# Using Sun Grid Engine on the BRC-MH Linux Cluster

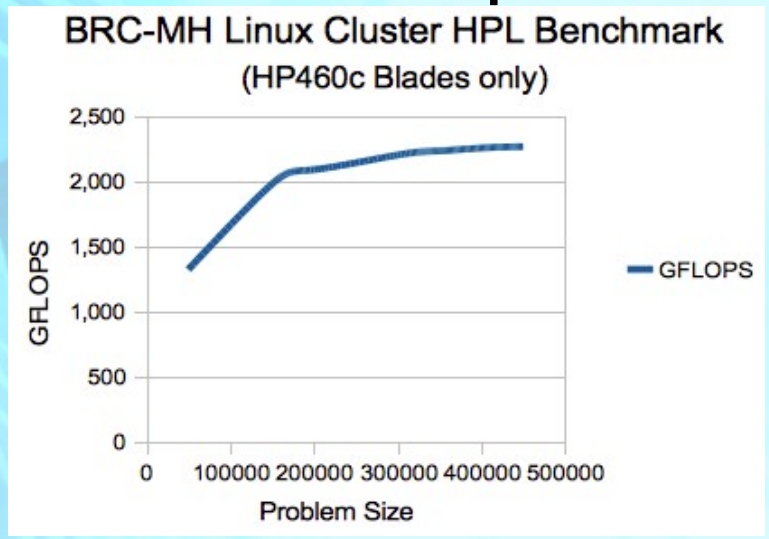
Updated: 20th April 2011

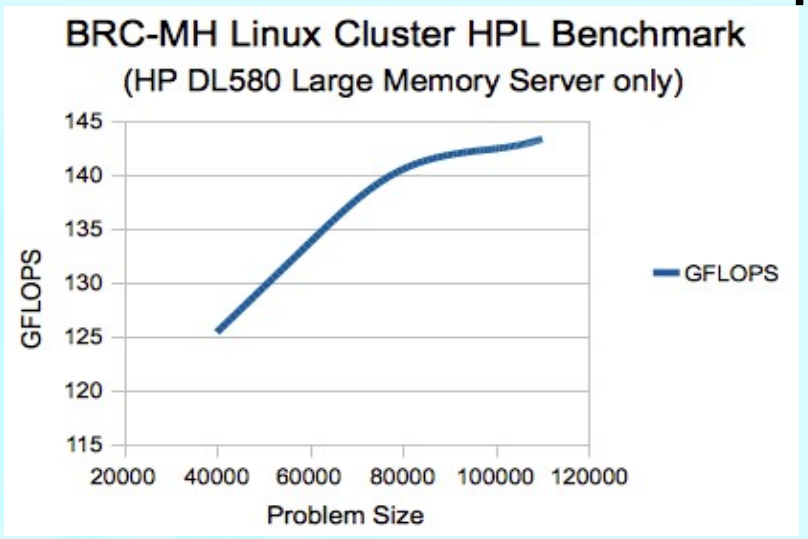
# Brief Introduction

- 30 x HP 460c G6 Blades (8 cores each)
- 2 x HP 460c G7 Blades (12 cores each)
- 1 x DL580 G5 (24 cores)
- Total: 288 cores
- 2.340 TB memory

# Brief Introduction

- Using HPL 2.0
  - 30 Blades: 2.272 TFlops
  - DL580: 142 GFlops
- Combined performance around 2.4 TFlops



# Brief Introduction

- 120TB of storage provided by 3 Panasas storage arrays.
- Snapshots provide immediately recoverable checkpoints
- Backed up to tape
  - /home – max usage 100GB (per user)
  - /project – max usage 500GB
- Not backed up to tape
  - /scratch

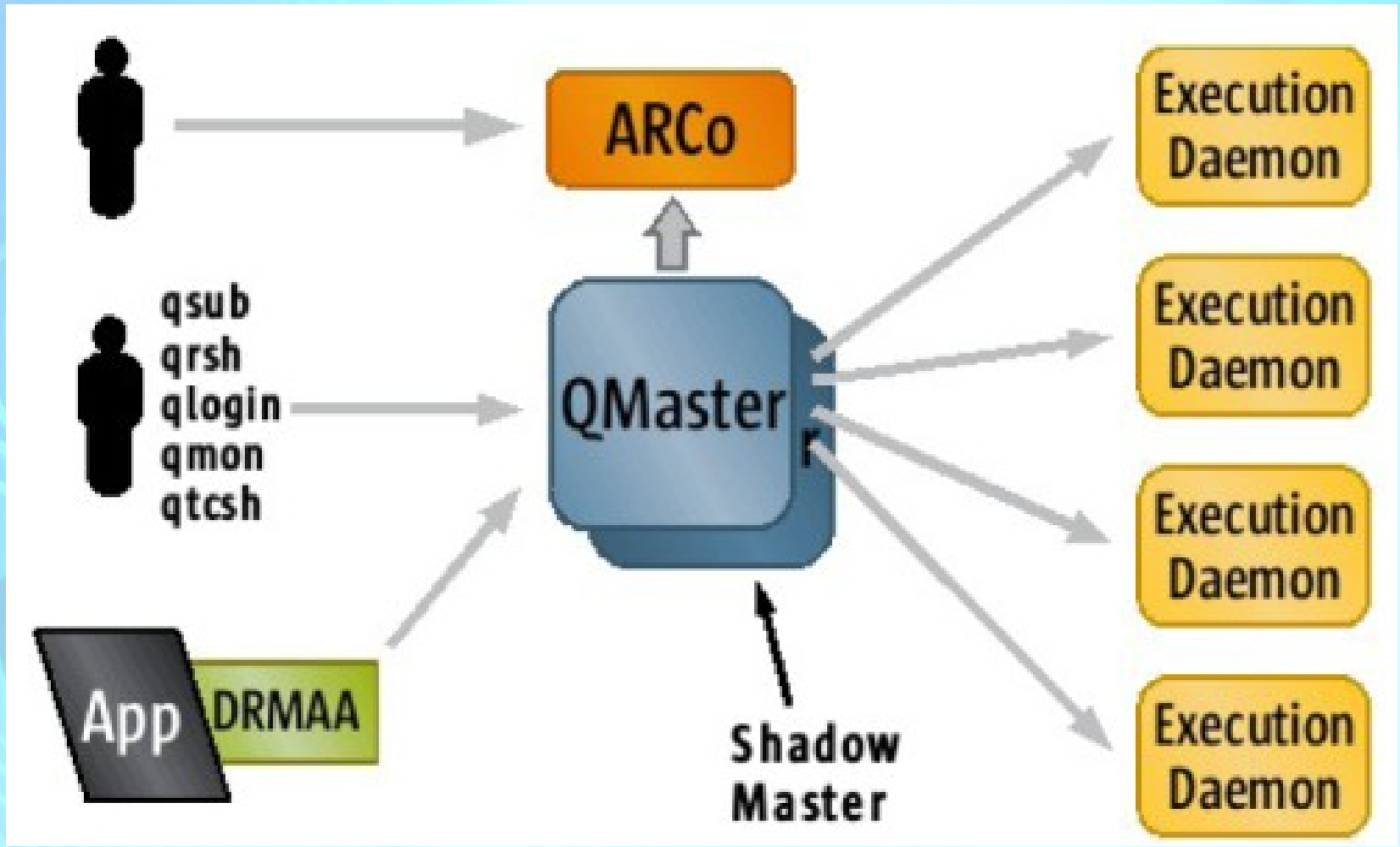
# Brief Introduction

- Wiki: https://compbio.brc.iop.kcl.ac.uk:8443/biowiki
  - Documentation for cluster
  - Description of datasets which reside on the cluster
  - Apart from some cluster documentation all users able to edit the wiki
  - Please document your datasets and add information where applicable

# Oracle (Sun) Grid Engine

- SGE is an open source batch-queuing system (version 6.2u5 installed)
- Permits automated allocation of the cluster resources for each job
- Documentation: http://wikis.sun.com/display/gridengine62u2/Grid+Engine+Documentation+(Printable)
- You must use it to run your jobs on the cluster!

# Oracle (Sun) Grid Engine

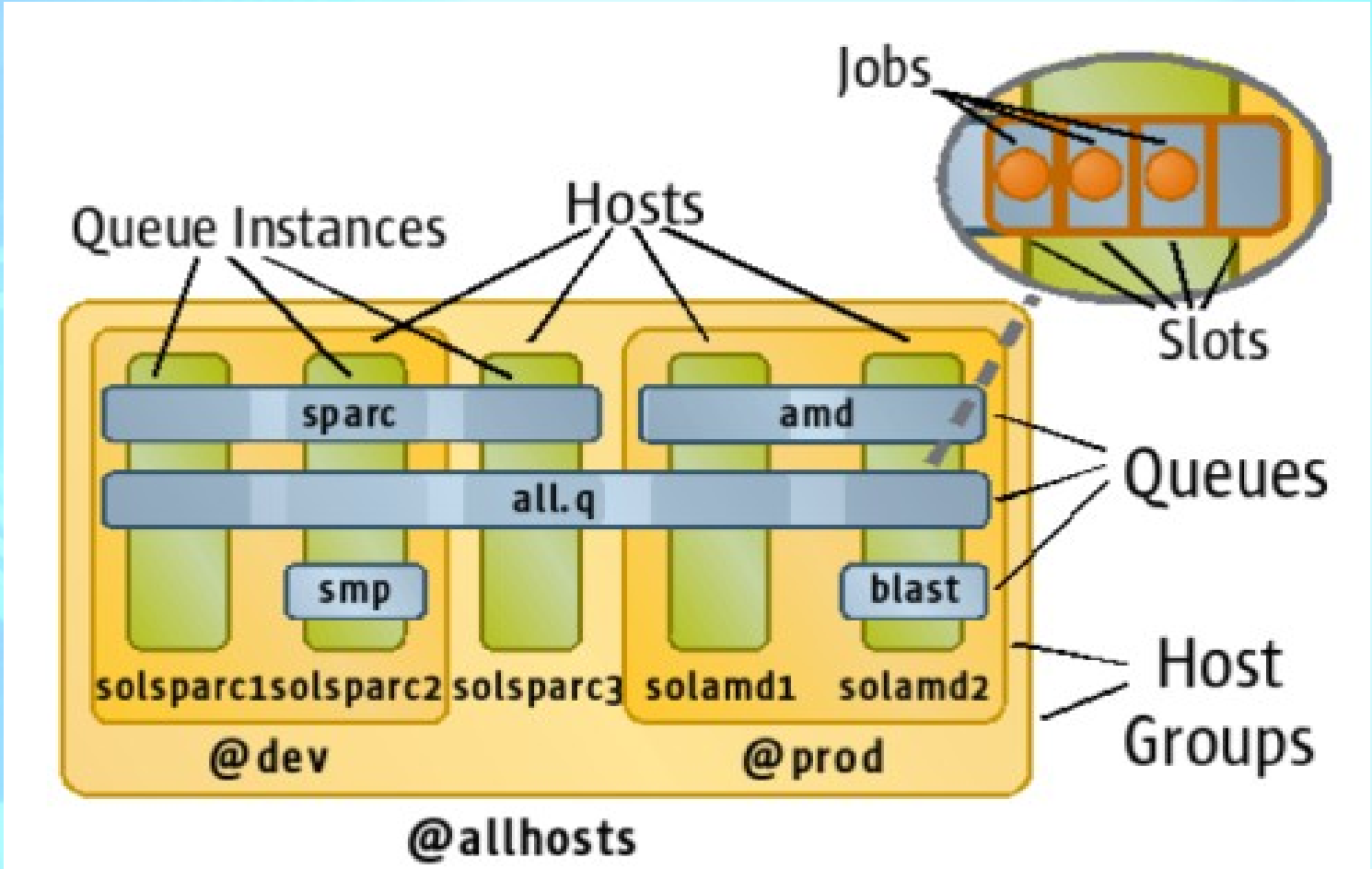


# Queues

- Each CPU core equates to a 'slot'.
- Nodes/slots are grouped into 'Queues'
- Jobs are submitted to queues

```
$ qconf -sql
all.q       ← disabled
short.q     ← jobs < 5 days
long.q      ← low priority jobs
bignode.q   ← bignode only jobs
test.q      ← for testing (2 hours)
```

# Queues


Jobs
Queue Instances
Hosts
Slots
sparc
amd
all.q
smp
blast
Queues
solsparc1
solsparc2
solsparc3
solamd1
solamd2
Host Groups
@dev
@prod
@allhosts

# Types of jobs in SGE

- 4 types of jobs
  - Batch job
  - Array job
  - Parallel job
  - Interactive job

# Batch job: An example

- Create a job script: my_sge_batch-job.sh

```
#!/bin/sh
#$-S /bin/sh
#$-o /home/dto/sge-demo/sge-output
#$-e /home/dto/sge-demo/sge-output
#$-q short.q,long.q,bignode.q
#$-l h_vmem=1G
#$-pe multi_thread 1
/home/dto/sge-demo/programs/get_freq.pl \
/home/dto/sge-demo/file-input/infile.txt
```

- Submit the job

```
$ qsub my_sge_batch-job.sh
```

# Array job: An example

- Create a job script: my_sge_array-job.sh

```
#!/bin/sh
#$-S /bin/sh
#$-t 1-100
#$-o /home/dto/sge-demo/sge-output
#$-e /home/dto/sge-demo/sge-output
#$-q short.q,long.q,bignode.q
#$-l h_vmem=1G
#$-pe multi_thread 1
/home/dto/sge-demo/programs/get_freq.pl \
/home/dto/sge-demo/file-input/infile.$SGE_TASK_ID.txt
```

- Submit the job

```
$ qsub my_sge_array-job.sh
```

# Array job: An example

- Use **`$SGE_TASK_ID`** to reference the task ID
- Managing array jobs
  - 1 to 100: #$-t 1-100
  - 50 to 100: #$-t 50-100
  - 1 to 100, but only evens: #$-t 1-100:2
  - 1 to 100, but only 5,10,15...:

    #$-t 1-100:5
- Max running tasks per array job: 200.
- Max tasks per array job: 20,000.
- Throttling: -tc max_running_tasks

# Parallel jobs

- Multi-threaded jobs: same as batch/array
- Submit jobs declaring number of threads

```
#$-pe multi_thread X
```

- Number of the threads == number of slots
- Remember number of cores on each node.
- We won't cover MPI jobs for now

# Memory Allocation

- Jobs use a default of 2GBytes of memory
- Submit jobs declaring amount of memory

```
#$-l h_vmem=XG
```

- Units are: G=GigaBytes, M=MegaBytes, K=KiloBytes
- This is the amount of memory your job will use **per thread**.
- Requested memory = Total memory/number of threads
- If your job uses a max of 12GBytes, but uses four threads, ask for 3GBytes

```
#$-l h_vmem=3G
```

# Rules

- All jobs should be submitted to SGE, and not run via the command line on bignode
- Array jobs that run longer than a day must be throttled to a maximum of 40 concurrent jobs

```
#$-tc 40
```

- Do not request more resources than you require
- Do not submit threaded jobs without specifying how many threads the job will use
- Do not abuse priority

# Priority of jobs

- Priority of jobs is set to -100 by default.
  - Users can change their priority from -1023 to 0
  - Admin can change their priority from -1023 to 1024
  - Higher the number, the higher the priority
- Use: `#$-p <priority>` to change the priority when you submit.
  - eg. `#$-p -50`
- Note: this priority gets mixed with other options to create a priority between 0 and 1 (when viewed with `qstat -u "*"`)
- Be wise about using this – only bump your jobs to a higher priority if they are urgent.

# Gotchas

- Use: `#$-S /bin/sh` not `#$-S /bin/bash`
- Request how much memory you'll need
  - Requesting too little, your job will fail
  - Requesting too much, prevents others using resources
  - Requesting more than exists – your job will never star
- Test your jobs on the **test.q**
  - View them running: `ssh testnode`

# Helpful commands

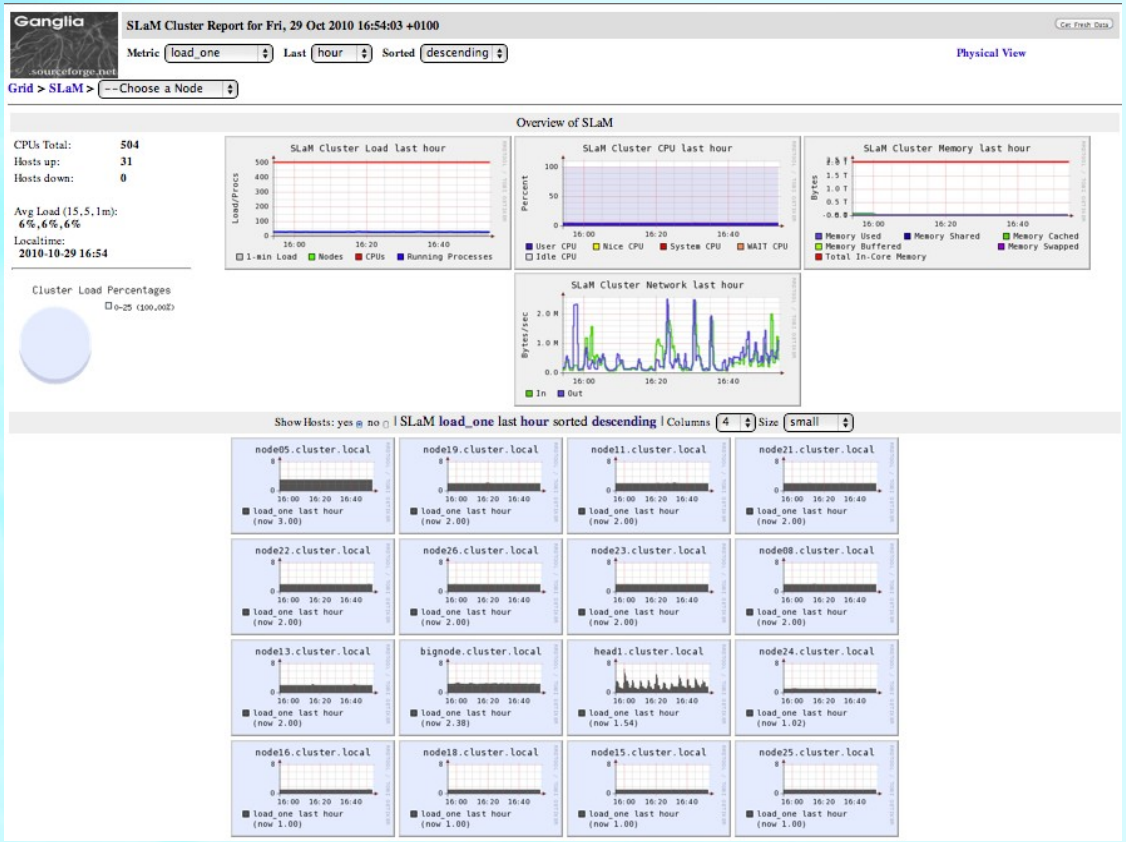
- View all cluster jobs: `qstat -f`
- View all running/queued jobs: `qstat -u "*"`
- View your jobs: `qstat -u <username>`
- Details on a specific job: `qstat -j <jobid>`
- Stop/remove a job: `qdel <jobid>`
- Change priority of queued jobs:
  - All: `qalter -p -500 -u <username>`
  - Specific: `qalter -p -500 <jobid>`
- List all queues: `qconf -sql`

# Helpful options

- Receive email when jobs start/finish
  - `#$ -M your.email@kcl.ac.uk`
  - `#$ -m be`
- Give a name to your job:
  - `# -N NameOfJob`
- Execute job in the current working directory:
  - `# -cwd`
- Execute job in a specified working directory:
  - `# -wd working-dir`

# Ganglia

- https://cluster.slam-services.org/ganglia/



# Questions

- See more at:
- http://compbio.brc.iop.kcl.ac.uk/cluster/
- https://compbio.brc.iop.kcl.ac.uk:8443/biowiki/
- Contact Cass via:
  - Email: caroline.johnston@kcl.ac.uk
  - Skype: brc-mh_linux_support