The University of Queensland

Research Computing Centre

FlashLite User Guide

Document Status

This update: May 31 2019 by David Green

General Information

The name?

In recent years, we've been using Queensland lake names (such as Wivenhoe, Weyba, Wobby, Wappa, Eacham, Barrine, Euramoo, Awoonga and Tinaroo) for our clusters.

FlashLite is special and its name is derived from its data intensive character and its design heritage.
FlashLite's design was inspired by Gordon, the United States' National Science Foundation supercomputer at San Diego.

FlashLite differs from other supercomputers: It relies on high-speed flash-based solid-state drives, rather than traditional hard-disk drives, for secondary storage.
It will serve data-intensive research by providing very rapid access to data sets and by supporting parallel processing.

FlashLite augments our other HPC platforms namely

This cluster is an integral part of the RCC Advanced Computing Strategy for UQ.
The currently available resources, and their primary purpose, are detailed in the following table.

Resource	Primary Purpose
Awoonga	A high throughput computing cluster for loosely coupled, small footprint jobs and job arrays (i.e. single-node or sub-node scale)
FlashLite	A specialized HPC cluster for data-intensive computation (e.g. very large RAM and/or extreme input/output requirements).
Tinaroo	A high performance computing cluster for tightly coupled message passing jobs (i.e. multi-node message passing).
Wiener	A HPC platform targeting image de-convolution using graphical processing units (GPUs).

This research computing loadout provides a competitive local capability, for UQ, that spans a wide range of requirements.

FlashLite has been purchased to fulfill a specific role within the RCC's Advanced Computing Strategy.
FlashLite is primarily for the Very large memory and very high input-output (IO) jobs

The batch system configuration on FlashLite prioritizes data intensive computational workloads.
Small footprint jobs are better served by the Awoonga cluster but small jobs can technically be submitted on FlashLite especially if they have atypical RAM requirements.
Usage accounting is being to used to allow for reporting of usage.

Who can use the FlashLite cluster?

- Any UQ researcher with a data intensive workload,
- Any researcher who is actively collaborating with a UQ researcher,
- Any researcher from a partner institution who has data intensive workloads,
- NCMAS awardees with data intensive workloads.

Getting a Login

Gaining access to FlashLite involves a technical justification. It is straightforward but is not automatic

It is strongly recommended that all FlashLite users from QCIF partners (but especially researchers that are new to HPC) apply first for an account on Awoonga to get a suitable grounding.

UQ staff and students should use your UQ signon.
It is preferable that UQ research higher degree students use their staff username (if they have one) instead of their student number login.

Accessing the HPCs

Currently, all valid users should connect to the cluster by using one of the following methods.

- A command line SSH client to the cluster login nodes
- A SFTP/SCP file transfer tool
- The web based access portal for Tinaroo at this link provides
    - A full graphical remote desktop style access to Tinaroo, and
    - Browser based command line sessions to each HPC cluster (Awoonga and Flashlite as well as Tinaroo).

If you have any problem with the web based access to HPCs, please submit a support request to rcc-support@uq.edu.au.

Your UQ username and password should always work when you access HPCs.
When you update your UQ password, that password will immediately become your password for accessing HPCs.

Use of SSH public/private key pairs is a convenience, especially when combined with a key agent utility.
Some care must be used and best practices for use of SSH keys are described elsewhere.

Unlike the experience on previous HPC clusters that used a software based load balancer, there is no performance penalty for connecting to a cluster's generic login alias (e.g. tinaroo.rcc.uq.edu.au) for data transfers.
Having said that, some users of file transfer tools have experienced connection failures when using the generic login node alias e.g. tinaroo.rcc.uq.edu.au
In those situations, we recommend you try **SFTP** mode (port 22) and connect to one of the two actual login nodes e.g. **tinaroo1.rcc.uq.edu.au** or **tinaroo2.rcc.uq.edu.au** for the Tinaroo HPC.

More details about the command line tools, and file transfer tools, are provided below and in separate topic specific user guides (Connecting to HPC and File Transfers)
Be aware that transferring text files in binary mode can result in files that are not usable on the HPC in some circumstances.

Individual Login Nodes and Cluster Aliases

Cluster	Login Nodes	Cluster Login Alias
Awoonga	awoonga1.rcc.uq.edu.au	awoonga.rcc.uq.edu.au
FlashLite	flashlite1.rcc.uq.edu.au	flashlite.rcc.uq.edu.au
	flashlite2.rcc.uq.edu.au	
Tinaroo	tinaroo1.rcc.uq.edu.au	tinaroo.rcc.uq.edu.au
	tinaroo2.rcc.uq.edu.au	
Wiener	wiener.hpc.dc.uq.edu.au	wiener.hpc.dc.uq.edu.au

HPC Nodes

The HPC clusters (Awoonga, FlashLite, Tinaroo and Wiener) all have outward facing login nodes for users to access the HPC.
They also have a number of compute nodes that are hidden from the internet on private network ranges.

Cluster Comparison

Cluster	Logins	Computes	Network
Awoonga	1	40	10 Gbps ethernet
FlashLite	2	66	InfiniBand Spine and Leaf Topology

Tinaroo	2	244	InfiniBand Hypercube Topology
Wiener	1	32	EDR InfiniBand

Detailed Configurations

Cluster	Cores/Node	CPU	GB	Max Job GB	GB/Core	TMPDIR
Awoonga	24	E5-2670	256	240	10	200GB
FlashLite	24	E5-2680	512	504	21	4.4TB
Tinaroo	24	E5-2680	128	120	5	852GB
Wiener	28	Intel Xeon Gold 6132	384	380	14	

CPUs

Cluster	CPU
Awoonga	Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz
FlashLite	Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
Tinaroo	Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz
Wiener	Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz

GPUs

Cluster	GPUs
Awoonga	4x Tesla K20*
FlashLite	0
Tinaroo	0
Wiener	96x NVIDIA Volta V100

*To be upgraded soon to Pascal P100

Login Nodes

As noted elsewhere, the Tinaroo and FlashLite HPCs have a pair of login nodes for redundancy and load sharing.
Awoonga and Wiener have a single login node.

When you log in to a HPC (e.g. tinaroo) via the cluster login alias (e.g. tinaroo.rcc.uq.edu.au), you will find yourself landing on one of two identical login nodes (e.g. tinaroo1 or tinaroo2).
We recommend that you connect to the hostname tinaroo.rcc.uq.edu.au to ensure you get an available login node.
You are able to connect to a specific login node if you need to.
The login nodes are similar configuration to the compute nodes.

Note: The Awoonga/FlashLite/Tinaroo HPC login nodes are publicly accessible machines.

You may choose to use UQ VPN to connect from off campus, but it is not essential.
To thwart password cracking attempts, a security tool will block repeated connection attempts to login nodes above a threshold.
For more information about this security feature and its ability to prevent you from logging in, please read the README module for SSH on the system.

Compute Nodes

A HPC job needs to leave some memory for the operating system to function with.
The batch system has been configured to ignore/reject jobs that request more RAM per node than the Usage GB figurein the table above.

The RCC cluster security model allows you to SSH to compute nodes directly, but only in situations where you have a batch job running there.
This prevents accidental harm to valid batch jobs running on the compute nodes.

vSMP Nodes

FlashLite was purchased with the Versatile SMP server aggregation and virtualisation software from ScaleMP.com

The compute nodes in FlashLite can be flexibly aggregated together into larger "supernodes" using the vSMP tool.
This software tool aggregates multiple physically separate servers into one single virtual high-end system.

Typically, multiple physical nodes in FlashLite are combined to create a single virtual server based on either

- all the cores and all of the memory of the physical machines (system expansion, SYX, mode), or,
- the cores and RAM from just one physical machine, combined with just the RAM from the rest of the participating machines (memory expansion, MEX, mode)

Referred to as a “vSMP supernode”, there is aggregation of the CPUs, memory, and I/O capabilities of multiple physical hosts into one virtual machine (VM).

Although it is straightforward to take down and rebuild the "supernodes", experience has shown that it is beneficial to have supernodes on standby and configured into the batch system.
Currently there are 3 supernodes.

Host Name	nCPUs	RAM	Batch Queue	vSMP Mode	Physical Servers
flvc0	48	1tb	BigSMP	SYX	2
flvc1	24	2tb	BigMemory	MEX	4
flvc2	24	1.5tb	BigMemory	MEX	3

Jobs will route to these queues and nodes according to the resource requests made in the batch system job submission.
You should not specify the queue.

Other configurations are possible but require manual intervention by the systems team and available nodes to aggregate, so may incur delays. If you anticipate the need for larger memory footprints, please submit a support request.

The Storage Subsystem

Storage Technologies on HPC

Local Disk on Awoonga/FlashLite/Tinaroo

These three HPC clusters have access to the same network storage options (see below) but also have local disks that are best accessed using the TMPDIR environment variable within batch jobs. The disk capacities are details in the section below about Quotas and Limits.

High Performance Network Scratch Space on Wiener

Instead of local disk on each Wiener node, space is available on a high performance network filestore mounted at /scratch.
You will find a folder for your Faculty/School/Institute under /scratch and you will be able to write into that subfolder.

Cluster Local GPFS Storage on Awoonga/FlashLite/Tinaroo

These three clusters share storage that is based on a high performance IBM GPFS storage system.
That storage was installed at the end of June 2021 to replace the DDN GridScaler device that was part of the procurement of FlashLite in 2015.

- Each node in Awoonga, FlashLite and Tinaroo clusters is a native GPFS client for that storage system. This enhances the data performance.
- The locations hosted on this are /sw /home and /scratch (/scratch/user and /scratch/project)
- There are strict quotas being enforced on /home and /scratch.
- There are purging policies being applied to the /scratch to maintain a usable capacity.

~~Cluster Local BeeGFS Storage on FlashLite~~ No Longer Supported

The nodes in the FlashLite cluster are equipped with 4.8TB of solid state NVME drives.
A prototype BeeGFS storage cluster had been deployed to aggregate space from multiple FlashLite nodes into a single distributed storage device.

However, the performance and capacity of our new /scratch filesystem has made it unnecessary to keep the BeeGFS storage, so it has been disassembled.

QRIScloud Collection Storage

The physical proximity to the QRIScloud infrastructure, means that HPC clusters can also directly mount QRIScloud collection storage via the NFS protocol. Historically these have been HSM based collections. More recently they have been handled as MeDiCI/RDM style collections.

UQ RDM / Medici Data Fabric Storage

The UQ RDM collection storage is presented to the HPCs like a QRIScloud collection via the Medici Data Fabric.
Currently that is done using the same NFS based mechanism used for other QRIScloud data collections.
We are evaluating improvements to the attachment of the Medici Data Fabric to the HPC to enhance the performance.
The Medici data fabric is also available on the Weiner HPC system.

The RDM collection storage is a multi-layer, multi-site data caching and replication system.
You see your collection in one of these layers at one of the sites.
When you make changes to your data at a site, those changes propagate laterally, to other sites, and vertically through layers at the primary site (which for HPC connected "Q collections" is our data centre at Springfield)
Your data may have been pushed offline due to inactivity and your collection reaching its low water mark (quota).
It will still show up in the file list, but the size may be smaller than expected.
This is a "file stub" and the file must be retrieved from the HSM layer before you will be able to work with its contents.

How is my collection data stored and accessed?

The RDM/MeDiCI Data Fabric stores vast amounts of data.
However, it does not do so solely by powering a lot of disk drives all of the time.
Instead, RDM storage (specifically the Q collections attached to the HPCs) are housed in a multi-level cache infrastructure (stack) built upon a "hierarchical storage management" system.

Layer	Storage Technology
GPFS Cache	Disk
HSM Cache	Disk
HSM Spin Down	Zero Watt Storage
HSM Archive	Robotic Tape Storage

This full storage stack resides at only one site ("home").
The other sites you may have need to connect to (e.g. for scientific instruments or your R:\ drive or the cloud.rdm.uq.edu.au) just have the GPFS cache that you interact with.
That GPFS cache layer stays in sync with "home".

Some newer collections and older ones with specific characteristics are held on a device called UQ03.
That device combines the GPFS cache and the HSM cache in a single layer.
This combination provides a more seamless experience for big collections and collections that are subject to a high rate of "churn".

On the HPCs, you always access your RDM Q collection data from the GPFS Cache Layer.
On Awoonga/FlashLite/Tinaroo, this cache layer gets mounted onto the HPC at a location such as `/QRISdata/Q1234`.
On Wiener, it is the QBI replica of your data that you interact with. That is mounted at `/afm01/Q1/Q1234`
The "HSM Spin Down Layer" powers down after a period of inactivity to conserve energy and to reduce our carbon footprint.

Your active data can and should reside in the GPFS Disk Cache layer.
When you don't access your data for a period of time, it can be evicted from the GPFS layer and pushed offline in the HSM.
Refer to the section below entitled "Pre-fetching files from HSM" about how to efficiently retrieve archived data from the HSM.

Ensuring that your RDM Collection is easily accessible from HPCs

When you apply for the RDM collection storage "record" that you want to access from UQ HPC systems (Awoonga/FlashLite /Tinaroo/Weiner), you must make sure

- that you check the box near the bottom that says **The project data needs to be mounted on UQ HPC facilities.**
- that you check any of the boxes that pertain to your data. Some types of data may not be permitted to be stored on the Medici fabric.
- that you read and understand the various terms and conditions.

This will allocate you an initial 1TB of storage almost immediately that you will be able to access from the HPCs.
The quota can easily be increased as needed.
Your RDM collection is the logical place to retain your results and to be able to share data with your supervisor and other researchers as necessary.

- Once you have a valid RDM MeDiCI collection
  - It will automatically become accessible (via the autofs mechanism) onto Awoonga/FlashLite/Tinaroo HPC systems.
  - It needs to be mounted onto Wiener by the systems team. You request that by submitting an email to helpdesk@qbi.uq.edu.au

What could adversely impact performance of RDM Collection Storage?

You should avoid running computational workloads against your RDM collection- especially if there is a lot of Input/Output.

Running heavy computational workloads in RDM collections can have diabolical impacts on other users of the HPC, QRIScloud storage and the research cloud.
Please use scratch disk space provided in all the HPCs and

- copy in your inputs,
- perform your computations in the scratch spaces ($TMPDIR, /scratch/user/, /scratch/project/ or /nvmescratch (Weiner), then
- copy back your outputs back to your RDM collection as "lumps" (i.e. tar or zip files)

Why does my collection sometimes appear to be owned by the root user?

When you look at the collections from their parent directory /QRISdata, any collection directory that is not mounted will appear to be owned by root.
Once a collection is mounted the ownership changes to that in the mounted fileset.

So to properly look at these, you would need to look inside the collection directory (and thus forcing the mount request). The `ls -l` command would help you do that.

Why do my collection files sometimes appear to be owned by the nobody user?

This can happen under some circumstances.
It is often caused because of the way that the files were added to the collection.

It is important for the usability of your collection that file permissions are correctly set.
RDM collection files and folders should have the following features

- owned by the Q system account for the collection, or, owned by the user who created the file
- belong to the collection RW group (eg. Q1234RW)
- be group readable and writeable but not accessible by others
- folders should have the group writable sticky bit set ("s")
- have other switches (including the "+") set for the RDM mechanism to work flawlessly.

An example is shown here for a directory with some files in it:

```
tinaroo1:/local/home/davidg # ls -salh /QRISdata/Q0286/UQ-RCC/tests
total 9.8G
 32K drwxrws---+ 2 Q0286 Q0286RW  32K Jun 24 10:24 .
 32K drwxrws---+ 6 Q0286 Q0286RW  32K Mar 18  2019 ..
   0 -rw-rwx---+ 1 Q0286 Q0286RW 1.5G Jun 11  2017 tensorflow-1.1.0.sapp.old.20200601
9.8G -rw-rwx---+ 1 Q0286 Q0286RW 9.8G Mar 17  2018 tensorflow-1.6.0-cuda-9.0.sapp
   0 -rw-rwx---+ 1 Q0286 Q0286RW 2.4G Jan 29  2018 tensorflow-gpu-1.4.1.sapp.old.20200601
   0 -rw-rwx---+ 1 Q0286 Q0286RW 2.1G Apr 24  2017 tensorflow.sapp.old.20200601
   0 -rw-rwx---+ 1 Q0286 Q0286RW   51 May 24  2018 textFile
```

If you find files and folders within your collection that show up on the HPC with incorrect ownership, or you are prevented from updating a file, then please submit a support request to rcc-support@uq.edu.au.

Why do my collection files sometimes look empty in output of the ls -salh command?

In the example directory shown in the previous section, you can see that only one file has a size on disk (the left column of `ls -salh` output) of 9.8G(B) that matches the expected file size (in column 6 of `ls -salh` output).
That means that currently the file is in GPFS cache disk and available to use immediately.

The other files in that folder have a size on disk (left column) of 0 instead of their file size.
This means that the file has been put into nearline HSM storage and will need to be retrieved from HSM before being available to use.
We recommend that you use the **recall_medici** command on the HPC to trigger a recall of a file you need.
You do not have to use the recall_medici command as long as the operation you are performing to trigger the recall from HSM is able to wait without failing for the file contents to be retrieved from the HSM layer.

When and how to use the recall_medici script

If the processing you are doing is likely to fail because the file is not in the GPFS cache layer (and the processing does not handle the wait properly), then you must `recall_medici` the files before you attempt the processing that might fail.

You can do that recall_medici step in one of three modes:

1. at the command line shell (login node or qsub -I session) - use a wildcard
2. as part of a preprocessing batch job (which subsequent jobs need to wait for completion) to retrieve all the files needed next
3. just-in-time - as a preceding step within each processing batch job

The third option could cause your processing batch jobs to need extra walltime because each processing job will be waiting for its file(s) to be recalled.

Why do my collection files sometimes fail to read or appear to have invalid format?

Very rarely we see situations where a file in GPFS cache is incompletely read back from HSM.
The output of an **@ls -salh** command shows a file size on disk (column 1) that is non-zero but less than the expected size (column 6).
In this situation, you should contact rcc-support@uq.edu.au

What can be done if I have deleted or damaged a collection file or folder accidentally?

The good news is that when a file is removed or modified in the GPFS layer it does not mean it is gone for good.
In most circumstances, we should be able to retrieve the contents of the file from the HSM and reinstate it.
The caveats on that is that the file has to have been successfully written to HSM some time before being deleted.
Depending on the time between creation and accidental deletion/modification we often can recover from a HSM copy which is not purged immediately.

In this situation, you should contact rcc-support@uq.edu.au

Sometimes my data appears to be missing

The RDM collection storage is a multi-layer, multi-site data caching and replication system.
Occasionally, the different sites can get "out of sync".
For example, files created on, or uploaded to, the HPC, may not appear immediately in your mapped network drive on your PC.
There have also been issues with the synchronisation to the cloud.rdm.uq.edu.au service endpoint.
Usually, just waiting a few hours is sufficient to allow for the data to be replicated between sites.

If the data at one site is out of sync for a prolonged period, you should submit a support request to

- ITS HelpDesk if the data is available at HPC but not available at R: drive or cloud.rdm.uq.edu.au
- Conversely, rcc-support@uq.edu.au if the data is available in the R drive but has not made it to the HPC.

Storage Quotas and Limits

See also the section on HPC Enhanced Access Mechanism

Current Settings

Cluster	File System	Access	GB Limit	Files Limit	Time Limit	Comment
A, F & T	**/home**	GPFS	50	1,048,576	none	Unlimited retention but no centralised backups. ~~Filesystem snapshots provide some scope for recovery from accidental deletion. (see storage related tricks section below)~~
A, F & T	**/scratch/user**	GPFS	150	100,000	Time based deletion policy.	Deletion is commencing soon.
A, F & T	**/scratch /project**	GPFS				Quota settings depend on project. Deletion policies are being implemented.
A, F & T	**/RDS/Q1234**	NFS				Quotas and timelimits as per QRIScloud collection allocation.
A, F & T	**/RDS/Q9876**	NFS/GPFS				Quotas and timelimits as per RDM/Medici collection allocation.
Awoonga	**/state /partition1**	local disk	200		job duration	Use **$TMPDIR** in running jobs on nodes. Purged on job completion.
FlashLite	**/state /partition1**	local disk	386		job duration	Available but not often used. You will need to purge manually.
FlashLite	**/nvme**	local disk	4400		job duration	Use **$TMPDIR** in running jobs on nodes. Purged on job completion.
Tinaroo	**/state /partition1**	local disk	852		job duration	Use **$TMPDIR** in running jobs on nodes. Purged on job completion.

All users should be using the variable $TMPDIR for their work and NOT hard coding the paths /state/partition1, or /nvme explicitly.

The rquota command

The `/usr/local/bin/rquota` command (usually, in your path) will inform you of your current usage, and quotas, across quota-controlled filesystems.
A quota limit value of 0 implies it is unlimited!

```
uqdgree5@tinaroo1:~/Tests/Tinaroo/PBS> rquota
FileSet        Used(GB) Limit    Files      Limit
30days_group   0        0        0          0
/home          10       20       49932      204800
```

```
/30days        395      1000     23219      3145728
/90days        74       400      71503      1048576

Group quotas
FileSet        Used(GB) Limit    Files      Limit
/groups         3071     2048     0          928383
```

What is $TMPDIR?

Inside a Batch Job

The TMPDIR is an environment variable (with value $TMPDIR) that is defined within every running batch job.
Each running batch job has a temporary location created that is

- is unique to the job and
- is owned and accessible to only the job owner.

No one else can access the contents of your $TMPDIR directory while your job is running.
Upon job completion or termination, the location specified by $TMPDIR is removed.

Outside of a running PBS job

The TMPDIR is also a defined environment variable in your unix command shell.
It has the value $TMPDIR that corresponds to your personal directory in the /30days filesystem.
This feature allows for ease of job script testing and for handling situations where software expects a $TMPDIR to exist.
The contents of the $TMPDIR (when not inside a running batch job) are left in their /30days location.

See also Storage Best Practices below about why and how to best use $TMPDIR in batch jobs.

Storage Best Practices

- Use $TMPDIR when you need local disk for your batch jobs.
  The $TMPDIR directory is created in /state/partition1 automatically as part of your batch job and is removed for you automatically at the end of the job.
- Saving user data randomly into local disk on a node (outside of $TMPDIR mechanism) can adversely impact other users. Please DON'T do that.
- Compute Nodes are periodically rebuilt and the local disk space is reformatted, so do not rely on using local disk on compute nodes unless via $TMPDIR.
- Don't forget that $TMPDIR is unique for each job and job-array sub-job.
  Although the path may be the same, the $TMPDIR directory will probably contain different files on different nodes and for different jobs.
- If you need to work with many small files, please keep them bound together in a single archive file (ZIP or tar).
  Copy the archive file to local disk (i.e. $TMPDIR) before unpacking the files to work on them in local disk space.

To figure out which directories have the most files, you can use the **`du`** command and the options **`--inodes`** and **`--max-depth`** combined with the `sort` and `head` commands to get a summary of your most populous folders.

```
uqdgree5@tinaroo1:~> du --inodes --max-depth 1 /home/uqdgree5/Tests | sort -nr | head -5
900     /home/uqdgree5/Tests
270     /home/uqdgree5/Tests/FlashLite
260     /home/uqdgree5/Tests/Tinaroo
115     /home/uqdgree5/Tests/Nimrod
84      /home/uqdgree5/Tests/PBSPro
```

Suggested Data Workflow

We suggest that you consider the following as a model for your data arrangements:

- use /home for small nonvolatile items such as documentation, source codes, PBS files and the like
- use /30days as scratch space for job outputs before copying them back to HSM or /90days or /home
- use /90days for reference or input data files (keep them tar-ed or zip-eds or like fastqScreenDB (if that is a reference input) or packed inputs.
- use the UQ research data HSM or a QRIScloud collection to hold (compressed) tar archives of your input data.
- unpack the compressed input data onto local disk on the compute nodes ($TMPDIR) if there is space
  Awoonga nodes have 200GB of local disk. Tinaroo and FlashLite have more local disk.

Storage Related Tricks and Tips

Stale File Handles

From time to time you may experience the dreaded "Stale file handle" message.

This means that the machine you are working on (usually a login node) has lost its connection to some part of the storage infrastructure backend.
It can affect /home and the other filesystem served by the HPC cluster GPFS device, as well as /QRISdata collection storage.
The stale file handle situation can be due to a failure of the backend storage sub-system, but can also be due to a mounting problem for just the node you are on.

In most situations, simply waiting will allow time for the connection to be re-established automatically.
However if you do not wish to wait then you can login to a different node (see the Connecting to HPC User Guide for how to connect to a specific login node, instead of the alias name)
If you waited, and/or tried another entry point and you still get stale file handles, then please check the RCC Active Incidents page.
Submit a support request (mailing rcc-support@uq.edu.au) if there is no mention of filesystem related problems on the Incidents page.

Addendum: The stale file handle issue is most prevalent on Tinaroo1.
If you logged into tinaroo.rcc.uq.edu.au or tinaroo1.rcc.uq.edu.au, then try logging into tinaroo2.rcc.uq.edu.au directly (see note above about Connecting to HPC User Guide)

Recovering from an accidental deletion in /home

There is a "backup", of sorts, that is done on the /home file system on the Awoonga/FlashLite/Tinaroo HPCs.

The GPFS storage subsystem has a feature called snapshots.
Approximately daily, it will note any differences (to the previous snapshot version) of all the files in your home directory.
It quietly works away each morning before dawn.

For each user, there are multiple "shadow" copies of your home directory and all the files therein.
For any file in your home directory, you should find a timestamped file system entry.
For example, `/home/.snapshots/20190408_0311/uqdgree5/` is a snapshot taken at 03:11AM on April 8 2019.
If I deleted a file from my home directory some time after that point in time, I would be able to "roll back" to that April 8 version.
You own all of your snapshot files so you can copy back over onto the proper home directory the one that you removed or messed up.

The snapshots only go back a month or so, so don't wait too long!

Pre-fetching files from HSM

Due to the finite amount of disk space available for collections, larger files, and files that have not been used for some time, may be evicted from the main disk layer of the UQ RDM and pushed offline to low energy disk or tape media.
The act of accessing any offline file will trigger a recall of the file.
Unfortunately, not all research software seems capable of waiting for the file to be retrieved.
Users sometimes experience file read or input/output errors.

To avoid this sort of disruption to your work, we recommend that you use the `/usr/local/bin/recall_medici` command.
It accepts a single filepath, or a glob (a wildcard).
That command will wait until the offline copy is read back into disk, before proceeding.

Here is an example of it being used.

> Notice that the `ls -salh` output has two columns with filesize information.
> The filesize figure in Column 1 is the size of the data actually stored on (MeDiCI/UQ-RDM) disk layer.
> The filesize figure in Column 6 is the size of the file as expected when it is on the disk.
> If the Column 1 size is zero it means that the file is "offline".
> If the Column 1 size is matches the Column 6 size, it means that the file is "online".
> Sometimes the Column 1 size will be slightly larger than Column 6. This can happen for small files due to rounding to whole disk blocks.
> Whenever a file is offline and required it will be recalled from the hierarchical storage (HSM) subsystem. This step can take some time to complete.

```
uqdgree5@tinaroo1:~> ls -salh /QRISdata/Q0837/demo.bam
32G -rwxr-x--- 1 uqdgree5 qris-uq 32G Nov 14  2018 /QRISdata/Q0837/demo.bam

Now the file has been evicted from the GPFS cache layer by a systems administrator. Check it again now ...

uqdgree5@tinaroo1:~> ls -salh /QRISdata/Q0837/demo.bam
0 -rwxr-x--- 1 uqdgree5 qris-uq 32G Nov 14  2018 /QRISdata/Q0837/demo.bam

So now the file exists in the HSM subsystem but not on MeDiCI disk.

Then use the recall_medici command to trigger the recall of the file from HSM
I am also curious to see how long it takes... About 15 minutes.

uqdgree5@tinaroo1:~> date; /usr/local/bin/recall_medici /QRISdata/Q0837/demo.bam ; date
Thu Nov 28 15:15:09 AEST 2019
```

```
Thu Nov 28 15:30:40 AEST 2019

uqdgree5@tinaroo1:~> ls -salh /QRISdata/Q0837/demo.bam
32G -rwxr-x--- 1 uqdgree5 qris-uq 32G Nov 14  2018 /QRISdata/Q0837/demo.bam

It does not hurt if you run recall_medici on a file that is already on disk.
In fact it should finish very quickly.

uqdgree5@tinaroo1:~> date; /usr/local/bin/recall_medici /QRISdata/Q0837/demo.bam ; date
Thu Nov 28 15:38:02 AEST 2019
Thu Nov 28 15:38:45 AEST 2019

uqdgree5@tinaroo1:~> ls -salh /QRISdata/Q0837/demo.bam
32G -rwxr-x--- 1 uqdgree5 qris-uq 32G Nov 14  2018 /QRISdata/Q0837/demo.bam
uqdgree5@tinaroo1:~>
```

Downloading Data from External Sources

If you have an option for where you pull data from, endeavour to use a source that is **"on-net"**.
The term **"on-net"** means that it does not involve commercial charging rates for the download.
Generally speaking, sites hosted within Australia, Universities and research institutions are on-net.
Because of peering arrangements that AARNet has, it is not necessarily the case that commercial providers are not on-net.
You need to check.

The on-net/off-net status of a download source can be ascertained using a tool provided by AARNet.
It is linked off their Network Operations page

> HOW TO CHECK IF AN IP ADDRESS IS ON-NET
> Enter an IP Address in the Network Address Query Tool

You just need to enter the server name into the tool as data-cbr.csiro.au (without the ftp:// or http:// and file path info).
For example the data server at CSIRO is On-Net

> Checking data-cbr.csiro.au
> 150.229.21.196 is domestic (On-Net) network 150.229.0.0/19 originated by AS6262 via path 6262

Cambridge University is on-net but outside of Australia, obviously:
Checking www.cambridge.ac.uk
131.111.150.22 is international research (On-Net) network 131.111.0.0/16 originated by AS786 via path 20965 786

The Toshiba corporation has two entries, one is on-net the other is off-net.
It would always be preferable to use the **on-net** location where ever possible.

> Checking www.toshiba.com
> 163.171.217.16 is international transit (Off-Net) network 163.171.217.0/24 originated by AS54994 via path 2914 7473 54994
> Checking www.toshiba.com
> 163.171.197.87 is domestic peering (On-Net) network 163.171.197.0/24 originated by AS54994 via path 4826 23686 23686 23686 23686 23686 23686 54994

un-tar to a different place

You don't always have to unpack your tar file into the same directory as the tar file

```
tar -xf archive.tar -C /target/directory
```

So for example if your compressed inputs were on /90days but you want to unpack them into $TMPDIR in a running job then do this

```
tar -zxf /90days/$USER/inputs.tgz -C $TMPDIR
```

Stacking Bricks

You may find that your data overwhelms you (or at least your quota settings).

To assist you with managing your files and folders, there are a few tools available:

- **fpart** - a partitioning tool for files and folders (`module help fpart`)
- **parallel** - a tasking tool that will manage parallel execution of a set of tasks (`module load parallel`)
- **bricks** - in-house tools for directory-aware partitioning (dpart) and HSM-aware copying (hsync).

If you **`module load bricks`**, you will automatically load the fpart and parallel modules.

The `fpart` and `dpart` utilities will create sets of files each containing a list of files and folders.

Those lists of filenames can be fed to `parallel` for processing.
The `hsync` command understands how to work well with the hierarchical storage (HSM).

Collections on the HSM

Some of the QRIScloud collections are housed on our hierarchical storage management (HSM) infrastructure (called Tier 3).

- HSM is the generic name for the technology that combines
  - a small amount of disk space,
  - a large amount of tape storage and
  - software that applies policies to try to keep active files on disk and inactive data on tape.
    Our HSM is based on the Data Migration Facility (DMF) product that was developed by SGI that is now been incorporated into Hewlett Packard Enterprise.
    DMF is one brand of HSM, like a ... is one brand of car.

Carpe Orbis (Seize the Disk)

If you have a QRIScloud collection that is housed on the hierarchical storage infrastructure (Tier 3) you are able to pre-emptively recall your data from the tapes so that the data is already in the disk cache when you need it.
If you don't use a HSM file for a period of time, it will be migrated to multiple tapes and the disk copy removed.
If the disk cache starts to fill up, then the oldest files will be pushed to tapes to make space.

If your collection is housed on HSM, it will look like this when you run the /bin/df -h command.
(the bare `df` command is actually a wrapper that kindly does not show you all of the mounted collections)

```
tinaroo1:~ # /bin/df -h /RDS/Q0275
Filesystem                          Size  Used Avail Use% Mounted on
10.255.120.226:/tier3a1/Q0275/Q0275  120T  116T  4.4T  97% /QRISdata/Q0275
```

The DMF Commands

The commands that are helpful when you want to be more proactive in your use of the HSM are:

- `dmls` allows you to list HSM file attributes as well as regular file system information
  - REG means regular file (on disk only)
  - DUL means dual state (on disk and on tape)
  - OFL means offline (on tape only)
  - MIG/UNM means transitioning MIG-rating to tape or UNM-igrating from tape
- `dmfind` allows you to search for filenames based on HSM attributes (e.g. -state OFL) .
- `dmget FILENAME` will queue up the retrieval of the file from the tape storage and place it on the disk cache.
- `dmput -r FILENAME` this copy a disk file to tapes and will release the disk space ASAP

It is often useful to combine dmfind with dmput or dmget (or other linux commands). See the dmfind manual page for further information and examples.

Why does it take so long to get a file?

Why the HSM can take a long time to retrieve a file...

There are only so many tape drives in our robotic tape silos.
The tape drives switch between reading data and writing data depending on what needs to be done.

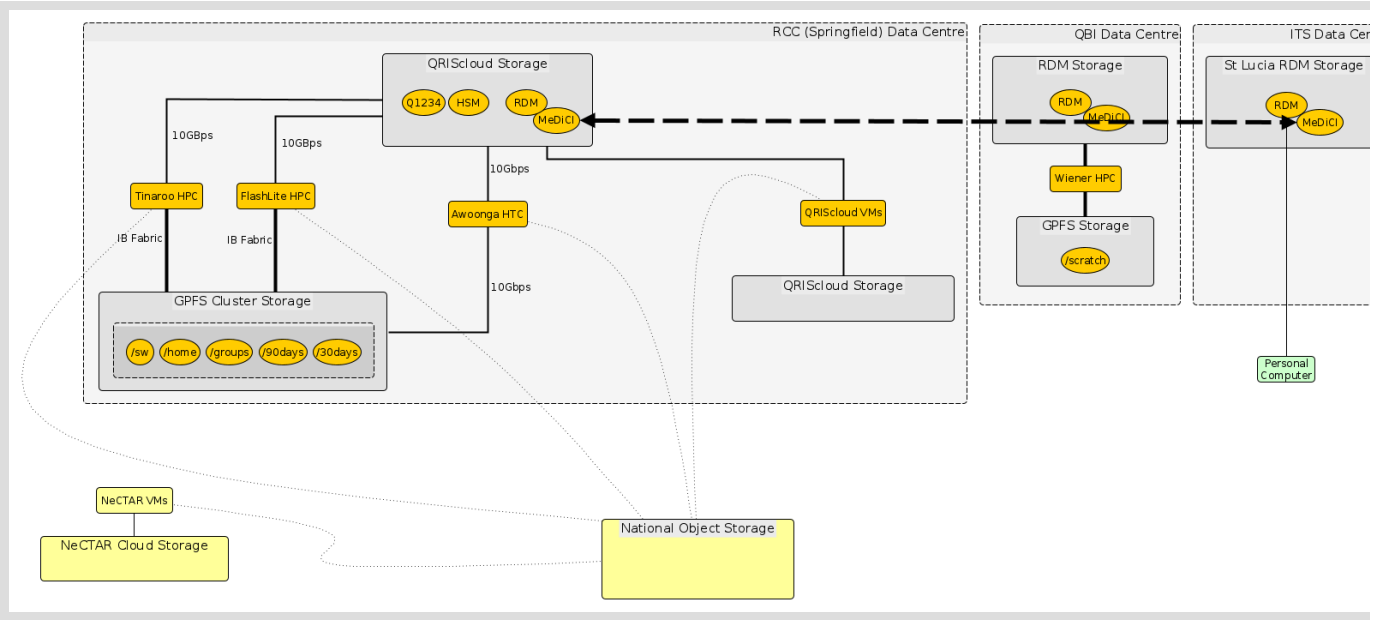
The access to data on tapes via those tape drives is handled by a queue based system.
The queue depth can change by orders of magnitude in short amounts of time.

Lately, there has been an unusually heavy amount of writing to tapes that has needed to be done.
This has meant increased competition for read operations.

If there are a lot requests for reads or writes then it takes longer to recall a specific file.

The best thing you can do to mitigate waiting for HSM data to be recalled is to fetch it ahead of when you need it by using the **`dmget`** command. (See DMF commands section for more information)

Storage Schematic



The PBSPro Batch System

In late 2017, all three clusters (Awoonga, FlashLite and Tinaroo) changed their batch system from Torque to PBSPro.

This was necessary to obtain better performance of the batch system under extreme loads and also to provide a number of diagnostic tools to help us, to help users.

A "wrapper script" has been deployed so that, in most cases, existing Torque job submissions will be understood and translated into PBSPro systax as the job is submitted.
You should work to migrate your job scripts to PBSPro syntax, and can use the `qstat -f JOB_NUMBER` command to learn about the PBSPro syntax corresponding to your submitted torque syntax.

Batch Computing in General

Using batch jobs, managed the batch system, is the preferred way for HPC users to fairly utilise HPC resources.
Because it it shared facility, you are expected to request resources and wait to be allocated those resources.
When the system is very busy, you may need to wait longer.

PBS Pro provides 2 categories of jobs (Interactive and Regular) that are described elsewhere in this document.
In addition, on UQ HPC, the Nimrod workload system is available to assist you with managing ensembles of many jobs.
Nimrod operates in two modes on the HPC, embedded (where the Nimrod subsystem is deployed as part of a regular batch job) and Portal which provides more convenient web access.

The following diagram is an overview of the modes for batch jobs.

```
.
└── Batch_Jobs
    ├── Interactive
    │   └── Use_qsub_-I_Command
    │       ├── 0.Wait_for_Job_to_Start
    │       ├── 1.Load_Software_Module
    │       ├── 2.Start_Using_Software
    │       └── 3.Exit_Job_Using_exit_Command
    ├── Nimrod
    │   ├── Embedded
    │   │   ├── 0.Write_nsh_Job_Script
    │   │   ├── 1.Submit_Job_Script_Using_qsub_Command
    │   │   ├── 2.Nimrod_Manages_Workload
    │   │   └── 3.Wait_for_Job_to_Finish
    │   └── Portal
    │       ├── 0.Access_Nimrod_Portal
    │       ├── 1.Create_Nimrod_Experiment
    │       ├── 2.Start_Experiment_in_Portal
    │       ├── 3.Monitor_Progress_in_Portal
    │       └── 4.Wait_for_Experiment_to_Finish
    └── Regular
        ├── 0.Write_PBS_Job_Script
        ├── 1.Submit_Job_Script_Using_qsub_Command
        ├── 2.Script_does_it_all_unattended
        └── 3.Wait_for_Job_to_Finish
```

Getting Started with the Batch System

Detailed information about PBSPro V 18.2.3 is available in the User Guide
The three clusters all use the same batch system although they have individual batch servers on each cluster.

PBSPro Commands

The manual pages on the system also have information about the various PBS related "q" commands:
qalter, qdel, qhold, qmove, qmsg, qrerun, qrls, qselect, qsig, qstat

Here is a summary of the commands that are most relevant to a HPC user.
All these commands have a man page and you should familiarize yourself with the syntax for using them.

Command	Purpose	Example
qalter	To alter the resource request of a queued job. (see Notes below)	`qalter -l walltime=02:00:00 jobID`
qdel	To delete a queued or running job immediately.	`qdel jobID`
qhold	Places a hold on a job so that it remains in the queue but unable to start. User can hold their jobs.	`qhold jobID`
qmgr	To read (and for admins modify) batch system configuration.	`qmgr -c "list queue Short"`
qmove	To move a queued job to a different queue. Should not need to be used.	`qmove queueName jobID@`
qmsg	To send a text message to a running job that gets written into the O or E file.	`qmsg -E "Are we there yet?" jobID`
qorder	To swap the order of two queued job.	`qorder jobID1 jobID2`
qrerun	To immediately kill and requeue a job.	`qrerun jobID`
qrls	To release a hold previously placed on a job.	`qrls -h u jobID`
qselect	To generate a list of job identifiers based on criteria.	`qstat -a `qselect s R -q Multiple``
qsig	To send a unix system signal to your job. Your job will need to be able to respond to that.	`qsig -s 10 jobID`
qstat	To generate an output of the status of queues or jobs.	`qstat -aw*`
qsub	To submit a job.	`qsub jobscript.pbs`

Command Notes:

- **qalter**
  - A job that is still queued may be altered by the owner.
  - The non-walltime resource settings are fixed once job starts running.
  - A regular user cannot increase walltime once a job has started.
  - HPC admin team can increase the walltime of a running job, however we are unable to do so immediately. Submit a support request at least 48 hours before the job is due to be killed.
- **qrerun**
  - You may wish that your job never restarts automatically when there is a problem, or a qrerun issued (i.e. it should not be rerunnable).
    To ensure this is the case you must mark it as with `#PBS -r n` in your job script, otherwise your job may restart if interrupted.
- **qrls**
  - A regular user can only release a hold that is of the user type (u).
    There are also operator (o) and system (s) holds that may be applied by the admins.
    To see the holds on your job/jobs use the following command pipeline as an example

```
qstat -f | grep -e ^Job -e Hold_Types
```

Additionally there are commands that are only available to the system administrators.

Command	Purpose
qterm	Systems Admin Only- Terminate the batch server.
qdisable/qenable	Systems Admin Only- Controlling queues.

qstart/qstop	Systems Admin Only- Controlling queues.
qrun	Systems Admin Only - To force a job to start.

Wrapped PBSPro Commands

Some of the PBSpro commands are provided with a "wrapper script".
A wrapper script affords additional logic and formatting which enhances the functionality of the "bare" PBSPro commands.
For most users, the wrapper script is encountered earlier in the $PATH than the "bare" version, so the wrapper is what gets run as the command.

Currently, two PBS commands have wrappers

- qstat
  - The command **`/usr/local/bin/qstat`** will produce a comprehensive list of your jobs across all 3 clusters.
  - If you wish to only see the output of the standard PBS Pro qstat command, you should use the "bare" command `/opt/pbs/bin/qstat`
- qsub
  - The wrapper does some syntax checking and fixes any Windows end-of-line character issues

If the wrapped **`/usr/local/bin/qstat`** command does not respond, then there could a problem with accessing the PBS Pro service on one of the other HPC clusters.
If this is the case then you should use the "bare" **`/opt/pbs/bin/qstat`** command to obtain status information for the local cluster you are running the command upon.

Queues, Limits, Nodes

Queues

The default queue (workq) is a routing queue and the workq will route jobs to the appropriate execution queue based on the resource requirements.

Most users will never need to specify the queue for their job, or if they do then use `#PBS -q workq`.

The exception would be if you have been told by RCC Support to use a specific queue as part of Enhanced Access or other some special arrangement that has been made.

The job execution queues that are available on each cluster vary depending on the emphasis of the HPC cluster.

You can find out the names, the basic properties, limits and activity levels on each batch queue by using the command **`qstat -q`**

```
[root@awonmgr2 ~]# /opt/pbs/bin/qstat -q

server: awonmgr2

Queue             Memory CPU Time Walltime Node   Run   Que   Lm  State
----------------- ------ -------- -------- ---- ----- ----- ----  -----
CVL                 12gb     --    168:00:0    1     0     0   --   E R
Short              250gb     --    24:00:00    1   154    70   --   E R
Interact           250gb     --    48:00:00    1     3     1   --   E R
Long               250gb     --    336:00:0    1    34     7   --   E R
Single             250gb     --    168:00:0    1   172   128   --   E R
GPU                 48gb     --    48:00:00    1     0     0   --   E S
DeadEnd              --      --       --     --      0     0   --   E S
Special             60gb     --    48:00:00    1     0     0   --   E R
workq                --      --       --     --      0     0   --   E R
                                                  ----- -----
                                                    363   206
```

Note that queued job array elements do not show up in the Q column count.

You can find out more detail about the level of activity on each batch queue by using the command **`qstat -Q`**

```
[root@awonmgr2 ~]# /opt/pbs/bin/qstat -Q
Queue              Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
CVL                  0     0 yes yes     0     0     0     0     0     0 Exec
Short                0 26540 yes yes    69   152     1     0     0     0 Exec
Interact             0     4 yes yes     1     3     0     0     0     0 Exec
Long                 0    43 yes yes     5    34     2     0     0     0 Exec
Single               0   375 yes yes   127   172     2     0     0     0 Exec
GPU                  0     0 yes  no     0     0     0     0     0     0 Exec
DeadEnd              0     0 yes  no     0     0     0     0     0     0 Exec
Special              0     0 yes yes     0     0     0     0     0     0 Exec
workq                0     0 yes yes     0     0     0     0     0     0 Rout
```

Note that due to a software bug, the total number of jobs (Tot) listed in qstat -Q output is usually bogus.

The number of queued and running jobs is fairly accurate, although job array elements are not counted separately in those statistics.

A job ends up in the DeadEnd queue when it cannot find a home in one of the other valid queues.
When this happens it usually means that your resource requirements are inappropriate and cannot be match by any queue.
Check the queue limits again and adjust your job's resource request.

Queue and Site Limits

Apart from the limits detailed in the `qstat -q` output, there are other limits in place.
These can be examined using the `qmgr` command.

Command	Purpose
`/opt/pbs/bin/qmgr -c "print queue Short"`	Details of a specific queue.
`/opt/pbs/bin/qmgr -c "print node aw009g"`	Details of a specific node.
`/opt/pbs/bin/qmgr -c "print server"`	All details of the server configuration (queues).

See **`man /opt/pbs/share/man/man8/qmgr.8B`** for further details.

The output of the "print server" option includes the site (or complex) wide settings such as

```
set server max_run_res.ncpus = [u:PBS_GENERIC=128]
set server max_run_res.cpu_hrs = [u:PBS_GENERIC=26600]
set server max_array_size = 10000
```

These settings may change to match demand so you should query the current settings if it is important to you.

Nodes Information

The `pbsnodes` command will provide you with information about the state and availability of nodes.
There are several options to the pbsnodes command:

- `pbsnodes -a` shows all nodes in full detail
- `pbsnodes -aS` shows all nodes as a table with one row per node
- `pbsnodes -l` option only shows offline nodes in a tabular format

See `man pbsnodes` for further options.

A node can be in one of the following states

- offline (unable to run any jobs)
- free (has some available capacity for running jobs - not necessarily that it is completely idle)
- job-busy (has no space left for running jobs due to committed CPUs or RAM)

When a node has a specific queue associated with it, it means it will only accept jobs from that queue.
Usually, you would need to eligible to use that queue and node.

Batch Jobs

Batch computing takes a small amount of additional organisation and testing, but the reward can be found in the volume of work that can be achieved with incremental human effort.

Job Submission Script Structure

Batch jobs in PBS Pro can be submitted completely from the command line, however, most people will use a job submission script.

A job submission script has a simple structure with three sections:

	Looks like	Description	Details
1	`#!/bin/bash`	(optional)	shell invocation line if you want to test the submission outside of the batch system
2	`#PBS ...`	the job setup/structure	multiple job specification lines
3	Linux commands/comments	the job payload	multiple lines of Unix commands to execute once the job commences

Note: Any #PBS directives are ignored if they occur after the first unix command in the script file.

Examples of job submission scripts appear elsewhere in this guide.

Transferring PBS Job Scripts from Windows Computers

Be careful if you are editing or saving your PBS script on a Windows computer and transferring it to HPC.

There can be control characters (^M) embedded in the file that are not visible but may cause the batch job to spontaneously fail when it starts to run.
You can avoid this by always transferring the text file using text transfer mode of WinSCP/FileZilla/CyberDuck.
If you want to be sure, you can also use the `dos2unix` command on the HPC to ensure the file is 100% Unix (and PBS) compatible.

GOOD NEWS:
The qsub wrapper script `/usr/local/bin/qsub` will now automatically perform a `dos2unix` on your PBS script before passing it to the batch system.

Batch Job Attributes and Options

The `#PBS` directives are used to specify a variety of attributes of the job.
Most important of these are the job resources which includes the number of "chunks".

One way to think about these chunks of resources is to imagine the job as running within a box (or a number of identical boxes).
The box (or boxes) would have "dimensions" of

- the number of CPU cores (`ncpus=`),
- the amount of RAM required (`mem=`),
- the amount of time the job is allowed to run for before it is forcibly stopped (walltime=).

Different jobs might use a combinations of these dimensions as their chunks. For example,

- A job might have a large number of CPUs and a short walltime- e.g. a single node shared memory (OpenMP) code.
- A job might run on a single CPU and have a long walltime and a large memory footprint- e.g. a genome assembly.
- A job might run on multiple whole nodes and have a moderate walltime- e.g. a CFD or computational chemistry simulation based on message passing (MPI) code.

The number of chunks is governed by the `select=` parameter.
A typical resource request (within a job script) might look like

```
#PBS -l select=1:ncpus=2:mem=4GB
#PBS -l walltime=08:00:00
```

Such a job would be requesting

- 1 chunk with "dimensions" of
- 2 CPUs
- 4 GB of RAM
- 8 hours of walltime

The `select=` parameter is a multiplier for the other non-walltime resources.
That is, `#PBS -l select=5:ncpus=2:mem=4GB` is requesting a total of 10 CPUs and 20GB of RAM

Jobs that are meant to run across multiple cpu cores, or across multiple physical nodes, will have additional parameters to specify (that is, there are additional dimensions to their resource "box(es)" referred to earlier). These parameters will include

- the number of parallel shared memory threads to use (`ompthreads=`)
- the number of message passing threads to run per node (`mpiprocs=`)
  It is perfectly reasonable, in some circumstances, for a job to request a smaller number of mpiprocs or ompthreads than the number of ncpus for that job.

Job submissions may have a number of other parameters to specify

- the job name,
- dependencies on other jobs,
- output options
- notification options, amongst others.

Job options can be set in a script file (using the #PBS) directives or on the command line as options.
Job options provided at the qsub command line will override the corresponding job options in a job script.

Option	Purpose	Default Behavior
`-A`	accounting string for usage reporting	if possible it will work out the value from your group memberships
`-l`	specify the resource requirements (cpu, mem, walltime etc.)	some reasonable defaults are set but a poorly specified job may be rejected
`-N`	job name upto 15 characters	will be the name of the submission script file
`-S`	the Unix shell variant to run the job with	same as your login shell- for most people this is /bin/bash

`-e`	the path (full or relative) for the job's stderr	a file called jobname.*e*JOBID in $PBS_O_WORKDIR
`-o`	the path (full or relative) for the job's stdout	a file called jobname.*o*JOBID in $PBS_O_WORKDIR
`-h`	place a hold on the job as it is submitted so it cannot start until released	
`-J`	specify a job array	you need to provide the range values for the array
`-v`	environment variables and shell functions to be exported to the job	
`-I`	interactive job submission - starts the job and takes you with it !	This option only makes sense at the command line.

The following email notification related options are currently not fully supported on Awoonga/FlashLite/Tinaroo HPCs.
You are unable to email to addresses outside of the HPC.

Option	Purpose	Default Behavior
`-m`	select when to be emailed ... i.e. on **a**(bort) **b**(egin) and/or **e**(nd)	This option will only be valid for email addresses that are internal to the HPC.
`-M`	email address email notifications	This currently has no effect for email addresses that are external to the HPC.

See **qsub manual page** or the PBS Pro User Guide for further details.

How long should the walltime be for my job?

This question is often asked.
The answer is nearly always, measure it, or grossly overestimate it!

We recommend that you do so with the following method

- over-estimate the required walltime (eg. 2 week is **`-l walltime=336:00:00`** is the maximum for most queues on the UQ HPC)
- let a first job(s) finish, use job holds to prevent other jobs from starting
- inspect that job's "e" file or use **`qstat -xf JOBID`** to assess the walltime used (not the cputime)
- submit all similar future jobs using that runtime plus 10% to allow for delays due to storage or network.

You will need to repeat this experimentation if you are using different software or markedly different inputs because the run time may vary considerably.

Note:
If you grossly overestimate your walltimes, your jobs may wait a little longer to get started.
For walltimes greater than 24 hours and upto 168 hours there will be very little difference in the wait time.
It is not possible for a regular user to `qalter` an increase to the walltime of their job once it starts running.
If you underestimate the amount of walltime, your jobs will be terminated once the requested walltime is exceeded.
Please do not expect that RCC staff will always be able to respond to a request to have the walltime of a job extended.
Avoid that situation by following the advice provided above.

A Basic Batch Job

The following job will be called SimpleDemo, will wait on hold until I qrls the job number, then it will run and finish quickly

```
uqdgree5@awoonga1:~/Tests/PBSPro> cat simpledemo.pbs
#!/bin/bash
#
#PBS -A UQ-RCC
#
#PBS -l select=1:ncpus=1:mem=4GB
#PBS -l walltime=01:00:00
#PBS -N SimpleDemo
#PBS -h
#
echo $HOSTNAME
```

```
uqdgree5@awoonga1:~/Tests/PBSPro> qsub simpledemo.pbs
40562.awongmgmr1

uqdgree5@awoonga1:~/Tests/PBSPro> qstat -aw1n

awongmgmr1:
                                                                                  Req'd  Req'd   Elap
Job ID                          Username        Queue           Jobname         SessID   NDS  TSK   Memory Time  S Time
------------------------------ --------------- --------------- --------------- -------- ---- ----- ------ ----- - -----
40562.awongmgmr1                uqdgree5        Short           SimpleDemo           --     1     1    4gb 01:00 H  --   --
uqdgree5@awoonga1:~/Tests/PBSPro>
uqdgree5@awoonga1:~/Tests/PBSPro> qrls 40562.awongmgmr1
uqdgree5@awoonga1:~/Tests/PBSPro> qstat -aw1n
```

Each job will generate an "e" file and an "o" file.
The "o" file contains any output from the commands in the script.
The "e" files contains any error messages as well as a summary of the job's performance.

```
uqdgree5@awoonga1:~/Tests/PBSPro> ls -salrt SimpleDemo.*
0 -rw------- 1 uqdgree5 qris-uq  12 Feb 23 02:13 SimpleDemo.o40562
0 -rw------- 1 uqdgree5 qris-uq 688 Feb 23 02:13 SimpleDemo.e40562

uqdgree5@awoonga1:~/Tests/PBSPro> cat SimpleDemo.o40562
aw031.local

uqdgree5@awoonga1:~/Tests/PBSPro> cat SimpleDemo.e40562
########################## Execution Started ############################
JobId:40562.awongmgmr1
UserName:uqdgree5
GroupName:qris-uq
ExecutionHost:aw031
################################################################################
########################## Job Execution History ############################
JobId:40562.awongmgmr1
UserName:uqdgree5
GroupName:qris-uq
JobName:SimpleDemo
SessionId:123879
ResourcesRequested:mem=4gb,ncpus=1,place=free,walltime=01:00:00
ResourcesUsed:cpupercent=0,cput=00:00:00,mem=0kb,ncpus=1,vmem=0kb,walltime=00:00:00
QueueUsed:Short
AccountString:UQ-RCC
ExitStatus:0
################################################################################
uqdgree5@awoonga1:~/Tests/PBSPro>
```

Note:
There have been problems with the delivery of e and o files at the end of otherwise successful batch jobs because the directory path included spaces or punctuation characters.
It is safer to use _ in place of a space and to avoid punctuation characters in your directory names.

Your Account String

In order to track and report usage on HPC, every user has been allocated to one or more accounting groups.
We need to be able to account for all usage on the cluster to satisfy our stakeholders that their entitlements are being met, and to assist with planning the resource management into the future.

The accounting groups are unix system groups.
Every UQ user belongs to a group like UQ-FACULTY-SCHOOL which reflects your organisational unit.
Additionally, for project work, users may have been allocated one or more project account groups.

- Individuals may be working on multiple projects, or have allocations granted via a number of pathways.
- Groups of users may belong to the same project and need to expend their project allocation collectively.
- Those with specific allocations will need to ensure that they expend their allocations within the granting period.
- We may need to be able to easily and accurately know how much resource was needed to run your jobs and what remains of your allocation or partner share.

A valid account string must be used when submitting jobs.
The job submission mechanism checks your membership of the group you selected and will reject a job that does not use a valid account string.

You can find out what your group memberships (valid account string values) are by using the command `groups`

```
uqdgree5@awoonga1:~> groups
qris-uq support1 flashlite1 Q0030RW Q0224RW Q0286RW Q0030RO UQ-RCC sw-SLIM sw-ORCA
```

You can see that this user is a member of several groups:

- primary group (qris-uq)
- admin support groups (support1 and flashlite1),
- collection storage groups (Q0030RW Q0224RW etc.)
- their main organisational unit group (UQ-RCC)
- and a couple of software access control groups (sw-SLIM and sw-ORCA).
  To submit jobs, always use the account string (group name) that begins with UQ- unless you are submitting project based workloads.

The first group named by the groups command is usually your primary group (unless you actively changed group using the newgrp command). The simplest way to check is to login afresh and issue the groups command.
Not all groups you belong to will necessarily be valid account strings. For example, the group qris-uq is not a valid account string.

To find out the list of all group names and members, issue the following command at the command line:

```
getent group
```

Consider the situation of a UQ researcher who is a member of two groups in additional to their primary group (qris-uq).
One groups is a special project group (ProjectXYZ) and the other is a UQ org unit based group (UQ-FACULTY-SCHOOL).
To submit computations that are under the umbrella of the project use

```
qsub -A ProjectXYZ ...
```

To submit computations that are under the umbrella of their UQ Org Unit use

```
qsub -A UQ-FACULTY-SCHOOL ...
```

Regular (Non-Interactive) Jobs

For a regular batch job, you should combine

- the resource settings and
- job task lines

into a submission script file (e.g. regular_batch_job.pbs)

Job Submission Script

```
#!/bin/bash
#
#This example will request 4 cores and 4x5GB of a compute node on Tinaroo (all nodes 24 CPUs and 120GB of usable memory) for
#PBS -l select=1:ncpus=4:mem=20GB
#PBS -l walltime=02:00:00

#CHANGE THIS TO YOUR UQ-FACULTY-SCHOOL group name.
#USE the groups command to find out your exact group name.
#PBS -A UQ-FACULTY-SCHOOL

#A job always starts running in your home directory $HOME.
echo "The job just started and my present location is ..."
pwd

#To get into the directory where you ran the qsub command for this job you need to
echo "This job was submitted from the PBS_O_WORKDIR directory $PBS_O_WORKDIR "
cd $PBS_O_WORKDIR
pwd

#Each job has a temporary directory created on the compute node running the job.
echo "The TMPDIR for this job is $TMPDIR so change into that directory."
cd $TMPDIR
pwd

echo "Hello World ... what else would we say?!"

echo "Now sleep for 30 minutes so we can check the job using qstat ... "
echo "otherwise it may finish before we have had a chance to look at it with qstat"
sleep 30m

#Refer to the table of job environment variables section of the RCC PBS Pro User Guide
```

Be aware that the batch system stops parsing those `#PBS` lines once it hits a shell command, so put all of your resource request settings towards to the top of the file.
Comments and blank lines are OK just not anything that looks like a command

You **must** change the -A option to a valid account string for you.

Submit Job to Batch System

Then to submit the jobs to the batch system you just need to qsub that script file:

```
qsub regular_batch_job.pbs
```

Checking Job Status

To check the job status, you just need to use the qstat command.

```
uqdgree5@tinaroo1:~/Tests/PBSPro> /opt/pbs/bin/qstat -aws1n

tinmgr2:
                                                                                  Req'd  Req'd   Elap
Job ID                      Username        Queue           Jobname         SessID   NDS  TSK   Memory Time  S Time
```

```
------------------------------ --------------- --------------- --------------- -------- ---- ----- ------ ----- - -----
79103.tinmgr2                  uqdgree5        Short           tsepp                --     1     4   16gb 12:00 H  --   --
    --
99598.tinmgr2                  uqdgree5        Short           regular_batch_j    45896    1     4   20gb 02:00 R 00:00:00 t
   Job run at Thu Sep 05 at 14:35 on (tn425a:ncpus=4:mem=20971520kb)
96608.tinmgr2                  uqdgree5        Training        TrainingTest         --     1     1    5gb 01:00 H  --   --
    --
```

Figuring out WHERE your job is running

The "`-1n`" option to qstat will list the name(s) of the node(s) that are running the job when it is running.

Armed with this information, you can ssh to node(s) that is/are running your job.
In the example above, the node name is **tn425a**.
Those other details ("@/1*4") are referring to cores and their placement on the CPUs and are not part of the server name.

Please note that UQ RCC policies will only allow users to access a HPC compute server who are owners of the actively running jobs on that node.
Connections by "non-stakeholders" may appear to initially succeed however the connection will soon be dropped if you are not the owner of a running job in the compute node you are connecting to.

Figuring out WHAT your job is doing

Sometimes, you may not be sure if your job is making progress.
Sometimes, your job may be running in an imbalanced, or sub-optimal way.
Sometimes, your job may not be running correctly, at all.

It is useful to be able to hop on to the compute node that is processing your work and check that things are working as expected.
Noting the section above about "Figuring out WHERE your job is running", you can access that compute server and run some diagnostics on it.

You would ascertain the name of the node, then, from the login node, **`ssh -x NODENAME`**.
Note, you will only be permitted to do this successfully while you have a job running on a compute node.

You should not need your password to ssh from a login node to a compute node in the same HPC.
You will not be able to ssh from a login node to a compute node on a different HPC.

Note, the values of `PBS_O_WORKDIR` and `TMPDIR` will be undefined, or incorrect, in your ssh session.

Use a command like the following to ascertain the value of the **`PBS_O_WORKDIR`** variable as set in your batch job.
So if your job's $PBS_JOBID is 529717.tinmgr2

```
/opt/pbs/bin/qstat -f -F json 529717.tinmgr2 | grep -e PBS_O_WORKDIR
```

The value of **`TMPDIR`** will be (assuming the job ID is JOBID)

- **`/state/partition1/pbs/tmpdir/pbs.JOBID`** on Awoonga and Tinaroo
- **`/nvme/pbs/tmpdir/pbs.JOBID`** on FlashLite

Once logged in, there are some useful diagnostic commands that include:

Diagnostic	Purpose
`uptime`	Shows 1, 5 and 15 minute average load on the server.
`top -u $USER`	Shows the most active processes that are running and are owned by you.
`pstree -u $USER`	Shows a tree structure of the processes you have running on the server.
`ls -salh directory`	Will show the names and sizes of file(s) in a directory.
`tail -f filename`	Will show you the last lines of a file and update as the file grows.

So, if your job is logging output to a text file (in PBS_O_WORKDIR or TMPDIR), then you could check that its size is growing, and check on its contents and look for trouble.

If your job is merely writing to STDOUT and STDERR,
you can still access those because as the job is running the files that ultimately are returned to you as the job's "o" and "e" files are located on the compute node at the locations like

- `/state/partition1/pbs/spool/JOBID.OU` and `/state/partition1/pbs/spool/JOBID.ER` on Awoonga and Tinaroo
- `/nvme/pbs/spool/JOBID.OU` and `/nvme/pbs/spool/JOBID.ER` on FlashLite

Figuring out WHEN your job is behaving badly

Sometimes a job can overload a single node due IO pressure or because a multitude of processing threads have all been started on that node.
Java and python software, and MATLAB, are observed to do this on occasions because they spawn many threads.

Other situations arise for jobs that are supposed to be running evenly across, say, 4 whole nodes.
Instead a job can have been misconfigured in your job script so that it is either

- performing only 24 threads on the first of the four nodes, or
- performing 4*24 threads in just the first of the four nodes thereby overloading the node and causing the job to run much more slowly than when distributed properly across 4 nodes.

Noting the two sections above, you can ssh to each of the nodes involved with your job and run some basic diagnostics to ascertain whether your multi-node job is well balanced and behaving as it should.

Using shell environment aliases in regular batch jobs

Sometimes a software module will define an alias in the shell environment.
This alias will be a shortcut for

For many of the RCC curated Software Containers, there is an alias defined that will run the container like a single command (e.g. the rjags alias for RJAGS container)

```
#We use an alias as a short cut.
#You need to allow this to work in non-interactive settings.
#Insert this line BEFORE you issue the command, or load the software module, that defines alias(es)

#For BASH shell
shopt -s expand_aliases

#For CSH/TCSH shells
```

Where am I when my batch job runs

The key points in relation to locations when your batch job runs are as follows:

- A batch job always starts running in your home directory **`$HOME`**.
  Think of it as the job needs to ssh into your account on the node that is running your job.
  The value of the environment variable **$PBS_O_HOME** should confirm this location.

- The value of the environment variable **`$PBS_O_WORKDIR`** is always the location you were when you issued the qsub command.
  This may be the location of your input data, or perhaps the directory above your Inputs directory.

- The value of environment variable **`$TMPDIR`** is a unique location that is created for each job (or job array element).
  You own it, and only you can access it, however once the job finishes the **`$TMPDIR`** location is completely removed.
  You must copy anything you wish to keep back from **`$TMPDIR`** before the job finishes.
  Using `cp` or `mv` or `rsync` commands at the end of your batch job script should achieve this for you.

Making Maximum Use of $TMPDIR

All our HPC compute nodes have local (i.e. on-board) disk.
The storage user guide provides information about the disk capacity for each cluster.

It is nearly always preferable to utilize the local disk on a node rather than to be reading and writing furiously across the network.

You should always utilize the local disk `$TMPDIR` location in your batch jobs unless there is a valid reason not to do so.
Some reasons you might give for not using `$TMPDIR` could include:

- My job runs on multiple nodes using MPI so I cannot use `$TMPDIR`. (OK, that is a valid reason but see note below about TMPDIR for multi-node jobs)
- My job needs more space than `$TMPDIR` can provide. (OK, that is a valid reason)
- My job writes barely any output until the end of the job. (OK, that maybe valid)
- I am not interested in making my jobs run better. (soooo not valid)
- Using `$TMPDIR` is too hard. (not valid ... it isn't really)

To make use of `$TMPDIR` you need to think about the following

- where your input data resides so you can fetch a copy,
- how your main task(s) will run and access the local copy of the inputs
- where you would like to deposit your outputs as the job comes to an end.

For the purposes of illustration, it will be assumed that

- you will `qsub` the batch job in a particular directory

- your code to be run is in the sub-directory Codes
- your input files are in a sub-directory called Inputs
- your output files should be returned to a sub-directory called Outputs

There are three important steps to follow to make maximum use of the `$TMPDIR`.

1. Copy your Inputs and Codes into `$TMPDIR` from a sub-directory called Inputs
2. Run your program in `$TMPDIR`
3. Copy your outputs back from `$TMPDIR` to a sub-directory called Outputs

The simplest way to achieve these 3 steps is to

```
#1
cp -rp $PBS_O_WORKDIR/Inputs $TMPDIR
cp -rp $PBS_O_WORKDIR/Codes $TMPDIR

#2
cd $TMPDIR
#Run your code that resides in Codes/
#Make sure your program reads inputs files from current working directory (i.e. $TMPDIR on local disk) or Inputs directory t
#That is, it uses relative paths (or $TMPDIR) instead of reading from /home, /30days/ or /90days/

#3
cp -rp $TMPDIR/Outputs/* $PBS_O_WORKDIR/Outputs/
```

You do not need to remove the contents of `$TMPDIR` because that will be automatically deleted when the job ends.

If you have hidden files or want to copy a subset of things you will need to modify this template.
If you do not have your inputs and outputs separated into folders, you should consider doing so for ease of working.

You might consider using the `rsync` command if you have inputs and outputs and code messed up in the same directory.
This will ensure that only the files that are missing files at the destination get copied.
See the Transferring Files user guide for more about using the `rsync` command.

A Note about TMPDIR for Multi-node (MPI) Jobs

There is a TMPDIR created for the job, but it only happens on the first (rank 0) node of the nodes assigned to the job.
For example,

```
-bash-4.2$ qstat -aw1n 230874.tinmgr2

tinmgr2:
                                                                                  Req'd  Req'd   Elap
Job ID                          Username        Queue           Jobname         SessID   NDS  TSK   Memory Time  S Time
------------------------------ --------------- --------------- --------------- -------- ---- ----- ------ ----- - -----
230874.tinmgr2                  uquser          Multiple        MyJobName          17227    4    96  400gb 168:0 R 23:46:55 t

-bash-4.2$ ssh -x tn417c "ls -sald /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2"
0 drwx------ 2 uquser   qris-uq 6 Jun  4 11:06 /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2

-bash-4.2$ ssh -x tn423d "ls -sald /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2"
ls: cannot access /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2: No such file or directory

-bash-4.2$ ssh -x tn425b "ls -sald /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2"
ls: cannot access /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2: No such file or directory

-bash-4.2$ ssh -x tn221b "ls -sald /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2"
ls: cannot access /state/partition1/pbs/tmpdir/pbs.230874.tinmgr2: No such file or directory
```

So TMPDIR is still usable provided it is only the rank0 process that reads from, or writes to, the TMPDIR on the first node in the list.

Interactive Job Submissions

Batch computing systems support the seemingly contradictory notion of an interactive batch job.
What it is, is a job allocated resources and a compute node destination by the batch scheduler mechanism, but your login session is "transported" to that compute node while you wait.

Please use interactive job submissions for all but the very basic of tasks instead of burdening login nodes and annoying your fellow users!
Using the following command will launch an interactive job with 2 cores and 4GB of RAM for an afternoon of productive code compilation and testing.

```
qsub -I -l select=1:ncpus=2:mem=4GB  -l walltime=03:00:00 -A YourAccountString
```

My personal preference is to use a shell script that I can edit as necessary.

```
uqdgree5@tinaroo1:~> cd $HOME
uqdgree5@tinaroo1:~> chmod 700 bin/interact
```

```
uqdgree5@tinaroo1:~> cat bin/interact
#!/bin/bash
qsub -I -A UQ-RCC -l select=1:ncpus=2:mem=4GB  -l walltime=03:00:00
```

The run it using `$HOME/bin/interact` or if you have your bin directory in your PATH then just type `interact` .

You cannot `qsub -I` a regular PBS batch job script and have it run the commands contained therein.
To achieve something like that, you will need to use the `sleep` trick described below for getting long walltime interactivity.

When you are finished with your interactive session, you can either wait for the wall time to elapse, or you can type the command `exit` at the command prompt and this will cause your connection to the allocated compute node to close, and your session will return to the login node from where you qsub-ed.

X11 Forwarding (-X option)

The `-X` option is optional and it enables X11 forwarding. This would allow you to see graphical output generated on the compute node.
See **`man qsub`** for details.

To use this feature you must have your DISPLAY environment variable set and also have a functioning X11 connection to the login node.
See the Connecting to HPC User Guide for details.

Longer Interactive Sessions or Multi-Node Interactive jobs

Unlike our previous (torque) batch system, jobs submitted with the `qsub -I` can have job attributes that exceed the limits on the Interact queue.
What happens in this situation is that the job will route to another queue- one that can accommodate the job's resource request.
So for example,

- if you require an interactive single node job of 50 hours walltime it will route to the Single queue because the walltime exceeds 48 hours,
- if you require an interactive multi-node job (on Tinaroo or FlashLite) it will route to the Multiple queue because the job selected more than 1 node

If you would prefer not to have to maintain your connection to the login node where you launched the `qsub -I`, there is a relatively simple solution.

The solution involves the following steps:

1. Create a regular batch job that just contains a "sleep" command.
   The job would call a command like `sleep 168h`.
2. Submit the "sleeping job" to the batch system as a regular batch job.
3. Use `qstat -a1n` command to ascertain whether the job has started and if so, which node your job has been assigned to.
4. For a multi-node job, the first named node is what you want to connect with.
5. Then `ssh NODENAME` to the node your sleeping job was allocated. Add the -X option if you need X11.
6. With MPI you may need to adapt your mpirun command so it knows which nodes to use.

The command `cat $PBS_NODEFILE|sort|uniq` will give the unique hostnames.
The $PBS_NODEFILE file may contain multiple entries for the same hostname depending on the resource request made in the sleeper job.

An Important Note about MPI Jobs

Whether your MPI job runs on a single node, or multiple nodes, PBSPro sets up that environment and communicates that environment via the API to the message passing interface framework.
So if, for example, your job request was

```
#PBS -l select=4:ncpus=12:mpiprocs=12:mem=60GB
```

then further in your job script you would probably use an `mpirun` command.

Whenever you use mpirun within a PBSPro job, you should avoid specifying the number of MPI ranks to use.
That is, do **not** include a **`-np 48`** in the mpirun command line in the example MPI job stated above.
Not only is it unnecessary, but it may actually be counterproductive as it may confuse the MPI library.

Just say
`mpirun program-name ...`
and leave it up to the batch system and the MPI library to sort out the "-np stuff".

Tips for Creating More Resilient Job Scripts

If you put to one side the #PBS directives, a PBSPro job script is just a unix shell script.
In its simplest form it is a sequence command lines to be executed (as illustrated in the earlier sections).

However you can, and it could be argued, should, be making your job scripts more resilient so that they can withstand, or better handle, situations that sometimes arise when a job runs. The way to do that is to take advantage of features of the shell environment.
The following is based on the "Bourne Again SH" /bin/bash shell which is what the vast majority of HPC users use as their login shell.

A significant tool for this is the `test` command aka the "[ ]" operator.
I normally use it with as the [] operator within an if statement.
You can negate the test using the ! symbol

If your job needs to work with an input file from RDM, it is useful to

1. check that the directory is available `if [ ! -d path_to_directory ] ; then ...` take action if directory is not mounting
2. check that the file is available `if [ -f full_file_path ] ; then ...` proceed
3. ensure that the file is ondisk `/usr/local/bin/recall_medici full_file_path` refer to storage user guide section about RDM

If your job script contains steps or stages and a latter stage needs an earlier stage to have been completed successfully, you should pay attention to the exit codes.

```
#do something
run_first_thing
#check the exit code
if [ $? -eq 0 ]; then
  run_second_thing
fi
```

OR abbreviated shell syntax to perform them together conditionally

```
#only do second if the first is successful
run_first_thing && run_second_thing

#only do the second if the first is unsuccessful
run_first_thing || run_second_thing
```

Utilising Exit Codes in Job Scripts

Another very useful thing you can do is to catch the various failure modes of your job script and exit your script immediately with an informative exit code.
The job "e" will then record the nature of the failure as the exit status for the job.
Job related exit codes need to positive and non-zero.
An exit code of 0 means the job was successful, and a negative error code will occur if the error was server related.

PBSPro Job Environment Variables

The following table is extracted from the PBSPro Reference Guide Table 13-1: PBS Environment Variables

Environment Variable	Meaning
NCPUS	Number of threads or CPUs
OMP_NUM_THREADS	Same as NCPUS (for shared memory jobs)
PBS_ARRAY_ID	Identifier for job arrays. Consists of sequence number and []
PBS_ARRAY_INDEX	Index number of subjob in job array
PBS_CONF_FILE	Path to pbs.conf
PBS_CPUSET_DEDICATED	Asserts exclusive use of resources in assigned cpuset
PBS_DEFAULT	Name of default PBS server
PBS_DATA_SERVICE_USER	Account used by data service
PBS_ENVIRONMENT	Indicates job type: PBS_BATCH or PBS_INTERACTIVE
PBS_JOBCOOKIE	Unique identifier for inter-MoM job-based communication
PBS_JOBDIR	Pathname of job-specific staging and execution directory
PBS_JOBID	The job identifier assigned to the job or job array by the batch system
PBS_JOBNAME	The job name supplied by the user
PBS_MOMPORT	Port number on which this job's MoMs will communicate

PBS_NODEFILE	The filename containing a list of vnodes assigned to the job
PBS_NODENUM	Logical vnode number of this vnode allocated to the job
PBS_O_HOME	Value of HOME from submission environment
PBS_O_HOST	The host name on which the qsub command was executed
PBS_O_LANG	Value of LANG from submission environment
PBS_O_LOGNAME	Value of LOGNAME from submission environment
PBS_O_MAIL	Value of MAIL from submission environment
PBS_O_PATH	Value of PATH from submission environment
PBS_O_QUEUE	The original queue name to which the job was submitted
PBS_O_SHELL	Value of SHELL from submission environment
PBS_O_SYSTEM	The operating system name where qsub was executed
PBS_O_TZ	Value of TZ from submission environment
PBS_O_WORKDIR	The absolute path of directory where qsub was executed
PBS_QUEUE	The name of the queue from which the job is executed
~~**PBS_SERVER**~~	~~The name of the default PBS server.~~
PBS_TASKNUM	The task (process) number for the job on this vnode
TMPDIR	The job-specific temporary directory for this job

About HPC Software

Overview

Awoonga, FlashLite and Tinaroo are ROCKS clusters.

- Standard operating system commands are installed in standard locations, such as `/bin`, `/usr/bin`
- Some utilities and customizations made by RCC are usually installed into `/usr/local/bin`
- A lot of application software lives on local disk of each compute node.
  - Application software is usually installed in /opt
  - Application gets deployed periodically as part of the cluster node reimaging mechanism.
- ROCKS uses software rolls to manage software
  - The command `rocks list roll` will summarize the deployed rolls.
  - Some rolls (eg. biotools) contain many individual applications.
- Some application software and software development tools are also located on the shared storage point under /sw which is available on all nodes.
- Project and discipline areas are encouraged to compile and maintain their own software if currency is an issue.

The ROCKS Roll Call

Currently the list of specific research software rolls on the HPCs is

```
abyss beagle beast biotools boost cern chemistry cilk fftw geo gnutools hdf jags llvm math mrbayes ncar netcdf octave
performance python R r-modules scipy vtk
```

Note that the `biotools` roll contains many bioinformatics software tools. Use the command **`module display biotools`** to learn more.

The command **`rocks list roll`** can be used to get a complete list of software rolls on Awoonga, FlashLite and Tinaroo.
The list is usually displayed in the order of installation.
Full details about the components of a software roll can be obtained via the SDSC GitHub page

The /sw Software

For a few different reasons software is sometimes installed into the /sw.
You do not need to go digging for it, because officially installed software in /sw will have a environment module, just like the other software.

Your Own Software

You can, of course, build and use your own software on the HPC.
We recommend that you should install it into /90days (or /home or collection storage for longevity).
You can also request access to a directory in `/sw` if you are installing software that is useful to other users (perhaps other research group members).

Software Containers

A relatively recent innovation is the Software Container.

- We support Singularity software containers.
- Containers are usually an entire linux operating system installation (one that is different to the HPC cluster operating system).
- Containers have specific software installed into that operating system.
- Software containers on HPC are usually, but not exclusively, installed by the systems team.
- A software container is treated like a regular piece of software and we have modules, aliases and wrapper scripts to make it as straightforward as possible.
- A regular user can use a singularity software container.
- A regular user cannot create a singularity software container on the HPC, however, they can create it offsite and copy it onto HPC.

The topic of Software Containers is covered in more detail in the Software Container User Guide.

More about Software

- The ~~environment modules mechanism~~ Lmod modules mechanism is the best way to work with installed software on HPCs.
- The environment modules modify your linux environment so you can find the software you want to use and their manual pages and other settings.
- Some background about modules structure on HPCs is provided by a README module
  `module display README/Modules`
- Some other information about the use of modules is available in the **README** and **HOWTO** sections of the modules.
  `module avail README`
  `module avail HOWTO`
- Modules are loaded when the software is required and unloaded or purged when no longer required.
- You can even create and use your own personal module files.
  `module display HOWTO/PersonalModules`
- The Lmod modules system uses a per-user cache file to speed operations up for you.
  You may need to refresh your Lmod cache to be able to access a freshly installed module.
  You can refresh your Lmod cache file by running the command
  `module --ignore-cache avail`

Summary of module command options

- `module avail`
  A list of all available modules.
- `module display ModuleName`
  `module show ModuleName`
  A summary of what the loading the module will do.
- `module help ModuleName`
  A copy of the embedded help in the module.
- `module whatis ModuleName`
  A brief synopsis of the module.
- `module load ModuleName`
  `module add ModuleName`
  Loads/adds the module ModuleName
- `module list`
  A list of previously loaded modules.
- `module unload ModuleName`
  `module rm ModuleName`

Unloads/removes the previously loaded module.

See module help for more information and options.

Usage Examples

- The list of available modules can be requested using

```
module avail
```

  Use the -t option for terse single column output.

- **Not all modules show up when you run the `module avail` command.**
  Some modules that depend on compiler modules are hidden from view until the compiler module has been loaded.
  Other modules have a replica that fails to load whenever the compiler module has not been loaded.

For example, if you need to the BOOST maths library that was built with the same compiler as the software you want to use it with, then you need to first load the compiler module that it was built with (intel or gnu).

To load a module you simply

```
module load Name_of_Module
```

Although the default (or highest version number) module will get loaded if you do not specify a version, **you are strongly advised to always load modules by specific version** so you will know which one you used in, for example, a job submission script.

To see what a module load would do for you

```
module display Name_of_Module
```

To see what help a module provides

```
module help Name_of_Module
```

Not all modules provide extensive help.

Sometimes ... module avail ... fails

Sometimes the `module avail` command will fail in rather inglorious fashion:

```
davidg@flashlite2:~> module avail
/usr/bin/lua: /usr/share/lmod/lmod/libexec/Spider.lua:308: stack overflow
stack traceback:
.
.
.
```

If this happens you should delete your cache directory using the command `rm -rf $HOME/.lmod.d/.cache` and try `module avail` again.
You may need to repeat these steps a couple of times, and perhaps logout and log back in again.

It is sometimes necessary to perform this step on a different login node (same cluster or different cluster).

If it still persists after all that, then please submit a support request via email to **rcc-support@uq.edu.au** as it may be syntax error in a recently added or edited module file.
We will usually check that updated modules work for a regular user logins but sometimes that step can be missed when things are busy.

Using Your Own Software Modules

You are able to write and use your own module files.
This may be useful if you build your own software, or have a special combination of modules you need to use on a regular basis.
It can make things more convenient for setting PATH and LD_LIBRARY_PATH and the like.

See environment modules project website for information: http://modules.sourceforge.net/ for help with syntax.
You could also copy a similar module file from the many that are on the system.
In some circumstances you may wish to link to a special or a not-yet-public module file.

The steps are as follows:

1. Create the directory for your private module files at $HOME/privatemodules
2. Create your module file(s) in that place.
3. Load the use.own module that is in the `/usr/share/Modules/modulefiles` section of `module avail` output.

4. Load your personal module file by its path relative to $HOME/privatemodules

See also `module help use.own`.

A Sample Session

So as a worked example ...

```
uqdgree5@tinaroo1:~> ls -sal privatemodules/embeddednimrod/7
0 lrwxrwxrwx 1 uqdgree5 qris-uq 42 Sep 11 13:25 privatemodules/embeddednimrod/7 -> /gpfs1/sw7/RCC/NimrodG/embedded/modulefil

uqdgree5@tinaroo1:~> module purge

uqdgree5@tinaroo1:~> module load use.own

uqdgree5@tinaroo1:~> module load embeddednimrod/7a

uqdgree5@tinaroo1:~> module list

Currently Loaded Modules:
  1) use.own   2) embeddednimrod/7a
```

A Sample Module File

The module file referred to in the sample session has most of the features you are likely to need.

```
uqdgree5@tinaroo1:~> cat privatemodules/embeddednimrod/7a.lua
--
-- Embedded Nimrod/G Modulefile, Lmod version
--
local nimrod_version = "nimrod-0.8.5"
local base_path = "/gpfs1/sw7/RCC/NimrodG/embedded2"
local nimrod_home = pathJoin(base_path, "opt/nimrod")
local java_home = pathJoin(base_path, "lib/jvm/jdk-10.0.1")
local qpid_home = pathJoin(base_path, "opt/qpid-broker/7.0.4")

whatis("Name: Embedded Nimrod/G")
whatis("Version: "..nimrod_version)
whatis("URL: https://rcc.uq.edu.au/nimrod")

help([[
To run a planfile:
    nimrun /path/to/planfile

To validate a planfile:
    nimrod compile --no-out /path/to/planfile
]])

setenv("JAVA_HOME", java_home)
setenv("NIMROD_HOME", nimrod_home)
setenv("QPID_HOME", qpid_home)

prepend_path("PATH", pathJoin(java_home, "bin"))
prepend_path("PATH", pathJoin(base_path, "bin"))

setenv("OMP_NUM_THREADS", os.getenv("OMP_NUM_THREADS") or "1")
```

The example above is written in the LUA language.
If you want to see more examples of module files, then locate a module file (`module -t avail` will tell you the location of the groups of modules) and use the `cat` command to view the content of the module file.
Be aware that some module files on the HPC are written in TCL/Tk syntax and get translated into LUA language syntax as they are loaded.

Compilers and Related Matters Debuggers

Because a number of the ROCKS software tools have been built for both intel and gnu compilers, you sometimes need to first load your preferred compiler module before loading the module for that software tool.

The list of such software is
`boost hdf4 hdf5 ipm lapack mvapich2 mxml nco openmpi2 openmpi papi parmetis pdt petsc scalapack slepc sprng sundials superlu tau`

Consider a scenario where you are compiling an OpenMPI v2 based code using the Intel compilers.

If you did not first load a compiler module, then you would get this (rather unhelpful) message:

```
uqdgree5@tinaroo1:~> module load openmpi2_ib/2.0.2
Lmod has detected the following error:  Unable to load module:
openmpi2_ib/2.0.2
     /opt/modulefiles/mpi/openmpi2_ib/2.0.2: Non-zero status returned

While processing the following module(s):
```

```
   Module fullname    Module Filename
   ---------------    ---------------
   openmpi2_ib/2.0.2  /opt/modulefiles/mpi/openmpi2_ib/2.0.2
```

however if you first load the compiler, then the openmpi things work as expected.

```
uqdgree5@tinaroo1:~> module load intel
uqdgree5@tinaroo1:~> module list

Currently Loaded Modules:
  1) intel/2018.2.046

uqdgree5@tinaroo1:~> module load openmpi2_ib

uqdgree5@tinaroo1:~> module list

Currently Loaded Modules:
  1) intel/2018.2.046   2) openmpi2_ib/2.0.2
```

```
uqdgree5@tinaroo1:~> module display openmpi2_ib
-------------------------------------------------------------------
   /opt/modulefiles/mpi/openmpi2_ib/2.0.2:
-------------------------------------------------------------------
whatis("openmpi2_ib mpi stack ")
whatis("Version: 2.0.2 ")
whatis("Compiler: gnu intel ")
setenv("MPIHOME","/opt/openmpi2/intel/ib")
prepend_path("PATH","/opt/openmpi2/intel/ib/bin")
prepend_path("LD_LIBRARY_PATH","/opt/openmpi2/intel/ib/lib")
```

PBS Script Snippets for Specific Applications

These sample scripts are intended to enhance the productivity of users of specific applications.
Use at your own risk.

Gaussian

- Gaussian creates checkpoint file(s) as it runs. This involves writing a lot of data to disk intermittently.
- It is best done using local disk on a compute node rather than a network drive.
- Our license for Gaussian is limited to single node computations, so it is well suited to using **`TMPDIR`**.
- The gaussian software module will automatically set the GAUSS_SCRDIR environment variable to the TMPDIR.
- Checkpoint files are written to PBS_O_WORKDIR so you can monitor progress.
- If you know roughly how much disk space a job will require for temporary files that get written into GAUSS_SCRDIR you should request it via PBS.

Job Script Snippets

```
#You can request a minimum amount of free scratch disk for the job start on a node
#PBS -l select=1:ncpus=12:mem=4000MB:scratch=50GB
```

```
module load gaussian/g09
printf "The location of temporary files is going to be %s\n" $GAUSS_SCRDIR
```

Cluster Connection Tools

The following table lists some of the tools available that can be used to connect to the HPCs so you can

- transfer files
- operate command line sessions
- display X11 graphics on your local computer

This list is not an endorsement of any particular tool.
Use at your own risk.

	Windows	**Linux**	**Mac**	**Android**
File Transfers	FileZilla CyberDuck WinSCP psftp	FileZilla CyberDuck scp at the command line gftp	FileZilla CyberDuck scp at the command line RBrowser Fugu (inactive)	Turbo Client Synchronise Ultimate
Command Line Sessions	PuTTY MobaXterm	ssh at the command line PuTTY	ssh at the command line Fugu (inactive)	Serverauditor JuiceSSH
X11 Servers	XMing MobaXterm	X11 is installed by default	X11 is now a separate project called XQuartz	Search for "X11 server" in Play Store

Just search the internet for them by name if any of these links are out-of-date.

Getting More Help

- Software man pages (if available) for all software should be accessible once you load the relevant software module.
- Most user documentation and tools are provided on the cluster via the module mechanism.

```
module avail README
module avail HOWTO
```

- The Intro to HPC training sessions are usually held monthly at UQ St Lucia.
  These are a half day hands-on session for new users.
  It is strongly recommended that all users attend training.
- There are weekly Hacky Hour UQ meetups where you can ask questions face-to-face.
- From time to time, we will hold more advanced training sessions or user discussion forums around particular technologies and tools.
- If you are experiencing difficulties with accessing the HPC clusters, it may be that there is a problem that has already been logged.
  Please check the RCC Service Status page before sending an email to rcc-susupport call.
- If you have an issue that is something that is not covered by attending training, dropping in to Hacky Hour, reading documentation or checking active incidents, then please submit a support request to rcc-support@uq.edu.au
- Please note that our ability to respond to support requests is limited, especially when there are operational difficulties to attend to.
- The building specialist software for individual users, is necessarily a low priority.
- We have provided a fairly complete suite of software and will continue to add to it as workloads permit. Users are encouraged to use the tools provided to compile fresher versions of software they need urgently. The **`module avail`** command will list the entire catalog of software that is covered by the modules mechanism.