



ARCHIE-WeSt

**Academic and Research Computer
Hosting Industry and Enterprise**
in the West of Scotland

HPC User Introduction

Dr Richard Martin

HPC Manager, ARCHIE-WeSt

Room JA8.30, x3265

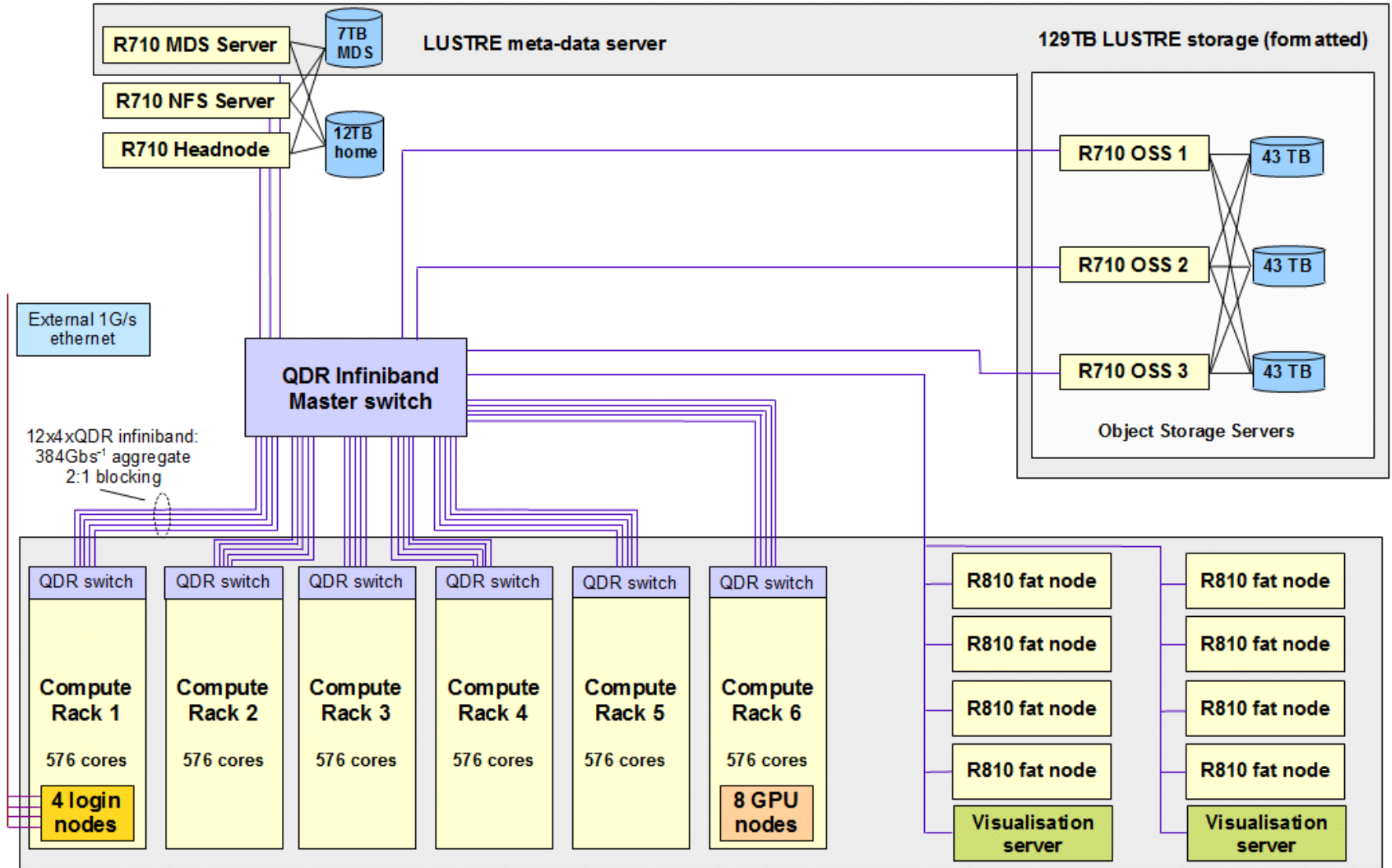
Richard.martin@strath.ac.uk



Technical Specification

- 276 compute nodes
 - Dell C6100
 - 2 Intel Xeon X5650 2.66 GHz CPU (6 cores each)
 - 48 GB RAM
 - 4xQDR Infiniband interconnect
 - 3312 core “main queue”
- 8 DELL R810 SMP nodes (“fat nodes”)
 - 4 Intel Xeon E7-430 2.3GHz eight core CPU’s
 - 512 GB RAM
- 8 Nvidia M2075 GPU nodes
- 2 Dell R5500 visualization nodes
- 150 TB high performance storage (LUSTRE)

QDR infiniband (32 Gbs⁻¹)



Access

ssh - secure **shell**

ssh is a program for logging into a remote machine and executing commands in the remote machine.

ssh cwb08102@archie-w.hpc.strath.ac.uk

log user cwb08102 to ARCHIE-WeSt

ssh -X cwb08102@archie-e.hpc.strath.ac.uk

log user cwb08102 to ARCHIE-WeSt using X terminal (graphic terminal)

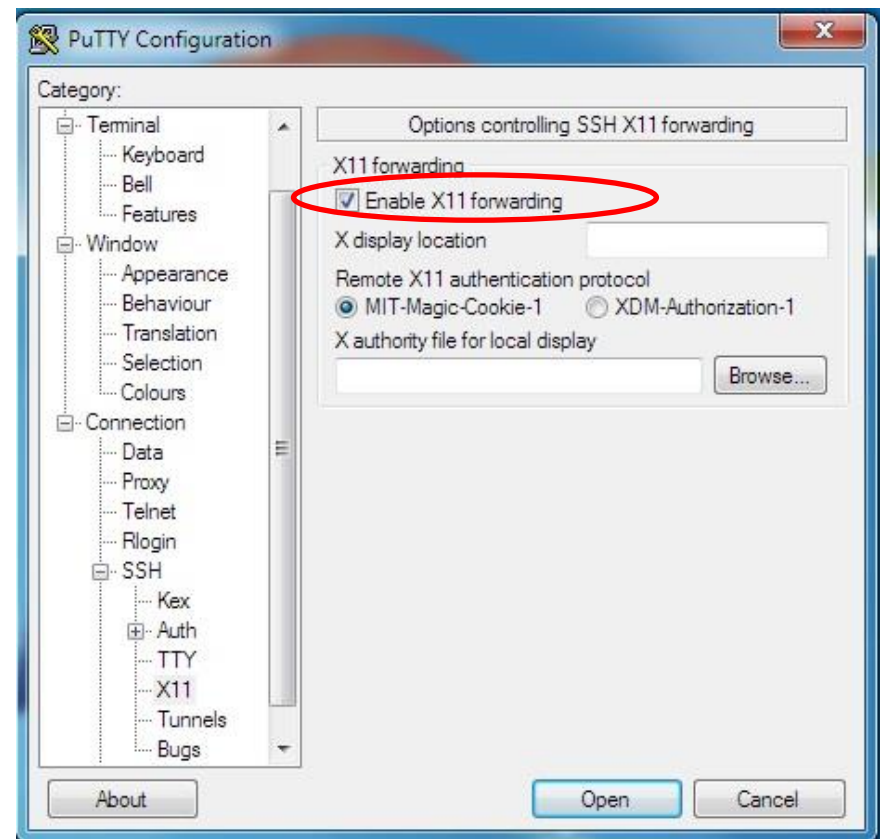
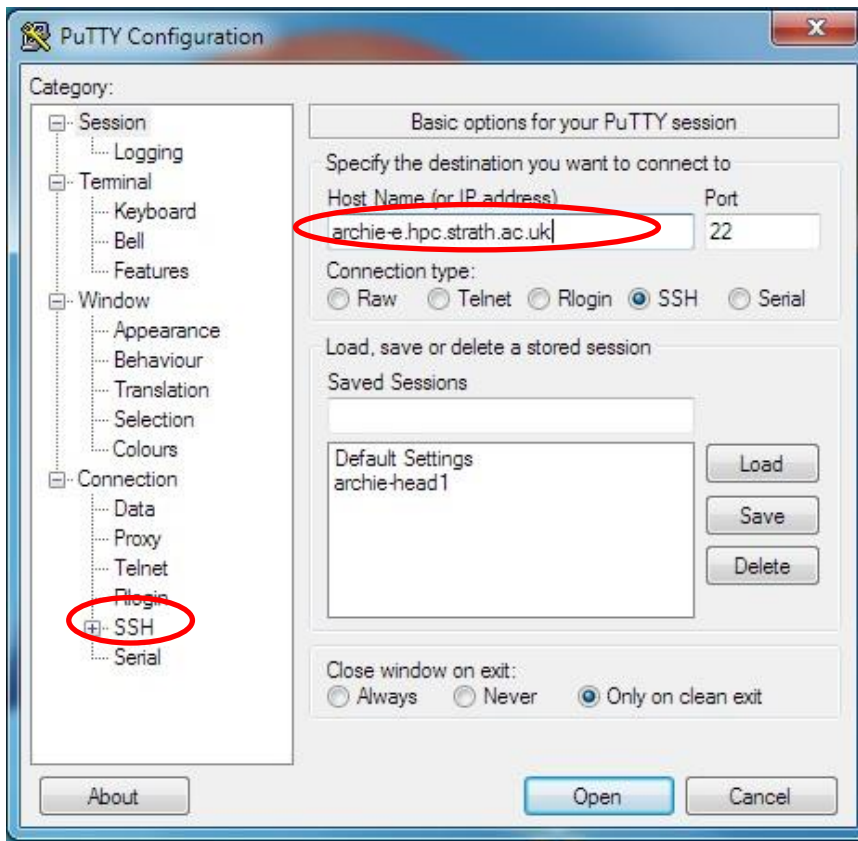
Use **archie-w**, **archie-e**, **archie-s** or **archie-t** and your **DS** **username**

Remote control - PuTTY

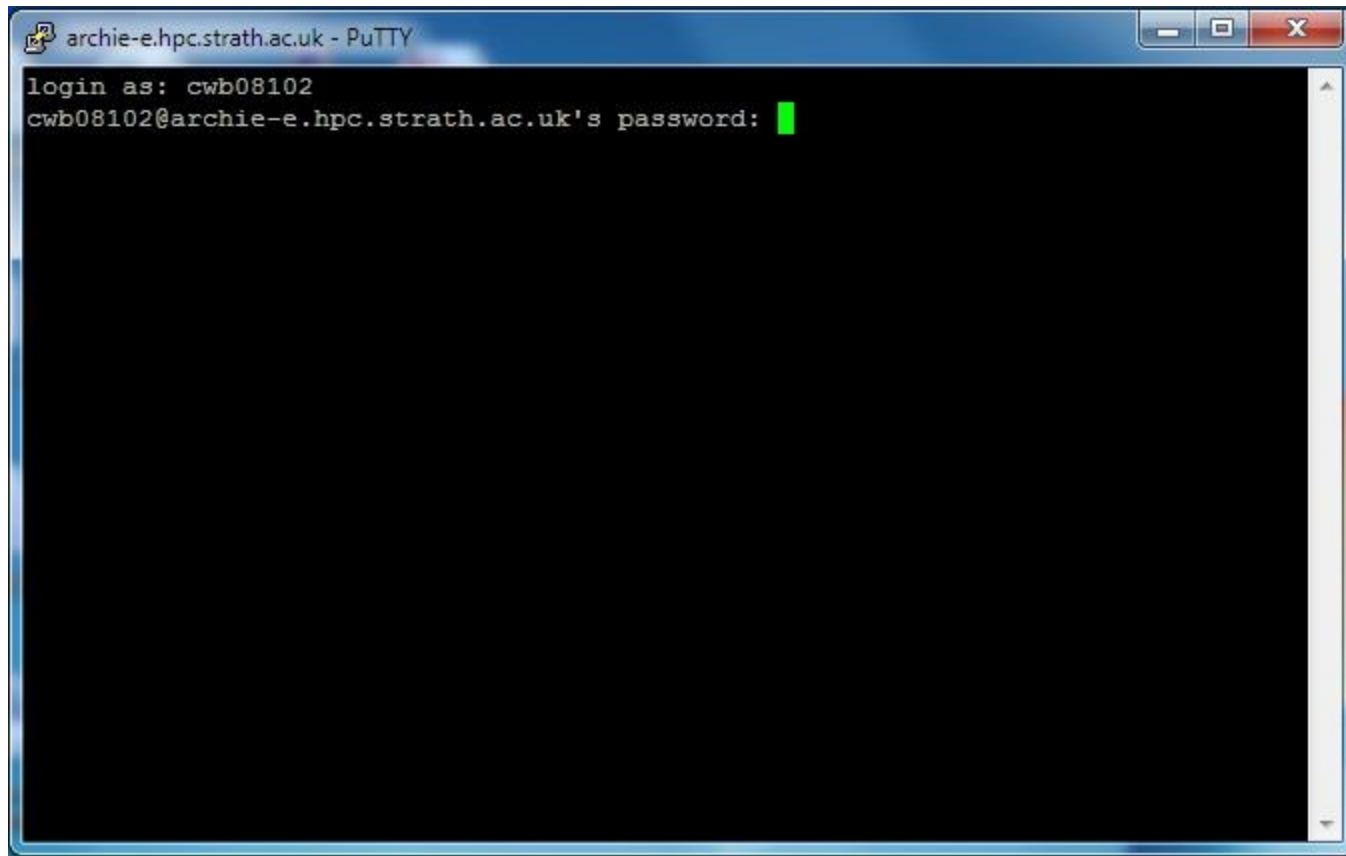
PuTTY is a free implementation (open source) of Telnet and SSH for Windows and Unix platforms, along with an xterm terminal emulator. It is also a SFTP client and FTP client for Windows.

To download visit <http://www.putty.org/>

Remote control - PuTTY



Remote control - PuTTY



The image shows a PuTTY terminal window titled "archie-e.hpc.strath.ac.uk - PuTTY". The terminal displays the following text:

```
login as: cwb08102  
cwb08102@archie-e.hpc.strath.ac.uk's password: █
```

The terminal has a black background with white text. A green cursor is visible at the end of the password prompt. The window has a standard Windows-style title bar with minimize, maximize, and close buttons.

Access with remote display

Using **ThinLinc** remote desktop client, we can run a remote desktop session on ARCHIE-WeSt

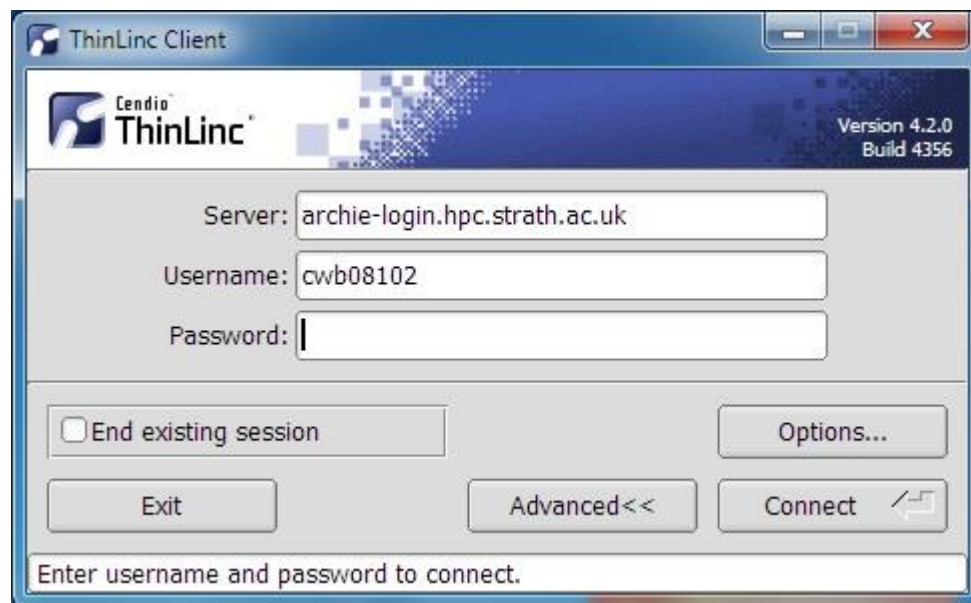
- Download from <https://www.cendio.com/downloads/clients>
- Instructions at <http://www.archie-west.ac.uk/archie-access>





Remote Desktop - ThinLinc

To have the remote **desktop session connect to** archie-login.hpc.strath.ac.uk, use your DS username and password.



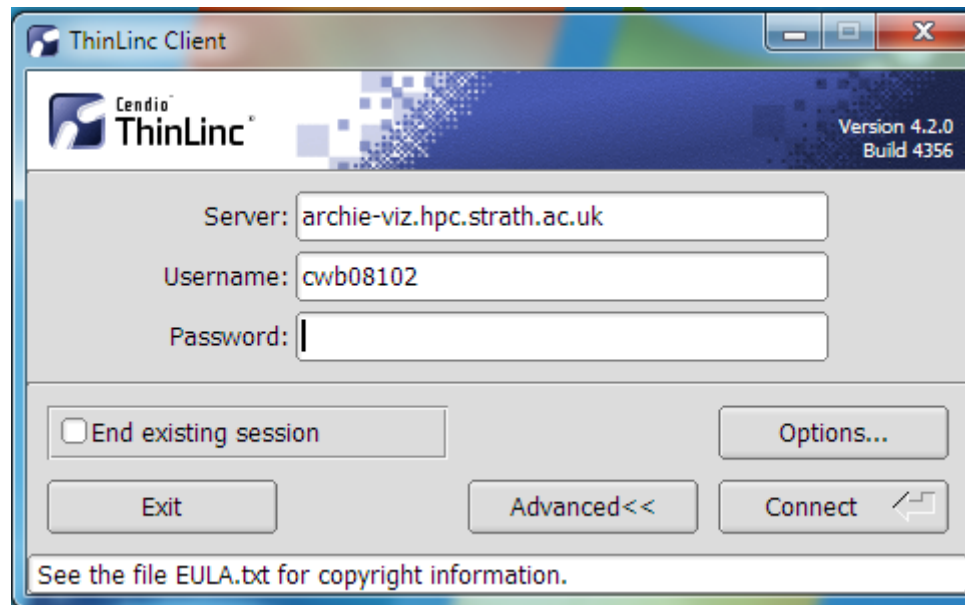
Pressing **F8** within the desktop session will give you the access to the **ThinLinc** client options.

You can “suspend” the session by simply closing the window via the “**X**” on the top corner. You can of course resume the suspended sessions. However, if you have no applications running we recommend that you **log out** and release the license.

Access to Visualization server

Using **ThinLinc** remote desktop client, we can run a remote desktop session on ARCHIE-WeSt visualization server

To use ARCHIE-WeSt visualisation server graphic card instead of your local graphic card use the command **vglrun**. For example: **vglrun vmd** instead of **vmd**.



Transferring Files

scp - **secure copy**

scp copies files over the network securely, uses ssh for data transfer

scp -r - copy the entire directory (with sub-directories)

scp -p - preserve file attributes and timestamps

Transferring Files

scp - secure copy

Sitting at your desktop machine:

```
scp -rp cwb08102@dm1.hpc.strath.ac.uk:/lustre/strath/physics/cwb08102/MY_DATA .
```

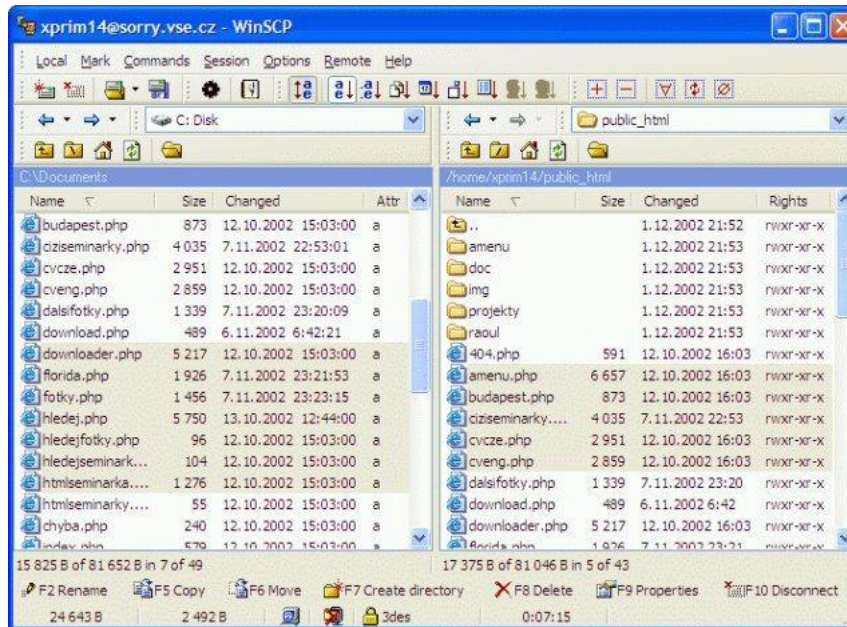
copy the entire directory (with sub-directories) from ARCHIE to your current working directory

- p** - preserve file attributes and timestamps
- r** - option to transfer entire directory

Transferring Files with GUI

WinSCP is a SFTP client and FTP client for Windows.

Its main function is the secure file transfer between a local and a remote computer. For fast transfer always connect to **dm1.hpc.strath.ac.uk**



Local Windows
Machine

Remote computer

Application interface similar to Norton Commander. To download visit <http://winscp.net/eng/index.php>

Transferring Files to H & I drives

(Strathclyde users only)

Our “data-mover” server `dm1.hpc.strath.ac.uk` has a 10 Gb/s link to the campus network.

ssh to `dm1.hpc.strath.ac.uk`

- **To copy files to your I drive space :**
on `dm1`, type: `mount_idrive` (you will be prompted for your DS password)

I drive will be mounted at `~username/i_drive`

- **To copy files to your H drive space:**

Type: `mount_hdrive`

H drive will be mounted at `~username/h_drive`

You can then copy files to `~username/i_drive` or `~username/h_drive`

Once finished, type `umount_idrive` or `umount_hdrive` as appropriate

User space

- Users home directory at /users/<userid> (5 TB disk array)
e.g. /users/abc03115
- Each user will have a “LUSTRE” directory at
/lustre/<univ>/<department>/<userid>
e.g. /lustre/strath/mecheng/abc03115
- Both directories will have quotas enabled
 - To view your /users quota type: **quota**
 - To view your lustre quota, type: **lfs quota /lustre**
 - A “soft quota limit” has been enabled (250GB, hard limit 1TB)
 - Can exceed the quota by an unspecified amount
 - will have 7 days to get back to within your limit
 - otherwise account is automatically write protected.
- Backup:
 - /users directory will be backed up (store essential files here)
 - /lustre **will not** be backed up

Environment

- Default shell is **bash** (shell: command interpreter)
- Different software packages requires environment variables to have different values e.g. PATH, LD_LIBRARY_PATH
- We handle this using linux **modules**:

available modules: **module avail**

loaded modules: **module list**

Intel compiler: **module load compilers/intel/15.0.3**

lapack: **module load libs/gcc/lapack/3.3.1**

remove a module: **module rm compilers/intel/15.0.3**

remove all modules: **module purge**

SUN Grid Engine (SGE)

- Job submission is handled via SGE
- Basic commands:

qconf -sql

- lists all queues

qstat -g c

- queue summary

qsub start-job.sh

- launches job using the script start-job.sh

qstat

- lists your jobs (qw - waiting, r - running)

qstat -u "*"

- lists all jobs in the queue

qstat -j <JOBID>

- gives fuller detail

qacct -j <JOBID>

- gives detail on completed jobs

qdel <JOBID>

- deletes job from queue

SUN Grid Engine (SGE) (cont ...)

- Queues

- | | | |
|-------------------------|-------------------------------|---|
| <code>parallel.q</code> | <code>(parallel-low.q)</code> | - main parallel queue (up to 3312 cores) |
| <code>serial.q</code> | <code>(serial-low.q)</code> | - main serial queue (up to 3312 cores) |
| <code>multiway.q</code> | <code>(multiway-low.q)</code> | - like parallel.q but specifically for Fluent |
| <code>smp.q</code> | <code>(smp-low.q)</code> | - “fat node” queue |
| <code>gpu.q</code> | <code>(gpu-low.q)</code> | - gpu queue |

- Note:

- parallel jobs have priority over serial jobs
- As you submit more and more jobs, the priority decreases

Project Identifiers

- Every job must be associated with a “project”
- You will be issued with a project identifier that you must use in your jobs scripts e.g.

```
#$ -P training.prj
```

Launching Jobs

- Serial job:

Sample script: `gedit /opt/gridware/job-scripts/start-serial.sh`

```
# Simple serial job submission script

# Specifies that all environment variables active within the qsub
# utility be exported to the context of the job.
#$ -V
# Execute the job from the current working directory. Standard output and
# standard error files will be written to this directory
#$ -cwd
# Associate with the "training project"
#$ -P training.prj
# Submit to the queue called serial-low.q
#$ -q serial-low.q
# Merges standard error stream with standard output
#$ -j y
# Specifies the name of the file containing the standard output
#$ -o out.$JOB_ID
#Indicate runtime
#$ -l h_rt=01:00:00

~/bin/hello-gcc-serial
```

Parallel Environments

- Each parallel queue has at least one associated “parallel environment”
- When launching a job, we determine which parallel queue the job will run on, by specifying the “parallel environment” (PE) in the job script.

- For the standard nodes there are PE's:

- The main parallel queue has two PE's:

mpi & mpi-verbose

- For Fluent:

multiway

- For SMP nodes there are PE's:

smp & smp-verbose

- Include a directive in your jobs script like:

```
#$ -pe mpi-verbose 1 (12 cores) or
```

```
#$ -pe smp-verbose 12 (12 cores)
```

Resource Reservation

- Add

```
#$ -R y
```

to your job script, then the queueing system will reserve nodes for parallel jobs

- Add a runtime by adding (for short jobs don't use the above)

```
#$ -l h_rt=6:00:00
```

for a runtime of 6 hours. If the job exceeds this time it will be terminated. Therefore, you should over-estimate the runtime by a reasonable amount

- ***Default runtime is 14 days – it may be to your advantage to specify a shorter runtime***

- The `#$ -l h_rt` option above should not be confused with the flag:

```
#$ -ac runtime="3hours"
```

which is sometimes provided for information purposes only. It is not necessary to supply this if using "`#$ -l h_rt`".

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/foam-openmpi-parallel.sh`

```
#
export PROCS_ON_EACH_NODE=12
# ***** SGE qsub options *****
#Export env variables and keep current working directory
#$ -V -cwd
# Associate with the "training project"
#$ -P training.prj
#Select parallel environment and number of parallel queue slots (nodes)
#$ -pe mpi-verbose 2
#Combine STDOUT/STDERR
#$ -j y
#Specify output file
#$ -o out.$JOB_ID
#Request resource reservation (reserve slots on each scheduler run until
# enough have been gathered to run the job
#$ -R y
#Indicate runtime
#$ -l h_rt=6:00:00
# ***** END SGE qsub options *****

export NCORES=`expr $PROCS_ON_EACH_NODE \* $NSLOTS`
export OMPI_MCA_btl=openib,self

mpirun -np $NCORES dsmcFoam -parallel
```

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/ansys-smp-parallel.sh`

```
#
# ***** SGE qsub options *****
#Export env variables and keep current working directory
#$ -V -cwd
#Select parallel environment and number of parallel queue slots (nodes)
#$ -pe smp-verbose 2
# Associate with the "training project"
#$ -P training.prj
#Combine STDOUT/STDERR
#$ -j y
#Specify output file
#$ -o out.$JOB_ID
# Indicate runtime
#$ -l h_rt=6:00:00
# ***** END SGE qsub options *****

ansys160 -b nolist -p $LICENSE -np $NSLOTS -j pv-nonlinear -i pv-nonlinear.txt -o pv-
nonlinear.out
```

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/fluent_multiway.sh`

```
#
# ***** SGE qsub options *****
#Export env variables and keep current working directory
#$ -V -cwd
# Associate with the "training project"
#$ -P training.prj
#Select parallel environment and number of parallel queue slots
#$ -pe multiway 24
#Combine STDOUT/STDERR
#$ -j y
#Specify output file
#$ -o out.$JOB_ID
#Request resource reservation (reserve slots on each scheduler run until enough have
been gathered to run the job
#$ -R y
# Add runtime indication
#$ -l h_rt=50:00:00
# ***** END SGE qsub options *****

fluent 3d -pib -ssh -sge -g -i fluent_input.txt > output
```

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/matlab_serial.sh`

```
#!/bin/bash
# Simple serial job submission script
# Specifies that all environment variables active within the qsub
# utility be exported to the context of the job.
#$ -V
# Execute the job from the current working directory. Standard output and
# standard error files will be written to this directory
#$ -cwd
# Submit to the queue called serial-low.q
#$ -q serial-low.q
#Specify Project identifier
#$ -P training.prj
# Merges standard error stream with standard output
#$ -j y # Specifies the name of the file containing the standard output
#$ -o out.$JOB_ID
# Add runtime indication
#$ -l h_rt=10:00:00
# ***** END SGE qsub options *****

matlab -nodisplay -nodesktop -singleCompThread -r "my_matlab_file;exit "
```

Name of your matlab script but without ".m" suffix

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/matlab_parallel.sh`

```
#
export PROCS_ON_EACH_NODE=12

# ***** SGE qsub options *****
#Export env variables and keep current working directory
#$ -V -cwd
#Select parallel environment and number of parallel queue slots (nodes)
#$ -pe mpi-verbose 1
#$ -q parallel-low.q
#$ -P training.prj
#Combine STDOUT/STDERR
#$ -j y
#Specify output file
#$ -o out.$JOB_ID
#Request resource reservation (reserve slots on each scheduler run until enough have
been gathered to run the job
#$ -R y
# Add runtime indication
#$ -l h_rt=05:00:00

# ***** END SGE qsub options *****
matlab -nodisplay -nodesktop -r "my_parallel_matlab_file;exit"
```

Launching Jobs

- A job script must be created in order to launch jobs with SGE

Sample script: `gedit /opt/gridware/job-scripts/start-starccm.sh`

```
#
export PROCS_ON_EACH_NODE=12

# ***** SGE qsub options *****
#Export env variables and keep current working directory
#$ -V -cwd
#Select parallel environment and number of parallel queue slots (nodes)
#$ -pe mpi-verbose 2
#$ -q parallel-low.q
#$ -P training.prj
#Combine STDOUT/STDERR
#$ -j y
#Specify output file
#$ -o out.$JOB_ID
#Request resource reservation (reserve slots on each scheduler run until enough have
been gathered to run the job
#$ -R y
# Add runtime indication
#$ -l h_rt=72:00:00
# ***** END SGE qsub options *****

export NCORES=`expr $PROCS_ON_EACH_NODE \* $NSLOTS`

starccm+ -batchsystem sge -rsh /usr/bin/ssh -mpidriver infiniband -mpi platform \
        -np $NCORES -batch -power -podkey IRqKvpmQHfXXXXCUGDWNSKg \
        -licpath 1999@flex.cd-adapco.com RunStarMacro.java \
        19MM20.sim > 19MM20output
```

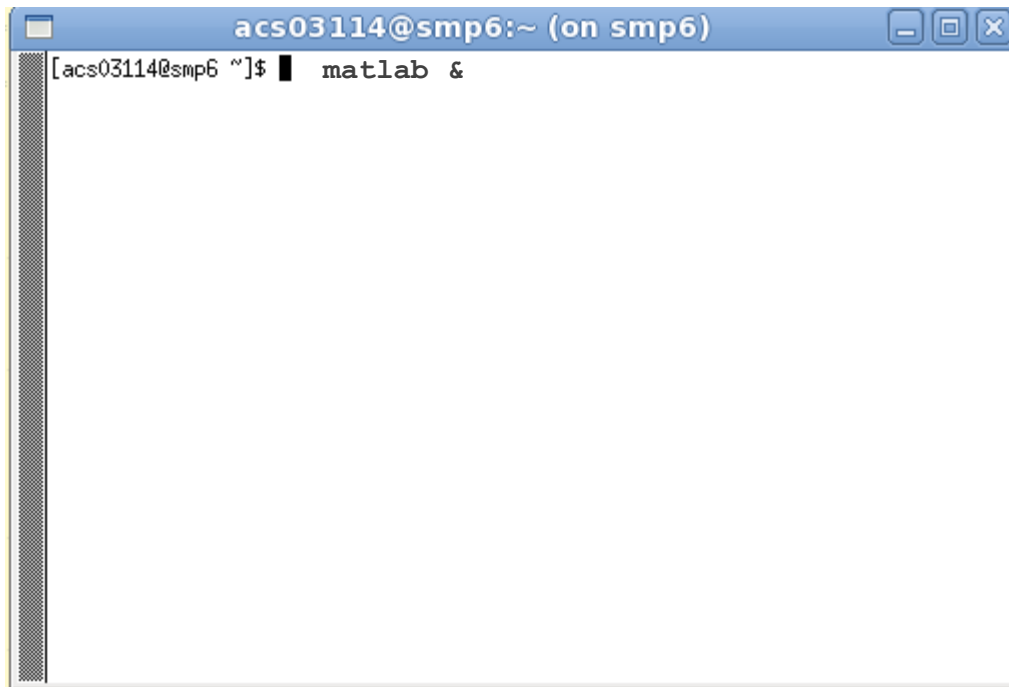
Interactive Jobs

- If you want to run a program *interactively* e.g. using a GUI
 - still needs to be done via SGE
 - Use the “qrsh” command for this

```
qrsh -P training.prj -V -pe smp 4 xterm
```

This will open up a terminal window from which you can launch your program (using 4 cores, in this example)

Interactive Jobs ... cont



```
acs03114@smp6:~ (on smp6)
[acs03114@smp6 ~]$ matlab &
```


Launching Jobs (cont ...)

Note:

- Lines starting with `#$` are SGE directives
- Lines starting with `#` are simply comments
- Submit the job script with the command:

```
qsub start_fluent_smp.sh
```

- View Progress by typing: `qstat`
- Delete the job by typing: `qdel <JOBID>`
- Remember to add a job runtime (in hours) in your script e.g.:

```
#$ -l h_rt=24:00:00
```

Ground Rules

- Be considerate
- Don't generate lots of data just because you can
- Do not launch a “production job” without knowing
 - A. How long it will take
 - B. How much data it will generate

You can run short jobs in order to produce an estimate of A & B
- Do not run jobs from the /users directory
 - You will not benefit from Infiniband
 - Disk space is more limited
- Keep source code and other important files in /users
- Post processing can be conducted on [archie-viz](#)

General Remarks

- In all graphical presentations such as conference presentations, posters, lectures etc., the graphical logo of ARCHIE-WeSt should be used. The logo can be downloaded from www.archie-west.ac.uk/acknowledge-archie-west-2/
- In papers, reports etc., include this statement in the Acknowledgement paragraph: “Results were obtained using the EPSRC funded ARCHIE-WeSt High Performance Computer (www.archie-west.ac.uk). EPSRC grant no. EP/K000586/1.”
- Update PURE (Strathclyde’s Current Research Information system) and associate your publication with UOSHPC

It is necessary to get the project extension

Documentation

- Website: <http://www.archie-west.ac.uk>
- Service Status <http://www.archie-west.ac.uk/status>
- Mailing list: archie-users@lists.strath.ac.uk
- Support: support@archie-west.ac.uk