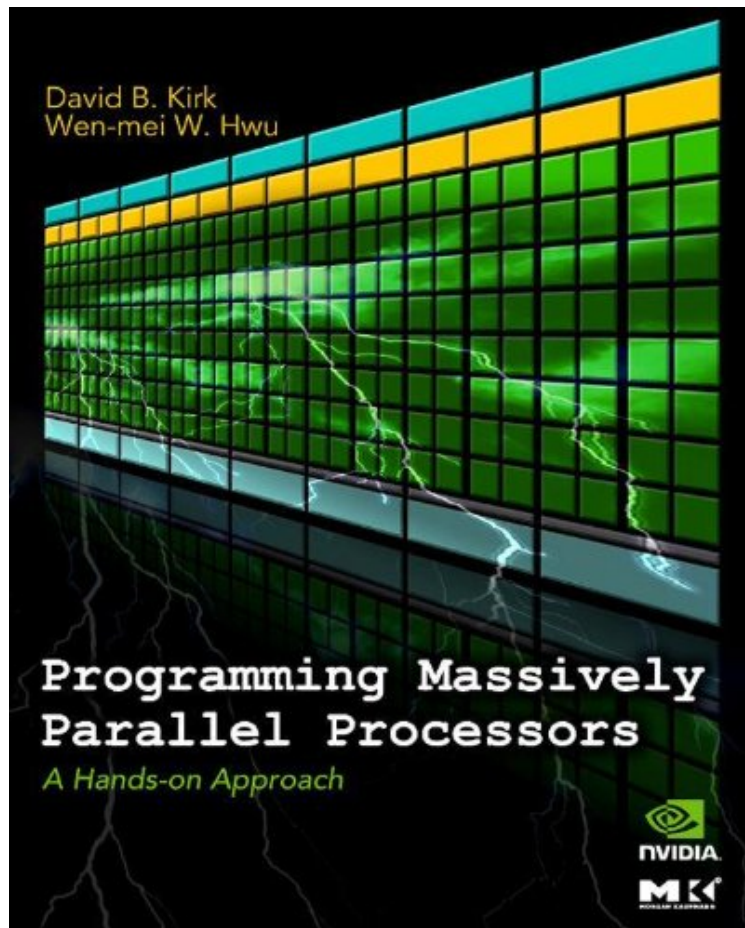


[Download free ebook] Programming Massively Parallel Processors: A Hands-on Approach

Programming Massively Parallel Processors: A Hands-on Approach

Von David B. Kirk, Wen-mei W. Hwu
ePub | *DOC | audiobook | ebooks | Download PDF



DOWNLOAD



+

READ ONLINE

Produktinformation -Verkaufsrang: #938454 in eBooksVerffentlicht am: 2010-02-22Erscheinungsdatum:
2010-02-22File Name: B003EH18TS | File size: 22.Mb

Von David B. Kirk, Wen-mei W. Hwu : Programming Massively Parallel Processors: A Hands-on Approach
before purchasing it in order to gage whether or not it would be worth my time, and all praised Programming
Massively Parallel Processors: A Hands-on Approach:

KundenrezensionenHilfreichste Kundenrezensionen15 von 21 Kunden fanden die folgende Rezension hilfreich. Solide
EinfhrungVon Dr. Chrilly DonningerDas Buch entstand aus mehreren Vorlesungen/Kursen der Autoren zur CUDA.
Die Autoren verwenden nicht die bliche CopyPaste Methode der SDK-Dokumentation. Sie geben dem Leser
stattdessen den klassischen Ratschlag RTFM. Sie konzentrieren sich auf die konzeptionelle Seite. An Hand einer
Matrizenmultiplikation wird schrittweise gezeigt, wie man die maximale Performance aus einer GPU herausholen
kann. Die unmittelbare Transformation des Problems in die CUDA ist sehr einfach. Allerdings wird dieser naive
Ansatz durch die Latenz und die Bandweite des globalen Grafikkarten Memory's ausgebremst. Ein klassisches
Problem in praktisch allen massiv-parallelen Techniken mit shared memory (bei distributed memory ist dafr die
Kommunikation der Flaschenhals). Die Autoren zeigen, wie man durch diverse Tricks den globalen Memory-Zugriff

verringert und lokales Memory besser ausnutzt. Sie gehen auch detailliert auf den dadurch erreichbaren Speedup ein. Die einzelnen Schritte sind didaktisch sehr gut aufgebaut. Man bekommt ein gutes Gefühl für die Stärken und Schwächen der GPU. Ich habe bereits eine HPC (High-Performance-Computing) Anwendung mit FPGAs gebaut. Die FPGA Community machte sich Hoffnungen, in diesen lukrativen Markt am Kuchen mitzuschneiden zu können. Für rein numerische (floating-point) HPC-Anwendungen sind diese Pläne m.E. mit dem Erscheinen der CUDA gestorben. Man muss auch mit der CUDA einiges Hirnschmalz aufwenden um einen Algorithmus effektiv zu implementieren. Aber im Verhältnis zum Aufwand für eine FPGA-Implementation ist das noch immer nix. Auch preislich liegen zwischen HPC-FPGA Karten und Grafikkarten Welten. Ich kenne auch keine mit diesem Buch vergleichbare Einführung in das HPC-Computing mit FPGAs. Die Sache wurde in den Kinderzimmern entschieden. Es schwebt mir vor, eine finanzmathematische Monte-Carlo Simulation auf die CUDA zu portieren. Allerdings habe ich das Problem, dass die Simulation auch am Pentium schnell genug ist. Ich muss wohl das Modell komplexer machen um mich mit gutem Gewissen mit der CUDA spielen zu können. Es war noch nie so leicht eine massiv-parallele Anwendung zu schreiben. Es ist aber auch nicht zu leicht. 2 von 4 Kunden fanden die folgende Rezension hilfreich. Unverzichtbar für die Performance-Optimierung von Jörn Dinkla. Dieses Buch ist vor etwa zwei Jahren erschienen. Aus diesem Grund sind Teile des Buches veraltet, da viele Konstanten für die G80- und die G200-Architekturen angegeben werden. Die momentan aktuelle Fermi-Architektur wird nur kurz in Kapitel 12 vorgestellt. Aber: Was dieses Buch von anderen unterscheidet, ist die genaue Beschreibung der Funktionsweise der Hardware, wie z. B. das Warp-Scheduling und die Speicherzugriffe. Auch wird anhand von zwei Fallbeispielen Schritt-für-Schritt sequentieller Code in möglichst optimalen CUDA-Code umgeformt. Hier lernt der Leser das Umschreiben und Anpassen von Code an die GPU und erfährt wichtige Tipps, wie z. B. "loop fission", "latency hiding" und "memory coalescing". Man erfährt so aber auch, dass das Schreiben von optimalem Code erhebliche (Denk-)Arbeit erfordert. Es gibt allerdings auch ein paar Kritikpunkte: die Beispiele erfordern Kenntnisse in der Mathematik, der Index ist unvollständig und der Abschnitt über OpenCL bringt nicht viel, da es für den Einstieg zu wenig ist. Auch werden CUDA Arrays (2D, 3D) und Texturen nicht ausführlich genug behandelt. Wer CUDA-Programme schreiben möchte, die die GPU möglichst gut ausnutzen, kommt an der Lektüre dieses Buches nicht vorbei. 0 von 10 Kunden fanden die folgende Rezension hilfreich. Programming Massively Parallel Processors: A Hands-On Approach von David Singleton. Das Buch war interessant zu lesen, und klärt, wie man einen Graphics-Processor programmieren kann. Ich muss jetzt die Zeit finden, es alles auszuprobieren!!

Kurzbeschreibung Programming Massively Parallel Processors discusses the basic concepts of parallel programming and GPU architecture. Various techniques for constructing parallel programs are explored in detail. Case studies demonstrate the development process, which begins with computational thinking and ends with effective and efficient parallel programs. This book describes computational thinking techniques that will enable students to think about problems in ways that are amenable to high-performance parallel computing. It utilizes CUDA (Compute Unified Device Architecture), NVIDIA's software development tool created specifically for massively parallel environments. Studies learn how to achieve both high-performance and high-reliability using the CUDA programming model as well as OpenCL. This book is recommended for advanced students, software engineers, programmers, and hardware engineers. Teaches computational thinking and problem-solving techniques that facilitate high-performance parallel computing. Utilizes CUDA (Compute Unified Device Architecture), NVIDIA's software development tool created specifically for massively parallel environments. Shows you how to achieve both high-performance and high-reliability using the CUDA programming model as well as OpenCL. **Pressestimmen** "For those interested in the GPU path to parallel enlightenment, this new book from David Kirk and Wen-mei Hwu is a godsend, as it introduces CUDA (tm), a C-like data parallel language, and Tesla(tm), the architecture of the current generation of NVIDIA GPUs. In addition to explaining the language and the architecture, they define the nature of data parallel problems that run well on the heterogeneous CPU-GPU hardware ... This book is a valuable addition to the recently reinvigorated parallel computing literature." - David Patterson, Director of The Parallel Computing Research Laboratory and the Pardee Professor of Computer Science, U.C. Berkeley. Co-author of Computer Architecture: A Quantitative Approach "Written by two teaching pioneers, this book is the definitive practical reference on programming massively parallel processors--a true technological gold mine. The hands-on learning included is cutting-edge, yet very readable. This is a most rewarding read for students, engineers, and scientists interested in supercharging computational resources to solve today's and tomorrow's hardest problems." - Nicolas Pinto, MIT, NVIDIA Fellow, 2009 "I have always admired Wen-mei Hwu's and David Kirk's ability to turn complex problems into easy-to-comprehend concepts. They have done it again in this book. This joint venture of a passionate teacher and a GPU evangelizer tackles the trade-off between the simple explanation of the concepts and the in-depth analysis of the programming techniques. This is a great book to learn both massive parallel programming and CUDA." - Mateo Valero, Director, Barcelona Supercomputing Center "The use of GPUs is having a big impact in scientific computing. David Kirk and Wen-mei Hwu's new book is an important contribution towards educating our students on the ideas and techniques of programming for massively parallel

processors." - Mike Giles, Professor of Scientific Computing, University of Oxford "This book is the most comprehensive and authoritative introduction to GPU computing yet. David Kirk and Wen-mei Hwu are the pioneers in this increasingly important field, and their insights are invaluable and fascinating. This book will be the standard reference for years to come." - Hanspeter Pfister, Harvard University "This is a vital and much-needed text. GPU programming is growing by leaps and bounds. This new book will be very welcomed and highly useful across interdisciplinary fields." - Shannon Steinfadt, Kent State University "GPUs have hundreds of cores capable of delivering transformative performance increases across a wide range of computational challenges. The rise of these multi-core architectures has raised the need to teach advanced programmers a new and essential skill: how to program massively parallel processors." - CNNMoney.com

Kurzbeschreibung Programming Massively Parallel Processors discusses the basic concepts of parallel programming and GPU architecture. Various techniques for constructing parallel programs are explored in detail. Case studies demonstrate the development process, which begins with computational thinking and ends with effective and efficient parallel programs. This book describes computational thinking techniques that will enable students to think about problems in ways that are amenable to high-performance parallel computing. It utilizes CUDA (Compute Unified Device Architecture), NVIDIA's software development tool created specifically for massively parallel environments. Studies learn how to achieve both high-performance and high-reliability using the CUDA programming model as well as OpenCL. This book is recommended for advanced students, software engineers, programmers, and hardware engineers. Teaches computational thinking and problem-solving techniques that facilitate high-performance parallel computing. Utilizes CUDA (Compute Unified Device Architecture), NVIDIA's software development tool created specifically for massively parallel environments. Shows you how to achieve both high-performance and high-reliability using the CUDA programming model as well as OpenCL.