VadeVecum for HPC Unitn

ovvero

Read Me before using the cluster

This is a very small guide about launching jobs on HPC@Unitn. It is meant to be *helpful* for the master students who attended *Computational Biophysics*, from now referred as **qcbimbi**.

You are supposed to have a basic knowledge of bash scripting language (i.e. `cd`, `ls`, `mkdir` and so on...) and a decent OS (i.e. not Windows).

For more info about the cluster, see HPC website. For more info about how to use the cluster and the batch scheduler, see section 2.

Important: the backslash `\` at the end of a line in a code block means that the line continues to the next line. Therefore, if you find

```
echo "Ciao, sono un \
pokemon"
```

you have to write in your file:

```
echo "Ciao, sono un pokemon"
```

1. How to connect

In order to connect to the HPC cluster, you have to be connected to the university network:

- you are connected to Unitn-x or similar (so you are in one of the university structures);
- if you log from home, you need to use the university VPN (install the Pulse secure and google how to connect to unitn).

Once one of the aforementioned criteria is fulfilled, then open a shell (Ctrl+Alt+t for Linux user) and type:

```
ssh <your-unitn-username>@hpc.unitn.it
```

and then your password will be requested.

NOW you are connected to the login *computers*. **DO NOT LAUNCH JOBS THERE**: those computers are meant just for the job submission, since they are shared among all connected users and they have to be available. If you want to do something on the fly, using only few cores for few minutes, go interactively (see 2.1.3)

2. How to launch a job

The cluster has a batch scheduler, a software that organises the execution of the jobs coming from different users with different requested resources.

Basic dictionary (probably wrong):

- core: part of a computer that does things
- node: the whole computer (with multiple cores in it) connected with other computers with an *ethernet* cable;
- walltime: maximum time you will be granted; if your job is not ended yet, it will be killed. So, choose it carefully and use the `Benchmark` NAMD provides you for a good estimate.

On HPC@Unitn each node has 20 cores.

We remind you that you should request maximum 20 cores (Herr Professor Doktor said 16, but at most ask for 20 without telling him). As far as I know, you will not be able to run, statistically, those jobs that request more than 10 cores per nodes. Therefore, either you ask for 10 cores only on a single node (the following line will be clear in a while):

```
#PBS -l select=1:ncpus=10:mpiprocs=10:mem=40GB
```

or you ask for 10 cores on two nodes (using then 20 cores in total)

```
#PBS -l select=2:ncpus=10:mpiprocs=10:mem=40GB
```

IMPORTANT: before launching *THE* simulation, just run small simulations (1000 timesteps maximum) asking for different numbers of cores. For example: launch 3 simulations requesting 5, 10 e 20 cores (*ça va sans dire*, set the walltime at 10 mins). Depending on your system, you will have a small gain in performances using 20 cores with respect to 10 cores. If this is your case, use only 10 cores, since it will speed up your time in queue.

2.1 What to do in practice

2.1.1 Create batch file

Go to your working folder, where you have the configuration file of namd, the `.psf` `.pdb` files and so on, and create a `submit_me.pbs` file as described below (the extension pbs is purely formal).

```
#!/bin/bash
#PBS -l select=1:ncpus=10:mpiprocs=10:mem=10GB
#PBS -l walltime=00:10:00
#PBS -q cpuq
#PBS -N bubu
#PBS -o mainagioia_out
#PBS -e mainagioia_err

# This is a comment.
# From the /home in the compute node we move to the
# directory from which we launched the job
# (and where you, hopefully, have your files)

cd $PBS_O_WORKDIR

/apps/NAMD_2.12_Linux-x86_64-ibverbs/charmrun \
 /apps/NAMD_2.12_Linux-x86_64-ibverbs/namd2 ++local +p10 conf.namd >& log.log
```

After the file is ready, you submit the job with:

```
qsub submit_me.pbs
```

Now you can wait, have a cup of coffee and pray.

2.1.2 What to modify

All the lines starting with `#PBS` are **NOT** comments but instructions for the batch scheduler.

```
#PBS -l select=<numberOfNodes>:ncpus=<numberOfCoresPerNode>\
:mpiprocs=<sameNumberOfNcpus>:mem=<GBofRAMnecessary>

#PBS -l walltime=hours:minutes:seconds

#PBS -q <queue to be used>

#PBS -N <name to be visualised using qstat>

#PBS -o/e <file to redirect stdout and stderr>
```

2.1.3 Interactive job

On the login shell type:

```
qsub -I -q cpuq -l select=1:ncpus=2:mpiprocs=2:mem=10GB,walltime=00:30:00
```

and then you can use the same commands you use in the `submit_me.pbs` file.

2.1.4 Check your job

To check the status of your submitted jobs, type in the shell:

```
qstat -u $USER
```

2.2 Further info

For more commands and the use of the batch scheduler:

- go to HPC@Unitn guide ;
- type `man qsub` in your login shell, aka *RTFM*.

Questions

For any *technical* questions, first check online (Google and StackOverflow are your best friends).

If in doubt, feel free to send me an e-mail: `gianfranco.abrusci@unitn.it`

Trivial questions, already discussed during the tutorial lessons, will be answered in a formally non-polite way.