



Toolbox: Using the CIP

Sometimes the personal computer is not powerful enough to run code and it doesn't have to be. Most of the time you will be able to outsource the computation to a cluster. For the practical course, you have this possibility as well. Using the computers in the CIP pool, you can run most common programming languages.

The computers at the CIP pool got a scratch partition, where your data can be saved. This folder is not reset upon reboot. However, for maintenance reasons, there are reinstallations of the computers. In case of such a reinstallation, the scratch partition is deleted as well. Therefore it is important, that you record your activities in [this shared spreadsheet](#). If you do not do so, your code or results might be deleted without warning.

1 Start computations locally

You can get an account to use the computers at the CIP pool, by talking to the supervising person there. Using this account, you can log in to any of the computers there. They already support most of the common programming languages. When running a computation, it is important that you save the results in the `/scratch/compphys` folder. Otherwise, your data will be lost upon reboot. Create a myname folder in this directory, so you will be able to find your data later on.

2 Start computations remotely

2.1 VPN and SSH

To be able to start a computation, you do not even have to be on site. When connected to the university network, code can be executed via the CIP server. If you are not at the university, you can still access its network via VPN. The VPN setup is explained in detail on the [RRZK website](#).

Once you are connected to the university network, you can access the CIP server via SSH. Their server is available under the address

```
neptun.ph-cip.uni-koeln.de.
```

The username and password read: `compphys`. Across all common operating systems, SSH access works in the terminal via:

```
ssh compphys@neptun.ph-cip.uni-koeln.de
```

2.2 How to start the computation

Once you successfully logged into the server, there are two important rules you have to keep in mind:

1. Use the scratch partition to save your data! Everything else will be lost in the case of a reboot.
2. NEVER start a computation on the server itself! The server will crash.

To follow the rules, use `ssh` again, to log into a specific computer. Their name is a concatenation of `cip` and a number (3-25). To log in to e.g. `cip14` type:

```
ssh cip14
```

Before continuing, you should use the `htop` command to check the computer's load. If the computational resources are already used up by another process, go back to the server by typing `exit` and pick a different computer.

Once you have found a computer, change to the `/scratch/compphys` folder, by typing

```
cd /scratch/compphys
```

and create a folder with your name.

```
mkdir myname
```

In this folder you can start any computation you like (change there, using `cd myname`). However, your code might use up all the available computing resources, which would make the computer unusable for someone sitting in front of it. To prevent this, the priority of your computation has to be lowered, using the `nice` command. Use the priority `-19`.

There is no need to stay connected all the time. Decoupling the computation from your connection to the server comes with the additional benefit that losing your connection does not abort the computation. Use the `screen` command for this purpose. This command opens a new terminal session, comparable to a window in your browser. Create a session `mysession`, start the computation there and detach from the session. You can reattach at any time. Close the session when you are done. To sum all of this up, the commands you want to use read:

- **start session**
`screen -S mysession`
- **start computation**
`nice -19 <start my calculation>`
- **detach via shortcut: ctrl+a d**
- **reattach**
`screen -r mysession`
- **To end a session, detach and type**
`screen -XS mysession`

If for example, you want to execute the python code *pycode.py*, `<start my computation>` would look like this:

```
python pycode.py
```

If you want to executed the julia code *jlcode.jl* `<start my computation>` would look like this:

```
julia jlcode.jl
```

Log out by typing `exit`.

2.3 Julia in the REPL

If you want to use Julia in the REPL, you can start it by typing `julia`. To exit the REPL, type `exit()`.

2.4 Copy data remotely

Crucial for working with a computer remotely, is the ability to copy files and documents to and from them. Before copying it is recommended to compress the files or folders in question to reduce the file size that is been copied. A terminal command that works across all platforms (Windows, macOS, Linux) is `tar`. It is executed from the terminal. Type:

```
tar -czf archive.tar.gz examplefile examplefolder
```

This command compresses the file `examplefile` and the folder `examplefolder` into `archive.tar.gz`. To extract the archive again, use:

```
tar -xzf archiv.tar.gz
```

The command used to copy data is the `scp` command. The syntax is analogous to a simple `cp` command. The command takes two paths as input and copies

data from the first to the second path.

```
scp -r <source> <destination>
```

The `-r` is only necessary when copying folders. Continuing the example started above, to copy the `pycode.py` from your local Documents folder to `cip14`, the source would read:

```
~/Documents/pycode.py
```

and the destination:

```
compphys@cip14.ph-cip.uni-koeln.de:/scratch/compphys/myname
```

Accordingly, getting an eventual output file from `cip14` to your personal computer, the source and destination are interchanged. Note that under Windows, the path to your local documents folder would read: `C:\Users\uname\Documents`. Replace `uname` by your username.

2.5 Summary

- connect to the university network via VPN
- use SSH to log into the CIP server `neptun` (`ssh`)
- pick a computer for your calculation (`ssh`) and check, whether there are already heavy calculations running with (`htop`)
- record your computation in the shared document
- create a folder in the scratch partition with your name (`mkdir`)
- copy your code to this folder (`scp`)
- start the computation (`nice`, `nohup`)
- copy the results back to your machine (`scp`)