Working remotely in the CIP

As we have to work remotely this semester, we hereby provide you a short walk through on how to setup your remote connection and get started working in the CIP pool.

- If you do not have an account for the CIP pools yet, you can create an account using your IdM login.

This tutorial is specifically important for exercise 4, where we assemble a deep learning model to train and test using PyTorch. To efficiently use PyTorch for training of deep neural networks with very large data sets, GPUs are utilized as a common-practice. Training your model on your local machine may take ages, if you do not have a NVIDIA GPU compatible with CUDA. Nonetheless, all CIP machines support SSH for remote access, only the machines in the Huber-CIP, CIP-3 and CIP-4 are equipped with suitable GPUs.

To make sure you have a remote machine assigned in your respective excercise we want to establish a schedule for the exercise hours. During previous semesters we held our exercises in the Huber-CIP and could assure that all Deep Learning students can access a machine in their respective exercise time. We cannot guarantee access to a specific computer at any time right now. So we ask you to use the computers moderately and considerately. To establish a remote computer schedule during our exercise hours, we have created Excel Sheets in your respective exercise in MS Teams. Please leave your TeamID, in a computer slot that you are working on in. This should at least resolve some conflicts for you.

For your SSH session you will need your loginname (username on CIP account) and your hostname, which consists of your machine name and '.cip.cs.fau.de'. There are several ways to use a remote machine and further information can also be found in the CIP-tutorial:

- Using Linux or MacOS you can log onto a virtual machine with the command

      ssh loginname@hostname

  in your terminal. You will be asked for your CIP password and then have access to your remote terminal.

- To login from a Windows system you'll need to install a SSH-Client, e.g. Putty.

- PyCharm Professional (which is free of cost for students) provides a remote SSH-Interpreter option to directly run your program from your PyCharm environment. The logon process is almost alike. **However the default interpreter 'usr/bin/python' points to a Python 2 version. To set your remote interpreter to a Python 3 version use: '/local/python3.7-Anaconda3-2019.07/installation/bin/python3'** Note: This path is hard-coded and may change. In case the above mentioned link is outdated do not hesitate to contact us in the forum or check your Python 3 location on a CIP machine. **Moreover your default 'Sync folder' (which will be used to store your files on your remote machine) (as seen in step number 6 in the linked description) will be set in the folder '/tmp/' on the CIP pool machine.** This is a local folder on the remote machine and read and write flags are set such that everyone can copy or edit your files. **To avoid confusion and plagiarism, we highly recommend to set this folder to a private folder.** Detailed information on how you set up remote connection in PyCharm is found here.

- Visual Studio Code offers a similar option.

- Last but not least, the CIP machines also provide a graphical interface that can be used with the Xpra HTML5 client. This comes in specifically handy for quick checks and all machines guarantee access to a GPU and PyTorch.

Common problems

During previous iterations of the Deep Learning Exercises, we have encountered certain issues repeatedly when working with the machines in the Huber-CIP. If you have an issue with your machine and something is not working as it should be, this sheet may help you to figure out what went wrong and how to solve it. **Important: Before deleting any files, make sure you do not need them anymore or save them somewhere else!**

How can I check if my graphics card is beeing used?

Just because you are logged onto a CIP machine, does not necessarily guarantee that your GPU is not beeing used by someone else at the same time. During the exercise hours we try to guarantee you a spot on a machine, however here are useful tips to monitor GPU usage outside of your exercise times.

- To monitor the progress on a NVIDIA GPU:

```
nvidia-smi
```

- Process viewer and manager:

```
htop
```

- While monitoring processes it can be useful to use terminal multiplexing.

What is quota and what can I do once I have exceeded it?

- The quota is the limit set by the admins. It's to prevent cip-users from allocating too much disk space.
- You can check your quota by execution of the following command:

```
cip-quota
```

- Once you have exceeded your limit you have two opportunities:
  1. Execute the

     ```
     ncdu
     ```

     command and browse through your personal files, identify the files that need much space, and delete those with the key 'd' or save them somewhere else outside of cip.
  2. Every student has the option to extend the quota by adding a user-specific folder in the directory '/proj/ciptmp'. For more informations go to /proj/ciptmp and carefully read the README there. When creating a folder, please set and check your directory rights accordingly. To set read, write and execute settings to owner only use:

     ```
     chmod 700 /path/to/folder
     ```

Problems with installing Python packages using pip:

- You may need or want to install additional Python packages.
- In order to install Python packages globally, one needs admin rights. Not suitable for us.
- Instead of global installations, local installations are possible. These local installations are then only available for the user who installed the respective package. In order to install locally, use the –user flag:

```
pip install --user scikit-image
```

- Consider: Installing packages costs you quota.

Problems with locally installed Python versions:

- The deep learning class is not the only class using Python. Thus you may have a local Python version installed, e.g., if you attended Pattern Recognition Deluxe.
- Since we are using the global install of python3, one can encounter some conflicts with an additional local install of Python, as we have experienced that multiple time.
- You can solve this conflict by getting rid of the local Python, i.e., by deleting the directory it is stored in.
- Note: You are the master of what you do with your account. We do not take any responsibilities if you delete one of your own files you shouldn't. Luckily nothing really can be seriously broken, since you typically have no admin rights.
- The most convenient way of deleting your local Python version is to remove the whole .local folder. In order to do this, execute the following command:

```
rm -r ~/.local
```

- Notice: /.local is also the place where all packages installed with the –user command are stored, i.e., they may have to be reinstalled.