

**Un script à envoyer sur le cluster va contenir les instructions sous la forme #SBATCH intructions en debut de ligne du script (fichier.sh)**

**Ensuite on envoie le batch pour execution**

**cluster-head> sbatch fichier.sh**

Quand le job est fini, ou s'exécute, on peut voir la sortie dans le repertoire où on a lancé le job. Un fichier slurm-NNN.out contient la sortie (NNN numéro du job)

**exemple 1:** je veux 50 taches indépendantes, peu m'importe où elles vont s'exécuter

```
#!/bin/bash

#SBATCH -n 50

cd repertoire_ou_se_trouve_mes_affaires

srun mon_executable
```

(recuperation du numéro de tache avec la variable d'environnement SLURM\_PROCID)

**exemple 2:** j'ai un code qui démarre plusieurs taches MPI, chaque tache se divisant en 8 avec openMP

```
#!/bin/bash

#SBATCH --job-name=cosmocmc

# nombre de noeuds (machine)
#SBATCH -N 4

# je veux les noeuds pour moi toute seule, je blinde les machines
#SBATCH --exclusive

# pour chaque tache mpi, elle spawne 8 threads (OpenMP), cosmocmc attention
num_thread dans .ini
#SBATCH -c 8

#liste des noeuds alloues pour le job
echo "On tourne sur: $SLURM_NODELIST"
# nombre total de noeuds
echo "Soit $SLURM_NNODES nodes."
# liste du nombre de CPU pour chaque noeuds
echo "job cpu per node = $SLURM_JOB_CPUS_PER_NODE"

# je ne sais pas combien j en veux, depends du nombre de CPUs sur les noeuds
temp=`echo $SLURM_JOB_CPUS_PER_NODE | sed s/(//g | sed s/)//g | sed
s/\/,\/+\/g | sed s/x/*\/g`
n=$(( $temp ))
n=$(( $n/8 ))
```

```
cd slurm_test/cosmomc_ias_dev/  
source environnement  
cd cosmomc  
  
# la je dis que je veux $n taches MPI, il sait avec -c 8 plus haut, qu'il  
doit dispatcher  
# une tache pour 8 processeurs.  
mpirun -n $n ./cosmomc test_SZs.ini
```

**Exemple 3 :** D'habitude je lance mes calculs en direct dans idl.

```
#!/bin/bash  
  
# je demande 1 noeud  
#SBATCH -N1  
  
#informatif  
echo "On tourne sur: $SLURM_NODELIST"  
  
# idl va prendre autant de CPUs que disponibles  
# il lui faut un fichier de commandes à executer  
# (tout ce que vous taperiez si vous etiez devant)  
# soit le creer avant, soit le creer a la volée  
# dans ce job  
  
cat > fichier_commande_idl.bat <<EOF  
mk_an2star  
EOF  
  
idl fichier_commande_idl.bat  
  
# j'ai fini plus besoin du fichier.bat  
rm fichier_commande_idl.bat
```

Exemple 4: mes calculs doivent s'exécuter sur une machine avec coprocesseurs (K20 ou PHI)

<CODE>

## #!/bin/bash

je voudrais phi ou gpu indifferemment

# SBATCH -p copro

si on veut seulement gpu mettre -p K20

si on veut seulement xeon phi, mettre -p PHI

mon code utilise une fois lancé 8 processeurs (multithreadé)

## SBATCH -c 8

## SBATCH --job-name "test\_copro"

echo "On tourne sur:  SLURMJOBSPERNODE"

slurm ajoute des variables qui cachent les coprocesseurs #####CUDA\_VISIBLE\_DEVICES=NoDevFiles

```
unset CUDAVISIBLEDEVICES unset GPUDEVICEORDINAL
```

```
cd ICL/CapsBasic make clean make ./capsbasic ./intelcaps
```

```
./capsbasic NVIDIA ./intelcaps NVIDIA </code>
```

From:

<https://docinfo.ias.u-psud.fr/> - **Informations, recommandations et conseils du service informatique de l'IAS**

Permanent link:

[https://docinfo.ias.u-psud.fr/doku.php/calcul:cluster:exemples\\_batches?rev=1445437925](https://docinfo.ias.u-psud.fr/doku.php/calcul:cluster:exemples_batches?rev=1445437925) 

Last update: **2015/10/21 16:32**