Introduction to High Performance Computing (HPC) Resources at GACRC

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

Outline

- What is GACRC?
- Concept of High Performance Computing (HPC)
- What is GACRC zcluster?
- What is GACRC New Cluster – Sapelo?

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)
- http://gacrc.uga.edu/help/ (Web Help)
  https://wiki.gacrc.uga.edu/wiki/Getting_Help (Wiki Help)

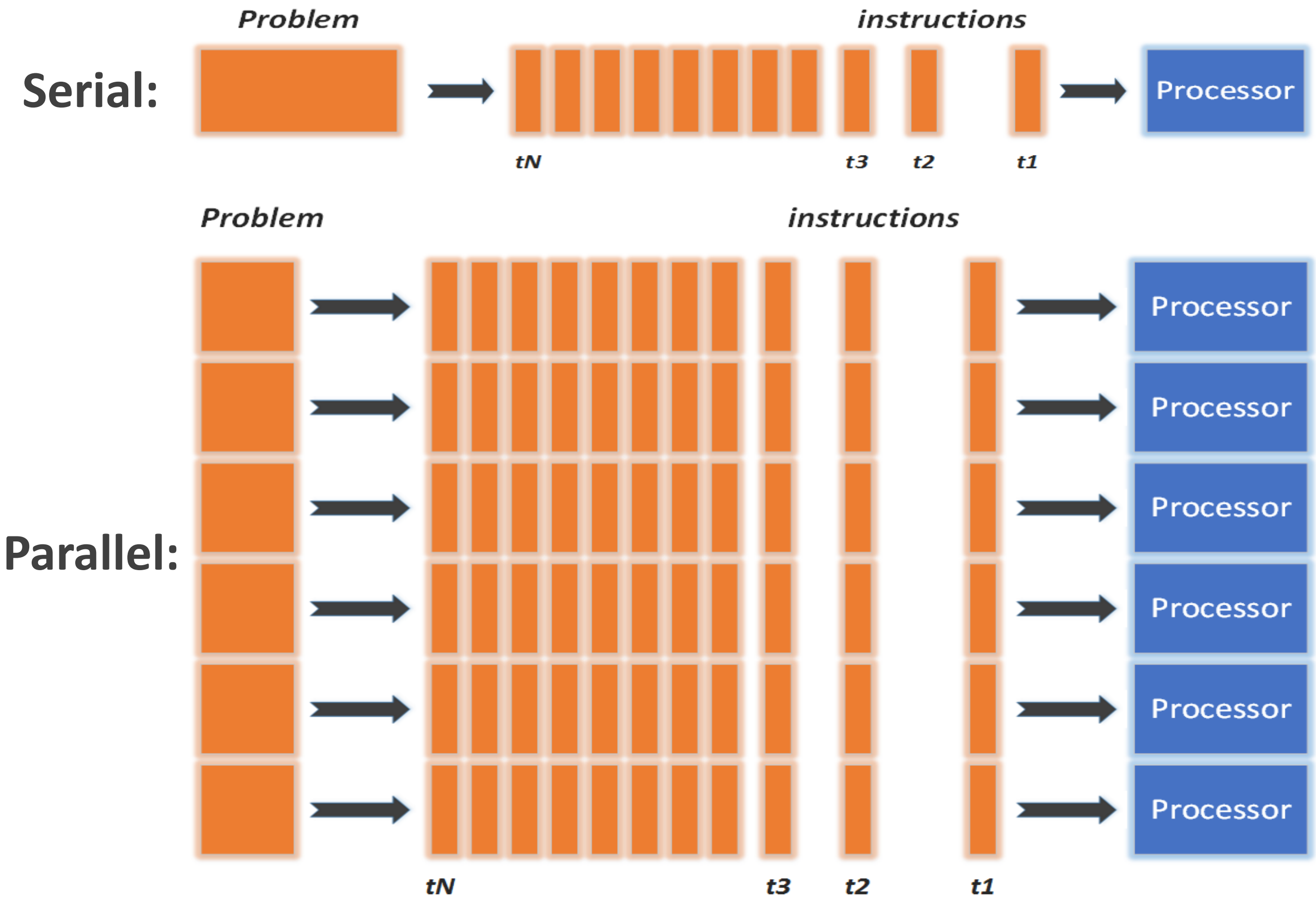
GACRC Users September 2015

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**

Concept of High Performance Computing (HPC)



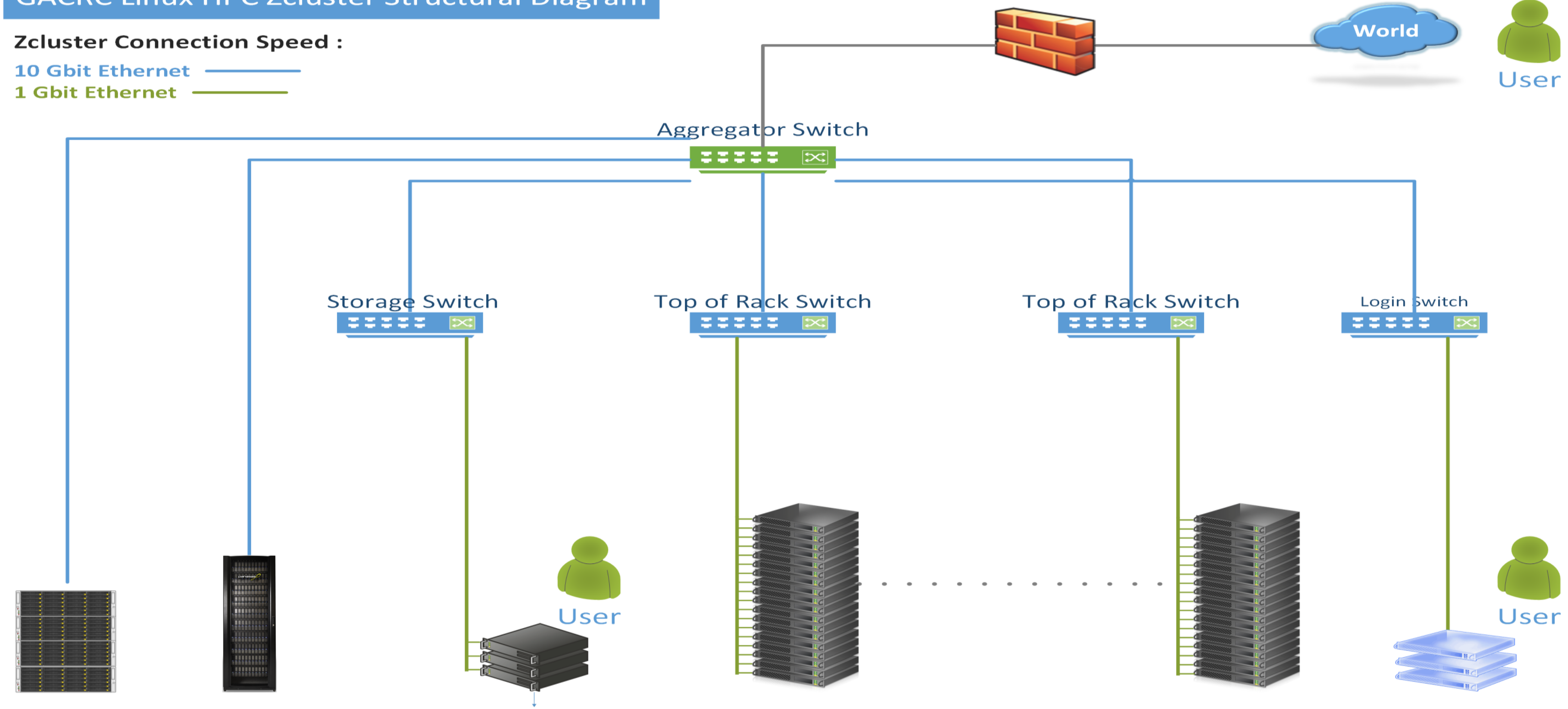
- ✓ Serial problem can not be broken
- ✓ *Discrete* instructions executed *sequentially*
- ✓ Only *1* instruction executed at any moment on a *single* processor

- ✓ Problem broken into *parallel* parts can be solved *concurrently*
- ✓ Instructions executed *simultaneously* on *multiply* processors
- ✓ Synchronization/communication employed
- ✓ Shared-memory multithreaded job or MPI job (Message Passing Interface)

What is GACRC zcluster?

- Cluster Structural Diagram
- General Information
- Computing Resources
- Software Installed
- Submit Jobs

GACRC Linux HPC Zcluster Structural Diagram



Zcluster Connection Speed :

10 Gbit Ethernet

1 Gbit Ethernet

World

User

Aggregator Switch

Storage Switch

Top of Rack Switch

Top of Rack Switch

Login Switch

User

User

Icebreaker Storage

/escratch4

Panasas Storage

/home
/usr/local
/db

Copy Nodes
copy.rcc.uga.edu

Node index: 1, 2, 3

Compute Nodes on Rack 1-11

Node index: c1-1, c1-2, ……, c11-1, c11-2, ……
CPU: Intel Xeon/AMD Opteron multicore
GPU: NVIDIA Tesla (Fermi/Kepler)

/lscratch

Login Node
zcluster.rcc.uga.edu

zcluster General Information

zcluster is a Linux high performance computing (HPC) cluster:

- Operating System: 64-bit Red Hat Enterprise Linux 5 (RHEL 5)
- User can login to:

  Login node: zcluster.rcc.uga.edu (for login & job submission)
  Copy node: copy.rcc.uga.edu (for data transferring & compression)

- Internodal Communication: 1Gbit network

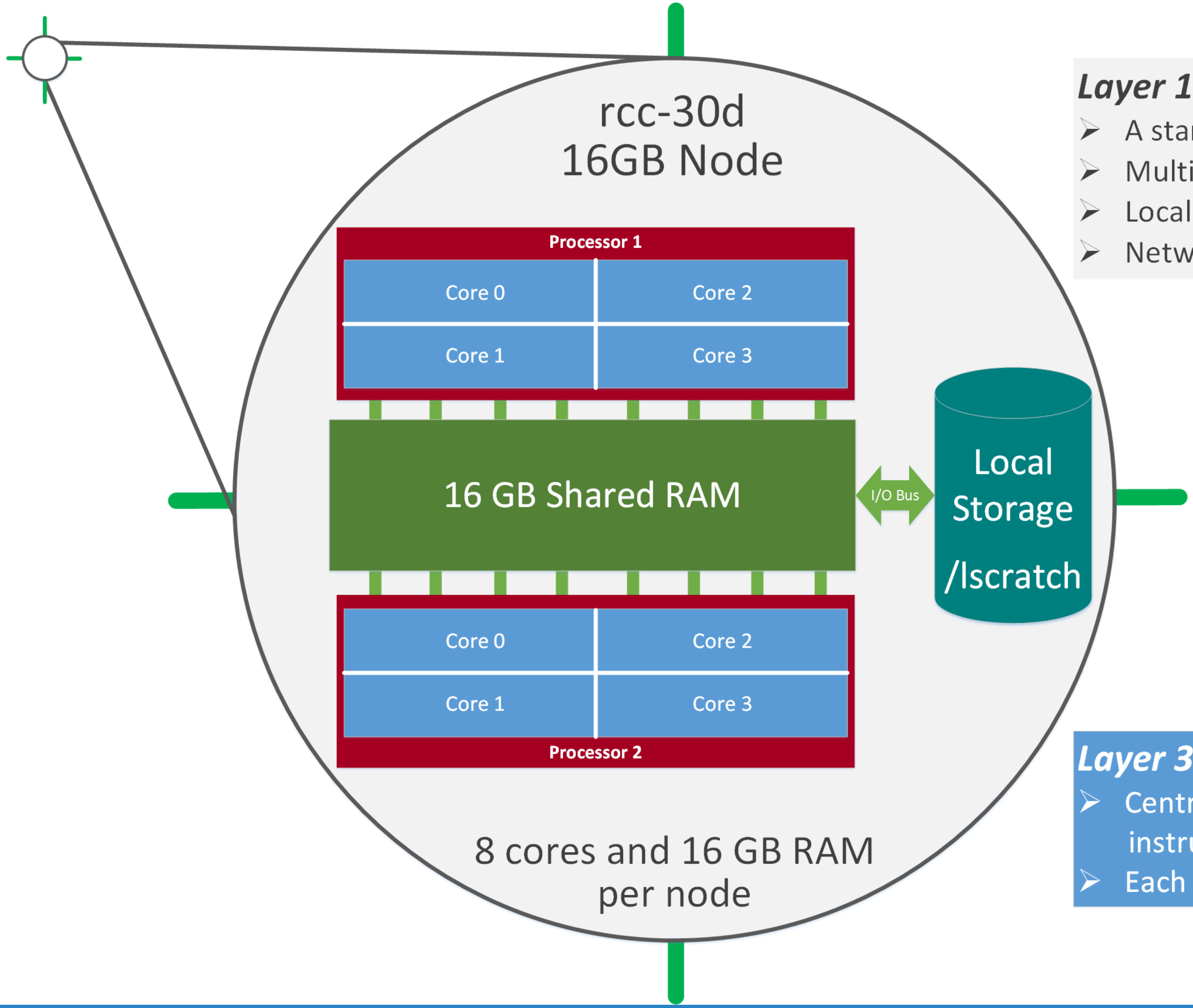
  compute nodes ⇔ compute nodes
  compute nodes ⇔ storage systems

- Queueing System: Sun Grid Engine (SGE) with qsub, qstat, qdel, etc. commands

zcluster Computing Resources

	Queue Type	Queue Name	Nodes	Processor	Cores/Node	RAM(GB)/Node	Cores	NVIDIA GPU
➡	Regular	rcc-30d	45	Intel Xeon	12	48	540	N/A
			150		8	16	1200	
	High Memory	rcc-m128-30d	4	Intel Xeon	8	192	32	N/A
			10		12	256	120	
		rcc-m512-30d	2		32	512	64	
	Multi Core	rcc-mc-30d	6	AMD Opteron	32	64	192	N/A
➡	Interactive	interq	2	AMD Opteron	48	132	96	N/A
	GPU	rcc-sgpu-30d	2	Intel Xeon	8	48	16	4 Tesla S1070 cards
		rcc-mgpu-30d	2		12	48	24	9 Tesla (Fermi) M2070 cards
		rcc-kgpu-30d	4		12	96	24	32 Tesla (Kepler) K20Xm cards

***Total peak performance:* 23 Tflops**



Layer 1: Node

- A standalone "computer in a box"
- Multiple processors, e.g. 2, sharing memory
- Local disk storage, network interface, etc.
- Networked into a cluster

Layer 2: Processor

- A single computing component
- Multicore processor, e.g. 4 cores

Layer 3: Core

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

Software Installed on zcluster

- Perl, Python, Java, awk, sed, C/C++ and Fortran compilers
- Matlab, Maple, R
- Many Bioinformatics applications: NCBI Blast+, Velvet, Trinity, TopHat, MrBayes, SoapDeNovo, Samtools, RaxML, etc.
- RCCBatchBlast (RCCBatchBlastPlus) to distribute NCBI Blast (NCBI Blast+) searches to multiple nodes.
- Many Bioinformatics Databases: NCBI Blast, Pfam, uniprot, etc.
- For a complete list of applications installed: https://wiki.gacrc.uga.edu/wiki/Software

Submit Jobs on zcluster

- To submit a batch job, you need:
  - ➢ Software installed
  - ➢ Job submission script to run the software,
    - ✓ Specifying working directory
    - ✓ Exporting environment variables, e.g.,
      OMP_NUM_THREADS (OpenMP threads number)
      LD_LIBRARY_PATH (searching paths for shared libraries)
- Job queueing commands:
  - ➢ qsub with specifying queue name
  - ➢ qstat, qdel
  - ➢ qacct, qsj, etc.

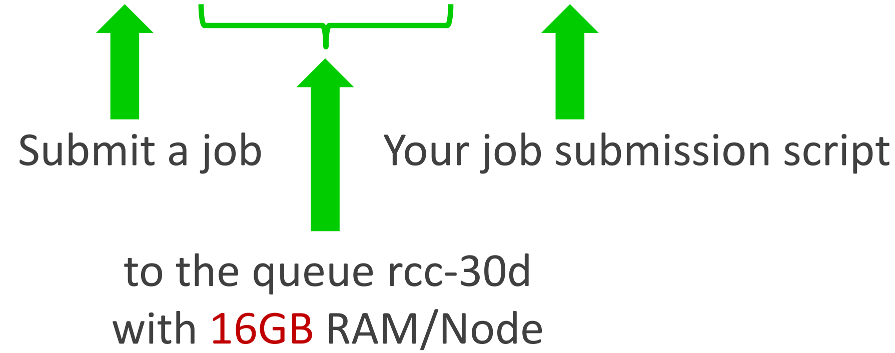
Submit Jobs on zcluster

- **Step 1**: Create a job submission script *sub.sh* running *Samtools*:

```
#!/bin/bash                   ➔ Linux shell (bash)

cd ${HOME}/testdir            ➔ Specify and enter (cd) the working directory (${HOME}/testdir)

time /usr/local/samtools/latest/samtools <command> [options]  ➔ Run samtools with ‘time’ command to measure amount of
                                                                 time it takes to run the application
```

- **Step 2**: Submit it to the queue:

```
$ qsub -q rcc-30d sub.sh
```



Submit a job

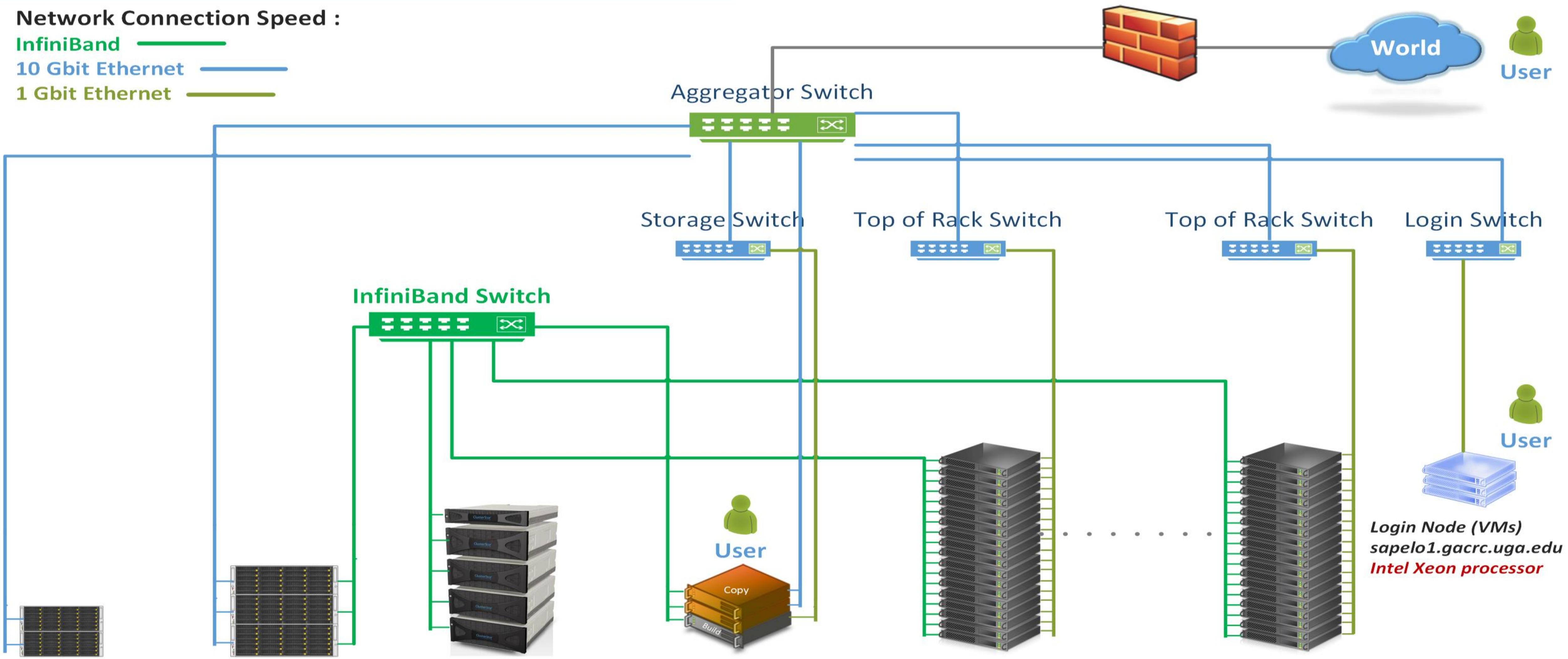
Your job submission script

to the queue rcc-30d
with 16GB RAM/Node

What is GACRC Sapelo

- Cluster Structural Diagram
- General Information
- Computing Resources
- Software installed
- Submit Jobs

The New GACRC Linux HPC Cluster Structural Diagram



Network Connection Speed :

InfiniBand

10 Gbit Ethernet

1 Gbit Ethernet

World

User

Aggregator Switch

Storage Switch

Top of Rack Switch

Top of Rack Switch

Login Switch

InfiniBand Switch

User

User

Copy

Build

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

Sapelo General Information

Sapelo is a Linux high performance computing (HPC) cluster:

- Operating System: 64-bit CentOS Linux 6.5
- 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)

- Internodal communication: InfiniBand network

  compute nodes ⇔ compute nodes

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

- Queueing System: Torque + Moab with qsub, qstat, qdel, etc. commands

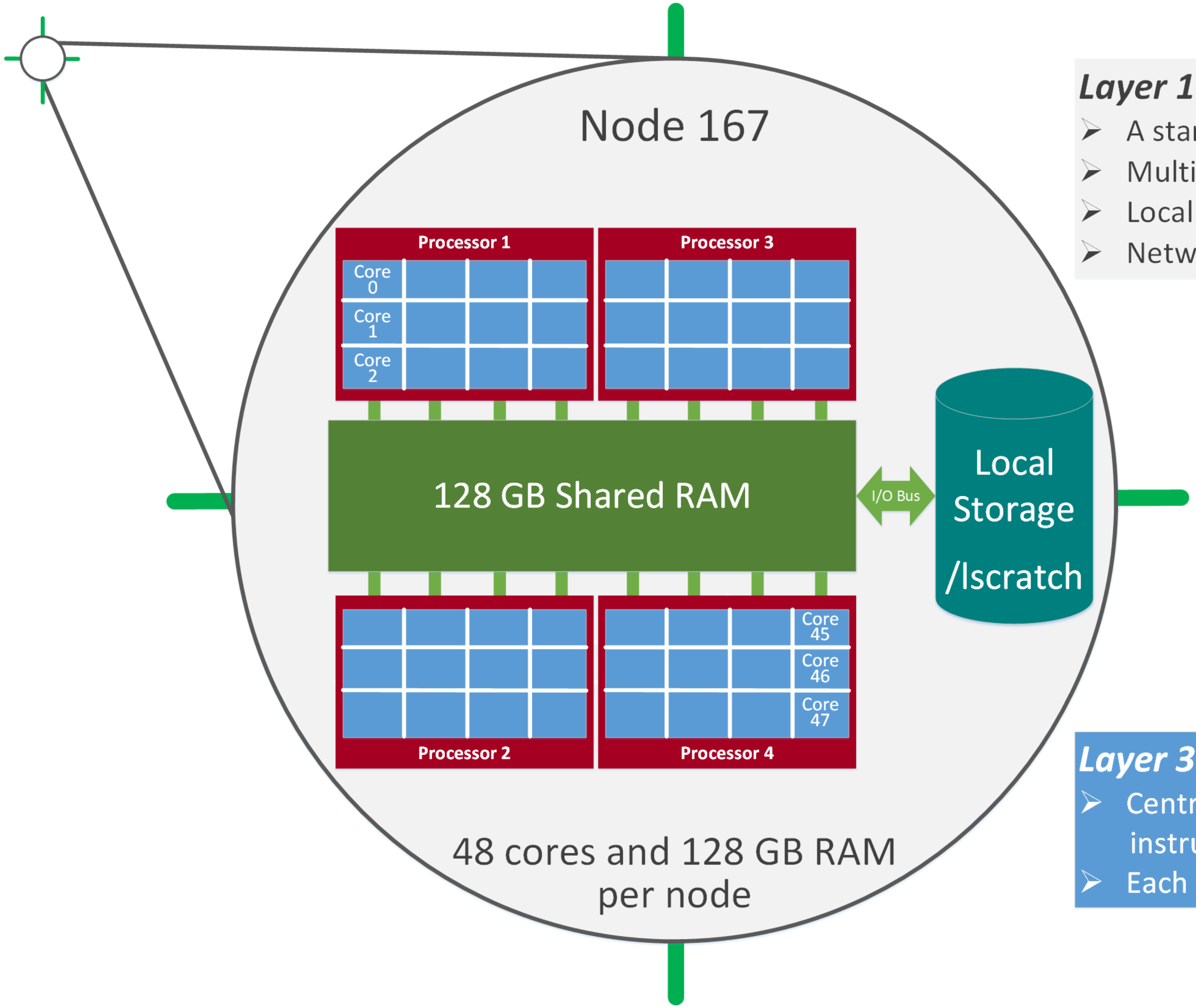
Sapelo Computing Resources

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!



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

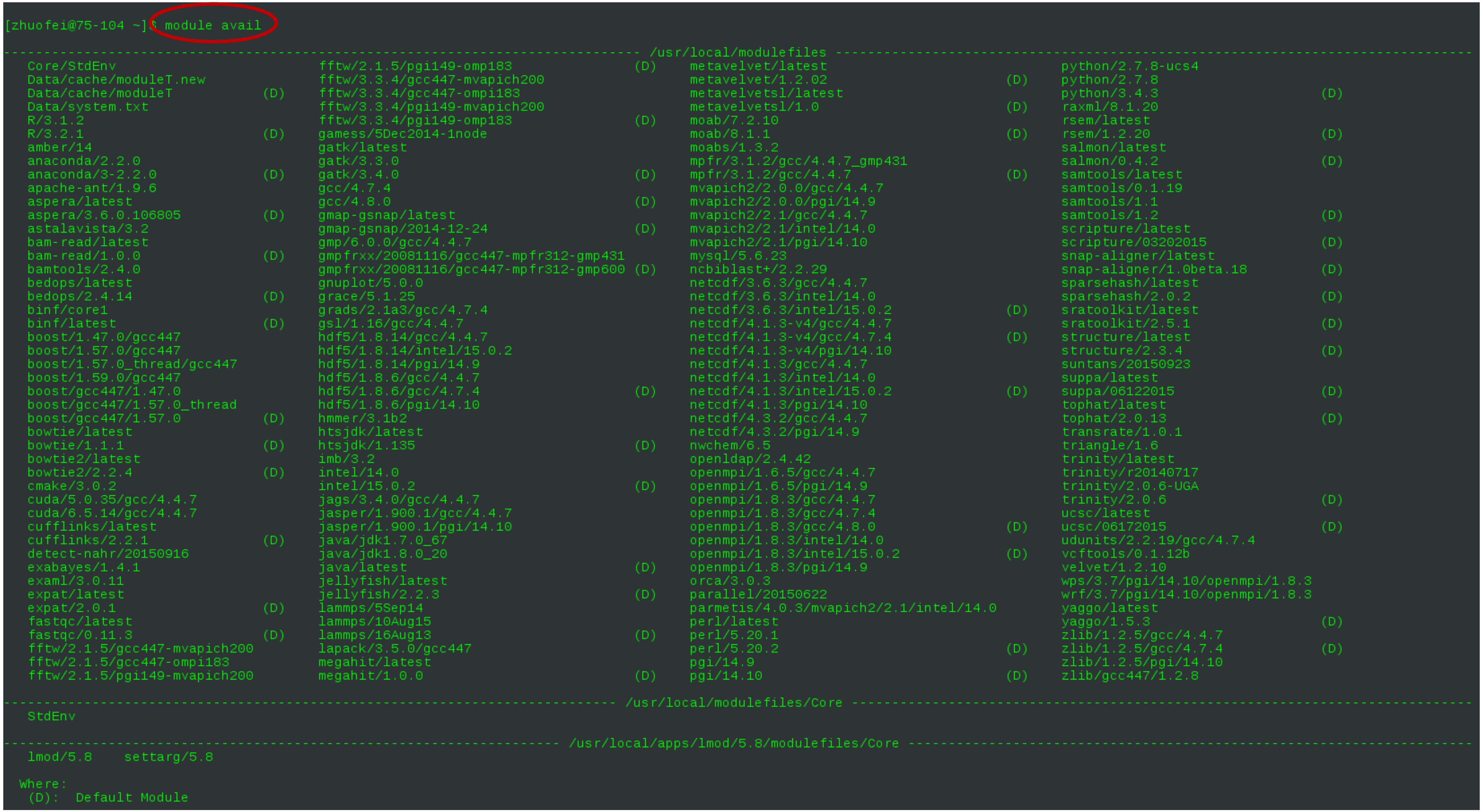
Software Installed on Sapelo

- Sapelo uses environment modules to define paths for software
- Current number of modules installed is ~90 and expanding daily!
- `module avail` ➔ List all modules available on Sapelo

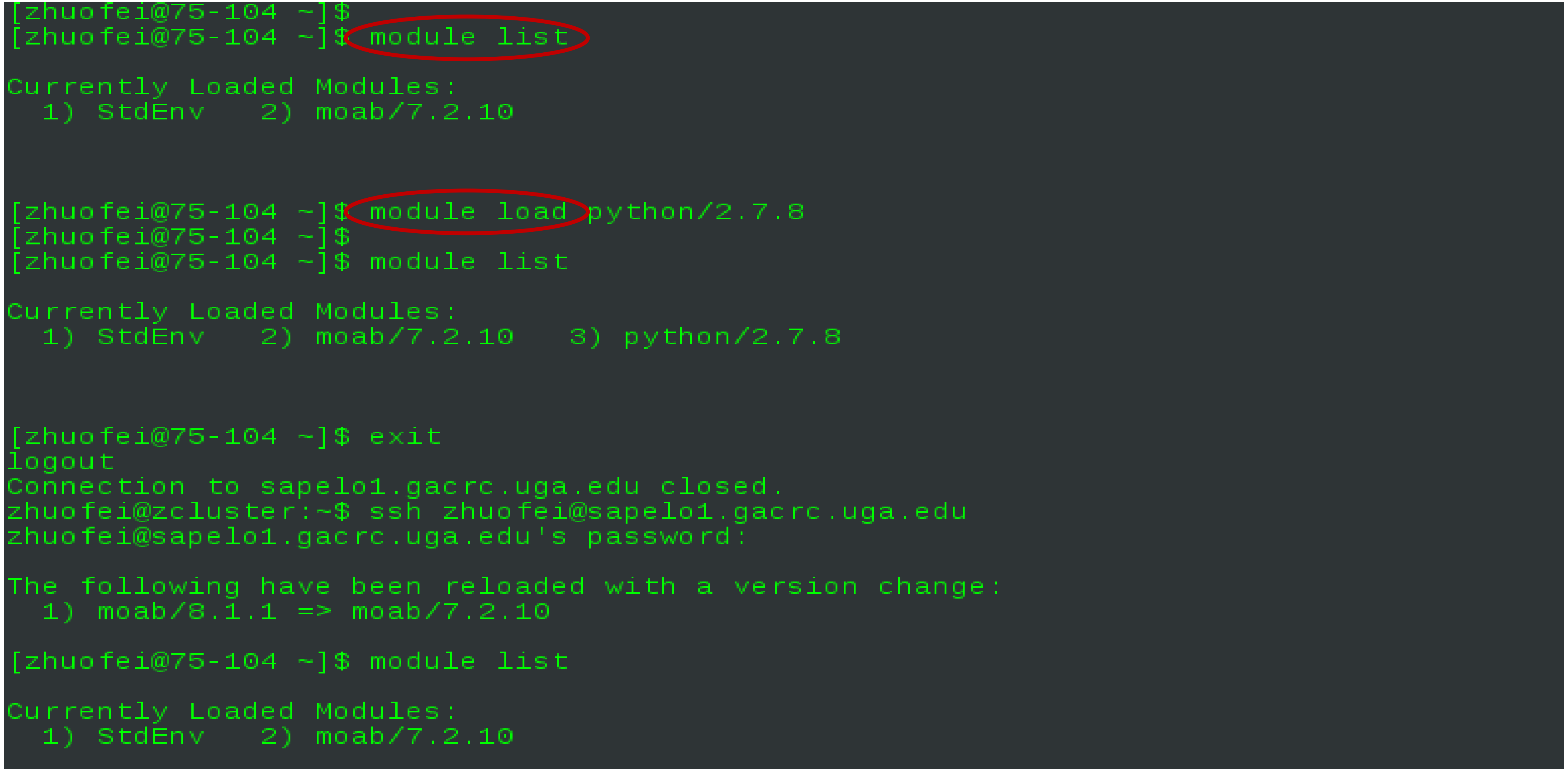
  `module list` ➔ List modules currently being loaded for use

  `module load` ➔ Load modules needed

  `module unload` ➔ Unload modules not needed



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

------------------------------------------------------------------------------- /usr/local/modulefiles --------------------------------------------------------------------------------
   Core/StdEnv                         fftw/2.1.5/pgi149-omp183               (D)    metavelvet/latest                              python/2.7.8-ucs4
   Data/cache/moduleT.new              fftw/3.3.4/gcc447-mvapich200                  metavelvet/1.2.02                       (D)    python/2.7.8
   Data/cache/moduleT           (D)    fftw/3.3.4/gcc447-ompi183                     metavelvetsl/latest                            python/3.4.3                    (D)
   Data/system.txt                     fftw/3.3.4/pgi149-mvapich200                  metavelvetsl/1.0                        (D)    raxml/8.1.20
   R/3.1.2                             fftw/3.3.4/pgi149-omp183               (D)    moab/7.2.10                                    rsem/latest
   R/3.2.1                      (D)    gamess/5Dec2014-1node                         moab/8.1.1                              (D)    rsem/1.2.20                     (D)
   amber/14                            gatk/latest                                   moabs/1.3.2                                    salmon/latest
   anaconda/2.2.0                      gatk/3.3.0                                    mpfr/3.1.2/gcc/4.4.7_gmp431                    salmon/0.4.2                    (D)
   anaconda/3-2.2.0             (D)    gatk/3.4.0                             (D)    mpfr/3.1.2/gcc/4.4.7                    (D)    samtools/latest
   apache-ant/1.9.6                    gcc/4.7.4                                     mvapich2/2.0.0/gcc/4.4.7                       samtools/0.1.19
   aspera/latest                       gcc/4.8.0                              (D)    mvapich2/2.0.0/pgi/14.9                        samtools/1.1
   aspera/3.6.0.106805          (D)    gmap-gsnap/latest                             mvapich2/2.1/gcc/4.4.7                         samtools/1.2                    (D)
   astalavista/3.2                     gmap-gsnap/2014-12-24                  (D)    mvapich2/2.1/intel/14.0                        scripture/latest
   bam-read/latest                     gmp/6.0.0/gcc/4.4.7                           mvapich2/2.1/pgi/14.10                         scripture/03202015              (D)
   bam-read/1.0.0               (D)    gmpfrxx/20081116/gcc447-mpfr312-gmp431        mysql/5.6.23                                   snap-aligner/latest
   bamtools/2.4.0                      gmpfrxx/20081116/gcc447-mpfr312-gmp600 (D)    ncbiblast+/2.2.29                              snap-aligner/1.0beta.18         (D)
   bedops/latest                       gnuplot/5.0.0                                 netcdf/3.6.3/gcc/4.4.7                         sparsehash/latest
   bedops/2.4.14                (D)    grace/5.1.25                                  netcdf/3.6.3/intel/14.0                        sparsehash/2.0.2                (D)
   binf/core1                          grads/2.1a3/gcc/4.7.4                         netcdf/3.6.3/intel/15.0.2               (D)    sratoolkit/latest
   binf/latest                  (D)    gsl/1.16/gcc/4.4.7                            netcdf/4.1.3-v4/gcc/4.4.7                      sratoolkit/2.5.1                (D)
   boost/1.47.0/gcc447                 hdf5/1.8.14/gcc/4.4.7                         netcdf/4.1.3-v4/gcc/4.7.4               (D)    structure/latest
   boost/1.57.0/gcc447                 hdf5/1.8.14/intel/15.0.2                      netcdf/4.1.3-v4/pgi/14.10                      structure/2.3.4                 (D)
   boost/1.57.0_thread/gcc447          hdf5/1.8.14/pgi/14.9                          netcdf/4.1.3/gcc/4.4.7                         suntans/20150923
   boost/1.59.0/gcc447                 hdf5/1.8.6/gcc/4.4.7                          netcdf/4.1.3/intel/14.0                        suppa/latest
   boost/gcc447/1.47.0                 hdf5/1.8.6/gcc/4.7.4                   (D)    netcdf/4.1.3/intel/15.0.2               (D)    suppa/06122015                  (D)
   boost/gcc447/1.57.0_thread          hdf5/1.8.6/pgi/14.10                          netcdf/4.1.3/pgi/14.10                         tophat/latest
   boost/gcc447/1.57.0          (D)    hmmer/3.1b2                                   netcdf/4.3.2/gcc/4.4.7                         tophat/2.0.13                   (D)
   bowtie/latest                       htsjdk/latest                                 netcdf/4.3.2/pgi/14.9                          transrate/1.0.1
   bowtie/1.1.1                 (D)    htsjdk/1.135                           (D)    nwchem/6.5                                     triangle/1.6
   bowtie2/latest                      imb/3.2                                       openldap/2.4.42                                trinity/latest
   bowtie2/2.2.4                (D)    intel/14.0                                    openmpi/1.6.5/gcc/4.4.7                        trinity/r20140717
   cmake/3.0.2                         intel/15.0.2                           (D)    openmpi/1.6.5/pgi/14.9                         trinity/2.0.6-UGA
   cuda/5.0.35/gcc/4.4.7               jags/3.4.0/gcc/4.4.7                          openmpi/1.8.3/gcc/4.4.7                        trinity/2.0.6                   (D)
   cuda/6.5.14/gcc/4.4.7               jasper/1.900.1/gcc/4.4.7                      openmpi/1.8.3/gcc/4.7.4                        ucsc/latest
   cufflinks/latest                    jasper/1.900.1/pgi/14.10                      openmpi/1.8.3/gcc/4.8.0                 (D)    ucsc/06172015                   (D)
   cufflinks/2.2.1              (D)    java/jdk1.7.0_67                              openmpi/1.8.3/intel/14.0                       udunits/2.2.19/gcc/4.7.4
   detect-nahr/20150916                java/jdk1.8.0_20                              openmpi/1.8.3/intel/15.0.2              (D)    vcftools/0.1.12b
   exabayes/1.4.1                      java/latest                            (D)    openmpi/1.8.3/pgi/14.9                         velvet/1.2.10
   examl/3.0.11                        jellyfish/latest                              orca/3.0.3                                     wps/3.7/pgi/14.10/openmpi/1.8.3
   expat/latest                        jellyfish/2.2.3                        (D)    parallel/20150622                              wrf/3.7/pgi/14.10/openmpi/1.8.3
   expat/2.0.1                  (D)    lammps/5Sep14                                 parmetis/4.0.3/mvapich2/2.1/intel/14.0         yaggo/latest
   fastqc/latest                       lammps/10Aug15                                perl/latest                                    yaggo/1.5.3                     (D)
   fastqc/0.11.3                (D)    lammps/16Aug13                         (D)    perl/5.20.1                                    zlib/1.2.5/gcc/4.4.7
   fftw/2.1.5/gcc447-mvapich200        lapack/3.5.0/gcc447                           perl/5.20.2                             (D)    zlib/1.2.5/gcc/4.7.4            (D)
   fftw/2.1.5/gcc447-ompi183           megahit/latest                                pgi/14.9                                       zlib/1.2.5/pgi/14.10
   fftw/2.1.5/pgi149-mvapich200        megahit/1.0.0                          (D)    pgi/14.10                               (D)    zlib/gcc447/1.2.8

---------------------------------------------------------------------------- /usr/local/modulefiles/Core ------------------------------------------------------------------------------
   StdEnv

--------------------------------------------------------------------- /usr/local/apps/lmod/5.8/modulefiles/Core -----------------------------------------------------------------------
   lmod/5.8    settarg/5.8

  Where:
   (D):  Default Module
```



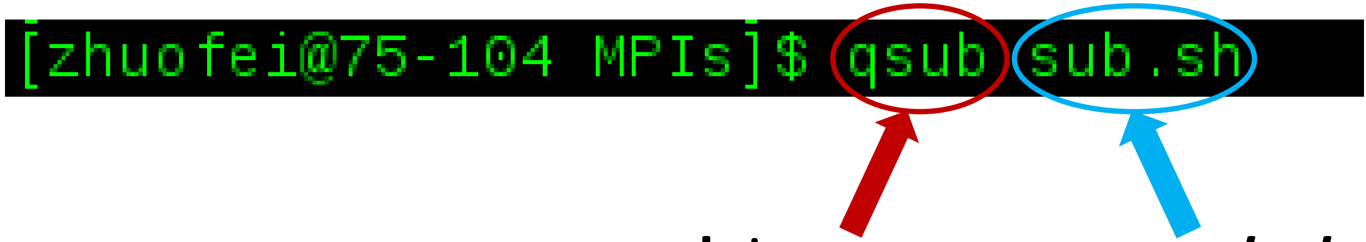
```
[zhuofei@75-104 ~]$
[zhuofei@75-104 ~]$ module list

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



[zhuofei@75-104 ~]$ module load python/2.7.8
[zhuofei@75-104 ~]$
[zhuofei@75-104 ~]$ module list

Currently Loaded Modules:
  1) StdEnv   2) moab/7.2.10   3) python/2.7.8



[zhuofei@75-104 ~]$ exit
logout
Connection to sapelo1.gacrc.uga.edu closed.
zhuofei@zcluster:~$ ssh zhuofei@sapelo1.gacrc.uga.edu
zhuofei@sapelo1.gacrc.uga.edu's password:

The following have been reloaded with a version change:
  1) moab/8.1.1 => moab/7.2.10

[zhuofei@75-104 ~]$ module list

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

Submit Batch Jobs on Sapelo

- To submit a batch job, you need:
  - ➢ Software loaded. If not, used `module load`
  - ➢ Job submission script to run the software, specifying working directory and computing resources:
    - ✓ Number of nodes and cores
    - ✓ Amount of memory
    - ✓ Type of nodes
    - ✓ Maximum wallclock time, etc.
- Job queueing commands:
  - ➢ qsub, qstat, qdel
  - ➢ showq, checkjob, etc.

Submit Batch Jobs on Sapelo

- 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!***

Submit Batch Jobs on Sapelo

- Example: Serial job submission 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=20gb	→ 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

Where to Find Useful Information?

- GACRC Web: http://gacrc.uga.edu/
- GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
- GACRC Help : http://gacrc.uga.edu/help/
- GACRC Training: https://wiki.gacrc.uga.edu/wiki/Training
- GACRC User Account: https://wiki.gacrc.uga.edu/wiki/User_Accounts
- GACRC Software: https://wiki.gacrc.uga.edu/wiki/Software

Telephone Support

EITS HELPDESK: 706-542-3106

MONDAY – THURSDAY: 8AM – 10PM

FRIDAY: 8AM – 6PM

SATURDAY – SUNDAY: 1PM – 7PM

Georgia Advanced Computing Resource Center
4098C Stegeman Coliseum
University of Georgia
Athens, GA 30602

Thank You!