Handout

Working in Jupyter

PC2 JupyterHub URL: jh.pc2.uni-paderborn.de

Log in with your credentials (same as for SSH access, e.g. `usrtr028`).

To spawn a Jupyter server, select a preset like e.g. “NHR Courseweek - Julia for HPC (32 cores, 1 GPU)”.

Most important commands

- Evaluate a cell: `Ctrl`+`Enter`
- Evaluate a cell and move to next: `Shift`+`Enter`
- Create a new cell below: `Esc B`
- Delete a cell: `Esc X`

Loading sofware modules

On the left side, on the vertical panel, click on the little blue hexagon (“Software”). Then type, e.g. “kernel” into the search bar at the top to find the Jupyter kernel modules. Hover over a module and click “load” on the right to load the module.

For “**Julia for HPC**” you will need to load the module “JupyterKernel-Julia/1.9.0-foss-2022a-CUDA-11.7.0”.

For “**Data Parallel Neural Networks**” you will need to load the module “JupyterKernel-Python/3.10.4-torchvision-0.13.1-CUDA-11.7.0”.

Working in the Terminal

SSH Access

```
ssh <username>@training.pc2.upb.de
```

then

```
ssh n2login2.ab2021.pc2.uni-paderborn.de
```

You should work in the directory `/scratch/hpc-prf-nhrgs/<username>`, where `<username>` is e.g. `usrtr028`.

Software modules

For "**Julia for HPC**" you will need to load

```
module load lang/JuliaHPC/1.9.0-foss-2022a-CUDA-11.7.0
```

For "**Data Parallel Neural Networks**" you will need to load

```
module load vis/torchvision/0.13.1-foss-2022a-CUDA-11.7.0
```

Starting Julia We've already pre-installed the course environment for you such that all Julia packages are available if you run Julia via `julia --project` inside of the course folder `/scratch/hpc-prf-nhrgs/<username>/julia-nhr-courseweek-2023` (or any of its subfolders).

SLURM allocations / Compute-node sessions

Interactive

To get an interactive session on a Noctua 2 compute node (e.g. 1 hour, 32 CPU-cores) run

```
srun -A hpc-prf-nhrgs -N 1 -n 1 -c 32 -p normal -t 1:00:00 --pty bash
```

If you want a similar session with an additional NVIDIA A100 GPU run

```
srun -A hpc-prf-nhrgs -N 1 -n 1 -c 32 --gres=gpu:a100:1 -p gpu -t 1:00:00
    --pty bash
```

Job submission

If you want to submit a non-interactive job, you first need to create a job file (see example below).

```
#!/bin/bash
#SBATCH -J myjob
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -c 128
#SBATCH --account=hpc-prf-nhrgs
#SBATCH --partition=normal
#SBATCH -t 00:15:00

# load necessary modules
module reset
module load lang/JuliaHPC/1.9.0-foss-2022a-CUDA-11.7.0

# run program (with 128 threads)
julia --project -t 128 your_program.jl
```

This requests a single node (`N`) for 15 minutes (`t`) on which we'll run 1 process (`n`) using 128 CPU-cores (`c`). If you need an entire node exclusively for yourself, add `#SBATCH --exclusive`. To submit this job to the scheduler use `sbatch`, e.g. `sbatch myjob.sh`. With `squeue` you can get a list of your scheduled/running jobs.

SLURM resource reservations

There are two SLURM reservations for this week: `nhrgs` (GPU nodes with 128 cores and 4xA100 each) and `nhrgscpu` (CPU nodes with 128 cores). In principle, these should be automatically used such that you don't have to wait much for your job to start. However, if nonetheless necessary, you can specify one of these reservations explicitly in your resource allocation by adding e.g. `--reservation=nhrgs`.