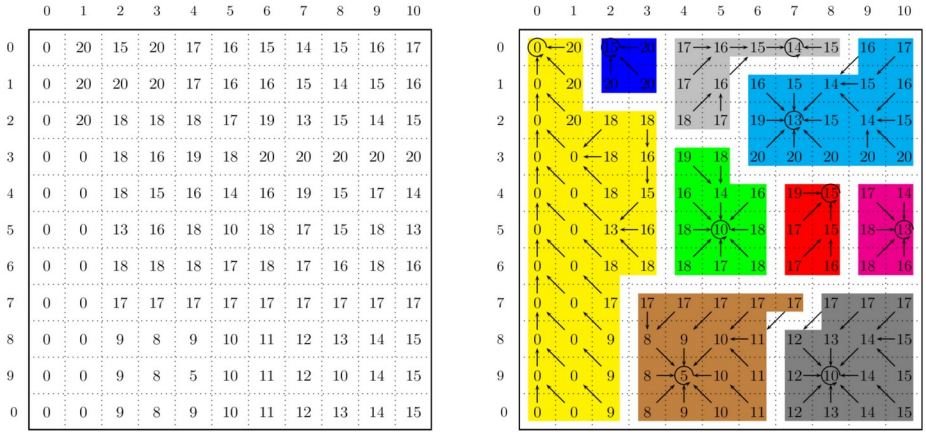
Projet programmation parallèle

DARBOUX

M1 Informatique, année 2024-2025
destiné aux étudiants ayant déjà suivi cette option en licence 2023-2024

Contexte



Ce projet utilise les Modèles Numériques de Terrain (ou MNT, voir https://fr.wikipedia.org/wiki/Modèle_numérique_de_terrain), qui sont simplement des cartes d'altitude d'une région géographique. Elles sont stockées dans des fichiers d'extension `.mnt` de format très simple, en texte clair : quelques caractéristiques (taille, position, etc.) suivies d'un tableau de nombres flottants donnant l'altitude de chaque point. Quelques exemples, de différentes tailles, se trouvent dans le répertoire MNT/`input/` fourni.

Le programme fourni consiste à calculer le remplissage des cuvettes de la carte par l'algorithme de Darboux. Cet algorithme relativement simple va calculer dans quelle direction s'écoule l'eau qui tombe au sol sur ce terrain, et donc quelles cuvettes vont se remplir (voir la figure ci-dessus). Puis il enregistre une carte modifiée (cuvettes remplies) dans le fichier passé en troisième argument du programme, ou sur la sortie standard s'il n'y a que deux arguments. Il affiche également sur la sortie standard une carte des cuvettes remplies dans ce cas, en format texte. La boucle while principale (fichier darboux.c, dernière fonction) calcule itérativement une nouvelle carte W à partir d'une carte existante Wprec, et s'arrête dès que W et Wprec sont identiques. Le calcul de W utilise en chaque point [i,j] la valeur de m[i,j] et les valeurs des 8 voisins de [i,j] dans Wprec :

W[i,j] = f(m[i,j], Wprec[i +/- 1, j +/- 1])

Travail à effectuer

Ce projet consiste à paralléliser cet algorithme de Darboux en MPI *et* en OpenMP. Vous découperez les matrices m, Wprec et W en bandes de tailles égales (ou similaires) : attention, la hauteur de la matrice n'est pas forcément divisible par le nombre de processus ou de threads.

Une fois que le programme parallèle MPI ou OpenMP fonctionne (il fonctionne s'il calcule exactement les mêmes résultats que le programme séquentiel - vérifiez que c'est le cas !), effectuez des mesures de performance de cette version de base.

Puis réfléchissez à son optimisation : les échanges des premières/dernières lignes peuvent être réalisés simultanément avec une partie du calcul. De même, si vous faites une réduction, elle peut être réalisée simultanément au calcul de l'itération suivante, et l'arrêt de la boucle principale pourra s'effectuer à l'itération suivante.

Vous utiliserez MPI et OpenMP simultanément pour exécuter des régions de code parallèles sur chaque processeur multi-coeurs et un seul processus MPI par nœud de calcul d'un cluster. Attention, par défaut `mpirun` contrôle la manière dont les coeurs sont assignés aux processus. Pour lancer un programme mpi avec un processus par machine et le maximum de threads OpenMP disponible (nombre de coeurs), utilisez la commande suivante :

```
mpirun -hostfile ... --map-by ppr:1:node ./a.out
```

Mesurez les gains de performances que vous obtenez grâce à ces optimisations sur la plateforme openstack mise à votre disposition (courant octobre).

Modalités de remise

Ce projet est à réaliser en binôme. Vous déposerez une archive (de préférence .tar.gz) contenant votre code et un rapport de quelques pages (format pdf) présentant la manière dont vous avez parallélisé ce programme et les mesures de performances que vous avez effectuées.

N'oubliez pas d'indiquer les noms des deux participants dans le rapport, et ne déposez qu'une seule archive par binôme. N'incluez pas les fichiers de données fournis (le répertoire input/), ils sont trop gros pour être déposés sur moodle !