GACRC

Introduction to HPC Using zcluster at GACRC

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

GACRC

OVERVIEW

- GACRC
- High Performance Computing (HPC)
- zcluster – Architecture, Operation
- Access and Working with zcluster

GACRC

Georgia Advanced Computing Resource Center

Who Are We?

- **G**eorgia **A**dvanced **C**omputing **R**esource **C**enter (**GACRC**)
- 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

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

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

GACRC

Computer Cluster

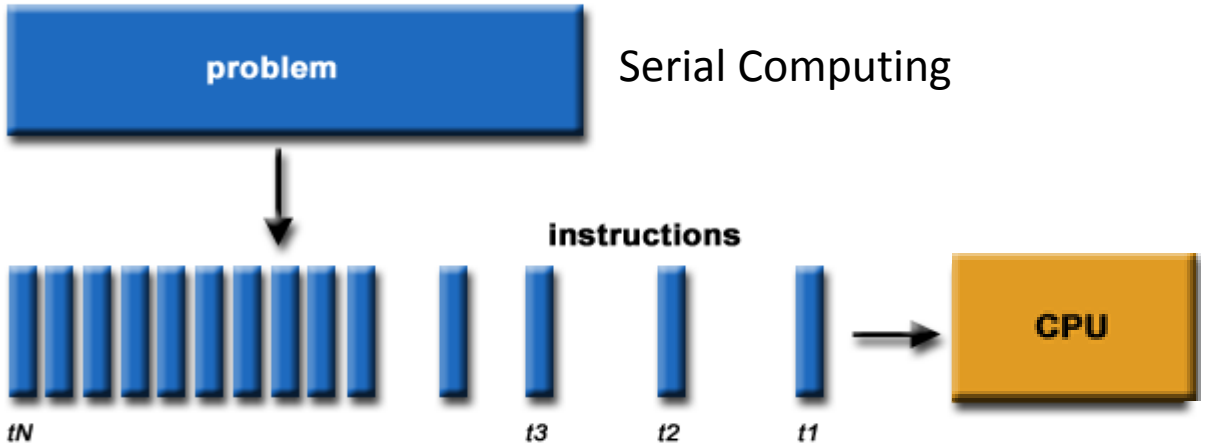
- Cluster is a widely used term meaning independent computers combined into a unified system through software and high speed dedicated network
  - It provides greater computational power than a single computer can provide – LARGER problems can be solved
- Used for:
  - High Availability (HA)
  - Greater reliability
  - High Performance Computing (HPC)

High Performance Computing (HPC)

Several “definitions” can be found, but, in general:

Use of Parallel processing for solving complex computational problems using advanced application programs efficiently, reliably and quickly

High Performance Computing (HPC)

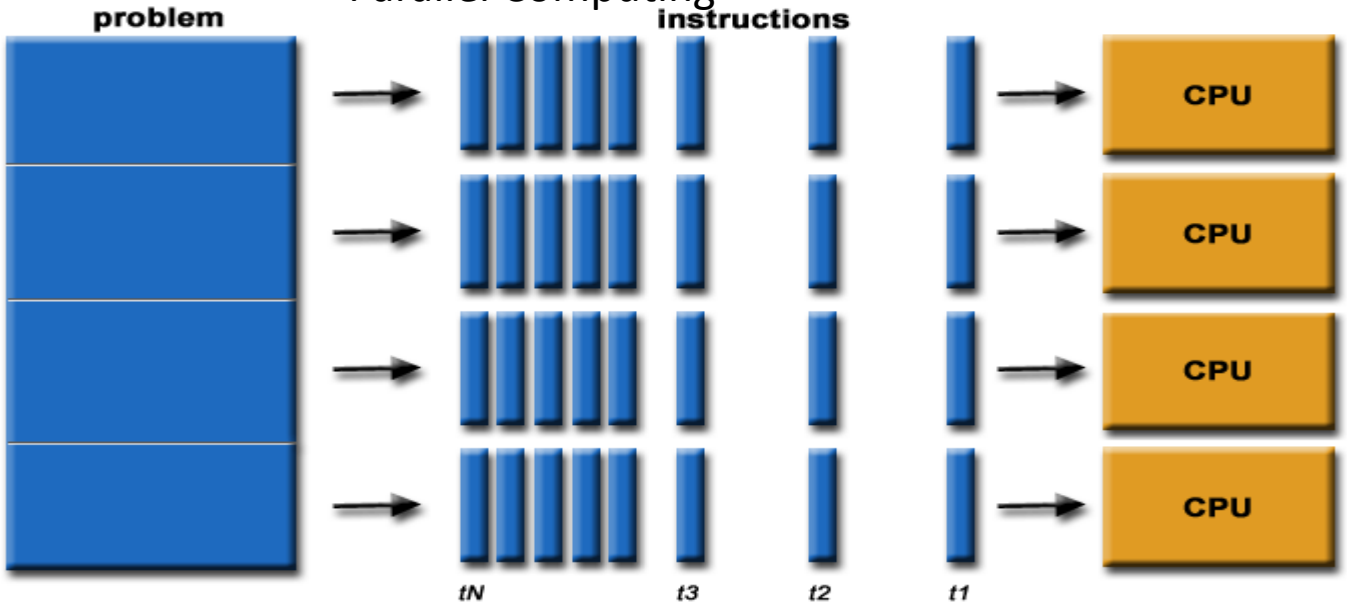


Serial Computing

Serial Computing

- A problem is broken into a discrete series of instructions
- Instructions are executed sequentially
- Executed on a single processor
- Only one instruction may execute at any moment in time

Parallel Computing



Parallel Computing

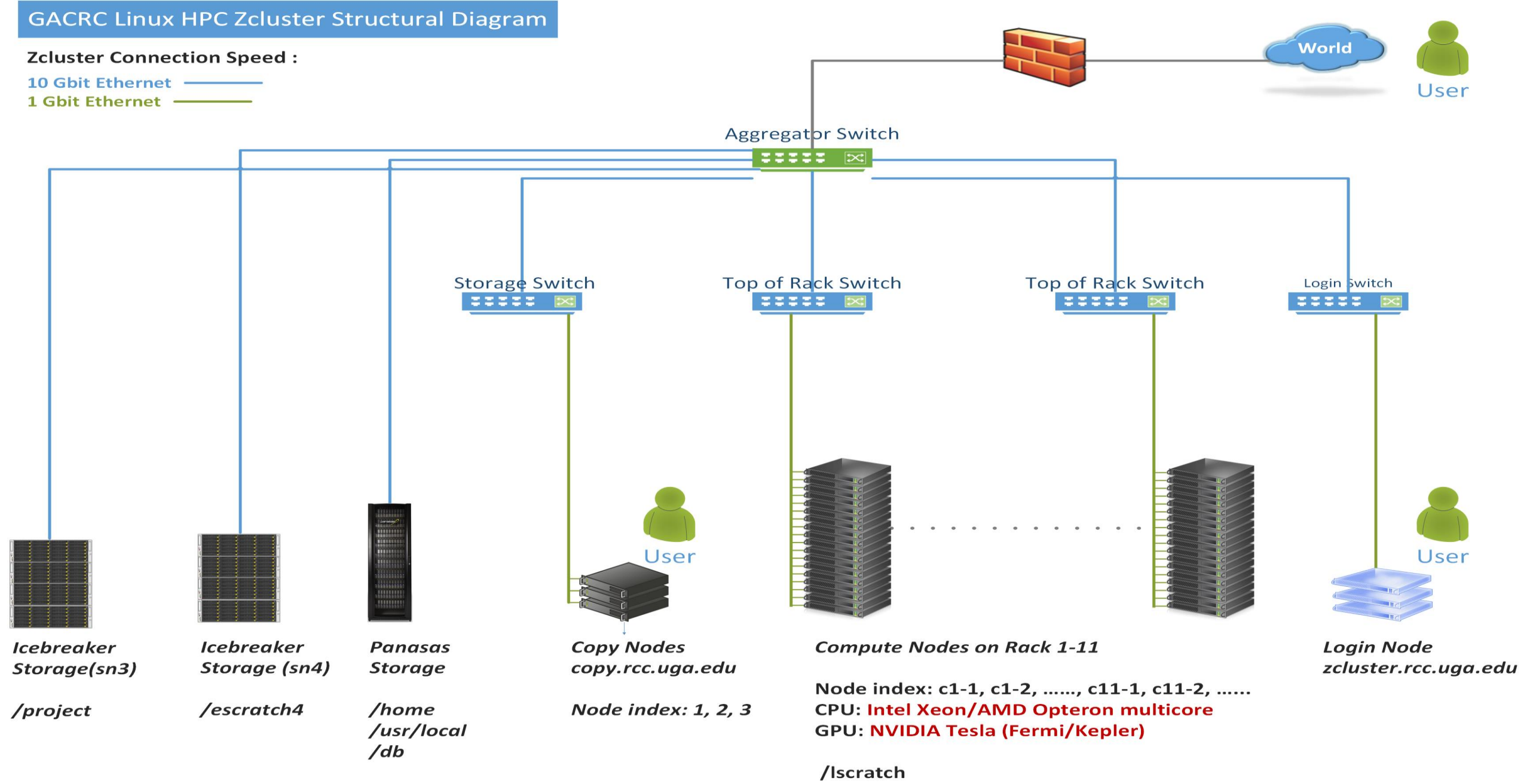
- A problem is broken into discrete parts that can be solved concurrently
- Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different processors
- An overall control/coordination mechanism is employed

GACRC

Zcluster Overview

- zcluster Structure
- General Information
- Computing Resources
- Storage Environment

GACRC Linux HPC Zcluster Structural Diagram



zcluster – General Information

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

- Operating System: 64-bit Red Hat Enterprise Linux 5 (RHEL 5)

- Login Node: zcluster.rcc.uga.edu

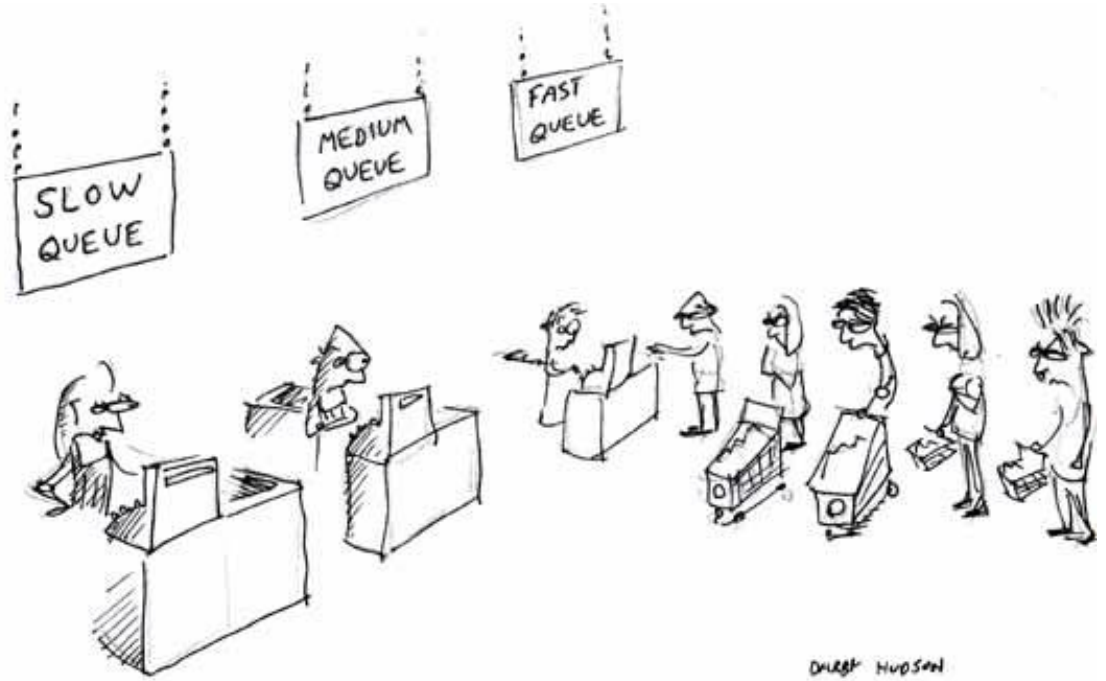
Interactive Node: compute-14-7/9
  Copy Node: copy.rcc.uga.edu

- Internodal Communication: 1Gbit network
  - compute nodes ⇔ compute nodes
  - compute nodes ⇔ storage systems

NOTE: Please Do Not run jobs on the zcluster login node - use the Queues or the Interactive Nodes.

zcluster – General Information

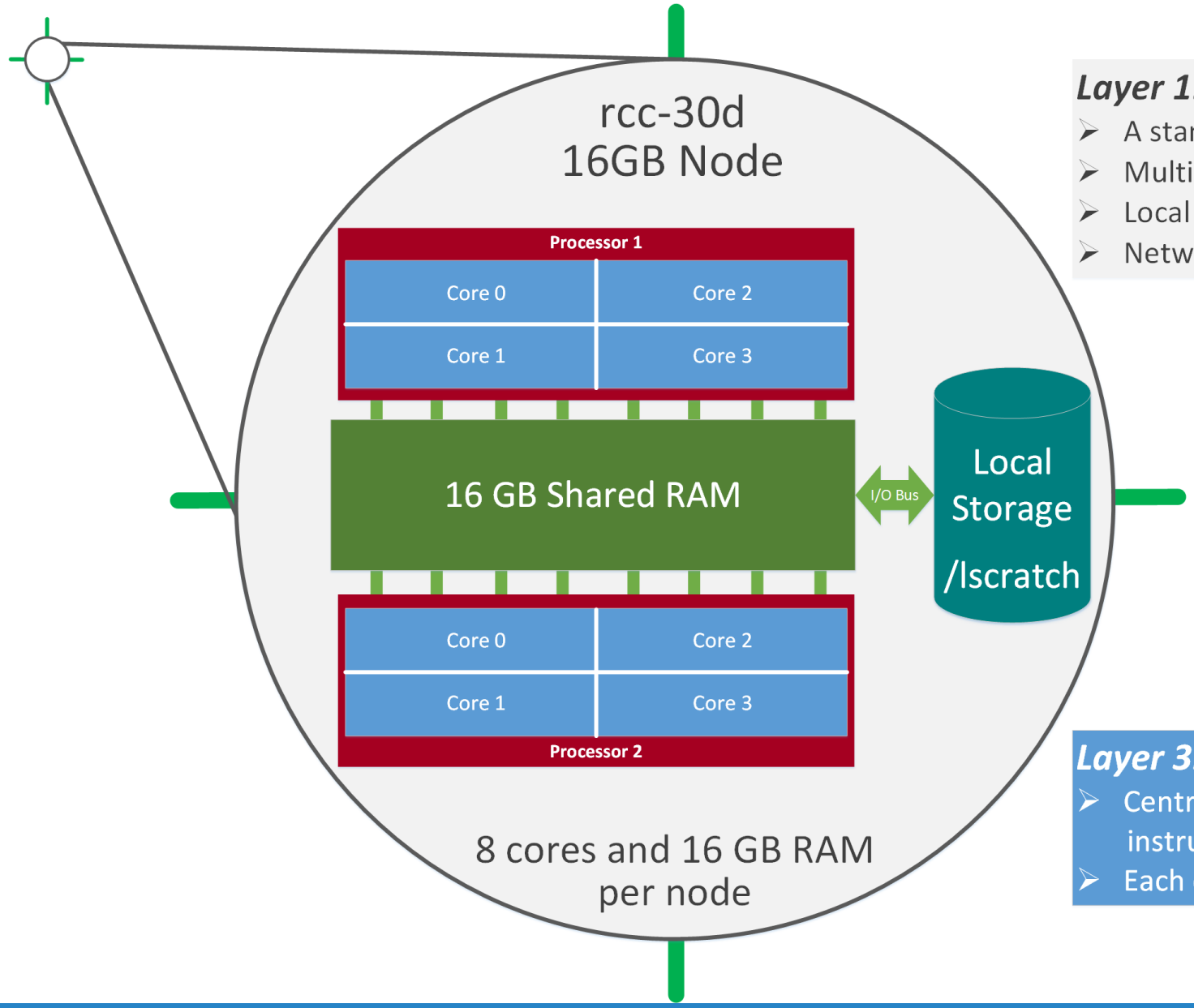
- Batch-queueing System:
  - Jobs can be started (submitted), monitored, and controlled
  - Determine which compute node is the best place to run a job
  - Determine appropriate execution priority for a job to run
- On zcluster: Sun Grid Engine (SGE)



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	1	Intel Xeon	8	128	8	N/A
		4		8	192	32	
		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

GACRC

zcluster – Storage Environment

- Mainly, there are 4 different storage locations:
  - **Home directory** - where you would land after logging in to the zcluster
  - **lscratch** - is the storage that is local to the computation nodes
  - **escratch4** - is temporary scratch which is visible to all nodes
  - **Project** storage area - is long term, and is created for a lab.

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
    - Taken once a day, and maintained 4 daily ones and 1 weekly one
    - Name: .snapshot, e.g., /home/abclab/jsmith/.snapshot
    - Completely invisible, however, user can "cd" into it and then "ls":

```
pakala@zcluster:~$ pwd
/home/rccstaff/pakala
pakala@zcluster:~$ ls -a
.              .bash_history.compute-14-7  .bash_history.zhead  .bashrc         cmd_kill  .java     RNA_SEQ
..             .bash_history.compute-14-9  .bash_logout         batchsub_demo   e4        .mozilla  .ssh
.bash_history  .bash_history.zcluster      .bash_profile        Blast           .emacs    ncbidb    .viminfo
pakala@zcluster:~$ cd .snapshot
pakala@zcluster:~/.snapshot$ ls
2015.11.29.00.00.01.weekly  2015.12.06.00.00.01.weekly  2015.12.07.01.00.01.daily  2015.12.09.01.00.01.daily
2015.12.05.01.00.01.daily   2015.12.06.01.00.01.daily   2015.12.08.01.00.01.daily
```

➔ .snapshot is NOT shown here!

➔ Can "cd" into " .snapshot

➔ And "ls" to list its contents

GACRC

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/pakala/pakala_Nov_13
  - Create with `make_escratch` command
  - Visible to all nodes with a quota of 4TB
  - No snapshot backup
  - To be deleted after 37 days

GACRC

zcluster – Storage Environment

Filesystem	Role	Quota	Accessible from	Intended Use	Notes
/home/abclab/username	Home	100GB	zcluster.rcc.uga.edu (Login) copy.rcc.uga.edu (Copy) Interactive nodes (Interactive) compute nodes (Compute)	Highly static data being used frequently	Snapshots
/escratch4/username	Scratch	4TB		Temporarily storing large data being used by jobs	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	copy.rcc.uga.edu (Copy)	Long-term data storage	Group sharing possible

Note: 1. /usr/local : Software installation directory
/db : bioinformatics database installation directory
2. To login to Interactive nodes, use `qlogin` from Login node

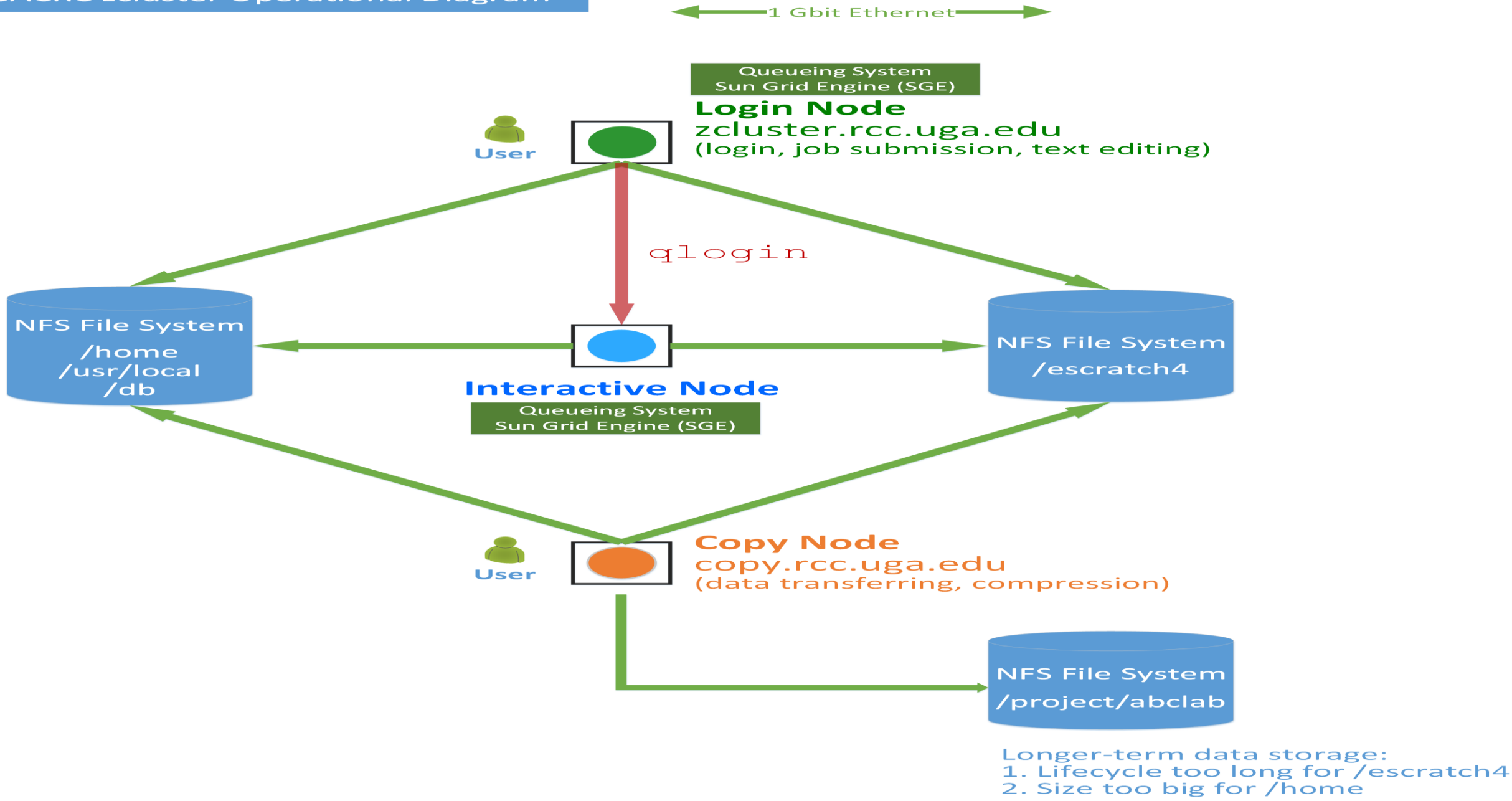
GACRC

zcluster – Storage Environment

6 Main Function	On/From-Node	Related Filesystem
Login Landing	Login or Copy	/home/abclab/username (Home) (Always!)
Batch Job Submitting	Login or Interactive	/escratch4/username (Scratch) (Suggested!) /home/abclab/username (Home)
Interactive Job Running	Interactive	/escratch4/username (Scratch) /home/abclab/username (Home)
Data Archiving , Compressing and Transferring	Copy	/escratch4/username (Scratch) /home/abclab/username (Home)
Job Data Temporarily Storing	Compute	/lscratch/username (Local Scratch) /escratch4/username (Scratch)
Long-term Data Storing	Copy	/project/abclab

How does zcluster operate?

GACRC zcluster Operational Diagram



Longer-term data storage:
1. Lifecycle too long for /escratch4
2. Size too big for /home

GACRC

Zcluster – Tips, Dos and Don'ts

Before we start:

- To get zcluster to be your best HPC buddy, go to
  GACRC Wiki (http://wiki.gacrc.uga.edu)
  GACRC Web (http://gacrc.uga.edu)
- To get the most effective and qualified support from us, go to
  GACRC Support (https://wiki.gacrc.uga.edu/wiki/Getting_Help)
- To work happily and productively, follow the cluster's Community Code of Conduct (CCOC)

zcluster – Tips, Dos and Don'ts - continued

- Cluster's CCOC:

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

6 rules of thumb to remember:

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

GACRC

How to work with zcluster? - Overview

- Start with zcluster
- Connect & Login
- Transfer Files
- Run Interactive Jobs
- Submit Batch Jobs
  - How to submit ***serial***, ***threaded***, and ***MPI*** batch jobs
  - How to check job status, cancel a job, etc.
- Software Installed

GACRC

Getting Started with zcluster

- You need a User Account : username@zcluster.rcc.uga.edu
- Procedure: https://wiki.gacrc.uga.edu/wiki/User_Accounts
- User receives an email notification once the account is ready
- User can use `passwd` command to change initial temporary password
- 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 members of his/her computing lab: http://help.gacrc.uga.edu/userAcct.php

GACRC

Connection & Login @ zcluster

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

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

or

```
ssh -X pakala@zcluster.rcc.uga.edu
```

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

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

- Logging in: You will be prompted for your zcluster password

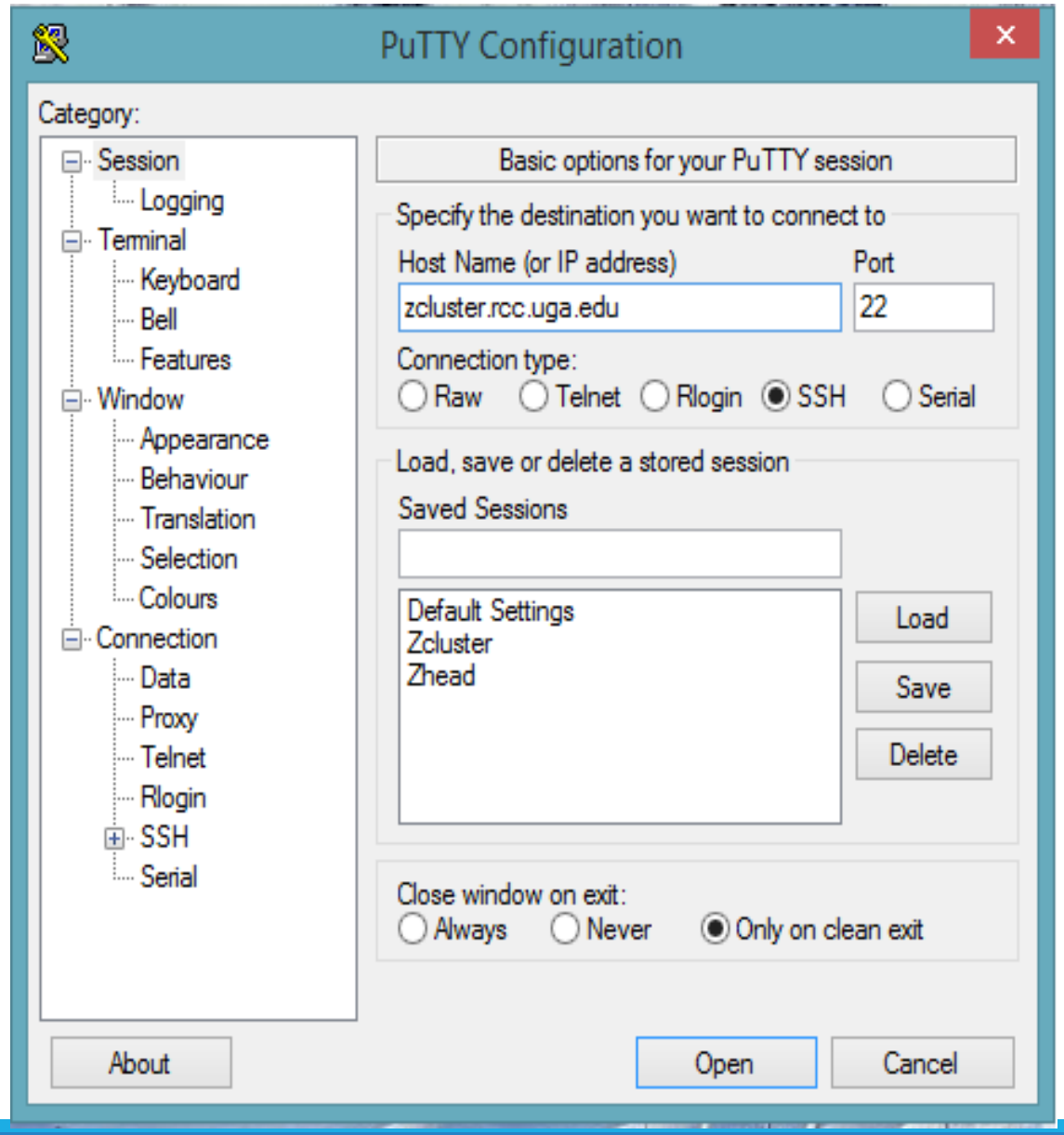
```
pakala@zcluster.rcc.uga.edu's password:
```

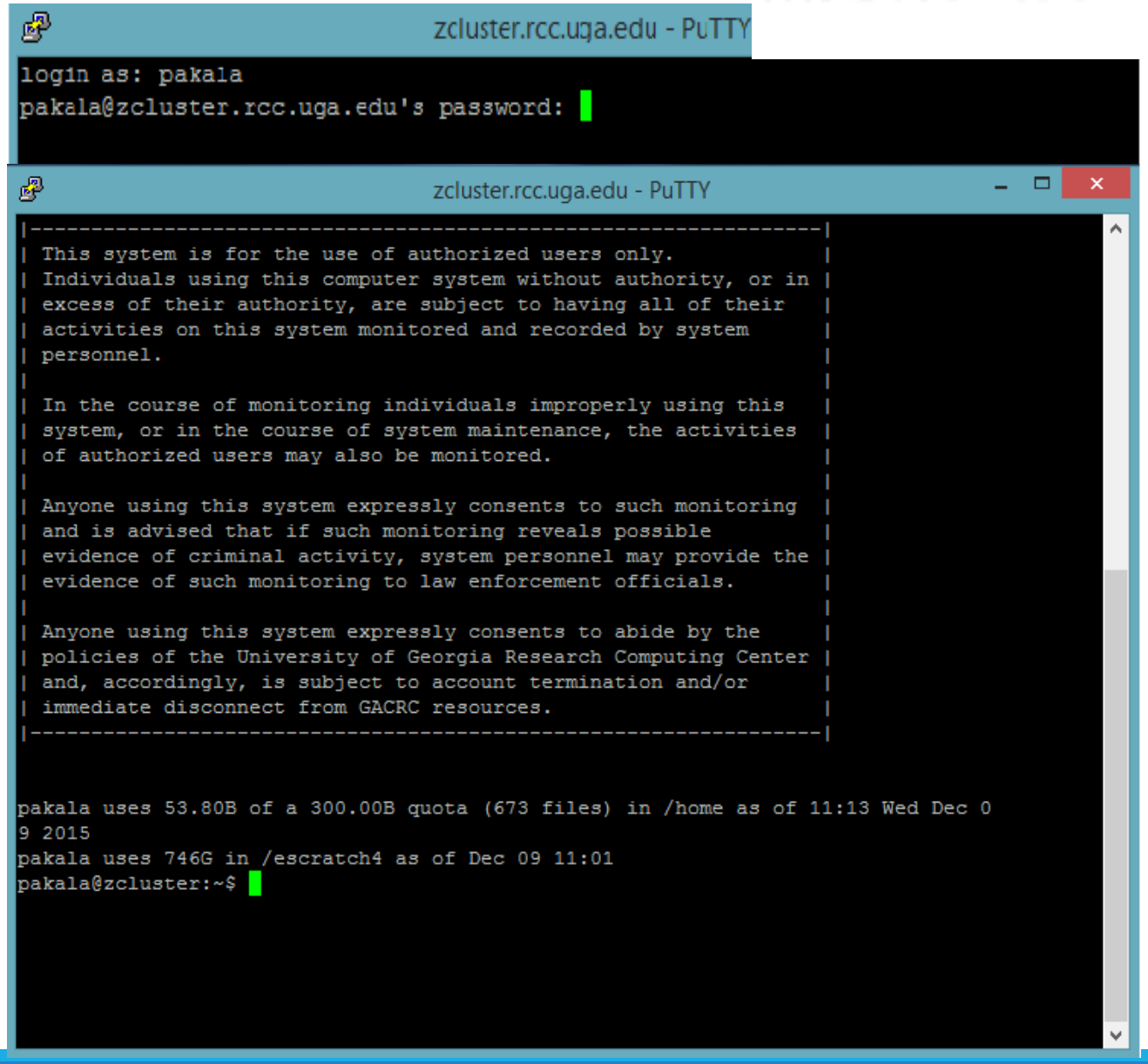
([3] On Linux/Mac, when you type in the password, the prompt blinks and does not move)

- Logging out: `exit` to leave the system

```
pakala@zcluster:~$ exit
```

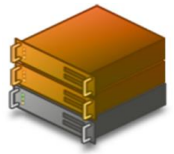
PUTTY: LOGIN


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
Telnet
Rlogin
SSH
Serial
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
zcluster.rcc.uga.edu
22
Connection type:
Raw
Telnet
Rlogin
SSH
Serial
Load, save or delete a stored session
Saved Sessions
Default Settings
Zcluster
Zhead
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Open
Cancel


zcluster.rcc.uga.edu - PuTTY
login as: pakala
pakala@zcluster.rcc.uga.edu's password:
zcluster.rcc.uga.edu - PuTTY
This system is for the use of authorized users only.
Individuals using this computer system without authority, or in
excess of their authority, are subject to having all of their
activities on this system monitored and recorded by system
personnel.
In the course of monitoring individuals improperly using this
system, or in the course of system maintenance, the activities
of authorized users may also be monitored.
Anyone using this system expressly consents to such monitoring
and is advised that if such monitoring reveals possible
evidence of criminal activity, system personnel may provide the
evidence of such monitoring to law enforcement officials.
Anyone using this system expressly consents to abide by the
policies of the University of Georgia Research Computing Center
and, accordingly, is subject to account termination and/or
immediate disconnect from GACRC resources.
pakala uses 53.80B of a 300.00B quota (673 files) in /home as of 11:13 Wed Dec 0
9 2015
pakala uses 746G in /escratch4 as of Dec 09 11:01
pakala@zcluster:~$

Transfer Files @ zcluster

User's local

 Copy node (copy.rcc.uga.edu) of zcluster

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

  E.g. 1: On local machine, do Local → zcluster

```
scp file1 username@copy.rcc.uga.edu:~/subdir
```

```
scp *.dat username@copy.rcc.uga.edu:~/subdir
```

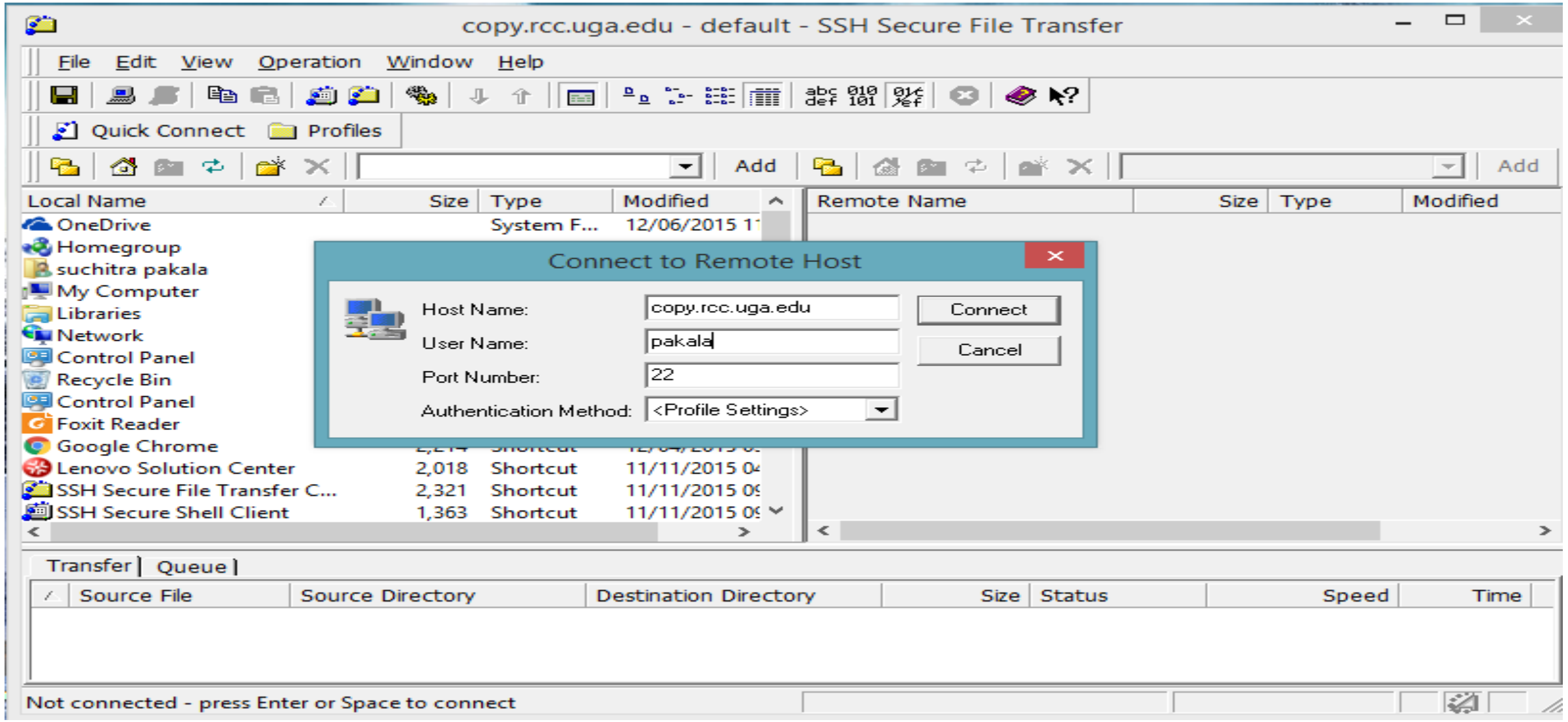
  E.g. 2: On local machine, do zcluster → Local

```
scp username@copy.rcc.uga.edu:~/subdir/file ./
```

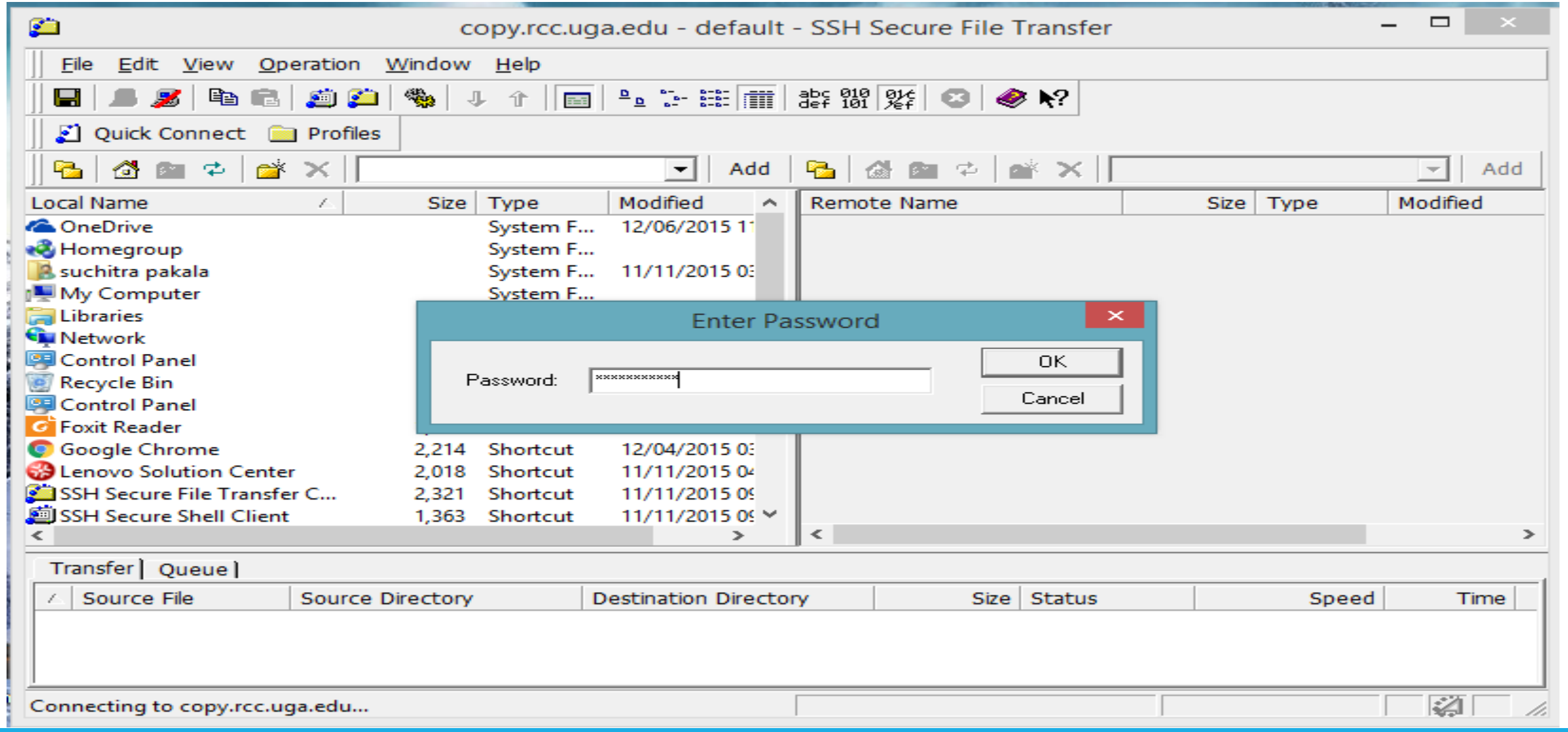
```
scp username@copy.rcc.uga.edu:~/subdir/\*.dat ./
```

- On Windows: FileZilla, WinSCP, SSH Secure Client, etc.

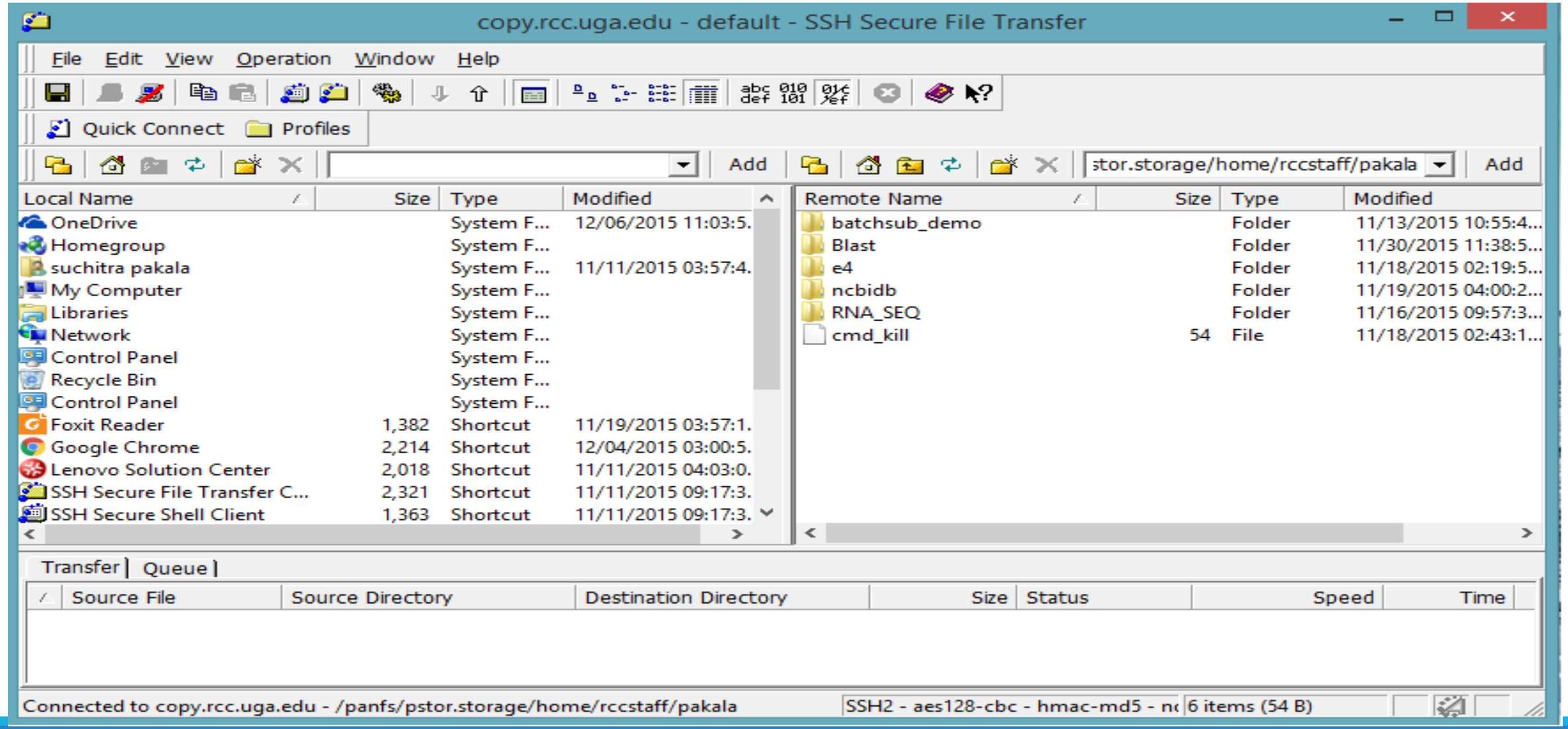
SSH Secure Client: Connecting to COPY Node



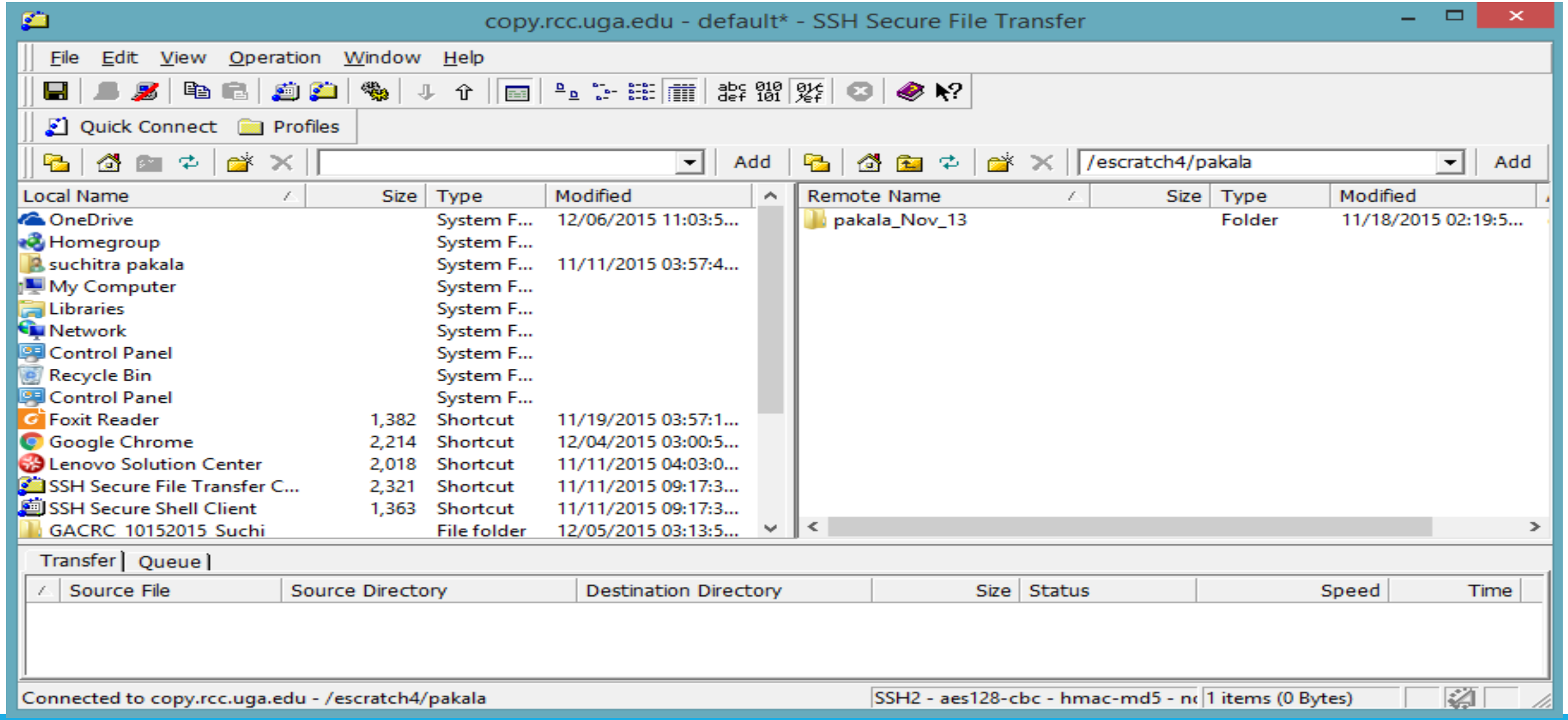
SSH Secure Client: Connecting to COPY Node



SSH Secure : Connected to Home Directory



SSH Secure : Navigated to /escratch4/pakala



Run Interactive Jobs @ zcluster

- To run an interactive job, you need to open a session on an interactive node using qlogin command:

```
pakala@zcluster:~$ qlogin
Your job 9559204 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 9559204 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, about interactive parallel jobs.
  https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_zcluster

GACRC

Submit Batch Jobs @ zcluster

- Components you need to submit a batch job:
  - Software already installed on zcluster
  - 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)
- Common commands you need:
  - qsub with specifying queue name, threads or MPI rank number
  - qstat, qdel
  - qacct, qsj, etc.

Batch *Serial* Job @ zcluster

Step 1: Create a job submission script fastqc_*sub.sh* running Fastqc tool:

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

cd /escratch4/pakala/pakala_Nov_13           → Specify and enter (cd) the working directory
export PATH=/usr/local/fastqc/latest:${PATH} → Export command helps to set the Environment variables
time fastqc SRR1369670.fastq -o Output_File  → Run Fastqc 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 fastqc_sub.sh
```

Submit a job

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

Your job submission script

OR

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

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

Batch *Threaded* Job @ zcluster

❖ Step 1: Create a job submission script blast.*sh* running Blast:

```
#!/bin/bash

cd /escratch4/pakala/pakala_Nov_13

time /usr/local/ncbiblast/latest/bin/blastall -p 2 [options]
```

➔ Run Blastall with 2 threads

Number of Threads =
Number of Cores Requested

❖ Step 2: Submit it to the queue:

```
$ qsub -q rcc-30d -l mem_total=20g -pe thread 2 ./blast.sh
```

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

2 cores requested

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

Batch *MPI* Job @ zcluster

❖ Step 1: Create a job submission script *sub.sh* running RAxML:

```
#!/bin/bash
cd /escratch4/pakala/pakala_Nov_13

export MPIRUN=/usr/local/mpich2/1.4.1p1/gcc 4.5.3/bin/mpirun

$MPIRUN -np $NSLOTS /usr/local/raxml/latest/raxmlHPC-MPI-SSE3 [options]
```

➔ Define and export environment variable (MPIRUN) for convenient usage

➔ Run RAxML with 20 MPI processes (-np $NSLOTS)

❖ Step 2: Submit it to the queue:

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

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

GACRC

Check and Cancel Jobs @ zcluster

- To check the status of all queued and running 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 –g t	➔ list all nodes used by your jobs

- To cancel a queued or running job: qdel

qdel –u pakala	➔ deleted all your jobs
qdel 12345	➔ deletes your job with JOBID 12345

- To list detailed information about a job: qsj, qacct

qsj 12345	➔ shows information, e.g., maxvmem, about the RUNNING job with JOBID 12345
qacct –j 12345	➔ shows information, e.g., maxvmem, about the ENDED job with JOBID 12345

GACRC

Software Installed @ 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. https://wiki.gacrc.uga.edu/wiki/Bioinformatics_Databases
- For a complete list of applications: https://wiki.gacrc.uga.edu/wiki/Software

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

GACRC

THANK YOU for your patience

GACRC

Questions?