Sujet : Solveur direct pour architectures multi-coeurs accélérées par des GPUs

Responsables : Emmanuel Agullo (INRIA/LaBRI), Alfredo Buttari (CNRS/ENSEEIHT/IRIT), Abdou Guermouche (U.B.1/INRIA/LaBRI)
Téléphones : 05 24 57 41 50 - 05 24 57 41 03 - 05 34 32 22 08
Courriels : Emmanuel.Agullo@inria.fr, Abdou.Guermouche@labri.fr, alfredo.buttari@enseeiht.fr

Présentation du sujet :

Mot-clés : GPU ; multi-coeur ; solveur creux ; factorisation QR ; plate-forme hétérogène ; méthode multifrontale.

Durant ces cinq dernières années, l'intérêt de la communauté de calcul scientifique pour les cartes d'accélération graphiques (GPUs) s'est fortement accru. La raison principale de cette engouement tient à la formidable capacité de calcul de ces composants, initialement pensés et mis au point pour effectuer les opérations de calcul graphique et de traitement d'image. Un effort important de recherche a été mené pour porter les codes numériques sur de tels composants, contribuant ainsi au développement significatif de librairies scientifiques de qualité industrielle ainsi que de modèles de programmation pour GPUs (GPGPU). Dans la même période, l'élaboration des cartes graphiques s'est fortement adaptée aux besoins de la communauté du calcul scientifique si bien que les GPUs modernes peuvent désormais exécuter les opérations arithmétiques à virgule flottante en précision double à des vitesses qui dépassent celles des processeurs (CPUs) standards jusqu'à un facteur 30.

La communauté d'algèbre linéaire dense a été pionnière dans l'utilisation des GPUs à des fins scientifiques; le projet MAGMA de l'Université du Tennessee, dont l'un des encadrant est co-auteur, peut par exemple être cité parmi les librairies ayant réussi à exploiter efficacement les GPUs. Les algorithmes d'algèbre linéaire dense courants sont en effet extrêmement riches en calcul et ont un schéma d'accès aux données très régulier, faisant d'eux d'excellents candidats à l'utilisation de GPUs. Au contraire, les algorithmes d'algèbre linéaire creuse ont ordinairement des patrons d'accès irréguliers et indirects rendant extrêmement difficile l'exploitation efficace du potentiel des GPUs. Ces méthodes, souvent employées au coeur des simulations numériques à grande échelle, peuvent être classées en deux familles :

- **méthodes itératives :** dans leur version basique, non préconditionnée, ces méthodes sont naturellement parallélisables. Pour cette raison, leur port sur GPUs a été l'objet de nombreuses recherches fructueuses. Néanmoins, le manque de précondionneurs robustes, extrêmement compliqués à implanter sur GPU, rend les solveurs itératifs sur ce type d'architecture généralement trop peu robustes pour le moment. De surcroît, les méthodes itératives sont basées sur des opérations, telles que le produit matrice-vecteur creux, caractérisées par un taux calcul/communication très faible, limitant leur performance.
- **méthodes directes :** dans les algorithmes appartenant à cette famille, tels que les factorisations de matrices creuses, les calculs sont scindés en termes d'opérations sur des matrices denses, riches en opérations à virgule flottante. Il est alors possible de mettre en oeuvre une parallélisation multithreadée massive et donc d'effectuer un portage efficace sur GPUs. Néanmoins, le patron d'accès aux données des méthodes directes est extrêmement complexe. Leur implantation haute-performance demeure donc un véritable challenge.

Le but de ce projet de fin d'étude est d'étudier et mettre au point les algorithmes et modèles de programmation parallèles permettant d'implémenter des méthodes directes de résolution de systèmes linéaires creux sur les plate-formes de calculs émergentes équipées processeurs multi-coeurs accélérés par des GPUs. Plusieurs tentatives ont été accomplies afin de porter ces méthodes sur de telles architectures. Les approches proposées jusqu'à présent ont essentiellement consisté à déporter

une partie des tâches de calculs (celle à gros grain) sur les GPUs et optimiser soigneusement le code afin d'accroître la performance. Ce stage propose une approche innovante qui se base sur l'efficacité et la portabilité des moteurs d'exécutions, tel que StarPU développé au LaBRI (Bordeaux). Les méthodes directes devront être repensées et implantées en conséquence, afin d'obtenir des propriétés rendant efficaces leur exécution sur des machines hétérogènes efficaces. Cela peut nécessiter le développement de nouvelles méthodes et de nouveaux schémas d'accès aux données qui conviennent mieux à l'ordonnancement dynamique des tâches de calcul sur des unités d'exécution de capacités fortement hétérogènes. L'efficacité, la fiabilité et la robustesse du solveur seront évalués sur des matrices issues de simulations numériques académiques et industrielles à grande échelle.

Commentaires :

Ce stage pourra se poursuivre dans le cadre d'un doctorat.

Références :

A. Buttari.
Fine-grained multithreading for the multifrontal QR factorization of sparse matrices.
2011. Submitted to SIAM SISC and APO technical report number RT-APO-11-6 [PDF].

E. Agullo, C. Augonnet, J. Dongarra, M. Faverge, H. Ltaief, S. Thibault, and S. Tomov.
QR Factorization on a Multicore Node Enhanced with Multiple GPU Accelerators.
Proceedings of the IPDPS 2011 International Conference. [PDF].