

# Introduction to Discovery

*<http://discovery.dartmouth.edu>*

# The Discovery Cluster



01/29/18

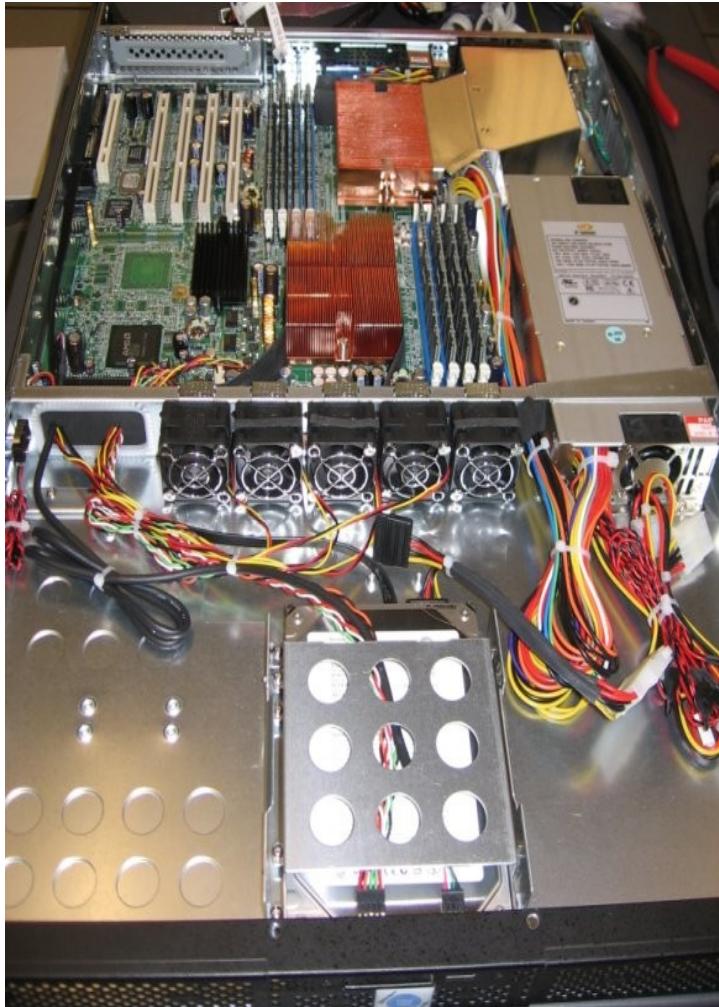
# Agenda

- What is a cluster and why use it
- Overview of computer hardware in cluster
- Help Available to Discovery Users
- Logging on to the cluster with “ssh”
- Transferring files to and from the cluster
- The Environment
- Scheduler basics
- Requesting resources - PBS scripts
- Checking on submitted jobs
- Cluster Etiquette - running jobs & disk space
- Publishing
- Labs

# Why Would You Need to Use Discovery ?

- Your program runs for a LONG time
- Your program needs a lot of memory
- You need to run your program many times
- Your data files use up a lot of disk space
- You need to run your program in parallel

# Cluster Nodes



**Cell E: AMD 4386 3.1 GHz Dual 8-core (16 cores)**

**Cell F: AMD 6348 2.8GHz Quad Dodeca-Core (48-cores)**

**Cell G: 2 NVidia K80 GPUs, Intel E5-2640 (16-cores)**

**Cell H: Intel Xeon E5-2470 2.3GHz Dual 8-Core (16-cores)**

**Cell J: Intel Xeon E5-2690 2.6GHz Dual 12-core (24 cores)**

**Cell K: Intel Xeon E5-2640V3 2.6GHz Dual 8-Core (16-cores)**

**Cell M: Intel Xeon E5-2667V4 3.2 GHz Dual 8-core (16-cores)**

# Help Available for Discovery Users

- Build and install requested applications
- Help getting your applications running
- Specialized help from RC application specialists:
  - Bioinformatics
  - Debugging, optimizing and parallelizing code
  - GIS
  - Statistics
  - Python, R, Java, C/C++, Fortran, Matlab
- Help setting up shared data repositories for research groups

# Logging On

- SSH (Secure Shell)

- Linux: `ssh -X username@discovery.dartmouth.edu`
- Mac: `ssh -Y username@discovery.dartmouth.edu`
- Windows
  - MobaXterm built in Xserver and sftp (free and recommended)
  - Ssh secure shell or putty

- Changing your password

- Use the **passwd** command to make the change.

# Discovery Converting to A New File System - DartFS

Two types of homes during transition period:

- New File System - DartFS: /dartfs-hpc/rc/home/e/d12345e (50GB)
- Old File System – Isilon: /ihome/username (20 GB)

New DartFS Lab Shared Directory:

- DartFS: /dartfs-hpc/rc/lab/X/XavierC [PI's name] (1TB)
- Shared space for members of your research group

Advantages of New DartFS File System:

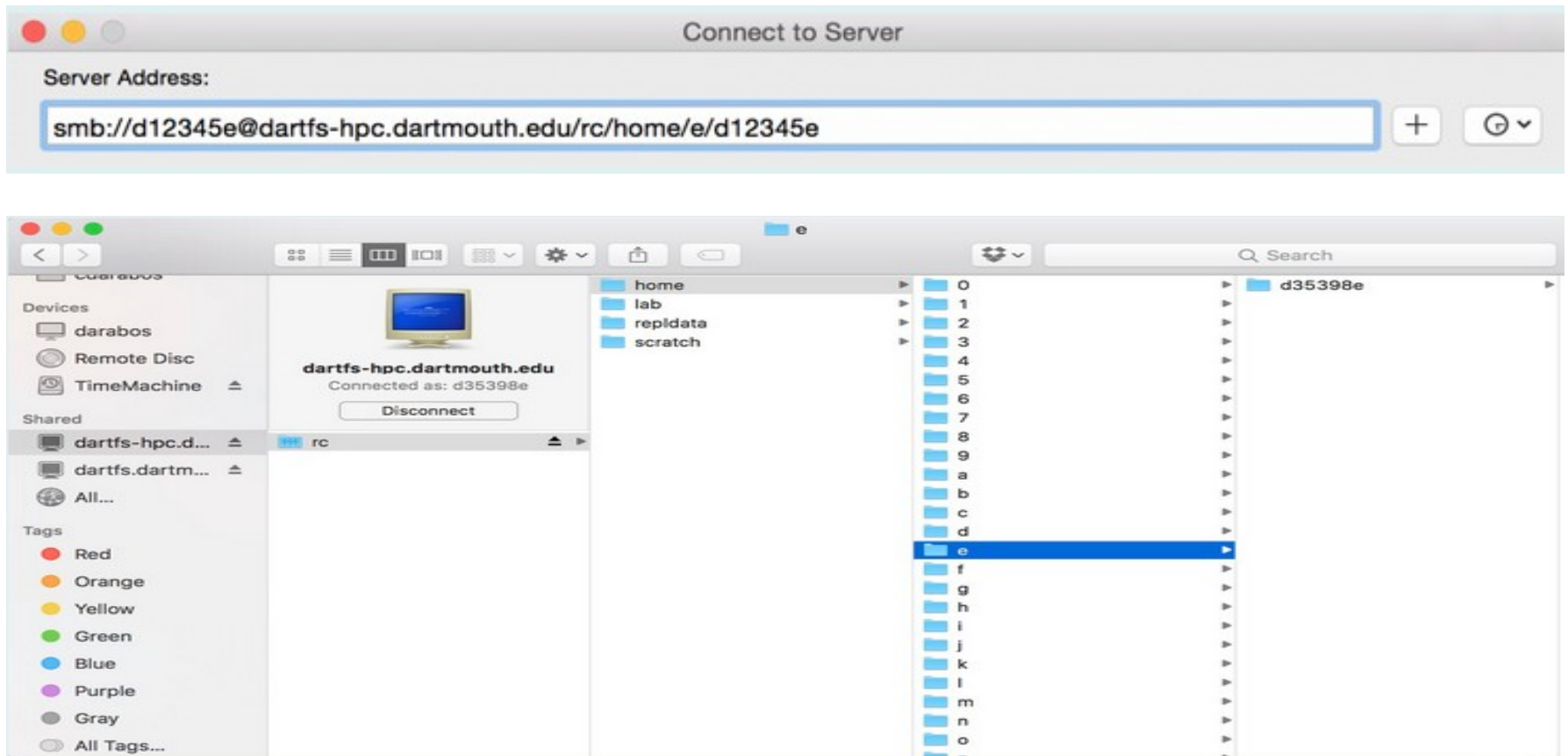
- Mount via SMB on Your Mac or Windows Laptop (easy to access files)
- More space available
- Login with netid (don't need to remember another login/password)
- Snapshots



# How To SMB Mount DartFS Files on Mac OS X

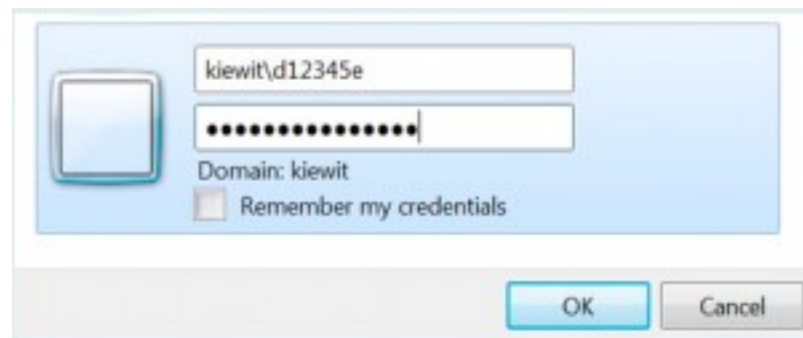
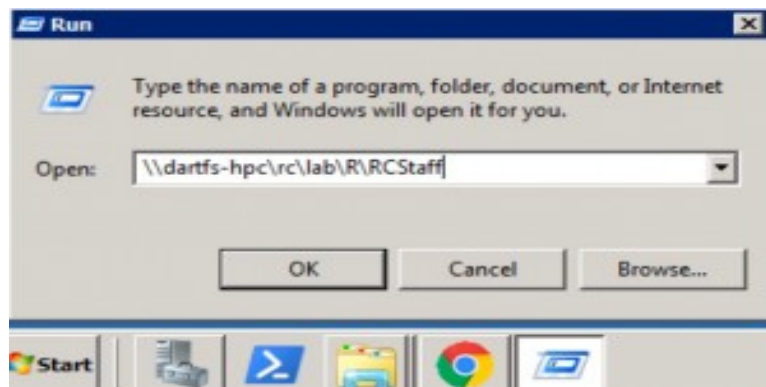
SMB Mount on Mac OS X:

Finder-> Go-> Connect to Server



# How To SMB Mount DartFS Files on Windows

- Click the Start or Windows menu icon (bottom left corner of the desktop).
- In the search text field, type Run. In Windows 10, just start typing run if the search field is missing.
- Select and click the Run application found towards the top of the menu.



# Transferring Files To/From Discovery (CLI)

- Linux or Mac (CLI): sftp & scp
  - CLI secure file transfer program – “sftp”
  - `sftp username@discovery.dartmouth.edu`
  - Use put, get, mput & mget
    - `put filename (mput filenames*)`
    - `get filename (mget filenames*)`
  - To copy from outside machine to discovery
    - `scp file(s) username@discovery.dartmouth.edu:`
    - `scp -r dir username@discovery.dartmouth.edu:`
      - **dir** will be created in your HOME directory on the cluster.

# Transferring Files To/From Discovery (GUI)

## GUI SFTP clients

- Windows
  - MobaXterm
  - WinSCP
- Macintosh
  - Fetch
- Both
  - Filezilla
  - Cyberduck

# Your Environment

## BASH

- The bash shell is the default shell you will be using on Discovery. The environment is tailored to use this shell.
- If you change to some other shell then queuing jobs, compiling parallel code is not guaranteed to work.
- **Warning:** Do not replace your `.bashrc` or `.bash_profile` files. Only add to them.

# Environment Modules I

- Using Modules to Manage Software
  - The Discovery cluster uses modules to manage the user environment for different third-party software versions.
  - The advantage of the modules approach is that the user is no longer required to specify paths for different versions, and to try to keep the PATH, MANPATH and related variables coordinated.
  - With the modules approach, users simply "load" and "unload" modules to control their environment.

# Environment Modules II

- Module commands
  - To get a usage list of module options type the following (the listing has been abbreviated to only those commands discussed in this webpage) :

- `$ module help`

Available Commands and Usage:

```
add|load      modulefile [modulefile ...]
rm|unload     modulefile [modulefile ...]
switch        modulefile1 modulefile2
display       modulefile [modulefile ...]
avail         path [path]
list
initadd       modulefile [modulefile ...]
help          modulefile [modulefile ...]
```

# Rstor Files

- If you have an AFS account...
- In order to have write access, to your AFS directory, you will need to use the **klog** command.
- The **klog** command will prompt you for your AFS account password.
- Once you have done this, you can use your AFS account to archive data and files from discovery.
- **AFS is only available from the discovery head node. It is not available from the compute nodes.**



# Disk Space

- You have write access to
  - \$HOME – your home directory (shortcut: ~ )
  - /scratch (local to nodes)
    - /scratch should be used for intermediate storage of the job data, if possible.
  - /dartfs-hpc/scratch (central scratch)
    - Data in /scratch and /dartfs-hpc/scratch cleaned by the system after 30 days.
- Home directories backed up daily offsite
  - Snapshots taken daily, weekly & monthly and are available in your
    - \$HOME/.snapshot

# Disk Space II

## Disk quotas

- \$HOME (50GB)
  - Email sent if quota usage reaches 95%
  - Use **quota** command to view your usage
- /scratch (no quota enforced)
  - Please have job cleanup
- /dartfs-hpc/scratch (no quota enforced)
  - Please have job cleanup

# Disk Space III

- If you need to store large quantities of data, we will work with you to arrange alternatives most suited to your needs.
- When over quota you can't write any files and sometimes can't login
- **Don't go over your quota**

# Publishing your work

- Discovery provides you a website to publish your work.
- The contents of your website is kept in a subdirectory below your HOME directory called **public\_html** or **dartmouth\_html**
  - **public\_html**: open to the world
  - **dartmouth\_html**: open only to the Dartmouth network
- The directory should be created as follows:
  - **\$ mkdir -m 711 ~/public\_html**
- URL: <http://rcweb.dartmouth.edu/~netid/>
- URL : <http://dartrcweb.dartmouth.edu/~netid>

# How to Get Started Running on Discovery

- Install your program(s) and copy any data to Discovery
- Run your program interactively on test nodes
- Debug your program if necessary
- Monitor and time your application
- Write a submit script and submit a sample job
- Look at job output and debug submit script
- Submit and monitor your job(s)

# Scheduler Basics

- Scheduling jobs
- PBS scripts
- Resources available
- Using the scheduler

# How The Scheduler Works

- Submit jobs to the scheduler - PBS scripts
- Torque – resource manager
  - Controls when and where jobs will run.
  - Does the work of putting the jobs on the nodes.
- Moab – job scheduler
  - Controls who can run on what resources for up to some period of time.
  - Determines Policies and Limits
- Priority, core count and walltime is based on your status
  - Part of a Membership Account(Buy-in)
  - Part of a Grant Account(3-months)
  - Part of a Free Access Account

# Example PBS Script

```
#!/bin/bash -l
# declare a name for this job to be sample_job
#PBS -N my_serial_job
# request the queue (enter the possible names, if omitted, default is the default)
# if more than 600 jobs use the largeq
#PBS -q default
# request 1 core on 1 node
# ensure you reserve enough cores for the projected memory usage
# figuring 4G/core
#PBS -l nodes=1:ppn=1
# request 4 hours and 30 minutes of wall time
#PBS -l walltime=04:30:00
# mail is sent to you when the job begins and when it exits or aborts
# you can use all or some or none. If you don't want email leave this
# and the following (#PBS -M) out of the script.
#PBS -m bea
# specify your email address
#PBS -M John.Smith@dartmouth.edu
# By default, PBS scripts execute in your home directory, not the
# directory from which they were submitted. The following line
# places you in the directory from which the job was submitted.
cd $PBS_0_WORKDIR
# run the program
./program_name arg1 arg2 ...
01/29/18
```



# Using The Scheduler

- *mksub (qsub) pbs\_script\_filename*
- *myjobs [-rn]*
- *qshow [-r]*
- *pbsmon*
- *checkjob [-v] jobID*
- *qr*
- *qdel jobID*
- *qnotify*

submit job

view job(s) status

view queue status

view nodes & status

view job(s) status

view your resources

remove job

notify near run end

# Things to Check Before Job Submission

- Have I saved all results (data and graphics)?
- Have I requested enough time?
  - #PBS -l walltime=2:00:00 (hr:min:sec)
- Have I requested enough cores?
  - Specify 1 core per 4GB of memory usage
- Have I specified any other needed features?
  - #PBS -l feature='cellk'

# Diagnosing Problems

## Blocked jobs

- Use **checkjob -v** see the reason
- Try changing parameters and resubmitting

## Jobs that do not return results

- Contact [research.computing@dartmouth.edu](mailto:research.computing@dartmouth.edu)

## Out of disk space (quota)

- The **quota** command will show your usage
- /scratch can also fill up (have job clean up)
- This condition can cause errors that are very hard to diagnose

# Scheduler Etiquette

Our goal is to provide fair use of the resources

Stage large quantity job submissions

- If more than 600 jobs, use the **largeq** (routing queue)

To maximize your use of the available resources

- Start modestly - test new or unfamiliar code
- Use test nodes x01, x02 or x03 for testing and timing
- *Use top or htop on Test nodes to check performance*

# Scheduler Etiquette II

- To maximize your use of the available resources (cont'd)
  - Know your code and what your cluster resources are
    - The **qr** (queue resources) command can help
  - Know cluster policies on runtime and resource limitations
    - available on the Discovery website
    - <http://discovery.dartmouth.edu>
  - Plan ahead for long jobs
    - Are the resources available?
  - If possible, compile code on the cluster
  - Ask us ([research.computing@dartmouth.edu](mailto:research.computing@dartmouth.edu))
    - if you must run in an unusual way

# Discovery: Helpful Commands

- `myjobs [-rbi]`
- `tnodeload`
- `quota`
- `pbsmon`
- `features [-h] [-a] <feature>`
- `qr [-h]`
- `qshow [-r]`
- `qnotify job-id hour(s)`

# myjobs

- myjobs [-rn]

\$ myjobs

active jobs-----

JOBID	USERNAME	STATE	PROCS	REMAINING	STARTTIME
3810851	ryanu	Running	1	14:09:05	Mon Mar 22 02:55:08
3810867	ryanu	Running	1	14:38:28	Mon Mar 22 03:24:31
3810873	ryanu	Running	1	14:52:15	Mon Mar 22 03:38:18

3 active jobs                    3 of 1548 processors in use by local jobs (0.33%)  
                                  88 of 114 nodes active                    (77.19%)

eligible jobs-----

JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUETIME
-------	----------	-------	-------	---------	-----------

0 eligible jobs

blocked jobs-----

JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUETIME
3811629	ryanu	Idle	1	1:00:00:00	Mon Mar 22 09:59:23
3811630	ryanu	Idle	1	1:00:00:00	Mon Mar 22 10:00:23
3811633	ryanu	Idle	1	1:00:00:00	Mon Mar 22 10:07:53

3 blocked jobs

Total jobs: 6

# tnodeload

**\$ tnodeoad**

Node	Users	Load	Memory	Scratch	Speed	Max	Chip Set
x01	0	0.04	64.5G	779G	2.4GHz	2.4GHz	AMD Opteron(tm) Processor 6136
x02	0	0.00	64.5G	779G	2.4GHz	2.4GHz	AMD Opteron(tm) Processor 6136
x03	1	0.00	64.6G	779G	2.4GHz	2.4GHz	AMD Opteron(tm) Processor 6136



quota

\$ quota

User: pete

-----

Quota: 20G

Used: 12G

Available: 8.7G

Use: 57%

quota

\$ quota

User: pete

-----

Quota: 20G

Used: 19G

Available: 2.0G

Use: 95%

# pbsmon

```
a01 a02 a03 a04 a13 a14 a15 a16 a17 a18 a19 a20 a21
b01 b02 b03 b04 b05 b06 b07 b08 b09 b10 b11 b12 b13 b14 b15 b16
c01 c02 c03 c04 c05 c06 c07 c08 c09 c10 c11 c12 c13 c14 c15 c16
c17 c18 c19 c20 c21 c22 c23 c24 c25 c26 c27
d01 d02 d03 d04 d05 d06 d07 d08 d09 d10 d11 d12 d13 d14 d15 d16
d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30 d31 d32
d33 d34 d35 d36 d37 d38 d39
e01 e02 e03 e04 e05 e06 e07 e08 e09 e10 e11 e12 e13 e14 e15 e16
e17 e18 e19 e20 e21 e22 e23 e24 e25 e26 e27 e28 e29 e30 e31 e32
e33 e34 e35
f01 f02 f03 f04 f05 f06 f07 f08
g01 g02
h01 h02 h03 h04 h05 h06 h07 h08
x01 x02 x03
```

---

```
nodes free           : 54      nodes down           : 9
<= 50% cores in use : 12      100% cores in use  : 64
> 50% cores in use  : 12      Total cores in use  : 1134
```

# features

```
[pete@discovery ~]$ features -a
```

Feature	Total Cores	Avail Cores	Free Nodes
cella	104	0	0
cellb	128	2	0
cellc	432	211	8
celld	624	202	9
celle	560	486	28
cellf	384	334	6
cellh	128	0	0
ib2	384	32	2
amd	1720	1201	49
intel	256	2	0
-----			
Totals	1976	1203	49

# features II

```
[pete@discovery ~]$ features -h
```

```
Syntax: features [-a] [-h] [-f feature]
```

Providing the -h option prints this help message.

If given the option "-a" then all features and their available resources are displayed. The Totals are the count of both the amd and intel features

If given one of the following features as an argument to "-f", then that feature's available resources will be displayed.

```
Features Available: cella cellb cellc celld celle cellf cellh ib2 amd intel
```

```
Current Feature Assignments:
```

a01-a04,a13-a21:	cella,amd	(Opteron 2.7Ghz 32G RAM 8-cores)
b01-b16:	cellb,intel	(Xeon Nahalem 2.3Ghz 32G RAM 8-cores)
c01-c27:	cellc,amd	(Opteron 2.4Ghz 64G RAM 16-cores)
d01-d39:	celld,amd	(Opteron 3.0Ghz 64G RAM 16-cores)
d01-d24:	ib2	(Infiniband)
e01-e34:	elle,amd	(Opteron 3.1Ghz 64G RAM 16-cores)
f01-f08:	cellf,amd	(Opteron 2.8Ghz 192G RAM 48-cores)
h01-h08:	cellh,intel	(Xeon 2.5Ghz 64G RAM 16-cores)

# qr (queue resources)

```
pete@discovery:~ — ssh — 68x23
[pete@discovery ~]$ qr

Queue Resources for pete on Fri Mar 14 18:06:01 EDT 2014

Account/User Resources
      Owned      MAX      UserMAX      MAX      UserMAX
Account CPUs      CPUs      CPUs      Wall      Wall
  Moore   796     1811      400     204480     102240

Account Usage
      Wall      Jobs      CPUs      FS %
  Moore  44109      895     1270     35.0

pete's Usage/Availability
      Rem  Running  Used
      Wall  Jobs    CPUs
      0      0      0

pete's Blocked Jobs
      Wall  Jobs    CPUs
      0      0      0
```

# qshow

```
pete@discovery:~ — ssh — 59x22
[pete@discovery ~]$ qshow
```

User	Running		Blocked		Eligible	
	Jobs	CPUs	Jobs	CPUs	Jobs	CPUs
-----	-----	-----	-----	-----	-----	-----
aglaser	5	80	0	0	0	0
bzhu	3	48	1	16	0	0
ccheng	2	2	0	0	0	0
chandana	2	48	0	0	0	0
denton	1	48	0	0	0	0
dfisher	1	64	0	0	0	0
ebrahimi	1	16	0	0	0	0
pandrews	75	75	0	0	0	0
piotr	2	2	0	0	0	0
qpan	400	400	198	198	0	0
rhughes	6	24	0	0	0	0
robertd	18	72	0	0	0	0
ryanu	400	400	200	200	0	0
rzhang	11	110	2	20	0	0
tingh	25	400	68	1088	0	0
-----	-----	-----	-----	-----	-----	-----
Total	952	1789	469	1522	0	0

# qnotify

```
$ qnotify
```

```
Syntax: qnotify job-id hours  
        qnotify -l (list notifications)
```

```
$ qnotify 3872942 1
```

QNotify will notify you when there are about 1 hours of walltime remaining on job 3872942.

```
$ qnotify -l
```

JobID	Remaining	Notify
3872942	1:59:20	1



# qshow -r

```
pete@discovery:~ -- ssh -- 69x22
[pete@discovery ~]$ qshow -r
```

User	Running		Blocked		Eligible		Routing
	Jobs	CPUs	Jobs	CPUs	Jobs	CPUs	Jobs
aglaser	5	80	0	0	0	0	0
bzhu	3	48	1	16	0	0	0
ccheng	2	2	0	0	0	0	0
chandana	2	48	0	0	0	0	0
denton	1	48	0	0	0	0	0
dfisher	1	64	0	0	0	0	0
ebrahimi	1	16	0	0	0	0	0
pandrews	75	75	0	0	0	0	0
piotr	2	2	0	0	0	0	0
qpan	400	400	199	199	0	0	11351
rhughes	6	24	0	0	0	0	0
robertd	18	72	0	0	0	0	0
ryanu	400	400	200	200	0	0	633
rzhang	11	110	2	20	0	0	0
tingh	25	400	68	1088	0	0	0
Total	952	1789	470	1523	0	0	11984

# Summary

- Cluster introduction
- Connecting/Transferring data
- Environment settings
- Submitting jobs (PBS script, qsub)
- Checking jobs
- Usage policies and etiquette overview
  - submitting jobs etiquette
  - monitoring disk usage