JSC HPC SUPPORT CORNER

HOW TO TRANSFER DATA TO/FROM HPC

23.04.2025 I ILYA ZHUKOV

ARCHIVING FILES

tar and zip

- **Goal: Combine multiple files/directories into a single archive, optionally compressing them.**
- Create an archive with **tar**:

  tar -cvf archive.tar file1 file2 dir1

  Creates "archive.tar" containing the listed files/directories
- Create a compressed archive (gzip): **tar -czvf archive.tar.gz file1 dir1**
- Extract an archive:

  tar -xvf archive.tar

  tar -xzvf archive.tar.gz (for gzip-compressed)

- Create a zip archive: **zip archive.zip file1 file2**
- Recursively zip a directory: **zip -r archive.zip dir1**
- Extract a zip archive: **unzip archive.zip**

DOWNLOADING FILES FROM THE INTERNET

wget and curl

- **Goal: Download files from HTTP, HTTPS, or FTP sources.**
- Works only on login nodes!
- Download a file:

  wget <link>

  curl -O <link> (download and save as original filename)

  wget --user=<username>@fz-juelich.de --ask-password <link address> (download from sciebo)

  - Upload file to sciebo with **curl**:
    - Recipe available at **/p/<filesystem_where_files_deleted_after_90_days>/share/ScieboEgg/sciebo_curl.txt**

JÜLICH Forschungszentrum

TRANSFERRING FILES I

scp and rsync

- **Goal: Copy files between local and remote systems.**
- Copy local file to remote (execute on local machine):

```
scp [options] /path/to/file/filename <username>@<remote_system>:/path/to/dest/
rsync -avzP /path/to/file/filename <username>@<remote_system>:/path/to/dest/
```

-a (archive) preserves the date and times, and permissions of the files;

-v (verbose) option gives verbose output to help monitor the transfer;

-z (compression) option compresses the file during transit to reduce size and transfer time;

-P (partial/progress) option preserves partially transferred files in case of an interruption and also displays the progress of the transfer.

TRANSFERRING FILES II

scp and rsync

- **Goal: Copy files between local and remote systems.**
- Copy from remote to local (execute on local machine):

```
scp [options] <username>@<remote_system>:/path/to/file/filename /path/to/dest
rsync -avzP <username>@<remote_system>:/path/to/file/filename /path/to/dest
```

Examples

- `rsync -avzP ./my_app.zip musterman12@jureca.fz-juelich.de:/p/project/project42/`
- `scp musterman12@judac.fz-juelich.de:/p/project/project42/test.tar .`

OTHER MEANS TO TRANSFER FILES I

- **SSHFS** allows you to mount a remote filesystem using SFTP.
  https://github.com/libfuse/sshfs

- **UFTP (UNICORE FTP)** is a file transfer tool similar to Unix' FTP. Its main features include high-performance file transfers from client to server (and vice versa), list directories, make/remove files or directories, sync files and data sharing. In addition, users can easily share their data even with users who do not have Unix-level access to the data.
  https://apps.fz-juelich.de/jsc/hps/judac/uftp.html

OTHER MEANS TO TRANSFER FILES II

- **GridFTP** is an extension of FTP used within large science projects. It includes features like parallelized FTP streams, fault tolerancy, download of portions of data and authentication and encryption for file transfers.

  https://apps.fz-juelich.de/jsc/hps/judac/gridftp.html

- On Windows you can use various clients, e.g. **WinSCP**, **FileZilla**, **PuTTY**, etc.

- **Jupyter-JSC**

  https://jupyter.jsc.fz-juelich.de/

- **VSCode**, etc.

MANAGING LONG-RUNNING TRANSFERS

nohub, tmux, screen

- **Goal: run commands that won’t get interrupted (like the terminal closing).**

- **nohup** – run in background, e.g.

  nohup rsync -avz test.tar <username>@<remote_system>:/path/to/dest/ &

  (Output is redirected to “nohup.out” by default)
- Use interactive terminal multiplexer
  - **tmux**: https://github.com/tmux/tmux/wiki
  - **screen**: https://www.gnu.org/software/screen/

BEST PRACTICE

- Always archive/compress many small files before transfer
- Use JUDAC for data transfers
  - https://www.fz-juelich.de/en/ias/jsc/systems/storage-systems/judac
- Use rsync/scp to transfer data.
- For long transfers, use nohup/tmux/screen to avoid interruption.
- For large data transfer, consider using UFTP/GridFTP