



Séminaire Informatique Haute Performance @ Campus Teratec

Séminaire n°50 du Jeudi 29 Septembre 2016, 10h, Ter@tec.
Décomposition Automatique des Programmes Parallèles pour
l'Optimisation et la Prédiction de Performance.

Jeudi 29 Septembre 2016, Mihail Popov, Doctorant à l'Université de Versailles Saint-Quentin-en-Yvelines nous présentera ses travaux de thèse.

Voici le résumé de cette présentation qui aura lieu dans la salle Paul Gauguin à Ter@tec, à 10h

Décomposition Automatique des Programmes Parallèles pour l'Optimisation et la Prédiction de Performance.

Dans le domaine du calcul haute performance, de nombreux programmes étalons ou benchmarks sont utilisés pour mesurer l'efficacité des calculateurs, des compilateurs et des optimisations de performance. Les benchmarks de référence regroupent souvent des programmes de calcul issus de l'industrie et peuvent être très longs. Le processus d'étalonnage d'une nouvelle architecture de calcul ou d'une optimisation est donc coûteux.

La plupart des benchmarks sont constitués d'un ensemble de noyaux de calcul indépendants. Souvent l'étalonneur n'est intéressé que par un sous-ensemble de ces noyaux, il serait donc intéressant de pouvoir les exécuter séparément. Ainsi, il devient plus facile et rapide d'appliquer des optimisations locales sur les benchmarks. De plus, les benchmarks contiennent de nombreux noyaux de calcul redondants. Certaines opérations, bien que mesurées plusieurs fois, n'apportent pas d'informations supplémentaires sur le système à étudier. En détectant les similarités entre eux et en éliminant les noyaux redondants, on diminue le coût des benchmarks sans perte d'information.

Cette thèse propose une méthode permettant de décomposer automatiquement une application en un ensemble de noyaux de performance, que nous appelons codelets. La méthode proposée permet de rejouer les codelets, de manière isolée, dans différentes conditions expérimentales pour pouvoir étalonner leur performance. Cette thèse étudie dans quelle mesure la décomposition en noyaux permet de diminuer le coût du processus de benchmarking et d'optimisation. Elle évalue aussi l'avantage d'optimisations locales par rapport à une approche globale.

De nombreux travaux ont été réalisés afin d'améliorer le processus de benchmarking. Dans ce domaine, on remarquera l'utilisation de techniques d'apprentissage machine ou d'échantillonnage. L'idée de décomposer les benchmarks en morceaux indépendants n'est pas nouvelle. Ce concept a été appliqué avec succès sur les programmes séquentiels et nous le portons à maturité sur les programmes parallèles.

Évaluer des nouvelles micro-architectures ou la scalabilité est 25x fois plus rapide avec des codelets qu'avec des exécutions d'applications.

Les codelets prédisent le temps d'exécution avec une précision de 94% et permettent de trouver des optimisations locales jusqu'à 1.06x fois plus efficaces que la meilleure approche globale.
