Transferring files

On this page:

A first recommendation: use the correct file system

Copy your files into the correct location

Remember that Pawsey systems have different file systems. Therefore, it is important that users put their files into the correct location. In general, user's personal folders in:

- /home should be left to operating system related matters
- /group should be used for installing and developing tools and for use case of persistent data storage
- /scratch and /astro should be used for running your cases. All transient (intermediate, input and output) data should be there temporarily. Results should be analyzed immediately and deleted afterwards. If files need to be kept, they should be transferred to your institution storage and removed from /scratch or /astro. If data will be used persistently it can then be stored in /group or HSM (if applicable).

Refer to HPC Filesystems for more information.

A second recommendation: for a large number of files use **tar** to package them into a single file first

Transfer tar files instead of thousands of raw files

Transferring clients spend a lot of time figuring out the list of files to be transferred and checking the correctness of each individual transfer. When the transfer involves a large amount of files and directories (for example, the result files of multiple OpenFOAM runs), this "listing and checking" time adds up dramatically. Therefore, it is always much more convenient to use the command **tar** to first pack the files that need be transferred into a single tar file (or into a very reduced number of tar files). Then transfer the tar file(s) and untar afterwards if needed.

A first warning: use the Datamover nodes

For transfering large amounts of data

NEVER USE THE LOGIN NODES for transferring large amounts of data. Users should use the datamover nodes for this purpose. These nodes are accessible through the generic hostname **hpc-data.pawsey.org.au** or through the "copyq" queue for batch processing through scheduler.

As these nodes are separate from the login nodes, the load for transferring large files or a large amount of data will not impact other users. All standard filesystems, such as /group, /scratch, /astro and /home, are accessible from these data-mover nodes.

A second warning: if the "group" property of transferred files/directories is not set as your project

If the "group" property has been overriden

By default, your personal directories in the file systems /scratch and /group (easily accessible with the environmental variables $MYSCRATCH and $MYGROUP) are set to belong to: "yourUsername-yourProject" in the "owner-group" properties. This combination of ownership properties has been defined through set-gid. Also by default, all the files and subdirectories created under your personal directories in /group and /scratch will inherit the same ownership properties: i.e. should belong to: "yourUsername-yourProject".

Unfortunately, some file transfer programs override our set-gid defaults for ownership properties. The typical unfortunate change is to set "yourUsername-yourUsername" instead of "yourUsername-yourProject" for the "owner-group" properties (see example below). This makes the files abide by the quota restrictions as set on /home and won't be able to make use of extended quotas as on /group, /scratch or /astro. You may receive errors during an upload if this is the case.

To resolve this problem, the "group" part of the ownership properties needs to be fixed. Secondly, the setup of the transferring tool needs to be fixed for avoiding the problem to happen again.

Consider the following example where the directory "badDirectory" resulted with the wrong settings after being transferred. The ownership properties of the directory show "espinosa-espinosa" instead of "espinosa-pawsey0001" which is the right "owner-group" combination of ownership properties. The rest of the directories listed have the correct properties. (The command "ls -la" was used to observe the files/directories properties.):

```
espinosa@zeus-1:/group/pawsey0001/espinosa/userSupport> ls -la
total 20
drwxrws---+  5 espinosa pawsey0001 4096 Jul 30 16:00 .
drwxr-s---+ 17 espinosa pawsey0001 4096 Jul 25 09:25 ..
drwxrws---+  2 espinosa espinosa   4096 Jul 30 16:00 badDirectory
drwxrws---+  2 espinosa pawsey0001 4096 Jul 30 15:52 borra
drwxrws---+  4 espinosa pawsey0001 4096 Jul 11 15:53 GS-8819-ONETEP
```

To resolve the problem, execute the **chgrp** command (change group) with the "-R" option to apply the change recursively through all the content (files and subdirectories) inside the affected initial directory. The general syntax is:

```
chgrp -R yourProject yourAffectedDirectory
```

This fixes that group association of the directory named badDirectory and all the files in it recursively:

```
espinosa@zeus-1:/group/pawsey0001/espinosa/userSupport> chgrp -R pawsey0001 badDirectory
espinosa@zeus-1:/group/pawsey0001/espinosa/userSupport> ls -la
total 20
drwxrws---+  5 espinosa pawsey0001 4096 Jul 30 16:00 .
drwxr-s---+ 17 espinosa pawsey0001 4096 Jul 25 09:25 ..
drwxrws---+  2 espinosa pawsey0001 4096 Jul 30 16:00 badDirectory
drwxrws---+  2 espinosa pawsey0001 4096 Jul 30 15:52 borra
drwxrws---+  4 espinosa pawsey0001 4096 Jul 11 15:53 GS-8819-ONETEP
```

If the problem has expanded extensively into most of your /group directory, you may need to use the tool "fix.group.permission.sh" provided by the module "pawseytools". More info about this can be found in the "File Permissions and Quota" of the documentation entry: Pawsey File Systems and their Usage.

Finally, in order to avoid this problem to happen again, the user needs to set up the configuration of their file transfer program to honour the set-gid (set-group identification default) so that newly created files and directories belong to your project on the "group" property. This will be explained for several documented tools within the following subsections.

Command-line Clients

The transfer of files to or from Pawsey systems is performed over encrypted data streams. The Secure Copy Protocol (SCP) and the Secure File Transfer Protocol (SFTP), which are based on the Secure SHell protocol (SSH), are installed in our systems and can send data over a normal ssh connection. Most of the tools we recommend in this page make use of this protocols.

Command line clients are a convenient way of moving data between computers. They are usually easy to include in a script, so can be a part of an automated workflow.

scp

scp stands for "Secure Copy Protocol". Linux and macOS include the **scp** command, easily available in a terminal or xterm. MobaXterm and Cygwin on Windows also include it. **scp** has been built on top of **ssh**, so if a ssh connection can be established, then a scp connection should work too.

scp is useful to copy few small files and not a lot or very large files. It is not recommended for the transfer of large amounts of data, as it can't resume transfers if the operation/connection is interrupted for any reason.

ssh is enabled on Pawsey systems for both incoming and outgoing traffic. This however may not be true for some firewalls on connections on client side. Most university, business and home internet connections only permit outgoing connections, and have their incoming ssh disabled within their firewall. This means that, scp is always invoked on client i.e. your Laptop/Desktop to copy the data to/from the Pawsey supercomputers.

The generic syntax is **scp** command is:

```
scp [options] SourceFileOrDir USER@HOST:DestFileOrDir
scp [options] USER@HOST:SourceFileOrDir DestFileOrDir
```

where the name of the "SourceFileOrDir" includes both the path and name of the files/directories to transfer. The "DestFileOrDir" (destination) also includes the path (and possibly the "new" name) of the destination file or directory.

For example, to transfer the file "/myDir/myfile.dat" in your local computer to the user's personal directory in "/scratch":

```
scp /myDir/myfile.dat espinosa@hpc-data.pawsey.org.au:/scratch/pawsey0001/espinosa
```

In this case "espinosa" is the user and "pawsey0001" is their project.

To copy all the files with extension "*.data" from the user's personal directory in "/group" to the local machine into the current directory (indicated by a dot "."):

```
scp espinosa@hpc-data.pawsey.org.au:/group/pawsey0001/espinosa/*.data .
```

Again, "espinosa" is the user and "pawsey0001" is their project.

A source directory can be copied recursively with the "-r" option. For example, to copy the whole directory "myScripts" and its contents into the user's personal directory in "/home":

```
scp -r ./myScripts espinosa@hpc-data.pawsey.org.au:/home/espinosa
```

Again, "espinosa" is the user.

If the transfer/connection fails

If connection fails, **scp** cannot be used to resume the file transfer from the interruption point although **rsync** may be used for that.

sftp

sftp stands for "Secure File Transfer Protocol" (not be confused with FTPs, which is FTP over SSL). **sftp** is an interactive transfer client that uses SSH to create a secure connection to the server. Its functionality is very similar to **ftp**, but not all of the ftp options are available.

Open and Close a connection to Pawsey

In order to establish a connection to Pawsey execute:

```
myLocalComputer:> sftp USER@hpc-data.pawsey.org.au
sftp>
```

Note that after establishing a connection, the prompt will change to "sftp>" indicating that the interactive sftp session has started.

In order to close the connection execute **bye**:

```
sftp> bye
```

Navigate in the local and remote systems

As for any linux interactive session, the basic navigation tool for moving among directories in the remote file system is **cd**:

```
sftp> cd RemoteDesiredDirectory
```

and you can check which is the current directory in the remote server with **pwd**, and its with **ls**:

```
sftp> pwd
Remote working directory: /scratch/pawsey0001/espinosa
sftp> ls
Nek5000            OpenFOAM           babadu             testingTransfers   userSupport
```

Within the sftp interactive session, you can also navigate in your local computer by using the prefix "l" as for "local" in the commands:

```
sftp> lpwd
Local working directory: /Users/esp025
sftp> lls
Applications        Downloads        Movies            Tools
Boostnote        Dropbox            Music            borra
Desktop            Library            Pictures        mac:esp025:espinosa
Documents        Maciej            Public            mnt
sftp> lcd Downloads
```

It is always a bit tricky to remember to use lcd,lls and lpwd to navigate in the local system. Therefore, it is recommended to navigate in your local system before establishing the sftp connection. In that way, your current directory in the local computer will be the desired local directory for file transfers and you may not need to navigate from there anymore.

Copy files into Pawsey file systems

Once an interactive **sftp** connection to the data mover system at Pawsey has been stablished and the current directory in the local and remote systems are the desired ones, users can put files into the remote system by executing the **put** command inside the interactive sftp session:

```
sftp> put [options] SourceFileOrDirInLocalSystem [DestFileOrDirInRemoteSystem]
```

For example, to put the file "myFile.dat" which is current working directory of the local computer (and which is going to be transferred to the current working directory of the remote computer):

```
sftp> put myFile.dat
```

As the general syntax suggests, there is still freedom to choose source path/files and destination path/file names. The following example takes another file from other directory and puts it into the personal directory of the user in /group, even if the current remote directory was in /scratch:

```
sftp> pwd
Remote working directory: /scratch/pawsey0001/espinosa
sftp> put /otherDir/otherFile.dat /group/pawsey0001/espinosa/someOtherFile.dat
```

As for **scp**, in order to put entire directories, the option "-r" needs to be used. In this case the directory "myScripts" will be transferred to the personal directory of the user in /home:

```
sftp> put -r ./myScripts /home/espinosa
```

Copy files from Pawsey file systems

The sftp command for copying data into the local file system is **get**. Its general syntax is:

```
sftp> get [options] SourceFileOrDirInRemoteSystem [DestFileOrDirInLocalSystem]
```

Basically, **get** operates in the contrary direction of **put**, but its syntax is exactly the same**.**

rsync

rsync is a utility for efficiently transferring and synchronizing files across computer systems, by checking the timestamp and size of files. Linux and macOS include the **rsync** command, easily available in a terminal or xterm. MobaXterm and Cygwin on Windows also include **rsync**. Similar to **scp** `(see above)`, **`rsync`** requires the specification of a source and of a destination; either of them may be remote, but not both.

rsync is more robust that **scp** in the sense that it can resume transfers after failure, and it can also be used for synchronizing and backing up. We recommend its use for preparing scripts that can be reused for transferring data.

The generic syntax is very similar to scp (see subsection above) :

```
rsync [options] SourceFileOrDir USER@HOST:DestFileOrDir
rsync [options] USER@HOST:SourceFileOrDir DestFileOrDir
```

where the name of the "SourceFileOrDir" includes both the path and name of the files/directories to transfer. The "DestFileOrDir" (destination) also includes the path (and possibly the "new" name) of the destination file or directory.

rsync has a lot of different options that can be consulted elsewhere. These are our recommendations:

```
rsync -vhsrl --chmod=Dg+s -e ssh ....
```

Do not use -a option for transferring files into our systems

Before listing the meaning of these options, it is worth to notice that, contrary to the common recommendations available on the internet, we do not recommend the use of the option "-a", specially for transferring files into our systems. This is because the behaviour of "-a" can override the default set-gid settings of the "group" property (see subsection named "A second warning" above).

The purpose of the recommended options is:

Option	Purpose
-e ssh	runs rsync over ssh
--chmod=Dg+s	forces all directories to get marked by the default set-gid
-v	this turns on verbosity, so that written messages about the progress of the transfer are displayed
-h	this turns on human readable numbers for the sizes of the transferred data diplayed by the verbosity option
-s	this allows for strange names of files (with spaces or strange characters) to be interpreted as part of the name (although we do not recommend to use spaces or strange characters in your filenames)
-r	turns on recursivity, so if a directory has been chosen to be transferred, then all the contents of the directory will be transferred
-l	when symlinks are encountered, recreate the symlink on the destination

Then, for example, for transfering the directory "~/myDir" and all its contents into the user's personal directory on /scratch:

```
rsync -vhsrl --chmod=Dg+s -e ssh ~/myDir espinosa@hpc-data.pawsey.org.au:/scratch/pawsey0001/espinosa
```

where "espinosa" is the user and "pawsey0001" is their project. The syntax for Source and Destination are the same as for **scp** (see subsection above for more examples).

Use preservation of times with care

Rsync has an option, -t, that activates the preservation of modification times of the files into the destination system. This can be useful for rsync choosing not to transfer files that already exist in the destination system with same modification dates.

However, this should not be used when transferring files to the /scratch filesystem, where a 30-day purge policy is in place (see Scratch Purge Policy). Using the -t option to transfer to /scratch files that have not been accessed for more than 30 days will result in the deletion of those files by the purge policy, and then data loss.

bbcp

bbcp is a point-to-point network file transfers application with particularly high network transfer rates. bbcp can be very useful especially when transferring large files (few GB and more).

bbcp documentation can be found here: http://www.slac.stanford.edu/~abh/bbcp/

bbcp is currently available on Zeus copyq as a module:

```
module load bbcp
```

There are 2 components required on your local machine to perform data transfers to/from Pawsey systems using bbcp:

1. SSH client, for most Unix-like systems (Linux//MacOS/Cygwin), the command ssh is sufficient.
2. bbcp installation in your local machine (download bbcp).

Two examples on how to use bbcp to transfer data between Pawsey system and local machine are given below.

Transferring from Pawsey data transfer nodes to local machine

```
module load bbcp
bbcp --port 40000:41000 -T "ssh -x -a -oFallBackToRsh=no %I -l %U %H PATH_TO_BBCP_ON_LOCAL_MACHINE/bbcp" file.
dat username@localsystem:/directory_on_local_system/
```

The *-T* option in the above command points to the bbcp binary on the local machine.

PATH_TO_BBCP_ON_LOCAL_MACHINE should be replaced with location (directory) of the bbcp binary on the local system.

file.dat is the name of the file located on Pawsey system.

username@localsystem:/directory_on_local_system/ points to the local machine (username,localsystem pair) and destination directory for the file.

Transferring to Pawsey data transfer nodes from local machine

```
bbcp -S "ssh -x -a -oFallBackToRsh=no %I -l %U %H /pawsey/sles12sp3/tools/sandybridge/gcc/4.8.5/bbcp/17.
12.00.00.0/bin/bbcp" file.dat username@hpc-data.pawsey.org.au:/scratch/PAWSEY_PROJECT/USERNAME/
```

The -S option in the above command points to the bbcp binary on the Pawsey system.

Please be aware that the path to bbcp on Zeus (/pawsey/sles12sp3/tools/sandybridge/gcc/4.8.5/bbcp/17.12.00.00.0/bin/bbcp) might change after system upgrades.

Known issues

Adding -z option to bbcp (*bbcp -z*) might be useful in case of the following error message:

```
bbcp: Accept timed out on port 5031
bbcp: Unable to allocate more than 0 of 4 data streams.
Killed by signal 15.
```

GridFTP (globus-url-copy)

This tool can be installed on Linux and macOS and on Cygwin on Windows. The data-mover nodes have GridFTP installed and configured to accept sshftp:// URLs (performing authentication using SSH rather than X509 Certificates).

The generic syntax is:

```
globus-url-copy [options] file://SourceFileOrDir sshftp://USER@HOST/DestFileOrDir
globus-url-copy [options] sshftp://USER@HOST/SourceFileOrDir file://DestFileOrDir
```

Note that SourceFileOrDir and DestFileOrDir may contain themselves a first "/" indicating that the paths are being referred from the root of the filesystems. This then results in some triple "///" and double "//" in the actual execution of the command that are not evident in the general syntax. This can be more clearly seen in the examples below.

If you are moving large data files, globus-url-copy provides an alternative to scp which can be faster, although transfer of the data itself may be insecure. Use is similar to that of scp in that both ends of the transaction must support the globus-url-copy protocol. Note that globes-url-copy requires absolute paths to be specified, and no wildcards are available (It can therefore be convenient to execute copies of more than one or two files via a batch script with ssh keys enabled; see below). Please note that the GLOBUS_PORT_RANGE for Pawsey is defined as 40000-41000

The recommended options for globus-url-copy are described in the following table.

Option	Purpose
`-tcp bs` ***`size`***	Set the TCP buffer size (bytes). To achieve top performance, this should be set to an appropriate value. However, an "appropriate value" depends on a number of factors which are not generally easy to anticipate such as the actual bandwidth, and the round-trip time for a packet. The value suggested gives reasonable performance in tests.
`-bs` ***`size`***	Set the buffer size (bytes) for the transfers (use the same value as for `-tcp-bs`)
`-p` ***`nstreams`***	Set the number of parallel streams (4 is reckoned to be a good starting point)
`-vb`	Verbose output

Then, for example, for transfering the fie "/home/myDir/myFile.dat" into the user's personal directory on /group:

```
globus-url-copy -tcp-bs 16M -bs 16M -p 4 -vb file:///home/myDir/myFile.dat sshftp://espinosa@hpc-data.pawsey.
org.au//group/pawsey0001/espinosa/myFile.dat
```

globus-url-copy can also be executed from the data-mover nodes. But for this you need to load the module "globus"

```
hpc-data1:~> module load globus
hpc-data1:~> globus-url-copy -tcp-bs 16M -bs 16M -p 4 -vb file:///home/user/file.dat sshftp://user@remote.dns.
addr//home/user/remote-file.dat
```

This will require standard password authentication at the remote site unless ssh keys have been enabled.

GUI clients

GUI clients for transferring files are a very attractive option for users because of their intuitive framework. They have the advantage that do not need to remember the several different options for the command-line tools. Although they are still based on the command-line clients listed above. In practice, the combined usage of both GUI and command-line clients within your workflows results in better efficiency.

Always pay attention the source and destination directories

Note that most GUI clients will start in your home directory when you first connect to a remote server, while some will start in your previous directory. This is almost never where you need to put the data. In most cases you will need to browse to your own scratch or group area.

Filezilla

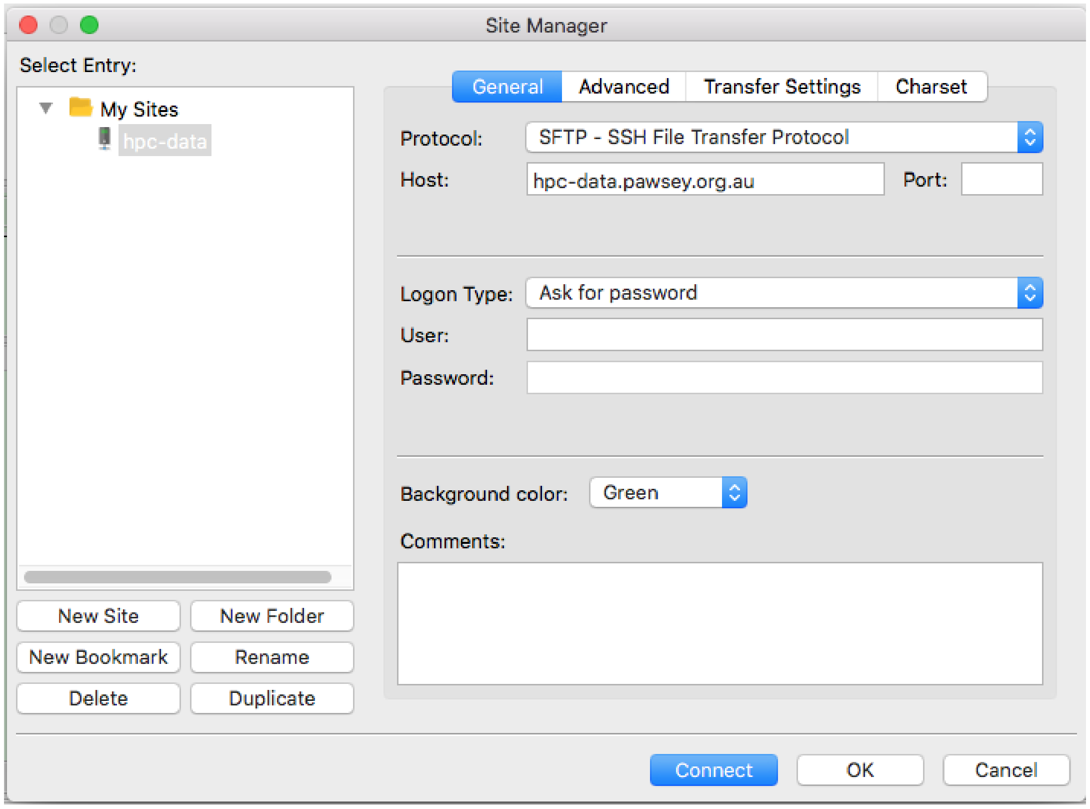
Filezilla is a fast and reliable file transfer client with an intuitive GUI. It works on multiple platforms (Windows, macOS and Linux) and supports **SFTP**, which is one of the supported protocols for transfering files to/from Pawsey systems. Filezilla supports simultaneous transfer of multiple files, transfer of large files (>4GB) and transfer resume after connection failure.

Avoid spyware

Only download Filezilla from its own website: https://filezilla-project.org/index.php . As for any software, be careful of not falling into "click tricks" that mislead you to download or install undesired software.

Basic setup

- Step 1: In the File menu, select Site Manager. A window for the "Site Manager" settings will pop-up.
- Step 2: In the lower left corner of the window, select "New Site". Name the site as you want (hpc-data is the name used here).
- Step 3: In the Protocol entry, select **SFTP** as the protocol. In the Host entry, type **hpc-data.pawsey.org.au**
- Step 4: In the Logon Type as "Ask for password".


Protocol: SFTP - SSH File Transfer Protocol
Host: hpc-data.pawsey.org.au
Port:
Logon Type: Ask for password
User:
Password:
Background color: Green
Comments:

- Step 5: Click "Connect". Use your username and your password for accessing Pawsey systems.

Do not save your password in Filezilla

Even if Filezilla can remember passwords we do not recommend this practice. The use of ssh-keys is safer. It is also less prone to multiple failed connection attempts, which may eventually block your account at Pawsey (see Account Blocking).

Use of ssh-keys

The use of ssh-keys is very practical, as users can avoid the need of typing their password at every connection.

In order to use this feature, users first need to generate SSH-Keys for connecting into Pawsey systems (as explained in Logging in with SSH keys). After generating the keys, users need to follow the instructions of the basic setup (above), but instead of choosing "Ask for password" in step 4, choose Logon Type as "Key file". Then add the path and name of the ssh-key file that you have generated (for example, in my case, /Users/espinosa/.ssh /pawsey_rsa_key). Finally Filezilla will ask for the passphrase of that key and convert it to a .ppk format, which is compatible with Filezilla itself.

Multiple simultaneous transfers

Filezilla supports simultaneous transfer of multiple files, so should be faster than WinSCP or command-line scp if you have many files to transfer. Multiple streams is most useful over long distances, in which case 12 may be useful.

Nevertheless, too many streams will create load on the data-mover node, and may reduce performance (and impact others). Over short distances, up to 4 streams may help improve performance while not creating too much load.

Users can limit the maximum simultaneous connections in the "Transfer Settings" section of the Site Manager settings window shown above.

Automation using scripts

To our knowledge, filezilla does not allow for automatic transfers through the use of scripts. (See https://superuser.com/questions/239860/how-do-i-send-a-file-with-filezilla-from-the-command-line). But you can always prepare scripts that make use of command-line clients (explained above) or use other tools that allow this (like WinSCP below).

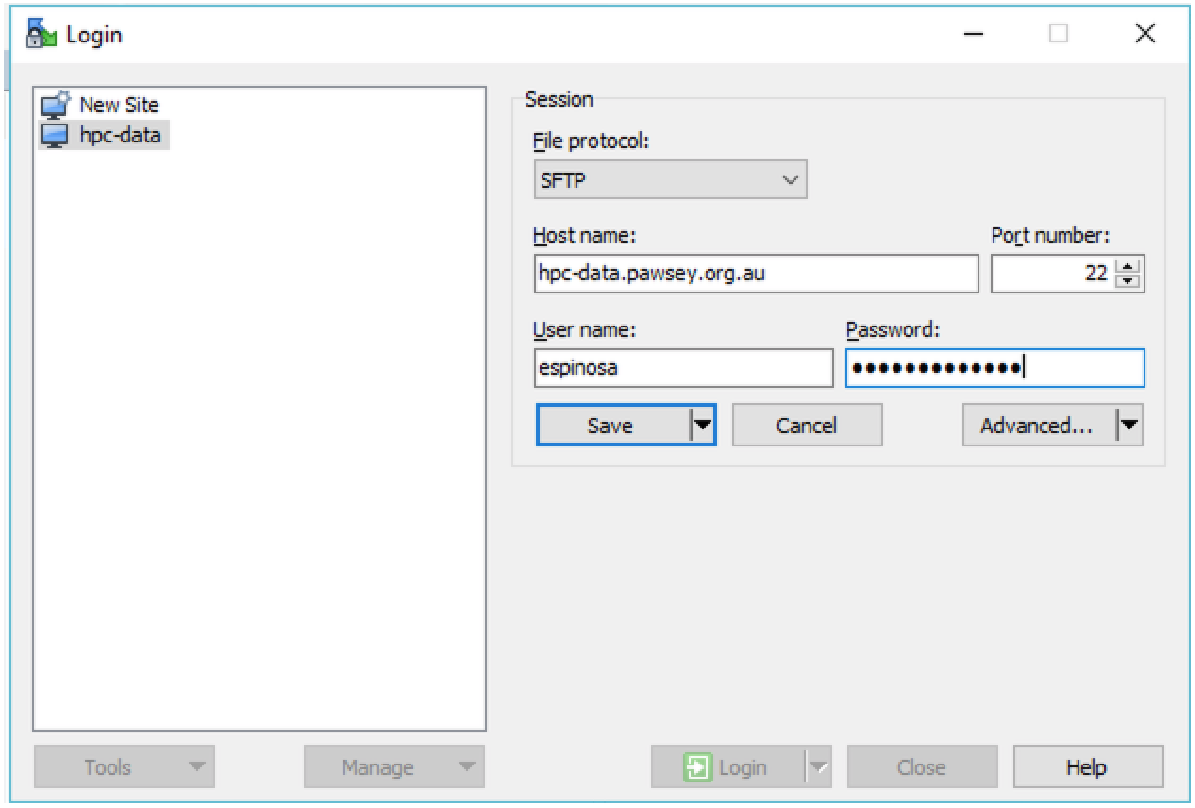
WinSCP

WinSCP is an open source free file transfer client for Windows. It has an intuitive GUI and supports **SFTP** and **SCP** protocols among others. Its main function is file transfer between a local and a remote computer. Beyond this, WinSCP offers scripting and basic file manager functionality. We recommend to use the SFTP protocol (default) as it allows to resume transfers after a connection interruption.

Only download WinSCP from its own website: https://winscp.net/eng/index.php . As for any software, be careful of not falling into "click tricks" that mislead you to download or install undesired software.

Basic setup

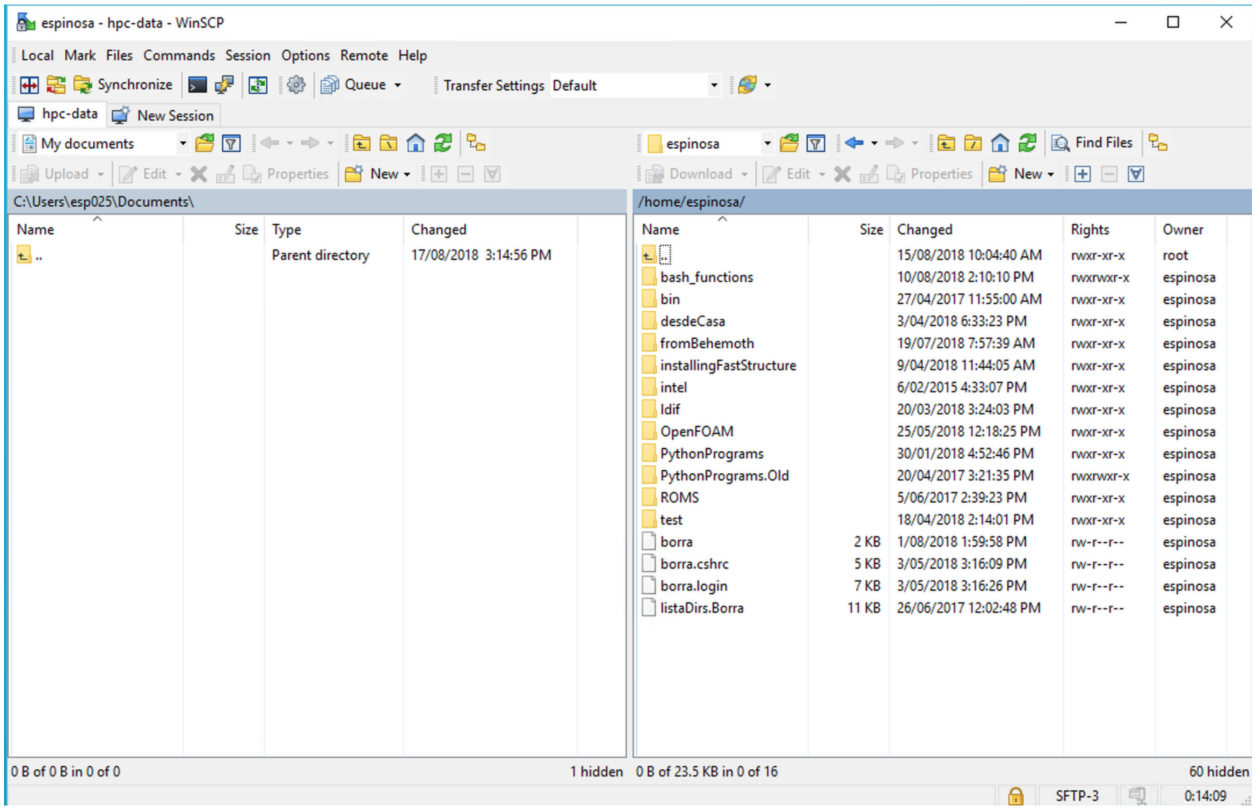
- Step 1: Open a New Login Setup window. If it is not opened by default, then from the Session menu, choose "New Session".
- Step 2: On the left panel choose "New Site" and on the bottom menu "Manage" (a bit on the left) choose "save as" and define a name for the connection (hpc-data is the name we are using here).
- Step 3: Use **hpc-data.pawsey.org.au** as the hostname, your username and your password. (Do not save the password in WinSCP).
- Step 4: Click save and then Login.



File protocol: SFTP
Host name: hpc-data.pawsey.org.au
Port number: 22
User name: espinosa
Password: •••••••••••••

Basic usage

After a connection has been established, the current directories in the local and remote filesystems are listed. You can navigate intuitively within the file systems by clicking into subdirectories or by going up one level using the "Parent directory" icon ( ). Several other navigating icons exist:


espinosa - hpc-data - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
hpc-data
New Session
My documents
Upload
Edit
Properties
New
espinosa
Find Files
Download

C:\Users\esp025\Documents\

Name	Size	Type	Changed
..		Parent directory	17/08/2018 3:14:56 PM

0 B of 0 B in 0 of 0 — 1 hidden

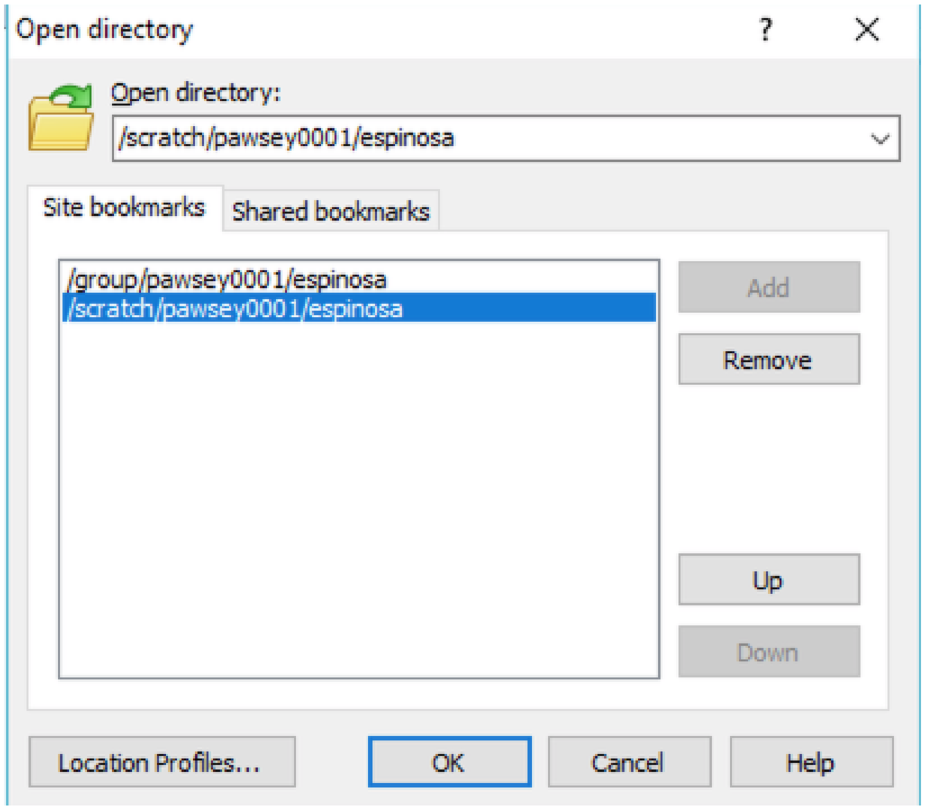
/home/espinosa/

Name	Size	Changed	Rights	Owner
..		15/08/2018 10:04:40 AM	rwxr-xr-x	root
bash_functions		10/08/2018 2:10:10 PM	rwxrwxr-x	espinosa
bin		27/04/2017 11:55:00 AM	rwxr-xr-x	espinosa
desdeCasa		3/04/2018 6:33:23 PM	rwxr-xr-x	espinosa
fromBehemoth		19/07/2018 7:57:39 AM	rwxr-xr-x	espinosa
installingFastStructure		9/04/2018 11:44:05 AM	rwxr-xr-x	espinosa
intel		6/02/2015 4:33:07 PM	rwxr-xr-x	espinosa
Idif		20/03/2018 3:24:03 PM	rwxr-xr-x	espinosa
OpenFOAM		25/05/2018 12:18:25 PM	rwxr-xr-x	espinosa
PythonPrograms		30/01/2018 4:52:46 PM	rwxr-xr-x	espinosa
PythonPrograms.Old		20/04/2017 3:21:35 PM	rwxrwxr-x	espinosa
ROMS		5/06/2017 2:39:23 PM	rwxr-xr-x	espinosa
test		18/04/2018 2:14:01 PM	rwxr-xr-x	espinosa
borra	2 KB	1/08/2018 1:59:58 PM	rw-r--r--	espinosa
borra.cshrc	5 KB	3/05/2018 3:16:09 PM	rw-r--r--	espinosa
borra.login	7 KB	3/05/2018 3:16:26 PM	rw-r--r--	espinosa
listaDirs.Borra	11 KB	26/06/2017 12:02:48 PM	rw-r--r--	espinosa

0 B of 23.5 KB in 0 of 16 — 60 hidden

SFTP-3 0:14:09

But this basic navigation may not be optimal for Pawsey filesystems. A more practical and faster way is to go directly to the directory we want to work with. For that, double click on the current directory path on the line shaded with grey colour (in this case, the line that has "/home/espinosa/"). A pop-up window will appear and there you can type directly the desired path (for example /scratch/[project]/[user]) and even save it as a bookmark for later use:



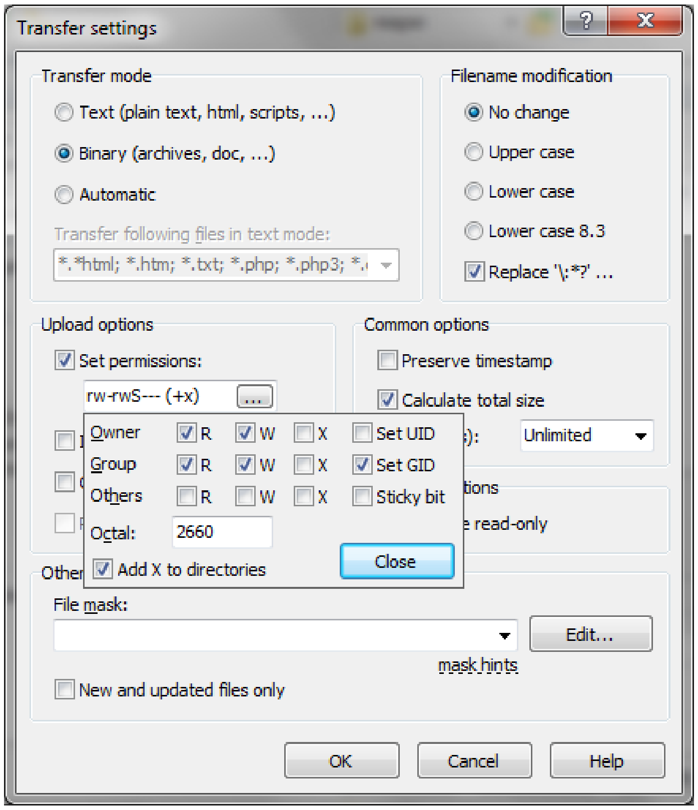
Open directory: /scratch/pawsey0001/espinosa

/group/pawsey0001/espinosa
/scratch/pawsey0001/espinosa

When finished, choose "Disconnect" within the Session Menu.

File permissions settings

You should set default permissions for transferred files. See the WinSCP UI permissions documentation for how to access the settings. See the below image for recommended settings, which gives read/write access to others in your project, and no access to anyone outside your project.



Transfer settings

Transfer mode
- ○ Text (plain text, html, scripts, ...)
- ● Binary (archives, doc, ...)
- ○ Automatic

Transfer following files in text mode: *.*html; *.htm; *.txt; *.php; *.php3; *.

Filename modification
- ● No change
- ○ Upper case
- ○ Lower case
- ○ Lower case 8.3
- ✓ Replace '\:*?' ...

Upload options
- ✓ Set permissions: rw-rwS--- (+x)

Owner	✓ R	✓ W	☐ X	☐ Set UID
Group	✓ R	✓ W	☐ X	✓ Set GID
Others	☐ R	☐ W	☐ X	☐ Sticky bit

Octal: 2660

✓ Add X to directories

Close

Common options
- ☐ Preserve timestamp
- ✓ Calculate total size

Unlimited

read-only

File mask:

Edit...

mask hints

☐ New and updated files only

OK Cancel Help

Use preservation of times with care

WinSCP has a setting to preserve timestamps (i.e. modification times) of the files in the destination system.

This should not be used when transferring files to the /scratch filesystem, where a 30-day purge policy is in place (see Scratch Purge Policy). Using the -t option to transfer to /scratch files that have not been accessed for more than 30 days will result in the deletion of those files by the purge policy, and then data loss.

Cyberduck

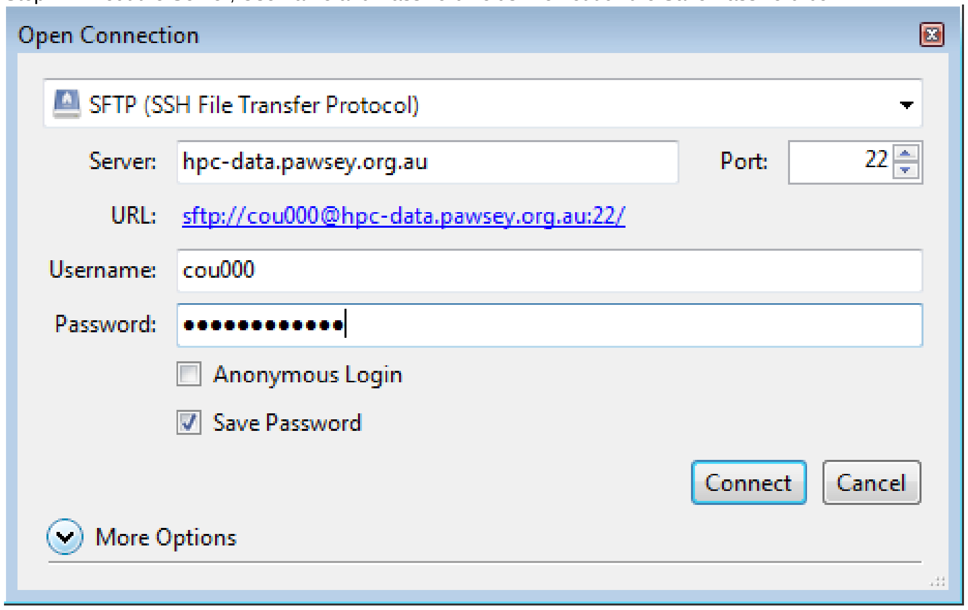
Cyberduck is described as "libre server" and cloud storage browser for macOS and Windows. It supports **SFTP** among many other protocols and cloud platforms. Cyberduck has a GUI although it appears less user-friendly than Filezilla or WinSCP.

Only download Cyberduck from its own website: https://cyberduck.io/. As for any software, be careful of not falling into "click tricks" that mislead you to download or install undesired software.

Basic setup

- Step 1: Click the Open Connection button (at the top left of the main GUI window).

- Step 2: Fill out the Server, Username and Password fields. Do not tick the Save Password box.


Open Connection

SFTP (SSH File Transfer Protocol)

Server: hpc-data.pawsey.org.au Port: 22

URL: sftp://cou000@hpc-data.pawsey.org.au:22/

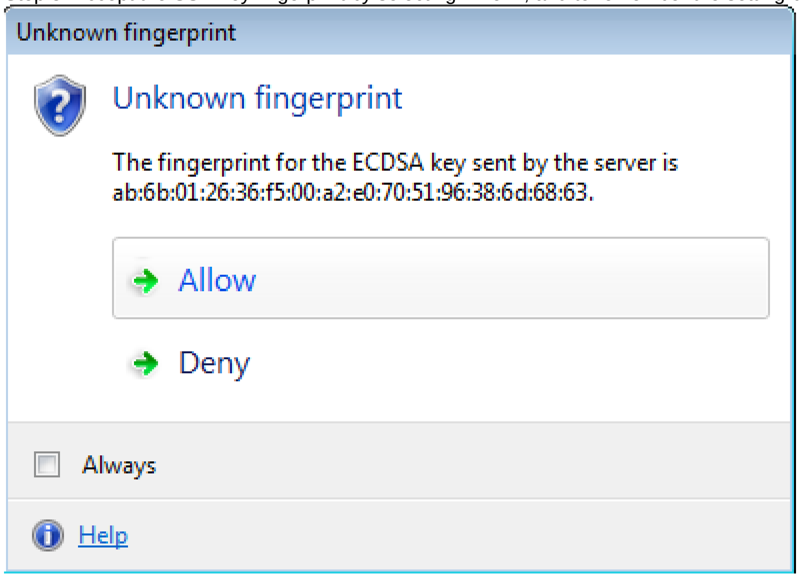
Username: cou000

Password: ●●●●●●●●●●●●

[ ] Anonymous Login

[✓] Save Password

- Step 3: Accept the SSH key fingerprint by selecting "Allow", and to remember the setting by ticking the "Always" box.


Unknown fingerprint

Unknown fingerprint

The fingerprint for the ECDSA key sent by the server is ab:6b:01:26:36:f5:00:a2:e0:70:51:96:38:6d:68:63.

[ ] Always

In Cyberduck it is useful to add a bookmark once you have navigated to the desired folder on the server. Add via the Bookmark menu.

Data Size Matters!

Use scp or GUI tools for the transfer of small amounts of data

Interactive use of **scp** may be the most appropriate and practical way of transferring of a small amount of data.

Use rsync or GUI tools for the transfer of large amounts of data

But for larger data transfers, interactive use of **scp** is discouraged. It is always preferred to perform large data transfers with tools that can resume transfers if the process/connection fails (for example **rsync** or **filezilla**).

tar your files before transfer

For a large number of files it is always better to **tar** them into a single file before transfer

Use script to automatize frequent transfers

For transfer processes that may occur routinely, the use of scripts with command-line tools for automating the processes is a great option. WinSCP also allows scripting.

As explained in the previous subsections, transfers initiated by remote sites should connect to the data-transfer nodes. When large data transfers are initiated from within the Pawsey systems, they should use the "copyq" queue.

Outgoing Connections via the copyq Partition

The **copyq** queue partition is available to provide batch access to the data-mover nodes for outgoing connections to a remote site. This partition is hosted by Zeus, and is specified using the "--partition=copyq" option to sbatch (alternatively "-p copyq"). Further, the copyq partition is available from all Pawsey Centre machines via Zeus. Submission must use the "--cluster=zeus" option (alternatively "-M zeus"). The usage of these options (together with their alternatives syntaxes) are demonstrated next, assuming that the user is logged into hpc-data, Zeus, Magnus, or Galaxy:

```
hpc-data1:~> sbatch --partition=copyq script-data-copy.sh
```

```
zeus-1:~> sbatch -p copyq script-data-copy.sh
```

```
magnus-1:~> sbatch --cluster=zeus --partition=copyq script-data-copy.sh
```

```
galaxy-1:~> sbatch -M zeus -p copyq script-data-copy.sh
```

Note that when logged into the hpc-data system, there is no need to indicate the cluster as hpc-data is indeed a subsystem part of Zeus that access directly to the data-mover nodes.

All the above options can be avoided from the command line if they are set as #SBATCH directives withing the submission script:

```
#!/bin/bash --login


# Submit a job to the copy queue ("--partition=copyq") on zeus (--cluster="zeus")
# This runs in serial, and starts with a default environment for the login shell
# on zeus ("--export=NONE")


#SBATCH --partition=copyq
#SBATCH --cluster=zeus
#SBATCH --ntasks=1
#SBATCH --account=[your-account]
#SBATCH --time=00:30:00
#SBATCH --export=NONE

# ...data transfer commands here...
```

Note that the "--export=NONE" (which basically erases all environmental variables set previous to the submission of the script) is essential for SLURM submissions between machines. It ensures a consistent login environment is used on the target machine, and not one from the submitting machine.

Passphrase-less secure transfers

Transferring data using a protocol based on SSH allows us to protect information and ensure its integrity. However, setting up a proper environment configuration can be tricky; if not done right, security risks arise. This is especially true when one wants to automate copy operations, for example through a SLURM job on data mover nodes. In such a scenario, a public key-based authentication method is recommended because the ssh client, running on a Pawsey's supercomputer node, will only need the private key to connect to a third-party system's ssh server, which in turn has the correspondent public key to be used to perform a secure handshake. The private key, however, must not be protected by a passphrase otherwise a human input is required. There are several issues to address in the described situation.

A user must generate a key-pair specifically for this purpose, i.e. data transfers from and to Pawsey systems (see Logging in with SSH keys). Let's call this key-pair: `COPYPAIR`. Do not repurpose an existing key-pair used to log in to Pawsey or other systems (which by the way, should use a passphrase). This allows isolation of unauthorised accesses due to a compromised key-pair.

The ssh server on the third party system should be configured to avoid using `COPYPAIR`'s public key to authorise connections not originating from Pawsey's data mover machines. This is a powerful capability that protects the third party server from unauthorised use of `COPYPAIR` from outside the Pawsey network. To enable the discussed feature users need to edit the `COPYPAIR.pub` public key file and prepend the following string:

```
from="hpc-data*.pawsey.org.au" no-port-forwarding no-pty
```

followed by a space and followed by the original key.

Here is an example:

```
from="hpc-data*.pawsey.org.au" no-port-forwarding no-pty ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDhGk1QdMVDVao1j9eclHPPhniU5x6rHYBhJp88DJZrEiDM3Kt70+gHvo
/fCGaHmOMWQX0hjqLs5uin42VGUW7w3y0FrIBB/hZJro+JKXJzhUJFpTE
/wR08CK8DI4c3GrxjrCqNRkd3ff4AOUIgS7VFGcmagg9aAj6iSas1ibvAMLMZuXkVyPcNcKhB+J38atc3u5
/zuRqU9QgKGQvTQgLL7lx4CrsHGKd8bPzjdEVDaCoeD1KBdRq
/S+am2wvaPwN5wqqgs6hVU83VvZggIBkGRLBbGEeMmnzu8dkG1osqE4S3RCmFVQ8MG9tiOiP0MN/jx/DpckP++NnuamJWcD/Z comment
```

Transferring data between `group` and `scratch`

dcp or distributed copy is a tool which uses `MPI` to directories and large files in an efficient manner. This tool can be used only on data mover nodes (**hpc-data.pawsey.org.au**).

It is available as a `module` on data mover nodes. Users may use it directly from command line when on a data mover node or submit a job on `copyq`.

```
module load mpifileutils
mpirun -np 4 dcp -p SOURCE DESTINATION
```

`SOURCE` can either be a file or directory. The option `-p` preserves the file attributes e.g permissions, group association and ownership of the files.

```
mpirun -np 4 dcp -f SOURCE DESTINATION
```

Option `-f` deletes the any files on in `DESTINATION` directory if an error occurs during the operation.

Here is a sample output of a 100GB file copied from `group` space to `scratch`. (The file was striped over 4 OSTs)

```
 mshaikh@hpc-data1:/scratch/pawsey0001/mshaikh/dcp2-test> mpirun -np 4 dcp $MYGROUP/dcp2-test/100gb.bin
$MYSCRATCH/dcp-test/
[2017-01-27T12:34:41] [0] [handle_args.c:315] Walking /group/pawsey0001/mshaikh/dcp2-test/str-c4/100gb.bin
[2017-01-27T12:34:42] [0] [dcp.c:222] Creating directories.
  level=6 min=0 max=0 sum=0 rate=0.000000/sec secs=0.000006
[2017-01-27T12:34:42] [0] [dcp.c:430] Creating files.
  level=6 min=0 max=1 sum=1 rate=1.587000 secs=0.630120
[2017-01-27T12:34:43] [0] [dcp.c:922] Copying data.
[2017-01-27T12:48:00] [0] [dcp.c:967] Fixing permissions.
[2017-01-27T12:48:00] [0] [dcp.c:1362] Syncing updates to disk.
[2017-01-27T12:48:00] [0] [dcp.c:146] Started: Jan-27-2017,12:34:41
[2017-01-27T12:48:00] [0] [dcp.c:147] Completed: Jan-27-2017,12:48:00
[2017-01-27T12:48:00] [0] [dcp.c:148] Seconds: 798.996
[2017-01-27T12:48:00] [0] [dcp.c:149] Items: 1
[2017-01-27T12:48:00] [0] [dcp.c:150]   Directories: 0
[2017-01-27T12:48:00] [0] [dcp.c:151]   Files: 1
[2017-01-27T12:48:00] [0] [dcp.c:152]   Links: 0
[2017-01-27T12:48:00] [0] [dcp.c:154] Data: 100.000 GB (107374182400 bytes)
[2017-01-27T12:48:00] [0] [dcp.c:158] Rate: 128.161 MB/s (107374182400 bytes in 798.996 seconds)
```