Joint High Performance Computing Exchange (JHPCE) Cluster Orientation

JOHNS HOPKINS
BLOOMBERG SCHOOL
of PUBLIC HEALTH

http://www.jhpce.jhu.edu/

Schedule

- **Introductions – who are we, who are you?**
- Terminology
- Logging in and account setup
- Basics of running programs on the cluster
- Details – limits and resources
- Examples

Who we are:

- JHPCE – Joint High Performance Computing Exchange
  - Co-Director: Brian Caffo
  - Co-Director: Mark Miller
  - Systems Engineer: Jiong Yang
- Beyond this class, when you have questions:
  - http://www.jhpce.jhu.edu - lots of good FAQ info
  - bitsupport@lists.johnshopkins.edu
    - System issues (password resets/disk space)
    - Monitored by the 3 people above
  - bithelp@lists.johnshopkins.edu
    - Application issues (R/SAS/perl...)
    - Monitored by dozens of application SMEs
    - All volunteers
  - Others in your lab
  - Web Search

Who are you?

- Name
- Department
- How do you plan on using the cluster? What data or applications will you be using?
- Will you be accessing the cluster from a Mac or a Windows system?
- What is your experience with Unix?
- Any experience using other clusters?

Schedule

- Introductions – who are we, who are you?
- **Terminology**
- Logging in and account setup
- Basics of running programs on the cluster
- Details – limits and resources
- Examples

What is a cluster?

- A collection of many powerful computers that can be shared with many users.

Node (Computer) Components

- Each computer is called a “Node”
- Each node, just like a desktop/laptop has
  - RAM
  - Intel and AMD CPUs
    - Cores
  - Disk space

- Unlike desktop/laptop systems, nodes do not make use of a display/mouse – they are used from a command line interface

- Range in size from a large pizza box to a long shoe box

The JHPCE cluster components

- Joint High Performance Computing Exchange (JHPCE)
  . . . kinda sounds out to “gypsy”
- Fee for service – nodes purchased by various PIs.
- Located at Bayview Colocation Facility

Software:

- Used for a wide range of Biostatistics – gene sequence analysis, population simulations, medical treatment analysis.
- Common applications: R, SAS, Stata, perl, python ...

Hardware:

- 11 Racks of equipment – 5 compute, 5 storage, 1 infrastructure
- 76 Nodes – 72 compute nodes, 2 transfer nodes, 2 login nodes
- 3100 Cores - Nodes have 2 to 4 CPUs, giving 24 to 64 cores per node
- 25 TB of RAM - Nodes ranges from 128 GB to 768 GB RAM per node.
- 9000 TB of Disk space – 6500 TB of project storage, 2000 TB of backup, 500TB of scratch/home/other storage.
  - Project, home, and scratch storage is network attached, so it available to all nodes of the cluster.

How do programs get run on the compute nodes?

- We use a product called “Sun Grid Engine” (SGE) that schedules programs (jobs)
- History:
  - 1990s - Developed by Gridware
  - 2000 - Gridware purchased by Sun Microsystems
  - 2001 - Sun makes source code open source
  - 2010 - Oracle buys Sun and discontinues support for SGE
  - 2013 - Univa picks up support for Sun customers
- Jobs are assigned to slots as they become available and meet the resource requirement of the job
- Jobs are submitted to queues
  - Shared Queue
  - Designated queues
- The cluster nodes can also be used interactively.

Why would you use a cluster?

- You need resources not available on your local laptop
- You need to run a program (job) that will run for a very long time
- You need to run a job that can make use of parallel computing

Schedule

- Introductions – who are we, who are you?
- Terminology
- **Logging in and account setup**
- Basics of running programs on the cluster
- Details – limits and resources
- Examples

How do you use the cluster?

- The JHPCE cluster is accessed using SSH (Secure SHell), so you will need an ssh client.
- Use **`ssh`** to login to "**`jhpce01.jhsph.edu`**"

- For Mac and Linux users, you can use **`ssh`** from Terminal Window.

- For MS Windows users, you need to install an ssh client – such as MobaXterm (recommended) or Cygwin, Putty and Winscp :

http://mobaxterm.mobatek.net/

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

http://www.cygwin.com

http://winscp.net

Quick note about graphical programs

To run graphical programs on the JHPCE cluster, you will need to have an X11 server running on your laptop.

- For Microsoft Windows, Mobaxterm has and X server built into it.
- For Windows, if you are using Putty, you will need to install an X server such as Cygwin.

- For Macs:

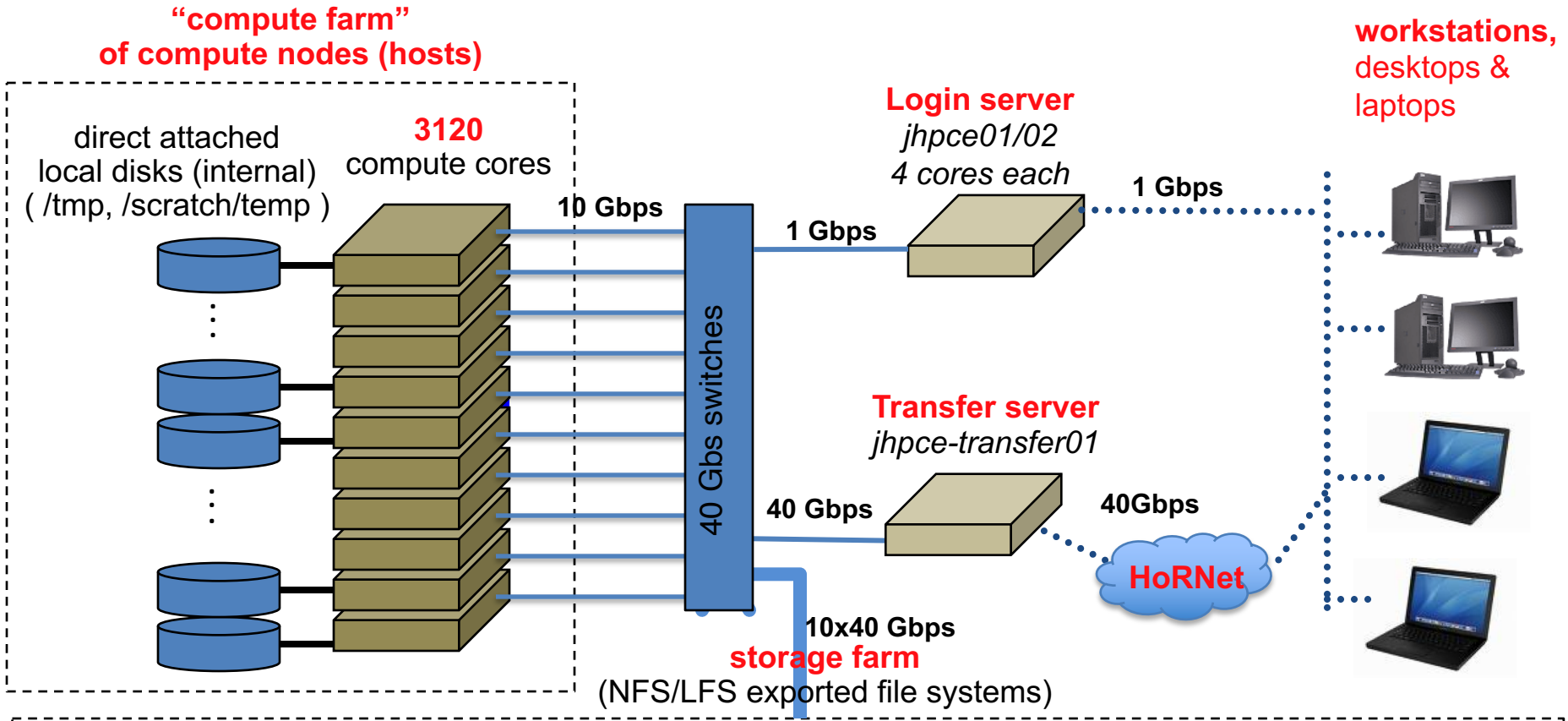
1) You need to have the Xquartz program installed on your laptop. This software is a free download from Apple, and does require you to reboot your laptop http://xquartz.macosforge.org/landing/
2) You need to add the "-X" option to your ssh command:

```
$ ssh -X mmill116@jhpce01.jhsph.edu
```

- For Linux laptops, you should already have an X11 server install. You will though need to add the –X option to ssh:

```
$ ssh -X mmill116@jhpce01.jhsph.edu
```

JHPCE user-model



/dcs01
ZFS/NFS

1080TB raw
800TB usable

8 SM847E26-RJBOD1 JBODs with 450 WD Red 3TB disks

/dcl01
Lustre

4500TB raw
3100TB usable

20 SM847E26-RJBOD1 JBODs with 450 WD Red 4TB disks and 450 WD Red 6TB disks

/dcl02
Lustre
(encrypted)

3520TB raw
2463TB usable

10 SM847E26-RJBOD1 JBODs with 440 8TB disks

/users, /legacy, /starter02
ZFS/NFS
160TB usable
1 SM JBOD with 40 WD Red 6TB disks and
SSDs for L2ARC and SLOG

/amber3
ZFS/NFS
100 TB useable Sun 7210

Example 1 – Logging in

- Bring up Terminal
- Run: **`ssh –X <USER>@jhpce01.jhsph.edu`**
- 2 Factor authentication
- Shell prompt

Lab 1 - Logging In

- For Mac/Linux laptop Users:

  - Bring up a Terminal
  - Run: **`ssh –X <USER>@jhpce01.jhsph.edu`**

  - Login with the initial Verification Code and Password that were sent to you

- For PC Users:

  - Launch MobaXterm
  - click on the “Sessions” icon 

in the upper left corner
  - On the “Session settings” screen, click on “SSH”
  - Enter “jhpce01.jhsph.edu” as the “Remote host”. Click on the “Specify username” checkbox, and enter your JHPCE username in the next field. Then click the “OK” button.
  - Login with the initial Verification Code and Password that were sent to you.
  - If dialog windows pop up, click "Cancel" when prompted for another Verification Code, or click "No" or “Do not ask this again” when prompted to save your password.

Lab 1 - Logging In - cont

- Change your password with the "**`kpasswd`**" command. You will be prompted for your current Initial Password, and then prompted for a new password.

- Setup 2 factor authentication
  - http://jhpce.jhu.edu/knowledge-base/how-to/2-factor-authentication/
  - 1) On your smartphone, bring up the "Google Authenticator" app
  - 2) On the JHPCE cluster, run "auth_util"
  - 3) In "auth_util", use option "5" to display the QR code (you may need to resize your ssh window)
  - 4) Scan the QR code with the Google Authenticator app
  - 5) Next in "auth_util" use option 2 to display your scratch codes
  - 6) In " auth_util", use option "6" to exit from "auth_util"

- Logout of the cluster by typing "exit".
- Login to the cluster again with 2 factor authentication

Lab 1 - Logging In - cont

- Note "Emergency Scratch Codes"
- Note: 100 GB limit on home directory. Home directories are backed up, but other storage areas may not be.
- (optional) Setup ssh keys

General Linux/Unix Commands

Navigating Unix:

- ls
- ls -l
- ls -al
- pwd
- cd
- . and ..
- man

Commands in example script:

- date
- echo
- hostname
- sleep
- control-C

Looking at files:

- more

Changing files with editors:

- nano
- vi/emacs

Good overviews of Linux: http://korflab.ucdavis.edu/Unix_and_Perl/unix_and_perl_v3.1.1.html
https://www.codecademy.com/learn/learn-the-command-line

Schedule

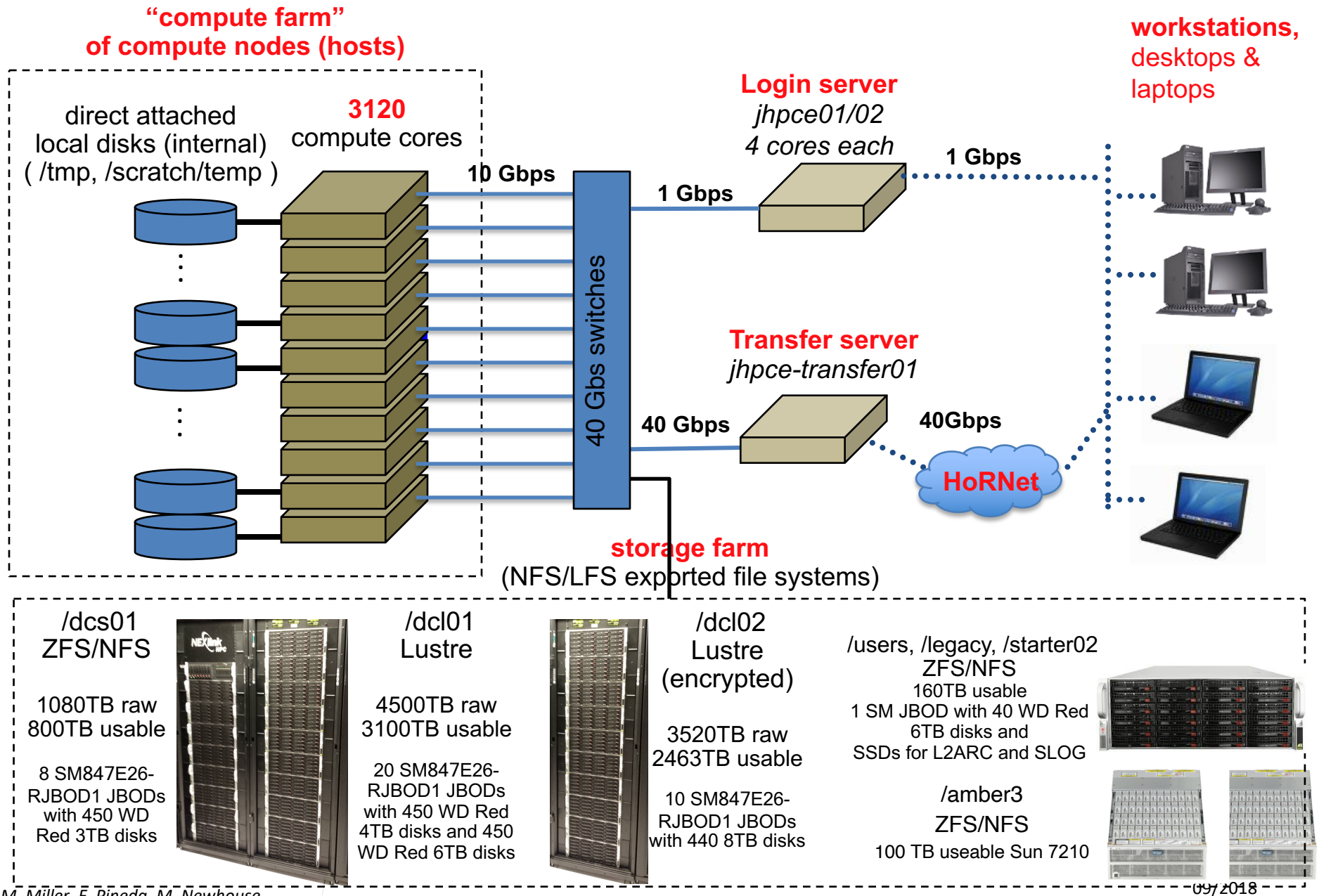
- Introductions – who are we, who are you?
- Terminology
- Logging in and account setup
- **Basics of running programs on the cluster**
- Details – limits and resources
- Examples

Submitting jobs to the queue with Sun Grid Engine (SGE)

- **`qsub`** – allows you to submit a batch job to the cluster
- **`qrsh`** – allows you establish an interactive session

- **`qstat`** – allows you to see the status of your jobs

JHPCE user-model



Lab 2 - Using the cluster

Example 2a – submitting a batch job

```
cd class-scripts
qsub -cwd script1
qstat
```

examine results files

Example 2b – using an interactive session

```
qrsh
```

Note current directory

Modules

Modules for R, SAS, Mathematica . . .

- module list
- module avail
- module load
- module unload

Never run a job on the login node!

- Always use "**`qsub`**" or **`qrsh`**" to make use of the compute nodes
- Jobs that are found running on the login node may be killed at will
- If you are going to be compiling programs, do so on a compute node via **`qrsh`**.
- Even something as simple as copying large files should be done via **`qrsh.`** The compute nodes have 10Gbps connections to the storage systems and jhpce01 only has a 1Gbps connection.

Other useful SGE commands

`qstat` – shows information about running jobs you are running

`qacct` – shows information about completed jobs

`qhost` – shows information about the nodes

`qdel` – deletes your job

Schedule

- Introductions – who are we, who are you?
- Terminology
- Logging in and account setup
- Basics of running programs on the cluster
- **Details – limits and resources**
- Examples

A few options for “qsub”

- To run the job in the current working directory (where qsub was executed) rather than the default (home directory)

  `-cwd`

- To send standard output (error) stream to a different file or directory

  `-o` *`path/filename`*

  `-e` *`path/filename`*

- To receive notification via email when your job completes

  `-m e -M` **`john@jhu.edu`**

File size limitations

- There is a default file size limitation of 10GB for all jobs.
- If you are going to be creating files larger than 10GB you need to use the "**`-l h_fsize`**" option.
- The "**`-l`**" option is used to set "limits" for **`qsub/qrsh`** and will be used quite a bit in the next several slides.

- Example:

```
$ qsub -l h_fsize=50G myscript.sh
```

Embedding options in your qsub script

- You can supply options to qsub in 2 ways:
  - On the command line

```
qsub -cwd -l h_fsize=100G -m e -M john@jhu.edu script1.sh
```

  - Setting them up in your batch job script. Lines that start with "#$" are interpreted as options to **qsub**

```
$ head script1-resource-request.sh
#$ -l h_fsize=100G
#$ -cwd
#$ -m e
#$ -M john@jhu.edu

$ qsub script1-resource-request.sh
```

Requesting resources in qsub or qrsh

- By default, when you submit a job, or run qrsh, you are allotted 5GB of RAM and 1 core for your job.
- These default settings can be adjusted by modifying the .sge_request file in your home directory.

Requesting additional RAM

- You can request more RAM by setting **mem_free** and **h_vmem**.
  - **mem_free** – is used to set the amount of memory your job will need. SGE will place your job on a node that has at least **mem_free** RAM available.
  - **h_vmem** – is used to set a high water mark for your job. If your job uses more than **h_vmem** RAM, your job will be killed. This is typically set to be the same as mem_free.

- Examples:

```
qsub -l mem_free=10G,h_vmem=10G job1.sh
```

or

```
qrsh -l mem_free=10G,h_vmem=10G
```

Estimating RAM usage

- No easy formula until you've actually run something
- A good place to start is the size of the files you will be reading in
- If you are going to be running many of the same type of job, run one job, and use "**`qacct -j <jobid>"`** to look at "**`maxvmem`**".

Requesting additional cores

To request multiple cores, the "-pe local N" option (where N is the number of cores) needs to be supplied to the qsub or qrsh commnd For example:

- A job that will use 4 cores:

```
qsub -cwd -pe local 4 myscript.sh
```

- A qrsh session that will use 6 cores:

```
qrsh -pe local 6
```

IMPORTANT NOTE: The **`mem_free`** and **`h_vmem`** RAM limits are per-core values, so the total RAM requested needs to be divided by the number of cores.

Examples:

– A job which will use 10 cores and need 120GB of RAM:

```
qsub -cwd -pe local 10 -l mem_free=12G,h_vmem=12G myscript2.sh
```

Types of parallelism

1. Embarrassingly (obviously) parallel …
http://en.wikipedia.org/wiki/Embarrassingly_parallel

2. Multi-core (or multi-threaded) – a single job using multiple CPU cores via program threads on a single machine (cluster node). Also see discussion of fine-grained vs coarse-grained parallelism at
http://en.wikipedia.org/wiki/Parallel_computing

3. Many CPU cores on many nodes using a Message Passing Interface (MPI) environment. Not used much on the JHPCE Cluster.

“How many jobs can I submit?”

Users frequently submit 1000s of jobs on the cluster. We don't impose a limit on the **submitting** of jobs, but users should not submit more than 10,000 jobs without notifying bitsupport@lists.jhu.edu first.

We do however impose a per-user limit on the number of cores and RAM for **running** jobs on the shared queue. Currently, the limit is set to **200 cores per user and 768GB of RAM per user.**

So, if a user submits 1000 single-core jobs, the first 200 will begin immediately (assuming the cluster has 200 cores available on the shared queue), and the rest will remain in the ‘qw’ state until the first 200 jobs start to finish. As jobs complete, the cluster will start running 'qw' jobs, and keep the number of running jobs at 200.

Similarly, if a user's job requests 100GB of RAM to run, the user would only be able to run 7 jobs before hitting their 768 GB limit, and subsequent jobs would remain in 'qw' state until running jobs completed.

The maximum number of slots per user may be temporarily increased by submitting a request to bitsupport@lists.jhu.edu. We will increase the limit, depending on the availability of cluster resources. There are also dedicated queues for stakeholders which may have custom configurations and limits.

Schedule

- Introductions – who are we, who are you?
- Terminology
- Logging in and account setup
- Basics of running programs on the cluster
- Details – limits and resources
- **Examples**

Lab 3

Running R on the cluster:

- In `$HOME/class-scripts/R-demo`, note 2 files – Script file and R file

- Submit Script file
  - `qsub -cwd plot1.sh`
- Run R commands interactively
  - `qrsh`
  - `module load conda_R`
  - `R`

Lab 4

Transferring files to the cluster

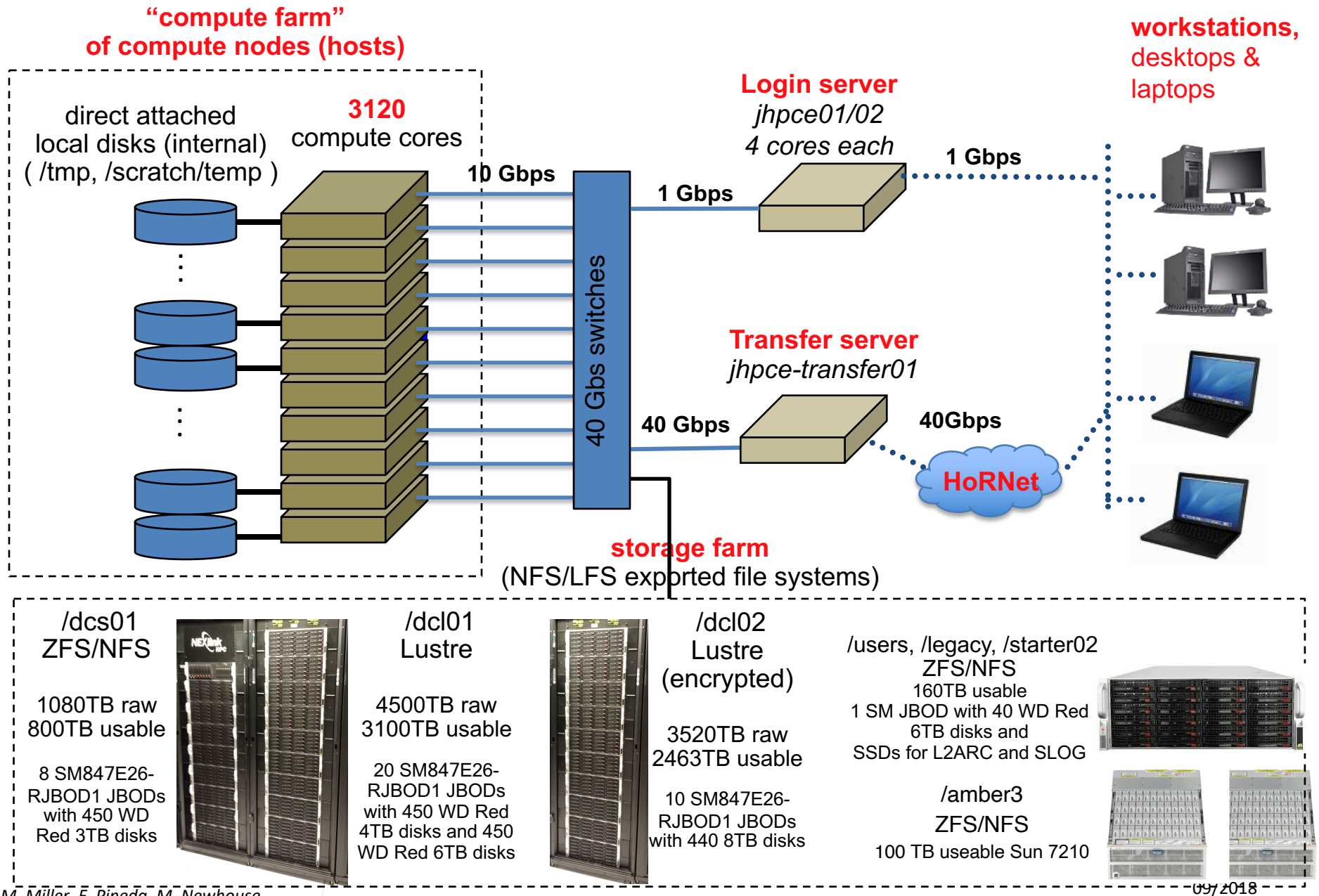
- Transfer results back

```
$ sftp mmill116@jhpce-transfer01.jhsph.edu
```

Or use WinSCP, Filezilla, Globus, mobaxterm…

- Look at resulting PDF file using viewer on laptop

JHPCE user-model



Lab 5

Running RStudio

- X Windows Setup
  - For Windows, Mobaxterm has and X server built into it
  - For Mac, you need to have the Xquartz program installed (which requires a reboot), and you need to add the "-X" option to ssh:

```
$ ssh -X mmill116@jhpce01.jhsph.edu
```

- Start up Rstudio

```
$ qrsh -l mem_free=10G,h_vmem=10G
$ module load conda_R
$ module load rstudio
$ rstudio
```

- Look at pdf from example 4 via `xpdf`

Other queues

- “shared” queue – default queue
- “download” queues
  - **`qrsh -l rnet`**
    - `(jhpce-transfer01.jhsph.edu )`
- “math” queue
  - **`qrsh -l math`**
- “sas” queue
  - **`qrsh -l sas`**

Lab 6

Running Stata and SAS

- Stata example:

Batch:

```
$ cd $HOME/class-scripts/stata-demo
$ ls
$ cat stata-demo1.sh
$ cat stata-demo1.do
$ qsub stata-demo1.sh
```

Interactive:

```
$ qrsh
$ stata
```

or

```
$ xstata
```

- SAS example:

Batch:

```
$ cd $HOME/class-scripts/SAS-demo
$ ls
$ cat sas-demo1.sh
$ cat class-info.sas
$ qsub sas-demo1.sh
```

Interactive:

```
$ qrsh -l sas
$ sas
```

Note – The name of the program and script need not be the same, but it is good practice to keep them the same when possible.

Note – Extensions sometimes are meaningful. SAS doesn't care, but Stata programs need to have ".do" as the extension. It is good practice for human readability.

Summary

- Review
  - Get familiar with Linux
  - Use ssh to connect to JHPCE cluster
  - Use qsub and qrsh to submit jobs
  - Never run jobs on the login node
  - Helpful resources
    - http://www.jhpce.jhu.edu/
    - bitsupport@lists.johnshopkins.edu - System issues
    - bithelp@lists.johnshopkins.edu - Application issues
- What to do next
  - Make note of your Google Authenticator scratch codes (option 2 in "auth_util")
  - Set up ssh keys if you will be accessing the cluster frequently https://jhpce.jhu.edu/knowledge-base/authentication/login/
  - Play nice with others – this is a shared community supported system.

Thanks for attending! Questions?