Un exemple de A à Z (III)

Programmation multi-threads avec OpenMP

Marc Tajchman

CEA Saclay, DEN/DM2S/STMF/LGLS

8 octobre 2012

Quelques règles :

- Choisir si une donnée est privée ou partagée
- Équilibrer le temps de calcul entre les threads
- Avoir les sections parallèles les plus grandes possibles

Exemple (version 1):

```
#pragma omp parallel for shared(a,n)
      for (i = 0; i < n; i++)
            a[i] = 2 * i;

c = a[0] + a[n-1];

#pragma omp parallel for shared(a,b,c,n)
      for (i = 0; i < n; i++)
            b[i] = a[i] + c;
```

Exemple (version 2):

```
#pragma omp parallel shared(a,b,c,n)
{
#pragma omp for
      for (i = 0; i < n; i++)
            a[i] = 2 * i;

#pragma omp single
      c = a[0] + a[n-1];

#pragma omp for
      for (i = 0; i < n; i++)
            b[i] = a[i] + c;
}
```

Préférer la version 2 à la version 1: démarrer et arrêter des threads prend du temps. Il faut que le calcul effectué par chaque thread soit assez important par rapport au temps nécessaire à la gestion des threads.

- Si possible, préférer les variables locales allouées par chaque thread aux variables privées

```
int a;
#pragma omp parallel private (a)
{
   a = 1
   ...
}
```

```
#pragma omp parallel
{
   int a = 1
   ...
}
```

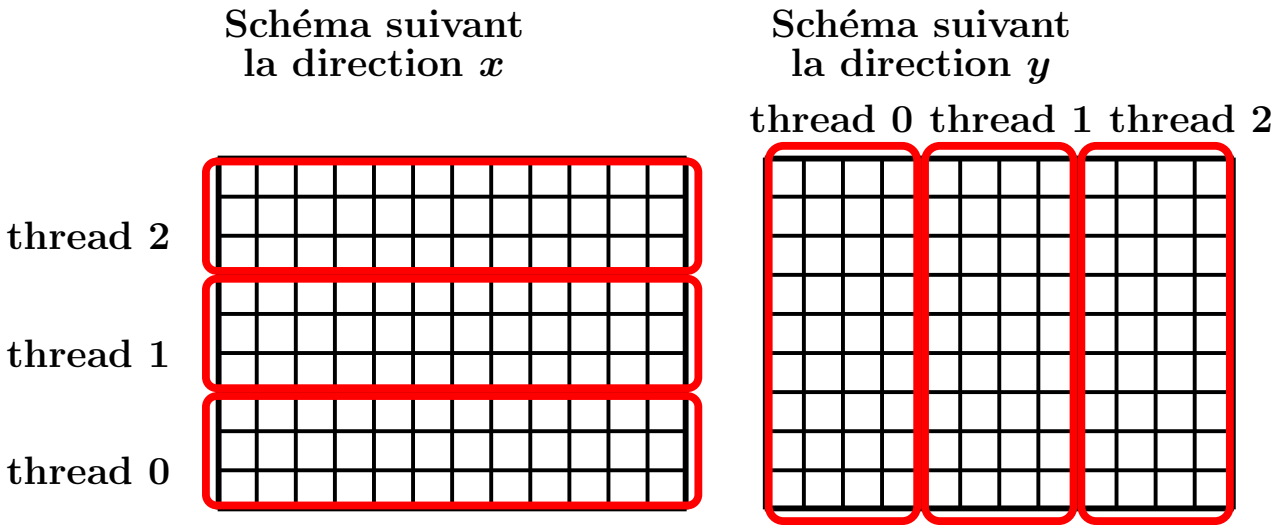
- Faire attention aux accès concurrents aux variables partagées
  En particulier s'il y a des écritures dans des vecteurs partagés ("False sharing" entre threads).

  Exemple : code fourni en annexe

Utilisation d'OpenMP pour construire une version multi-threads du code.

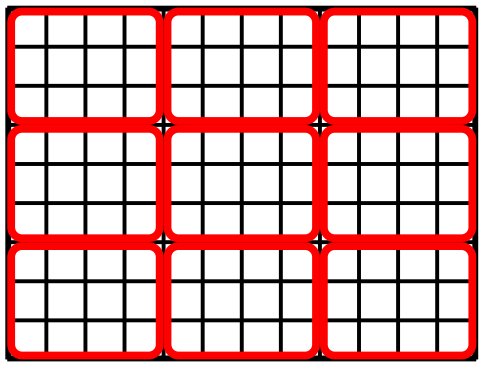
Il existe plusieurs façon de distribuer les calculs parmi les threads.

- Le domaine est découpé en sous-domaines suivant la direction x pour la 1ère étape du schéma AD et suivant y pour la seconde.



- Programmation très simple.
- Ce découpage convient pour un petit nombre de threads, mais est de moins en moins efficace quand on augmente le nombre de threads.

Autre découpage:



- Plus efficace quand le nombre de threads disponibles augmente.
- La programmation est plus compliquée.

  En particulier, la gestion des barrières de synchronisation peut être délicate pour être sur de ne pas utiliser de valeur non encore calculée.