Atelier T7.A1 : Calcul parallèle hybride avec OpenMP, MPI et OpenCL : Session questions/exercices

P.F. Lavallée [1], A. Sartirana [2], G. Grasseau [2]

[1] IDRIS, Orsay
email{lavallee}@idris.fr

[2] Laboratoire Leprince-Ringuet, École polytechnique, Palaiseau
email{sartirana, grasseau}@llr.in2p3.fr

JDEV 2013, Palaiseau, 4-6 septembre 2013

Atelier

Connexion

- Configuration réseau de votre `PC` , voir la note :

  `http://jdev2013.lix.polytechnique.fr/Ateliers/T7/A1/connexion.html`

- Connexion :

  ```
  ssh -X training@polgrid47.in2p3.fr
  passwd:yyyyyyyyx
  ```

- Supports ateliers :

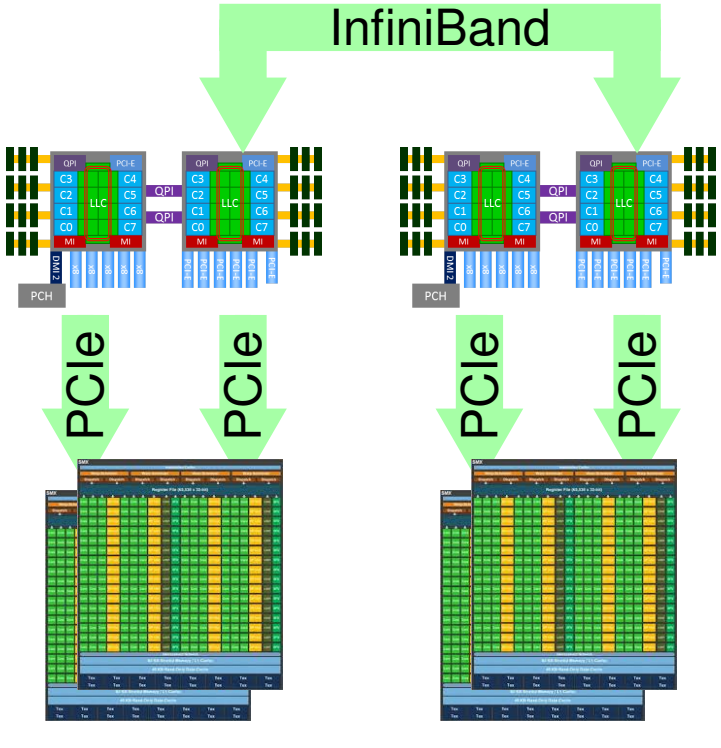
  `http://jdev2013.lix.polytechnique.fr/Ateliers/T7/A1`

Atelier

Configurations

Plateforme `CC-IN2P3`

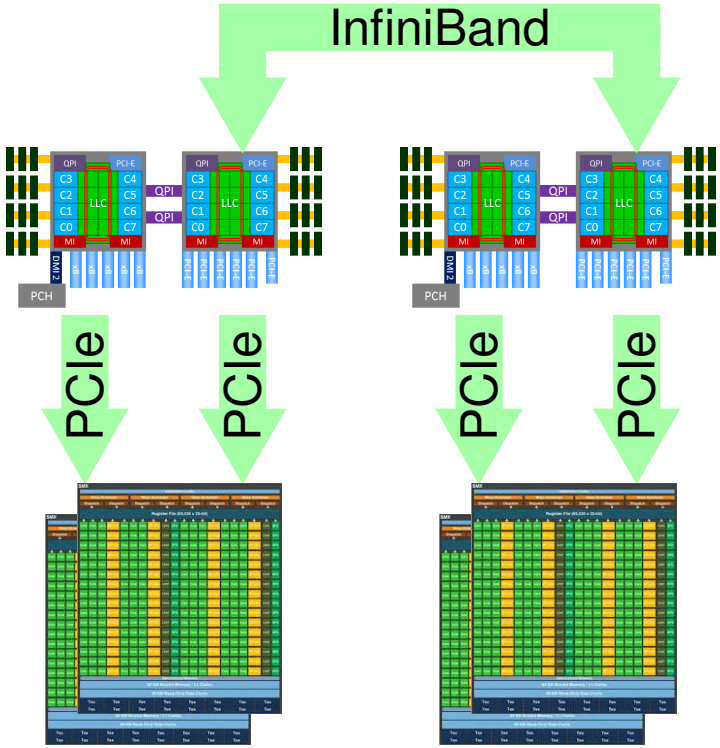
2 noeuds $\times$2 processeurs $\times$8 coeurs



2 $\times$ 2$\times$ `NVidia M2090`

Plateforme `P2IO GridCL`

2 noeuds $\times$2 processeurs $\times$8 coeurs



2 $\times$ 2$\times$ `NVidia K20c`

Configurations :

2 noeuds au `CC-IN2P3`, chacun :

- 2× `Intel E5-2670`
- 2× `NVidia M2090`
- `SL6`
- *Hyper-threading* activé

2 noeuds sur la plateforme `GridCL`, chacun :

- 2× `Intel E5-2670`
- 2× `NVidia K20C`
- `SL6`
- *Hyper-threading* activé

Atelier

Système de *Batch*

Principes :

- Le gestionnaire de *Batch* se trouve sur `polgrid47` qui soumet au 4 noeuds possibles
- A la soumission d'un job :
  - un `tar` du répertoire est effectué,
  - il est recopié (+ `tar -xvf ...`) vers la machine cible
  - le `script` est exécuté,
  - enfin le répertoire de soumission est écrasé lors du retour du job
- Attention :
  - modifications de fichiers perdues lors d'un retour de job
  - le répertoire grossit . . .

Les commandes du système de *Batch* TS

CC-IN2P3

```
# Execution permission
$ chmod +x job.cc.sh

# Submit to CC-Lyon
$ batch_submit -q lyon ./job.cc.sh

# Job status
$ batch_list [ -q lyon ]
...
# Wait the job is 'finished'
...
# Remove from queue
$ batch_end -q lyon 60

# List the job output
$ cat ./job.cc.sh.lyon.60
```

GridCL

```
# Execution permission
$ chmod +x job.gridcl.sh

# Submit to GridCL
$ batch_submit -q llr ./job.gridcl.sh

# Job status
$ batch_list [ -q llr ]
...
# Wait the job is 'finished'
...
# Remove from queue
$ batch_end -q llr 5

# List the job output
$ cat ./job.gridcl.sh.llr.5
```

Volet `MPI/OpenMP`

Avec le fichier d'*input* : `input_sedov_noio_10000x10000.nml`

- Exécuter le code parallèle dans sa version MPI (HYDRO/F90/MPI/MPI2D) sur 16 coeurs
- Exécuter le code parallèle dans sa version hybride MPI-OpenMP FG (HYDRO/F90/Hybride/MPI_OMPFG) sur 16 coeurs, en faisant varier le nombre de threads OpenMP par processus MPI
- Idem avec la version hybride MPI-OpenMP CG (HYDRO/F90/Hybride/MPI_OMPCG2DSync).
- Quelle version donne les meilleurs résultats ? Pour quel ratio de threads `OpenMP` par processus `MPI` ?

Atelier

Questions (2)

Volet `MPI/OpenMP/OpenCL`

- Optimisation à l'exécution :
  - trouver la configuration optimale
  - exécution sur le même *device*
- Portabilité :
  - déployer les *kernels* sur `CPU`
  - observer une très nette baisse des performances, pourquoi ?
- Écrire (sur le papier) une réduction **portable** (pour les fans d'`OpenCL`) :

Soumission de job :

- `job.cc.sh` → `CC-IN2P3`
- `job.gridcl.sh` → `GridCL`