Parietal Computational Resources: Using Margaret

Thomas Moreau INRIA Saclay

Parietal Computational Resources

Single Machines: small scale

- `drago/2/4` | `paradigm/paradox/parametric/parabolic` (*CPU*)
- `drago3/5` (*GPU*) (avoid CPU intensive tasks)

Clusters: large scale

- `margaret` (*SLURM*; 32 × 40 CPUs)
- `Jean-Zay` (*SLURM*; huge GPU cluster, more complex)

Storage:

- `$HOME` (*10GB* / user; `/home/parietal/$USER` shared on `drago*`)
- `dragostore/2` (≈*3 x 130TB*; `/storage/store*` on `drago*`)

Virtual Machines:

- `minidrago` (*windows*; connection RDP)
- `Gulliver` (*openStack*; see documentation)

Margaret

SLURM: Intelligent scheduler for a cluster.

 `$ ssh margaret` $\Rightarrow$ on the front node, do not run your computation!

- On `drago`, you call your script that runs directly.

  $\neq$

- On `margaret`, you put the script in a queue and it runs **when the resources are available**.

With extra features: multiple queues, batched submissions, priority, quotas, ...

Launching one script with `srun`

To run your computations:

- `$ srun -c 1 --time 01:00 -p parietal hostname`

  Run cmd `hostmane` on parietal resources with 1 CPU.

  - `-c` : Number of CPUs per task,
    Use `-c 20` to have 20 CPUs.
  - `-p` : partition to request resources from
    Use `-p parietal,default` to prefer `parietal` resources
    Use `-p gpu` to use GPU nodes partition.
  - `--time` : Time before interruption.
    Use `--time 24:00:00` to get a job for 24h.

Lauching interactive session

The important parameter: `--pty` – ask for a pseudo-terminal.

- `$ srun -c 10 --pty bash -i`
  Launch an interactive bash session with 10 CPUs.

- Opening an interactive interpreter `srun -pty -c 10 ipython`:

```
In [1]: !hostname
marg034

In [2]: import joblib

In [3]: joblib.cpu_count()
Out[3]: 10
```

Don't forget to release it!

Some more advance info

Other cmd:

- `sinfo -s` : state of the cluster,
- `squeue -u $USER` : current queued job,
- `sbatch` , `salloc` , `sacct` , ...: `https://slurm.schedmd.com/`

Python user tips:

- Submitting multiple jobs with `submitit` and a `concurrent.futures` API: An example of submit script is present in the parietal-wiki

- If you have installed `conda` on `dragostore` , you can access your installation in `/data/parietal/store` .