Calculco Quick Reference

by calculco@univ-littoral.fr - version 2021-07-16

« Pôle Calcul »user account

Get your account : fill in the form :
https://www-calculco.univ-littoral.fr/
`login`,`password` (and e-mail) open access to :

- **calculco.univ-littoral.fr :** front end access for computing (compile and *very short* tests)
- **pcsdata.univ-littoral.fr :** direct access of all filesystems (data and result transfer)
- **web :** www private access (monitoring, Matlab downloading ...)
- **services annexes :** (gogs/git) and « Dropbox like's »service pcsbox/Seafile (e-mail et `password`)

Calculco access and settings

Tool :	SSH protocol
Windows :	bitvise (or ssh , Win 10)
OS X / Linux :	ssh (terminal or console)
connexion :	`ssh login@calculco.univ-littoral.fr`

bitvise settings :

- `Server` block of `Login`'s tab
  Host : `calculco.univ-littoral.fr`
  Port : 22
- `Authentication` block of `Login`'s tab
  Username : `yourlogin`

(save your *profile* !)

ssh-keys configuration (OS X / Linux) :

- generate the keys : `ssh-keygen -t ed25519 -o -a 64`
- copy it : `ssh-copy-id -i ~/.ssh/id_ed25519.pub monlogin@calculco.univ-littoral.fr`
- create (or add in) your `.ssh/config` file :

```
Host calculco (your alias 1)
    HostName calculco.univ-littoral.fr
    User monlogin
    IdentityFile ~/.ssh/id_ed25519.pub
Host pcsdata (your alias 2)
    HostName pcsdata.univ-littoral.fr
    User monlogin
    IdentityFile ~/.ssh/id_ed25519.pub
```

- Once, logging to the front end :
  `ssh calculco` (according to alias 1)
- (Or... uploading your project :)
  `rsync -azuv ~/project pcsdata:` (accord. to alias 2)

Filesystems

`$HOME`	Homedirectory : quota (100 GB), shared (by NFS) on all nodes. . Backup every day.
`$DATA`	`/nfs/data/unprotected/labo`. quotaless filesystem (limit is the filesystem's size, different for each laboratory/`labo`). Appropriate for large dataset, large file. No backup. `/nfs/data/protected/labo`. Idem , but smaller (about 10 % of `unprotected`) . Backup every day.
`/scratch`	Local disk in every nodes, suitable for high I/O loads. Shared (by NFS) on the front end (calculco) (`/scratch/orvalXX/labo/login`). Auto cleaning every month.
`/workdir`	Parallel filesystem BeeGFS of 30 To, quotaless, suitable for high I/O loads. Auto cleaning every 6 months.

File transfers

2021 (new) : Please use directly `pcsdata` instead of `calculco` for large datasets.

Commands : Linux, MacOS (,Win10). With the appropriate *ssh-keys* and *.ssh/config* settings :

- `rsync -azuv project pcsdata:`
- `rsync -azuv pcsdata:project/results .`
- `scp -rp smalltest calculco:`
- `scp newfile calculco:smalltest/`

Graphical tools :

- **Windows :** bitvise or Filezilla or ...
- **OS X/Linux :** Filezilla or ...
- **Bitvise settings :**
  Open the *SFTP* tab (settings are already performed (see «Calculco access and settings» frame)
  the best : duplicate your *profile* and switch `calculco` by `pcsdata`
- **Filezilla settings :**
  Host : `sftp://pcsdata.univ-littoral.fr` .
  User : `yourlogin`
  Port : 22 .

Batch scheduler & jobs

Multi-user cluster, concurrent computations, a *Batch Scheduler* is required : OAR

basic commands :

`oarsub -I`	**I**nteractive mode (short test)
`oarsub -S ./script`	launch **S**cript,
`oarsub -C` *JobID*	**C**onnection (console) to the *JobID*
`oardel` *JobID*	delete *JobID*
`oardel` –array *arID*	delete job array
`oarstat -sj` *JobID*	*JobID*status (running, waiting ...)
`oarstat -fj` *JobID*	**f**ull jobs status of *JobID*
`oarstat -u mylogin`	status of my jobs

usefull options :

`-l, --resource <LIST>`	set ressources
`-n, --name=<txt>`	rename the job
`-q, --queue <QUEUE>`	set queue
`-p, --property "<LIST>"`	propertie settings.
`-t, --type <TYPE>`	set kind of job
`--notify <TXT>`	mail or script
`-O <FILE>`	set output file
`-E <FILE>`	set error file

ressources hierarchy & properties (examples)

`-l core=1,walltime=2:00:00`	one core for 2 hours (defaults)
`-l nodes=1/cpu=1/core=8`	8 cores, 1 cpu
`-l nodes=1/cpu=2/core=4`	8 cores on 2 cpu (note the multiplication)

add some properties

`-t token:matlab=1`	reserve a Matlab token
`-t besteffort`	submit besteffort
`-p 'host="orval02"'`	a node selection
`-p 'nodemodel="Dell_R630"'`	node model

list of equipments

Queues : submitting policies (option -q)

- **besteffort :** no constraints but null priority (can be killed by the scheduler at any time), no garanty of success (cf. `-t idempotent`)
- **default :** 3000 hours*core (and 128 core max.), on non-dedicated nodes.
- **long :** max 12 cores on max 12 jobs. No walltime limits.
- **foo-project** dedicated to *foo-project* members, on project dedicated nodes.

Workflow examples

Example 1 upload myproject

```
monPc$rsync -azuv ~/myproject calculco:todo
monPc$ ssh calculco # (with appropriate ~/.ssh/config settings)
calculco$ cd todo
calculco$ oarsub -S ./launch.oar
OAR_JOB_ID=123456
oarstat -fj 123456 # (check eventually the job status)
```

where the script `launch.oar` may be (a minima) :

```
#/bin/bash
#OAR -l /core=4/cpu=2,walltime=10:00:00
#OAR -q besteffort
./myprogram
```

Example 2 connect to the submitted job

```
monPc$ ssh calculco
calculco$ oarsub -C 123456
```

in order to check CPU and memory usage of `myprogram`(for example) :

```
orval02:~/acalculer/$ htop -u monlogin
```

Example 3 interactive session with matlab .

```
monPc$ ssh -X calculco
calculco$ nedit test.m
calculco$ oarsub -I
orval01$ ml matlab (ou module load matlab)
orval01$ matlab -nodesktop -nodisplay < test.m
```

Web monitoring and tools

www-calculco.univ-littoral.fr/monitoring (authentication required)

- **drawgantt :** Gantt diagramm of all running and scheduled jobs.
- **monika :** HPC cluster status, array of all running jobs. **NB :** *properties form* (may help for selecting nodes,cpu etc.)
- **ganglia :** nodes load (cpu, memory, network, I/O etc.)
- **zabbix :** other boards, including license server (matlab tokens ...)

FAQ and support

Q : Where is the documentation ?
R : www-calculco.univ-littoral.fr
Q : I need help ?
R : Check tutorial sur www-calculco.univ-littoral.fr (in french :-(, but all code blocks are universal!) , or send a mail to the userlist calculco-users@univ-littoral.fr.
Finally contact calculco@univ-littoral.fr.
Q : Quick reference for Linux commands ?
R : Yes : tinyurl.com/unixquickref-pdf
Q : Cannot connect to the front-end `calculco` ?
R : Your IP is probably banned, consequently of two many errors of `login` or/and `password`. Ban actions are temporary (few hours or few days in case of recidive) : send us you « real »IP address (e.g. : http://ip4.me/)
Q : I need a lot of memory. What to do ?
R : cf. www-calculco.univ-littoral.fr/utilisation/memoire-recommandations

Modules & software environnement

The debian distribution is deployed on all nodes, with a lot of **packages**. Other software are accessible thanks to the lmod *module system* (by Univ. Texas -TACC-)

commande	**description**
`module help` (ou `ml help`)	list sub-commands
`module avail` (ou `ml av`)	list availale modules
`module spider` (ou `ml spider`)	detailed informations
`module spider Matlab`	detailed infos on Matlab
`module load Matlab` (ou `ml Matlab`)	load Matlab
`module list` (ou `ml`)	list all loaded modules
`module unload Matlab`(ou `ml -Matlab`)	unload Matlab
`module purge` (ou `ml purge`)	unload all modules
tips	
completion works	(⇆ key)
save and restore your environment	
`module save` (ou `ml save`)	save modules
`module save env1`(ou `ml save env1`)	backup in `~/.lmod.d/env1`
`module savelist` (ou `ml savelist`)	list of backups
`module restore env1`	restore defined modules in `~/.lmod.d/env1`
astuce	**grep filter**
`ml av |& grep -i opencv`	...keep `grep` working!

batch OAR script and modules

modules are accessibles in interactive mode. In batch mode, you should previously launch *module* with the following instruction (in your `OAR` script) :

`source /nfs/opt/env/env.sh` (or `. /nfs/opt/env/env.sh`)

OAR script example :

```
#/bin/bash
#OAR -l core=4,walltime=0:30:00
#OAR -q default
#OAR -n myproject
#OAR --notify mail:prenom.nom@univ-littoral.fr
#OAR -O myproject.%jobid%.out
#OAR -E mayproject.%jobid%.err
#OAR -t token:matlab=1
. /nfs/opt/env/env.sh
module load matlab
PROG=testcalcul.m
matlab -nodesktop -nodisplay -nosplash < $PROG > matlab.out
```