IDRIS / Jean Zay

IDRIS / Jean Zay

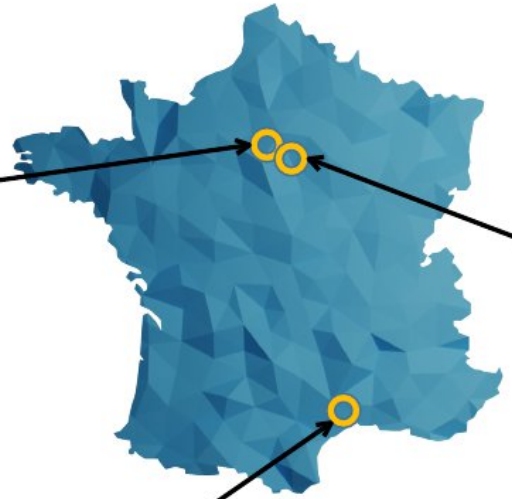
- **IDRIS computing center**
- Jean Zay supercomputer
- Work environment on Jean Zay
- Scaling up with Jean Zay

IDRIS: National computing center of the CNRS

GENCI

Ressources gratuites
- financement français
- recherche ouverte

Jean Zay Supercomputer
HPE SGI 8600,
36.8 PFlop/s
2019
IDRIS CNRS



Joliot-Curie Supercomputer
Bull Sequana X1000 and XH2000,
22 PFlop/s
2017
TGCC CEA

Adastra Supercomputer
HPE Cray EX4000,
74.5 PFlop/s
2022
CINES Universities

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATION
Liberté
Égalité
Fraternité

cnrs

Access to resources

Nombre d'heures demandées

+

500 000 heures CPU
ou
50 000 heures GPU normalisées*

-

* i.e. 50 000 h V100 ou 25 000 h A100 ou 12 500 h H100

Accès régulier

Accès dynamique

Hours request + account opening

https://www.edari.fr

GENCI — CINES — CNRS (Institut du Développement et des Ressources en Informatique Scientifique) — TGCC

DARI Demande d'Attribution de Ressources Informatiques — Se connecter ou se créer un compte eDARI

Comment obtenir des ressources de calcul et de stockage ?

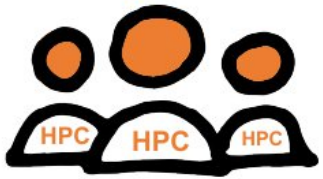
Demande de ressources ou demande d'accès	Pour qui, quel usage et quelles conditions ?	Documents de référence	Actualités
...	...	...	...

Allocation d'heures

Compte de calcul IDRIS

cnrs

Les missions de l'équipe support

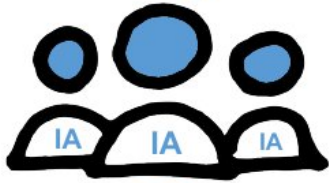


Permanence
(emails et téléphone)

Installations
logicielles

Documentation
en ligne

Support avancé



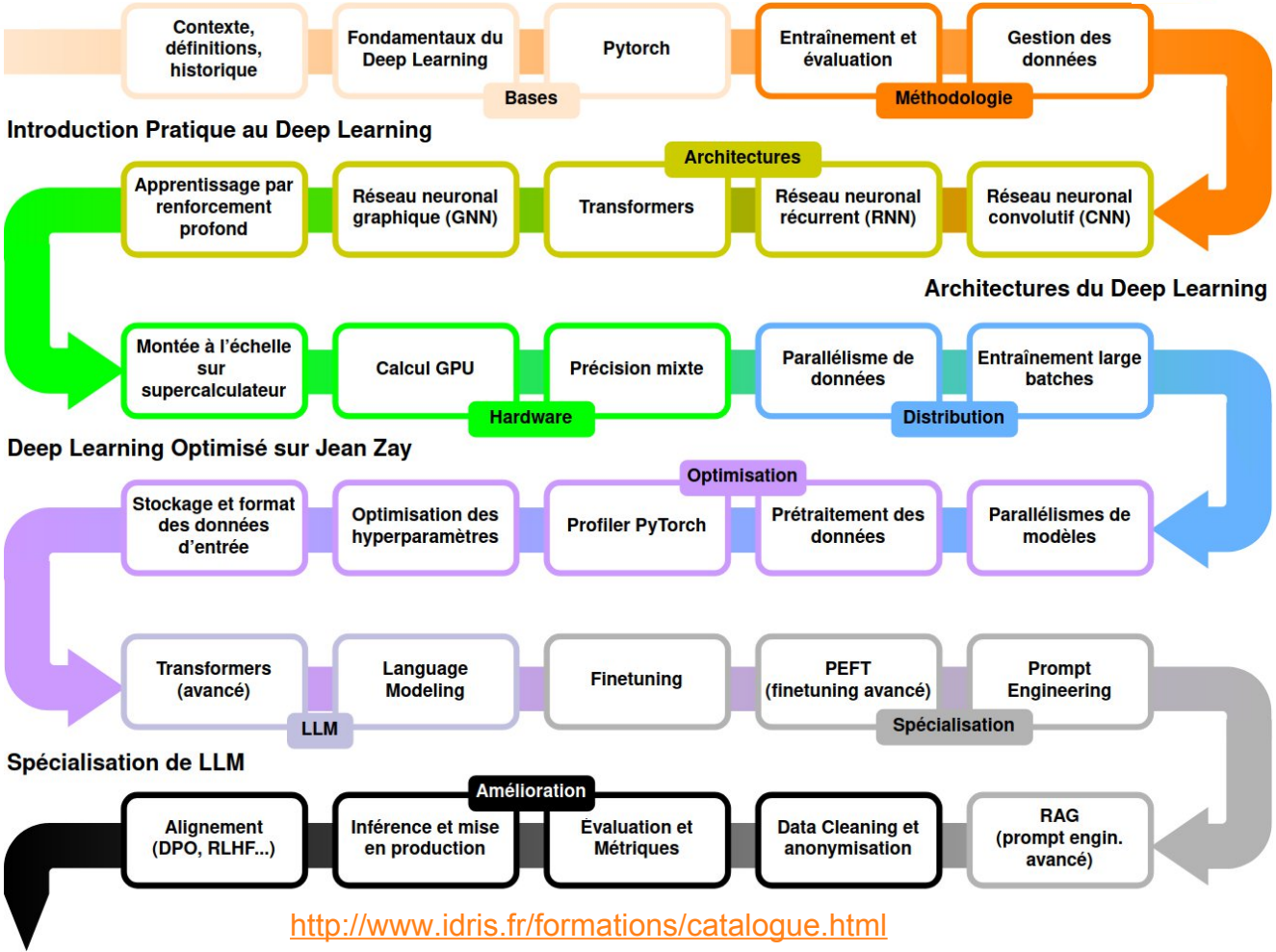
Hackathon
international

Workshop

Formations

Veille
technologique

AI courses - 2024



http://www.idris.fr/formations/catalogue.html

We are on YouTube

- **FIDLE ANF:** www.youtube.com/@CNRS-FIDLE
  - Deep Learning course online
  - 20 sessions (2h each)
  - On Thursday at 2PM

- **IDRIS YouTube channel:** www.youtube.com/@idriscnrs
  - Video documentation
  - Monthly magazine - technology watch (1h)
    → *Friday, August 23rd at 10AM*

L'IDRIS recrute des ingénieur·e·s sur concours externes

- **Concours N°65 : Expert·e calcul scientifique, fonction mutualisée Maison de la Simulation (IR)**
  - équipe Support HPC de l'IDRIS + Maison de la Simulation (60% / 40%)
- **Concours N°67 : Expert·e calcul scientifique (IR)**
  - équipe Support IA de l'IDRIS
- **Concours N°221 : Gestionnaire d'infrastructures (AI)**
  - équipe Réseau de l'IDRIS
- **Concours N°235 : Assistant·e en gestion administrative (AI)**
  - équipe Administration de l'IDRIS, administration + communication

Deadline de soumission des dossiers : 10 juillet 2024 13h

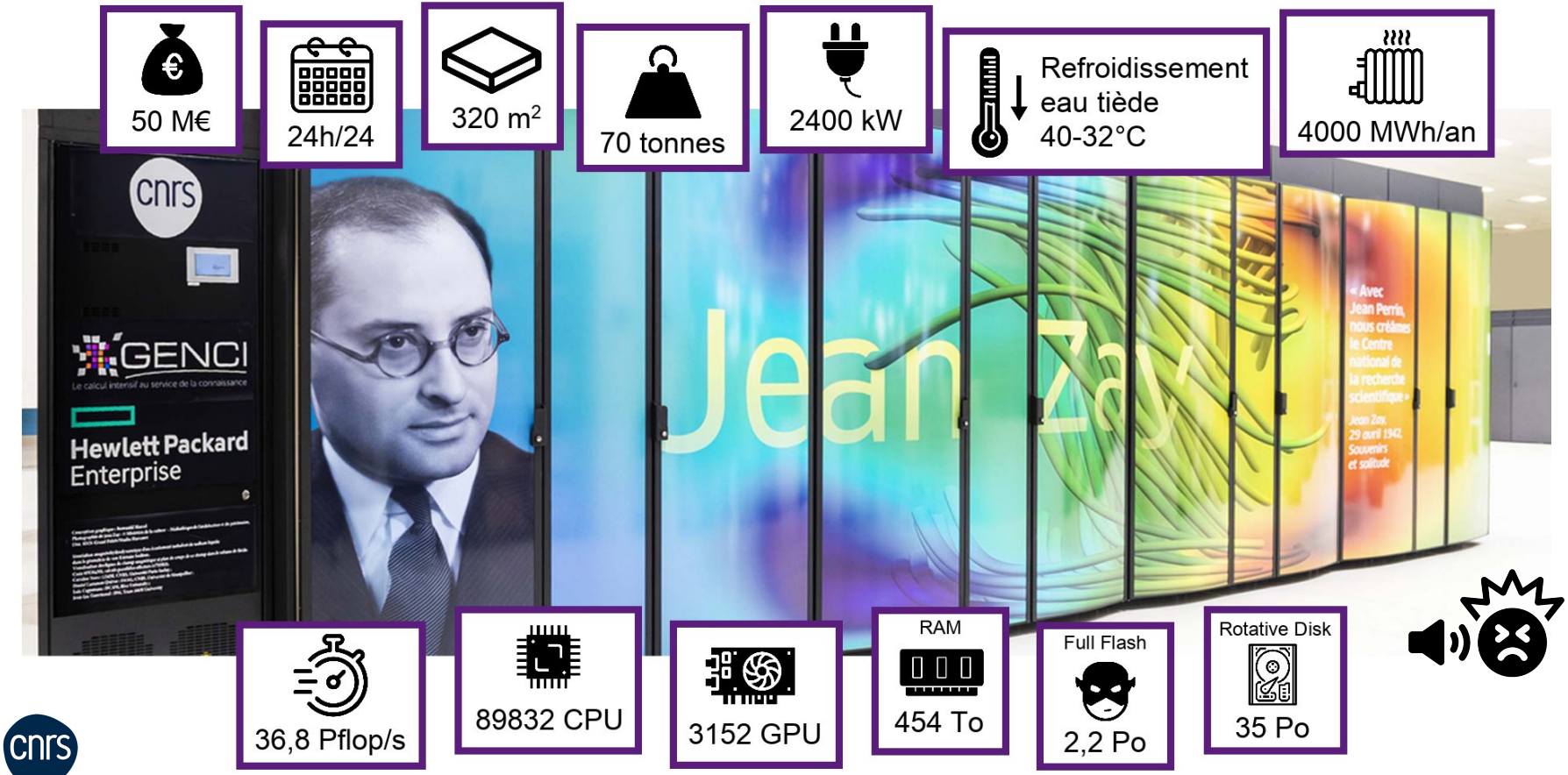
http://www.idris.fr/annonces/idris-recrute.html

IDRIS / Jean Zay

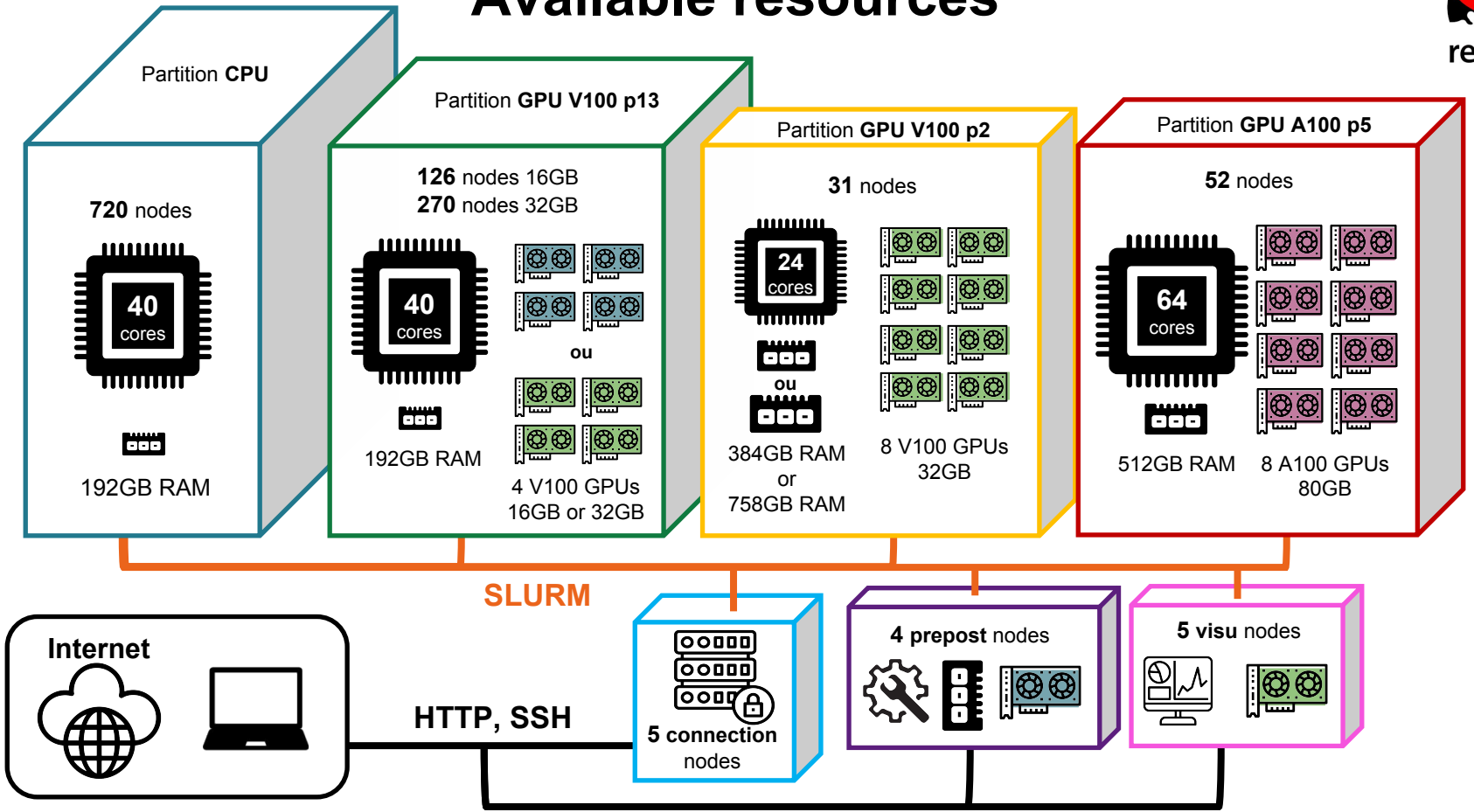
- IDRIS computing center
- ➢ **Jean Zay supercomputer**
- Work environment on Jean Zay
- Scaling up with Jean Zay

Jean Zay

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



Available resources



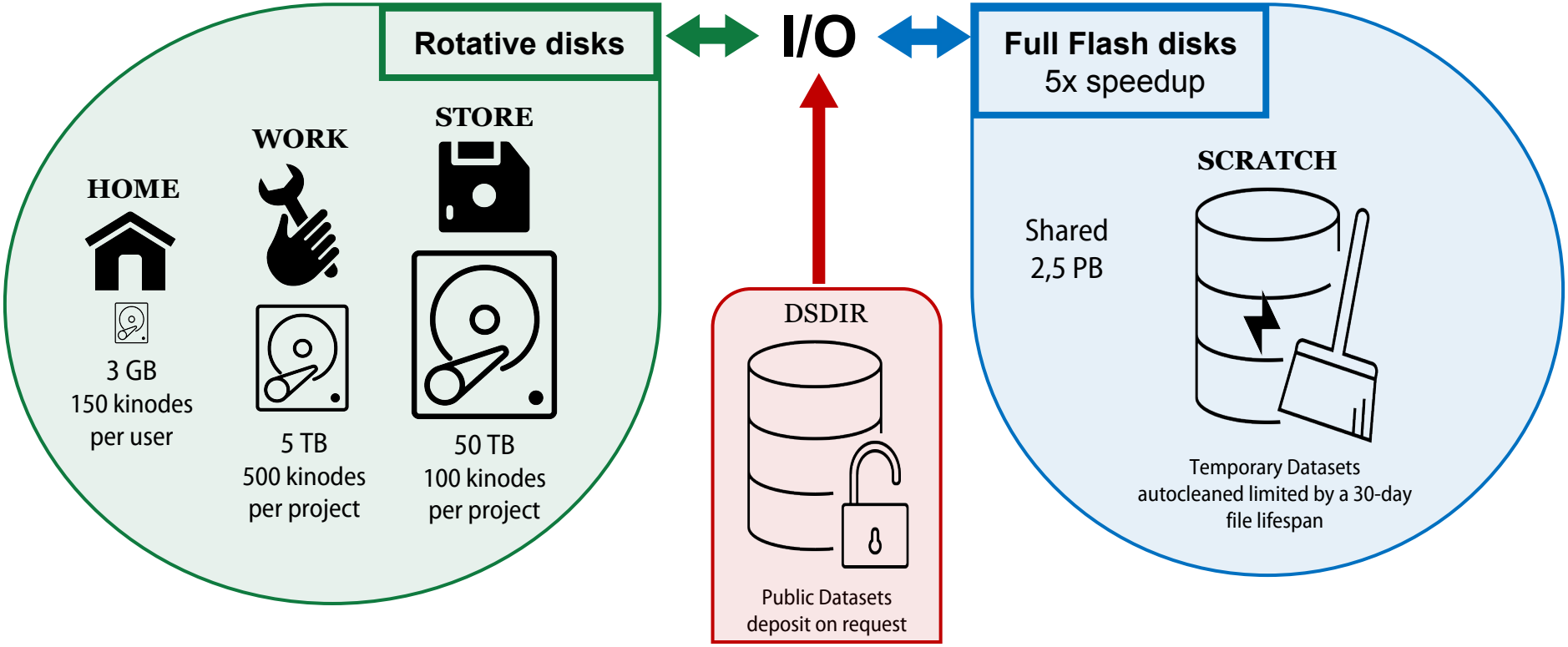
Available resources

Available to all users on September 2024

Soon on Jean Zay...

- Resources: **+ 1456 NVIDIA H100 80GB GPUs**
- New interconnection network: OmniPath **+ InfiniBand**
- New parallel file system: from IBM Spectrum Scale to **Lustre**

Storage spaces

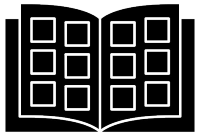


IDRIS / Jean Zay

- IDRIS computing center
- Jean Zay supercomputer
- ➢ **Work environment on Jean Zay**
- Scaling up with Jean Zay

Work environment

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 ++

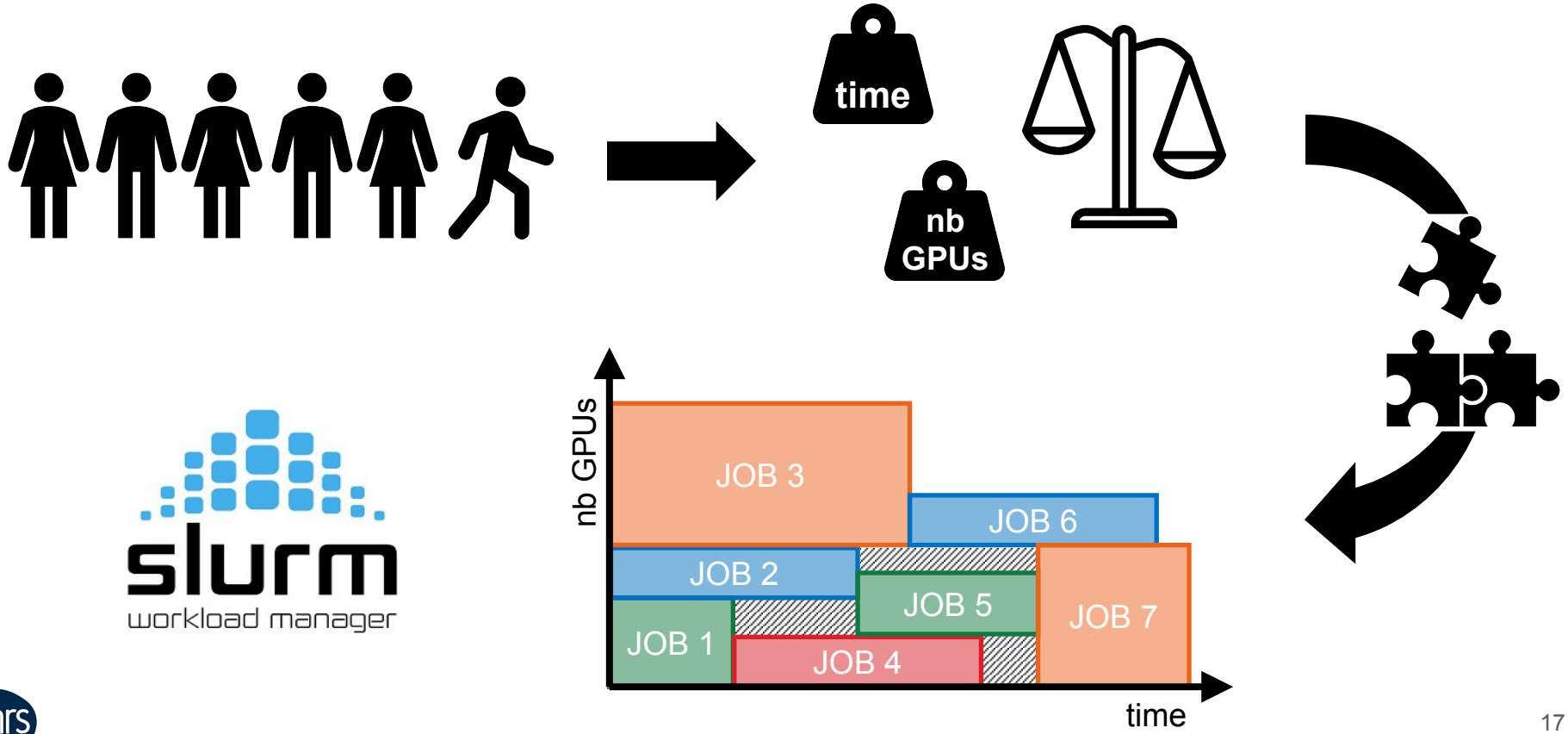
Singularity containers

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

Import SIF images on Jean Zay

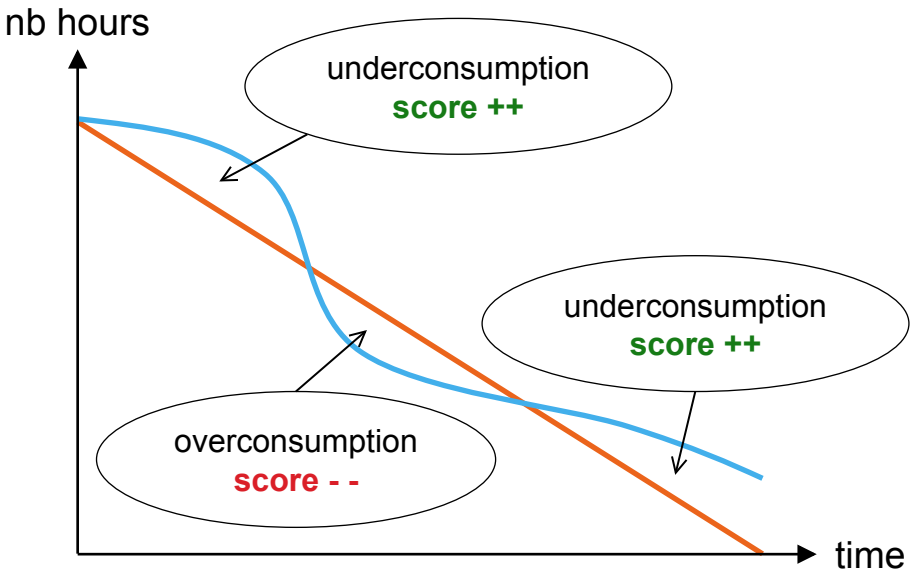
- From your PC
- From public deposits
- Possible conversion from docker

Job submission



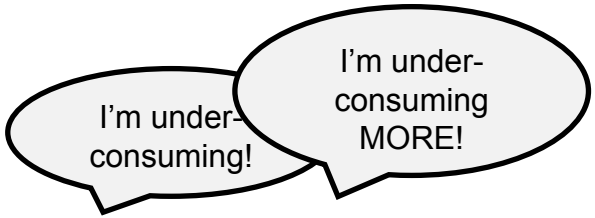
Job submission

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



theoretical consumption

real consumption



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

Job submission

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

QoS	Max elapsed time	Resource limits			
		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 (V100)	100h	16 GPUs	96 GPUs	96 GPUs	256 GPUs

\+ priority -

dev

prod

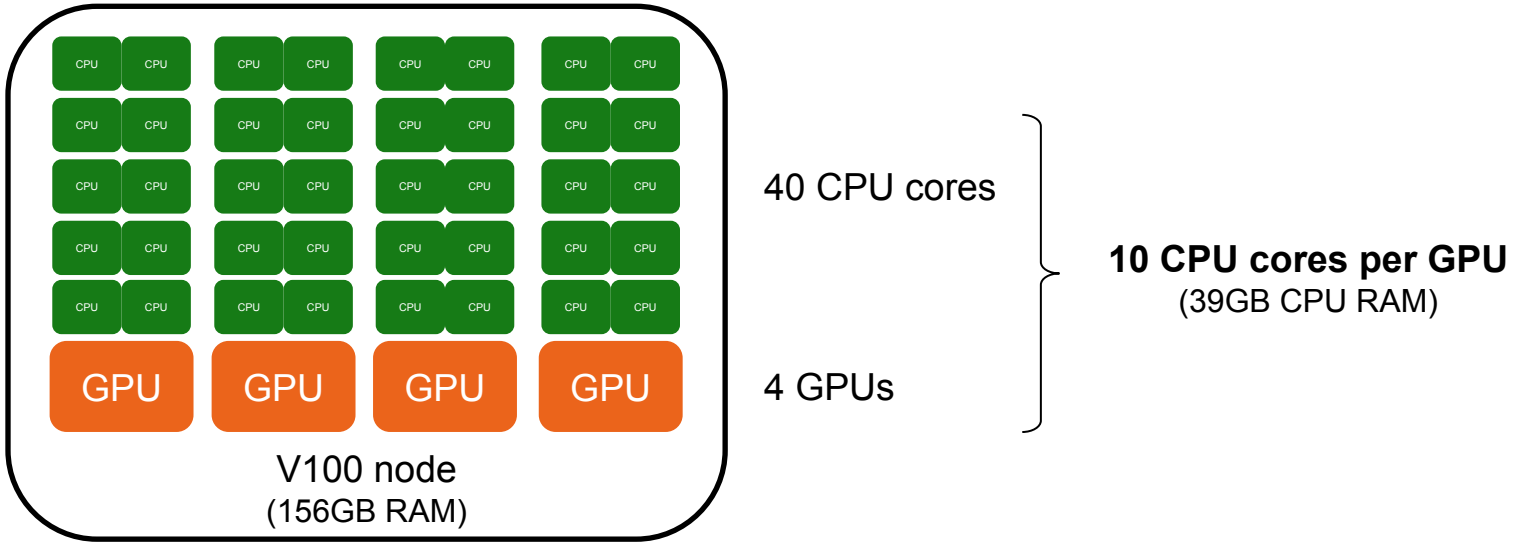
And keep your job in the queue, its priority score will increase with time.

Job submission

How to configure my job?

- How many GPUs?
- How many 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)

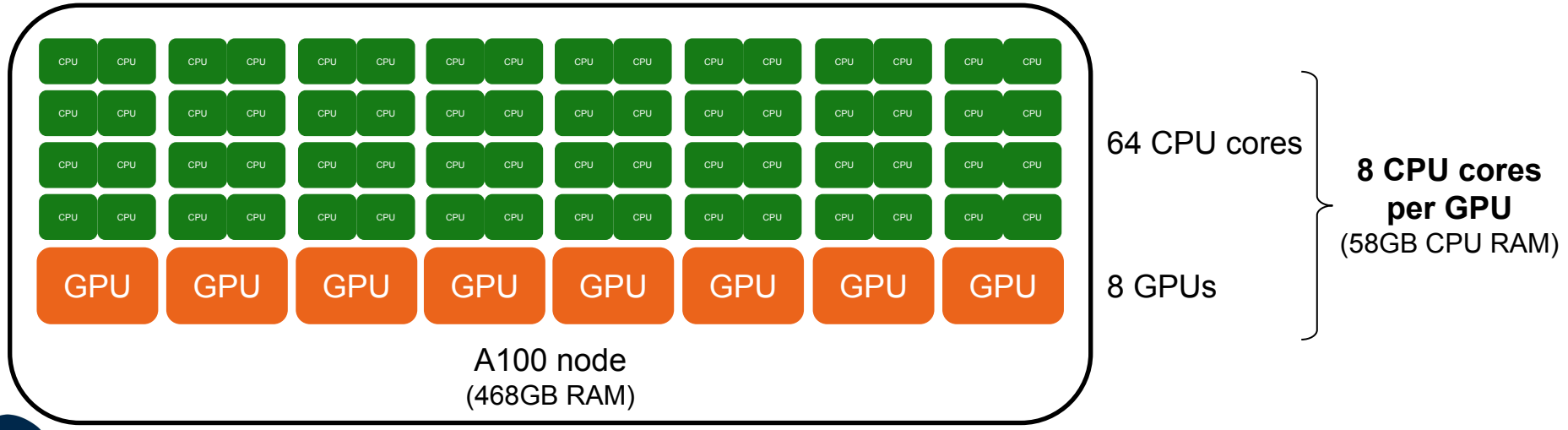


Job submission

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

Example: reservation of 2 x 4 V100 GPUs

script.slurm

```
#!/bin/bash

#SBATCH --job-name="dlojz"                # name of the job
#SBATCH --output="dlojz%j.out"            # output file
#SBATCH --error="dlojz%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/py3/2.2.0         # environment

srun python script.py                     # run script
```

Job submission

script.slurm

```
#!/bin/bash

#SBATCH --job-name="dlojz"
#SBATCH --output="dlojz%j.out"
#SBATCH --error="dlojz%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/py3/2.2.0

srun python script.py
```

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

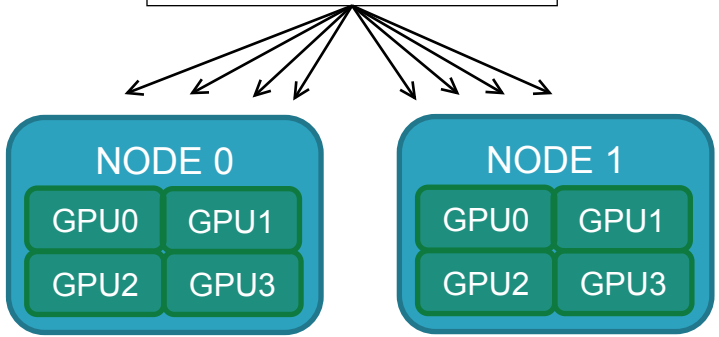
Job submission

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)
```

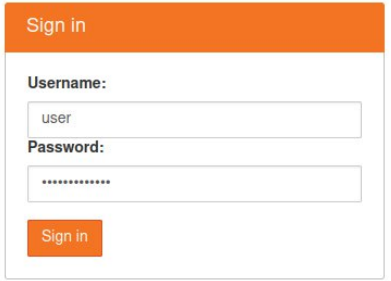
Lauching job

srun python script.py

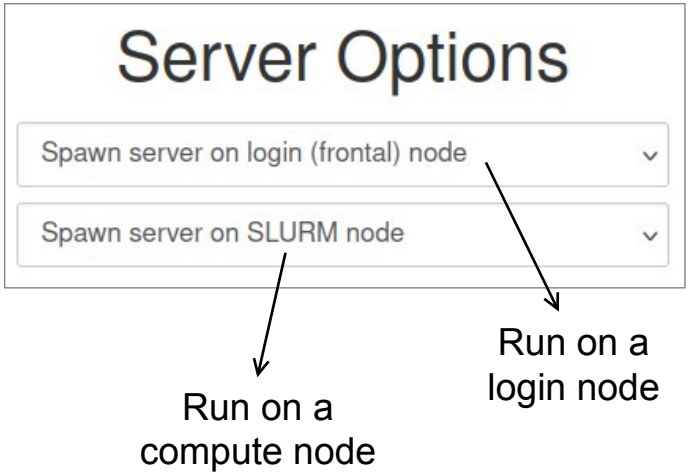


JupyterHub

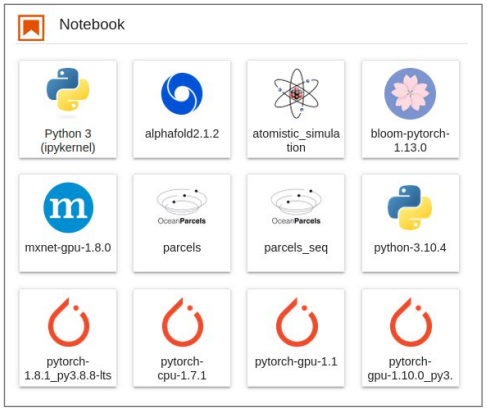
1. Authentication on https://jupyterhub.idris.fr



2. Choose and configure an instance



3. Choose a kernel (pytorch-gpu-2.2.0)



IDRIS / Jean Zay

- IDRIS computing center
- Jean Zay supercomputer
- Work environment on Jean Zay
- ➢ **Scaling up with Jean Zay**

Some use cases

- Submitting a collection of similar jobs - Slurm Job Arrays
- Handle Slurm jobs dependencies
- Distributing training on several GPUs

Submitting a collection of similar jobs

- Slurm Job Arrays

job_array.slurm

```
#!/bin/bash
#SBATCH --job-name=job-array    # name of job
#SBATCH --ntasks=1              # total number of MPI processes
#SBATCH --ntasks-per-node=1     # number of MPI processes per node
#SBATCH --hint=nomultithread    # 1 MPI process per physical core (no hyperthreading)
#SBATCH --time=01:00:00         # maximum execution time requested (HH:MM:SS)
#SBATCH --output=%x_%A_%a.out   # output file name containing the ID and the index value
#SBATCH --error=%x_%A_%a.out    # name of error file (here common with the output)
#SBATCH --array=0-19%5          # total of 20 jobs but maximum of 5 jobs in the Slurm queue

# load modules
module load pytorch-gpu/py3/2.2.0

# Execution of the "myscript.py" script with different inputs for each job
# The value of ${SLURM_ARRAY_TASK_ID} is different for each job
srun python myscript.py --config config_${SLURM_ARRAY_TASK_ID}.json \
                        --checkpoint checkpoint_${SLURM_ARRAY_TASK_ID}.safetensors
```


IDRIS doc: http://www.idris.fr/eng/jean-zay/cpu/jean-zay-cpu-exec_jobarray-eng.html

Submitting a collection of similar jobs

- Slurm Job Arrays

```
$ sbatch job_array.slurm
$ squeue --me
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
305813_[5-19%5]     cpu_p1 job-array  mylogin PD       0:00      1 (JobArrayTaskLimit)
        305813_4     cpu_p1 job-array  mylogin  R       1:23      1 r7i1n0
        305813_3     cpu_p1 job-array  mylogin  R       1:23      1 r8i6n3
        305813_2     cpu_p1 job-array  mylogin  R       1:23      1 r7i2n0
        305813_1     cpu_p1 job-array  mylogin  R       1:23      1 r8i6n0
        305813_0     cpu_p1 job-array  mylogin  R       1:23      1 r8i5n3
```

IDRIS doc: http://www.idris.fr/eng/jean-zay/cpu/jean-zay-cpu-exec_jobarray-eng.html

Handle Slurm jobs dependencies

- Creating dependencies between Slurm jobs

submit_jobs.sh

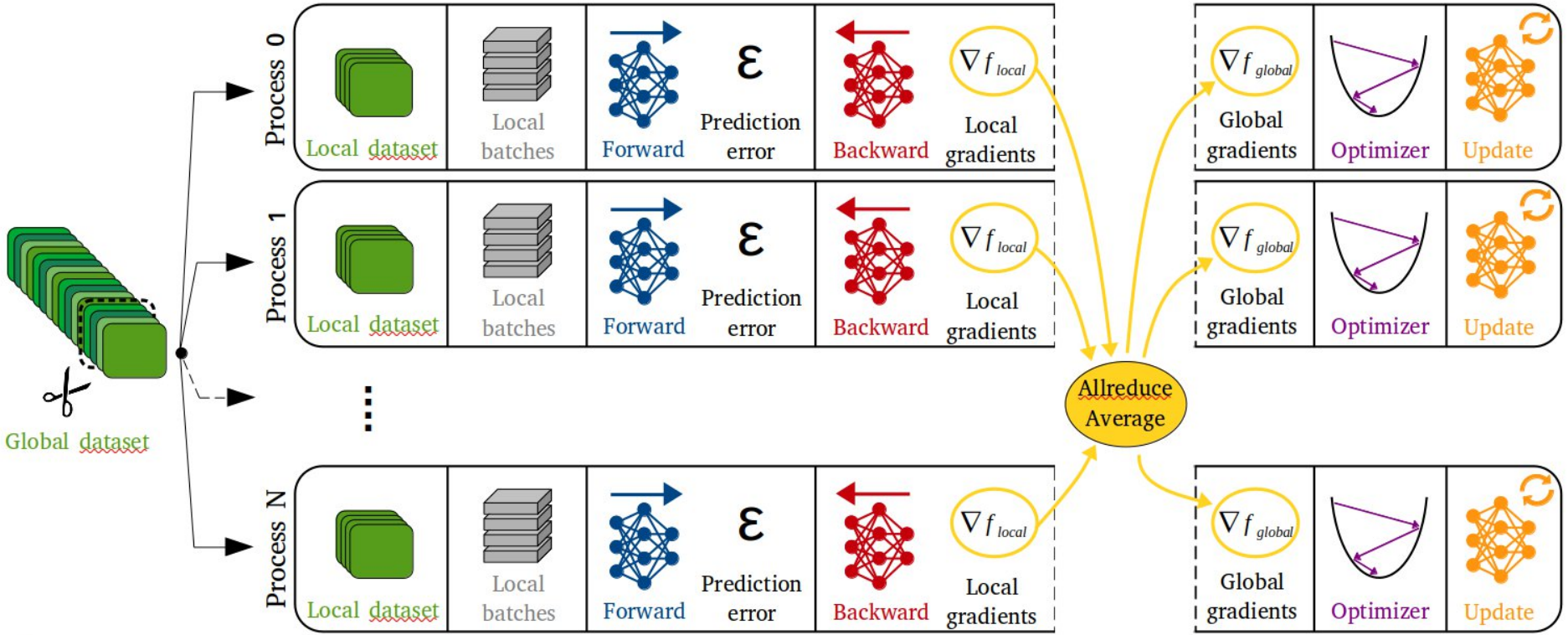
```
#!/bin/bash
# launch first job and get its ID
JOBID1=`sbatch  job1.slurm | cut -d " " -f 4`
# launch second job after the first one if it succeeds
sbatch --dependency=afterok:$JOBID1 job2.slurm
```

```
$ ./submit_jobs.sh
$ squeue --me
Thu Jun 27 15:15:10 2024
             JOBID PARTITION     NAME     USER    STATE       TIME TIME_LIMI  NODES NODELIST(REASON)
            241230    cpu_p1     job2  mylogin  PENDING       0:00     10:00      1 (Dependency)
            241229    cpu_p1     job1  mylogin  RUNNING       0:29     10:00      1 r1i1n16
```


IDRIS doc: http://www.idris.fr/eng/jean-zay/cpu/jean-zay-cpu-exec_cascade-eng.html

Distributing training on several GPUs

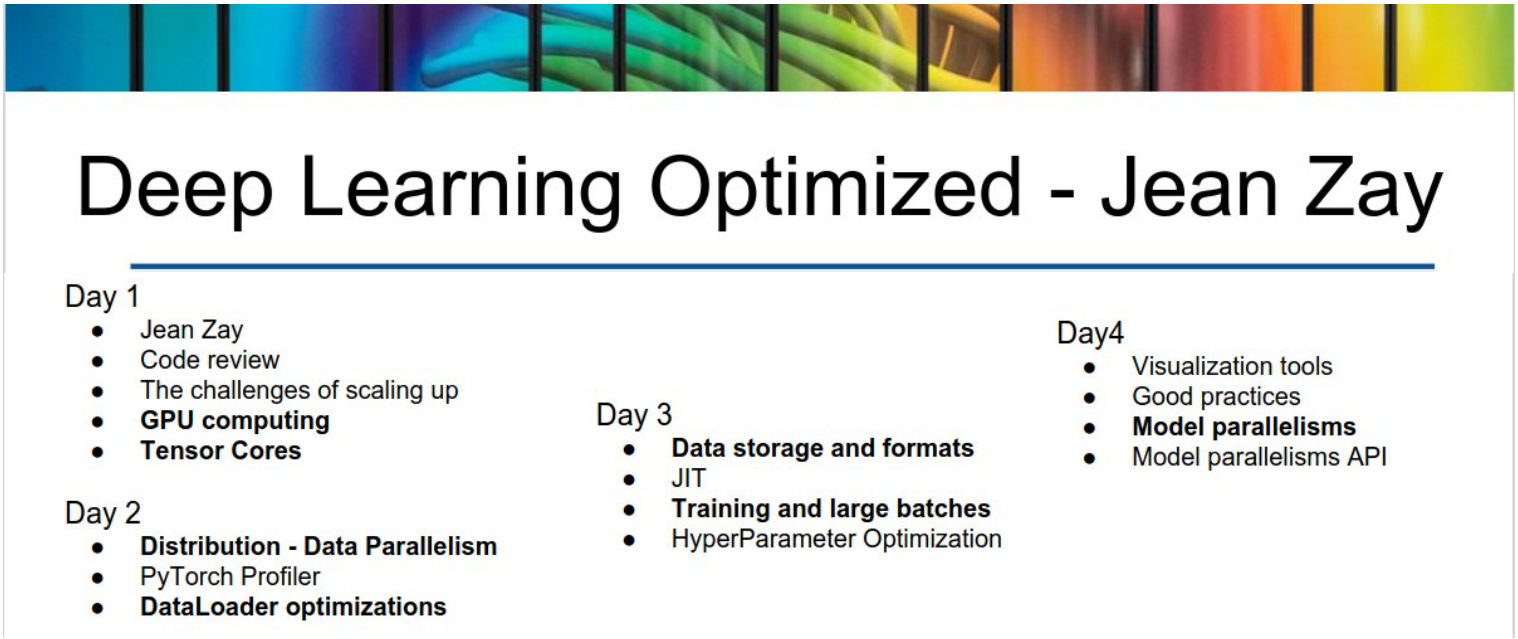
- Data Parallelism



IDRIS doc: http://www.idris.fr/eng/ia/apprentissage-distribue-eng.html

Distributing training on several GPUs

- Data Parallelism
  - PyTorch **DistributedDataParallel**: http://www.idris.fr/eng/jean-zay/gpu/jean-zay-gpu-torch-multi-eng.html
  - TensorFlow **MultiWorkerMirroredStrategy**: http://www.idris.fr/eng/jean-zay/gpu/jean-zay-gpu-tf-multi-eng.html



DLO-JZ course: http://www.idris.fr/formations/dlo-jz/fiche-dlo-jz.html

IDRIS / Jean Zay

- **Thank you for your attention**

Useful links and contacts

Compute hours request:
- eDARI website: www.edari.fr
- GENCI website: www.genci.fr

Account opening: http://www.idris.fr/eng/info/gestion/demandes-heures-eng.html

IDRIS website: www.idris.fr

Jean Zay documentation:
- Main documentation: http://www.idris.fr/eng/jean-zay/
- Getting started: http://www.idris.fr/eng/su/debutant-eng.html
- Artificial Intelligence: http://www.idris.fr/eng/ia/
- Cheat sheet: http://www.idris.fr/jean-zay/pense-bete.html

Contacts:
- Compute jours request: acces@genci.fr
- Account opening: gestutil@idris.fr
- Support team: assist@idris.fr