

## Istruzioni tecniche d'uso del cluster GPU DISI (aggiornamento Gennaio 2025)

Per l'utilizzo del cluster il primo passo è l'**abilitazione dell'account istituzionale** unibo.it all'accesso ai sistemi dipartimentali e al cluster stesso. In caso non si sia già abilitati si riceverà una e-mail di avvenuta abilitazione. Con le credenziali istituzionali si avrà accesso, anche in remoto, a tutte le macchine del laboratorio Ercolani. Nella e-mail si trova il link dedicato ai servizi informatici dipartimentali (<https://disi.unibo.it/it/dipartimento/servizi-tecnici-e-amministrativi/servizi-informatici>) e nella sezione *Accesso remoto* si trovano dettagli su tali macchine e su come accedervi. Inoltre è possibile accedere allo stesso modo alla macchina **giano.cs.unibo.it** da cui è possibile utilizzare lo schedatore del cluster e in cui è necessario predisporre l'ambiente di esecuzione dei job in quanto sono presenti le versioni aggiornate di Python ed eventuali ulteriori librerie richieste.

La **quota utente** massima è ad oggi impostata a 400 MB. In caso di necessità di maggiore spazio è possibile ricorrere alla creazione di una propria directory in **/scratch.hpc/**, in cui i singoli file vengono cancellati se non acceduti negli ultimi 40 giorni (la directory **/public/** viene invece di norma cancellata ogni prima domenica del mese, mentre la directory **/public.hpc/** è in dismissione). La home utente è uno spazio di archiviazione condiviso tra le macchine, quindi l'ambiente di esecuzione e i file necessari all'elaborazione presenti nella macchina **giano.cs.unibo.it** da cui poi avviare il job che verrà eseguito nelle macchine dotate di GPU saranno visibili anche in tutte le altre macchine di laboratorio. Le directory **/scratch.hpc/** e **/public.hpc/** sono invece visibili esclusivamente dalla macchina **giano.cs.unibo.it**.

Nel cluster sono presenti due code di schedulazione:

- **rtx2080**: con nodi di elaborazione (CPU singola quad-core, RAM 44 GB) contenenti ognuno una scheda grafica Nvidia GeForce RTX 2080 Ti (GPU Turing TU102 con 4352 core, memoria 11 GB) pilotate con driver Nvidia v. 535 e librerie di computazione CUDA 11.8.
- **l40**: con nodi di elaborazione (CPU singola octa-core, RAM 64 GB) contenenti ognuno una scheda grafica Nvidia L40 (GPU Ada Lovelace AD102GL con 18176 core, memoria 48 GB) pilotate con driver Nvidia v. 535 e librerie di computazione CUDA 11.8.

Una possibile **impostazione del lavoro** è quella di creare nella macchina **giano** un virtual environment Python inserendo all'interno tutto ciò di cui si ha bisogno e utilizzando **pip** per l'installazione dei moduli necessari; ad esempio in caso di installazione di *pytorch* bisognerà utilizzare il comando **pip3 install torch --no-cache-dir --index-url <https://download.pytorch.org/whl/cu118>** (rif. <https://pytorch.org/>).

**N.B.:** Il gestore di pacchetti pip utilizza per default una cache nello spazio utente, e la relativa quota potrebbe esaurirsi velocemente. Si consiglia quindi di includere sempre il parametro **--no-cache-dir** nel comando di installazione di moduli, e nel caso si debba cancellare una cache esistente di utilizzare il comando **pip3 cache purge**.

Il cluster utilizza uno schedatore SLURM (<https://slurm.schedmd.com/overview.html>) per la distribuzione dei job. Per effettuare il submit di un job bisogna predisporre nella propria area di lavoro un file di script SLURM (ad es. *script.sbatch*) in cui inserire le direttive per la configurazione del job stesso. Dopo le direttive è possibile inserire comandi di script (ad es. BASH). Un esempio di script è il seguente:

```
#!/bin/bash
#SBATCH --job-name=nomejob
#SBATCH --mail-type=ALL
#SBATCH --mail-user=nome.cognome@unibo.it
#SBATCH --time=01:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=8
#SBATCH --mem=31G
#SBATCH --partition=nomepartition
#SBATCH --output=nomeoutput
#SBATCH --chdir=/scratch.hpc/nome.cognome
#SBATCH --gres=gpu:1
```

```
venv/bin/python3 test.py # per attivare il virtual environment python
```

Nell'esempio precedente la direttiva da riportare immutata è `--gres=gpu:1` (ogni nodo di computazione ha un'unica GPU a disposizione e deve essere attivata per poterla utilizzare). Le altre possono essere personalizzate. Per la definizione di queste e altre direttive si rimanda alla documentazione di SLURM (<https://slurm.schedmd.com/sbatch.html>). Nell'esempio, dopo le direttive è stato invocato il programma. Il processo deve essere accodato dalla macchina *giano.cs.unibo.it* (accessibile via *ssh*) lanciando il comando **`sbatch nomescrpt`** (ad es. **`sbatch script.sbatch`**). Con le direttive specificate nell'esempio saranno inviate e-mail all'indirizzo specificato alla partenza del job, al termine dello stesso e in caso di errori. I risultati dell'elaborazione saranno presenti nel file *nomeoutput* come indicato nella direttiva.

L'esecuzione nelle macchine avviene all'interno dello stesso path relativo che, essendo condiviso, viene visto dalle macchine di laboratorio, dalla macchina *giano* e dai relativi nodi di elaborazione (ad eccezione delle directory */scratch.hpc/* e */public.hpc/* che non sono visibili dalle macchine di laboratorio).