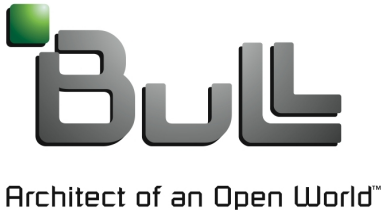


Cyril Mazauric

Cyril.Mazauric@bull.net

BULL - APPLICATIONS & Performances TEAM

Emmanuel Courcelle

emmanuel.courcelle@inp-toulouse.fr

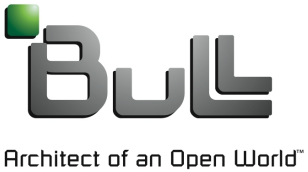
CNRS - CALMIP

29-30/01/2015

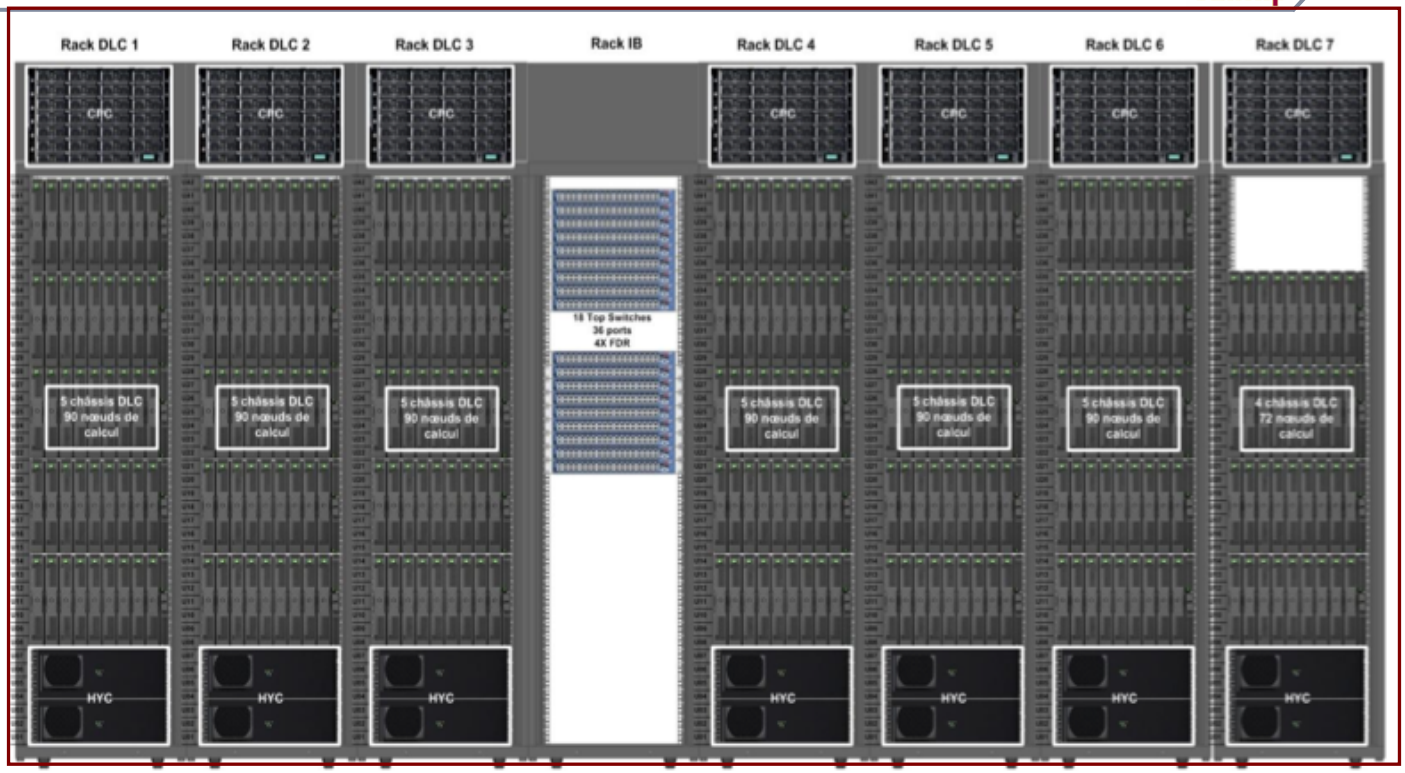
16/02/2016

- Le supercalculateur EOS
- Le batch scheduler : SLURM
- Les modules

Présentation d'EOS



eos : Les nœuds de calcul



Distributed Memory Cluster BULLx DLC

12240 cœurs (612 nœuds, bi-proc)

Processeurs Intel(r) IVYBRIDGE 2,8 Ghz 10-cœurs

64Go de ram / nœud

Interconnection : infiniband FDR

topologie : full fat-tree

eos : Les nœuds de service

4 frontales de connexion

20 cœurs

128GB de RAM

2 nœuds de visualisation

20 cœurs IVYBRIDGE

128GB de RAM

2 cartes graphiques Nvidia Quadro 6000

Virtual GL / Turbo VNC

Printemps 2016

Stockage permanent

NFS

111 Tb

Quota

5 Gb/personne

Stockage temporaire

Lustre

780 Tb

Pas de quota

Ménage

Stockage permanent

gpfs

3 Pb

Quota

1 Tb/projet

eos : Le nœud grande mémoire

1 nœud Grande mémoire

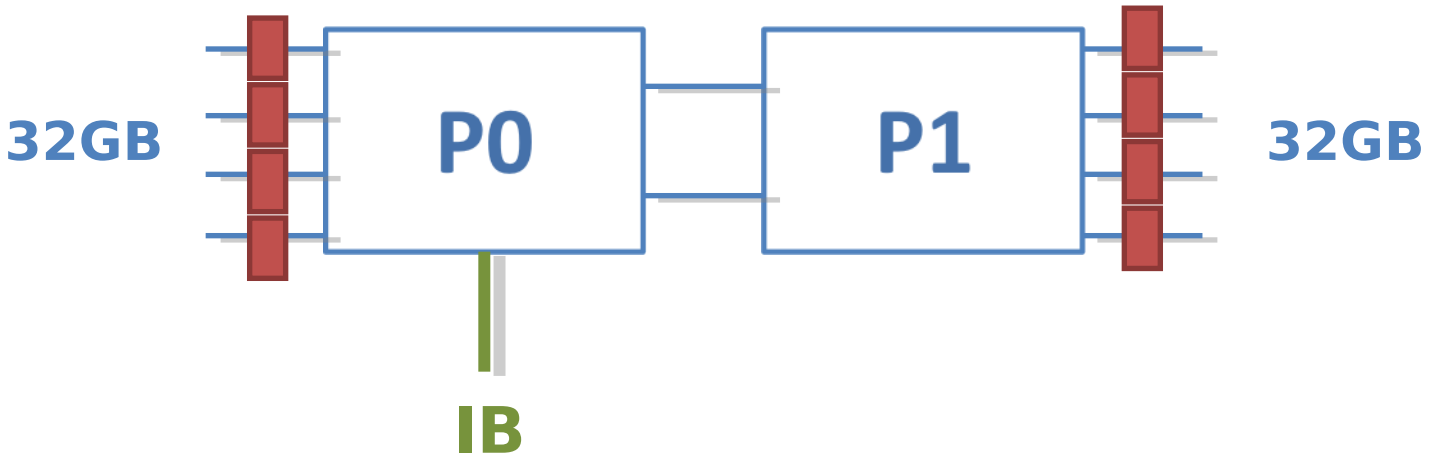
128 cœurs (8 sockets)

2,6 Ghz haswell EX 16-cœurs

2To de ram

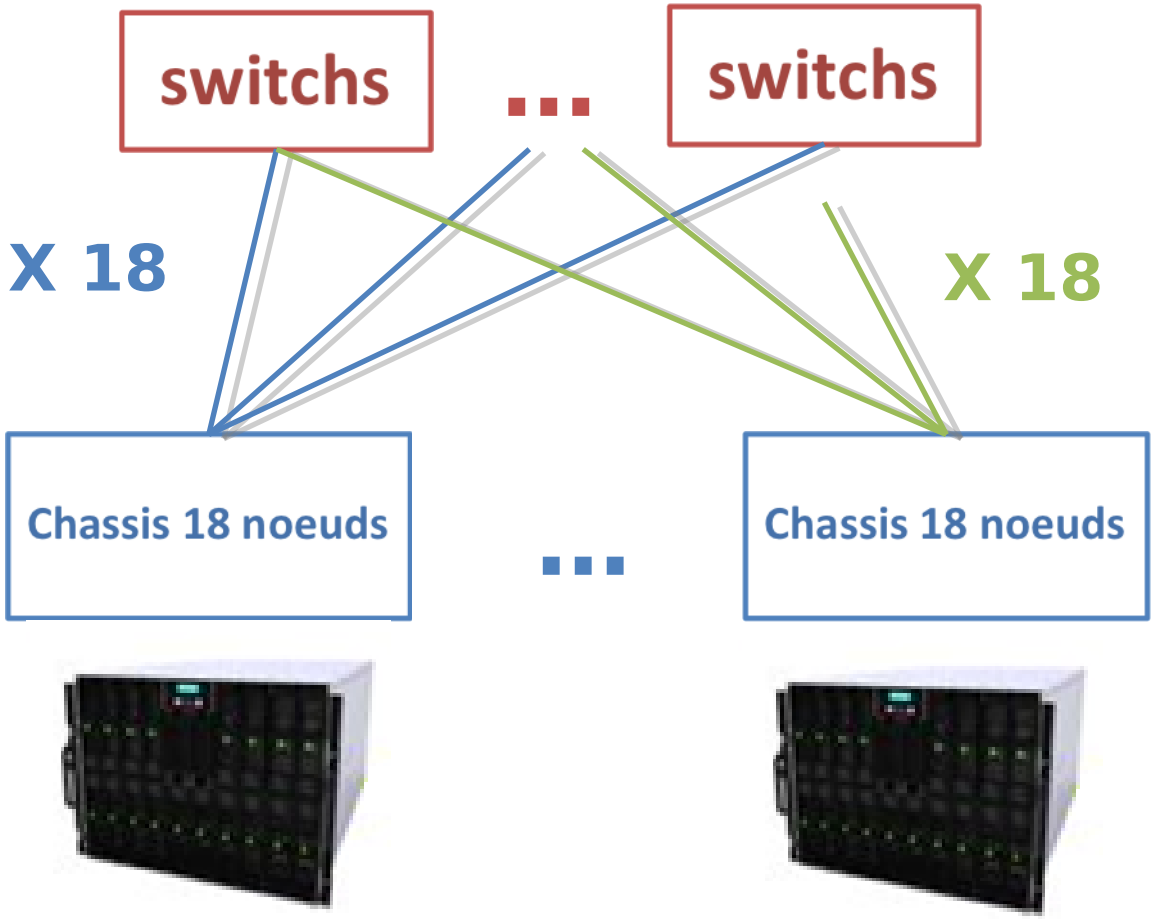
eos : Les caractéristiques

- **612** nœuds IvyBridge bi-sockets :
  - **2 x 10** cœurs par nœuds
  - **12240** cœurs de calculs
- Hyperthreading activé :
  - chaque nœud possède donc 2 x 10 x 2 cœurs
  - l'HT peut être utile pour les jobs hybride MPI/OpenMP
- Les processeurs sont cadencés à **2,8 GHz**
- **64 Go** de mémoire par nœuds (4 x 8 Go à 1866 MHz x 2)
  - bande passante mémoire : 4 x 8 x 2 x 1866MHz = **96 Go/s**



eos : L'interconnection

- Infiniband **FDR**
- Topologie **full fat tree**

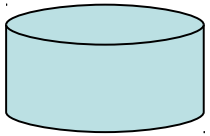


eos : Les fichiers

Espace permanent

- HOME directory
  - **5 Go** par compte
  - **Sauvegardé**

```
cd /users/$GROUPE/$LOGNAME
cd $HOME
cd
```

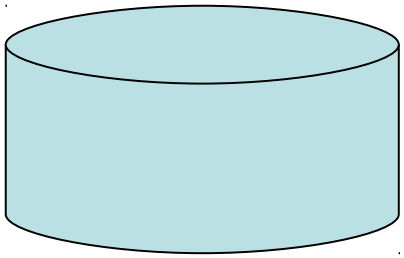


5 Go

Espace temporaire

- Répertoire temporaire
  - **Pas de quota**
  - **Pas de sauvegarde**
  - **Ménage (100 jours max)**

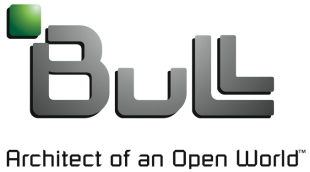
```
cd /tmpdir/$LOGNAME
```



780 To

Chaque espace de fichiers est monté sur tous les nœuds de calcul

L'ordonnanceur slurm (batch scheduler)



slurm : Le gestionnaire de batch

Connexion : sur la frontale

```
ssh -X {username}@eos.calmip.univ-toulouse.fr
```

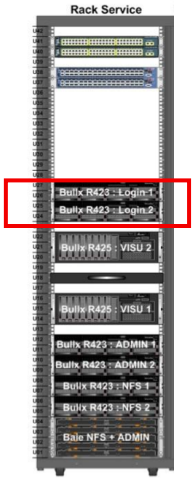
WEB → spip.php?rubrique98

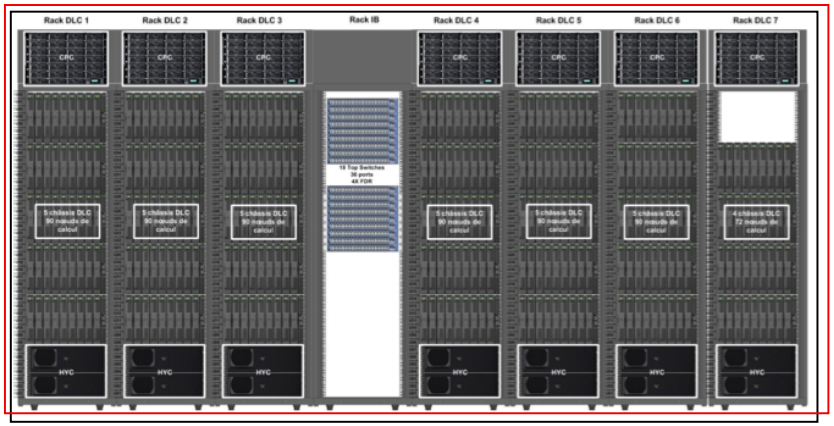
Les calculs sont lancés par slurm :

- En différé
- Sur les nœuds de calcul

```
sbatch mon_job.bash
```

WEB → spip.php?rubrique96





slurm : Le gestionnaire de batch

- **SLURM permet de :**
  - réserver les nœuds pour votre calcul et pour un temps
  - Placer les processus sur les cœurs *(optimisation)*

- **Les nœuds sont séparés en trois partitions :**
  - Partition **exclusive** : nœuds attribuables à usage exclusif
    - 606 nœuds disponibles
  - Partition **shared** : Chaque nœud est partagé entre utilisateurs
    - Seulement 6 nœuds disponibles
  - Partition mesca : la machine à grande mémoire *(partagée aussi)*
    - Seulement 1 nœud disponible

- **Choix automatique de la file d'attente :**
  - Nombre de tâches demandées
  - Mémoire demandée
  - Nombre de nœuds demandés

- **Décomptage des heures :**
  - Partition shared : → **cœurs x tps elapsed**
  - Partition exclusive : → **20 x nœuds x tps elapsed**
  - Partition mesca : → **cœurs x tps elapsed**

slurm : Les files d'attente

- Choix (presque) automatique de la queue
- Limitations en temps et en nombre de jobs simultanés
- Pas plus de **3 jobs** et **90 nœuds** (1 rack)

File d'attente	Nombre de cœurs	Nombre de nœuds	Walltime	Jobs/user	Ram	Partition
mono	< 10	1	400h	3 max	32 Go max	shared
noeud	20	1	300h	3 max	60 Go/nœud	exclusive
noeud9	40 à 180	2 à 9	200h	2 max	60 Go/nœud	exclusive
noeud18	200 à 360	10 à 18	150h	2 max	60 Go/nœud	exclusive
noeud36	380 à 720	19 à 36	100h	1 max	60 Go/nœud	exclusive
noeud72	740 à 1440	37 à 72	48h	1max	60 Go/nœud	exclusive
noeud90	1460 à 1800	73 à 90	36h	1 max	60 Go/nœud	exclusive
visu	20	1	2h	1 max	126 Go max	visu
mesca	1 à 64	1	100h	1 max	1 To max	mesca

WEB → spip.php?article405

slurm : Les commandes

sinfo : y a-t-il des nœuds disponibles ?

```
[manu@eoslogin2 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
exclusiv*    up   infinite      4 drain* eoscomp[33,431,545,557]
exclusiv*    up   infinite      2  down* eoscomp[201,270]
exclusiv*    up   infinite    252  alloc eoscomp[18-32,34-35,53,88-106,
exclusiv*    up   infinite    348   idle eoscomp[0-17,36-52,54-87,107-141,143-152,1
shared       up   infinite      5    mix eoscomp[606-609,611]
shared       up   infinite      1   idle eoscomp610
visu         up   infinite      2   idle eosvisu[1-2]
uvprod       up   infinite      1    mix uvprod
```

slurm : Les commandes

squeue -u <user>: Où en sont mes jobs ?

```
[manu@eoslogin2 ~]$squeue -u  manu
JOBID PARTITION     NAME     USER    STATE       TIME TIMELIMIT        QOS  NODES NODELIST(REASON)
85231 exclusive Ag1mlMgO     manu  RUNNING 5-14:58:36 8-08:00:00     noeud9      4 eoscomp[177,179,288-28
85232 exclusive Ag2mlMgO     manu  RUNNING 5-16:24:18 6-06:00:00    noeud18     10 eoscomp[234-237,241-24
85378 exclusive Agslab35     manu  PENDING       0:00 12-12:00:00     noeud      1 (AssociationResourceLi
85783 exclusive  MLMgHCP     manu  RUNNING    1:44:15 8-08:00:00     noeud9      4 eoscomp[598-601]
85784 exclusive  MLMgTOP     manu  PENDING       0:00 8-08:00:00     noeud9     80 (QOSResourceLimit)
85889 exclusive MLMgO_TO     manu  PENDING       0:00 8-08:00:00     noeud9     80 (QOSResourceLimit)
85892 exclusive MLMgO_HC     manu  PENDING       0:00 8-08:00:00     noeud9     80 (QOSResourceLimit)
85905 exclusive MgO2_FCC     manu  RUNNING 2-15:06:34 6-06:00:00    noeud18     10 eoscomp[216-217,219-22
85908 exclusive MgO2_HCP     manu  PENDING       0:00 8-08:00:00     noeud9     80 (QOSResourceLimit)
85969 exclusive MgO2_HCT     manu  PENDING       0:00 8-08:00:00     noeud9     80 (QOSResourceLimit)
```

Scancel 85378 : Supprimer un job

slurm : Les commandes

Scontrol show jobid=85231 : Tout savoir sur un job

```
[manu@eoslogin2 ~]$ scontrol show jobid=85231
JobId=85231 Name=Ag1mlMgO
   UserId=manu(679) GroupId=p1143(545)
   Priority=11838 Account=p1143 QOS=noeud9
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=0 Restarts=0 BatchFlag=1 ExitCode=0:0
   RunTime=5-15:09:41 TimeLimit=8-08:00:00 TimeMin=N/A
   SubmitTime=2015-03-03T17:35:44 EligibleTime=2015-03-03T17:35:44
   StartTime=2015-03-03T19:22:27 EndTime=2015-03-12T03:22:27
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=exclusive AllocNode:Sid=eoslogin2:2970
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=eoscomp[177,179,288-289]
   BatchHost=eoscomp177
   NumNodes=4 NumCPUs=80 CPUs/Task=1 ReqS:C:T=*:*:*
   MinCPUsNode=20 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) Gres=(null) Reservation=(null)
   Shared=0 Contiguous=0 Licenses=(null) Network=(null)
   Command=(null)
   WorkDir=/eos2/p1143/manu/ALDO/Ag_1MLMgO
```

slurm : Script de soumission

- Pour soumettre un job il faut :
  - Écrire un script de lancement (commandes bash)
  - Appeler la commande sbatch

```
[manu@eoslogin1 ~]$ sbatch script_slurm
Submitted batch job 235209
```

Se connecter à eos

Depuis l'ECA :

ssh -X eos

Depuis le campus :

ssh -X eos.calmip.univ-toulouse.fr

slurm : exercices

Créer un script slurm qui utilise la commande srun et le lancer sur deux nœuds (4 tâches par nœud).

```
#!/bin/bash
#SBATCH --nodes=?
#SBATCH --ntasks=?
#SBATCH --ntasks-per-node=?
#SBATCH -J EXO_1

srun hostname
jobinfo $SLURM_JOBID
infoincidentjob
```

slurm : exercices

Créer un script slurm qui utilise la commande srun et le lancer sur deux nœuds (4 tâches par nœud).

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks=8
#SBATCH --ntasks-per-node=4
#SBATCH -J EXO_1

srun hostname
jobinfo $SLURM_JOBID
infoincidentjob
```

Script de soumission : files nœud*

```
#!/bin/bash
#SBATCH --nodes=5
#SBATCH --ntasks=100
#SBATCH --ntasks-per-node=20
#SBATCH --J mon_job
#SBATCH --time=01:00:00
#SBATCH --mail-user=moi@mon_labo.fr
#SBATCH --mail-type=BEGIN,END,FAIL (ou ALL)
```

- **Recommandations :**
  - Même si nodes, ntasks, ntasks-per-node peuvent être redondants, **il vaut mieux tout spécifier**
  - Utilisez -J pour **nommer votre job dans la queue**
  - Essayez d'**évaluer correctement time**, cela peut contribuer à la réduction de l'attente

WEB → spip.php?article413

Script de soumission : file mono

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=5
#SBATCH --ntasks-per-node=5
#SBATCH --ntasks-per-core=1
#SBATCH --mem=10000
#SBATCH --time=01:00:00
#SBATCH --J mon_job
#SBATCH --mail-user=moi@mon_labo.fr
#SBATCH --mail-type=BEGIN,END,FAIL (ou ALL)
```

- **Recommandations :**
  - Spécifiez 1 nœud et < **10 tâches**
  - Spécifiez la mémoire demandée : **< 32 Go**
  - Essayez d'**évaluer correctement time**

WEB → spip.php?article422

Script de soumission : file mesca

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --qos=mesca
#SBATCH --mem=80000
#SBATCH --time=01:00:00
#SBATCH --J mon_job
#SBATCH --mail-user=moi@mon_labo.fr
#SBATCH --mail-type=BEGIN,END,FAIL (ou ALL)
```

- **Recommandations :**
  - Spécifiez explicitement la queue : **--qos=mesca**
  - Travaux multithreadés de préférence (~~MPI~~/**openMP**)
  - Spécifiez la mémoire demandée : **< 32 Go**

WEB → spip.php?article490

slurm : Variables d'environnement

Vous pouvez utiliser ces variables dans votre script

VARIABLE	DESCRIPTION
SLURM_NODELIST	Liste des nœuds réservés eoscomp[1-6,9]
SLURM_SUBMIT_DIR	Répertoire de lancement
SLURM_NNODES	Nombre de nœuds réservés
SLURM_NTASKS_PER_NODE	Nombre de tâches MPI par nœud
SLURM_JOBID	ID du job slurm
SLURM_NTASKS	Nombre de tâches MPI

slurm : Lancement de votre programme

```
#!/bin/bash
#SBATCH ... (cf. diapos précédentes)

# environnement (cf. diapos suivantes)
module purge
module load intel intelmpi

# répertoire de travail sur tmpdir
dirname=$SLURM_JOBID
mkdir /tmpdir/$LOGNAME/$dirname
cp mes_inputs  /tmpdir/$LOGNAME/$dirname
cd /tmpdir/$LOGNAME/$dirname

# lancement du programme
srun mon_programme avec_ses_arguments

# mise en sécurité de la sortie
mv mes_outputs $SLURM_SUBMIT_DIR

# fin pour savoir ce qui s'est passé
jobinfo $SLURM_JOBID
infoincidentjob
```

WEB → spip.php?article490

slurm : intelmpi vs Bullx mpi

- srun permet de lancer un job mpi depuis le script de soumission
  - Avec Intelmpi (**recommandé par CALMIP**) :

```
export I_MPI_PMI_LIBRARY=/usr/lib64/libpmi.so
srun mon_programme_mpi
```

  - Ou avec Bullxmpi (repose sur openmpi)

```
module load bullxmpi/bullxmpi-1.2.7.1
export OMPI_MCA_ess=^pmi
export OMPI_MCA_pubsub=^pmi
export OMPI_MCA_mpi_leave_pinned=1
srun --resv-ports mon_programme_mpi
```

WEB → spip.php?article413

Les modules

Architect of an Open World™

module : les commandes

- La commande module permet de :
  - charger un environnement
  - le supprimer proprement

COMMANDE	DESCRIPTION
module av	Liste des modules disponibles
module li	Liste des modules déjà chargés
module load mod_name	Charge le module dans votre environnement
module unload mod_name	Supprime le module de votre environnement
module show mod_name	Permet de voir l'environnement
module purge	Supprime tous les modules chargés
module sw mod1 mod2	charge un module à la place d'un autre

module : ce qui existe sur eos

Ce qui est chargé par défaut sur EOS :

```
[bull@eoslogin1 ~]$ module list
Currently Loaded Modulefiles:
  1) oscar-modules/1.0.3  2) intel/14.0.2.144  3) intelmpi/4.1.3.049
```

Ce qui est disponible sur EOS

```
[manu@eoslogin2 ~]$ module available
```

module : Exercice

Changer **intelMPI** *(chargé par défaut)* pour **BullXMPI**

```
module li
which mpirun

module unload ??
Which mpirun

module av

module load ??
which mpirun

Module li
```

module : Exercice

Supprimer tous les modules chargés

```
module li

module purge

module li
```

Ouf !

calmip

Bon appétit

Un cas complet

Architect of an Open World™

Cas complet : Exercice

Lancer un exécutable

```
cp -a /tmpdir/manu/formation/exec* .
cp -a /tmpdir/manu/formation/script_slurm .

less script_slurm
sbatch script_slurm
```

WEB → spip.php?article413

Cas complet : Exercice

Surveiller son exécutable avec top

```
squeue ???
ssh ???
top
    1  → voir/cacher tous les cœurs
    fj → voir/cacher l'utilisation des cœurs
    Fj → trier sur les cœurs utilisés
    H  → voir/cacher les tHreads
    q  → sortir de top
    h  → help !

scancel ???
```

Cas complet : Exercice

Aller au bout de l'exécution

vi script_slurm
ou
nano script_slurm

Remplacer ./exec_10k.x par ./exec_1k.x
(deux lignes)

Sbatch script_slurm

Rechercher :
- **Les deux fichiers de sortie**
- **Le répertoire temporaire**

slurm : srun sans utiliser un script

On peut aussi soumettre un job sans utiliser de script slurm

```
module load env_compilo + env_mpi
srun –N 2 –n 40  mon_programme
```

srun propage votre environnement

slurm : salloc, réservation interactive

- On peut réserver des nœuds de manière interactive

```
salloc –N 1 –n 20  --time=02:00:00
```

- Soit on se connecte au premier nœud réservé :
  - **squeue** puis **ssh eoscompXXX**

- Soit on reste sur la frontale
  - Et on peut utiliser **les variables d'environnement SLURM**