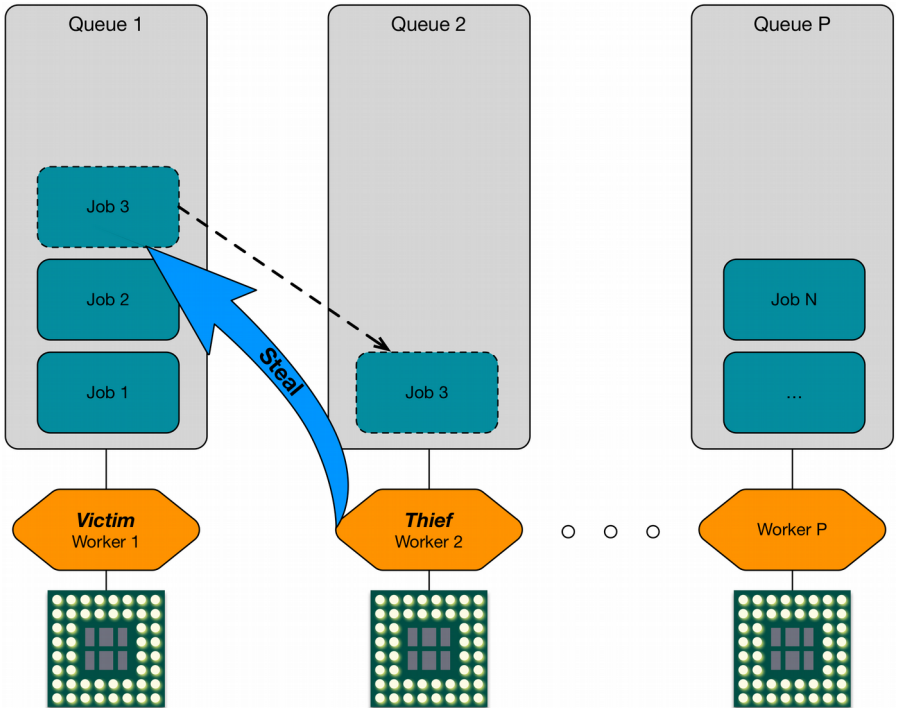
Rapport de Projet
de Programmation Système :

Ordonnanceur par work stealing



Johann HOSPICE – Desse KANDE
Université Paris Diderot
Année : 2017/2018

I) Introduction :

Ce projet de programmation système porte sur le parallélisme d'exécution d'un programme. En effet très souvent nos machines disposent de plusieurs cœurs dont chacun peut exécuter, à un moment donné, un thread ou processus.
Un programme purement séquentiel n'exploitera qu'une fraction de la puissance de calcul du processeur ; pour l'exploiter davantage, il faut écrire des programmes parallèles.

L'objectif du projet fut d'implémenter un ordonnanceur (scheduler), composant logiciel dont le rôle est de décider sur quel thread exécuter une tâche.
Dans un premier temps en work sharing où les tâches sont distribuées parmi les threads, puis dans un second temps en work stealing, où les threads oisifs volent des tâches aux autres processeurs.

Work sharing :

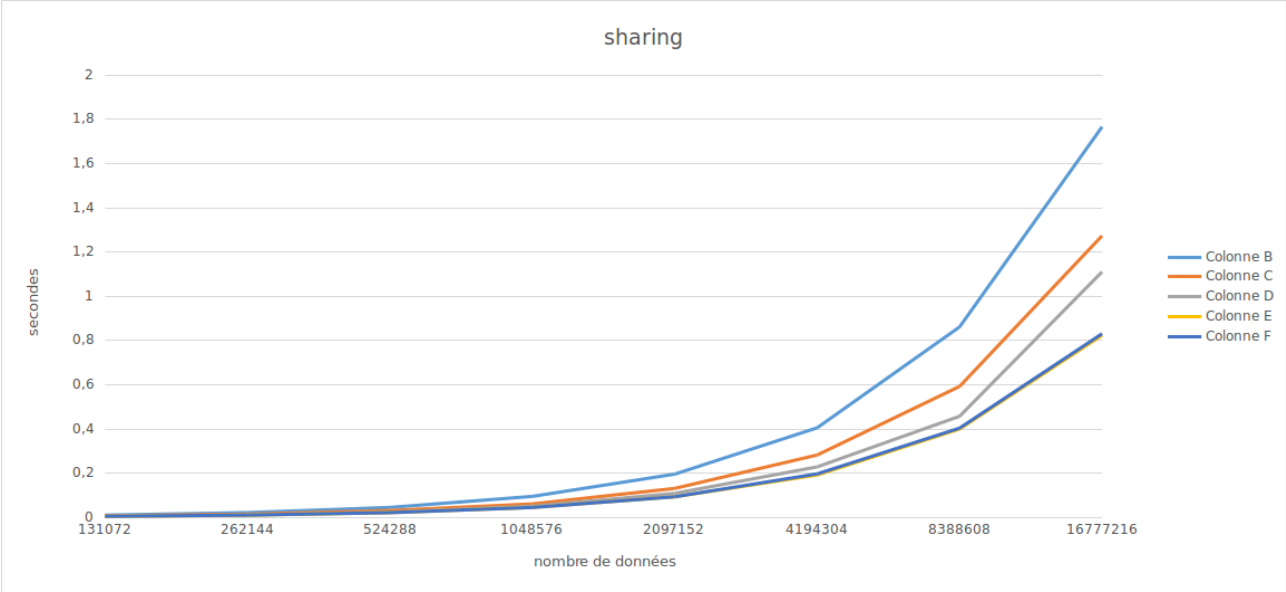
L'implémentation du mode work sharing s'est fait comme suit :
Nous avons tout d'abord procédé à la création d'une structure pile (le fichier lifo.c, last in, first out), ainsi que l'ensemble des méthodes nécessaire à son bon fonctionnement. Puis nous avons implémenté l'interface sched.h fournit avec le sujet via le fichier sharing.c. Enfin nous avons effectués des tests pour le mode work sharing.

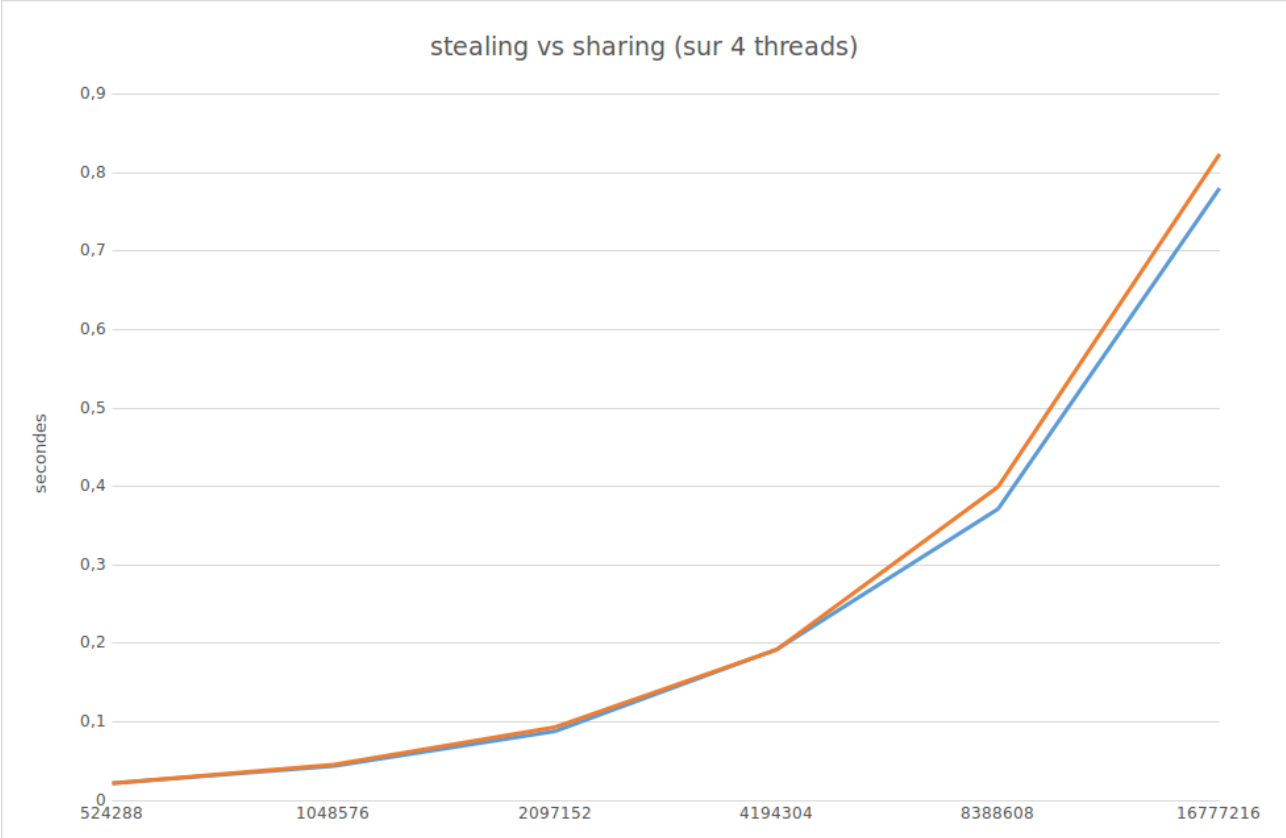
Work stealing :

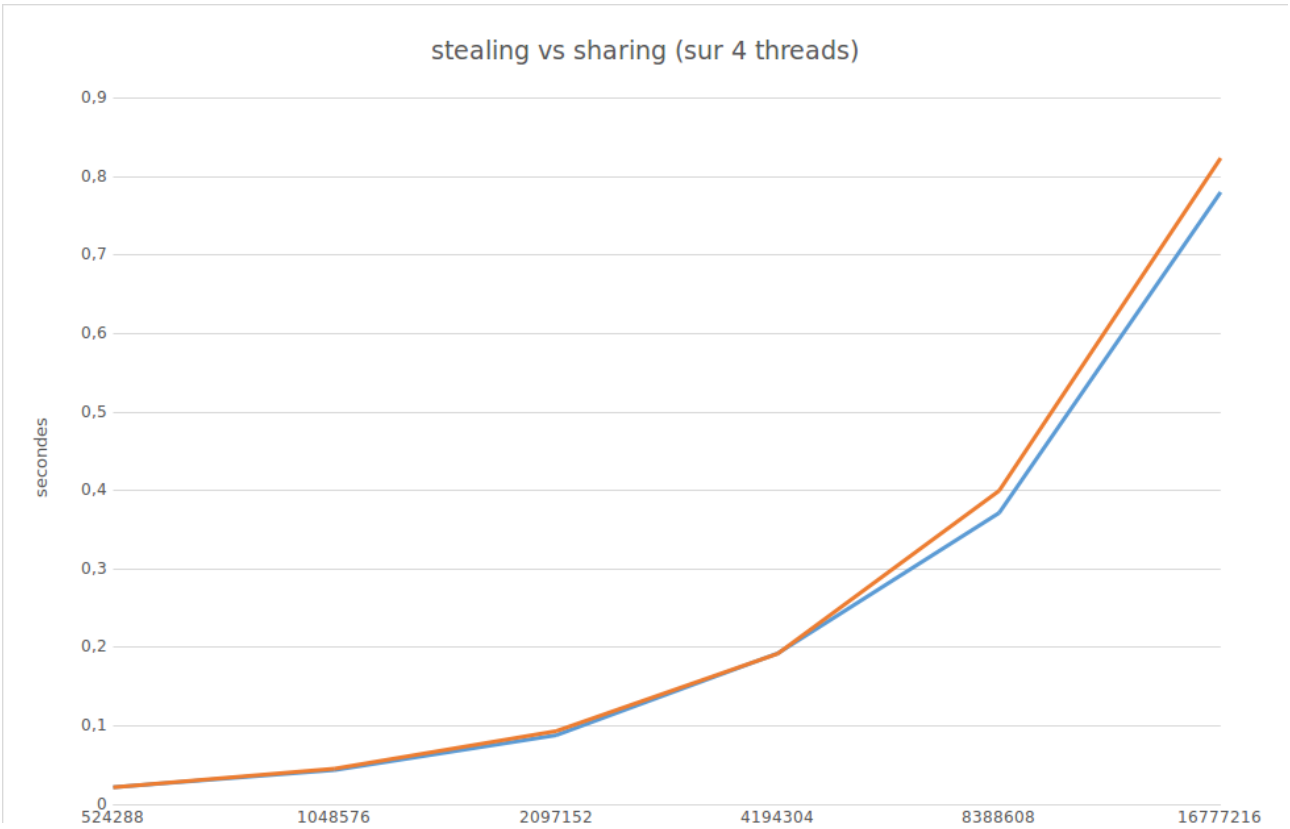
De même pour le mode work stealing , ce fut d'abord la création d'une structure deque (le fichier deque.h, utilisation d'une liste doublement chaînée), puis l'implémentation de sched.h via le fichier stealing .c. Enfin nous avons également effectués des tests pour le mode work stealing.

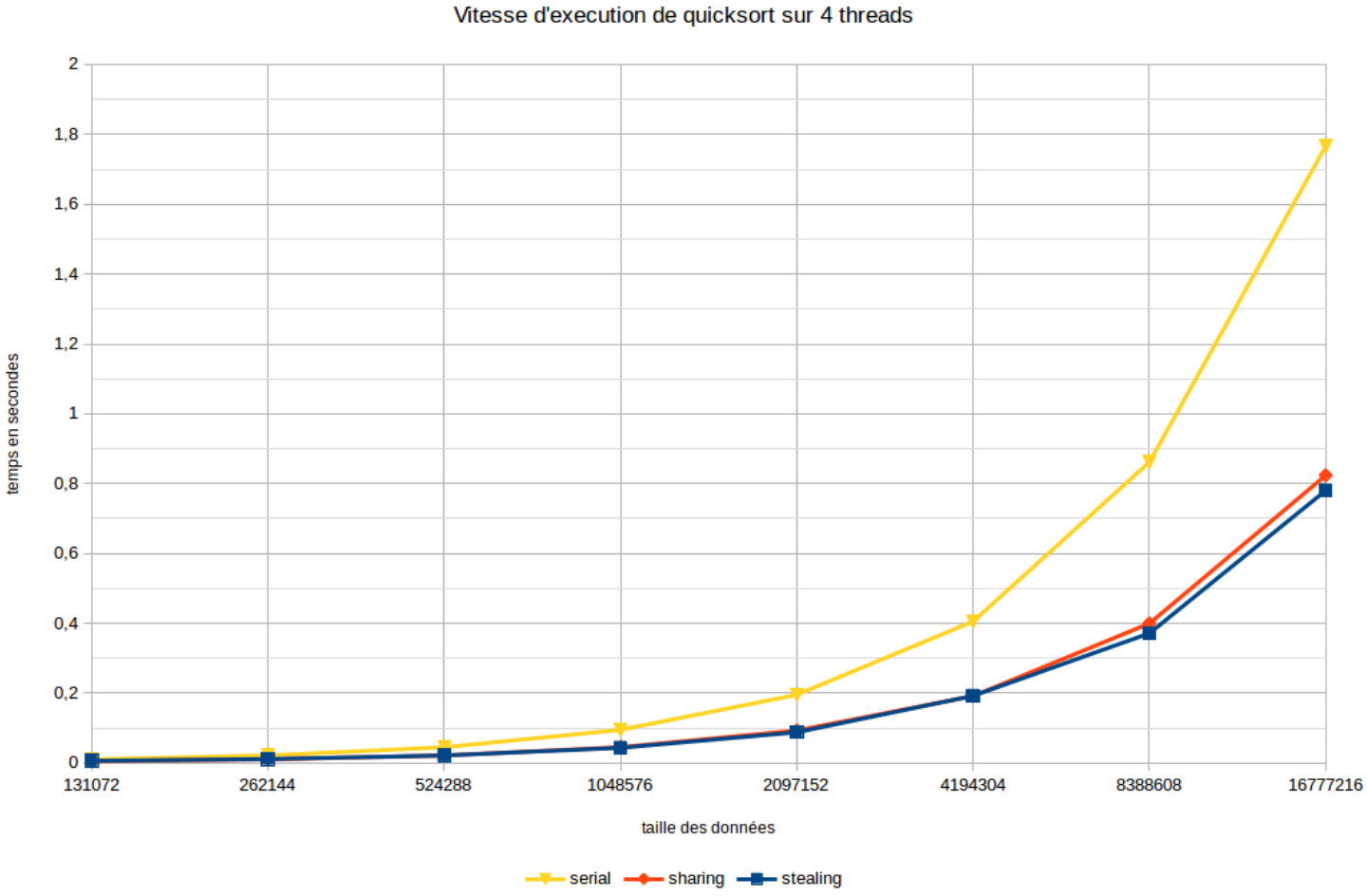
Extensions :

Statistiques et benchmarks :






stealing vs sharing (sur 4 threads)
secondes
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
524288
1048576
2097152
4194304
8388608
16777216


Vitesse d'execution de quicksort sur 4 threads
temps en secondes
2
1,8
1,6
1,4
1,2
1
0,8
0,6
0,4
0,2
0
131072
262144
524288
1048576
2097152
4194304
8388608
16777216
taille des données
serial
sharing
stealing

Programmes d’exemple :

Le programme contenu dans le fichier copymatrix.c permet de faire la copie d’une matrice, séquentiellement, tout comme parallèlement.

Conclusion :

Ce projet correspond à la mise en œuvre de certains points du cours d’algorithmique que nous avons pu voir au semestre 1. Il nous a permis de nous familiariser avec la manipulation des threads et processus en langage c. Cela fut très bénéfique pour nous, car il nous a éclairé sur certain point.