Data Transfer

There are different ways to move data back and forth between different areas.

Using **ftp** (file transfer protocol) is not permitted on CINECA systems for security reasons, but other tools are available and the following presented.

SCP secure copy protocol

This program allows moving data, using ssh encryption, between two servers. syntax:

```
scp -r *.bin myusername@remote.host:/my_directory/.
```

In this way I'm coping all *.bin files in the local system current directory to "/my_directory/" of the user "myusername" in the remote machine "remote.host". Unless you are not using the public ssh-key, the "myusername" password will be requested before proceeding. The "-r" flag allows ou to recursively copy entire directories.

scp is useful to move a small amount of data since it is not optimized. For example, if a file is present in the remote directory it will be overwritten (even if it is exactly the same). For further info type

`man scp`

RSYNC remote sync

With this program is possible to synchronize your local directory with a remote one.

```
rsync --r vzu -e ssh *.bin myusername@remote.host:/my_directory/.
```

In this way I'm coping all *.bin file in the current directory to "/my_directory/" of the user "myusername" in the remote machine "remote.host" using the ssh protocol. Using the public ssh-key is no more necessary to enter the password. The "-z" flag compress/decompress file to save bandwidth.

Respect to **scp**, before copying there is a check of the file in the remote directory. If the file to be copied already exists with the same size, the copy is not performed. For further info type

`man rsync`

or consult this documentation

SFTP secure ftp (ssh encryption)

With this program is possible to get/put files from a remote directory.

```
sftp myusername@remote.host
>...
>mget *.bin
```

In this way I'm entering in the home directory of the user "myusername" in the remote machine "remote.host" using the ssh protocol. Now I can get/put files using command mget (for getting from remote directory) or mput (to put into the remote directory). For further info type

`man sftp`

gridFTP clients

Such as globus-url-copy via command line, or GlobusOnline web browser GUI.

These programs allow to move huge quantities of data between two grid-enable endpoint. Refer to GridFTP client link for finding information about globus-url-copy and GlobusOnline usage in CINECA.

UNICORE UFTP

The UNICORE UFTP daemon has been installed from the corresponding rpm package on the CINECA HPC facility:

https://grid.hpc.cineca.it:9111/CINECA-MARCONI

previously also on PICO and GALILEO in the same way. The UFTP daemon receives requests from the UNICORE client through the UNICORE gateway (grid.hpc.cineca.it).

However in order to use the UFTP file transfer mechanism you need to be registered on our UserDB (userdb.hpc.cineca.it) and get an HPC account.

In order to move data using the UNICORE UFTP daemon, in general, the user can exploit both the UNICORE ucc (command line) client or the UFTP Standalone client, see UNICORE documentation (https://www.unicore.eu/documentation).

1) In order to use the ucc command line client (https://sourceforge.net/projects/unicore/files/Clients/Commandline%20Client/) you can download it directly from the UNICORE portal, then:

cd your-unicoreclient-location/bin

copy one file from your local home directory to your remote home directory on the hpc cluster (i.e. on MARCONI), see example below

```
./ucc put-file -s ./test.txt -t https://grid.hpc.cineca.it:9111/CINECA-MARCONI/services/StorageManagement?res=username-Home#/username/test.txt --protocols UFTP -v
```

but you can also move files vs any location on the shared file system.

Before doing this you have to configure the ucc configuration file adding you personal .p12 certificate (as described in the documentation) and also seting the CINECA gateway endpoint (grid.hpc.cineca.it)

2) The UFTP client (https://sourceforge.net/projects/unicore/files/Clients/UFTP-Client/) is a Java-based client for UFTP. It allows to

- list remote directories
- upload/download files
- sync files
- make remote directories
- delete remote files or directories

The *uftp* client will connect to a UFTP Authentication Server to authenticate and then to the *uftpd* server for transferring data or making a file operation. The *uftp* client supports username/password authentication, OIDC token authentication and (on UNIX) ssh-key authentication.

Actually the UFTP client needs the UNICORE authentication service, this is currently not supported anymore in CINECA

3) in order to list storages you have to execute the ucc connect -v command, example:

```
/UNICORE/ucc/unicore-ucc-7.5.0/bin$ ./ucc connect -v
[ucc connect] UCC 1.7.8, http://www.unicore.eu
[ucc connect] Reading properties file </home/username/.ucc/preferences>
[ucc connect] Current directory is </home/username/Desktop/UNICORE/ucc/unicore-ucc-7.5.0/bin>
[ucc connect] Output goes to <.>
[ucc connect] Registry = https://grid.hpc.cineca.it:9111/PRACE/services/Registry?res=default_registry
[ucc connect] Checking registry connection.
[ucc connect] Registry connection status: OK
[ucc connect] Current directory is </home/username/Desktop/UNICORE/ucc/unicore-ucc-7.5.0/bin>
[ucc connect] Output goes to <.>
[ucc connect] Using site default for TSS lifetime.
[ucc connect] Connecting to https://grid.hpc.cineca.it:9111/CINECA-GALILEO/services/TargetSystemFactoryService?res=default_target_system_factory
[ucc connect] Connecting to https://grid.hpc.cineca.it:9111/CINECA-PICO/services/TargetSystemFactoryService?res=default_target_system_factory
[ucc connect] Connecting to https://grid.hpc.cineca.it:9111/CINECA-MARCONI/services/TargetSystemFactoryService?res=default_target_system_factory
[ucc connect] Connecting to https://unicore.surfsara.nl:4014/SURFsara-Cartesius/services/TargetSystemFactoryService?res=default_target_system_factory
You can access 4 target system(s).
```

From the UFTP documentation (https://www.unicore.eu/docstore/uftp-2.4.0/uftp-manual.html) you can also perform third party transfers across HPC sites, example:

```
ucc copy-file -s https://hbp-unic.fz-juelich.de:7112/HBP_JUQUEEN/services/StorageManagement?res=BP0#/bp000002/output_data.txt -t https://grid.hpc.cineca.it:9111/CINECA-PICO/services/StorageManagement?res=bp000002-Home#/data_final.txt -P UFTP  -v
```

External devices (USB disks)

It is possible, but only in a particular situation, to copy all data to an external device like USB disks. Keep in mind that, unless you have an USB 3.0, the bandwidth is about 10 MB/s. Contact User Support staff (mailto: superc@cineca.it) to check this option.

Some Caveat/Hints about data transfer

- On our HPC platforms, all interactive programs are killed after 10 minutes of cpu time, so big data movements have to be done using a batch script
- To reduce the size of the file to transfer use compressing program (e.g. gzip)
- The latency of transfering a file is quite high, so to exploit bandwith it is better to copy 1 big file (using unix tar command) instead of many little ones
- Moving data outside CINECA NAS (Network Area Storage) heavily depends on the bandwidth of the network you are using

About data transfer efficiency

The technology of choice for moving your data depends mainly on the actual size.

Here we present some examples about data transfer efficiency, in order to let you estimate the time required for your own data transfer.

data structure	source --> target	tool	speed (Gb/s)
Single file, **size 150 GB**	**GALILEO:** CINECA_SCRATCH <--> DRES-FileSystem	rsync (quick check enabled)	0.9 +- 0.2
		cp	2.4 +- 1.2
		gridftp (Globus OnLine - checksum enabled)	1.7 +-0.5
	PICO: CINECA_SCRATCH <--> DRES-FileSystem	rsync (quick check enabled)	1.1 +- 0.3
		cp	2.1 +- 1.0
		gridftp (Globus OnLine - checksum enabled)	0.7 +- 0.2
1500 files, **size per file: 100MB** **total size: 150GB**	**GALILEO:** CINECA_SCRATCH <--> DRES-FileSystem	rsync with quick check	0.9+-0.2
		cp	1.8 +-0.6
		gridftp (Globus OnLine - checksum enabled)	1.8 +- 0.8
	PICO: CINECA_SCRATCH <--> DRES-FileSystem	rsync (quick check enabled)	1.0 +- 0.2
		cp	1.4 +- 0.4

		gridftp (Globus OnLine - checksum enabled)	1.3 +- 0.5

data structure	source --> target	tool	speed (Gb/s)
Single file, **size 150 GB**	**DRES-REPO** --> GALILEO (CINECA_SCRATCH)	icommand (iget - checksum enabled)	1.2 +- 0.3
		gridftp (Globus OnLine - checksum enabled)	0.50 +- 0.01
	DRES-REPO --> PICO (CINECA_SCRATCH)	icommand (iget - checksum enabled)	1.3 +- 0.5
		gridftp (Globus OnLine - checksum enabled)	0.4 +- 0.1
	GALILEO (CINECA_SCRATCH) --> **DRES-REPO**	icommand (iput - checksum enabled)	0.6 +- 0.3
		gridftp (Globus OnLine - checksum enabled)	0.4 +- 0.2
	PICO (CINECA_SCRATCH) --> **DRES-REPO**	icommand (iput - checksum enabled)	0.8 +- 0.2
		gridftp (Globus OnLine - checksum enabled)	0.46 +- 0.08
1500 files, **size per file: 100MB** **total size: 150GB**	**DRES-REPO** --> GALILEO (CINECA_SCRATCH)	icommand (iget - checksum enabled)	0.63 +- 0.03
		gridftp (Globus OnLine - checksum enabled)	0.68 +- 0.01
	DRES-REPO --> PICO (CINECA_SCRATCH)	icommand (iget - checksum enabled)	0.8 +- 0.2
		gridftp (Globus OnLine - checksum enabled)	0.6 +- 0.1
	GALILEO (CINECA_SCRATCH) --> **DRES-REPO**	icommand (iput - checksum enabled)	0.4 +- 0.2
		gridftp (Globus OnLine - checksum enabled)	0.4 +- 0.2
	PICO (CINECA_SCRATCH) --> **DRES-REPO**	icommand (iput - checksum enabled)	0.5 +- 0.1
		gridftp (Globus OnLine - checksum enabled)	0.9 +- 0.3

Some Hints about I/O

According to the figures presented here, I/O is crucial to achieve high performance on HPC systems.

According to the technological trends about HPC system, now I/O (and so storage issues) are a serious bottleneck for achieving high performances. So, please, follow this "**golden rules**" about I/O, keeping in mind that I/O issues will be even much more relevant for future HPC systems.

- Reduce I/O as much as possible: only relevant data must be stored on disks
- Save data in binary/unformatted form:
  - asks for less space comparing with ASCI/formatted ones, more than a factor 3!!!
  - It is faster: using ASCI/formatted format the data must be translated in a human-readable format. It is a time-consuming operation.
- Before starting to produce simulations data, plan a clean directory structure. It helps to find where the data are and which data are important and which are not.
- Save only what is necessary to save for restart/checkpointing, everything else, unless for debugging reason or quality check should be computed ***on the fly***.
- After each simulation clean all unnecessary files. It is better to remove them step by step...
- Remove duplicated restart files, use if possible symbolic links;
- Remove all test directories before starting data production simulations;
- Take a log of each simulation, it is very useful to keep track of all your simulation
- Use, if possible MPI/IO instead of standard I/O. Or use I/O libraries like HDF5 or netCDF, the parallel ones.
- Dump all the quantities you need once, instead of using multiple I/O calls: if necessary use a buffer array to store all the quantities and save the buffer using only a few I/O calls.
- Synchronize as much as possible the important data with your remote directory instead of doing that only at the end of your grant. **Transfering TB can ask for days or weeks**, depending on the interconnection and protocol used. It must be planned very carefully.
- Check periodically your storage usage using commands like "cindata" (for global occupation) or "du -sh" (for single directory occupation).
- Look for hidden files (the one starting with "**.**", i.e.".name_of_the_file"). They usually are temporary files that in general could be removed and are not visible using the "ls" command.
- Do not ask for more storage before removing all unnecessary data.

Use Cases

Data to be transferred back at home. I'll need them for further processing on my local cluster	use rsync o gridFTP, or even GridFTP if your local host supports it. Depending on the throughput of your local network, consider several days to move 1TB of data.
Data to be transferred to another PRACE site I'll need them for further processing on another Tier-0 system	gridFTP is strongly recommended. Since a very efficient network is in place among PRACE sites, you can expect to spend a few hours to move 1TB of data.