

# Calculco Quick Reference

by polecs@univ-littoral.fr - version 2017-05-11

## accès plateforme Calculco

**Prérequis :** avoir ou demander un compte (!)

<https://www-calculco.univ-littoral.fr/>

**Outils :** Protocol ssh

*Windows :* [putty](#)

*OS X / Linux :* [ssh](#) ( terminal ou console )

connexion : `ssh monlogin@calculco.univ-littoral.fr`

### Configuration de putty :

•Session →*Host Name* : `calculco.univ-littoral.fr`

*Port* : 22

*Connection type* : ssh

•Connexion →*Data* : *Auto login username* : `monlogin`

### Configuration de ssh (OS X / Linux) :

•ajouter ou créer votre fichier `.ssh/config`

Host calculco

*HostName* `calculco.univ-littoral.fr`

*User* `monlogin`

*IdentityFile* `/.ssh/votre-clef-privée`

•désormais la connexion se résume à :

`ssh calculco`

## Espaces disques

`$HOME` 25 GB «monté»(NFS) sur tous les nœuds. Répertoire personnel. Code source et résultats (si non volumineux). Espace sauvegardé.

`/nfs/data` 58 To. Espace sans quota, pour données et résultats volumineux. Non sauvegardé.

`/scratch` Espace local à chaque nœuds. utilisation indispensable pour des jobs qui requièrent beaucoup d'accès en lecture et écriture. Accessible (post-raitement) par NFS sur calculco (`/scratch/orvalXX/labo/login`).

## Transfert de fichiers

**Outils :** protocole SFTP.

— **Windows :** [Filezilla](#) ou [WinSCP](#) .

— **OS X/Linux :** [Filezilla](#) ou `rsync - scp` .

— **Configuration de Filezilla :**

*Host* : `sftp://calculco.univ-littoral.fr` .

*User* : `yourlogin`

*Port* : 22 .

## Gestionnaire de ressources & tâches

Plateforme multi-utilisateurs, calculs concurrents, un *batch scheduler* est indispensable : [OAR commandes basiques :](#)

`oarsub -I` mode Interactif  
`oarsub -S ./script` lancement par Script,  
`oarsub -C JobID` se Connecter (console) au *JobID*

`oardel JobID` supprimer le *JobID*  
`oarstat -sj JobID` status du *JobID* (running, waiting ...)

`oarstat -fj JobID` full status du *JobID*  
`oarstat -u` status des jobs (user)

### principales options :

`-l, --resource <LIST>` spécifie les ressources  
`-n, --name=<txt>` donner un nom au job  
`-q, --queue <QUEUE>` choix de la queue  
`-p, --property "<LIST>"` spécifier certaines propr.  
`-t, --type <TYPE>` spécifie le type de job  
`--notify <TXT>` au choix : mail ou script  
`-O <FILE>` nom fich. sortie standard  
`-E <FILE>` nom fich. erreur standard

### hiérarchie des ressources & propriétés (exemples)

`-l core=1,walltime=2:00:00` un cœur pour 2 heures (temps par défaut)  
`-l nodes=1/cpu=1/core=8` 8 cœurs sur 1 cpu  
`-l nodes=1/cpu=2/core=4` 8 cœurs sur 2 cpu (noter la multiplication)

### ajouts de propriétés ou de type :

`-t token:matlab=1` réserve un jeton matlab  
`-t besteffort` réserve en mode besteffort  
`-p 'host="orval02"'` précise le nœud  
`-p 'nodemodel="Dell_R630"'` précise le type de nœud

## Politique de réservation : queues (option -q)

— **besteffort** : aucune contrainte mais priorité la plus basse, donc aucune garantie d'arriver à terme (cf. `-t idempotent`)  
— **default** : 900h de calcul sur au plus 64 cœurs.  
— **long** : au plus 5 cœurs à répartir sur au plus 5 jobs sans limite de durée.

## Exemples de séquence

**E1** transférer monprojet et lancer le calcul  
`monPc$rsync -azuv ~/monprojet calculco:acalculer`  
`monPc$ ssh calculco #` (si vous avez édité votre `~/ssh/config`)  
`calculco$ cd acalculer`  
`calculco$ oarsub -S ./launch.oar`  
`OAR_JOB_ID=123456`  
`oarstat -fj 123456 #` (contrôler éventuellement l'état du job)

où le script `launch.oar` pourra contenir (a minima) :  
`#!/bin/bash`  
`#OAR -l /core=4/cpu=2,walltime=10:00:00`  
`#OAR -p network_address = 'orval02'`  
`#OAR -n montest`  
`#OAR -O /scratch/Labo/Login/masortie.%jobid%.out`  
`#OAR -E /scratch/Labo/Login/masortie.%jobid%.err`

**E2** se connecter au job précédemment lancé .

`monPc$ ssh calculco`  
`calculco$ oarsub -C 123456`  
`orval02:~/acalculer/$ ...blabla`

**E3** session interactive 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`

## Outils-Web de supervision

[www-calculco.univ-littoral.fr/monitoring](http://www-calculco.univ-littoral.fr/monitoring)

(identification obligatoire)

— **drawgantt** : diagramme de Gantt de tous les jobs en cours d'exécution.  
— **monika** : status du cluster, synthèse de l'usage de tous les cœurs, nœud par nœud.  
— **ganglia** : détail des nœuds (usage : cpu, mémoire, réseau etc.)  
— **zabbix** : autres tableaux de bord et usage de jetons logiciels (Matlab).

## Question, support

**Q** : Où trouver la documentation du cluster ?

**R** : [www-calculco.univ-littoral.fr](http://www-calculco.univ-littoral.fr)

**Q** : J'ai besoin d'aide, que faire ?

**R** : tout d'abord, lire les tutoriaux sur [www-calculco.univ-littoral.fr](http://www-calculco.univ-littoral.fr), ou envoyer un mail sur la liste des utilisateurs [calculco-users@univ-littoral.fr](mailto:calculco-users@univ-littoral.fr).

et finalement envoyer un mail aux administrateurs [calculco@univ-littoral.fr](mailto:calculco@univ-littoral.fr).

**Q** : Y a-t-il un guide de référence pour les commandes Linux ?

**R** : Oui : [tinyurl.com/unixquickref-pdf](http://tinyurl.com/unixquickref-pdf)

## Modules & environnement logiciel

L'OS du cluster est une [debian](#) qui dispose de nombreux **packages**, accessibles dans le **path** par défaut du système d'exploitation. D'autres logiciels / librairies sont installés dans des répertoire non standard du système. Inexistants (sous forme de package) ou recompilés (avec d'autres librairies pour des raisons de versions, de performances, etc.). L'utilisateur modifie ses variables d'environnements pour sélectionner les applications de son choix avec le *gestionnaire de modules* **lmod** (projet [Univ. Texas -TACC-](#))

<b>commande</b>	<b>description</b>
<code>module help</code> (ou <code>ml help</code> )	lister les sous commandes
<code>module avail</code> (ou <code>ml av</code> )	lister les modules disponibles
<code>module spider</code> (ou <code>ml spider</code> )	informations supplémentaires
<code>module spider opencv</code> (ou <code>ml spider opencv</code> )	infos suppl. sur opencv
<code>module spider Matlab</code> (ou <code>ml spider Matlab</code> )	infos suppl. sur Matlab
<code>module list</code> (ou <code>ml</code> )	lister les modules chargés dans l'environnement
<code>module load OpenCV</code> (ou <code>ml OpenCV</code> )	charger OpenCV
<code>module unload OpenCV</code> (ou <code>ml -OpenCV</code> )	enlever OpenCV
<code>module purge</code> (ou <code>ml purge</code> )	enlever tous les modules
<b>astuce</b>	
la complétion fonctionne	(touche  )
<b>sauvegarder restaurer son environnement</b>	
<code>module save</code> (ou <code>ml save</code> )	sauvegarde des modules actuellement chargés.
<code>module save env1</code> (ou <code>ml save env1</code> )	sauvegarde des modules dans <code>~/.lmod.d/env1</code>
<code>module savelist</code> (ou <code>ml savelist</code> )	liste les sauvegardes
<code>module restore env1</code> (ou <code>ml savelist</code> )	charge les modules définis dans <code>~/.lmod.d/env1</code>
<b>astuce</b>	<b>filtrage grep</b>
<code>ml av  &amp; grep -i opencv</code>	...rend l'usage du <b>grep</b> !

## module et scripts OAR

Les commandes modules sont accessibles en mode interactif. En mode batch, il **faudrait ajouter** l'instruction suivante dans le script de lancement OAR :

```
source /nfs/opt/env/env.sh
```

### Exemple de script OAR

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