Titre :

Algorithmes parallèles de partitionnement de graphes pour les architectures exaflopiques

Parallel graph partitioning algorithms for exaflopic architectures

Encadrant :

François PELLEGRINI, Pr

francois.pellegrini@labri.fr

LaBRI et INRIA Bordeaux Sud-Ouest
Université Bordeaux 1
351, cours de la Libération
33405 Talence cedex

Mots clés :

Architectures hétérogènes, NUMA, Placement statique, Repartitionnement dynamique

Heterogeneous architectures, NUMA, Static mapping, Dynamic repartitioning

Description du sujet :

Le partitionnement de graphes est une technique utilisée pour la résolution de nombreux problèmes en calcul scientifique, tels que la décomposition d'un maillage en domaines afin d'équilibrer la charge de calcul sur une architecture parallèle. Du fait de la taille sans cesse croissante des maillages à manipuler, les outils de partitionnement eux-mêmes ont dû être parallélisés. Les versions parallèles de ces outils (comme par exemple l'outil PT-Scotch que nous développons) fournissent de bons résultats pour et sur plusieurs milliers de processeurs, mais l'arrivée d'architectures comprenant plus d'un million d'unités de traitement pose de nouveaux défis.

Tout d'abord, les résultats de partitionnement produits par ces logiciels doivent prendre en compte l'hétérogénéité de ces architectures, en calculant des placements des processus sur les processeurs, à la place de simples partitions, de façon à réduire le coût de communication global en privilégiant la communication locale par rapport à la communication distante. De plus, l'exécution efficace des logiciels de partitionnement sur ces architectures nécessite des algorithmes bine plus sophistiqués pour prendre en compte cette hétérogénéité au sein des algorithmes eux-mêmes.

Le but de cette thèse est de résoudre les défis algorithmiques et d'implémentation permettant de réaliser un outil capable de partitionner et repartitionner dynamiquement des graphes comprenant jusqu'à un billion (c'est-à-dire 10^{12}) de sommets, distribués sur un million d'unités de traitement.

En particulier, les machines hétérogènes de grande taille peuvent poser des problèmes de synchronisme, qui doivent être évalués. La plupart des algorithmes existants utilisent des échanges entre voisins de type « halo », c'est-à-dire une sorte de communication all-to-all synchrone entre sommets voisins localisés sur des processeurs voisins, ainsi que des réductions parallèles, pour informer tous les processeurs du résultat d'un calcul distribué. Avec l'émergence de machines comprenant plusieurs centaines de milliers d'éléments de

traitement, et en dépit de l'amélioration continuelle des sous-systèmes de communication, le besoin de plus d'asynchronisme dans les algorithmes parallèles est voué à augmenter. De nouveaux algorithmes, vraisemblablement hiérarchiques, doivent être étudiés.

———

Graph partitioning is a technique used for the solving of many problems in scientific computing, such as the decomposition of a mesh into domains so as to evenly balance the compute load on the processors of a parallel architecture. Because of the ever increasing size of the meshes to handle, partitioning tools themselves had to be parallelized. The parallel versions of these software (such as the PT-Scotch software that we are developing) provide good results for and on several thousands of processors, but the advent of architectures comprising more than a million processing elements raises new concerns.

First, the partitioning results produced by these software have to take into account the heterogeneity of these architectures, by computing process-processor mappings instead of plain partitions, so as to reduce overall communication cost by privileging local communication over remote communication. Moreover, the efficient execution of the partitioning software on these architectures requires much more sophisticated algorithms to account for this heterogeneity in the algorithms themselves.

The purpose of this thesis is to tackle algorithmic and implementation challenges leading to a software tool able to partition and dynamically repartition graphs comprising a up to a trillion (that is, 10^{12}) vertices, distributed across a million processing elements.

In particular, large heterogeneous machines may pose synchronicity problems, which have to be evaluated. Most existing algorithms make use of halo exchanges, that is, some form of synchronous all-to-all communication between neighbor vertices borne by different processors, as well as of parallel reduction, to inform all processors of the result of a distributed computation. With the advent of machines comprising several hundred thousands of processing elements, and in spite of the continuous improvement of communication subsystems, the demand for more asynchronicity in parallel algorithms is likely to increase. New, possibly hierarchical, algorithms, have to be investigated.