HPC Computing at TUIL This documentation should help to get started with the High Performance Computing (HPC) cluster and GPU computing at our university.

Basics There are two possibilities to access the system, either via SSH with username@cslogin.tu-ilmenau.de or via an interactive desktop session to be started via the browser interface (just click on „Desktop Session“ to start a new one).
You need to be inside the university network if you want to access the system, so at home use the VPN. If you want to use the interactive Desktop Session, it is strongly recomended to install the DCV client and configure the „Viewer Type“ of interactive desktop sessions via „Settings“ to „Desktop Client“ in NICE (see https://www1.tu-ilmenau.de/hpcwiki/doku.php?id=rem_vis).

It is very important to know that there are two different „types“ of computing nodes: interactive nodes and non-interactive nodes. Interactive nodes can be accessed via SSH, to run something on non-interactive nodes you would need to use the batch system. Usually you only have access to the internet in **interactive nodes**. That means, you need to configure everything in the interactive nodes and then run it either via the batch system or on a interactive GPU node.

Lesson's learned - Python setup, Miniconda and tcsh It is strongly recommended to use Miniconda for a proper Python setup, see https://conda.io/projects/conda/en/latest/user-guide/install/linux.html

As the default shell is not bash, but tcsh, you need to run `conda init tcsh` after installation of Miniconda before you want to use Miniconda. Always remember if you run into some weird problems that your default shell is tcsh. It may be the case that applications that are added to your path are not used on the interactive nodes (because there already a pre-defined version is installed), but they will be most likely used on the interactive GPU node and nodes of the batch system.

Another important hint: Many python-based software need access to the internet if you run them for the first time. The reason is that sometimes some pre-trained model data needs to be downloaded. So it may be useful to execute the Python scripts on an interactive node with internet acces, before you execute the code in the batch system or on the interactive GPU node not having access to the internet.

Lesson's learned - pytorch If your software uses the pytorch package, it is strongly recommended to install one of the most recent pytorch versions with the pip command you can get from the official website https://pytorch.org/get-started/locally

A valid command to install pytorch would be e.g. `pip3 install torch==1.9.0+cu111 torchvision==0.10.0+cu111 torchaudio==0.9.0 -f https://download.pytorch.org/whl/torch_stable.html`

Take care to run your Python package installations on an interactive node, if you do it e.g. on makalu86 it won't work because you don't have internet access on this node. Even if your software uses an older pytorch version, it is usually worth it to try out a recent version supporting CUDA version >= 11.0. It is important to have a pytorch version with CUDA >= 11.0 as the NVIDIA A100 graphics cards only support CUDA >= 11.0.

Interactive GPU node The easiest way to get started with running some GPU-based computing stuff is to access the interactive GPU node. There is currently one interactive GPU node with 4x NVIDIA A100 graphics cards, the name of this machine is makalu86. **Important:** due to security reasons, **interactive GPU nodes don't have access to the internet** so you need to download/prepare everything on a standard interactive node and then execute your command „offline“ on the interactive GPU node. Further, you can access the interactive GPU nodes only if you

already are logged in into an interactive node (either via SSH or via an interactive Session).

To access an interactive GPU node, enter `ssh makalu86`, type in your CS login password and there you are. To check if you are on the correct node, enter `nvidia-smi`, four A100 graphics cards should appear. Now you can start to test your scripts, etc. Always bear in mind that the interactive GPU nodes are only there for the purpose of testing, you are not the only one using this interactive node. If you want to run something for a longer period of time or more than one GPU, it is recommended to use the batch system (cf. respective section).

Estimate efficiency of GPU usage On makalu86, `nvidia-smi` is for sure the best way to monitor how good your software is using the GPU. Here, the power usage is for sure the best way to find out how efficient your software is using the GPU. Another good indicator is the temperature, if the GPU is not busy it will have around 33°, starting from ~60° you can say the GPU is getting busy.

GPU Batch system Before you use the batch system, prepare a Shell script which will be then executed via the batch system. There already are a few pre-installed batch scripts, which usually can be directly used. If you want to start a batch job using 1 GPU, simply enter `batch.1gpu script.sh` and your batch job will be executed when there are free computing resources. Usually batch jobs are executed immediately after submission if you don't have „special requirements“ like more than 1 GPU, etc. If you want to start a batch job using 4 GPUs, enter `batch.4gpu script.sh` etc. If you want to get an overview of your batch jobs, enter `bjobs`.

In the directory where you started the batch job, you will also find a .log and .err file per job, giving you some information on stdout and stderr of your script.

You can also check the NICE –> Jobs view, ~20s after submitting your batch job it should also appear there. You can also check the output of your job there, but it won't automatically refresh.