

TP n° 6 : OpenMP

0. Introduction

L'objectif de ce TP est de mesurer l'impact de la parallélisation OpenMP et de la comparer à l'utilisation des extensions SIMD

1. Calcul de π par intégration

Mesurer les temps d'exécution de la version séquentielle et des différentes versions OpenMP pour le calcul de π par intégration, en utilisant le programme fourni sur le site du M1 Archi (<http://www.lri.fr/~de/ArchiM1-1112.htm>) pour 2/4/8 threads.

2. Calcul de π par arc tangente

π peut être calculé par la formule suivante : $\frac{\pi}{4} = \arctan(1) = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$

- Ecrire une version scalaire utilisant des flottants en double précision pour calculer π et mesurer le temps d'exécution (en nombre de cycles/k) et en vérifiant le résultat obtenu sans l'option `-sse2`.
- Tester la possibilité de « vectorisation » par le compilateur et mesurer le temps d'exécution
- Ecrire une ou plusieurs versions OpenMP (sans SIMD) et mesurer les temps d'exécution pour 2/4/8 threads.
- Ecrire une version OpenMP avec SIMD (par « vectorisation » automatique ou utilisation d'intrinsics) et mesurer les temps d'exécution pour 2/4/8 threads.

3. Jacobi

Ecrire une version OpenMP du code séquentiel « Jacobi » fourni sur le site du M1 Archi et mesurer les temps d'exécution et l'accélération pour 2/4/8 threads

4. Exercices supplémentaires (optionnel)

Ecrire les versions séquentielles et les versions OpenMP et mesurer les accélérations pour

- le produit matrice-vecteur
- le produit de matrices ikj

Rendre un compte rendu de TP par binôme au plus tard le 19 Décembre.