Projet

OS202

1.  **Regarder le nombre de cœurs physiques sur votre machine et la taille des différentes mémoires caches que vous donnerez dans votre rapport.**

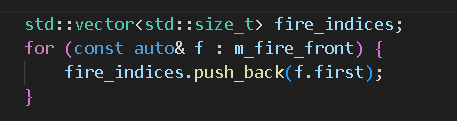
* Model name: 13th Gen Intel(R) Core(TM) i7-13650HX
* CPU family: 6
* Model: 183
* Thread(s) per core: 2
* Core(s) per socket: 10
* Caches (sum of all):
* L1d:480 KiB (10 instances)
* L1i: 320 KiB (10 instances)
* L2: 12.5 MiB (10 instances)
* L3: 24 MiB (1 instance)

2**. Mesurer les temps moyen pris pour chaque pas de temps, puis pour l'affichage et l'avancement en temps**

* Temps moyen d'avancement : 179.684 ms
* Temps moyen d'affichage : 7015.42 ms
* Temps moyen total par pas de temps : 7195.11 ms

3.

* **Récupérez dans un tableau toutes les clés contenues dans le dictionnaire et à l'aide d'un indice, parcourez ces clefs pour l'avancement en temps**



* **Tableau**

| **Threads** | **Temps moyen Avancement (ms)** | **Temps moyen Total (ms)** | **Speedup Avancement** | **Speedup Total** |
| --- | --- | --- | --- | --- |
| 2 | 0.209 | 4.268 | 0.58 | 1.11 |
| 4 | 0.232 | 4.604 | 0.53 | 1.02 |
| 8 | 0.298 | 6.271 | 0.41 | 0.75 |
| 12 | 0.347 | 5.276 | 0.35 | 0.89 |
| 14 | 0.572 | 5.257 | 0.21 | 0.90 |
| 16 | 0.533 | 4.228 | 0.23 | 1.12 |
| 18 | 0.521 | 4.193 | 0.23 | 1.13 |
| 20 | 1.401 | 6.342 | 0.09 | 0.74 |

**2. Deuxième étape**

**2.1** Voici un résumé des mesures moyennes de Temps Total(ms) par itération pour chaque jeu de résultats :

* Code séquentiel d’origine : ~4,41 ms/itération
* MPI avec 1 processus : ~4,82 ms/itération
* MPI avec 2 processus : ~3,89 ms/itération
* MPI avec 4 processus : ~5,42 ms/itération

**Interprétation**

1. Processus MPI 1 (~4,82 ms)

L'utilisation de MPI avec un seul processus introduit une petite surcharge par rapport au code séquentiel d'origine (~ 4,41 ms). Cela est dû à l'initialisation de l'environnement MPI et aux appels supplémentaires.

1. Processus MPI 2 (~3,89 ms)

Ici, nous constatons un léger gain de performance par rapport au séquentiel. En divisant la charge de travail (et en supposant que la partie parallélisable est suffisamment grande), le coût de communication est compensé par le parallélisme effectif. Par conséquent, nous obtenons un temps moyen par itération inférieur à celui du code séquentiel.

1. Processus MPI 4 (~5,42 ms)

En passant à 4 processus, on observe une dégradation des performances moyennes. Deux raisons principales possibles Augmentation de la synchronisation et de la surcharge de communication. Répartition de la charge : si la part de travail par processus n'est pas suffisamment importante, le temps perdu en communication et en synchronisation l'emporte sur le gain en parallélisme.

| Nombre de Threads | | Temps moyen Avancement (ms) | | --- |  |  | | --- | | | Temps Total moyen (ms) | | --- |  |  | | --- | | | Accélération Avancement | | --- |  |  | | --- | | | Accélération Globale | | --- |  |  | | --- | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |