

# Un exemple de A à Z (V)

## Programmation hybride MPI/OpenMP

Marc Tajchman

CEA Saclay, DEN/DM2S/STMF/LGLS

8 octobre 2012

Quand on compare les versions “OpenMP” et “MPI” du code, on remarque que les instructions “communications entre sous-domaines” et “calcul multi-threads” n’apparaissent pas dans les mêmes parties du code.

C’est la situation idéale: il y a une séparation nette entre les phases de communication et les calculs multi-threads

Il faut vérifier que l’implémentation MPI utilisée est suffisamment compatible

*Les différents niveaux de compatibilité sont :*

- `MPI_THREAD_SINGLE` : *pas de compatibilité*
- `MPI_THREAD_FUNNELED` : *toutes les communications doivent être faites depuis le thread 0*
- `MPI_THREAD_SERIALIZED` : *tous les threads peuvent faire des communications, un seul à un instant donné*
- `MPI_THREAD_MULTIPLE` : *compatibilité totale.*

Ici, le niveau `MPI_THREAD_FUNNELED` suffit.

Pour utiliser un ensemble de machines multi-cœurs, 3 possibilités :

- tout MPI (i.e. lancer 1 processus MPI par cœur)
- hybride MPI + OpenMP (i.e. 1 processus MPI par nœud, 1 thread OpenMP par cœur)
- hybride MPI + MPI/OpenMP (par exemple, 1 processus MPI par socket qui lance 1 thread OpenMP sur les cœurs contrôlé par le socket)

Le but théorique est d'utiliser toutes les ressources de calcul (i.e. tous les cœurs de tous les nœuds de calcul). En pratique, la situation optimale est souvent atteinte pour un ensemble incomplet de cœurs.

Avantages (théoriques) de la programmation hybride par rapport au tout MPI (pour une même configuration matérielle) :

- moins de processus MPI  $\Rightarrow$  opérations collectives plus rapides
- moins de messages MPI et de taille plus grande
- moins de duplication mémoire
  - pas de cellules miroir entre des sous-domaines gérés par des threads dans un même processus
  - pas de buffers MPI pour des calculs parallèles à l'intérieur d'un même nœud)

Inconvénients (théoriques) :

- programmation plus complexe (parallélisme à deux niveaux)
- amélioration non garantie