Software Optimization Cookbook Second Edition

The Software Optimization Cookbook
The Compiler Design Handbook
Business Optimization Using Mathematical Programming
Introduction to Software for Chemical Engineers
High Performance MySQL
Power Distribution Planning Reference Book, Second Edition
Scientific Computing
IPython Interactive Computing and Visualization Cookbook
Introduction to High Performance Computing for Scientists and Engineers
Optimization Software Guide
Introduction to Software for Chemical Engineers, Second Edition
Introduction to Software for Chemical Engineers, Second Edition
Handbook of Research on Computational Science and Engineering: Theory and Practice
Engineering a Compiler
Code Complete

Optimization Modeling with Spreadsheets
Power and Performance
Performance Optimization of Numerically Intensive Codes
Optimization Methods in Finance
Pyomo — Optimization Modeling in Python
Pyomo – Optimization Modeling in Python
Optimizing Compilers for Modern Architectures: A Dependence-Based Approach
Julia Programming for Operations Research
Programming Multicore and Many-core Computing Systems
Embedded System Design
Write Great Code, Volume 2, 2nd Edition
Parallel Programming with Intel Parallel Studio XE
Software Optimization for High-performance Computing
The Software Optimization Cookbook
Embedded Computing for High Performance
Search Engine Optimization
Convex Optimization
NumPy Cookbook - Second Edition
Proceedings of the ... ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming
The Software Optimization Cookbook
High Performance Computing
C++ High Performance
Working Effectively with Legacy Code
Docker Cookbook - Second Edition

ALINA

The Software Optimization Cookbook "O'Reilly Media, Inc." 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 maintainabilit y but also maintains the central ideas in modern AMLs.

The Compiler Design Handbook

Pearson Education

Providing more than twice the content of the original edition, this new edition is the premier source on the selection, development, and provision of safe, high-quality, and cost-effective electric utility distribution systems, and it promises vast improvements in system

reliability and layout by spanning every aspect of system planning including load forecasting, scheduling, performance, and economics. Responding to the evolving needs of electric utilities, Power Distribution Planning Reference Book presents an abundance of real-world examples, procedural and managerial issues, and engineering and analytical methodologies that are crucial to efficient and enhanced system performance. *Business Optimization Using Mathematical Programming* CRC Press 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.

Introduction to Software for Chemical Engineers

Springer Science & Business Media

Reflects the latest applied research and features state-of-the-art software for building and solving spreadsheet optimization models Thoroughly updated to reflect the latest topical and technical advances in the field, Optimization Modeling with Spreadsheets, Second Edition continues to focus on solving real-

world optimization problems through the creation of mathematical models and the use of spreadsheets to represent and analyze those models. Developed and extensively classroom-tested by the author, the book features a systematic approach that equips readers with the skills to apply optimization tools effectively without the need to rely on specialized algorithms. This new edition uses the powerful software package Risk Solver Platform (RSP) for optimization, including its Evolutionary Solver, which employs many recently developed ideas for heuristic programming. The author provides expanded coverage of integer programming and discusses linear and nonlinear programming using a systematic approach that emphasizes the use of spreadsheet-based optimization tools. The Second Edition also features: Classifications for the various problem types, providing the reader with a broad framework for building and recognizing optimization models Network models that allow for a more general form of mass balance A systematic introduction to Data Envelopment Analysis (DEA) The

identification of qualitative patterns in order to meaningfully interpret linear programming solutions An introduction to stochastic programming and the use of RSP to solve problems of this type Additional examples, exercises, and cases have been included throughout, allowing readers to test their comprehension of the material. In addition, a related website features Microsoft Office® Excel files to accompany the figures and data sets in the book. With its accessible and comprehensive presentation, Optimization Modeling with Spreadsheets, Second Edition is an excellent book for courses on deterministic models, optimization, and spreadsheet modeling at the upper-undergraduate and graduate levels. The book can also serve as a reference for researchers, practitioners, and consultants working in business, engineering, operations research, and management science.

High Performance MySQL

Springer Science & Business Media

Leverage Docker to deploying software at scale Key Features Leverage practical examples to manage containers efficiently Integrate with

orchestration tools such as Kubernetes for controlled deployments Learn to implement best practices on improving efficiency and security of containers Book Description Docker is an open source platform for building, shipping, managing, and securing containers. Docker has become the tool of choice for people willing to work with containers. Since the market is moving toward containerizatio n, Docker will definitely have a big role to play in the future tech market. This book starts with setting up Docker in different environment, and helps you learn how to work with Docker images. Then, you will take a deep dive into network and data management for containers. The book explores the RESTful APIs provided by Docker to perform different actions, such as image/contain er operations. The book then explores logs and troubleshootin g Docker to solve issues and bottlenecks. You will gain an understanding of Docker use cases, orchestration, security, ecosystems, and hosting platforms to make your applications easy to deploy, build, and collaborate on. The book covers the new features of Docker 18.xx (or

later), such as working with AWS and Azure, Docker Engine, Docker Swarm, Docker Compose, and so on. By the end of this book, you will have gained hands-on experience of finding quick solutions to different problems encountered while working with Docker. What you will learn Install Docker on various platforms Work with Docker images and containers Container networking and data sharing Docker APIs and language bindings Various PaaS solutions for Docker Implement container orchestration using Docker Swarm and Kubernetes Container security Docker on various clouds Who this book is for Book is targeted towards developers, system administrators , and DevOps engineers who want to use Docker in his/her development, QA, or production environments. It is expected that the reader has basic Linux/Unix skills such as installing packages, editing files, managing services, and so on. Any experience in virtualization technologies such as KVM, XEN, and VMware will be an added advantage *Power Distribution Planning Reference Book, Second Edition* SIAM 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

Scientific Computing

Elsevier

This book constitutes the thoroughly refereed post-conference proceedings of the 9th International Conference on High Performance Computing for Computational Science, VECPAR 2010, held in Berkeley, CA, USA, in June 2010. The 34 revised full papers presented together with five invited contributions were carefully selected during two rounds of reviewing and revision. The papers are organized in topical sections on linear algebra and solvers on emerging architectures, large-scale simulations, parallel and distributed computing, numerical algorithms.

IPython Interactive Computing and Visualization Cookbook

The Software Optimization CookbookThe Software Optimization CookbookAnnotation 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.In troduction to High Performance Computing for Scientists and Engineers 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

Introduction to High Performance Computing for Scientists and Engineers

CRC Press

This book differs from traditional numerical analysis texts in that it focuses on the motivation and ideas behind the algorithms presented rather than on detailed analyses of them. It presents a

broad overview of methods and software for solving mathematical problems arising in computational modeling and data analysis, including proper problem formulation, selection of effective solution algorithms, and interpretation of results.? In the 20 years since its original publication, the modern, fundamental perspective of this book has aged well, and it continues to be used in the classroom. This Classics edition has been updated to include pointers to Python software and the Chebfun package, expansions on barycentric formulation for Lagrange polynomial interpretation and stochastic methods, and the availability of about 100 interactive educational modules that dynamically illustrate the concepts and algorithms in the book. Scientific Computing: An Introductory Survey, Second Edition is intended as both a textbook and a reference for computationally oriented disciplines that need to solve mathematical problems. Optimization Software Guide Packt Publishing Ltd Power and Performance: Software Analysis and Optimization is a guide to solving performance problems in modern Linux systems. Power-efficient chips

are no help if the software those chips run on is inefficient. Starting with the necessary architectural background as a foundation, the book demonstrates the proper usage of performance analysis tools in order to pinpoint the cause of performance problems, and includes best practices for handling common performance issues those tools identify. Provides expert perspective from a key member of Intel's optimization team on how processors and memory systems influence performance Presents ideas to improve architectures running mobile, desktop, or enterprise platforms Demonstrates best practices for designing experiments and benchmarking throughout the software lifecycle Explains the importance of profiling and measurement to determine the source of performance issues

Introduction to Software for Chemical Engineers, Second Edition

Prentice Hall Professional

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.

Introduction to Software for Chemical Engineers, Second Edition

Springer

Until the late 1980s, information processing was associated with large mainframe computers and huge tape drives. During the 1990s, this trend shifted toward information processing with personal computers, or PCs. The trend toward miniaturizatio n continues and in the future the majority of information processing systems will be small mobile computers, many of which will be embedded into larger products and interfaced to the physical environment. Hence, these kinds of systems are called embedded systems. Embedded systems together with their physical environment are called cyber-physical systems. Examples include systems such as transportation and fabrication equipment. It is expected that the total market volume of embedded systems will be significantly larger than that of traditional information processing systems such as PCs and mainframes. Embedded systems share a number of common characteristics . For example, they must be dependable, efficient, meet real-time constraints

and require customized user interfaces (instead of generic keyboard and mouse interfaces). Therefore, it makes sense to consider common principles of embedded system design. Embedded System Design starts with an introduction into the area and a survey of specification models and languages for embedded and cyber-physical systems. It provides a brief overview of hardware devices used for such systems and presents the essentials of system software for embedded systems, like real-time operating systems. The book also discusses evaluation and validation techniques for embedded systems. Furthermore, the book presents an overview of techniques for mapping applications to execution platforms. Due to the importance of resource efficiency, the book also contains a selected set of optimization techniques for embedded systems, including special compilation techniques. The book closes with a brief survey on testing. Embedded System Design can be used as a text book for courses on embedded systems and as a source which provides pointers to relevant material in the

area for PhD students and teachers. It assumes a basic knowledge of information processing hardware and software. Courseware related to this book is available at http://ls12-www.cs.tu-dortmund.de/~marwedel.

Handbook of Research on Computational Science and Engineering: Theory and Practice 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

Engineering a Compiler

No Starch Press

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 programming Gain 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.

Code Complete

CRC Press

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 heterogeneou s 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.

Optimization Modeling with Spreadsheets SIAM 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.

Power and Performance Morgan Kaufmann Revealing the secrets of the software tuning process, The Software Optimization Cookbook provides recipes for high-performance applications on the Intel? Pentium? III and Pentium? 4 processors. Simple explanations and C language examples show you how to address performance issues with algorithms, memory access, branching, SIMD instructions, multiple threads, and floating-point calculations.

With this book, you need not be a processor architect or assembly language expert to get the full power out of your software on the 32-bit Intel Architecture. Learn how to: Use performance tools and tested concepts to analyze and improve applications. Determine which portions of an application should be given highest priority for optimizations. Identify the reasons that certain portions of your application are slower than they should be. Improve an application by working directly on the root cause of a software bottleneck. Design an application from the ground up for maximum performance.

Performance Optimization of Numerically Intensive Codes

Changhyun Kwon

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

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.

Pyomo — Optimization Modeling in Python CRC Press

Your visual, step-by-step guide to search engine optimization, from an Internet marketing expert Techniques and best practices for search engine optimization are constantly evolving. This visual guide to SEO is fully updated with information on the latest and most effective ways to move your website up in the search engine rankings. Internet marketing guru Kristopher Jones, a frequent keynote speaker at interactive marketing conferences, explains all the key concepts in a visual format so you can see how they work and what to do. Learn about keyword generation, internal linking, URL structure, content creation, using social media, and more. More than 70 percent of businesses today have websites; search engine optimization is a vital factor in growing a business by gaining new customers while increasing business from existing customers This two-color book is the

only guide to search engine optimization that is presented in a visual format Presents search engine marketing principles including keyword generation, on-site optimization involving website structure, internal linking, URL structure, content creation, off-site optimization, social media optimization and more Author is a popular keynote speaker and panelist at interactive marketing and technology conferences Search Engine Optimization: Your visual blueprint for effective Internet marketing, Third Edition helps visual learners master and maximize SEO techniques.