Cluster cheat sheet

Connection to saphir (gate to connect to the lab network)

ssh prenom.nom@saphir2.lis-lab.fr

Connection to sms, the head node of the cluster. Note: there is no X server.

ssh sms-ext.lis-lab.fr

Jobs are lunched from sms. These jobs are run on other servers: working nodes.

You can use your home dir.
There are also 2 shared spaces:

- **data1/data/expes** for working or sharing experiments
- **data1/data/corpus** for storing datasets

When working on a server one should use screen or tmux like tools to avoid problems due to lost connections.
Other interesting commands include **scp, rsync, sshfs**...

Build a conda environment to install locally all the stuff you need.

conda env list
conda activate NAME_OF_YOUR_FAVORITE_ENV
conda create…...

Run an interactive job. Interactive jobs are similar to a ssh connection on one of the working nodes.

srun --partition gpu --ntasks=1 --cpus-per-task=1 --gres=gpu:1 --time=2:00:00 --pty bash
squeue

Specify partition to use CPU or GPU and gres to select the number of GPU
Specify ntasks and cpu per tasks

Some information about the GPU on the machine you are connected to.
nvidia- smi

Run one or many jobs
sbatch YOUR_SCRIPT.sh
squeue

```
#!/bin/bash
#SBATCH --job-name="JOB_NAME"      # Job Name
#SBATCH --partition=gpu                # Name of the Slurm partition used
#SBATCH --gres=gpu:1                   # number of GPU per node
#SBATCH --time=96:00:00                 # time (HH:MM:SS)
#SBATCH --output="output_log.out"       # STDOUT
#SBATCH --error="output_error_log.txt"       # STDERR

# echo "$SLURM_ARRAY_TASK_ID"
# LINE=$(sed -n "$SLURM_ARRAY_TASK_ID"p parameters_combinations/chairs_expes.txt)
# echo $LINE

/PATH_TO_YOUR_FAVORITE/python -u /PATH_TO_YOUR_WONDERFUL/code.py

echo "All Done!"
wait      # Wait for the end of the "child" processes (Steps) before finishing the parent process (Job).
```

A *job* consists in one or more *steps*, each consisting in one or more *tasks* each using one or more *CPU*.

Jobs are typically created with the `sbatch` command, steps are created with the `srun` command, tasks are requested, at the job level with `--ntasks` or `--ntasks-per-node`, or at the step level with `--ntasks`. CPUs are requested **per task** with `--cpus-per-task`. Note that jobs submitted with `sbatch` have one implicit step; the Bash script itself.