

<https://ccub.u-bourgogne.fr/dnum-ccub/spip.php?article1285>

Exemple de job batch avec dépendance pour copier des données depuis l'archive

- Site Public - Calcul -

Date de mise en ligne : jeudi 12 septembre 2019

Copyright © Site du Centre de Calcul de l'Université de Bourgogne - Tous

droits réservés

Voici un exemple de jobs avec dépendances (qsub -hold_jid) utilisé pour copier des données depuis le /archive vers le /work via la file "transfer" dédiée à cet usage.

Pour rappel : l'espace de stockage /archive n'est pas disponible directement sur les noeuds batch.

Script 1 : copie de données

Dans l'exemple, le nom du fichier est copy_job.ksh :

```
1.      #!/bin/ksh
2.
3.      # copy task from ARG $1 to ARG $2
4.
5.      # transfer queue (/user1, /work and /archive are available)
6.      #$ -q transfer
7.      #$ -pe smp 1
8.
9.      # no mail
10.     #$ -m n
11.
12.     [[ -e "$1" ]] || {
13.         echo "error arg 1 : source data not found : $1"
14.         exit 1
15.     }
16.
17.     [[ -z "$2" ]] && {
18.         echo "error arg 2 : destination empty"
```

```
19.     exit 2
20.
21.   }
22.
23.     echo "Copying data from $1 to $2"
24.
25.     cp -pr "$1" "$2"
26.
27.     # wait 10 seconds
28.     sleep 10
```

[Télécharger](#)

Script 2 : traitement des données

Dans l'exemple, le nom du fichier est compute_job.ksh :

```
1.  #!/bin/ksh
2.
3.  # compute task
4.
5.  # $ -q batch
6.  # $ -pe smp 2
7.
8.  # $ -M antoine.migeon@u-bourgogne.fr
9.
10. [[ -e "$1" ]] || {
11.
```

```
12. echo "error arg 1 : compute data not found : $1"
13. exit 1
14. }
15.
16. echo "compute data located in $1"
17.
18. cd $1
19. #....
```

[Télécharger](#)

Exécution sur un lot de données

Dans cet exemple nous allons soumettre pour chaque jeu de données (nommés data_to_compute_1 data_to_compute_2 data_to_compute_3), un job de traitement des données (nommés compute_1 compute_2 compute_3) qui aura comme dépendance un job de copie (nommé cp_1 cp_2 cp_3).

```
1. for job in 1 2 3
2. do
3.
4. # copy job
5. qsub -N cp_${job} copy_job.ksh $ARCHIVEDIR/data_to_compute_${job} $WORKDIR/project/
6.
7. # compute job will wait for the copy job to be finished
8. qsub -N compute_${job} -hold_jid cp_${job} compute_job.ksh $WORKDIR/project/data_to_compute_${job}
9.
10. done
```

[Télécharger](#)

Le job compute_1 commencera dès que le job cp_1 sera terminé.