Administrator Training for University of Richmond

advanced clustering
technologies, inc.

2

Agenda

- Cluster overview
  - Physics hardware
  - Chemistry hardware
- Software
  - Modules, ACT Utils, Cloner
- GridEngine overview

advanced clustering technologies, inc.

3

Cluster overview

- 2 similar but separate systems (physics and chemistry)
- Physics
  - 1 head node
  - 1 storage node
  - 20x compute nodes
- Chemistry
  - 1 head node
  - 2x 12GB “quantum” nodes
  - 33x 24GB “quantum” nodes
  - 4x 48GB “quantum nodes”
  - 22x “molecular” nodes with InfiniBand

advanced clustering technologies, inc.

4

Head Node (qty 1) - Physics

- Hardware Specs
  - Dual Six-Core X5650 “Westmere” 2.66GHz processors
  - 24GB of RAM (6x 4GB DIMMs, 6 available for expansion)
  - 4x 500GB drives drives in hardware RAID5 with battery backed cache
  - One dual port ConnectX-EN card (2x 10G ethernet)
- Hostname / networking
  - head
  - 10 Gige: 10.1.1.254
  - ipmi: 10.1.3.254
  - public: 141.166.8.121
- Roles
  - DHCP/TFTP servers
  - Torque queue master

advanced clustering technologies, inc.

5

Storage Node (qty 1) - Physics

- Hardware Specs
  - Single Six-Core X5650 “Westmere” 2.66GHz processors
  - 12GB of RAM (3x 4GB DIMMs, 3 available for expansion)
  - 8x 1TB drives drives in hardware RAID5 with battery backed cache
  - One dual port ConnectX-EN card (2x 10G ethernet)
- Hostname / networking
  - storage
  - 10 Gige: 10.1.1.253
  - ipmi: 10.1.3.253
- Roles
  - NFS Server

6

Compute Nodes (qty 20) - Physics

- Hardware Specs
  - Dual Six-Core X5650 “Westmere” 2.66GHz processors
  - 24GB of RAM (6x 4GB DIMMs, 6 available for expansion)
  - 2x 500GB drive for operating system (software RAID-0)
- Hostname / networking
  - physics01 - physics20
  - gige: 10.1.1.1 - 10.1.1.20
  - ipmi: 10.1.3.1 - 10.1.3.20

advanced clustering
technologies, inc.

7

Head Node (qty 1) - Chemistry

- Hardware Specs
  - Dual Six-Core X5650 “Westmere” 2.66GHz processors
  - 12GB of RAM (6x 2GB DIMMs, 6 available for expansion)
  - 13x 2TB drives drives in hardware RAID6 with battery backed cache
  - One dual port ConnectX-EN card (2x 10G ethernet)
- Hostname / networking
  - head
  - 10 Gige: 10.1.1.254
  - ipmi: 10.1.3.254
  - public: 141.166.8.122
- Roles
  - DHCP/TFTP servers
  - SGE qmaster / Nagios monitor
  - NFS server

8

“Quantum” (qty 39) - Chemistry

- Hardware Specs
  - Dual Six-Core X5650 “Westmere” 2.66GHz processors
  - Multiple RAM configurations
    - 2x nodes with 12GB of RAM
    - 33x nodes with 24GB of RAM
    - 4x nodes with 48GB of RAM
  - 2x 500GB drive for operating system (software RAID-0)
- Hostname / networking
  - quantum01 - quantum39
  - gige: 10.1.1.1 - 10.1.1.20
  - ipmi: 10.1.3.1 - 10.1.3.20

advanced clustering
technologies, inc.

9

“Molecular” (qty 33) - Chemistry

- Hardware Specs
  - Dual Six-Core X5650 “Westmere” 2.66GHz processors
  - 12GB of RAM (6x 2GB DIMMs, 6 available for expansion)
  - 2x 500GB drive for operating system (software RAID-0)
  - ConnectX DDR (20GB/s InfiniBand on motherboard)
- Hostname / networking
  - molecular01 - molecular33
  - gige: 10.1.1.1 - 10.1.1.20
  - ipmi: 10.1.3.1 - 10.1.3.20

advanced clustering
technologies, inc.

10

Compute Nodes enclosure

1U enclosure that holds 2x Compute blades

advanced clustering technologies, inc.

11

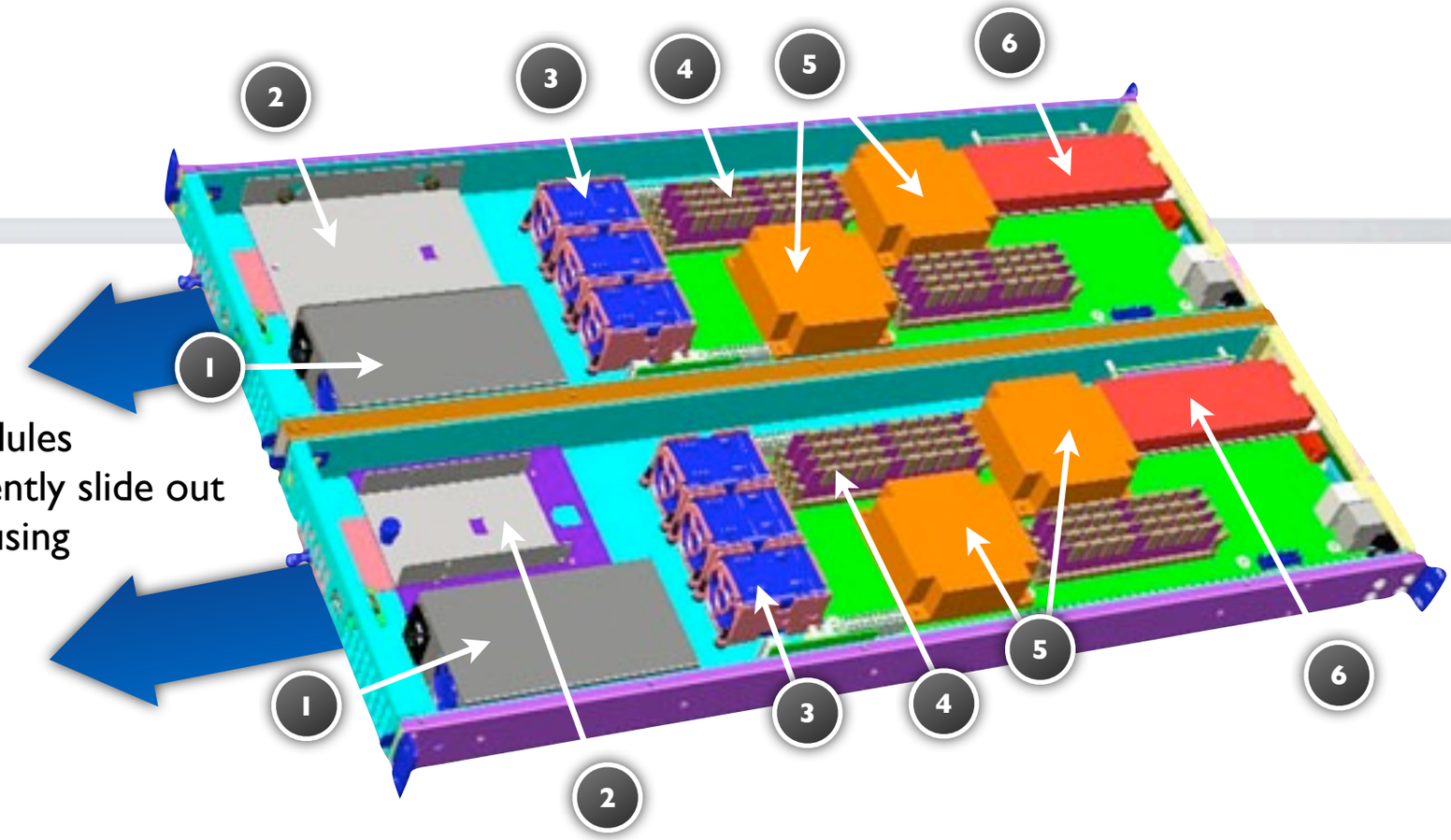
Compute Nodes blade

1	Power LED	4	Slide out ID label area
2	Power Switch	5	Quick release handles
3	HDD LED		

advanced clustering
technologies, inc.

Compute Nodes inside

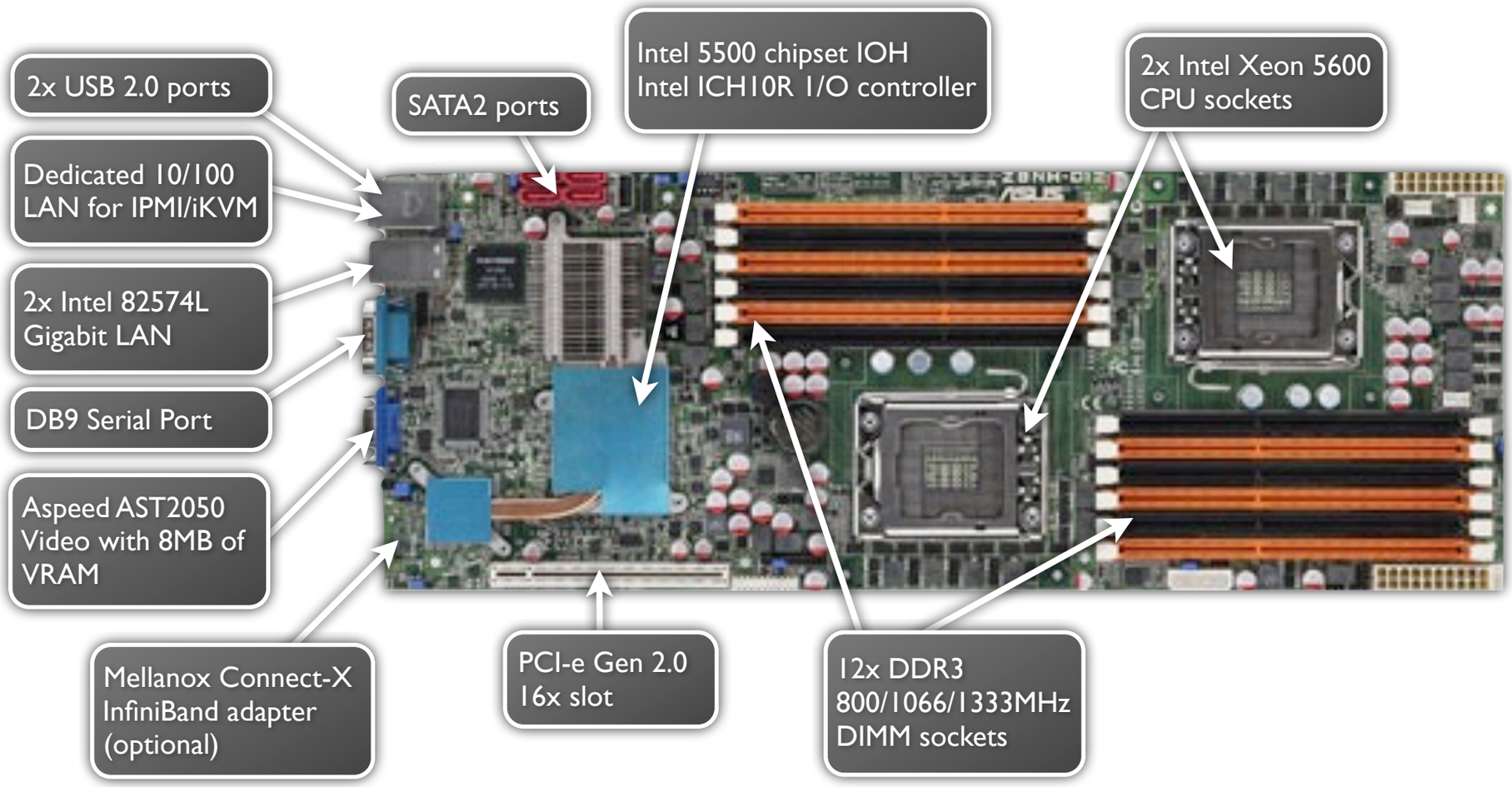
blade modules independently slide out of the housing



1	Power supply	4	System memory
2	Drive bay (1x 3.5”)	5	Processors
3	Cooling Fans	6	Low-profile expansion card

13

Compute Node motherboard



advanced clustering technologies, inc.

14

Management / IPMI network

- Each server has a dedicated 100Mb IPMI network interface that is independent of the host operating system

advanced clustering technologies, inc.

15

Software

- Modules
- ACT Utils
- Cloner

advanced clustering technologies, inc.

16

Modules command

- Modules is an easy way to setup the user environment for different pieces of software (path, variables, etc).
- Setup your .bashrc or .cshrc
  - source /act/etc/profile.d/actbin.[sh|csh]
  - source /act/Modules/3.2.6/init/[bash|csh]
  - module load null

advanced clustering technologies, inc.

17

Modules continued

- To see what modules you have available:
  - module avail
- To load the environment for a particular module:
  - module load modulename
- To unload the environment:
  - module unload modulename
  - module purge (removes all modules from environment)
- Modules are stored /act/Modules/3.2.6/modulefiles - can customize for your own software

18

ACT Utils

- ACT Utils is a series of commands to assist in managing your cluster, the suite contains the following commands:
  - act_authsync - sync user/password/group information across nodes
  - act_cp - copy files across nodes
  - act_exec - execute any Linux command across nodes
  - act_netboot - change network boot functionality for nodes
  - act_powerctl - power on, off, or reboot nodes via IPMI or PDU
  - act_sensors - retrieve temperatures, voltages, and fan speeds
  - act_locate - turn on/off node locator LED
  - act_console - connect to the hosts's serial console via IPMI

19

ACT Utils common arguments

- All utilities have a common set of command line arguments that can be used to specify which nodes to interact with
  - --all all nodes defined in the configuration file
  - --exclude a comma seperated list of nodes to exclude from the command
  - --nodes a comma separated list of node hostnames (i.e. node001,storage01)
  - --groups a comma separated list of group names (i.e. nodes, storage, etc)
  - --range a “range” of nodes (i.e. node001-node050)
- Configuration (including groups and nodes) defined in /act/etc/act_utils.conf

20

Groups defined on your cluster

- Physics
  - nodes
- Chemistry
  - quantum-12g
  - quantum-24g
  - quantum-48g
  - molecular
  - quantum
  - nodes

21

ACT Utils examples

- Find the current load on all the compute nodes
  - act_exec -g nodes uptime
- Copy the /etc/resolv.conf file to all the login nodes
  - act_cp -g nodes /etc/resolv.conf /etc/resolv.conf
- Shutdown 1 of the storage nodes and every compute node except node001
  - act_exec --node=storage01 --group=nodes --exclude=node001 /sbin/poweroff
- tell nodes node001 and node003 to boot into cloner on next boot
  - act_netboot --nodes=node001,node003 --set=cloner-v3.14

22

Shutting the system down

- To shut the system down for maintenance (run from head node):
  - act_utils -g nodes,gpu,storage /sbin/poweroff
- Make sure you shut down the node you are on last by just issuing the poweroff command
  - /sbin/poweroff

23

Cloner

- Cloner is used to easily replicate and distribute the operating system and configuration to nodes in a cluster
- Two main components:
  - Cloner image collection command
  - A small installer environment that is loaded via TFTP/PXE (default), CD-ROM, or USB key

24

Cloner image collection

- Login to the node you'd like to take an image of, and run the cloner command (this must execute on the machine you want to take the image, it does not pull the image, it pushes it to a server)
  - /act/bin/cloner --image=IMAGENAME --server=SERVER
- Arguments:
  - IMAGENAME = a unique identifier label for the type of image (i.e. node, login, etc)
  - SERVER = the hostname running the cloner rsync server process

25

Cloner data store

- Cloner data and images in /act/cloner/data
  - /act/cloner/data/hosts - individual files named with the system's hardware MAC address. These files files are used for auto-installation of nodes. File format include two lines:
    - IMAGE="imagename"
    - NODE="nodename"

26

Cloner data store

- /act/cloner/data/images - a sub directory is automatically created for each image with the name specified with the --image argument when creating
  - sub directory called “data” that contains the actual cloner image (an rsync’d copy of the system cloned).
  - sub directory called “nodes” and a series of subdirectories with the name of each node (i.e. data/images/storage/nodes/storage01, data/images/storage/nodes/storage02)
    - the node directory is an overlay that gets applied after the full system image
    - can be used for customizations that are specific to each node (i.e. hostname, network configuration, etc).

27

Creating node specific data

- ACT Utils includes a command to assist in creating all the node specific configuration for cloner (act_cfgfile)
- Example: create all the node specific information
  - act_cfgfile --cloner --cloner_path=/act/cloner --cloner_os=redhat
  - Writes /act/cloner/data/images/IMAGENAME/nodes/NODENAME/etc/sysconfig/networking, ifcfg-eth0, etc for each node

28

Installing a cloner image

- Boot a node from the PXE server to start re-installation
- Use act_netboot to set the network boot option to be cloner
  - act_netboot --set=cloner-v3.14 --node=node001
- On next boot system will unattended install the new image
- After node re-installation set network boot option to boot from the local disk
  - act_netboot --set=localboot --node=node001

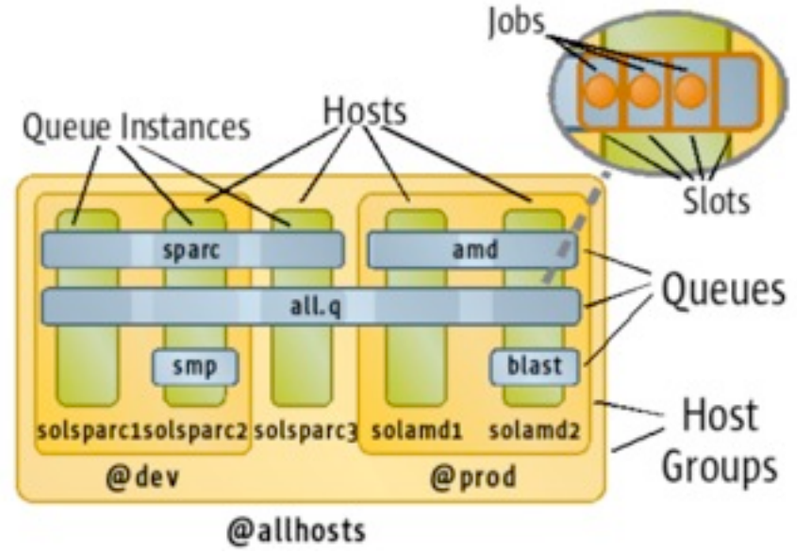
29

GridEngine introduction

- Basic setup
- Parallel environments
- Complexes
- Submitting batch jobs
- Job status
- Interactive jobs
- Graphical user control
- Graphical queue status

advanced clustering technologies, inc.

Basic setup



- Each node has N slots which can be used for any number of jobs
- Queues are currently defined which have different rules applied to them]
- Queues can over-lap on the same host but running a job in one queue deducts from the total number of slots on that node available to other queues (viewable with qhost -q)

31

Parallel Environments

- In addition to selecting a queue to run in using the "-q" qsub/qrsh parameter, you can select the parallel environment using the "-pe" environment.
- This does two things:
  - It's the only way to select more than one slot at a time--that is, any kind of parallel job. (Most of your jobs will be parallel.)
  - It adds additional environmental control over the top of that already provided by the queue. For example, running a job under "-q sas.q -pe threaded 12", "stacks" the environment configured for SAS and threaded applications in to the same run making it possible to both run SAS and to ensure that threaded applications have the needed environment.

32

Complexes

- Due to the limited number of software licenses and large memory consumption, jobs that use software licenses or large amounts of memory must request a "complex" resource by name. For example, in the case of Stata, a Stata license is requested by passing "-l stata=1" to request 1 Stata software license. Also in the case of Stata, "-l mem_free=20G" is used to request 20 gigabytes of RAM for the job.
- Grid Engine examines your request and will avoid over-scheduling instances of a program which require software licenses or memory which are currently being used by others or other instances of software being run by you.
- To check available license or memory, use "qstat -F" to show resource consumption by host.

33

Submitting batch jobs

- Basic syntax: qsub -q queuename.q -pe parallel_env slots jobscript
- jobscripts are simple shell scripts in either SH or CSH which at a minimum contain the name of your program. Here is the minimum jobscript:

```
#!/bin/sh
/path/to/executable
```

34

Submitting batch jobs

- A moderately complex job script can suggest command line parameters to SGE (prefixed with #$) that you may have left off of qsub as well as perform environment setup before running your program:

```
#!/bin/sh
#$ -N testjob
#$ -cwd
#$ -j y
#$ -q sas.q
#$ -pe threaded 12
echo Running on $(hostname).
echo I was given ${NSLOTS} slots on the following hosts:
cat ${PE_HOSTFILE}
echo It is now $(date).
sleep 60
echo It is now $(date).
```

35

Job status

- You can check your own job submission status by looking at the output of "qstat". qstat only shows your own jobs, be default.
- To show all jobs, run "qstat -u '*'". The priority listed in the left hand column affects the order in which jobs are executed.
- To examine a detailed explanation why a job hasn't started yet, use "qstat -f -j jobid"

36

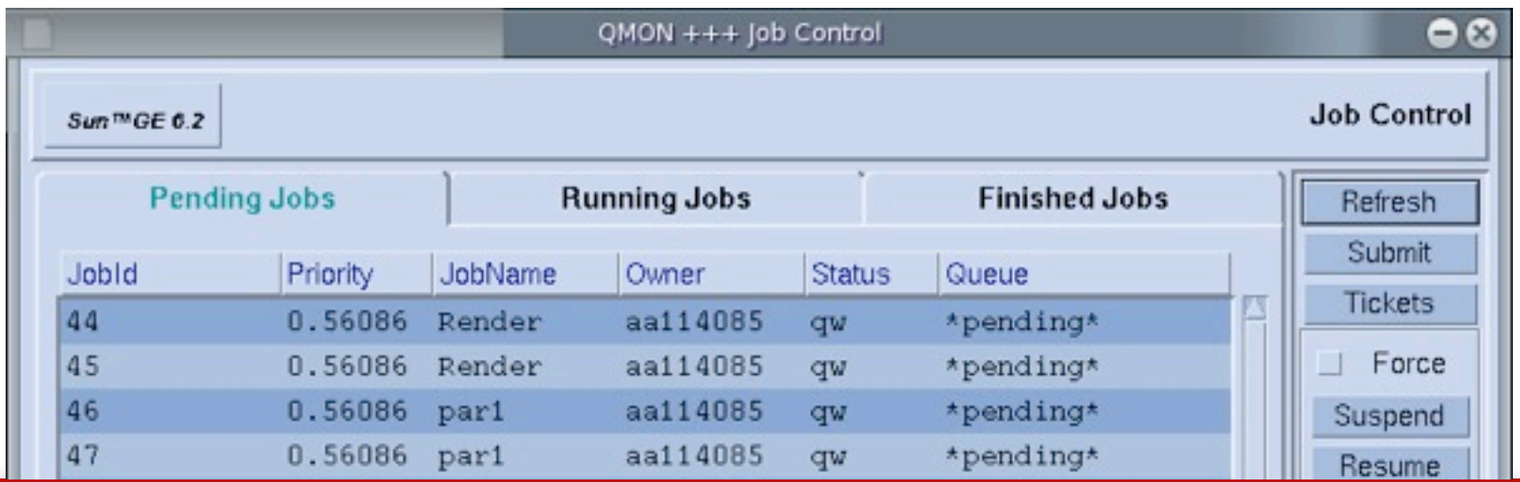
Interactive jobs

- Using "qrsh", you can start a job immediately under your control or not at all. If there aren't any free nodes to run the job, the command will exit telling you.
- When a job is scheduled, it lands you in a shell on the remote machine or immediately executes the program that you have provided as an argument to qrsh. This includes X11/graphical forwarding.
- You can pass any argument to qrsh that you can pass to qsub including "-pe" to take more than one slot on a node.
- When you exit, the resources are immediately freed for others to use.

37

Graphical user control

- qmon can be used to do anything done at the command line including submitting jobs. qmon is launched from the login nodes.
- The below screen shot shows buttons on the right which can control any of the related job functions. A job which is held, will not schedule. A suspended job is frozen but its memory is not freed. "Why?" will provide output similar to "qstat -f jobid" explaining why a job hasn't yet run.



38

Graphical queue status

- An output similar to "qstat -q" can be seen by examining the queues screen, especially "queue instances". For nodes are that are marked in an "E" (error) state, "Explain" will explain why. Admins can clear this error state.