Getting started with HPC (TinyGPU and Alex)

Documentation: https://hpc.fau.de/systems-services/documentation-instructions/

SSH Connection

If you are not in the university network, you might need to perform either step a) or b) first:

a) Connect to vpn.fau.de
b) Connect to the dialogserver: `ssh USERNAME@cshpc.rrze.fau.de`

Connect either to the TinyGPU or Alex cluster using PuTTY or a terminal:

```
ssh USERNAME@tinyx.nhr.fau.de
ssh USERNAME@alex.nhr.fau.de
```

You can either use PuTTY, a terminal, or use the SSH extension of Visual Studio Code.

File systems & data transfer

All clusters use Linux operating systems. The standard directory at login is called `$HOME`. Here you can store important files e.g. your python script. You can store larger files at `$WORK`. You can read more about the file systems here.

For copying data from your local machine to the cluster, you can use `scp` or WinSCP. Download WinSCP here: https://winscp.net/eng/download.php

Check available GB using: `shownicerquota.pl`

Setting up an Anaconda environment

There are anaconda installations provided as modules on HPC. List available modules:

```
module avail
module avail python
```

Next, you can load the python module e.g.

```
module load python/3.9-anaconda
```

In order to use conda environments on HPC, you need to create a *.profile* and *.condarc* file. Please follow these instructions. Now you can import an existing environment or create a new one using:

```
conda create ...
```

Job script

For submitting a job to the cluster, you should write a job script i.e. a bash file. The following job script runs a job “example” on a single GPU on the cluster:

example_jobscript.sh

This job “example” will run max. 8h. The time limit for jobs is 24h.

Using one GPU Nvidia A100. For multi-GPU, set `gres=gpu:a100:2`. For multi-processing, set e.g. `cpus-per-task=8`

Here, a folder “output” was created in $HOME beforehand. The outputs and errors are written to this folder.

Activate conda environment that was created beforehand

Configuring paths. This might need to be set for TensorFlow

Execute python script

```
#!/bin/bash -l
#SBATCH --job-name=example
#SBATCH --time=8:00:00
#SBATCH --partition=a100
#SBATCH --gres=gpu:a100:1
#SBATCH --cpus-per-task=1
#SBATCH -o /home/hpc/…/username/output/slurm-%j.out
#SBATCH -e /home/hpc/…/username/output/slurm-%j.err
#SBATCH --export=NONE
unset SLURM_EXPORT_ENV

module load python/3.9-anaconda
source activate my_env
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$CONDA_PREFIX/lib/


python myscript.py
```

In this example, all required packages such as CUDA were already installed in the conda environment “my_env”. Alternatively, you can use CUDA and cuDNN versions that are preinstalled on HPC. This works by adding `module load cuda/versionxy` and `module load cudnn/versionxy` in the job script.

In case you have trouble setting up an environment with TensorFlow and CUDA, these versions have worked for me:

```
conda install -c conda-forge cudatoolkit=11.2 cudnn=8.1.0
pip install --upgrade pip
pip install tensorflow==2.9.2
```

The job time is limited to 24 hours. However, extending the time limit should be possible by adding `#SBATCH --qos=a100_aibe` in the job script.

Submitting a job

Submit a job using the following command. To submit a job to the TinyGPU cluster, add “tinygpu”:

```
sbatch example_jobscript.sh
sbatch.tinygpu example_jobscript.sh
```

See queue of jobs (ST = State: R = Running, PD = Pending, CG = Completing):

```
squeue
```

Cancel a job:

```
scancel JOB_ID
```

How to check GPU utilization of a running job is explained here.
You can also monitor jobs using the ClusterCockpit.

Run an interactive job

Instead of submitting a job, you can also get an interactive shell on a compute node. Use the command `salloc` or `salloc.tinygpu` to open the interactive shell. The allocation will be revoked as soon as you close the terminal.

Example:

```
salloc --job-name=test --partition=a40 --time=00:30:00 --gres=gpu:a40:1
```
(wait until resources become available)
```
module load python
conda activate my_env
...
```

Further information

Getting started:
https://hpc.fau.de/systems-services/documentation-instructions/getting-started/

Python:
https://hpc.fau.de/systems-services/documentation-instructions/special-applications-and-tips-tricks/python-and-jupyter/

HPC in a nutshell:
https://www.rrze.fau.de/files/2019/05/2019-04-26_HPC_in_a_Nutshell1.pdf
https://www.rrze.fau.de/files/2019/05/2019-05-09_HPC_in_a_Nutshell2-2.pdf

Job scripts:
https://hpc.fau.de/systems-services/documentation-instructions/batch-processing/
https://www.aibe.wiki.fau.de/index.php?title=Hilfe:HPC_Introduction&title=Hilfe%3AHPC_Introduction