Gestion des communications sur grappes de noeuds massivement multi-coeurs

Responsable: Alexandre DENIS

lieu: INRIA/LaBRI

téléphone: 05.24.57.41.14

e-mail: `Alexandre.Denis@labri.fr`

équipe: Satanas, projet: Runtime

Mots-clés: communications réseau, multi-coeur, programmation hybride.

L'émergence des processeurs multi-coeurs a radicalement changé la physionomie des machines courantes, ce qui a eu des repercussions sur les grappes de calcul. Alors qu'il était typique auparavant qu'une grappe soit constituée de noeuds à deux processeurs, la tendance actuelle se porte plutôt vers des noeuds dotés de deux ou quatre processeurs, dotés chacun de quatre à huit coeurs, pour un total jusqu'à trente-deux coeurs par noeud. Cette tendance à l'augmentation du nombre de coeurs par noeud est nette et semble durable.

Dans le même temps, d'autres type d'architectures massivement multi-coeur font leur apparition, comme l'architecture MIC (*Many Integrated Core*) d'Intel. Cette architecture est massivement multi-coeur — plus de 50 coeurs par puce — et distribuée. Elle est en effet implémentée sous forme d'accélérateurs disposant de leur propre mémoire, dans un espace d'adressage distinct des coeurs des processeurs principaux.

Cette révolution du matériel a des répercussions à plusieurs niveaux, en particulier sur le modèle de programmation et sur la gestion des communications.

Le modèle de programmation qui était *standard* jusque là, à savoir une programmation du parallélisme au travers de la seule interface MPI (Message Passing Interface) avec un processus de calcul par coeur, n'est plus adapté. Il conduirait à de très nombreux processus par noeud, entrainant une important consommation mémoire et un échange de nombreux messages MPI au sein d'un noeud, conduisant à des performances suboptimales. L'objet de ce sujet de mémoire est de s'intéresser aux problématiques posées à bas niveau dans la gestion des communications consécutivement aux évolutions des architectures et des modèles de programmation hybrides.

Il s'agira alors pour le candidat de s'intéresser aux aspects suivants :

- Imaginer des mécanismes de coordination des processus pour accéder simultanément à la carte réseau de manière concertée plutôt que concurrente. En effet, avec les implémentations MPI actuelles, chaque porcessus accède directement à la carte réseau sans tenir compte des accès des autres processus, ce qui peut mener à une contention en cas de nombreux processus.

- Étudier, implémenter, et comparer les surcoûts et les avantages de différentes stratégies de routage sur de telles topologies hierarchiques, telles que : envoi direct avec concertation, envoi indirect avec route statique, envoi indirect avec routage adaptatif.

- Tirer profit de ce nouveau type de topologies pour de nouvelles opportunités d'optimisations inter-processus. Il semble opportun d'envisager une optimisation et un ordonnancement des messages à un niveau global sur chaque noeud. Par exemple, notre implémentation actuelle NewMadeleine est capable d'agréger des messages consécutifs d'une application en un seul paquet émis sur le réseau ; il est attendu d'appliquer des schémas d'optimisation similaires mais à un niveau global entre plusieurs processus.

- Étudier les interactions entre communications réseaux, communications inter-processus en mémoire partagée, communications vers les accélérateurs.

- Étudier le passage à l'échelle des mécanismes impliqués, aussi bien en nombre de processus qu'en nombre de noeuds et de cartes réseau. Il semble désormais acquis qu'il n'est plus approprié de ne considérer que des cas particuliers avec un ou deux processeurs et une ou des cartes réseau, mais qu'il faille au contraire considérer N ressources de chaque type.

- Valider la proposition par une implémentation, évaluer les performances par des micro-benchmarks, et évaluer le comportement global de la solution proposée avec des applications complètes. À cette occasion, l'étudiant sera amené à travailler en collaboration avec les autres membres de l'équipe s'intéressant aux modèles de programmation hybrides.

Références bibliographiques

- François Trahay. *De l'interaction des communications et de l'ordonnancement de threads au sein des grappes de machines multi-coeurs.* PhD thesis, Université Bordeaux 1, November 2009.

- François Trahay and Alexandre Denis. *A scalable and generic task scheduling system for communication libraries.* In Proceedings of the IEEE International Conference on Cluster Computing, New Orleans, LA, September 2009. IEEE Computer Society Press.

- François Trahay, Élisabeth Brunet, Alexandre Denis, and Raymond Namyst. *A multithreaded communication engine for multicore architectures.* In CAC 2008: Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2008, Miami, FL, April 2008. IEEE Computer Society Press.
- Elisabeth Brunet, François Trahay, Alexandre Denis, and Raymond Namyst. *A sampling-based approach for communication libraries auto-tuning.* In International Conference on Cluster Computing (IEEE Cluster), Austin, Texas, pages 299-307, September 2011. IEEE Computer Society Press.