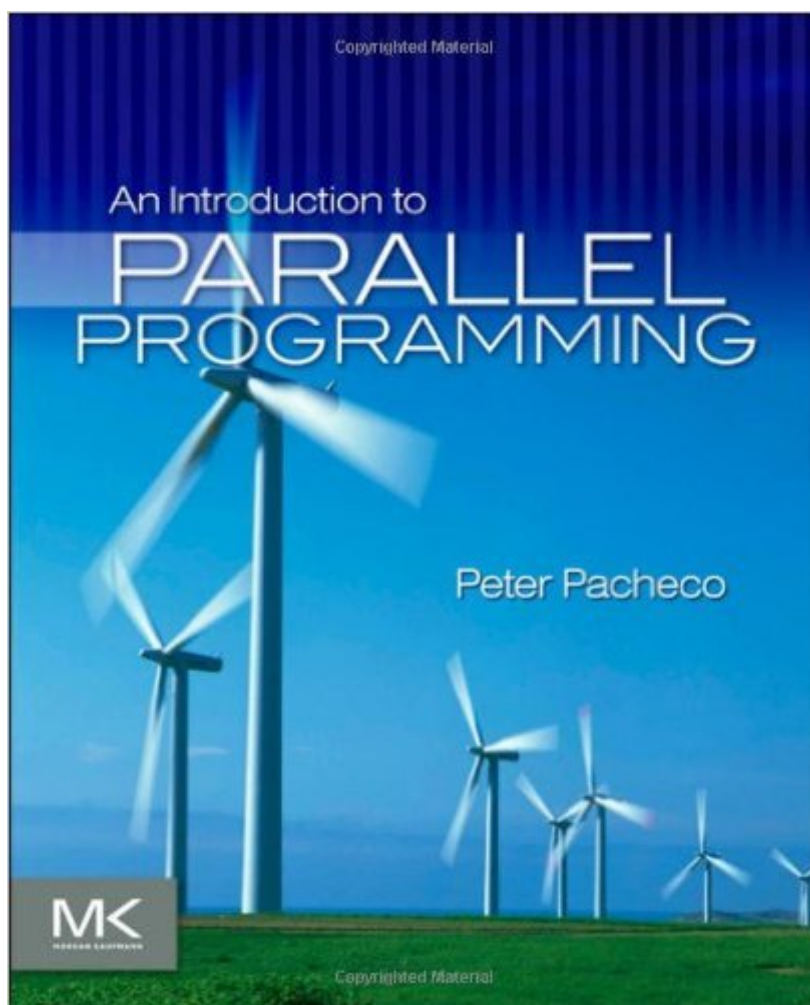


The book was found

# An Introduction To Parallel Programming



## Synopsis

Author Peter Pacheco uses a tutorial approach to show students how to develop effective parallel programs with MPI, Pthreads, and OpenMP. The first undergraduate text to directly address compiling and running parallel programs on the new multi-core and cluster architecture, *An Introduction to Parallel Programming* explains how to design, debug, and evaluate the performance of distributed and shared-memory programs. User-friendly exercises teach students how to compile, run and modify example programs. Takes a tutorial approach, starting with small programming examples and building progressively to more challenging examples. Focuses on designing, debugging and evaluating the performance of distributed and shared-memory programs. Explains how to develop parallel programs using MPI, Pthreads, and OpenMP programming models -- This text refers to the Paperback edition.

## Book Information

Hardcover: 392 pages

Publisher: Morgan Kaufmann; 1 edition (January 21, 2011)

Language: English

ISBN-10: 0123742609

ISBN-13: 978-0123742605

Product Dimensions: 7.7 x 0.9 x 9.3 inches

Shipping Weight: 2.2 pounds (View shipping rates and policies)

Average Customer Review: 4.0 out of 5 stars See all reviews (21 customer reviews)

Best Sellers Rank: #248,419 in Books (See Top 100 in Books) #27 in Books > Computers & Technology > Programming > Parallel Programming #564 in Books > Computers & Technology > Programming > Introductory & Beginning #1146 in Books > Textbooks > Computer Science > Programming Languages

## Customer Reviews

This book is a good start for anyone that had done quite a bit of programming and is confident in their abilities to program traditional non-parallel programs and wants to learn parallel programming. Good intro and review of material you will have learned in any computer architecture course in first chapters and reasons why you should learn parallel programming. In later chapters you learn about task-parallelism vs data-parallelism in learning to rewrite serial program for parallel. The two main types of parallel systems, shared-memory, and distributed-memory are covered and explained. MPI is used for the distributed-memory programming. Pthreads and OpenMP is used for

the Shared-Memory programming examples. Examples like the Trapezoid Rule and sorting algorithms that lend themselves to parallel programming are demonstrated using MPI, Pthreads and OpenMP. There is also the pi approximation example using alternating series that I remember coding in my first programming course and wondering why it was so slow. It's used in the book to warn of the dangers of improperly scoping variables in parallel blocks. MPI does seem to be favored over the other two for some reason that I didn't get though. OpenMP seems to be the easiest to use and works out of the box on Windows, Linux, and Mac OS X with little to no extra work too so I don't know why the author didn't start off with OpenMP first. You'll need to be running Linux to be able to easily follow along with all the examples in the book if you are a hands on person like myself and want to actually compile and run all the example yourself. On Windows, to build MPI programs, you need the Windows HPC Server 2008 SDK installed. You can run the Pthreads examples on the Mac too but on Windows you'll need to install the Unix for Windows. I did get the sample code to work pretty easily using OpenMP. All the examples were text based it was also easy to time them and verify that the parallel version of the programs were faster. It would've been nice to see some graphical examples but I'm guessing OpenCL, CUDA or some other API is better suited for that since I've seen graphical examples using those API's. The final chapters does provide some guidance to which API to choose and their strengths and weaknesses though. Since the book is intended for self-study it sure doesn't provide much help on setting up your development environment for parallel programming and seems to assume you'll figure it out on your own. Also, no answers were provided for any of the end of chapter problems which would really be helpful for self-study. The book website does provide some excellent slides to go with the book though. Finally, parallel programming with GPU's seems to be gaining quite a bit of popularity lately but this book seems to totally ignore the topic except to refer you to other books on the subject.

I bought this book for a dual purpose: To use in a post-grad class I am taking, and to help me build a distributed systems that I am currently developing. The book is pedagogically exquisite. The presentation and explanations are ordered, logical and non-assuming, and that is where this books shines - and probably why it is getting all the good reviews here and elsewhere. However, the book stands very short of being practical to anything but the simplest of applications - compare it to other books that have similar titles, even with the word introduction in the title and you immediately see the difference. Having said that, I think this the best place to start, it the most approachable introduction to the subject, but in my opinion, it could have accommodated more content either on the design and analysis of parallel algorithms or on the details of implementing more than plain

vanilla concurrent solutions.

I am a first year graduate student with no computer science background, other than one class I took five years ago on scientific computing where we used fortran and focused on writing simple programs to solve simple physical science problems. I am working in theoretical chemistry/physics and required to program extensively. One area we are focusing on is programming in parallel to get better performance. This text has been invaluable to me as an introduction to the field of parallel computing. While not complete self-contained (he states in the introduction that it is assumed the reader has taken an introductory computer science course) it is not hard for one mostly unfamiliar with computer science terms to come up to speed. His explanations are clear and concise and the examples he expounds clarify the topics masterfully. The exercises are reasonable and well thought out to help the reader gain a better understanding of how to efficiently program in parallel. One particularly elucidating example for me, which he comes back to more than once throughout the text, is that of computing a sum. In the first chapter he uses this example to show that substantial speedups can be obtained by not just writing the program to run in parallel, but making the best use of resources. As a novice to computer science this was one my "ah-ha!" moments I had while reading this book and working through the exercises. This will forever be an invaluable addition to my library, and I highly recommend it for the uninitiated. The only drawback is that it is written with the C language in mind, we still use fortran77 for several reasons, though this was only minor. With a freely available online text on C and other references on MPI and openMP in fortran I have been able to make a smooth transition for the work that I do. So even if you don't know C, or use it, this is still a useful source and reference.

[Download to continue reading...](#)

Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition) Parallel Programming with Intel Parallel Studio XE Introduction to Parallel Computing: Design and Analysis of Parallel Algorithms Parallel Scientific Computing in C++ and MPI: A Seamless Approach to Parallel Algorithms and their Implementation 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) An Introduction to 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) Structured Parallel Programming: Patterns for Efficient Computation Principles of Parallel Programming 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](#)