Modèles de performance des applications et des plates-formes parallèles pour le placement de processus

Encadrants : Brice Goglin (Brice.Goglin@labri.fr, HDR)
Lieu : Inria/LaBRI, équipe Satanas, projet Runtime
Mots-clés : Calcul haute performance, applications parallèles, placement de processus, modèles de performance.

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 d'abord utilisé des ordinateurs parallèles homogènes. Cependant, les récentes évolutions 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). A l'avenir, les machines parallèles auront couramment des milliers de nœuds composés de processeurs de plusieurs dizaines de cœurs. Du fait de l'hétérogénéité et de la hiérarchie des infrastructures, programmer efficacement et simplement de tels environnements est un des défis majeur du calcul haute performance. En particulier, le placement des processus de calcul sur de telles machines joue un rôle crucial sur les performances de l'application du fait de l'affinité de certains processus entre eux et des performances hétérogènes des communications. Ainsi, les affinités des tâches de calcul vis-à-vis de la localité matérielle deviennent de plus en plus importantes pour la bonne exploitation de la puissance de calcul des machines multicœurs modernes. En effet, certaines paires de processus communiquent d'avantage que d'autres paires et il est donc important de placer correctement ces tâches sur les ressources afin d'optimiser les communications et les accès mémoire.

Pour optimiser ce placement, les algorithmes développés dans notre équipe utilisent des modèles simples, fournis par la bibliothèque hwloc, qui représentent la hiérarchie mémoire par un arbre. A ce jour hwloc ne donne que des informations structurelles et pas des informations qualitatives (e.g. vitesse du bus, distance NUMA, etc). La structure de cet arbre est utilisée par des algorithmes de placement pour placer les processus et les threads sur les ressources de calcul. Cependant, ces algorithmes utilisent tous les niveaux de la hiérarchie pour prendre leurs décisions alors que certains niveaux (e.g. de cache) ont plus d'impact sur les performances que d'autres. D'autre part cette importance varie beaucoup en fonction du type d'application (celles qui communiquent beaucoup vs. celles qui font beaucoup d'accès mémoire vs. celles qui calculent beaucoup). Il est donc nécessaire de comprendre quels sont les niveaux de la hiérarchie qui impactent le plus la performance en fonction du type d'application.

L'objectif de ce travail est de comprendre l'interaction entre les performances d'une application en fonction du partage des ressources matériels. Par exemple, en prenant des applications simples (multiplication de matrices, stencil, etc.) on voudra savoir comment varient les performances si, dans une machine multicœurs, on confine l'exécution des threads à un certain niveau de la hiérarchie (cache L2, cache L3, banc mémoire, nœuds, etc.).

La thèse comportement les axes de travail suivants :

- Étude par micro-benchmarks des besoins des applications parallèles et définitions de différentes métriques (intensité des calculs séquentiels, intensité des accès mémoire, partage de mémoire, etc).
- Enrichissement du modèle hwloc pour annoter la topologie des machines avec des informations qualitatives sur le comportement de l'architecture selon les métriques

logicielles définies ci-dessus.

- Extension au réseau (capacité des liens, etc) dans le logiciel netloc.
- Mise en place de techniques évoluées de placement de processus en fonction des modèles proposés ci-dessus :
    - Adaptation des les algorithmes de placement comme TreeMatch pour tenir compte de ces nouvelles informations matérielles et logicielles.
    - Placement dans des topologies qui ne sont pas des arbres puis dans des topologies arbitraires. Pour cela, nous utiliserons des bibliothèques adaptées à ces topologies comme Scotch ou LibTopoMap.
    - Etude des aspects Non Uniform Memory I/O Access (NUIOA) des processeurs. En effet, dans les architectures modernes, certains cœurs sont privilégiées pour faire des entrées-sorties (e.g. accès au réseau). Le placement des calculs doit donc être optimisé en fonction de ces contraintes architecturales pour prendre en compte l'affinité en entrée-sortie de certaines tâches/threads de l'application.

Le cadre applicatif de la thèse comprendra des applications plus grand publique comme le multimédia (e.g. décodage H264/AVC), mais aussi des applications classiques de HPC (factorisation de Cholesky, gradient conjugué, etc.). Les plates-formes cibles iront de la grappe contenant de nombreux nœuds simples (une dizaine de cœurs) à des machines massivement multicoeurs (jusqu'à 160 cœurs dans PlaFRIM).

Références :

- Brice Goglin. *Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality (hwloc).* In Proceedings of 2014 International Conference on High Performance Computing & Simulation (HPCS 2014), Bologna, Italy, July 2014.
- Bertrand Putigny, Brice Goglin, and Denis Barthou. *A Benchmark-based Performance Model for Memory-bound HPC Applications.* In Proceedings of 2014 International Conference on High Performance Computing & Simulation (HPCS 2014), Bologna, Italy, July 2014.
- François Broquedis, Jérôme Clet-Ortega, Stéphanie Moreaud, Nathalie Furmento, Brice Goglin, Guillaume Mercier, Samuel Thibault, and Raymond Namyst. *hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications.* In Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2010), pages 180-186, Pisa, Italia, February 2010. IEEE Computer Society Press.
- Stéphanie Moreaud, Brice Goglin, and Raymond Namyst. *Adaptive MPI Multirail Tuning for Non-Uniform Input/Output Access.* In Edgar Gabriel Rainer Keller and Jack Dongarra, editors, Recent Advances in the Message Passing Interface. The 17th European MPI User's Group Meeting (EuroMPI 2010), volume 6305 of Lecture Notes in Computer Science, pages 239-248, Stuttgart, Germany, September 2010. Springer-Verlag
- Emmanuel Jeannot, Guillaume Mercier, and François Tessier. *Process Placement in Multicore Clusters : Algorithmic Issues and Practical Techniques*. In : IEEE Transactions on Parallel and Distributed Systems 25.4, p. 993–1002. April 2014.
- François Broquedis, Nathalie Furmento, Brice Goglin, Pierre-André Wacrenier, and Raymond Namyst. *ForestGOMP: an efficient OpenMP environment for NUMA architectures.* International Journal on Parallel Programming, Special Issue on OpenMP; Guest Editors: Matthias S. Müller and Eduard Ayguadé, 38(5):418-439, 2010.

Logiciels :

- Portable Hardware Locality (hwloc). http://www.open-mpi.org/projects/hwloc/
- Portable Network Locality (netloc). http://www.open-mpi.org/projects/netloc/
- TreeMatch, Process placement for multicore clusters. http://treematch.gforge.inria.fr/