



université PARIS-SACLAY

## «DÉPORT DE DIFFÉRENTES PHASES DE COMPILATION AU RUNTIME» PAR KHAWAR SAJJAD

Présentée par : Monsieur Khawar SAJJAD Discipline : Informatique Laboratoire : PRISM

### Résumé :

Les compilateurs modernes consacrent beaucoup d'efforts pour générer un code à la fois correct et efficace. De nombreuses optimisations ont été développées pour améliorer les performances du programme. Classiquement les techniques de compilation dynamique ne se sont pas imposées comme norme. Il a été démontré que la compilation dynamique peut aider à optimiser le code au-delà des limites traditionnelles des compilateurs statiques. Les compilateurs dynamiques incluent généralement des optimisations telles que : l'évaluation partielle d'expression arithmétique ou logique, la propagation de constante/zéro, déroulage de boucles, etc... en vue de l'optimisation des programmes. Durant cette thèse, je présente les techniques de compilation dynamique qui étendent le champ d'application d'optimisations dynamiques. Les techniques que j'ai développées génèrent du code binaire sous forme des composants simples appelés code blocks. On peut interconnecter ces composants en diverses manières, plus tard, selon le problème sous la main et selon le modèle d'ILP de la machine cible. Je montre que le code généré par mes techniques est souvent plus performant que du code C

généralisé statiquement. En plus, mes techniques étant dynamiques sont plus flexibles que des techniques statiques traditionnelles. Je montre que mes techniques facilitent l'utilisation des opérations matérielles spécialisées de manière portable avec la possibilité d'exploitation du contexte au temps d'exécution, par exemple on peut générer des opérations spéciales de chargement de mémoire en se basant sur l'alignement mémoire de variables au temps d'exécution. Ensuite, je montre des techniques dynamiques permettant de choisir les jeux d'instructions (ISA) pour les noyaux d'un programme en fonction de conditions d'exécution sur une architecture multicore hétérogène. Finalement, je montre que la compilation dynamique peut être appliquée pour exploiter les zones mémoires qui se chevauchent sur une application de compression vidéo. Au cours de mes expérimentations, j'ai comparé la performance des codes que je génère avec des codes optimisés/tunés à la main. J'ai observé que dans certains cas, le code généré dynamiquement a des performances similaires à ceux optimisés à la main et dans certains cas il les dépasse en terme de performance.

### **Abstract :**

Modern compilers work hard to generate both correct and efficient code. A number of compiler optimizations have been developed overtime to achieve better program performance. State-of-art compilation tools have largely remained static. Although dynamic compilation techniques have failed to make it to the mainstream, it has been shown that dynamic compilation can help optimize code beyond the limits of traditional static compilers. Classical dynamic compilers have generally relied upon optimizations like partial evaluation, constant/zero propagation, loop unrolling etc... to improve program performance. In this thesis I present dynamic compilation techniques that extend the scope of dynamic optimizations. I have developed techniques that allow binary code generation in the form of simple code components called code blocks that can be interconnect in different ways according to the problem at hand and the ILP-model of the target machine. I show that in most cases the code generated by this technique exhibits performance superior to optimized static C-code. Moreover the technique being at runtime is more flexible as compared to the corresponding static code. I show that my techniques help exploit special purpose machine operations in a portable manner with possible context exploitation e.g. the use of special load operations based on the runtime memory alignment of different variables. I demonstrate dynamic techniques to choose the instruction set (ISA) for several program kernels based on the runtime conditions on a heterogeneous multi-core architecture. Finally I demonstrate that dynamic compilation can be used to exploit overlapping memory regions in video compression applications. Program performance in some cases has been compared to that of hand tuned assembly codes where available. It has been shown that dynamically generated code

demonstrates similar performances to the hand tuned codes in some cases and outperform them in others.

## INFORMATIONS COMPLÉMENTAIRES

**Philippe CLAUSS**, Professeur des Universités, à l'Université de Strasbourg/Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT) - Illkirch - Rapporteur

**Tanguy RISSET**, Professeur des Universités, à l'INSA/Laboratoire Centre of Innovation in Telecommunications and Integration of services (CITI) - Villeurbanne - Rapporteur

**Henri-Pierre CHARLES**, Chercheur, au CEA/Laboratoire Intégration et Atelier Logiciel pour Puces (LIALP) - Gières - Directeur de thèse

**William JALBY**, Professeur des Universités, à l'Université de Versailles Saint-Quentin-en-Yvelines/Laboratoire Parallélisme, Réseaux, Système, Modélisation (PRISM) - Versailles - Examineur

**Yves LHUILLIER**, Ingénieur de Recherche, au CEA/Laboratoire d'Intégration des Systèmes et des Technologies (LIST) - Gif/Yvette - Examineur

**Jean-François MEHAUT**, Professeur des Universités, à l'Université Joseph Fourier Grenoble 1/Laboratoire Informatique de Grenoble (LIG) - Montbonnot Saint Martin - Examineur

**Contact :** dredval service FED : [theses@uvsq.fr](mailto:theses@uvsq.fr)