Programmation des architectures hétérogènes à l'aide de tâches divisibles

Raymond Namyst
INRIA « Runtime » group
Phone: 05 24 57 40 34
email: Raymond.Namyst@labri.fr

Mots clés : Calcul parallèle, Accélérateur, GPU, multicoeur, StarPU.

Contexte et objectifs du travail

Les ordinateurs multicoeurs équipés d'accélérateurs réalisent une percée remarquable dans le paysage du calcul haute performance. Parmi les machines parallèles les plus puissantes au monde (selon le classement www.top500.org), une sur deux est équipée d'accélérateurs. Cette récente évolution vers des architectures hétérogènes a entraîné un regain d'efforts de recherche visant à concevoir des outils permettant de programmer facilement des applications capables d'exploiter efficacement toutes les unités de calcul de ces machines.

Le support d'exécution StarPU, développé dans l'équipe Runtime, a été conçu pour servir de cible à des compilateurs de langages parallèles et des bibliothèques spécialisées (algèbre linéaire, développements de fourier, etc.) La fonction principale de StarPU est d'ordonnancer des graphes dynamiques de tâches de manière efficace sur l'ensemble des ressources hétérogènes de la machine. Pour ce faire, le support s'appuie sur des modèles adaptatifs de prédiction de coût des calculs et des transferts de données, ainsi que sur une mémoire virtuellement partagée destinée à minimiser les mouvements de données entre les différentes mémoires. StarPU est avant tout une plateforme pour expérimenter de nouvelles stratégies d'ordonnancement, celles-ci pouvant aisément être construites en redéfinissant les fonctions appelées en réaction à certains événements (nouvelle tâche prête, unité de calcul oisive, etc.) StarPU a récemment été utilisé avec succès pour l'implémentation d'algorithmes parallèles en algèbre linéaire sur configurations multi-GPU, en collaboration avec d'autres équipes françaises et étrangères.

L'un des aspects les plus difficiles, lorsque du découpage d'une application en graphe de tâches, est de choisir la granularité de ce découpage, qui va typiquement de pair avec la taille des blocs utilisés pour partitionner les données du problème. Les granularités trop petites ne permettent pas d'exploiter efficacement les accélérateurs de type GPU, qui ont besoin de mettre en oeuvre un parallélisme massif pour « tourner à plein régime ». À l'inverse, les processeurs traditionnels exhibent souvent des performances optimales à des granularités beaucoup plus fines. Le choix du découpage est donc non seulement difficile, mais il a en outre une influence sur la quantité de parallélisme disponible dans le système : trop de petites tâches risque d'inonder le système en introduisant un surcoût inutile, alors que peu de grosses tâches risque d'aboutir à un déficit de

parallélisme. Fixer un découpage manuellement demande donc de nombreux tâtonnements avant de trouver le bon compromis.

L'objectif de ce travail de recherche est d'introduire dans StarPU la notion de tâches divisibles, c'est-à-dire de tâches que le support d'exécution pourra décider (ou non) de redécouper en plusieurs sous-tâches à l'exécution, en fonction de différents critères tels que la quantité de parallélisme que l'on souhaite générer, l'opportunité d'exploiter certains types d'unités de calcul à un moment donné, etc. Une grande partie du travail consistera à étudier comment gérer efficacement une partition non uniforme des données (pour autoriser la co-existence de sous-données de différentes granularités) ainsi qu'une gestion des dépendances entre tâches s'adaptant à des raffinements locaux du graphe de tâches.

On s'intéressera dans un premier temps à une version simple du découpage, où le support d'exécution sait à l'avance en combien de sous-tâches il peut découper chaque tâche (voire même contrôler leur taille), ce qui permettra de se concentrer dans un premier temps sur l'ordonnancement, la calibration de la stratégie de découpage et la gestion des granularités multiples.

Bibliographie

- Cédric Augonnet, Samuel Thibault, Raymond Namyst, and Pierre-André Wacrenier. StarPU : A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. In Proceedings of the 15th International Euro-Par Conference, volume 5704 of Lecture Notes in Computer Science, Delft, The Netherlands, pages 863-874, August 2009.
- Emmanuel Agullo, Cédric Augonnet, Jack Dongarra, Mathieu Faverge, Hatem Ltaief, Samuel Thibault, and Stanimire Tomov. QR Factorization on a Multicore Node Enhanced with Multiple GPU Accelerators. In 25th IEEE International Parallel & Distributed Processing Symposium (IEEE IPDPS 2011), Anchorage, Alaska, USA, May 2011.

Liens utiles

- Équipe Runtime : runtime.bordeaux.inria.fr
- Logiciel StarPU : runtime.bordeaux.inria.fr/starpu

Détails administratifs

- Lieu du stage : INRIA, bâtiment A29 bis.
- Financement : INRIA (gratification d'environ 400€ par mois)