

Sujet : Méthode multipôle rapide pour architectures multi-cœurs.

Encadrement

Olivier Coulaud - INRIA – équipe projet Hiepacs

Olivier.Coulaud@inria.fr

Emmanuel Agullo - INRIA – équipe projet Hiepacs

Emmanuel.Agullo@inria.fr

Mots-clés : GPU; multi-cœurs ; méthode multipôle rapide (FMM), octree.

Présentation du sujet

De nombreux problèmes issus de la chimie moléculaire, de la physique des particules, de l'astrophysique,... nécessitent de calculer rapidement des interactions entre deux corps par exemple les interactions coulombiennes ou gravitationnelles. Pour calculer rapidement ces interactions, le champ est décomposé en un champ proche et un champ lointain. Cette décomposition est basée sur un octree. Le champ direct est calculé classiquement tandis que le champ lointain est approché par une méthode hiérarchique. Cette méthode conduit donc à deux algorithmes très différents en terme de calcul et de schéma de communication. Le calcul du champ lointain peut conduire à beaucoup de communications avec les 189 cellules voisines dans le cas des distributions uniformes de particules. Nous avons déjà développé un code parallèle en C++ pour calculer les interactions coulombiennes ou gravitationnelles. L'algorithme développé est basé sur un parallélisme à grain fin et a été implémenté au dessus de MPI et OpenMPI. Cette approche donne de très bons résultats sur les architectures classiques, mais pour des architectures manycores comme les GPU, la granularité des tâches n'est pas adaptée.

Une première tentative de cette approche montre que si l'on garde la même granularité de calcul que celle dans l'approche classique, les temps de calculs peuvent être très supérieurs. L'objectif de ce travail est de revoir la granularité des tâches en les grossissant pour pouvoir alimenter le CPU puis le GPU. Pour cela, après avoir fait un état de l'art sur le portage de la méthode multipôle rapide sur les GPU, nous choisirons les meilleures solutions pour le calcul champ proche puis pour le calcul du champ lointain. Cette première étape, devrait conduire rapidement à des améliorations sur l'approche en tâches de notre code ScalFMM, soit avec OpenMP (3.0), soit avec StarPU. Dans un deuxième temps, on développera les noyaux de calcul sur CUDA pour tirer complètement avantage de l'utilisation du runtime StarPU sur les machines hybrides CPU+GPU. Les travaux seront validés sur le cluster mirage (96 cores et 24 GPUs) de la plateforme PlaFRIM.

Ce stage aura lieu à Inria Bordeaux - Sud-Ouest et donnera lieu à une gratification.

Ce travail se fera en collaboration étroite avec Pierre Fortin (Lip6), Béranger Bramas (Hieapacs).

Références

Hamada, T.; Narumi, T.; Yokota, R.; Yasuoka, K.; Nitadori, K. & Taiji, M. 42 TFlops hierarchical N-body simulations on GPUs with applications in both astrophysics and turbulence, *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, ACM*, 2009, 62:1-62:12.

Darve, E.; Cecka, C. & Takahashi, T. The fast multipole method on parallel clusters, multicore processors, and graphics processing units, *Comptes Rendus Mécanique*, 2011, 339, 185-193.