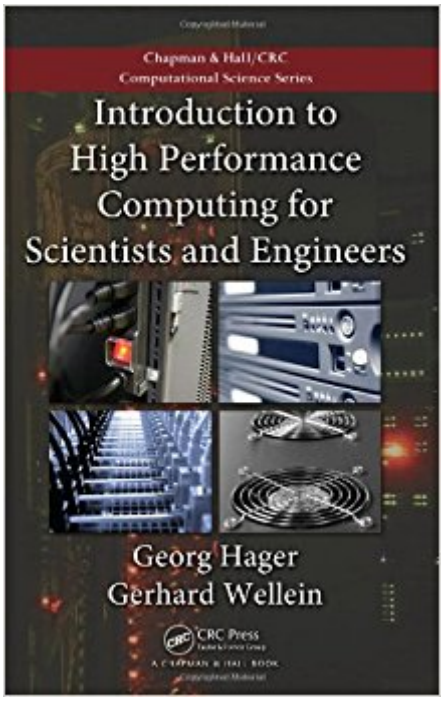
Introduction To High Performance Computing For Scientists And Engineers (Chapman & Hall/CRC Computational Science)



Synopsis

Written by high performance computing (HPC) experts, Introduction to High Performance Computing for Scientists and Engineers provides a solid introduction to current mainstream computer architecture, dominant parallel programming models, and useful optimization strategies for scientific HPC. From working in a scientific computing center, the authors gained a unique perspective on the requirements and attitudes of users as well as manufacturers of parallel computers. The text first introduces the architecture of modern cache-based microprocessors and discusses their inherent performance limitations, before describing general optimization strategies for serial code on cache-based architectures. It next covers shared- and distributed-memory parallel computer architectures and the most relevant network topologies. After discussing parallel computing on a theoretical level, the authors show how to avoid or ameliorate typical performance problems connected with OpenMP. They then present cache-coherent nonuniform memory access (ccNUMA) optimization techniques, examine distributed-memory parallel programming with message passing interface (MPI), and explain how to write efficient MPI code. The final chapter focuses on hybrid programming with MPI and OpenMP. Users of high performance computers often have no idea what factors limit time to solution and whether it makes sense to think about optimization at all. This book facilitates an intuitive understanding of performance limitations without relying on heavy computer science knowledge. It also prepares readers for studying more advanced literature. Read about the authorsÃ¢â ¬â„¢ recent honor: Informatics Europe Curriculum Best Practices Award for Parallelism and Concurrency

Book Information

Series: Chapman & Hall/CRC Computational Science (Book 7)

Paperback: 356 pages

Publisher: CRC Press; 1 edition (July 4, 2010)

Language: English

ISBN-10: 143981192X

ISBN-13: 978-1439811924

Product Dimensions: 5.9 x 0.9 x 9.1 inches

Shipping Weight: 1.3 pounds (View shipping rates and policies)

Average Customer Review: 4.3 out of 5 stars 8 customer reviews

Best Sellers Rank: #129,477 in Books (See Top 100 in Books) #10 inÃ Â Books > Computers & Technology > Programming > Parallel Programming #11 inÃ Â Books > Science & Math >

Mathematics > Number Systems #25 inÃ Â Books > Science & Math > Physics > Waves & Wave Mechanics

Customer Reviews

Georg Hager and Gerhard Wellein have developed a very approachable introduction to high performance computing for scientists and engineers. Their style and description is easy to read and follow. Ã¢â ¬Â¦ This book presents a balanced treatment of the theory, technology, architecture, and software for modern high performance computers and the use of high performance computing systems. The focus on scientific and engineering problems makes this both educational and unique. I highly recommend this timely book for scientists and engineers. I believe this book will benefit many readers and provide a fine reference.Ã¢â ¬â ¢From the Foreword by Jack Dongarra, University of Tennessee, Knoxville, USA

Georg Hager is a senior research scientist in the high performance computing group of the Erlangen Regional Computing Center at the University of Erlangen-Nuremberg in Germany. Gerhard Wellein leads the high performance computing group of the Erlangen Regional Computing Center and is a professor in the Department for Computer Science at the University of Erlangen-Nuremberg in Germany.

This book assumes you have a lot of background knowledge in CS both in coding and hardware makeup of a computer.And it is written in a way that is so much denser than it needs to be. For example, it is very annoying how uses vocabulary like buses before he defines them and doesn't write about paging in any sort of illuminating way at all, but still goes on to talk about the problems with it. The concepts themselves are very easy once you slowly get through the passage. It's no calculus for first time learners or 5 star sudoku logic, but I wonder if it is the overabundance of jargon that obfuscates or his extremely dry writing style (as opposed to something like Griffiths texts).It's the first time I have read such a book, and it is okay. I learned something, but there is something to be said about his writing style if online resources were significantly more illuminating using similar amounts of jargon for various people who had/have currently varying degrees of knowledge of general programming/HPC knowledge as well as myself.

I have read many texts covering high performance and parallel programming, but none of them provides the fundamental coverage of the core concepts as well as this one. The authors' emphasis

on maximizing performance of serial programs _before_ applying parallelization strategies is something drastically missing from every other text on parallel programming I have read. Additionally, their discussion of ccNUMA was a breath of fresh air. Most texts never mention it, but this one has a whole chapter! Each chapter also has a set of questions that extend the concepts covered therein. The solutions are in the back, so it may not be best suited as a classroom text (although it certainly could) but it makes self-study very easy.The choice of using FORTRAN for most of the code examples is, in my mind, unfortunate but understandable as the array syntax in that language does simplify demonstrating certain programming features. There is also C++ code sprinkled here and there, but contains disproportionately more problems than the FORTRAN code (e.g., the use of static_cast instead of reinterpret_cast in the discussion of allocators and placement new). It would have been nice to see a deeper coverage of SIMD programming, as well. However, my biggest complaint is that "classical" coverage of OpenMP for shared memory programming and MPI for distributed memory programming. Indeed, these two technologies are ubiquitous in the HPC world, but there are many great technologies that provide several new features and fill in the gaps left by these classic tools. Yet little to no mention is given to them (e.g., Intel's TBB, Cilk Plus, C++11's threading library, etc.).

I have learned some basic, fundamental facts from this book, despite the fact that I've been producing (good) code for some 30 years. For instance, that the leading consideration in optimization is memory management--- getting data from RAM to the CPU--- this is orders of magnitude more important than number of mathematical operations. I also learned that while the CPU does its best to cache and pipeline operations by guessing what I'm going to call next, the compiler doesn't help as much as I thought it did. (For instance, loop unrolling is considered an advanced optimization, not usually performed). This book has effectively suggested a few experiments I can perform with my code to see if I can get it more into a form that the CPU expects and maybe gain a factor of 10 or 100 on tight loops. (Plots of throughput in the book have sharp edges of at least that magnitude.)

Thanks! Arrived as advertised.

Excellent overview on many relevant aspects of HPSC, including hardware, algorithm design, and parallel processing.

I would give this 4.5 stars if I could. I find it to be pretty well written but not always organized the greatest. This means if you want to study a certain concept (e.g., OpenMP) you need to hop around the book a lot. This is a stylistic thing -- clearly the authors just think of organizing things differently than me, but I would prefer a different ordering. Having said that, all of the material is there and helpful.

This is a great guide for someone looking to optimize code. I find that their way of presenting the material is great: they start off with basic performance optimizations that can be applied to serial code and then go on to address parallel code. Even computer scientists who are already familiar with the relevant computer architecture issues can benefit.

There are not many books available on the subject of high-performance programming, but this is the best of them, in part due to it being up-to-date. I also ownÃ Â Introduction to Parallel Computing (Oxford Texts in Applied and Engineering Mathematics),High Performance Computing (RISC Architectures, Optimization & Benchmarks), andÃ Â Performance Optimization of Numerically Intensive Codes (Software, Environments and Tools), all of which I like, but none of these address the current generation of computer hardware.This book should be required reading for anyone who programs supercomputers or needs to write performance-critical scientific code. I find the examples to be relevant and well-written; I currently use them in workshops at the Argonne Leadership Computing Facility.The topics covered include serial optimization, OpenMP programming and MPI programming. I find the serial optimization and treatment of memory hierarchies to be the most useful. The OpenMP coverage is also good and not duplicated elsewhere. There is nothing wrong with the treatment of MPI other than it is too short to be useful for some.This book intentionally overlooks performance issues related to programming language, particularly C++, so if that is your interest, there are plenty of other good books. Most of the examples are in Fortran but I have no trouble reproducing them in C.

Hall/CRC Mathematical and Computational Biology) Computational Statistics Handbook with MATLAB, Third Edition (Chapman & Hall/CRC Computer Science & Data Analysis) Flexible, Reliable Software: Using Patterns and Agile Development (Chapman & Hall/CRC Textbooks in Computing) Statistical Computing with R (Chapman & Hall/CRC The R Series) Making Music with Computers: Creative Programming in Python (Chapman & Hall/CRC Textbooks in Computing) Statistics and Data Analysis for Microarrays Using R and Bioconductor, Second Edition (Chapman & Hall/CRC Mathematical and Computational Biology) Variational Methods in Image Processing (Chapman & Hall/CRC Mathematical and Computational Imaging Sciences Series) RNA-seq Data Analysis: A Practical Approach (Chapman & Hall/CRC Mathematical and Computational Biology) Design of Experiments: An Introduction Based on Linear Models (Chapman & Hall/CRC Texts in Statistical Science) Modern Data Science with R (Chapman & Hall/CRC Texts in Statistical Science) Measure and Integral: An Introduction to Real Analysis, Second Edition (Chapman & Hall/CRC Pure and Applied Mathematics) Introduction to Set Theory, Third Edition, Revised and Expanded (Chapman & Hall/CRC Pure and Applied Mathematics) Introduction to Scientific Programming and Simulation Using R (Chapman & Hall/CRC The R Series) Introduction to Scientific Programming and Simulation Using R, Second Edition (Chapman & Hall/CRC The R Series) Introduction to Modern Cryptography, Second Edition (Chapman & Hall/CRC Cryptography and Network Security Series) Modeling and Analysis of Stochastic Systems, Second Edition (Chapman & Hall/CRC Texts in Statistical Science)