



# Foundations of LLM Mastery: Fine-tuning with multi GPUs

25 February 2025  
ONLINE



# How to use a supercomputer

SLURM Job Scheduler

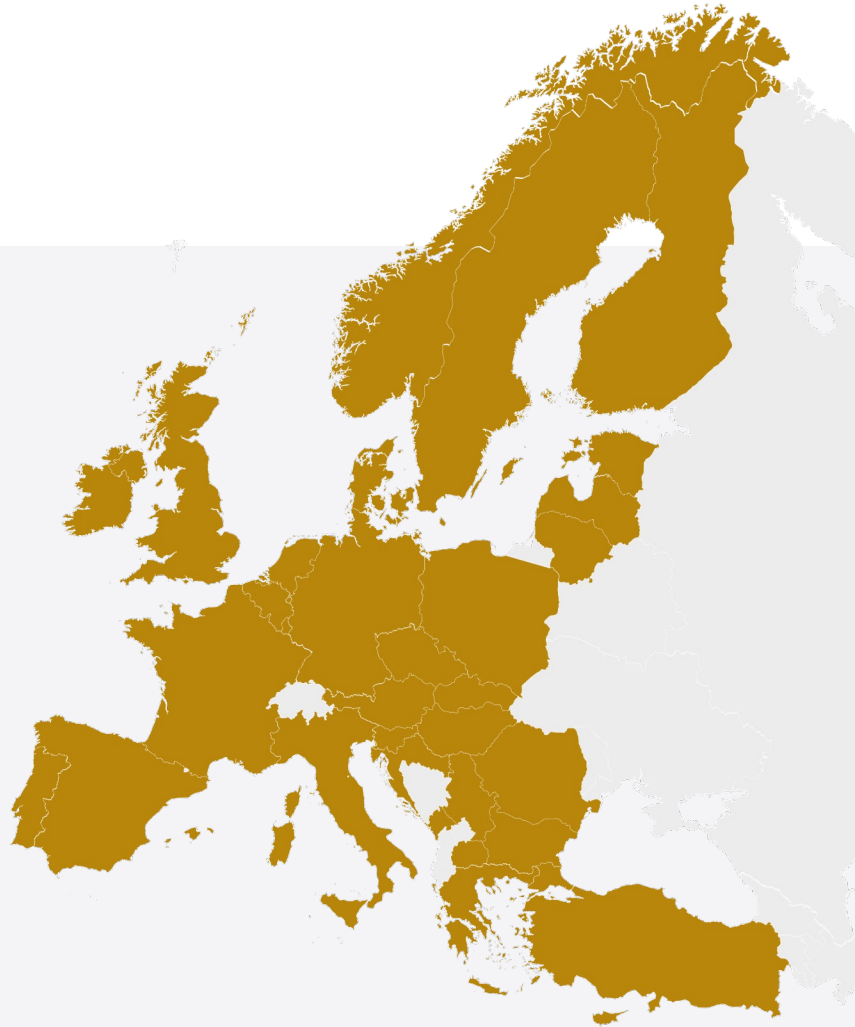
---

Speaker: Martin Pfister  
HPC / AI Team, EuroCC Austria

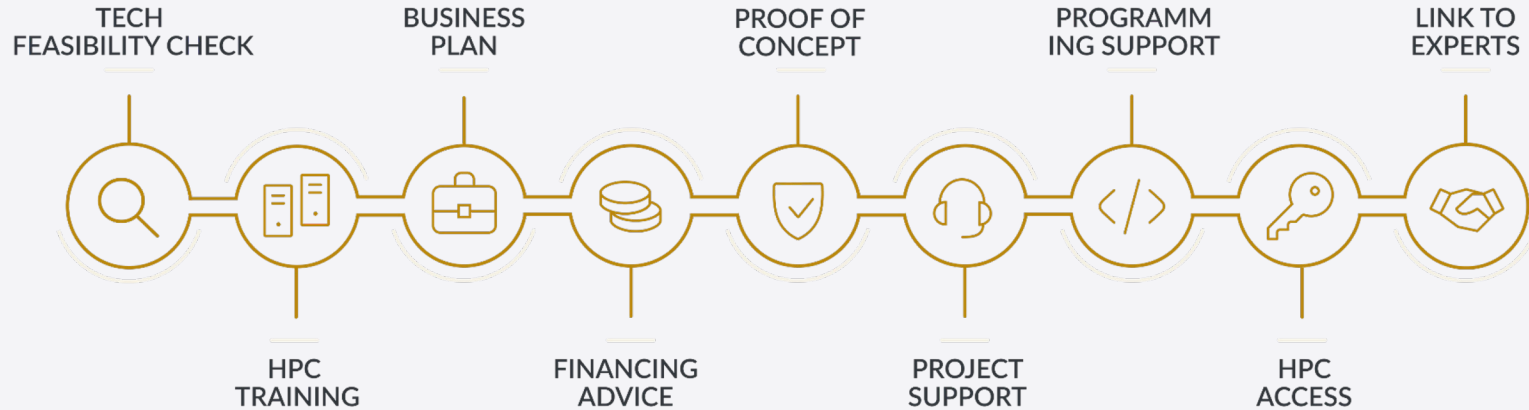
# EuroCC

## Fully funded EU project

- EuroCC is EU and nationally funded (50/50) international initiative aimed to support the uptake of AI and High-Performance Computing (HPC) in Europe
- Set up of 32 National Competence Centres (NCCs) across Europe
- EuroCC Austria is one of them
- Service Provider for AI, HPC and HPDA



# EuroCC Austria's Services



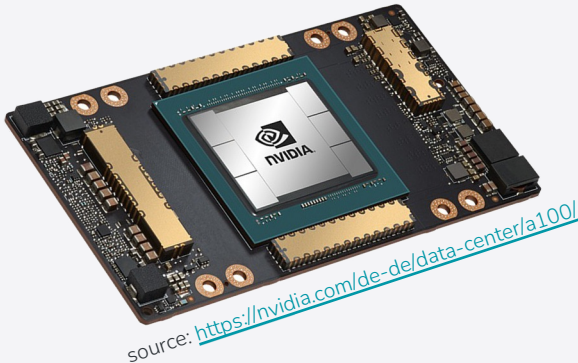
CONSULTING – TRAINING – INFRASTRUCTURE

# The Vienna Scientific Cluster

## VSC-4 (2019)

790 CPU nodes

- 2x Intel Skylake Platinum CPUs
- 2x 24 cores per CPU
- 96 GB of memory per node



source: <https://nvidia.com/de-de/data-center/a100/>

## VSC-5 (2022)

770 CPU nodes

- 2x AMD EPYC Milan
- 2x 64 cores per CPU
- 512 GB of memory per node

60 GPU nodes 2x NVIDIA A100,

- 40 GB memory per GPU

40 GPU nodes 2x NVIDIA A40

- 48 GB memory per GPU

# Need More Compute Power?

## LUMI

- #8 in Top500
- Linpack: 380 PFlop/s
- AMD EPYC CPUs
- AMD Instinct MI250X GPUs (128 GB)

<https://www.lumi-supercomputer.eu/>

## Leonardo

- #9 in Top500
- Linpack: 240 PFlos/p
- Intel Xeon CPUs
- NVIDIA A100 GPUs (64GB)

<https://leonardo-supercomputer.cineca.eu/>

# Supercomputers in Europe

## EuroHPC JU systems

Apply for access to EuroHPC supercomputers

Different access modes:

- Benchmark Access
- Development Access
- ...
- Extreme Scale Access

<https://eurohpc-ju.europa.eu/>



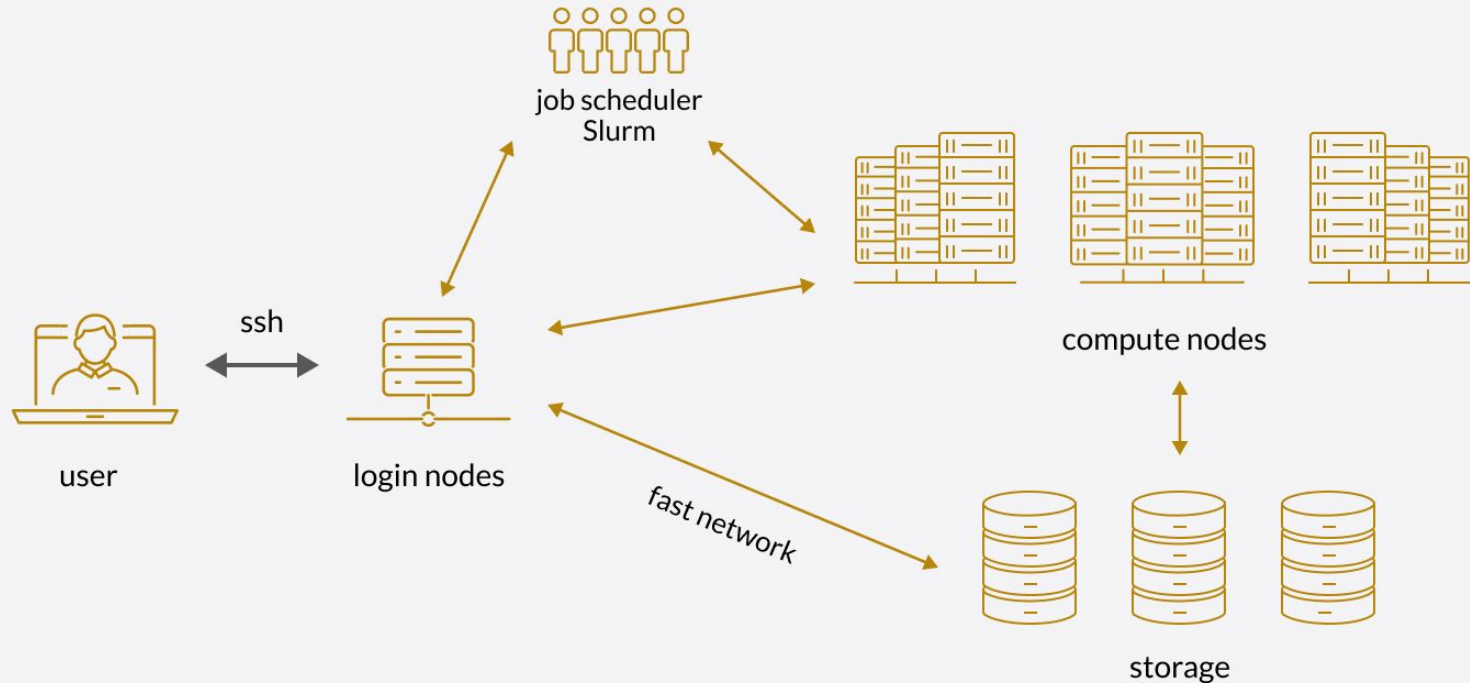
# Supercomputers in Europe

- VSC5 (60 nodes with 2 Nvidia A100 40 GB)
- LEONARDO (3456 nodes with 4 Nvidia A100 64 GB)
- LUMI-G (2978 nodes with 4 AMD MI250x 128 GB)
- MUSICA (~200 nodes with 4 Nvidia H100 96 GB)

[https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers\\_en](https://eurohpc-ju.europa.eu/supercomputers/our-supercomputers_en)



# Typical Setup of a Supercomputer



# SLURM: Job Scheduler

## Leonardo job script (1 GPU)

```
#!/bin/bash
#SBATCH --partition=boost_usr_prod
#SBATCH --qos=boost_qos_dbg      # High priority QOS. Remove line for normal priority.
#SBATCH --nodes=1                # Number of nodes
#SBATCH --ntasks-per-node=1      # Number of `srun` tasks executed per node
#SBATCH --gpus-per-task=1        # Number of GPUs (up to 4 on Leonardo)
#SBATCH --mem=120GB              # Fair share on Leonardo: 120GB * gpus-per-task
#SBATCH --cpus-per-task=8        # Fair share on Leonardo: 8 * gpus-per-task
#SBATCH --time=0:30:00           # Time limit in HH:MM:SS, up to 0:30:00 for boost_qos_dbg
```

<whatever command should be executed on the compute node>

# SLURM: Job Scheduler

## Leonardo job script (1 GPU)

```
#!/bin/bash
#SBATCH --partition=boost_usr_prod
#SBATCH --qos=boost_qos_dbg      # High priority QOS. Remove line for normal priority.
#SBATCH --nodes=1                # Number of nodes
#SBATCH --ntasks-per-node=1      # Number of `srun` tasks executed per node
#SBATCH --gpus-per-task=1        # Number of GPUs (up to 4 on Leonardo)
#SBATCH --mem=120GB              # Fair share on Leonardo: 120GB * gpus-per-task
#SBATCH --cpus-per-task=8        # Fair share on Leonardo: 8 * gpus-per-task
#SBATCH --time=0:30:00           # Time limit in HH:MM:SS, up to 0:30:00 for boost_qos_dbg

module purge                     # Start in a clean environment
module load anaconda3            # Load conda
eval "$(conda shell.bash hook)"  # Initialize conda
conda activate /leonardo/pub/userexternal/mpfister/conda_env_martin24

python3 script.py
```

# SLURM: Job Scheduler

## Leonardo job script (2 GPU)

```
#!/bin/bash
#SBATCH --partition=boost_usr_prod
#SBATCH --qos=boost_qos_dbg      # High priority QOS. Remove line for normal priority.
#SBATCH --nodes=1                # Number of nodes
#SBATCH --ntasks-per-node=1      # Number of `srun` tasks executed per node
#SBATCH --gpus-per-task=2        # Number of GPUs (up to 4 on Leonardo)
#SBATCH --mem=240GB              # Fair share on Leonardo: 120GB * gpus-per-task
#SBATCH --cpus-per-task=16       # Fair share on Leonardo: 8 * gpus-per-task
#SBATCH --time=0:30:00           # Time limit in HH:MM:SS, up to 0:30:00 for boost_qos_dbg

module purge                     # Start in a clean environment
module load anaconda3            # Load conda
eval "$(conda shell.bash hook)"  # Initialize conda
conda activate /leonardo/pub/userexternal/mpfister/conda_env_martin24

python3 script.py
```

# SLURM: Job Scheduler

## Leonardo job script (4 GPU)

```
#!/bin/bash
#SBATCH --partition=boost_usr_prod
#SBATCH --qos=boost_qos_dbg      # High priority QOS. Remove line for normal priority.
#SBATCH --nodes=1                # Number of nodes
#SBATCH --ntasks-per-node=1      # Number of `srun` tasks executed per node
#SBATCH --gpus-per-task=4        # Number of GPUs (up to 4 on Leonardo)
#SBATCH --mem=480GB              # Fair share on Leonardo: 120GB * gpus-per-task
#SBATCH --cpus-per-task=32       # Fair share on Leonardo: 8 * gpus-per-task
#SBATCH --time=0:30:00           # Time limit in HH:MM:SS, up to 0:30:00 for boost_qos_dbg

module purge                     # Start in a clean environment
module load anaconda3            # Load conda
eval "$(conda shell.bash hook)"  # Initialize conda
conda activate /leonardo/pub/userexternal/mpfister/conda_env_martin24

python3 script.py
```

# SLURM: Job Scheduler

## Leonardo job script (2 nodes with 4 GPUs each)

```
#!/bin/bash
#SBATCH --partition=boost_usr_prod
#SBATCH --qos=boost_qos_dbg          # High priority QOS. Remove line for normal priority.
#SBATCH --nodes=2                    # Number of nodes
#SBATCH --ntasks-per-node=1          # Number of `srun` tasks executed per node
#SBATCH --gpus-per-task=4            # Number of GPUs (up to 4 on Leonardo)
#SBATCH --mem=480GB                  # Fair share on Leonardo: 120GB * gpus-per-task
#SBATCH --cpus-per-task=32           # Fair share on Leonardo: 8 * gpus-per-task
#SBATCH --time=0:30:00               # Time limit in HH:MM:SS, up to 0:30:00 for boost_qos_dbg

module purge                         # Start in a clean environment
module load anaconda3                # Load conda
eval "$(conda shell.bash hook)"      # Initialize conda
conda activate /leonardo/pub/userexternal/mpfister/conda_env_martin24

srun python3 script.py               # Or replace python3 with torchrun or accelerate launch
```

# SLURM: Job Scheduler

## Useful SLURM commands

# Submit a job:

\$ sbatch job.sh

# Check submitted jobs:

\$ squeue --me

# Look at the output from a job:

\$ cat slurm-<job\_id>.out

# or follow the output as the job runs:

\$ tail -c +0 -f slurm-<job\_id>.out

# Cancel job:

\$ scancel <job\_id>

# Get a shell at a node while a job is running:

\$ ssh <compute\_node\_name>

# or:

\$ srun --overlap --pty -jobid=<job\_id> bash