Introduction to Discovery

The Discovery Cluster

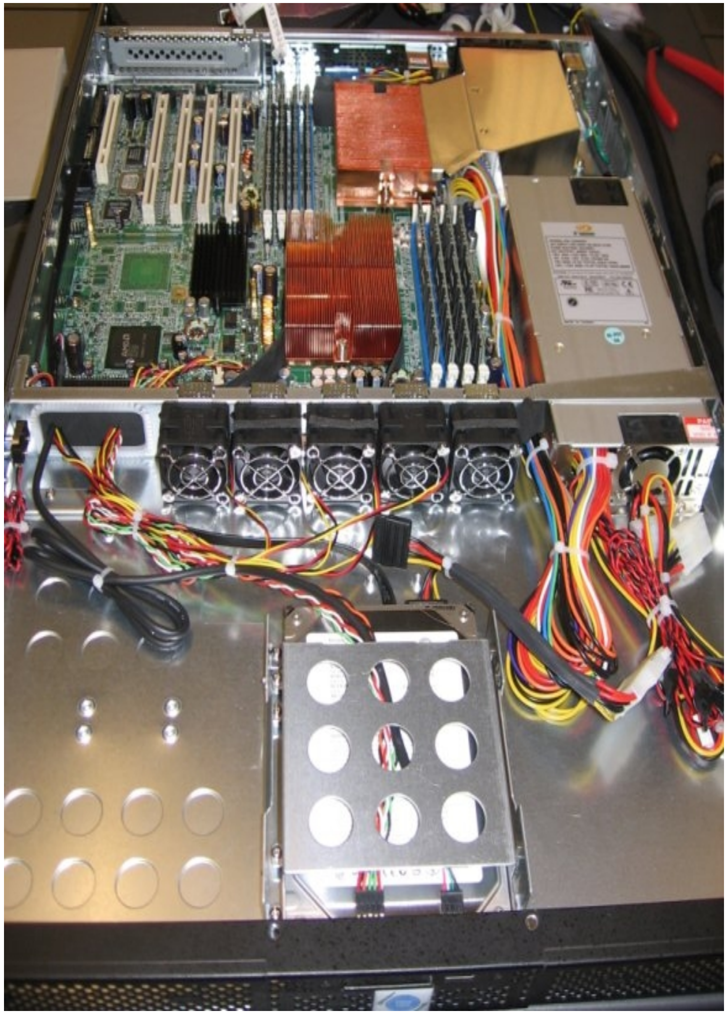
Agenda

- 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 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)

Cell N: Intel Xeon Gold 6148 2.4 GHz Dual 20-core (40 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@discovery7.dartmouth.edu
- Mac: ssh -Y username@discovery7.dartmouth.edu
- Install Xquartz for graphical interface
- Windows
  - MobaXterm built in Xserver and sftp (free and recommended)
  - Ssh secure shell or putty

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

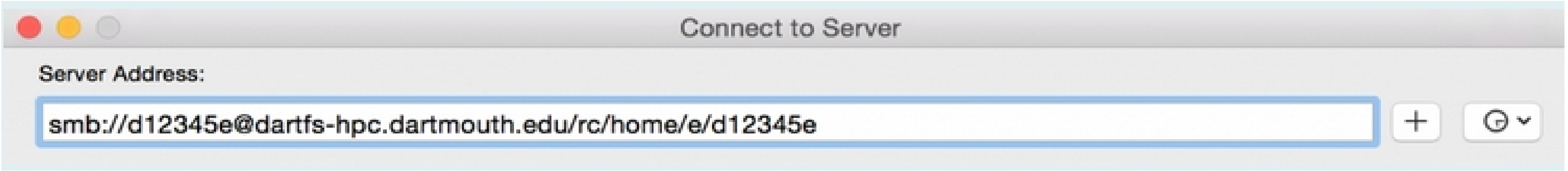
Advantages of New DartFS File System:

- 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:

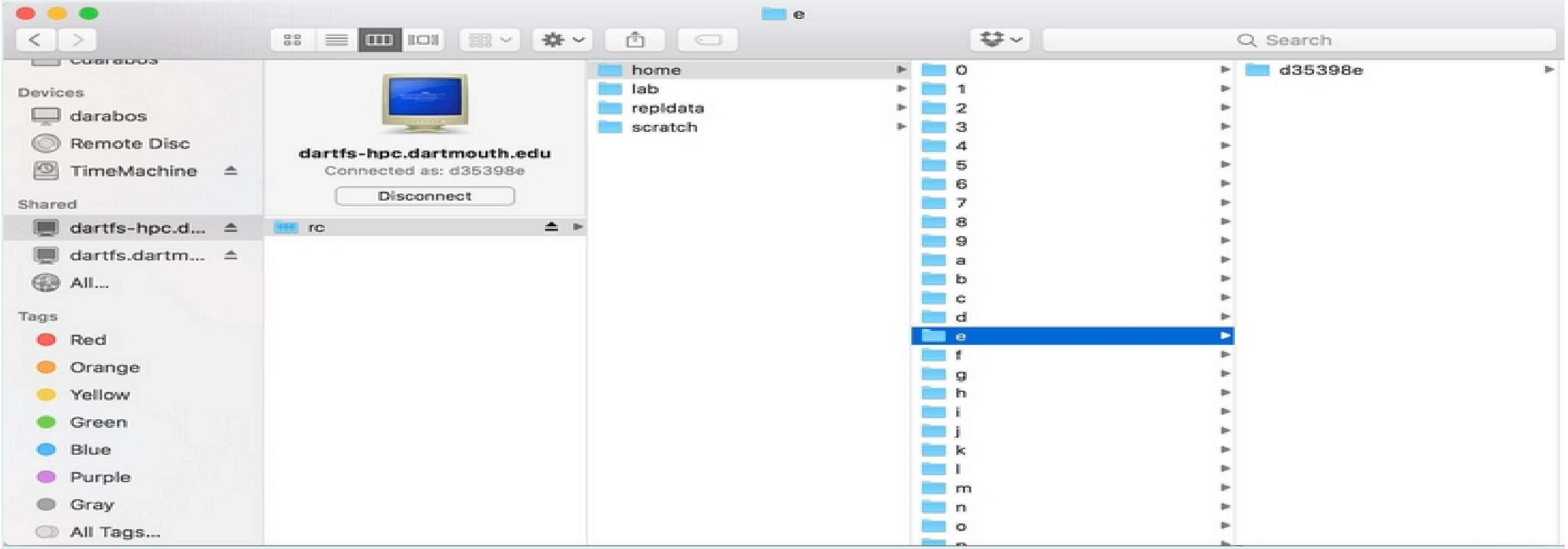
Finder-> Go-> Connect to Server



Connect to Server

Server Address:

smb://d12345e@dartfs-hpc.dartmouth.edu/rc/home/e/d12345e

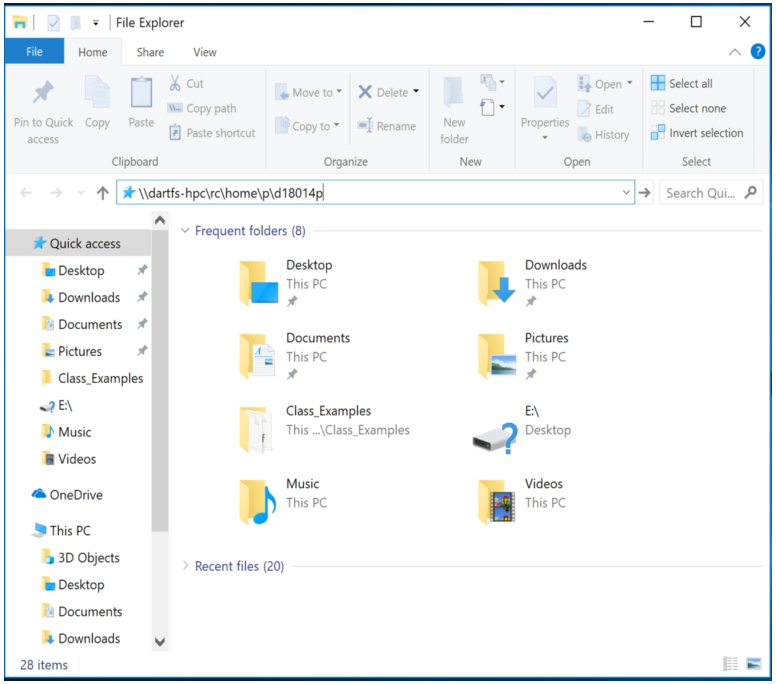


dartfs-hpc.dartmouth.edu

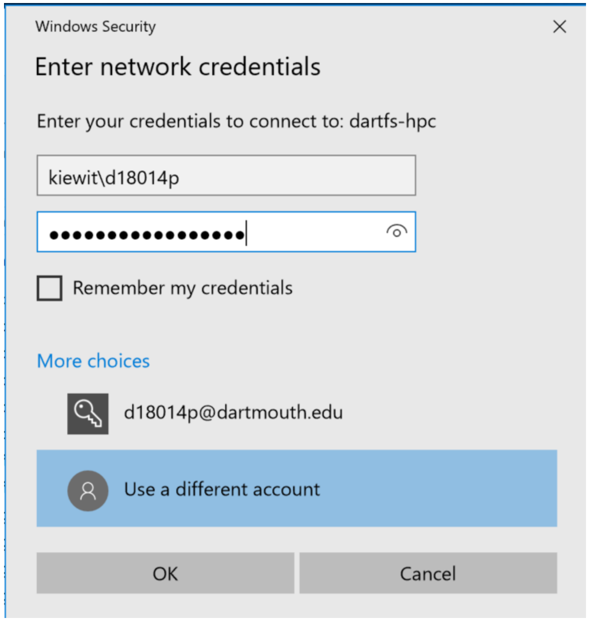
Connected as: d35398e

How To SMB Mount DartFS Files on Windows

- Open File Explorer and enter \\dartfs-hpc\rc\home\p\d18014p
- Enter your netid as **kiewit\d18014p** and your netid password
- You can also map a network drive



\\dartfs-hpc\rc\home\p\d18014p



Windows Security

Enter network credentials

Enter your credentials to connect to: dartfs-hpc

kiewit\d18014p

•••••••••••••••••

□ Remember my credentials

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 when 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 ...]
```

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 45 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**
    - Email Research Computing to have **public_html** created**.**
- URL: **`http://rcweb.dartmouth.edu/homes/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 then 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_O_WORKDIR
# run the program
./program_name arg1 arg2 ...
```

Using The Scheduler

mksub pbs_script_filename	submit job
myjobs [-rn]	view job(s) status
qshow [-r]	view queue status
pbsmon	view nodes & status
checkjob [-v] jobID	view job(s) status
qr	view your resources
qdel *jobID*	remove job
qnotify	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

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 then 600 jobs, use the **largeq** (routing queue)

To maximize your use of the available resources

- Start modestly - test new or unfamiliar code
- Use test node x01 for testing and timing
- *Use **top** or **htop** 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*)
    - if you must run in an unusual way

Discovery: Helpful Commands

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

myjobs

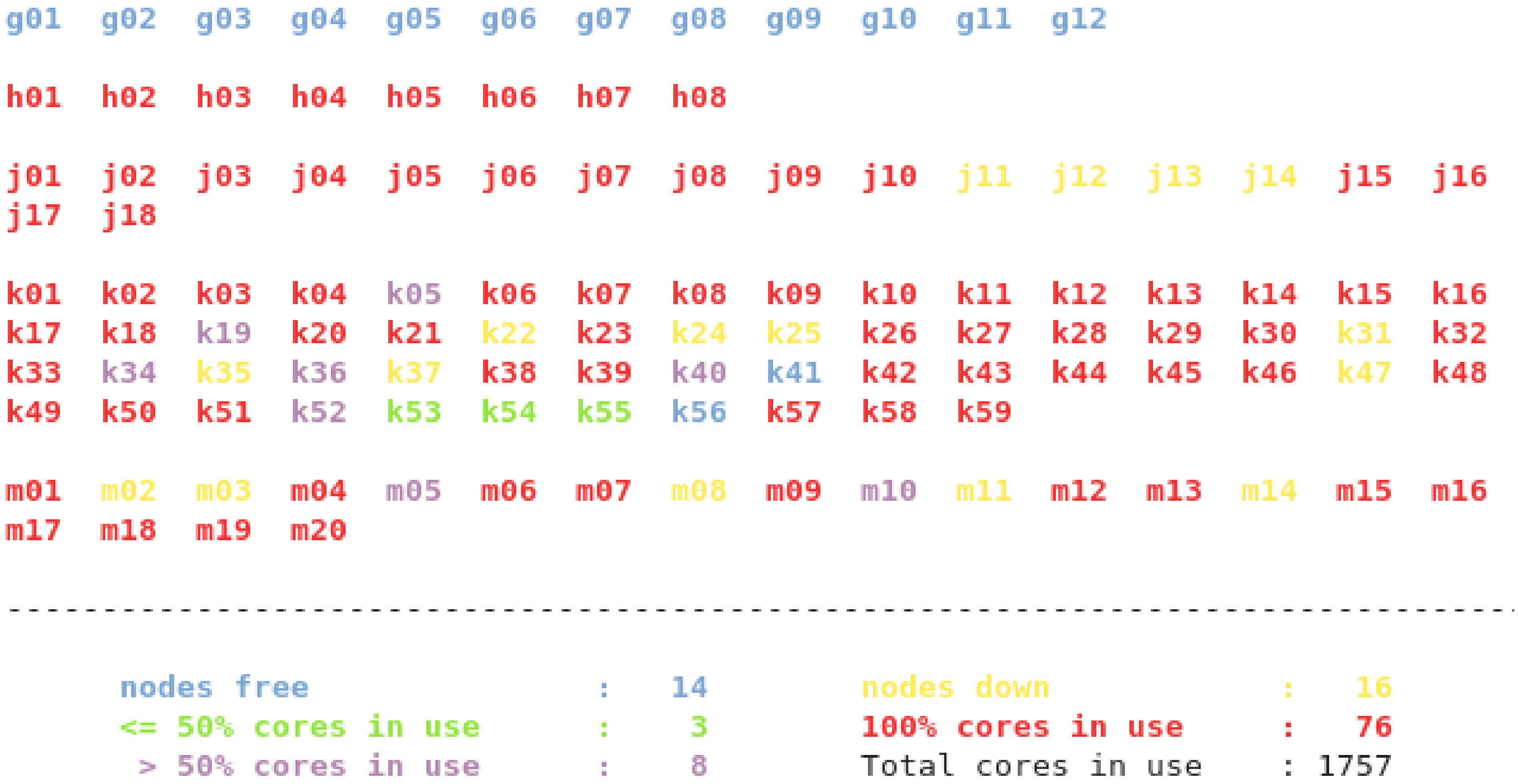
- `myjobs [-rn]`

```
$ 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
```

quota

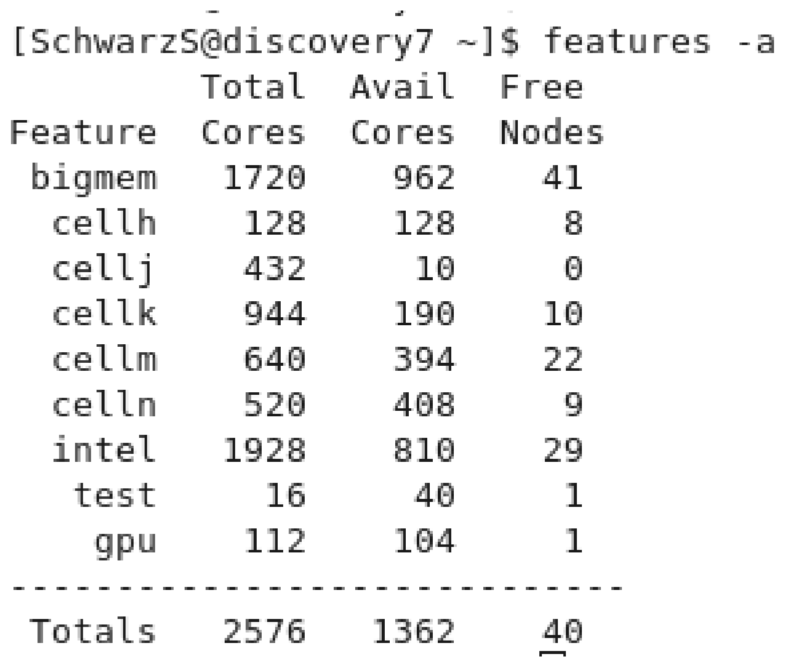
```
$ quota

        User: pete
        ----- ----
       Quota: 20G
        Used: 12G
   Available: 8.7G
         Use: 57%
```

pbsmon



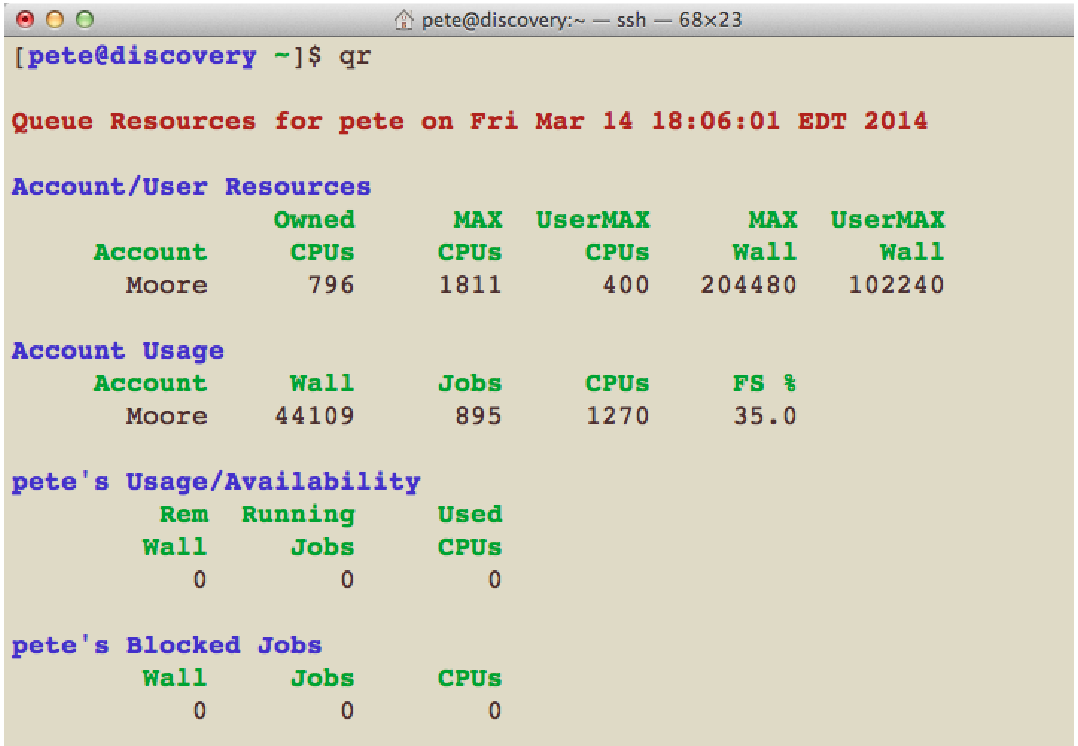
```
g01   g02   g03   g04   g05   g06   g07   g08   g09   g10   g11   g12

h01   h02   h03   h04   h05   h06   h07   h08

j01   j02   j03   j04   j05   j06   j07   j08   j09   j10   j11   j12   j13   j14   j15   j16
j17   j18

k01   k02   k03   k04   k05   k06   k07   k08   k09   k10   k11   k12   k13   k14   k15   k16
k17   k18   k19   k20   k21   k22   k23   k24   k25   k26   k27   k28   k29   k30   k31   k32
k33   k34   k35   k36   k37   k38   k39   k40   k41   k42   k43   k44   k45   k46   k47   k48
k49   k50   k51   k52   k53   k54   k55   k56   k57   k58   k59

m01   m02   m03   m04   m05   m06   m07   m08   m09   m10   m11   m12   m13   m14   m15   m16
m17   m18   m19   m20

-------------------------------------------------------------------------------

      nodes free                 :    14        nodes down                :    16
      <= 50% cores in use        :     3        100% cores in use         :    76
       > 50% cores in use        :     8        Total cores in use        : 1757
```

features



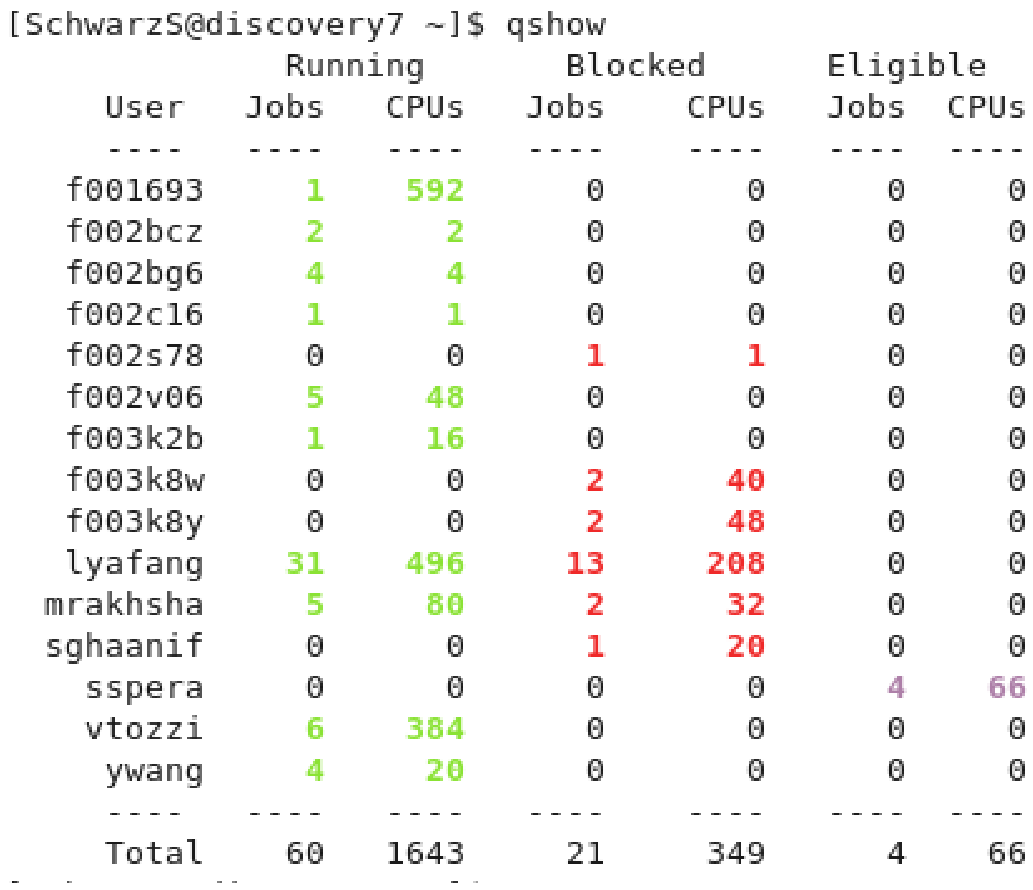
```
[SchwarzS@discovery7 ~]$ features -a
         Total  Avail  Free
Feature  Cores  Cores  Nodes
 bigmem   1720    962    41
  cellh    128    128     8
  cellj    432     10     0
  cellk    944    190    10
  cellm    640    394    22
  celln    520    408     9
  intel   1928    810    29
   test     16     40     1
    gpu    112    104     1
-----------------------------
 Totals   2576   1362    40
```

qr   (queue resources)



```
[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
     Account     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

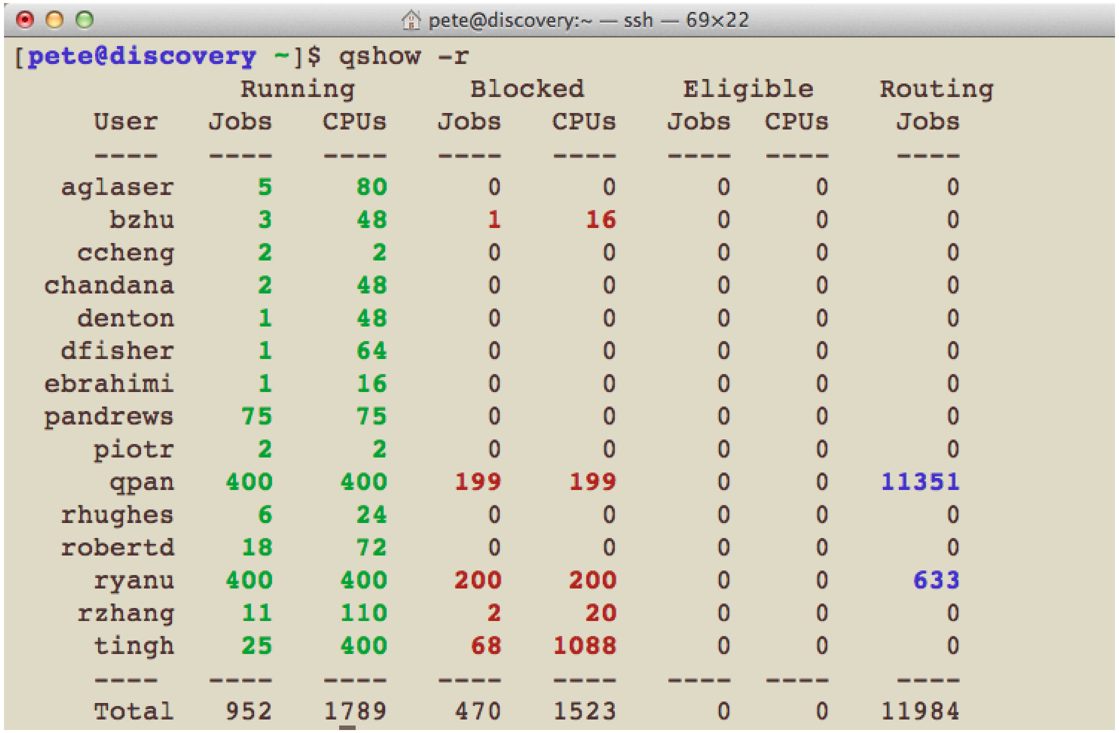


```
[SchwarzS@discovery7 ~]$ qshow
             Running          Blocked        Eligible
     User    Jobs    CPUs    Jobs     CPUs    Jobs   CPUs
     ----    ----    ----    ----     ----    ----   ----
   f001693      1     592       0        0       0      0
   f002bcz      2       2       0        0       0      0
   f002bg6      4       4       0        0       0      0
   f002c16      1       1       0        0       0      0
   f002s78      0       0       1        1       0      0
   f002v06      5      48       0        0       0      0
   f003k2b      1      16       0        0       0      0
   f003k8w      0       0       2       40       0      0
   f003k8y      0       0       2       48       0      0
   lyafang     31     496      13      208       0      0
  mrakhsha      5      80       2       32       0      0
  sghaanif      0       0       1       20       0      0
    sspera      0       0       0        0       4     66
    vtozzi      6     384       0        0       0      0
     ywang      4      20       0        0       0      0
     ----    ----    ----    ----     ----    ----   ----
     Total     60    1643      21      349       4     66
```

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 ~]$ qshow -r
              Running          Blocked        Eligible     Routing
     User    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