

# Parallélisme

## TP3

Ce travail pratique est évalué, le rendu doit se faire au plus tard le mercredi 21 novembre 2018 sur la plateforme Moodle → cours Parallélisme (13X007) → TP3. Vous devez rendre une archive au format zip portant votre nom et prénom, contenant le Makefile et le code source. Ce travail est individuel.

## 1 L'équation de la chaleur

Dans ce TP, vous allez devoir résoudre l'équation de la chaleur comme cela vous à été présenté dans le TP2. La différence est que vous travaillerez maintenant dans un environnement à mémoire partagée.

## 2 Travail à accomplir

Vous devez implémenter un programme qui résout en parallèle l'équation de la chaleur en C ou C++ sur une machine à mémoire partagée. Comme lors du TP2, vous devez partager le domaine en bandes et attribuer le calcul de chaque bande à un *thread*.

Travailler en mémoire partagée apporte de nombreuses simplifications dans l'implémentation de l'algorithme parallèle. La différence la plus importante est que vous n'avez plus à gérer de communication entre les threads pour calculer les valeurs des sites se trouvant aux bords des sous-domaines.

Pour vous aider, vous pouvez utiliser la barrière de synchronisation fournie sur Moodle (TP3) (fichier `Barrier.hpp`). Une barrière de synchronisation est un point dans le code où les différents threads d'une section parallèle vont s'attendre mutuellement. La barrière fournie est cyclique car elle peut être réutilisée une fois que tous les threads en attente ont été libérés.

## 3 Consignes

- Votre implémentation doit être réalisée en C/C++ à l'aide threads et doit fonctionner avec un nombre arbitraire de threads.
- Vous devez rendre votre code dans une archive au format **zip** portant votre nom et prénom qui doit être structurée de la façon suivante :
  - un répertoire `src` qui contient vos fichiers sources
  - un fichier `Makefile` à la racine de l'archive
- Le fichier `Makefile` doit contenir au moins les deux cibles suivantes :
  - la cible `all` compile votre application dans le répertoire courant et produit un exécutable nommé `laplace_threads`
  - la cible `clean` supprime les fichiers compilés

- Votre application doit prendre quatre arguments à la ligne de commande dans l'ordre suivant : la largeur de la matrice, la hauteur de la matrice, le nombre d'itérations et le nombre de threads. Par exemple la commande suivante

```
$ ./laplace_threads 100 200 5000 4
```

permet de résoudre l'équation de Laplace avec 4 threads sur un domaine de taille 100x200 en appliquant 5000 itérations du schéma numérique

- Le fichier produit par l'application doit se nommer **chaleur.dat**
- L'archive produite doit être déposée sur Moodle dans

Parallélisme → TP3

au plus tard le 21 novembre 2018.