

README - MIOSSEC MAEL.

Description rapide de la structure du programme

Ce programme en C effectue le **calcul des minimums** entre deux tableaux **A** et **B** de grande taille en utilisant le **multi-threading** avec **pthread**. Il supporte trois stratégies de répartition des calculs :

1. **Répartition cyclique** : chaque thread traite les éléments un par un, en sautant `nb_threads` éléments à chaque itération.
 2. **Répartition par blocs** : chaque thread traite des **blocs de `BLOCK_SIZE`** éléments, en sautant `nb_threads * BLOCK_SIZE`.
 3. **Répartition dynamique (farming)** : un thread récupère dynamiquement des blocs à traiter en demandant **un nouvel index via un mutex** après chaque bloc terminé.
-

Structure du programme

- Déclarations et allocations

- Trois tableaux **A, B, C** sont **alloués dynamiquement** (`malloc`).
- Un `enum Strat` définit les trois stratégies.
- Une structure `ThreadArgs` est utilisée pour passer les paramètres aux threads.

- Lancement des threads

- Les threads sont créés avec `pthread_create()` et chacun exécute la fonction `calcul_min()`.
- Selon la stratégie choisie, les threads appliquent la bonne méthode de répartition des calculs.
- Pour le **farming**, un `mutex` protège l'accès à l'index des blocs à traiter.

- Synchronisation et mesure du temps

- Le temps d'exécution est mesuré avec `gettimeofday()`.
- Les threads sont synchronisés avec `pthread_join()`.
- Le programme affiche **le temps moyen sur plusieurs exécutions**.

- Affichage et exportation des résultats

- Les résultats sont affichés au format **CSV** pour analyse dans un tableur.
- Pour le **farming**, le programme affiche **le nombre min et max de blocs traités par thread**.

- Libération de la mémoire

- À la fin, les tableaux **A**, **B**, et **C** sont libérés (**free**).

Résumé

Le programme **optimise le calcul parallèle** du minimum entre deux grands tableaux en testant différentes stratégies de répartition du travail entre threads. Il mesure **l'efficacité** et exporte les résultats pour analyse.

Quelle est la meilleur configuration pour ma machine

Ma machine possède 8 coeurs et la meilleure configuration pour elle est d'utiliser la **répartition cyclique des blocs avec 128 threads et migration autorisé** avec un temps moyen de **0.441952 secondes**.

Temps d'exécution selon le nombre de threads (échelle logarithmique)

