File transfer to/from the cluster

There is a special node dedicated to data transfer with the cluster system. Whenever you transfer data with the cluster system, please use this node:

```
transfer.cluster.uni-hannover.de
```

Please note: Use the dedicated transfer node for file transfers, because processes that use more than 30 minutes of cpu time on the login machines will be aborted by the system. Since ssh/scp use encryption, you may eventually use enough cpu time to get the transfer killed on the login nodes.

Files can be transferred to and from the cluster system in a number of ways. What has been said in the section about the availability of the tools on different operating systems also holds true here: the command line tools like `scp`, `sftp` or `rsync` are usually already available on Linux and Mac OS computers. For Windows, you may need to install additional software such as MobaXterm which offers a command line in addition to graphical interface for file transfer. You'll usually want to initiate the connection from your own workstation, since you need a running ssh server for most of the commands, and a Windows workstation usually does not have (should not have without good reason) that.

The following command, e.g., copies a whole directory `mydata` from the cluster to a local directory on your workstation:

```
[myworkstation]$ scp -r <username>@transfer.cluster.uni-
hannover.de:/path/to/mydata .
```

Replace `<username>` above with your cluster user name.

An very efficient, but also very powerful and more demanding tool is `rsync` which can synchronize directories across sites and also complete interrupted file transfers. If you want to sync the contents of directory `mydir` (source) on your workstation with directory `mydir` (target) on your BIGWORK, run:

```
[myworkstation]$ rsync -av --delete /path/to/mydir/
<username>@transfer.cluster.uni-hannover.de:/bigwork/<username>/mydir/
```

Watch out for the slashes '/' at the end of both source and target path, they are important. If in doubt, use `--dry-run` to check what you are about to do, and possibly create an extra subdirectory for sync'ing until you get familiar with the particular syntax. Whether you put a slash ('/') at the end of source or target or not is interpreted as either “sync the *contents* of this directory” vs. “sync the directory itself (including everything in it)”. So `/path/to/mydir/` and `/path/to/mydir` have completely different meanings. Ask `man rsync` to get the complete picture.

The `--delete` option removes files in the target that are missing in the source directory, so the target is in sync with the source. Take care that you are in the correct subdirectory, or you may delete

local files that you still wanted. If in doubt, test without this option first.

Alternatively, you can use FileZilla if you want to use a graphical tool. FileZila supports Linux, Mac OS, as well as Windows platforms. Information on how to configure FileZilla for use with the cluster system can be found in the next section.

File transfer using FileZilla

The FileZilla client may be used to exchange files between your workstation and the cluster system in sftp mode. In the following section we provide instructions on how to install and configure FileZilla client version `3.14.1_win64` on Windows, but the software is available for other platforms as well. FileZilla can be obtained from the following URL. After downloading, you can install the FileZilla client to a directory of your choice.

After installing, open the *Site Manager* and create a new server which you can then connect to. The following options have to be set (cf. the red boxes in figure figure 1).

- **Host**: transfer.cluster.uni-hannover.de
- **Protocol**: SFTP - SSH File Transfer Protocol
- **Logon** Type: Ask for password
- **User**: Your user name

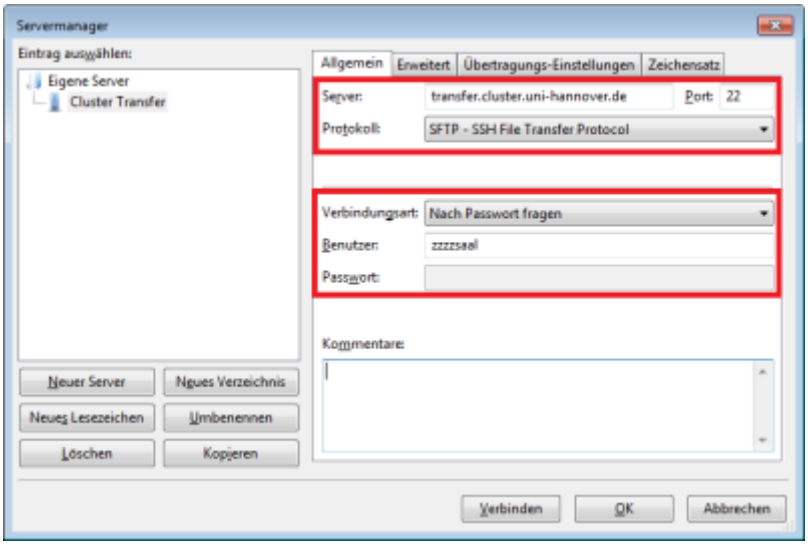

Fig. 1: Site Manager: General

Furthermore, it is possible to open the remote connection directly to $BIGWORK. Without further configuration, the remote directory will be set to $HOME. In order to configure this option, go to the *Advanced* tab and set *Default remote directory* accordingly, see figure figure 2.

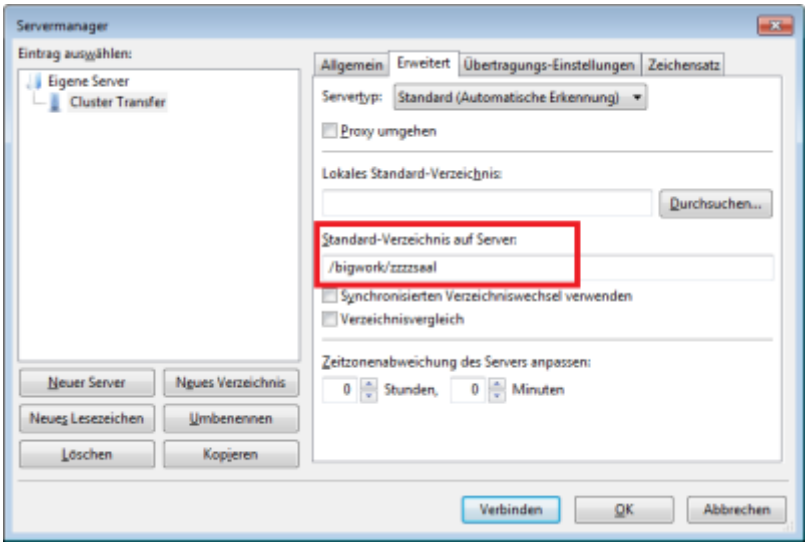

Fig. 2: Site Manager: Advanced

The first time a connection to the transfer node is made, you will need to confirm the authenticity of the node's host-key - cf. figure figure 3.

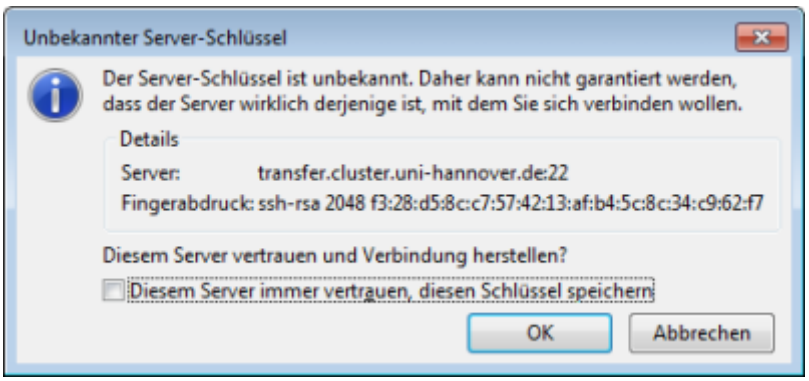

Fig. 3: Host-key verification on first connection attempt

After a connection is successfully established – cf. figure figure 4 – you can exchange data with the cluster system.

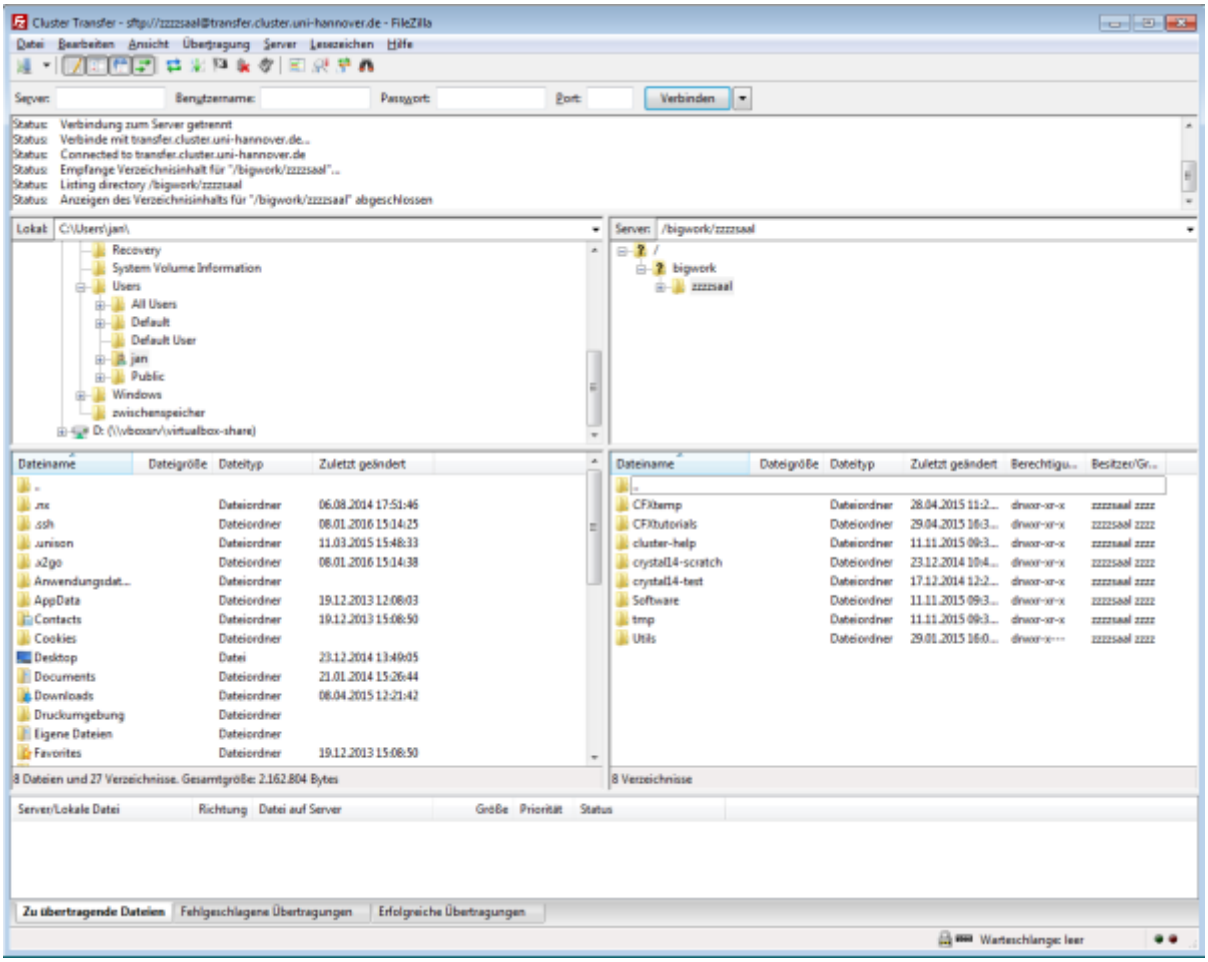

Fig. 4: Connection to transfer-node, established with FileZilla

If your own workstation runs Windows, you should now have a look into `Settings::Transfers::FTP` File Types and check that the transfer type selected will be correct for the files you will want to transfer. See the section "Converting jobscripts written under Windows"

In case you see different file sizes or if you experience corrupt files after the transfer, you may/will probably need to switch to “binary transfer mode”. FTP is an old protocol that once took care that different encodings of ASCII text files were translated properly, and you'll quite probably don't want that for binary data files.

File transfer using web browser

To transfer a limited number of relatively small files, you can use the cluster's OOD web portal - after logging into the portal, go to the `File` menu. Currently, the size of each uploaded file must not exceed 1 GB. You can also directly edit your files through this interface, eliminating the need to transfer files to your computer for editing, see figure figure 5.

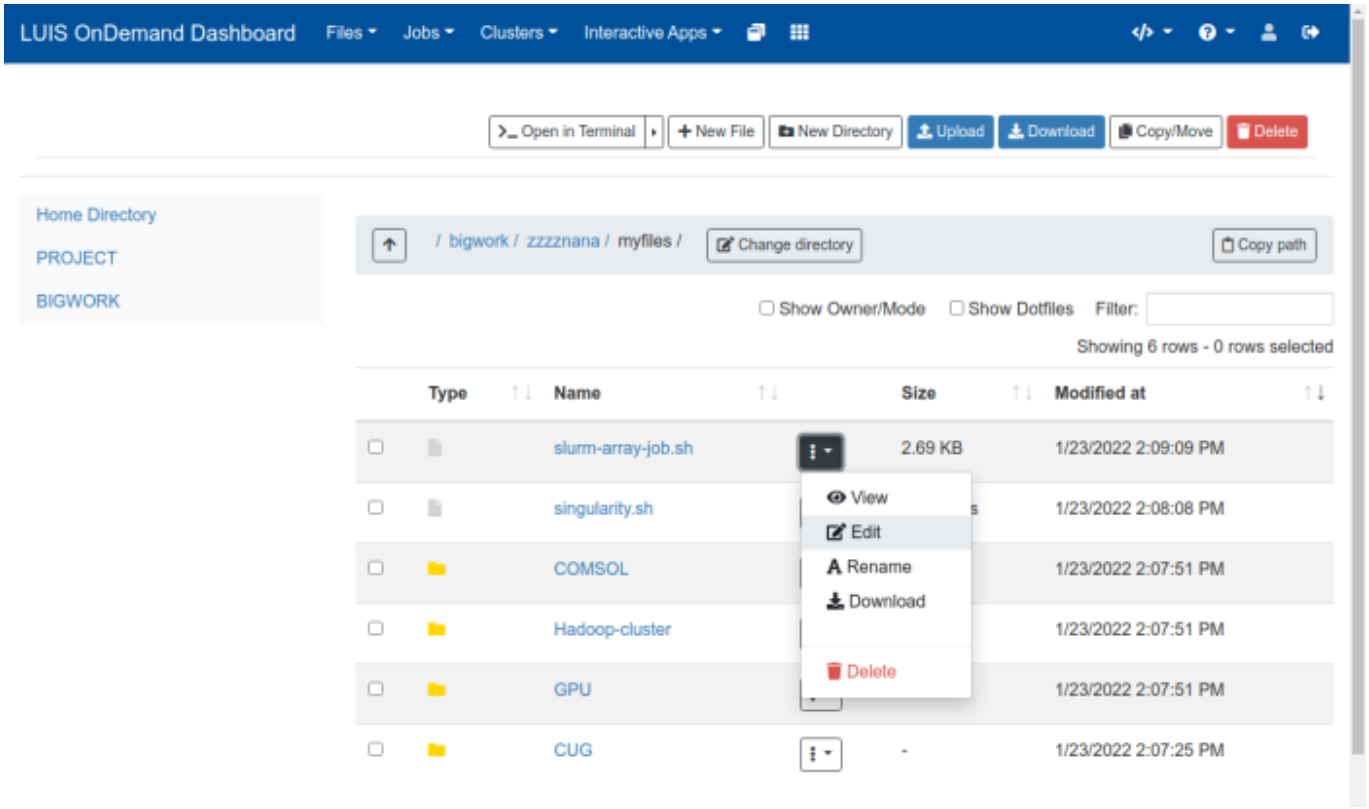

Fig. 5: Cluster web portal, file manager

To ensure your files are really identical

There's a variety of tools with which you can create a checksum for your files to check that not only the size of the file is correct, but to also make it highly unlikely that it has been somehow corrupted during transfer. A good tool could e.g. be sha256sum, which you could use both on the source and the target end to check whether you get the same checksum for

```
sha256sum <file-to-check>
```

If the results are identical, your files probably are too.

In case you experience strange aborts transferring data to Windows machines

… check whether your target storage is formatted with a file system like FAT32 that does not support files larger than 4 Gigabytes. Open a file explorer (LeftWindows+E), select the disk, right click, properties → common. In case you are using an unsuitable file system, you'll need to reformat to a file system that supports large files (e.g. exFAT or NTFS).

Converting jobscripts written under Windows

Windows and Unix/Linux use a different byte/character to mark the end of a line in a text file. You usually won't see these characters, of course, since they are converted into a line change, but when

you transfer a file to a different operating system without taking care about this, you may get into trouble.

In Windows, the end of a line is marked by a combination of CR/LF (“carriage return” and “line feed”). In Linux, it's just LF.

So when you transfer text files - and scripts usually are text files - without taking care of that difference, you'll probably see “^M” characters at the end of the line (when you created a file on Windows, transferred it to the cluster and edit it there) or (when the file was created on the cluster and you transferred it back to a Windows machine) just one very long line.

To automatically take care about this, tools like e.g. FileZilla usually provide two transfer modes, ASCII and binary, and for text files (like job scripts), ASCII should be used. There's also frequently an “Auto” mode that selects the (hopefully) correct mode by looking at the file extension, like .bat, .job, .py, .txt and so on.

The problem is sometimes hidden, since Editors like WordPad may recognize the “wrong” format and display the file correctly, and not all editors show “^M”, since this is a so-called control character. But you should always be able to check the type of your files with the “file” command (on Linux).

Example:

Creating a jobscript under windows and copying it onto the cluster system may create the following error message when submitting that jobscript with `sbatch`:

```
zzzz0001@login02:~$ sbatch file-from-windows.txt
sbatch:  script is written in DOS/Windows text format
```

Check this file with the `file` command:

```
zzzz0001@login02:~$ file file-from-windows.txt
file-from-windows.txt: ASCII text, with CRLF line terminators
```

Convert the file to Unix format:

```
zzzz0001@login02:~$ dos2unix file-from-windows.txt
dos2unix: converting file file-from-windows.txt to UNIX format ...
```

Check the file again with the `file` command to see if conversion was successful:

```
zzzz0001@login02:~$ file file-from-windows.txt
file-from-windows.txt: ASCII text
```

From:
https://docs.cluster.uni-hannover.de/ - **Cluster Docs**

Permanent link:
https://docs.cluster.uni-hannover.de/doku.php/guide/to_pdf/250_ransferring_data

Last update: **2024/04/26 17:22**