

Projet multicore programming

Thibault BÈZIERS LA FOSSE, Benjamin MOREAU

28 Février 2016

1 Introduction

L'objectif de ce projet est d'implémenter une version parallélisée avec le langage *C++* de l'algorithme de branch & bound par intervalles afin d'encadrer le minimum d'une fonction de deux variables réelles.

La parallélisation du code se fera en deux étapes. Dans un premier temps, nous utiliserons *MPI* pour effectuer le calcul en coopération sur plusieurs machines. Dans un second temps, nous paralléliserons le code au sein de chaque machine à l'aide de *OpenMP*. Dans chaque partie, nous justifierons nos choix d'implémentation et présenterons les résultats obtenus.

2 Parallélisation avec *MPI*

2.1 Implémentation

2.2 résultats

3 Parallélisme à mémoire partagée avec *OpenMP*

Nous avons choisis *OpenMP* pour la parallélisation à mémoire partagée car il très facile d'utilisation. De plus, l'utilisation de *pragma* permet à l'utilisateur d'utiliser le code source sans forcément posséder la librairie *OpenMP*. Malgré sa souplesse d'utilisation, il offre de bonnes performances.

3.1 Implémentation

La parallélisation du code se fait au sein de l'algorithme de *Branch & Bound*. Dans cette fonction, si la précision n'est pas suffisante, l'espace cubique contenant la fonction est coupé en 4 sous-espaces et la fonction est rappelée récursivement sur ces sous-boîtes. Nous parallélisons donc l'appelle récursif de cette fonction. A chaque appelle de la fonction, 4 exécutions parallèles de code sont donc lancés.

Ces exécutions parallèles accèdent à une variable commune qu'il faut donc protéger : la liste de minimum courant. Nous utilisons donc un *pragma* définissant une section critique à chaque fois que la variable est modifiée au sein de la fonction parallélisée.

3.2 résultats

4 Conclusion