Quicksort Multithreadé

Description

Ce projet implémente une version multithreadée de l'algorithme **Quicksort** en C++. Il permet de trier efficacement des vecteurs tout en exploitant la puissance des threads pour paralléliser les tâches.

Le programme utilise un moniteur de Mesa pour synchroniser les threads à l'aide de PcoConditionVariable et PcoMutex.

Fonctionnalités principales

- Implémentation du tri Quicksort avec un nombre défini de threads.
- Synchronisation efficace via un moniteur de Mesa.
- Gestion dynamique des tâches à trier à l'aide d'une file protégée.

Structure du projet

- quicksort.h : Implémentation de l'algorithme Quicksort multithreadé.
- utils.h : Fonctions utilitaires pour générer des séquences et vérifier leur tri.
- tests/main.cpp : Tests unitaires avec Google Test.
- benchmark/main.cpp : Mesures des performances avec différents nombres de threads.

Tests

Les tests sont réalisés à l'aide de **Google Test** et incluent :
- Vérification que le tableau trié n'est pas vide.
- Validation que le tableau est trié dans l'ordre croissant.

```
TEST(SortingTest, Test1) {
   int size = 16;
   int nbThreads = 2;
   int seed = 23;
```

```
    test(nbThreads, size, seed);
}
```

Résultats attendus

- Un gain de performance linéaire avec l'augmentation du nombre de threads jusqu'à un certain seuil (limité par les capacités du matériel et les coûts de synchronisation).
- Validation fonctionnelle pour des tableaux de différentes tailles.

Résultats réels

Après plusieurs tests effectués sur notre ordinateur :

- Performance : Aucun changement notable des performances entre différents nombres de threads. Cela pourrait être dû aux limites du processeur ou à d'autres facteurs (comme le coût de la synchronisation ou une faible charge sur les threads).
- Benchmark sur la VM : Il a été impossible de lancer les benchmarks sur la machine virtuelle, même après des modifications de fichiers.

Voici un exemple des résultats obtenus sur notre machine locale :

```
-------------------------------------------------------------------------
Benchmark                               Time             CPU   Iterations
-------------------------------------------------------------------------
BM_QS_MANYTHREADS/1/real_time  2024994333 ns    411111000 ns            1
BM_QS_MANYTHREADS/2/real_time  2037261917 ns    411263000 ns            1
BM_QS_MANYTHREADS/4/real_time  2033776834 ns    414182000 ns            1
BM_QS_MANYTHREADS/8/real_time  2045370625 ns    418866000 ns            1
BM_QS_MANYTHREADS/16/real_time 2063411250 ns    417238000 ns            1
```

Difficultés rencontrées

- Gestion des threads : L'implémentation de la file de tâches protégée par un moniteur de Mesa a demandé une attention particulière pour éviter les conditions de course et les blocages.
- Limitations matérielles : Les performances sont impactées par les capacités du matériel utilisé pour les tests.
- VM non opérationnelle : Malgré plusieurs tentatives, les benchmarks n'ont pas pu être exécutés correctement sur la machine virtuelle.

Auteurs

- Maxime Regenass

- Nathan Füllemann