BIOS Virtual Workshop Introduction to the new RSPH HPC Cluster

November 20, 2020

Instructor: Jingchao Zhang

Moderators: Ying Guo, David Benkeser

Schedule

- Overview of the RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

Teaching Objectives

- Login to RSPH cluster using PuTTY(Windows)/Terminal(Mac).

- File management on the cluster.

- Use modules (software) installed on the clusters.

- Submit jobs using a job scheduler.

Logistics

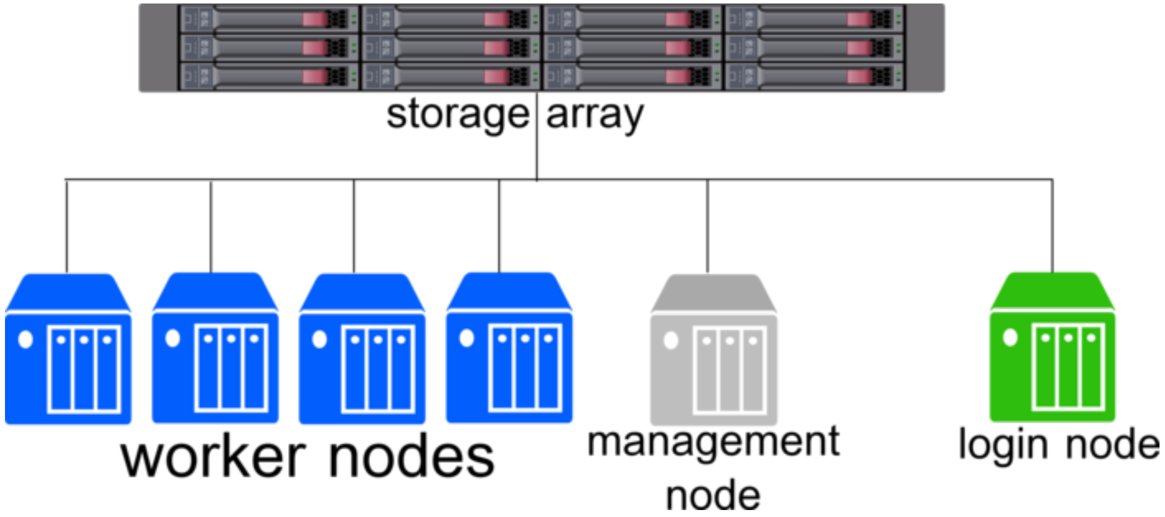
- During Session, all attendees are muted by default.
  - Please stay muted during talks.
  - During the Q&A, you are welcome to unmute yourself and ask questions.
- Feel free to add questions and/or comments to the chat at any time.
  - Co-hosts will answer Q's and/or ask speaker during session break.
- You need access to the RSPH cluster for hands-on exercises during the workshop.

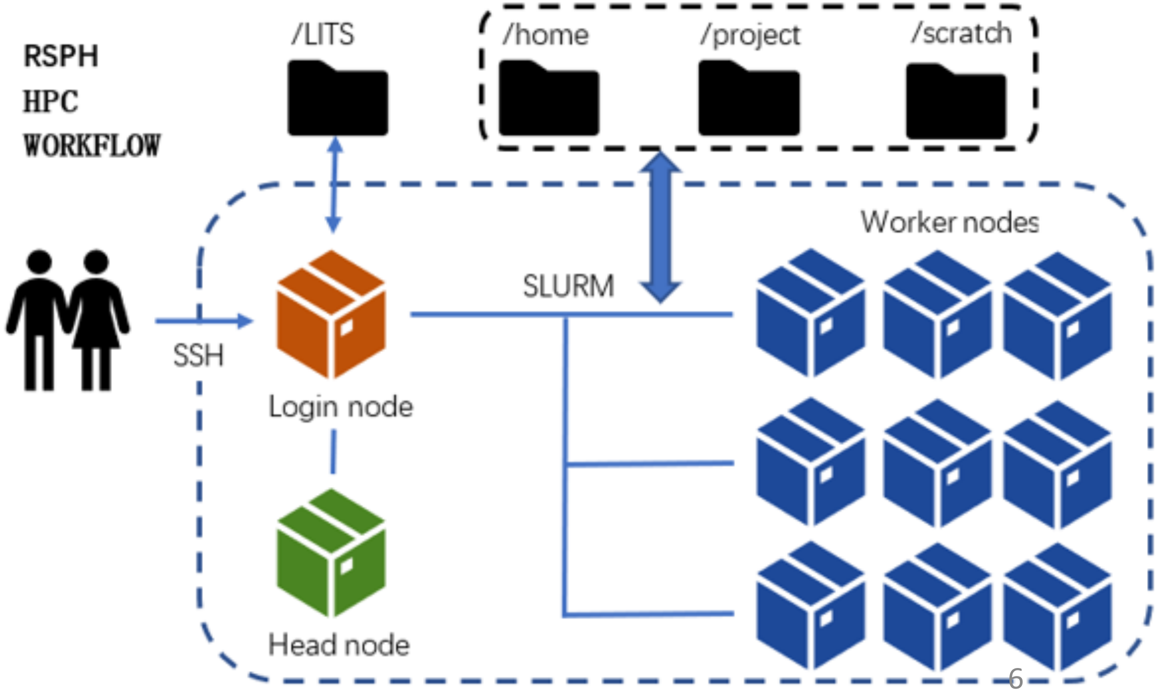
The success of the virtual workshop depends on all of us working together within the limitations of being virtual.

Schedule

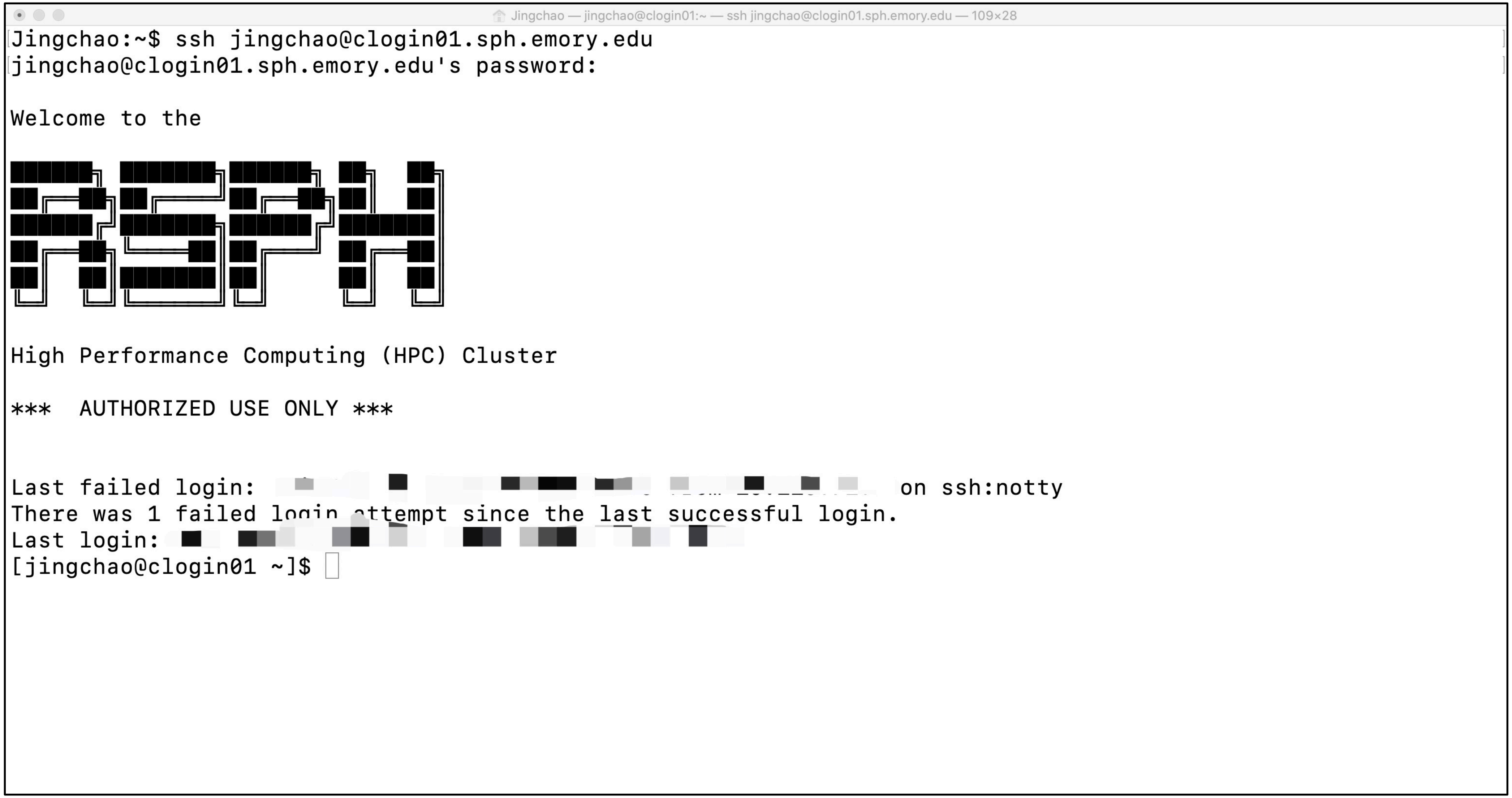
- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

What is a Cluster?





Command line interface (CLI)


Jingchao — jingchao@clogin01:~ — ssh jingchao@clogin01.sph.emory.edu — 109×28

```
Jingchao:~$ ssh jingchao@clogin01.sph.emory.edu
jingchao@clogin01.sph.emory.edu's password:

Welcome to the

██████╗ ███████╗██████╗ ██╗  ██╗
██╔══██╗██╔════╝██╔══██╗██║  ██║
██████╔╝███████╗██████╔╝███████║
██╔══██╗╚════██║██╔═══╝ ██╔══██║
██║  ██║███████║██║     ██║  ██║
╚═╝  ╚═╝╚══════╝╚═╝     ╚═╝  ╚═╝

High Performance Computing (HPC) Cluster

***  AUTHORIZED USE ONLY ***


Last failed login: [redacted] on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: [redacted]
[jingchao@clogin01 ~]$ 
```

clogin01

- 1 login node
- 1 management/head node
- 25 compute node, 32 cores each
- 24 nodes have 192 GB RAM
- 1 node has 1.5 Tb RAM
- 1 Pb shared Panasas storage
- 25GigE Mellanox Ethernet

- CPU specs:
  - Intel Xeon 6242 "Cascade Lake-SP" 2.8 GHz 16-core 14nm CPU
- Memory Specs:
  - 16GB DDR4 2933 MHz ECC
  - 192GB Total Memory @ 2933MHz
  - 6GB memory per core

Total Resources (growing)

800 cores

Approximately 1 PetaBytes of storage

192 GB to 1.5 Terabyte memory per node

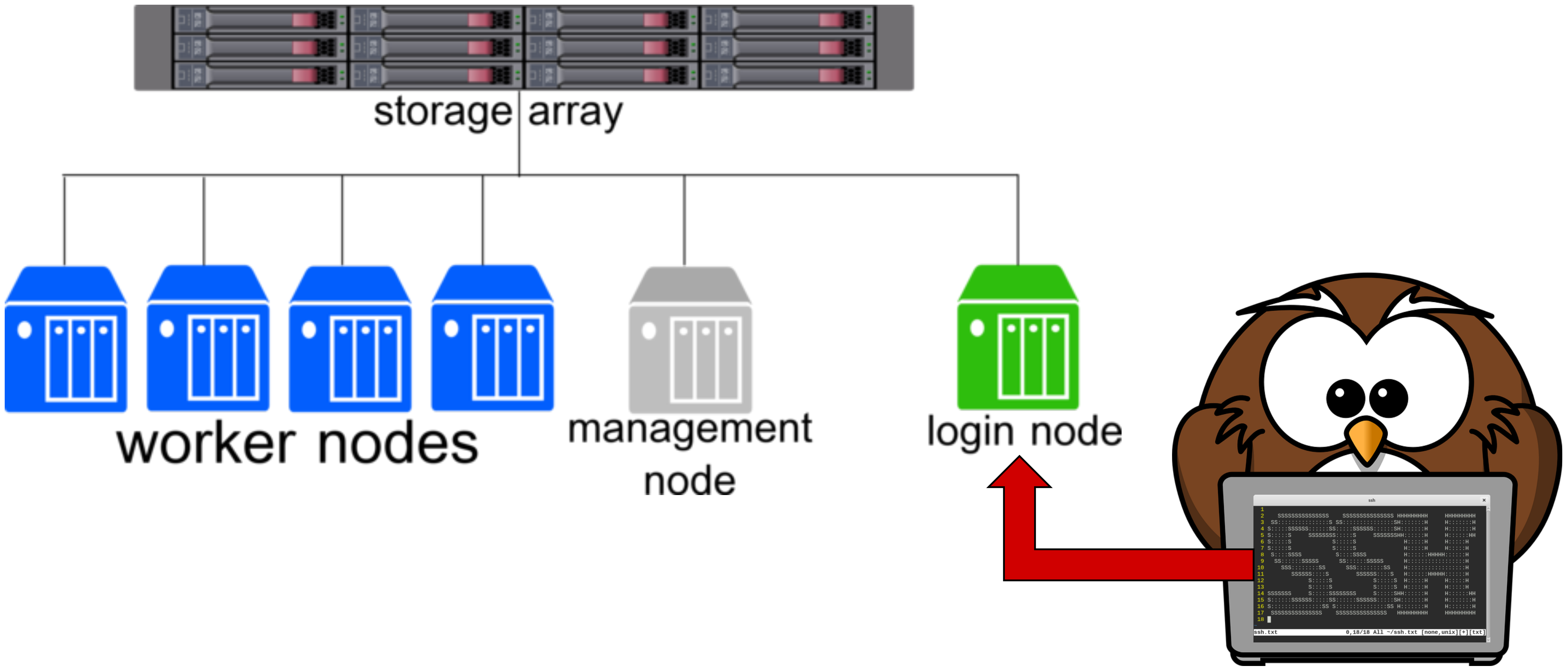
FREE! FREE! FREE!

- Access to the HPC cluster is free for all members of the RSPH community.
- All RSPH faculty may request access to the cluster for themselves, and non-faculty may request accounts via sponsorship by an RSPH faculty member.
- Accounts are requested by emailing “help@sph.emory.edu".
- Fees are requires only for dedicated computing resources and extra storage beyond the free quota limit.

Schedule

- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

Connecting to the Clusters



```
   SSSSSSSSSSSSSSS    SSSSSSSSSSSSSSS HHHHHHHHH     HHHHHHHHH
 SS:::::::::::::::S SS:::::::::::::::SH:::::::H     H:::::::H
S:::::SSSSSS::::::SS:::::SSSSSS::::::SH:::::::H     H:::::::H
S:::::S     SSSSSSSS:::::S     SSSSSSSHH::::::H     H::::::HH
S:::::S            S:::::S              H:::::H     H:::::H
S:::::S            S:::::S              H:::::H     H:::::H
 S::::SSSS          S::::SSSS           H::::::HHHHH::::::H
  SS::::::SSSSS       SS::::::SSSSS     H:::::::::::::::::H
    SSS::::::::SS       SSS::::::::SS   H:::::::::::::::::H
       SSSSSS::::S         SSSSSS::::S  H::::::HHHHH::::::H
            S:::::S             S:::::S H:::::H     H:::::H
            S:::::S             S:::::S H:::::H     H:::::H
SSSSSSS     S:::::SSSSSSSS     S:::::SHH::::::H     H::::::HH
S::::::SSSSSS:::::SS::::::SSSSSS:::::SH:::::::H     H:::::::H
S:::::::::::::::SS S:::::::::::::::SS H:::::::H     H:::::::H
 SSSSSSSSSSSSSSS    SSSSSSSSSSSSSSS   HHHHHHHHH     HHHHHHHHH

ssh.txt                    0,18/18 All ~/ssh.txt [none,unix][+][txt]
```

How to connect

VPN

- To connect the the HPC cluster, one first requires access to the Emory VPN HIPAAcore.
- All users can self manage access to the general VPN by following the instructions at (http://it.emory.edu/vpntools/access.html).
- Once the VPN has been configured by the user and a general connection has been successfully made, the secondary access to the HIPAA-core will be granted that will allow access to the HPC cluster login node. This access is requested on behalf of the user from LITS.

Before your login to the RSPH HPC, please make sure you have connected Emory VPN. https://vpn.emory.edu

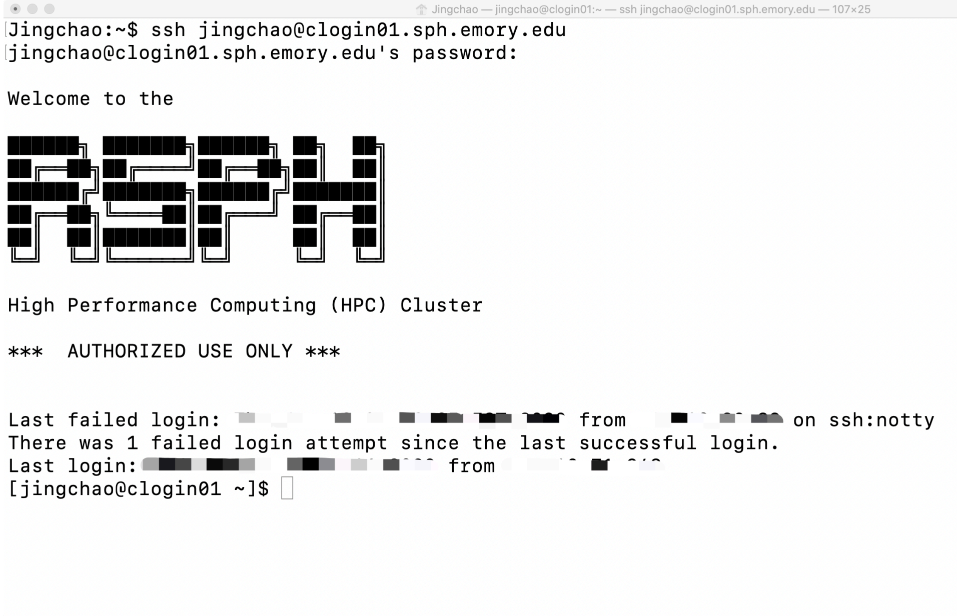
How to connect

MacOS / Linux

- Open Terminal
- Type in the following command and press Enter:

  `ssh <user_name>@clogin01.sph.emory.edu`

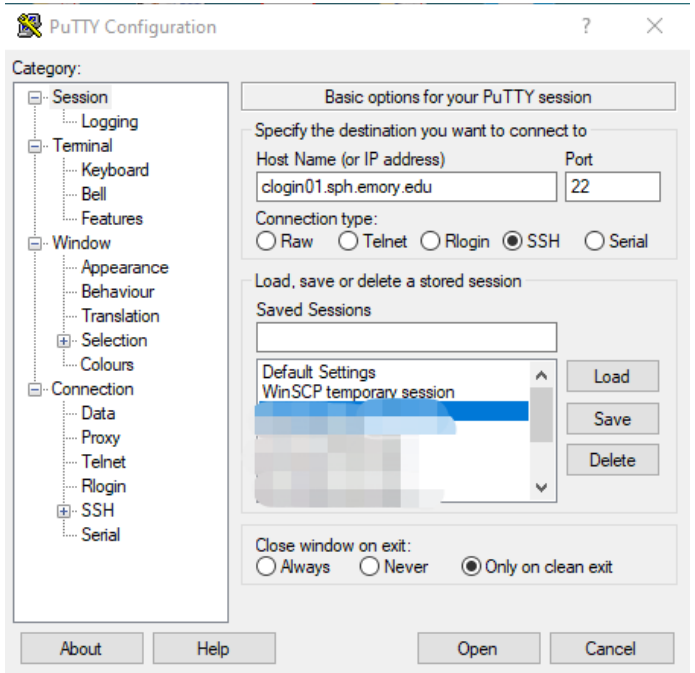
  (Replace <user_name> with your RSPH login)



```
Jingchao — jingchao@clogin01:~ — ssh jingchao@clogin01.sph.emory.edu — 107×25
Jingchao:~$ ssh jingchao@clogin01.sph.emory.edu
jingchao@clogin01.sph.emory.edu's password:

Welcome to the



High Performance Computing (HPC) Cluster

***  AUTHORIZED USE ONLY ***


Last failed login: [redacted] from [redacted] on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: [redacted] from [redacted]
[jingchao@clogin01 ~]$ 
```

Windows

- Open PuTTY (direct download link)
- Type **`clogin01.sph.emory.edu`** for Host Name and click Open



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): clogin01.sph.emory.edu

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
WinSCP temporary session
[redacted]

Close window on exit:
○ Always ○ Never ● Only on clean exit

- On the second screen, click Yes

Exercises

1. Connect to the RSPH cluster

   (hostname: `clogin01.sph.emory.edu`)

Shell Command Review (link)

Command	What it does	Example Usage
ls	list: Lists the files and directories located in the current directory	**ls** • Lists all files and directories in the current directory
cd	change directory: this allows users to navigate in or out of file directories	**cd <dir_name>** • Navigates into the directory “dir_name”
pwd	print working directory: print name of current/working directory	[jzhan61@clogin01 ~]$ **pwd** /home/jzhan61
rm	rm - remove files or directories	**rm <fileName>** • permanently delete the file “**fileName**” **rm –r <dirName>** • permanently delete the folder “**dirName**” and its content
nano	nano text editor: opens the nano text editor Note: To access the menu options, ^ indicates the control (CTRL) key.	**nano <file_name>** • opens the text editor with “file_name” open. If “file_name” does not exist, it will be created if the file is saved
less	less: opens an extended view of a file Note: scroll using up and down arrows. To exit, press 'q'	**less <file_name>** • opens an extended view of the file "file_name"
man	man - an interface to the reference manuals	**man <CMD>** • opens the manual page of the for command ”**CMD**"

Schedule

- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

Home vs Project vs Scratch vs Isilon

Home - /home/user

- Quota-limited to 25GB *per user*
- **Daily snapshots for best-effort data-loss recovery**
  - /home/.snapshot
- Meant for items that take up less space.
  - Source code
  - Program binaries
  - Configuration files
  - Small jobs

Project - /work/group

- Designed for group projects
- 1 TB quota *per group*
- **No snapshots**
- **Purge policy**
  - Files will be deleted after 1 year inactivity
- Additional storage can be purchased $75/TB/year

Home vs Project vs Scratch vs Isilon

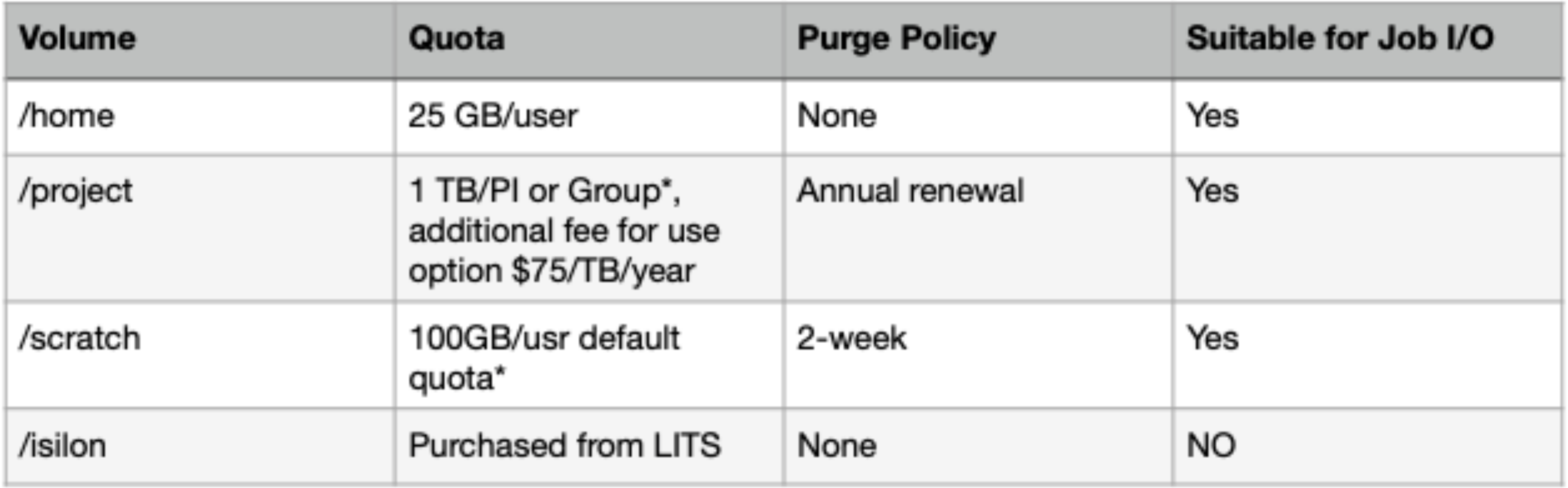
Scratch - /scratch/user

- Created upon request
- 100 Gb quota *per* user
- **No snapshots**
- **Purge policy**
  - Files will be deleted after 2-week inactivity
- Quota can be temporary increased for large jobs

Isilon - /isilon

- Can only be access on login node
- Designed for archiving
- Governed by LITS policy

Home vs Project vs Scratch vs Isilon



Volume	Quota	Purge Policy	Suitable for Job I/O
/home	25 GB/user	None	Yes
/project	1 TB/PI or Group*, additional fee for use option $75/TB/year	Annual renewal	Yes
/scratch	100GB/usr default quota*	2-week	Yes
/isilon	Purchased from LITS	None	NO

Home vs Project vs Scratch vs Isilon

Best Practices

- Home vs Project vs Scratch are all part of the Panasas parallel file system, so there is no performance difference.
- Take advantage of snapshot feature of Home for data recovery.
- For large jobs that do not fit in Home and Project, request for temporary space on Scratch.
- Millions of small files will slow down the filesystem. Zip or tar them into one file when unused.
- Home vs Project vs Scratch are NOT BACKED UP. Data loss can occur when hard drive fails. Regularly backup important files.

Transferring Files

- **Transfer files using an SCP client**
    - WinSCP (https://winscp.net)
        - Documentation page
    - Cyberduck (https://cyberduck.io)
    - FileZilla (https://filezilla-project.org/) – Windows, MacOS and Linux
        - Documentation Page
- **`rsync or scp`**
    - Usage: **`rsync -avr user@host:source_file user@host:target_file`**
    - Example:

```
rsync my_file.txt USER@clogin01.sph.emory.edu:/home/jzhan61
```

Migrating Files from Old Cluster to The New One

1. Login to the new RPSH cluster (clogin01.sph.emory.edu)

2. On the login node, check if you can find your files in **/isilon/home** and **/isilon/projects**

   - If yes, then you can directly copy them from /isilon/home or /isilon/projects to your /home or /project. For instance: cp /isilon/home/jzhan61/* /home/jzhan61

3. If you cannot find your files in /isilon, use **rsync** to migrate files from the old cluster to the new one.

   - For example, rsync –avr jzhan61@hpc5.sph.emory.edu:/home/jzhan61/* /home/jzhan61
   - rsync supports file transfer resumption after interruptions. To resume an interrupted file transfer, use rsync –avr --append.

Refer to this link for more details:

https://scholarblogs.emory.edu/rsph-hpc/migrating-files-from-the-old-cluster-to-the-new-one/

Exercises

1. On your local computer create a file called ‘bio.txt’ – edit this file to include your name and department.

2. If you are using a Windows laptop, download and setup WinSCP.
   Or if you are using a Mac, download and setup FileZilla.
   - WinSCP instructions
   - FileZilla instructions

3. Copy the ‘bio.txt’ file from your local laptop to the cluster’s Home.
   (Mac/Linux users can use scp/rsync if preferred)

4. Clone the tutorial files to your /home directory by entering the command:
   - cd $HOME
   - git clone https://github.com/JingchaoZhang/BIOS-job-examples.git

Schedule

- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

Running Applications

- All applications installed on RSPH clusters are loaded in individual modules implemented with Lmod
  - Lua based module system

- Modules dynamically change the user environment
  - $PATH
  - $LD_LIBRARY_PATH

- Typically follows the naming convention <software>/<version>
  - Example: python/2.7

- Load using the **`module`** command

Lmod Commands

Command	What it does
`module available`	Lists all modules available to be loaded
`module spider <name>`	Information about a specific module – can also be used to search
`module load <module_name>`	Load module(s) – can load a list of space delimitated modules
`module unload <module_name>`	Unload module(s) – can unload a list of space delimitated modules
`module purge`	Unloads all currently loaded modules
`module list`	Lists all currently loaded modules

For more information:

- `module --help`

Software installation

What to do when a software or package you need is not on the cluster?

- R/Python packages:
  - Users can install R and Python packages in their home directory.
    - For R, use command install.packages('PACKAGE') within the R shell. (Example: vegan) Tutorial link
    - For Python, use command pip install PACKAGE --user within the Python shell. (Example: matplotlib)
- Using Conda package manager:
  - conda can be used to create isolated environments containing thousands of packages at repo.anaconda.com Tutorial link
- Create your own modules:
  - Yes, users can create their own module files and load them the same way as system installed modules Tutorial link
- To get a software installed system wide, send a ticket to 'help@sph.emory.edu' and mention "HPC cluster" in the subject line of your request

Exercises

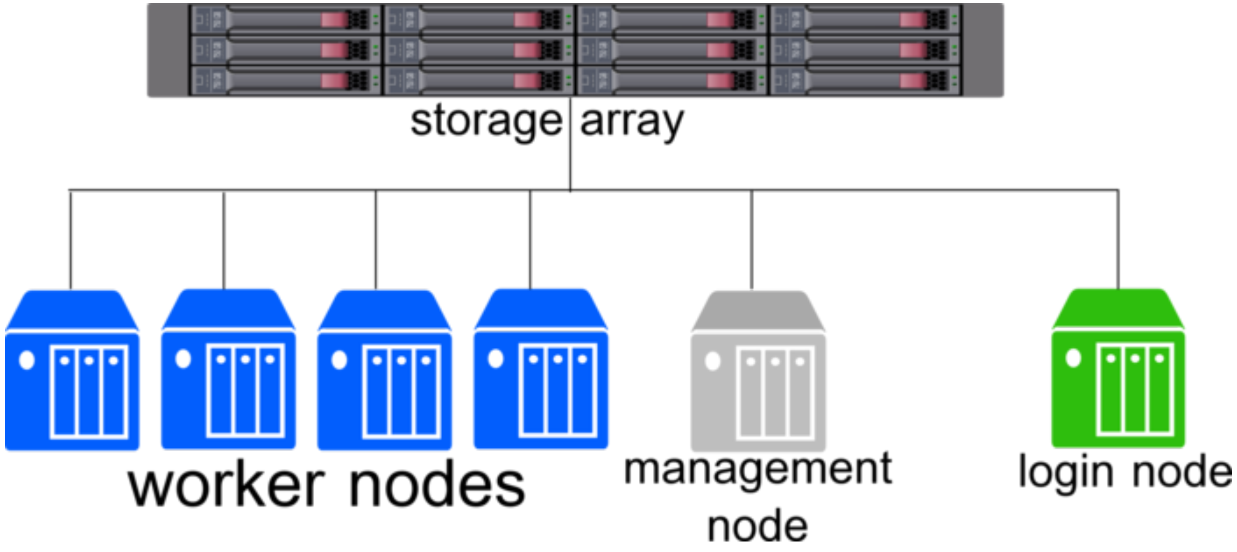
1. Load MATLAB/R2020a, R/4.0.2, python/3.8 in one command
2. Unload all modules in one command
3. Load the module R/4.0.3 and install a missing package named ‘bayestestR’.
    - After installation, type ‘library(bayestestR)’ to confirm the package can be loaded.

- Bonus: Download and install miniconda in your /home directory. Refer to this tutorial.

Schedule

- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A

Running Jobs



- All calculation and analysis must be done on the worker nodes
- Processes started on the login node will be killed
    - Limit usage to brief, non-intensive tasks like file management and editing text files
- RSPH uses the SLURM scheduler to manage and allocate resources
    - Resources are allocated based on the Fair Use Algorithm
- Jobs can be run in interactive or batch format

Interactive vs Batch Jobs

Batch Jobs

- The user creates a **submit file** which contains all commands and job information and add its to the queue
    - Similar to running Bash scripts
- Holds job information until resources become available
    - User can disconnect and the job will remain queued
- Uses the `sbatch` command

Interactive Jobs

- Allows the user to type in commands and run programs **interactively**
- Must remain connected and wait for resources to be allocated
- Job can be interrupted
- Uses the `srun` command

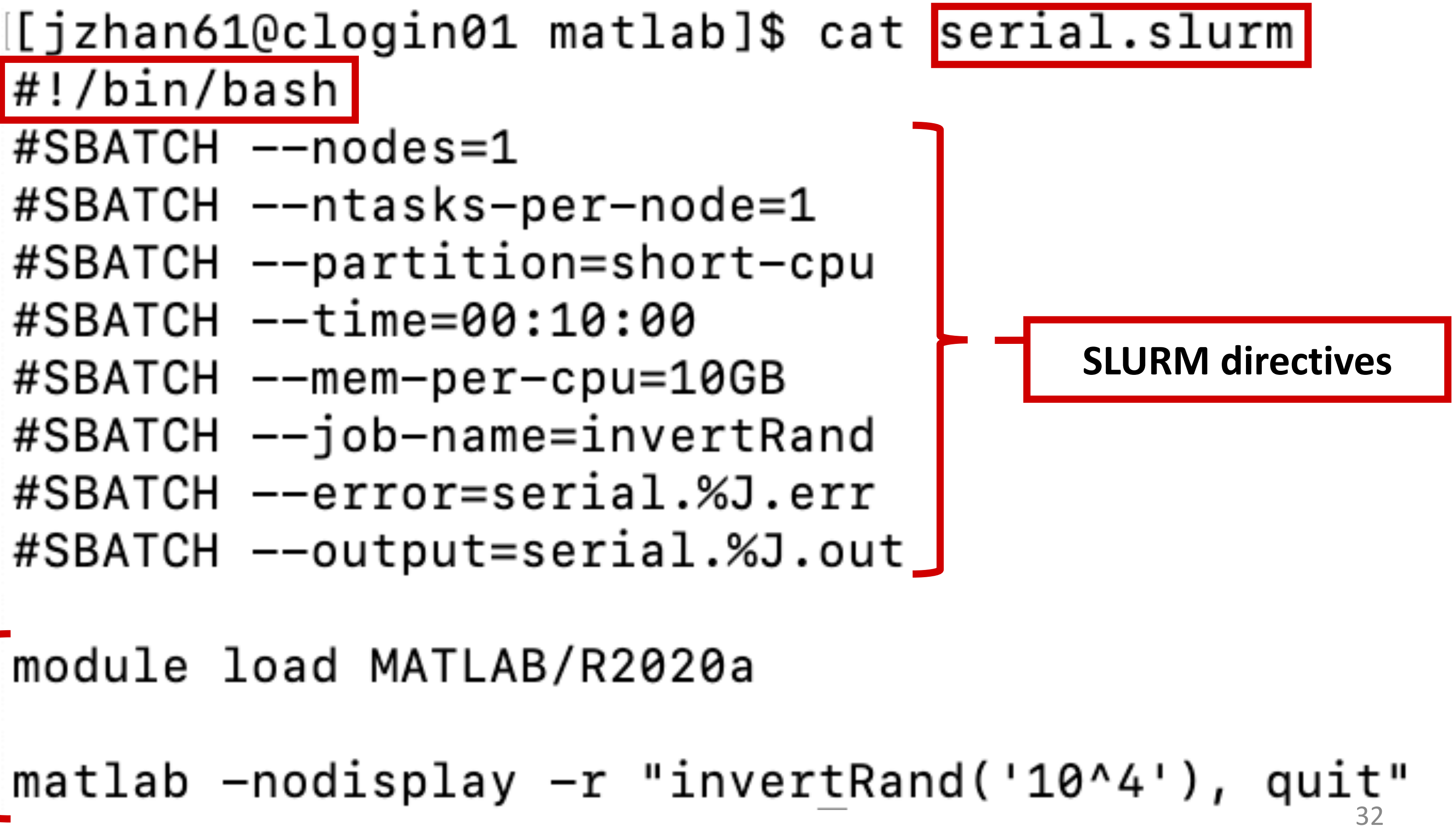
Batch Jobs

- To create a batch job, the user must first make a submit script
  - Submit scripts include all job resource information:
    - number of nodes/cores
    - required memory
    - Runtime
  - **If the job exceeds the requested memory or time, it will be killed.**
- Submit script is then added to the job queue using the **`sbatch`** command
- **`squeue`** will show queued and running jobs
- **`sacct`** can be used to find information about completed jobs

Submit Scripts

Name of the submit file
This can be anything. Here we are using "serial.slurm" the .slurm makes it easy to recognize that this is a submit file.

Shebang
The shebang tells Slurm what interpreter to use for this file. This one is for the shell (Bash)



```
[jzhan61@clogin01 matlab]$ cat serial.slurm
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=short-cpu
#SBATCH --time=00:10:00
#SBATCH --mem-per-cpu=10GB
#SBATCH --job-name=invertRand
#SBATCH --error=serial.%J.err
#SBATCH --output=serial.%J.out

module load MATLAB/R2020a

matlab -nodisplay -r "invertRand('10^4'), quit"
```

SLURM directives

Commands
Any commands after the SBATCH lines will be executed by the interpreter specified in the shebang – similar to what would happen if you were to type the commands interactively

Common SBATCH Options

Command	What it does
`--nodes`	Number of nodes requested
`--ntasks-per-node`	Number of tasks per node – used to request a specific number of cores
`--mem`	Real memory (RAM) required per node - can use KB, MB, and GB units – default is MB Request less memory than total available on the node - The maximum available on a 192 GB RAM node is 186 GB
`--mem-per-cpu`	required per allocated core
`--time`	Maximum walltime for the job – in DD-HHH:MM:SS format
`--output`	Filename where all STDOUT will be directed – default is slurm-<jobid>.out
`--error`	Filename where all STDERR will be directed – default is slurm-<jobid>.out
`--job-name`	How the job will show up in the queue

For more information:

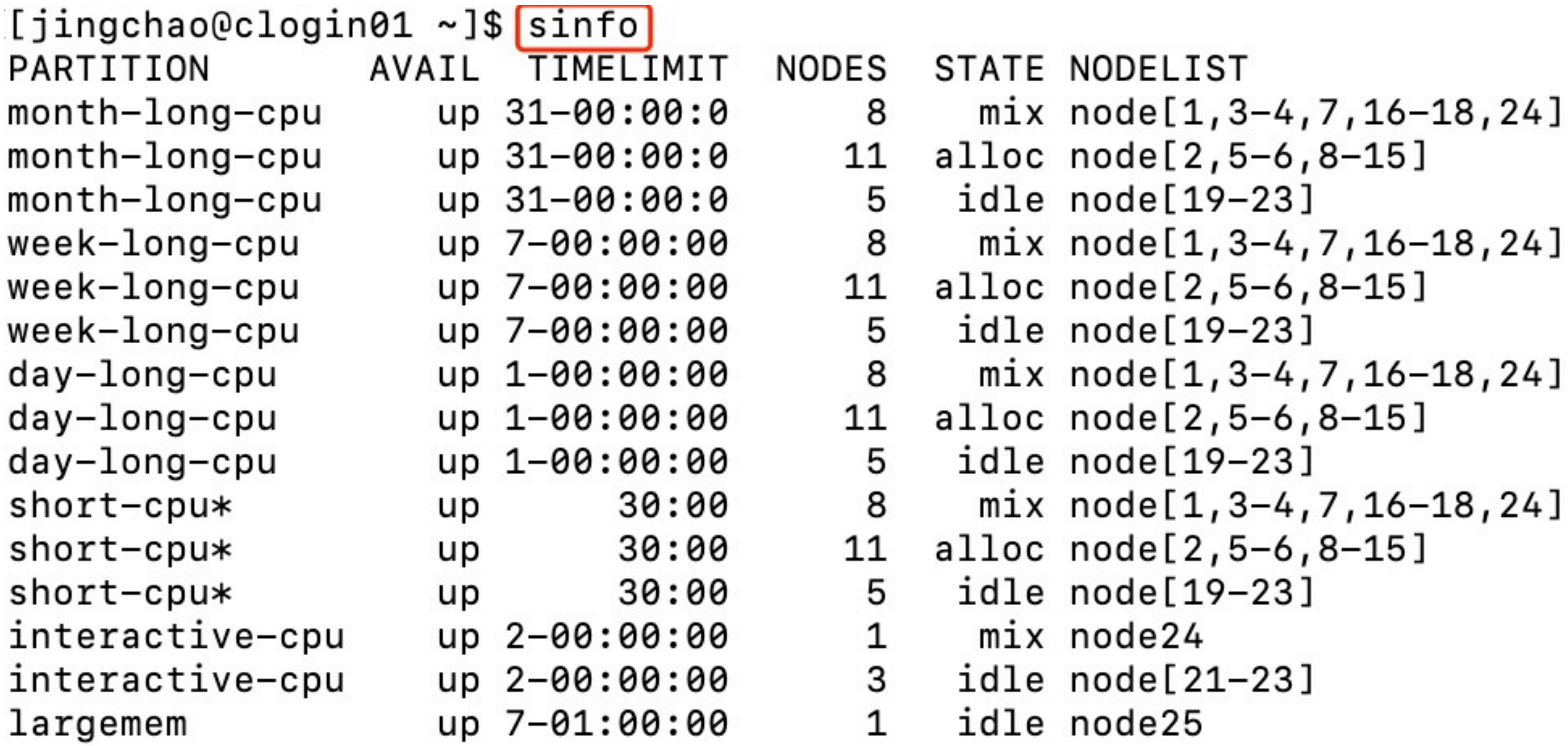
- `sbatch –help`
- SLURM Documentation: https://slurm.schedmd.com/sbatch.html

Available Partitions

Partition Name	Time Limit
short-cpu	30:00
day-long-cpu	1-00:00:00
week-long-cpu	7-00:00:00
month-long-cpu	31-00:00:00
interactive-cpu	2-00:00:00
largemem	7-01:00:00

- Fairshare: different partitions have different priority values.
  Jobs submitted to shorter queues have higher priorities, meaning less waiting time in the queue.

Available Partitions



```
[jingchao@clogin01 ~]$ sinfo
PARTITION          AVAIL  TIMELIMIT  NODES  STATE NODELIST
month-long-cpu        up 31-00:00:0      8    mix node[1,3-4,7,16-18,24]
month-long-cpu        up 31-00:00:0     11  alloc node[2,5-6,8-15]
month-long-cpu        up 31-00:00:0      5   idle node[19-23]
week-long-cpu         up 7-00:00:00      8    mix node[1,3-4,7,16-18,24]
week-long-cpu         up 7-00:00:00     11  alloc node[2,5-6,8-15]
week-long-cpu         up 7-00:00:00      5   idle node[19-23]
day-long-cpu          up 1-00:00:00      8    mix node[1,3-4,7,16-18,24]
day-long-cpu          up 1-00:00:00     11  alloc node[2,5-6,8-15]
day-long-cpu          up 1-00:00:00      5   idle node[19-23]
short-cpu*            up      30:00      8    mix node[1,3-4,7,16-18,24]
short-cpu*            up      30:00     11  alloc node[2,5-6,8-15]
short-cpu*            up      30:00      5   idle node[19-23]
interactive-cpu       up 2-00:00:00      1    mix node24
interactive-cpu       up 2-00:00:00      3   idle node[21-23]
largemem              up 7-01:00:00      1   idle node25
```

Determining Parameters

How many nodes/memory/time should I request?

- **Short answer:** We don’t know.
- **Long answer:** The amount of time and memory required is highly dependent on the application you are using, the input file sizes and the parameters you select.
  - Sometimes it can help to speak with someone else who has used the software before.
  - Ultimately, it comes down to trial and error
    - Check the output and utilization of each job will help you determine what parameters you will need in the future.
    - Trying different combinations and seeing what works and what doesn’t.

Submit Files Best Practices

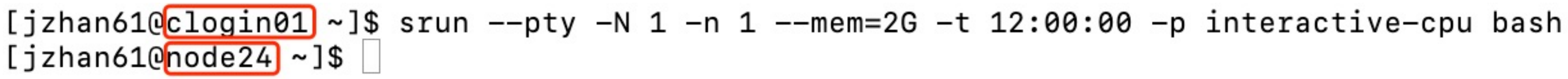
- **Put all module loads immediately after SBATCH lines**
  - Quickly locate what modules and versions were used.
- **Specify versions on module loads**
  - Allows you to see what versions were used during the analysis
- **Use a separate submit file for each job**
  - Instead of editing and resubmitting a submit files – copy a previous one and make changes to it
  - Keep a running record of your analysis
- **Redirect output and error to separate files**
  - Allows you to see quickly whether a job completes with errors or not
- **Separate individual workflow steps into individual jobs**
  - Avoid putting too many steps into a single job

Exercises

1. Navigate into the matlab directory of the BIOS-job-examples directory and locate the serial.slurm file
   - This submit file runs a MATLAB script which inverts a 10,000 x 10,000 randomly generated matrix and outputs the length of time it took to perform the inversion.
   - Look at the contents of serial.slurm - How many nodes this will run on? How many cores? How much memory and time is requested?
   - Submit the serial.slurm job. Check the output to see how long it took to invert the matrix.
2. Locate the parallel.slurm file.
   - Submit the job again and compare the time to your initial run. How much faster or slower is it?
   - Compare times with others. Did you see the same amount of improvement?
3. If there’s time, try different combinations of SBATCH commands and see how the running time changes.

Refer to slide 21 exercise 4 if you cannot find the BIOS-job-examples directory.

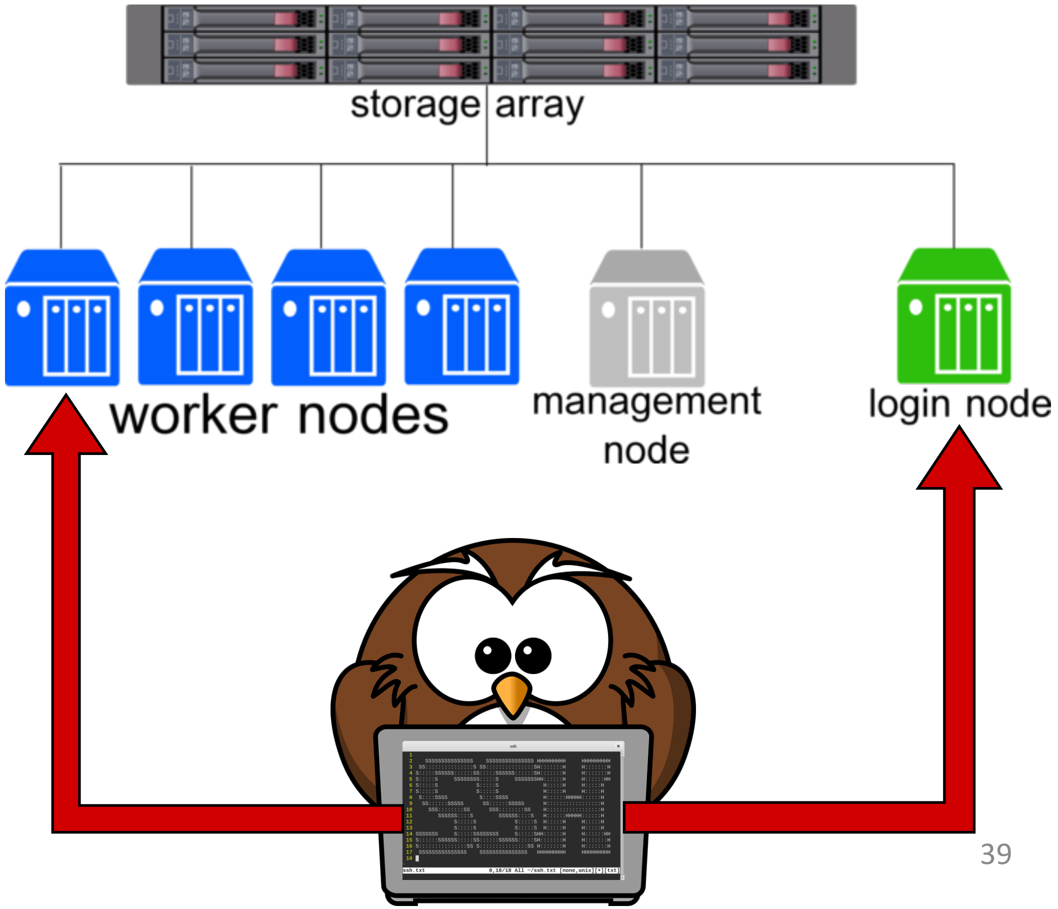
Interactive Jobs



```
[jzhan61@clogin01 ~]$ srun --pty -N 1 -n 1 --mem=2G -t 12:00:00 -p interactive-cpu bash
[jzhan61@node24 ~]$ 
```

- Interactive jobs work very similarly to batch jobs

- Once resources are allocated, commands can be input interactively
  - All output is directed to the screen



Exercises

1. Request an interactive job for certain amount of resources
   - If you can’t think of a setup, use one of these:
     - 1 node, 1 core with 2 GB RAM each core
     - 2 nodes, 1 core per node with 2 GB RAM each core
   - How long did it take to allocate resources? Compare your results with others.

2. Using your interactive job from above, load the module for your favorite programming language (Python, R, MATLAB, etc.)
   - Run the program interactively

Asking for help

- *When your job fails, look for error messages in your error or output file. Google is your friend!*
- Read Documentations
  1. RSPH HPC Official Documentation
  2. BIOS HPC Website (report issues to jingchao.zhang@emory.edu)
- Need more help?
  1. If you are from BIOS, contact me at jingchao.zhang@emory.edu
  2. Create a ticket by sending an email to help@sph.emory.edu and mention “HPC cluster” in the subject line

Schedule

- Overview of RSPH New Cluster
- Cluster Login & Basic Linux Command
- File System
- Software Management
- SLURM Job Scheduler
- Q&A