Projet de Programmation Parallèle
ES : Architecture matérielle et logicielle des ordinateurs
2021-2022

claude.tadonki@mines-paristech.fr

A remettre au plus tard le 11 mars 2022

Objectif: Ecrire un programme « multi-threadé » + « vectoriel » en C, puis comprendre les divers aspects liés à son fonctionnement correct ainsi qu'à ses performances.

Remarque: Le programme doit être écrit, en individuel, en langage C, en considérant Pthread et AVX. A remettre au plus tard le lundi 11 mars 2022 par mail.

Exercice 1: Ecrire la fonction
double rnorm(float *U, int n);
qui calcule (en double précision)

$$d(u,n) = \sum_{i=0}^{n-1} \sqrt{u_i}$$

pour un vecteur U de n réels (en simple précision).

Exercice 2: Ecrire une version vectorielle (AVX) de la fonction précédente en supposant que l'adresse de U est alignée.
double vect_rnorm (float *U, int n);
Indication : Le calcul en vectoriel de la fonction « racine carrée » est illustré dans la dernière page du polycopié, il s'agit d'une macro-instruction vectorielle (i.e. « intrinsics »).

Exercice 3: Proposer une version multithreadée de la fonction rnorm
void rnormPar(double *U, int n, int nb_threads, int mode);
qui considère une exécution avec nb_threads threads. On peut par exemple supposer que chaque thread traitera un block du vecteur U (partitionnement statique). Le paramètre *mode* indique le type de calcul (0 : scalaire et 1 : vectoriel). Pensez à créer une structure pour les données du thread.
Remarque : Vous utiliserez l'approche d'une variable globale pour la somme finale, laquelle sera mise à jour par chaque thread après son calcul en demandant le « mutex ».
Il ne s'agit donc pas de créer un tableau intermédiaire dans lequel chaque thread inscrira sa somme partielle

Exercice 4: Ecrire la fonction principale de votre projet
Le main() doit

a) Créer et initialiser le vecteur U avec des valeurs aléatoires comprises entre 0 et 1.
b) Calculer la rnorm(U) (toutes les versions : scalaire/vectorielle/multithreadée/totale)
c) Afficher l'accélération par rapport au temps d'exécution (prendre $n = 1024^2$)

Remarque : Votre affichage devra être de la forme ci-dessous :
VALEURS
Séquentiel (scalaire : … vectoriel : ….) Parallèle (nb_thread : … scalaire : … vectoriel : ….)
TEMPS D'EXECUTION
Séquentiel (scalaire : … vectoriel : ….) Parallèle (nb_thread : … scalaire : … vectoriel : ….)
Accélération (vectoriel : … multithread : … vectoriel + multithread : …)

Où les « … » sont à remplacer par les valeurs correspondantes. Toutes les accélérations sont calculées par rapport à la version originale (exercice 1)