JOHNS HOPKINS BLOOMBERG SCHOOL of PUBLIC HEALTH

Enigma, Sun Grid Engine (SGE), and the High Performance Scientific Computing Center (HPSCC) Cluster

http://www.biostat.jhsph.edu/bit/SGE_lecture.ppt.pdf

Marvin Newhouse

The Dept. of Biostatistics HPSCC:

Director: *Fernando J. Pineda*

Technology Director/Manager: *Marvin Newhouse*

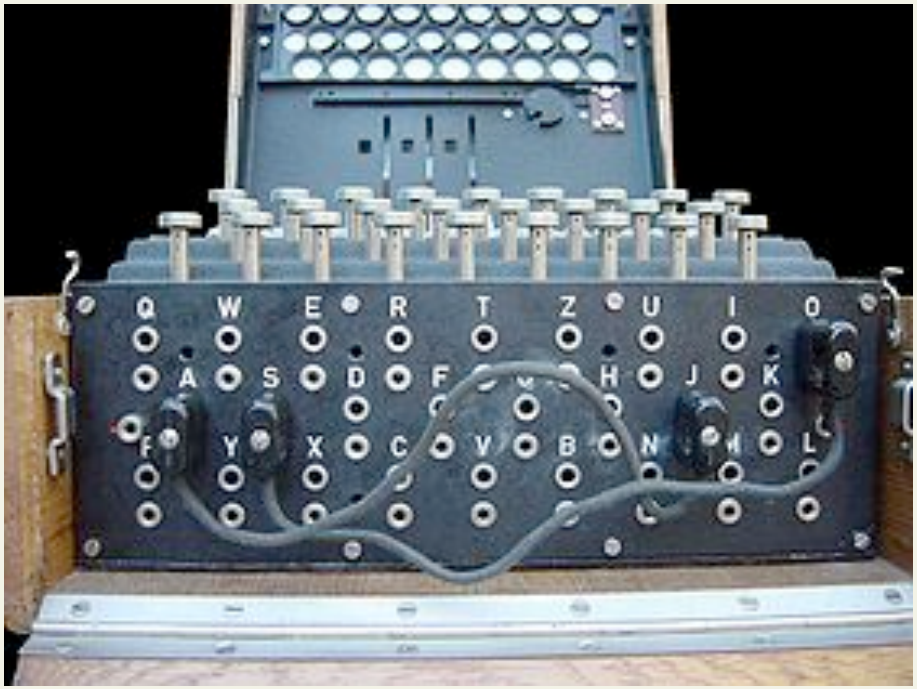
Systems Engineer: *Jiong Yang*

Financial Administrator: *Cindy Hockett*

2010Oct28

The first Enigma machine ?

http://en.wikipedia.org/wiki/Enigma_machine



" … rotor machines used to generate ciphers for the encryption and decryption of secret messages. The first Enigma was invented by German engineer Arthur Scherbius at the end of World War I."

HPSCC

- High Performance Scientific Computing Center (HPSCC)

  A Johns Hopkins Service Center under the Department of Biostatistics that operates the computing cluster on a fee-for-service basis

- Mission

  To provide a **shared** high-performance computing environment for research and teaching in biostatistics, genetics, computational biology and bioinformatics.

- Administrative team (approx 1.55 FTE)
  - –Fernando Pineda (Director, 20%)
  - –Marvin Newhouse (Manager of Computing Systems, 80%)
  - –Jiong Yang (Systems Engineer, 50%)
  - –Cindy Hockett (Financial Admin, 5%)
  - –BIT committee (Advisory committee)

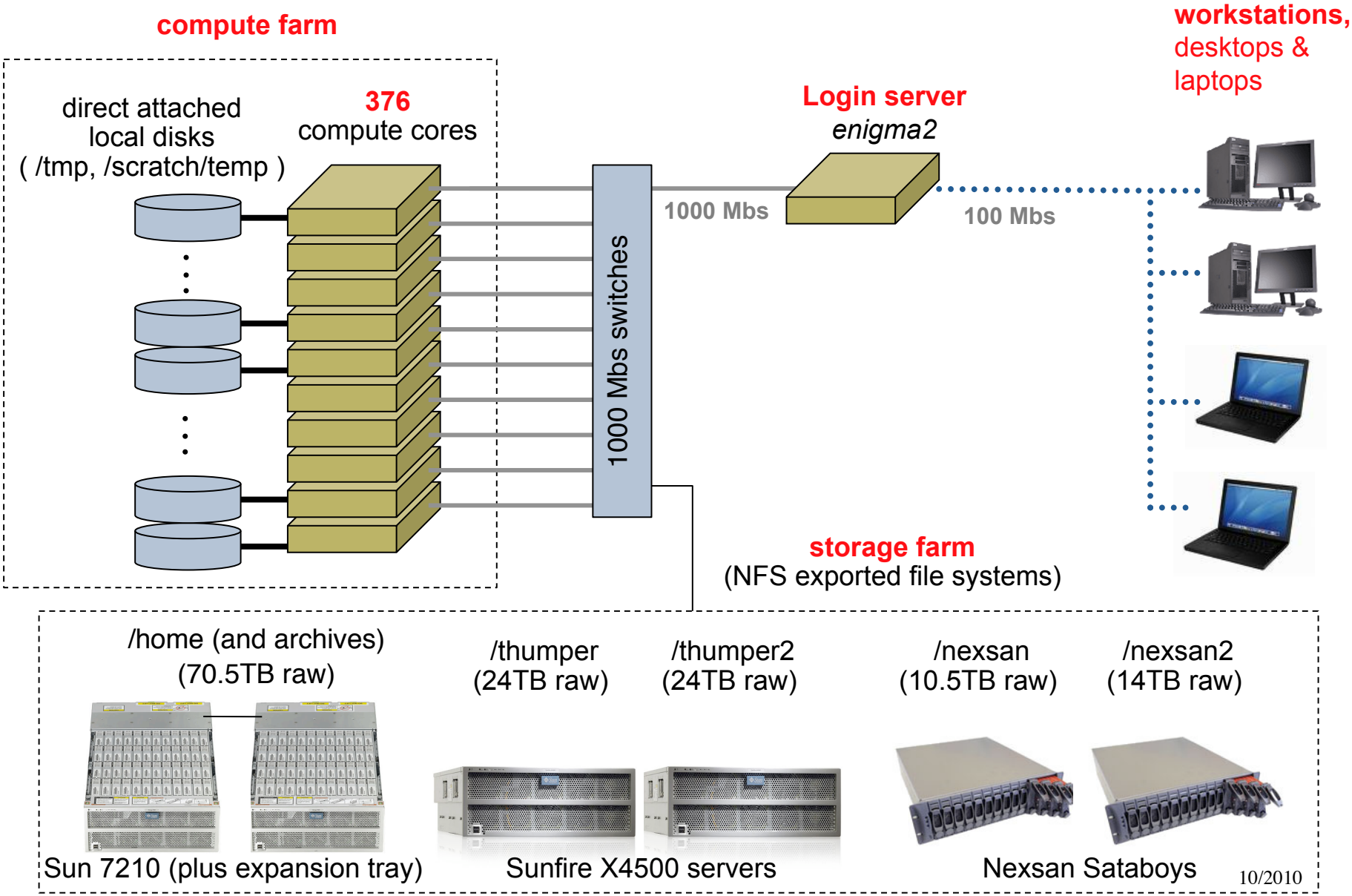
Sharing: back to kindergarten

1. The cluster is a compute farm: i.e. a collection of compute servers that can be shared and accessed through a single “portal” (enigma2).
2. Sun Grid Engine (SGE) is the software that allows you to initiate and/or send jobs to the cluster compute servers (also known as compute hosts or nodes). SGE manages cluster resources and assigns your jobs to appropriate compute server nodes.
3. Originally the servers and hardware was purchased by Biostat, MMI and EPI, however many research groups and departments have contributed resources and everyone now shares the resources.

Getting help

- on-line support for HPSCC systems and biostat-specific
  - http://www.biostat.jhsph.edu/bit
    - *Web site with tutorials and instructions, primarily with a Biostat bent, but also has tutorials on using the cluster*
  - bitsupport@jhsph.edu
    - *System support for the cluster*
    - *Monitored by faculty and staff with systems maintenance responsibilities and/or connections to the BIT committee*
    - *Please use bitsupport rather than contacting Jiong and Marvin directly.*
  - bithelp@jhsph.edu
    - *Application support for R and Biostat applications*
    - *Monitored by current and previous faculty/staff/students*
  - http://hpscc.jhsph.edu
    - *The HPSCC web site, still under development, but has material for proposal preparation*

HPSCC user-model



HPSCC Cluster
Current Compute Node Characteristics

Quantity	CPU Cores	Memory (GB)	Disk 1 (GB)	Disk 2 (GB)	Comments
4	4 x 2.6–3.0 GHz	32	750	750	Sun/Biostat ('bst' nodes)
5	8 x 2.3GHz	20	250	250	Sun/HPSCC
3	8 x 2.3GHz	32	250	250	Sun/HPSCC
1	8 x 2.3GHz	64	1000	1000	Sun/HPSCC
1	8 x 2.3GHz	64	1000	1000	Sun/EpiGenetics
1	8 x 2.7GHz	64	1000	1000	Sun/Oncology
3	8 x 2.7GHz	64	1000	1000	Sun/Infectious Diseases
1	8 x 2.3GHz	40	250	250	Sun/HPSCC
1	8 x 2.7GHz	40	1000	1000	Sun/ Infectious Diseases
5	8 x 2.7GHz	32	500	500	Sun/CEGS(EpiGenetics)
6	12 x 2.6GHz	32	500	500	Penguin Twin Nodes/ Oncology(MCMC)
10	12 x 2.6GHz	64	500	500	Penguin Twin Nodes/ CEGS(EpiGenetics)

Why use a compute cluster?

1. Share high value resources or new technology among a large community of users …

2. Resources on other available computers may be limited:
   - *limited number of cpus*
     - user may want to run many job processes at once
   - *limited amount of memory*
     - user might need lots of memory per process (8GB? 16GB? 32GB? 64GB?)
   - *limited availability*
     - jobs may need to run for many hours/days/weeks

Server characteristics and functions

- *Enigma2* (AMD Opteron, 8-cpu cores, 16 GB)
  - login host (remote login from outside school allowed)
    Use "Good" password practices - do not share your login password
  - interactive work, program development
  - SGE submit host for the cluster
- *Cluster software server* (AMD Opteron, 8-cpu cores, 8 GB)
  - Provides software (apps) over the network for the cluster nodes (*you never see it*)
- *Cluster head node* (AMD Opteron, 4-cpu cores, 8 GB)
  - Manages/monitors cluster jobs (*you never see it*)
- *compute-0-NN* (AMD Opteron, [see earlier slide])
  - Cluster compute nodes (SGE execution hosts)
  - Some cluster nodes can also be SGE submit hosts
- *Post ** (Intel Xeon, 4-cpu cores, 8 GB)
  - * Dedicated access for a specific study-group

System Software

Red Hat Enterprise Linux 5

Login server (and each compute node) has its own instance of RHEL
(login server may not be at the same OS revision level as compute nodes)
See "The Linux Documentation Project" http://tldp.org/ or
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/index.html/

Solaris with ZFS/NFS (Network File System)

Sun 7210 (Amber1) NFS server with multi-gigabit network connectivity
exports /home file systems to login server and compute nodes

Other NFS exports

NFS on Thumper(s) exports mostly static project data to login server and cluster
NFS on Nexsan host servers exports static project data to login server and cluster

Rocks 5.2.2

Cluster build and maintenance tool

Sun Grid Engine 6.2u2

Used for job submission to cluster
Provides cluster resource and load management
Cluster accounting

File systems

- ZFS - accessible from Enigma and all cluster compute nodes (via NFS - see below)
  - Partitions (ZFS shares) under ***/home*** are on the Sun 7210
    - ***/home/bst*** biostat partitions
    - ***/home/mmi*** MMI partition
    - ***/home/epi*** EPI partition
    - ***/home/bstsas*** Dedicated study-group partition
    - *More to come*
  - Partitions under **/thumper** and **/thumper2** on the Thumpers

- Linux ext3 - accessible from Enigma and all cluster compute nodes (via NFS - see below)
  - Partitions under ***/nexsan*** and ***/nexsan2*** are on the NEXSAN and NEXSAN2

- Local Linux ext3 file system
  - All servers, including compute nodes, have a local file system which has **/tmp** and **/scratch/temp** directories for temporary local storage.
    Use this for temporary (scratch) mass storage. See discussion of file staging for I/O intensive jobs and SGE's $TMPDIR on a later slide.

- NFS (network file system)
  - Not actually a separate file system, but a client/server system we use for exporting file systems to enigma2 and the cluster nodes.

Using the cluster

http://biostat.jhsph.edu/bit/cluster-usage.html

All computing on the cluster is done by logging into enigma and submitting batch or interactive jobs via **Sun Grid Engine (SGE).**

- Job submission
  - **qsub** submit batch job
  - **qrsh** establish interactive session or submit interactive job
- Job management
  - **qdel** delete a job from a queue
  - **qhold** hold back a pending submitted job from execution
  - **qrls** release held job for execution
  - **qalter** modify a pending batch job

Cluster information display

- **qstat** displays status listings
- **qhost** displays information about execution hosts
- **qconf** displays information about cluster and queue configuration

A first look at job submission

- create a shell script

```
vi script1.sh            # create a shell script
```

- The shell script

```
ps -ef           # see what's running (on the cluster node)
sleep 10         # sleep for 10 seconds
echo -e "\n --------- Done"
```

- submit shell script with a given job name (note use of -N option)

```
qsub -N mytest script1.sh
```

- check the status of your job (note use of -u option)

```
qstat -u manfred
```

- The output (stdout and stderr) goes to files in your home directory

```
ls mytest.o* mytest.e*
```

BEWARE of text files made on Windows

Windows text files may have (probably do have) CR-LF (carriage return, line feed) line terminators and Linux/Unix shells won't correctly parse such lines.

A simple solution:

On *enigma* run

```
dos2unix script1.sh
```

where **`script1.sh`** is the suspect file.

See

http://www.biostat.jhsph.edu/bit/cluster-troubleshooting.html

A few options

- run the job in the current working directory (where qsub was executed) rather than the default (home directory)

  –cwd

- send standard output (error) stream to a different file

  -o *path/filename*

  -e *path/filename*

- merge the standard error stream into the standard output

  -j y

“How many jobs can I submit?”

As many as you want … BUT only **maxujobs** will run. The rest will have the queue wait state **‘qw’** and will run as your other jobs finish.

The value of **maxujobs** may change depending on the availability of cluster resources.

```
[enigma]$ qconf -ssconf | grep maxujobs
maxujobs                              16
```

or simply …

```
[enigma]$ qmax
maxujobs                              16
```

“Background and/or multi-threaded processes?”

We have SGE configured to assume that EACH job or session, unless otherwise specified, uses 1 cluster job slot and no more than 100% of one CPU core.

Do not spawn background process (even if you are familiar with Linux and you know how to do this)!

Do not run a process in the background and then log off of an interactive session that you started with qrsh!

Submit batch jobs via qsub to accomplish the same end.

Requesting multiple slots or parallel environments for an SGE job/session is a topic for a future lecture
(or contact Marvin for advice on how to proceed).

qstat

- full dump of queue and job status

  `qstat -f`

- info for a given user

  `qstat -u` *username* (*or* **`qu`**)

- What do the column labels mean?

  `job-ID` a unique identifier for the job

  `name` the name of the job

  `state` the state of the job

  r: running

  s: suspended

  t: being transferred to an execution host

  qw: queued and waiting to run

  Eqw: an error occured with the job

- Why is my job in Eqw state?

  `qstat -j` ***`job-ID`*** **`-explain E`**

qstat -f

```
[enigma]$ qstat -f
queuename                      qtype used/tot. load_avg arch          states
---------------------------------------------------------------------------------
. . .
standard.q@compute-0-0.local   BIP   1/2       1.00     lx26-amd64
  94752 2.93288 simu_V2_24 john         r     11/17/2008 11:15:06     1
---------------------------------------------------------------------------------
standard.q@compute-0-1.local   BIP   0/2       0.84     lx26-amd64
---------------------------------------------------------------------------------
standard.q@compute-0-13.local  BIP   1/2       1.00     lx26-amd64
  94740 2.93354 simu_V2_18 john         r     11/17/2008 03:16:44     1
---------------------------------------------------------------------------------
standard.q@compute-0-14.local  BIP   2/2       2.00     lx26-amd64
  94739 2.93354 simu_V2_17 john         r     11/17/2008 03:16:44     1
  94819 2.87651 dR-2reps3. mary         r     11/17/2008 13:51:59     1
---------------------------------------------------------------------------------
standard.q@compute-0-15.local  BIP   2/2       0.97     lx26-amd64
  94369 10.45000 QRLOGIN   martha       r     11/14/2008 07:57:09     1
  94893 2.89601 simu_V2_2. john         r     11/17/2008 18:51:08     1
---------------------------------------------------------------------------------
standard.q@compute-0-16.local  BIP   1/2       1.00     lx26-amd64
  94849 3.04957 boot113.tx harry        r     11/17/2008 16:25:05     1
---------------------------------------------------------------------------------
standard.q@compute-0-17.local  BIP   1/2       1.00     lx26-amd64
  94570 2.99906 PEF-exp-n1 john         r     11/16/2008 00:06:11     1
---------------------------------------------------------------------------------
standard.q@compute-0-18.local  BIP   1/2       1.01     lx26-amd64
  94791 3.06098 estresampl harry        r     11/17/2008 11:41:11     1
---------------------------------------------------------------------------------
standard.q@compute-0-19.local  BIP   1/2       1.00     lx26-amd64
  94790 3.06098 estresampl harry        r     11/17/2008 11:41:11     1
---------------------------------------------------------------------------------
. . .
```

qstat

```
[enigma]$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
  94752 2.88228 simu_V2_24 john         r     11/17/2008 11:15:06 standard.q@compute-0-0.local       1
  94736 2.88294 simu_V2_14 john         r     11/17/2008 03:16:44 standard.q@compute-0-1.local       1
  94740 2.88294 simu_V2_18 john         r     11/17/2008 03:16:44 standard.q@compute-0-13.local      1
  94739 2.88294 simu_V2_17 john         r     11/17/2008 03:16:44 standard.q@compute-0-14.local      1
  94819 2.84273 dR-2reps3. mary         r     11/17/2008 13:51:59 standard.q@compute-0-14.local      1
  94369 10.45000 QRLOGIN   martha       r     11/14/2008 07:57:09 standard.q@compute-0-15.local      1
  94893 2.84540 simu_V2_2. john         r     11/17/2008 18:51:08 standard.q@compute-0-15.local      1
  94849 3.01641 boot113.tx harry        r     11/17/2008 16:25:05 standard.q@compute-0-16.local      1
  94570 2.94845 PEF-exp-n1 john         r     11/16/2008 00:06:11 standard.q@compute-0-17.local      1
  94791 3.02782 estresampl harry        r     11/17/2008 11:41:11 standard.q@compute-0-18.local      1
  94790 3.02782 estresampl harry        r     11/17/2008 11:41:11 standard.q@compute-0-19.local      1
  94569 2.94846 PEF-exp-n5 john         r     11/16/2008 00:06:11 standard.q@compute-0-2.local       1
  94889 3.01325 nboot11.tx harry        r     11/17/2008 17:43:52 standard.q@compute-0-20.local      1
  94898 3.24159 QRLOGIN    peter        r     11/17/2008 18:54:44 standard.q@compute-0-21.local      1
  94792 3.02782 estresampl harry        r     11/17/2008 11:41:11 standard.q@compute-0-22.local      1
  94795 3.02734 estresampl harry        r     11/17/2008 11:53:03 standard.q@compute-0-23.local      1
  94809 2.84274 cond1-3rep mary         r     11/17/2008 13:51:59 standard.q@compute-0-26.local      1
  94628 2.90735 test2.sh   mary         r     11/16/2008 11:03:50 standard.q@compute-0-27.local      1
  94629 2.90735 test3.sh   mary         r     11/16/2008 11:03:50 standard.q@compute-0-28.local      1
  94784 2.85268 sim5.sh    mary         r     11/17/2008 09:44:33 standard.q@compute-0-29.local      1
  94737 2.88294 simu_V2_15 john         r     11/17/2008 03:16:44 standard.q@compute-0-3.local       1
  94420 8.38095 get_probes sarah        r     11/14/2008 13:43:37 standard.q@compute-0-30.local      1
  94817 2.84273 dR-2reps1. mary         r     11/17/2008 13:51:59 standard.q@compute-0-30.local      1
```

•••

New SGE defult for qstat shows only your jobs. You can use `qstat –u \*` *to see all users jobs.*

qdel

To terminate a job, first get the job-id with qstat

`qstat -u` *username* *(or* **`qu`** *)*

- Terminate the job

  `qdel` ***job-id***

- Forced termination of a running job (admins only)

  `qdel -f` ***job-id***

Requesting particular resources

- SGE performs resource allocation and load balancing
- Request resources by specifying options in SGE submission commands

- How to request particular resources or queues
  - –Specify a resource constraint/requirement list with `-l` option (GOOD!)

```
qsub -l mem_free=3G,h_vmem=5G foo_job.sh
qsub -l h_rt=0:30:0 foo_job.sh
qsub -l sas sas_batchfile
```

  - –Specify a particular queue instance (usually BAD)

```
qsub -q standard.q@compute-0-30.local foo_job.sh
```

Rule of thumb:

Do not specify queues yourself.
Use resource constraints and let SGE balance the load!

Requesting particular resources

- In general qsub expects a *batch script*
  - lines that start with **#** are shell comments
  - lines that start with **#$** are interpreted by **qsub**

- Example script (**script2.sh**)

```
# Following are lines containing qsub options, one per line
# exactly as you would have typed them on the
# qsub command line, e.g.
#$ -l h_rt=0:30:0
#$ -l mem_free=3.0G
#$ -l h_vmem=5G
# Finally, the body of the script, i.e. the part
# that is executed by the shell,
ps -ef
sleep 10
```

- Running the script

```
qsub script2.sh
```

Running memory intensive jobs

- Resources are limited and shared

Ask for the amount of memory that you expect your job will need so that it runs on a cluster node with sufficient memory available
… AND …
place a limit on the amount of memory your job can use so that you don't accidentally crash a compute node and take down other users' jobs along with yours:

```
qsub -l mem_free=6.5G,h_vmem=10G  myscript.sh

qrsh -l mem_free=8G,h_vmem=12G  R
```

See http://www.biostat.jhsph.edu/bit/cluster-usage.html#LargeMem for more details.

- Specify your resource requirements

Don't assume that any node will do or that a 64G node will always have enough RAM.

Interactive jobs

`qrsh` establishes a remote shell connection (ssh) on an unspecified node.

- Examples:

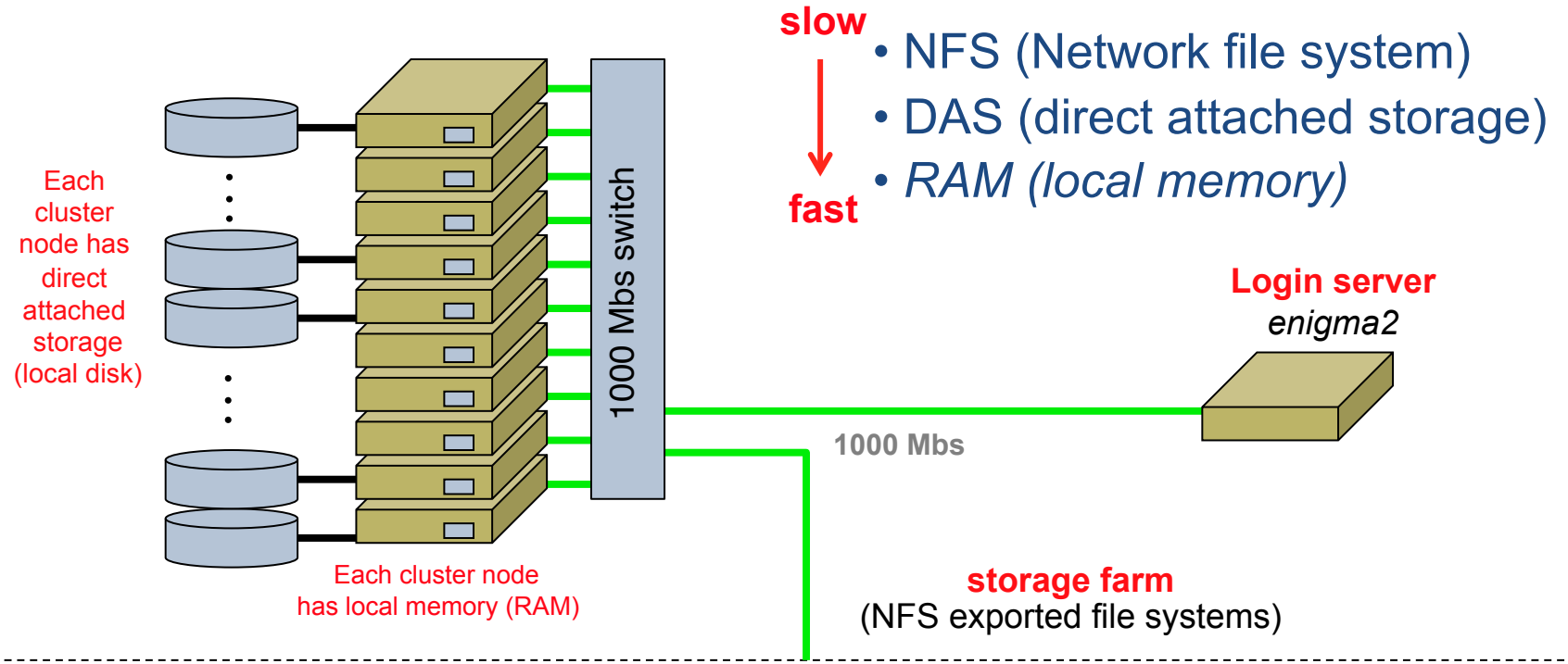
```
qrsh        # open remote shell
qrsh R      # open R in remote shell
qrsh vi     # open vi in remote shell
```

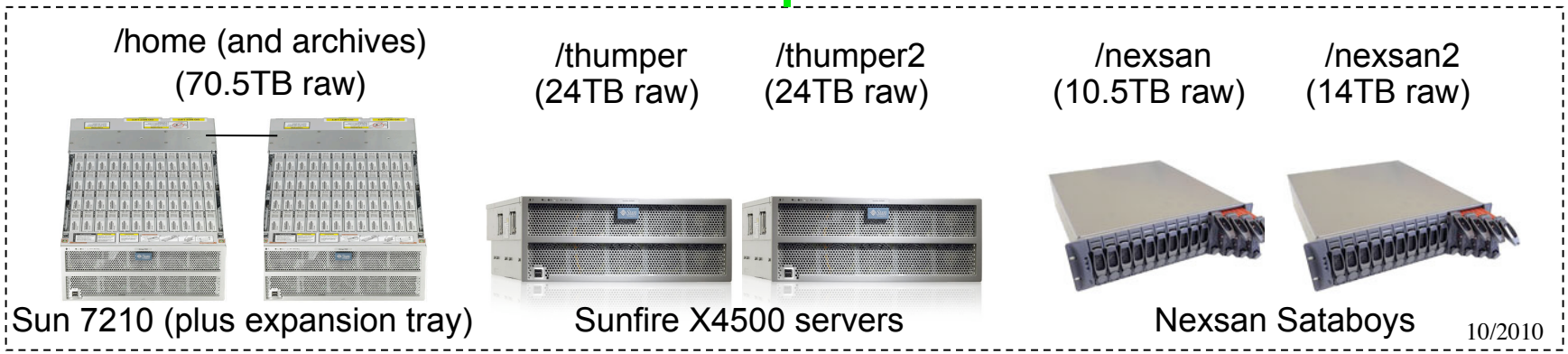
- Establish an ssh connection and run executable with given resource constraints:

```
qrsh -l mem_free=3.0G R
qrsh -l mem_free=5G R CMD BATCH my.R
```

Running I/O intensive jobs

Understanding bandwidth constraints





Other than scratch disk space on a given compute node, ALL file input/output (`I/O`) is over a network connection using NFS.

Use file staging for `I/O` intensive jobs

- Performing sequential `I/O` to your home or data directories over NFS can be very very slow.

- Break the bottleneck with the "staging" technique
  - For each job executed, SGE creates a local temporary directory under **`/tmp`** and exports the path name in the $TMPDIR environment variable.
  - start processing by copying input files into $TMPDIR.
  - perform all your `I/O` from/to files in $TMPDIR
  - end processing by copying output files out of $TMPDIR into a permanent location (under your home directory)
  - when the job exits, $TMPDIR, and anything in it, is deleted by SGE.

script3.sh

```
#!/bin/bash
#####################
# SGE options & parameters
#####################
# (1) the name of the job
#$ -N SGE_DEMO
# (2) resource requirements
#$ -l h_rt=0:30:0
#$ -l mem_free=1.0G
# (3) output files
#$ -cwd
#$ -o demo.out
#$ -e demo.err
# (4) It's a big job, so set priority so low
that
#     everyone else can get in ahead of me
#$ -p -1023
#$ (5) E-mail me when it's finished
#$ -m e
#$ -M john@jhu.edu
#####################
# stage the input data
#####################
# (1) copy 628MB of compressed data
mkdir $TMPDIR/data
cp ~/data/test.tar.Z $TMPDIR/data/.
cd $TMPDIR/data
ls -lh test.tar.Z

# (2) decompress and untar the data
tar -xzvf test.tar.Z
du -ch $TMPDIR/data | grep total

#####################
# Do the processing
#####################
#
# (1) in this case just move the data
#     to a temporary output directory
#     and tar it again

mkdir $TMPDIR/output
mv $TMPDIR/data $TMPDIR/output
cd $TMPDIR/output
tar -czf results.tar.Z data

#####################
# save the results of the computation
# and quit
#####################

cp results.tar.Z $HOME/.
```

SGE history

- DQS was developed from 1991- 2000 at Florida State University by the Supercomputer Computations Research Institute under contract from the Department of Energy.
- Gridware, inc made a commercial product out of DQS and called it Codine.
- Sun acquired Codine in 2000, renamed it Sun Grid Engine (aka “N1 Grid Engine”) and released it under an open source license.
- Open source version now simply know as Grid Engine.
- Commercial version and open source are now based on the same code base (~1 million lines of code)
- Over 10,000 deployments
- The current version of SGE is 6.2
- Download docs from

  `http://gridengine.sunsource.net/documentation.html`

A little SGE terminology

- **Hosts**
  - Physical servers that run SGE processes (clients or servers)
- **Queues**
  - An SGE queue is a class of jobs allowed to execute on a particular host (or set of hosts) concurrently.
- **Complexes**
  - The set of attributes associated with a queue, a host, or an entire cluster is called a complex

Hosts

- **Submit host**: the server(s) where you submit jobs and execute SGE client commands (enigma)

- **Execution hosts**: the machines where jobs actually run (cluster nodes)

- `qconf -sel` `#shows list of execution host names`

- `qhost` `#shows status of execution hosts`

- `qhostw` `#summarizes load and memory usage of execution hosts by memory capacity (local mod. of qhost)`

qhost

```
[enigm]$ qhost
HOSTNAME                ARCH         NCPU  LOAD  MEMTOT  MEMUSE  SWAPTO  SWAPUS
----------------------------------------------------------------------------------
global                  -               -     -       -       -       -       -
compute-0-0             lx26-amd64      2  1.04    2.0G   84.1M    2.0G   14.5M
compute-0-1             lx26-amd64      2  1.05    2.0G  106.2M    2.0G     0.0
compute-0-14            lx26-amd64      2  2.00    3.9G  758.7M    3.9G   10.5M
compute-0-15            lx26-amd64      2  0.96    3.9G  719.2M    3.9G     0.0
compute-0-16            lx26-amd64      2  1.04    3.9G   97.9M    3.9G     0.0
compute-0-17            lx26-amd64      2  1.03    3.9G  148.5M    3.9G     0.0
compute-0-18            lx26-amd64      2  1.05    3.9G   97.7M    3.9G     0.0
compute-0-19            lx26-amd64      2  1.03    3.9G  122.6M    3.9G     0.0
compute-0-2             lx26-amd64      2  1.29    2.0G  166.0M    2.0G     0.0
compute-0-20            lx26-amd64      2  1.00    7.7G  104.5M    7.8G     0.0
compute-0-21            lx26-amd64      2  0.09    7.7G  530.3M    7.8G     0.0
compute-0-22            lx26-amd64      2  1.04    7.7G  102.0M    7.8G     0.0
compute-0-23            lx26-amd64      2  1.03    7.7G  106.0M    7.8G  688.0K
compute-0-24            lx26-amd64      2     -   15.6G       -   15.6G       -
compute-0-25            lx26-amd64      2  0.30   15.6G   83.1M   15.6G  144.0K
compute-0-26            lx26-amd64      2  1.00    7.7G  240.0M    7.8G     0.0
compute-0-27            lx26-amd64      2  1.02    7.7G  678.4M    7.8G     0.0
compute-0-28            lx26-amd64      2  1.02    7.7G  727.4M    7.8G  548.0K
compute-0-29            lx26-amd64      2  1.02    7.7G  615.7M    7.8G     0.0
compute-0-3             lx26-amd64      2  1.05    2.0G  128.1M    2.0G     0.0
compute-0-30            lx26-amd64      2  2.00   15.6G    9.6G   15.6G     0.0
compute-0-31            lx26-amd64      2  1.01   15.6G  465.5M   15.6G     0.0
compute-0-32            lx26-amd64      2  1.01    3.9G  764.2M    3.9G     0.0
compute-0-33            lx26-amd64      2  1.04    2.0G  129.0M    2.0G     0.0
compute-0-35            lx26-amd64      4  0.00   31.4G    4.0G   31.2G   18.2M
compute-0-36            lx26-amd64      4  0.00   31.4G    9.7G   31.2G     0.0
compute-0-37            lx26-amd64      4  0.00   31.4G  124.5M   31.2G   12.6M
compute-0-38            lx26-amd64      4  0.00   31.4G   81.2M   31.2G   10.3M
compute-0-39            lx26-amd64      8  1.99   31.4G  320.9M   31.2G   10.1M
compute-0-4             lx26-amd64      2  0.07    2.0G   62.1M    2.0G  144.0K
compute-0-40            lx26-amd64      8  4.54   31.4G    6.2G   31.2G   28.0M
compute-0-41            lx26-amd64      8  3.00   31.4G  968.5M   31.2G     0.0
compute-0-42            lx26-amd64      8  2.97   11.7G    1.0G   11.7G     0.0
compute-0-43            lx26-amd64      8  3.27   11.7G    9.0G   11.7G     0.0
compute-0-44            lx26-amd64      8  3.02   11.7G 1006.4M   11.7G  208.0K
compute-0-45            lx26-amd64      8  3.97   11.7G 1000.7M   11.7G  208.0K
compute-0-46            lx26-amd64      8  3.02   11.7G    3.3G   11.7G     0.0
compute-0-47            lx26-amd64      8  2.16   62.9G   62.8G   61.1G   19.6G
compute-0-5             lx26-amd64      2  0.44    2.0G   46.0M    2.0G   13.7M
compute-0-6             lx26-amd64      2  1.00    2.0G   92.1M    2.0G   44.0K
compute-0-7             lx26-amd64      2  1.00    2.0G  126.4M    2.0G     0.0
compute-0-8             lx26-amd64      2  0.96    2.0G  132.6M    2.0G     0.0
compute-0-9             lx26-amd64      2     -    2.0G       -    2.0G       -
```

qhostw

```
[enigma]$ qhostw
               LOAD  MEMTOT  MEMUSE    SWAPTOT  SWAPUSE
 2GB nodes -----
  compute-0-0  1.15    2.0G   84.2M       2.0G   14.5M
  compute-0-1  1.19    2.0G  106.3M       2.0G     0.0
  compute-0-2  1.87    2.0G  166.0M       2.0G     0.0
  compute-0-3  1.19    2.0G  128.1M       2.0G     0.0
  compute-0-4  0.23    2.0G   62.0M       2.0G  144.0K
  compute-0-5  1.14    2.0G  738.2M       2.0G   15.8M
  compute-0-6  1.03    2.0G   92.1M       2.0G   44.0K
  compute-0-7  1.00    2.0G  126.5M       2.0G     0.0
  compute-0-8  0.91    2.0G  132.6M       2.0G     0.0
 compute-0-13  1.00    2.0G  105.7M       2.0G     0.0
 compute-0-33  1.16    2.0G  128.7M       2.0G     0.0
 4GB nodes -----
 compute-0-14  2.00    3.9G  758.8M       3.9G   10.5M
 compute-0-15  0.92    3.9G  719.1M       3.9G     0.0
 compute-0-16  1.11    3.9G   97.8M       3.9G     0.0
 compute-0-17  1.12    3.9G  148.7M       3.9G     0.0
 compute-0-18  1.12    3.9G   97.8M       3.9G     0.0
 compute-0-19  1.11    3.9G  122.6M       3.9G     0.0
 compute-0-32  1.00    3.9G  740.6M       3.9G     0.0
 8GB nodes -----
 compute-0-20  1.00    7.7G  104.6M       7.8G     0.0
 compute-0-21  0.26    7.7G  530.4M       7.8G     0.0
 compute-0-22  1.11    7.7G  102.0M       7.8G     0.0
 compute-0-23  1.10    7.7G  106.0M       7.8G  688.0K
 compute-0-26  1.00    7.7G  239.9M       7.8G     0.0
 compute-0-27  1.07    7.7G  689.7M       7.8G     0.0
 compute-0-28  1.06    7.7G  727.4M       7.8G  548.0K
 compute-0-29  1.07    7.7G  538.0M       7.8G     0.0
 12GB nodes -----
 compute-0-42  2.97   11.7G    1.0G      11.7G     0.0
 compute-0-43  3.52   11.7G    9.0G      11.7G     0.0
 compute-0-44  3.05   11.7G  969.1M      11.7G  208.0K
 compute-0-45  3.96   11.7G 1000.2M      11.7G  208.0K
 compute-0-46  3.09   11.7G    2.0G      11.7G     0.0
 16GB nodes -----
 compute-0-25  0.90   15.6G   83.1M      15.6G  144.0K
 compute-0-30  2.00   15.6G    9.6G      15.6G     0.0
 compute-0-31  1.00   15.6G  465.5M      15.6G     0.0
 32GB nodes -----
 compute-0-39  1.99   31.4G  321.3M      31.2G   10.1M
 compute-0-40  4.10   31.4G    6.0G      31.2G   28.0M
 compute-0-41  3.04   31.4G  983.6M      31.2G     0.0
 64GB nodes -----
 compute-0-47  2.19   62.9G   44.3G      61.1G   18.9G

 32GB nodes ----- 'FAT nodes' (for special projects)
 compute-0-35  0.00   31.4G    4.0G      31.2G   18.2M
 compute-0-36  0.00   31.4G    9.7G      31.2G     0.0
 compute-0-37  0.00   31.4G  124.7M      31.2G   12.6M
 compute-0-38  0.00   31.4G   81.7M      31.2G   10.3M

[enigma]$
```

Queues

Adapted from Sun Microsystems' N1 Grid Engine 6 User's Guide
http://docs.sun.com/app/docs/doc/817-6117?q=N1GE

1. A queue is a container for a class of jobs that are allowed to run on one or more hosts concurrently. A queue determines certain job attributes, for example, how long (wall clock time) the job is allowed to run. Throughout its lifetime, a running job is associated with its queue. Association with a queue affects some of the things that can happen to a job. For example, if a queue is suspended, all jobs associated with that queue are also suspended.

2. Jobs need not be submitted directly to a queue. You need to specify only the requirement profile of the job. A profile might include requirements such as memory, operating system, available software, and so forth. The grid engine software automatically dispatches the job to a suitable queue and a suitable host with a light execution load. If you submit a job to a specified queue, the job is bound to this queue. As a result, the grid engine system daemons are unable to select a lighter-loaded or better-suited device.

3. A queue can reside on a single host, or a queue can extend across multiple hosts. For this reason, grid engine system queues are also referred to as cluster queues. Cluster queues enable users and administrators to work with a cluster of execution hosts by means of a single queue configuration. Each host that is attached to a cluster queue receives its own queue instance from the cluster queue.

Queues - 1

A queue is a container for a class of jobs that are allowed to run on one or more hosts concurrently. A queue determines certain job attributes, for example, how long (wall clock time) the job is allowed to run. Throughout its lifetime, a running job is associated with its queue.

```
[enigma]$ qconf -sql   # shows queue list
express.q
fat_hi.q
sas.q
special.q
standard.q
[enigma]$
```

Association with a queue affects some of the things that can happen to a job. For example, if a queue is suspended, all jobs associated with that queue are also suspended.

Adapted from Sun Microsystems' N1 Grid Engine 6 User's Guide

Queues - 2

Jobs need not be submitted directly to a queue. ***You need to specify only the requirement profile of the job.*** A profile might include requirements such as memory, operating system, available software, and so forth. The grid engine software automatically dispatches the job to a suitable queue and a suitable host with a light execution load.

```
#submit a job to run on any node (host) with at least 9G memory available
[enigma]$ qsub -l mem_free=9G large_memjob.sh

#open a remote shell on any express queue node (host)
[enigma]$ qrsh -l express
```

If, however, you submit a job to a *specific* queue (or host), the job is bound to that queue. As a result, the grid engine system daemons are unable to select a lighter-loaded or better-suited device.

Adapted from Sun Microsystems' N1 Grid Engine 6 User's Guide

Queues - 3

A queue can reside on a single host, or a queue can extend across multiple hosts. For this reason, grid engine system queues are also referred to as cluster queues.

Cluster queues enable users and administrators to work with a cluster of execution hosts by means of a single queue configuration.

Each host that is attached to a cluster queue receives its own *queue instance* from the cluster queue.

Adapted from Sun Microsystems' N1 Grid Engine 6 User's Guide

Parting shots

- Improved SGE scheduling coming soon
- Link to this lecture posted on bit page
- Also see *http://www.biostat.jhsph.edu/bit/cluster-usage.html*
- queue names may change, so specify resource constraints rather than specific queues.
- Be a good citizen
  - *Don't run jobs on the cluster other than through SGE*
  - *Do large* I/O *jobs using file staging*
  - *Don't hog the large memory machines*
- Talk to us:
  - *Use bithelp@jhsph.edu for application support*
  - *Use bitsupport@jhsph.edu for system support*

*Composite Theoretical Performance (CTP) calculations ("Calculations") for AMD Opteron microprocessors. Official calculations stated by AMD are in Millions of Theoretical Operations Per Second (MTOPS) and are based upon a formula in the United States Department of Commerce Export Administration Regulations 15 CFR 774 (Advisory Note 4 for Category 4).

HPSCC physical network topology