GACRC Teaching Cluster Training

Georgia Advanced Computing Resource Center (GACRC)
Enterprise Information Technology Services (EITS)

UNIVERSITY OF GEORGIA
Georgia Advanced Computing Resource Center

Agenda

- GACRC
- Teaching Cluster
    - Cluster Diagram and Overview
    - Directories
    - Computational Partitions
    - Software Environment
- Job Submission Workflow
- Useful Commands: sq --me, sacct-gacrc -X
- GACRC wiki and User Support
- Appendices

GACRC

- A high-performance-computing (HPC) center at UGA
- Advanced computing environment for the UGA research and education community
  - HPC infrastructure location in the Boyd Data Center
  - Comprehensive collection of scientific, engineering and business applications
  - Consulting and training services

- GACRC Wiki: http://wiki.gacrc.uga.edu
- Help and Support: http://help.gacrc.uga.edu
- Main website: http://gacrc.uga.edu
- Kaltura Channel (tutorial videos): https://kaltura.uga.edu/channel/GACRC/176125031

Running Computations

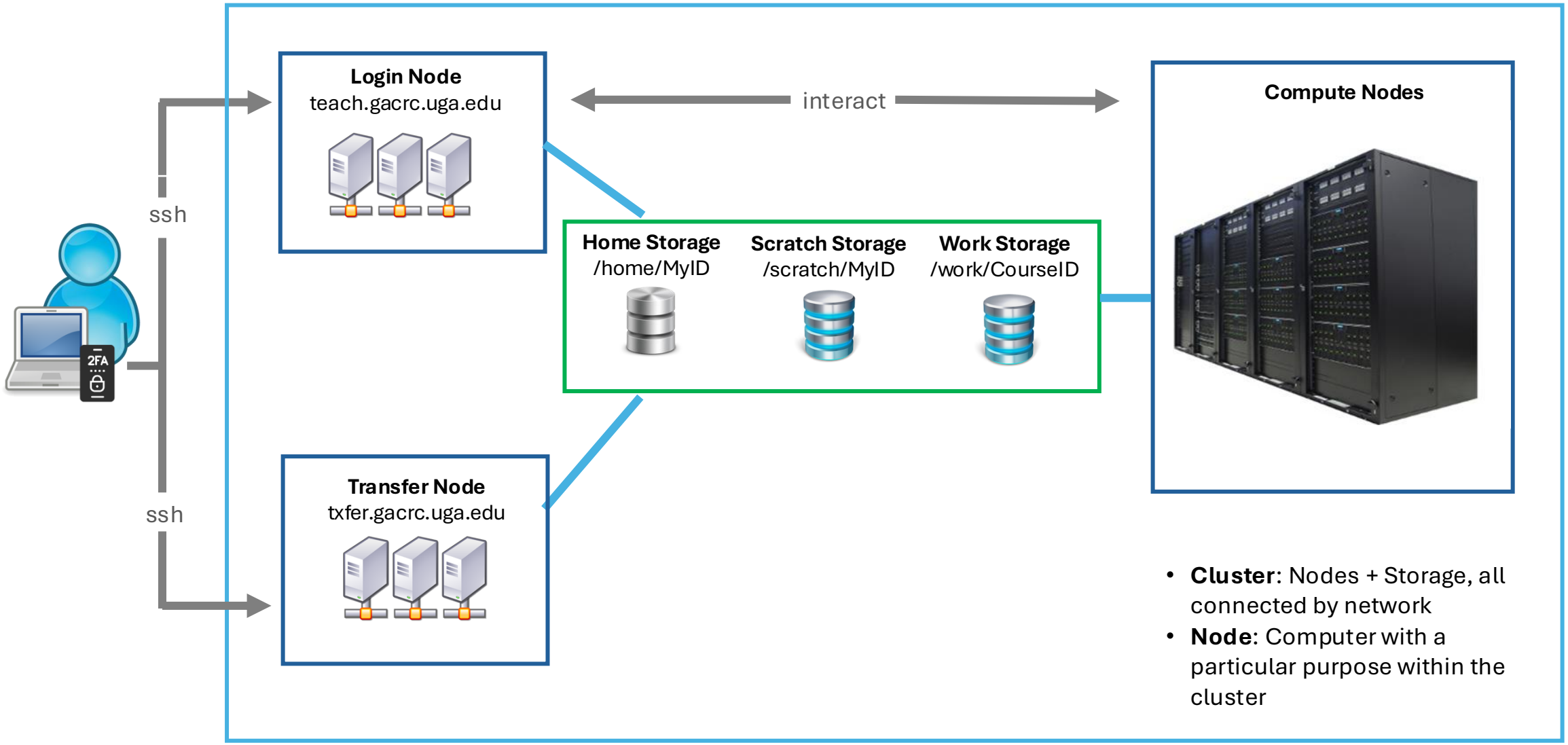
- You can run your computations/programs/scripts in **jobs** on the Teaching Cluster
- There are a few different types of **jobs** on the cluster depending on how you want to run your computation
- A few we will talk about today are:

Submission script jobs (batch jobs)

Interactive jobs

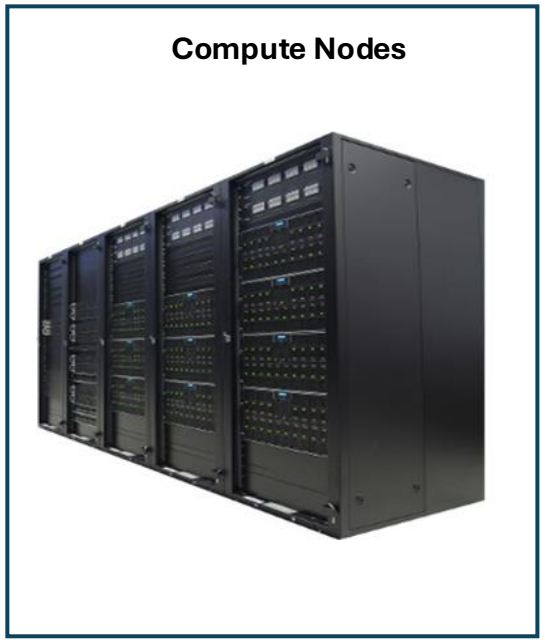
OnDemand jobs

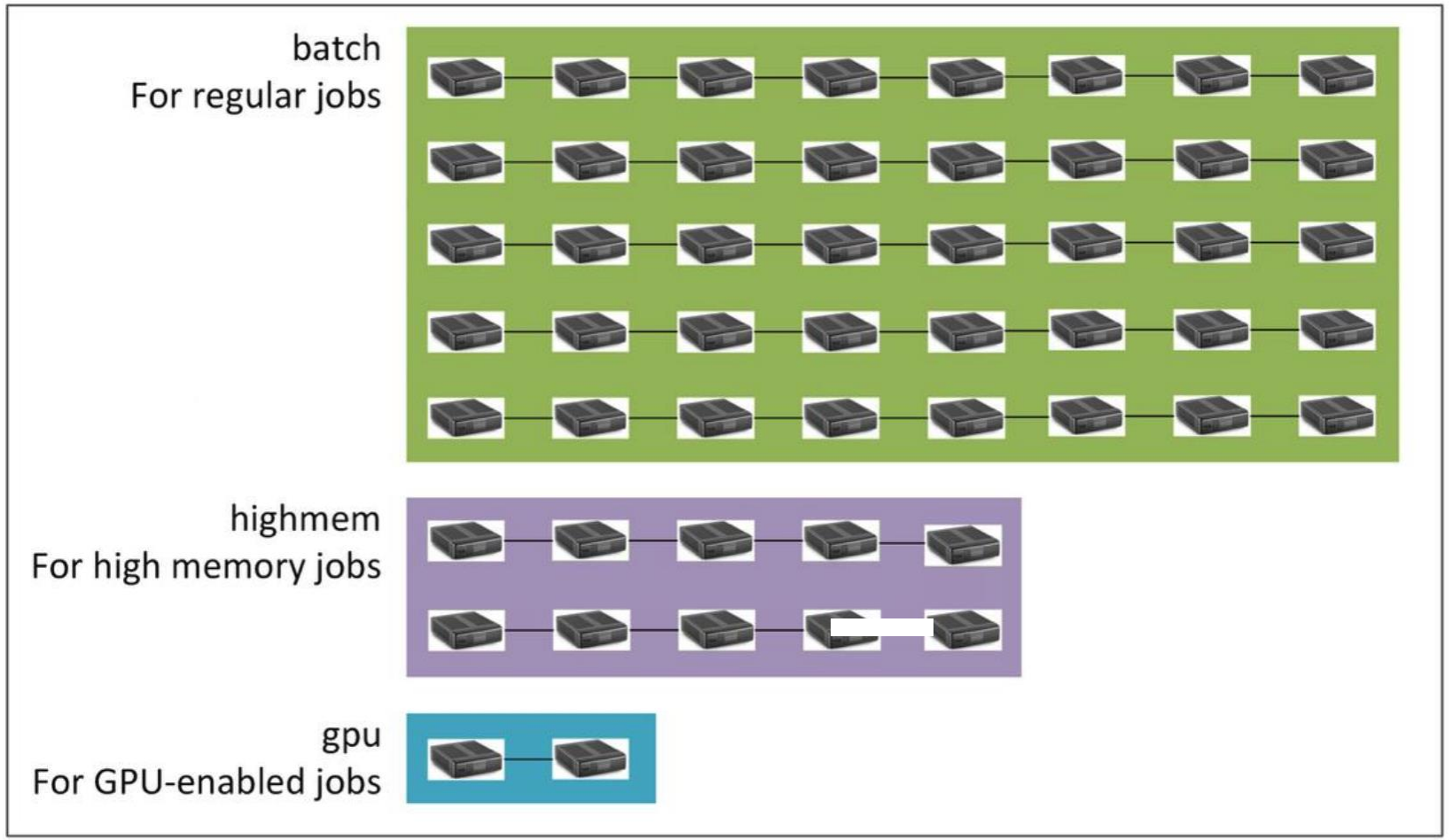
Teaching Cluster



- **Cluster**: Nodes + Storage, all connected by network
- **Node**: Computer with a particular purpose within the cluster

Computational Partitions





Compute nodes are divided into groups called **partitions**. A **partition** is a collection of compute nodes for a particular computing need.

Computational Partitions - Resources

*subject to change

Partition	Number of Nodes	Description	Notes
batch	8	32-core, 128GB RAM, AMD Naples	Regular nodes
highmem	2	64-core, 1TB RAM, Intel Xeon	For running high memory jobs
gpu	1	32-core, 190 GB RAM, Intel Skylake, 1 P100 GPU	For running GPU-enabled jobs
interactive	2	32-core, 128GB RAM, Intel Xeon	For interactive jobs
franklin_gpu	2	96-core, 1TB RAM, Intel Sapphire Rapids, 4 A30 GPU	For Franklin College only

Software Environment

- Software is installed as “software modules”
- Naming format of software modules: **Name/Version-Toolchain**
  - e.g., **Python/3.12.3-GCCcore-13.3.0**
- Module commands:
  - ml spider *pattern* : Search module names matching a *pattern*
  - ml *ModuleName* : Load a module into your submission script
  - ml : List modules currently loaded
  - ml -*ModuleName* : Remove a module
  - ml purge : Remove all currently loaded modules

NOTE: DO NOT LOAD/USE MODULES ON THE LOGIN/SUBMIT NODES! (teach-sub)

Software Environment - toolchains

- When you load more than one software module, you must pay attention to their toolchain compatibility
- Having modules loaded with incompatible toolchains can cause errors in your job, such as:

```
Lmod has detected the following error:
These module(s) exist but cannot be loaded as requested
```

- The toolchain compatibility table can be found at https://wiki.gacrc.uga.edu/wiki/Available_Toolchains_and_Toolchain_Compatibility

Job Submission Workflow

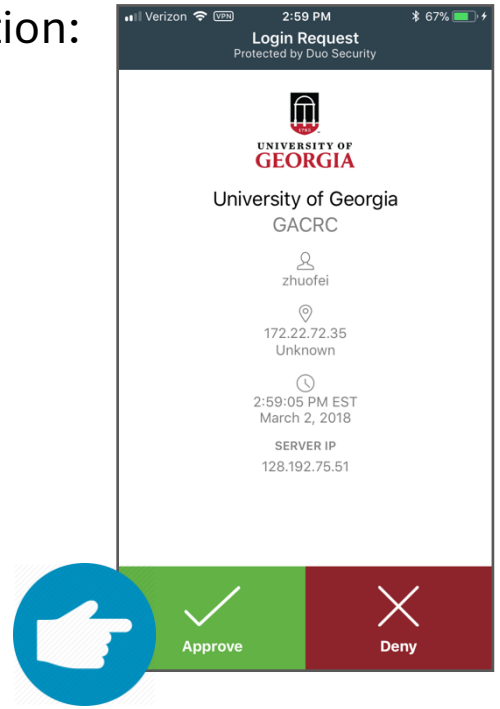
1. Log on to Login node using MyID and password, and two-factor authentication with Archpass Duo: **ssh MyID@teach.gacrc.uga.edu**
2. On Login node, change directory to your scratch space: **cd /scratch/MyID**
3. Create a new directory for a job : **mkdir training**
4. Change your working directory to training : **cd training**
5. Transfer data from local computer to training : use scp or WinSCP via the transfer node (txfer.gacrc.uga.edu)
6. Make a job submission script: **nano sub.sh**
7. Submit the job: **sbatch sub.sh**
8. Check job status : **sq --me** and **sacct-gacrc -X** or Cancel a job : **scancel jobID**

Step 1: Login to login node – Mac users

1. Open Terminal utility
2. Type in the command line: ssh MyID@teach.gacrc.uga.edu
3. You will be prompted for your MyID password
4. Sapelo2 access requires ID verification using two-factor authentication with Archpass Duo. If you are not enrolled in Archpass Duo, please refer to https://eits.uga.edu/access_and_security/infosec/tools/archpass/ on how to enroll

Step 1: Login to login/submit node (Mac users)

Duo authentication:



Steps on the terminal utility on Mac:

```
[zhuofei@localhost ~]$
[zhuofei@localhost ~]$ ssh zhuofei@teach.gacrc.uga.edu      ← Log on
Password:                                  ← Input MyID password!
......
Enter a passcode or select one of the following options:
 1. Duo Push to XXX-XXX-5758
 2. Phone call to XXX-XXX-5758
 3. Phone call to XXX-XXX-1925
 4. SMS passcodes to XXX-XXX-5758

Passcode or option (1-5): 1          ← Select Duo authentication option!
Success. Logging you in...
Last login: Tue Sep 15 11:22:42 2020 from 128.192.75.65

zhuofei@teach-sub1 ~$  ← I am on login node teach-sub1!
```

Step 1: Login to login/submit node (Windows users) - Download PuTTY

1. Download and install PuTTY: https://www.putty.org/

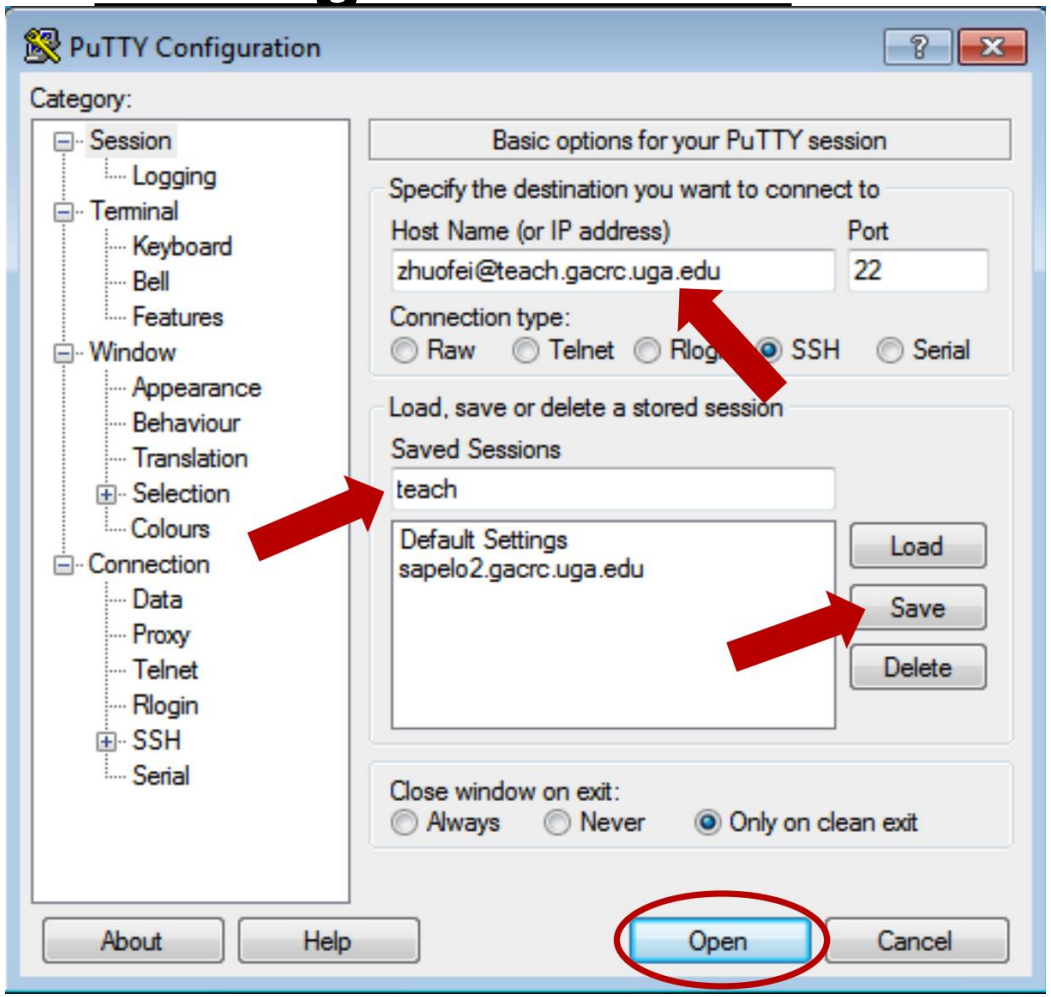
2. Detailed downloading and installation instructions:

   https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY

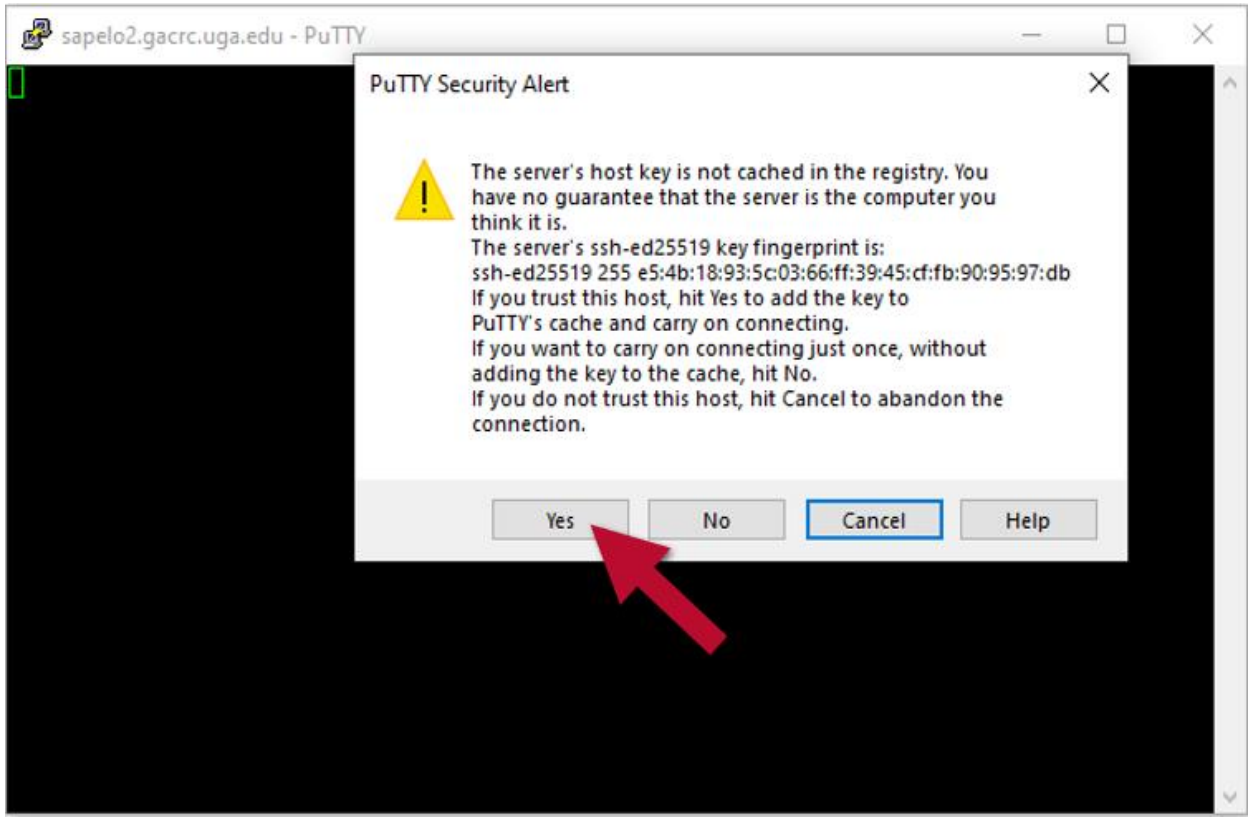
3. Detailed configuring and usage instructions:

   https://wiki.gacrc.uga.edu/wiki/How_to_Install_and_Configure_PuTTY#Configuring_PuTTY

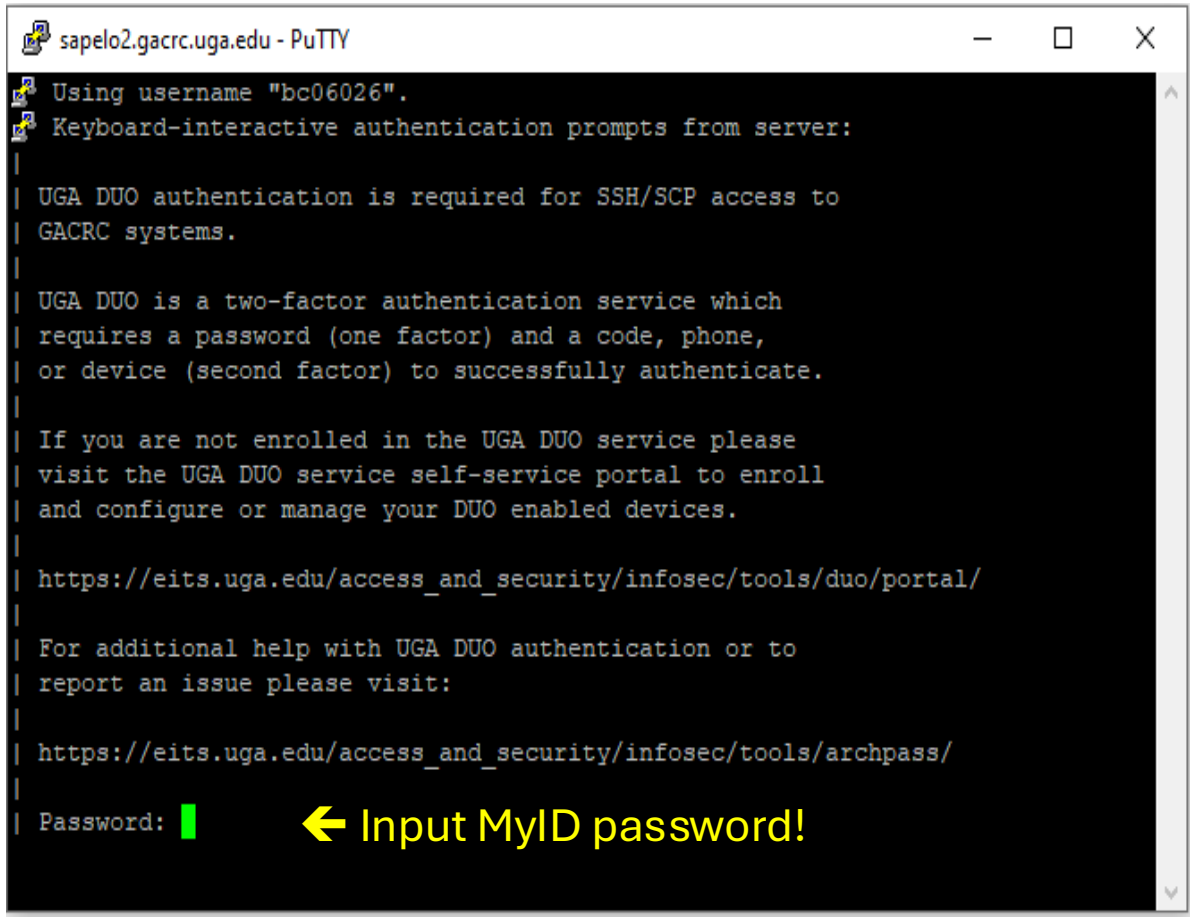
Step 1: Login to login/submit node (Windows users) – Configure PuTTY

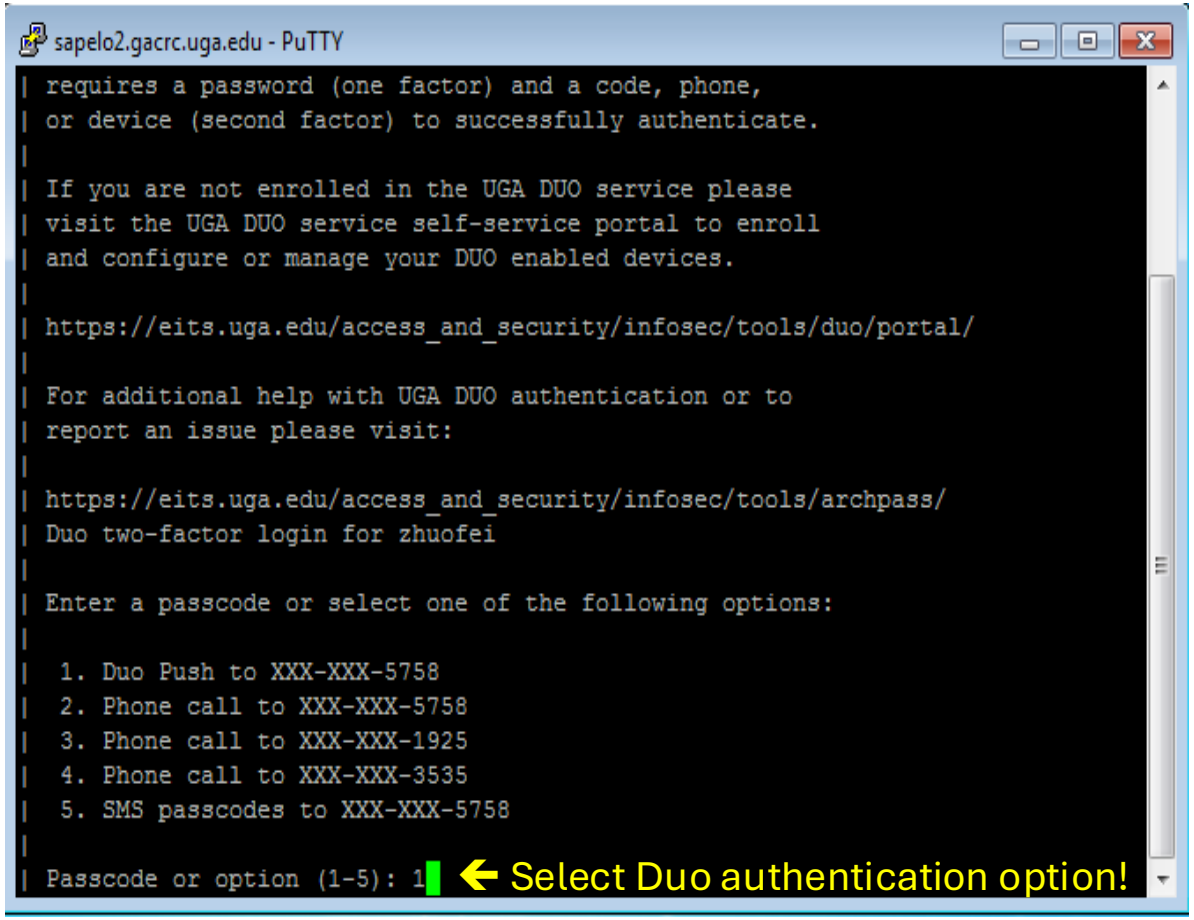


The first time you connect to a login node, PuTTY will give you this security alert window. Please click "Yes" or "Accept"



Step 1: Login to login/submit node (Windows users) – SSH via PuTTY





Step 2: Change Directory to “scratch” dir

- Once logged on, your current location will be your home directory

```
zhuofei@teach-sub1 ~$ pwd
/home/zhuofei                    ← this is my home directory!
```

- Use cd command to change your current directory to /scratch/MyID

```
zhuofei@teach-sub1 ~$ cd /scratch/zhuofei/
zhuofei@teach-sub1 zhuofei$ pwd
/scratch/zhuofei                 ← this is my scratch space!
```

- Use ls command to take a look in /scratch/MyID

```
zhuofei@teach-sub1 zhuofei$ ls
apps  input_1.txt
```

Step 3: Create a subdirectory

- Use the mkdir command to create a new directory within /scratch/MyID

```
zhuofei@teach-sub1 zhuofei$ mkdir jobdir
zhuofei@teach-sub1 zhuofei$ ls
apps  input_1.txt  jobdir
```

Step 4: Change directories to workDir

- Use the cd command to change your directory from /scratch/MyID to /scratch/MyID/workDir
- Can use a relative path or an absolute path

- Using a relative path (relative to where we currently are which is /scratch/MyID):

```
zhuofei@teach-sub1 zhuofei$ cd workDir
zhuofei@teach-sub1 zhuofei$ pwd
/scratch/zhuofei/workDir
```

- Using an absolute path (works no matter where you currently are):

```
zhuofei@teach-sub1 zhuofei$ cd /scratch/zhuofei/workDir
zhuofei@teach-sub1 zhuofei$ pwd
/scratch/zhuofei/workDir
```

Step 5: Transfer data to the Cluster

- You can transfer files between your computer and the Teaching Cluster with scp (for Mac users) and WinSCP (for Windows users)

Step 5: Transfer data to the Cluster - Mac

1. Open the terminal application (do not log in to the Cluster)
2. Use the scp command with the Transfer node following the examples below

- The syntax for the scp command is

```
scp [source file] [target location]
```

Ex 1: transfer a file from your local computer onto the Teaching Cluster

```
scp ./file MyID@txfer.gacrc.uga.edu:/scratch/MyID/jobdir
```

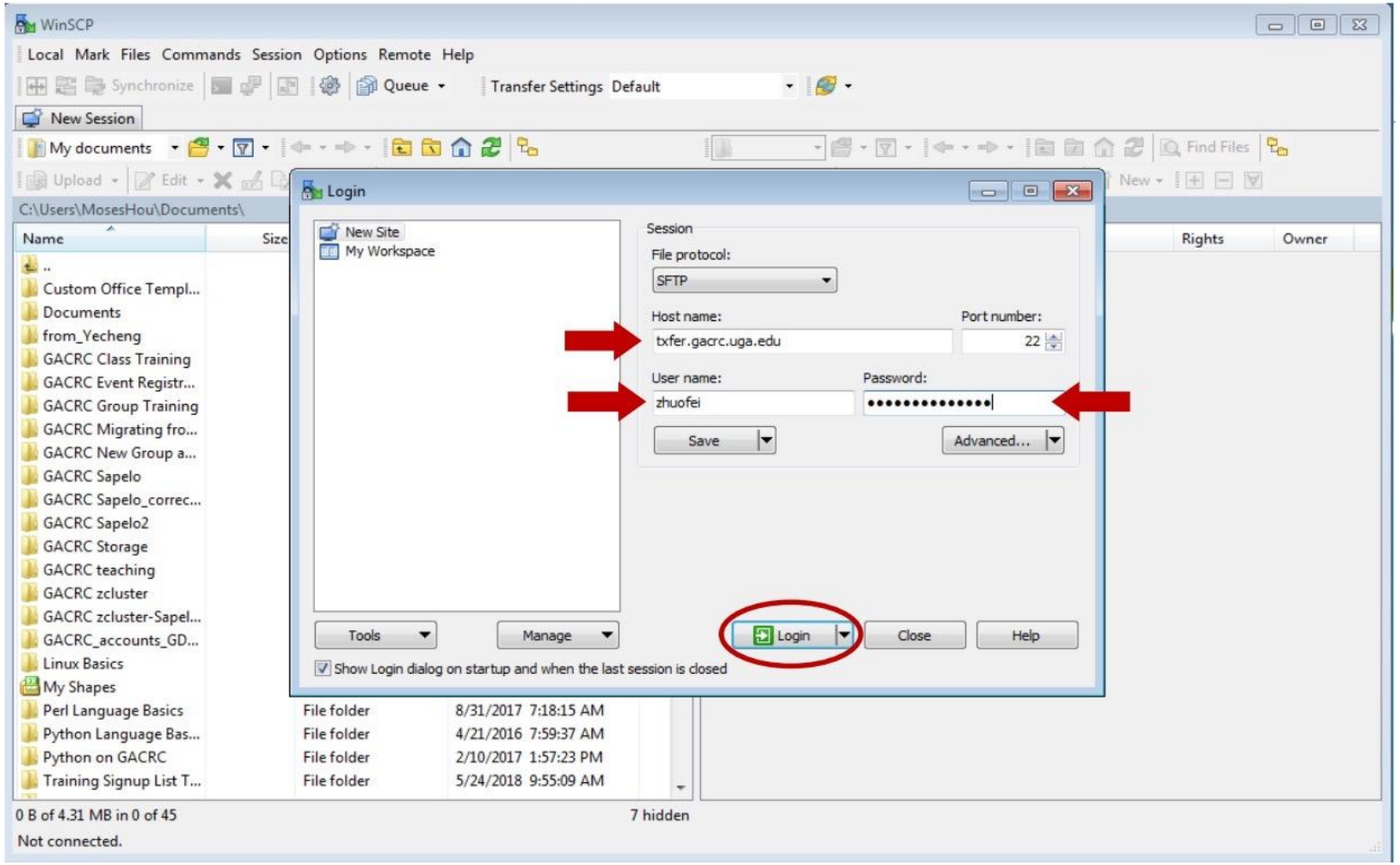
Ex 2: transfer a file from the Teaching Cluster onto your local computer

```
scp MyID@txfer.gacrc.uga.edu:/scratch/MyID/jobdir/file .
```

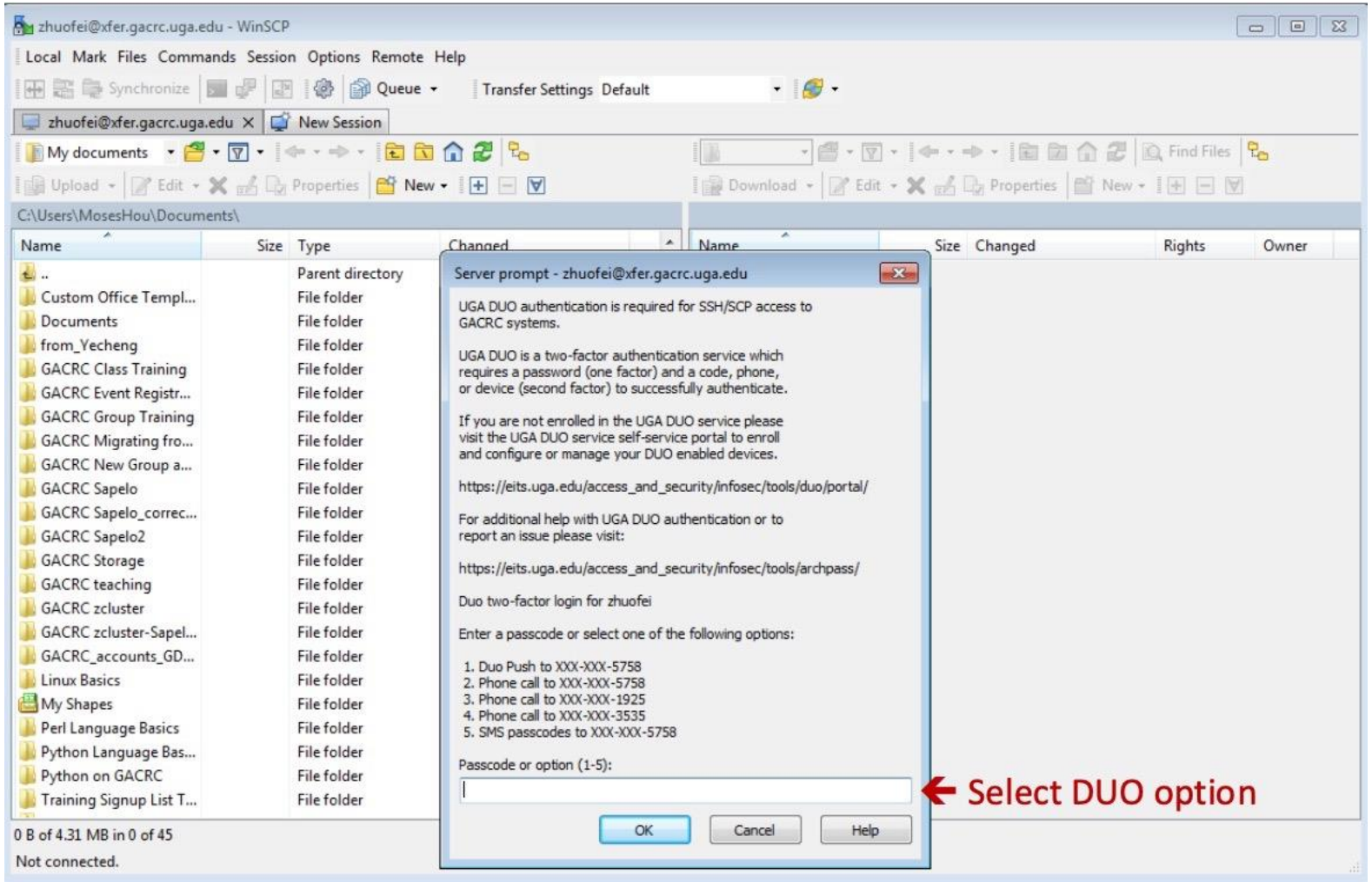
Step 5: Transfer data to the Cluster – Windows (1)

1. Download WinSCP on your own computer: https://winscp.net/eng/index.php
2. FileZilla is an alternative to WinSCP: https://wiki.gacrc.uga.edu/wiki/Transferring_Files#Using_FileZilla_2
3. Follow the steps in the following screenshots

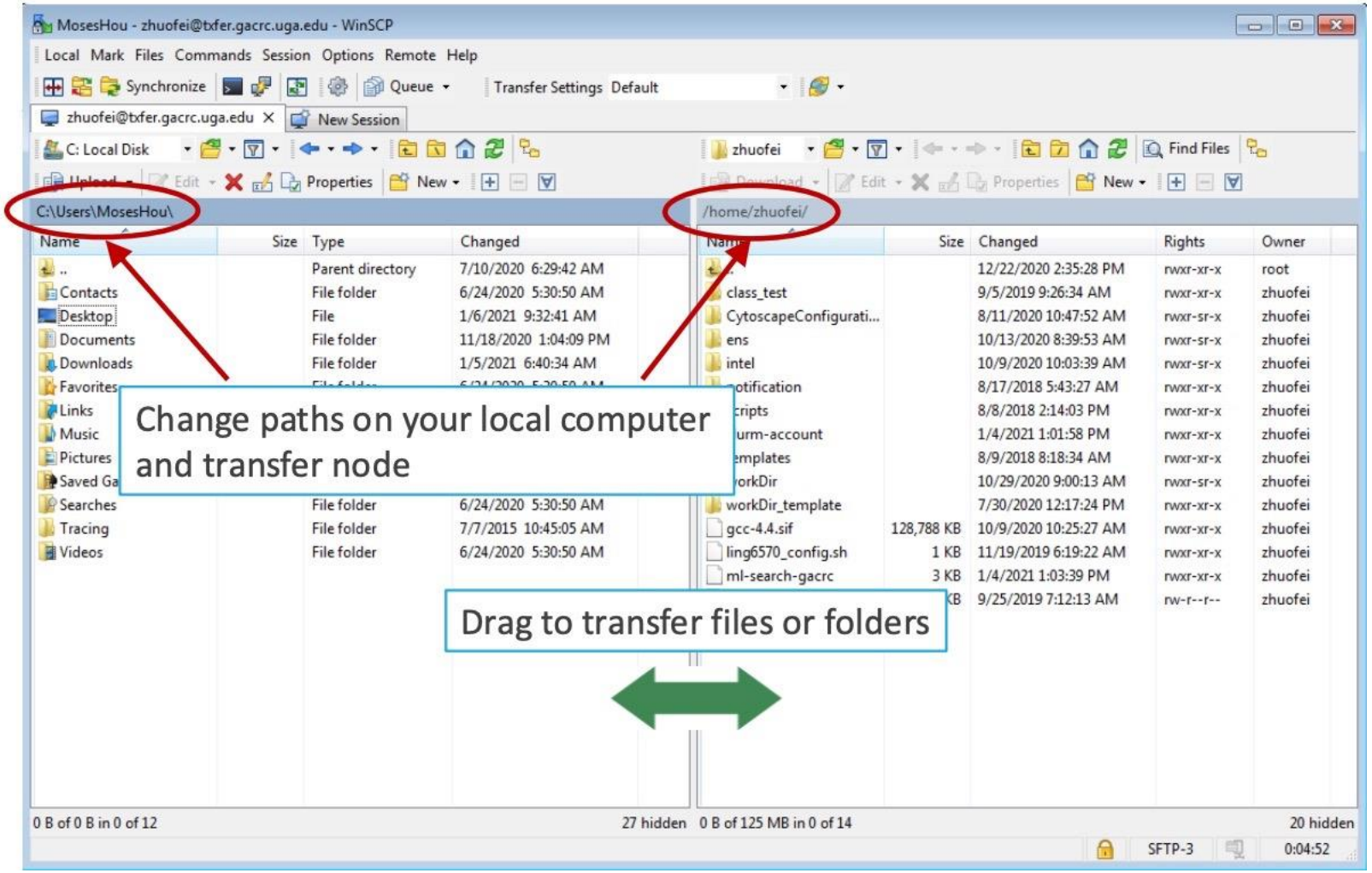
Step 5: Transfer data to the Cluster – Windows (2)



Step 5: Transfer data to the Cluster – Windows (3)



Step 5: Transfer data to the Cluster – Windows (4)



Step 6: Make a job submission script – GPU job

- ”nano” is a simple text editor in Linux. You are welcome to use others like “vim” or “emacs”
- Open the sub_gpu.sh file with a Linux text editor:

```
$ nano sub_gpu.sh
```

```
 GNU nano 5.6.1                                                    sub_gpu.sh
#!/bin/bash
#SBATCH --job-name=Amber_GPU                    # Job name
#SBATCH --partition=franklin_gpu                # Computational partition (franklin_gpu or gpu)
#SBATCH --gres=gpu:1                            # Request 1 GPU device
#SBATCH --ntasks=1                              # Run a single task
#SBATCH --cpus-per-task=4                       # Number of CPU cores per task
#SBATCH --mem-per-cpu=2gb                       # Memory per CPU core
#SBATCH --time=24:00:00                         # Time limit hrs:mins:secs
#SBATCH --output=log.%j.out                     # Standard output log
#SBATCH --error=log.%j.err                      # Standard error log
#SBATCH --mail-user=MyID@uga.edu                # Where to send mail
#SBATCH --mail-type=ALL                         # Mail events (BEGIN, END, FAIL, ALL)

cd $SLURM_SUBMIT_DIR                            # cd /scratch/MyID/workDir

ml Amber/24.3-foss-2022a-AmberTools-24.10-CUDA-12.1.1   # Load Amber module
source ${AMBERHOME}/amber.sh                    # Config software environment for Amber

# PMEMD: Job1: minimization, solvent
$AMBERHOME/bin/pmemd.cuda -O \
 -i min_solvent.in \
 -o min_solvent.out \
 -p gfp.parm7 \
 -c gfp.rst7 \
 -ref gfp.rst7 \
 -r gfp_min_solvent.rst7
```

Step 6: Make a job submission script – alternative job

- There is also an example of an MPI job which does **not** use GPU resources
- You can open sub_mpi.sh if you'd like with:

```
$ nano sub_mpi.sh
```

```
  GNU nano 5.6.1                                                    sub_mpi.sh
#!/bin/bash
#SBATCH --job-name=Amber_MPI                    # Job name
#SBATCH --partition=batch                       # Computational partition
#SBATCH --ntasks=10                             # MPI processes/rank number
#SBATCH --cpus-per-task=1                       # Number of CPU cores per task/MPI process
#SBATCH --mem-per-cpu=2gb                       # Memory per CPU core
#SBATCH --time=24:00:00                         # Time limit hrs:mins:secs
#SBATCH --output=log.%j.out                     # Standard output log
#SBATCH --error=log.%j.err                      # Standard error log
#SBATCH --mail-user=MyID@uga.edu                # Where to send mail
#SBATCH --mail-type=ALL                         # Mail events (BEGIN, END, FAIL, ALL)

cd $SLURM_SUBMIT_DIR                            # cd /scratch/MyID/workDir

ml Amber/24.3-foss-2022a-AmberTools-24.10-CUDA-12.1.1   # Load Amber module
source ${AMBERHOME}/amber.sh                    # Config software environment for Amber

# PMEMD: Job1: minimization, solvent
srun ${AMBERHOME}/bin/pmemd.MPI -O \
 -i min_solvent.in \
 -o min_solvent.out \
 -p gfp.parm7 \
 -c gfp.rst7 \
 -ref gfp.rst7 \
 -r gfp_min_solvent.rst7
```

Step 6: Make a job submission script – Slurm headers

- Submission scripts do not have to be complex!

This script works the same way as sub.sh, but with only required Slurm headers specified:

```
#!/bin/bash

#SBATCH --partition=batch
#SBATCH --ntasks=1
#SBATCH --mem=4G
#SBATCH --time=1:00:00

ml Bowtie2/2.4.1-GCC-8.3.0

bowtie2 -x index/lambda_virus -U myreads.fq
```

The only required Slurm headers are:

- --partition
- --ntasks
- --mem
- --time

(Slurm will let you know if you forgot one when you try to submit your submission script)

Some default values if not specified:

- --cpus-per-task=1
- --nodes=1 (this may not default to 1 if you have requested more than one process with ntasks)

Step 7: Submit your job

- Submission scripts can be submitted to the Cluster with the command sbatch
- Be sure to submit jobs from within your /scratch/MyID directory

```
zhuofei@teach-sub workDir$ pwd
/scratch/zhuofei/workDir
zhuofei@teach-sub workDir$ ls
index  myreads.fq  sub.sh
zhuofei@teach-sub workDir$ sbatch sub.sh
Submitted batch job 32860
```

← check current location

← view files in current location

← submit job

sub.sh is a job submission script to

1. specify computing resources:
2. load software using ml *moduleName*
3. run software

Step 8: Monitor Job with sq

- The command sq --me will show you information about your pending and running jobs

```
zhuofei@teach-sub1 workdir$ sq --me
JOBID    TIME    TIME_LIMIT   NAME          PARTITION    USER      NODES  CPUS   MIN_MEMORY  PRIORITY  STATE     NODELIST(REASON)
4618668  1:18    4:00:00      test-job      highmem_p    zhuofei   1      1      200G        5993      RUNNING   d4-11
4618666  1:21    1:00:00      Bowtie2-2cpu  batch        zhuofei   1      2      4G          5991      RUNNING   c1-23
4618665  5:25    2:00:00      testBowtie2   batch        zhuofei   1      1      4G          5991      RUNNING   c1-23
```

- Running sq --help will show you the help page for this command:

```
[cft07037@ss-sub3 workdir]$ sq --help

Usage: sq [OPTIONS]

Description: sq - preformatted wrapper for squeue.  See man squeue for more information.

    -j                          Displays squeue output for a given job
    --me                        Displays squeue output for the user executing this command
    -p                          Displays squeue output for a given partition
    -u                          Displays squeue output for a given user
    -T                          Displays submit and start time columns
    -h, --help                  Displays this help output
```

Step 8: Monitor Job with sacct-gacrc

- The command sacct-gacrc -X will show you information about your pending, running, and completed jobs

```
zhuofei@teach-sub1 workdir$ sacct-gacrc -X
     JobID         JobName      User  Partition NNode NCPUS   ReqMem    CPUTime    Elapsed  Timelimit      State ExitCode   NodeList
---------- --------------- --------- ---------- ----- ----- -------- ---------- ---------- ---------- ---------- -------- ----------
   4618312        interact   bc06026    inter_p     1     1      2Gn   00:07:01   00:07:01   12:00:00  COMPLETED      0:0       c4-20
   4618665     testBowtie2   bc06026      batch     1     1      4Gn   00:23:12   00:23:12   01:00:00  COMPLETED      0:0       c1-23
   4618666    Bowtie2-2cpu   bc06026      batch     1     2      4Gn   00:20:24   00:10:12   01:00:00  COMPLETED      0:0       c1-23
   4618668        test-job   bc06026  highmem_p     1     1    200Gn   00:09:46   00:09:46   04:00:00  COMPLETED      0:0       d4-11
```

- Running sacct-gacrc --help will show you the help page for this command:

```
[cft07037@ss-sub3 workdir]$ sacct-gacrc --help

Usage: sacct-gacrc [OPTIONS]

Description: preformatted wrapper for sacct.  See man sacct for more information.

    -E, --endtime              Display information about jobs up to a date, in the format of yyyy-mm-dd (default: now)
    -j, --jobs                 Display information about a particular job or jobs (comma-separated list if more than one job)
    -r, --partition            Display information about jobs from a particular partition
    -S, --starttime            Display information about jobs starting from a date in the format of yyyy-mm-dd (default: Midnight of today)
    -T                         Display the end time of a particular job or jobs
    -u, --user                 Display information about a particular user's job(s) (default: current user)
    -X, --allocations          Only show one line per job (do not display job steps)
    --debug                    Display the sacct command being executed
    --prio                     Display the priority of a particular job or jobs
    -h, --help                 Display this help output
```

Step 8: Monitor Job with sacct-gacrc options

- You can use the --starttime and --endtime options to modify the timeframe sacct-gacrc uses to show your jobs (default is midnight of previous day to your current time)

```
[cft07037@ss-sub1 ~]$ sacct-gacrc -X --starttime 2025-01-17 --endtime 2025-01-29
```

Step 8: Monitor Job with scancel

- You may cancel pending or running jobs with the command scancel JobID where you would replace JobID with your job's ID:

```
[cft07037@ss-sub1 workdir]$ scancel 34714806
```

- There is no way to undo cancelling a job, so use this command wisely!

Step 8: Monitor Job with seff

- You can check the resource usage of a completed job with seff JobID

```
[cft07037@ss-sub3 workdir]$ seff 34714807
Job ID: 34714807
Cluster: tc2
User/Group: cft07037/gacrc-appadmin
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 3
CPU Utilized: 00:13:48
CPU Efficiency: 98.22% of 00:14:03 core-walltime
Job Wall-clock time: 00:04:41
Memory Utilized: 493.74 MB
Memory Efficiency: 12.05% of 4.00 GB
```

← How many CPU cores were given to your job

← How well your job used all of its CPU cores

← How much memory your job actually used

← How much memory was requested

Monitoring jobs overview

- sq --me for details of pending and running jobs
- sacct-gacrc -X for details of pending, running, and completed jobs
- scancel to cancel a pending or running job
- seff to check the resource usage of a finished job

Interactive jobs - Overview

- Interactive jobs are an alternative way to use software on the Teaching cluster rather than using a submission script
- Interactive jobs will place you directly onto a compute node (from the login/submit node you were on originally)
- Pros:
  - Can load software directly from command line
  - Can run computations directly from command line
  - More hands-on and interactive than a submission script job
- Cons:
  - You must remain in the interactive job for the duration of your computations/processes (if you leave your interactive job, it will automatically end) so interactive jobs can be better for short jobs
    - Whereas submission script jobs (batch jobs) will continue to run in the background which makes them more versatile (for short or long jobs)

Interactive jobs – Starting a job

- To start an interactive job, run the command interact from a login/submit node
  - This will queue you up for an interactive job with default resources:

```
[cft07037@ss-sub3 workdir]$ interact

srun --pty  --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=2GB /bin/bash -l

[cft07037@c4-16 workdir]$
```

- To start an interactive job will specific resources, type the desired resource after the interact command. For example, to request 7GB of memory (rather than use the default 2GB):

```
[cft07037@ss-sub3 workdir]$ interact --mem=7GB

srun --pty  --cpus-per-task=1 --job-name=interact --ntasks=1 --nodes=1 --partition=inter_p --time=12:00:00 --mem=7GB /bin/bash -l

[cft07037@c4-16 workdir]$
```

- You can leave an interactive job to go back to the login/submit node with the command exit
- You can type interact --help to see all of the options for this command

Helpful GACRC Links

- GACRC Wiki http://wiki.gacrc.uga.edu
- Kaltura channel (GACRC video trainings) https://kaltura.uga.edu/channel/GACRC/176125031
- Connecting https://wiki.gacrc.uga.edu/wiki/Connecting#Connecting_to_the_teaching_cluster
- Software https://wiki.gacrc.uga.edu/wiki/Software_on_Sapelo2
- Running Jobs https://wiki.gacrc.uga.edu/wiki/Running_Jobs_on_the_teaching_cluster
- Monitoring Jobs https://wiki.gacrc.uga.edu/wiki/Monitoring_Jobs_on_the_teaching_cluster
- Sample submission scripts
  https://wiki.gacrc.uga.edu/wiki/Sample_batch_job_submission_scripts_on_the_teaching_cluster
- Linux Commands Help https://wiki.gacrc.uga.edu/wiki/Command_List
- Open OnDemand https://wiki.gacrc.uga.edu/wiki/OnDemand

GACRC Ticketing Help/Support

https://uga.teamdynamix.com/TDClient/2060/Portal/Requests/ServiceCatalog?CategoryID=18080

- For Job Troubleshooting help:
  - Please include as many details as you can including:
    - Your UGA MyID
    - Your JobID
    - The full path to your submission script
    - The command you use to submit the job
- For Software Installations:
  - Include
    - Specific name and version of software
    - Download website

File Storage Guidelines

- No directory should have too many files inside! A good rule of thumb is 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