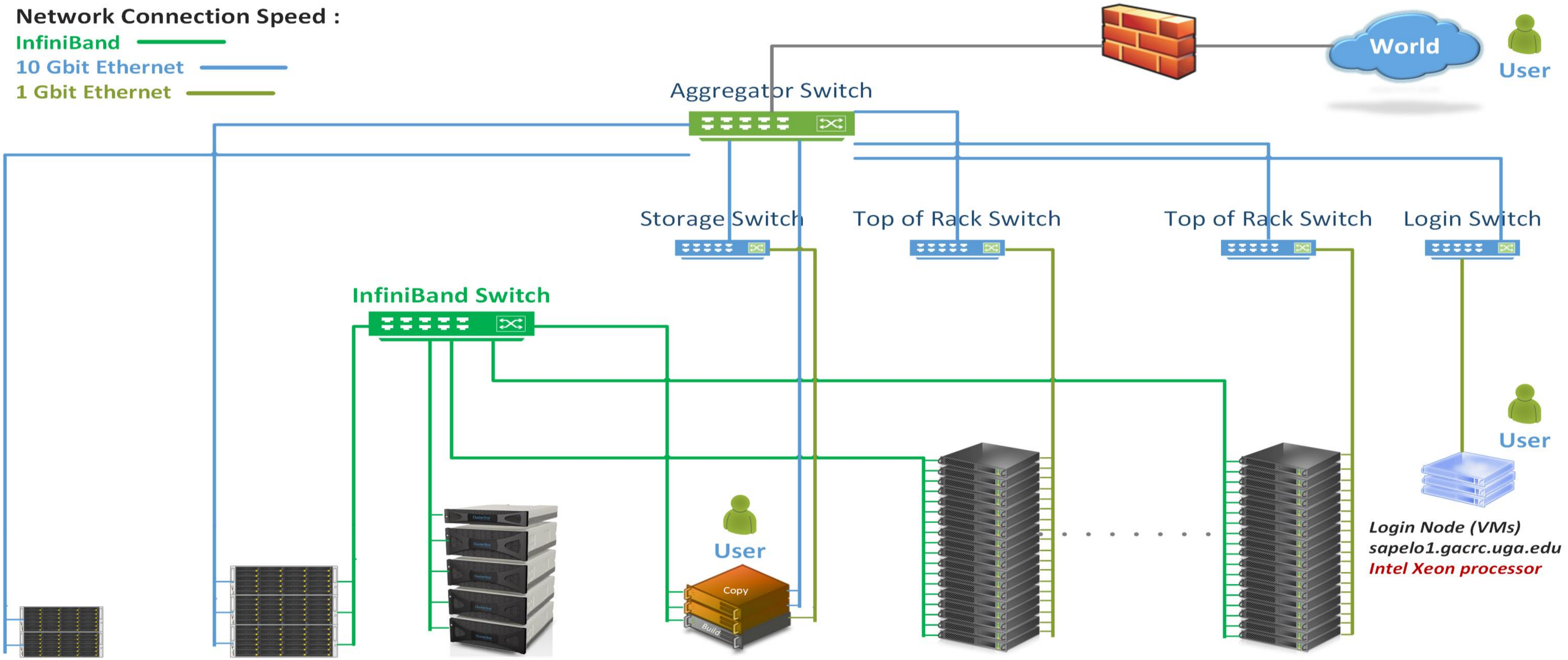
GACRC Users September 2015

Centers & Institutes	PIs	Users
Center for Applied Isotope Study	1	1
Center for Computational Quantum Chemistry	3	10
Complex Carbohydrate Research Center	6	28
Georgia Genomics Facility	1	5
Institute of Bioinformatics	1	1
Savannah River Ecology Laboratory	3	9
Skidaway Institute of Oceanography	2	2
Center for Family Research	1	1
Carl Vinson Institute of Government	1	2
	19	**59**

What is the new cluster at GACRC?

- Cluster Structural Diagram
- General Information
- Computing Resources

The New GACRC Linux HPC Cluster Structural Diagram



Network Connection Speed :

InfiniBand 🟩

10 Gbit Ethernet 🟦

1 Gbit Ethernet 🟩

Icebreaker Storage
Node index: sn3

/project

Icebreaker Storage
Node index: sn[0-2]

/home
/usr/local
/db

Seagate Storage
Lustre Parallel File System

/lustre1

Transfer Node
xfer2.gacrc.uga.edu

Build Node
build1.gacrc.uga.edu
AMD Opteron 16-core processor, 64 GB RAM

Compute Nodes
Node index: n0, n1, n2, …… n165, n166, n167
CPU: AMD Opteron / Intel Xeon multicore processors
GPU: NVIDIA Tesla (Kepler) K40m GPU cards

/lscratch

Login Node (VMs)
sapelo1.gacrc.uga.edu
Intel Xeon processor

What is the new cluster – General Information

The new cluster is a Linux high performance computing (HPC) cluster:

- 64-bit CentOS 6.5 operating system
- User can login to:

  Login node: sapelo1.gacrc.uga.edu (for login & job submission)

  Transfer mode: xfer2.gacrc.uga.edu (for data transferring & compression)

  Build node: build1.gacrc.uga.edu (for code compilation)

- **InfiniBand network** provides internodal communication:

  compute nodes ⇔ compute nodes

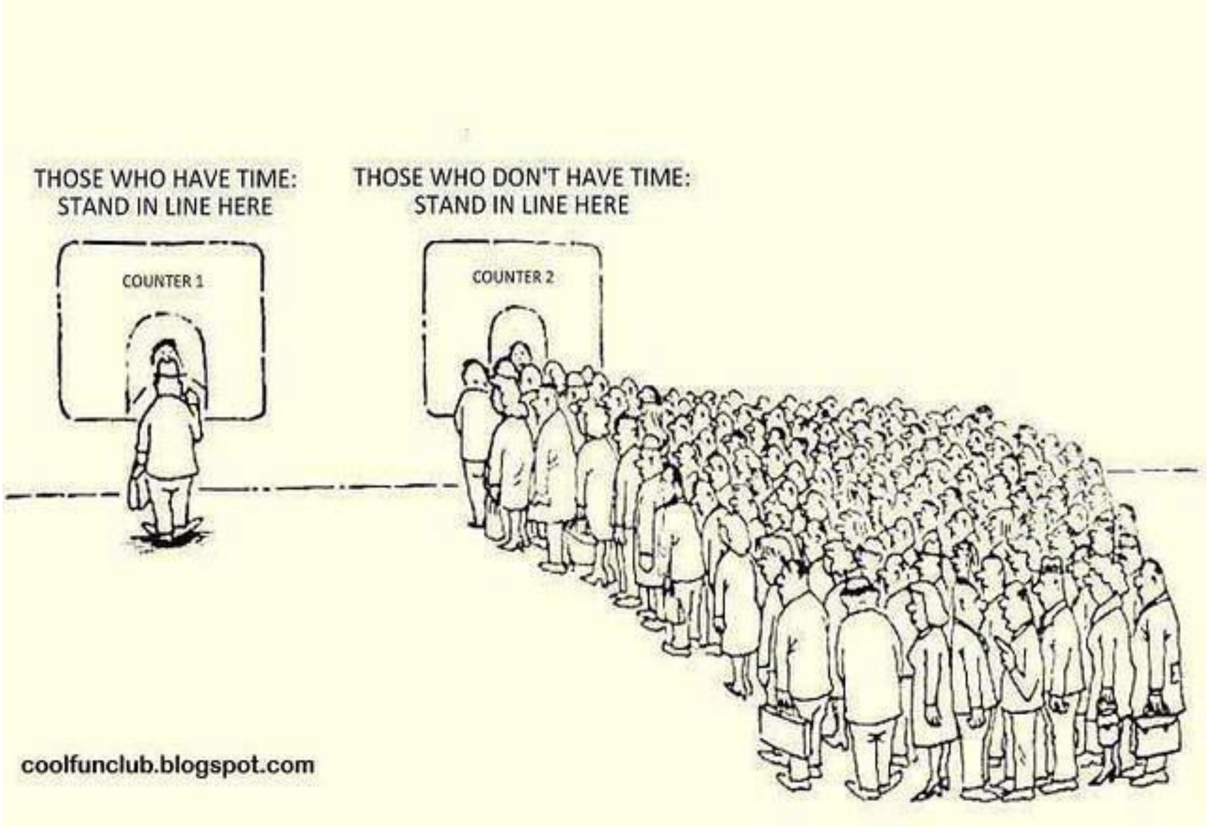
  compute nodes ⇔ storage systems, e.g., /home and /scratch

What is the new cluster – General Information

- Batch-queueing System:
  - ➢ Jobs can be started (submitted), monitored, and controlled
  - ➢ Determine which compute node is the best place to run a job
  - ➢ Determine appropriate execution priority for a job to run

- On new cluster:

  Torque Resource Manager
  Moab Workload Manager



What is the new cluster – Computing Resources

Queue	Node Type	Total Nodes	Processor	Cores / Node	RAM (GB) / Node	GPU	GPU Cards / Node	InfiniBand
batch	AMD	120	AMD Opteron	48	128	N/A	N/A	Yes
	HIGHMEM	3	AMD Opteron	48	512 (2) 1024 (1)	N/A	N/A	Yes
	GPU	2	Intel Xeon	16	128	NVIDIA K40m	8	Yes

Peak Performance per Node: 500 Gflops/Node

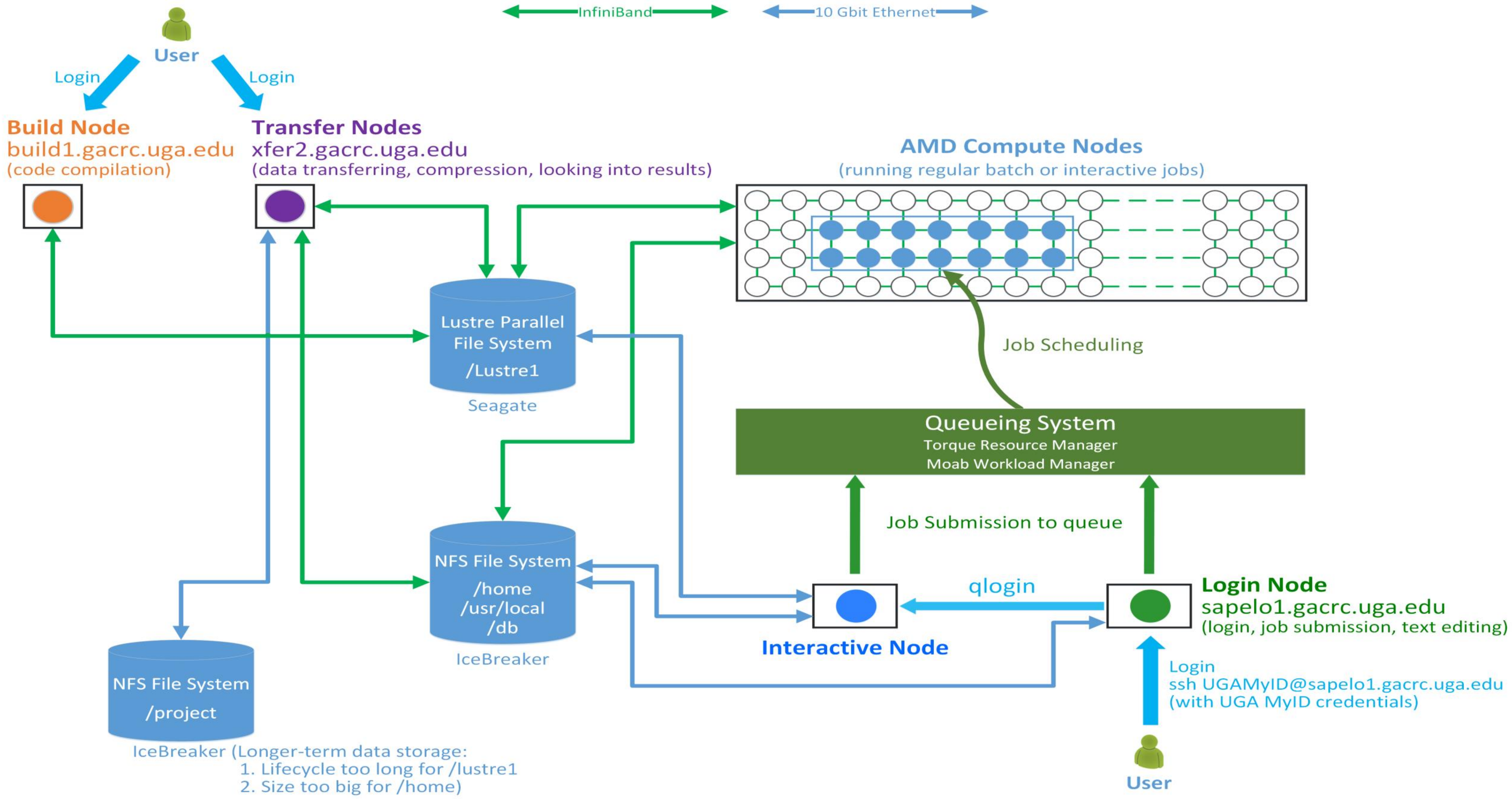
Home directory : 100 GB

Scratch directory on /lustre1 : NO quota limit, auto-moved to /project,
if no modification in 30 days!

How does it operate?

Next Page

The New GACRC Linux HPC Cluster Operational Diagram



InfiniBand

10 Gbit Ethernet

User

Login

Login

Build Node
build1.gacrc.uga.edu
(code compilation)

Transfer Nodes
xfer2.gacrc.uga.edu
(data transferring, compression, looking into results)

AMD Compute Nodes
(running regular batch or interactive jobs)

Lustre Parallel File System
/Lustre1
Seagate

Job Scheduling

Queueing System
Torque Resource Manager
Moab Workload Manager

Job Submission to queue

NFS File System
/home
/usr/local
/db
IceBreaker

qlogin

Interactive Node

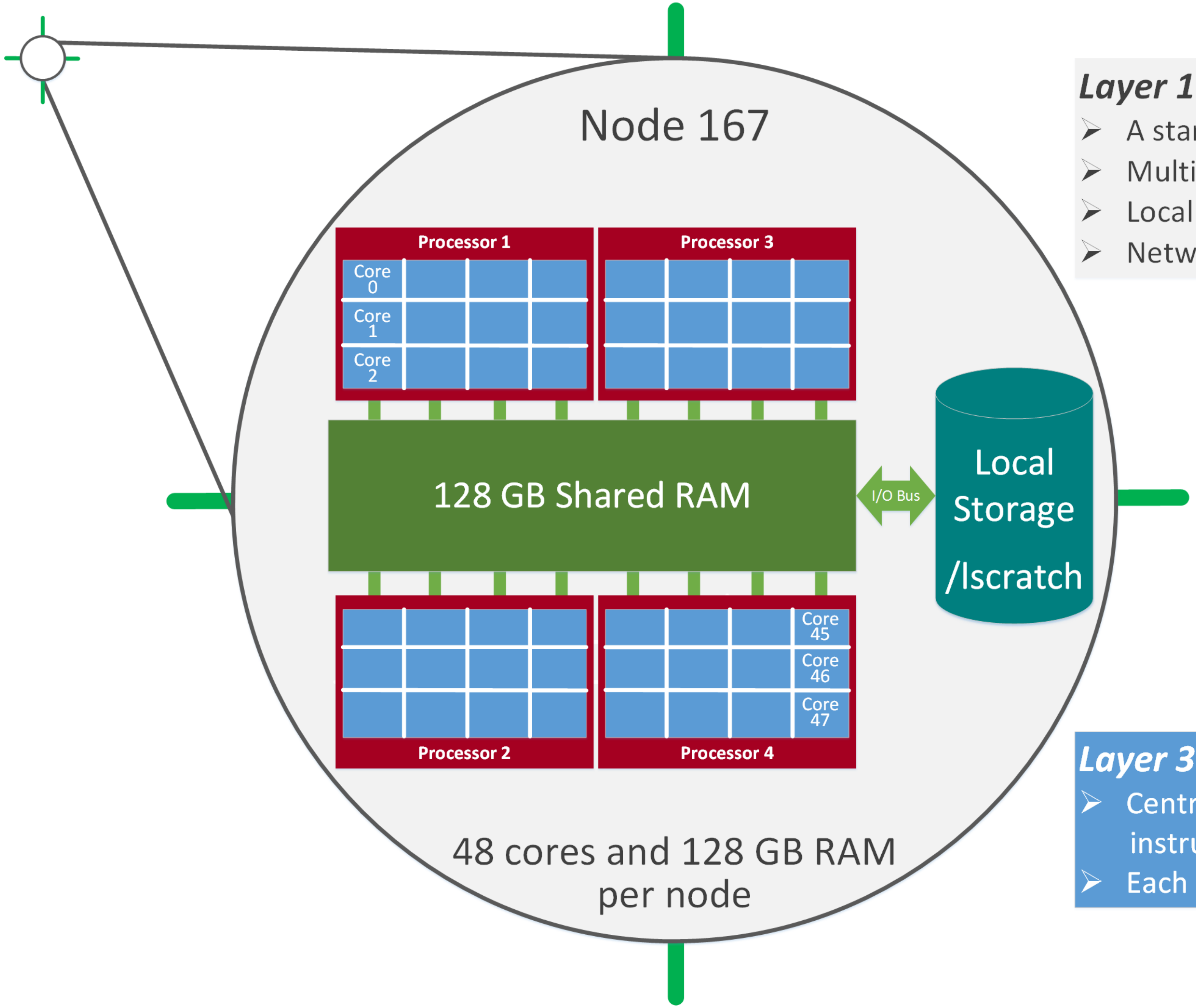
Login Node
sapelo1.gacrc.uga.edu
(login, job submission, text editing)

Login
ssh UGAMyID@sapelo1.gacrc.uga.edu
(with UGA MyID credentials)

User

NFS File System
/project

IceBreaker (Longer-term data storage:
1. Lifecycle too long for /lustre1
2. Size too big for /home)



Layer 1: Node

- A standalone "computer in a box"
- Multiple processors, e.g. 4, sharing memory
- Local disk storage, network interface, etc.
- Networked into a cluster

Layer 2: Processor

- A single computing component
- Multicore processor, e.g. 12 cores

Layer 3: Core

- Central processing unit (CPU) reading and executing instructions independently
- Each core is assigned to a software thread

How to work with it?

Before we start:

- To get the new cluster to be your best HPC buddy, go to

  GACRC Wiki (http://wiki.gacrc.uga.edu)

  GACRC Web (http://gacrc.uga.edu)

- To get the most effective and qualified support from us, go to

  GACRC Support (https://wiki.gacrc.uga.edu/wiki/Getting_Help)

- To work happily and productively, follow the new cluster's Community Code of Conduct (CCOC)

How to work with it?

- Cluster’s CCOC:

On cluster, you are not alone……. Each user is sharing finite resources, e.g., CPU cycles, RAM, disk storage, network bandwidth, with other researchers. *What you do may affect other researchers on the cluster*.

6 rules of thumb to remember:

- NO jobs running on login node
- NO multi-threaded job running with only 1 core requested
- NO large memory job running on regular nodes
- NO long job running on interactive node
- NO small memory job running on large memory nodes
- Use the copy node for file transfer and compression

How to work with it?

- Start with the Cluster
- Connect & Login
- Software Packages
- Run Jobs
  - How to submit a job
  - How to make a job submission script
  - How to check job status, cancel a job, etc.

How to work with it – Start with the Cluster

- You need a User Account: UGAMyID@sapelo1.gacrc.uga.edu

  To create your account correctly, you must provide us with your official UGA MyID, not a UGA MyID alias! ⚠

- To get a user account, follow 4 steps:

1. New user training (http://gacrc.uga.edu/help/training/)
2. Tell us your Name, UGA MyID, Lab name and PI's name, via GACRC Support (https://wiki.gacrc.uga.edu/wiki/Getting_Help)
3. We send you an invitation letter with instructions to start account initialization
4. With Step 3 finished successfully, we send you a welcome letter with whole package of information about your account created successfully

How to work with it – Connect & Login

- Open a connection: Open a terminal and ssh to your account

```
ssh zhuofei@sapelo1.gacrc.uga.edu
```

or

```
ssh -X zhuofei@sapelo1.gacrc.uga.edu
```

([1] −X is for X windows application running on the cluster to be forwarded to your local machine

[2] If using Windows, use SSH client to open connection, get from UGA download software page)

- Logging in: You will be prompted for your UGA MyID password

```
zhuofei@sapelo1.gacrc.uga.edu’s password: █
```

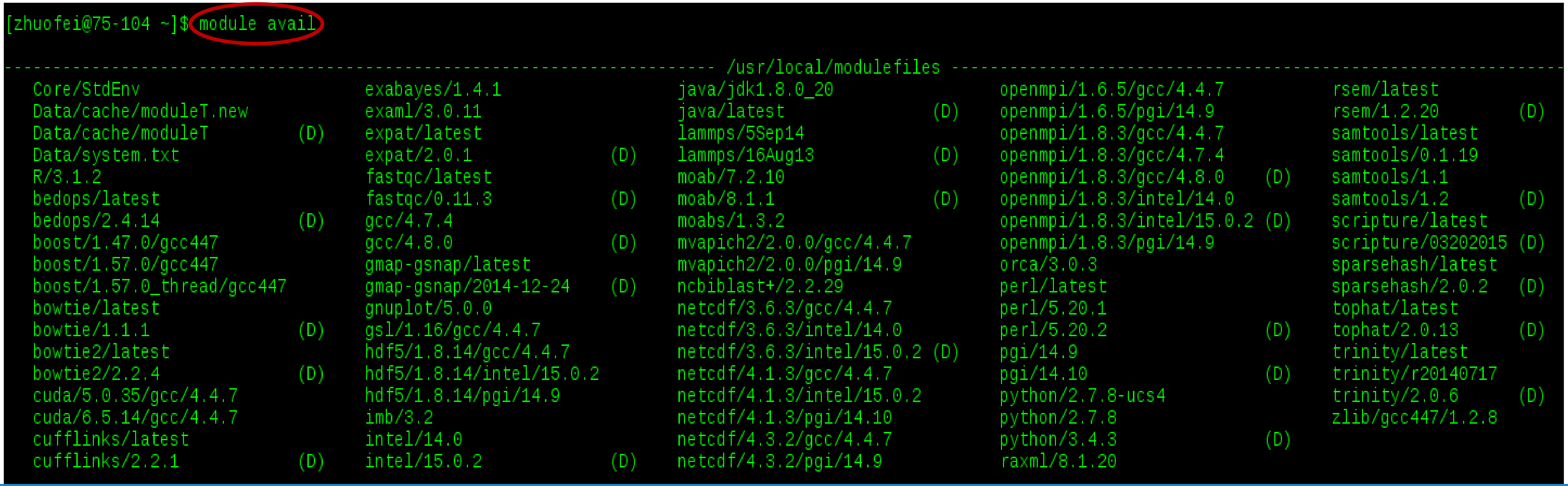
([3] On Linux/Mac, when you type in the password, the prompt blinks and does not move)

- Logging out: exit to leave the system

```
[zhuofei@75-104 ~]$ exit
```

How to work with it – Software Packages

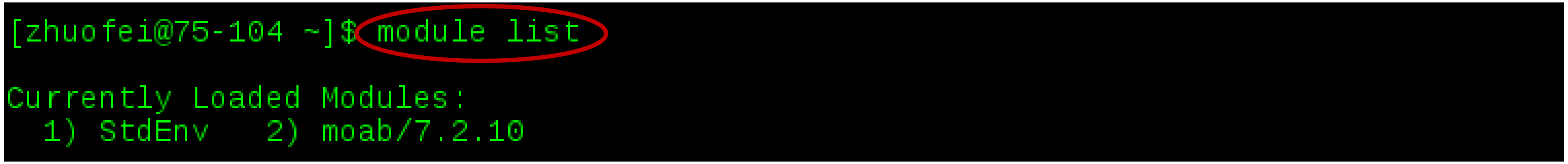
- The cluster uses environment modules to define the various paths for software packages
- Current number of modules installed is ~70 and expanding daily!
- `module avail` to list all modules available on the cluster:



```
[zhuofei@75-104 ~]$ module avail

------------------------------------------------------------------------ /usr/local/modulefiles ---------------------------------------------------------------
   Core/StdEnv                       exabayes/1.4.1                  java/jdk1.8.0_20                 openmpi/1.6.5/gcc/4.4.7           rsem/latest
   Data/cache/moduleT.new            examl/3.0.11                    java/latest               (D)    openmpi/1.6.5/pgi/14.9            rsem/1.2.20        (D)
   Data/cache/moduleT         (D)    expat/latest                    lammps/5Sep14                    openmpi/1.8.3/gcc/4.4.7           samtools/latest
   Data/system.txt                   expat/2.0.1              (D)    lammps/16Aug13            (D)    openmpi/1.8.3/gcc/4.7.4           samtools/0.1.19
   R/3.1.2                           fastqc/latest                   moab/7.2.10                      openmpi/1.8.3/gcc/4.8.0    (D)    samtools/1.1
   bedops/latest                     fastqc/0.11.3            (D)    moab/8.1.1                (D)    openmpi/1.8.3/intel/14.0          samtools/1.2       (D)
   bedops/2.4.14              (D)    gcc/4.7.4                       moabs/1.3.2                      openmpi/1.8.3/intel/15.0.2 (D)    scripture/latest
   boost/1.47.0/gcc447               gcc/4.8.0                (D)    mvapich2/2.0.0/gcc/4.4.7         openmpi/1.8.3/pgi/14.9            scripture/03202015 (D)
   boost/1.57.0/gcc447               gmap-gsnap/latest               mvapich2/2.0.0/pgi/14.9          orca/3.0.3                        sparsehash/latest
   boost/1.57.0_thread/gcc447        gmap-gsnap/2014-12-24    (D)    ncbiblast+/2.2.29                perl/latest                       sparsehash/2.0.2   (D)
   bowtie/latest                     gnuplot/5.0.0                   netcdf/3.6.3/gcc/4.4.7           perl/5.20.1                       tophat/latest
   bowtie/1.1.1               (D)    gsl/1.16/gcc/4.4.7              netcdf/3.6.3/intel/14.0          perl/5.20.2                (D)    tophat/2.0.13      (D)
   bowtie2/latest                    hdf5/1.8.14/gcc/4.4.7           netcdf/3.6.3/intel/15.0.2 (D)    pgi/14.9                          trinity/latest
   bowtie2/2.2.4              (D)    hdf5/1.8.14/intel/15.0.2        netcdf/4.1.3/gcc/4.4.7           pgi/14.10                  (D)    trinity/r20140717
   cuda/5.0.35/gcc/4.4.7             hdf5/1.8.14/pgi/14.9            netcdf/4.1.3/intel/15.0.2        python/2.7.8-ucs4                 trinity/2.0.6      (D)
   cuda/6.5.14/gcc/4.4.7             imb/3.2                         netcdf/4.1.3/pgi/14.10           python/2.7.8                      zlib/gcc447/1.2.8
   cufflinks/latest                  intel/14.0                      netcdf/4.3.2/gcc/4.4.7           python/3.4.3               (D)
   cufflinks/2.2.1            (D)    intel/15.0.2             (D)    netcdf/4.3.2/pgi/14.9            raxml/8.1.20
```

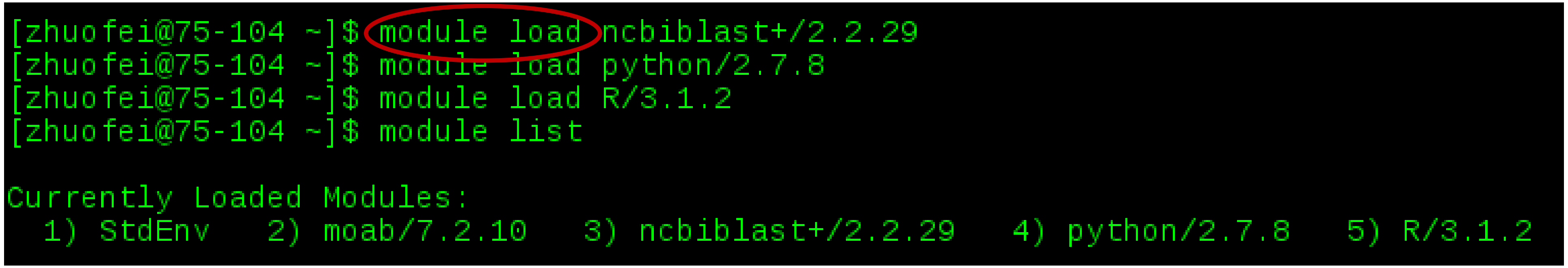
How to work with it – Software Packages

- `module list` to list which modules currently loaded:



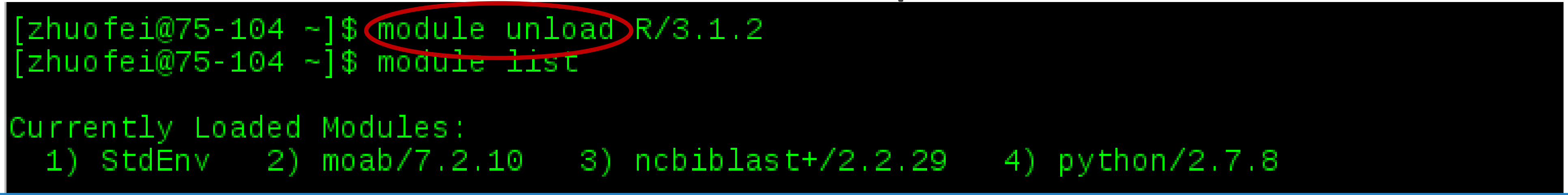
```
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10
```

- `module load` to load the needed modules:



```
[zhuofei@75-104 ~]$ module load ncbiblast+/2.2.29
[zhuofei@75-104 ~]$ module load python/2.7.8
[zhuofei@75-104 ~]$ module load R/3.1.2
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10   3) ncbiblast+/2.2.29   4) python/2.7.8   5) R/3.1.2
```

- `module unload` to remove the specific module:



```
[zhuofei@75-104 ~]$ module unload R/3.1.2
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10   3) ncbiblast+/2.2.29   4) python/2.7.8
```

How to work with it – Run Jobs

- To run a batch job, you need:
  - ➢ binary compiled on build1 or interactive node (AMD Opteron)
  - ➢ Job submission script to run the binary, specifying computing resources:
    - ✓ Number of nodes and cores
    - ✓ Amount of memory
    - ✓ Type of nodes
    - ✓ Maximum wallclock time, etc.

- Common commands:
  - ➢ `qsub, qstat, qdel`
  - ➢ `showq, checkjob,` etc.

How to work with it – Run Jobs

- How to compile binary code on build1 or interactive node
  - Login to username@build1.gacrc.uga.edu or
    interactive node (using `qlogin` from Login node)
  - Load compilers using `module load`, e.g.:

```
module load pgi/14.9          ➔ PGI 14.9 compilers
module load intel/15.0.2      ➔ Intel 15.0.2 compilers
module load mvapich2/2.0.0/gcc/4.4.7 ➔ MVAPICH2 2.0.0 using GNU 4.4.7 compilers
module load openmpi/1.8.3/pgi/14.9   ➔ OpenMPI 1.8.3, using PGI 14.9 compilers
```

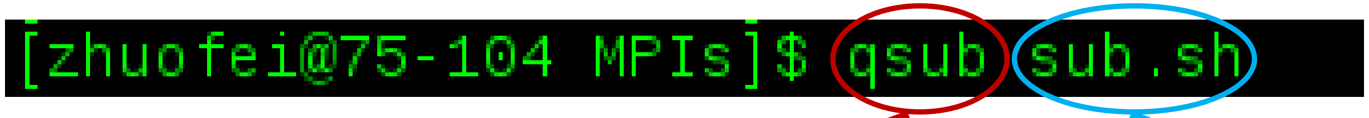
  - Compile source codes into binary

Note: 1. Default GNU 4.4.7 compilers; 2. No MPICH2, which uses Ethernet only

- Refer to https://wiki.gacrc.uga.edu/wiki/Code_Compilation_on_Sapelo
  https://wiki.gacrc.uga.edu/wiki/MPI

How to work with it – Run Jobs

- How to submit a job? ***Easy!***



```
[zhuofei@75-104 MPIs]$ qsub sub.sh
```

qsub is to submit a job

sub.sh is your **job submission script** specifying:

- ✓ Number of nodes and cores
- ✓ Amount of memory
- ✓ Type of nodes
- ✓ Maximum wallclock time, etc.

- How to make a job submission script? ***Next Page!***

How to work with it – Run Jobs

- Example 1: Serial job script *sub.sh* running NCBI Blast +

`#PBS -S /bin/bash`	→ Linux shell (bash)
`#PBS -q batch`	→ Queue name (batch)
`#PBS -N testBlast`	→ Name of the job (testBlast)
`#PBS -l nodes=1:ppn=1:AMD`	→ Number of nodes (1), number of cores/node (1), node type (AMD)
`#PBS -l mem=20g    b`	→ Maximum amount of physical memory (20 GB) used by the job
`#PBS -l walltime=48:00:00`	→ Maximum wall clock time (48 hours) for the job, default 6 minutes
`cd $PBS_O_WORKDIR`	→ Use the directory from which the job is submitted as the working directory
`module load ncbiblast+/2.2.29`	→ Load the module of ncbiblast+, version 2.2.29
`time blastn [options] > outputfile`	→ Run blastn with ‘time’ command to measure the amount of time it takes to run the application

How to work with it – Run Jobs

- Example 2: Threaded job script *sub.sh* running NCBI Blast + with 4 threads

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=4:AMD            → Number of nodes (1), number of cores/node (4), node type (AMD)
#PBS -l walltime=480:00:00              Number of threads (4) = Number of cores requested (4)
#PBS -l mem=20gb

#PBS -M jSmith@uga.edu               → Email to receive a summary of computing resources used by the job
#PBS -m ae                           → Receive an email when the job finishes (e)
#PBS -j oe                           → Standard error file (testBlast.e1234) will be merged into standard
                                        out file (testBlast.o1234)

cd $PBS_O_WORKDIR

module load ncbiblast+/2.2.29

time blastn -num_threads 4 [options]  > outputfile     → Run blastn with 4 threads (-num_threads 4)
```

How to work with it – Run Jobs

- Example 3: MPI job script *sub.sh* running RAxML with 50 MPI processes

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=48:AMD        ➔ Number of nodes (2), number of cores/node (48), node type (AMD)
#PBS -l walltime=48:00:00           Total cores requested = 2 × 48 = 96
#PBS -l mem=20gb                    We suggest, Number of MPI Processes (50) ≤ Number of cores requested (96)
#PBS -j oe

cd $PBS_O_WORKDIR

module load raxml/8.1.20          ➔ To run raxmlHPC-MPI-AVX, MPI version using OpenMPI 1.8.3/Intel 15.0.2
module load intel/15.0.2
module load openmpi/1.8.3/intel/15.0.2
              ⬇
mpirun –np 50 raxmlHPC-MPI-AVX [options] > outputfile ➔ Run raxmlHPC-MPI-AVX with 50 MPI processes (–np 50)
```

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=27:AMD        → ppn number (27) fewer than 48 MUST be a multiplier of 3!
#PBS -l walltime=48:00:00
#PBS -l mem=20gb
#PBS -j oe

cd $PBS_O_WORKDIR

# Context Sharing
CONTEXTS=$(/usr/local/bin/set_contexts.sh $PBS_NUM_PPN)
if [[ "$?" -eq "0" ]] ; then                                  } New lines added!
  export PSM_SHAREDCONTEXTS_MAX=$CONTEXTS
fi

module load raxml/8.1.20
module load intel/15.0.2
module load openmpi/1.8.3/intel/15.0.2

mpirun –np 50 raxmlHPC-MPI-AVX [options] > outputfile → Run raxmlHPC-MPI-AVX with 50 MPI processes (–np 50)
```

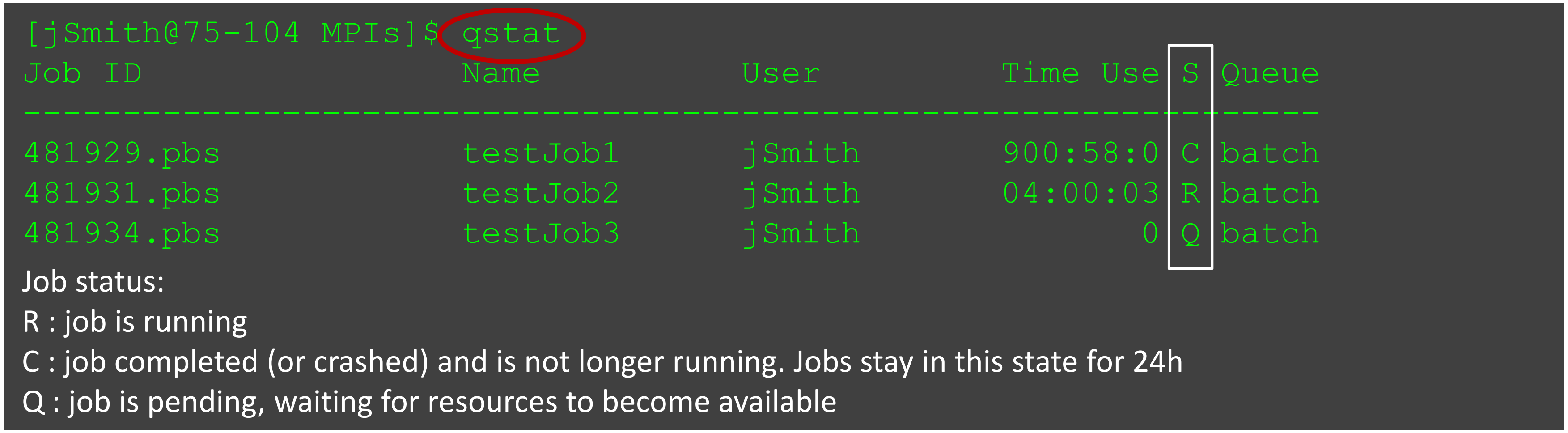
How to work with it – Run Jobs

- Example 4: Job script *sub.sh* running a GPU-enabled (e.g. CUDA) application:

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=1:ppn=4:gpus=1:GPU          ➔  Number of GPU card (1), node type (GPU)
#PBS -l walltime=48:00:00
#PBS -l mem=20gb
#PBS -j oe

cd $PBS_O_WORKDIR

module load cuda/6.5/gcc/4.4.7            ➔ Load CUDA 6.5 module

Time ./a.out > outputfile                 ➔ Run binary (a.out) with 4 threads and 1 GPU card
```

How to work with it – Run Jobs

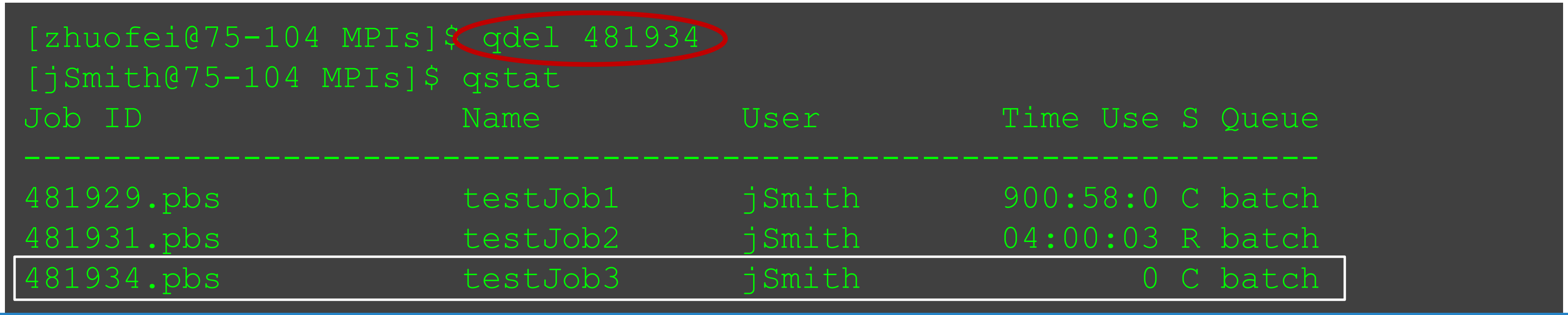
- How to check job status? ***qstat!***



```
[jSmith@75-104 MPIs]$ qstat
Job ID                    Name             User            Time Use S Queue
-----------------------------------------------------------------------------
481929.pbs                testJob1         jSmith          900:58:0 C batch
481931.pbs                testJob2         jSmith          04:00:03 R batch
481934.pbs                testJob3         jSmith                 0 Q batch
```

Job status:
R : job is running
C : job completed (or crashed) and is not longer running. Jobs stay in this state for 24h
Q : job is pending, waiting for resources to become available

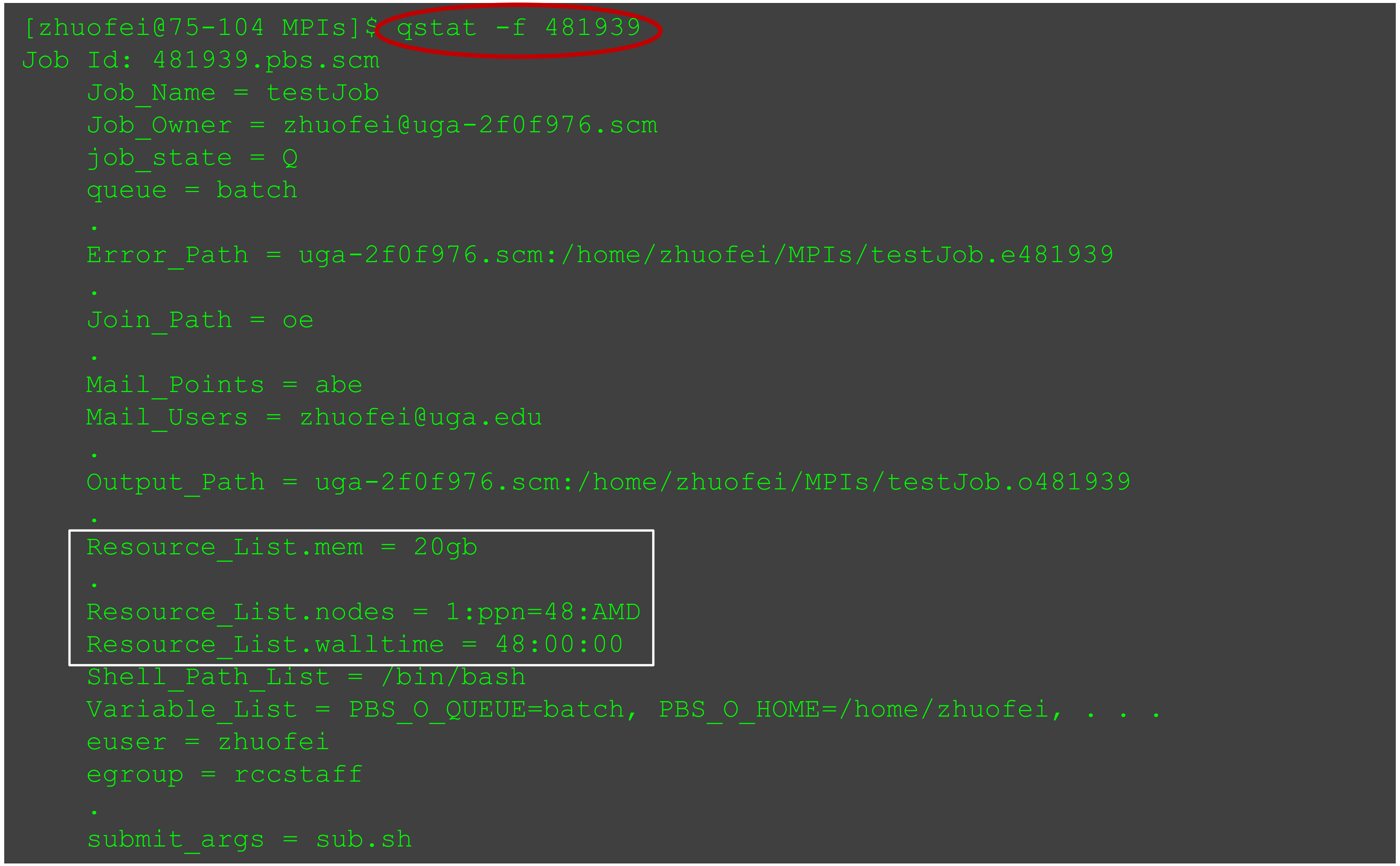
- How to cancel *testJob3* with jobID *481934*? ***qdel!***



```
[zhuofei@75-104 MPIs]$ qdel 481934
[jSmith@75-104 MPIs]$ qstat
Job ID                    Name             User            Time Use S Queue
-----------------------------------------------------------------------------
481929.pbs                testJob1         jSmith          900:58:0 C batch
481931.pbs                testJob2         jSmith          04:00:03 R batch
481934.pbs                testJob3         jSmith                 0 C batch
```

How to work with it – Run Jobs

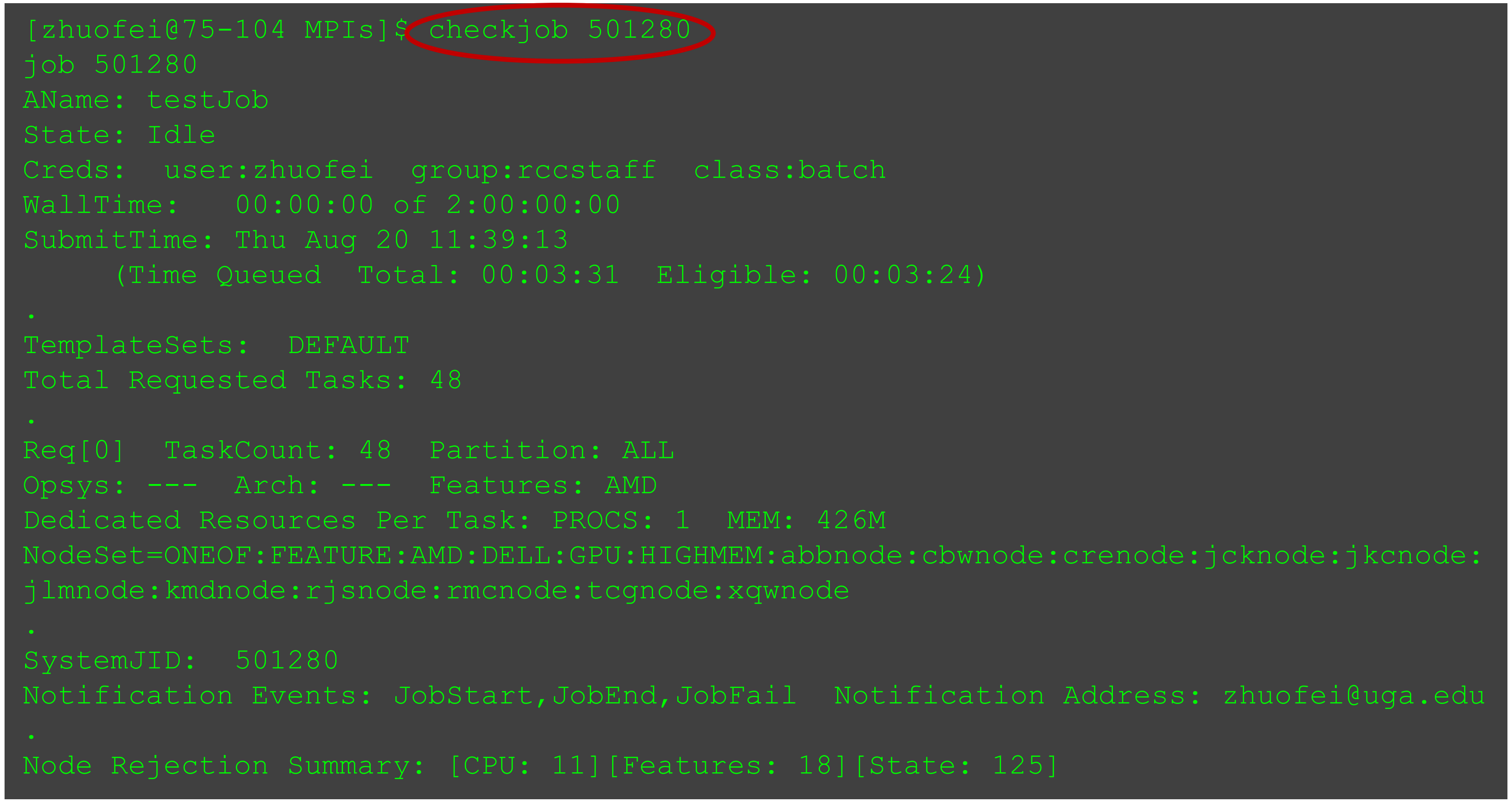
- How to check resource utilization of a job? ***qstat -f***



```
[zhuofei@75-104 MPIs]$ qstat -f 481939
Job Id: 481939.pbs.scm
    Job_Name = testJob
    Job_Owner = zhuofei@uga-2f0f976.scm
    job_state = Q
    queue = batch
    .
    Error_Path = uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.e481939
    .
    Join_Path = oe
    .
    Mail_Points = abe
    Mail_Users = zhuofei@uga.edu
    .
    Output_Path = uga-2f0f976.scm:/home/zhuofei/MPIs/testJob.o481939
    .
    Resource_List.mem = 20gb
    .
    Resource_List.nodes = 1:ppn=48:AMD
    Resource_List.walltime = 48:00:00
    Shell_Path_List = /bin/bash
    Variable_List = PBS_O_QUEUE=batch, PBS_O_HOME=/home/zhuofei, . . .
    euser = zhuofei
    egroup = rccstaff
    .
    submit_args = sub.sh
```

How to work with it – Run Jobs

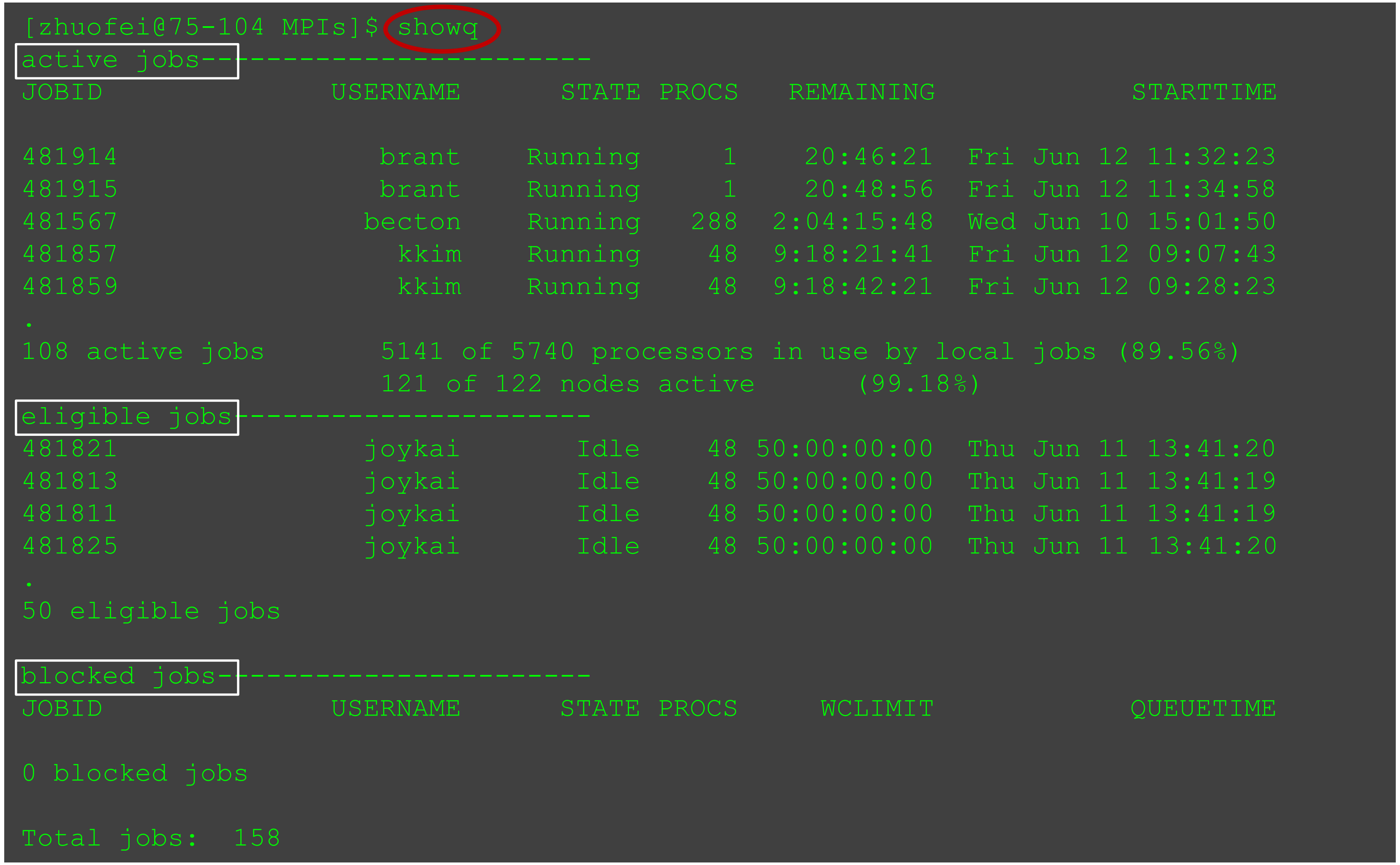
- How to check resource utilization of a job? ***checkjob***



```
[zhuofei@75-104 MPIs]$ checkjob 501280
job 501280
AName: testJob
State: Idle
Creds:  user:zhuofei  group:rccstaff  class:batch
WallTime:   00:00:00 of 2:00:00:00
SubmitTime: Thu Aug 20 11:39:13
     (Time Queued  Total: 00:03:31  Eligible: 00:03:24)
.
TemplateSets:  DEFAULT
Total Requested Tasks: 48
.
Req[0]  TaskCount: 48  Partition: ALL
Opsys: ---  Arch: ---  Features: AMD
Dedicated Resources Per Task: PROCS: 1  MEM: 426M
NodeSet=ONEOF:FEATURE:AMD:DELL:GPU:HIGHMEM:abbnode:cbwnode:crenode:jcknode:jkcnode:
jlmnode:kmdnode:rjsnode:rmcnode:tcgnode:xqwnode
.
SystemJID:  501280
Notification Events: JobStart,JobEnd,JobFail  Notification Address: zhuofei@uga.edu
.
Node Rejection Summary: [CPU: 11][Features: 18][State: 125]
```

How to work with it – Run Jobs

- How to check queue status? ***showq!***



```
[zhuofei@75-104 MPIs]$ showq
active jobs------------------------
JOBID               USERNAME      STATE PROCS   REMAINING            STARTTIME

481914                 brant    Running     1    20:46:21  Fri Jun 12 11:32:23
481915                 brant    Running     1    20:48:56  Fri Jun 12 11:34:58
481567                becton    Running   288  2:04:15:48  Wed Jun 10 15:01:50
481857                  kkim    Running    48  9:18:21:41  Fri Jun 12 09:07:43
481859                  kkim    Running    48  9:18:42:21  Fri Jun 12 09:28:23
.
108 active jobs       5141 of 5740 processors in use by local jobs (89.56%)
                      121 of 122 nodes active      (99.18%)
eligible jobs----------------------
481821                joykai       Idle    48 50:00:00:00  Thu Jun 11 13:41:20
481813                joykai       Idle    48 50:00:00:00  Thu Jun 11 13:41:19
481811                joykai       Idle    48 50:00:00:00  Thu Jun 11 13:41:19
481825                joykai       Idle    48 50:00:00:00  Thu Jun 11 13:41:20
.
50 eligible jobs

blocked jobs-----------------------
JOBID               USERNAME      STATE PROCS     WCLIMIT            QUEUETIME

0 blocked jobs

Total jobs:  158
```

Thank You!