DISI GPU Cluster Technical Instructions for Use
(Updated November 2024)

To use the cluster, the first step is to **enable** your unibo.it **institutional account** to access the departmental systems and the cluster itself. If you are not already enabled, you will receive an email confirming your authorization. With your institutional credentials, you will have access, even remotely, to all the machines in the Ercolani laboratory. In the email you will find the link dedicated to the departmental IT services (https://disi.unibo.it/en/department/technical-and-administrative-services/it-services) and in the *Remote Access* section you will find details on these machines and how to access them. In addition, you can access the ***giano.cs.unibo.it*** *machine in the same way*, from which you can use the cluster scheduler and where you need to set up the job execution environment as the updated versions of Python and any additional required libraries are present.

The maximum **user quota** is currently set to 400 MB. If you need more space, you can create your own directory in ***/scratch.hpc/***, which is not subject to any forced deletion policy (the ***/public/*** *directory* is normally deleted every first Sunday of the month, while the **/public.hpc/** directory *is being* decommissioned). The user home is a shared storage space between the machines, so the execution environment and the files needed for processing present in the *giano.cs.unibo.it* machine from which to start the job that will be executed on the machines equipped with GPUs will also be visible on all the other lab machines*. The /scratch.hpc/* and */public.hpc/* directories are visible exclusively from the *giano.cs.unibo.it* machine.

There are two scheduling queues in the cluster:

- **rtx2080**: with compute nodes (single quad-core CPU, 44 GB RAM) each containing an Nvidia GeForce RTX 2080 Ti graphics card (Turing TU102 GPU with 4352 cores, 11 GB memory) driven with Nvidia drivers v. 535 and CUDA 11.8 compute libraries.
- **l40**: with compute nodes (single octa-core CPU, 64 GB RAM) each containing an Nvidia L40 graphics card (Ada Lovelace AD102GL GPU with 18176 cores, 48 GB memory) driven with Nvidia drivers v. 535 and CUDA 11.8 compute libraries.

One possible **setting for the job** is to create a virtual machine in the *giano* machine environment Python by inserting everything you need inside and using **pip** to install the necessary modules; for example, if you install *pytorch* you will need to use the command **pip3 install torch --no-cache-dir --index- url https://download.pytorch.org/whl/cu118** (ref. https://pytorch.org/).
NB: The pip package manager uses a user-space cache by default, and its quota may run out quickly. It is therefore recommended to always include the **--no-cache-dir** parameter in the module installation command, and to use the **pip3 cache purge** command if you need to delete an existing cache.
The cluster uses a SLURM scheduler (https://slurm.schedmd.com/overview.html) for job distribution. To submit a job, you need to prepare a SLURM script file (e.g. *script.sbatch*) in your workspace, in which you can insert the directives for configuring the job itself. After the directives, you can insert script commands (e.g. BASH). An example of a script is the following:

```
#!/bin/bash
#SBATCH --job-name=jobname
#SBATCH --mail-type=ALL
#SBATCH --mail-user=name.surname@unibo.it
#SBATCH --time=01:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --partition=partitionname
#SBATCH --output=outputname
#SBATCH --gres=gpu:1

. bin/activate  # to activate the virtual environment python

python test.py
```

In the previous example, the directive to be reported unchanged is *--gres=gpu:1* (each computation node has a single GPU available and it must be activated to use it). The others can be customized. For the definition of these and other directives, please refer to the SLURM documentation (https://slurm.schedmd.com/sbatch.html). In the example, after the directives the program was invoked. The process must be queued from the machine *giano.cs.unibo.it* (accessible via *ssh) by launching the* **sbatch *scriptname*** command (e.g. **sbatch script.sbatch**). The directives specified in the example will send emails to the specified address at the start of the job, at the end of the job and in case of errors. The results of the processing will be present in the file *outputname* as indicated in the directive.

Execution on the machines occurs within the same relative path which, being shared, *is* seen by the laboratory machines, the *giano* machine and the related processing nodes (except for the */scratch.hpc/* and */public.hpc/ directories* which are not visible to the laboratory machines).