HPC Filesystems

Overview of HPC Filesystems

There are a number of filesystems mounted by Pawsey supercomputers:

- /scratch is a large, high-performance filesystem intended for short term use by jobs actively running on the system.
- /group is a mid-tier filesystem intended for actively used files needed for the length of a project.
- /home is a smaller filesystem that should only be used for configuration files that are expected by software to be located there.
- /pawsey is a read-only filesystem where system-wide modules are located.
- /astro is a filesystem that supports the operation of the ASKAP and MWA radio telescopes, refer to The Astronomy Filesystem for more information.

The important differences between these filesystems are summarised in the following table:

Filesystem	User Directory	Variable	Quota	Time Limit	Permissions
/scratch	/scratch/$PAWSEY_PROJECT/$USER	$MYSCRATCH	1 million entries	30 days	750
/group	/group/$PAWSEY_PROJECT/$USER	$MYGROUP	1 TB	project duration	750
/home	/home/$USER	$HOME	1 GB / 10,000 entries	account duration	700
/pawsey	-	-	read only	-	read only

The $PAWSEY_PROJECT environment variable is used for your Pawsey project, if you have access to multiple projects, set the desired value through the file ~/.pawsey_project.

To avoid the need to copy files between systems, the same filesystems are visible on all of the Pawsey supercomputers.

As Pawsey facilities are shared resources accessed by a wide range of users, it is important that appropriate File Permissions are used.

Note that Pawsey staff have access to data on these filesystems, as detailed in Staff Access to Project Directories.

For longer term data storage refer to the Managed Storage documentation.

Scratch Filesystem

Intended for **data files that are actively used by jobs**, the /scratch directory is provided by a high performance 3 PB Lustre parallel filesystem.

Each user of a project has a /scratch directory with the following location:

```
/scratch/$PAWSEY_PROJECT/$USER
```

The $MYSCRATCH environment variable can be used as a shortcut to this directory, for example:

```
cd $MYSCRATCH
```

The /scratch filesystem has the highest performance of the available filesystems, and does not have a capacity quota. This is to allow jobs to temporarily use large amounts of storage while running. However, each user has an inode quota (set at 1 million per user) restricting the number of files/directories they can create, which is to maintain the high performance.

To ensure that /scratch remains available to support jobs actively running on the system, it is critical to move files off the filesystem to a more permanent storage as workflows complete. The copyq on Zeus can be used for this.

⊘ The /scratch filesystem is not intended for long-term storage, and files that remain unused for more than 30 days may be deleted.

For more details, refer to the Scratch Purge Policy.

Leaving files to be removed by the 30 day purge policy places an unnecessary load on the filesystem as the system is scanned for these files, and causes less capacity to be available for other users.

⊘ To minimise load on the filesystem, use the munlink command to delete files.

For more details, refer to Deleting large numbers of files.

Group Filesystem

Intended for **jobscripts, data, and project-specific software installations that are frequently used for the duration of a project**, the /group filesystem is a high performance Lustre filesystem.

Each user of a project has a /group directory with the following location:

```
/group/$PAWSEY_PROJECT/$USER
```

The $MYGROUP environment variable can be used as a shortcut to this directory, for example:

```
cd $MYGROUP
```

By default, projects are provided with 1TB of storage on /group, which is persistent for the duration of the project allocation.

The current quota can be displayed using the pawseyAccountBalance tool in the pawseytools module (loaded by default), for example:

```
pawseyAccountBalance -project $PAWSEY_PROJECT -storage
```

This quota is shared for the project group rather than individual user accounts, and files must belong to the appropriate group to be created in or copied to a project /group directory.

⊘ If you encounter a write error, compiler error, or file transfer error on /group, then it is most likely that this is because the files are counting against your personal group quota rather than your project's group quota.

Refer to Group Permissions for HPC Filesystems for more details.

Home Filesystem

```
/home/$USER
```

Intended for **user-specific configuration files**, the /home directory is provided by a Network File System (NFS).

Each user has a directory in the home file system with a 1 GB or 10,000 inode (files & directories) quota, which is the default location of the command line interface when logging in.

The current usage of the home filesystem can be displayed using the quota command:

```
quota -s
```

The /home filesystem has a significantly lower capacity and performance compared to /scratch and /group, and is **not** suitable for job scripts, job data, or software installations.

It is intended that the /home filesystem is only used to store relatively small numbers of important system and program configuration files that should be located there by convention, such as login profiles and shell configurations.

⊘

Where possible, the /group filesystem should be used instead of /home.

> ✅ It is strongly recommended that software environments (such as module load commands, and setting environment variables) are **not** included in login profiles, but are instead included in the job scripts that are submitted to the scheduler.
>
> This results in more portable and reproducible work flows that can be shared more easily with colleagues and Pawsey staff that may be providing assistance.
>
> See also Shell Initialisation Scripts and Jobscript Reproducibility.

Pawsey Filesystem

The /pawsey filesystem provides **configuration files and system-wide software installations** for all Pawsey users.

```
/pawsey
```

Users should not need to access files in the /pawsey filesystem directly.

Instead, loading system-wide software modules will add the relevant directories to the shell environment.

See the Modules documentation page for more details.