Georgia Advanced Computing Resource Center
UNIVERSITY OF GEORGIA

Introduction to HPC Using Sapelo at GACRC

Georgia Advanced Computing Resource Center
University of Georgia
Suchitra Pakala
pakala@uga.edu
Slides courtesy: Zhuofei Hou

Outline

- GACRC
- Sapelo Cluster at GACRC
- Job Submission Workflow
- Work with Sapelo Cluster
- 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)

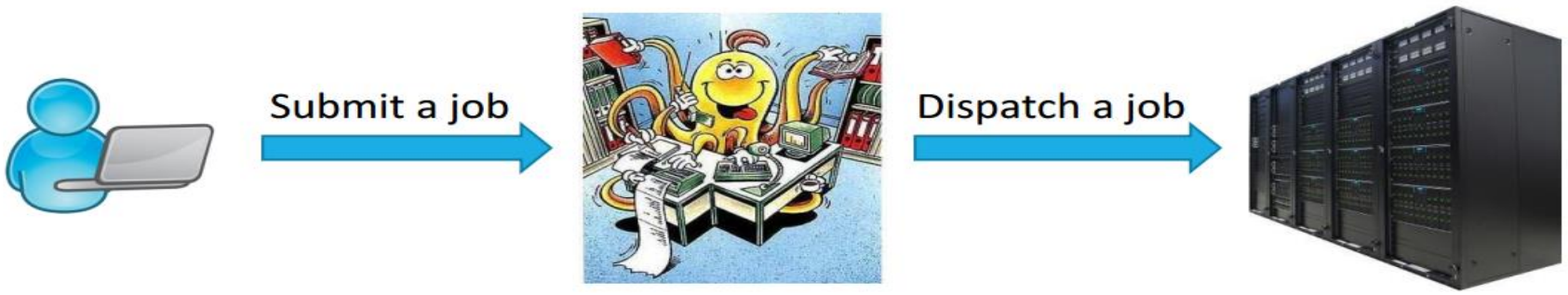
Sapelo cluster at GACRC

- Cluster Diagram
- Cluster Overview
- Computing Resources
- Storage Environment

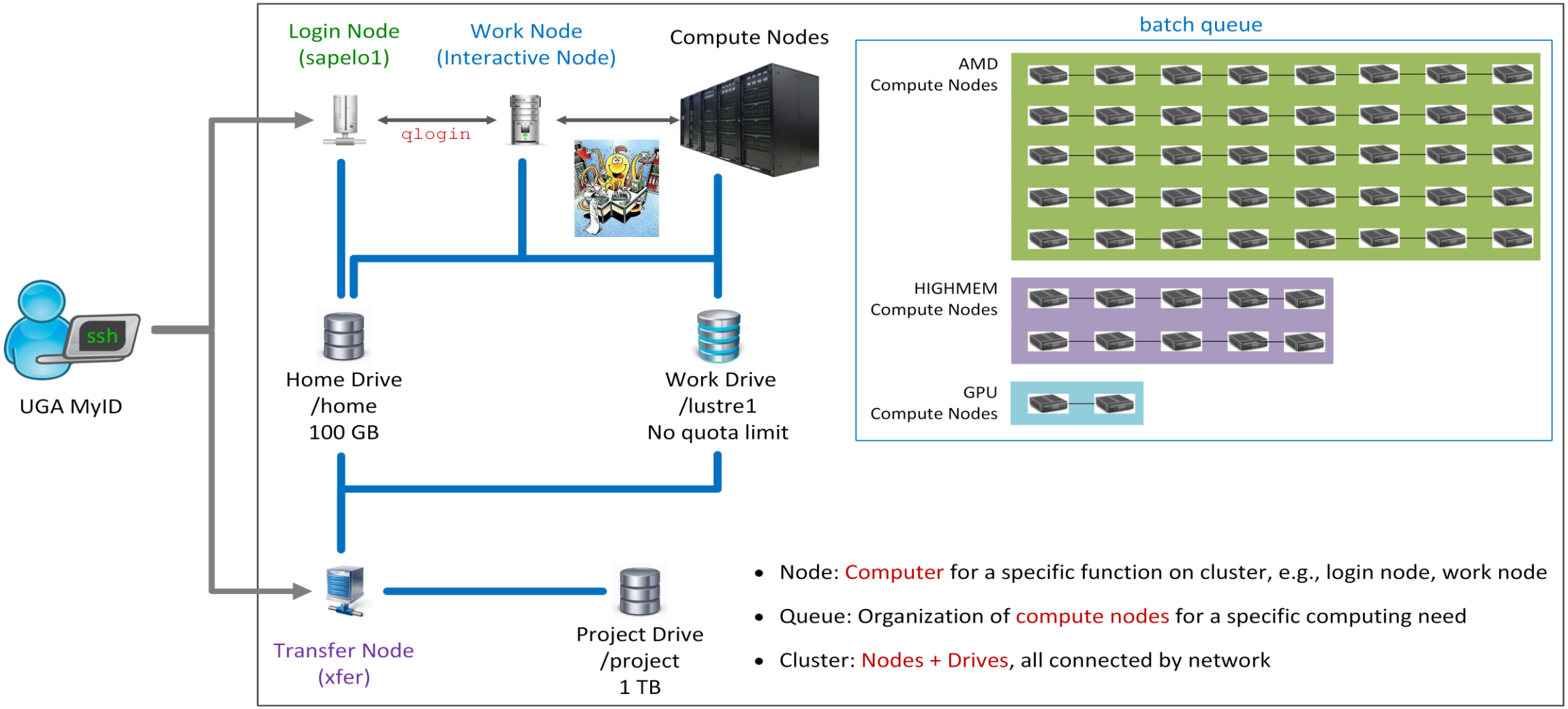
What is a Cluster?

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

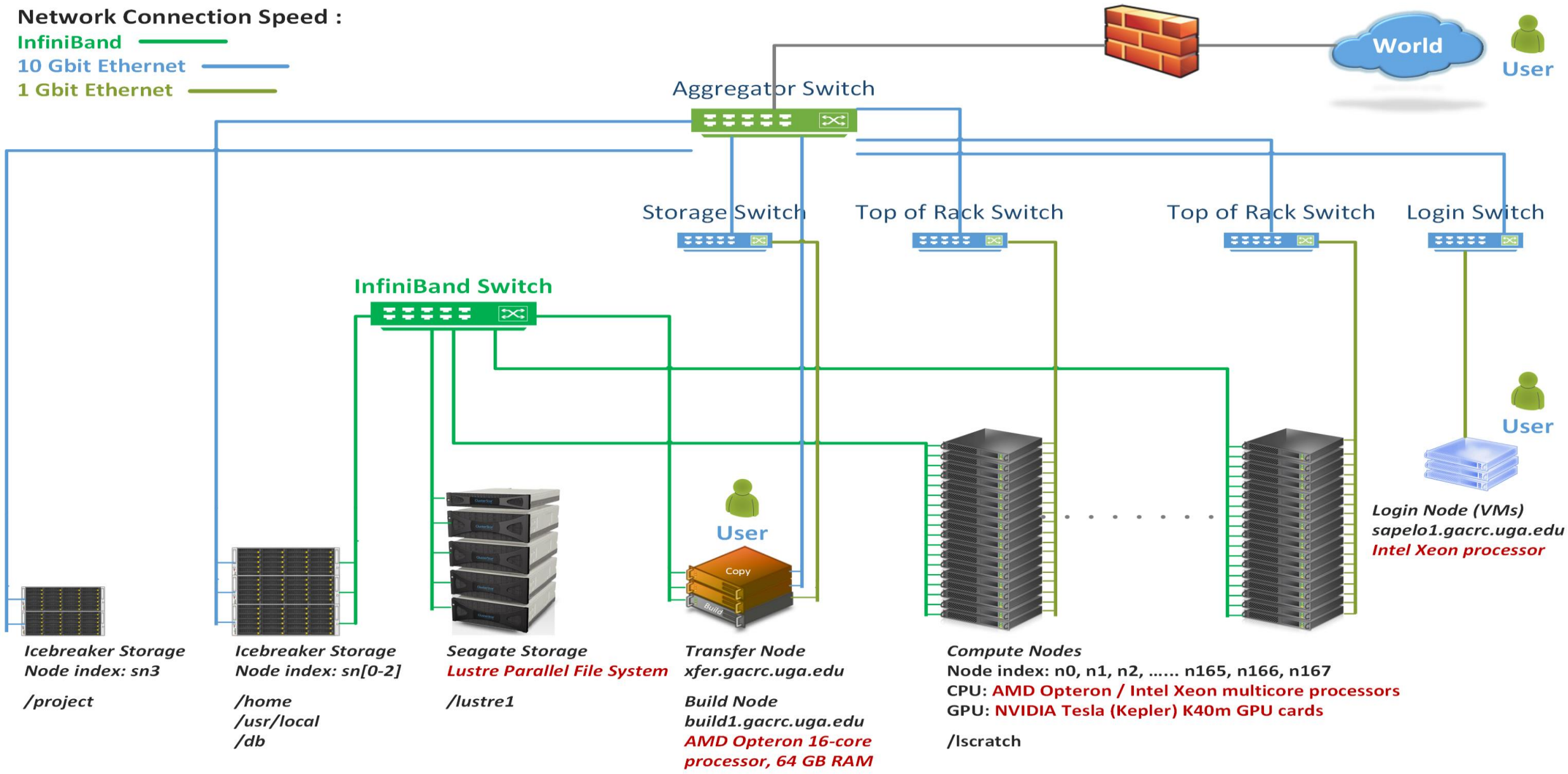


Sapelo Cluster Diagram



- 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



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

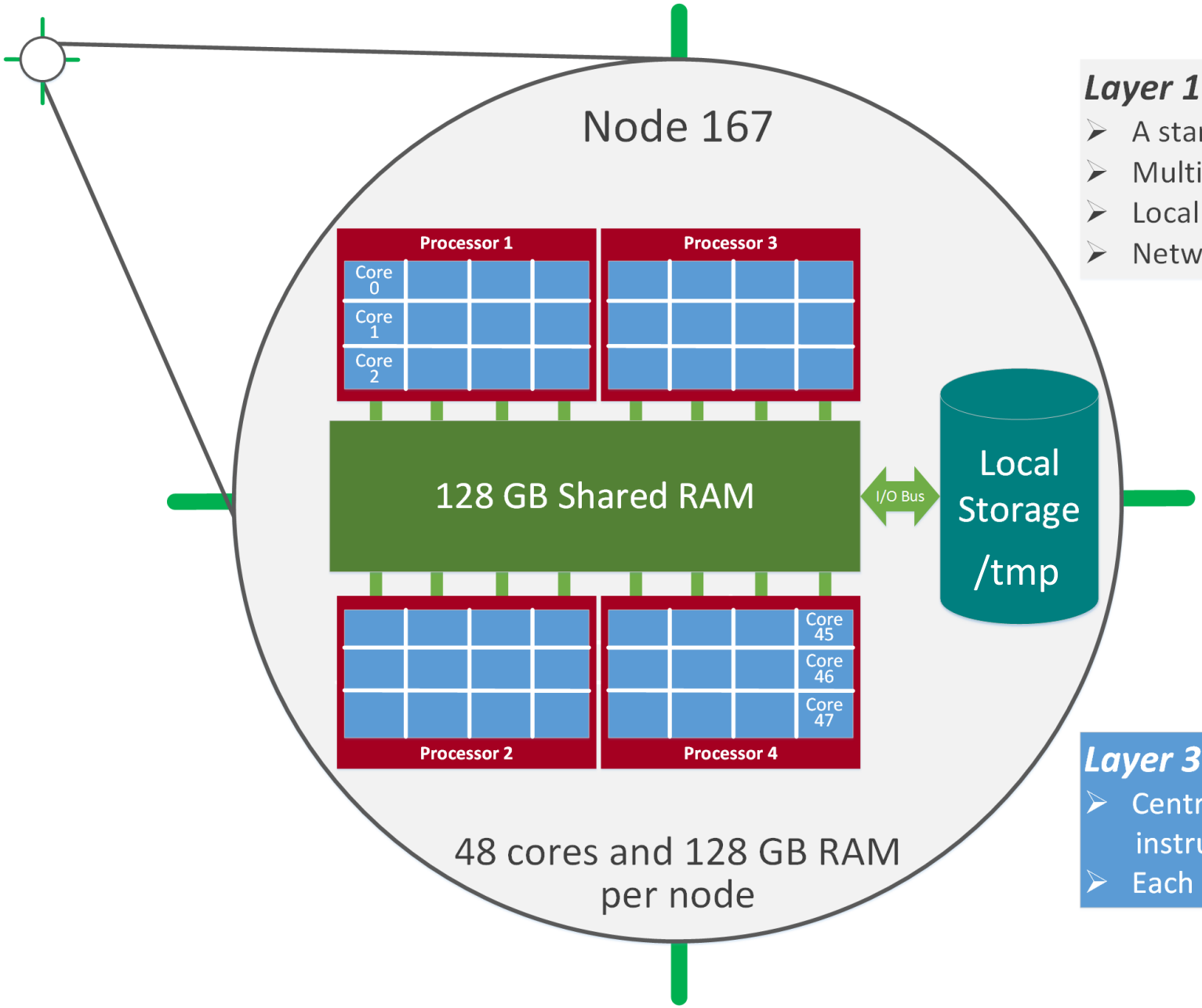
Sapelo Cluster Overview

- 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 Nodes	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					
			1024 (3)		28	Intel Xeon			
	GPU	2	128	126	16		NVIDIA K40m	8	
	abcnode (buy-in)	85	variable	variable	variable	variable	variable	variable	



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

Mainly, there are 3 different storage locations

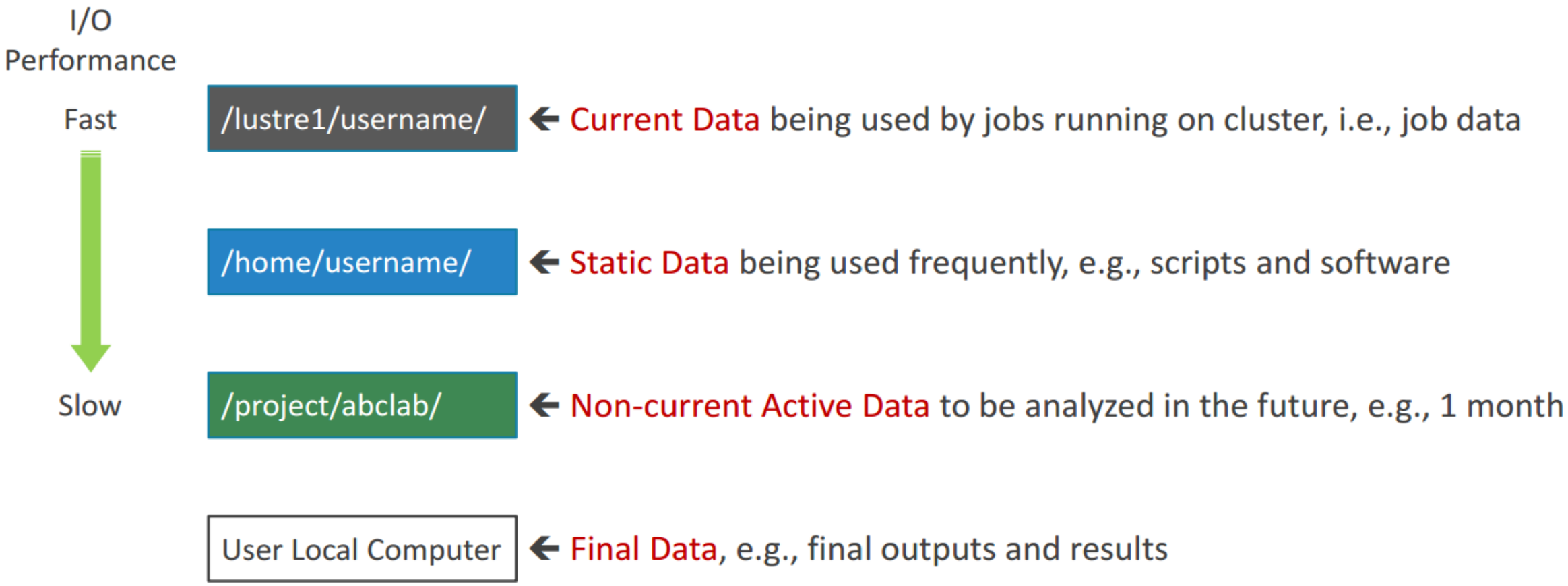
- **Home directory** ➔ */home/username*
  - With a quota of ~100GB
  - Any directory on /home has snapshot backups
- **Global Scratch directory** ➔ /lustre1/username
  - "No quota limit"
  - No snapshot backup
  - User's responsibility to clean up data
- **Group project Storage directory** ➔ /project/abclab/
  - Created for a lab, e.g., abclab
  - Temporary data parking for non-current active data

Storage Environment

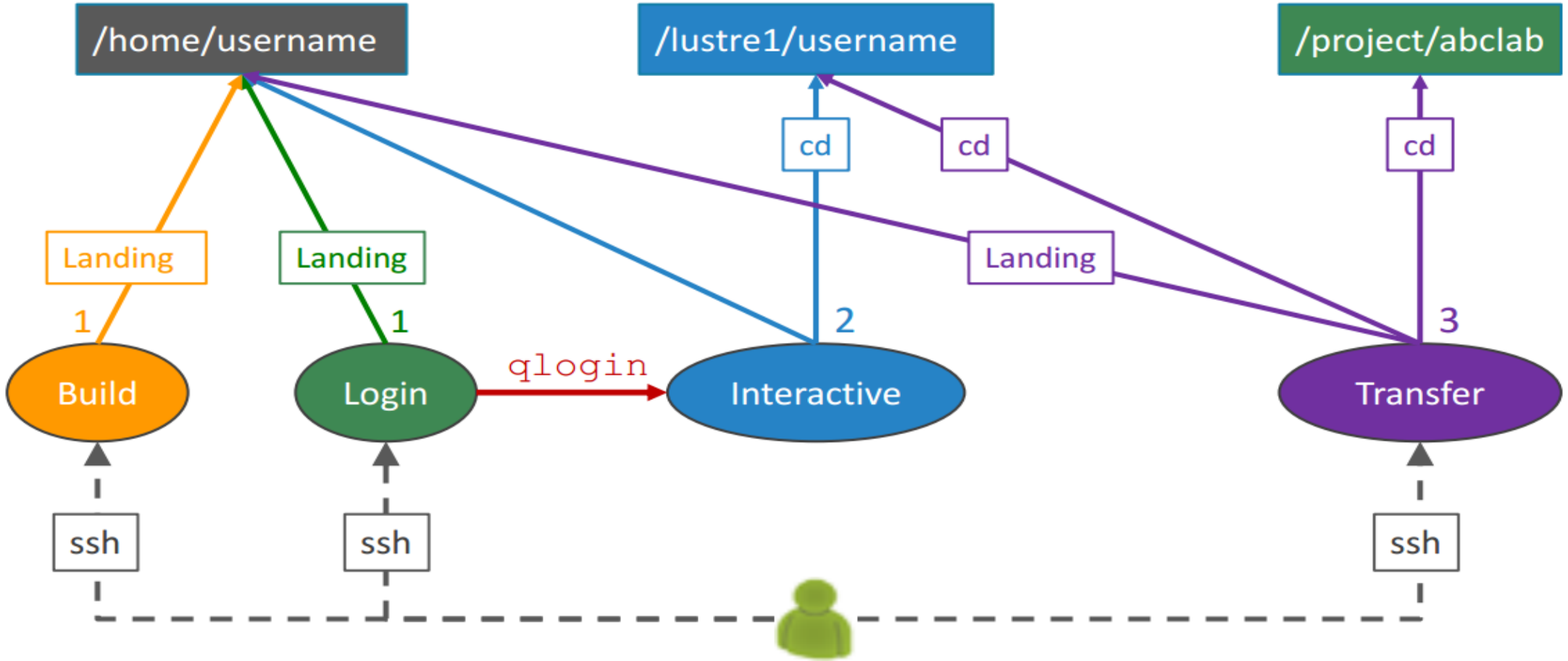
4 Filesystems	Role	Quota	Accessible from	Intended Use	Notes
/home/username/	Home	100GB	sapelo1.gacrc.uga.edu (Login) Interactive nodes (Interactive) xfer.gacrc.uga.edu (Transfer) build1.gacrc.uga.edu (Build) compute nodes (Compute)	Static data being used frequently, e.g., scripts, local software	Snapshots
/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 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) /tmp/lscratch (Local Scratch)	Compute

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

Interactive

```
2. qlogin
```

```
3. cd /lustre1/username
4. mkdir ./workDir
5. cd ./workDir
```

lustre1

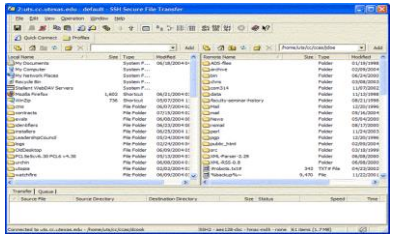
```
7. nano ./sub.sh
```

```
#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
```

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

Job Submission Workflow

1. Log on to Sapelo Login node: `ssh username@sapelo1.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 /lustre1/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`

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

Working With Sapelo

- Start with the Cluster
- Connect and Login
- Transfer Files Using Transfer Node
- Software Packages
- Run Interactive Jobs
- Run Jobs
  - ✓ How to submit a job
  - ✓ Job submission scripts for ***serial***, ***threaded***, and ***MPI*** batch jobs
  - ✓ How to check job status, and cancel a job
  - ✓ How to check memory usage of 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 pakala@sapelo1.gacrc.uga.edu
```

or

```
ssh -x pakala@sapelo1.gacrc.uga.edu
```

([1] `-X` is for *X windows application* running on the cluster with its GUI 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:

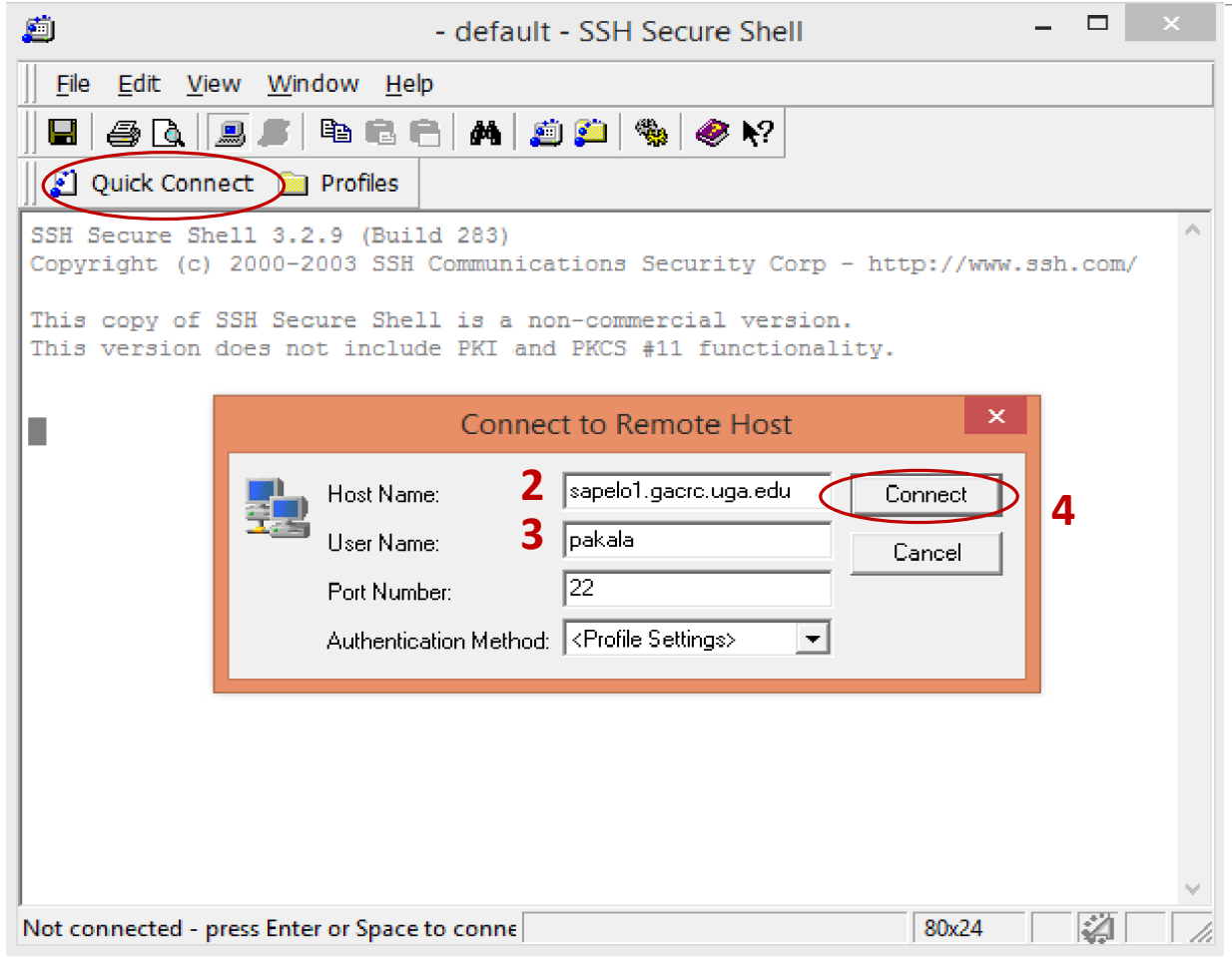
```
pakala@sapelo1.gacrc.uga.edu's password :
```

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

- Logging out: `exit` to leave the system:

```
[pakala@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 Files Using 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 Secure Shell 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 spot)
  - /lustre1/username/ : Sapelo global scratch
  - /panfs/pstor.storage/home/abclab/username/ : zcluster home
  - /escratch4/username/ : zcluster scratch
  - /project/abclab/ : long-term active data storage
- ✓ Most file systems on Transfer are *auto-mounted* upon *the first time full-path access*, e.g., `cd /project/abclab/`

Transfer Files Using Transfer Node xfer.gacrc.uga.edu

User's local  scp/sftp/rsync  Transfer (xfer.gacrc.uga.edu)

- On Linux, Mac or cygwin on Windows : `scp [Source] [Target]`

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

```
scp ./file pakala@xfer.gacrc.uga.edu:/lustre1/pakala/suchi_Sept_14/
```

```
scp -r ./folder/ pakala@xfer.gacrc.uga.edu:/lustre1/pakala/suchi_Sept_14/
```

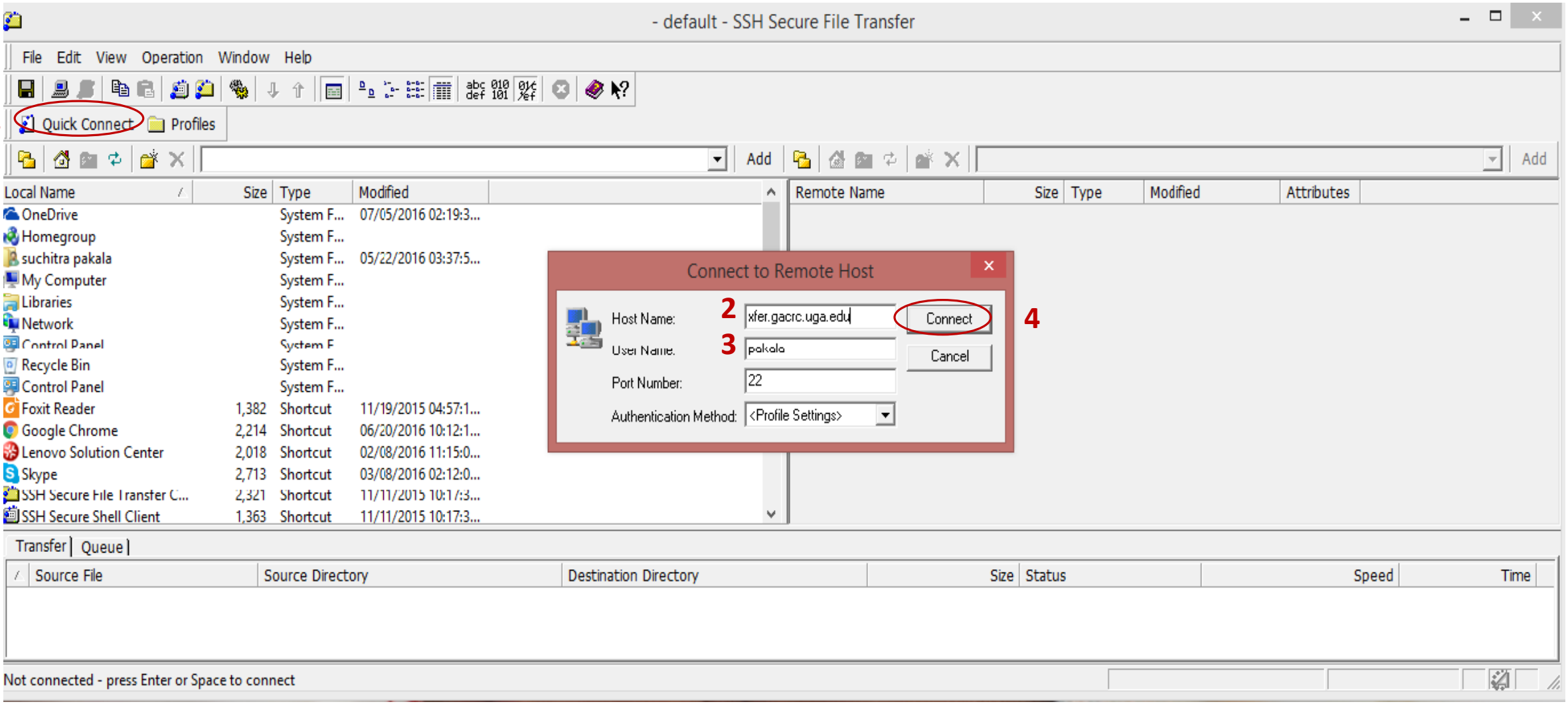
E.g. 2: working on local machine, from Sapelo global scratch ➔ Local

```
scp pakala@xfer.gacrc.uga.edu:/lustre1/pakala/suchi_Sept_14/file ./
```

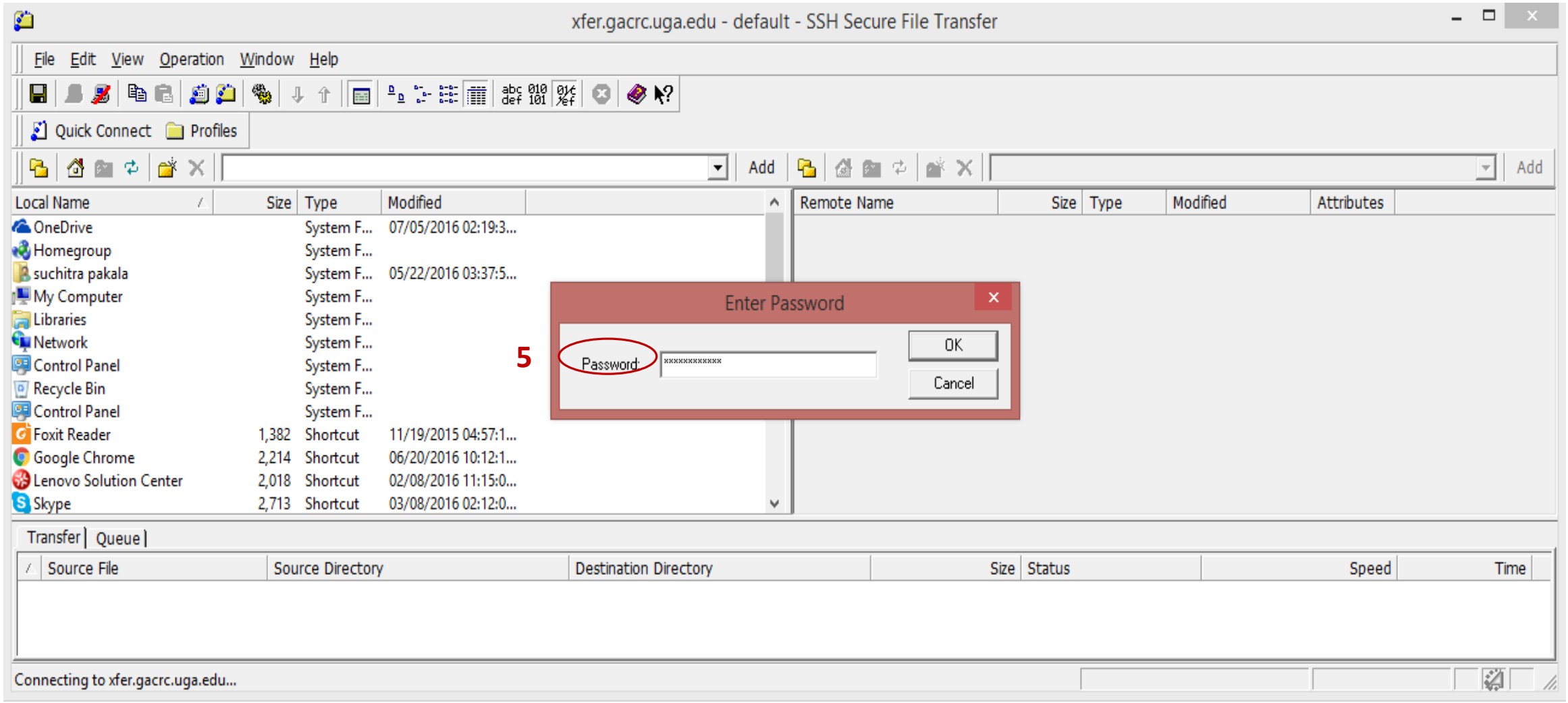
```
scp -r pakala@xfer.gacrc.uga.edu:/lustre1/pakala/suchi_Sept_14/folder/ ./
```

- On Window: SSH Secure Shell File Transfer, FileZilla, WinSCP (next page)

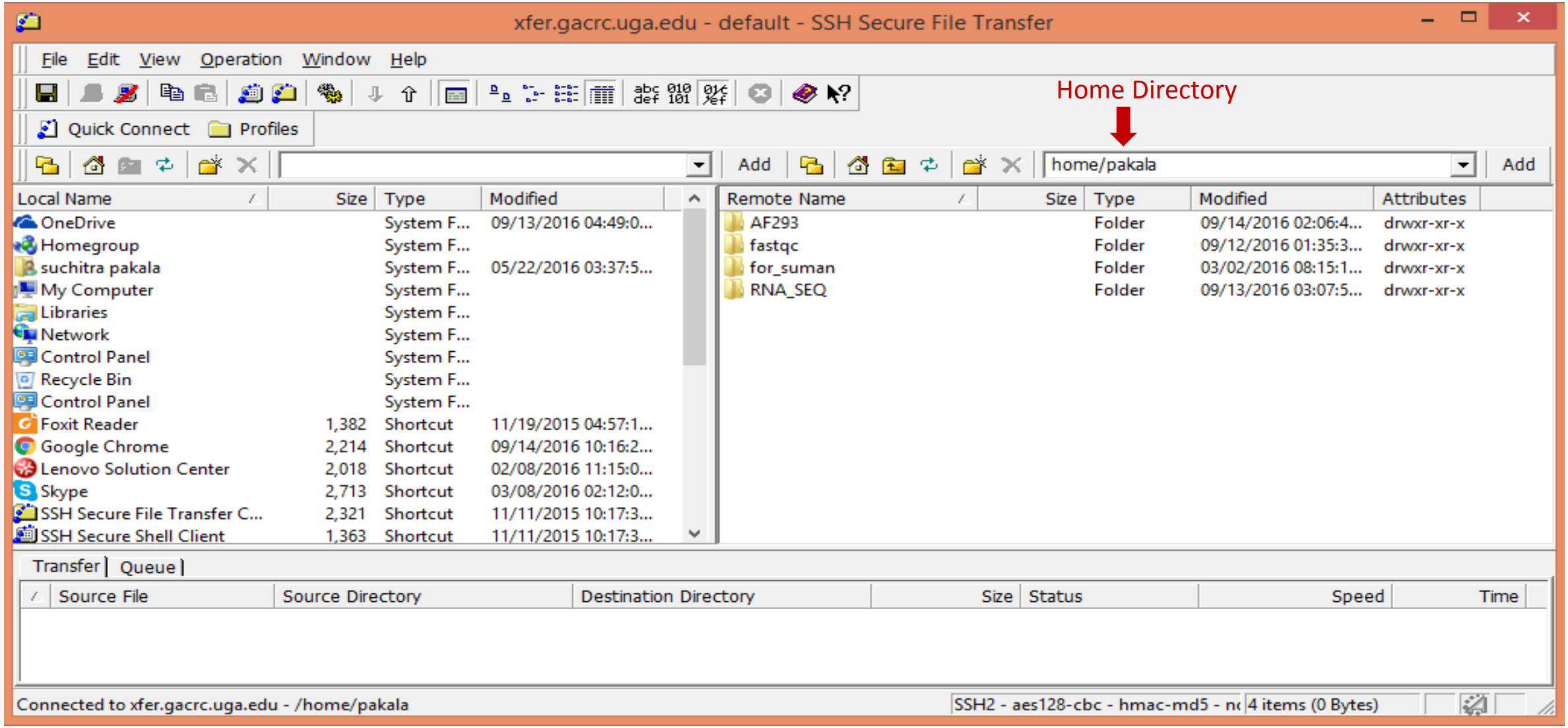
SSH Secure Client: Connecting to Transfer



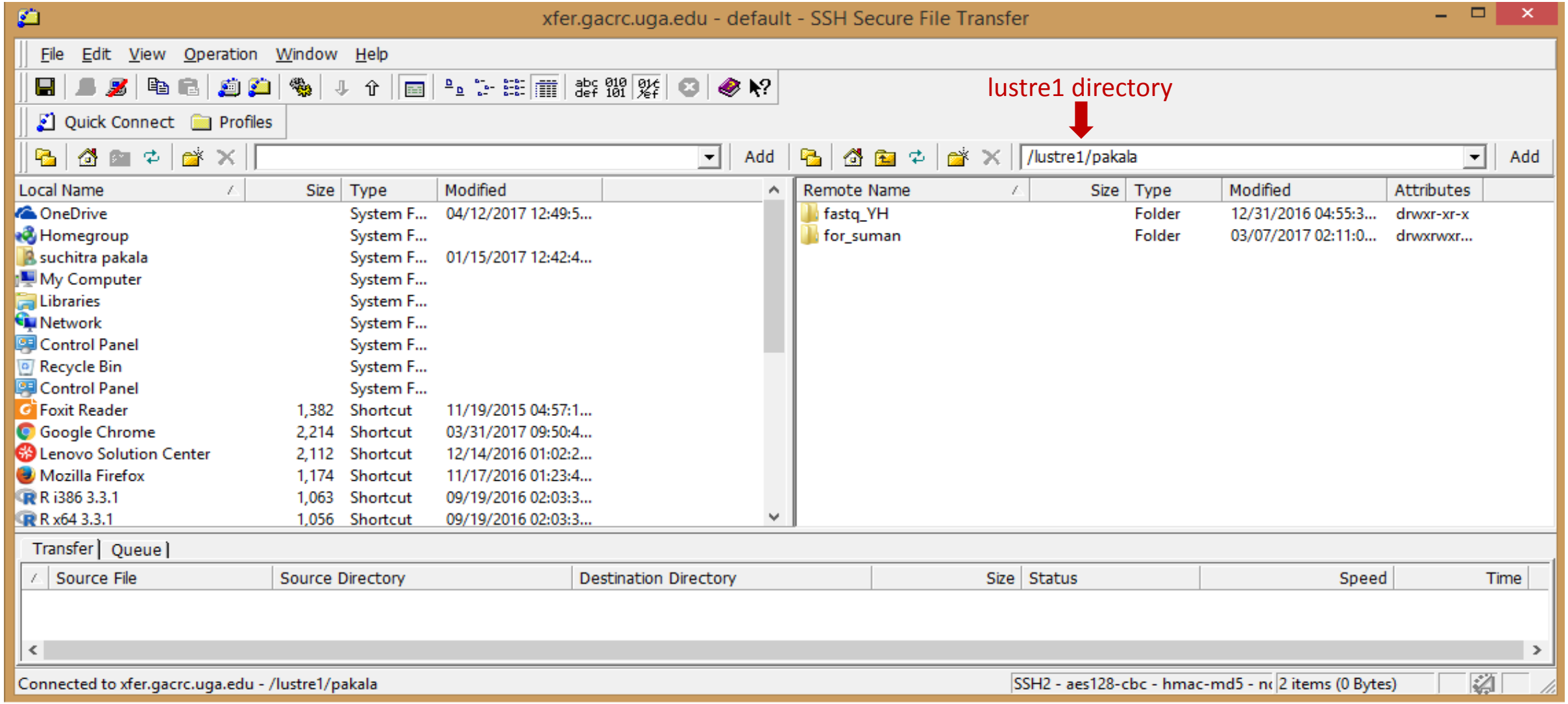
SSH Secure Client: Connecting to Transfer Node



SSH Secure : Connected to Home



SSH Secure : Navigated to /lustre1/pakala



Software Packages

- The cluster uses environment modules to define the various paths for software packages
- Current number of modules installed is ~600 and expanding daily!
- `module avail` to list all available modules (centralized installation)

Software Packages

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

------------------------------------------------------------------------/usr/local/modulefiles -------------------------------------------------------------------------
   Core/StdEnv                          fftw/3.3.4/pgi149-omp183              (D)    mcscanx/0.8                                 python/2.7.8-del
   Data/cache/moduleT.new               find_cns/07212016                            mcscanx-transposed/0.8                      python/2.7.8-ucs4
   Data/cache/moduleT           (D)     flash/1.2.11                                 megahit/latest                              python/2.7.8
   Data/system.txt                      flex/2.6.0/gcc/4.7.4                         megahit/1.0.0                         (D)   python/3.4.3                        (D)
   R/3.1.2                              flex/2.6.0/pgi/14.10                         megan/6.4.3                                 qiime/1.9.1
   R/3.2.1                              freebayes/1.0.1                              meme/4.10.2                                 quast/4.1
   R/3.2.3                              gam-ngs/1.1b                                 meme/4.11.0                           (D)   quast/4.2                           (D)
   R/3.3.1                      (D)     gamess/5Dec2014-1node                        meraculous/2.0.5                            quota-alignment/07142016
   abyss/1.5.2                          gatk/latest                                  mercurial/3.9                               raxml/8.1.20
   abyss/1.9.0                  (D)     gatk/3.3.0                                   metabat/0.26.1                              raxml/8.2.4                         (D)
   allmaps/0.6.6                        gatk/3.4.0                                   metavelvet/latest                           ray/2.3.1-kmer32
   amber/14-at15                        gatk/3.5                                     metavelvet/1.2.02                     (D)   ray/2.3.1-kmer121                   (D)
   amber/14                             gatk/3.6                              (D)    metavelvetsl/latest                         rdptools/2.0.2
   amber/14-20160315            (D)     gblocks/0.91b                                metavelvetsl/1.0                      (D)   reademption/0.3.7
   anaconda/2-4.1.1                     gcc/4.7.4                                    methylpy/08252016                           repeatmasker/4.0.5_perl_threaded
   anaconda/2.2.0-fix                   gcc/4.8.0                                    miinconda2/07152016                         repeatmasker/4.0.5                  (D)
-------------------------------------------------------------------------/usr/local/modulefiles/Core --------------------------------------------------------------------

   StdEnv

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

   (D):  Default Module

Use "module spider" to find all possible modules.

Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

Software Packages

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

```
[pakala@75-104 ~]$ module list
Currently Loaded Modules:
  1) moab/8.1.1   2) StdEnv
```

- `module load` to load the needed modules:

```
[pakala@75-104 ~]$ module load ncbiblast/2.2.26
[pakala@75-104 ~]$ module load bamtools/2.4.0
[pakala@75-104 ~]$ module list
Currently Loaded Modules:
  1) moab/8.1.1   3) ncbiblast/2.2.26
  2) StdEnv       4) bamtools/2.4.0
```

- `module unload` to remove the specific module:

```
[pakala@75-104 ~]$ module unload bamtools/2.4.0
[pakala@75-104 ~]$ module list
Currently Loaded Modules:
  1) moab/8.1.1   2) StdEnv   3) ncbiblast/2.2.26
```

Run Interactive Jobs

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

```
[pakala@75-104 ~]$ qlogin
qsub: waiting for job 1329698.pbs.scm to start
qsub: job 1329698.pbs.scm ready
[pakala@n15 ~]$                ← Now connected to n15, which is an interactive node
[pakala@n15 ~]$ pwd
/home/pakala
[pakala@n15 ~]$ cd /lustre1/pakala
[pakala@n15 pakala]$ module load ncbiblast+/2.2.29
[pakala@n15 pakala]$ module load bowtie/1.1.1
[pakala@n15 pakala]$ module list

Currently Loaded Modules:
  1) moab/8.1.1   3) ncbiblast/2.2.26   5) ncbiblast+/2.2.29
  2) StdEnv       4) bamtools/2.4.0     6) bowtie/1.1.1
```

- 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

```
[pakala@n15 AF293]$ pwd          ← n15: interactive node
/lustre1/pakala/AF293            ← /lustre1/pakala/: global scratch
[pakala@n15 AF293]$
[pakala@n15 AF293]$ qsub sub.sh
1412941.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

Job Submission Script

- 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
```

Job Submission Script

- 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

cd $PBS_O_WORKDIR


module load ncbiblast+/2.2.29

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

➔ Job will be dispatched to run on AMD 256GB node

Job Submission Script

- Example 2: Threaded job script *sub.sh* running NCBI Blast+ with 24 threads

```
#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
cd $PBS_O_WORKDIR

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


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)

Job Submission Script

- Example 3: MPI job script *sub.sh* running RAxML with 94 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=480:00:00          Total cores requested = 2 × 48 = 96
#PBS -l mem=20gb                        We suggest, Number of MPI Processes (94) ≤ 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 94 raxmlHPC-MPI-AVX [options]  > outputfile ➔ Run raxmlHPC-MPI-AVX with 94 MPI processes (–np 94)
```

```
#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=480:00:00
#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
  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)
```

New lines copied from GACRC Wiki

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***

- "**showjobs**" command should be run in a "**qlogin**" session, not on the login node.

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

if using:
```
#PBS -M pakala@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 15:24:32 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

```
[pakala@n15 for_suman]$ showq

active jobs------------------------
JOBID              USERNAME      STATE PROCS   REMAINING            STARTTIME
1412927                gent    Running     4 19:22:58:29  Tue Mar  7 13:35:02
1412931             jl03308    Running     4 19:23:03:11  Tue Mar  7 13:39:44
1412944              pakala    Running     1 19:23:37:33  Tue Mar  7 14:14:06
1412945             jl03308    Running     4 19:23:38:19  Tue Mar  7 14:14:52

370 active jobs       4306 of 7472 processors in use by local jobs (57.63%)
                        114 of 170 nodes active      (67.06%)

eligible jobs----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME
1410619               rzzmh       Idle    12 30:00:00:00  Thu Mar  2 17:02:57
1411044             apjaeger      Idle    42 28:00:00:00  Fri Mar  3 13:30:32
1412702              mesbahi      Idle    24 41:04:00:00  Tue Mar  7 10:08:01
81 eligible jobs

blocked jobs-----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME
1409386            lferreri  BatchHold     1 20:00:00:00  Tue Feb 28 13:16:59
1 blocked job

Total jobs:  452
```

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

How to Submit Tickets to GACRC

- For Installation/Downloading Software:
  - User needs to provide the name, version (or latest), and website
  - Applications need to be compatible with Linux
  - Note – only FREE software will be installed

- For Troubleshooting:
  - List the path of the working directory, path of the script that is producing errors, Job ID, and the command sent to the queue or interactive node
  - No need to attach the script or huge error messages

- For Testing:
  - Please have a sample dataset at your working directory, so that it can be used for debugging

- These steps will help us in responding quickly and efficiently

Useful Links

- GACRC Wiki: https://wiki.gacrc.uga.edu/wiki/Main_Page
- GACRC Software: https://wiki.gacrc.uga.edu/wiki/Software
- GACRC Support: http://gacrc.uga.edu/help/
- GACRC Training: https://wiki.gacrc.uga.edu/wiki/Training

Georgia Advanced Computing Resource Center
4098C Stegeman Coliseum
University of Georgia

Telephone Support
EITS HELPDESK: 706-542-3106
MONDAY – THURSDAY: 8AM – 10PM
FRIDAY: 8AM – 6PM
SATURDAY – SUNDAY: 1PM – 7PM

THANK YOU for your patience