WORKLOAD MANAGEMENT WITH GRID ENGINE

Dr. Jeffrey Frey
University of Delaware, IT

Dare to be first.
UNIVERSITY OF DELAWARE

…IN OLDEN TIMES

- A cluster was a few shelves of standard PC towers.
  - Loosely constructed
  - Order of 10^1 cores ➙
  - Handful of users

…IN OLDEN TIMES

- A cluster was a few shelves of standard PC towers.
- User has work to do, he/she "calls" a node
  - Whiteboard in the lab
  - Write-in your name, erase when done

…ORDER OF 10^3 CORES

That's a mighty large whiteboard!

MODERN CLUSTERS

- Scale of modern systems precludes the free-for-all method of resource allocation
  - Homogeneity no longer the norm
    - Coprocessors, differing memory size/core count
  - Number of users also much greater, geographically diverse

DEFINE THE PROBLEM

- Collect work from users (*jobs*)
- Match work to available hardware (*resources*)
- Arrange for execution of work (*scheduling*)

DEFINE THE PROBLEM

job |jäb| noun : encapsulation of work to be performed

- Users provide data and steps that do something with/to the data

DEFINE THE PROBLEM

resource |rē zôrs| noun : consumable work-processing materials

`nodeX`

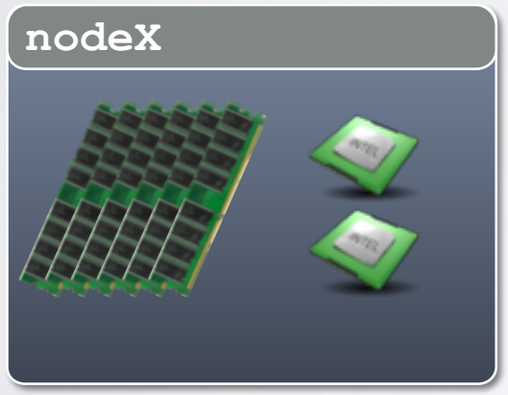
`nodeY`

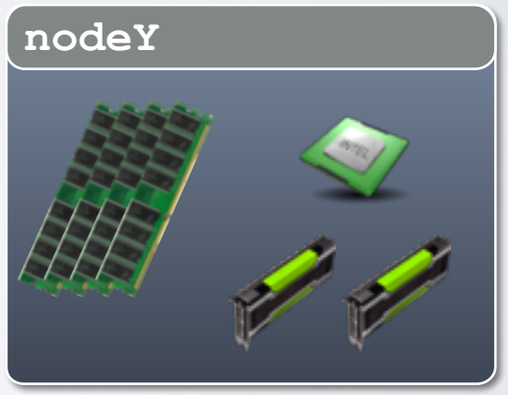
Dare to be first.
UNIVERSITY OF DELAWARE

- How much RAM? How many sockets, cores, threads?
- Any coprocessors?

DEFINE THE PROBLEM

resource |rē zôrs| noun : consumable work-processing materials





Dare to be first.

UNIVERSITY OF DELAWARE

- How much RAM? How many sockets, cores, threads?
- Any coprocessors?

DEFINE THE PROBLEM

schedule |ske jōōl| noun : a plan for processing work

Dare to be first.
UNIVERSITY OF DELAWARE

- Scheduling is a matter of packing jobs onto the resources in an optimal fashion

DEFINE THE PROBLEM

schedule |ske jōōl| noun : a plan for processing work

Dare to be first.
UNIVERSITY OF DELAWARE

- Scheduling is a matter of packing jobs onto the resources in an optimal fashion

DEFINE THE PROBLEM

- Scheduling = complex data processing
  - Jobs must indicate resource requirements and scheduling constraints
  - Scheduling needs access to resource location and ongoing availability
  - Include time as a variable in the equation
    - Deadline or planned start date for execution
    - Knowledge of future resource availability

- Job scheduling can itself become an intensive computational load

RESOURCE MANAGER

- "Knows" what resources are present and where
- Periodically checks for availability, current consumption level
- May include mechanism(s) for dispatch of work

SCHEDULER

- Maintains a list of jobs both active and waiting for execution
- Uses information from resource manager to match jobs to resources
  - Plans ahead for optimal placement…
  - …but can be influenced by per-job priority levels.

resource manager

job scheduler

SCHEDULER

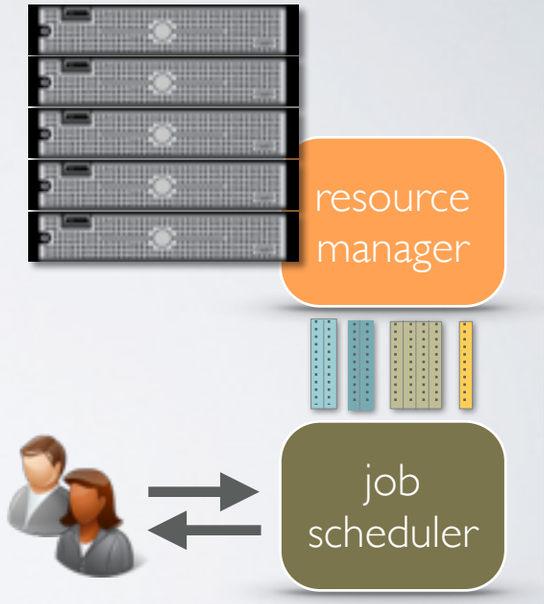
- Users interact with the job scheduler, not with the resources themselves
  - The more per-job information provided by users, the more optimal the work ⇆ resource
  - Minimize "unknown" work on resources = scheduler can better plan ahead

resource manager

job scheduler

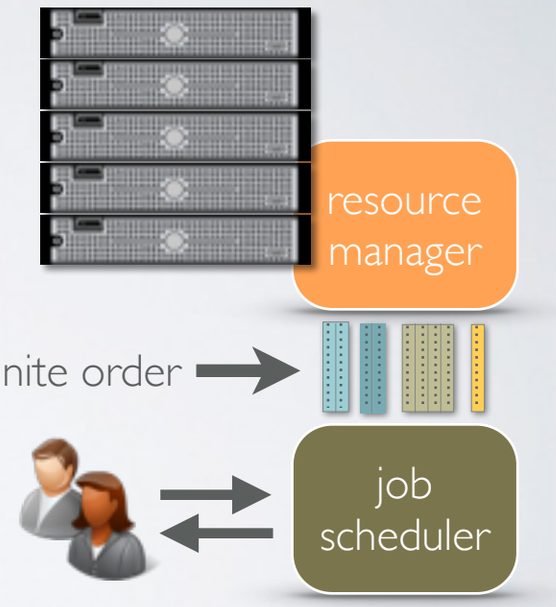
SCHEDULER

- Work is spread into lists of jobs similar in nature
  - Initially in order of submission
  - Resource availability, priority differences may cause reordering of the list



- These lists feed jobs to the resource manager (or directly to resources)

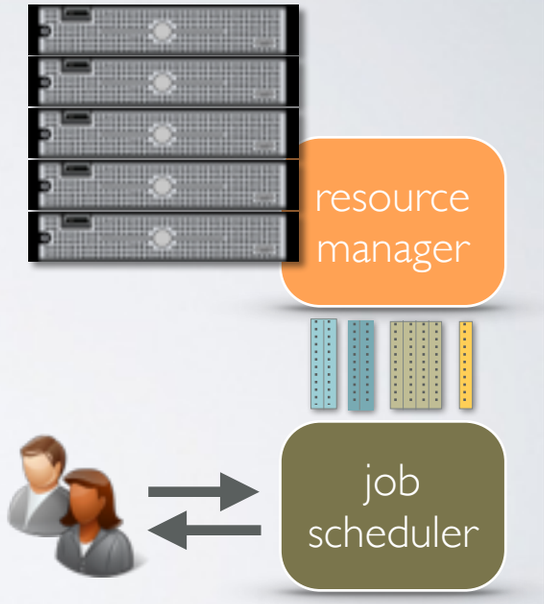
SCHEDULER



queue |kyōō| noun : a list of items in definite order

SCHEDULER

- A queue is parameterized to determine
  - what jobs will be accepted
    - which users are allowed
  - how many jobs can be present
  - what resources will process the jobs



GRID ENGINE

- One solution to the problem of automated job scheduling and resource management
- Originally SUN Grid Engine
- SUN bought by Oracle, Oracle Grid Engine
- Oracle sells to Univa, Univa Grid Engine (what we use on Farber)
- Several non-commercial variants thanks to open-source spinoff of the source by SUN many years ago
    - OpenGridScheduler (what we use on Mills)

WHAT IS GRID ENGINE?

- Grid Engine functions as both a resource manager and a job scheduler

WHAT IS GRID ENGINE?

- Compute nodes run an *execution dæmon*
  - Periodically transmits resource info/usage
  - Starts programs on behalf of a user

WHAT IS GRID ENGINE?

- Compute nodes run an
  - Periodically transmits resource info/usage
  - [illegible] half of

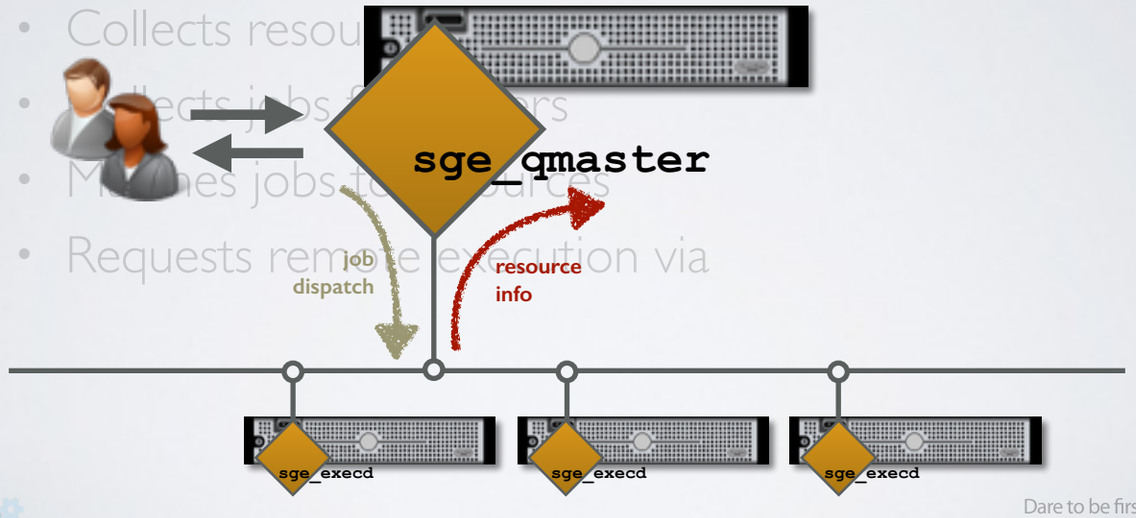
WHAT IS GRID ENGINE?

- Scheduler node(s) run a *queue master dæmon*
  - Collects resource information from *sge_execd*'s
  - Collects jobs from users
  - Matches jobs to resources
  - Requests remote execution via *sge_execd*'s

- The daemons communicate via a network of some sort — typically, standard ethernet

WHAT IS GRID ENGINE?

- Scheduler node(s) run a
  - Collects resou
  - ects jobs f rs
  - M nes jobs to urces
  - Requests remote execution via



- The daemons communicate via a network of some sort — typically, standard ethernet

RESOURCE INFORMATION

- Grid Engine categorizes resource information into types
  - String, integer, real, boolean, date/time, memory
- Each data point (a *complex*) is given a name
- Each *sge_execd* communicates values for applicable *complexes*

RESOURCE INFORMATION

complex	type	description
`m_mem_free`	memory	memory required to be unused
`m_mem_total`	memory	total memory present in node
`scratch_free`	memory	local scratch disk space available
`hostname`	host	name of the node
`h_rt`	time	hard runtime limit for job
`exclusive`	bool	job uses empty nodes only

RESOURCE INFORMATION

```
[(user:group)@farber ~]$ qhost -h n000 -F m_mem_free,m_mem_total,scratch_free,hostname,h_rt,exclusive
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR NLOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -       -
n000                    lx-amd64       20    2   20   20  0.00   63.0G    1.1G    2.0G     0.0
    Host Resource(s):      hl:scratch_free=433.100G
   hl:m_mem_total=63.955G
   hl:m_mem_free=62.832G
   hc:exclusive=1.000000

[(user:group)@farber ~]$
```


- query per-node resource info using the qhost command
- Grid Engine commands use a letter "q" prefix (short for…queue)
- NLOAD: normalized system load (load divided by processor count)
- h_rt and hostname not displayed? no value in this context

RESOURCE INFORMATION

```
[(user:group)@farber ~]$ qhost -h n021 -j
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR NLOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -       -
n021                    lx-amd64       20    2   20   20  0.57   63.0G   29.5G    2.0G   37.4M
   job-ID     prior   name       user         state submit/start at     queue      master ja-task-ID
   ----------------------------------------------------------------------------------------------
        40618 0.53022 T21Het6.   user_ex      r     03/02/2015 16:50:02 hmichael.q MASTER 50
        41013 0.52980 T30Het6.   user_ex      r     03/03/2015 12:24:57 hmichael.q MASTER 17
        41568 0.52815 T21Het5.   user_ex      r     03/07/2015 07:02:02 hmichael.q MASTER 40
        42188 0.52623 T71Het1.   user_ex      r     03/10/2015 11:46:59 hmichael.q MASTER 36
        45919 0.52155 T30Het5.   user_ex      r     03/19/2015 13:39:12 hmichael.q MASTER 5
        45919 0.52155 T30Het5.   user_ex      r     03/19/2015 13:39:42 hmichael.q MASTER 17
        45921 0.52154 T30Het6.   user_ex      r     03/19/2015 13:52:57 hmichael.q MASTER 9
        45924 0.52154 T31Het6.   user_ex      r     03/19/2015 13:57:27 hmichael.q MASTER 2
        59939 0.50655 D10Het5.   user_ex      r     04/17/2015 15:30:47 hmichael.q MASTER 11
        59940 0.50655 D10Het6.   user_ex      r     04/17/2015 15:31:32 hmichael.q MASTER 2
        59941 0.50655 D10Het6.   user_ex      r     04/17/2015 15:31:47 hmichael.q MASTER
```

Dare to be first.
UNIVERSITY OF DELAWARE

- NLOAD of 0.57 means ~ 11 cores in use (20 * 0.57 = 11.4)
- Use the "-j" flag to see what jobs are using the node

RESOURCE INFORMATION

- Mills/Farber nodes are owned by workgroups
  - How do I know which nodes are mine?

```
[(user:group)@farber ~]$ qconf -shgrp @it_nss
group_name @it_nss
hostlist n000 n001 n002 n003 n004


[(user:group)@farber ~]$ qhostgrp
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR NLOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -       -
n000                    lx-amd64       20    2   20   20  0.00   63.0G    1.1G    2.0G     0.0
n001                    lx-amd64       20    2   20   20  0.00   63.0G    3.0G    2.0G     0.0
n002                    lx-amd64       20    2   20   20  0.00   63.0G    1.1G    2.0G     0.0
n003                    lx-amd64       20    2   20   20  0.00   63.0G    2.2G    2.0G     0.0
n004                    lx-amd64       20    2   20   20  0.00   63.0G    2.0G    2.0G     0.0
```


- Note the "qconf" command — short for "queue configuration"

RESOURCE INFORMATION

- Mills/Farber nodes are
  - How do I know whic

The *qhostgrp* command was created here at UD and is not part of Grid Engine itself.

```
[(user:group)@farber ~]$ qconf -shgrp @it_nss
group_name @it_nss
hostlist n000 n001 n002 n003 n004

[(user:group)@farber ~]$ qhostgrp
HOSTNAME                ARCH         NCPU NSOC NCOR NTHR NLOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------------------
global                  -               -    -    -    -     -       -       -       -       -
n000                    lx-amd64       20    2   20   20  0.00   63.0G    1.1G    2.0G     0.0
n001                    lx-amd64       20    2   20   20  0.00   63.0G    3.0G    2.0G     0.0
n002                    lx-amd64       20    2   20   20  0.00   63.0G    1.1G    2.0G     0.0
n003                    lx-amd64       20    2   20   20  0.00   63.0G    2.2G    2.0G     0.0
n004                    lx-amd64       20    2   20   20  0.00   63.0G    2.0G    2.0G     0.0
```


- Note the "qconf" command — short for "queue configuration"

QUEUES

- On many systems users are expected to explicitly choose a queue for each job submitted
- Mills/Farber use owned queues
  - Jobs submitted as workgroup ***[wg]*** are eligible ONLY for that workgroup's queue(s)
  - Workgroup's queue(s) ONLY feed specific nodes
  - Other parameters of the job automatically select the correct owner queue = no need to specify queue

QUEUES

- Mills: multiple job-specific queues
  - If own node(s) are occupied by standby jobs†, your jobs must wait

queue name	description
***[wg]*.q**	generic MPI jobs
***[wg]*.q+**	tightly-integrated MPI jobs; serial/threaded jobs
***[wg]*-qrsh.q**	interactive login to compute node

† Standby jobs will be discussed later…

QUEUES

- Farber: single owner queue
  - If own node(s) occupied by standby jobs, your jobs can draw from the spillover queue

queue name	description
[wg]`.q`	general owned queue
`spillover.q`	temporary resource fulfillment

QUEUES

```
[(user:group)@farber ~]$ qconf -sql
afwallace.q
arce.q
cadsr_cluster.q
ccei_biomass.q
ccm_gillespi.q
clouds_wind_climate.q
dditoro.q
disasters.q
ececis_research.q
geography.q
gpu-demo.q
hmichael.q
ifsa.q
it_css.q
it_nss.q
jayaraman_lab.gpu.q
jayaraman_lab.q
jayaraman_lab.spillover.q
jneun.q
kirby.q
kukulka_lab.q
phys.q
rauscher.q
resource_econ.q
safrono.q
sding stat.q
```

Dare to be first.

UNIVERSITY OF DELAWARE

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby4h=0
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby4h=0
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- Queue instances on all nodes in the @base host group
- Sorting order is
  - 10 by default
  - 50 for nodes in @unowned
  - 100 for nodes in @owned
- Queue instance is "overloaded" when node's 5-minute load average ≥ 100%

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby4h=0
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- Accepts both interactive and batch jobs
- Accepts parallel jobs that use threads (OpenMP) or mpi (Open MPI)

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby4h=0
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- Each queue instance can have at most 20 concurrently-executing jobs
- Job's $TMPDIR = a path under /tmp
- Job's are BASH scripts…
- …with the shebang ignored.

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- Jobs from any user may run in this queue…
- …as long as the user is NOT in the *deny_spillover* user list

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,standby4h=0
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- Jobs with non-zero values for the *standby* or *standby4h* complexes will not be accepted in this queue

QUEUES

```
[(user:group)@farber ~]$ qconf -sq spillover.q
qname                 spillover.q
hostlist              @base
seq_no                10,[@unowned=50],[@owned=100]
load_thresholds       np_load_avg=1.00
  :
qtype                 BATCH INTERACTIVE
pe_list               threads,mpi
  :
slots                 20
tmpdir                /tmp
shell                 /bin/bash
  :
shell_start_mode      posix_compliant
  :
user_lists            all_groups
xuser_lists           deny_spillover
  :
complex_values        standby=0,sta
  :
s_rt                  INFINITY
h_rt                  INFINITY
d_rt                  INFINITY
  :
```

- The queue places no default limit on how long a job may continuously execute

QUEUES

```
[(user:group)@farber ~]$ qstat -q spillover.q -u \* -g c
CLUSTER QUEUE                   CQLOAD   USED    RES  AVAIL  TOTAL aoACDS  cdsuE
--------------------------------------------------------------------------------
spillover.q                       0.74     86      0   1174   1740    420     80


[(user:group)@farber ~]$ qstat -q standby.q -u \* -g c
CLUSTER QUEUE                   CQLOAD   USED    RES  AVAIL  TOTAL aoACDS  cdsuE
--------------------------------------------------------------------------------
standby.q                         0.75    334      0     93   1740   1500     80
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Another Grid Engine command — "queue status"

QUEUES

```
[(user:group)@farber ~]$ qstat -q spillover.q -u \* -g c
CLUSTER QUEUE                   CQLOAD   USED    RES  AVAIL  TOTAL aoACDS  cdsuE
--------------------------------------------------------------------------------
spillover.q                       0.74     86      0   1174   1740    420     80

[(user:group)@farber ~]$ qstat -q standby.q -u \* -g c
CLUSTER QUEUE                   CQLOAD   USED    RES  AVAIL  TOTAL aoACDS  cdsuE
--------------------------------------------------------------------------------
standby.q                         0.75    334      0     93   1740   1500     0
```

- **c**onfiguration is botched
- **d**isabled by administrator
- **s**uspended
- **u**nknown (sge_execd down)
- **E**rror

Dare to be first.
UNIVERSITY OF DELAWARE

- Another Grid Engine command — "queue status"

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q spillover.q -u \*
job-ID     prior   name       user         state submit/start at     queue
jclass                         slots ja-task-ID
-----------------------------------------------------------------------------------------------------------
-----------------------------------------
     61914 0.50513 gluc2mann_ jeffc        r     04/21/2015 09:56:15 spillover.q@n057
20
     55528 0.50834 openmpi.qs zcheng       r     04/15/2015 00:23:44 spillover.q@n059
6
     61923 0.50506 DMFDehySte rpatet       r     04/21/2015 10:26:00 spillover.q@n060
10
     61921 0.50512 propylene- gjenness     r     04/21/2015 10:25:15 spillover.q@n088
20
     56873 0.50809 QLOGIN     lccannon     r     04/15/2015 15:52:53 spillover.q@n089
20
     61922 0.50506 DMFDehySte rpatet       r     04/21/2015 10:25:45 spillover.q@n090
10
     61933 0.50511 ipa-e1.qs  saikonda     qw    04/21/2015 10:52:33
20
     61926 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:02
10
     61927 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:17
10
```

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q spillover.q -u \*
job-ID     prior   name       user         state submit/start at     queue
jclass                         slots ja-task-ID
-----------------------------------------------------------------------------------------------------------
-----------------------------------------
     61914 0.50513 gluc2mann_ jeffc        r     04/21/2015 09:56:15 spillover.q@n057
20
     55528 0.50834 openmpi.qs zcheng       r     04/15/2015 00:23:44 spillover.q@n059
6
     61923 0.50506 DMFDehySte rpatet       r     04/21/2015 10:26:00 spillover.q@n060
10
     61921 0.50512 propylene- gjenness     r     04/21/2015 10:25:15 spillover.q@n088
                                                 04/15/2015 15:52:53 spillover.q@n089
                                                 04/21/2015 10:25:45 spillover.q@n090
10
     61933 0.50511 ipa-e1.qs  saikonda     qw    04/21/2015 10:52:33
20
     61926 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:02
10
     61927 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:17
10
```

- A job that requests/uses more than one slot = parallel job

Dare to be first.

UNIVERSITY OF DELAWARE

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q spillover.q -u \*
job-ID     prior   name       user         state submit/start at     queue
jclass                         slots ja-task-ID
-----------------------------------------------------------------------------------------------------------
-----------------------------------------
     61914 0.50513 gluc2mann_ jeffc        r     04/21/2015 09:56:15 spillover.q@n057
20
     55528 0.50834 openmpi.qs zcheng       r     04/15/2015 00:23:44 spillover.q@n059
6
     61923 0.50506 DMFDehySte rpatet       r     04/21/2015 10:26:00 spillover.q@n060
10
     61921 0.50512 propylene- gjenness     r     04/21/2015 10:25:15 spillover.q@n058
10
     61933 0.50511 ipa-e1.qs  saikonda     qw
20
     61926 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:02
10
     61927 0.50506 FDehyStep4 rpatet       qw    04/21/2015 10:27:17
10
```

- A job that requests/uses more than one slot = parallel job

- Primary node for job (first slot)
- Additional slots may be allocated on the same node or on other nodes

Dare to be first.
University of Delaware

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q kirby.q -u \* -g t
job-ID     prior   name       user         state submit/start at     queue
jclass                          master ja-task-ID
---------------------------------------------------------------------------------------------------
-----------------------------------------
     46489 0.52175 fyshi_b.qs fyshi        r     03/20/2015 09:27:35 kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
     46489 0.52175 fyshi_b.qs fyshi        r     03/20/2015 09:27:35 kirby.q@n108
MASTER
                                                                     kirby.q@n108
SLAVE
```

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q kirby.q -u \* -g t
job-ID     prior   name       user         state submit/start at     queue
jclass                         master ja-task-ID
---------------------------------------------------------------------------------------------------
-----------------------------------------
     46489 0.52175 fyshi_b.qs fyshi        r     03/20/2015 09:27:35 kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
                                                                     kirby.q@n106
SLAVE
     46489                                                   15 09:27:35 kirby.q@n108
MASTER
                                                                     kirby.q@n108
SLAVE
```

- Primary slot for job 46489 on n108

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -q kirby.q -u \* -g t
job-ID     prior   name       user         state submit/start at     queue
jclass                         master ja-task-ID
-----------------------------------------------------------------------------------------------------------
-----------------------------------------
     46489 0.52175 fyshi_b.qs fyshi        r     03/20/2015 09:27:35 kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
                                                                    kirby.q@n106
SLAVE
     46489                                                   15 09:27:35 kirby.q@n108
MASTER
                                                                    kirby.q@n108
SLAVE
```

- Additional slots allocated from n108 and n106
- Likely an MPI job since it spans multiple nodes

- Primary slot for job 46489 on n108

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -j 46489
==============================================================
job_number:                 46489
jclass:                     NONE
exec_file:                  job_scripts/46489
submission_time:            03/20/2015 09:27:30.089
owner:                      fyshi
uid:                        1047
group:                      kirby
gid:                        1026
sge_o_home:                 /home/1047
sge_o_log_name:             fyshi
sge_o_path:                 /home/1047/DELFT3D/delft3d_repository/src/bin:/home/1047/DELFT3D/
delft3d_repository/src/bin:/opt/bin:/opt/shared/valet/2.0.1/bin/bash:/opt/shared/valet/2.0.1/bin:/opt/
shared/univa/current/bin/lx-amd64:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/
usr/sbin:/sbin:/opt/ibutils/bin:/home/1047/bin
sge_o_shell:                /bin/bash
sge_o_workdir:              /home/1047/MCR/ideal_fine_nonhy
sge_o_host:                 login000
account:                    sge
cwd:                        /home/1047/MCR/ideal_fine_nonhy
merge:                      y
hard resource_list:         h_vmem=2G,standby=0,standby4h=0,m_mem_free=2G
mail_list:                  fyshi@login000
notify:                     FALSE
job_name:                   fyshi_b.qs
jobshare:                   0
```


- sge_o_* parameters reflect the environment in which the job was submitted
- cwd: working directory for job
  - Grid Engine configured so that by default job starts in the working directory from time of submission
  - Not the case in many other machines, and job scripts always start with e.g. cd PBS_O_WORKDIR

QUEUES AND JOBS

```
[(user:group)@farber ~]$ qstat -j 46489
==============================================================
job_number:                 46489
jclass:                     NONE
exec_file:                  job_scripts/46489
submission_time:            03/20/2015 09:27:30.089
owner:                      fyshi
uid:                        1047
group:                      kirby
gid:                        1026
sge_o_home:                 /home/1047
sge_o_log_name:             fyshi
sge_o_path:                 /home/1047/DELFT3D/delft3d_repository/src/bin:/home/1047/DELFT3D/
delft3d_repository/src/bin:/opt/bin:/opt/shared/valet/2.0.1/bin/bash:/opt/shared/valet/2.0.1/bin:/opt/
shared/univa/current/bin/lx-amd64:/usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/
usr/sbin:/sbin:/opt/ibutils/bin:/home/1047/bin
sge_o_shell:                /bin/bash
sge_o_workdir:              /home/1047/MCR/ideal_fine_nonhy
sge_o_host:                 logi
account:                    sge
cwd:                        /home
merge:                      y
hard resource_list:         h_vme
mail_list:                  fyshi
notify:                     FALSE
job_name:                   fyshi
jobshare:                   0
```

- stdout and stderr for the job script go to a single file
- file named according to the pattern: **<job_name>.o<job_number>**
- …unless provided explicitly

- sge_o_* parameters reflect the environment in which the job was submitted
- cwd: working directory for job
  - Grid Engine configured so that by default job starts in the working directory from time of submission
  - Not the case in many other machines, and job scripts always start with e.g. cd PBS_O_WORKDIR

QUEUES AND JOBS

```
env_list:
script_file:                fyshi_b.qs
parallel environment:  mpi range: 48
verify_suitable_queues:     1
department:                 defaultdepartment
binding:                    NONE
mbind:                      NONE
submit_cmd:                 /opt/shared/univa/current/bin/lx-amd64/qsub fyshi_b.qs
granted_license       1:
usage                 1:    cpu=1475:01:46:38, mem=165171444.14715 GBs, io=6.03360, vmem=72.198G,
maxvmem=74.260G, rss=59.889G, pss=59.457G, smem=528.652M, pmem=59.373G, maxrss=59.889G, maxpss=59.457G
scheduling info:            (Collecting of scheduler job information is turned off)
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Indeed, it is an MPI job
- What's a parallel environment? We'll get to that in a moment
- Running jobs show resource utilization: accumulated CPU, memory; memory "waterline"
    - Accumulated memory in units of GBs (gigabyte-second), just like power usage is kWh (kilowatt-hour)

PARALLEL ENVIRONMENT

- Serial job occupies a single *queue slot*
- Parallel job occupies *N queue slots*
  - What are the allowed limits on *N*?
  - How should the *N* slots be allocated?
  - Is there any "work" required to setup the runtime environment for the job? tear-down?

PARALLEL ENVIRONMENT

- Serial job occupies a single *queue slot*
- Parallel job occupies *N queue slots*
- Grid Engine allows the administrator to implement *parallel environments* that answer those questions.

PARALLEL ENVIRONMENT

```
[(user:group)@farber ~]$ qconf -spl
generic-mpi
mpi
threads

[(user:group)@farber ~]$ qconf -sp threads
pe_name                 threads
slots                   9999
user_lists              NONE
xuser_lists             NONE
start_proc_args         NONE
stop_proc_args          NONE
allocation_rule         $pe_slots
control_slaves          FALSE
job_is_first_task       FALSE
urgency_slots           min
accounting_summary      TRUE
daemon_forks_slaves     FALSE
master_forks_slaves     FALSE
```


- For threaded execution, a job use AT MOST the number of cores present on a single node
    - pe_slots indicates this rule
- There can be AT MOST 9999 queue slots occupied by jobs running in this parallel environment

PARALLEL ENVIRONMENT

```
[(user:group)@farber ~]$ qconf -sp mpi
pe_name               mpi
slots                 9999
user_lists            NONE
xuser_lists           NONE
start_proc_args       /bin/true
stop_proc_args        /bin/true
allocation_rule       $fill_up
control_slaves        TRUE
job_is_first_task     FALSE
urgency_slots         min
accounting_summary    FALSE
daemon_forks_slaves   TRUE
master_forks_slaves   FALSE
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Allocate slots by picking a primary slot queue instance
    - …then grab any unused slots in that queue instance
    - …move to another queue instance and repeat until all slots allocated
- Used for so-called "tightly integrated" MPI libraries (Open MPI, Intel MPI, MVAPICH)

PARALLEL ENVIRONMENT

```
[(user:group)@farber ~]$ qconf -sp generic-mpi
pe_name               generic-mpi
slots                 9999
user_lists            NONE
xuser_lists           NONE
start_proc_args       /opt/shared/univa/local/generic_mpi_start.sh $pe_hostfile
stop_proc_args        /opt/shared/univa/local/generic_mpi_stop.sh
allocation_rule       $fill_up
control_slaves        TRUE
job_is_first_task     FALSE
urgency_slots         min
accounting_summary    FALSE
daemon_forks_slaves   FALSE
master_forks_slaves   TRUE
```


- Similar to "mpi" but includes environment setup/tear-down scripts
    - The start script creates a standard MPI "machines" file from Grid Engine's allocation map
    - The stop script does nothing right now
- Used with e.g. MPICH 1

JOBS

- Your primary concern with Grid Engine is submitting and managing *jobs*.
- A *job* is essentially a Bash shell script
  - When job is submitted, Grid Engine notes the working directory and makes a copy of the script
  - At a later time, an *sge_execd* on a node goes to that working directory and executes Grid Engine's copy of the script

JOBS

- Jobs are assigned a numerical identifier when submitted
  - Use this *job number* (or *job id*) to identify the job when query Grid Engine w.r.t. it
- Examples that follow were performed on Farber; most would function the same on Mills

JOBS

```
[(user:group)@farber ~]$ qsub -N 'When did I run'

echo "I am going to execute now, and the time is"
echo
date
echo
echo "Now I go to sleep for five seconds…"
sleep 5
date
echo
echo "Now I'm done, bye-bye."
echo

^D
Your job 61983 ("When did I run") has been submitted
[(user:group)@farber ~]$
```


- A new Grid Engine command — "queue submit"
- No script mentioned, so qsub let's you type a script line-by-line

JOBS

```
[(user:group)@farber ~]$ qsub -N 'When did I run'

echo "I am going to execute now, and the time is"
echo
date
echo
echo "Now I go to sleep for five seconds…"
sleep 5
date
echo
echo "Now I'm done, bye-bye."
echo

^D
Your job 61983 ("When did I run") has been submitted
[(user:group)@farber ~]$
```

- Grid Engine assigned job number 61983 to the script I just typed.
- Note there was no shebang at the top

Dare to be first.
UNIVERSITY OF DELAWARE

- A new Grid Engine command — "queue submit"
- No script mentioned, so qsub let's you type a script line-by-line

JOBS

```
[(user:group)@farber ~]$ $ cat When\ did\ I\ run.o61983

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 1073741824 bytes (vmem 1073741824 bytes) on n004 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

I am going to execute now, and the time is

Tue Apr 21 12:17:58 EDT 2015

Now I go to sleep for five seconds…
Tue Apr 21 12:18:03 EDT 2015

Now I'm done, bye-bye.
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Output ends up in a file matching the job name and number

JOBS

```
[(user:group)@farber ~]$ $ cat When\ did\ I\ run.o61983

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 1073741824 bytes (vmem 1073741824 bytes) on n004 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

I am going t

Tue Apr 21 1

Now I go to
Tue Apr 21 1

Now I'm done, bye bye.
```

- Specific to Farber: jobs run in "containers" that have a dedicated set of processor cores and a hard limit on memory usage.
- No such limits on Mills, jobs can use more resources than they're granted
- 1073741824 bytes = 1024 x 1024 x 1024 bytes = 1 GiB

Dare to be first.
UNIVERSITY OF DELAWARE

- Output ends up in a file matching the job name and number

JOBS

```
[(user:group)@farber ~]$ qsub
#$ -N "I ran now"

echo "I am going to execute now, and the time is"
echo
date
echo
echo "Now I go to sleep for five minutes…"
sleep 300
date
echo
echo "Now I'm done, bye-bye."
echo

^D
Your job 61985 ("I ran now") has been submitted
[(user:group)@farber ~]$
```


- Instead of using -N on the command line, embed it in our script

JOBS

```
[(user:group)@farber ~]$ qsub
#$ -N "I ran now"
```

- Lines that start with "#$" can contain the same arguments that would be typed with the *qsub* command
- If provided both in script and with *qsub*, the latter takes precedence

```
echo
echo "Now I'm done, bye-bye."
echo

^D
Your job 61985 ("I ran now") has been submitted
[(user:group)@farber ~]$
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Instead of using -N on the command line, embed it in our script

JOBS

```
[(user:group)@farber ~]$ qstat -j 61985
==============================================================
job_number:                 61985
  :
exec_file:                  job_scripts/61985
  :
hard resource_list:         h_vmem=1G,standby=0,m_mem_free=1G,standby4h=0
  :
job_name:                   I ran now
  :
script_file:                STDIN
  :
usage                 1:    cpu=00:00:00, mem=0.00000 GBs, io=0.00561, vmem=16.141M, maxvmem=16.141M,
rss=2.184M, pss=908.000K, smem=1.609M, pmem=588.000K, maxrss=2.184M, maxpss=908.000K
scheduling info:            (Collecting of scheduler job information is turned off)
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Again use the qstat (queue status) command to check what a job is doing
- Resource list shows that (by default) the job requested a memory cap of 1 GiB…

JOBS

```
[(user:group)@farber ~]$ qstat -j 61985
==============================================================
job_number:                 61985
  :
exec_file:                  job_scripts/61985
  :
hard resource_list:         h_vmem=1G,standby=0,m_mem_free=1G,standby4h=0
  :
job_name:                   I ran now
  :
script_file:                STDIN
  :
usage                 1:    cpu=00:00:00, mem=0.00000 GBs, io=0.00561, vmem=16.141M, maxvmem=16.141M,
rss=2.184M, pss=908.000K, sm                                                    ss=908.000K
scheduling info:                                                                urned off)
```

- Grid Engine's copy of the script
- Original script was typed on STDIN

- Again use the qstat (queue status) command to check what a job is doing
- Resource list shows that (by default) the job requested a memory cap of 1 GiB…

JOBS

```
[(user:group)@farber ~]$ qdel 61985
user has registered the job 61985 for deletion

[(user:group)@farber ~]$ cat I\ ran\ now.o61985

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 1073741824 bytes (vmem 1073741824 bytes) on n000 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

I am going to execute now, and the time is

Tue Apr 21 12:27:58 EDT 2015

Now I go to sleep for five minutes…
[(user:group)@farber ~]$
```

Dare to be first.
UNIVERSITY OF DELAWARE

- The qdel (queue delete) command is used to remove a job
- Job got that 1 GiB memory limit

JOBS

```
[(user:group)@farber ~]$ qdel 61985
user has registered the job 61985 for deletion

[(user:group)@farber ~]$ cat I\ ran\ now.o61985

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 1073741824 bytes (vmem 1073741824 bytes) on n000 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

I am going to execute now, and the time is

Tue Apr 21 12:27:58 EDT 2015

Now I go to sleep for five minutes…
[(user:group)@farber ~]$
```

- Job script exited during the five-minute sleep command (sleep 300)
- Remainder of script was NOT executed

Dare to be first.
UNIVERSITY OF DELAWARE

- The qdel (queue delete) command is used to remove a job
- Job got that 1 GiB memory limit

JOBS

```
[(user:group)@farber ~]$ qsub -S /bin/csh
#$ -l m_mem_free=4G
#$ -S /bin/bash
#$ -N "Can I use csh"

setenv MY_VAR "This is a test."
echo $MY_VAR

^D
Your job 61988 ("Can I use csh") has been submitted
[(user:group)@farber ~]$
```


- The "-S" option indicated which shell should be used to execute the script

JOBS

```
[(user:group)@farber ~]$ cat Can\ I\ use\ csh.o61987

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 4294967296 bytes (vmem 4294967296 bytes) on n004 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

Warning: no access to tty (Bad file descriptor).
Thus no job control in this shell.
This is a test.

[(user:group)@farber ~]$
```


- Yes, I can use C shell
- To run a Perl script in the queue, pass the path to the Perl interpreter (/usr/bin/perl)
- And obviously an option provided on the command line overrides the option provided in the script.

JOBS

```
[(user:group)@farber ~]$ cat Can\ I\ use\ csh.o61987

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 4294967296 bytes (vmem 4294967296 bytes) on n004 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

Warning: no access to
Thus no job control in this shell.
This is a test.

[(user:group)@farber ~]$
```

- I asked for 4 GiB memory limit, and the job got it.

Dare to be first.
UNIVERSITY OF DELAWARE

- Yes, I can use C shell
- To run a Perl script in the queue, pass the path to the Perl interpreter (/usr/bin/perl)
- And obviously an option provided on the command line overrides the option provided in the script.

JOBS

```
[(user:group)@farber ~]$ qsub
#$ -l m_mem_free=4G
#$ -l scratch_free=1000G
#$ -N Lotsa_scratch_disk

echo "I ran."

^D
warning: no suitable queues
warning: no suitable queues
Your job 61989 ("Lotsa_scratch_disk") has been submitted
[(user:group)@farber ~]$
```


- I want a 4 GiB memory limit and a terabyte of local scratch disk space
- All nodes in Farber have 500 GB hard disks, so no node satisfies this request
- BUT, if such a node were to be added in the future, it could run this job…thus, Grid Engine does not outright reject it

JOBS

```
[(user:group)@farber ~]$ qstat -j 61989 | grep resource_list
hard resource_list:         h_vmem=4G,m_mem_free=4G,scratch_free=1000G,standby=0,standby4h=0


[(user:group)@farber ~]$ qalter -l h_vmem=4G,m_mem_free=4G,scratch_free=400G,standby=0,standby4h=0 \
> 61989
modified hard resource list of job 61989



   :
 several minutes go by…
   :



[(user:group)@farber ~]$ cat Lotsa_scratch_disk.o61989

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 4294967296 bytes (vmem 4294967296 bytes) on n004 (master)
[CGROUPS]   with 1 core = 0
[CGROUPS] done.

I ran.
[(user:group)@farber ~]$
```

Dare to be first.

UNIVERSITY OF DELAWARE

- The qalter (queue alter) command can be used to change a job's parameters after submission
    - The entire resource list must be passed to qalter — it does not add/remove piecemeal
- 400 GB of scratch disk CAN be satisfied by a Farber node

JOBS

```
[(user:group)@farber ~]$ qsub
#$ -l m_mem_free=4G
#$ -pe threads 4
#$ -N four_threads

export OMP_NUM_THREADS="$NSLOTS"

echo "My OpenMP code would run with ${OMP_NUM_THREADS} threads."
echo

^D
Your job 61990 ("four_threads") has been submitted
[(user:group)@farber ~]$
```

Dare to be first.
UNIVERSITY OF DELAWARE

- This job will use the "threads" parallel environment, grabbing 4 slots from a single node.

JOBS

```
[(user:group)@farber ~]$ cat four_threads.o61990

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 17179869184 bytes (vmem 17179869184 bytes) on n004 (master)
[CGROUPS]   with 4 cores = 0-3
[CGROUPS] done.

My OpenMP code would run with 4 threads.


[(user:group)@farber ~]$
```


- Turns out that's actually 16 GiB — the number of slots multiples the requested memory.

JOBS

```
[(user:group)@farber ~]$ cat four_threads.o61990

[CGROUPS] UD Grid Engine cgroup setup commencing
[CGROUPS] Setting 17179869184 bytes (vmem 17179869184 bytes) on n004 (master)
[CGROUPS]   with 4 cores = 0-3
[CGROUPS] done.

My OpenMP code would

[(user:group)@farber ~]$
```

- I asked for 4 GiB memory limit, and that's not 4294967296 bytes!

Dare to be first.
UNIVERSITY OF DELAWARE

- Turns out that's actually 16 GiB — the number of slots multiples the requested memory.

JOBS

- IT has created queue script templates that users can copy and modify to their workflows
  - See **`/opt/templates/gridengine`** on Mills/Farber
  - Files are heavily-documented to guide you through the necessary modifications

SUMMARY

- Complex computing environments demand automated workload management
  - A resource manager keeps track of what's available and what it's doing
  - A job scheduler keeps track of users' work and dispatches it, making decisions based on
    - Policies
    - Resource manager's information

SUMMARY

- Complex computing environments demand automated workload management
- UD uses Grid Engine on the community clusters
  - Product with a long history
  - Both a resource manager and job scheduler
  - Template scripts, IT support are available

ADDITIONAL INFO

RESOURCE QUOTAS

```
[(user:group)@farber ~]$ qstat -f -q '*@n085'
queuename                      qtype resv/used/tot. np_load  arch          states
---------------------------------------------------------------------------------
spillover.q@n085               BIP   0/0/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby.q@n085                 BP    0/4/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby-4h.q@n085              BP    0/16/20        0.00     lx-amd64
```

Dare to be first.
UNIVERSITY OF DELAWARE

- If the three queues on n085 each present 20 slots (for 60 total slots), but the node has just 20 cores, won't the node end up running too much stuff?
- YES, except that

RESOURCE QUOTAS

```
[(user:group)@farber ~]$ qstat -f -q '*@n085'
queuename                      qtype resv/used/tot. np_load  arch          states
---------------------------------------------------------------------------------
spillover.q@n085               BIP   0/0/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby.q@n085                 BP    0/4/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby-4h.q@n085              BP    0/16/20        0.00     lx-amd64
```

- Doesn't this mean n085 has 60 queue slots available for scheduling jobs?
- If the node only has 20 cores, doesn't that mean it's possible too much stuff will run on the node?

Dare to be first.
UNIVERSITY OF DELAWARE

- If the three queues on n085 each present 20 slots (for 60 total slots), but the node has just 20 cores, won't the node end up running too much stuff?
- YES, except that

RESOURCE QUOTAS

```
[(user:group)@farber ~]$ qstat -f -q '*@n085'
queuename                      qtype resv/used/tot. np_load  arch          states
---------------------------------------------------------------------------------
spillover.q@n085               BIP   0/0/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby.q@n085                 BP    0/4/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby-4h.q@n085              BP    0/16/20        0.00     lx-amd64

[(user:group)@farber ~]$ qquota -h n085 | grep slots_per_node
slots_per_node/default slots=20/20          hosts n085

[(user:group)@farber ~]$ qconf -srqs slots_per_node
{
   name         slots_per_node
   description  Per host, no more than $num_proc slots should be in-use across \
   all queues
   enabled      TRUE
   limit        name default hosts {*} to slots=$num_proc
}
```

Dare to be first.
UNIVERSITY OF DELAWARE

- We set an explicit maximum using a resource quota
- On n085, the 4 and 16 slots sum to 20, so the host is "full"

RESOURCE QUOTAS

```
[(user:group)@farber ~]$ qstat -f -q '*@n085'
queuename                      qtype resv/used/tot. np_load  arch          states
---------------------------------------------------------------------------------
spillover.q@n085               BIP   0/0/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby.q@n085                 BP    0/4/20         0.00     lx-amd64
---------------------------------------------------------------------------------
standby-4h.q@n085              BP    0/16/20        0.00     lx-amd64

[(user:group)@farber ~]$ qquota -h n085 | grep slots_per_node
slots_per_node/default slots=20/20          hosts n085

[(user:group)@farber ~]$ qconf -srqs slots_per_node
{
   name         slots_per_node
   description  Per host, no more than $num_proc slots should be in-use across \
   all queues
   enabled      TRUE
   limit        name default hosts {*} to slots=$num_proc
}
```

- …aggregated across all queues…
- …on each individual host…

- the number of occupied slots cannot exceed the core count

Dare to be first.
University of Delaware

- We set an explicit maximum using a resource quota
- On n085, the 4 and 16 slots sum to 20, so the host is "full"

RESOURCE QUOTAS

```
[(user:group)@farber ~]$ qconf -srqs per_workgroup | less
{
   name         per_workgroup
   description  Limit the number of non-standby slots each workgroup can use \
   concurrently.
   enabled      TRUE
   limit        name it_nss users @it_nss queues it_nss.q,spillover.q to \
   slots=100
   limit        name it_css users @it_css queues it_css.q,spillover.q to \
   slots=100
     :
```

Dare to be first.
UNIVERSITY OF DELAWARE

- Similar rules used to limit each workgroup's use of the spillover queue

RESOURCE QUOTAS

```
[(user:group)@farb
{
   name         pe
   description  Li                                  orkgroup can use \
   concurrently.
   enabled      TRUE
   limit        name it_nss users @it_nss queues it_nss.q,spillover.q to \
   slots=100
   limit        name it_css users @it_css queues it
```

- …for users in the it_nss workgroup…
- …are limited to a total of 100 slots (5 nodes)
- …jobs running in the owner queue OR the spillover queue…

Dare to be first.
UNIVERSITY OF DELAWARE

- Similar rules used to limit each workgroup's use of the spillover queue

STANDBY JOBS

- An attempt to make use of idle cpu cycles on the clusters
  - Jobs can run on any node, not just workgroup's nodes
- Time-limited use of high core counts
  - Benchmark software/job scaling

STANDBY JOBS

- Standard queue
  - Add "`-l standby=1`" to your qsub parameters
  - Maximum run time = 8 hours
    - When 8 hours is exceeded, job is killed
    - Jobs (software) with checkpointing features can be rerun

STANDBY JOBS

- Standard queue
  - Add "`-l standby=1`" to your qsub parameters
  - Maximum run time = 8 hours

Cluster	Maximum Cores, 8-hour
Mills	240
Farber	200

STANDBY JOBS

- Standard queue
  - Add "`-l standby=1`" to your qsub parameters
  - Maximum run time = 8 hours

Cluster	Maximum Cores, 8-hour
Mills	240
Farber	200

per user, across all the user's running standby jobs

STANDBY JOBS

- 4-hour queue
  - Add "`-l standby=1`" to your qsub parameters
  - Indicate a maximum runtime < 4 hours
    - E.g. add "`-l h_rt=3:00:00`" to your qsub parameters

Cluster	Maximum Cores, 4-hour
Mills	816
Farber	800

STANDBY JOBS

- Aggregate per-user limit to concurrent usage
  - Applies to a user's jobs running across BOTH standby queues
  - E.g. on Farber, user may run (4) 200-core jobs in the standard standby queue
  - …or, (1) 800-core job in the 4-hour queue

STANDBY JOBS

- Aggregate per-user limit to concurrent usage

Cluster	Maximum Concurrent Cores
Mills	816
Farber	800