Rapport projet IN203

Aurélien Laforge

Note : Il y a deux colonisation.cpp. Celui portant ce nom est la version avec les modifications demandées, l'autre est la version « basique » du code, auquel j'ai simplement ajouté du code permettant de calculer le temps moyen par boucle, pour mesurer la méthode OMP.

Pour la partie parallélisation en mémoire distribuée, j'ai « découpé » la galaxie par tranches horizontales.

Conflits mémoire

Dans la partie mémoire partagée, d'éventuels conflits se trouvent au niveau des variables situées dans le « private » d'omp, mais ce n'est pas un problème puisqu'elles sont protégées par le private. Ensuite, il serait possible que également que des threads calculant les valeurs pour des zones adjacentes se retrouvent à vouloir accéder à la même zone mémoire (en parcourant les valeurs des voisins dans les conditions). Cependant, il ne s'agit que d'une simple lecture dans tous les cas puisque rien n'est écrit dans « galaxy_previous » dans la fonction mise_a_jour de parametres.cpp. Ainsi, la valeur lue sera la bonne, et si la lecture ne peut se faire que pour un thread, l'autre devra simplement attendre. Dans tous les cas, au vu de la taille des sous parties, il y a peu de chance qu'un thread veuille accéder au tout début de sa zone pendant que le thread adjacent veut accéder à la fin de la sienne, puisque les threads parcourent normalement leur zone dans le même « sens ». Il faudrait qu'un thread ai subi un fort ralentissement par rapport à celui qui le précède.

Dans la partie parallélisation en mémoire distribuée, les seuls conflits qu'il pourrait y avoir se trouve au niveau des cellules fantômes, représentant la même chose qu'un autre objet, mais ce soucis est résolu par les comparaisons effectuées après réceptions des données. (Ce n'est pas réellement un problème de conflit de mémoire, plus un problème de contenu de la mémoire qui pourrait faire doublon avec des valeurs présentent ailleurs en mémoire, censées représenter la même chose.)

Calcul du Speed Up

Les calculs sont effectués sur une moyenne de 1000 prises. Toutefois, il est à noter que les valeurs présentent ici sont à prendre uniquement comme « ordre d'idée ». En effet, en faisant plusieurs tests consécutifs, certaines valeurs ont beaucoup changé, passant presque du simple au double. Les valeurs retenues sont celles qui étaient les plus « centrales » après avoir testé plusieurs fois, ou simplement la première valeur mesurée si celle ci était stable.

Version « basique » : 22.081 ms

parallélisation de boucle en mémoire partagée : 25.281 ms (environ 0,87 fois plus rapide) Pour cette version j'ai repris le code de colonisation.cpp de base, en ajoutant simplement ma « cellule » de calcul du speed up, pour éviter les interactions entre omp et MPI (n'ayant pas eu le temps de faire la combinaison des deux)

recouvrement entrée sortie via thread : 15,629 ms (environ 1,41 fois plus rapide)

recouvrement entrée sortie via MPI (2 processus) : 15.743 ms (environ fois plus rapide)

Calcul en mémoire partagée avec : - 3 processus : 15.119 ms (environ 1,46 fois plus rapide)
- 4 processus : 12.161 ms (environ 1,82 fois plus rapide)
- 5 processus : 10.716 ms (environ 2,06 fois plus rapide)
- 6 processus : 9.622 ms (environ 2,29 fois plus rapide)
- 7 processus : 17.034 ms (environ 1,30 fois plus rapide)
- 8 processus : 13.212 ms (environ 1,67 fois plus rapide)
- 9 processus : 14.171 ms (environ 1,56 fois plus rapide)

On onbserve bien un « ralentissement » du speed up à partir d'un certain nombre, avec « stabilisation » de la valeur.