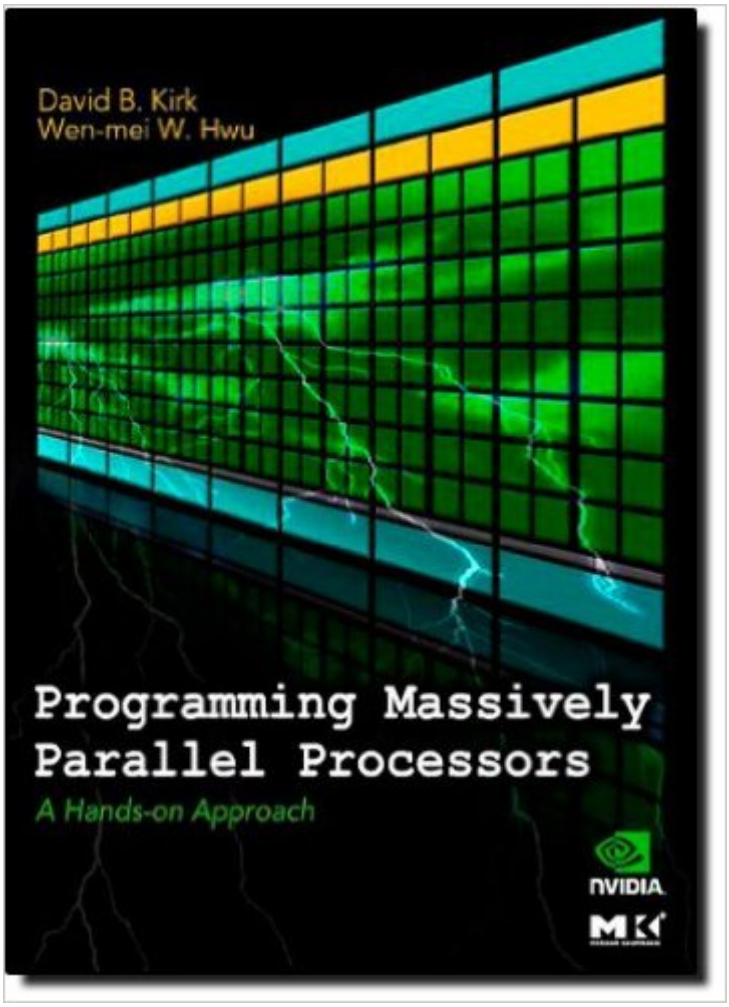
Programming Massively Parallel Processors: A Hands-on Approach (Applications Of GPU Computing Series)



Synopsis

Programming Massively Parallel Processors discusses basic concepts about parallel programming and GPU architecture. ""Massively parallel"" refers to the use of a large number of processors to perform a set of computations in a coordinated parallel way. The book details various techniques for constructing parallel programs. It also discusses the development process, performance level, floating-point format, parallel patterns, and dynamic parallelism. The book serves as a teaching guide where parallel programming is the main topic of the course. It builds on the basics of C programming for CUDA, a parallel programming environment that is supported on NVI- DIA GPUs. Composed of 12 chapters, the book begins with basic information about the GPU as a parallel computer source. It also explains the main concepts of CUDA, data parallelism, and the importance of memory access efficiency using CUDA. The target audience of the book is graduate and undergraduate students from all science and engineering disciplines who need information about computational thinking and parallel programming.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.

Book Information

Series: Applications of GPU Computing Series
Paperback: 280 pages
Publisher: Morgan Kaufmann; 1 edition (February 5, 2010)
Language: English
ISBN-10: 0123814723
ISBN-13: 978-0123814722
Product Dimensions: 7.5 x 0.6 x 9.2 inches
Shipping Weight: 1.2 pounds
Average Customer Review: 4.0 out of 5 starsÂ Â See all reviewsÂ (34 customer reviews)
Best Sellers Rank: #785,010 in Books (See Top 100 in Books) #56 inÂ Books > Computers & Technology > Hardware & DIY > Mainframes & Minicomputers #77 inÂ Books > Computers & Technology > Programming > Parallel Programming #125 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Computer Design

Customer Reviews

This book is a much better introduction to programming GPUs via CUDA than CUDA manual, or some presentation floating on the web. It is a little odd in coverage and language. You can tell it is written by two people with different command of English as well as passion. One co-author seems to be trying very hard to be colorful and looking for idiot-proof analogies but is prone to repetition. The other co-author sounds like a dry marketing droid sometimes. There are some mistakes in the codes in the book, but not too many since they don't dwell too long on code listings. In terms of coverage, I wish they'd cover texture memories, profiling tools, examples beyond simple matrix multiplication, and advice on computational thinking for codes with random access patterns. Chapters 6, 8, 9, and 10 are worth reading several times as they are full of practical tricks to use to trade one performance limiter for another in the quest for higher performance.

I think this book was written with the beginner in mind - if you're new to CUDA and having issues with understanding NVIDIA's documentation on the subject then this is the book to get. The author(s) took time to clarify and solidify some of the more difficult terms to understand e.g. memory bandwidth utilization, optimizing strategies but there are shortcomings in the book and two i could think of are typos (this really an issue cos it happens to every other book i've read) and the other would be using more examples to solidify concepts and illustrating them.In a nutshell, a great beginner's book but not a handbook sort of book.

This book fills a nice gap between the SDK samples, technical specifications, and online course content. If you are just getting started with GPGPU computing, this book leads you smoothly through the computation model, hardware architecture, and the programming model required to take advantage of the hardware.As others have pointed out, this is not a large book and fairly expensive. But, for the first book on the market it's surprisingly useful, effective, and readable. Definitely recommended for newcomers to the platform. Experienced GPGPU developers should only pick it up as a "hand out" for the people you need to train up, though.

As a beginning text this book has a significant advantage that beginning texts written for MPI, OpenMP, and so on don't have: there are 200 million CUDA-capable GPUs already deployed, and the odds are pretty good that most readers either have, or can readily get access to, a computer on which they can meaningfully learn parallel programming. If you are new to parallel programming and have access to a Tesla GPU, this book is a fine place to start your education. Readers already comfortable with parallel programming will find clear explanations of the Tesla GPU architecture and

the performance implications of its hardware features, as well as a solid introduction to the principles of programming in CUDA, though they'll probably do a lot of skimming over the already-familiar basics.

The book contains everything a C programmer needs to learn to be able to program NVIDIA GPUs using CUDA: the device architecture, the memory model, the execution model, and optimization techniques. Unfortunately, the book is also often infuriatingly repetitive and wordy. The authors belong to the tell-them-what-you-are-going-to-say-then-say-it-then-tell-them-what-you-said school of writing --- at the paragraph granularity. As a result, the last one or two sentences of a very great number of paragraphs are a mindless repetition of the first sentence of the same paragraph. And explanations are often far longer than they need to be, given that their target reader is a C programmer. For example, their multi-paragraph explanation of "row-major" order can be replaced with the single sentence "Just as in C; look at the picture." In another case, they provide an example of a topic using the "max" function, and then they actually go on to provide another (isomorphic) example using the "min" function! If you want to learn CUDA, you will learn it from this book, but be prepared to do a lot of skimming. Incidentally, the fact that CUDA requires a syntactic extension to C is a horrible botch.

Executive Summary - if you really want to dig into CUDA, go to the "CUDA Zone" on NVidia's web site. Also, this book concentrates on using CUDA on a single GPU.I think the target audience of this book is an undergraduate taking a CUDA or parallel programming class with the university supplying access to a pre-installed CUDA development system.This book is very readable (compared to the usual stuff programmers read). I particularly enjoyed the parts about GPU architecture and how various CUDA commands and structures map onto the architecture.As far as "hands-on" ... ummmm ... no. The code snippets look like they were taken from a linux system (or maybe windows with posix in there) but there isn't any real discussion about setting up a programming environment. To me, a true "hands-on" book should have the reader creating and running a "hello world" app ASAP. This book doesn't deliver that.An experienced professional (not just an MCSE or script kiddie) might enjoy this book, but not if the goal is to sling code under a deadline. In that case the CUDA zone is your friend.

The authors do a very good job of showing you how to write and run your own matrix-matrix multiplication GPU code in CUDA. However, if you don't know CUDA before hand, this book is

beyond frustrating. I tried to start with this book to start writing some scientific computing code and was nothing more than frustrated. I picked up CUDA by Example and after I learned about the basic structure of CUDA (which this book does not explain) I revisited this book and was much happier. This book is now a little dated as well since CUDA keeps being updated so rapidly though these gaps can be filled by getting the latest CUDA manual from NVIDIA (which also walks you though, in much less detail, how to write a matrix-matrix multiplication CUDA program).