

TP OpenMP

Exercice1 :

Le code dans le fichier `prod_mat.c`, calcule le produit de matrices : $C = A \times B$

Dans cet exercice, il s'agit :

1. d'insérer les directives OpenMP appropriées.
2. de tester les différents modes (STATIC, DYNAMIC, GUIDED) de répartition des itérations d'une boucle et faire varier la taille des paquets

Exercice 2 :

Le programme dans le fichier `reduction_tab.c` est extrait d'un code de chimie. Il s'agit de réduire un tableau tridimensionnel en un vecteur.

Le but de ce TP est de paralléliser ce noyau de calcul sans toucher à l'ordre initial des boucles (i.e. k, j, i).

1. Analyser le statut des variables et adapter le code source de façon à paralléliser la boucle la plus externe en k.

Exercice 3 :

Paralléliser le programme "`prod_cons.c`". Il s'agit d'un modèle bien connu appelé le modèle producteur-consommateur.

- Un thread produit des valeurs qu'un autre thread consomme.
- Souvent utilisé avec un flux de valeurs produites pour mettre en œuvre le "parallélisme de pipeline".

L'idée est d'implémenter une synchronisation par paires entre les threads.

Exercice 4 :

Créez une version parallèle du programme `pi_mc.c`, sans changer les interfaces des fonctions dans "`random.c`". Le générateur de nombres aléatoires doit être *threadsafe*.