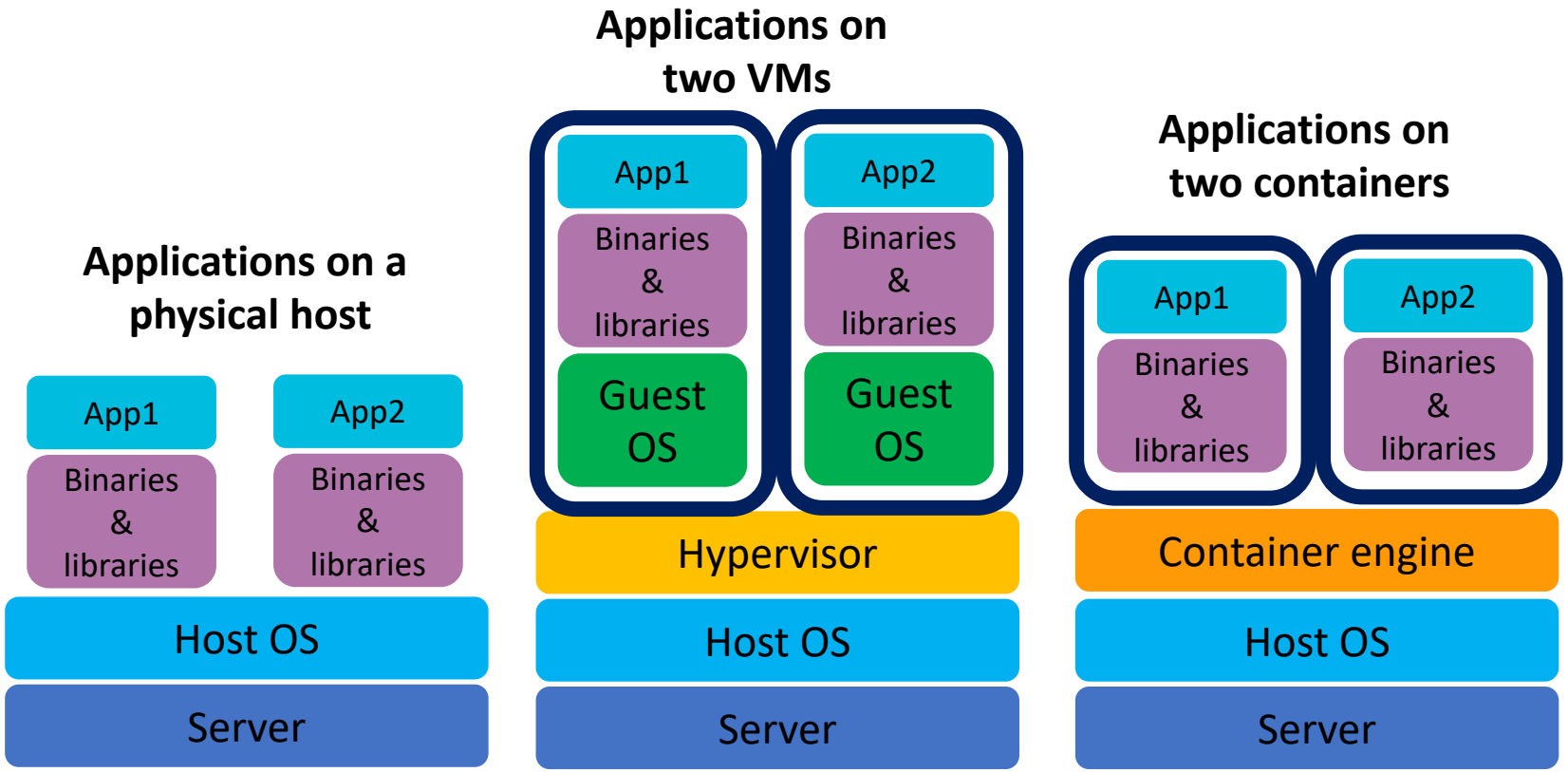
POLITECNICO
MILANO 1863

Containers and schedulers

Physical hosts, virtual machines and containers



Containers

POLITECNICO MILANO 1863

- Software should run properly on multiple environments:
  - A developer's laptop
  - A test environment
  - Potentially, several production environments
- Possible problems:
  - Different versions of libraries and middleware
  - Different network and security configurations
  - …
- A container consists of an entire runtime environment: an application, plus all its dependencies, libraries and other binaries, and configuration files needed to run it, bundled into one package

Containers vs VMs

Containers

- Rely on the underlying operating system
- A container image is in the order of MBs
- Are available almost instantly

VMs

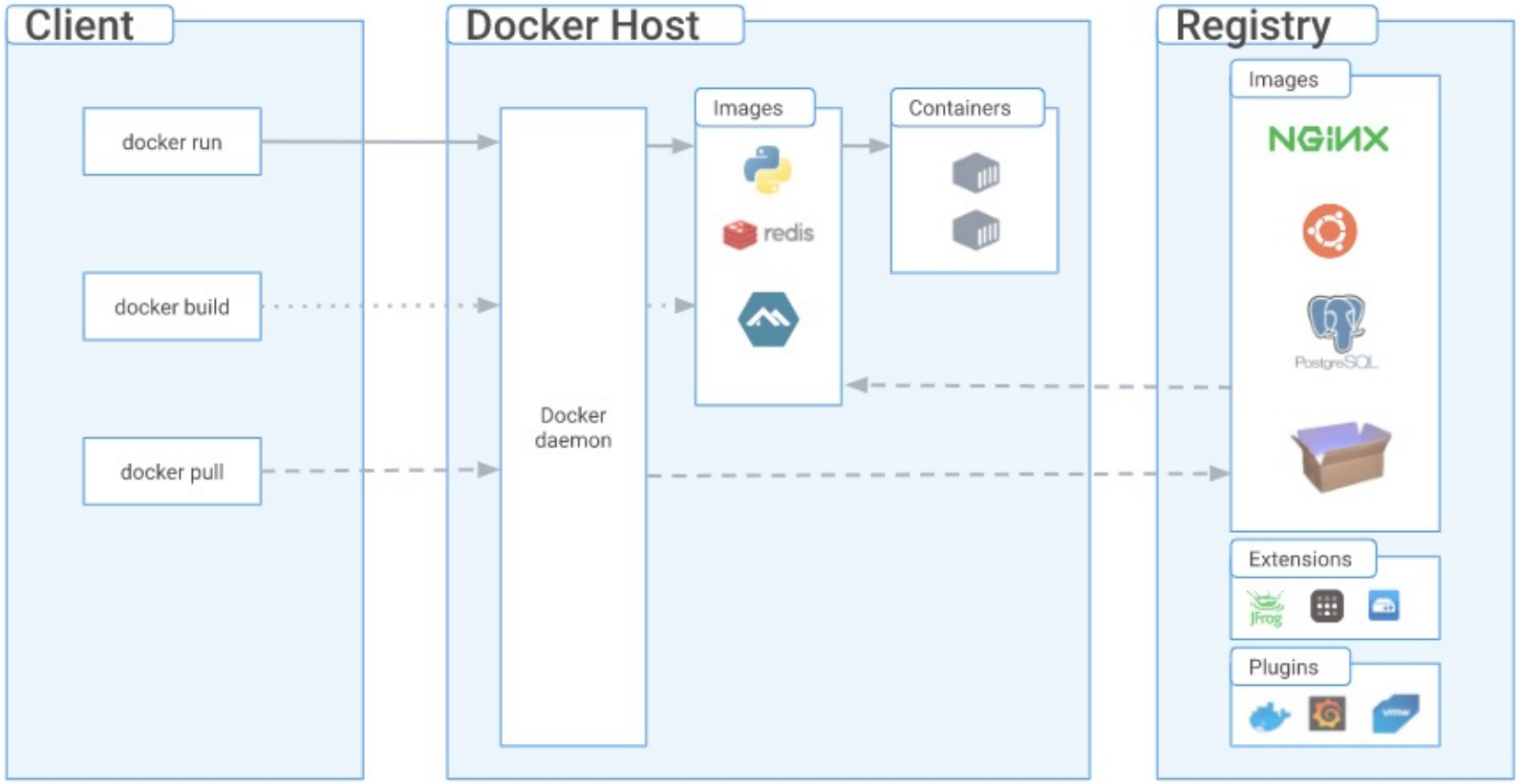
- Each has its own operating system
- A VM image is in the order of GBs
- Require several minutes to boostrap

Container frameworks

- The first container technologies from the Unix/Linux domain -> LXC https://linuxcontainers.org/
- Docker
- Singularity
- Open Container Initiative (OCI) https://opencontainers.org/

Docker

https://www.docker.com/



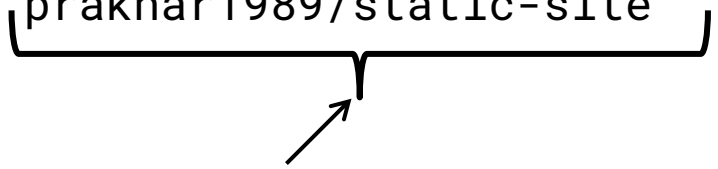
Terminology

- Image
  - Read-only template with instructions for creating a Docker container
  - An image can be based on another image and add some customization to it
- Container
  - A runnable instance of an image
  - Containers can be created, started, stopped, moved or deleted
  - They can be connected to one or more networks and attach storage to them
- Dockerfile
  - Contains the instructions to create and run a new image

Quick demo

https://docker-curriculum.com/

- Running a container
  - `docker run prakhar1989/static-site`



- If we want to publish the exposed ports
  - `docker run -P prakhar1989/static-site`
- If we want to map an exposed port on a specific port
  - `docker run -p 8888:80 prakhar1989/static-site`
- If we want to reuse the terminal where docker is running and give a name to the container
  - `docker run -d --name static-site -p 8888:80 prakhar1989/static-site`

Dockerfile

POLITECNICO MILANO 1863

- Defines what goes in the environment inside a container.
- It permits to:
  - Define starting image
  - Set up environment (download extra packages, ..)
  - Configure host (map ports..)
  - Execute commands
  - Copy files inside the container

Example of a Dockerfile

https://docker-curriculum.com/

```
FROM python:3.8

# set a directory for the app
WORKDIR /usr/src/app

# copy all the files to the container
COPY . .

# install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# define the port number the container should expose
EXPOSE 5000

# run the command
CMD ["python", "./app.py"]
```

- Dockerfile, requirements.txt and app.py should be in the same folder. You can clone the repo
  - https://github.com/prakhar1989/docker-curriculum.git
  - Edit requirements.txt replacing the existing line with `Flask==3.0.3`
- To create the image
  - `docker build –t your_usename/name_you_like .`
- Then
  - `docker run –p 8888:5000 your_usename/name_you_like`

Singularity

https://docs.sylabs.io/guides/latest/user-guide/index.html

POLITECNICO MILANO 1863

- Created to run complex applications on HPC clusters in a simple, portable, and reproducible way
- Main focuses:
  - Verifiable reproducibility and security, using cryptographic signatures, an immutable container image format, and in-memory decryption.
  - Integration over isolation by default. Easily make use of GPUs, high speed networks, parallel filesystems on a cluster or server by default.
  - Mobility of compute. The single file SIF container format is easy to transport and share.
  - A simple, effective security model. You are the same user inside a container as outside, and cannot gain additional privilege on the host system by default.
- Works with any version and distribution of Linux, it does not work with MacOs or Windows
- Can run Docker images

HPC resource managers and job schedulers

POLITECNICO MILANO 1863

- Resource manager: coordinates the actions of all other components in the batch system by maintaining a database of all resources, submitted requests and running jobs.
- Job scheduler: takes the node and job information from the resource manager and produces a list sorted by the job priority telling the resource manager when and where to run each job.

Slurm

POLITECNICO MILANO 1863

- Simple Linux Utility for Resource Management (Slurm): an open-source cluster management and job scheduling system
- Used by most of the top 500 HPC clusters
- Large community, multiple contributors
  - Repo: https://github.com/SchedMD/slurm
  - Documentation: https://slurm.schedmd.com/documentation.html
  - Overview: Jette, M.A., Wickberg, T. (2023). Architecture of the Slurm Workload Manager. In: Klusáček, D., Corbalán, J., Rodrigo, G.P. (eds) Job Scheduling Strategies for Parallel Processing. JSSPP 2023. Lecture Notes in Computer Science, vol 14283. Springer, Cham. https://doi.org/10.1007/978-3-031-43943-8_1

Slurm main characteristics

POLITECNICO MILANO 1863

- Originally designed as a simple resource manager capable only of allocating whole nodes to jobs,
- Evolved into a comprehensive workload scheduler capable of managing the most demanding workflows on many of the largest computers in the world.
- Design goals:
  - Being open source, portability, scalability, fault tolerance, security, Sys Admin friendly

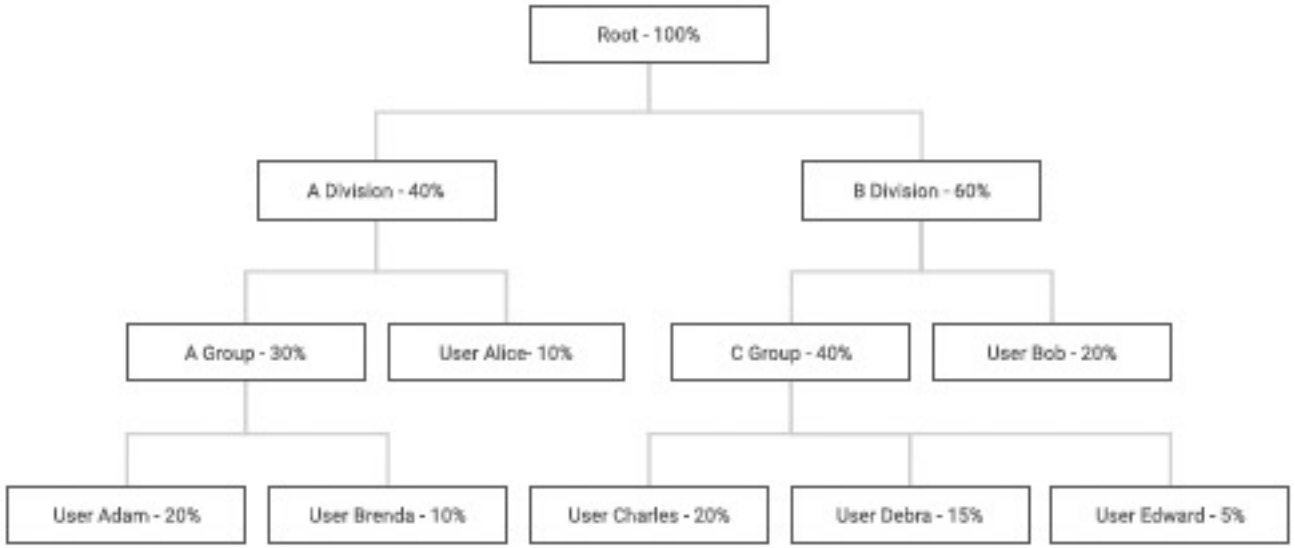
Slurm main concepts

POLITECNICO MILANO 1863

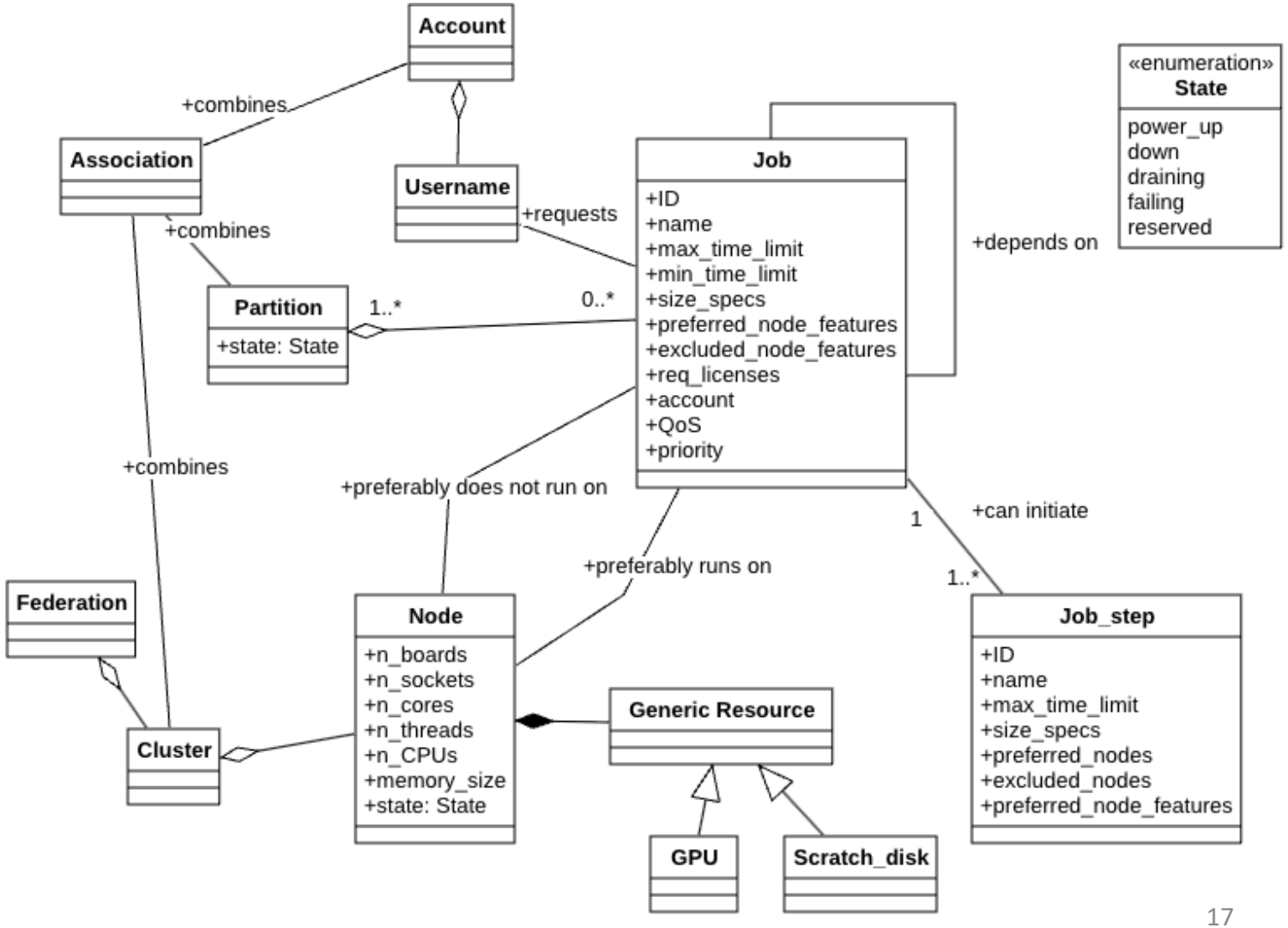
- **Job**: resource allocation request, two types
  - batch script to be queued for later execution,
  - interactive job for which the user awaits resource allocation and then makes real time use of it
- **Job step**: a set of parallel tasks, typically an MPI application
- **Node**: computational element
- **Cluster**: a collection of nodes sharing the same network
- **Federation**: a collection of clusters sharing a common configuration DB
- **Job partition**: is a job queue. Note that a job can belong to multiple partitions

Slurm main concepts

- **Account**: access group that gives users access to resources on a cluster
- **Slurm association**: combination of cluster, account, username, and (optional) partition name

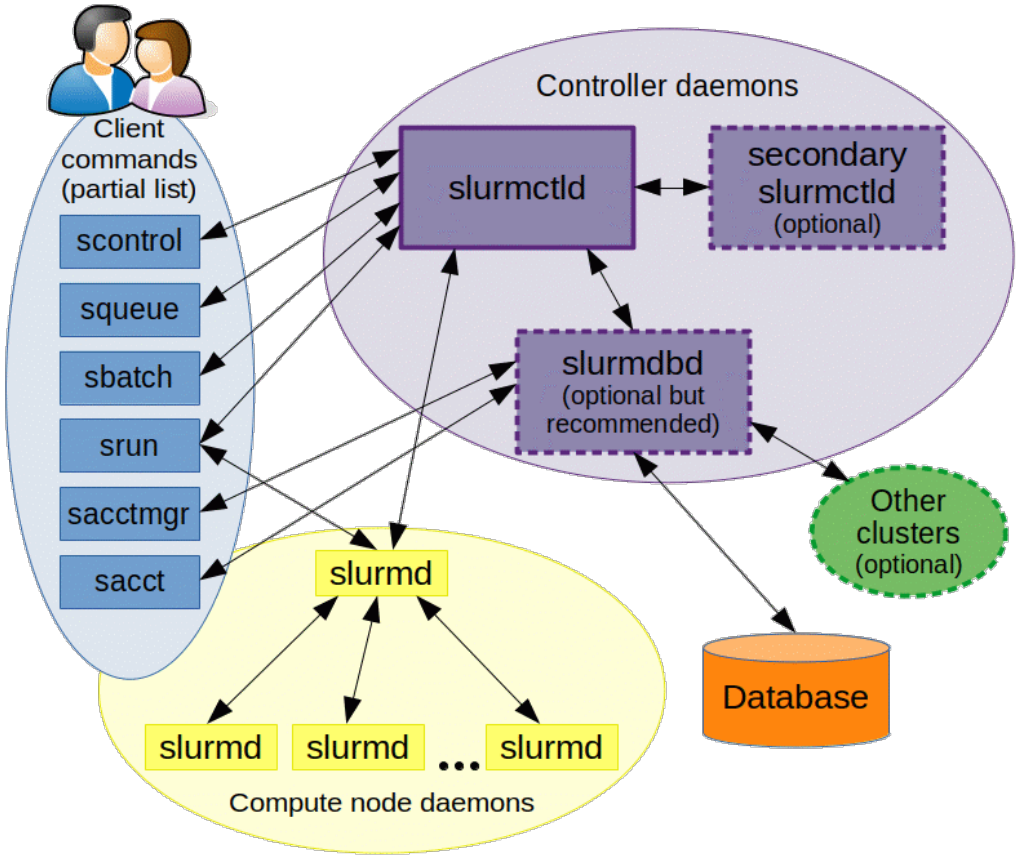


Slurm main concepts



Slurm architecture

- **slurmd**: controls the execution of jobs and job steps on a node
- **slurmctld**: monitors the state of resources, decides when and where to initiate jobs and job steps, and processes almost all user commands
- **slurmdbd**:
  - interfaces with a DBMS MariaDB or MySQL
  - records accounting information and manages some configuration information centrally (many limits, fair share information, QOS, licenses, etc.)
- **slurmrestd**: interface to Slurm via REST
- Component replication and caching to increase fault tolerance



Slurm commands

POLITECNICO MILANO 1863

- **sbatch**: submits a job script for later execution
- **srun**: submits a job for execution or initiate job steps in real time
- **salloc**: allocates resources for a job in real time

- **squeue**: reports the state of jobs or job steps
- **sinfo**: reports the state of partitions and nodes managed by Slurm

- https://slurm.schedmd.com/pdfs/summary.pdf

Example: script.py

```
import sys
import os
if len(sys.argv) != 2:
  print('Usage: %s MAXIMUM' % (os.path.basename(sys.argv[0])))
        sys.exit(1)
maximum = int(sys.argv[1])
n1 = 1
n2 = 1
while n2 <= maximum:
    n1, n2 = n2, n1 + n2
print('The greatest Fibonacci number up to %d is %d' % (maximum, n1))
```

Receives an integer as parameter and finds the closest but not higher Fibonacci number

Slurm one step script example

```
#!/bin/bash

#SBATCH --job-name=maxFib      ## Name of the job
#SBATCH --output=maxFib.out    ## Output file
#SBATCH --time=10:00           ## Job Duration
#SBATCH --ntasks=1             ## Number of tasks (analyses) to run
#SBATCH --cpus-per-task=1      ## The number of threads the code will use
#SBATCH --mem-per-cpu=100M     ## Real memory(MB) per CPU required by the job.

## Load the python interpreter
module load python

## Execute the python script and pass the argument/input '90'
srun python script.py 90
```

Step

What should we do to run it on an HPC cluster with Slurm?

POLITECNICO MILANO 1863

- We create on the cluster the files script.py and the Slurm script (e.g., script.sh)
- We assign execution permissions to script.sh
  - `chmod +x script.sh`
- We use the sbatch command to submit the script
  - `sbatch script.sh`
- We can use squeue to see the progress
  - `squeue -u <your username>`
- Once completed, we will find the file maxFib.out with the result

Scheduling approach

- Slurm Default: FIFO
- With the Multifactor Priority Plugin: ordered by priority
- But if a job can be scheduled without delaying the start of any other jobs with higher priority, the backfill scheduler may launch it first

Multifactor Priority Plugin

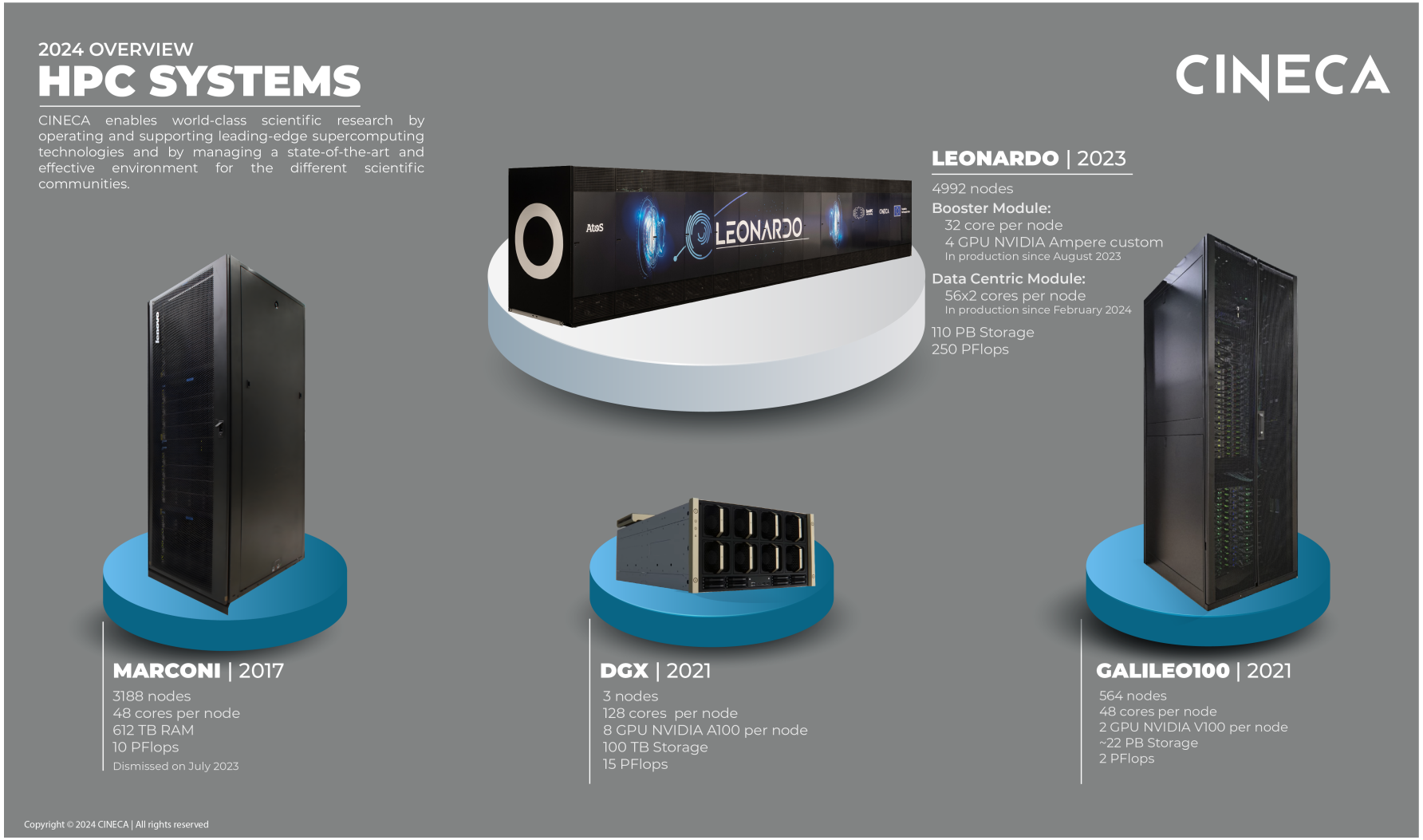
POLITECNICO MILANO 1863

- Job_priority = (PriorityWeightAge * age_factor) + (PriorityWeightQOS * QOS_factor) + (PriorityWeightPartition * partition_factor) + (PriorityWeightJobSize * job_size_factor) + (PriorityWeightFairshare * fair-share_factor) + (PriorityWeightAssoc * assoc_factor) + SUM(TRES_weight_<type> * TRES_factor_<type>, ...) - nice_factor + site_factor
- All factors are between 0 and 1
- All weights are large positive numbers

Scheduling approach

- Simple and practical material on the scheduling approach
  - https://mmesiti.github.io/slurmprio200428/
  - https://youtu.be/p8ebqQ9CHFU?feature=shared

HPC clusters at CINECA



HPC @ CINECA

POLITECNICO MILANO 1863

- Documentation
  - https://wiki.u-gov.it/confluence/display/SCAIUS/HPC+User+Guide
  - https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.3%3A+GALILEO100+UserGuide
- Steps to follow
  - Create groups of two to three people
  - Elect a group representative who will fill in the form concerning the acceptance of the CINECA privacy policy and of the user responsibilities:
    - https://eventi.cineca.it/en/hpc/software-engineering-hpc-politecnico-di-milano/20240520
  - Fill in the form available here https://forms.office.com/e/skv50CAeLm providing the evidence that you have filled in the CINECA form
  - I'll send the account to each group representative and further information on how to proceed