Sujet : Étude et évaluation de la prise en charge des échanges de données inter-noeuds dans un support d'exécution à base de tâches, à l'aide d'une interface de communication spécialisée

Responsable (contact) : Olivier Aumage, Inria Bordeaux - Sud-Ouest, 200 avenue de la vieille tour, 33405 Talence Cedex
Téléphone : 05 24 57 41 19
Courriel : olivier.aumage@inria.fr

Encadrant : Emmanuel Agullo, Inria Bordeaux
Téléphone : 05 24 57 41 50
Courriel : emmanuel.agullo@inria.fr

Encadrant : Aurélien Esnard, Inria Bordeaux
Téléphone : 05 24 57 41 08
Courriel : aurelien.esnard@inria.fr

Encadrant : Samuel Thibault, Inria Bordeaux
Téléphone : 05 24 57 41 43
Courriel : samuel.thibault@labri.fr

Présentation du sujet :

Contexte

L'équipe STORM (Inria/LaBRI) travaille à la conception de supports d'exécution d'applications scientifiques sur des machines de calcul parallèles et distribuées, dans le domaine que l'on nomme calcul intensif. Le but fondamental de ces supports d'exécution est d'assurer la portabilité des performances des applications en permettant à celles-ci de maximiser l'exploitation du potentiel de performances des machines, tout en minimisant ou en éliminant l'effort d'adaptation nécessaire à l'utilisation de telles applications sur une machine donnée.
Le support exécutif StarPU, développé par l'équipe STORM, fournit un cadre d'ordonnancement de tâches sur des plates-formes hétérogènes et une API permettant d'implémenter diverses classes d'algorithmes d'ordonnancement. Ce cadre d'ordonnancement travaille en coopération avec un gestionnaire de mémoire virtuellement partagée assurant l'élimination des transferts de données redondants et les recouvrements calculs/communications.
StarPU implémente le modèle de programmation STF (sequential task flow), selon lequel les tâches sont soumises dans l'ordre séquentiel de l'algorithme applicatif puis exécutées en parallèle par le moteur d'exécution de StarPU. Dans le cas d'une session impliquant plusieurs noeuds, le modèle STF est conservé et étendu à l'ensemble de la session: un unique graphe de tâches est soumis de manière identique sur l'ensemble des noeuds. StarPU gère automatiquement les transferts de données inter noeuds pour satisfaire les dépendances entre tâches. La distribution

initiale des données reste cependant à la charge du code applicatif. Par ailleurs, StarPU ne prend pas, à l'heure actuelle, l'initiative d'altérer cette distribution de données. Il revient donc au code applicatif de détecter ou anticiper d'éventuels déséquilibres, et d'effectuer des redistributions à bon escient.
Or, les informations collectées par StarPU notamment sur la progression de l'exécution et le volume d'échanges de données pourraient être mises à profit pour prendre déclencher automatiquement ces opérations de redistribution, ou fournir des éléments de prise de décision au code applicatif pour en faciliter le travail.

Travail demandé

En s'appuyant initialement sur une application synthétique, présentant un déséquilibre de charge maitrisé, il s'agit de faire évoluer l'interface distribuée de StarPU de façon à détecter un déséquilibre de charge induit par la distribution de donnée en vigueur et déterminer les mesures correctives à appliquer. Dans une optique de passage à l'échelle, il est indispensable d'éviter tout point de contention, qui s'avèrerait rapidement prohibitif. C'est notamment la raison pour laquelle le graphe de tâche est soumis sur chaque noeud, au lieu d'utiliser un noeud maitre qui piloterait l'ensemble de la session. Il est donc essentiel, pour préserver ces propriétés de passage à l'échelle de StarPU, que les opérations de détection de déséquilibre de charge et de redistribution des données fonctionnent de manière locale, décentralisée.

Mot-clés : redistribution, parallélisme, tâche, communication, calcul distribué

Commentaires :
Une bonne maîtrise de la programmation parallèle et de la programmation système est souhaitable pour aborder le sujet dans de bonnes conditions.

Références :

- Équipe Storm: https://team.inria.fr/storm
- Équipe HiePACS: https://team.inria.fr/hiepacs
- StarPU: http://starpu.gforge.inria.fr

[1] Cédric Augonnet, Samuel Thibault, Raymond Namyst, and Pierre-André Wacrenier. StarPU : A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. Concurrency and Computation : Practice and Experience, Special Issue : Euro-Par 2009, 2011.
[2] Matthias Lieber, Kerstin Gößner, and Wolfgang E. Nagel. The potential of diffusive load balancing at large scale. In 23rd European MPI Users' Group Meeting, 2016.
[3] Kirk Schloegel, George Karypis, and Vipin Kumar. Wavefront diffusion and LMSR: algorithms for dynamic repartitioning of adaptive meshes. Transactions on Parallel and Distributed Systems, 2001.
[4] James D. Teresco, Karen D. Devine, Joseph E. Flaherty, Are Magnus Bruaset, and Tveito Aslak. Partitioning and dynamic load balancing for the numerical solution of partial differential equations. In Numerical Solution of Partial Differential Equations on Parallel Computers, 2006.