

Modèle de performance par auto-tuning

Sujet de master2

Lieu du stage : Equipe-Projet RUNTIME, INRIA, Bordeaux

Encadrants : Denis Barthou, Bertrand Putigny

Contact : 05.24.57.41.16

Contexte

La mise au point des performances d'un code parallèle est une étape essentielle pour l'obtention de codes à hautes performances. La complexité croissante des architectures fait que la construction de modèles et d'outils fiables permettant l'analyse des performances et l'optimisation de code est de plus en plus difficile et est l'objet de nombreuses recherches actuellement [SM06, MMSW01].

Les analyses de performances sur les architectures récentes d'Intel sont très complexes, en raison de la complexité de l'architecture sous-jacente. L'un des points de départ est la caractérisation des instructions assembleur de la machine et les ressources qu'elles mobilisent. Actuellement, aucun outil à notre connaissance ne réalise automatiquement cette étape, alors que ces données changent d'une génération de puce à l'autre. Outre les données constructeur, peu fiables, quelques travaux donnent ces performances, mesurées à l'aide de microbenchmarks réalisés à la main [Fog11, GA05]. La génération automatique de ces benchmarks permettrait de mieux capturer le fonctionnement détaillé de l'architecture, de mieux comprendre les points clés pour les performances et d'obtenir un modèle fin d'architecture.

Le but du stage est d'étudier les travaux relatifs à la modélisation fine des performances, à l'échelle d'un cœur (par exemple, [BCRJ⁺10, HRTV11]). Pour le volet algorithmique, le stagiaire pourra commencer par proposer une méthode systématique permettant de construire automatiquement des microbenchmarks évaluant les performances des instructions individuelles, puis on évaluera l'impact de l'allocation des registres et de l'ordonnancement pour une machine super-scalaire comme les processeurs récents d'Intel et l'interaction entre différentes instructions.

Ce travail sur les méthodes pourra s'accompagner d'un travail de développement d'un prototype. Ce travail sera mené en collaboration avec un travail de thèse sur les modèles de performance et optimisations pour les architectures

multicoeurs, et en partenariat avec le laboratoire Exascale Lab (Université de Versailles St Quentin, CEA, Genci, Intel).

References

- [BCRJ⁺10] Denis Barthou, Andres Charif Rubial, William Jalby, Souad Koliai, and Cedric Valensi. Performance tuning of x86 openmp codes with maqao. In Matthias S. Muller, Michael M. Resch, Alexander Schulz, and Wolfgang E. Nagel, editors, *Tools for High Performance Computing 2009*, pages 95–113. Springer Berlin Heidelberg, 2010.
- [Fog11] Agner Fog. Optimization manuals, 2011. www.agner.org.
- [GA05] Torbjörn Granlund and Swox Ab. Instruction latencies and throughput for amd and intel x86 processors, 2005.
- [HRTV11] R. Hundt, E. Raman, M. Thuresson, and N. Vachharajani. Mao : An extensible micro-architectural optimizer. In *Code Generation and Optimization (CGO), 2011 9th Annual IEEE/ACM International Symposium on*, pages 1–10, april 2011.
- [MMSW01] Bernd Mohr, Allen D. Malony, Sameer Shende, and Felix Wolf. Towards a performance tool interface for openmp: An approach based on directive rewriting. In *Workshop on OpenMP*, 2001.
- [SM06] Sameer S. Shende and Allen D. Malony. The tau parallel performance system. *Intl. Journal on High Performance Computing Applications*, 20:287–331, 2006.