

Rédigé par : équipe pédagogique du cours de prog parallèle.

Ref : *ING2-PROGPAR-TP-OPENMP*

A l'intention de : Etudiants d'ING2 MI

Créé le : 21/11/2014

### Exercices de synthèse sur l'utilisation d'OpenMP

#### Exercice 1 : produit de matrices

Le code, contenu dans le fichier `prod_mat.c`, calcule le produit de matrices :

$$C = A \times B$$

Dans cet exercice, il s'agit de :

1. Insérer les directives OpenMP appropriées dans le fichier `prod_mat.c`.
2. Analyser les performances du code sur 2, 4, 6 et 8 tâches par rapport à une exécution séquentielle (utiliser un Makefile). Ne pas hésiter à tester les différents modes (**STATIC**, **DYNAMIC**, **GUIDED**) de répartition des itérations d'une boucle, ainsi qu'à faire varier la taille des paquets.
3. de donner le tableau des accélération obtenues.

#### Exercice 2 : Calcul de $\pi$ par la méthode des rectangles (point milieu)

Il s'agit de calculer  $\pi$  par intégration numérique sachant que :

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

Le fichier `calcul_pi.c` contient le programme permettant de calculer la valeur de  $\pi$  par la méthode des rectangles (point milieu). Soit  $f(x) = \frac{4}{1+x^2}$  la fonction à intégrer et N et  $h = 1/n$  respectivement le nombre de points et le pas de discrétisation de l'intervalle d'intégration  $[0, 1]$ .

Le calcul de  $\pi$  est simple, il ne s'agit que d'une intégrale calculée par la méthode des trapèzes ou des rectangles. Le calcul est réalisé par une boucle sur un grand nombre d'éléments

1. Analyser le statut des variables et insérer les directives OpenMP appropriées dans le fichier `calcul_pi.c`. Cet exercice peut être parallélisé de trois façons différentes (i.e. utilisation de directives OpenMP différentes pour chaque version).
2. Analyser les performances des trois codes sur 2, 4, 6 et 8 threads par rapport à une exécution séquentielle.
3. Donner un Makefile pour lancer la compilation et l'exécution du programme séquentiel, parallèle et avec les différents threads.
4. Tracer les courbes d'accélération obtenues

#### Méthode de travail :

1. Implémentez la méthode séquentielle sans regarder le corrigé
2. Si vous n'arrivez pas, récupérez le corrigé et commencez à placer les directives pour une implémentation parallèle
3. Si vous n'y arrivez pas, le cas échéant le corrigé vous sera proposé

Rédigé par : équipe pédagogique du cours de prog parallèle.

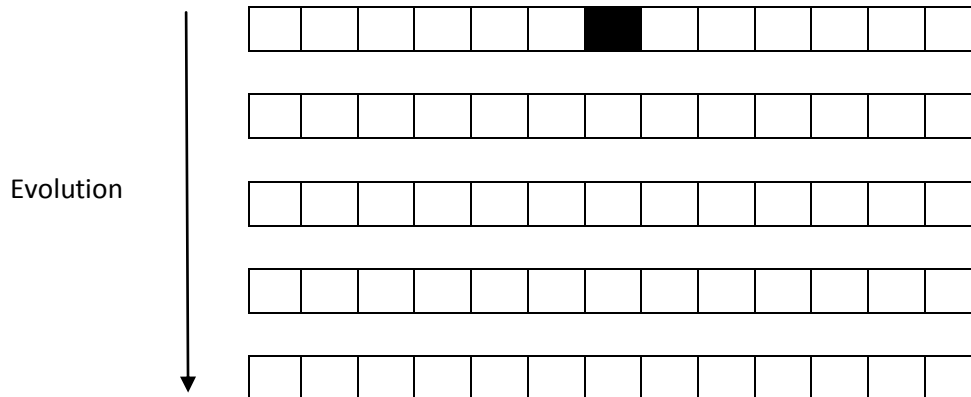
Ref : *ING2-PROGPAR-TP-OPENMP*

A l'intention de : Etudiants d'ING2 MI

Créé le : 21/11/2014

### Exercice 3 : automate cellulaire

Soit une ligne de M cases blanches. Pour générer la ligne suivante, on définit des règles pour déterminer la couleur d'une cellule suivante en fonction des couleurs de ses voisines sur la ligne actuelle. On définit un état initial (première ligne). Si on affiche la ligne à chaque étape du traitement, en fonction des règles et de l'état initial choisis, on obtient des motifs plus ou moins complexes, qui peuvent être répétitifs, présenter des régularités ou ne plus évoluer du tout.



### Règles :

- une case noire entourée de 2 cases blanches devient blanche à l'étape suivante sinon elle reste noire.
- Une case blanche devient noire à l'étape suivante si elle possède au moins une voisine noire.

1. Donnez le code séquentiel permettant de simuler ce phénomène
2. Paralléliser ce code en utilisant l'API OpenMP
3. Paralléliser ce mécanisme en utilisant le multithreading en java