Rapport – Travaux Pratique 6

Sommaire

Exercice 1

Taille des tableaux	Moyenne du temps d'exécution du patron MAP sur GPU (µs)	Moyenne du temps d'exécution du patron MAP sur CPU (µs)
1 024	10	20
16 384	10	140
262 144	35	2 000
4 194 304	410	35 000
67 108 864	6 450	590 000
268 435 456	21 500	2 350 000

On observe que le temps d'exécution pour des petits tableaux est 10 fois plus rapide sur GPU que sur CPU, alors qu'avec des grands tableaux on atteint un temps d'exécution qui est plus de 100 fois plus rapide sur GPU que sur CPU.

On remarque aussi que le patron MAP pour calculer l'addition des éléments de deux tableaux est 5 fois plus rapide sur GPU (avec des tableaux de grandes tailles) qu'avec la fonction MAP de la librairie standard C++ utilisé pour le TP3.

Exercice 2

Pour cette algorithme, j'ai choisi de faire un découpage en deux dimensions de threads et de blocs. En effet chaque pixel de l'image peut être retrouvé grâce à des coordonnées 2D (x, y) avec x pour la largeur et y pour la hauteur. Pour découper les threads, j'ai décidé de faire du 32 x 32 car les images sont en moyenne plutôt carré puisque les images peuvent être au format paysage ou au format portrait. Pour les blocs j'ai découpé en deux dimensions aussi avec la première dimension qui est découpe selon la largeur et la seconde qui est découpé selon la hauteur.

Exercice 3

Pour le patron GATHER avec CUDA, j'ai fait le même découpage que pour l'exercice précédent et j'ai appliqué la même formule que le modèle PRAM vu en cours.

Exercice 4

Mêmes remarques que pour l'exercice précédent.

Exercice 5

Schémas de répartition des blocs	Moyenne du temps d'exécution sur l'image Raffael_012_filtered.ppm (µs)
(1, 1 024)	2 849 000
(8, 128)	477 000
(16, 64)	304 000
(32, 32)	205 000
(64, 16)	205 000
(128, 8)	205 000
(1 024, 1)	205 000