Abstract

Approximate computing is a paradigm that deliberately trades accuracy for better performance or lower energy consumption. Emerging applications such as image processing and deep learning provide inherent resilience towards approximation. Approximating emerging workloads makes an important contribution to tackle the evergrowing requirements on performance and energy consumption. This thesis presents techniques to approximate applications on multicore CPUs, GPUs, and embedded GPUs. The proposed approximation approaches perforate applications by skipping loop iterations and memory accesses to gain performance.

The first part of the thesis presents Kernel Perforation, a novel technique that perforates the input buffer of GPU applications. Kernel Perforation exploits the GPU's local memory to improve the accuracy of the approximations. The evaluation shows that the technique accelerates applications up to 3×, while improving the accuracy significantly when compared to state-of-the-art approaches. The second part of the thesis shows how Kernel Perforation can be employed to accelerate applications on embedded systems. These systems are architecturally different from desktop GPUs. They often share their memory with the CPU, and they provide no dedicated local memory in some cases. The results show, that the technique can accelerate applications on embedded GPUs up to 1.38×. Finally, the last part focuses on ALONA, a framework for advanced automatic compilation techniques based on polyhedral analysis. Many computationally demanding applications such as scientific simulations or deep learning applications feature deeply nested, multidimensional loops. We introduce a novel compiler-based approach that perforates deeply nested loops using multidimensional perforation schemes; improves the accuracy by signal reconstruction; and efficiently handles the large transformation space by automatically selecting promising code versions.

Overall, the proposed approximation framework improves performance and accuracy of applications on multicore CPUs, GPUs, and embedded GPUs when compared to state-of-the-art perforation approaches.

Zusammenfassung

Approximate Computing ist Rechenparadigma, das bewusst Genauigkeit gegen Leistung, oder Energieverbrauch tauscht. Aufstrebender Anwendungen wie Bildverarbeitung und Deep Learning bieten inhärente Resilienz gegenüber Approximierung. Die Approximierung dieser Anwendungen ist ein wichtiger Beitrag zur Bewältigung der stets wachsenden Anforderungen an Rechenleistung und Energieverbrauch. In dieser Dissertation werden Techniken zur Approximierung von Applikationen auf mehrkernigen CPUs, GPUs und eingebetteten GPUs vorgestellt. Die vorgeschlagenen Techniken perforieren Applikationen in dem Schleifendurchläufe und Speicherzugriffe ausgelassen werden um die Rechenleistung zu steigern.

Im ersten Teil wird Kernel Perforation, eine neuartige Technik, die den Eingabepuffer von GPU-Applikationen perforiert, vorgestellt. Dieser Ansatz nutzt den lokalen Speicher der GPU um die Genauigkeit der Approximierungen zu verbessern. Unsere Evaluation zeigt, dass der Ansatz im Vergleich zum Stand der Technik die Ausführung von Applikationen um bis zu 3× beschleunigt und die Genauigkeit signifikant verbessert. Im zweiten Teil wird gezeigt, wie der Ansatz zur Beschleunigung von Applikationen auf eingebetteten Systemen, die sich von der Architektur von Desktop-GPUs deutlich unterscheiden, da sie oftmals den Hauptspeicher mit der CPU teilen und keinen dedizierten lokalen Speicher haben, verwendet werden kann. Die Auswertung zeigt, dass der Ansatz Applikationen um bis zu 1.38× beschleunigen kann. Schließlich wird im letzten Teil der Fokus ALONA, einem Framework für automatische Perforation auf CPUs mit einem Compiler, gesetzt. Viele Programme mit hoher Rechenlast wie zum Beispiel wissenschaftliche Simulationen oder Deep Learning verwenden tief verschachtelte, mehrdimensionale Schleifen. Deshalb werden mehrdimensionale Perforationsansätze benötigt. Dadurch wird allerdings der Approximationsraum zu groß um eine manuelle Approximierung und ein manuelles Optimieren der Parameter in Betracht zu ziehen. Deshalb stellen wir Technik vor, die es erlaubt vielversprechende Varianten automatisch auszuwählen.

Insgesamt verbessern die vorgestellte Perforationstechniken die Leistung und Genauigkeit von Applikationen auf CPUs, GPUs und eingebetteten GPUs im Vergleich zum Stand der Technik.