

Laboratoire de High Performance Coding

semestre printemps 2023

Laboratoire 6 : Parallélisme des tâches

Temps à disposition : 4 périodes (2 séances de laboratoire).

1 Objectifs de ce laboratoire

L'objectif de ce laboratoire est d'explorer le parallélisme des tâches dans le contexte du calcul haute performance. Vous serez amenés à paralléliser deux tâches distinctes, à savoir la statistique de fichier parallélisée et l'algorithme A* parallélisé.

Pour les deux parties du laboratoire, vous devrez rédiger un rapport détaillant vos choix d'implémentation de la version parallélisée. Vous devrez expliquer la stratégie de répartition du travail que vous avez choisie et comment vous avez attribué les tâches. De plus, vous devrez réaliser une analyse comparative des performances entre la version non parallélisée et la version parallélisée de chaque application, en mettant en évidence les gains obtenus grâce à la parallélisation. Des expérimentations devront être menées en utilisant des fichiers de tailles variées pour évaluer les performances du programme parallélisé.

2 Statistique de fichier parallélisée

Dans cette première partie, vous serez confrontés à la tâche de lecture d'un fichier volumineux et au besoin de compter les occurrences de caractères dans ce fichier. Une version non parallélisée du code vous a été fournie.

L'objectif de cet exercice est d'implémenter une version parallélisée de cette application de comptage d'occurrences en utilisant les outils de parallélisme de votre choix, tels que OpenMP ou pthreads. Le but est de tirer parti des capacités de parallélisme offertes par les processeurs modernes afin d'accélérer le traitement et d'améliorer les performances de l'application.

Afin de déterminer la pertinence du parallélisme, nous vous encourageons à utiliser le script `gen_file.py` fourni, qui vous permet de générer des fichiers de différentes tailles. En utilisant des fichiers de tailles variées, vous pourrez évaluer les performances de la version parallélisée par rapport à la version non parallélisée et analyser l'impact du parallélisme sur les temps de traitement.

Vous êtes libre de modifier les codes et le Makefile comme vous le souhaitez !

Vous devez rédiger un rapport détaillant vos choix d'implémentation de la version parallélisée. Expliquez en détail la stratégie de répartition du travail que vous avez choisie et comment vous avez attribué les tâches. Incluez également une analyse comparative des performances entre la version non parallélisée et la version parallélisée de l'application, en mettant en évidence les gains de performances obtenus grâce à la parallélisation. Utilisez les fichiers générés par le script `gen_file.py` pour étayer votre analyse.

3 A* parallélisé

Dans cette deuxième partie du laboratoire, nous abordons l'algorithme de recherche de chemin A*. Vous avez déjà travaillé plusieurs fois sur cet algorithme et possédez donc une version séquentielle fonctionnelle de celui-ci. L'objectif de cette partie est de parvenir à paralléliser la tâche afin d'améliorer les performances ou la précision en exploitant les capacités de parallélisme offertes par le matériel à notre disposition.

Le but de la parallélisation de l'algorithme A* est de répartir efficacement la charge de travail entre les threads pour accélérer la recherche du chemin optimal. Les avantages potentiels de la parallélisation sont une réduction du temps de calcul global ou une meilleure exploration de l'espace de recherche, permettant de trouver des chemins plus précis.

Dans votre rapport, détaillez vos choix de parallélisation. Expliquez comment vous avez identifié les parties de l'algorithme qui peuvent être exécutées en parallèle et comment elles sont réparties entre les threads. Il est important de justifier vos choix de parallélisation en fonction des caractéristiques de l'algorithme A* et des contraintes de performance ou de précision.

Dans le rapport, incluez des comparaisons entre l'implémentation séquentielle et l'implémentation parallélisée de l'algorithme A*. Ces comparaisons devraient mettre en évidence les gains de performances ou de précision obtenus grâce à la parallélisation. Vous devrez également décrire les situations spécifiques dans lesquelles le choix de parallélisation se révèle le plus avantageux, en vous basant sur des exemples concrets et des expérimentations réalisées.