TITRE : Placement d'applications hybrides sur machine non-uniformes multicoeurs

COURRIELS :emmanuel.jeannot@labri.fr,mercier@labri.fr,namyst@labri.fr
DIRECTEURS: Emmanuel Jeannot, Guillaume Mercier, Raymond Namyst

Contexte

Dans de nombreux domaines scientifiques et d'ingénierie, il est nécessaire de résoudre des problèmes complexes réclamant d'énormes besoins en calculs. Dans ce but, on a utilisé des ordinateurs parallèles homogènes. Cependant, les récentes avancées en architectures font que les calculateurs haute performance sont de plus en plus hiérarchiques et hétérogènes (ex: une grappe de nœuds multiprocesseurs avec plusieurs cœurs, interconnectés avec différentes technologies de réseau haut débit). À l'avenir les machines parallèles auront couramment des milliers de nœuds composés de processeurs de plusieurs dizaines de cœurs. Du fait de leur hétérogénéité et de leur hiérarchie ces infrastructures programmer efficacement et simplement de tels environnements est un des défis majeur du calcul haute performance.

En effet, à l'heure actuelle, aucune des solutions et des technologies disponibles ne permet de se rapprocher de manière satisfaisante des performances optimales de ces environnements. D'une part, les environnements à base de passage de messages (ex. MPI) ne sont pas adaptés à la granularité fine des cœurs. D'autre part, les environnements à base de threads (ex. OpenMP) sont bien adaptés pour les environnements à mémoire partagée (et donc pour les processeurs multicœurs), mais sont limités dès que les échanges de donnée doivent passer par un réseau. Une solution intermédiaire, appelée programmation hybride, mixant la puissance des environnements à passage de messages pour les échanges inter-nœuds, et à base de threads pour l'exécution sur un processeur multicœur est très prometteuse.

Au sein de l'équipe Satanas du LaBRI et en particulier de l'EPI INRIA Runtime, de nombreux outils et techniques existent ou sont en cours de développement. En particulier, la bibliothèque HWLOC permet d'obtenir la topologie d'un environnement cible. D'autre part, l'équipe travaille depuis de nombreuses années sur les aspects MPI (collaboration avec MPICH2) et OpenMP/thread (bibliothèque Marcel). Enfin, la dimension compilation et approche statique a été renforcée par l'arrivée de Denis Barthou. La thèse décrite ici, se basera sur ces différents apports et visera à les enrichir en permettant, par exemple une validation de ces modèles ou l'intégration des solutions proposées dans ces outils.

Sujet

L'objectif de cette thèse est de travailler sur les aspects placement des processus et des threads pour les applications haute-performance. En effet, le placement par défaut proposé par MPI est bien adapté pour les architectures à plat (sans hiérarchie), mais pose des problèmes important dans le cas hiérarchique car il ne tient pas compte de l'affinité des processus entre eux.

Pour traiter ce problème, nous proposons le plan de travail suivant:

1. **Approche algorithmique** : Cette première partie a pour but d'étudier des stratégies de placement efficaces. À partir d'un modèle d'application et d'architecture, il s'agit mettre en place des stratégies qui affectent les processus aux cœurs. Le problème étant par nature très combinatoire l'utilisation de la structure du problème (hiérarchie, symétrie, etc.) est nécessaire pour obtenir des approches garanties voir optimales.
2. **Approche hybride**: L'objectif est de combiner les modèles à base de processus et ceux à base threads. Placer efficacement les threads sur les cœurs permet d'optimiser les accès mémoire mais nécessite de bien contrôler le grain de l'application. Cela implique d'enrichir le modèle

avec les aspects mémoire et d'améliorer les stratégies de placement en incluant un nouveau niveau dans la hiérarchie de l'application.

3. **Approche mixte** : Ici, il s'agit d'intégrer les informations issuent d'une analyse statique du code source effectuée à la compilation. L'objectif est double. Premièrement, de telles informations permettront d'améliorer le placement et de s'affranchir de la collecte de trace. Deuxièmement, Ces informations seront utiles pour construire des modèles paramétriques, les seuls permettant de passer à l'échelle.
4. **Approche dynamique** : Cette dernière partie consiste à mettre en place des outils et des algorithmes dynamiques permettant, à l'exécution, d'adapter le placement pour prendre en compte les imperfections de celui-ci dues aux limites des modèles et aux variations des conditions d'exécutions.