

Deep Learning Optimized - Jean Zay

Introduction – Jean Zay – GPU

cnrs IDRIS

CC BY NC SA

DLO-JZ presentation

IDRIS ◂

Agenda ◂

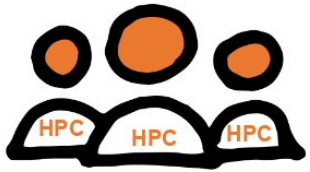
Presentation of the participants ◂

IDRIS

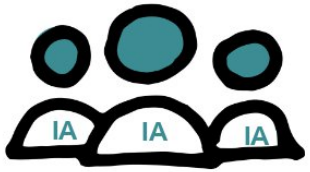
System



User Assistance



10 ingénieur·e·s



10 ingénieur·e·s

BLOOM on Jean Zay



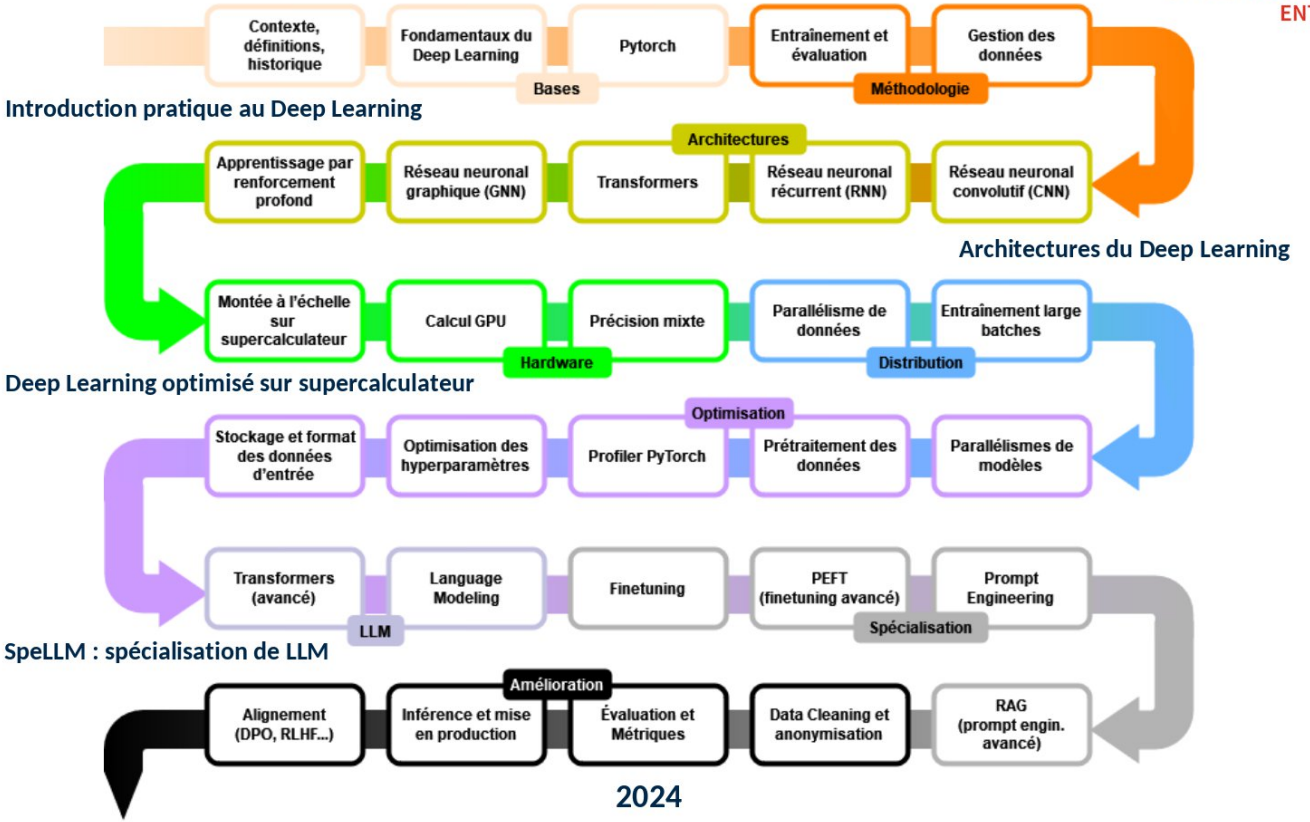
Pre-training :
117 days (2022)
384 x 80GB A100 GPUs

Train modulaire des formations IDRIS

IDRIS

Les formations IA

Pour les inscriptions ou une formation sur mesure contacter
CNRS FORMATION ENTREPRISES







Spécialisation des LLM

cnrs



DLO-JZ & SpeLLM 

Learn about LLM specializing to meet specific needs.

Learn to make a prototype in 4 days → The training is hands-on centered

1st day
Transformers
Classic Fine-Tuning
Dev environment



2nd day
Metrics and Evaluations
Data Cleaning
Prompt Engineering

3rd day
Retrieval Augmentation Generation
Parameter Efficient Fine-Tuning
Alignment

4th day
Inference
Multi-modality
Agents



DLO-JZ

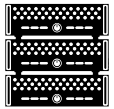
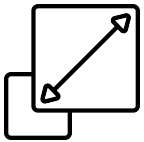
Agenda – Covered topics

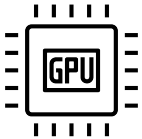
Day 1
- Presentation of Jean Zay
- The challenges of scaling up
- **GPU computing + Mixed Precision**
- **Tensor optimisation**
- The PyTorch profiler

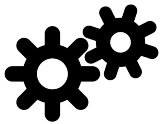
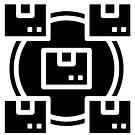
Day 2
- **DataLoader optimizations**
- **Distribution + Data Parallelism**
- **Data Augmentation**

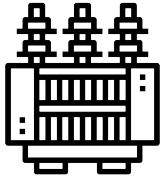
Day 3
- **Data storage and formats**
- **Training and large batches**
- JIT

Day4
- Good practices
- **Large Model Scaling + FSDP**
- **Model parallelisms**
- Visualization tools
- **HyperParameter Optimization**

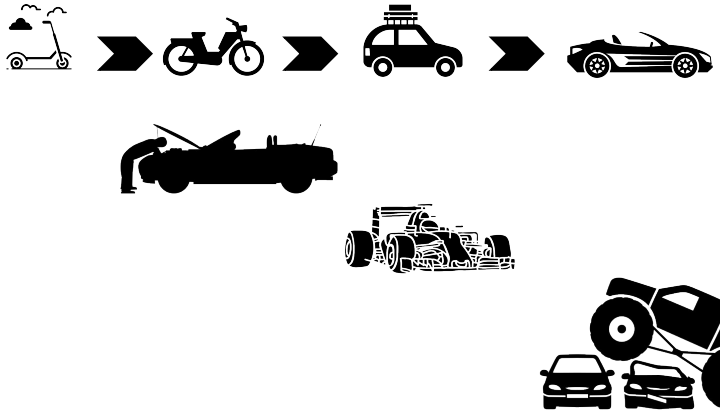





Practical Workshop

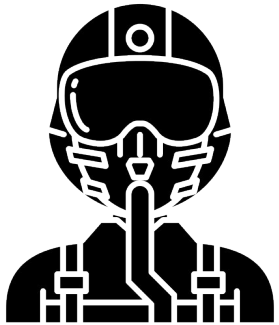
- Day 1, 2, 3, 4 :
  - System Optimization : GPU, Mixed Precision
  - Profiler
  - DataLoader
  - Distributed Data Parallelism
  - Data Augmentation
  - *Optimizer*s & LR scheduler
  - torch.compile & torchscript
  - *FSDP*
  - *Hyper-Parameters Optimization (HPO)*



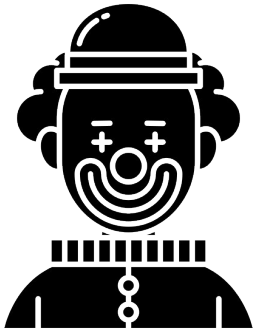
- Reservation partition A100 JZ



Presentation of participants









Jean Zay

Supercomputer ◂

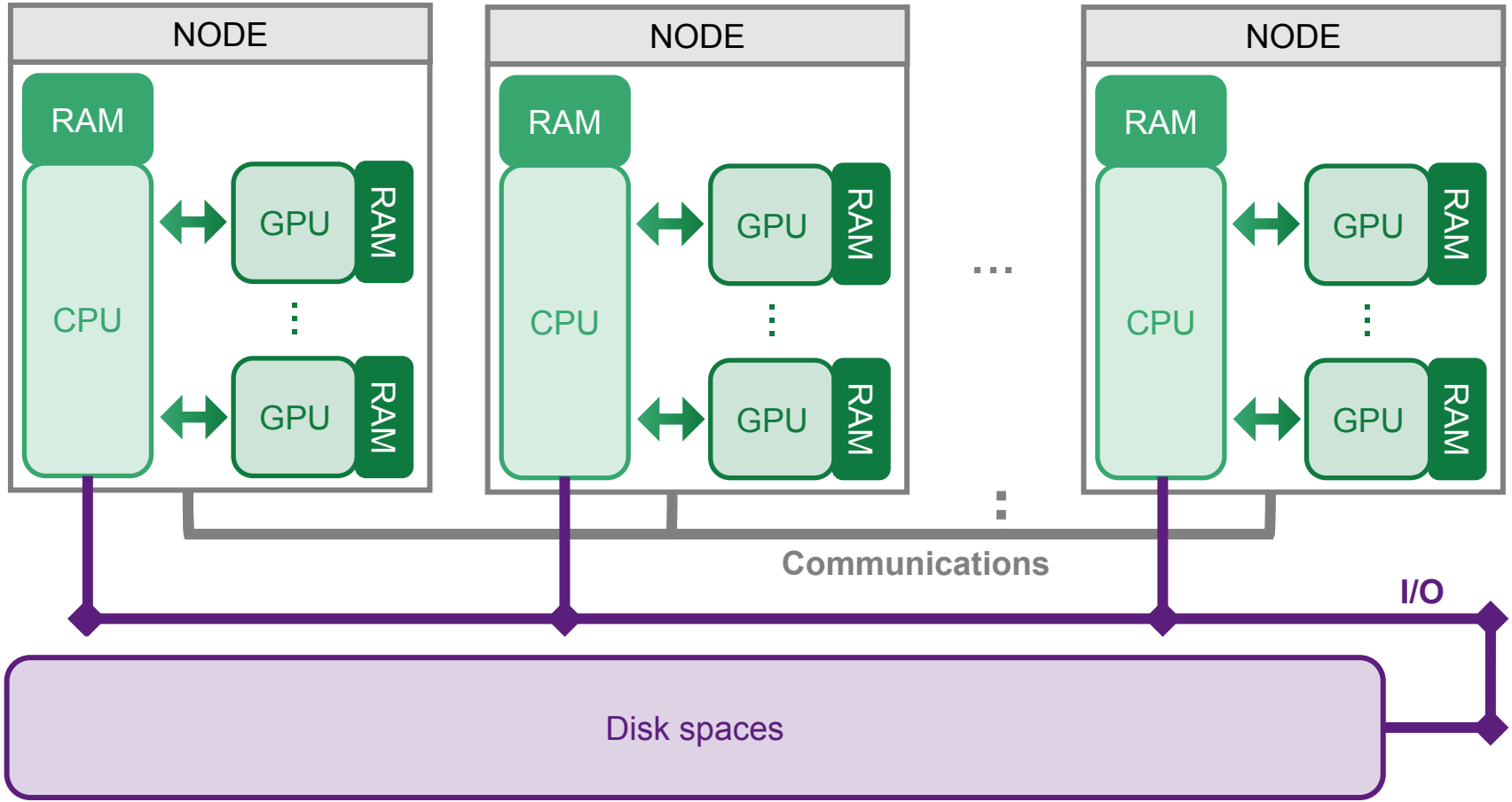
Jean Zay ◂

Job submission with Slurm ◂

JupyterHub on Jean Zay ◂

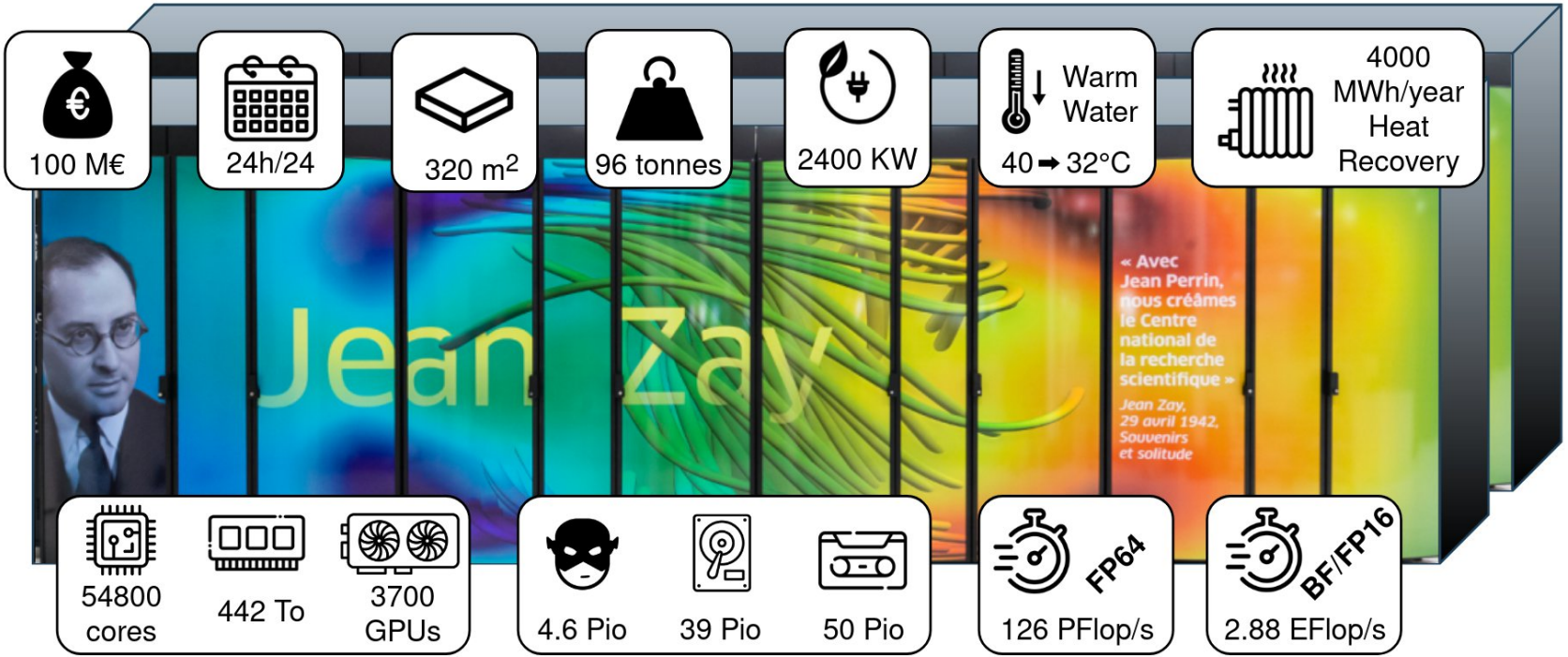
Slurm tools for python notebooks ◂

What's a supercomputer?

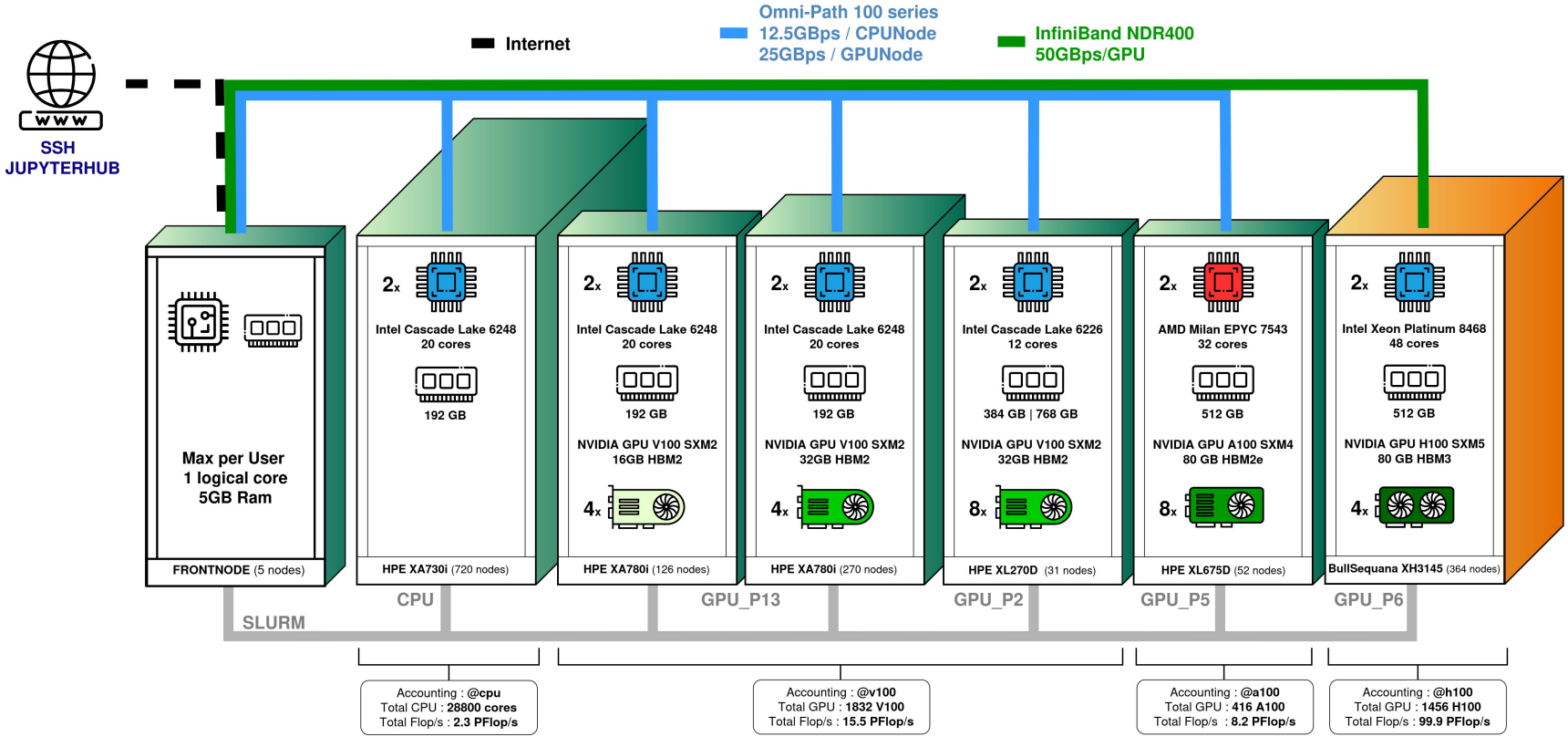


Jean Zay

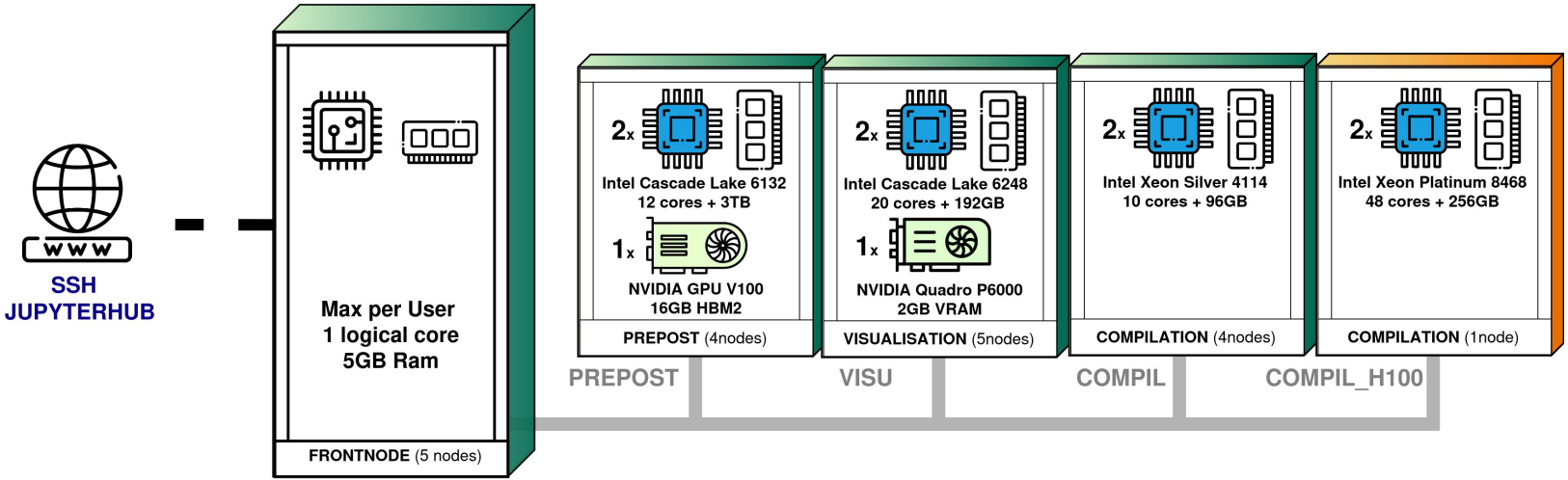
First national converged supercomputer dedicated to Artificial Intelligence (AI) and High Performance Computing (HPC)



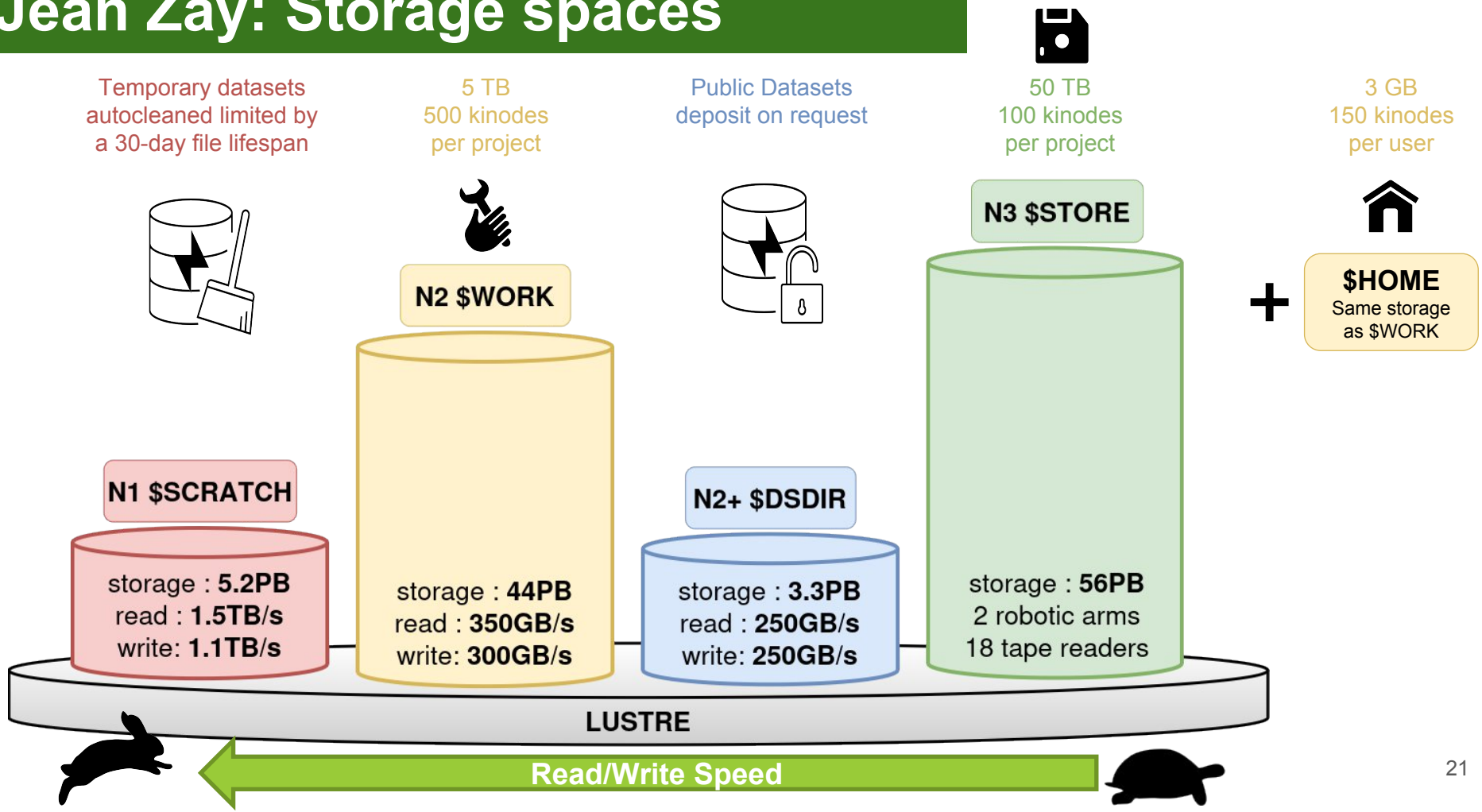
Jean Zay


Internet
Omni-Path 100 series
12.5GBps / CPUNode
25GBps / GPUNode
InfiniBand NDR400
50GBps/GPU
www
SSH
JUPYTERHUB
Max per User
1 logical core
5GB Ram
FRONTNODE (5 nodes)
2x
Intel Cascade Lake 6248
20 cores
192 GB
HPE XA730i (720 nodes)
CPU
2x
Intel Cascade Lake 6248
20 cores
192 GB
NVIDIA GPU V100 SXM2
16GB HBM2
4x
HPE XA780i (126 nodes)
GPU_P13
2x
Intel Cascade Lake 6248
20 cores
192 GB
NVIDIA GPU V100 SXM2
32GB HBM2
4x
HPE XA780i (270 nodes)
2x
Intel Cascade Lake 6226
12 cores
384 GB | 768 GB
NVIDIA GPU V100 SXM2
32GB HBM2
8x
HPE XL270D (31 nodes)
GPU_P2
2x
AMD Milan EPYC 7543
32 cores
512 GB
NVIDIA GPU A100 SXM4
80 GB HBM2e
8x
HPE XL675D (52 nodes)
GPU_P5
2x
Intel Xeon Platinum 8468
48 cores
512 GB
NVIDIA GPU H100 SXM5
80 GB HBM3
4x
BullSequana XH3145 (364 nodes)
GPU_P6
SLURM
Accounting : @cpu
Total CPU : 28800 cores
Total Flop/s : 2.3 PFlop/s
Accounting : @v100
Total GPU : 1832 V100
Total Flop/s : 15.5 PFlop/s
Accounting : @a100
Total GPU : 416 A100
Total Flop/s : 8.2 PFlop/s
Accounting : @h100
Total GPU : 1456 H100
Total Flop/s : 99.9 PFlop/s

Jean Zay


www
SSH
JUPYTERHUB
Max per User
1 logical core
5GB Ram
FRONTNODE (5 nodes)
2x
Intel Cascade Lake 6132
12 cores + 3TB
1x
NVIDIA GPU V100
16GB HBM2
PREPOST (4nodes)
2x
Intel Cascade Lake 6248
20 cores + 192GB
1x
NVIDIA Quadro P6000
2GB VRAM
VISUALISATION (5nodes)
2x
Intel Xeon Silver 4114
10 cores + 96GB
COMPILATION (4nodes)
2x
Intel Xeon Platinum 8468
48 cores + 256GB
COMPILATION (1node)
PREPOST
VISU
COMPIL
COMPIL_H100

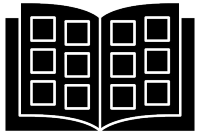
Jean Zay: Storage spaces



Jean Zay: Work environment

ENVIRONMENT MODULES

Catalogue of shared modules (conda environments)



- Installed by IDRIS
- Completed on request

```
login@jean-zay3:~$ module load pytorch-gpu/py3/1.11.0
  Loading requirement: ...
(pytorch-gpu-1.11.0+py3.9.12) login@jean-zay3:~$
```

- Customizable

```
~$ pip install --user --no-cache-dir <paquet>
```

Conflicts between versions
Storage spaces saturation

Personal conda environments

```
login@jean-zay3:~$ module load anaconda-py3/2023.03
(base) login@jean-zay3:~$ conda create -n myenv
```

Storage spaces saturation ++

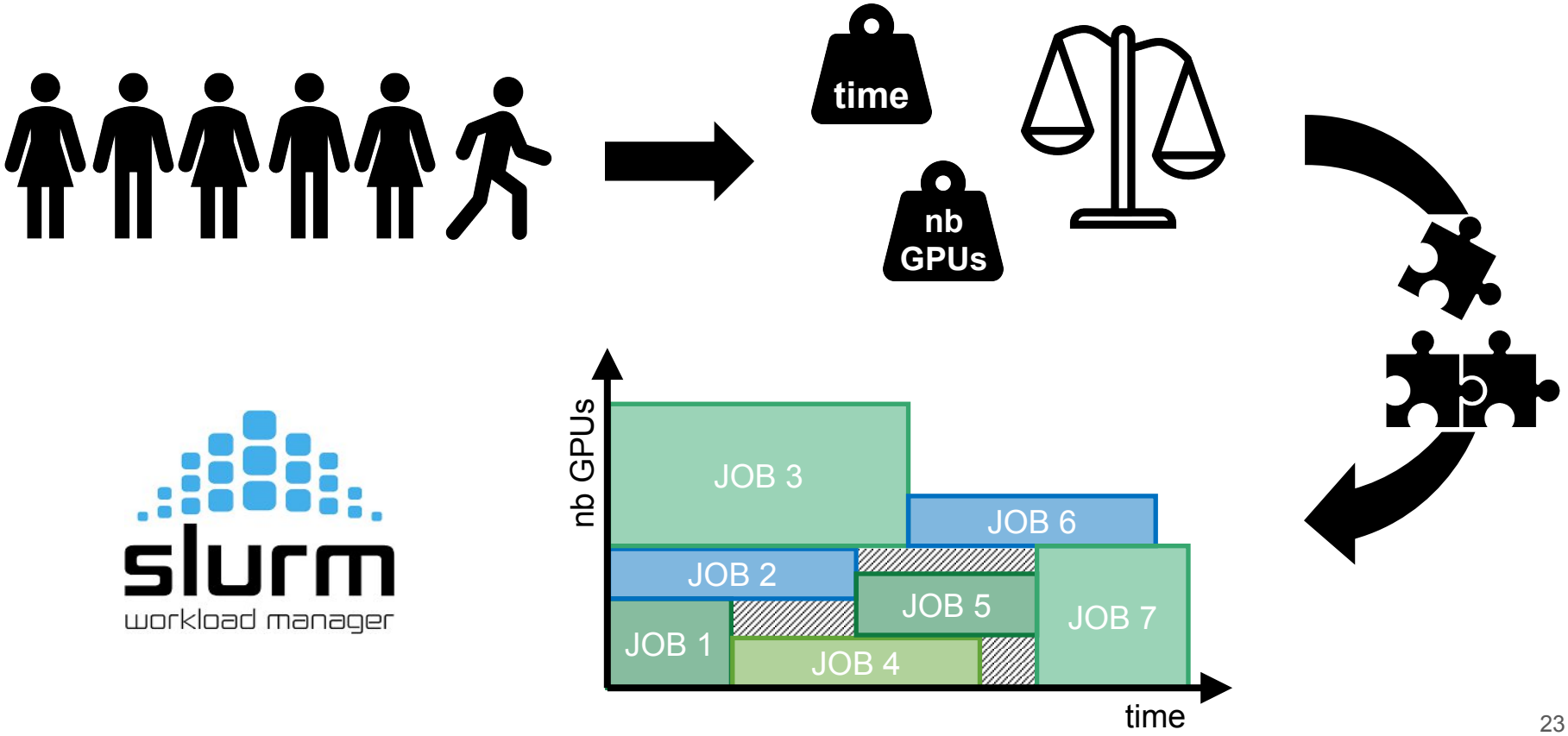
Singularity containers

```
login@jean-zay3:~$ module load singularity
```

Import SIF images on Jean Zay

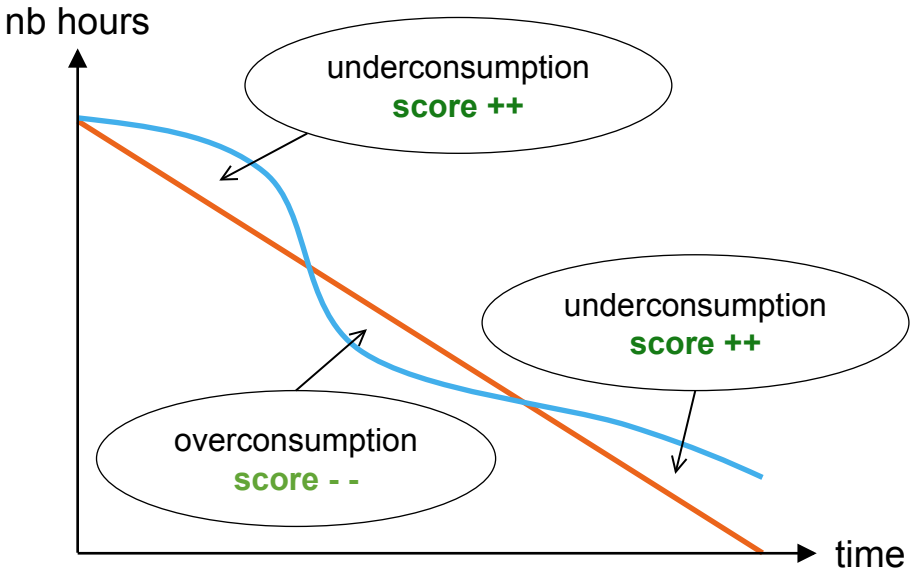
- From your PC
- From public deposits
- Possible conversion from docker

Job submission with Slurm



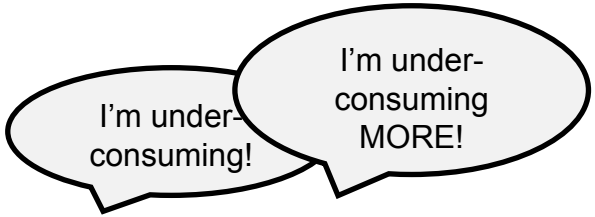
Job submission with Slurm

Slurm gives your job a **priority score** depending on your consumption.



theoretical consumption

real consumption



Slurm compares all users' scores to define job position in the queue.

Job submission with Slurm

Adjust the **QoS (Quality of Service)** to improve the priority score of your job!

QoS	Max elapsed time	Resource limits (V100 partition)			
		Per job	Per user	Per project	Per QoS
qos_gpu-dev	2h	32 GPUs	32 GPUs (10 jobs max at the same time)	32 GPUs	512 GPUs
qos_gpu-t3 (default)	20h	512 GPUs	512 GPUs	512 GPUs	
qos_gpu-t4	100h	16 GPUs	96 GPUs	96 GPUs	256 GPUs

\+ priority -

dev

prod

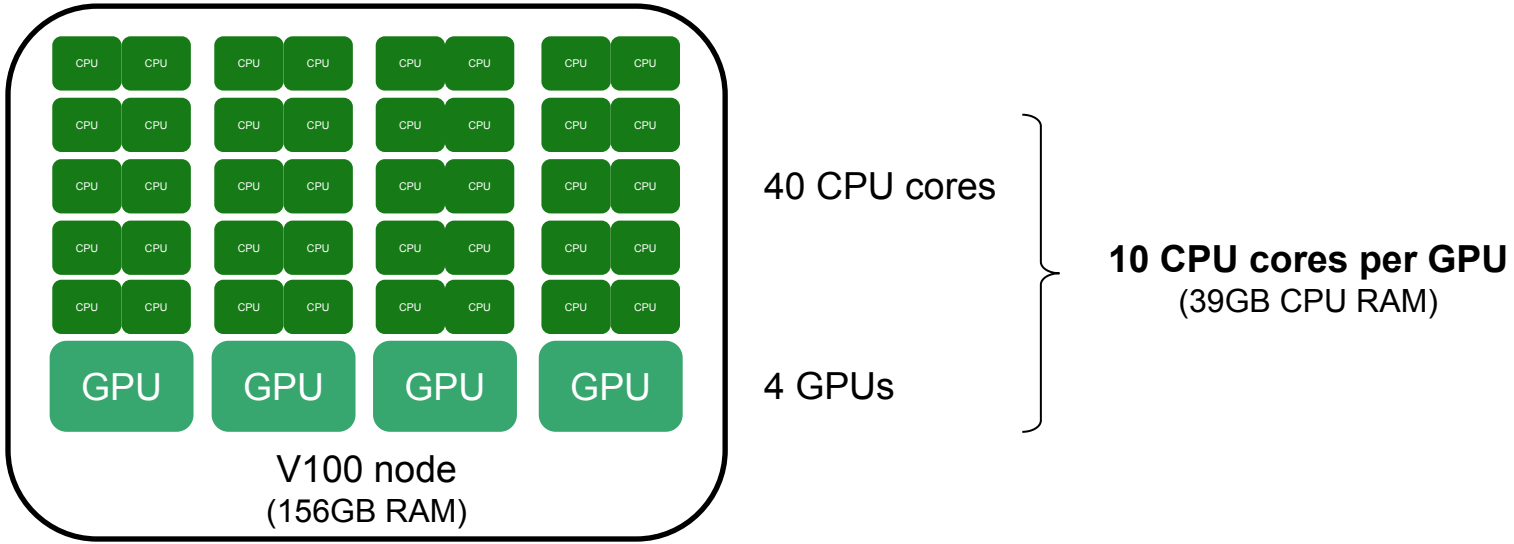
And keep your job in the queue, its priority score will increase with time.

Job submission with Slurm

How to configure my job?

- How many GPUs?
- How much time?
- How many CPUs per GPU?

→ **1 CPU core = 3.9GB RAM** (on V100 nodes)
→ interesting to have **several CPU cores to feed a GPU**
(cf “DataLoader optimizations” part of this course)

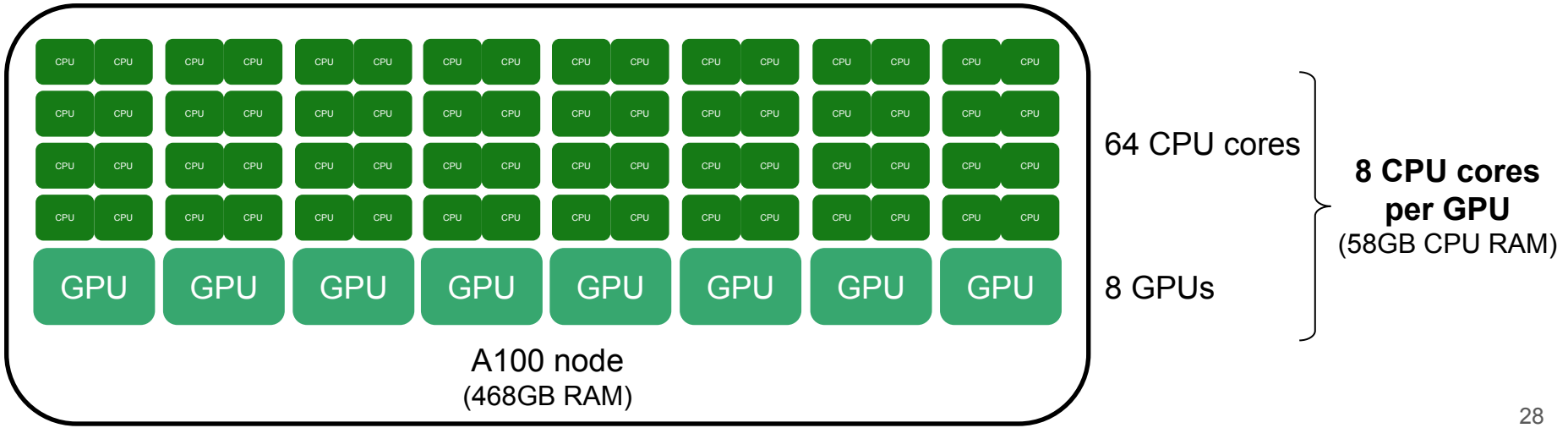


Job submission with Slurm

How to configure my job?

- How many GPUs?
- How many time?
- How many CPUs per GPU?

→ **1 CPU core = 7.3GB RAM** (on A100 nodes)
→ interesting to have **several CPU cores to feed a GPU**
(cf “DataLoader optimizations” part of this course)

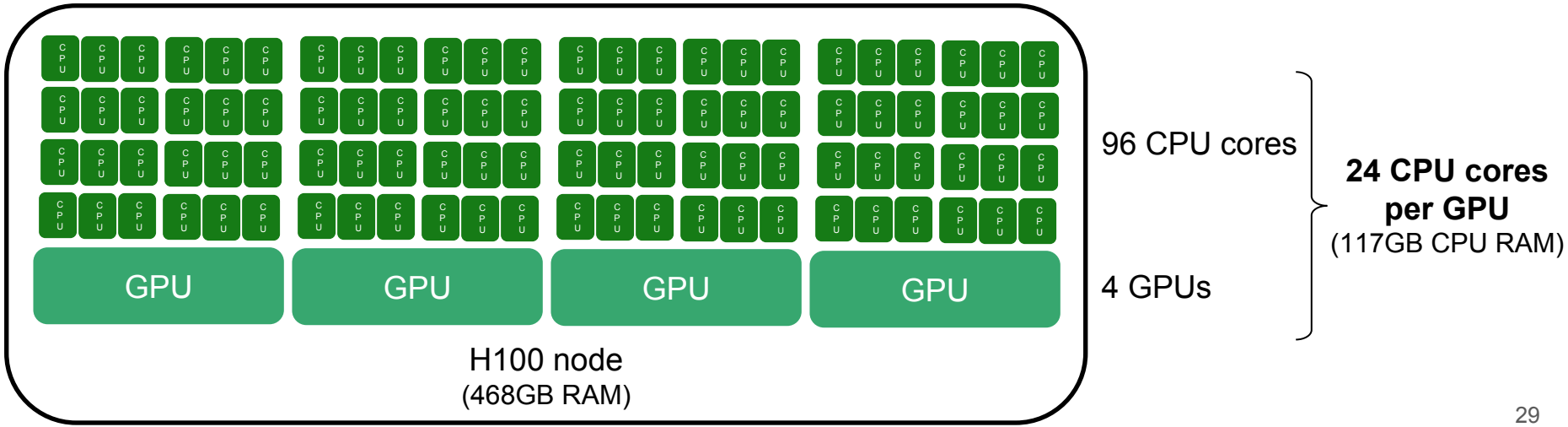


Job submission with Slurm

How to configure my job?

- How many GPUs?
- How many time?
- How many CPUs per GPU?

→ **1 CPU core = 4.87GB RAM** (on H100 nodes)
→ interesting to have **several CPU cores to feed a GPU**
(cf “DataLoader optimizations” part of this course)



Job submission with Slurm

Example: reservation of 2 x 4 V100 GPUs

script.slurm

```
#!/bin/bash

#SBATCH --job-name="dlojz"                # name of the job
#SBATCH --output="log%j.out"              # output file
#SBATCH --error="log%j.err"               # error file
#SBATCH --nodes=2                         # nb of nodes
#SBATCH --gres=gpu:4                      # nb of GPUs/node
#SBATCH --ntasks-per-node=4               # nb of tasks/node
#SBATCH --cpus-per-task=10                # nb of CPU cores/task
#SBATCH --hint=nomultithread              # no hyperthreading
#SBATCH --time=01:00:00                   # max execution time
#SBATCH --qos=qos_gpu-dev                 # adjust QoS

module load pytorch-gpu/py3/2.11.0        # environment

srun python script.py                     # run script
```

Job submission with Slurm

script.slurm

```
#!/bin/bash

#SBATCH --job-name="dlojz"
#SBATCH --output="log%j.out"
#SBATCH --error="log%j.err"
#SBATCH --nodes=2
#SBATCH --gres=gpu:4
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-task=10
#SBATCH --hint=nomultithread
#SBATCH --time=01:00:00
#SBATCH --qos=qos_gpu-dev

module load pytorch-gpu/py3/2.11.0

srun python script.py
```

```
login@jean-zay3:~$ sbatch script.slurm
```

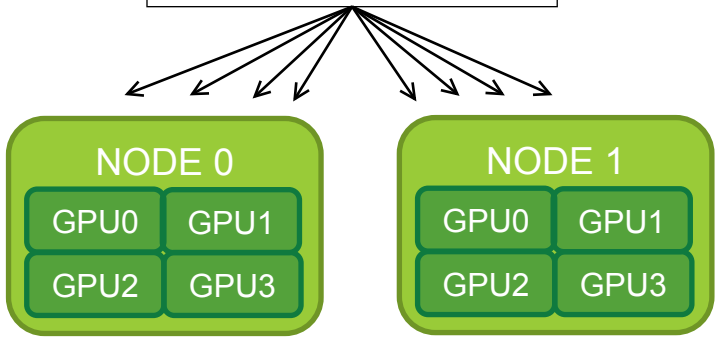
Job submission

Waiting in queue

```
login@jean-zay3:~$ squeue --me
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            223225   gpu_p13    dlojz          PD       0:00      2 (Priority)
```

Lauching job

srun python script.py

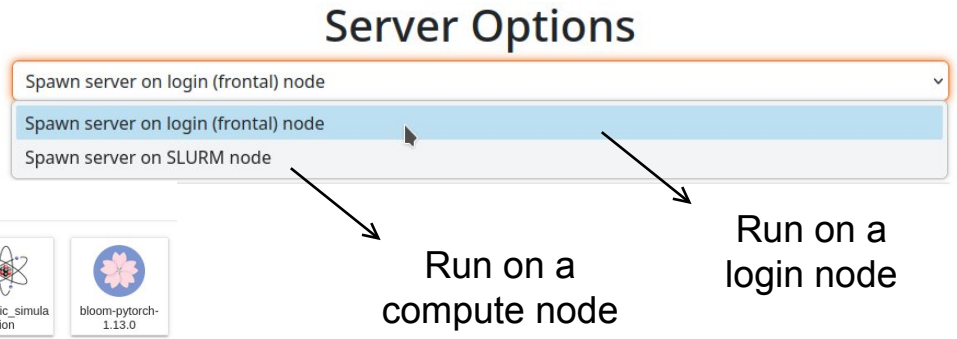


JupyterHub on Jean Zay

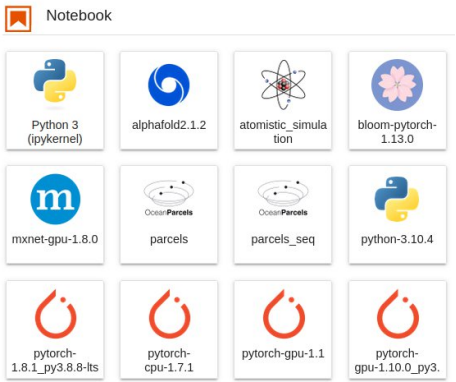
1. Authentication on https://jupyterhub.idris.fr



2. Choose and configure an instance



3. Choose a kernel (pytorch-gpu-2.11.0)

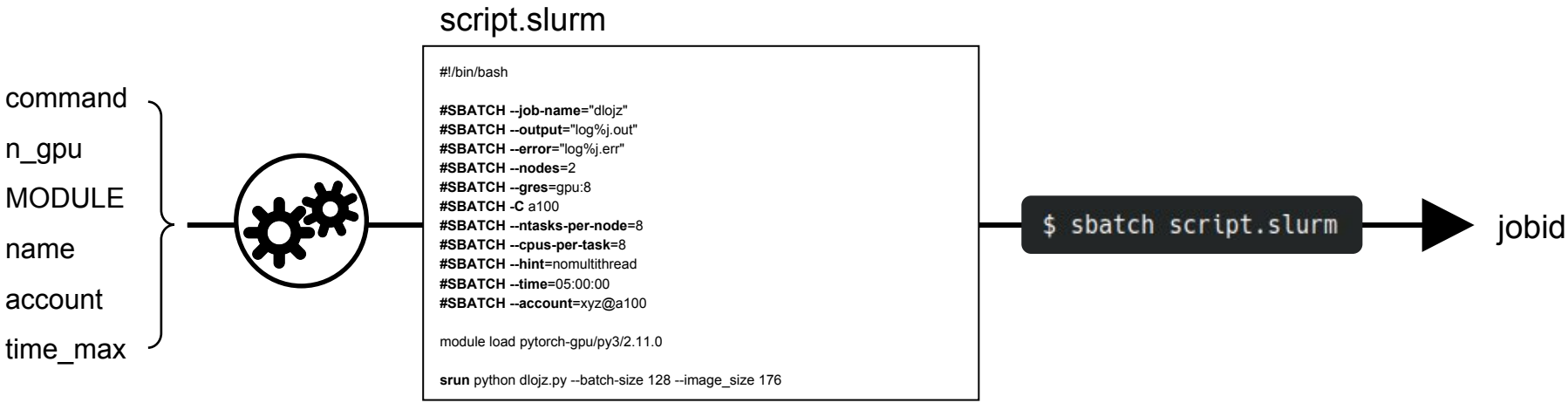


Slurm tools for python notebooks

```
from idr_pytools import gpu_jobs_submitter
```

```
command = 'dlojz.py --batch-size 128 --image_size 176'
n_gpu = 8
MODULE = 'pytorch-gpu/py3/2.11.0'
name = 'dlojz'

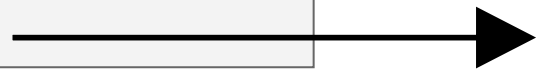
jobid = gpu_jobs_submitter(command, n_gpu, MODULE, name=name, account='xyz@a100', time_max='05:00:00')
```



Slurm tools for python notebooks

```
from idr_pytools import display_slurm_queue
```

```
name = 'dlojz'
display_slurm_queue(name)
```



```
$ squeue --me -n <name>
```

```
from idr_pytools import search_log
```

```
jobid = ['12345']

search_log(contains=jobid)[0]

search_log(contains=jobid, with_err=True)[0]
```



output filename



error filename