DISI GPU cluster technical instructions

(updated July 2022)

To use the cluster, the first step is to enable the unibo.it institutional account to access departmental systems and the cluster itself. If you are not already enabled you will receive an e-mail of successful enablement. With the institutional credentials you will have access, also remotely, to all the machines in the Ercolani and Ranzani laboratories. In the e-mail you will find the link dedicated to departmental computer services (https://disi.unibo.it/en/department/technical-and-administrative-services/it-services/) and in the Remote Access section you will find details about those machines. In addition to this, the machine slurm.cs.unibo.it, on which the cluster scheduler is located and on which the job execution environment needs to be set up as updated versions of Python and any additional libraries required are present, can be accessed in the same way.

The maximum user quota is to date set at 400 MB. If more space is needed, it is possible to resort to creating one's own directory in /public.hpc/, which is not subject to any forced deletion policy (the similar directory /public/ is instead usually deleted every first Sunday of the month). The user home is a shared storage space between the machines; therefore, the execution environment and files needed for processing present on the machine slurm.cs.unibo.it from which to then start the job that will run on the GPU-equipped machines will also be visible on all other machines in the labs. In contrast, the /public.hpc/ directory is visible exclusively from the slurm.cs.unibo.it machine.

One possible work setup is to create a Python virtual environment on the slurm machine by putting everything you need inside and using pip to install the necessary modules. Note that in order to use Python 3 it is necessary to invoke it explicitly since on the machines the default is Python 2. In the cluster there are Turing GPUs driven with Nvidia v. 470 drivers and CUDA 11.4 computation libraries, so in case of installing pytorch it will be necessary to use the command pip3 install torch --no-cache-dir --extra-index-url https://download.pytorch.org/whl/cu113 (ref. https://pytorch.org/).

N.B.: The pip package manager uses a cache in user space 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 in case you need to clear an existing cache to use the pip3 cache purge command.

The cluster uses a SLURM scheduler (https://slurm.schedmd.com/overview.html) for job distribution. To submit a job, you must prepare a SLURM script file (e.g., script.sbatch) in your workspace in which to enter directives for configuring the job. After the directives you can insert script commands (e.g., BASH). An example of a script is as follows:

```
#!/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 --output=nomeoutput
```

```
#SBATCH --gres=gpu:1

. bin/activate  # per attivare il virtual environment python

python test.py
```

In the above example, the directive to be carried over unchanged is --gres=gpu:1 (each compute node has only one GPU available and must be enabled to use it). The others can be customized. Refer to the SLURM documentation (https://slurm.schedmd.com/sbatch.html) for definitions of these and other directives. In the example, the program was invoked after the directives.

The process must be queued from the machine slurm.cs.unibo.it (accessible via ssh) and run the sbatch nomescript command (e.g., sbatch script.sbatch). With the directives specified in the example, e-mails will be sent to the specified address when the job starts, when it finishes, and in case of errors. The results of the processing will be in the nomeoutput file as specified in the directive.

Execution on the machines occurs within the same relative path, which, being shared, is seen by the lab machines, the slurm machine, and the related processing nodes (with the exception of the /public.hpc/ directory, which is not visible from the lab machines).

We also point out this Github repository (https://github.com/dezingaro/slurm-job-template-disi) with some resources for using the cluster.