
Software Optimization Cookbook Second Edition

This is likewise one of the factors by obtaining the soft documents of this Software Optimization Cookbook Second Edition by online. You might not require more times to spend to go to the book launch as well as search for them. In some cases, you likewise do not discover the declaration Software Optimization Cookbook Second Edition that you are looking for. It will unquestionably squander the time.

However below, behind you visit this web page, it will be appropriately extremely simple to get as well as download lead Software Optimization Cookbook Second Edition

It will not bow to many era as we notify before. You can do it while pretense something else at home and even in your workplace. appropriately easy! So, are you question? Just exercise just what we allow under as well as review Software Optimization Cookbook Second Edition what you behind to read!



Software Optimization for High-performance Computing Springer

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 author *Working Effectively with Legacy Code* CRC Press

Last Updated: December 2020
Based on Julia v1.3+ and JuMP v0.21+ The main motivation of writing this book was to help the

author himself. He is a professor in the field of operations research, and his daily activities involve building models of mathematical optimization, developing algorithms for solving the problems, implementing those algorithms using computer programming languages, experimenting with data, etc. Three languages are involved: human language, mathematical language, and computer language. His team of students need to go over three different languages, which requires "translation" among the three languages. As this book was written to teach his research group how to translate, this book will also be useful for anyone who needs to learn how to translate in a similar situation. The Julia Language is as fast as C, as convenient as MATLAB, and as general as Python with a flexible algebraic modeling language for mathematical

optimization problems. With the great support from Julia developers, especially the developers of the JuMP—Julia for Mathematical Programming—package, Julia makes a perfect tool for students and professionals in operations research and related areas such as industrial engineering, management science, transportation engineering, economics, and regional science.

For more information, visit:

<http://www.chkwon.net/julia>

The Software Optimization Cookbook

Morgan Kaufmann

The field of Chemical Engineering and its link to computer science is in constant evolution and new engineers have a variety of tools at their disposal to tackle their everyday problems. *Introduction to Software for Chemical Engineers, Second Edition* provides a quick guide to the use of various computer packages for chemical engineering applications. It covers a range of software applications from Excel and general mathematical packages such as MATLAB and MathCAD to process simulators, CHEMCAD and ASPEN, equation-based modeling languages, gProms, optimization software such as GAMS and AIMS, and specialized software like CFD or DEM codes. The different packages are introduced and applied to solve typical problems in fluid mechanics, heat and mass transfer, mass and energy balances, unit operations, reactor engineering, process and equipment design and control. This new edition offers a wider view of packages including open source software such as R, Python and Julia. It also includes complete examples in ASPEN Plus,

adds ANSYS Fluent to CFD codes, Lingo to the optimization packages, and discusses Engineering Equation Solver. It offers a global idea of the capabilities of the software used in the chemical engineering field and provides examples for solving real-world problems. Written by leading experts, this book is a must-have reference for chemical engineers looking to grow in their careers through the use of new and improving computer software. Its user-friendly approach to simulation and optimization as well as its example-based presentation of the software, makes it a perfect teaching tool for both undergraduate and master levels.

High Performance MySQL CRC Press

By using computer simulations in research and development, computational science and engineering (CSE) allows empirical inquiry where traditional experimentation and methods of inquiry are difficult, inefficient, or prohibitively expensive. *The Handbook of Research on Computational Science and Engineering: Theory and Practice* is a reference for interested researchers and decision-makers who want a timely introduction to the possibilities in CSE to advance their ongoing research and applications or to discover new resources and cutting edge developments. Rather than reporting results obtained using CSE models, this comprehensive survey captures the architecture of the cross-disciplinary field, explores the long term implications of technology choices, alerts readers to the hurdles facing CSE, and identifies trends in future development.

High Performance Computing John Wiley & Sons

Optimize code for multi-core processors with

Intel's Parallel Studio Parallel programming is rapidly becoming a "must-know" skill for developers. Yet, where to start? This teach-yourself tutorial is an ideal starting point for developers who already know Windows C and C++ and are eager to add parallelism to their code. With a focus on applying tools, techniques, and language extensions to implement parallelism, this essential resource teaches you how to write programs for multicore and leverage the power of multicore in your programs. Sharing hands-on case studies and real-world examples, the authors examine the challenges of each project and show you how to overcome them. Explores conversion of serial code to parallel Focuses on implementing Intel Parallel Studio Highlights the benefits of using parallel code Addresses error and performance optimization of code Includes real-world scenarios that illustrate the techniques of advanced parallel programming situations Parallel Programming with Intel Parallel Studio dispels any concerns of difficulty and gets you started creating faster code with Intel Parallel Studio.

Optimization Methods in Finance

Morgan Kaufmann

Programming multi-core and many-core computing systems Sabri Pllana, Linnaeus University, Sweden Fatos Xhafa, Technical University of Catalonia, Spain Provides state-of-the-art methods for programming multi-core and many-core systems The book comprises a selection of twenty two chapters covering: fundamental techniques and algorithms; programming approaches; methodologies and frameworks; scheduling and management; testing and evaluation methodologies; and case studies for programming multi-core and many-core systems. Program development for multi-core processors, especially for heterogeneous multi-core processors, is significantly more complex than for single-core processors. However, programmers have been traditionally trained for the development of sequential programs, and

only a small percentage of them have experience with parallel programming. In the past, only a relatively small group of programmers interested in High Performance Computing (HPC) was concerned with the parallel programming issues, but the situation has changed dramatically with the appearance of multi-core processors on commonly used computing systems. It is expected that with the pervasiveness of multi-core processors, parallel programming will become mainstream. The pervasiveness of multi-core processors affects a large spectrum of systems, from embedded and general-purpose, to high-end computing systems. This book assists programmers in mastering the efficient programming of multi-core systems, which is of paramount importance for the software-intensive industry towards a more effective product-development cycle. Key features: Lessons, challenges, and roadmaps ahead. Contains real world examples and case studies. Helps programmers in mastering the efficient programming of multi-core and many-core systems. The book serves as a reference for a larger audience of practitioners, young researchers and graduate level students. A basic level of programming knowledge is required to use this book.

Convex Optimization Packt Publishing Ltd Explains how compilers translate high-level language source code (like code written in Python) into low-level machine code (code that the computer can understand) to help readers understand how to produce the best low-level, computer readable machine code. In the beginning, most software was written in assembly, the CPU's low-level language, in order to achieve acceptable performance on relatively slow hardware. Early programmers were sparing in their use of high-level language code, knowing

that a high-level language compiler would generate crummy, low-level machine code for their software. Today, however, many programmers write in high-level languages like Python, C/C++/C#, Java, Swift. The result is often sloppy, inefficient code. But you don't need to give up the productivity and portability of high-level languages in order to produce more efficient software. In this second volume of the Write Great Code series, you'll learn:

- How to analyze the output of a compiler to verify that your code does, indeed, generate good machine code
- The types of machine code statements that compilers typically generate for common control structures, so you can choose the best statements when writing HLL code
- Just enough 80x86 and PowerPC assembly language to read compiler output
- How compilers convert various constant and variable objects into machine data, and how to use these objects to write faster and shorter programs

NEW TO THIS EDITION, COVERAGE OF:

- Programming languages like Swift and Java
- Code generation on modern 64-bit CPUs
- ARM processors on mobile phones and tablets
- Stack-based architectures like the Java Virtual Machine
- Modern language systems like the Microsoft Common Language Runtime

With an understanding of how compilers work, you'll be able to write source code that they can translate into elegant machine code. That understanding starts right here, with *Write Great Code, Volume 2: Thinking Low-Level, Writing High-Level*.

Docker Cookbook - Second Edition SIAM

Learn to use IPython and Jupyter

Notebook for your data analysis and

visualization work. Key Features Leverage

the Jupyter Notebook for interactive data

science and visualization Become an

expert in high-performance computing and

visualization for data analysis and scientific

modeling A comprehensive coverage of scientific computing through many hands-on, example-driven recipes with detailed, step-by-step explanations Book Description Python is one of the leading open source platforms for data science and numerical computing. IPython and the associated Jupyter Notebook offer efficient interfaces to Python for data analysis and interactive visualization, and they constitute an ideal gateway to the platform. IPython Interactive Computing and Visualization Cookbook, Second Edition contains many ready-to-use, focused recipes for high-performance scientific computing and data analysis, from the latest IPython/Jupyter features to the most advanced tricks, to help you write better and faster code. You will apply these state-of-the-art methods to various real-world examples, illustrating topics in applied mathematics, scientific modeling, and machine learning. The first part of the book covers programming techniques: code quality and reproducibility, code optimization, high-performance computing through just-in-time compilation, parallel computing, and graphics card programming. The second part tackles data science, statistics, machine learning, signal and image processing, dynamical systems, and pure and applied mathematics. What you will learn Master all features of the Jupyter Notebook Code better: write high-quality, readable, and well-tested programs; profile and optimize your code; and conduct reproducible interactive computing experiments Visualize data and create interactive plots in the Jupyter Notebook Write blazingly fast Python programs with NumPy, ctypes, Numba, Cython, OpenMP, GPU programming (CUDA), parallel IPython, Dask, and more Analyze data with Bayesian or frequentist statistics (Pandas, PyMC, and R), and learn from actual data through machine learning (scikit-learn) Gain

valuable insights into signals, images, and sounds with SciPy, scikit-image, and OpenCV Simulate deterministic and stochastic dynamical systems in Python Familiarize yourself with math in Python using SymPy and Sage: algebra, analysis, logic, graphs, geometry, and probability theory Who this book is for This book is intended for anyone interested in numerical computing and data science: students, researchers, teachers, engineers, analysts, and hobbyists. A basic knowledge of Python/NumPy is recommended. Some skills in mathematics will help you understand the theory behind the computational methods.

Embedded Computing for High Performance Morgan Kaufmann Publishers

Mathematics of Computing -- Numerical Analysis.

Business Optimization Using Mathematical Programming Packt Publishing Ltd

If you are a Python developer with some experience of working on scientific, mathematical, and statistical applications and want to gain an expert understanding of NumPy programming in relation to science, math, and finance using practical recipes, then this book is for you.

Introduction to Software for Chemical Engineers, Second Edition CRC Press

The field of Chemical Engineering and its link to computer science is in constant evolution and new engineers have a variety of tools at their disposal to tackle their everyday problems.

Introduction to Software for Chemical Engineers, Second Edition provides a quick guide to the use of various computer packages for chemical engineering applications. It covers a range of software applications from

Excel and general mathematical packages such as MATLAB and MathCAD to process simulators, CHEMCAD and ASPEN, equation-based modeling languages, gProms, optimization software such as GAMS and AIMS, and specialized software like CFD or DEM codes. The different packages are introduced and applied to solve typical problems in fluid mechanics, heat and mass transfer, mass and energy balances, unit operations, reactor engineering, process and equipment design and control. This new edition offers a wider view of packages including open source software such as R, Python and Julia. It also includes complete examples in ASPEN Plus, adds ANSYS Fluent to CFD codes, Lingo to the optimization packages, and discusses Engineering Equation Solver. It offers a global idea of the capabilities of the software used in the chemical engineering field and provides examples for solving real-world problems. Written by leading experts, this book is a must-have reference for chemical engineers looking to grow in their careers through the use of new and improving computer software. Its user-friendly approach to simulation and optimization as well as its example-based presentation of the software, makes it a perfect teaching tool for both undergraduate and master levels.

Engineering a Compiler Elsevier

This book provides a complete and comprehensive reference/guide to Pyomo (Python Optimization Modeling Objects) for both beginning and advanced modelers, including students at the undergraduate and graduate

levels, academic researchers, and practitioners. The text illustrates the breadth of the modeling and analysis capabilities that are supported by the software and support of complex real-world applications. Pyomo is an open source software package for formulating and solving large-scale optimization and operations research problems. The text begins with a tutorial on simple linear and integer programming models. A detailed reference of Pyomo's modeling components is illustrated with extensive examples, including a discussion of how to load data from data sources like spreadsheets and databases. Chapters describing advanced modeling capabilities for nonlinear and stochastic optimization are also included. The Pyomo software provides familiar modeling features within Python, a powerful dynamic programming language that has a very clear, readable syntax and intuitive object orientation. Pyomo includes Python classes for defining sparse sets, parameters, and variables, which can be used to formulate algebraic expressions that define objectives and constraints. Moreover, Pyomo can be used from a command-line interface and within Python's interactive command environment, which makes it easy to create Pyomo models, apply a variety of optimizers, and examine solutions. The software supports a different modeling approach than commercial AML (Algebraic Modeling Languages) tools, and is designed for flexibility, extensibility, portability, and maintainability but also maintains the central ideas in modern AMLs.

Introduction to High Performance Computing for Scientists and Engineers SIAM

Get more out of your legacy systems: more performance, functionality, reliability, and manageability Is your code easy to change? Can you get nearly instantaneous feedback when you do change it? Do you understand it? If the answer to any of these questions is no, you have legacy code, and it is draining time and money away from your development efforts. In this book, Michael Feathers offers start-to-finish strategies for working more effectively with large, untested legacy code bases. This book draws on material Michael created for his renowned Object Mentor seminars: techniques Michael has used in mentoring to help hundreds of developers, technical managers, and testers bring their legacy systems under control. The topics covered include Understanding the mechanics of software change: adding features, fixing bugs, improving design, optimizing performance Getting legacy code into a test harness Writing tests that protect you against introducing new problems Techniques that can be used with any language or platform—with examples in Java, C++, C, and C# Accurately identifying where code changes need to be made Coping with legacy systems that aren't object-oriented Handling applications that don't seem to have any structure This book also includes a catalog of twenty-four dependency-breaking techniques that help you work with program elements in isolation and make safer changes.

Write Great Code, Volume 2, 2nd Edition John Wiley & Sons

A comprehensive guide to help aspiring and professional C++ developers elevate the performance of their apps by allowing them to run faster and consume fewer resources Key

FeaturesUpdated to C++20 with completely revised code and more content on error handling, benchmarking, memory allocators, and concurrent programmingExplore the latest C++20 features including concepts, ranges, and coroutinesUtilize C++ constructs and techniques to carry out effective data structure optimization and memory managementBook Description C++ High Performance, Second Edition guides you through optimizing the performance of your C++ apps. This allows them to run faster and consume fewer resources on the device they're running on without compromising the readability of your codebase. The book begins by introducing the C++ language and some of its modern concepts in brief. Once you are familiar with the fundamentals, you will be ready to measure, identify, and eradicate bottlenecks in your C++ codebase. By following this process, you will gradually improve your style of writing code. The book then explores data structure optimization, memory management, and how it can be used efficiently concerning CPU caches. After laying the foundation, the book trains you to leverage algorithms, ranges, and containers from the standard library to achieve faster execution, write readable code, and use customized iterators. It provides hands-on examples of C++ metaprogramming, coroutines, reflection to reduce boilerplate code, proxy objects to perform optimizations under the hood, concurrent programming, and lock-free data structures. The book concludes with an overview of parallel algorithms. By the end of this book, you will have

the ability to use every tool as needed to boost the efficiency of your C++ projects. What you will learnWrite specialized data structures for performance-critical codeUse modern metaprogramming techniques to reduce runtime calculationsAchieve efficient memory management using custom memory allocatorsReduce boilerplate code using reflection techniquesReap the benefits of lock-free concurrent programmingGain insights into subtle optimizations used by standard library algorithmsCompose algorithms using ranges libraryDevelop the ability to apply metaprogramming aspects such as constexpr, constraints, and conceptsImplement lazy generators and asynchronous tasks using C++20 coroutinesWho this book is for If you're a C++ developer looking to improve the efficiency of your code or just keen to upgrade your skills to the next level, this book is for you.

Introduction to Software for Chemical Engineers IGI Global

High Performance MySQL is the definitive guide to building fast, reliable systems with MySQL. Written by noted experts with years of real-world experience building very large systems, this book covers every aspect of MySQL performance in detail, and focuses on robustness, security, and data integrity. High Performance MySQL teaches you advanced techniques in depth so you can bring out MySQL's full power. Learn how to design schemas, indexes, queries and advanced MySQL features for maximum performance, and get detailed guidance for tuning your MySQL server, operating system, and hardware to their fullest potential. You'll also learn practical, safe, high-performance ways to scale your applications with replication, load balancing, high availability, and failover. This second edition is completely revised and

greatly expanded, with deeper coverage in all areas. Major additions include: Emphasis throughout on both performance and reliability Thorough coverage of storage engines, including in-depth tuning and optimizations for the InnoDB storage engine Effects of new features in MySQL 5.0 and 5.1, including stored procedures, partitioned databases, triggers, and views A detailed discussion on how to build very large, highly scalable systems with MySQL New options for backups and replication Optimization of advanced querying features, such as full-text searches Four new appendices The book also includes chapters on benchmarking, profiling, backups, security, and tools and techniques to help you measure, monitor, and manage your MySQL installations.

Parallel Programming with Intel Parallel Studio XE Springer Science & Business Media

Full treatment, from model formulation to computational implementation, of optimization techniques that solve central problems in finance.

Pyomo – Optimization Modeling in Python Packt Publishing Ltd

The hands-on guide to high-performance coding and algorithm optimization. This hands-on guide to software optimization introduces state-of-the-art solutions for every key aspect of software performance - both code-based and algorithm-based. Two leading HP software performance experts offer comparative optimization strategies for RISC and for the new Explicitly Parallel Instruction Computing (EPIC) design used in Intel IA-64 processors. Using many practical examples, they offer specific techniques for: Predicting and measuring performance - and identifying your best optimization opportunities Storage optimization: cache, system memory, virtual memory, and I/O Parallel processing: distributed-memory and shared-memory (SMP and ccNUMA) Compilers and loop optimization Enhancing parallelism: compiler directives, threads, and message passing Mathematical libraries and algorithms Whether you're a

developer, ISV, or technical researcher, if you need to optimize high-performance software on today's leading processors, one book delivers the advanced techniques and code examples you need: *Software Optimization for High Performance Computing*.

Performance Optimization of Numerically Intensive Codes John Wiley & Sons

Annotation Four Intel experts explain the techniques and tools that you can use to improve the performance of applications for IA-32 processors. Simple explanations and code examples help you to develop software that benefits from Intel? Extended Memory 64 Technology (Intel? EM64T), multi-core processing, Hyper-Threading Technology, OpenMP*, and multimedia extensions. This book guides you through the growing collection of software tools, compiler switches, and coding optimizations, showing you efficient ways to get the best performance from software applications.

Changhyun Kwon

Performance Optimization of Numerically Intensive Codes offers a comprehensive, tutorial-style, hands-on, introductory and intermediate-level treatment of all the essential ingredients for achieving high performance in numerical computations on modern computers. The authors explain computer architectures, data traffic and issues related to performance of serial and parallel code optimization exemplified by actual programs written for algorithms of wide interest. The unique hands-on style is achieved by extensive case studies using realistic computational problems. The performance gain obtained by applying the techniques described in this book can be very significant. The book

bridges the gap between the literature in system architecture, the one in numerical methods and the occasional descriptions of optimization topics in computer vendors' literature. It also allows readers to better judge the suitability of certain computer architecture to their computational requirements. In contrast to standard textbooks on computer architecture and on programming techniques the book treats these topics together at the level necessary for writing high-performance programs. The book facilitates easy access to these topics for computational scientists and engineers mainly interested in practical issues related to efficient code development.

C++ High Performance CRC Press

Widely considered one of the best practical guides to programming, Steve McConnell's original **CODE COMPLETE** has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that

help you: Design for minimum complexity and maximum creativity
Reap the benefits of collaborative development
Apply defensive programming techniques to reduce and flush out errors
Exploit opportunities to refactor—or evolve—code, and do it safely
Use construction practices that are right-weight for your project
Debug problems quickly and effectively
Resolve critical construction issues early and correctly
Build quality into the beginning, middle, and end of your project