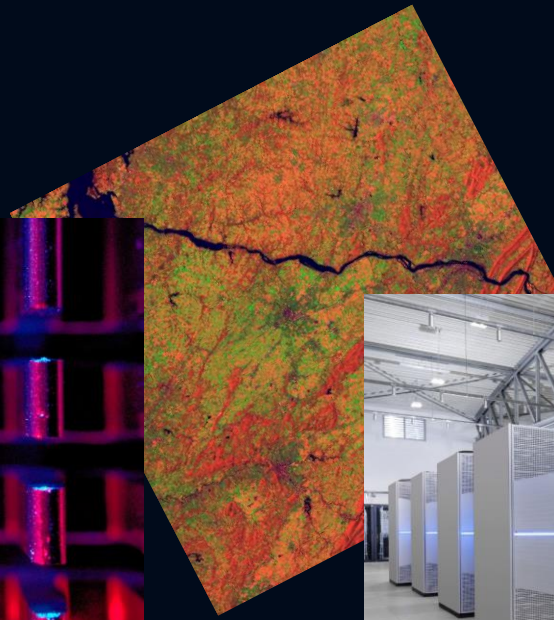
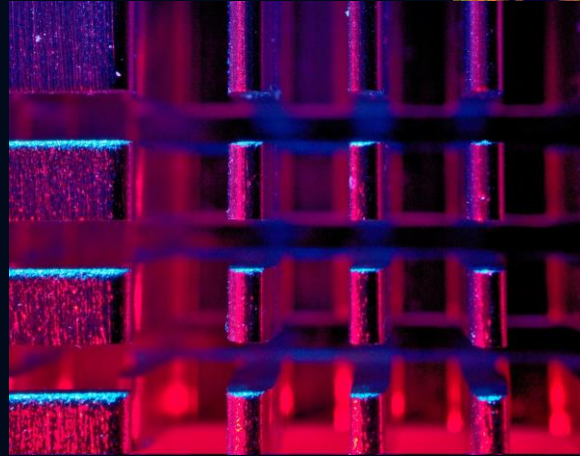




# Hands-on Distributed Deep Learning

IEEE IGARSS Tutorial on Scalable Machine Learning with High Performance and Cloud Computing  
July 10-11, 2021

ROCCO SEDONA, PHD STUDENT

HIGH PRODUCTIVITY DATA PROCESSING RESEARCH GROUP

JÜLICH SUPERCOMPUTING CENTRE

UNIVERSITY OF ICELAND

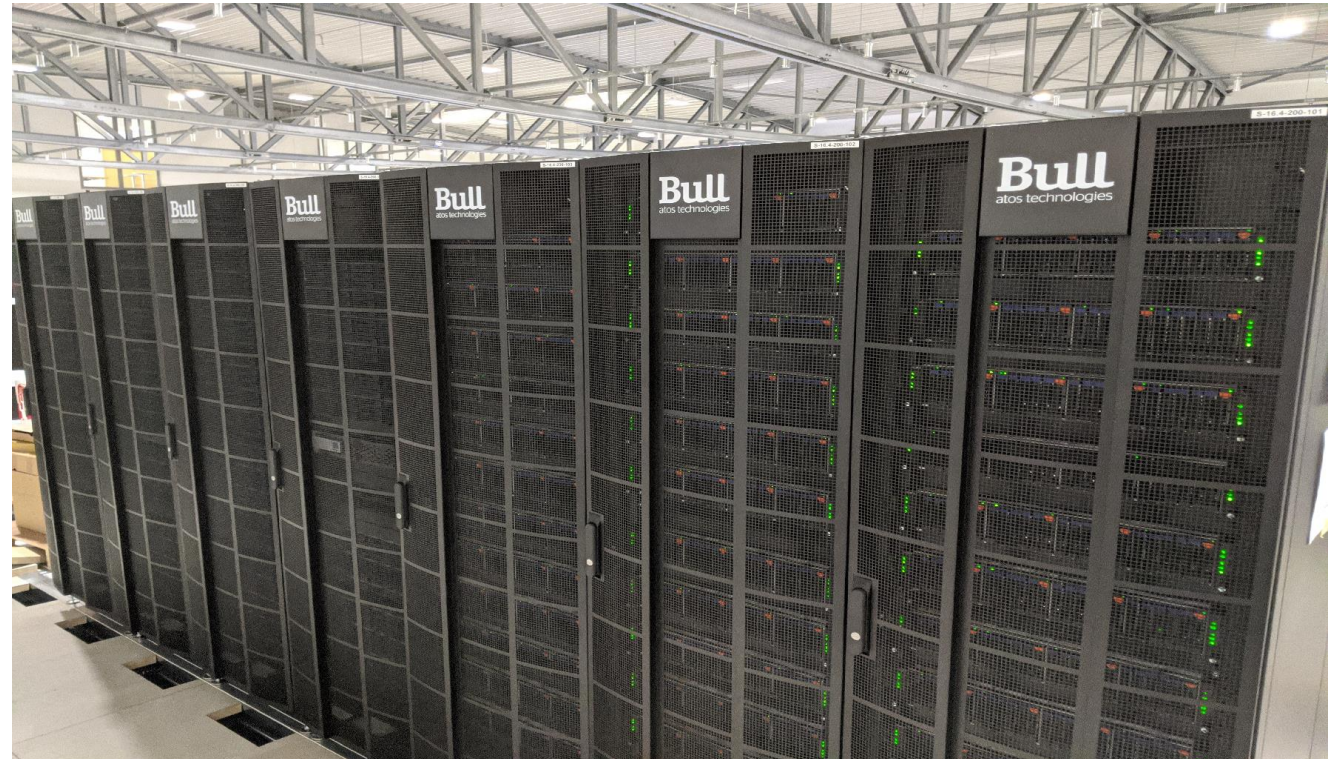
# JUSUF

144 standard compute nodes

**61 accelerated compute nodes**

- 2× AMD EPYC 7742, 2× 64 cores, 2.25 GHz
- 256 (16× 16) GB DDR4, 3200 MHz
- InfiniBand HDR100 (Connect-X6)
- local disk for operating system (1× 240 GB SSD)
- 1 TB NVMe
- 1× NVIDIA V100 GPU with 16 GB HBM2e

4 login nodes

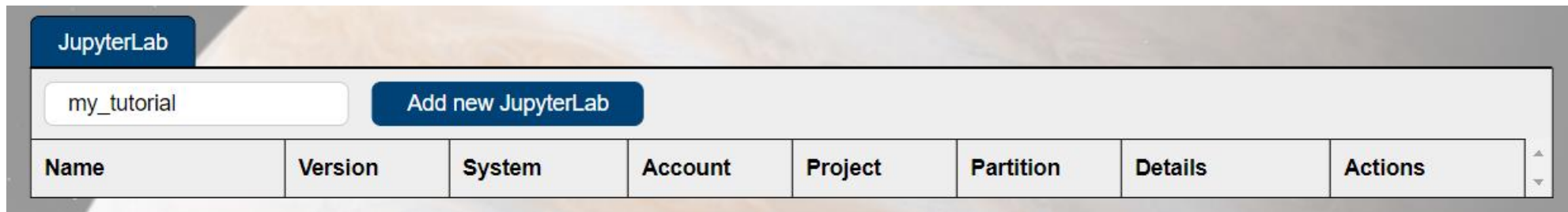


# Instructions I

Open Jupyter-JSC and log in:

<https://jupyter-jsc.fz-juelich.de/>

Add a new Jupyter Lab



The screenshot shows the JupyterLab interface. At the top, there is a blue button labeled "JupyterLab". Below it, there is a text input field containing "my\_tutorial" and a blue button labeled "Add new JupyterLab". Below these elements is a table with the following columns: Name, Version, System, Account, Project, Partition, Details, and Actions. The table is currently empty of data rows.

| Name | Version | System | Account | Project | Partition | Details | Actions |
|------|---------|--------|---------|---------|-----------|---------|---------|
|------|---------|--------|---------|---------|-----------|---------|---------|

# Instructions II

## JupyterLab Options

|           |                |
|-----------|----------------|
| Version   | JupyterLab ▼   |
| System    | JUSUF ▼        |
| Account   | sedona3 ▼      |
| Project   | training2118 ▼ |
| Partition | LoginNode ▼    |

Start

# Instructions III

Navigate to your working folder

```
[sedona3@jsfl01 ~]$ cd /p/project/training2118/$USER  
[sedona3@jsfl01 sedona3]$ █
```

Copy exercises into your working folder

```
[sedona3@jsfl01 sedona3]$ cp -r /p/project/training2118/exercise_* .
```

```
[sedona3@jsfl01 exercise_2]$ squeue -u $USER  
[sedona3@jsfl01 exercise_2]$ sbatch submit job iusuf.sh ID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

# Instructions Exercises

## Submit job script (SLURM command)

```
[sedona3@jsfl01 exercise_2]$ sbatch submit_job_jusuf.sh  
Submitted batch job 106384  
[sedona3@jsfl01 exercise_2]$
```

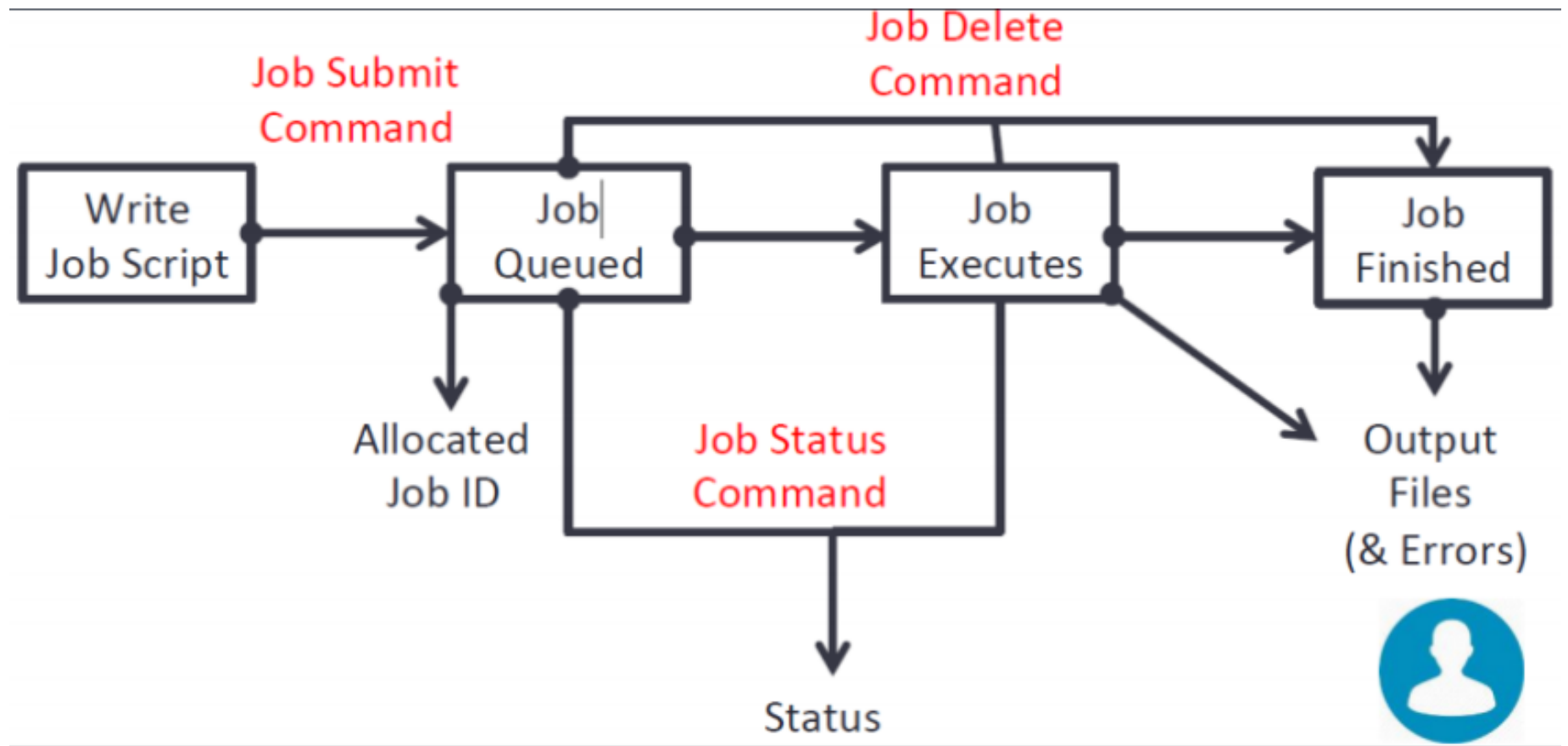
## Check job status

```
[sedona3@jsfl01 exercise_2]$ squeue -u $USER
```

| JOBID  | PARTITION | NAME     | USER    | ST | TIME | NODES | ODELIST(REASON) |
|--------|-----------|----------|---------|----|------|-------|-----------------|
| 106382 | gpus      | HOROVOD_ | sedona3 | PD | 0:00 | 4     | (Reservation)   |

# Resource Access

- Slurm



# Job script I

```
#!/usr/bin/env bash
```

shebang, what interpreter to use

```
# Slurm job configuration
```

```
#SBATCH --nodes=4
```

```
#SBATCH --ntasks-per-node=1
```

```
#SBATCH --account=training2118
```

```
#SBATCH --output=bench_tf2_4gpus.out
```

```
#SBATCH --error=bench_tf2_4gpus.er
```

```
#SBATCH --time=0:30:00
```

```
#SBATCH --job-name=HOROVOD_benchmark
```

```
#SBATCH --gres=gpu:1 --partition=gpus
```

```
#SBATCH --mail-type=ALL
```

```
#SBATCH --reservation=tutorial-2021-07-11
```

remove reservation after the tutorial

# Job script II

# Load the required modules

ml Stages/2020 GCC/9.3.0 OpenMPI/4.1.0rc1

ml Horovod/0.20.3-Python-3.8.5

ml TensorFlow/2.3.1-Python-3.8.5

#activate the virtual environment

source /p/project/training2118/.env\_tutorial\_jusuf/bin/activate:

# Run the program in parallel

srun --cpu-bind=none,v --accel-bind=gn python -u tensorflow2\_synthetic\_benchmark.py

# Horovod

1. Initialize Horovod
2. Pin each GPU to a single process
3. Scale the learning rate by the number of workers.
4. Wrap the optimizer with the distributed optimizer function
5. Broadcast the initial variable states from rank 0 to all other processes
6. Modify your code to save checkpoints only on worker 0 to prevent other workers from corrupting them.



[2]

# EUROSAT

- Patches extracted from Sentinel-2 tiles
- 27000 labeled samples
- 10 classes
- 13 bands (RGB only also available)
- Patch based land use and land cover classification



# References

- [1] [https://fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUSUF/JUSUF\\_node.html](https://fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUSUF/JUSUF_node.html)
- [2] <https://horovod.readthedocs.io/en/stable/tensorflow.html>
- [3] <https://github.com/phelber/EuroSAT>
- [4] Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. Patrick Helber, Benjamin Bischke, Andreas Dengel, Damian Borth. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019.