Clusters in the Faculty of Mathematics and Computer Science

In the Faculty we have two clusters for general usage:

- Mcluster01 with 39 nodes (machines running under a master), and
- Mcluster03 with 25 nodes.

Mcluster01

In order to connect to the server to run a job/ jobs you can

```
rsh mcluster01
```

or

```
ssh mcluster01
```

Once logged into the master machine then you can run your script, as so:

```
qsub -q amd32g.q  -l mem_total=20GB ./myscript.sh
```

This will run a batch job to queue amd32g.q with and will reserve a total of 20GB of memory.

The queues of mcluster01.

Under mcluster01 there are 3 queues, as specified here:

all.q - This is a general queue which includes all the nodes in the cluster.
amd32g.q - This queue is a queue with AMD cpu nodes and a total of 32GB memory.

There are 14 nodes under this queue.
amd64g.q - This queue contains nodes with AMD cpu's and a total of 64GB memory.
There are 7 nodes under this queue.

Nodes in Mcluster01:

Host Name	CPU Architecture	Number of CPU's	Total Memory
n01	AMD 64Bit	4	7.7G
n02	AMD 64Bit	4	7.7G
n03	AMD 64Bit	4	7.7G
n04	AMD 64Bit	4	7.7G

Host Name	CPU Architecture	Number of CPU's	Total Memory
n05	AMD 64Bit	4	7.7G
n06	AMD 64Bit	4	7.7G
n07	AMD 64Bit	4	7.7G
n08	AMD 64Bit	4	7.7G
n09	AMD 64Bit	4	7.7G
n10	AMD 64Bit	4	7.7G
n11	AMD 64Bit	4	7.7G
n12	AMD 64Bit	4	7.7G
n13	AMD 64Bit	4	7.7G
n14	AMD 64Bit	4	7.7G
n15	AMD 64Bit	4	7.7G
n16	AMD 64Bit	4	7.7G
n17	AMD 64Bit	4	7.7G
n18	AMD 64Bit	4	5.7G
n19	AMD 64Bit	8	63.0G
n20	AMD 64Bit	8	63.0G
n21	AMD 64Bit	8	63.0G
n22	AMD 64Bit	8	63.0G
n23	AMD 64Bit	8	63.0G
n24	AMD 64Bit	8	63.0G
n25	AMD 64Bit	8	63.0G
n26	AMD 64Bit	8	31.4G
n27	AMD 64Bit	8	31.4G
n28	AMD 64Bit	8	31.4G
n29	AMD 64Bit	8	31.4G
n30	AMD 64Bit	8	31.4G
n31	AMD 64Bit	8	31.4G
n32	AMD 64Bit	8	31.4G
n33	AMD 64Bit	8	31.4G
n34	AMD 64Bit	8	31.4G
n35	AMD 64Bit	8	31.4G
n36	AMD 64Bit	8	31.4G
n37	AMD 64Bit	8	31.4G
n38	AMD 64Bit	8	31.4G
n39	AMD 64Bit	8	31.4G

Mcluster03

In order to connect to the server to run a job/ jobs you can

```
rsh mcluster03
```

or

```
ssh mcluster03
```

The queues of mcluster03.

There are 3 queues in mcluster03, as specified here:

all.q	- This is a general queue which includes all the nodes in the cluster.
intel16g.q	- This is a queue with Intel Cpu's and a total of 16GB of memory. In this queue there are 16 nodes.
intel32g.q	- This is a queue with Intel Cpu's and a total of 32GB of memory. In this queue there are 6 nodes.

Nodes in Mcluster03:

Host Name	CPU Architecture	Number of CPU's	Total Memory
nm01	Intel Xeon	8	15.7G
nm02	Intel Xeon	8	15.7G
nm03	Intel Xeon	8	15.7G
nm04	Intel Xeon	8	15.7G
nm05	Intel Xeon	8	15.7G
nm06	Intel Xeon	8	15.7G
nm07	Intel Xeon	8	15.7G
nm08	Intel Xeon	8	15.7G
nm09	Intel Xeon	8	15.7G
nm10	Intel Xeon	8	15.7G
nm11	Intel Xeon	8	15.7G
nm12	Intel Xeon	8	15.7G
nm13	Intel Xeon	8	15.7G
nm14	Intel Xeon	8	15.7G
nm15	Intel Xeon	8	15.7G
nm16	Intel Xeon	8	15.7G
nm17	AMD 64Bit	4	23.5G
nm18	AMD 64Bit	4	23.5G
nm19	AMD 64Bit	4	23.5G
nm20	Intel Xeon	8	31.4G
nm21	Intel Xeon	8	31.4G
nm22	Intel Xeon	8	31.4G
nm23	Intel Xeon	8	31.4G
nm24	Intel Xeon	8	31.4G
nm25	Intel Xeon	8	31.4G

About the SGE queuing system.

The Sun Grid Engine (SGE) is a software package to facilitate "grid" computing. The Math SGE cluster is a set of systems provided for doing large computations.

Basic Commands:

The main submit commands are qsub, qrsh and qtcsh. See the man pages for submit(1) and qtcsh(1) for more details.

Showing Information

qstat - Show job/queue status

no arguments - Show currently running/pending jobs

- -f Show full listing of all queues.
- -j Shows detailed information on pending/running job.
- -u Shows current jobs by user.

qhost - Show job/host status

no arguments - Show a table of all execution hosts and information about their configuration

- -l attr=val Show only certain hosts.
- -j Shows detailed information on pending/running job.
- -q Shows detailed information on queues at each host.

Submitting Jobs:

qsub - submit scripts

Started with no arguments it accepts input from STDIN (^D to send submit input)

- -cwd Run the job from the current working directory (Default: Home directory).
- -v Pass the variable VAR (-V passes all variables).
- -o Redirect standard output (Default: Home directory).
- -e Redirect standard error (Default: Home directory).

Example:

```
qsub -cwd -v SOME_VAR -o /dev/null -e /dev/null myjob.sh
```

- **Submitting a job to the queue: qsub**

  Qsub is used to submit a job to SGE. The qsub command has the following syntax:

```
qsub [ options ] [ scriptfile | -- [ script args ]]
```

Binary files may not be submitted directly to SGE. For example, if we wanted to submit the "date" command to SGE we would need a script that looks like:

```
#!/bin/bash
```

```
/bin/date
```

If the script were called sge-date, then we could simply run the following:

```
$ qsub sge-date
```

SGE will then run the program, and place two files in your current directory:

```
$ls
sge-date.e#
sge-date.o#
```

where # is the job number assigned by SGE. The sge-date.e# file contains the output from standard error and the sge-date.o# file contains the output form standard out. The following basic options may be used to submit the job.

-A [account name] -- Specify the account under which to run the job
-N [name] -- The name of the job
-l *h rt=hr:min:sec* -- Maximum walltime for this job
-r *[y,n]* -- Should this job be re-runnable (default y)
-pe *[type] [num]* -- Request [num] amount of [type] nodes.
-cwd -- Place the output files (.e,.o) in the current working directory. The default is to place them in the users home directory.
-S *[shell path]* -- Specify the shell to use when running the job script.

Although it is possible to use command line options and script wrappers to submit jobs, it is usually more convenient to use just a single script to include all options for the job.

In general, qsub is used for traditional batch submit, that is where I/O is directed to a file. Note that qsub only accepts shell scripts, not executable files.

See the qsub(1) man page for more details.

qsh – Open an interactive shell

Will choose the most available computer in the cluster, copy your shell environment to this chosen computer, where you will be able to operate the computer in the interactive mode.

qlogin - will choose the most available computer in the cluster and will

allow access via login.

qrsh – rsh command

Qrsh acts similar to the rsh command, except that a host name is not given. Instead, a shell script or an executable file is run, potentially on any node in the cluster. I/O is directed back to the submitter's terminal window. By default, if the job cannot be run immediately, qrsh will not queue the job. Using the '-now no' flag to qrsh will allow jobs to queue. Note that I/O can be redirected with the shell redirect operators. For example, to run the uname -a command:

qrsh uname -a

The uname of some machine the scheduler selects in the cluster will then be displayed on the submitting terminal. To redirect the output,

qrsh uname -a > /tmp/myfile

The output from uname will be written to /tmp/myfile on the submitting host. To allow the command to queue:

qrsh -now no uname -a

If a suitable host is not immediately available the command will block until a suitable host is available. At that time, the command output will be displayed on the submitting terminal.

See the qrsh(1) man page for more details.

Deleting a Job: qdel

Jobs may be deleted by using the `qdel` command as follows:

```
$ qdel job-id
```

The *job-id* job number is the number assigned by SGE when you submit the job using `qsub`. You can only delete you jobs.

Preserving Your Environment

If you want to make sure your current environment variables are used on you SGE jobs, include the following in your submit script:

```
#$ -V
```

or

```
qsub -V ./job.sh
```

Memory Intense Jobs

If you need to run your job with large memory usage, then it is necessary to specify how much your job will need. For example in the following

```
qsub -l mem_total=6GB ./myscript.sh
```

we are running a job named myscript with a total memory of 6GB.

In case of problems requests and large amounts of disk usage please contact :

Leon Samuel
Email: Leon.Samuel@Weizmann.ac.il
Internal Tel: 08-934- 4304