GÉNÉRALISATION DE L'ANALYSE DE PERFORMANCE DÉCRÉMENTALE VERS L'ANALYSE DIFFÉRENTIELLE PAR ZAKARIA BENDIFALLAH

Présentée par : Zakaria Bendifallah Discipline : informatique Laboratoire : PRISM

Résumé :
Une des étapes les plus cruciales dans le processus d'analyse des performances d'une application est la détection des goulets d'étranglement. Un goulet étant tout évènement qui contribue au temps d'exécution, la détection de ses causes est importante pour les développeurs d'applications afin de comprendre les défauts de conception et de génération de code.
Cependant, la détection de goulets devient un art difficile. Dans le passé, des techniques qui reposent sur le comptage du nombre d'évènements arrivaient facilement à trouver les goulets. Maintenant, la complexité accrue des micro-architectures modernes et l'introduction de plusieurs niveaux de parallélisme ont rendu ces techniques beaucoup moins efficaces. Par conséquent, il y a un réel besoin de réflexion sur de nouvelles approches.
Notre travail porte sur le développement d'outils d'évaluation de performance des boucles de calculs issues d'applications scientifiques. Nous travaillons sur DECAN, un outil d'analyse de performance qui présente une approche intéressante et prometteuse appelée l'Analyse Décrementale. DECAN repose sur l'idée d'effectuer des changements contrôlés sur les boucles du programme et de comparer la version obtenue (appelée

variante) avec la version originale, permettant ainsi de détecter la présence ou pas de goulets d'étranglement.
Tout d'abord, nous avons enrichi DECAN avec de nouvelles variantes, que nous avons conçues, testées et validées. Ces variantes sont, par la suite, intégrées dans une analyse de performance poussée appelée l'Analyse Différentielle. Nous avons intégré l'outil et l'analyse dans une méthodologie d'analyse de performance plus globale appelée PAMDA.
Nous décrirons aussi les différents apports à l'outil DECAN. Sont particulièrement détaillées les techniques de préservations des structures de contrôle du programme, ainsi que l'ajout de support pour les programmes parallèles.
Finalement, nous effectuons une étude statistique qui permet de vérifier la possibilité d'utiliser des compteurs d'évènements, autres que le temps d'exécution, comme métriques de comparaison entre les variantes DECAN.

Abstract :
A crucial step in the process of application performance analysis is the accurate detection of program bottlenecks. A bottleneck is any event which contributes to the execution time, to application developers as it enables to detect code design and generation flaws.
Bottleneck detection is becoming a difficult art. Techniques such as event counts, which succeeded to find bottlenecks easily in the past, became less efficient because of the increasing complexity of modern micro-processors, and the introduction of parallelism at several levels. Consequently, a real need for new analysis approaches is present in order to face these challenges.
Our work focuses on performance analysis and bottleneck detection of compute intensive loops in scientific applications. We work on DECAN a performance analysis and bottleneck detection tool which offers an interesting and promising approach called Decremental Analysis. The tool, which operates at binary level, is based on the idea of performing controlled modifications on the instructions of a loop, and comparing the new version (called variant) against the original one. The goal is to assess the cost of specific events, and thus the existence or not of bottlenecks.
Our first contribution consists of extending DECAN with new variants that we designed, tested and validated. Based on these variants, we developed analysis methods which we used to characterize hot loops and find their bottlenecks. We later, integrated the tool into a performance analysis methodology (PAMDA) which coordinates several analysis tools in order to achieve a more efficient application performance analysis.Second, we introduce several improvements on the DECAN tool. Techniques developed to preserve the control flow of the modified programs, allowed to use the tool on real applications

instead of extracted kernels. Support for parallel programs (thread and process based) was also added. Finally, the tool relying primarily on execution time as the main event for its analysis process, we study the possibility to use of other hardware generated events as well, through a study of their stability, precision and overhead.

INFORMATIONS COMPLÉMENTAIRES

Albert COHEN, Directeur de Recherche, à l'ENS - Département d'Informatique - Paris - Rapporteur
Michael KRAJECKI, Professeur des Universités, à l'Université de Reims Champagne-Ardenne/Centre de Recherche en STIC - EA 3804 - Reims - Rapporteur
William JALBY, Professeur des Universités, à l'Université de Versailles Saint-Quentin-en-Yvelines/Laboratoire Parallélisme, Réseaux, Système, Modélisation (PRISM) - Versailles - Directeur de thèse
Edouard AUDIT, Ingénieur, à la Maison de la Simulation - CEA Saclay - Gif/Yvette - Examinateur
Lee BAUGH, Chercheur, à Google - Seattle (Etats-Unis) - Examinateur
Jean-Thomas ACQUAVIVA, Ingénieur, à Data DirectNetWork - Meudon-la-Forêt - Examinateur
Andry RAZAFINJATOVO, Ingénieur, à Bull SAS - Echirolles - Invité

Contact : dredval service FED : theses@uvsq.fr