Pawsey File Systems and their Usage

iss

On this page:

Introduction

There are multiple filesystems mounted to each of Pawsey supercomputers. /home and /pawsey which are two Network Filesystems (NFS) mounted on all the Pawsey systems, are connected TCP/IP Ethernet. For higher performance, there are three Lustre filesystems, /group, /scratch and /astro, which are mounted via InfiniBand interconnect and can deliver higher throughput at lower latency between compute nodes and the filesystems. /astro is specific for Radioastronomy use whereas /group and /scratch are for general purpose use.

Here we present a detailed description of these file systems, which you can also find in HPC Filesystems page.

Users can read and write data on /home, /group, /scratch and /astro but /pawsey is a **read only** filesystem where applications are installed. With the exception of /scratch not mounted on Galaxy, all the filesystems are mounted on all Pawsey supercomputers, namely, Magnus, Zeus, Topaz and Galaxy . A user is able to transfer data between them using dedicated data mover nodes. Jobs submitted to SLURM partition called the **copyq** will run the job script on these data mover nodes. Thus from users' perspective, files and directories on a filesystem are accessible from any node of the Pawsey supercomputer.

```
mshaikh@magnus-1:~> df
Filesystem                                          1K-blocks          Used     Available Use% Mounted on
tmpfs                                               264019152       4179788     259839364   2% /run
/dev/sda2                                            30627688      13849640      15199196  48% /
devtmpfs                                            264000564             0     264000564   0% /dev
/dev/sdb1                                          1608252164      35439196    1572812968   3% /var/opt/cray
/persistent
tmpfs                                               264019152           116     264019036   1% /dev/shm
tmpfs                                               264019152             0     264019152   0% /sys/fs/cgroup
146.118.44.32:/vol/ivec                            8160437888    1439168640    6721269248  18% /pawsey
146.118.44.32:/vol/home                           15461882304    2314615744   13147266560  15% /home
10.10.110.17@o2ib4:10.10.110.18@o2ib4:/pgfs     3344840229600 2065737207888 1110464276312  66% /group
10.10.36.128@o2ib4:10.10.36.129@o2ib4:/astrofs  1980913738624 1575715871240  305326225932  84% /astro
```

```
mshaikh@zeus-1:~> df
Filesystem                                          1K-blocks          Used     Available Use% Mounted on
devtmpfs                                            132017456             8     132017448   1% /dev
tmpfs                                               132028560      24812336     107216224  19% /dev/shm
tmpfs                                               132028560       1692012     130336548   2% /run
tmpfs                                               132028560             0     132028560   0% /sys/fs/cgroup
/dev/mapper/LVMDisk-Root                            104806400       6096920      98709480   6% /
/dev/sda1                                             1002316        107288        842448  12% /boot
/dev/mapper/LVMDisk-Var                             104806400       3632440     101173960   4% /var
/dev/mapper/LVMDisk-Opt                             209612800        415412     209197388   1% /opt
/dev/mapper/LVMDisk-Tmp                             419225600        579784     418645816   1% /tmp
146.118.44.32:/vol/home                           15461882304    2314616384   13147265920  15% /home
146.118.44.32:/vol/ivec                            8160437888    1439167744    6721270144  18% /pawsey
10.10.100.23@o2ib1:10.10.100.24@o2ib1:/snx11038 3341599881984 1263553775264 2039979442220  39% /scratch
10.10.110.17@o2ib4:10.10.110.18@o2ib4:/pgfs     3344840229600 2065736492756 1110459539560  66% /group
10.10.36.128@o2ib4:10.10.36.129@o2ib4:/astrofs  1980913738624 1575704619268  305337462756  84% /astro
```

```
mshaikh@galaxy-1:~> df
Filesystem                                         1K-blocks          Used     Available Use% Mounted on
tmpfs                                              264022444       1947828     262074616   1% /run
/dev/sda2                                           30627688       9459196      19589640  33% /
devtmpfs                                           264003816             8     264003808   1% /dev
/dev/sda3                                           22938040         45080      21704712   1% /var/crash
/dev/sda4                                             475736         70313        376342  16% /boot
/dev/sda6                                          492020640        851076     466153196   1% /tmp
/dev/sdb1                                         1022384388      40919708     981464680   5% /var/opt/cray
/persistent
tmpfs                                              264022444         32788     263989656   1% /dev/shm
tmpfs                                              264022444             0     264022444   0% /sys/fs/cgroup
146.118.44.32:/vol/home                          15461882304    2314412736   13147469568  15% /home
146.118.44.32:/vol/ivec                           8160437888    1439168448    6721269440  18% /pawsey
10.10.110.17@o2ib4:10.10.110.18@o2ib4:/pgfs    3344840229600 2061771925392 1114424866424  65% /group
10.10.36.128@o2ib4:10.10.36.129@o2ib4:/astrofs 1980913738624 1575835278832  305207115792  84% /astro
```

As shown above, the same /home, /group, /scratch and /astro filesystems are mounted on both Magnus and Zeus. All but /scratch are mounted on Galaxy as shown in the last block. This is because Galaxy is a dedicated platform for Radioastronomy research. **/astro is a Lustre filesystem dedicated to a real time processing of the radioastronomy data**.

The four filesystems are different in many ways and are designed to facilitate different activities in supercomputing. The intended usage for each of them is explained below. Use outside of these purposes is subject to poor performance to a particular activity as well as detrimental impact to other users.

Home File System

```
/home/[username]
```

Each user has a default login directory which is in the /home file system.
Each user has a default quota of 1 GB, and 1000 individual files.
The location of the default login directory can always be found by examining the environment variable $HOME.

It is intended that the home file system is used to store relatively small numbers of important system files such as your Linux profile, shell configuration etc.

Current usage of the home file system can be found by using

```
magnus-1:~> quota -s
```

Owing to its small quota limit and low performance, the home filesystem is **not** suitable for launching / storing production work. Files such as jobs, executables, input data, and batch scripts should be stored in the group file system. Job output should use the /scratch file system.

Each user has a uniform view of a single home directory across Pawsey Centre machines.

Group File System

```
/group/[project]/[username]
```

Each project has a directory /group/[project] in which each project member has a subdirectory /group/[project]/[username] allocated in the group file system. The default allocation for each project (not each member) is 1 TB.
More can be allocated upon justified request.

The location can be found by examining the environment variable $MYGROUP.

The group file system is intended for storage of executables, input datasets, important output data, and so on, for the life time of the project.

All members of a project have read and write access to the /group/[project] directory, so it can be used for sharing files within a project. Your allocation of space on /group lasts for the duration of the project - it is not subject to any automatic purging.

Quotas are managed per project group. If any member of the project exceeds the shared project quota on /group, it will affect the whole project and will be unable to save data (you may see a 'quota exceeded' message')

/group is a Lustre file system and has a much higher throughput than /home. The quota can be queried using the following command:

```
magnus-1:~> pawseyAccountBalance -storage
...
Storage Information
-------------------
/group usage for project123, used = 899.54 GiB, quota = 1024.00 GiB
```

The group file system is not backed up.

We reccommend that users use /scratch for the best performance for running jobs, and save any precious data in their /group directory.

Scratch File System

```
/scratch/[project]/[username]
```

Each project has a directory /scratch/[project] in which each project member has a subdirectory /scratch/[project]/[username].

/scratch is a Lustre filesystem and its location can be found by examining the environment variable $MYSCRATCH.

Prior to January 2019, /scratch was not subject to quotas, so a large amount of space was available. Since then, Pawsey have now limited the inode quota to 1000000.

It is intended for temporary storage related to production runs in progress.

The scratch file system **is not intended for long-term storage**: it is not backed-up and is purged on a regular basis.This means files on $MYSCRATCH which have not been accessed for the purge period will be deleted automatically, and WILL BE LOST - see Scratch Purge Policy. If you wish to retain files, they should be moved to $MYGROUP.

/scratch faster than /group so *it's the primary area where you should set up jobs to run*. This could require copying data from other filesystems onto /scratch before running the job, and similarly copying files back to /group after the job.

File Permissions and Quota

The effect of file permissions and ownership on storage quotas vary depending on which filesystem data is located. The **default** behavior can be summarized as such:

- Files/directories created in a user's **home** are only accessible by the user.
- Files/directories created in a user's **group** or temporary (**scratch** or **astro**) are only accessible to the user and members of the same project.

/home quotas are based on file ownership, not group owernship. **/scratch** has an inode quota of 1000000, and /**astro** have no quotas, and free space is managed by the purge policy mentioned above.

As shown in the example below, default group membership when creating files and directories in /home is the users ID whereas on Lustre filesystem any file created is by default associated to the user's project ID. For the **group** filesystem, Pawsey uses a file's group ownership to calculate storage quotas. As mentioned before, the default quota for a project on /group is 1TB. Therefore, only the files with group association as the project ID will be able to make use of the group quota.

`A file created in /home filesystem`

```
mshaikh@magnus-1:~> touch new_file
mshaikh@magnus-1:~> ls -l new_file
-rw-r--r-- 1 mshaikh mshaikh 0 Aug 30 17:12 new_file
```

`A file created in /group filesystem`

```
mshaikh@magnus-1:/group/pawsey0001/mshaikh> pwd
/group/pawsey0001/mshaikh
mshaikh@magnus-1:/group/pawsey0001/mshaikh> touch new_file_group
mshaikh@magnus-1:/group/pawsey0001/mshaikh> ls -l new_file_group
-rw-r--r-- 1 mshaikh pawsey0001 0 Aug 30 17:13 new_file_group
```

This is important to know because a user can be member of more than one projects and is always a member of the group namely its own username (mshaikh in the example above). Files created with group associated to "username" are limited to have a default quota of 1GB and there can be at most 100 of them.

If you encounter a write error, compiler error, or file transfer error on /**group** filesystem, 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.

File permissions are also important to consider. Here are the default permissions of a file (myscript.sh) created in the home directory of user bskjerven:

```
bskjerven@magnus-1 ~ $ ls -ld myscript.sh
-rwxr-xr-x 1 bskjerven bskjerven 0 Jul  7 08:57 myscript.sh
```

Recall that Linux file permissions are broken down into groups of three:

- `rwx`
  - The first set of permissions corresponds to the owner's permissions. In this case `bskjerven` is the owner, and is allowed to read (r), write (w), and execute the file (x).
- r-x
  - The second set of permissions corresponds to the group's permission. The group here is the same as the username (bskjerven). Group members are allowed to read and execute.
- r-x
  - The final set of permissions are all other users' permissions. While the permissions are set to read and execute, the top-level user directory (/home/bskjerven) is locked to just the user, so no others are able to read, write, or execute files in another user's home directory.

Now look at the difference of a file created in group:

```
bskjerven@magnus-1 /group/pawsey0001/bskjerven $ ls -ld my_group_script.sh
-rwxr-xr-x 1 bskjerven pawsey0001 0 Jul  7 09:13 my_group_script.sh
```

The file permissions are the same as before, but with a different group ownership (pawsey0001). Other members of pawsey0001 will be able to read and execute this script. Similar to first script, all other users' permissions are set to read and execute, but the top level group directory (/group/pawsey0001) is locked to just the group so that others not in the group cannot access any files within it:

```
bskjerven@magnus-1 /group/pawsey0001/bskjerven $ ls -ld /group/pawsey0001
drwxrws--- 46 root pawsey0001 4096 Jul  4 14:20 /group/pawsey0001
```

Note there is a new flag in the group permissions, the SETGID flag (s). With the SETGID flag set on the directory, whenever a user creates a new file under /group/[projectID], the group ownership is set to the same as the group owner of the directory, as opposed to setting the group ownership to the user who created it. So, in the example above, any file created under /group/pawsey0001 will have a group ownership of pawsey0001 instead of bskjerven.

The SETGID flag on your project's **group** directory is set when Pawsey staff first set up the new project so there's no need for users to modify this. However, there are situations where a user might accidentally modify permissions or ownership when moving files. For example, if a user moves a file from /home to /group (instead of copying it) the group ownership is not changed:

```
bskjerven@magnus-1 ~ $ touch foo.txt
bskjerven@magnus-1 ~ $ ls -ld foo.txt
-rw-r--r-- 1 bskjerven bskjerven 0 Jul  7 09:31 foo.txt
bskjerven@magnus-1 ~ $ mv foo.txt $MYGROUP
bskjerven@magnus-1 ~ $ ls -ld $MYGROUP/foo.txt
-rw-r--r-- 1 bskjerven bskjerven 0 Jul  7 09:31 /group/pawsey0001/bskjerven/foo.txt
```

In the example above a file "foo.txt" was created in directory on /home. As a result, the group ownership is set to user's group (bskjerven). The file was then moved it to the /group filesystem, and you can see that the original permissions and group ID remained. The file foo.txt will count against the user's 1GB home quota, even though it is located in /group.

The solution is to use the copy command (cp) instead of move (mv) when transferring files from /home to /group. The reason is because cp actually creates a new file, which inherits the SETGID flag from the top-level group directory:

```
bskjerven@magnus-1 ~ $ touch bar.txt
bskjerven@magnus-1 ~ $ ls -ld bar.txt
-rw-r--r-- 1 bskjerven bskjerven 0 Jul  7 09:31 bar.txt
bskjerven@magnus-1 ~ $ cp bar.txt $MYGROUP
bskjerven@magnus-1 ~ $ ls -ld $MYGROUP/bar.txt
-rw-r--r-- 1 bskjerven pawsey0001 0 Jul  7 09:31 /group/pawsey0001/bskjerven/bar.txt
```

When transferring files from **scratch** to **group**, you will see the above behaviour and require the same workaround. When using cp, do not use the -a or -p flags. if you want to preserve timestamps, use 'cp --preserve=timestamps'.

File transfer programs like WinSCP can also cause issues with permissions and groups. You should consult the documentation of your preferred transfer program. rsync users should avoid using the '-a' and '-p' flags; these flags will preserve permissions of the source files, which may conflict with the default behavior on Pawsey systems. Some additional information about file transfer programs is at: File Systems, File Transfers and File Management.

Pawsey has provided a tool that allows you to fix a file and directory permissions on **group**. The script **fix.group.permission.sh** is available in the pawseytools module (which is loaded by default). To use it simply type

```
fix.group.permission.sh ProjectGroupID
```

where ProjectGroupID is your project ID (e.g. pawsey0001, director1234, etc.). Note that this will only fix files and directories owned by the user executing the command (i.e. $USER), and will only work on the directory tree in /group/[ProjectGroupID]. Please be aware that this may take some time to complete, and that you can only run one instance of the script at a time.

A quick way of doing this in your own /group area is:

```
find /group/ProjectGroupID/Username ! -group ProjectGroupID -exec chgrp ProjectGroupID \{} \;
find /group/ProjectGroupID/Username -type d ! -perm /g=s -exec chmod g+s \{} \;
```

The extra tests for the 'find' command above speed up the process for many files, by only changing files/directories that need to be changed.

Lustre File System

Lustre is a high performance parallel file system provided by [Whamcloud]. The filesystem uses multiple servers to store data and metadata, improving throughput. A Lustre filesystem can be accessible by all nodes in a cluster. Lustre provides high throughput for large data transfers, however it can perform very poorly for frequent small I/O. Take this into account when writing programs to do I/O.

To check your quota on a Lustre filesystem use

```
lfs quota –g project_code /group
lfs quota –g project_code /scratch
```

Astro File System

```
/astro/[project]/[username]
```

The Astonomy Filesystem /astro is a lustre filesystem provided for the needs of temporary storage of the ASKAP and MWA groups who perform computations on the Galaxy cluster. It is an SGI/HPe provided cluster of nodes backed by DDN storage. The system currently contains 2 Metadata servers (MDS) with 2 Metadata targets (MDT), one each for the metadata of ASKAP and MWA files. It has 4 Object Store servers (OSS) for storing data and they have 32 Object Store targets (OST) which are organised into pools so the two radio astronomy groups can have dedicated access to resources. This gives approximately 1.9 PB of usable storage. It has a possible read and write speed of over 10GB/s and Pawsey staff have been easily getting 7-8GB/s when using only the four copyq nodes to transfer data around using dcp.

The expandability of lustre means that the filesystem can be expanded, without downtime, by adding more OSS's and OST Disk behind them in groups of 2 (for high availability).

Using /astro

The Astronomy filesystem is mounted on all Galaxy nodes, data moving nodes and ASKAP ingest nodes at /astro. The top level directory has directories for all the areas:

```
galaxy-1:/astro # ls -l /astro/
total 28
drwxr-x--- 4 root askap      4096 May 19 10:11 askap
drwxr-x--- 2 root askaprt    4096 May  8 13:27 askaprt
drwxr-x--- 2 root askap      4096 May  8 13:28 casda
drwxr-x--- 2 root mwaeor     4096 May  8 13:28 mwaeor
drwxr-x--- 3 root mwaops     4096 May  9 10:35 mwaops
drwxr-x--- 3 root mwasci     4096 May 18 23:04 mwasci
drwxr-x--- 2 root pawsey0001 4096 May  8 13:29 pawsey0001
```

Each of the ASKAP directories use the ASKAP MDT (MDT0) and are set to write their data to the dedicated ASKAP OST Pool (askappool). Similarly each of the MWA directories use the MWA MDT (MDT1) and are set to write their data to the dedicated MWA OST Pool (mwapool). The pawsey0001 directory is for Pawsey testing of the system and can be set up in different ways as needed, it will not often be used.

Quotas

Each group has access to their own OST pool which limits the amount of data that they can write to half of the filesystem each. However there are also additional quotas that can be applied to assist in data management. At the time of writing MWA have requested that mwaeor, mwaops and mwasci are assigned 300TB each and ASKAP have not asked for any further quotaing but this is subject to change as and when requested by the groups.

To check the current quota use:

```
lfs quota -g projectcode /astro
```

Usage

To check usage you can use the normal unix df command to check the entire filesystem. But if you want just your pool usage you also have access to lustre commands to give you that.

```
galaxy-1:/astro # lfs df -h --pool mwapool /astro/
UUID                       bytes        Used   Available Use% Mounted on
astrofs-MDT0000_UUID      542.1G      469.2M      505.0G   0% /astro[MDT:0]
astrofs-MDT0001_UUID      542.1G      619.0M      504.8G   0% /astro[MDT:1]
astrofs-OST0010_UUID       57.7T        1.4T       53.3T   3% /astro[OST:16]
astrofs-OST0011_UUID       57.7T        1.2T       53.5T   2% /astro[OST:17]
astrofs-OST0012_UUID       57.7T        1.4T       53.4T   3% /astro[OST:18]
astrofs-OST0013_UUID       57.7T        1.8T       53.0T   3% /astro[OST:19]
astrofs-OST0014_UUID       57.7T        1.9T       52.8T   3% /astro[OST:20]
astrofs-OST0015_UUID       57.7T        1.1T       53.6T   2% /astro[OST:21]
astrofs-OST0016_UUID       57.7T        1.3T       53.4T   2% /astro[OST:22]
astrofs-OST0017_UUID       57.7T        1.5T       53.2T   3% /astro[OST:23]
astrofs-OST0018_UUID       57.7T        1.5T       53.2T   3% /astro[OST:24]
astrofs-OST0019_UUID       57.7T     1012.7G       53.8T   2% /astro[OST:25]
astrofs-OST001a_UUID       57.7T      848.9G       53.9T   2% /astro[OST:26]
astrofs-OST001b_UUID       57.7T        1.6T       53.2T   3% /astro[OST:27]
astrofs-OST001c_UUID       57.7T        2.2T       52.6T   4% /astro[OST:28]
astrofs-OST001d_UUID       57.7T      944.9G       53.8T   2% /astro[OST:29]
astrofs-OST001e_UUID       57.7T        1.2T       53.5T   2% /astro[OST:30]
astrofs-OST001f_UUID       57.7T        1.5T       53.2T   3% /astro[OST:31]

filesystem summary:       922.4T       22.4T      853.5T   3% /astro
```

This gives a breakdown by OST and a summary at the bottom.