TP3

Laura Martin

1 Exemple MPI

Question 1

	1º fois	1º fois	3º fois	moyenne
sinus_seq	3.6745	3.6853	3.7011	3.6869
sinus_mpi	1.3497	1.3571	1.3438	1.3502

Ce sont les données obtenus pour un code sequentiel et le même code parallélisé sur $n = 3$ processus. On attend que le temps de calcul soit la troisième partie, mais on obtient un peu plus de temps. On ne peut pas tirer de conclusions définitives avec un exemple assez petit mais commence a voir que il existent des opérations non réductibles en temps (par exemple allumer et éteindre les différents processeurs).

2 Exemple MPI-OpenMP "grain fin" et "grain grossier"

Questions 2 et 3

	1º fois	1º fois	3º fois	moyenne
mpirun -n 4 sinus_mpi	1.1554	1.0716	1.0783	1.1017
mpirun -n 2 sinus_mpi_openmp_fine	1.0940	1.0645	1.1256	1.0947
mpirun -n 2 sinus_mpi_openmp_coarse	1.0779	1.0573	1.0486	1.0612

Grosso modo, la différence entre sinus_mpi et les fichers sinus_mpi_openmp est la parallélisation des boucles avec *#pragma* dans chaque processus MPI. Après il y a plusiers façons de le faire mais ici on ne voit qu'une :

- **Case Fine :** Les boucles *#pragma* ont été mis dans les fonctions *init* et *stat*. C'est l'ordinateur qui décide comment couper les boucles.
- **Case Coarse :** Les boucles *#pragma* ont été mis dans le *main*. Pour ça, on a eu besoin de définir nous mêmes le débout et la fin du morceau du vecteur avec lequel il travail chaque thread.

On voit une petite amélioration du temps de code "coarse" par rapport au code "fine". Dans le deux cas, les threads sont allumés deux fois, donc la cause de ce différence peut se trouver dans le fait que on le donne directement la division des boucles. Ça permet faire une meilleur répartition du travail, de façon qu'on sait que tous les threads travaillent à peu près le même temps.

3 Versions hybrides MPI-OpenMP du mini-code

Questions 4 et 5

(Fichiers envoyés par mail)