Auteur : WOLFGER Killian
Date : 11/10/2019

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.

8/10 : Malgré quelques hésitations et erreurs au début du TP je pense avoir relativement bien compris le but de l'exercice et en quoi les différentes méthodes sont plus ou moins efficaces. J'ai notamment une bien meilleur compréhension de la fonction synchronized et de son rapport coût/effet en fonction de la situation.

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.

Je pense que le but principal de l'exercice était de retirer l'approche et la vision naïve que nous avions sur les thread : au plus il y en a au mieux c'est !
De plus il a permis, pour ma part en tout cas, de mieux saisir le fonctionnement des thread, de savoir la bonne manière de partager un travail pour un nombre de thread donné ; ou encore comment optimiser le temps de calcul d'une tâche qui a besoin d'être synchronisé pour en diminuer l'impact sur le temps d'exécution des autres thread.

Relevé de mesures observées

i	**Programme**	**Temps de calcul *t* (en s.)**	**Gain** *(t.0/t.i)*
0	Séquentiel	30,0 (+- 1,0)	1
1	Statique (Q. 3)	13,7 (+- 0,2)	2,190
2	Dynamique (Q. 4)	8,5 (+-0,1)	3,529
3	500 threads	8,2 (+- 0,05)	3,659
4	250 000 threads	52 (+- 2,0)	0,577

Note : Y (+- x) pour x étant l'écart-type de Y

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.*

Valeur **max** utilisée : 88 000
Caractéristiques de la machine :
-Processeur : Intel*(R)* Core*(TM)* i5-7500
-Fréquence d'horloge : 3,40 GHz
-Nombre de processeurs physique : 4 | Nombre de processeurs logiques : 4

Analyse

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

Si j'avais obtenu ces résultats au tout début du TP il n'aurait pas du tout étaient conformes à mes attentes, je me serais probablement dis naïvement qu'un thread par pixel ce n'est pas si mal.
Or durant ce tp j'ai travaillé et modifié le travail et les tâches des thread de nombreuses fois, et je pense en avoir tiré de bonnes conclusions.

Premièrement « au plus il y en a au mieux c'est » n'a définitivement pas de sens quand on parle de thread. Ici créer un thread par pixel ne fais que freiner le calcul, en effet la création des 250 000 objets à tout de même un coût (0,5 s.) même si on pourrait dire qu'il est négligeable à l'échelle du temps d'exécution du programme (52 s.). Or même si on les lance à la suite le plus vite possible (i.e à l'aide d'un for) la tâche est si petite que les 5000 premiers vont avoir finis avant que les 5000 suivant n'ai commencé. L'image ne se fait donc pas d'un coup comme on pourrait naïvement le penser et nous avons donc des thread qui n'accélèrent pas du tout le déroulement du programme, au contraire.
Deuxièmement, pour une tâche spécifique un certain ordonnancement des « sous tâches » est optimal. La fonction synchronized est par exemple à utiliser avec parcimonie, si du code non bloquant se trouve tout de même dans cette dernière cela peut ralentir considérablement l'exécution des autres processus et donc du programme, c'est en effet un temps supplémentaire non justifié pendant lequel nous prenons le verrou.