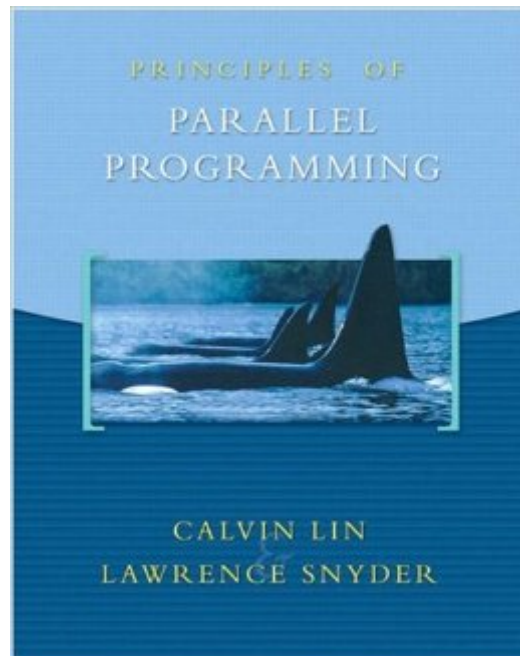# Principles Of Parallel Programming

Synopsis

With the rise of multi-core architecture, parallel programming is an increasingly important topic for software engineers and computer system designers. Written by well-known researchers Larry Snyder and Calvin Lin, this highly anticipated first edition emphasizes the principles underlying parallel computation, explains the various phenomena, and clarifies why these phenomena represent opportunities or barriers to successful parallel programming. Ideal for an advanced upper-level undergraduate course, Principles of Parallel Programming supplies enduring knowledge that will outlive the current hardware and software, aiming to inspire future researchers to build tomorrowâ ™s solutions.

Book Information

Hardcover: 352 pages

Publisher: Pearson; 1 edition (March 7, 2008)

Language: English

ISBN-10: 0321487907

ISBN-13: 978-0321487902

Product Dimensions: 7.3 x 1 x 9.2 inches

Shipping Weight: 1.6 pounds (View shipping rates and policies)

Average Customer Review: 3.3 out of 5 starsÂ Â See all reviewsÂ (6 customer reviews)

Best Sellers Rank: #367,139 in Books (See Top 100 in Books)   #41 inÂ Books > Computers & Technology > Programming > Parallel Programming   #1086 inÂ Books > Computers & Technology > Hardware & DIY   #2630 inÂ Books > Computers & Technology > Programming > Languages & Tools

Customer Reviews

I have done parallel programming on a variety of machines for many years, and have written some widely-used parallel numerical software. Now that I have graduate students of my own and teach courses in numerical and parallel computation, I've been hoping for a book like this to help my students understand the basic techniques, concepts, and problems common to most parallel programming, as well as to use as a reference for courses, without resorting to manuals bogged down in the details of specific architectures. A colleague of mine (who has a large company developing parallel tools and who for many years has taught a course on parallel scientific computing) pointed me to this text, and I'm much more pleased with it than with any recent book on the subject that I can recall.The authors of this book clearly introduce key concepts of extracting

parallelism, load balancing, performance analysis, and memory management with a number of well-selected examples and advice clearly stemming from long experience in the field. They describe numerous general principles in an accessible way, without getting bogged down in the theoretical models of dubious utility that are too common in this field. The book is timely, in that it exhibits a clear awareness of current architectural trends, but remains rightly focused on timeless ideas.I suppose the authors cannot be blamed for devoting a chapter to the parallel programming language they have developed in their own work (ZPL), and it is balanced by chapters on the current popular low-level techniques like MPI and threads as well as brief discussions of other proposed high-level languages (although the mention of Cilk is a bit too brief for my tastes). But the real strength of the book is that it is not tied too closely to any particular language or implementation, and instead helps you to recognize fundamental ideas as they appear in various forms.I do wish the book were a bit cheaper, but high textbook prices seem to be a fact of life. A more basic introduction to caches, and the connection between memory locality on serial computers and locality on parallel machines, would probably be helpful. The mention of the powerful idea of work stealing is too brief. And I'm sure I'll find many other things I dislike as I continue to use this book, but overall I'm quite happy with this book as a way to get students into this subject.

Years ago I briefly worked on a hypercube, and when I got this book, I wondered how it had fared. Alas, the hypercube, at least under this name, rated no mention. Though there is a passing reference to a binary 3 cube which is a 3 dimensional hypercube.The authors explain the current state of multiprocessor architectures. The few remaining computer CPU makers have efforts in this field. Intel, AMD, Sun and IBM. The book describes qualitatively the salient aspects of each. One nice thing about the discussion is that it focuses on this, without drowning you in unnecessary hardware details. This turns out to be a key theme of the book. It abstracts out essential hardware properties, so that you can appreciate these and apply the book's ideas without being tied to any given chip.The book also describes an important type of multiprocessor. Cluster machines, where each node is typically some off the shelf CPU, buffed up with a lot of local memory. The key differences between clusters are often related to how the nodes are hooked to each other, by some type of bus or crossbar. Affordability is an important property of clusters; thus the maximal use of commodity hardware. (The hypercube that I mentioned earlier would be a cluster.)For a programmer, there is one overriding idea that you should get from the book. For optimal performance, minimise the internodal communication, compared to the use of a node's cache. The access time of the former can be 2-5 orders of magnitude slower. Details vary with the given

architectures, of course. But typically nothing else comes close, in terms of effects on your throughput.

As a professor, I use this book in my classes since it covers all the main topics in Parallel Programming while keeping an adequate balance between programming (my main interest) and computer architecture.

Book is alright, covers a lot of concepts, provides background discussions - but has many drawbacks, maybe not the best, especially given its ridiculous price tag.

This book is overpriced and the content is very very dated. Who wants to learn ZPL in today's world? If you want to learn about parallelism pickup a copy of Patterns for Parallel Programming Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill. Don't waste ur money on this book.

This book is absolutely riddled with errors. The sheer density of errors make the book unusable for any in-depth study.

Download to continue reading...

Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition) Parallel Programming with Intel Parallel Studio XE Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation Introduction to Parallel Computing: Design and Analysis of Parallel Algorithms Short Stories in Spanish: New Penguin Parallel Text (New Penguin Parallel Texts) (Spanish and English Edition) Learn German: Parallel Text - Easy, Funny Stories (German - English) - Bilingual (Learning German with Parallel Text Book 1) Learn German III: Parallel Text - Easy Stories (German - English) Bilingual - Dual Language (Learning German with Parallel Text 3) (German Edition) Principles of Parallel Programming Java: The Simple Guide to Learn Java Programming In No Time (Programming,Database, Java for dummies, coding books, java programming) (HTML,Javascript,Programming,Developers,Coding,CSS,PHP) (Volume 2) CoArrays: Parallel Programming in Fortran (Chapman & Hall/CRC Computational Science) P-Prolog: A Parallel Logic Programming Language (World Scientific Series in Computer Science) CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing) Foundations of Multithreaded, Parallel, and Distributed Programming Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series) An Introduction to

Parallel Programming Structured Parallel Programming: Patterns for Efficient Computation Python Parallel Programming Cookbook Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation) Using MPI - 2nd Edition: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation) Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures (Patterns & Practices)

Dmca