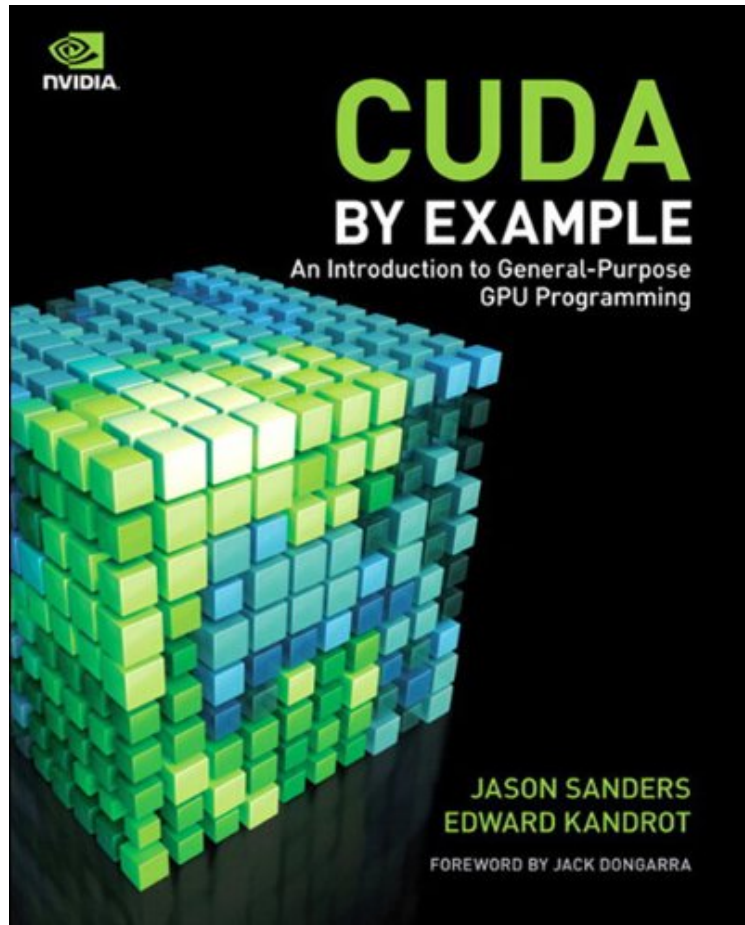


[Read now] CUDA by Example: An Introduction to General-Purpose GPU Programming

# CUDA by Example: An Introduction to General-Purpose GPU Programming

Von Jason Sanders, Edward Kandrot  
audiobook / \*ebooks / Download PDF / ePub / DOC



DOWNLOAD



READ ONLINE

Produktinformation - Verkaufsrang: #340018 in eBooks Veröffentlicht am: 2010-07-19 Erscheinungsdatum: 2010-07-19 File Name: B003VYBOSE | File size: 65.Mb

**Von Jason Sanders, Edward Kandrot : CUDA by Example: An Introduction to General-Purpose GPU Programming** before purchasing it in order to gage whether or not it would be worth my time, and all praised CUDA by Example: An Introduction to General-Purpose GPU Programming:

Kundenrezensionen Hilfreichste Kundenrezensionen 0 von 0 Kunden fanden die folgende Rezension hilfreich. Empfehlenswert Von Alexander Breitenstein Guter Einstieg, saubere Gestaltung und Erklärungen. Würde ich weiter empfehlen! Anmerkung zum kostenlosen ebook: Ebook Version ist so lala und nur 45 Tage online zugreifbar. 12 von 12 Kunden fanden die folgende Rezension hilfreich. Der beste Einstieg in CUDA Von Jörn Dinkla Parallele Algorithmen waren mal ein "esoterisches" Thema für theoretische Informatiker. Mit CUDA kann jetzt jeder massiv parallel programmieren und 10- bis zu 100-fache Geschwindigkeitssteigerungen erreichen. Es gibt diese Bücher - gerade bei neuen Themen - bei denen man den Eindruck hat, die Autoren haben die Dokumentation abgeschrieben und nur ein

wenig ausgeschmückt. Das ist hier definitiv nicht der Fall. Dieses Buch ist der perfekte Start in CUDA für Einsteiger. Aber auch manche Fortgeschrittene und Profis könnten Ihre Freude an den guten Erklärungen und Beispielen haben. Der Buchtitel "CUDA by Example" sagt es schon: Anhand von Beispielen soll gelernt werden. Und da numerische Algorithmen, wie z. B. Vektoraddition, Matrizen-Multiplikation zwar auch behandelt werden, aber von vielen nicht als "Spa" empfunden werden, haben die Autoren graphische Beispiele zusammengestellt: Julia-Mengen, einen einfachen Ray Tracer, Wärmeübertragung. Die Parallelität wird schrittweise, langsam und behutsam eingeführt und immer ausführlich erklärt. Ausgangspunkt ist sequentieller Code für die CPU, der dann umgeformt wird. Das Ziel des Buches, ist es, dem Leser die Grundlagen von CUDA beizubringen, damit er selber darin entwickeln kann. Die Theorie paralleler Algorithmen wird nicht behandelt, wird aber im Rahmen des Buches auch nicht benötigt. Als parallele Lektüre und als Nachschlagewerk benötigt man den "NVIDIA CUDA Programming Guide" und die "NVIDIA CUDA Best Practices", die beide von NVIDIA mit dem CUDA Toolkit mitgeliefert werden. Allerdings ist es ein Einsteigerbuch und die folgenden fortgeschrittenen Themen werden nicht behandelt. - Performance-Optimierung von Kernen - Optimierung von Speicherzugriffen ("coalescing") - CUDA Arrays, 3D-Speicher - CUDA Hardware im Detail, z. B. Warp-Scheduling - Vermeidung von Divergenz - Neuerungen bei Fermi, z. B. Caching - CUDA Driver API Fortgeschrittene Kenntnisse in C oder C++ sind notwendig, d.h. man sollte schon Programme geschrieben und gelesen haben. Auch sind Erfahrungen auf der Kommandozeile oder einer C-Entwicklungsumgebung von Vorteil. Grundwissen über parallele Algorithmen ist nicht schlecht, aber nicht unbedingt notwendig. Und hier noch eine stichwortartige Auflistung, um die Bewertung 4 von 5 zu rechtfertigen. Positiv - Gute Beispiele, gute Erklärungen, langsames Tempo - Die Autoren benutzen erst Grid-Blöcke und im zweiten Schritt Thread-Blöcke, um Parallelität einzuführen. Das ist eine gute Idee und das pädagogisch Sinnvollste. - Schöne Beispiele zum Anschauen, wie z. B. Julia-Mengen, Ray Tracer als Beispiel für konstanten Speicher, Shared Memory und Bitmaps, Wärmeübertragung für 1D- und 2D-Texturen - Schöne Einleitung mit Historie und Anwendungsbeispielen - Installation des Treibers, des Toolkits und des SDKs werden erklärt - Der Quellcode ist herunterladbar und funktioniert (meistens) auch - Gutes Beispiele für atomare Operationen im Anhang (Hash-Tabelle) Negativ - Der Code wurde immer vollständig abgedruckt. Hier kommt es im Laufe des Buchs zu Wiederholungen. Für Einsteiger mag diese Entscheidung richtig sein, den Fortgeschrittenen nervt es, eine Funktionsdefinition mehr als einmal lesen zu müssen. Ganz schlimm ist dieses bei den Ereignissen / Events, wenn der gesamte Ray Tracer noch zweimal abgedruckt wird (einmal mit globalem, einmal mit konstantem Speicher). - Sehr schnell veraltende Informationen wurden abgedruckt, z. B. die Liste der Devices auf S. 15f. - Die Dateinamen der Beispielprogramme stehen nicht im Buch. Man muss daher manchmal ein wenig suchen. - Abschnitt 3.3 'Querying Devices' ist zu diesem Zeitpunkt zu ausführlich. Das will man nicht alles wissen. Man kann auch zu diesem Zeitpunkt mit 95% der Begriffe nichts anfangen. Da hätte auch ein Verweis zur Dokumentation genügt. - Für Anfänger wäre ein kurzer Abschnitt über grundlegende Design-Patterns bzw. parallele Algorithmen sehr praktisch. Fazit Wer CUDA lernen will, findet in diesem Buch die beste Einleitung, die es momentan gibt. 3 von 3 Kunden fanden die folgende Rezension hilfreich. Sehr gute Kursunterlagen von Dr. Chrilly Donninger Ich spiele mich schon seit einiger Zeit mit dem Gedanken ein CUDA Programm zu schreiben. Bisher fehlte mir das passende Problem dafür. Ich habe aber nun eine nichtlineare (und nicht konvexe) Portfolio-Optimierung mit Hilfe der Differential Evolution Heuristik geschrieben. Die könnte einen Speedup brauchen und sollte auch parallelisierbar sein. Im Vorwort schreibt BLAS-Papst J. Dongarra "This book is required reading for anyone working with accelerator-based computing systems". Wahrscheinlich kann man CUDA-Programmierung auch mit der sehr guten SDK-Dokumentation lernen. Dieses Buch ist aber sicher nützlich. Die Autoren beginnen wie es sich gehört mit "Hello World" und enden mit einem für massiv-paralleles Computing garstigen Problem: einer Hashtabelle. Das war eines der Hauptprobleme beim massiv-parallelen Schachprogramm Hydra (siehe [1]). Die Beispiele sind sehr gut aufgebaut, die Autoren haben eine didaktische Ader. Nachdem sie aus dem CUDA-Developer Team stammen kennen sie sich auch aus. Bei massiv-paralleler Programmierung kämpft man immer gegen die Kommunikations- und Memory-Latenz. Wie beschäftigt man die im Überfluss vorhandenen Recheneinheiten? Ohne auf die - sich ohnehin ndernden - Details der GPU-Hardware einzugehen behandeln die Autoren diesen zentralen Aspekt mit steigendem Schwierigkeitsgrad. Um dann im Anhang am Beispiel der Hashtabelle aufzuzeigen, dass sich manche Probleme einfach nicht effektiv parallelisieren lassen (es gibt allerdings gefinkelte Lösungen für das Problem. Das würde aber den Rahmen eines Lehrbuchs sprengen). Besonders gefallen hat mir, dass der Code vollständig abgedruckt ist. Ich will nicht zwischen Buch und dem Bildschirm hin- und herspringen. Die Autoren kommentieren ihn auch mit Akribie. Der Inhalt des Buches dient auch als (Video-)Skriptum für universitäre CUDA-Kurse. Die etwas flapsige Sprache ist diesem Zielpublikum angepasst. Mir gefällt der etwas subtilere Humor von D. Knuth besser. Die Autoren werden aber nie peinlich. Sehr viele Programmierbücher sind heutzutage zu Bildl-Anklick Anleitungen verkommen. Als Bücher- und Programmierernarr erfreut es mein Herz, dass noch richtige Bücher für richtige Programmierer geschrieben werden. Möglicherweise liegt es daran, dass CUDA auch nix für Bildl-Anklicker ist. Nachdem ein Problem vorhanden ist gibt es nach dem Studium dieses Buches keine Ausrede mehr die rümel hochzukrempeln und mit richtiger CUDA-Programmierung loszulegen. Werma segne uns was wird. Ein sehr gutes weiterführendes Buch ist [2]. [1] Chrilly Donninger, Ulf Lorenz: The Chess Monster Hydra. [2] D. Kirk, Wen-Mei Hwu: Programming Massively Parallel Processors. P.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.

Kurzbeschreibung CUDA is a computing architecture designed to facilitate the development of parallel programs. In conjunction with a comprehensive software platform, the CUDA Architecture enables programmers to draw on the immense power of graphics processing units (GPUs) when building high-performance applications. GPUs, of course, have long been available for demanding graphics and game applications. CUDA now brings this valuable resource to programmers working on applications in other domains, including science, engineering, and finance. No knowledge of graphics programming is required just the ability to program in a modestly extended version of C. CUDA by Example, written by two senior members of the CUDA software platform team, shows programmers how to employ this new technology. The authors introduce each area of CUDA development through working examples. After a concise introduction to the CUDA platform and architecture, as well as a quick-start guide to CUDA C, the book details the techniques and trade-offs associated with each key CUDA feature. Youll discover when to use each CUDA C extension and how to write CUDA software that delivers truly outstanding performance. Major topics covered include Parallel programming Thread cooperation Constant memory and events Texture memory Graphics interoperability Atomics Streams CUDA C on multiple GPUs Advanced atomics Additional CUDA resources All the CUDA software tools youll need are freely available for download from NVIDIA. <http://developer.nvidia.com/object/cuda-by-example.html>

Kurzbeschreibung CUDA is a computing architecture designed to facilitate the development of parallel programs. In conjunction with a comprehensive software platform, the CUDA Architecture enables programmers to draw on the immense power of graphics processing units (GPUs) when building high-performance applications. GPUs, of course, have long been available for demanding graphics and game applications. CUDA now brings this valuable resource to programmers working on applications in other domains, including science, engineering, and finance. No knowledge of graphics programming is required just the ability to program in a modestly extended version of C. CUDA by Example, written by two senior members of the CUDA software platform team, shows programmers how to employ this new technology. The authors introduce each area of CUDA development through working examples. After a concise introduction to the CUDA platform and architecture, as well as a quick-start guide to CUDA C, the book details the techniques and trade-offs associated with each key CUDA feature. Youll discover when to use each CUDA C extension and how to write CUDA software that delivers truly outstanding performance. Major topics covered include Parallel programming Thread cooperation Constant memory and events Texture memory Graphics interoperability Atomics Streams CUDA C on multiple GPUs Advanced atomics Additional CUDA resources All the CUDA software tools youll need are freely available for download from NVIDIA.

<http://developer.nvidia.com/object/cuda-by-example.html> Buchrckseite ""This book is required reading for anyone working with accelerator-based computing systems."" -From the Foreword by Jack Dongarra, University of Tennessee and Oak Ridge National Laboratory CUDA is a computing architecture designed to facilitate the development of parallel programs. In conjunction with a comprehensive software platform, the CUDA Architecture enables programmers to draw on the immense power of graphics processing units (GPUs) when building high-performance applications. GPUs, of course, have long been available for demanding graphics and game applications. CUDA now brings this valuable resource to programmers working on applications in other domains, including science, engineering, and finance. No knowledge of graphics programming is required-just the ability to program in a modestly extended version of C. "CUDA by Example, " written by two senior members of the CUDA software platform team, shows programmers how to employ this new technology. The authors introduce each area of CUDA development through working examples. After a concise introduction to the CUDA platform and architecture, as well as a quick-start guide to CUDA C, the book details the techniques and trade-offs associated with each key CUDA feature. You'll discover when to use each CUDA C extension and how to write CUDA software that delivers truly outstanding performance. Major topics covered include Parallel programming Thread cooperation Constant memory and events Texture memory Graphics interoperability Atomics Streams CUDA C on multiple GPUs Advanced atomics Additional CUDA resources All the CUDA software tools you'll need are freely available for download from NVIDIA. <http://developer.nvidia.com/object/cuda-by-example.html>