Jouve Léonard, Kunzli Christophe

Introduction

Dans ce labo, nous devons implémenter un algorithme quicksort exploitant plusieurs threads ainsi que le principe de producteur/consommateur.

Conception

Nous avons écrit une classe monitor chargée de distribuer la charge de travail. Le travail consiste à partitioner et trier plusieurs sous-séquences du tableau à trier.

Chaque tâche est représentée par une instance de la structure Task. Et ces tâches sont stockées dans une file de la classe Monitor (tasks).

La méthode scheduleTask permet de produire une tâche en l'insérant dans la file. executeTask est appelée par chaque thread qui bloque en attendant qu'une tâche soit planifiée, et l'exécute.

Quicksort quant à elle prend une tâche en paramètre, l'exécute, schedule les 2 tâches découlant puis se remet en attente d'une nouvelle tâche.

Afin d'éviter l'attente active des threads, nous avons utilisé un PcoConditionVariable.

Test

1. Test de tri simple avec un seul thread :
    - Description : Vérifie si l'algorithme QuickSort fonctionne correctement avec un seul thread.
    - Entrée : Tableau non trié avec 1000 éléments aléatoires.
    - Attendu : Le tableau trié correctement.
2. Test avec plusieurs threads (4 threads) :
    - Description : Vérifie la parallélisation correcte et le tri avec plusieurs threads.
    - Entrée : Tableau non trié avec 1000 éléments aléatoires.
    - Attendu : Le tableau trié correctement.
3. Test avec tableau de petite taille (10 éléments) :
    - Description : Teste si l'algorithme gère les petits tableaux efficacement.
    - Entrée : Tableau avec 10 éléments aléatoires.
    - Attendu : Le tableau trié correctement.
4. Test avec tableau de grande taille (10^4 éléments) :
    - Description : Évalue les performances et la stabilité avec un grand tableau.
    - Entrée : Tableau non trié avec 10^4 éléments aléatoires.
    - Attendu : Le tableau trié correctement.
5. Test avec un tableau déjà trié :
    - Description : Vérifie si l'algorithme gère les cas optimaux efficacement.
    - Entrée : Tableau de 1000 éléments déjà triés.
    - Attendu : Le tableau reste trié.

Benchmark

Nombre de threads	Temps d'exécution (ms)	CPU (ns)	Iterations
1	7.095	444,716	98
2	12.704	466,507	56
4	18.551	610,538	36
8	34.977	1,344,726	20
16	43.207	2,820,687	13

- Note : Nous avons dû réduire la taille du tableau à 10'000 afin de pouvoir run le benchmark. Ceci à cause d'un segfault lorsque la taille du tableau est trop grande