

Modèles de programmation pour les architectures distribuées hétérogènes

Sujet de thèse

Mars 2011

Équipe/thème: Équipe SATANAS/RUNTIME

Directeurs: Olivier Aumage, Samuel Thibault

Directeur habilité: Raymond Namyst

Contact:

- Olivier Aumage 05 24 57 41 19, olivier.aumage@inria.fr
- Samuel Thibault 05 24 57 41 43, samuel.thibault@labri.fr
- Raymond Namyst 05 24 57 40 34, raymond.namyst@labri.fr

Mots-clés: ordonnancement, support exécutif, réseau, multi-coeur, langages à directives, calcul intensif

Contexte

L'équipe Runtime (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 conçu et développé au sein de Runtime dans le cadre de la thèse de Cédric Augonnet est destiné plus particulièrement à l'exploitation des architectures *hétérogènes*, c'est-à-dire les architectures comportant des unités de calcul généralistes (processeurs multicœurs habituels) associées à des unités de calcul accélératrices (processeurs graphiques/GPU, notamment). StarPU offre une interface de programmation à base de tâches élémentaires potentiellement parallèles et de relations de dépendances explicites ou implicites entre ces tâches. Le rôle de StarPU est ensuite principalement d'assurer l'ordonnancement de ces tâches élémentaires de calcul en prenant les décisions d'affecter telle ou telle tâche sur telle ou telle unité de calcul, et ceci en fonction de divers paramètres tels que les dépendances, les coûts estimés des différentes solutions, etc. StarPU s'occupe également des travaux afférents tels que les transferts de données vers ou depuis les mémoires embarquées des unités accélératrices.

Problématique

Le domaine du calcul intensif est actif depuis plusieurs décennies. Pour les nombreux codes scientifiques existant, souvent conçus au prix d'efforts de développement considérables, le travail d'adaptation à des modèles de programmation de type tâches tels que l'interface de StarPU peut nécessiter d'importantes modifications structurelles, voire une réécriture intégrale prohibitive. De plus, l'évolution rapide du matériel constituant les plateformes de calcul rend la pérennité d'un tel effort particulièrement incertaine.

Afin d'atténuer l'impact de telles évolutions sur le code applicatif, les extensions de langages à base de directives de compilation sont une piste séduisante et le succès d'OpenMP en réponse à l'apparition des processeurs multicœurs est là pour le démontrer. L'intérêt immédiat réside en une intrusivité relativement limitée : un programme OpenMP, par exemple, reste généralement valide lorsque les directives sont ignorées.

Au-delà, cependant, l'avantage d'une telle approche est de déplacer une partie de l'effort d'adaptation depuis l'application vers le couple formé du compilateur et de son support exécutif. Le bénéfice de cette coopération compilation/support est double. En premier lieu, elle permet d'améliorer l'objectif de portabilité des performances en introduisant une indirection dynamique entre le code applicatif et l'interface de programmation sous-jacente, réduisant ainsi l'empreinte générale dans le code scientifique parallélisé et, en second lieu, elle permet également d'améliorer l'objectif d'efficacité en permettant la récolte et l'exploitation automatisée d'informations structurelles sur l'application.

Travail demandé

En s'appuyant sur l'expérience acquise par l'équipe Runtime dans le développement de supports exécutifs tels que StarPU et ForestGOMP (OpenMP), d'une part et d'autre part sur les efforts de la communauté dans le domaine, notamment le langage XcalableMP développé à l'université de Tsukuba pour la programmation distribuée sur des clusters de calcul et le langage StarSS développé au Barcelona Supercomputing Center, il s'agit de proposer un modèle de programmation unifié permettant d'exploiter efficacement les architectures distribuées hétérogènes associant des processeurs multicœurs généralistes et des accélérateurs spécialisés reliés par des réseaux rapides. Ce modèle de programmation devra proposer le niveau d'expressivité adéquat afin de maximiser le potentiel d'optimisation à la compilation et à l'exécution tout en préservant la portabilité applicative.

