

TP 5. Programmation hybride MPI - OpenMP (2)

Préparation

Récupérer l'archive compressée `TP5.tar.gz` et extraire les fichiers qui sont contenus dans cette archive:

```
cd <repertoire dans votre espace de travail>
tar xvfz TP5.tar.gz
```

Se placer dans le répertoire TP5:

```
cd TP5
```

et préparer les compilations dans les points suivants avec la commande ci-dessous:

```
./build.sh
```

1 Parallélisation des échanges de messages

Il est possible avec les sections OpenMP, d'exécuter plusieurs instructions par des threads différents, par exemple :

```
#pragma omp parallel sections
{
    #pragma omp section
    {
        /* Execute par le thread 1 */
    }
    #pragma omp section
    {
        /* Execute par le thread 2 */
    }
    #pragma omp section
    {
        /* Execute par le thread 3 */
    }
    /* ... */
}
```

Question 1.

On fournit une version du code `src/PoissonMPI_OpenMP_FineGrain` (parallélisé par MPI et par OpenMP - grain fin).

Utiliser des sections OpenMP dans le fichier `value.cxx`, lignes 130 à 227, pour exécuter les échanges MPI dans plusieurs threads différents.

Ne pas oublier de spécifier un niveau suffisant de compatibilité OpenMP-MPI.

2 Recouvrement communications/calculs

Il est possible de faire du recouvrement communications/calculs en MPI seul, on propose de le faire ici avec une programmation hybride MPI-OpenMP.

Question 2.

On fournit une version `src/PoissonMPI` parallélisée par MPI mais prête à accueillir des pragma OpenMP.

Pour chaque itération en temps, séparer le calcul des inconnues en deux parties:

1. une première partie qui ne dépend pas des valeurs transmises par des échanges MPI (au cours de la même itération)
2. une seconde partie qui en dépend

Le but est de calculer les inconnues qui ne dépendent pas des échanges MPI en même temps que ces échanges.

Question 3.

Faire exécuter dans deux threads différents :

- les communications MPI et les calculs qui en dépendent,
- les autres calculs.

Ne pas oublier de spécifier un niveau suffisant de compatibilité OpenMP-MPI.