GACRC

Introduction to HPC Using zcluster at GACRC

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

Outline

- What is GACRC?
- What is HPC Concept?
- What is zcluster?
- How does zcluster operate?
- How to work with zcluster?

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

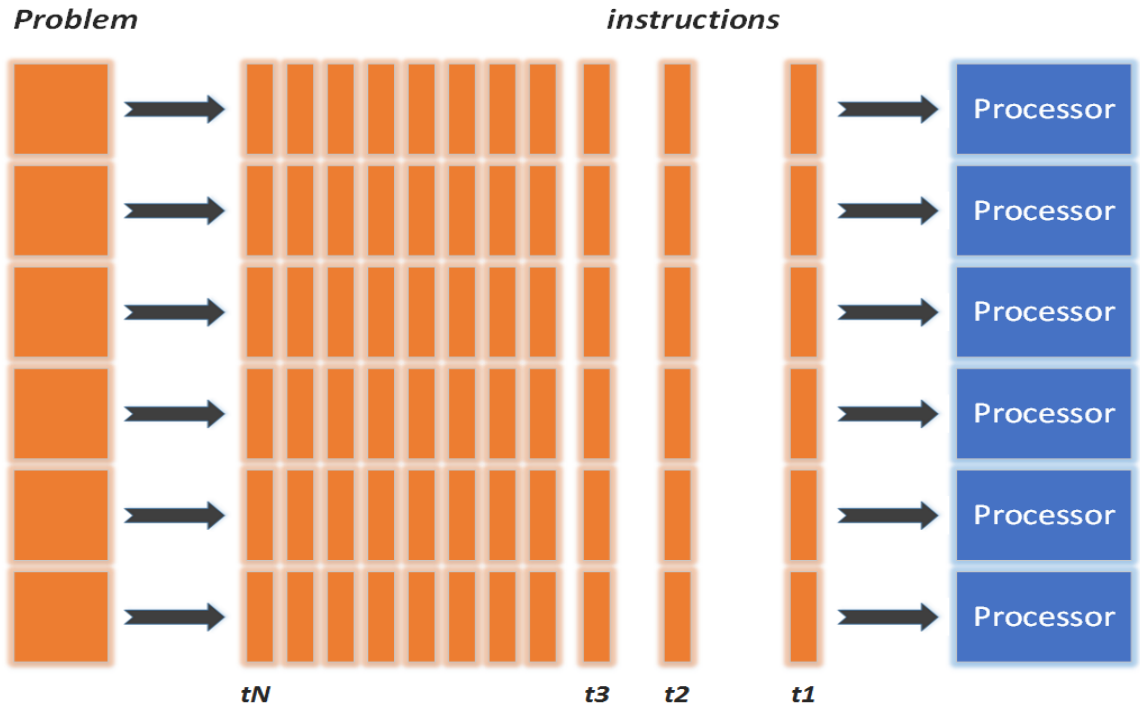
Centers & Institutes	PIs	Users
Center for Applied Isotope Study	1	1
Center for Computational Quantum Chemistry	3	10
Complex Carbohydrate Research Center	6	28
Georgia Genomics Facility	1	5
Institute of Bioinformatics	1	1
Savannah River Ecology Laboratory	3	9
Skidaway Institute of Oceanography	2	2
Center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

Concept of High Performance Computing (HPC)

Serial:

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

Parallel:



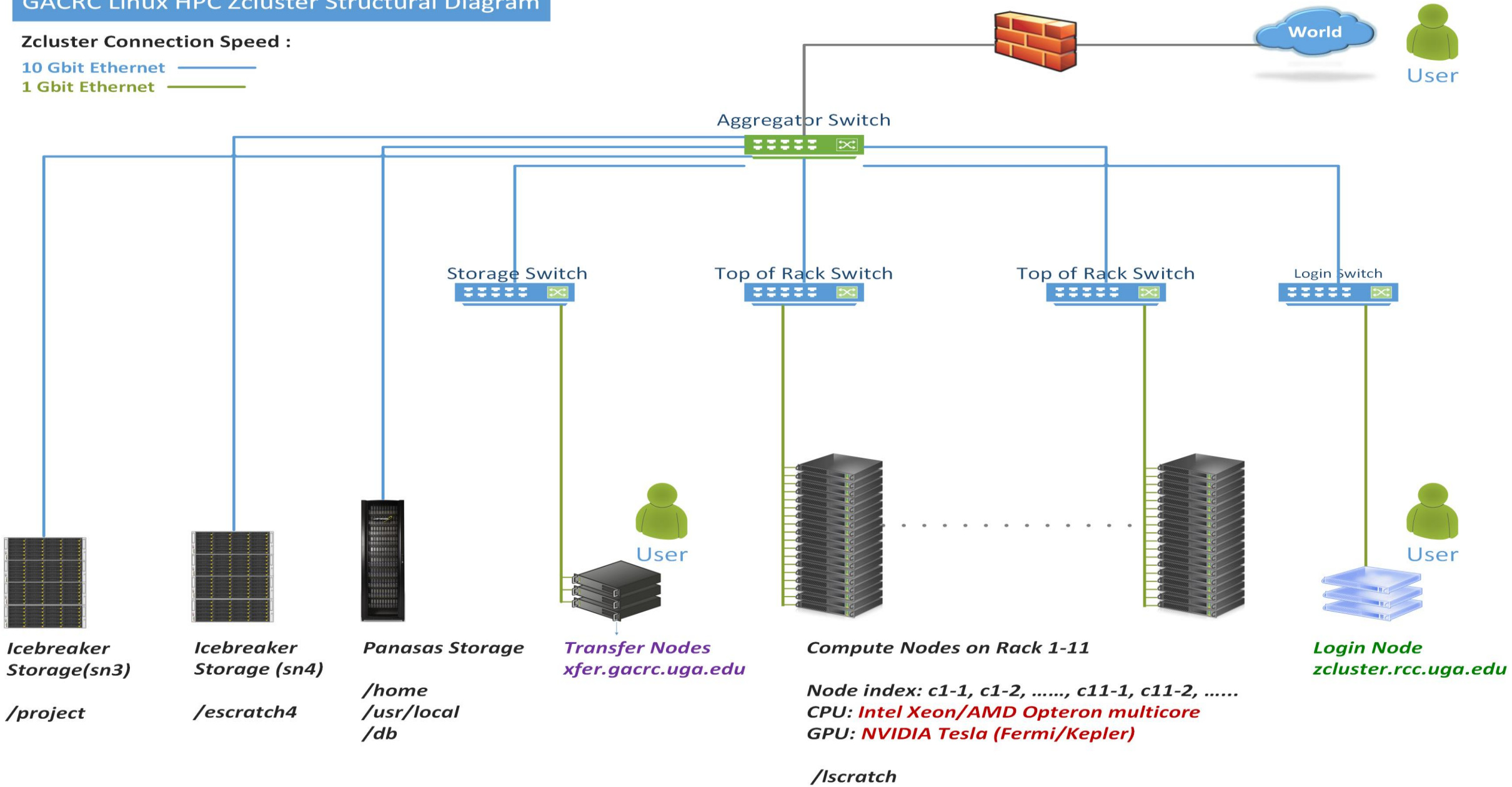
- ✓ Problem broken into *parallel* parts can be solved *concurrently*
- ✓ Instructions executed *simultaneously* on *multiply* processors
- ✓ Synchronization/communication employed
- ✓ Shared-memory multithreaded job or MPI job (Message Passing Interface)

GACRC

What is zcluster?

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

GACRC Linux HPC Zcluster Structural Diagram



What is zcluster – Cluster Overview

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

- OS: 64-bit Red Hat Enterprise Linux 5 (RHEL 5)
- Login Node: zcluster.rcc.uga.edu `qlogin` → Interactive Node: compute-14-7/9
  Transfer Node: xfer.gacrc.uga.edu
- Internodal Communication: 1Gbit network
  compute nodes ⇔ compute nodes
  compute nodes ⇔ storage systems

What is zcluster – Cluster Overview

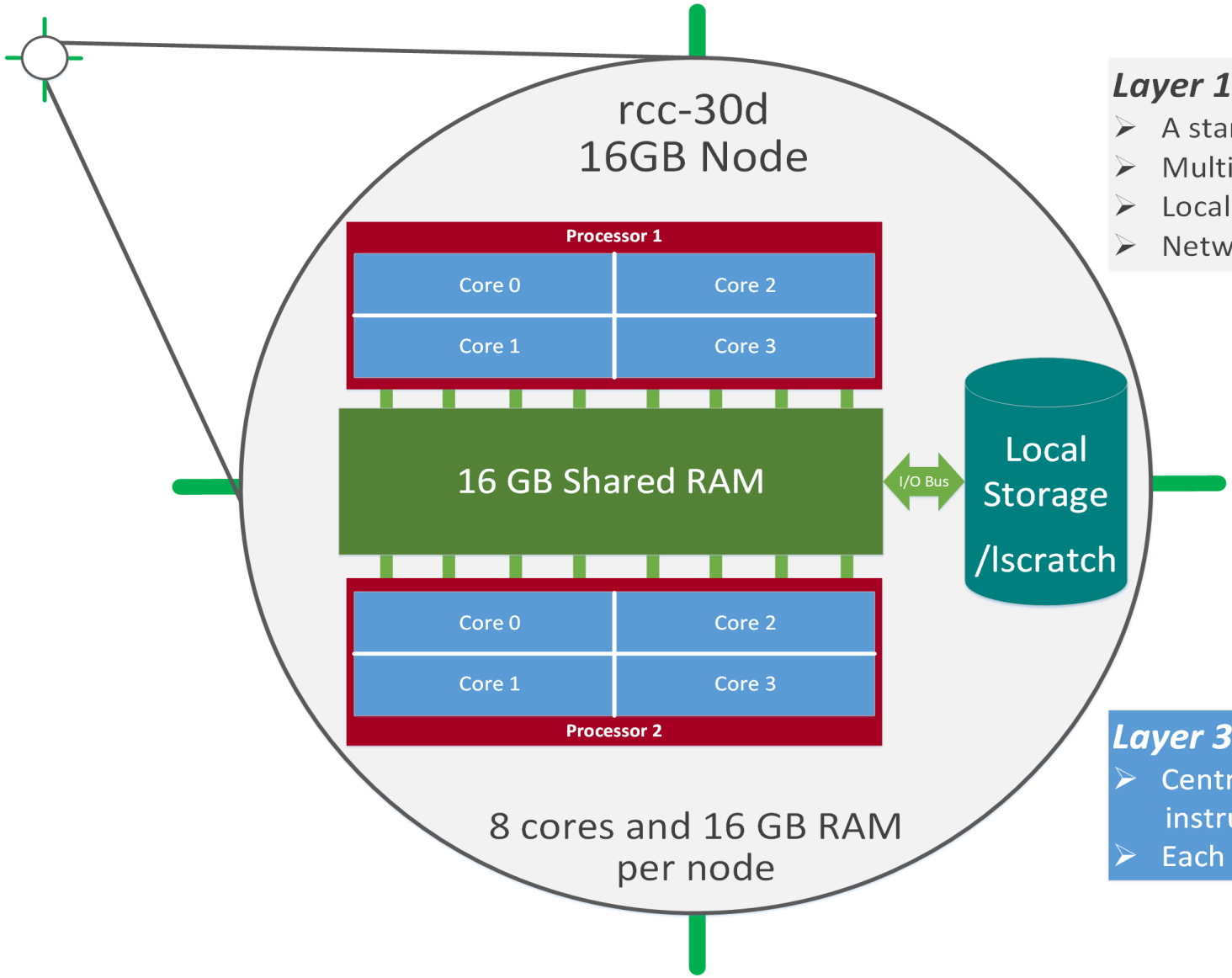
- Batch-queueing System:
  - Jobs can be started (submitted), monitored, and controlled
  - Determine which compute node is the best place to run a job
  - Determine appropriate execution priority for a job to run

- On zcluster:
  - Sun Grid Engine (SGE)
  - Queueing commands: `qsub`, `qstat`, `qdel` `qsj`, `qacct`

What is zcluster – Computing Resources

Queue	Queue Name	Total Nodes	Cores/Node	Max Threads	RAM(GB)/Node	Processor	NVIDIA GPU
Regular	rcc-30d	45	12	6	48	Intel Xeon	N/A
		150	8		16		
High Memory	rcc-m128-30d	1	8	5	128		
		3	8		192		
		10	12		256		
	rcc-m512-30d	2	32	8	512		
Multi Core	rcc-mc-30d	4	32	32	64	AMD Opteron	
Interactive	interq	2	48	N/A	132		
GPU	rcc-sgpu-30d	2	8		48	Intel Xeon	4 Tesla S1070 cards
	rcc-mgpu-30d	2	12		48		9 Tesla (Fermi) M2070 cards
	rcc-kgpu-30d	2	12		96		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

What is zcluster – Storage Environment

- Home directory ➔ */home/groupname/username/*
  - Mounted and visible on all nodes, with a quota of ~100GB
  - Any directory on /home has snapshot backups
    - /home/abclab/jsmith/.snapshot
    - Completely invisible, however, user can “cd” into it and then “ls”:

```
zhuofei@zcluster:~$ ls -a
.               .bash_profile  .emacs.d    .fontconfig        .maple_history  MPIs       scripts  test.sh
..              .bashrc        .ENV_file   .gnuplot_history   .Mathematica    openMPs    serials  .viminfo
.bash_history   downloads      exe         .history           .mc             .profile   sht      .Xauthority
.bash_logout    .emacs         .flexlmrc   .lesshst           .mozilla        Pthreads   .ssh
zhuofei@zcluster:~$ cd .snapshot
zhuofei@zcluster:~/.snapshot$ ls
2015.06.21.00.00.01.weekly  2015.06.27.01.00.01.daily  2015.06.28.01.00.01.daily  2015.06.30.01.00.01.daily
2015.06.26.01.00.01.daily   2015.06.28.00.00.01.weekly 2015.06.29.01.00.01.daily
```

← .snapshot is NOT shown here!

← can “cd” into .snapshot

← then “ls” to list its contents

GACRC

What is zcluster – Storage Environment

- Local scratch ➔ /lscratch/username/
  - On **local disk** of each **compute** node ➔ node-local storage
  - rcc-30d 8-core nodes: ~18GB, rcc-30d 12-core nodes: ~370GB
  - No snapshot backup
  - Usage Suggestion: *If your job writes results to /lscratch, job submission script should move the data to your home or escratch before exit*

- Ephemeral Scratch ➔ /escratch4/zhuofei/zhuofei_Jul_01/
  - Use `make_escratch` from Login to create working subdirectory …/username_mmm_dd/
  - Accessible from Login, Transfer, Interactive, and Compute nodes
  - Each user 4TB quota, No snapshot backup!
  - To be deleted after 37 days

What is zcluster – Storage Environment

Filesystem	Role	Quota	Accessible from	Intended Use	Notes
/home/abclab/username/	Home	100GB	zcluster.rcc.uga.edu (Login) xfer.gacrc.uga.edu (Transfer) Interactive nodes (Interactive) compute nodes (Compute)	Highly static data being used frequently	Snapshots
/escratch4/username/ username_mmm_dd/	Scratch	4TB		Temporarily storing large data being used by jobs	`make_escratch` to create daily; Auto deleted in 37 days!
/lscratch/username/	Local Scratch	18 ~ 370GB	Individual compute node	Jobs with heavy disk I/O	User to clean up
/project/abclab/	Storage	Variable	xfer.gacrc.uga.edu (Transfer)	Long-term data storage	Group sharing possible

Note: 1. /usr/local : Software installation directory
/db : bioinformatics database installation directory
2. use `qlogin` from Login node to log on Interactive node

What is zcluster – Storage Environment

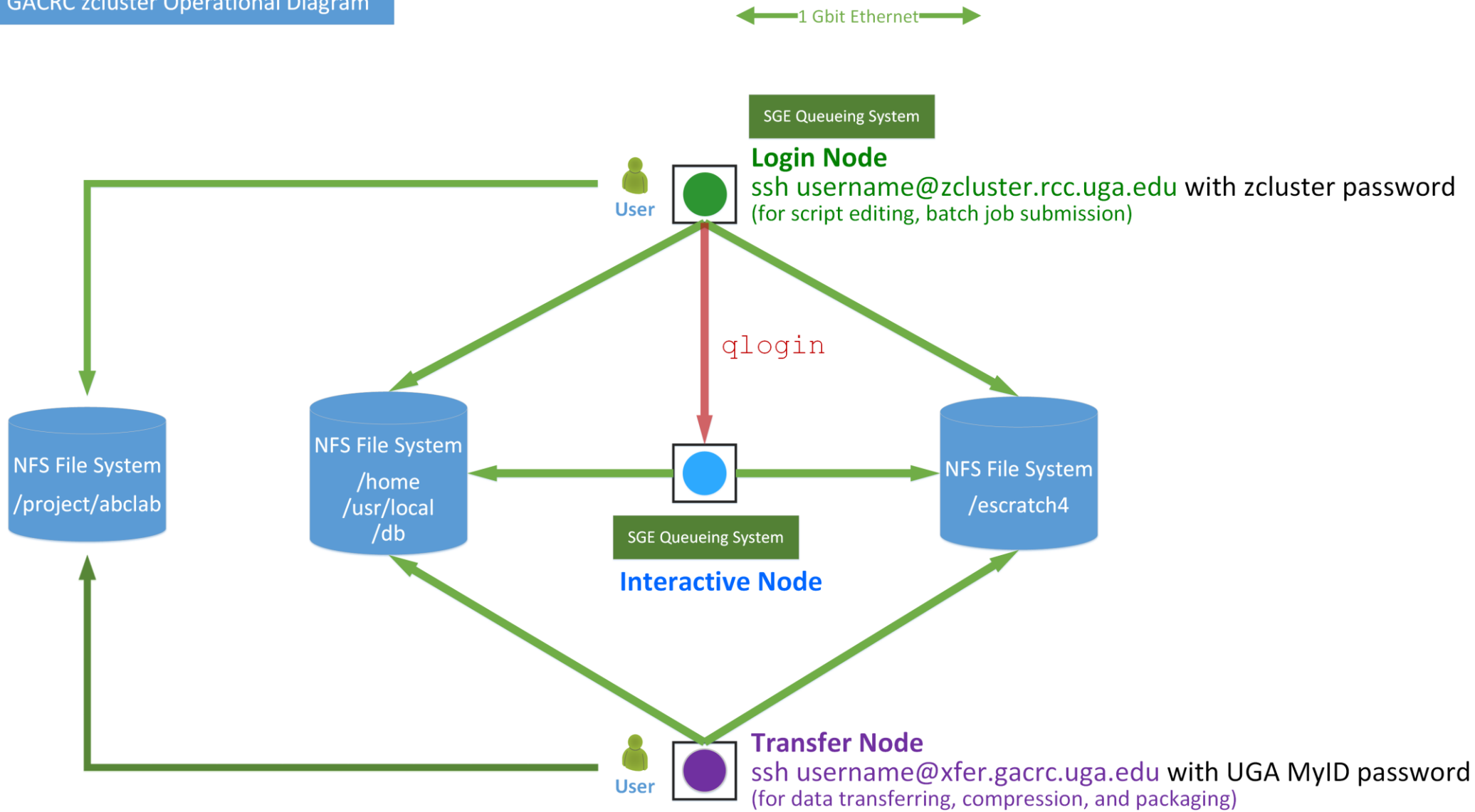
6 Main Functions	On/From Node	Related Filesystem
Login Landing	Login	/home/abclab/username/ (Home)
	Transfer	/home/username/ (Transfer Home) *
Batch Job Submitting	Login or Interactive	/escratch4/username/username_mmm_dd/ (Scratch) (Suggested!) /home/abclab/username/ (Home)
Interactive Job Running	Interactive	/escratch4/username/username_mmm_dd/ (Scratch) /home/abclab/username/ (Home)
Data Transferring, Archiving , and Compressing	Transfer	/escratch4/username/username_mmm_dd/ (Scratch) /panfs/pstor.storage/home/abclab/username/ (Home) *
Job Data Temporarily Storing	Compute	/escratch4/username/username_mmm_dd/ (Scratch) /lscratch/username/ (Local Scratch)
Long-term Active Data Storing	Login or Transfer	/project/abclab/

GACRC

How does zcluster operate?

Next Page

GACRC zcluster Operational Diagram



Workflow 1 – Home as Job Working Space

1. Linux/Mac user:
`ssh userID@zcluster.rcc.uga.edu`

Windows user:

Login

2. `mkdir ./workDir`

3. `cd ./workDir`

Home

5. `nano ./sub.sh`

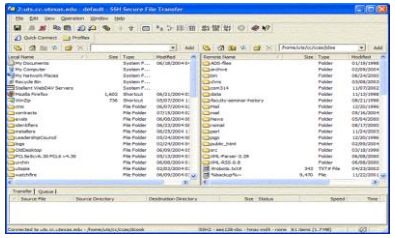
```
#!/bin/bash
cd `pwd`
/usr/local/samtools/latest/samtools...
```

6.

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

4. Linux/Mac user:
`scp input userID@copy.rcc.uga.edu:~/workDir`

Windows user:



Note: inputDate could be files or data folder (`scp -r`)

Other qsub options:
`-l mem_total=20g` : use 48GB rcc-30d nodes
`-pe thread 4` : request 4 cores for 4 threads

GACRC

Workflow 1 – Home as Job Working Space

1. Log on to zcluster Login node: `ssh userID@zcluster.rcc.uga.edu`
2. Create a working subdirectory in home: `mkdir ./workDir`
3. Change directory to `workDir`: `cd ./workDir`
4. Transfer data to `workDir` using `scp` or SSH Client File Transfer (with `tar` or `gzip`)
5. Make a zcluster job submission script: `nano ./sub.sh`
6. Submit job: `qsub -q rcc-30d ./sub.sh`

Useful `qsub` options: `-l mem_total=20g` : use 48GB high-RAM rcc-30d nodes

`-pe thread 4` : request 4 cores for 4 threads, max 6 is suggested!

Workflow 2 – Global Scratch as Job Working Space

1. Linux/Mac user:
`ssh userID@zcluster.rcc.uga.edu`

Windows user:

Login

2. `make_escratch`

3. `cd /escratch4/userID/userID_Jul_01`

escratch4

5. `nano ./sub.sh`

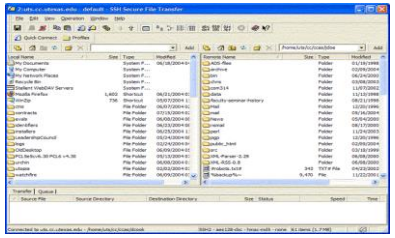
```
#!/bin/bash
cd `pwd`
/usr/local/samtools/latest/samtools...
```

6.

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

4. Linux/Mac user:
`scp input userID@copy.rcc.uga.edu:/escratch4/userID/userID_Jul_01`

Windows user:



Other qsub options:
`-l mem_total=20g : use 48GB rcc-30d nodes`
`-pe thread 4 : request 4 cores for 4 threads`

Note: inputDate could be files or data folder (`scp -r`)

Workflow 2 – Global Scratch as Job Working Space

1. Log on to zcluster Login node: `ssh userID@zcluster.rcc.uga.edu`
2. Create a working subdirectory on global scratch: `make_escratch`
3. Change directory to `userID_Jul_01`: `cd /escratch4/userID/userID_Jul_01`
4. Transfer data to `userID_Jul_01` using `scp` or SSH Client File Transfer (with `tar` or `gzip`)
5. Make a zcluster job submission script: `nano ./sub.sh`
6. Submit job: `qsub -q rcc-30d ./sub.sh`

Useful `qsub` options: `-l mem_total=20g` : use 48GB high-RAM rcc-30d nodes

`-pe thread 4` : request 4 cores for 4 threads, max 6 is suggested!

How to work with zcluster?

Before we start:

- To get zcluster to be your best HPC buddy

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

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

How to work with zcluster?

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!

How to work with zcluster?

- You are not alone on cluster… Each user is sharing finite computing resources, e.g., CPU cycles, RAM, disk storage, network bandwidth, with other researchers:

What you do may affect others on the cluster

- Do NOT run jobs on login node ➔ use the queues or the interactive nodes
- Do NOT use login node to move data into/out of cluster ➔ use Transfer xfer.gacrc.uga.edu
- NO multi-threaded job running with only 1 core requested ➔ threads # = cores # requested
- NO large memory job running on regular nodes ➔ HIGHMEM queue
- NO long job running on interactive node ➔ 12 hours
- NO small memory job running on large memory nodes ➔ Saving memory for others

How to work with zcluster?

- Start with zcluster
- Connect and Login
- Transfer Files Using Transfer Node
- Software Installed
- Run Interactive Jobs
- Submit Batch Jobs
  - ✓ How to submit ***serial***, ***threaded***, and ***MPI*** batch jobs; useful qsub options
  - ✓ How to check job status, cancel a job
  - ✓ How to check memory usage of a job

Start with zcluster

- You need a User Account : username@zcluster.rcc.uga.edu
- Procedure: https://wiki.gacrc.uga.edu/wiki/User_Accounts
  - A UGA faculty member (PI) may register a computing lab: http://help.gacrc.uga.edu/labAcct.php
  - The PI of a computing lab may request user accounts for group members: http://help.gacrc.uga.edu/userAcct.php
- User receives a welcome email once the account is ready
- User uses `passwd` to change initial temporary password to a permanent one upon the first time of login

Connect and Login

- On Linux/Mac: use Terminal utility and `ssh` to your account:

```
ssh zhuofei@zcluster.rcc.uga.edu
```

or

```
ssh -X zhuofei@zcluster.rcc.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 zcluster password:

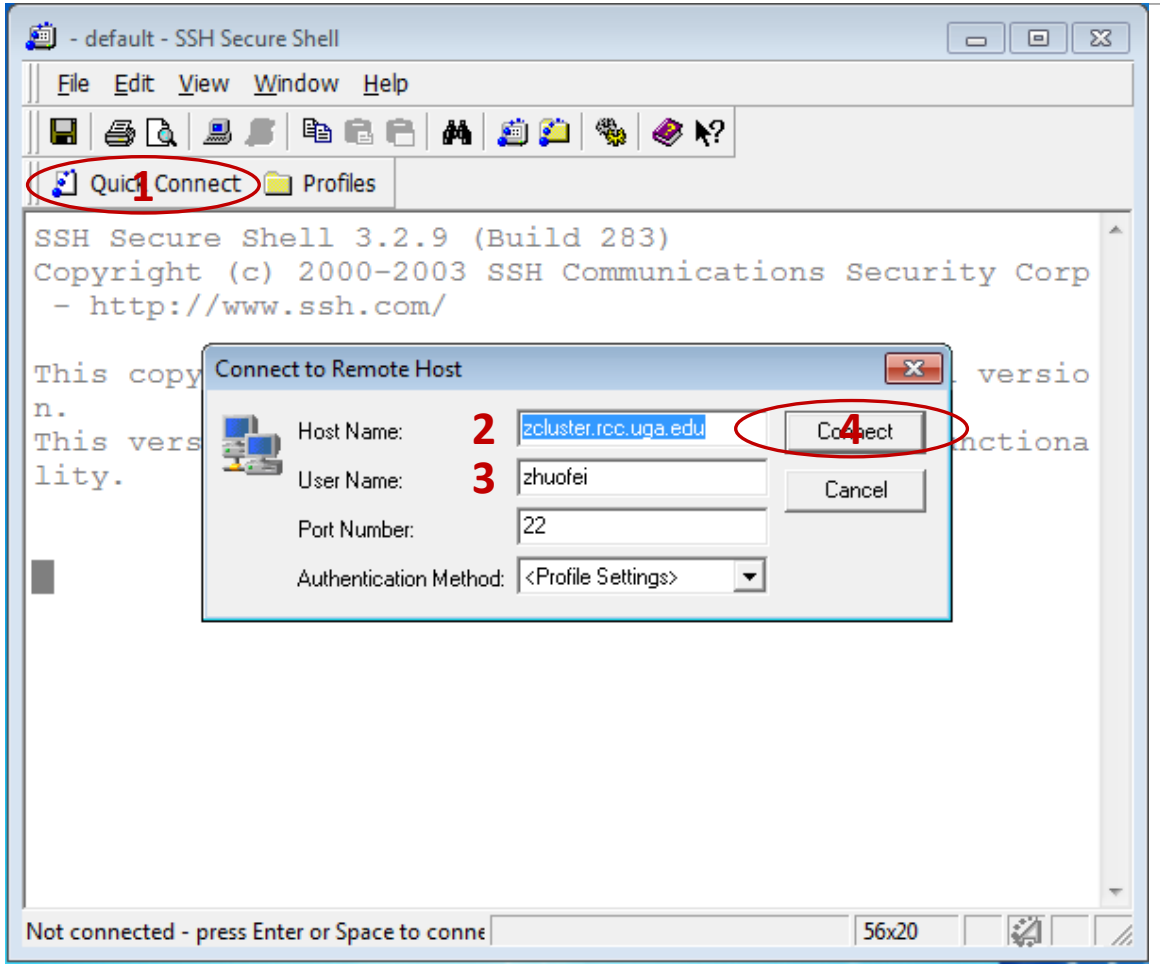
```
zhuofei@zcluster.rcc.uga.edu's password:
```

([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@zcluster:~$ 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`
- ✓ Move data into/out of zcluster (`scp`, `sftp`, `rsync`, SSH Secure Shell File Transfer, FileZilla)
- ✓ Compress or package data on zcluster (`tar`, `gzip`)
- ✓ Transfer data between zcluster and Sapelo (`cp`, `mv`)
- ✓ Filesystems you can access:
  - /home/username/ : Transfer home (landing spot)
  - /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  

 Transfer (xfer.gacrc.uga.edu)

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

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

```
scp ./file zhuofei@xfer.gacrc.uga.edu:/escratch4/zhuofei/zhuofei_Jul_1/
```

```
scp -r ./folder/ zhuofei@xfer.gacrc.uga.edu:/escratch4/zhuofei/zhuofei_Jul_1/
```

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

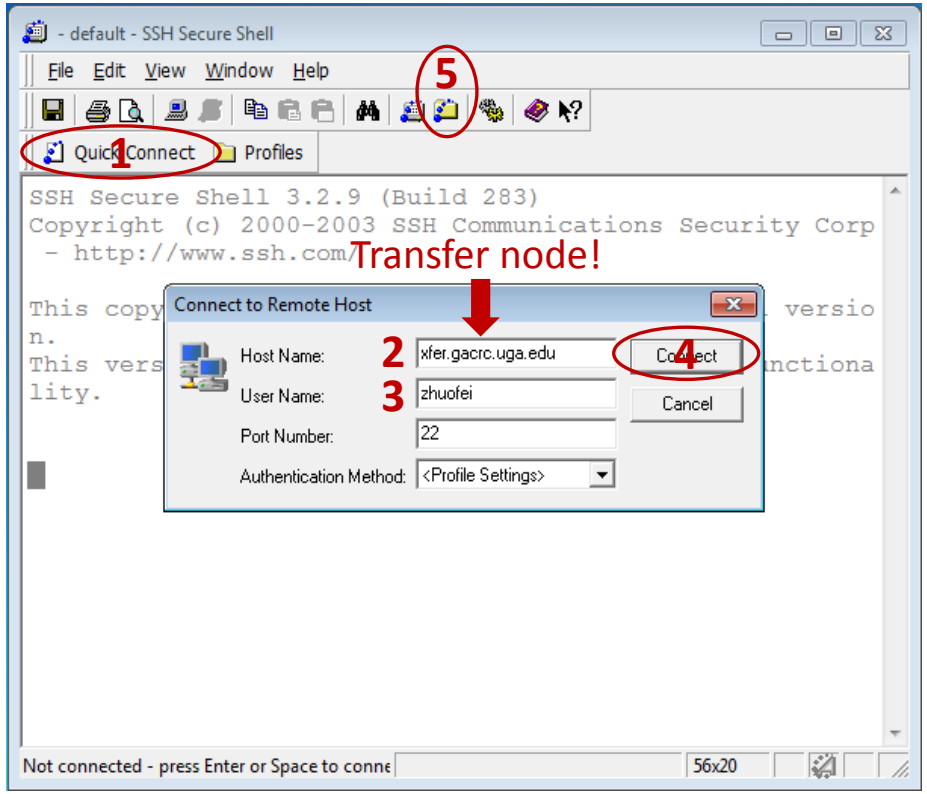
```
scp zhuofei@xfer.gacrc.uga.edu:/escratch4/zhuofei/zhuofei_Jul_1/file ./
```

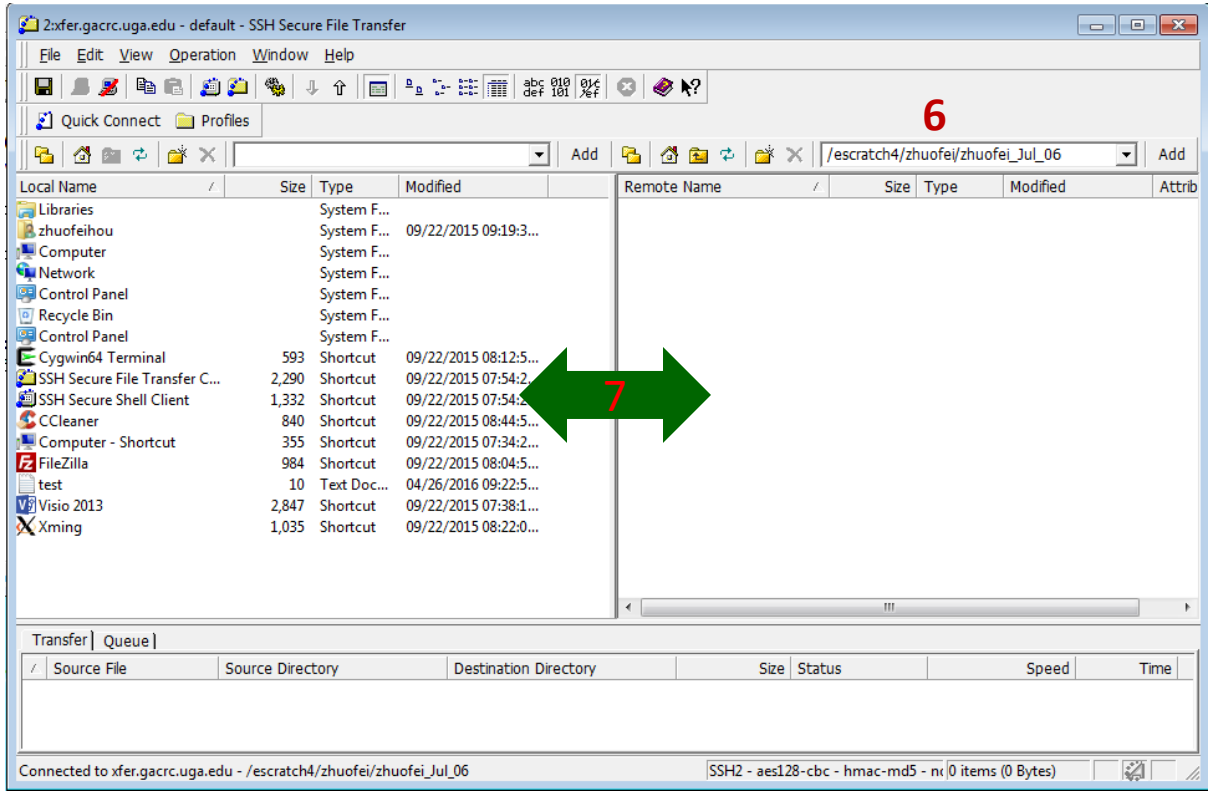
```
scp -r zhuofei@xfer.gacrc.uga.edu:/escratch4/zhuofei/zhuofei_Jul_1/folder/ ./
```

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

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

User's local  SSH client  Transfer (xfer.gacrc.uga.edu)





Software Installed

- 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

Run Interactive Jobs

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

```
zhuofei@zcluster:~$ qlogin
Your job 1391816 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 1391816 has been successfully scheduled.
...
compute-14-7.local$
```

← Now I am on compute-14-7, which is an interactive node

- Current maximum runtime is 12 hours
- When you are done, remember to `exit` the session!
- Detailed information, like interactive parallel job? Go to:

  https://wiki.gacrc.uga.edu/wiki/Running Jobs on zcluster

Submit Batch Jobs

- Components you need to submit a batch job:
  - Software already installed on zcluster
  - Job submission script to run the software, and
    - ✓ Specify working directory
    - ✓ Export environment variables, e.g.,
      PATH (searching path for executables)
      LD_LIBRARY_PATH (searching paths for shared libraries)
- Common commands you need:
  - qsub with specifying queue name, cores to be requested
  - qstat, qdel
  - qsj, qacct

Submit Batch *Serial* Job

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

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

cd /escratch4/zhuofei/zhuofei_Feb_1                          ➔ Specify and enter (cd) working directory (/escratch4/zhuofei/zhuofei_Feb_1)

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 *st.sh* to the queue:

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

Submit a job

job submission script

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

OR

```
$ qsub -q rcc-30d -l mem_total=20g  st.sh
```

to the queue rcc-30d
with 48GB RAM/Node

Submit Batch *Threaded* Job

- Step 1: Create a job submission script *blastn.sh* running NCBI Blast +:

```
#!/bin/bash

cd /escratch4/zhuofei/zhuofei_Feb_1

time /usr/local/ncbiblast+/latest/bin/blastn -num_threads 4 [options]
```

➔ Run blastn with 4 threads

- Step 2: Submit *blastn.sh* to the queue:

```
$ qsub -q rcc-30d -l mem_total=20g -pe thread 4 blastn.sh
```

to the queue rcc-30d with 48GB RAM/Node

4 cores requested

Number of Threads = Number of Cores Requested

Note:
Please use the rcc-mc-30d queue, If using threads more than 8!

Submit Batch *MPI* Job

- Step 1: Create a job submission script *raxml.sh* running RAxML:

```
#!/bin/bash

cd /escratch4/zhuofei/zhuofei_Dec_25

export MPIRUN=/usr/local/mpich2/1.4.1p1/gcc 4.5.3/bin/mpirun  ➔ Define and export environment variable (MPIRUN)

$MPIRUN -np $NSLOTS /usr/local/raxml/latest/raxmlHPC-MPI-SSE3 [options]  ➔ Run RAxML with 20 MPI processes (-np $NSLOTS )
```

- Step 2: Submit *raxml.sh* to the queue:

```
$ qsub -q rcc-30d -pe mpi 20  ramxl.sh
```

20 cores requested,
$NSLOTS will be assigned to 20 automatically, before the job submission script is interpreted

GACRC

Useful qsub Command Options

qsub options	Explanation
-q queue_name	Defines the queue to run your job, e.g. -q rcc-30d
-l mem_total=20g	Request a compute node with at least 20GB total physical RAM installed
-pe thread 4	Request 4 cores for a threaded job with 4 threads; maximum of 6 on rcc-30d
-pe mpi 20	Request 20 cores for a MPI job with 20 MPI processes, maximum of 75 on rcc-30d
-cwd	Run in current working directory
-M MyID@uga.edu	Defines the email address to send an email notification
-m ea	Send an email notification when job ends or aborts
-N name	Defines the name of a job

Check and Cancel Jobs

- To check the status of your jobs: qstat

```
qstat                 ➔ shows your job in the pool
qstat –u “*”          ➔ shows all the jobs in the pool
qstat –j 12345        ➔ shows detailed information, e.g., maxvmem, about the job with JOBID 12345
```

```
$ qstat
job-ID     prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
9707321 0.50766 sub1.sh    jsmith       r     01/28/2016 13:39:23 rcc-30d@compute-7-12.local  1
9707322 0.50383 sub2.sh    jsmith       Eqw   01/28/2016 13:39:23 rcc-30d@compute-7-12.local  1
9707323 0.00000 sub3.sh    jsmith       qw    01/28/2016 13:39:28                             1
```

- To cancel your job with a JobID: qdel

```
$ qdel 9707322
job-ID     prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
9707321 0.50766 sub1.sh    jsmith       r     01/28/2016 13:39:23 rcc-30d@compute-7-12.local  1
9707323 0.00000 sub3.sh    jsmith       qw    01/28/2016 13:39:28                             1
```

Check Memory Usage

- For a running job: qsj

```
$ qsj 9707368

==========================================
job_number:          9707368
owner:               s_110
cwd:                 /escratch4/s_110/s_110_Jan_28
hard_queue_list:     rcc-30d
script_file:         sub.sh
.....
usage    1:          cpu=00:01:27, mem=0.96498 GBs,
                     io=0.00014, vmem=73.734M,
                     maxvmem=75.734M
```

- For a finished jobs: qacct

```
$ qacct -j  970732

==============================
qname        rcc-30d
hostname     compute-7-12.local
jobname      sub.sh
jobnumber    9707323
.....
cpu          183.320
mem          2.021
io           0.000
maxvmem      6.530G
```

GACRC

Thank You!