

## TP HPC 4

Master Informatique - 1ère année

Année 2020-2021

L'objectif de ce TP est d'expérimenter une approche de la parallélisation d'une application à l'aide de OpenMP. Il s'appuyera principalement sur des exemples déjà effectués dans les TP précédents.

### Exercice 1

L'objectif est dans un premier temps d'examiner, de manière visuelle, la façon dont OpenMP répartit les calculs. Pour ce faire, le fichier `hpcOpenmpSchedule.cpp` vous est fourni. Celui-ci permet de générer l'image en niveau de gris qui apparaît sur la figure 1.



FIGURE 1 – Vue de l'image initiale

1. Compilez puis testez ce programme, pour vérifier que l'image obtenue est bien correcte. Vous lancerez son exécution par la commande :

```
. /hpcOpenmpSchedule 400 300 toto.pnm
```

où les valeurs 400 et 300 représentent respectivement la largeur et la hauteur de l'image, et le dernier paramètre le nom du fichier de sortie.

2. Parallélisez la boucle externe sur 3 threads et modifiez le calcul effectué de manière à affecter à chaque pixel, un niveau de gris identifiant le thread qui l'a calculé. On précise qu'un niveau de gris est ici nécessairement une valeur comprise entre 0 (noir) et 255 (blanc). Après exécution, vérifiez que l'image obtenue correspond bien à ce qui est attendu.
3. Spécifiez une répartition par segment de 50 pixels et vérifiez à nouveau que le résultat obtenu est correct.
4. Appliquez à présent la parallélisation sur la boucle imbriquée et vérifiez que l'image obtenue est correcte.
5. Examinez le temps de calcul à l'issue de l'exécution des deux parallélisations, en faisant une moyenne sur plusieurs essais. Qu'en concluez-vous ?
6. Echanger le sens de parcours de l'image, en changeant l'ordre des deux boucles. Qu'en concluez-vous ?
7. Appliquez à présent la parallélisation sur les deux boucles en même temps (directive `collapse`), avec des segments de 50 pixels, puis de 32 pixels. Que pensez-vous des résultats obtenus ?

### Exercice 2

On reprend dans cet exercice l'application qui permet de calculer les sommes suivantes (voir TP2) :

$$S_1 = \sum_{i=1}^N i$$

$$S_2 = \sum_{i=1}^N \sqrt{i}$$

$$s_3 = \sum_{i=1}^N \sqrt{i} \times \log i$$

1. Ecrire une application parallèle utilisant OpenMP permettant de réaliser le calcul de ces trois sommes ;
2. Comparez les résultats obtenus par cette application avec ceux obtenus lors du TP2 (sans routines AVX ...). Vous pourrez dresser un tableau de statistiques pour 1 à N threads, N représentant le nombre de coeurs logiques que perçoit votre système d'exploitation.

On précise que pour avoir des mesures relativement fiables, il est nécessaire d'effectuer plusieurs exécutions avec le même nombre de threads et d'effectuer la moyenne des résultats obtenus.

## Exercice 3

Après avoir récupéré les sources du lancer de rayon séquentiel fournies pour le TP3, étudiez différentes implantations parallèles de celui-ci en utilisant OpenMP :

1. parallélisation statique de la boucle externe ;
2. parallélisation statique de la boucle interne ;
3. parallélisation statique des deux boucles ;
4. parallélisations dynamiques

Evaluez différentes solutions et comparez-les à la version multi-threadée dynamique qui a été développée dans le TP2.