Queuing Systems

Rocks-A-Palooza I

Track 1

Session III

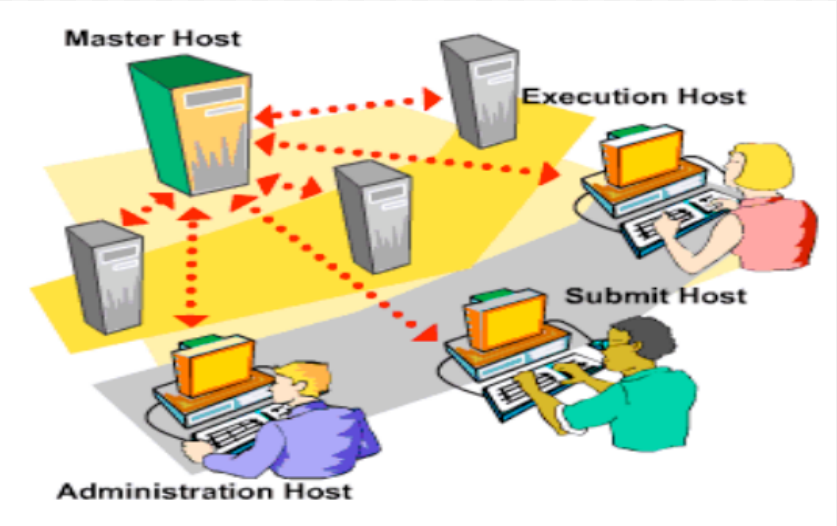
SDSC

SGE

Overview of Sun Grid Engine

Introduction to Sun Grid Engine

- What is a grid?
  - A collection of computing resources that perform tasks
    - A grid node can be a compute server, data collector, visualisation terminal..
- SGE is a resource management software
  - Accepts jobs submitted by users
  - Schedules them for execution on appropriate systems based on resource management policies
  - Can submit 100s of jobs without worrying where it will run



What is SGE?

- SUN marketing terminology
  - Cluster grids
    - Standard SGE.. found in Rocks
  - Campus grids
    - SGE Enterprise Edition
  - Global grids
    - SGE Enterprise Edition
- SGE Licence?
  - Standard Edition – free
  - Enterprise Edition – free if from opensource gridengine site, you can pay for “rigorously tested” SGEE package from SUN though..

Too many ‘E’s in SGE

- SGE Standard Edition
  - Linux cluster
- SGE Enterprise Edition
  - when you want to aggregate a few clusters together
    - And manage them as one resource
  - When you want sophisticated policy management
    - User/Project share
    - Deadlines
    - User, Department, Project level
- Rocks comes standard with SGE Enterprise

Job Management

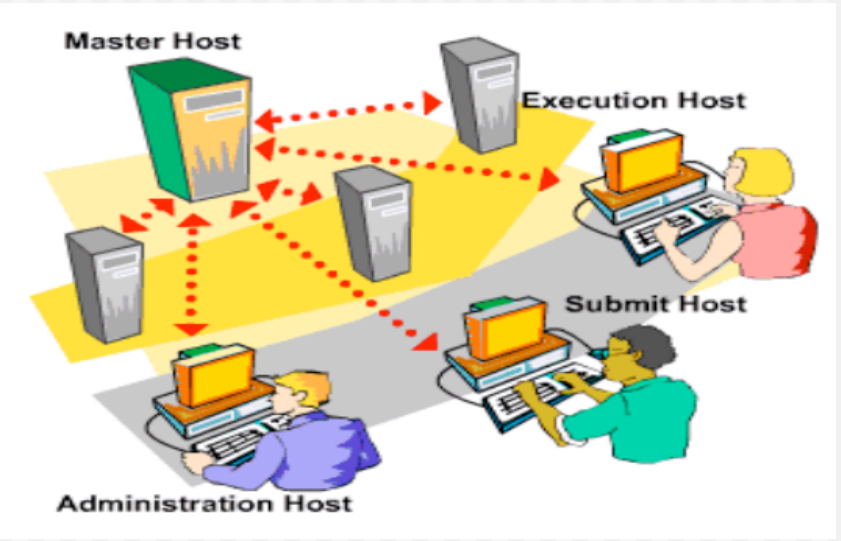
- Not recommended to run jobs directly!
- Use installed load scheduler
  - SUN Grid Engine
    - Load management tool for HETEROGENEOUS distributed computing environment
  - PBS/Torque
    - Not covered in this material
    - More sophisticated scheduling
- Why?
  - You can submit multiple jobs and have it queued (and go home!)
  - Fair Share
    - Allow other people to use the cluster also! (for Myrinet MPI jobs)

Host Roles

- Master Host
  - Controls overall cluster activity
  - Frontend, head node
  - It runs the master daemon: sge_qmaster, controlling
    - queues, jobs, status, user access permission
  - Also the scheduler: sge_schedd
- Execution Host
  - executes SGE jobs
  - execution daemon: sge_execd
    - Runs jobs on its hosts
    - Forwards sys status/info to sge_qmaster

Host Roles
continued

- Submit Host
  - They are allowed for submitting and controlling batch job only
  - No daemon required to run in this type of host.
- Administration Host
  - SGE administrator console usually



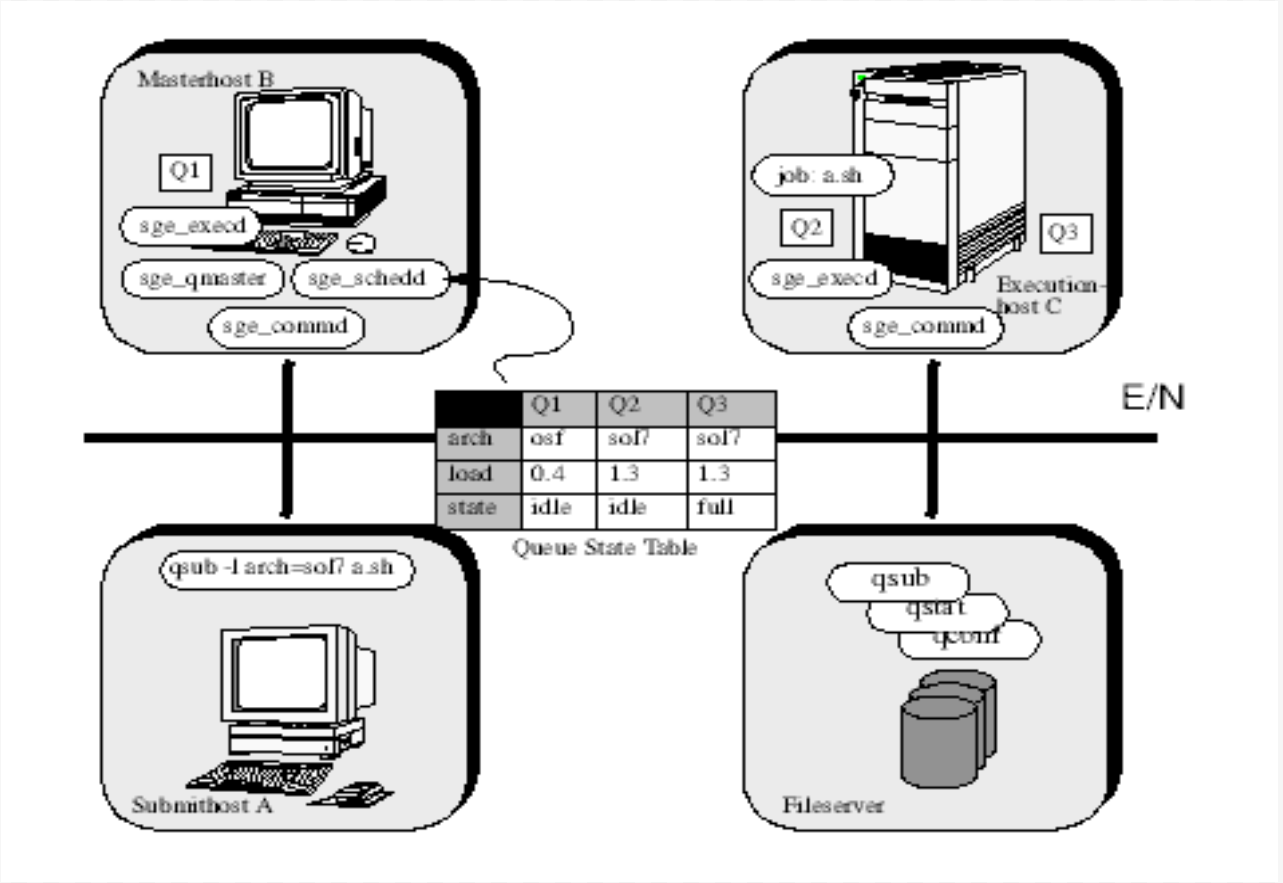
What is a Queue?

- A container for a class of jobs allowed to execute on a host concurrently
- A queue determines jobs types
  - Cpu (itanium.q, xeon.q)
  - Mem (himem.q)
  - Time (short.q, long.q)
  - Licences (Fluent.q)
- No need to submit job to a particular queue!
  - Only need to specify your job requirements
    - OS, software, mem
  - SGE will dispatch to suitable queue on a low-loaded host

Job Management

- Your administrator (or Rocks) would have setup default queues (compute-0-0.q) for the cluster
- More fine-tunned queues can be setup depending on cluster/user community
  - short.q, long.q, weekend.q, fluent.0.q, fluent.1.q
- As a user, you only need to know how to
  - Submit your jobs (serial or MPI)
  - Monitor your jobs
  - Get the results

Another View of SGE



Some SGE Commands

Command	Description
qacct	Extract accounting information from cluster
qalter	Changes the attributes of submitted but pending jobs
qconf	SGE's cluster, queue etc configuration
qdel	Job deletion
qhold	Holds back submitted jobs for execution
qhost	Shows status information about SGE hosts
qmod	Modify queue statues: enabled or suspended
qmon	X-windows Motif interface
qrsh	SGE queue based rsh facility
qselect	List queue matching selection criteria
qsh	Opens an interactive shell on a low-loaded hosts
qstat	Status listing of jobs and queues
qsub	Commandline interface to submit jobs to SGE
qtcsh	SGE queue based TCSH facility

qtcsh, qsh - extended command shells that can transparently distribute execution of programs/applications to least loaded hosts via SGE.

Command Access Control

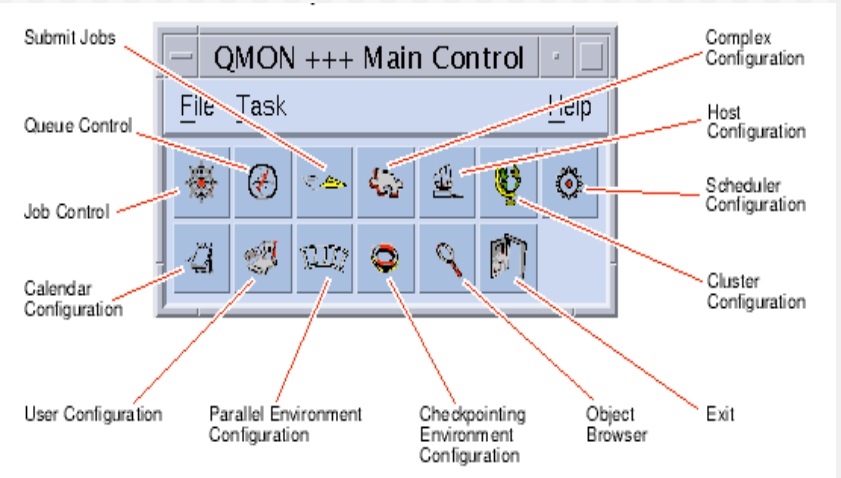
Command	Manager	Operator	Owner	User
qacct	Full	Full	Own jobs only	One jobs only
qalter	Full	Full	Own jobs only	Own jobs only
qconf	Full	Non-sys only	Show	Show
qdel	Ful	Full	Own jobs only	Own jobs only
qhold	Full	Full	Own jobs only	Own jobs only
qhost	Full	Full	Full	Full
qmod	Full	Full	Own jobs/queues	Own jobs only
qmon	Full	Non-sys only	No conf changes	No conf changes
qrsh	Full	Full	Full	Full
qstat	Full	Full	Full	Full
qsh	Full	Full	Full	Full
qstat	Full	Full	Full	Full
qsub	Full	Full	Full	Full
qtcsh	Full	Full	Full	Full

$ qhost

```
[scsadmin@hydra3 ]$ qhost
HOSTNAME                ARCH         NPROC  LOAD   MEMTOT   MEMUSE   SWAPTO   SWAPUS
-------------------------------------------------------------------------------------
global                  -                -     -       -        -        -        -
compute-0-1             glinux           4  6.00     3.9G   585.3M  1000.0M      0.0
compute-0-10            glinux           4  4.01     3.9G   647.0M  1000.0M      0.0
compute-0-11            glinux           4  4.00     3.9G   472.0M  1000.0M      0.0
compute-0-12            glinux           4  4.00     3.9G   529.2M  1000.0M      0.0
compute-0-13            glinux           4  4.00     3.9G   535.5M  1000.0M      0.0
compute-0-14            glinux           4  4.00     3.9G   530.0M  1000.0M      0.0
compute-0-2             glinux           4  5.00     3.9G   560.9M  1000.0M      0.0
compute-0-3             glinux           4  4.00     3.9G   534.3M  1000.0M      0.0
compute-0-4             glinux           4  5.00     3.9G   555.0M  1000.0M      0.0
compute-0-5             glinux           4  5.00     3.9G   559.8M  1000.0M      0.0
compute-0-6             glinux           4  5.00     3.9G   561.5M  1000.0M      0.0
compute-0-7             glinux           4  5.01     3.9G   551.6M  1000.0M      0.0
compute-0-8             glinux           4  5.00     3.9G   557.9M  1000.0M      0.0
compute-0-9             glinux           4  4.02     3.9G   541.8M  1000.0M      0.0
hydra3
```

QMON

- User friendly X-applications
- Requires you to run either Linux/Unix on your desktop or have a X-emulator (Hummingbird) on your Windows PC.
- Make sure you source the SGE settings, usually found in:
  `<SGE_ROOT>/common/settings.sh`




```
[scsadmin@hydra3 examples]$ which qmon
/home/sge/bin/glinux/qmon

[scsadmin@hydra3 examples]$ qmon
```

Submitting Jobs

- Command line (qsub) & Graphical (qmon)
  - Standard, Batch, Array, Interactive, Parallel
- SGE schedule jobs based on
  - Job priorities
    - User -> fifo
    - Admin -> can affect with priority settings
  - Equal-Share-Scheduling
    - Scheduler -> user_sort setting
    - Prevents a single user from hogging the queues
    - Recommended!!!

$ qsub

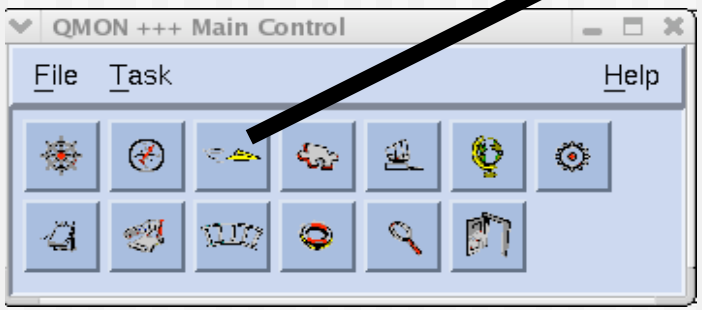
- Output/error by default in home directory

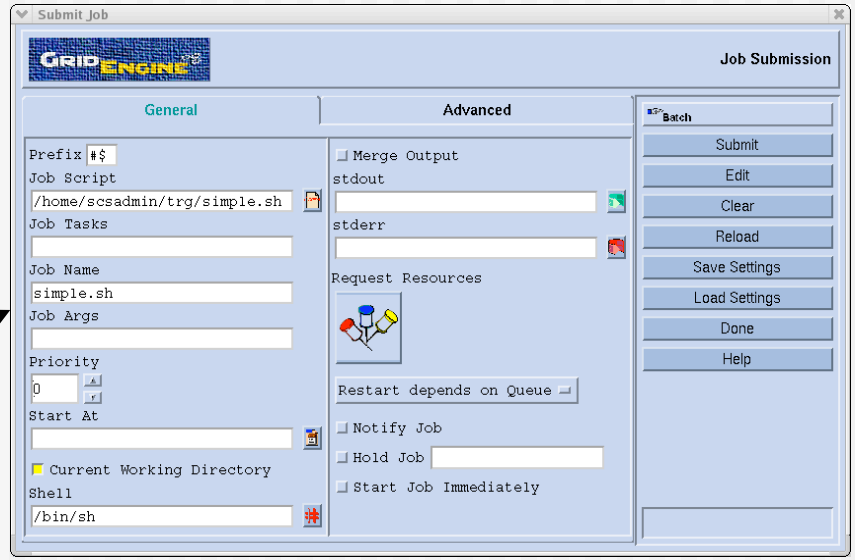
```
[scsadmin@hydra3 trg]$ cp /home/sge/examples/jobs/simple.sh .

[scsadmin@hydra3 scsadmin trg]$ more simple.sh
#!/bin/sh
date
sleep 20
date
[scsadmin@hydra3 trg]$ qsub simple.sh
your job 224 ("simple.sh") has been submitted

WAIT
[scsadmin@hydra3 trg] cd ~
-rw-r--r--    1 scsadmin scsadmin        0 May 12 18:50 simple.sh.e224
-rw-r--r--    1 scsadmin scsadmin       58 May 12 18:51 simple.sh.o224
[scsadmin@hydra3 scsadmin]$ more simple.sh.e224
[scsadmin@hydra3 scsadmin]$ more simple.sh.o224
Mon May 12 18:50:51 SGT 2003
Mon May 12 18:51:11 SGT 2003
[scsadmin@hydra3 scsadmin]$
```

- Use qstat to check job status

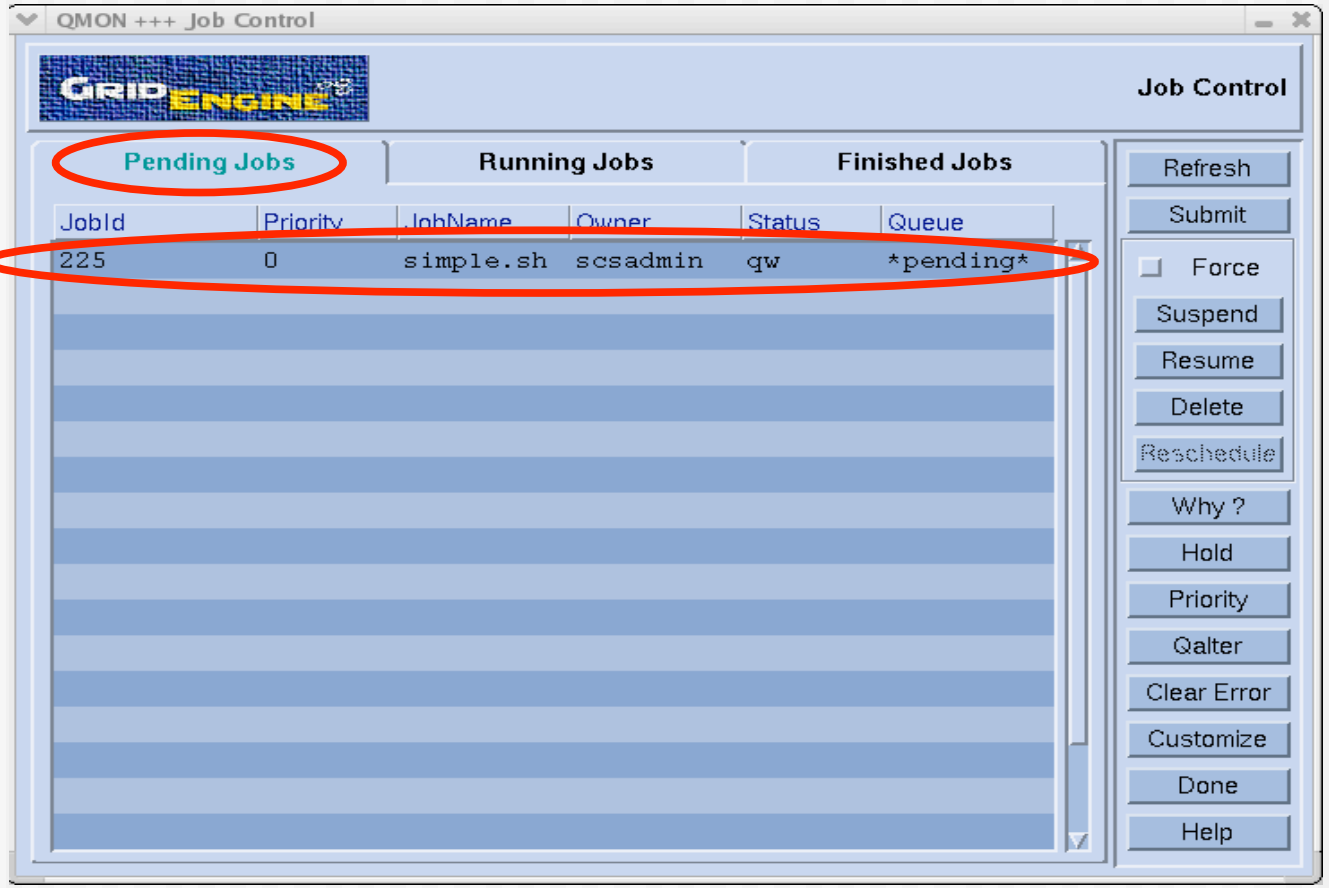
GUI

- Submit
- Monitor
- Control

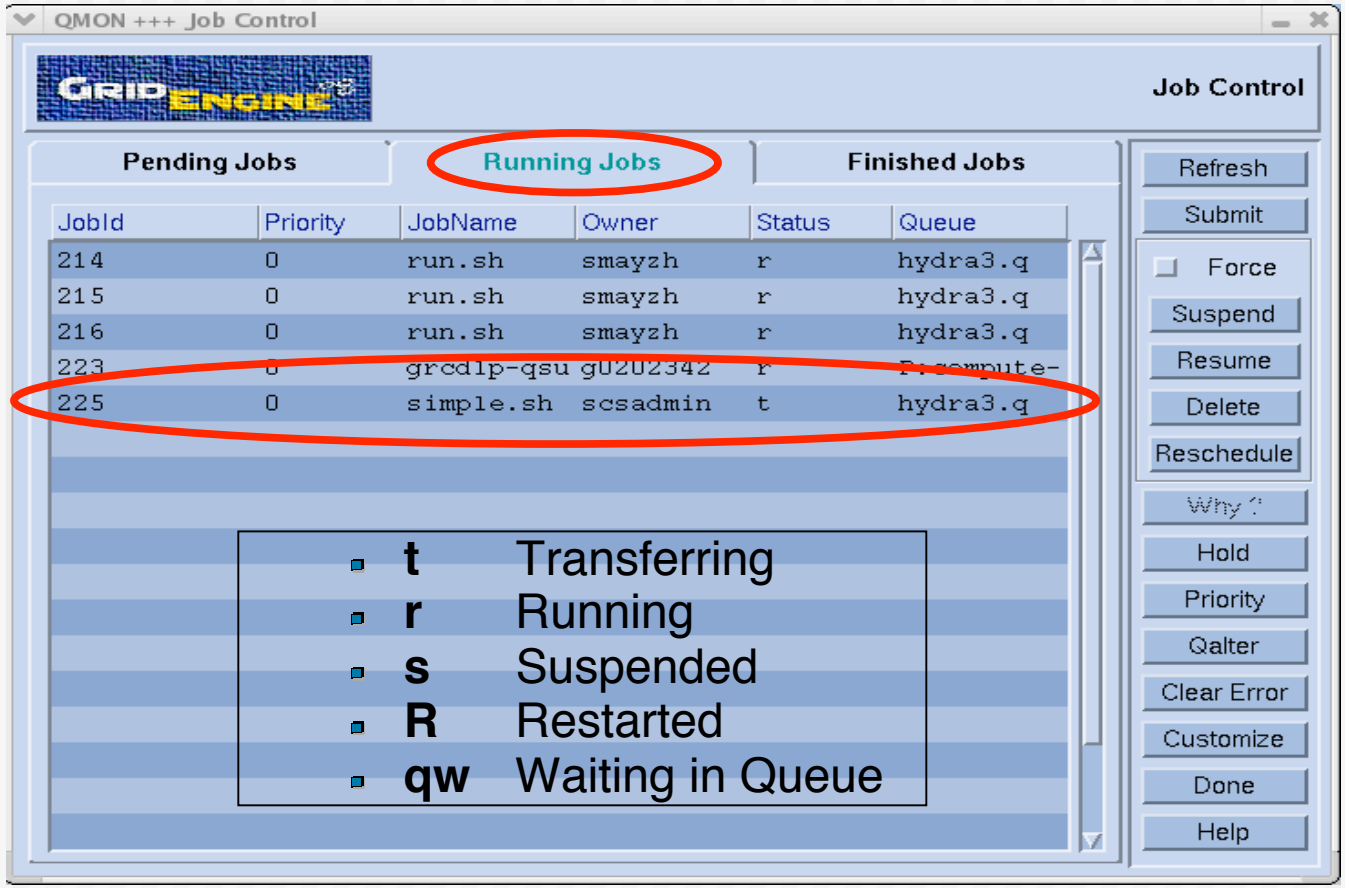




Monitoring



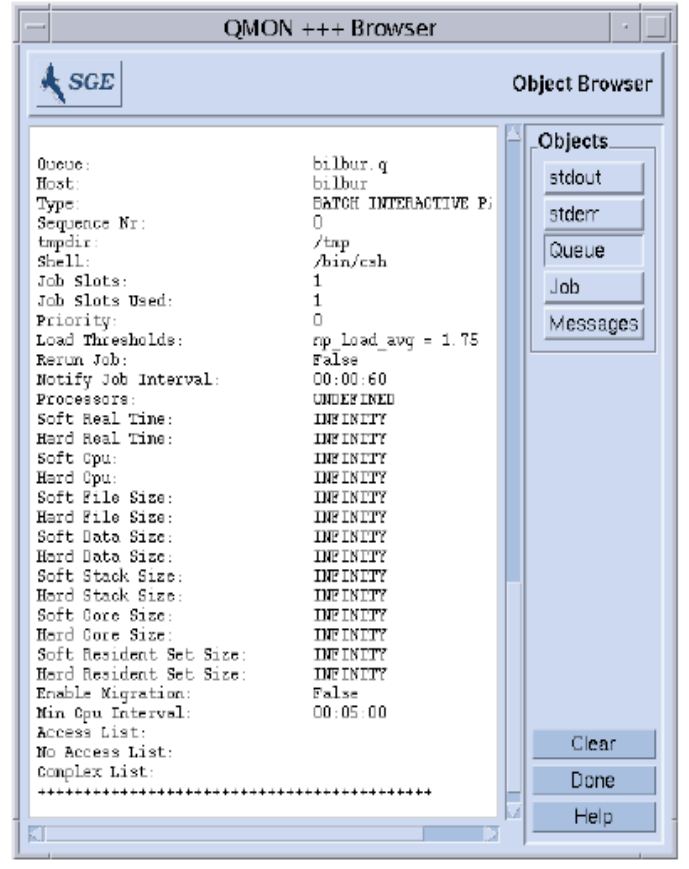
Monitoring
continued



Monitoring
continued

$ qconf

```
[scsadmin@hydra3 scsadmin]$ qconf hydra3.q
qname                 hydra3.q
hostname              hydra3
seq_no                0
load_thresholds       np_load_avg=1.75
suspend_thresholds    NONE
nsuspend              1
suspend_interval      00:05:00
priority              0
min_cpu_interval      00:05:00
processors            UNDEFINED
qtype                 BATCH INTERACTIVE PARALLEL
rerun                 FALSE
slots                 4
<snipped>
user_lists            NONE
xuser_lists           NONE
subordinate_list      NONE
<snipped>
s_rss                 INFINITY
h_rss                 INFINITY
s_vmem                INFINITY
h_vmem                INFINITY
```



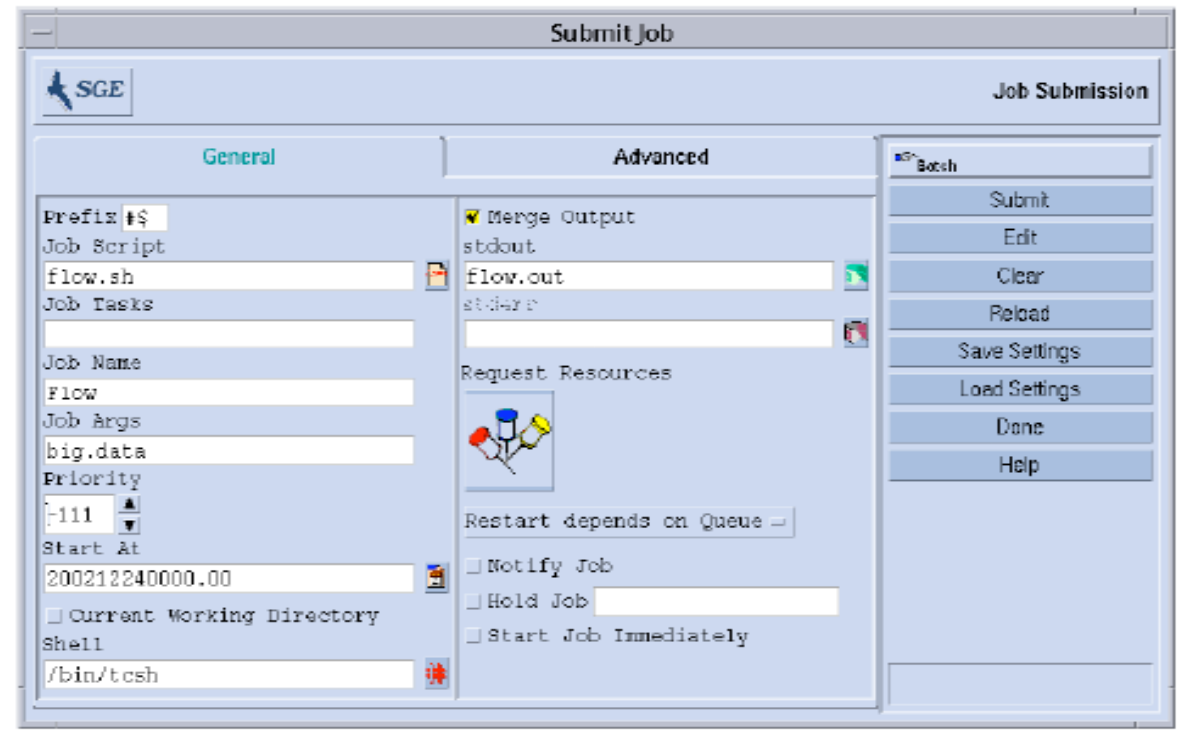
Advanced Submit

- Advanced or Batch jobs == shell scripts
- Can be as complicated as you want or even an application!

```
#!/bin/bash
#
# compiles my program everytime and create the executable and run it!
#

# change to my working directory
cd TEST

# compile the job
f77 flow.f -o flow -lm -latlas

# run the job
./flow myinput.dat
```

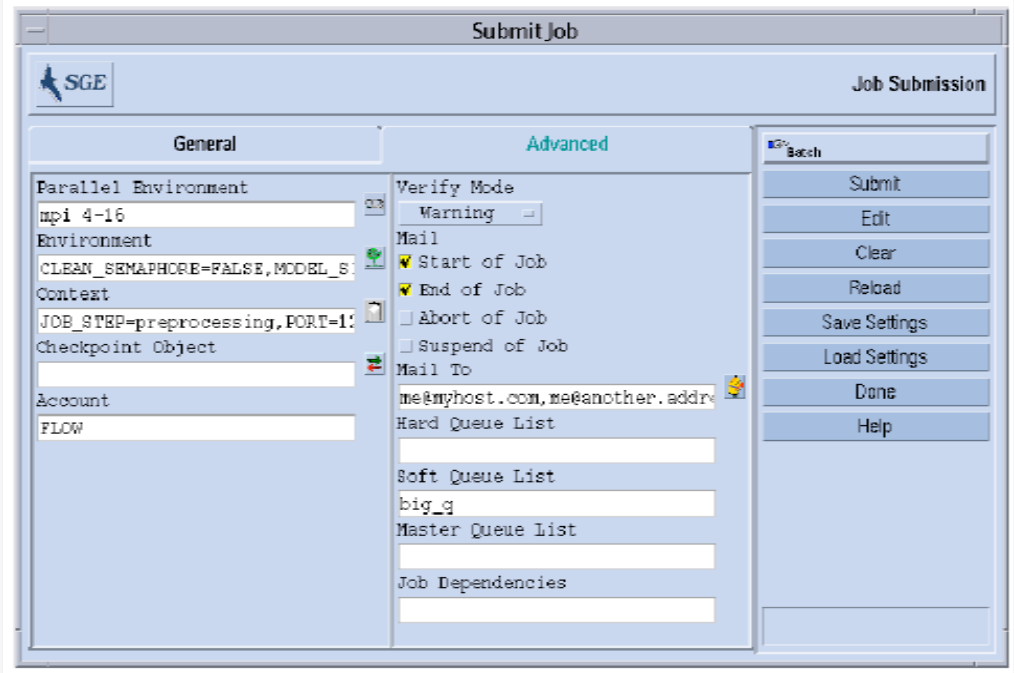
CLI vs. GUI

```
[scsadmin@hydra3 scsadmin]$ qsub -N Flow -p -111 -a 20001224000.00 -cwd -S
    /bin/tcsh -o flow.out -j y flow.sh big.data
```



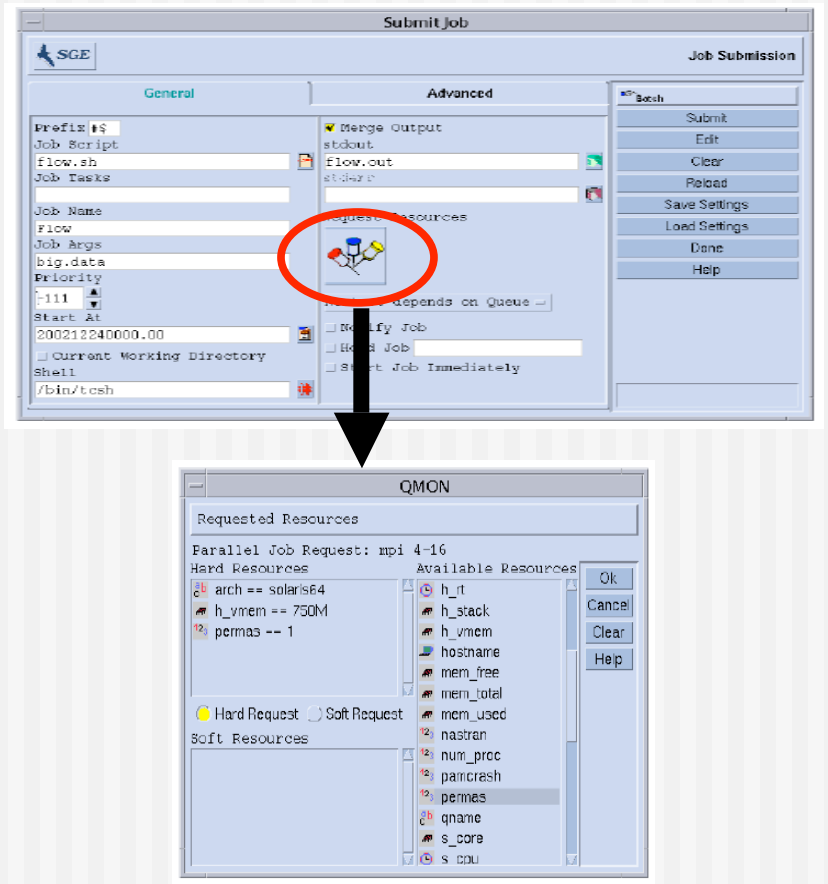
CLI vs. GUI

```
[scsadmin@hydra3 scsadmin]$ qsub -N Flow -p -111 -a 20001224000.00 -cwd -S
    /bin/tcsh -o flow.out -j y -pe mpi 4-16 -v SHARED_MEM=TRUE,
    MODEL_SIZE=LARGE -ac JOB_STEP=preprocessing, PORT=1234 -A FLOW -w w -r y -m
    s,e -q big.q -M me@myhostname.com, me@other.address flow.sh big.data
```



Requestable Attributes

- User submit jobs by specifying a job requirement profile of the hosts or of the queues
- SGE will match the job requirements and run on suitable hosts
- Attributes
  - Disk space
  - CPU
  - Memory
  - Software (Fluent lic)
  - OS



Attributes
continued

- Relop
  - Relational operation used to compute whether a queue meets a user request
- Requestable
  - Can be specified by user or not (eg in qsub)
- Consumable
  - Manage limited resources, eg licence or cpu

```
EXAMPLE EXPLANATION
#name               shortcut   type    value           relop requestable consumable default
arch                a          STRING  none            ==    YES         NO         none
num_proc            p          INT     1               ==    YES         NO         0
load_avg            la         DOUBLE  99.99           >=    NO          NO         0
slots               s          INT     0               <=    YES         YES        1

scsadmin@hydra3 scsadmin]$ qsub -l arch=glinux load_avg=0.01 myjob.sh
```

Attributes

continued

```
[scsadmin@hydra3 scsadmin]$ qconf -scl
host
queue

[scsadmin@hydra3 scsadmin]$ qconf -sc queue
#name               shortcut   type   value           relop requestable consumable default
#------------------------------------------------------------------------------------------
qname               q          STRING NONE            ==    YES         NO         NONE
hostname            h          HOST   unknown         ==    YES         NO         NONE
tmpdir              tmp        STRING NONE            ==    NO          NO         NONE
calendar            c          STRING NONE            ==    YES         NO         NONE
seq_no              seq        INT    0               ==    NO          NO         0
rerun               re         INT    0               ==    NO          NO         0
s_rt                s_rt       TIME   0:0:0           <=    YES         NO         0:0:0
h_rt                h_rt       TIME   0:0:0           <=    YES         NO         0:0:0
s_cpu               s_cpu      TIME   0:0:0           <=    YES         NO         0:0:0
h_cpu               h_cpu      TIME   0:0:0           <=    YES         NO         0:0:0
s_data              s_data     MEMORY 0               <=    YES         NO         0
h_data              h_data     MEMORY 0               <=    YES         NO         0
s_stack             s_stack    MEMORY 0               <=    YES         NO         0
h_stack             h_stack    MEMORY 0               <=    YES         NO         0
s_core              s_core     MEMORY 0               <=    YES         NO         0
h_core              h_core     MEMORY 0               <=    YES         NO         0
s_rss               s_rss      MEMORY 0               <=    YES         NO         0
h_rss               h_rss      MEMORY 0               <=    YES         NO         0
min_cpu_interval    mci        TIME   0:0:0           <=    NO          NO         0:0:0
```

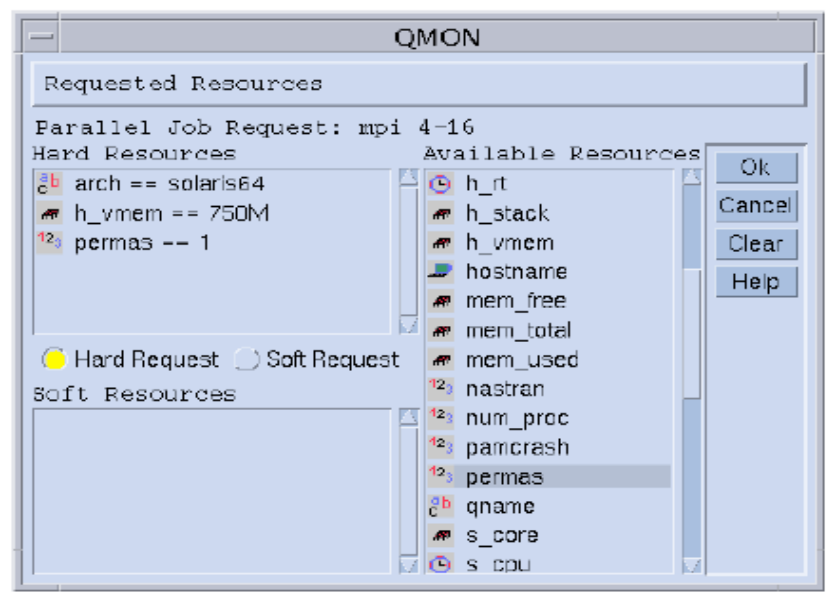
Attributes

continued

```
[scsadmin@hydra3 scsadmin]$ qconf -sc host
#name               shortcut   type   value           relop requestable consumable default
arch                a          STRING none            ==    YES         NO         none
num_proc            p          INT    1               ==    YES         NO         0
load_avg            la         DOUBLE 99.99           >=    NO          NO         0
load_short          ls         DOUBLE 99.99           >=    NO          NO         0
load_medium         lm         DOUBLE 99.99           >=    NO          NO         0
load_long           ll         DOUBLE 99.99           >=    NO          NO         0
np_load_avg         nla        DOUBLE 99.99           >=    NO          NO         0
np_load_short       nls        DOUBLE 99.99           >=    NO          NO         0
np_load_medium      nlm        DOUBLE 99.99           >=    NO          NO         0
np_load_long        nll        DOUBLE 99.99           >=    NO          NO         0
mem_free            mf         MEMORY 0               <=    YES         NO         0
mem_total           mt         MEMORY 0               <=    YES         NO         0
swap_free           sf         MEMORY 0               <=    YES         NO         0
swap_total          st         MEMORY 0               <=    YES         NO         0
virtual_free        vf         MEMORY 0               <=    YES         NO         0
virtual_total       vt         MEMORY 0               <=    YES         NO         0
mem_used            mu         MEMORY INFINITY        >=    YES         NO         0
swap_used           su         MEMORY INFINITY        >=    YES         NO         0
virtual_used        vu         MEMORY INFINITY        >=    YES         NO         0
swap_rsvd           srsv       MEMORY 0               >=    YES         NO         0
swap_rate           sr         MEMORY 0               >=    YES         NO         0
slots               s          INT    0               <=    YES         YES        1
s_vmem              s_vmem     MEMORY 0               <=    YES         NO         0
h_vmem              h_vmem     MEMORY 0               <=    YES         NO         0
s_fsize             s_fsize    MEMORY 0               <=    YES         NO         0
h_fsize             h_fsize    MEMORY 0               <=    YES         NO         0
cpu                 cpu        DOUBLE 0               >=    YES         NO         0
```

Attributes
continued

- Hard requests are checked first, followed by soft requests
- Attributes are parsed left to right, top to bottom




```
Scsadmin@hydra3 scsadmin]$ qsub -l arch=solaris64, h_vmem=750M, permas=1 permas.sh
```

Array Jobs

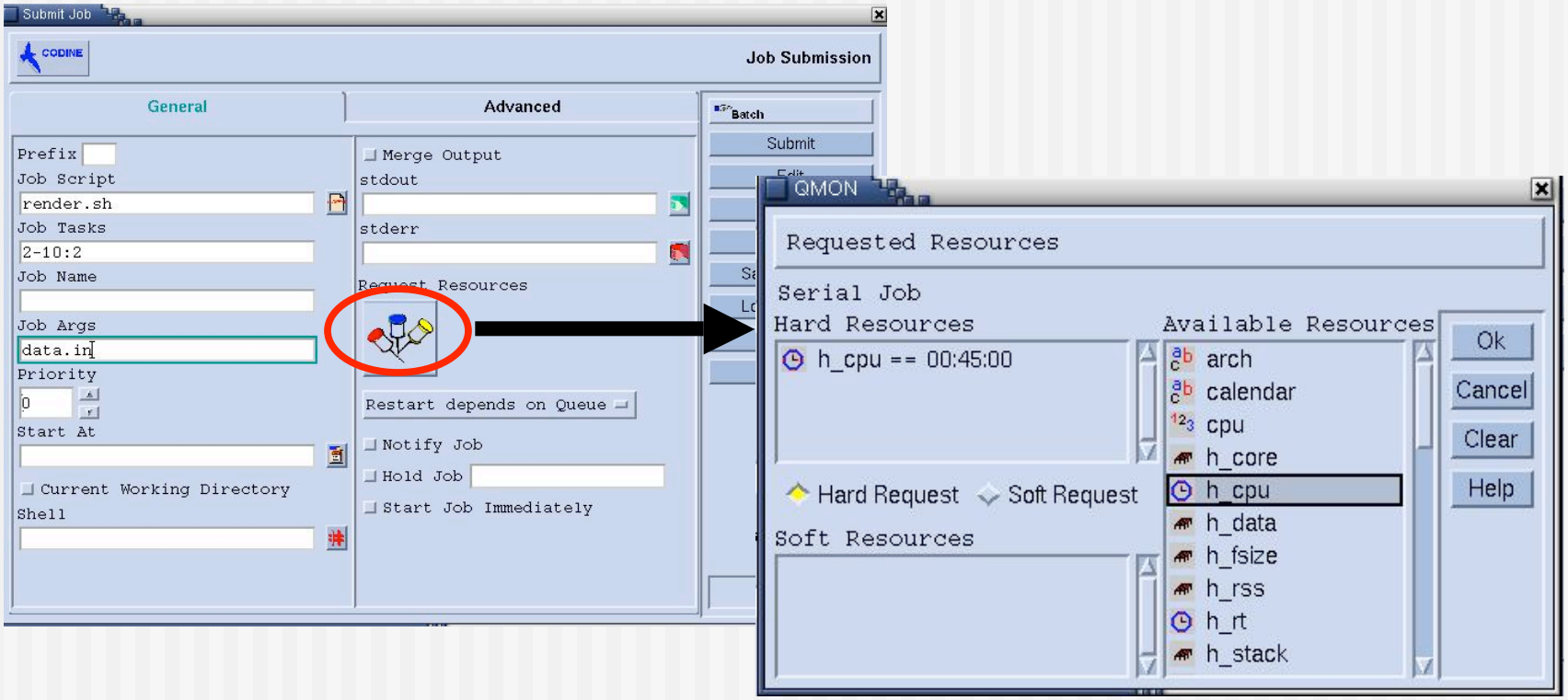
- Parameterized and repeated execution of the same program (in a script) is ideal for the *array job* facility
- High Throughput Computing (HTC)
  - Render farm for digital content creation
  - Lifescience -> BLAST!
- SGE provides efficient implementation of array jobs
  - Handle computations as an array of independent tasks joined into a single job
  - Can monitor and controlled as a total or by individual tasks or subset of tasks

$ qsub

- Submitting an Array Job from command line
  - -l option requests for a hard CPU time limit of 45mins
  - -t option defines the task index range
    - 2-10:2 specifies 2,4,6,8,10
- Uses **$SGE_TASK_ID** to find out whether they are task 2, 4, 6, 8 or 10
  - To find input record
  - As seed for random number generator

```
Scsadmin@hydra3 scsadmin]$ qsub -l h_cpu=0:45:0 -t 2-10:2 render.sh data.in
```

CLI vs. GUI

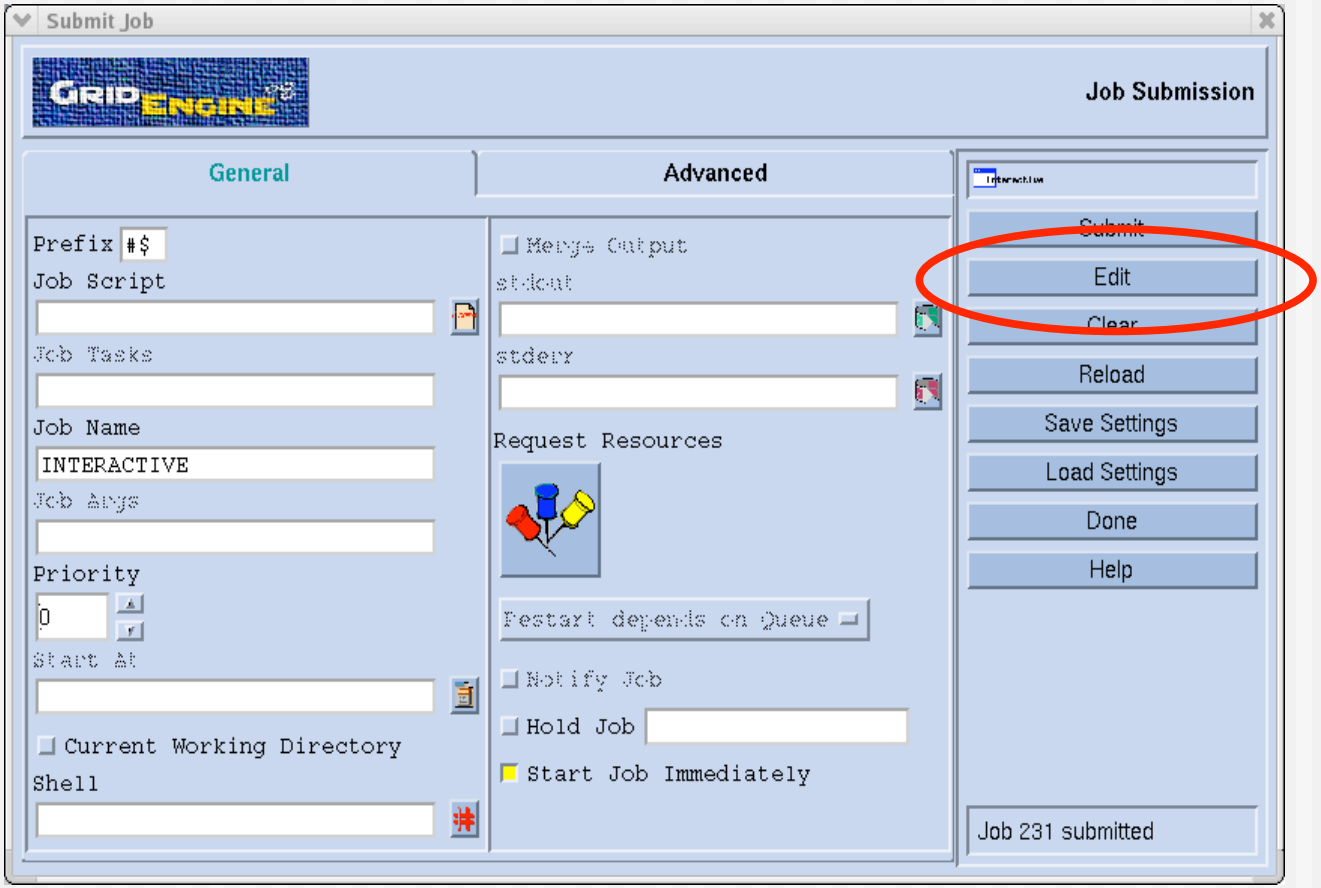



```
Scsadmin@hydra3 scsadmin]$ qsub -l h_cpu=0:45:0 -t 2-10:2 render.sh data.in
```

Interactive Submit

- Interactive jobs useful when your program needs your manual intervention, eg.
  - data input, x-applications, input to influence results
- A few methods in SGE
  - qsh
    - Submit an interactive X-windows login session to SGE
    - An xterm is brought up from the executing machine with display directed to $DISPLAY
  - qlogin
    - Submit an interactive login session to SGE like qsh above but does not open an xterm. It uses the current terminal for user I/O
  - qrsh
    - Submit an interactive rsh session to SGE
    - Similar to qlogin above.
    - Usually establish a rsh (ssh) connection and executes the command given

QMon



$ qsh

- Start xterm on any host that have at least 2 free slots

```
scsamdin@hydra3$ qsh -l slots=2
```

- Start xterm on a host that have a Fluent licence and a queue which have a minimum of 6 hours of hard CPU time limit

```
scsadmin@hydra3$ qsh -l fluent=1, h_cpu=6:0:0
```

$ qrsh
continued

- SGE provides qrsh, qtcsh (and qmake) for transparent remote execution of certain computational tasks
- Remote execution with qrsh
  - Remote execution of applications like rsh
  - Interactive login session like rlogin
  - Submission of batch jobs with terminal I/O
  - Submitting a standalone program not in shell script
  - Etc

$ qrsh
continued

- Qrsh understands almost all of qsub options plus additionals ones:
  - -now yes|no
    - scheduled immediately? Rejected if cannot start immediately
  - -verbose
    - More output, useful for debugging

```
scsadmin@hydra3 scsadmin]$ qrsh [options] program|shell-script
  [arguments] [> stdout_file] [>&2 stderr_file] [< stdin_file]
```

$ qtcsh

- Transparent Job Distribution with qtcsh
- Fully compatible replacement for UNIX C-shell (csh)
- Provides command-shell with capability of transparent distribution of designated applications to lightly loaded hosts
- Use of .qtask file in $HOME directory
- Default modes: remote, immediate, non-verbose

.qtask file

```
# .qtask file
netscape -v DISPLAY=myhost:0
```

Run this

```
scsadmin@hydra3$ netscape
```

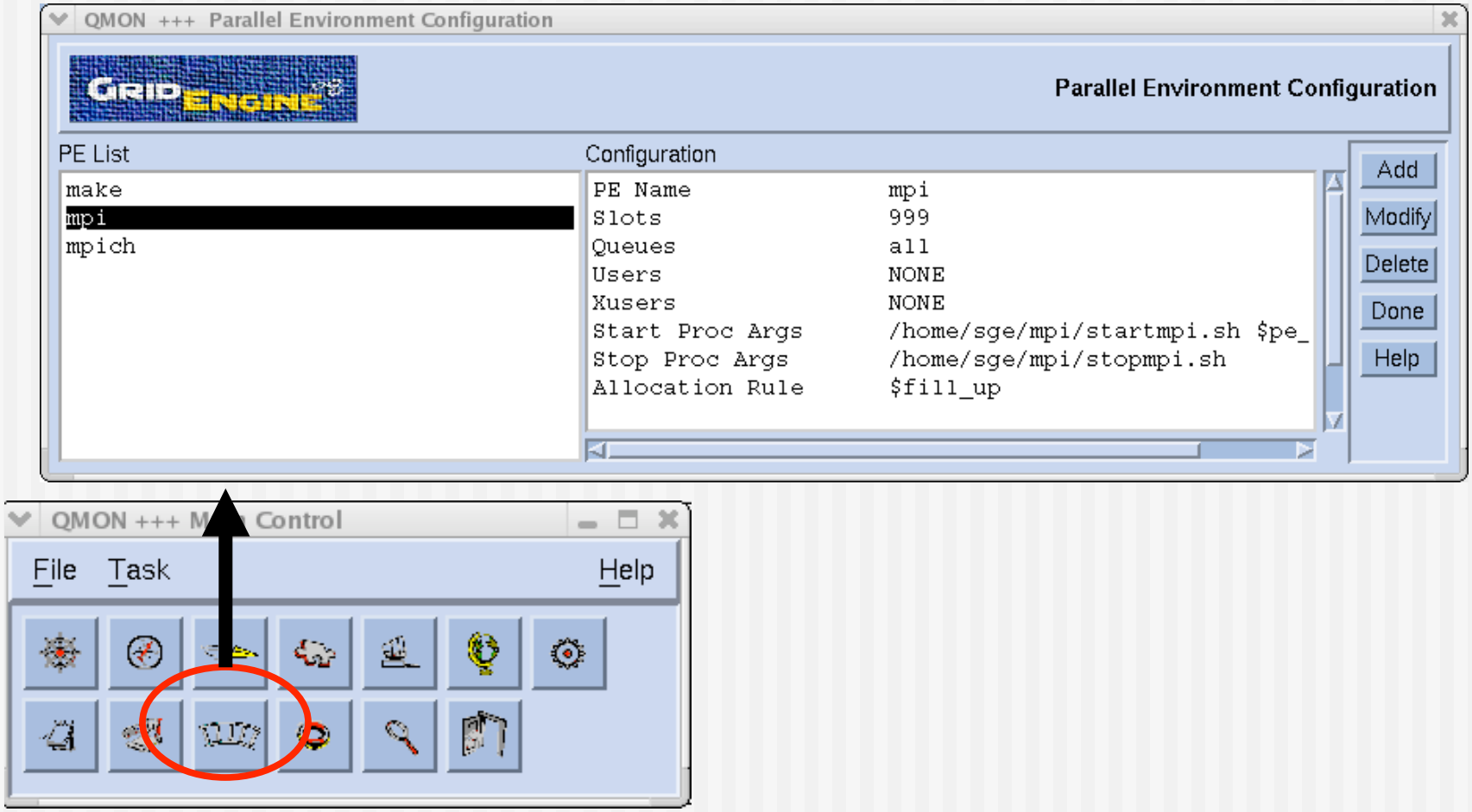
Get this!

```
Scsadmin@hydra3$ qrsh -v DISPLAY=myhost:0 netscape
```

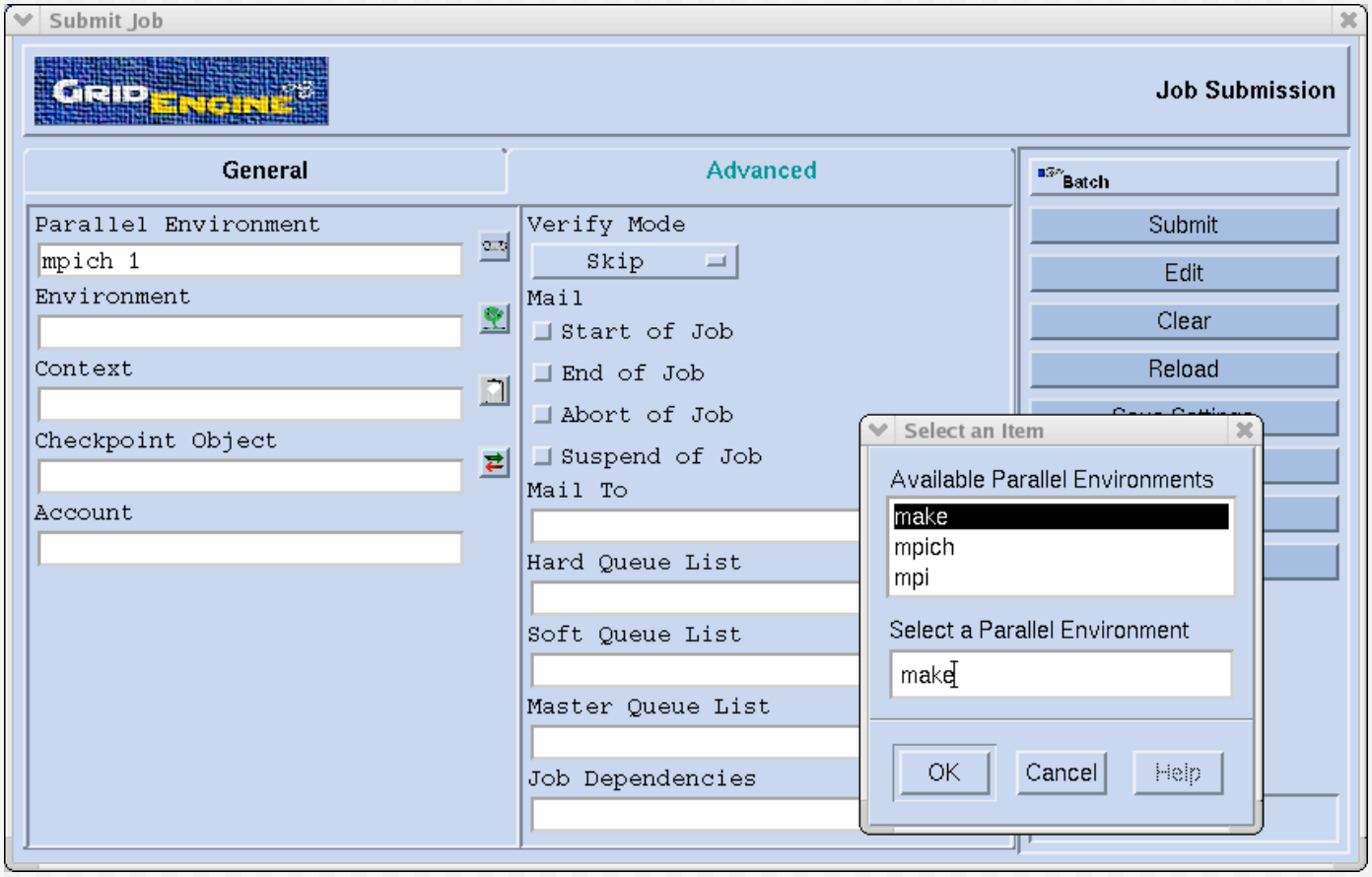
Parallel Submit

- SGE can execute parallel jobs
  - MPI or PVM
  - or shared memory parallel programs on multiple slots in single queues or distributed across multiple queues
  - And distributed memory parallel jobs across machines in multiple queues
- Use of Parallel Environment (PE)
  - Designed for concurrent/parallel computing
- Rocks has a default PE environment setup for MPICH

Parallel Submit



GUI



$ qsub
pe example

- Use PE = mpich
- Asks for 16 slots (16 CPUS)
- Needs to have Fluent licence and architecture must be Alpha
- Run the shell-script nastran.sh

```
scsadmin@hydra3$ qsub -pe mpich 16 -l fluent, arch=alpha nastran.sh
```

Using the PBS roll, a quick tour through the woods.

Roy Dragseth,
PBS roll maintainer.
royd@cc.uit.no

SDSC

Running jobs

- Provide as much info and as accurate as you can.
- All info can be provided as comments in the runscript or on as options to qsub.
- Options take precedence over comments.

A simple runscript

```
#/bin/bash
#PBS -lnodes=4:ppn=2
#PBS -lpmem=1000mb
#PBS -lwalltime=1:0:0
#PBS -M abe

mpiexec my-mpi-app
```

Reg shell heading

8 cpus

1000 MB per cpu

1 hour walltime

mail me on abort, begin and end

start the application.

mpiexec

- Uses the pbs task management (tm) interface to start mpi processes
- Much faster job startup than mpirun
- Cleaner interface than mpirun
- Better cleanup facilities at job failure.
- Reports real cpu-usage and memory utilization back to PBS.

The default setup

- The default setup is minimalistic.
  - One queue
  - FIFO scheduling
  - No restrictions on usage

Tuning the setup

- Most things can be achieved through maui reconfiguration
- Almost no need for setting up new queues

Maui setup

- Edit /opt/maui/maui.cfg
- run service maui restart

Job distribution

- Default is pack jobs into as few nodes as possible.
- Distribute jobs evenly over all nodes:

  NODEALLOCATIONPOLICY PRIORITY

  NODECFG[DEFAULT] PRIORITYF='- JOBCOUNT'

Prioritizing short jobs

- Use the Xfactor

  Xfactor = (walltime+queuetime)/walltime

XFACTORWEIGHT 1000

- I recommend to combine this with fairshare, or else the users will start running lots of short jobs.

Fairshare

- Prioritize based on historical usage

```
FSPOLICY              PSDEDICATED
FSDEPTH               7
FSINTERVAL              24:00:00
FSDECAY               0.80
USERCFG[DEFAULT]      FSTARGET=25.0
```

Imposing limits

- Limits can be set on users, groups, queues and QOS.
- Limiting how many cpus a user can allocate

USERCFG[DEFAULT] MAXPROC=90,150

- 90 is soft limit, 150 is hard limit.

Getting info about the system

- showq list the current jobs both idle and running
  - showq -r stats of runing jobs
- checkjob list info about one job
- checknode list info about one node
- qstat -a, -f, -n, -u
- tracejob -n days, trace a jobs history

Troubleshooting

- When a job won't die
  - First try qdel (but that probably wont help)
  - Then try qsig -sNULL
  - Then bring out the big hammer:
    - Stop pbs_server: qterm -t quick
    - Delete the .SC and .JB file from /opt/torque/server_priv/jobs/
    - Start the server: /opt/torque/sbin/pbs_server

Recover from node failures

- PBS detects most node failures and marks the node as down.
- Sometimes a pbs bug makes the node semi down, e.g. the node is up but pbs_mom isn't accepting jobs.
  - This is nasty and hard to correct without a node restart.
  - Check if the / filesystem on the node is full.

Rocks errors.

- Sometimes rocks detects wrong number of cpus.
  - Esp. on Itaniums.
  - Hyperthreading.
- Fix the database, then the nodelist:

```
mysql -u apache -Dcluster -e 'update nodes set cpus = 2 where
    name like “compute-0-0”;'
dbreport pbs-nodes | grep compute-0-0 | bash
```

Odds and ends.

- Jobs will survive daemon restarts.
  - Be careful with mpi-jobs and restarting pbs_mom.
  - Reports have indicated that jobs are killed on maui restart.
    - might be due to missing walltime limits on jobs??
  - Restarting pbs_server through the init.d script will destroy the job-node mapping for existing jobs, only affects ganglia.

Future plans.

- Switch to moab?
  - moab has more features than maui.
  - Licensing issues.
- Allocation manager.
  - qbank is dead.
  - have no experience with gold.
- Tighter integration with ganglia.

Resources

- Maui and torque homepage
  http://www.clusterresources.com
- The snowstorm cluster page
  http://uit.no/itavd/HPC-Cluster/
  - Contains some user guides in english.
- Hopefully a pbs-roll homepage soon...