

# Modèles et techniques en programmation parallèle hybride et multi-cœurs

## Travail pratique 3

Marc Tajchman

CEA - DEN/DM2S/STMF/LMES

mise à jour le 10/01/2021

## Travail pratique 2

On part de deux code qui calculent une solution approchée du problème suivant :

*Chercher  $u: (x, t) \mapsto u(x, t)$ , où  $x \in \Omega = [0, 1]^3$  et  $t \geq 0$ , qui vérifie :*

$$\frac{\partial u}{\partial t} = \Delta u + f(x, t)$$

$$u(x, 0) = g(x) \quad x \in \Omega$$

$$u(x, t) = g(x) \quad x \in \partial\Omega, t > 0$$

*où  $f$  et  $g$  sont des fonctions données.*

Le code utilise des différences finies pour approcher les dérivées partielles et découpe  $\Omega$  en  $n_0 \times n_1 \times n_2$  subdivisions.

# Codes de départ

Le premier code utilise la parallélisation avec MPI et la décomposition de domaines (ce code parallélisée est similaire au code MPI utilisé dans le TP2)

Récupérer et décompresser un des fichiers [TP3\\_MPI.tar.gz](#) ou [TP3\\_MPI.zip](#).

Le second code utilise la parallélisation avec Cuda sur carte graphique.

Récupérer et décompresser un des fichiers [TP3\\_Cuda.tar.gz](#) ou [TP3\\_Cuda.zip](#).

**Le but du TP est d'obtenir une version hybride destinée à être utilisée sur une machine parallèle dont chaque nœud contient une carte graphique.**

On pourra utiliser la machine `rhum.ensta.fr`, qui n'a qu'un seul nœud mais qui contient 2 cartes graphiques.

On comparera les performances du code :

- ▶ avec 1 processus MPI, en utilisant ou pas une carte graphique
- ▶ avec 2 processus MPI, chacun utilisant une des 2 cartes graphiques

Envoyez par mail à [marc.tajchman@cea.fr](mailto:marc.tajchman@cea.fr) :

- ▶ une description du travail réalisé (1-2 pages maximum)
- ▶ le code source, avec vos modifications, dans une archive (n'envoyez pas les répertoires `build` et `install` qui contiennent des binaires)
- ▶ autant que possible, les fichiers qui contiennent les sorties écran.

**avant le 14/02/2021.**

**Envoyez vos fichiers source même s'ils contiennent des erreurs.**