Sujet : Hierarchical QR factorization on STF programming model

Responsables : Mathieu Faverge
Téléphone : +33 5 24 57 40 73
Courriel :
Mathieu.Faverge@inria.fr

Présentation du sujet :

Le développement d'algorithmes par blocs a vu l'apparition de nouvelles techniques pour la factorisation QR des matrices. Cette nouvelle technique de factorisation QR permet de mieux entrelacer les différentes étapes de la factorisation afin de permettre une meilleure parallélisation. En effet, l'algorithme d'origine utilisé dans la bibliothèque LAPACK à base de parallélisme fork-join ne permet pas de passer facilement à l'échelle.

L'algorithme par tile par défaut est celui dit Flat-TS car il l'utilise un algorithme qui orthogonalise les blocs lignes par lignes. Il est ainsi possible de commencer la deuxième étape de la factorisation dés que la première étape à terminé avec la deuxième ligne de blocs. Cet algorithme est implémenté dans les bibliothèques PLASMA, CHAMELEON, et DPLASMA. Dans le cadre des matrices grandes et fines, le parallélisme de cet algorithme est malheureusement limité par le nombre de blocs de colonnes disponibles. Une évolution de cet algorithme dit TSQR ou communication-avoiding QR a été développé pour créer du parallélisme en découpant le problème horizontalement en plusieurs sous-matrices que l'on factorise indépendemment. La première ligne de bloc de chacune des ces sous-matrices sont ensuite factorisées ensembles en suivant un arbre binomial. Le parallélisme est ainsi multiplié par le nombre de sous-problèmes créé. Cet algorithme est disponible dans plusieurs bibliothèque pour accélérer la factorisation de matrices grandes et fines. Basé sur cette idée, nous avons proposé dans le cadre de la bibliothèque DPLASMA, l'algorithme Hierarchical QR, ou HQR, pour prendre en compte de manière hiérarchique à la fois la forme du problème considéré et l'architecture visée. Cet algorithme se base sur une structure d'arbre à plusieurs niveaux : le niveau le plus bas exploite l'algorithme Flat-TS décrit plus haut pour sa meilleure utilisation des ressources de calcul, un deuxième niveau d'arbre permet de créer du parallélisme localement à chaque noeud de mémoire partagée, enfin le troisième niveau permet de créer du parallélisme entre les noeuds en mémoire distribuée. Il a été prouvé théoriquement et expérimentalement que cette hiérarchie d'arbre permet d'accélérer la factorisation QR sur toutes les configurations de matrices et d'architecture, mais également l'algorithme de réduction de matrice générale à matrice bande qui est utilisé dans le calcul des valeurs singulières d'un problème.

L'objet du travail proposé sera l'extraction des fonctionnalités de gestion d'arbres de la bibliothèque DPLASMA dans une bibliothèque indépendante pour pouvoir être exploitée de façon similaire dans les bibliothèques DPLASMA et Chameleon. On identifie plusieurs étapes dans le développement de ce travail:

- Extraire dans une bibliothèque indépendante les fonctionnalité de création d'arbres adapté à une structure de matrice données et une architecture donnée. Cette extraction donnera lieu à

une étude des performances des fonctionnalités de la bibliothèques en la comparant à une nouvelle implémentation *statique* de la bibliothèque.

- Développer une version de l'algorithme HQR dans la bibliothèque Chameleon en respectant le modèle de programmation par flot de tâches séquentiel.
- Étude de scalabilité et comparative entre les deux iplémentations.

Mot-clés : Algèbre linéaire dense, calcul haute performance, modèles de programmation, DAG, arbres.

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

Références :

[1]
A. Buttari, J. Langou, J. Kurzak, J. Dongarra (2009).
A class of parallel tiled linear algebra algorithms for multicore architectures.
In *Parallel Computing, 35(1), 38-53, 2009*.

[2]
J. Dongarra, M. Faverge, T. Hérault, M. Jacquelin, J. Langou, Y. Rovert.
Hierarchical QR factorization algorithms for multi-core clusters.
Parallel Computing, Elsevier, 2013, 39 (4-5), pp.212-232. hal link

[3]
M. Faverge, J. Langou, Y. Robert, J. Dongarra.
Bidiagonalization with Parallel Tiled Algorithms.
RR-8969, INRIA. 2016. hal link