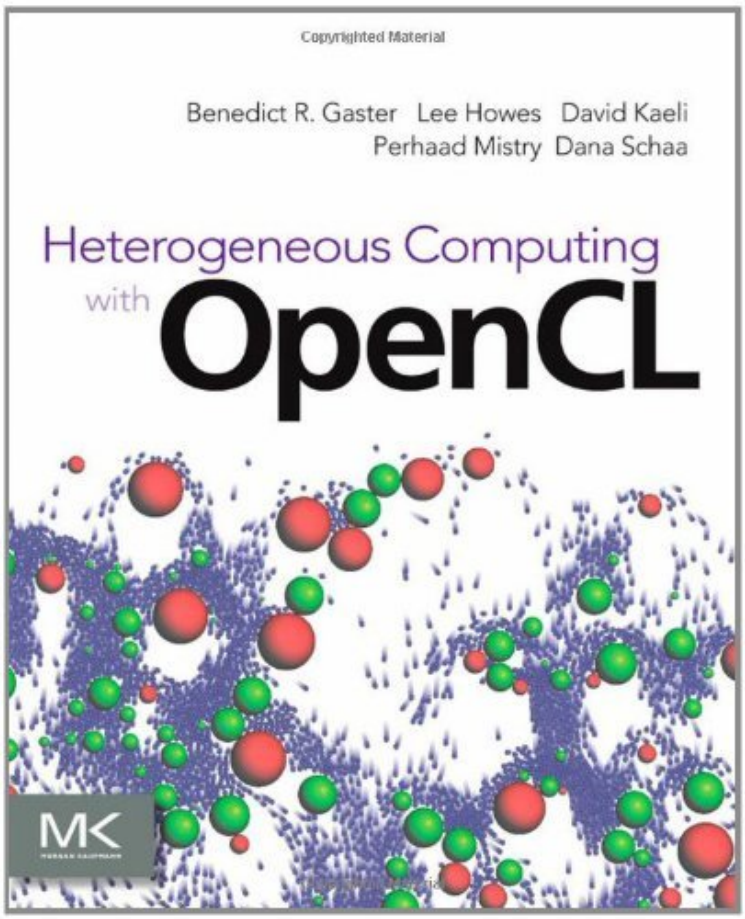
Heterogeneous Computing with OpenCL

Von Benedict Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, Dana Schaa
*ePub | *DOC | audiobook | ebooks | Download PDF*



Produktinformation -Verkaufsrang: #748924 in eBooksVerffentlicht am: 2011-09-30Erscheinungsdatum: 2011-09-30File Name: B005JRHYUS | File size: 15.Mb

Von Benedict Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, Dana Schaa : Heterogeneous Computing with OpenCL before purchasing it in order to gage whether or not it would be worth my time, and all praised Heterogeneous Computing with OpenCL:

KundenrezensionenHilfreichste Kundenrezensionen8 von 21 Kunden fanden die folgende Rezension hilfreich. Gutes Buch, aber CUDA ist besserVon Dr. Chrilly DonningerIch habe eine nichtlineare (und nicht konvexe) Portofolio-Optimierung mit Hilfe der Differential Evolution Heuristik geschrieben. Fr ein typisches Problem bentigt ein Lauf 2 Stunden. Ich mchte die Laufzeit mit Hilfe der Parallelisierung in einer GPU auf unter 10 Minuten drcken. Es bietet sich dafr CUDA und OpenCL an. Zur Entscheidung dieser Frage habe ich mir fr CUDA [1] (und [2]) zugelegt. Fr OpenCL dieses Buch. Die Bcher sind hnlich aufgebaut. Das erste Beispiel ist eine Vektor-Addition. Das ist quasi das GPU "Hello World". Der OpenCL-Kode ist ca. 10-Mal so lang. Das liegt nicht am Buch, sondern an OpenCL. Im Grunde ist meine Entscheidung damit schon gefallen.Man muss auch den Open-CL Kernel Code Zeile fr Zeile unter Anfhrungszeichen setzen. CUDA-Kode ist hingegen reines C mit ein paar zustzlichen Keywords. Ich seh auch keine Notwendigkeit fr die OpenCL Gnsefsschen. Der Compiler kann auf Grund des Kontextes unterscheiden ob es sich um

Host oder GPU Kode handelt. Wer sich das ausgedacht hat, dem fehlt jedes Empfinden fr die sthetik eines Programmkodes. Ich finds jedenfalls abstossend.Nach den Recherchen ist die CUDA in jeden fr mich relevanten Aspekt besser. Die Entwicklungstools sind wesentlich ausgereifter und die Performance ist - bei nichttrivialen Anwendungen - um eine Grssenordnung besser. OpenCL wirbt mit der Platform-Unabhngigkeit. Es ist eine Art massiv-paralleles Java. Angeblich kann man damit seinen C-Kode auch auf FPGAs laufen lassen. Ich habe das massiv-parallele FPGA-Schachprogramm Hydra geschrieben und entwickle Computer-Tomographie FPGA Anwendungen. Diese Ankndigung ist eine reine Werbedurchsage. FPGA-Programme muss man immer noch in VHDL oder Verilog schreiben. Die Synthese von C-Kode funktioniert nur fr triviale Probleme halbwegs. Es hilft einfach nichts: Man schreibt ein GPU (oder FPGA) Programm, weil es auf der CPU zu langsam luft. Man will die speziellen Fhigkeiten der Hardware ausntzen. Das geht aber nur "down-the-metal" und nicht gerteunabhngig.Im Grunde wird das offene und berall laufende OpenCL nur auf AMD Chipstzen wirklich untersttzt. Es gibt auch eine NVIDIA Version. Die ist nach allen mir bekannten Benchmarks grottenschlecht. NVIDIA will wahrscheinlich nur den Fuss in der OpenCL Tr haben. Es gibt aber fr NVIDIA keinen Grund OpenCL zum Durchbruch zu verhelfen.Dementsprechend kommen die Autoren dieses Buches von AMD. Sie bringen hnliche Beispiele wie im CUDA Buch. Sie kennen sich gut aus und haben auch eine gute didaktische Ader. Die Qualitt der Darstellung ist in beiden Bchern gleich gut. Ich kann dieses Buch OpenCL-Interessierten nur empfehlen. Nur bin ich halt der Meinung, dass OpenCL zumindest im Moment keine reale Alternative zur CUDA ist. Es gibt aber auch andere Meinungen. Man sollte daher [1] und dieses Buch lesen und danach selbst seine Schlsse ziehen.[1] J.Sanders, E.Kandrot: CUDA by Example[2] D.B.Kirk, Wen-mei W. Hwu: Programming Massively Parallel ProcessorsP.S.: Ich habe das Problem nun (6.12.2011) parallelisiert. Eine direkte CUDA-Portierung brachte auf einer nicht sehr leistungsstarken Silent-NVIDIA-GPU nur Speedup 1.1. Man kmpft - wenig berraschend - gegen die Memory-Latenz. Durch Cachen und entsprechende Umorganisation der Berechnung ist der Speedup der GPU-Routine 25. Die Gesamtanwendung ist natrlich nicht 25x so schnell geworden. Da kmpft man auch gegen Amdahls Law. Aber es erfllt meine Vorgaben.

KurzbeschreibungHeterogeneous Computing with OpenCL teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs) such as AMD Fusion technology. Designed to work on multiple platforms and with wide industry support, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, this book will give you hands-on OpenCL experience to address a range of fundamental parallel algorithms. The authors explore memory spaces, optimization techniques, graphics interoperability, extensions, and debugging and profiling. Intended to support a parallel programming course, Heterogeneous Computing with OpenCL includes detailed examples throughout, plus additional online exercises and other supporting materials.Explains principles and strategies to learn parallel programming with OpenCL, from understanding the four abstraction models to thoroughly testing and debugging complete applications.Covers image processing, web plugins, particle simulations, video editing, performance optimization, and more.Shows how OpenCL maps to an example target architecture and explains some of the tradeoffs associated with mapping to various architecturesAddresses a range of fundamental programming techniques, with multiple examples and case studies that demonstrate OpenCL extensions for a variety of hardware platformsPressestimmenWith parallel computing now in the mainstream, this book provides an excellent reference on the state-of-the-art techniques in accelerating applications on CPU-GPU systems. -David A. Bader, Georgia Institute of TechnologyKurzbeschreibungHeterogeneous Computing with OpenCL teaches OpenCL and parallel programming for complex systems that may include a variety of device architectures: multi-core CPUs, GPUs, and fully-integrated Accelerated Processing Units (APUs) such as AMD Fusion technology. Designed to work on multiple platforms and with wide industry support, OpenCL will help you more effectively program for a heterogeneous future. Written by leaders in the parallel computing and OpenCL communities, this book will give you hands-on OpenCL experience to address a range of fundamental parallel algorithms. The authors explore memory spaces, optimization techniques, graphics interoperability, extensions, and debugging and profiling. Intended to support a parallel programming course, Heterogeneous Computing with OpenCL includes detailed examples throughout, plus additional online exercises and other supporting materials.Explains principles and strategies to learn parallel programming with OpenCL, from understanding the four abstraction models to thoroughly testing and debugging complete applications.Covers image processing, web plugins, particle simulations, video editing, performance optimization, and more.Shows how OpenCL maps to an example target architecture and explains some of the tradeoffs associated with mapping to various architecturesAddresses a range of fundamental programming techniques, with multiple examples and case studies that demonstrate OpenCL extensions for a variety of hardware platforms