

Sujet : Memory-aware algorithms for task-based sparse direct solver

Responsables : Mathieu Faverge, Abdou Guermouche, Pierre Ramet, Jean Roman

Téléphone : +33 5 24 57 40 73

Courriel :

Mathieu.Faverge@inria.fr

Abdou.Guermouche@inria.fr

Pierre.Ramet@inria.fr

Jean.Roman@inria.fr

Présentation du sujet :

Du fait des quantités considérables de données et de calculs mises en jeu, la résolution des simulations numériques en vraie grandeur imaginées aujourd'hui par les chercheurs n'est réalisable qu'en ayant recours au calcul parallèle haute performance au sens large. Dans la plupart des simulations numériques utilisant un schéma implicite, l'étape la plus consommatrice en ressources (CPU et mémoire) est la résolution de systèmes linéaires creux de très grandes tailles. En plus de la difficulté numérique inhérente à certains modèles physiques employés, l'augmentation de la taille des systèmes linéaires conduit même les méthodes réputées scalables, telles que les méthodes itératives classiques, à ne plus converger de manière efficace. Nous avons déjà développé depuis de nombreuses années des solveurs linéaires creux parallèles, utilisant tant des méthodes directes qu'itératives. Ces solveurs, d'une grande robustesse, ont déjà permis de traiter des problèmes de très grandes tailles.

Le solveur PaStiX intègre une gestion fine et efficace des multiples niveaux de parallélisme exploitables dans un solveur linéaire creux. En effet, les nouvelles architectures de calcul intensif intègrent de plus en plus de microprocesseurs qui eux-mêmes intègrent un nombre croissant de cœurs de calcul accompagnés d'accélérateurs de type GPU. Afin d'exploiter cette hétérogénéité, des supports d'exécution génériques ont été proposés afin de simplifier le travail du développeur. Ces supports d'exécution reposent sur la représentation d'un algorithme sous la forme d'un graphe de tâches. Récemment, nous avons pu intégrer et évaluer les performances du solveur PaStiX sur deux supports d'exécution :

- StarPU qui utilise un modèle de découverte dynamique des dépendances en parcourant linéairement le graphe;
- PaRSEC qui utilise un modèle de graphe paramétré pour décrire algébriquement les dépendances de données.

Nous avons également montré récemment sur le support d'exécution PaRSEC que les ordonnancements à bases de tâches dites 1D montrent leur limites sur des architectures avec de grands nombres de cœurs, et éventuellement hétérogènes avec la présence de GPUs. Les schémas à base de tâches 2D permettent un grain de parallélisme plus fin et d'alimenter plus facilement l'ensemble des ressources disponibles permettant une meilleure scalabilité et une meilleure utilisation de ces ressources. Une des limites de ces schémas 2D est la création d'un très grand

nombre de tâches, de l'ordre de plusieurs dizaines de millions, qui saturent les systèmes d'ordonnement.

L'objet du travail proposé est de contraindre les algorithmes d'ordonnement afin de prendre en compte la limitation de la mémoire du calculateur. En effet, indépendamment de la gestion d'un grand nombre de tâches générées, le schéma 2D engendre un surcout mémoire très important qui doit être maîtrisé. L'écriture du code sous la forme d'un graphe de tâches permet d'hybrider plus facilement les algorithmes proposés dans la littérature. On identifie deux étapes dans le développement de ce travail:

- Le premier travail s'intéressera à la manière de parcourir l'arbre d'élimination pour minimiser le pic mémoire algorithmique tout en maintenant un bon niveau de parallélisme. Ce parcours devra guider l'ordonnement calculé dynamiquement par le support d'exécution.
- Le second travail consistera à réaliser une étude de l'approche Fan-Both introduite par C. Ashcraft. Cette généralisation des méthodes Fan-Out et Fan-In, classiquement implémentées dans les solveurs directs creux utilisant une formulation supernodale, permet la mise en oeuvre d'un compromis intéressant entre stockage mémoire et volume de communication. Une approche hybride supernodale-multifrontale pourra également être envisagée suivant l'avancement de cette étude.

Mot-clés : Algèbre linéaire creuse, calcul haute performance, modèles de programmation, DAG, architectures hétérogènes.

Commentaires : Ce sujet pourra être poursuivi dans le cadre d'une thèse.

Références :

[1]

C. Ashcraft and A. George and J. Gilbert and J. Liu.
The Fan-Both Family of Column-Based Distributed Cholesky Factorization Algorithms.
In *Graph Theory and Sparse Matrix Computation*, Springer 1993.

[2]

P. Hénon, P. Ramet, and J. Roman.
PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems.
Parallel Computing, 28(2):301-321, January 2002.

[3]

X. Lacoste, M. Faverge, P. Ramet, S. Thibault, G. Bosilca.
Taking advantage of hybrid systems of sparse direct solvers via task-based runtimes.
HCW'2014 workshop of IPDPS, May 2014.