



Introduction to HPC Using Sapelo Cluster at GACRC

Georgia Advanced Computing Resource Center (GACRC)

EITS/University of Georgia

Zhuofei Hou zhuofei@uga.edu



Outline

- GACRC
- Sapelo Cluster
- Job Submission Workflow
- Work with Sapelo
- Guideline and Practical Tips



GACRC

- We are a high-performance-computing (HPC) center at UGA
- We provide to the UGA research and education community an advanced computing environment:
 - HPC computing and networking infrastructure located at the Boyd Data Center
 - Comprehensive collection of scientific, engineering and business applications
 - Consulting and training services
- <http://wiki.gacrc.uga.edu> (GACRC Wiki)
- https://wiki.gacrc.uga.edu/wiki/Getting_Help (GACRC Support)
- <http://gacrc.uga.edu> (GACRC Web)



Sapelo Cluster

- Cluster Diagrams
- Cluster Overview
- Computing Resources
- Storage Environment

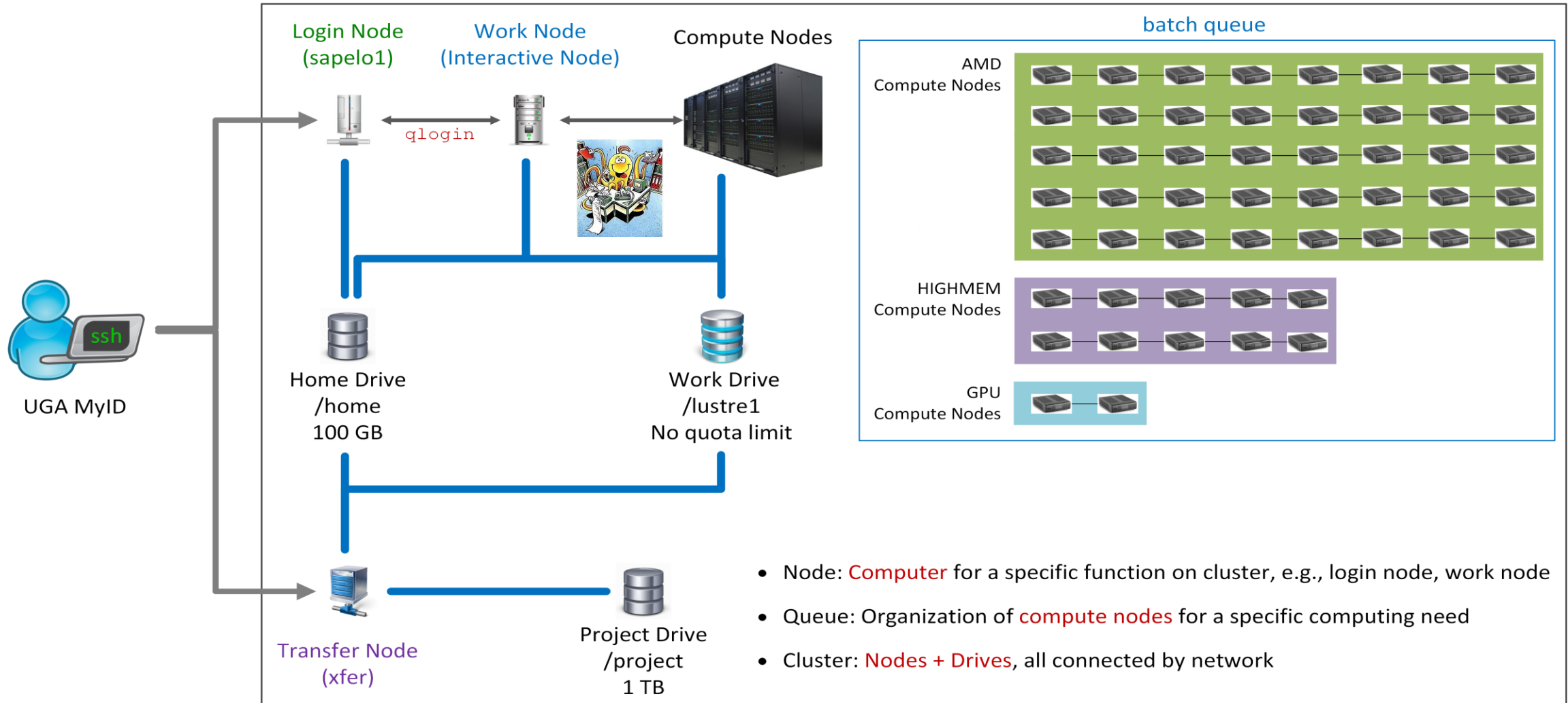
What is a Cluster?

When you work on cluster, **3** roles are working:

- User: to submit a job
- Queueing System: to dispatch a job to run on cluster
- Cluster: to run a job



Sapelo Cluster Diagram

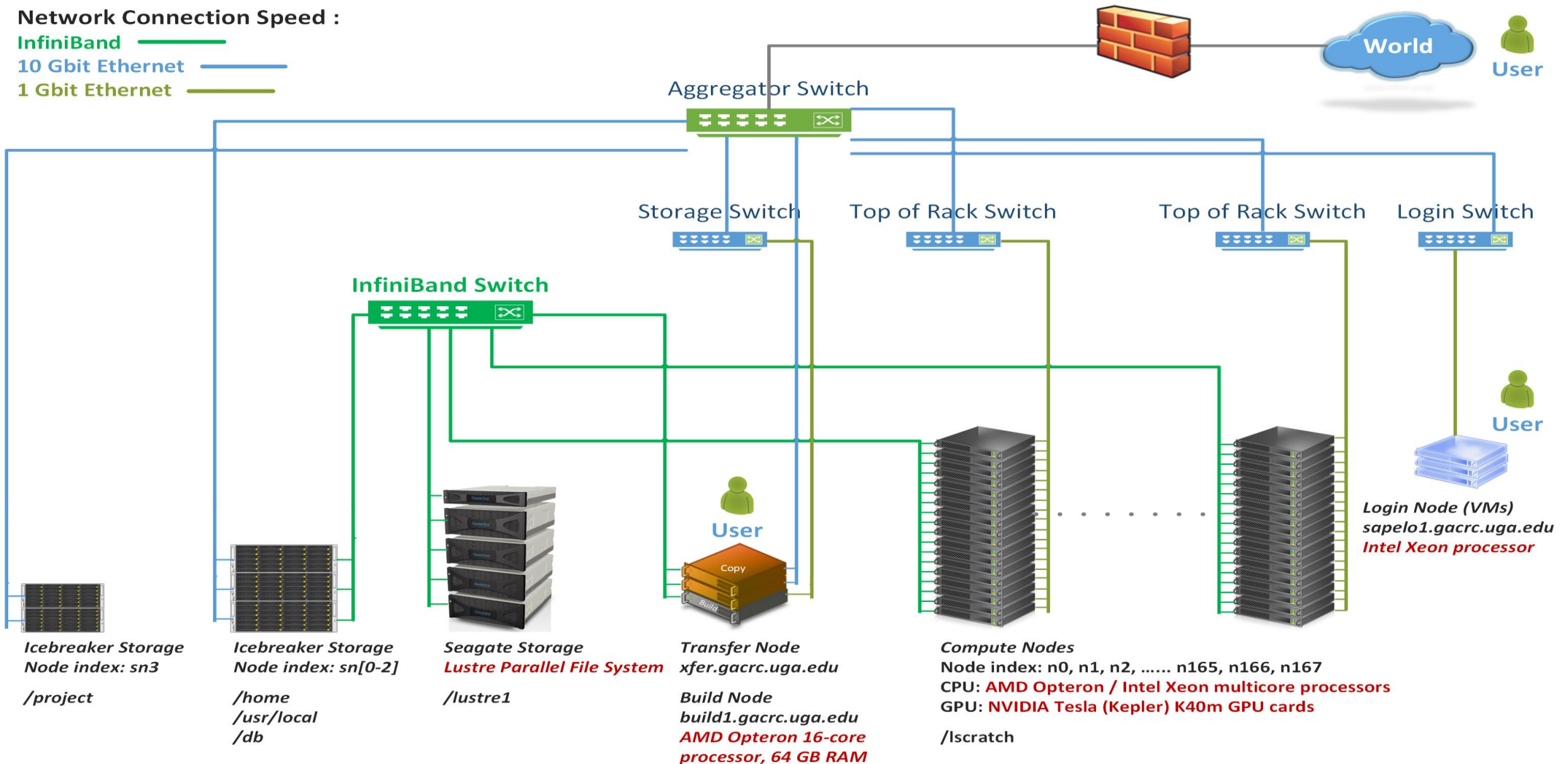


- Node: **Computer** for a specific function on cluster, e.g., login node, work node
- Queue: Organization of **compute nodes** for a specific computing need
- Cluster: **Nodes + Drives**, all connected by network

The New GACRC Linux HPC Cluster Structural Diagram

Network Connection Speed :

- InfiniBand —
- 10 Gbit Ethernet —
- 1 Gbit Ethernet —





Cluster Overview

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

- OS: 64-bit CentOS Linux 6.5
- You can log on to:

Login (username@sapelo1.gacrc.uga.edu) : edit script, submit batch job (not suggested)

Transfer (username@xfer.gacrc.uga.edu) : transfer, compress, package data

Build (username@build1.gacrc.uga.edu) : compile, build, test

- **Login** ^{qlogin}  **Work Node** : edit script, submit batch job, run interactive job

- Internodal Communication: **InfiniBand network**

compute nodes ↔ compute nodes ↔ storage systems

Cluster Overview

- 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 Sapelo:
 - **Torque** Resource Manager + **Moab** Workload Manager
 - Queueing commands: `qsub`, `qstat`, `qdel`
`showjobs`, `showq`

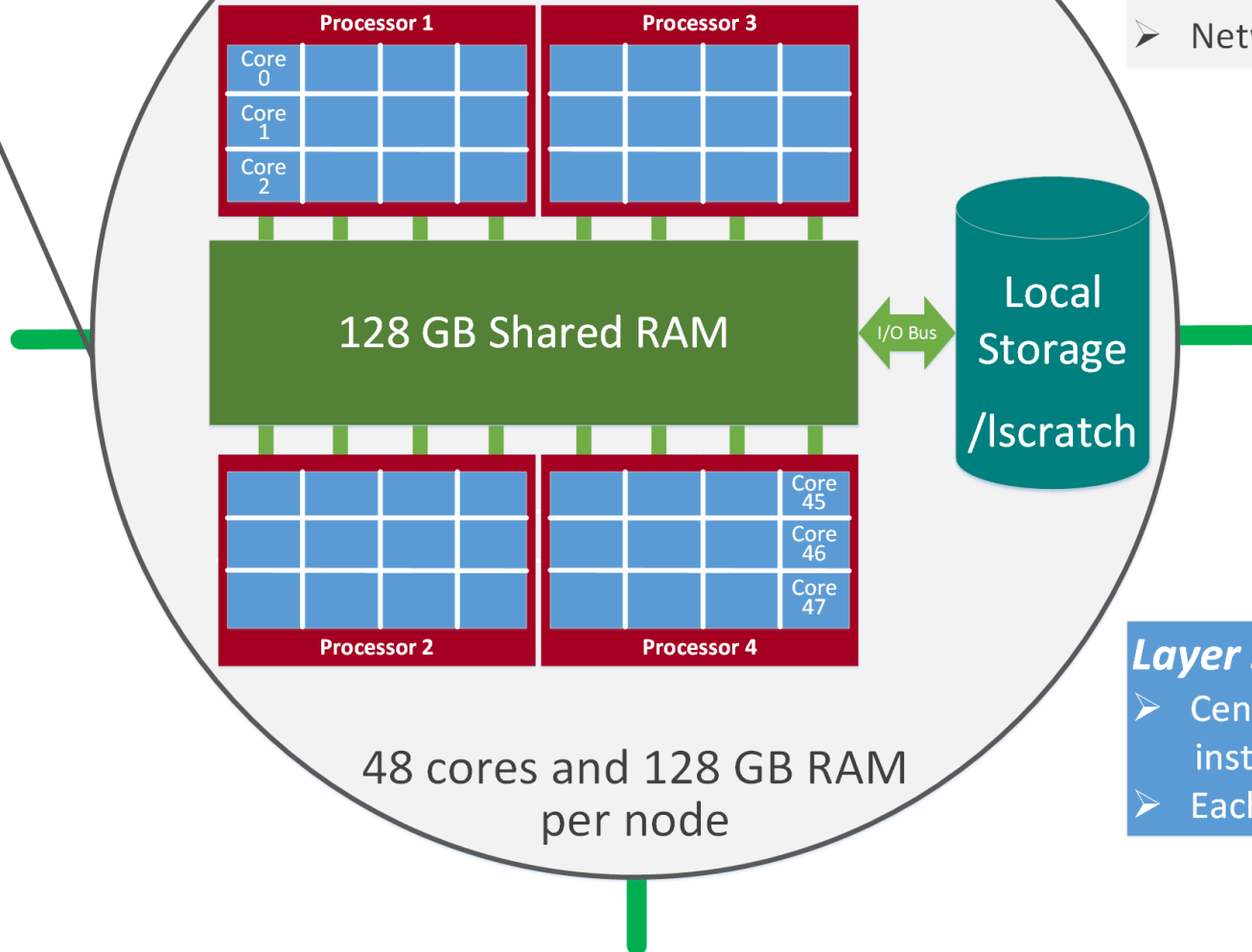




Computing Resources

Queue	Node Feature	Total	RAM(GB) /Node	Max RAM(GB) /Single-node Job	Cores /Node	Processor	GPU	GPU Cards /Node	InfiniBand
batch	AMD	112	128	126	48	AMD Opteron	N/A		Yes
		4	256	252					
	HIGHMEM	6	512	504					
		4	1024 (1)	997	28				
	1024 (3)								
	GPU	2	128	126	16	Intel Xeon	NVIDIA K40m	8	
abcnode (buy-in)	35	variable	variable	variable	variable	variable	variable		

Node 167



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



Storage Environment

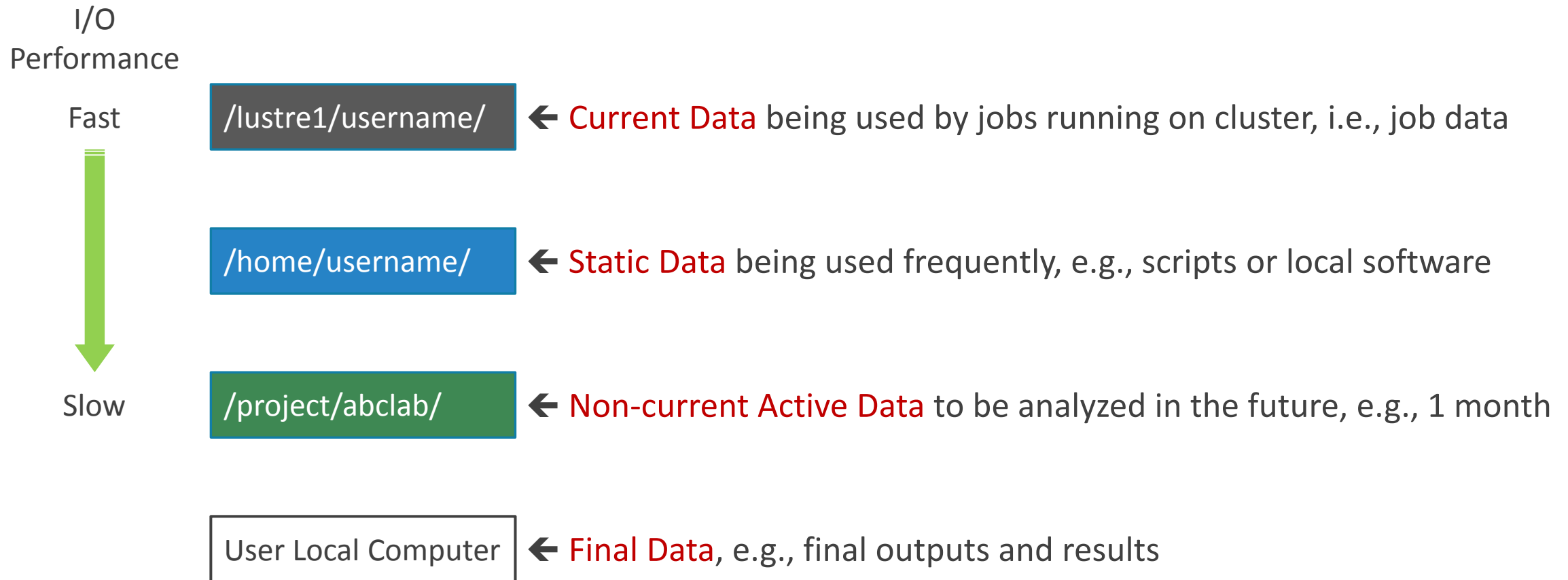
4 Filesystems	Role	Quota	Accessible from	Intended Use	Notes
→ /home/username/	Home	100GB	sapelo1.gacrc.uga.edu (Login) Interactive nodes (Interactive) xfer.gacrc.uga.edu (Transfer) build1.gacrc.uga.edu (Build) compute nodes (Compute)	Static data being used frequently, e.g., scripts, local software	Snapshots
→ /lustre1/username/	Global Scratch	No Limit	Interactive nodes (Interactive) xfer.gacrc.uga.edu (Transfer) compute nodes (Compute)	Current data being read from/written into by jobs running on cluster	User to clean up! Subject to deletion in 30 days*
/tmp/lscratch/	Local Scratch	N/A	Individual compute node	Jobs with heavy disk I/O	User to clean up!
→ /project/abclab/	Storage	1TB (Initial)	xfer.gacrc.uga.edu (Transfer)	Temporary data parking for non-current active data	Group sharing possible

Note: /usr/local/apps : Software installation directory ; /db : Bioinformatics database installation directory

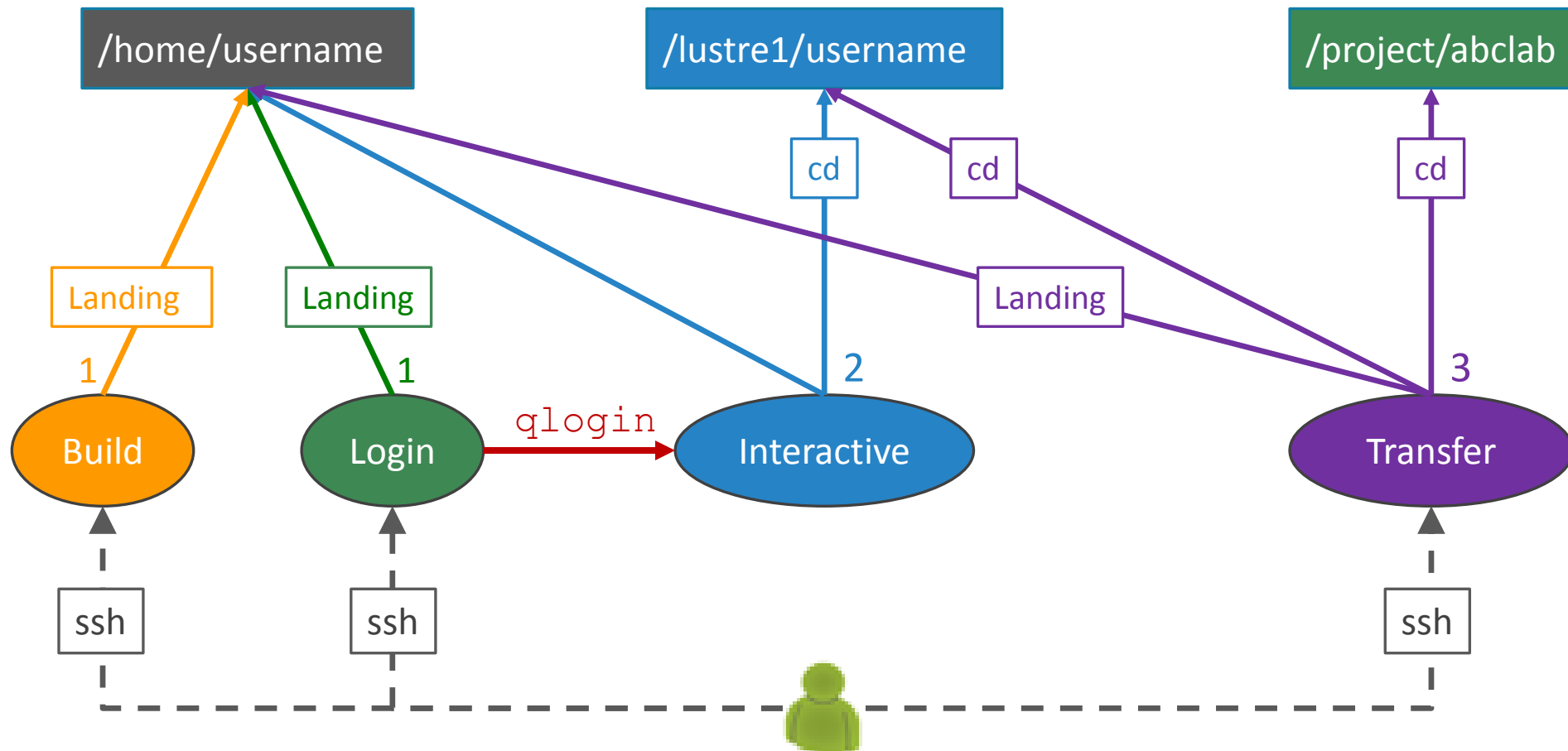
* denotes component or policy to be implemented in the future



Storage Environment - *Data Storing Rule*



Storage Environment - *Accessing Rule of 123*





Storage Environment

7 Main Functions	Related Filesystem	Related Node
Login Landing	/home/username (Home) (Always!)	Login or Transfer or Build
Batch Job Submitting	/home/username (Home)	Login
	/lustre1/username (Scratch) (Suggested!) /home/username (Home)	Interactive
Interactive Job Running	/lustre1/username (Scratch) /home/username (Home)	
Data Transferring, Archiving , Compressing	/lustre1/username (Scratch) /home/username (Home)	Transfer
Long-term Active Data Storing	/project/abclab (Project)	
Code Compilation, Test	/home/username (Home)	Build
Job Data Temporarily Storing	/lustre1/username (Scratch) /lscratch/username (Local Scratch)*	Compute

Note: * denotes component or policy to be implemented in the future

Job Submission Workflow

Job Working Space is Global Scratch: `/lustre1/username/`

- **Why?**

No quota limit + The I/O fastest filesystem

- **How to access?**

From **Interactive Node**, `cd /lustre1/username/`

- **What need to pay attention?**

Clean up! Non-current Active Data → `/project/abclab/`

Non-active Data → local storage

} using **Transfer Node**





Job Submission Workflow

1. Linux/Mac user:

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



Windows user:



Login



2. `qlogin`

Interactive



3. `cd /lustre1/username`

4. `mkdir ./workDir`

5. `cd ./workDir`

7. `nano ./sub.sh`



lustre1

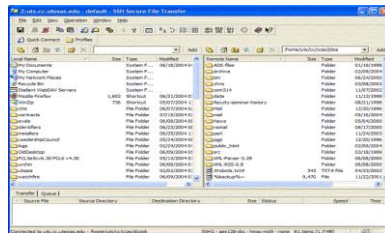
```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=1:AMD
#PBS -l mem=20gb
#PBS -l walltime=48:00:00
cd $PBS_O_WORKDIR
module load ncbiblast+/2.2.29
blastn [options] > outputfile
```

6. Linux/Mac user:

`scp file username@xfer.gacrc.uga.edu:/lustre1/username/workDir`



Windows user:



8.

```
$ qsub sub.sh
```

Useful PBS headers:

`#PBS mem=200gb` : use 256GB AMD nodes
`ppn=24` : request 24 cores for 24 threads

Note: `-r` option of `scp` command will recursively copy a directory



Job Submission Workflow

1. Log on to Sapelo **Login** node: `ssh username@sapelol.gacrc.uga.edu`
2. From Sapelo **Login** node, transfer to **Interactive** node by issuing the command: `qlogin`
3. From **Interactive** node: Change directory to global scratch: `cd /lustrel/username`
4. Create a working subdirectory on global scratch: `mkdir ./workDir`
5. Change directory to `workDir`: `cd ./workDir`
6. Transfer data to `workDir` using `scp` or **SSH File Transfer** (with `tar` or `gzip`)
7. Make a Sapelo job submission script: `nano ./sub.sh`
8. Submit job: `qsub ./sub.sh`

Useful PBS headers: `#PBS -l mem=200gb` : use 256GB high-RAM AMD nodes
`#PBS -l nodes=1:ppn=24:AMD` : request 24 cores for 24 threads, max **48!**



Work on Sapelo

Before we start:

GACRC Wiki: <http://wiki.gacrc.uga.edu>

GACRC Wiki Running Jobs: https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_Sapelo

GACRC Wiki Software: <https://wiki.gacrc.uga.edu/wiki/Software>

GACRC Wiki Linux Command: https://wiki.gacrc.uga.edu/wiki/Command_List

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

Work on Sapelo

To submit a ticket to us?

➤ **Job Troubleshooting:**

Please tell us details of your question or problem, including but not limited to:

- ✓ Your user name
- ✓ Your job ID
- ✓ Your working directory
- ✓ The queue name and command you used to submit the job

➤ **Software Installation:**

- ✓ Specific name and version of the software
- ✓ Download website
- ✓ Supporting package information if have

Note:

It's **USER's** responsibility to make sure the **correctness of datasets** being used by jobs!





Work on Sapelo

- User Account
- Connect and Login
- Transfer Files Using Transfer Node
- Software Packages
- Run Interactive Jobs
- Run Batch Jobs
 - ✓ Submit Batch Jobs
 - ✓ ***Serial, Threaded,*** and ***MPI*** Batch Job Submission Scripts
 - ✓ Check Job Status and Cancel a Job
 - ✓ Check Computing Resources Used by a Job

User Account

- User Account: **UGAMyID@sapelo1.gacrc.uga.edu**

A **valid official UGA MyID** is a MUST to create a user account!



- To get a user account:

1. Computing Lab Registration: <http://help.gacrc.uga.edu/labAcct.php> (for PI of a new group)
2. User Account Request: <http://help.gacrc.uga.edu/userAcct.php> (for PI of an existing group)
3. New User Training: <http://gacrc.uga.edu/help/training/>
4. **Welcome letter** with whole package of information about your Sapelo user account



Connect and Login

- On Linux/Mac: use Terminal utility 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 with its UGI to be forwarded to local

⁽²⁾ On Windows, use a *SSH client* to open the connection (next page))

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

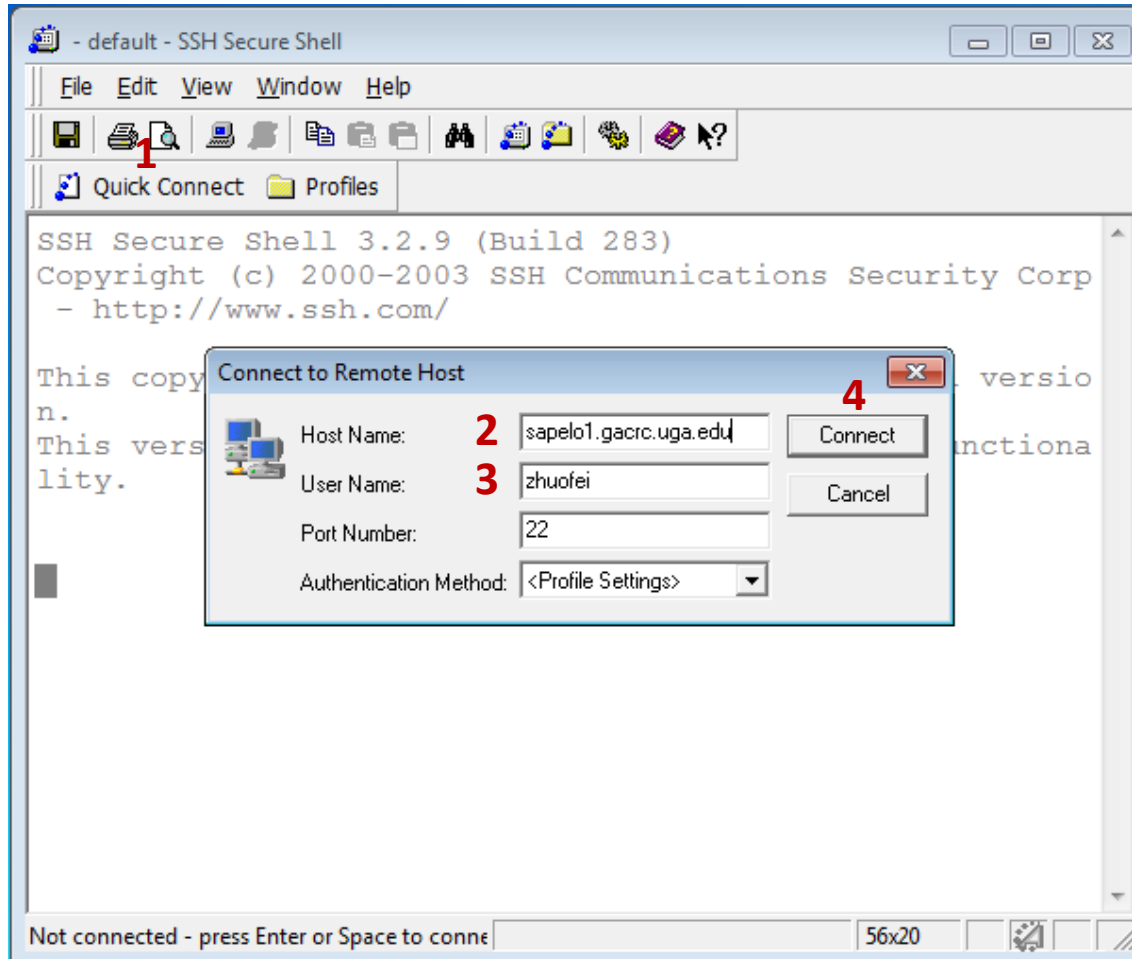
```
zhuofei@sapelo1.gacrc.uga.edu's password: █
```

⁽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
```

Connect and Login



1. To download:

http://eits.uga.edu/hardware_and_software/software/

with your UGA MyID and password

2. After connection is built, working environment is Linux, same as Linux/Mac users'



Transfer Node `xfer.gacrc.uga.edu`

- ✓ `ssh username@xfer.gacrc.uga.edu` with your **UGA MyID password**
- ✓ Landing directory: `/home/username` (Sapelo home)
- ✓ Move data into/out of Sapelo (`scp`, `sftp`, `rsync`, **SSH File Transfer**, **FileZilla**)
- ✓ Compress or package data on Sapelo (`tar`, `gzip`)
- ✓ Transfer data between Sapelo and zcluster (`cp`, `mv`)
- ✓ Filesystems you can access:
 - `/home/username/` : Sapelo home (landing directory)
 - `/lustre1/username/` : Sapelo global scratch
 - `/project/abclab/` : long-term active data storage
 - `/panfs/pstor.storage/home/abclab/username/` : zcluster home
 - `/escratch4/username/` : zcluster scratch
- ✓ Most file systems on Transfer are *auto-mounted* upon *the first time full-path access*, e.g.,
`cd /project/abclab/`

Linux/Mac Users Transfer Files Using scp/sftp



Type in Terminal utility on your local computer: `scp (-r) [Source] [Target]`

E.g. 1: working on local machine, from Local → Sapelo global scratch

```
scp ./file zhuofei@xfer.gacrc.uga.edu:/lustre1/zhuofei/workDir/
```

```
scp -r ./folder/ zhuofei@xfer.gacrc.uga.edu:/lustre1/zhuofei/workDir/
```

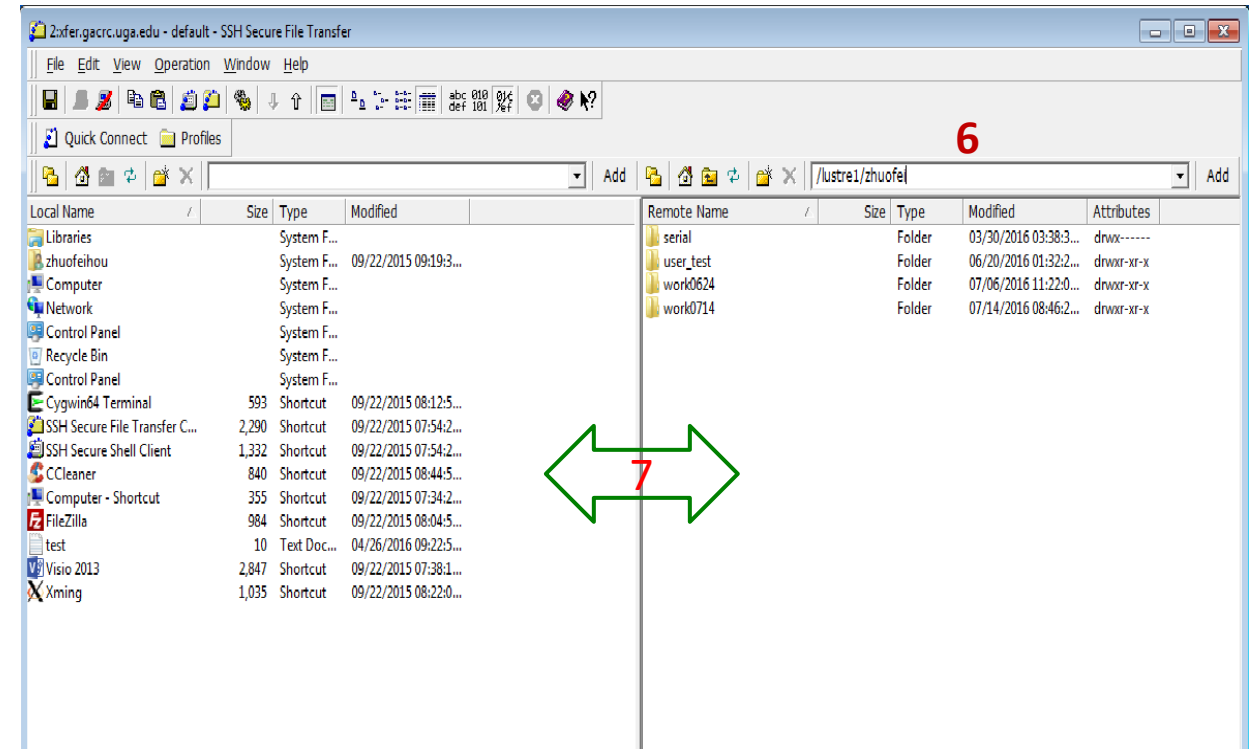
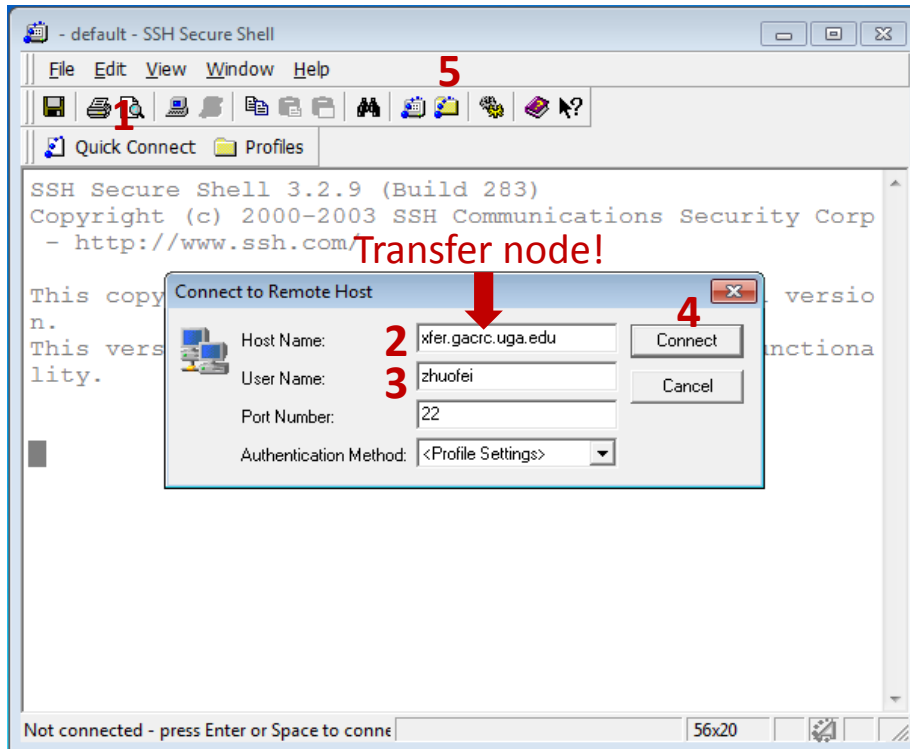
E.g. 2: working on local machine, from Sapelo global scratch → Local

```
scp zhuofei@xfer.gacrc.uga.edu:/lustre1/zhuofei/workDir/file ./
```

```
scp -r zhuofei@xfer.gacrc.uga.edu:/lustre1/zhuofei/workDir/folder/ ./
```



Windows Users Transfer Files Using SSH File Transfer/FileZilla





Software Packages

- The cluster uses **environment modules** to define the various paths for software packages (~300 and expanding daily!)
- module avail**: List all available modules (centralized installation):

```
$ module avail
```

```
----- /usr/local/modulefiles -----
Core/StdEnv          fftw/3.3.4/pgi149-mvapich200      mafft/7.273-e          python/2.7.8
Data/cache/moduleT.new  fftw/3.3.4/pgi149-omp183          (D) mageck/0.5             python/2.7.13
Data/cache/moduleT      (D) find_cns/07212016             mageck-vispr/02242017  python/3.4.3
Data/system.txt        fineradstructure/12092016         magickwand/0.2         python/3.5.1          (D)
R/3.0.3               finestructure/2.0.7               magma/2.22-9           pytorch/0.1.9
R/3.1.2               flash/1.2.11                      mahotas/1.4.3         pytorch/02272017    (D)
R/3.2.1               flex/2.6.0/gcc/4.7.4             masurca/3.1.3         qiime/1.9.1
R/3.2.3               flex/2.6.0/pgi/14.10             matlab/R2015aSP1       qt/4.8
R/3.3.1               (D) freebayes/1.0.1              matlab/R2015b          quast/4.1
StdEnv                 (D) freesurfer/6.0.0             matlab/R2016b          (D) quast/4.2          (D)
.....
```



Software Packages

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

```
$ module list
```

```
Currently Loaded Modules:  
1) moab/8.1.1  2) StdEnv
```

- `module load`: Load the needed modules:

```
$ module load python/2.7.8  
$ module load R/3.2.3  
$ module list
```

```
Currently Loaded Modules:
```

```
1) moab/8.1.1  2) StdEnv  3) python/2.7.8  4) R/3.2.3
```

- `module unload`: Remove a module:

```
$ module unload R/3.2.3  
$ module list
```

```
Currently Loaded Modules:
```

```
1) moab/8.1.1  2) StdEnv  3) python/2.7.8
```



Run Interactive Jobs

- To run an interactive job, using `qlogin` command from **Login** node:

```
[zhuofei@75-104 ~]$ qlogin
qsub: waiting for job 1058157.pbs.scm to start
qsub: job 1058157.pbs.scm ready
[zhuofei@n14 ~] ← Now I am on n14, which is an interactive node
[zhuofei@n14 ~]$ module load R/3.2.3
[zhuofei@n14 ~]$ R

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

...
[Previously saved workspace restored]

>
```

- When you are done, remember to `exit` the session



Run Batch Jobs

- Components you need to run a job:
 - **Software** already installed (cluster software or the one installed by yourself)
 - **Job submission script** to
 1. specify computing resources:
 - ✓ number of nodes and cores
 - ✓ amount of memory
 - ✓ node's feature
 - ✓ maximum wallclock time
 2. load software using **module load** (for cluster software)
 3. run any Linux commands you want to run, e.g., pwd, mkdir, cd, echo, etc.
 4. run the software
- Common queueing commands you need:
 - **qsub, qdel**
 - **qstat, showjobs, checkjob, showq**



Submit Batch Jobs

```
[zhuofei@n15 workDir]$ pwd          ← n15: interactive node
/lustrel1/zhuofei/workDir          ← /lustrel1/zhuofei/: global scratch
[zhuofei@n15 workDir]$
[zhuofei@n15 workDir]$ qsub sub.sh
1165617.pbs.scm
```

qsub is to
submit a job

sub.sh is the job submission script to

1. specify computing resources:
2. load software using **module load**
3. run any Linux commands you want to run
4. run the software



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=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 2>&1 → Run blastn with 'time' command to measure the amount of time it
                                         takes to run the application
```



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

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=1:AMD
#PBS -l mem=200gb
#PBS -l walltime=48:00:00
```

→ Job will be dispatched to run on AMD **256GB** node

```
cd $PBS_O_WORKDIR
```

```
module load ncbiblast+/2.2.29
```

```
time blastn [options] >outputfile 2>&1
```



Example 2: Threaded job script *sub.sh* running NCBI Blast+

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testBlast
#PBS -l nodes=1:ppn=24:AMD
#PBS -l mem=200gb
#PBS -l walltime=480:00:00

#PBS -M jSmith@uga.edu
#PBS -m ae
#PBS -j oe

cd $PBS_O_WORKDIR

module load ncbiblast+/2.2.29

time blastn -num_threads 24 [options] >outputfile 2>&1
```

→ Number of nodes (1), number of cores/node (24), node type (AMD)
Number of threads (24) = Number of cores requested (24)

→ Email address to receive a notification for computing resources
→ Send email notification when job aborts (a) or terminates (e)
→ Standard error file (testBlast.e1234) will be merged into standard out file (testBlast.o1234)

→ Run blastn with 24 threads (-num_threads 24)



Example 3: MPI job script *sub.sh* running RAxML

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=48:AMD
#PBS -l walltime=480:00:00
#PBS -l mem=20gb

cd $PBS_O_WORKDIR

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 2>&1
```

→ Number of nodes (2), number of cores/node (48), node type (AMD)
Total cores requested = $2 \times 48 = 96$
We suggest, Number of MPI Processes (50) \leq Number of cores requested (96)

→ To run raxmlHPC-MPI-AVX, MPI version using OpenMPI 1.8.3/Intel 15.0.2

→ Run raxmlHPC-MPI-AVX with 50 MPI processes (-np 50), default 96

```
#PBS -S /bin/bash
#PBS -q batch
#PBS -N testRAxML
#PBS -l nodes=2:ppn=27:AMD
#PBS -l walltime=480:00:00
#PBS -l mem=20g
```

→ ppn number (27) fewer than 48 MUST be a multiplier of 3!

```
cd $PBS_O_WORKDIR
```

```
# Context Sharing
```

```
CONTEXTS=$(/usr/local/bin/set_contexts.sh $PBS_NUM_PPN)
if [[ "$?" -eq "0" ]]; then
  export PSM_SHAREDCONTEXTS_MAX=$CONTEXTS
fi
```

} New lines copied from GACRC Wiki

```
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 2>&1
```

→ Run raxmlHPC-MPI-AVX with 50 MPI processes (-np 50), default 54



Check Job Status (`qstat`) and Cancel a Job (`qdel`)

```
$ 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 canceled or crashed) and is not longer running. Jobs stay in this state for 1h

Q : job is pending, waiting for resources to become available

```
$ qdel 481934
```

```
$ 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

← Stay on list 1 hr



Check Computing Resources Used by a Job

Option 1: `qstat -f JobID` for *running jobs* or *finished jobs in 1 hour*

Option 2: `showjobs JobID` for *finished jobs over 1 hour, but ≤ 7 days*

Option 3: Email notification from *finished jobs (completed, canceled, or crashed),*

if using:

```
#PBS -M jSmith@uga.edu  
#PBS -m ae
```



`qstat -f JobID` for running jobs or finished jobs in 1 hour

```
$ qstat -f 1426778
Job Id: 1426778.pbs.scm
  Job_Name = BS2_GFF_range_methylator5-10.py
  Job_Owner = gent@uga-2f0f976.scm
  resources_used.cput = 76:54:46
  resources_used.energy_used = 0
  resources_used.mem = 44136kb
  resources_used.vmem = 90200kb
  resources_used.walltime = 76:50:04
  job_state = C
  queue = batch
.
Error_Path = uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.e1426778
exec_host = n2/7
Output_Path = uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.o1426778
.
Resource_List.mem = 2gb
Resource_List.nodect = 1
Resource_List.nodes = 1:ppn=1:AMD
Resource_List.walltime = 480:00:00
.
Variable_List = PBS_O_QUEUE=batch,PBS_O_HOME=/home/zhuofei, .....,
                PBS_O_WORKDIR=/home/zhuofei/MPIs,
```




`showjobs JobID` for finished jobs over 1 hour, but ≤ 7 days

```
$ showjobs 1426778
Job Id       : 1426778.pbs.scm
Job Name     : BS2_GFF_range_methylator5-10.py
Output File  : uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.o1426778
Error File   : uga-2f0f976.scm:/home/gent/scripts/BS2_GFF_range_methylator5-10.py.e1426778
Working Directory : /home/gent/scripts
Home Directory : /home/gent
Submit Arguments : BS2_GFF_range_methylator5-10.py.sh
User Name    : gent
Group Name   : krdlab
Queue Name   : batch
Wallclock Limit : 20:00:00:00
Wallclock Duration: 3:04:50:04
CPUtime      : 3:04:54:46
Memory Used  : 43.1Mb
Memory Limit : 2gb
vmem Used    : 88.1Mb
Submit Time  : Mon Mar 27 20:51:01 2017
Start Time   : Tue Mar 28 03:06:41 2017
End Time     : Fri Mar 31 07:58:32 2017
Exit Code    : 0
Master Host  : n2
```



Email notification from finished jobs

PBS Job Id: 700009.pbs.scm
Job Name: testJob
Exec host: n1/4-27
Execution terminated
Exit_status=0
resources_used.cput=00:05:12
resources_used.energy_used=0
resources_used.mem=410984kb
resources_used.vmem=6548516kb
resources_used.walltime=04:00:59
Error_Path: uga-
2f0f976.scm:/home/zhuofei/MPIs/testJob.o700009
Output_Path: uga-
2f0f976.scm:/home/zhuofei/MPIs/testJob.o700009

PBS Job Id: 700097.pbs.scm
Job Name: testJob
Exec host: n5/4-27
Execution terminated
Exit_status=271
resources_used.cput=00:11:22
resources_used.energy_used=0
resources_used.mem=412304kb
resources_used.vmem=6548524kb
resources_used.walltime=05:00:41
Error_Path: uga-
2f0f976.scm:/home/zhuofei/MPIs/testJob.o700097
Output_Path: uga-
2f0f976.scm:/home/zhuofei/MPIs/testJob.o700097



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
.
50 eligible jobs

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

0 blocked jobs
Total jobs: 158
```



Guideline Tips

- Do NOT use Login Node to run jobs → Interactive Node or the queue
 - Do NOT use Login Node upload or download data to/from cluster
 - Do NOT use Login Node to transfer data to the home dir
- } Transfer Node
- NO large memory job running on AMD nodes → HIGHMEM nodes
 - NO small memory job running on HIGHMEM nodes → AMD nodes
 - As a general rule, threads # = cores # requested

Practical Tips

- **Each directory should not have too many files inside!** A rule of thumb would be to try to keep no more than a few tens of thousands of files (<10000 would be even better) in any single directory which is accessed frequently



All files are in ONE single dir! ❌



Files are organized in subdirs! ✅



Practical Tips

- Job name should have a specific computational meaning

Good Examples: `#PBS -N blastn_dataSet1_trail2 ; #PBS -N M-10-1121`

Bad Examples: `#PBS -N job1 ; #PBS -N bowtie ; #PBS -N 20160930`

- The stdout .o file and stderr .e file are to be written into files at the finishing time of a job.

Redirect standard output and error of the application to a file, instead of letting it be written in the stdout .o file and stderr .e file of the job, e.g.:

```
time application >file 2>&1
```

- Monitor job progress from time to time, to catch if a job gets stuck



Useful Links

- GACRC Web: <http://gacrc.uga.edu/>
- GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
- GACRC Support : https://wiki.gacrc.uga.edu/wiki/Getting_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>

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

Telephone Support

EITS HELPDESK: 706-542-3106

MONDAY – THURSDAY: 8AM – 10PM

FRIDAY: 8AM – 6PM

SATURDAY – SUNDAY: 1PM – 7PM



Thank You!