Auteurs : *Barry Jean-Noël & Allio David (Groupe 2)*
Date : 06/10/2019

Compte-rendu du TP A : Mandelbrot sur plusieurs threads

Auto-évaluation

Indiquez ici par une note globale entre 0 et 10 votre degré de réussite à l'exercice 2 du TP A. Expliquez brièvement cette note.

8 : Techniquement, notre programme est entièrement fonctionnel, mais nous n'avons utilisé les threads que de facon très basique, n'augmentant au final que peu la vitesse d'execution du programme, j'attribue donc une note globale élevée, mais pas maximale, puisque grandement améliorable.

Pédagogie

Indiquez quel a été selon vous le principal intérêt pédagogique de cet exercice en termes de programmation. Vous pourrez en particulier souligner les difficultés éventuelles rencontrées pour la compréhension du sujet, l'élaboration d'une solution ou le codage en Java.

Le principal intérêt pédagogique de cet exercice a été une meilleur compréhension des threads, des différentes techniques pour les générer, et de l'optimisation qui peut en résulter.
La difficulté majeure fut le passage de threads statiques en thread dynamiques puisque cela nécessitait une fonction de dessin bien plus complexe, s'adaptant au travail déjà effectué pour optimiser celui restant.

Relevé de mesures observées

Programme	Temps de calcul (en s.)	Gain
Séquentiel	30.998	0
Statique (Q. 3)	24.033	6.965
Dynamique (Q. 4)	7.55	16.483
500 threads	4.379	3.171
250 000 threads	39.318	-34.939

Conditions matérielles des observations

*Vous préciserez ici la valeur de **max** utilisée et les spécificités matérielles de votre machine : fréquence d'horloge, marque du microprocesseur, nombre de processeurs logiques ou physiques.*

Max=75 000
Processeur Amd Radeon with Vega Mobile Gfx à 2.00 Ghz, 4 coeurs et 8 processeurs logique

Analyse

Les gains observés sont-ils conformes à vos attentes ? Pourquoi ? Quelles leçons d'ordre général tirez-vous de ces observations ?

Les gains observés sont effectivemment conformes à nos attentes puisque chaque méthode a bien été plus rapide que la précédente, avec le plus grand gain résultant du passage de threads statiques à dynamiques, exception faite du passage a 250 000 threads, qui a beaucoup trop divisés l'image initiale pour etre encore efficace.
De façon générale, nous tirons de ces observations une connaissance claire de ce que l'optimisation de threads peut apporter a la rapidité d'exécution d'un programme.