

# Configurazione di SLURM in ambiente HPC

+ esperienza al CNAF

26 Novembre 2024 [ 17:00 - 18:30 ]

Speaker: Alessandro Pascolini

# Overview

## 1. [Introduzione](#)

- Termini utili
- Calcolo HPC
- SLURM Overview

## 2. [Setup Cluster HPC con SLURM](#)

- Step preliminari
- Configurazione di base

## 3. [Gestione cluster](#)

- Comandi base di SLURM
- Accounting & QoS

## 4. [Esempi di Sottomissione locale](#)

## 5. [Esempio Cluster HPC@CNAF](#)

## 6. [Configurazione per accesso GRID](#)

# Introduzione

# Termini utili

## Demoni

→ processi, solitamente di sistema, eseguiti in background.

No interazione da parte dell'utente.

Di solito i nomi finiscono per «d» (e.g. sshd)

Batch System → software che gestisce un insieme di macchine per l'esecuzione di processi non interattivi (batch)

processi non

## Batch Job

→ processo tipicamente non interattivo (batch) che l'utente finale vuole

eseguire

Acceleratori → hardware specifico, migliora le prestazioni per specifici use-case (e.g. GPU, FPGA, ecc.)

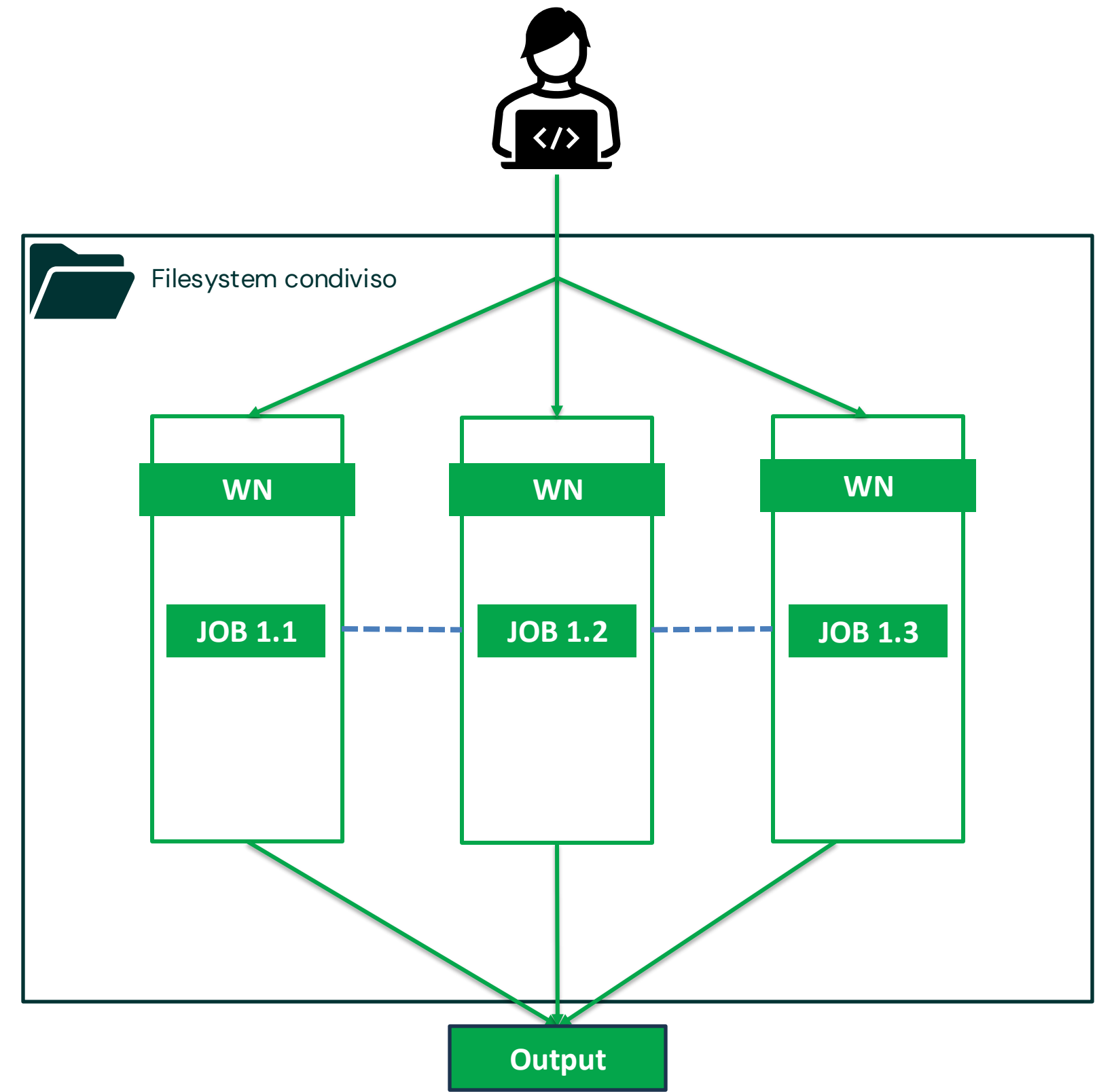
Worker-Node → macchina che esegue i Job dell'utente. Può montare solo CPU o anche

acceleratori

# Calcolo HPC

## HPC – High Performance Computing

- Calcolo **parallelo**, distribuito su più nodi
- Necessita di hardware ad alte prestazioni:
  - Connessione a **bassa latenza e ampia banda** (e.g. infiniband)
  - Acceleratori (e.g. GPU, FPGA)
- Solitamente i nodi di calcolo **condividono il filesystem**, in modo da sfruttare al massimo la parallelizzazione dei workflow



# SLURM workload manager

## Introduzione

Slurm is an **open source, fault-tolerant, and highly scalable** cluster management and **job scheduling system** for large and small Linux clusters. Slurm requires no kernel modifications for its operation and is relatively self-contained.

Da documentazione di SLURM<sup>[1]</sup>



[1] <https://slurm.schedmd.com/overview.html>

# SLURM workload manager

## Caratteristiche

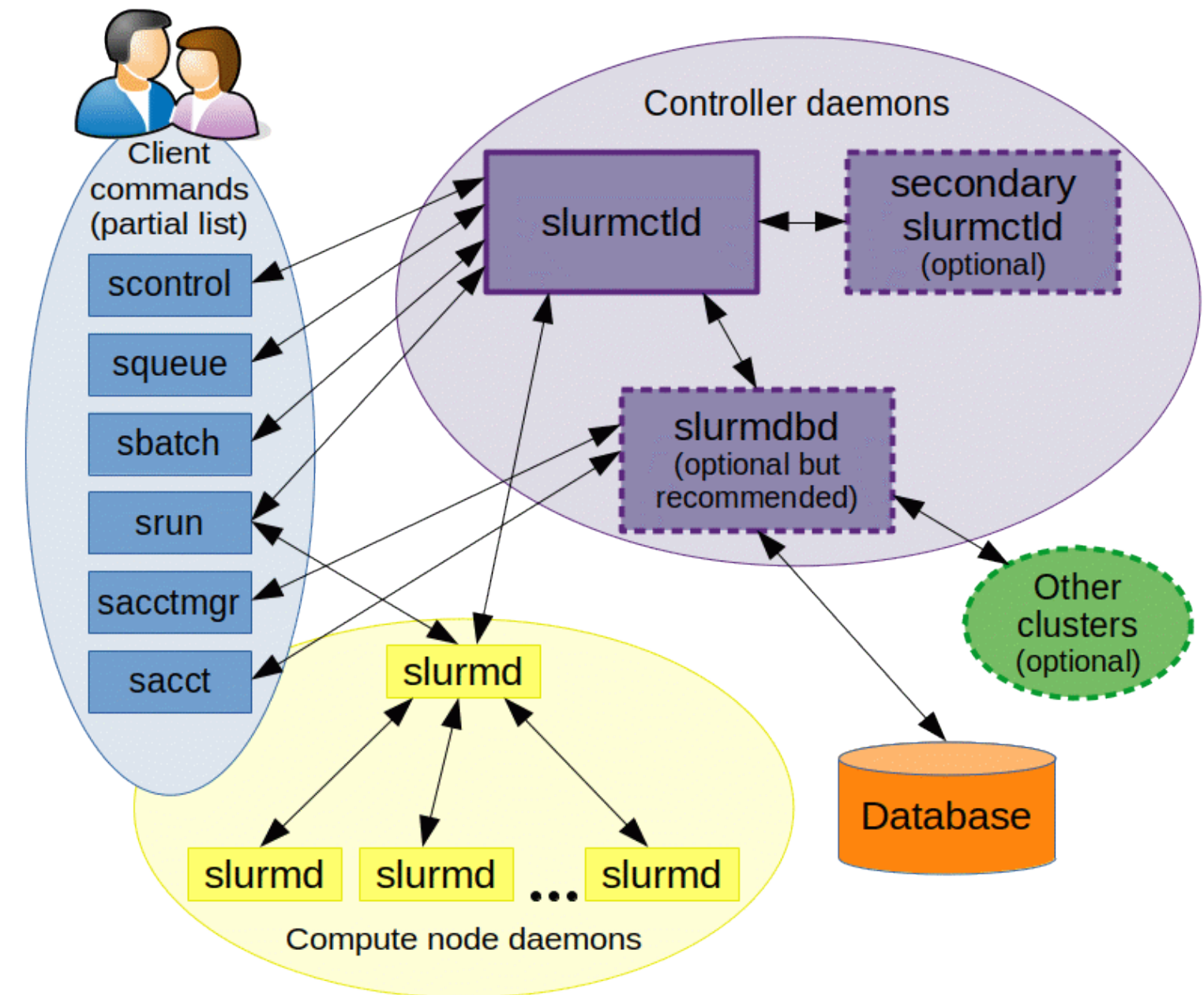
- **Batch System ottimizzato per calcolo HPC**
  - Multi-nodo (mpi)
  - Utilizzo di acceleratori (e.g. GPU)
- **Modulare**
  - Ampio ecosistema di plugin per la gestione di utenti e gruppi
- **Semplice**
  - Pochi demoni
  - Tool per la configurazione



# SLURM workload manager

## Architettura

- **slurmctld**
  - controller del cluster,
  - possibile configurazione in HA
- **slurmd**
  - gestisce l'esecuzione dei job
- **slurmdbd**
  - demone che si occupa di accounting
  - non necessario ma raccomandato





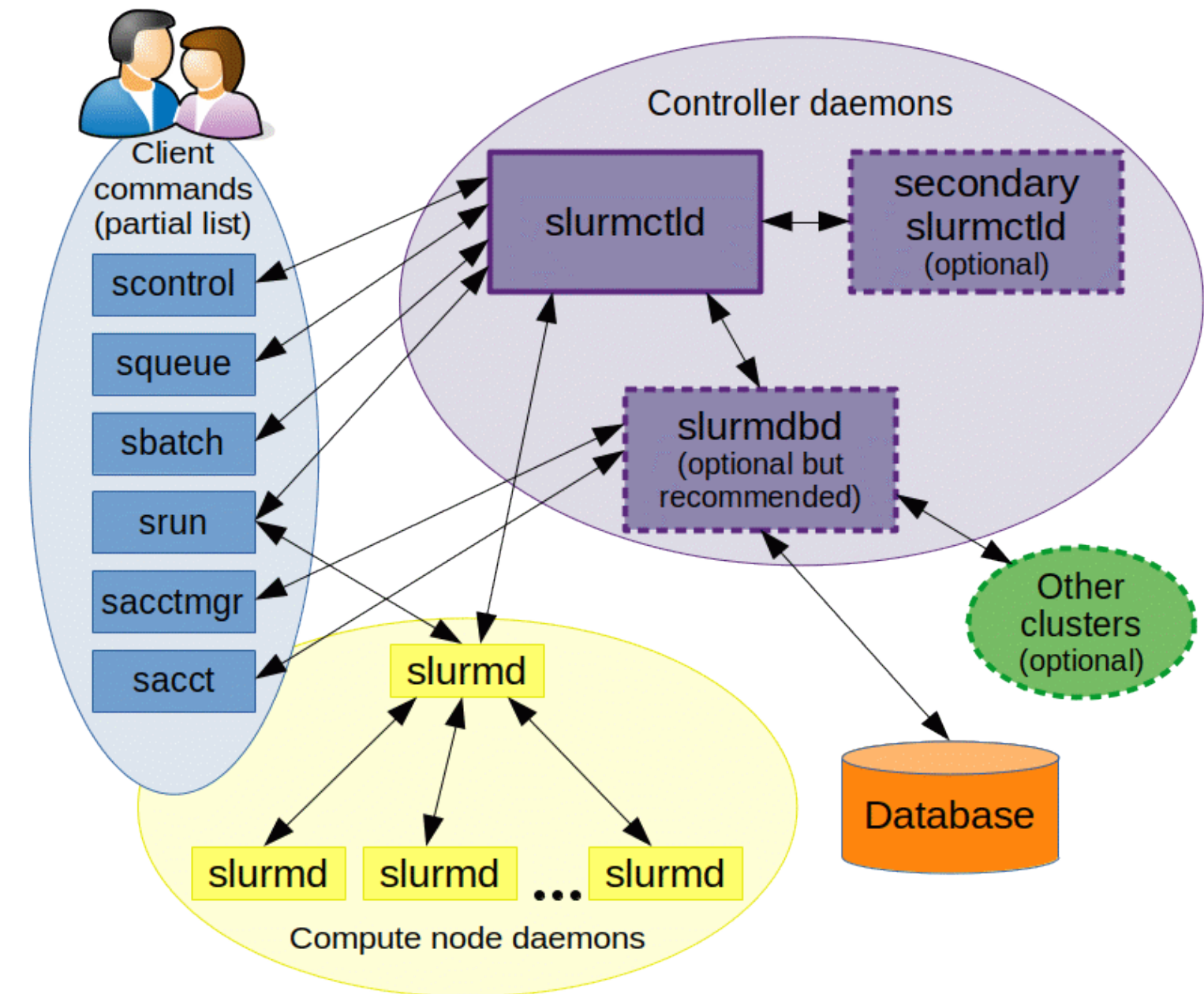


# Setup cluster HPC con SLURM

# Cluster SLURM

## Configurazione di un semplice cluster

- **Obiettivo:** setup di un semplice cluster con:
  - 1 controller
  - N worker-nodes (anche supporto GPU NVIDIA)



La procedura che mostrata è riferita ad un **cluster EL9**, in base al Sistema Operativo alcuni passaggi potrebbero variare

# Step Preliminari

## Lasciati al gentile lettore :)

1. Configurazione (pacchetti, kernel, OS) più possibile allineata tra i nodi
2. Utenti e gruppi devono essere sincronizzati tra tutte le macchine del cluster (e.g. LDAP)
3. **MUNGE** (per autenticazione tra nodi)
4. **DIVER**
  - Nel caso di nodi con acceleratori sarà necessario installare i driver necessari ad interagire con l'hardware del nodo
    - Ad esempio: GPU NVIDIA Tesla H100 → da installare i driver nvidia
5. **SLURM**
  - Buildare rpm su uno dei nodi del cluster (permette a SLURM di essere ottimizzato per la specifica configurazione)
  - Preparare i file di configurazione per il cluster

# MUNGE<sup>[2]</sup>

Permette l'autenticazione dei processi tra un nodo e l'altro tramite UserID (UID) e GroupID (GID)

## SETUP:

1. Installazione del pacchetto

```
dnf install munge
```

2. Creare il file `/etc/munge/munge.key`

- Almeno 32B
- Owner: munge
- Permessi: 600

3. Distribuire la chiave in tutti i nodi del cluster

4. Avviare munged

```
systemctl enable --now munged
```



[2] <https://dun.github.io/munge/>

# DRIVER NVIDIA\*

Procedura piuttosto complessa, due possibili strade:

- 1) Download dell'eseguibile per installare i driver
- 2) Via RPM, complessa ma automatizzabile

\*GPU NVIDIA Tesla H100 sono le più comuni nel nostro use-case



# DRIVER NVIDIA

## 1) Download eseguibile generico

- Selezionare Linux 64-bit nella pagina di ricerca dei driver <https://www.nvidia.com/en-us/drivers/>
- Si scaricherà un eseguibile da eseguire sul nodo (NVIDIA-Linux-x86\_64-\*.run)



### Manual Driver Search

Search by product, product type or series



Data Center / Tesla



H-Series

NVIDIA H100 PCIe

Linux 64-bit

Any CUDA Toolkit Version

English (US)

Find

# DRIVER NVIDIA

## 2) Via RPM

- Selezionare Linux 64-bit  $\langle + os-version \rangle$  nella pagina di ricerca dei driver
- Scaricare l'RPM
- Da seguire le istruzioni che riporta il sito

### Release Highlights   Supported Products   Additional Information

Once you accept the download please follow the steps listed below

- ``sudo rpm -i nvidia-driver-local-repo-rhel9-550.127.08-1.0-1.x86_64.rpm``
- ``sudo dnf clean all``
- ``sudo dnf -y module install nvidia-driver:latest-dkms``

#### Piccola aggiunta personale

- `sudo dracut --regenerate-all --force`
- `sudo systemctl reboot`



### Manual Driver Search

Search by product, product type or series



Data Center / Tesla



H-Series

NVIDIA H100 PCIe

Linux 64-bit RHEL 9

Any CUDA Toolkit Version

English (US)

Find



# DRIVER NVIDIA

Anyway...

- L'importante è riuscire a comunicare con la GPU

```
[almalinux@slurm-worker-gpu ~]$ nvidia-smi
Tue Nov 19 20:10:29 2024

+-----+-----+-----+
| NVIDIA-SMI 550.127.08                | Driver Version: 550.127.08   | CUDA Version: 12.4   |
+-----+-----+-----+
| GPU  Name           Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp    Perf     Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
|  0  Tesla V100-SXM2-32GB | Off          | 000000000:00:05.0 Off |   0
| N/A   29C    P0      36W / 300W | 1MiB / 32768MiB |    3%      Default  | |
|                               |                      |                      |     MIG M.           |
|                               |                      |                      |     N/A              |
+-----+-----+-----+

+-----+-----+-----+
| Processes:                               |
| GPU  GI    CI           PID  Type  Process name          | GPU Memory |
|      ID    ID                                     |            | Usage        |
+-----+-----+-----+
| No running processes found              |             |              |
+-----+-----+-----+
```



# SLURM

## Build RPM<sup>[3]</sup>

- Download SLURM<sup>[4]</sup>
- Build RPM su una macchina del cluster\*  
`rpmbuild -ta slurm-*.tar.bz2`
- Una volta finito il build, gli rpm saranno in  
`$HOME/rpmbuild/RPMS/x86_64/` da lì basterà un:  
`yum install -y $HOME/rpmbuild/RPMS/x86_64/*`

[3] [https://slurm.schedmd.com/quickstart\\_admin.html#rpmbuild](https://slurm.schedmd.com/quickstart_admin.html#rpmbuild)

[4] <https://www.schedmd.com/download-slurm/>

### Prerequisiti

- munge-devel  
(da crb → `dnf config-manager --set-enabled crb`)
- pam-devel
- perl
- readline-devel
- automake
- autoconf
- mariadb-devel  
→ per accounting
- dbus-devel  
→ per cgroups

\*è importante che kernel, librerie e pacchetti siano uguali a quelli disponibili poi sulle machine

# SLURM

## File di configurazione

- Tool per una configurazione di base<sup>[5]</sup>
- Path di default per la conf: **/etc/slurm/**
- Produce il file **slurm.conf**<sup>[6]</sup> con la descrizione della conf di tutto il cluster
- File da diffondere su **tutte le macchine** del cluster\*
- Contiene:
  - Hostname dei Controller
  - Lista di Worker-Nodes con risorse disponibili
  - Configurazione di Accounting
  - Gestione processi
  - Scheduling

[5] <https://slurm.schedmd.com/configurator.html>

[6] <https://slurm.schedmd.com/slurm.conf.html>

## Slurm Version 24.05 Configuration Tool

This form can be used to create a Slurm configuration file with you controlling many of the important configuration parameters.

This is the full version of the Slurm configuration tool. This version has all the configuration options to create a Slurm configuration file. There is a simplified version of the Slurm configuration tool available at [configurator.easy.html](https://slurm.schedmd.com/configurator.easy.html).

**This tool supports Slurm version 24.05 only.** Configuration files for other versions of Slurm should be built using the tool distributed with it in *doc/html/configurator.html*. Some parameters will be set to default values, but you can manually edit the resulting *slurm.conf* as desired for greater flexibility. See *man slurm.conf* for more details about the configuration parameters.

Note that while Slurm daemons create log files and other files as needed, it treats the lack of parent directories as a fatal error. This prevents the daemons from running if critical file systems are not mounted and will minimize the risk of cold-starting (starting without preserving jobs).

Note that this configuration file must be installed on all nodes in your cluster.

After you have filled in the fields of interest, use the "Submit" button on the bottom of the page to build the *slurm.conf* file. It will appear on your web browser. Save the file in text format.

For more information

**Cluster Name**

to record information

**Control Machine**

Define the hostname of the control machine and its domain name (e.g. use

**Compute Machine**

Define the machines of the compute nodes. The possible parameters will print its physical name into a single partition. Manually edit the *slurm.conf* file.

The following parameters are used to define resources:

ThreadsPerCore.

**Sockets:** Number of physical processor sockets/chips on the node. If Sockets is omitted, it will be inferred from: CPUs, CoresPerSocket, and ThreadsPerCore.

```
# slurm.conf file generated by configurator.html.
# Put this file on all nodes of your cluster.
# See the slurm.conf man page for more information.
#
ClusterName=cluster
SlurmctldHost=linux0
#SlurmctldHost=
#
#DisableRootJobs=NO
#EnforcePartLimits=NO
#Epilog=
#EpilogSlurmctld=
#FirstJobId=1
#MaxJobId=67043328
#GresTypes=
#GroupUpdateForce=0
#GroupUpdateTime=600
#JobFileAppend=0
#JobRequeue=1
#JobSubmitPlugins=lua
#KillOnBadExit=0
#LaunchType=launch/slurm
#Licenses=foo*4,bar
#MailProg=/bin/mail
#MaxJobCount=10000
#MaxStepCount=40000
#MaxTasksPerNode=512
#MpiDefault=
#MpiParams=ports=#-#
```

# SLURM

## Partizioni

- **Partizione**
  - insieme di nodi con caratteristiche comuni
- Ogni nodo può appartenere a una o più partizioni
- Con l'accounting è possibile riservare partizioni a specifici utenti o workflow

```
###  
# slurm.conf per nodi e partizioni  
###  
  
# COMPUTE NODES  
NodeName=slurm-worker-gpu NodeAddr=192.168.0.114 CPUs=8 State=UNKNOWN Gres=gpu:tesla:1  
  
PartitionName=debug Nodes=ALL Default=YES MaxTime=INFINITE State=UP
```

# SLURM

## File di configurazione

- Altri file:
  - **gres.conf**<sup>[7]</sup>  
→ file in cui sono listate le Generic RESources (GRES) dei nodi

```
#####  
# Slurm's Generic Resource (GRES) configuration file  
# Define GPU devices with MPS support, with AutoDetect sanity checking  
##### AutoDetect=nvml  
Name=gpu Type=gtx560 File=/dev/nvidia0 COREs=0,1  
Name=gpu Type=tesla File=/dev/nvidia1 COREs=2,3
```

Esempio di gres.conf

[7] <https://slurm.schedmd.com/gres.conf.html>

# SLURM

## File di configurazione

- Altri file:
  - **gres.conf**<sup>[7]</sup>  
→ file in cui sono listate le Generic RESources (GRES) dei nodi
  - **cgroup.conf**<sup>[8]</sup>  
→ definisce i parametri che verranno usati dal plugin di cgroup

```
###  
# slurm.conf per abilitare cgroups  
###  
ProctrackType=proctrack/cgroup  
TaskPlugin=task/cgroup,task/affinity  
JobAcctGatherType=jobacct_gather/cgroup #optional for gathering metrics  
PrologFlags=Contain #X11 flag is also suggested
```

```
###  
# cgroup.conf usual config  
###  
ConstrainCores=yes  
ConstrainDevices=yes  
ConstrainRAMSpace=yes  
ConstrainSwapSpace=yes
```

[7] <https://slurm.schedmd.com/gres.conf.html>

[8] <https://slurm.schedmd.com/cgroup.conf.html>

# SLURM

## Avvio Cluster

- Assicurarsi che esista lo SlurmUser
  - definito in **slurm.conf**
  - default: **slurm**
- Files/directories definiti in slurm.conf non vengono creati da SLURM
  - assicurarsi che esistano e siano accessibili all'utente **slurm**
- Avviare i demoni nel cluster:
  - slurmctld → controller
  - slurmd → workers
- Controllare che i nodi siano visibili al cluster

```
[almalinux@slurm-login ~]$ sinfo -N
NODELIST      NODES PARTITION STATE
slurm-worker-gpu  1  debug* idle
...
...
...
```



# Gestione SLURM

# Comandi SLURM<sup>[9]</sup>

## Sottomissione/Gestione Job

- **sbatch<sup>[10]</sup>**  
→ sottomissione di batch jobs a slurm
- **squeue<sup>[11]</sup>**  
→ lista i job presenti in coda  
(by default di tutti gli utenti)
- **scancel<sup>[12]</sup>**  
→ rimuove job dalla coda

```
[almalinux@slurm-login tests]$ sbatch test.sub  
Submitted batch job 9
```

```
[almalinux@slurm-login tests]$ squeue --me  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
9 debug test.sub almalinu R 0:10 1 slurm-worker-gpu
```

```
[almalinux@slurm-login tests]$ scancel 9
```

```
[almalinux@slurm-login tests]$ squeue --me  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
9 debug test.sub almalinu CG 0:27 1 slurm-worker-gpu
```

```
[almalinux@slurm-login tests]$ squeue --me  
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

[9] <https://slurm.schedmd.com/pdfs/summary.pdf>

[10] <https://slurm.schedmd.com/sbatch.html>

[11] <https://slurm.schedmd.com/squeue.html>

[12] <https://slurm.schedmd.com/scancel.html>



# Comandi SLURM<sup>[9]</sup>

## Interazione con SLURM-Workers

- **sinfo**<sup>[13]</sup>
  - lista i nodi appartenenti al cluster
  - info su risorse, stato e disponibilità dei nodi
- **scontrol**<sup>[14]</sup>
  - modifica dello stato dei nodi/job
  - la maggior parte delle opzioni sono limitate a utente **root** o **amministratore**

```
[almalinux@slurm-login tests]$ sinfo -N
NODELIST      NODES PARTITION STATE
slurm-worker-gpu  1  debug* idle
```

```
[almalinux@slurm-login tests]$ scontrol update NodeName=slurm-worker-gpu State=drain Reason="Test"
slurm_update error: Invalid user id
```

```
[almalinux@slurm-login tests]$ sudo !!
sudo scontrol update NodeName=slurm-worker-gpu State=drain Reason=«Test»
```

```
[almalinux@slurm-login tests]$ sinfo -N
NODELIST      NODES PARTITION STATE
slurm-worker-gpu  1  debug* drain
```

[9] <https://slurm.schedmd.com/pdfs/summary.pdf>

[13] <https://slurm.schedmd.com/sinfo.html>

[14] <https://slurm.schedmd.com/scontrol.html>

# Accounting<sup>[15]</sup>

## SlurmDBD

- Demone di SLURM per la gestione dell'accounting
- Gestisce:
  - Salvataggio di dati di accounting su DB/file di testo
  - AuthN/Z per accesso a dati di accounting da parte degli utenti
- Ottimizzato e ben integrato con gli altri demoni

→ **Altamente consigliato**



[15] <https://slurm.schedmd.com/accounting.html>

# Accounting<sup>[15]</sup>

## Setup

In **slurm.conf**

- **AccountingStorageType**  
→ indica come salvare i dati
- **AccountingStorageEnforce**  
→ specifica su cosa basare l'accounting

SlurmDBD ha un suo file di configurazione: **slurmdbd.conf**<sup>[16]</sup>

```
###  
# slurm.conf  
###  
  
AccountingStorageType=accounting_storage/slurmdbd # abilita l'accounting via SlurmDBD  
AccountingStorageEnforce=associations           # permette di sottomettere solo se  
                                                    # l'utente è registrato nell'accounting
```

[15] <https://slurm.schedmd.com/accounting.html>

[16] <https://slurm.schedmd.com/slurmdbd.conf.html>

# Accounting<sup>[15]</sup>

## Database

- L'accounting SLURM può essere salvato in:
  - File di testo
  - Database MySQL/MariaDB (richiesto mysql/mariadb-devel)
- Per MySQL/MariaDB è necessario:
  - Creare l'utente **<StorageUser>** nel DB
  - Creare il DB **<StorageLoc>** e dare accesso all'utente **<StorageUser>**

```
###  
# slurmdbd.conf  
###  
  
StorageType=accounting_storage/mysql # sets accounting to a mysql/mariadb database  
  
StorageHost=<database-host>  
StoragePort=3306 #default port  
StorageUser=slurm  
StoragePass=<password>  
StorageLoc=<db_name> # default slurm_acct_db
```

```
mysql> create user 'slurm'@'slurmdbd-host' identified by 'password';  
  
mysql> grant all on slurm_acct_db.* TO 'slurm'@'localhost';
```

[15] <https://slurm.schedmd.com/accounting.html>

# Accounting<sup>[15]</sup>

## Comandi

- **sacct**<sup>[17]</sup>
  - per ottenere informazioni riguardo a job running o completati
- **sacctmgr**<sup>[18]</sup>
  - per la modifica delle entries nel DB, utile per gestire utenti, account, QoSs etc.
  - comando per admin!
- **sreport**<sup>[19]</sup>
  - genera report sull'utilizzo del cluster

[15] <https://slurm.schedmd.com/accounting.html>

[17] <https://slurm.schedmd.com/sacct.html>

[18] <https://slurm.schedmd.com/sacctmgr.html>

[19] <https://slurm.schedmd.com/sreport.html>

# Gestione Utenti e Gruppi

- **Cluster**  
→ gruppo di macchine che fanno capo a uno **slurmctld**
- **Account**  
→ gruppo di utenti a cui sono assegnate delle risorse
- **User**  
→ singolo utente UNIX che può appartenere a più di un account



```
# add new cluster to accounting
[root@slurm-login ~]# sacctmgr add cluster test-cluster

# add «test» account to «test-cluster»
[root@slurm-login ~]# sacctmgr add account test Cluster=test-cluster

# add user «almalinux» to «test» account
[root@slurm-login ~]# sacctmgr add user almalinux Account=test
```

# QoS & resource limits

- **QoS (Quality of Service)**<sup>[20]</sup>
  - di una partizione (in **slurm.conf**)
  - di un utente/account
- I limiti possono essere essere su:
  - risorse (e.g. num. cpus, num. gpus)
  - cputime
  - priority & preemption
  - tanto altro

[20] <https://slurm.schedmd.com/qos.html>

```
# listare qos
$ sacctmgr show qos format=name,priority
  Name  Priority
-----
  normal  0

# aggiungere una qos
$ sacctmgr add qos zebra
Adding QOS(s)
      zebra
Settings
      Description = QOS Name

$ sacctmgr show qos format=name,priority
                                Name  Priority
-----
                                normal  0
                                zebra   0

$ sacctmgr modify qos zebra set GrpTRES=cpu=24
Modified qos...
zebra
$ sacctmgr show qos format=name,priority,GrpTRES
  Name  Priority  GrpTRES
-----
  normal  0
  zebra   10   cpu=24
```



# Esempi di sottomissione



# Submit File

## Struttura

```
#!/bin/bash
```

```
#SBATCH -o job.out  
#SBATCH -e job.err
```

```
export PATH=$HOME/.local/bin:$PATH
```

```
srun sleep 1000
```

- **shebang**
- **Direttive SLURM**
  - caratteristiche e requirements del job
- **setup environment**
  - setta variabili d'ambiente utili per l'esecuzione
- **comando da eseguire**

# Submit File

## SBATCH options

- **#SBATCH --partition=<NAME>**  
→ specifica la partizione in cui sottomettere
- **#SBATCH --nodelist=<NODES>**  
→ specifica una lista di nodi (comma separated o “regex”)
- **#SBATCH --nodes=<INT>**  
→ numero di nodi su cui eseguire il/i job
- **#SBATCH --ntasks=<INT>**  
→ numero di CPU da utilizzare
- **#SBATCH --ntasks-per-node=<INT>**  
→ numero di CPU per nodo
- **#SBATCH --gres=<GRES>:<#GRES>**  
→ tipologia e numero di GRES

```
###  
# test-gpu.sub  
###  
  
#!/bin/bash  
  
#SBATCH --job-name=v100  
#SBATCH --nodes=1  
#SBATCH --ntasks=8  
#SBATCH --gres=gpu:4  
  
srun <gpu_script>
```

# Submit File

May get a bit complex

```
[a07cna00@login01 CNAF-VM-new]$ cat cnaf.job
#!/bin/bash
#SBATCH --qos=qos_cnaf
#SBATCH --constraint=kvmcpu
#SBATCH --requeue
#SBATCH --open-mode=append
#SBATCH --output="%x-%j.out"
#SBATCH -A CNAF4_WLCG
#SBATCH -p dcgp_usr_prod
#SBATCH -n 112
#SBATCH --mem=480g
#SBATCH -t 00:00:00
#SBATCH --gres=tmpfs:3T

repodir="/leonardo/home/usera07cna/a07cna00/CNAF-VM-new"
jobnameprefix="cn-leo-"
vm_root_disk="wn-leo-v1.5.qcow2"
vm_map_file="$repodir/vm_addresses_map_file.txt"

if [ ! -r "$vm_map_file" ]; then
    echo "ERROR: $vm_map_file not found or not readable"
    exit 1
elif [ ! -r "$repodir/$vm_root_disk" ]; then
    echo "ERROR: $repodir/$vm_root_disk not found or not readable"
    exit 1
fi

...
...
...

./virt-install.sh
```

Parte del submit file che usiamo al CNAF per creare WN HTCondor su Leonardo

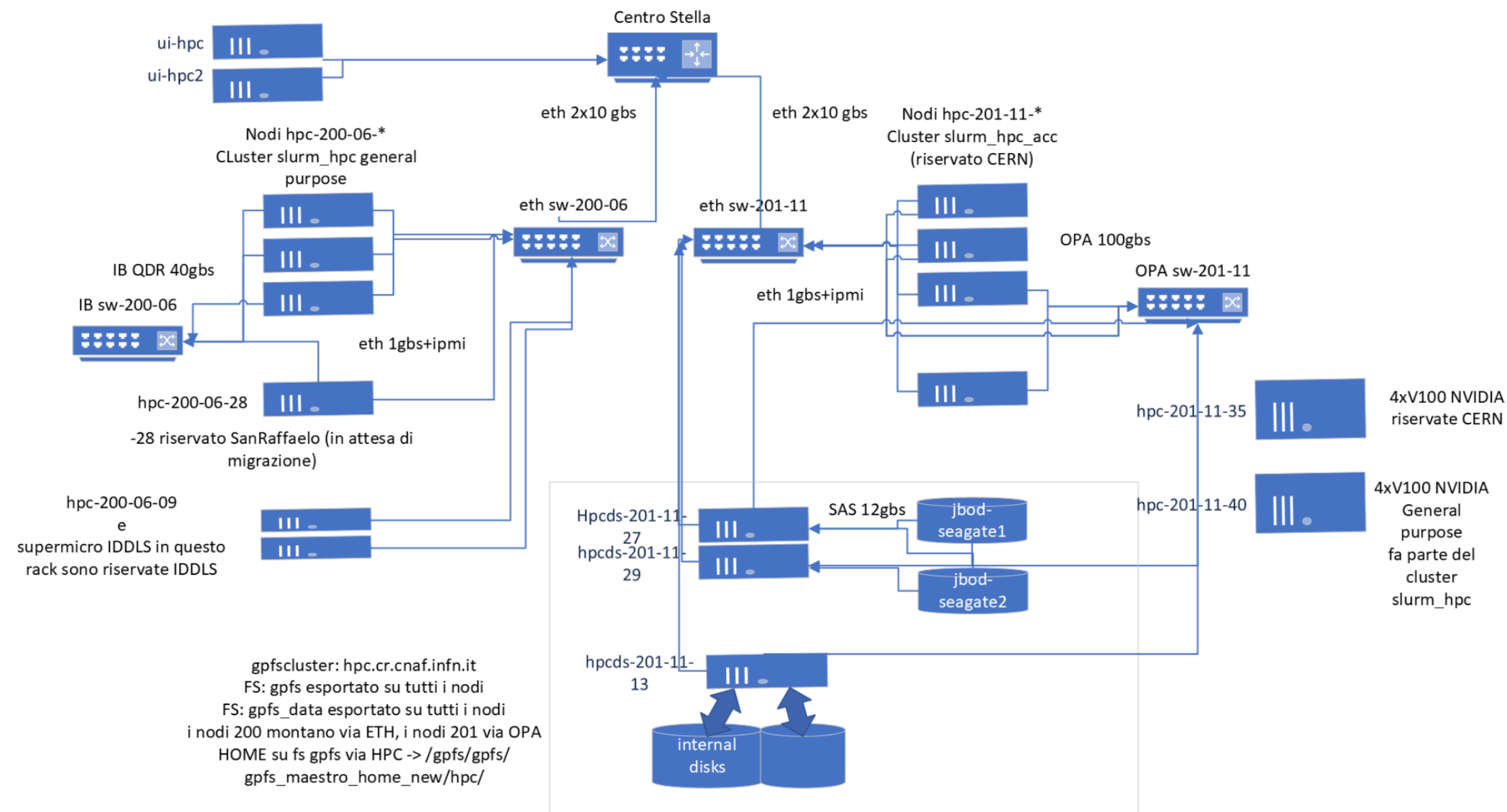


# Esempio cluster HPC@CNAF

# HPC@CNAF

## Struttura cluster

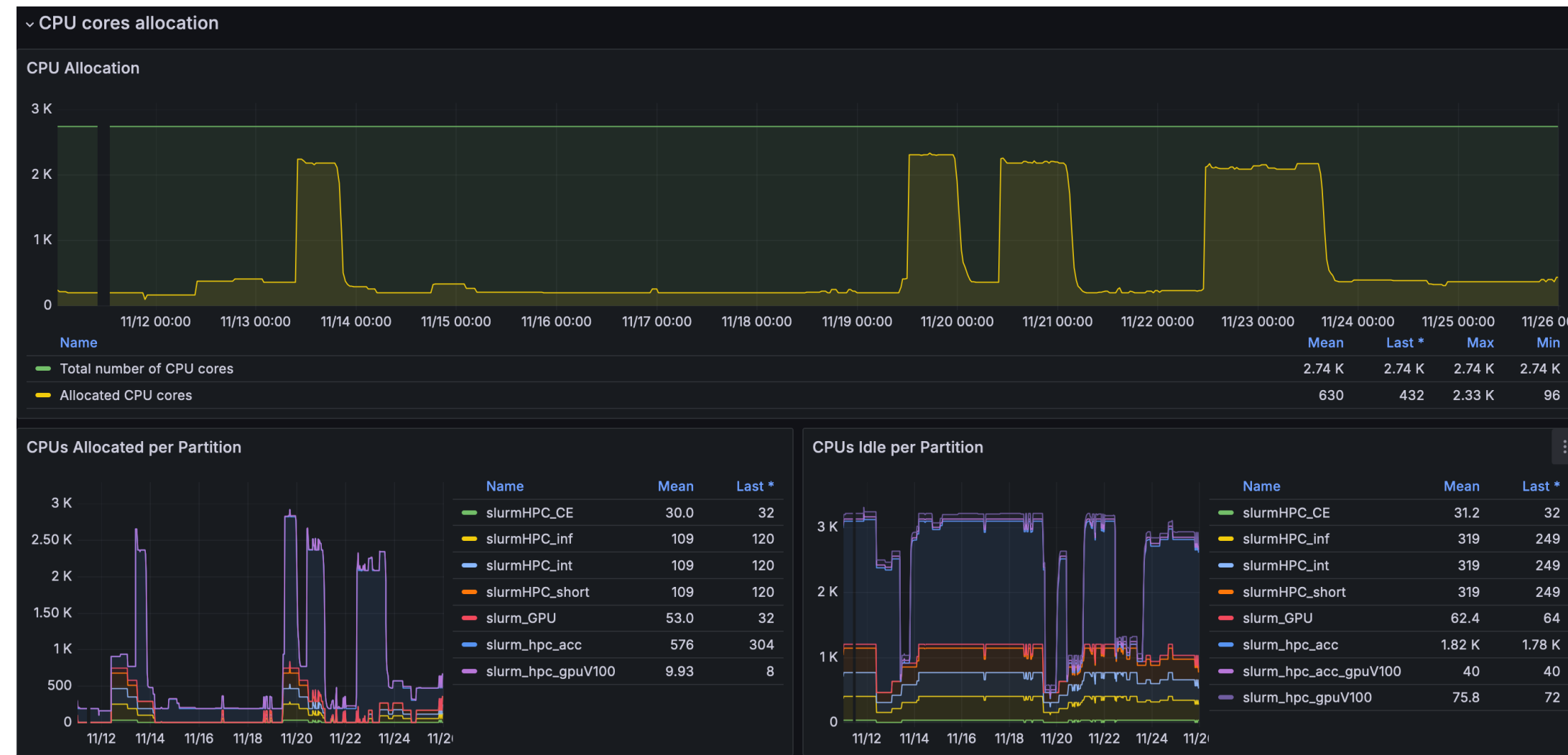
- Cluster con architetture miste:
  - nodi CPU
  - nodi GPU
    - NVIDIA Tesla k40
    - NVIDIA Tesla V100
- Filesystem condiviso (GPFS) per:
  - /home
  - scratch dir
- Connessione tra nodi:
  - infiniband (General purpose)
  - omnipath (CERN)



# HPC@CNAF

## Conf SLURM

- **slurmctld** in HA su 2 login nodes:
  - ui-hpc
  - ui-hpc2
- **slurmdbd** su ui-hpc con backup su ui-hpc2
- Cluster diviso in 2 partizioni
  - Generale
    - accessibile a TUTTI gli utenti
  - CERN
    - nodi riservati a solo utenti CERN
- DB Accounting su cluster Percona
- **Probe prometheus<sup>[21]</sup>** per monitoraggio



[21] <https://github.com/vpenso/prometheus-slurm-exporter>

# HPC@CNAF

## Partizioni

- **slurmHPC\_int**
  - partizione di default
  - MaxTime allowed for computation = 79h
- **slurm\_GPU**
  - partizione che include i nodi con Tesla k40
- **slurm\_hpc\_gpuV100**
  - partizione con nodi con GPU Tesla V100
- **slurm\_hpc\_acc**
  - nodi dedicati a utenti CERN
- **slurm\_hpc\_acc\_gpuV100**
  - nodi GPU dedicati a utenti CERN

## Accesso Interattivo

- L'accesso via ssh da rete CNAF è garantito su tutti i nodi
- Utenti del cluster GENERAL non possono accedere ai nodi CERN

## Accounting & QoS

- Non prevediamo enforcement di nessun tipo nel cluster
- Tranne limiti di tempo, non ci sono limiti di risorse per gli utenti

# HPC@CNAF

## Software

- Distribuito tramite **Environment Modules**<sup>[22]</sup>
  - **ampiamente utilizzati** in ambito HPC
  - alcune applicazioni per calcolo HPC **vengono con un loro modulo** (e.g. CUDA)
- Flessibilità e semplicità di utilizzo
- È possibile per un utente creare un proprio modulo



```
apascolinihpc@ui-hpc ~
$ module avail

----- /shared/software/modulefiles/ --
...
...
compilers/gcc-11_sl7
compilers/gcc-12.3_sl7
compilers/gcc-4.6.4
compilers/gcc-4.7.3
...
...

apascolinihpc@ui-hpc ~
$ gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-44)

apascolinihpc@ui-hpc ~
$ module load compilers/gcc-12.3_sl7

apascolinihpc@ui-hpc ~
$ gcc --version
gcc (GCC) 12.3.0
```

[22] <https://modules.readthedocs.io/en/latest/>





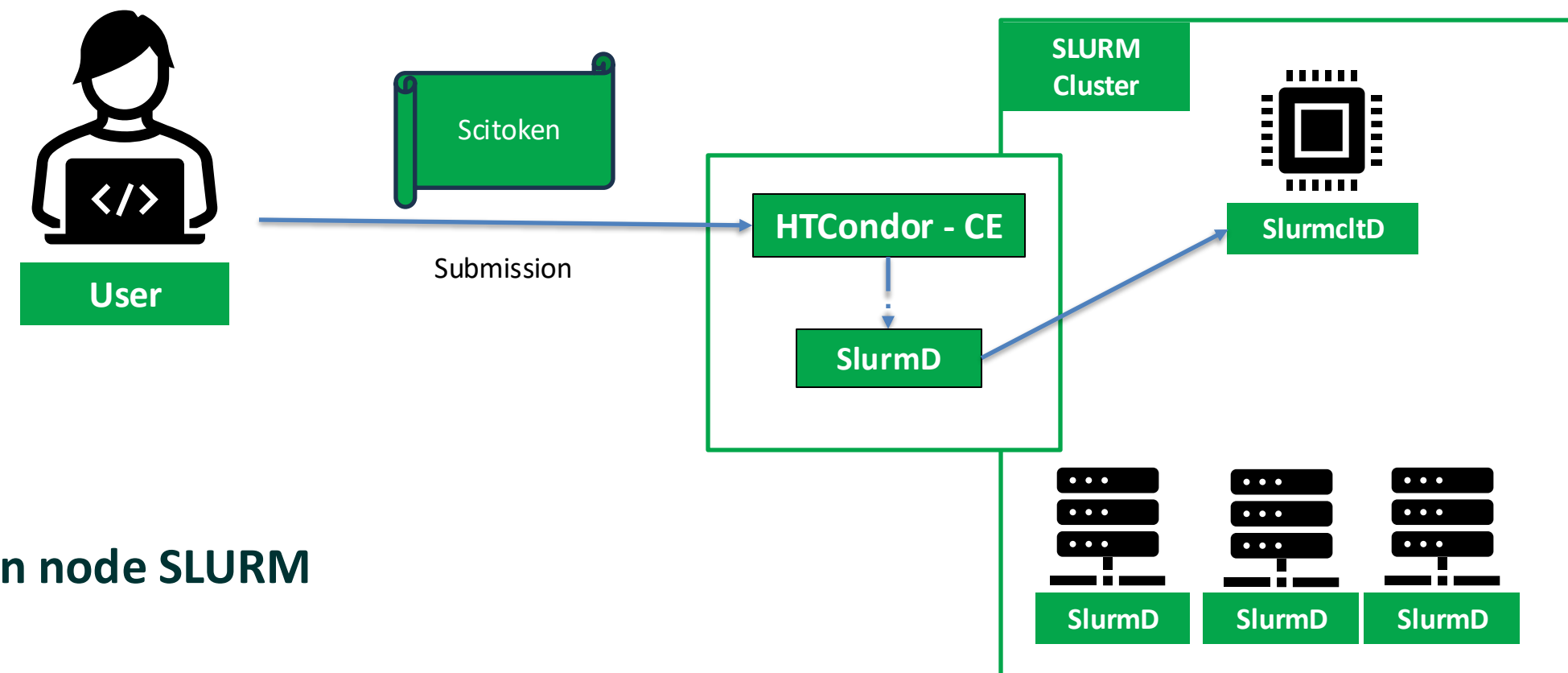
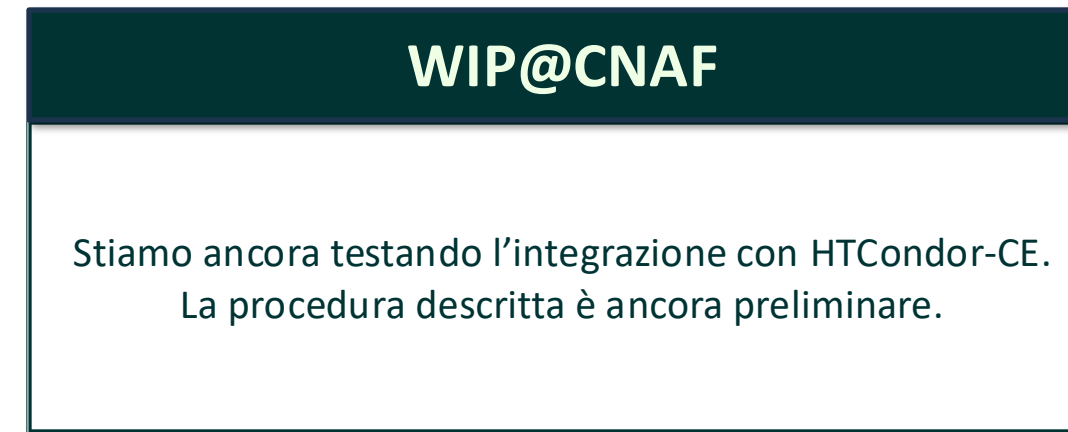
# Configurazione per accesso GRID

# HTCondor-CE<sup>[23]</sup>

HTCondor-CE (**Compute Entrypoint**) è un software per **accesso GRID**

- AuthN/Z (**SCITOKEN** o SSL)
- Sottomissione a un cluster remoto  
→ non è necessario accedere a un nodo del cluster SLURM

HTCondor-CE è un demone che **viene eseguito su un login node SLURM**  
→ interagisce con SLURM per sottomettere job

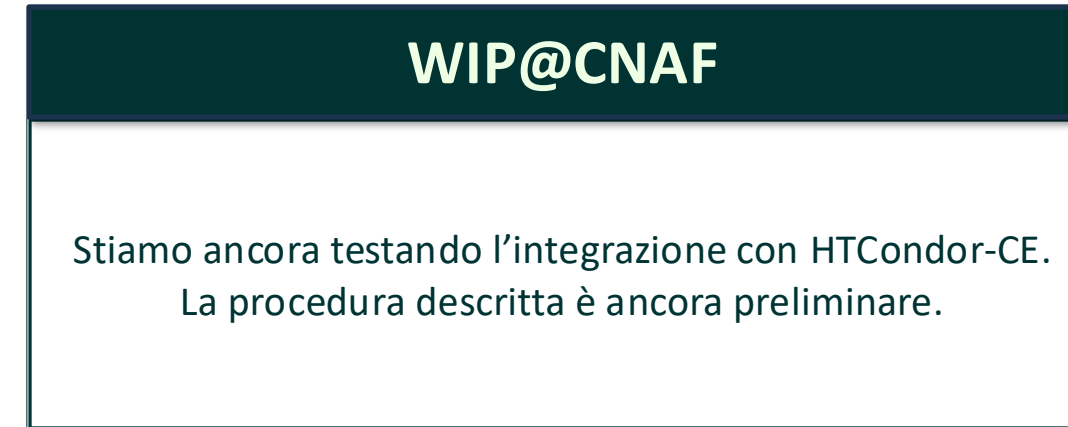


[23] <https://htcondor.com/htcondor-ce/>

# HTCondor-CE [23]

## Installazione & Configurazione

- Installazione di HTCondor-CE per SLURM  
`dnf install htcondor-ce-slurm`
- Configurazione mapping SCITOKENS  
→ in `/etc/condor-ce/mapfiles.d/`  
`SCITOKENS /<TOKEN ISSUER>,<TOKEN SUBJECT>/`  
`<USERNAME>`
- Setup certificati HTCondor-CE



```
###  
# /etc/condor-ce/mapfiles.d/01-scitokens.conf  
###  
  
SCITOKENS /^https:\\/\\/cms-auth(\\.web)?\\.cern\\.ch\\,08ca855e-d715-410e-a6ff-ad77306e1763/ cmssgm  
SCITOKENS /^https:\\/\\/cms-auth(\\.web)?\\.cern\\.ch\\,490a9a36-0268-4070-8813-65af031be5a3/ pilcms  
SCITOKENS /^https:\\/\\/cms-auth(\\.web)?\\.cern\\.ch\\,bad55f4e-602c-4e8d-a5c5-bd8ffb762113/ cmsprd
```

```
###  
# /etc/condor-ce/config.d/01-ce-auth.conf  
###  
  
AUTH_SSL_SERVER_CERTFILE = /etc/grid-security/hostcert.pem  
AUTH_SSL_SERVER_KEYFILE = /etc/grid-security/hostkey.pem  
AUTH_SSL_SERVER_CADIR = /etc/grid-security/certificates  
AUTH_SSL_SERVER_CAFILE =  
AUTH_SSL_CLIENT_CERTFILE = /etc/grid-security/hostcert.pem  
AUTH_SSL_CLIENT_KEYFILE = /etc/grid-security/hostkey.pem  
AUTH_SSL_CLIENT_CADIR = /etc/grid-security/certificates  
AUTH_SSL_CLIENT_CAFILE =
```

[23] <https://htcondor.com/htcondor-ce/>



Grazie per l'attenzione!