PLACEMENT

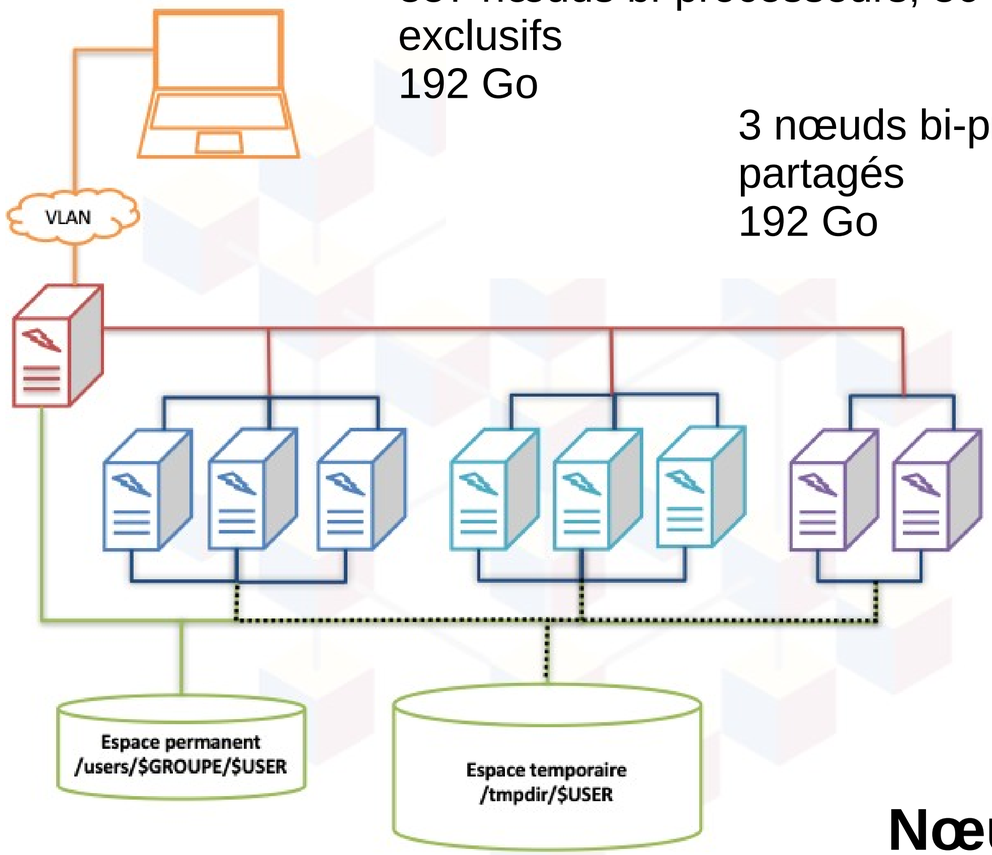
POUR CELLES ET CEUX QUI ONT DES CŒURS

emmanuel.courcelle@toulouse-inp.fr

Café calcul – BBB - Jeudi 21 janvier 2021

CALMIP (UMS 3667)
Espace Clément Ader
www.calmip.univ-toulouse.fr

Olympe, un cluster de calcul



357 nœuds bi-processeurs, 36 cœurs
exclusifs
192 Go

3 nœuds bi-processeurs, 36 cœurs
partagés
192 Go

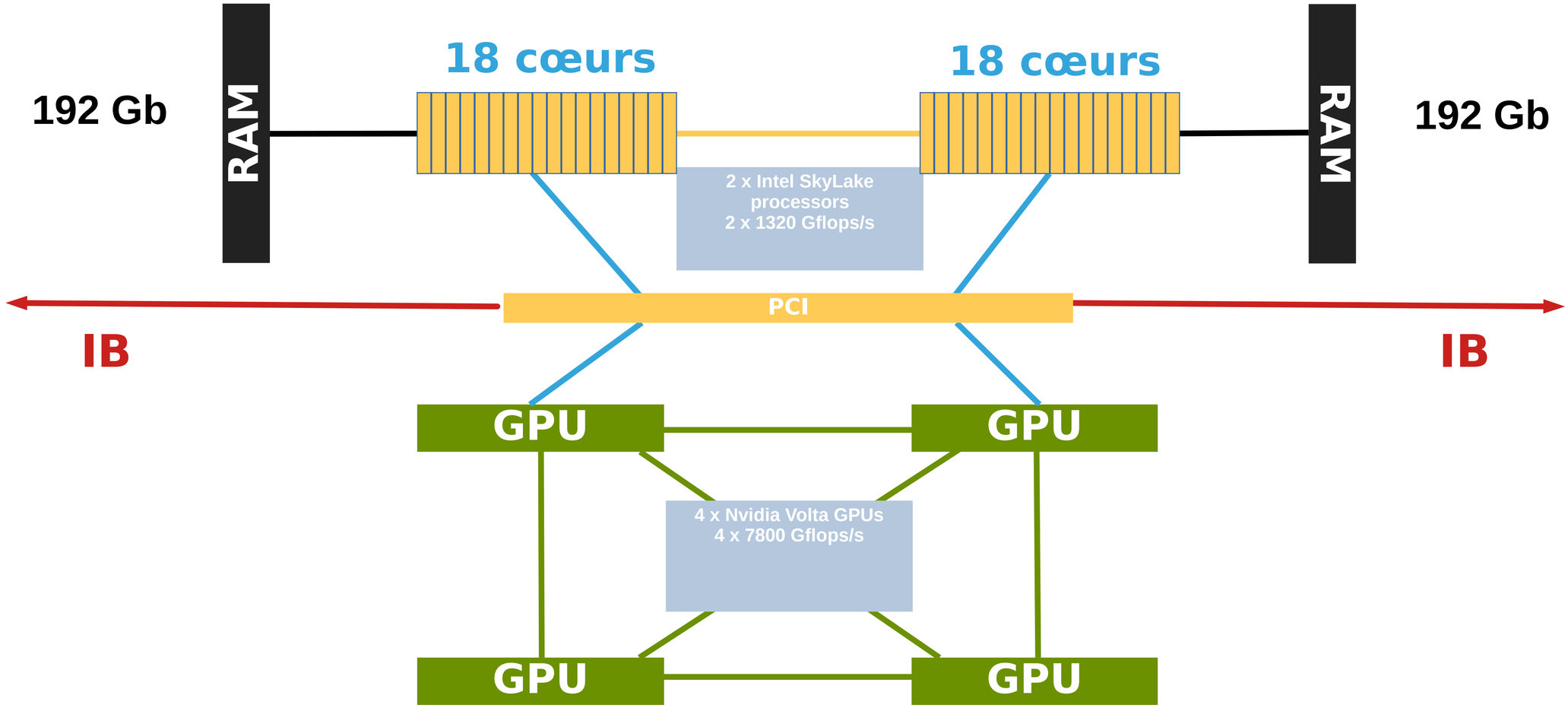
2 nœuds bi-processeurs, 36 cœurs
partagés
1.5 To

11 nœuds bi-processeurs, 36 cœurs
partagés
384 Go
4 GPUs

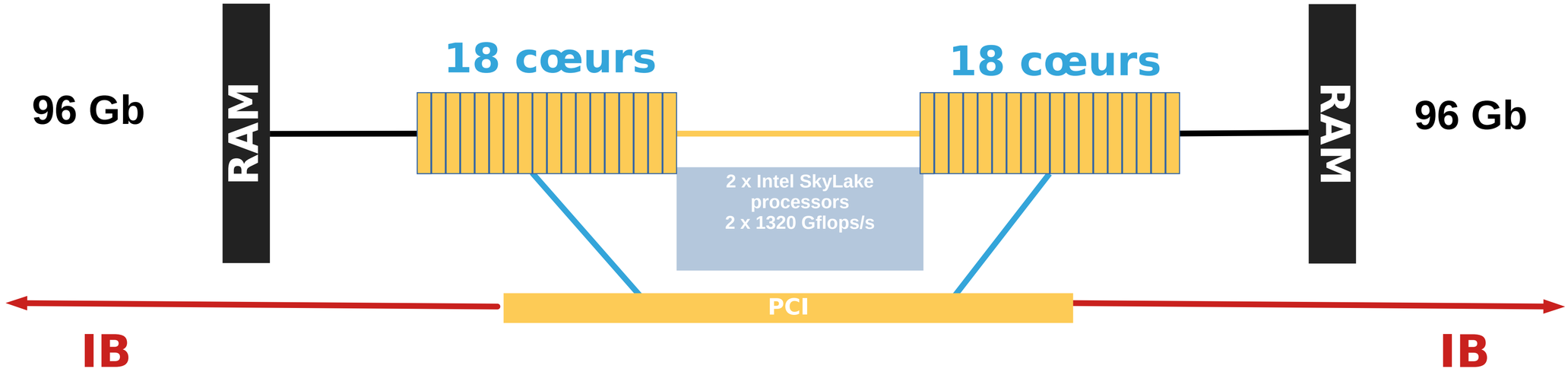
Nœuds homogènes

Plusieurs sortes de nœuds

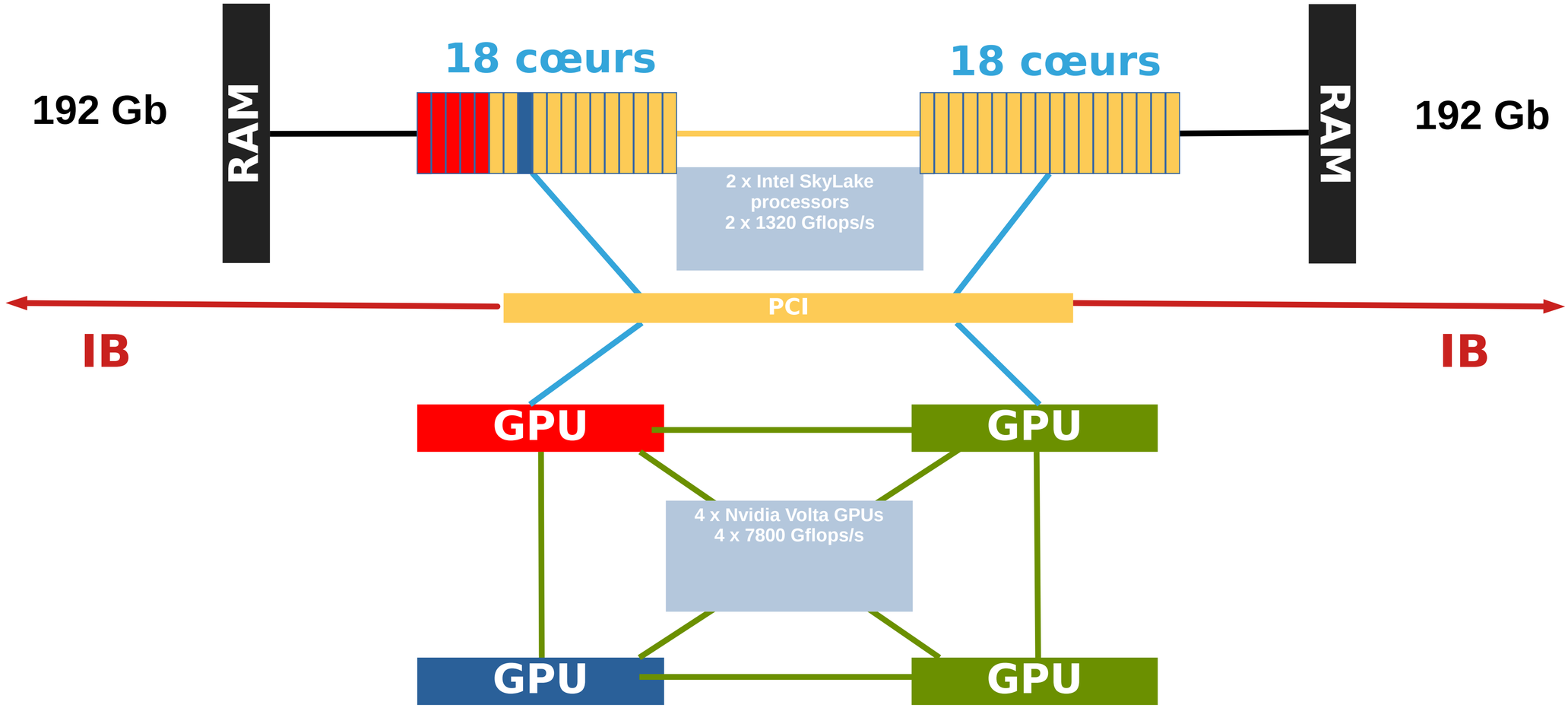
Un nœud avec des gpus...



...ou pas

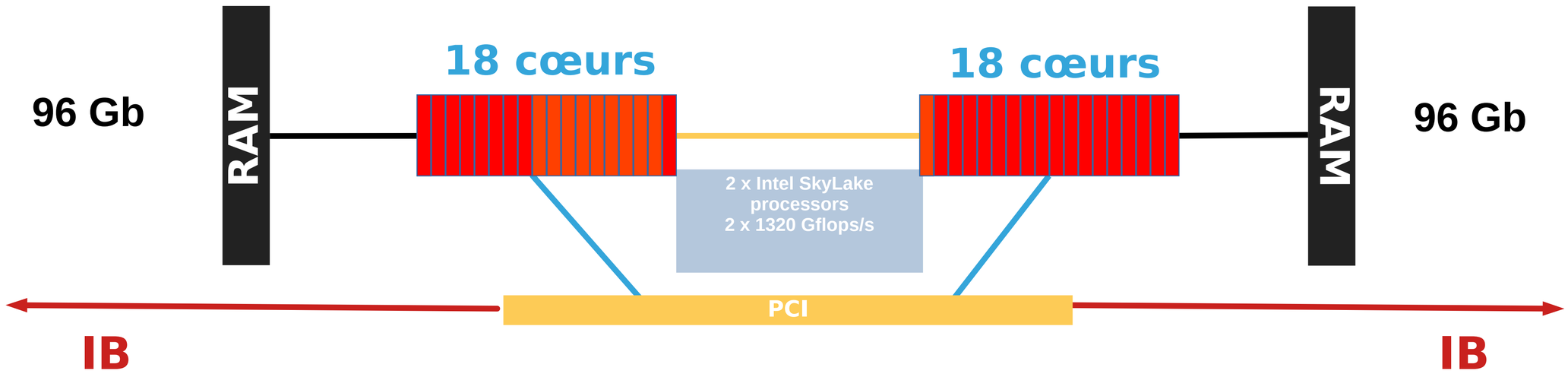

96 Gb
RAM
18 cœurs
18 cœurs
RAM
96 Gb
2 x Intel SkyLake
processors
2 x 1320 Gflops/s
PCI
IB
IB

Des nœuds partagés…



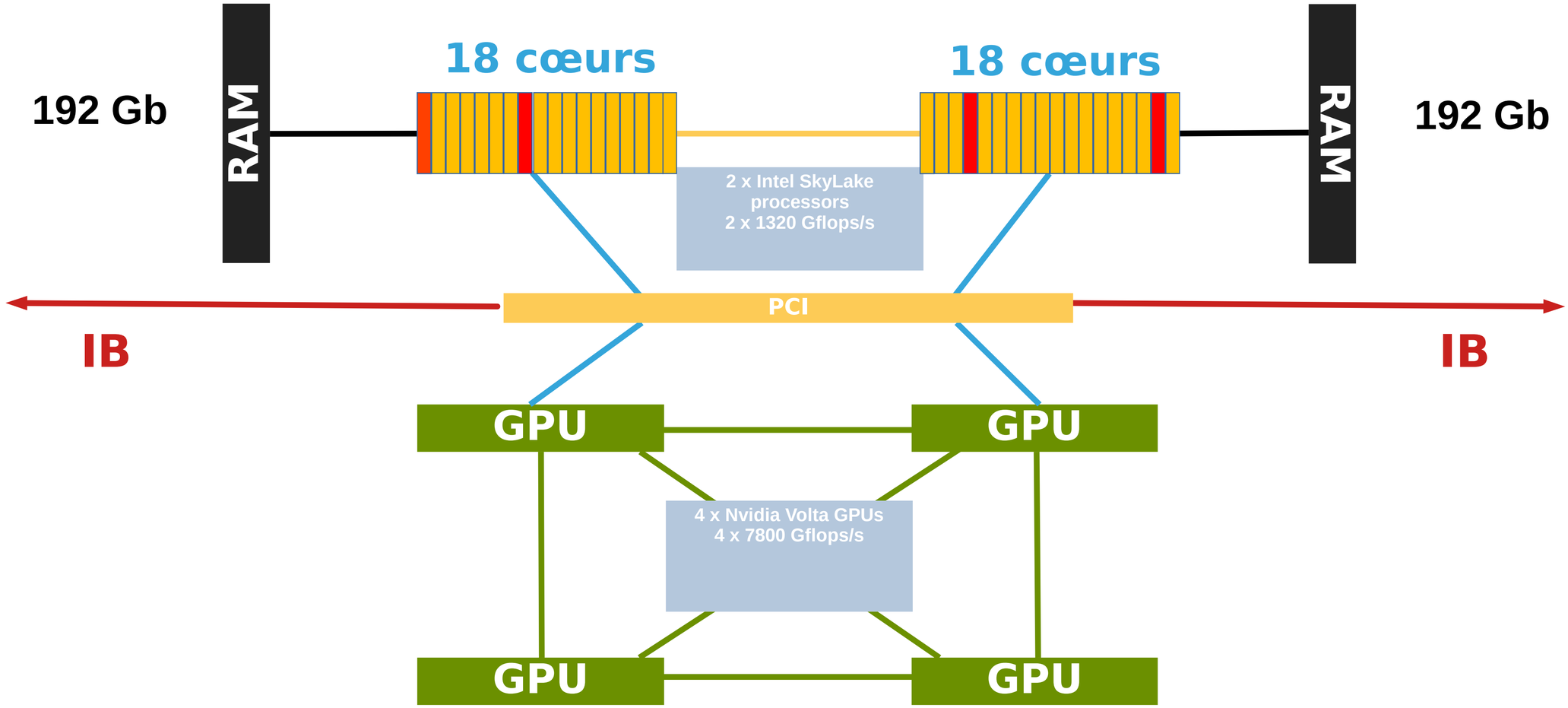
Deux utilisateurs se partagent le nœud

...ou dédiés



Un utilisateur a lancé un job mpi, 36 processus

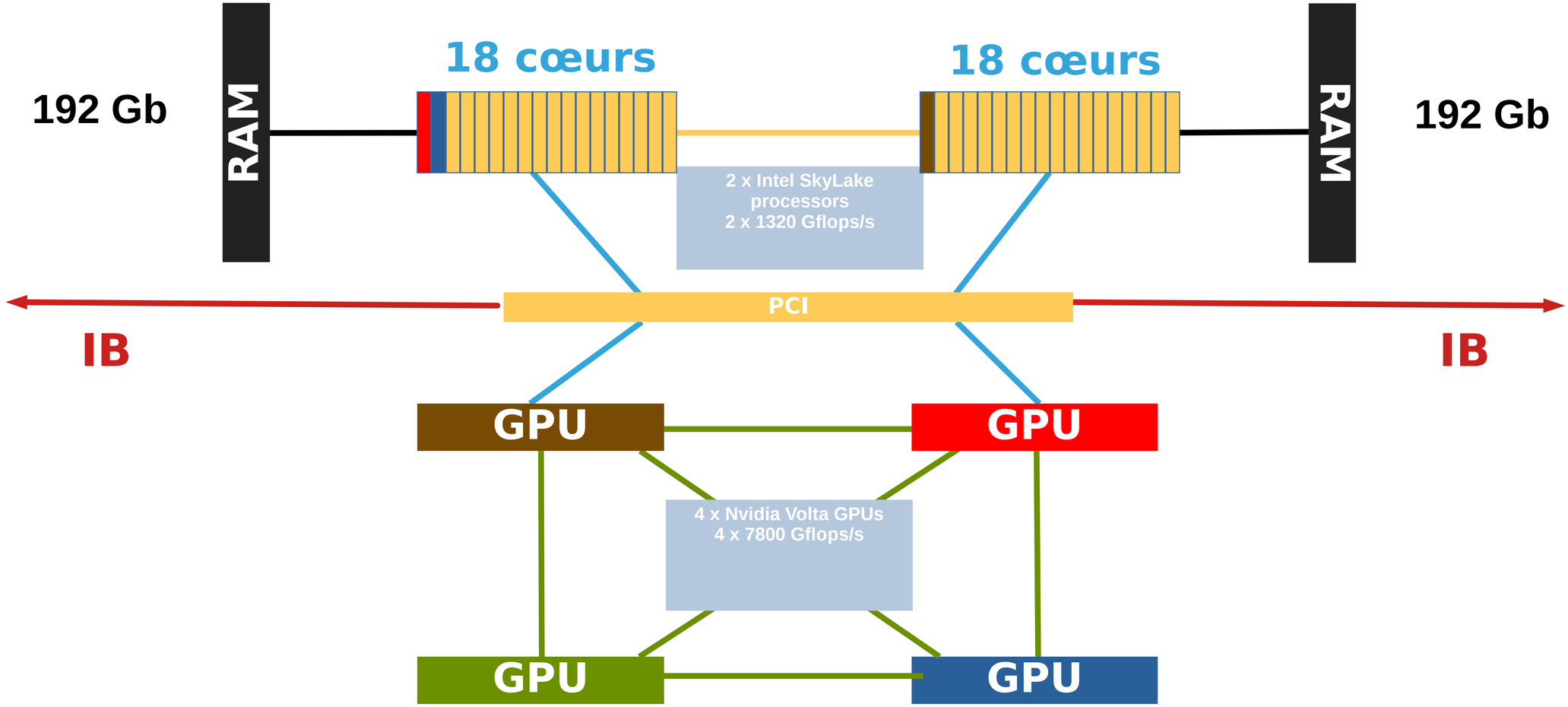
Ça ne se passe pas toujours bien



Placement permet de le détecter

Un utilisateur a réservé 4 GPUS
mais ne s'en sert pas !

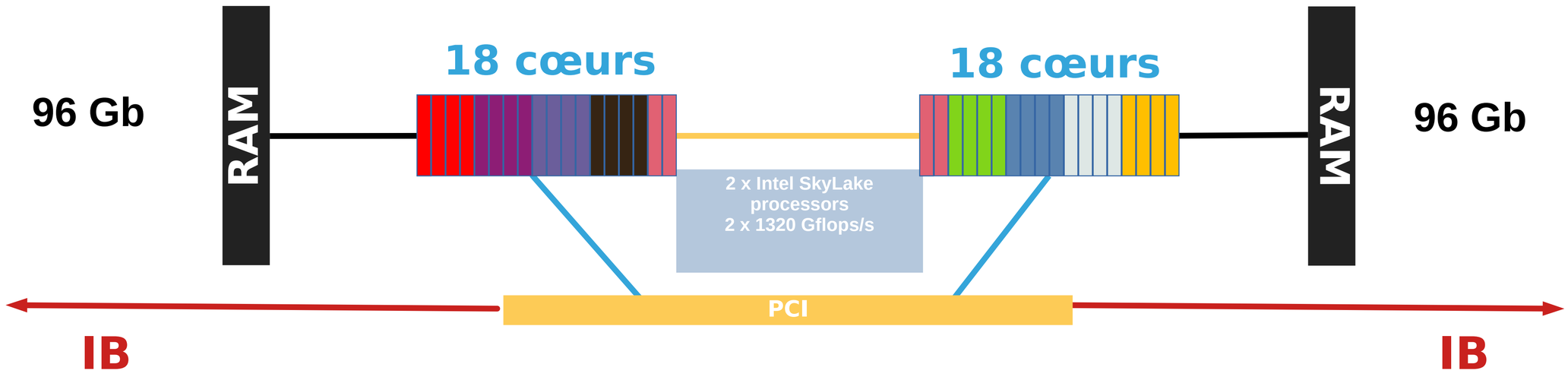
Ça ne se passe pas toujours bien



Placement permet de le détecter

Trois utilisateurs se partagent le nœud
Les transferts de données ne sont pas optimaux

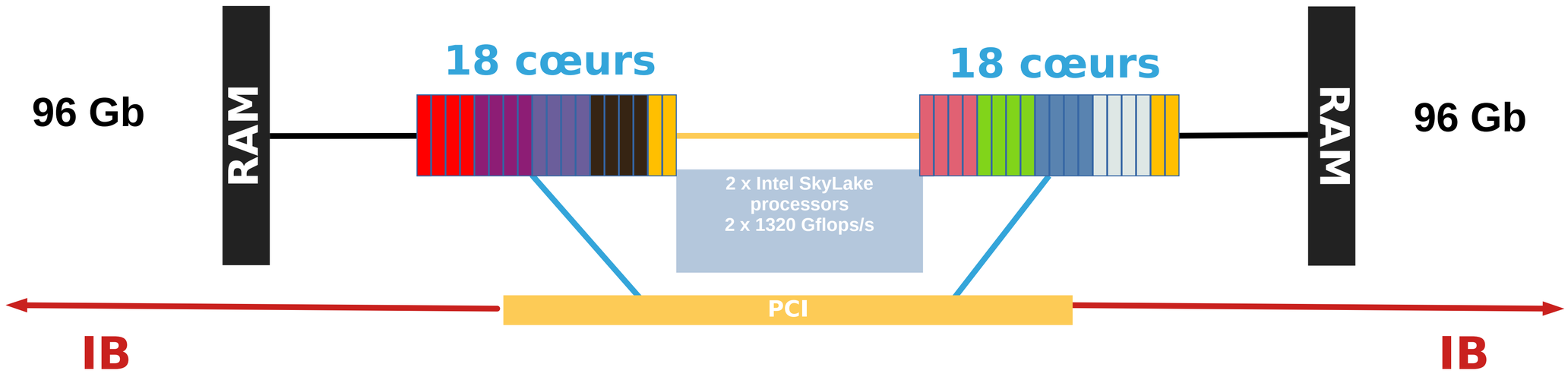
Ça ne se passe pas toujours bien



Le nœud est dédié (mode exclusif)
Un utilisateur a lancé un job hybride en mode dépeuplé
8 processus utilisant 4 threads
Le cinquième est **à cheval sur deux sockets**
Ce n’est pas optimal

Placement permet de l’empêcher
Placement permet de le détecter

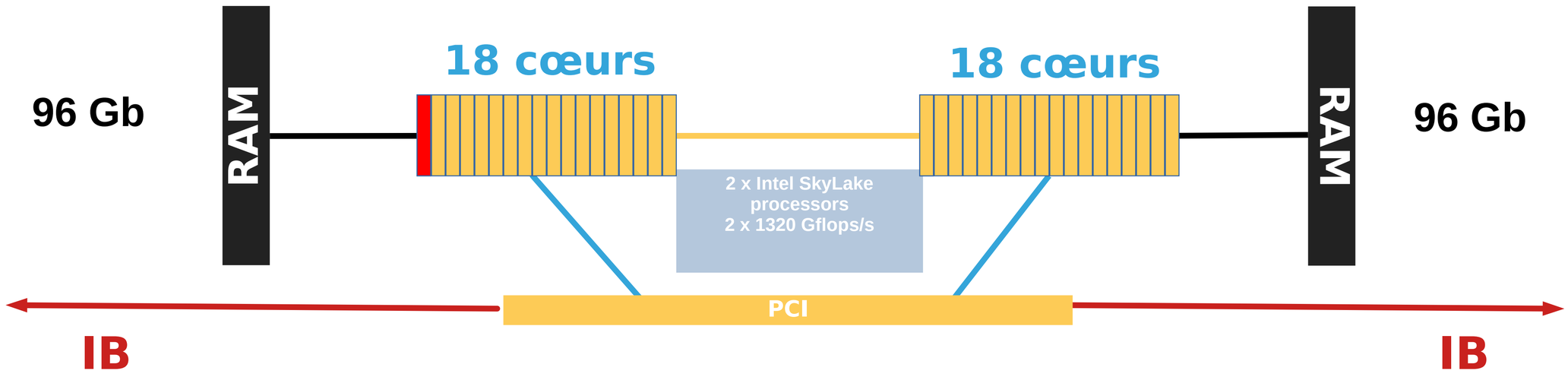
On aimerait bien avoir ça !



Le nœud est dédié (mode exclusif)
Un utilisateur a lancé un job hybride en mode dépeuplé
8 processus utilisant 4 threads
Cette fois les deux sockets sont équilibrés !

Placement permet de le contrôler aisément
Placement permet de le vérifier

Ça se passe parfois très mal



Le nœud est dédié (mode exclusif)
Un utilisateur a lancé un job mpi
36 processus mono-threadés
Ils utilisent tous le cœur **zéro**
La performance est catastrophique

Placement permet de l’empêcher
Placement permet de le détecter

Contrôler le placement...

Quel est mon objectif ?

Utiliser le plus de cœurs possibles avec des processus à 4 threads

```
# placement --ascii  9 4
PLACEMENT_ERROR_FOUND
PLACEMENT ERROR - One task is straddling two sockets !
Please lower the number of tasks/node, max is 8
```

...je peux le forcer… :

```
# placement --ascii --mode=compact  9 4
P AAAABBBBCCCCDDDDEE EEFFFFGGGGHHHHIIII
```

...voire même :

```
# placement --ascii --mode=compact --hyper  9 4
P AABBCCDDEEFFGGHHII ..................
L AABBCCDDEEFFGGHHII ..................
```

Contrôler le placement...

Quel est mon objectif ?

Travailler en dépeuplé, 16 processus/4 threads (64 cœurs utilisés)
Équilibrer la charge des deux sockets :

```
# placement --ascii  16 4
  S0---------------- S1----------------
P AABBCCDDEEFFGGHH.. IIJJKKLLMMNNOOPP..
L AABBCCDDEEFFGGHH.. IIJJKKLLMMNNOOPP..
```

Travailler en dépeuplé, 8 processus/4 threads (32 cœurs utilisés)
Équilibrer la charge des deux sockets :

```
# placement --ascii  8 4
  S0---------------- S1----------------
P AAAABBBBCCCCDDDD.. EEEEFFFFGGGGHHHH..
```

Contrôler le placement...

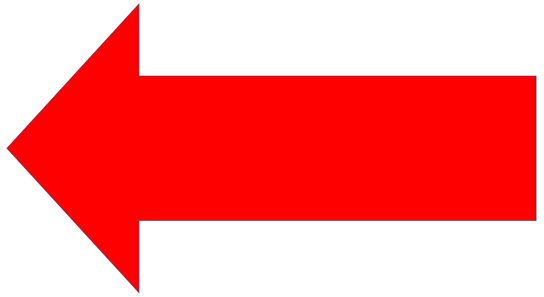
Quel est mon objectif ?

Travailler en dépeuplé, 16 processus/4 threads (64 cœurs utilisés)
Équilibrer la charge des deux sockets :

```
# placement --ascii  16 4
  S0---------------- S1----------------
P AABBCCDDEEFFGGHH.. IIJJKKLLMMNNOOPP..
L AABBCCDDEEFFGGHH.. IIJJKKLLMMNNOOPP..
```

Travailler en dépeuplé, 8 processus/4 threads (32 cœurs utilisés)
Équilibrer la charge des deux sockets :

```
# placement --ascii  8 4
  S0---------------- S1----------------
P AAAABBBBCCCCDDDD.. EEEEFFFFGGGGHHHH..
```



Contrôler le placement… avec srun

Quel est mon lanceur ?

Je travaille avec srun, je place de manière explicite

```
# srun --cpu_bind=mask_cpu:0xf,0xf0,0xf00,0xf000,0x3c0000, \
  0x3c00000,0x3c000000,0x3c0000000 mon_appli.exe
```

...c'est plus cool de l'écrire comme ça :

```
# srun $(placement --srun 8 4) mon_appli.exe
```

Contrôler le placement… avec numactl

Quel est mon lanceur ?

Je travaille avec mpirun, j'utilise numactl :

```
# numactl --physcpubind=0-3,4-7,8-11,12-15,18-21,22-25,\
           26-29,30-33 mpirun -np 8 mon_appli.exe
```

...c'est plus cool de l'écrire comme ça :

```
# numactl $(placement --numactl 8 4) mpirun -np 8 mon_appli.exe
```

Contrôler le placement… avec mpiexec.hydra

Quel est mon lanceur ?

Je travaille avec mpihydra.exe (intelmpi) :

```
# export I_MPI_PIN_DOMAIN=[f,f0,f00,f000,3c0000,3c00000,3c000000,3c0000000]
# mpiexec.hydra -np 8 mon_appli.exe
```

...c'est plus cool de l'écrire comme ça :

```
# eval $(placement --intel_pin_domain 8 4)
# mpiexec.hydra -np 8 mon_appli.exe
```

Vérifier que ça fonctionne !

Lorsque le programme est en train de tourner taper :

```
# placement --host olympevolta9 --memory
# placement --jobid=12345 --memory
# placement --checkme --memory
```

Les processus sont bien déployés

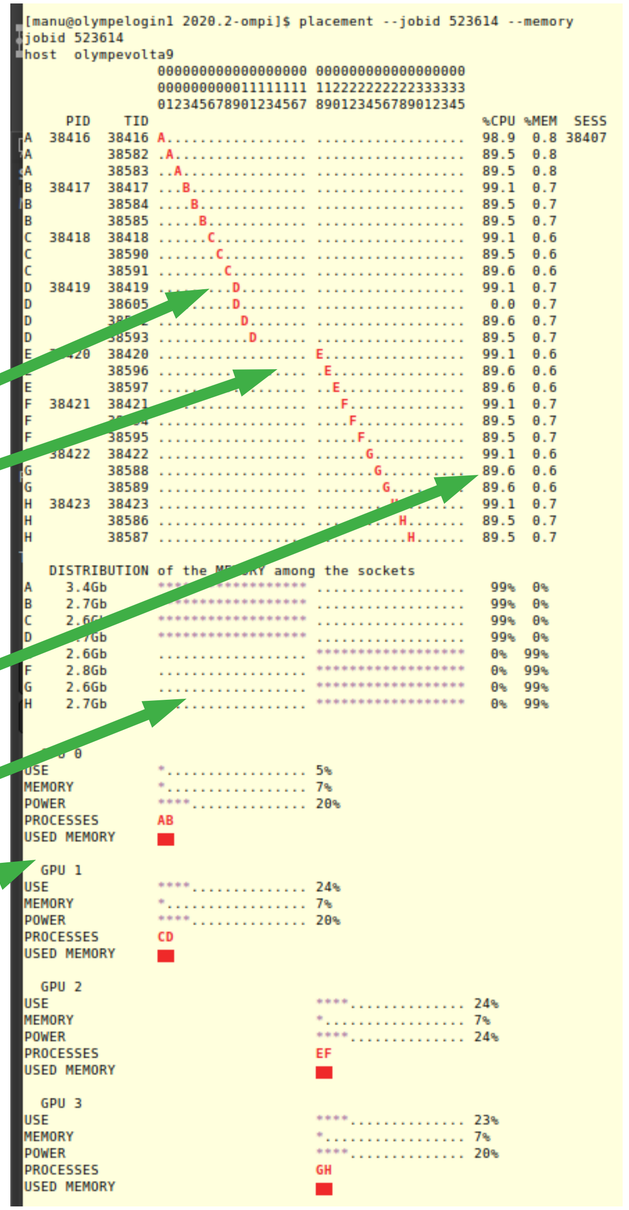
Les sockets sont équilibrés

Tous les cœurs alloués sont utilisés

La mémoire n’est pas saturée

Les bancs de mémoire sont affectés au plus proche

Les GPUs sont utilisés



Vérifier que ça fonctionne !

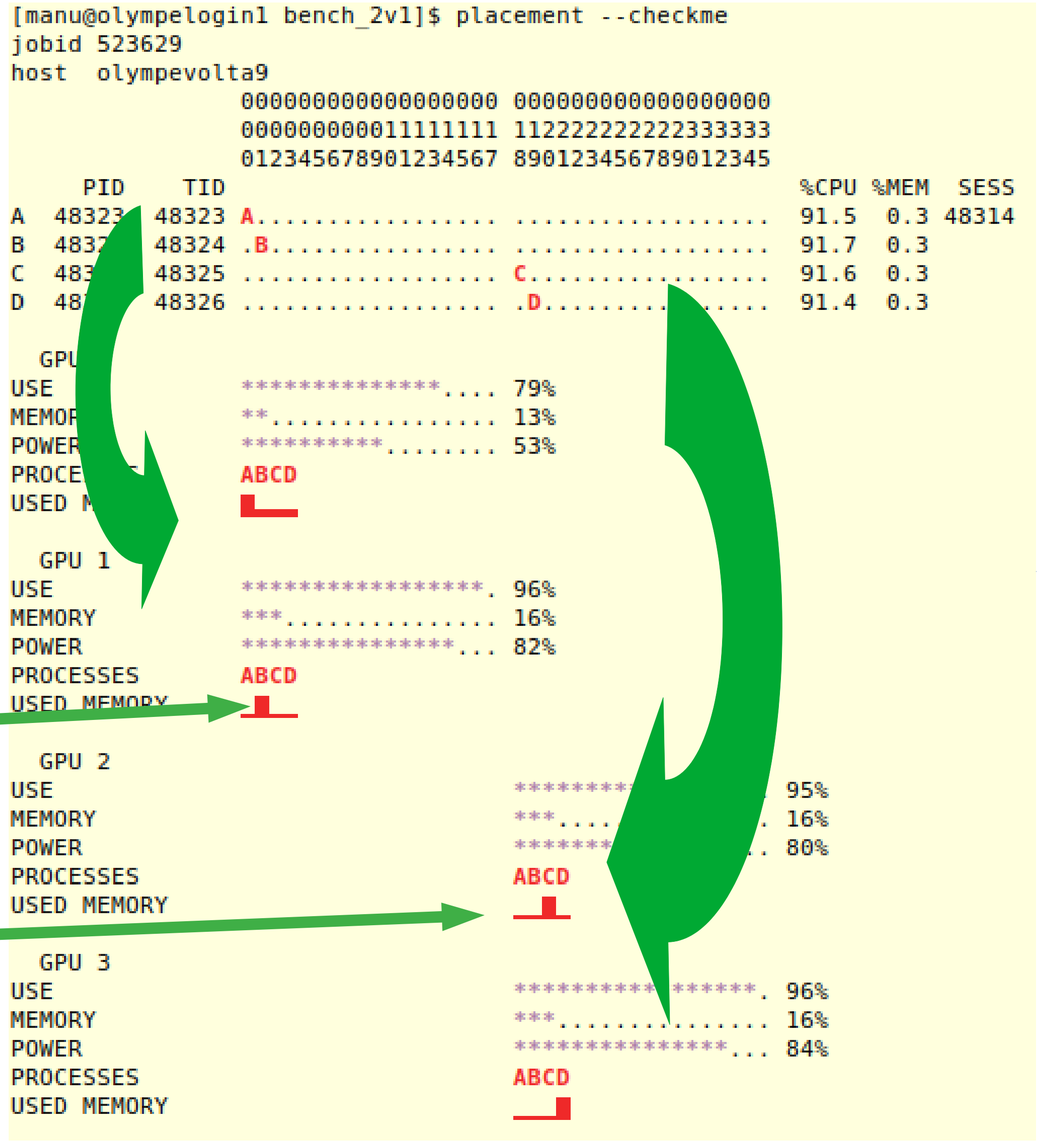
Lorsque le programme est en train de tourner

`# placement --checkme`

Le processus tourne sur le gpu le plus proche de lui

Chaque GPU calcule pour un processus

Chaque GPU « voit » les quatre processus



Installer placement...

Prérequis :

Avoir accès à python 3 (> 3.5.3)

éventuellement via un module

Accéder aux nœuds de calcul en ssh lorsque le job est lancé

Installer placement…

Pas besoin d’être root

```
./install.sh
```

Configuration – voir après

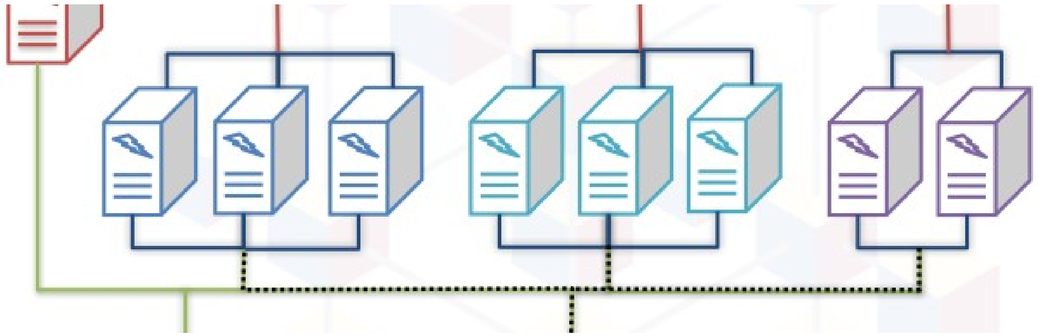
```
cp bin/placement-dist bin/placement
```

et

Éditer **bin/placement** *(appeler la bonne version de python)*

Configurer placement...

Placement.conf



Décrire les trois **types** de nœuds

Correspondance entre **noms** des nœuds et **types** des nœuds

Configurer placement...

1/ Les types de nœuds :

```
[sequana]
SOCKETS_PER_NODE:  2
CORES_PER_SOCKET:  18
HYPERTHREADING:    True
THREADS_PER_CORE:  2
MEM_PER_SOCKET:    98304
IS_SHARED:         False

[sequana_gpu]
SOCKETS_PER_NODE: 2
CORES_PER_SOCKET: 18
HYPERTHREADING:   True
THREADS_PER_CORE: 2
MEM_PER_SOCKET:   192000
IS_SHARED:        True
GPUS:             0-1,2-3
```

Configurer placement...

2/ La correspondance noms – type des nœuds :

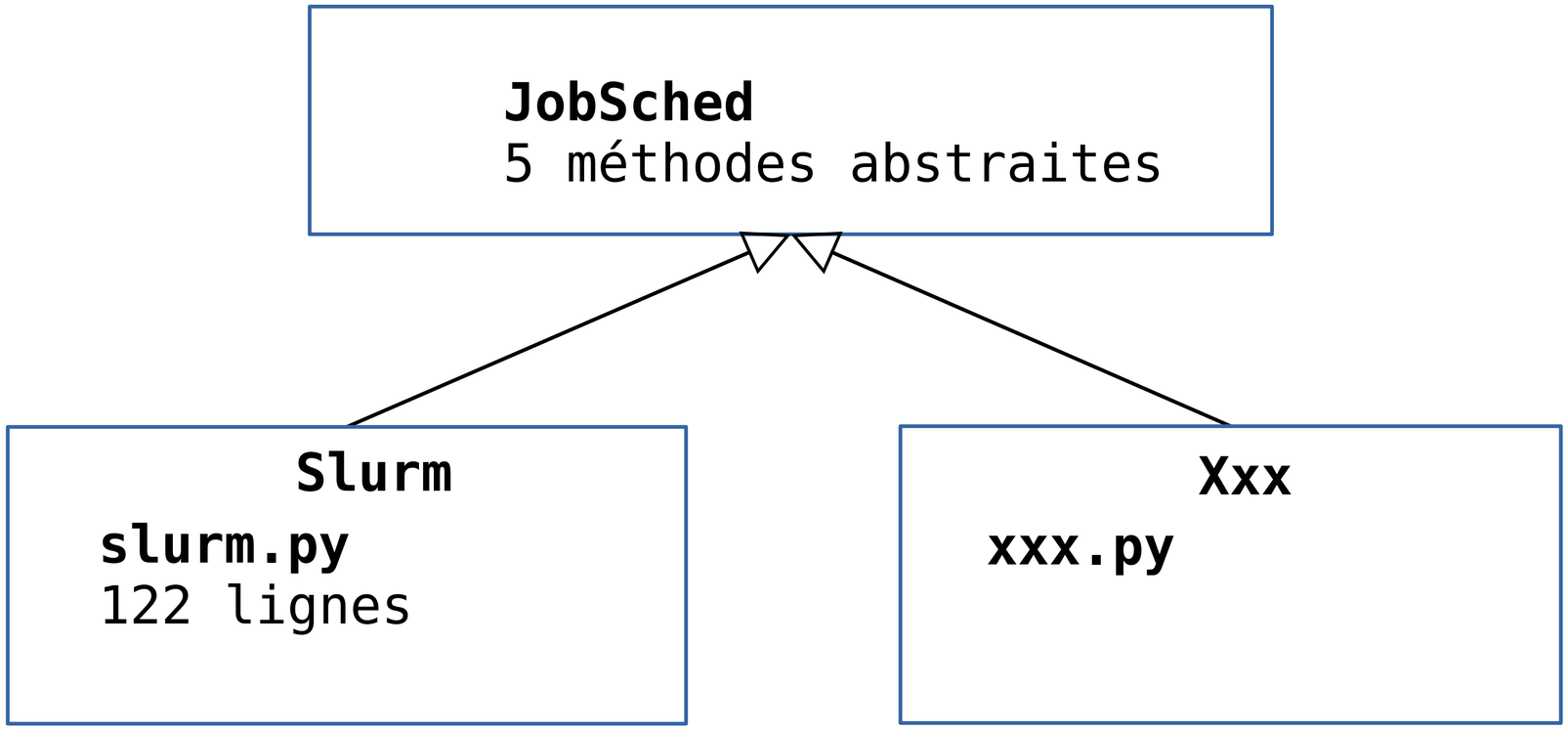
```
# The cluster compute nodes:
olympecomp[0-356]:  sequana



# Compute nodes with gpu
olympevolta[0-11]:  sequana_gpu
```

Adapter placement à un autre gestionnaire

1°)



2°)

Fichier **front.py**
classe **FrontNode :**

Ligne 45 remplacer **Slurm()** par **Xxx()**

Télécharger placement...

https://github.com/calmip/placement/

emmanuel.courcelle@toulouse-inp.fr

...merci pour ce café !