Sujet : Méthode des multipoles rapides sur GPU.

Responsables : O. Coulaud, B. Bramas E. Agullo
Téléphones : 05 24 57 40 11
Courriels : Olivier.Coulaud@inria.fr, berenger.bramas@inria.fr, Emmanuel.Agullo@inria.fr

Présentation du sujet :

De nombreux problèmes issus de la chimie ou de la physique moléculaire, de la physique des particules, de l'astrophysique,... nécessitent de calculer rapidement des interactions paire à paire, par exemple les interactions coulombiennes ou gravitationnelles. Les algorithmes classiques conduisent à des complexités en $O(N^2)$ où N est le nombre de particules du système. De nouveaux algorithmes hiérarchiques, basés sur des notions d'arbre comme les *octrees*, ont été introduits pour diminuer la complexité des calculs en $O(N \ln N)$ voir même en $O(N)$ pour évaluer l'énergie ou les forces exercées sur les particules. Historiquement, le premier algorithme introduit a été l'algorithme de Barnes et Hut en 1985, puis une analyse plus mathématique réalisée par L. Greengard [1] a conduit aux méthodes multipôles rapides (FMM : *fast multipole methods, 1987)*. Depuis lors, ces méthodes se sont fortement développées dans beaucoup d'autres domaines (équations intégrales, mécanique des fluides, électromagnétisme...) et sont désormais indispensables pour traiter des problèmes de très grande taille. Le calcul du potentiel et des forces se décomposent en deux parties : 1) la contribution du champ proche où l'on calcule explicitement toutes les interactions et 2) la contribution du champ lointain. Dans cette dernière, le champ lointain est approché par différentes techniques par exemple soit par des polynômes soit par des méthodes d'interpolations.

La parallélisation de ces méthodes pour les architectures modernes comprenant à la fois de nombreux cœurs et plusieurs accélérateurs (GPU, Xeon Phi, …) est un enjeu majeur pour ces méthodes. De nombreuses approches spécifiques ont été étudiées pour tirer partie de la puissance de ces architectures. Récemment l'utilisation des runtimes (DAGuE, StarPU, …) en algèbre linéaire a conduit à des gains intéressants. Nous avons étendu cette approche pour la méthode FMM basée sur une interpolation de Chebyschev [2]. Cette implémentation utilise un runtime (StarPU) pour répartir les calculs soit sur les CPUs soit sur les GPUs. Fort de cette expérience, nous souhaitons disposer de tous les noyaux sur GPU dans notre bibliothèque ScalFMM [3].

L'objectif de ce stage est donc de développer une implémentation efficace sur GPU des différents opérateurs de la FMM classique basée sur des développements en harmoniques sphériques. Nous disposons déjà d'une implémentation sur CPU de ces opérateurs. Ces opérateurs peuvent être fortement accélérés en utilisant des matrices de rotation qui permettent de ramener la complexité en O(p^3) au lieu de O(p^4) ou p est le nombre de termes du développement. Cette optimisation devra être aussi réalisée sur GPU. Une étude devra être faite pour savoir si ces matrices de rotation devront être construites à la volée sur le GPU ou pré-calculées puis envoyées au GPU.

Ces travaux seront intégrés dans notre bibliothèque ScalFMM et une comparaison fine sera réalisée entre cette approche et la méthode basée sur l'interpolation de Chebyschev [2]. Cette comparaison se fera sur les machines hybrides (12 cœurs et 3GPU Fermi 2070 ou 2090) de PlaFRIM.

Mot-clés : méthode hiérachique, Octree, GPU, runtime, parallélisme, Cuda

Commentaires :

Le stage est rémunéré et se déroulera dans l'équipe projet INRIA HiePACS.

Références :

[1] Greengard, L. & Rokhlin, V. A fast algorithm for particle simulations, *Journal of Computational Physics,* **1987***, 73*, 325 - 348

[2] **Pipelining the Fast Multipole Method over a Runtime System**, E. Agullo, B. Bramas, O. Coulaud, E. Darve, M. Messner, T. Takahashi. RR-7981 (pdf)

[3] SCALFMM, http://scalfmm.gforge.inria.fr