Introduction to Discovery

http://discovery.dartmouth.edu

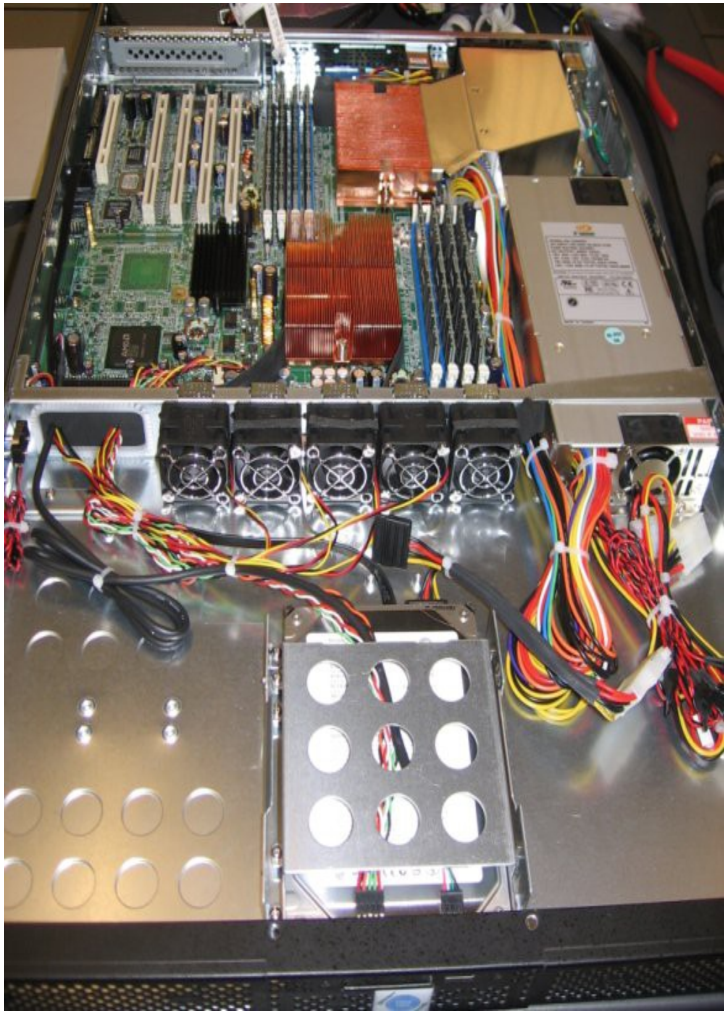
October 2014

The Discovery Cluster

Agenda

- Resource overview
- 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
- Getting help
- Cluster Etiquette - running jobs & disk space
- Publishing
- Labs

Cluster Nodes



A) AMD 2384 2.7GHz Dual Quad-Core (8-cores)

B) Intel Xeon L5520 2.3GHz Dual Quad-Core (8-cores)

C) AMD 6136 2.4GHz Dual Octa-Core (16-cores)

D) AMD 4284 3.0GHz Dual Octa-Core (16-cores)

E) AMD 4286 3.1GHz Dual Octa-Core (16-cores)

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

G) 3 GPU Nodes

H) Intel Xeon E5-2470 2.xGHz Dual Octa-Core (16-cores)

Logging On

- SSH (Secure Shell)
  - Linux: ssh -X username@discovery.dartmouth.edu
  - Mac: ssh -X username@discovery.dartmouth.edu
    - On Lion and later, this should start a local X-server
  - Windows
    - Use secure shell or putty
      - http://www.dartmouth.edu/comp/soft-comp/software/downloads/windows/ssh-sftp/
      - http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
    - Use xwin32 for GUI
      - https://caligari.dartmouth.edu/downloads/x-win32/

- Changing your password
  - Use the **`passwd`** `command to make the change.`

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)

- Windows GUI SSH/SFTP clients:
    - **SecureShell**
        - http://www.dartmouth.edu/comp/soft-comp/software/downloads/windows/ssh-sftp/index.html
    - **Cyberduck**
        - http://cyberduck.io
    - **WinSCP**
        - http://winscp.net/eng/download.php
- Macintosh
    - **Cyberduck**
        - http://cyberduck.ch/

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 ...]
```

Andrew File System: AFS

- 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.
  - /global/scratch (central scratch)
    - Data in /scratch and /global/scratch cleaned by the system after 7 days.
  - /global/data (members data space)
    - If you are part of a member's Discovery account (**qr** command)
- Home directories backed up daily offsite
  - Snapshots taken daily,weekly & monthly and are available in your
    - $HOME/.zfs/snapshot directory (if home path is /home or /cgl/home)
    - $HOME/.snapshot (if home path is /ihome)

Disk Space II

Disk quotas

- $HOME (20GB)
  - Email sent if quota usage reaches 95%
  - Use **quota** command to view your usage
- /global/data
  - Quota dependent on members investment
- /scratch (no quota enforced)
  - Please have job cleanup
- /global/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**.
- The directory should be created as follows:
  -
    ```
    $ mkdir -m 711 ~/public_html
    ```
- URL: `http://discovery.dartmouth.edu/~username/`

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

qsub *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

Diagnosing Problems

Blocked jobs

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

Jobs that do not return results

- Contact *help@discovery.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 nodes x01, x02 or x03 for testing and timing
- Use pmap to test memory usage.
    - Last line will show total usage for process in Kbytes
    - pmap <*process-id*>

Test cluster with new or unfamiliar code

Learn scheduler commands for checking job and queue status

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 (*help@discovery.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

```
$ tnodeload
```

```
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
```

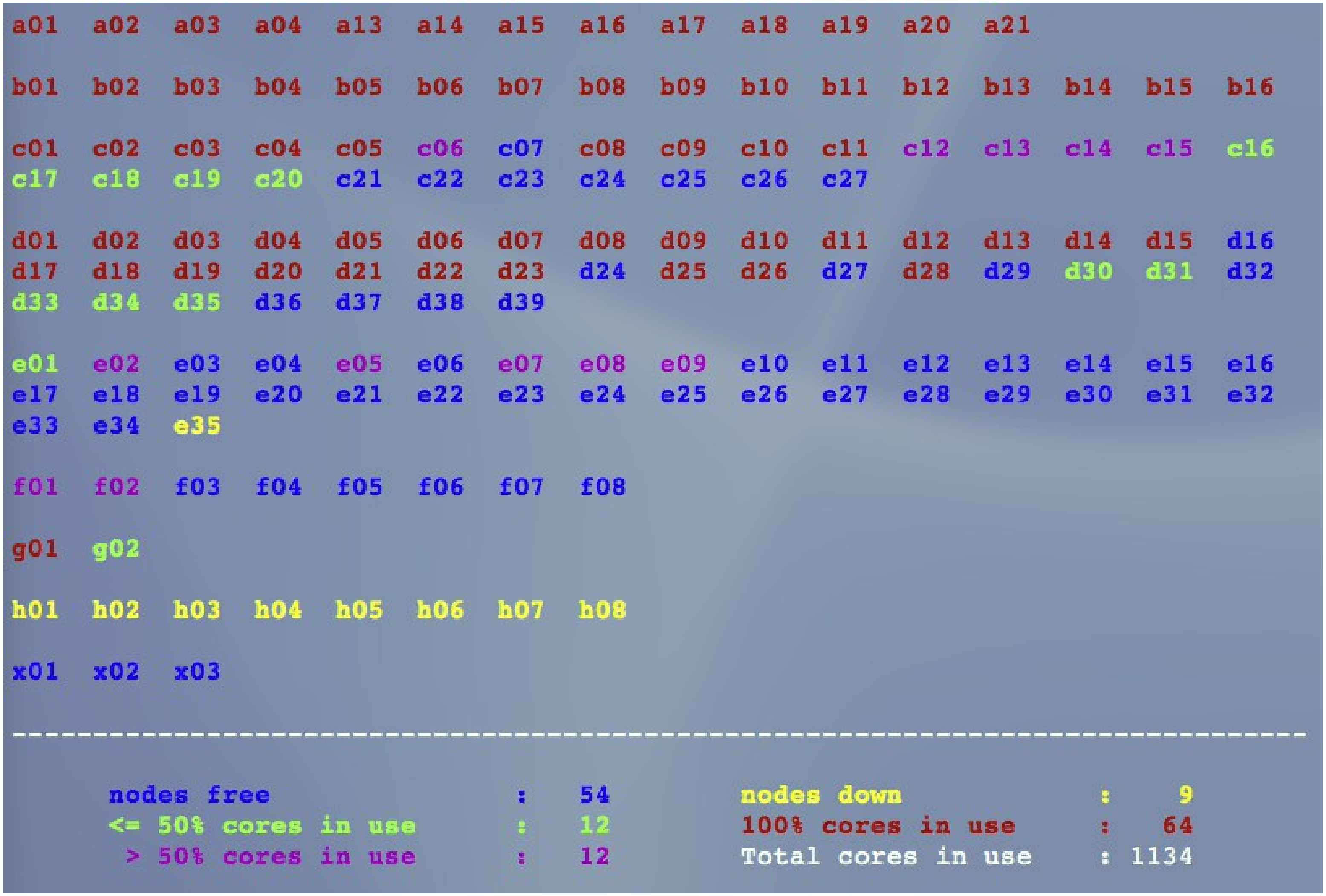
quota

```
$ quota

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

quota

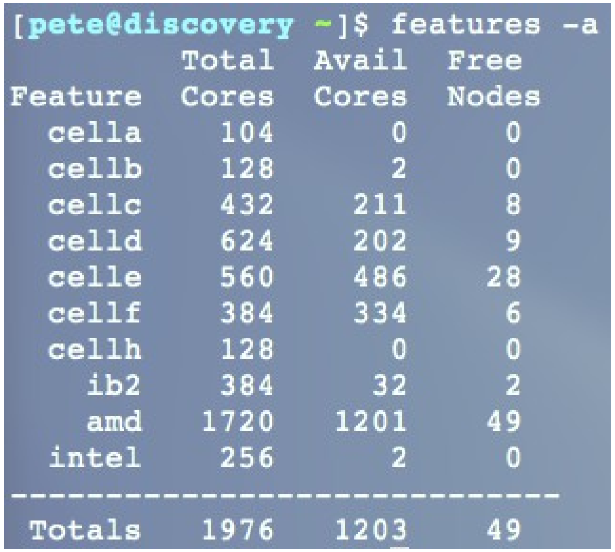
```
$ quota

        User: pete
        ----- ----
       Quota: 20G
        Used: 19G
   Available: 2.0G
         Use: 95%
```

pbsmon



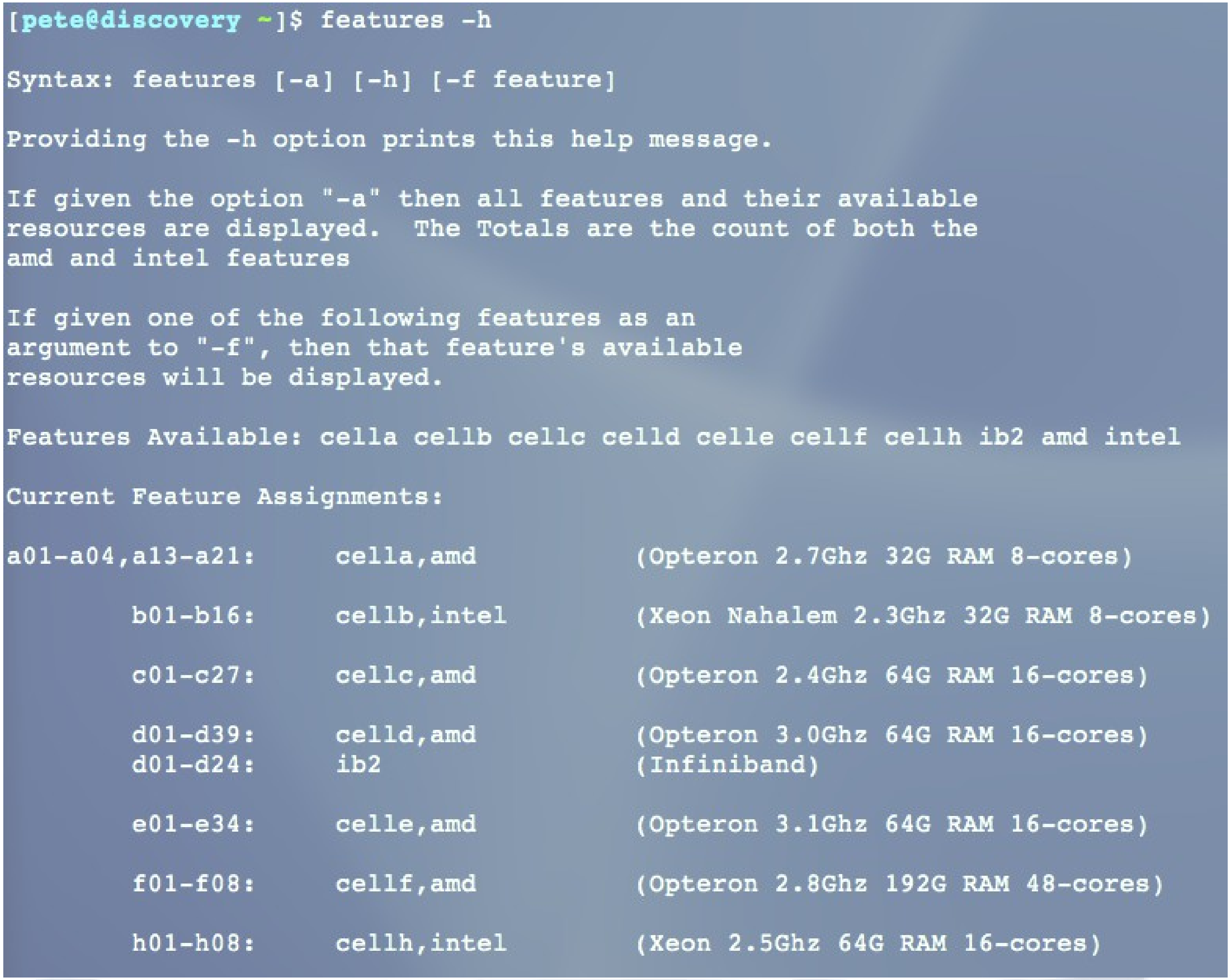
```
a01  a02  a03  a04  a13  a14  a15  a16  a17  a18  a19  a20  a21

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



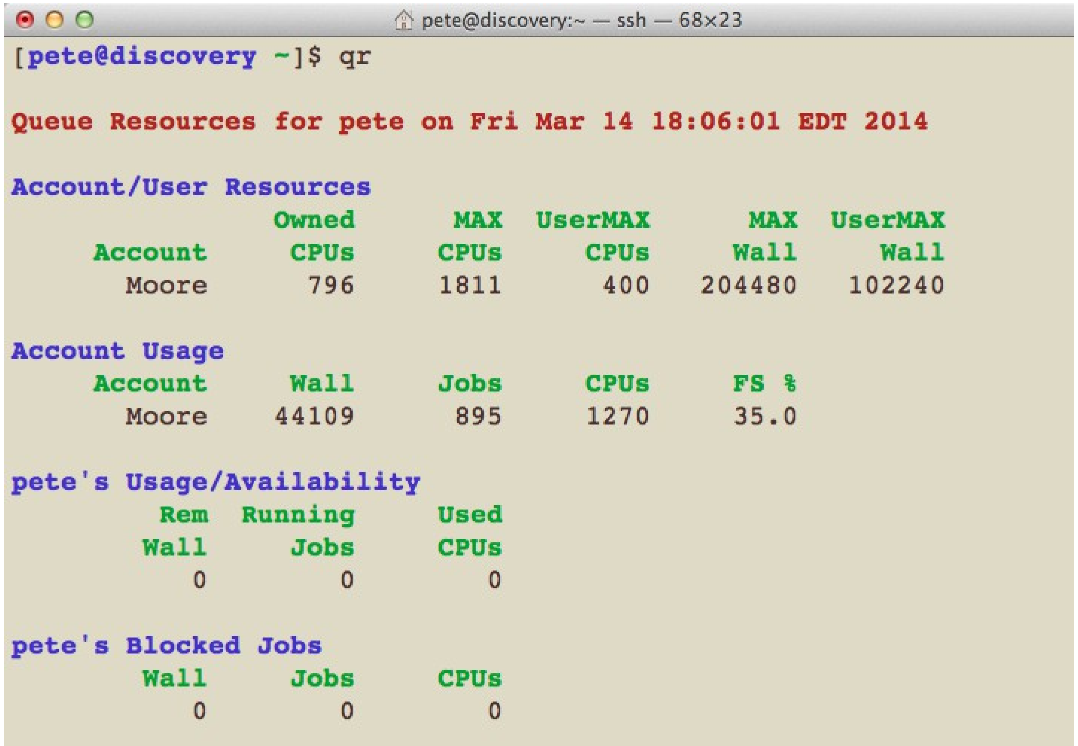
```
[pete@discovery ~]$ features -a
         Total  Avail  Free
Feature  Cores  Cores  Nodes
  cella    104      0     0
  cellb    128      2     0
  cellc    432    211     8
  celld    624    202     9
  celle    560    486    28
  cellf    384    334     6
  cellh    128      0     0
    ib2    384     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:     celle,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)



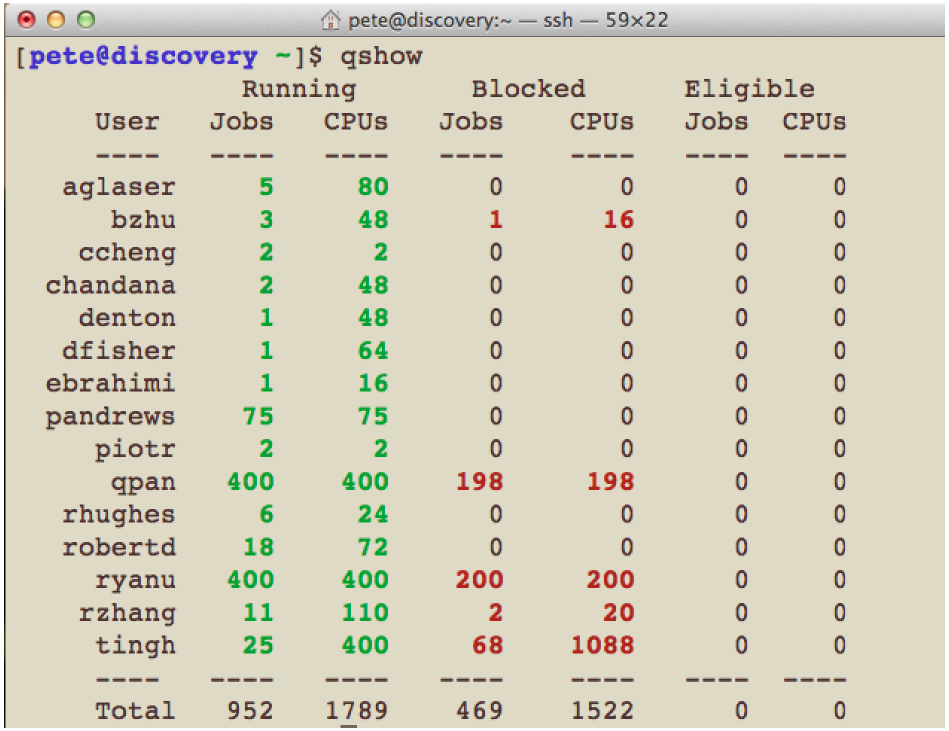
```
[pete@discovery ~]$ qr

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
```

qhist   (review past usage)

The purpose of qhist is to provide statistics on past cluster usage. It can report a summary or line-by-line report for a time period. It can also provide a more detailed report on a single job.

```
$ qhist -h

     qhist -h                 Syntax

     qhist -j <jobnum>        Single Job Report

     qhist -r                 Job records

     qhist -s                 Summary
```

qshow

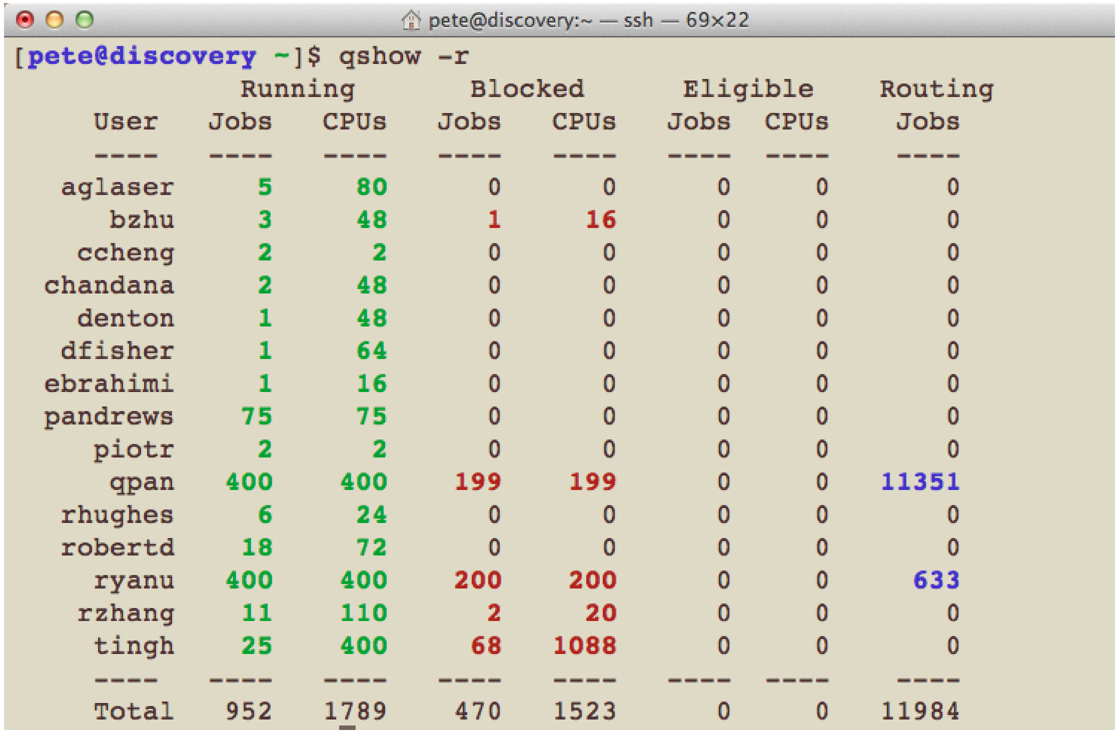


```
[pete@discovery ~]$ qshow
               Running          Blocked         Eligible
     User    Jobs   CPUs    Jobs     CPUs    Jobs   CPUs
     ----    ----   ----    ----     ----    ----   ----
  aglaser       5     80       0        0       0      0
     bzhu       3     48       1       16       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 ~]$ 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