

Computational Skills for Researchers

Intro to Quest

Alper Kinaci, Sr. Computational Specialist
Research Computing Services

<https://github.com/nuitrcs/intro-quest-ciera-workshop>

Northwestern

INFORMATION TECHNOLOGY

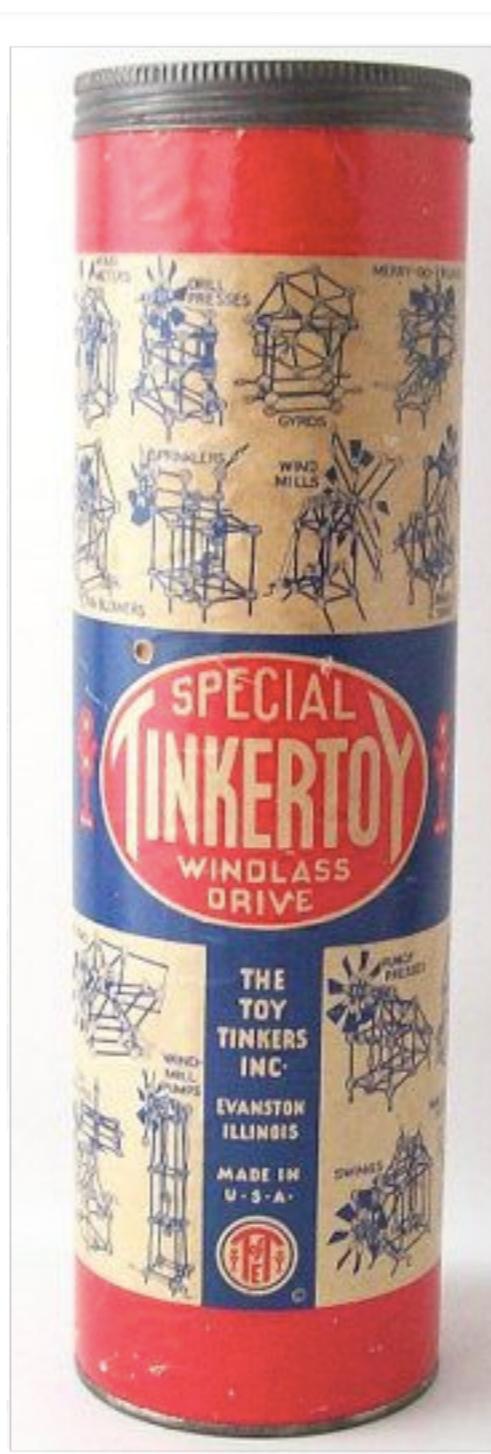
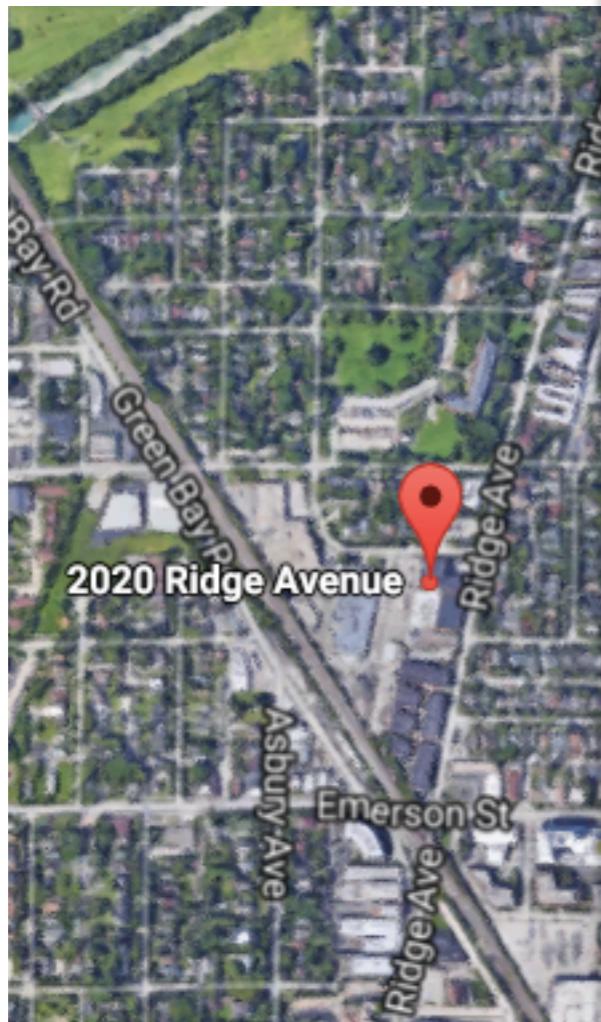
Learn About

- Quest System Architecture
- Job Scheduler & Software
- Parallel Computing
- Getting Started: Logging in, file transfers,
batch & interactive job
submission

High Performance Computing



Quest: High Performance Compute Cluster



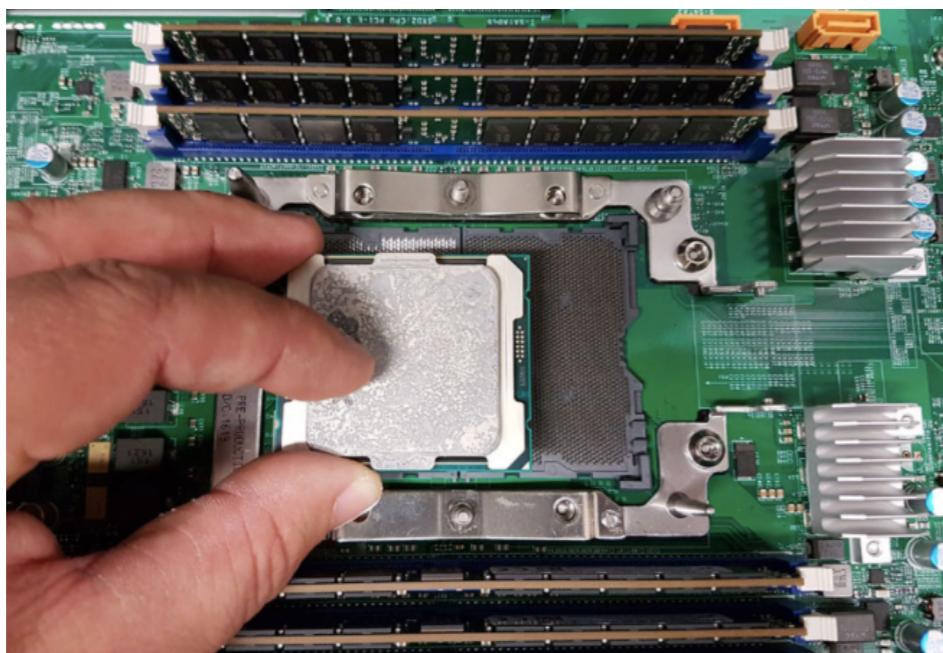
24 hour on-site service
firewalls and intrusion detection systems,
dedicated power station, high-throughput network

or authentication,

Quest: High Performance Compute Cluster

“node”: a computer

>670



“core”: a processor
~18,000

Quest: High Performance Compute Cluster

Desktop vs. Quest

1

Processor Speed: 2.2 GHz
Number of Processors: 1
Total Number of Cores: 2
L2 Cache (per Core): 256 KB
L3 Cache: 4 MB
Memory: 8 GB

670+

Processor Speed: 2.5-3.3 GHz
Number of Processors: 2
Total Number of Cores: 24+
L2 Cache (per Core): 256 KB
L3 Cache: 30 MB
Memory: 128 GB

Quest: High Performance Compute Cluster

“infiniband”: high-speed
inter-connect



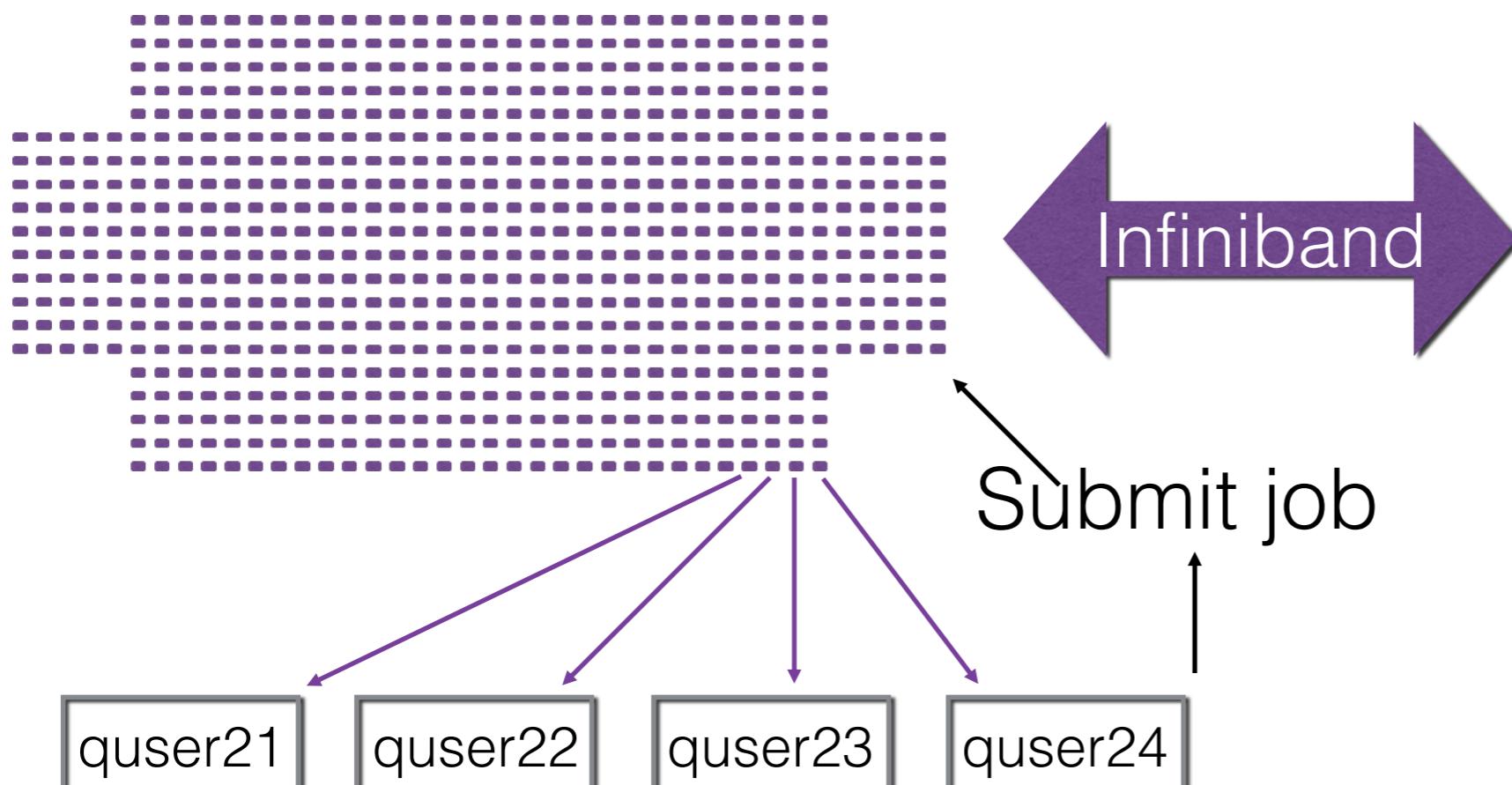
nodes in racks

Quest: High Performance Compute Cluster

Access, compute & storage

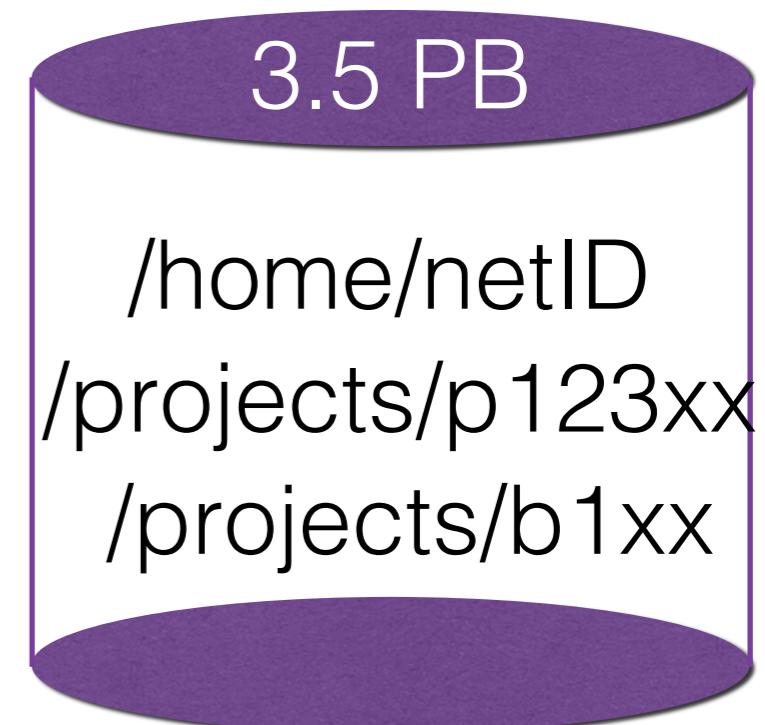
670+ compute nodes

128 GB or 96 GB RAM



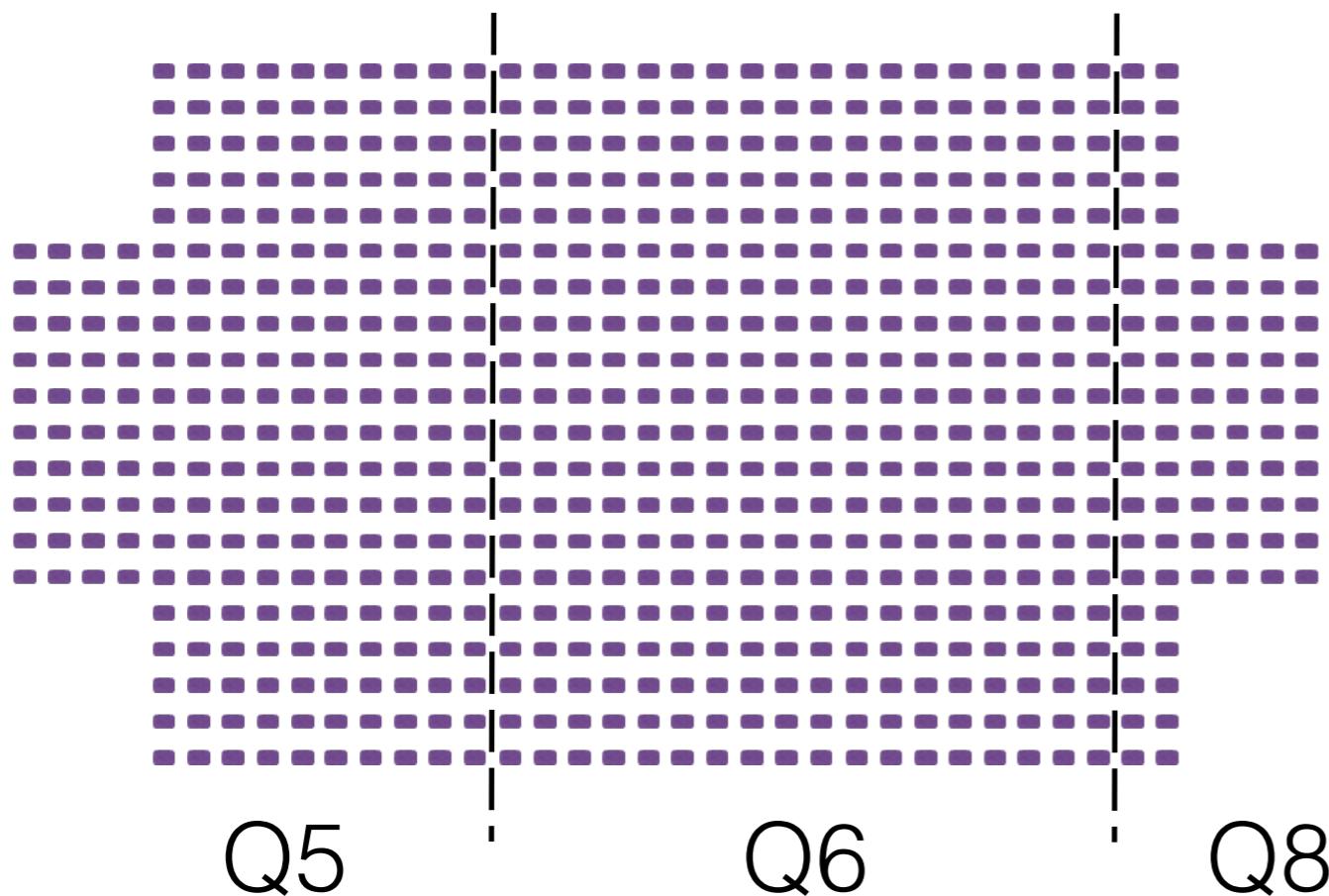
login nodes

ssh netID@quest.northwestern.edu



Quest: High Performance Compute Cluster

Node Architectures: 3 Generation of Nodes



670+ compute nodes

128 GB RAM (Q5, Q6)

96 GB RAM (Q8)

Q5 → 24 cores/node

Q6 → 28 cores/node

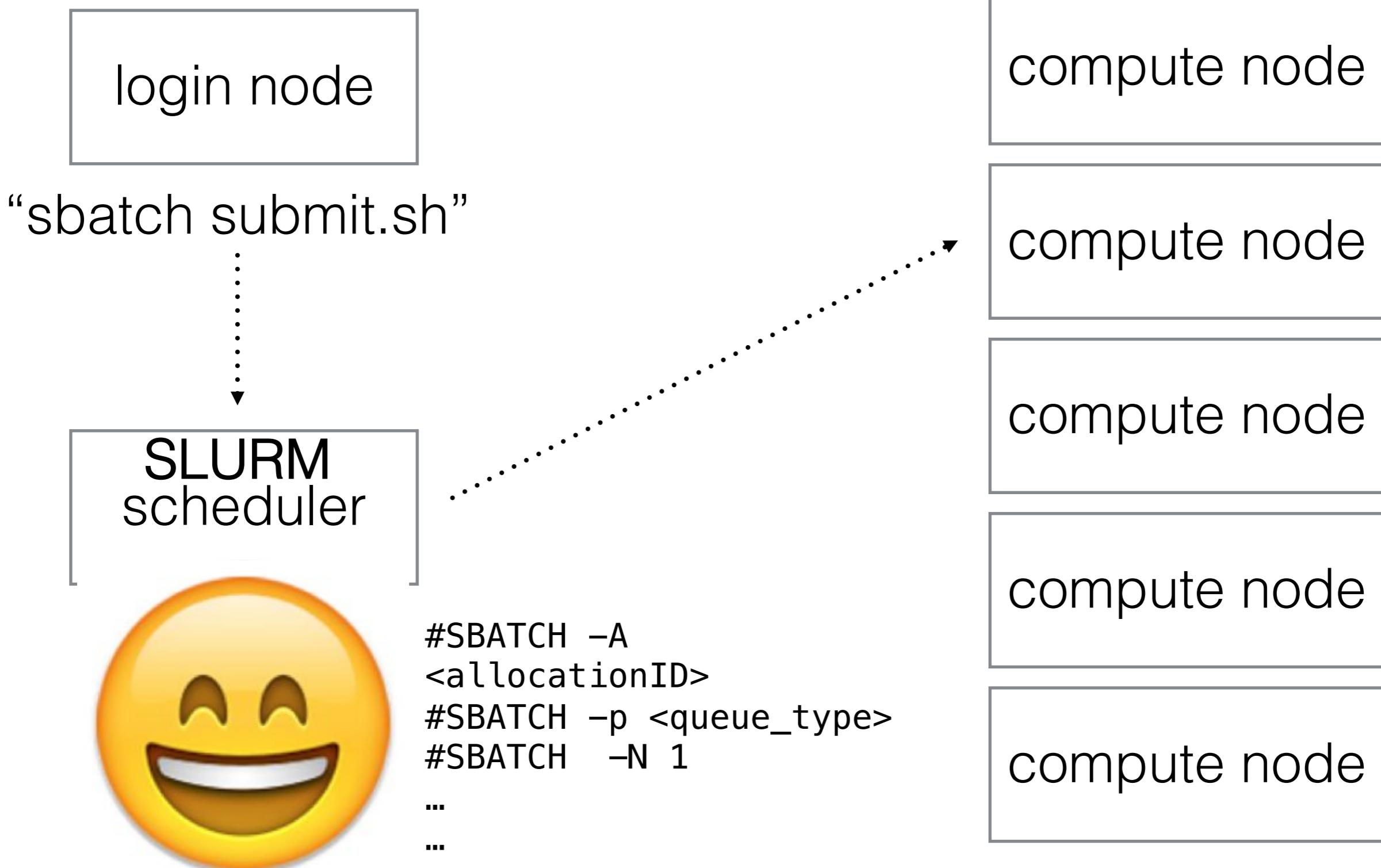
Q8 → 28 cores/node

Quest: High Performance Compute Cluster

Job Scheduler



Quest: High Performance Compute Cluster



Quest: High Performance Compute Cluster

Software

- Operating system
- Loadable modules

<https://kb.northwestern.edu/quest-software>

Quest: High Performance Compute Cluster

Parallel Computing

ACTIVITY

Quest: High Performance Compute Cluster

Parallel Computing

Multiple jobs working independently

“embarrassingly parallel”

“pleasingly parallel”

high-throughput computing

Single job communicating across nodes

message passing:

MPI, Spark, OpenMP

Getting Started: Logging in

Mac users: launch the Terminal App

PC users: launch Git Bash

Linux users: launch the Terminal App

Getting Started: Logging in

Connect to Quest using ssh (secure shell)

```
$ ssh <NetID>@quest.northwestern.edu
```

Print working directory

```
$ pwd
```

Names of your allocations (storage in /projects/<allocationID>)

```
$ groups
```

Summarize disk usage in your home folder

```
$ du -h ~
```

Logout from Quest, return to “work-folder” in your computer

```
$ logout
```

Getting Started: Software modules

Start Python

```
$python  
Python 2.7.5  
>>> exit()
```

It seems system has python 2.7

Check available software provided by modules

```
$module avail
```

Load python/anaconda3.6

```
$module load python/anaconda3.6
```

Getting Started: Software modules

Start Python 3.6

```
$python  
Python 3.6.0 | Anaconda 4.3.0 (64-bit)|  
>>> exit()
```

Now python 3.6 can be used

Unload python/anaconda3.6

```
$module unload python/anaconda3.6
```

Start Python again

```
$python  
Python 2.7.5  
>>> exit()
```

Getting Started: Clone GitHub repo

Clone the GitHub repository

```
$git clone  
https://github.com/nuitrcs/intro-quest-  
ciera-workshop
```

Change directory to cloned folder

```
$cd intro-quest-ciera-workshop
```

List the files/folders in the cloned folder

```
$ls
```

Change directory to the parent folder

```
$cd .. /
```

Getting Started: File transfers

To connect to Quest, start Cyberduck and then :

- 1) **Click Open Connection** in the upper left of the Cyberduck window
At the top of the Open Connection window that appears, Select SFTP (SSH File Transfer Protocol) from the drop-down menu.
- 2) **Enter `quest.it.northwestern.edu`** for server specification
- 3) **Enter your NetID** in the Username: box and leave the Password: box empty to prevent your NetID password from being saved in a file on your personal computer. Public Key Authentication is not supported.
- 4) **Click Connect.** You will see a Login failed window.
- 5) **Enter your NetID password** in the Password: field.
- 6) **Click Login.**

Quest: Transfer with Cyberduck

On your local machine

Drag "intro-quest-ciera-workshop" folder into your Quest home directory in Cyberduck

Connect to Quest using ssh (secure shell) from Git Bash

```
$ ssh <NetID>@quest.northwestern.edu
```

List the files/folders in the home folder

```
$ ls
```

Change directory to copied folder

```
$ cd intro-quest-ciera-workshop
```

List the files/folders in intro-quest-ciera-workshop folder

```
$ ls
```

Quest: Transfer with sftp

Logout from Quest, return to your computer

```
$ logout
```

Remove (i.e. delete) cloned folder from your computer

```
$ rm -Rf intro-quest-ciera-workshop
```

List the files/folders to see if the folder is removed

```
$ ls
```

Connect to Quest with secure file transfer protocol (sftp)

```
$ sftp <NetID>@quest.northwestern.edu
```

Download intro-quest-ciera-workshop folder to your computer

```
>get -r intro-quest-ciera-workshop
```

Quest: Transfer with sftp

List existing files on Quest

```
>ls
```

Locally list existing files (i.e. in your computer)

```
>lss
```

Change directory to cloned folder on Quest

```
>cd intro-quest-ciera-workshop
```

Remove “submit_generic.sh” file on Quest

```
>rm submit_generic.sh
```

List existing files on Quest

```
>ls
```

Quest: Transfer with sftp

Change directory to cloned folder on your local computer

```
>lcd intro-quest-ciera-workshop
```

Upload submit_generic.sh to Quest

```
>put submit_generic.sh
```

List existing files on Quest again

```
>ls
```

Quit sftp, return to intro-quest-ciera-workshop on your computer

```
>quit
```

Print working directory

```
$pwd
```

Quest: Transfer with scp

Change directory to “intro-quest-ciera-workshop” folder on your local computer if you are not already there.

Remove “submit_generic.sh” file from your computer

```
$ rm submit_generic.sh
```

List existing files in intro-quest-ciera-workshop

```
$ ls
```

Secure copy submit_generic.sh from Quest to your computer

```
$ scp <NetID>@quest.northwestern.edu:~/intro-quest-ciera-workshop/submit_generic.sh  
./submit_generic.sh
```

List files in intro-quest-ciera-workshop on your computer

```
$ ls
```

Getting Started: Batch Job Submission

Concatenate (print out) submission script, “submit_generic.sh”

```
$cat submit_generic.sh
```

```
#!/bin/bash
#SBATCH --account=<allocationID> ## <-- EDIT THIS TO BE YOUR ALLOCATION
#SBATCH --partition=<queue_type> ## <-- EDIT THIS TO BE YOUR QUEUE NAME
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=00:10:00
#SBATCH --mem-per-cpu=1G
#SBATCH --job-name=sample_job
#SBATCH --output=outlog
#SBATCH --error=errlog

module purge all          ## Unload existing modules
module load python         ## Load necessary modules (software, libraries)

bash whereami              ## Run the program
python helloworld.py       ## Run the program
```

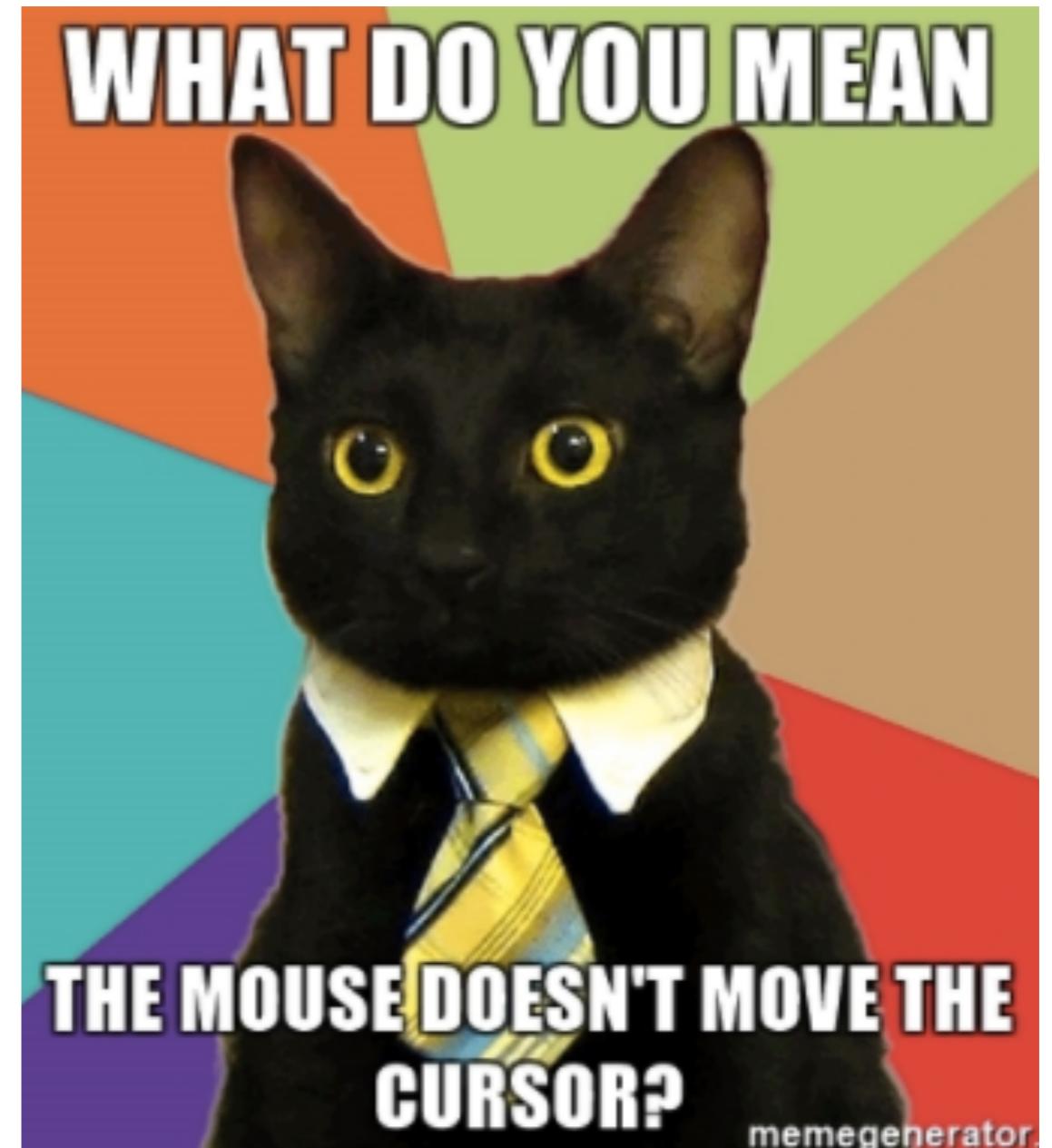
Getting Started: Batch Job Submission

Open submit_generic.sh with VIM text editor

```
$vim submit_generic.sh
```

text editor: vi/vim

- You land in command mode
- To enter insert mode: “i”
- Navigate with arrow keys
- To exit insert mode: “esc”
- To save & exit: “:wq”
- To exit without saving: “q!”



Getting Started: Batch Job Submission

What is the scheduler looking for in your script?

This is a BASH script: `#!/bin/bash`

Account: `#SBATCH --account=w10001`

Partition/Queue: `#SBATCH --partition=w10001`

Number of nodes: `#SBATCH --nodes=1`

Number of cores: `#SBATCH --ntasks-per-node=1`

Length of the job: `#SBATCH --time=00:10:00`

Required memory: `#SBATCH --mem-per-cpu=1G`

Name of the job: `#SBATCH --job-name=sample_job`

Generate an output log: `#SBATCH --output=outlog`

Generate an error log: `#SBATCH --error=errlog`

Getting Started: Batch Job Submission

Submit your job to batch processing

```
$sbatch submit_generic.sh
```

Note your job_id returned after sbatch

What is the status of your job(s) in the queue

```
$squeue -u <NetID>
```

Detailed information about the job

```
$checkjob <job_id>
```

When the job ends, investigate errlog, outlog and python_output.txt

Getting Started: Interactive Job Submission (with display)

PC users: Launch Xming

In Git Bash on your local computer

```
$export DISPLAY=localhost:0  
$ssh -XY <NetID>@quest.northwestern.edu
```

Mac users: Launch XQuartz (may be already running)

In terminal on your local computer

```
$ssh -X <NetID>@quest.northwestern.edu
```

Linux users: In terminal on your local computer

```
$ssh -X <NetID>@quest.northwestern.edu
```

Getting Started: Interactive Job Submission (with display)

Alternatively you can use FastX for display

PC, Mac & Linux users: FastX:

Hostname : quest.northwestern.edu

Username : your Northwestern NetID

Password : your Northwestern NetID password

From browser: <https://quest.northwestern.edu:3000>

Username : your Northwestern NetID

Password : your Northwestern NetID password

Getting Started: Interactive Job Submission

By submitting an interactive job, you can access the command line on a compute node to do your work

- 1) submit your job from the command line

```
$ srun --x11          \ → If display is required (optional)
    --account=w10001    \
    --partition=w10001   \
    --time=00:10:00     \
    --nodes=1           \
    --ntasks-per-node=1 \
    --mem-per-cpu=1G   \
    --pty bash -l       → Starts the bash terminal
```

Getting Started: Interactive Job Submission

When the job starts on a compute node, run matlab graphical user interface (GUI)

Check available matlab modules

```
$module avail matlab
```

Load matlab/r2018a module

```
$module load matlab/r2018a
```

Run matlab

```
$matlab
```

Close GUI and logout from the interactive session

```
$logout
```

Getting Started: Job Submission Examples

Jobs on Quest

<https://kb.northwestern.edu/page.php?id=69247>

Example Jobs

<https://kb.northwestern.edu/page.php?id=70719>

Questions?

email: quest-help@northwestern.edu

Research Computing Services

Northwestern

INFORMATION TECHNOLOGY