
Writing High Performance Net Code

Right here, we have countless book **Writing High Performance Net Code** and collections to check out. We additionally come up with the money for variant types and next type of the books to browse. The tolerable book, fiction, history, novel, scientific research, as well as various additional sorts of books are readily approachable here.

As this Writing High Performance Net Code, it ends in the works instinctive one of the favored book Writing High Performance Net Code collections that we have. This is why you remain in the best website to see the incredible books to have.

Writing High Performance Net Code

2023-10-10

WERNER YARELI

For Better Code, Performance, and Scalability Apress

The free book "Fundamentals of Computer Programming with C#" is a comprehensive computer programming tutorial that teaches programming, logical thinking, data structures and algorithms, problem solving and high quality code with lots of examples in C#. It starts with the first steps in programming and software development like variables, data types, conditional statements, loops and arrays and continues with other basic topics like methods, numeral systems, strings and string processing, exceptions, classes and objects. After the basics this fundamental programming book enters into more advanced programming topics like recursion, data structures (lists, trees, hash-tables and graphs), high-quality code, unit testing and refactoring, object-oriented principles (inheritance, abstraction, encapsulation and polymorphism) and their implementation the C# language. It also covers fundamental topics that each good

developer should know like algorithm design, complexity of algorithms and problem solving. The book uses C# language and Visual Studio to illustrate the programming concepts and explains some C# / .NET specific technologies like lambda expressions, extension methods and LINQ. The book is written by a team of developers lead by Svetlin Nakov who has 20+ years practical software development experience. It teaches the major programming concepts and way of thinking needed to become a good software engineer and the C# language in the meantime. It is a great start for anyone who wants to become a skillful software engineer. The books does not teach technologies like databases, mobile and web development, but shows the true way to master the basics of programming regardless of the languages, technologies and tools. It is good for beginners and intermediate developers who want to put a solid base for a successful career in the software engineering industry. The book is accompanied by free video lessons, presentation slides and mind maps, as well as hundreds of exercises and live examples. Download the free C# programming book, videos, presentations

and other resources from <http://introprogramming.info>. Title: Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) ISBN: 9789544007737 ISBN-13: 978-954-400-773-7 (9789544007737) ISBN-10: 954-400-773-3 (9544007733) Author: Svetlin Nakov & Co. Pages: 1132 Language: English Published: Sofia, 2013 Publisher: Faber Publishing, Bulgaria Web site: <http://www.introprogramming.info> License: CC-Attribution-Share-Alike Tags: free, programming, book, computer programming, programming fundamentals, ebook, book programming, C#, CSharp, C# book, tutorial, C# tutorial; programming concepts, programming fundamentals, compiler, Visual Studio, .NET, .NET Framework, data types, variables, expressions, statements, console, conditional statements, control-flow logic, loops, arrays, numeral systems, methods, strings, text processing, StringBuilder, exceptions, exception handling, stack trace, streams, files, text files, linear data structures, list, linked list, stack, queue, tree, balanced tree, graph, depth-first search, DFS, breadth-first search, BFS, dictionaries, hash tables, associative arrays, sets, algorithms, sorting algorithm, searching algorithms, recursion, combinatorial algorithms, algorithm complexity, OOP, object-oriented programming, classes, objects, constructors, fields, properties, static members, abstraction, interfaces, encapsulation, inheritance, virtual methods, polymorphism, cohesion, coupling, enumerations, generics, namespaces, UML, design patterns, extension methods, anonymous types, lambda expressions, LINQ, code quality, high-quality code, high-quality classes, high-quality methods, code formatting, self-documenting code, code refactoring, problem solving, problem solving methodology,

9789544007737, 9544007733

[A Comprehensive Guide for Writing Simple Code to Solve Complex Problems](#) "O'Reilly Media, Inc."

Take performance to the next level! This book does not just teach you how the CLR works---it teaches you exactly what you need to do now to obtain the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. This second edition incorporates the advances and improvements in .NET over the last few years, as well as greatly expanded coverage of tools, more topics, more tutorials, more tips, and improvements throughout the entire book. New in the 2nd Edition: 50% increase in content! New examples, code samples, and diagrams throughout entire book More ways to analyze the heap and find memory problems More tool coverage, including expanded usage of Visual Studio More benchmarking New GC configuration options Code warmup techniques New .NET features such as ref-returns, value tuples, SIMD, and more More detailed analysis of LINQ Tips for high-level feature areas such as ASP.NET, ADO.NET, and WPF Also find expanded coverage and discover new tips and tricks for: Profiling with multiple tools to quickly find problem areas Detailed description of the garbage collector, how to optimize your code for it, and how to diagnose difficult memory-related issues How to analyze JIT and diagnose warmup problems Effective use of the Task Parallel Library to maximize throughput Which .NET features and APIs to use and which to avoid Instrument your program with performance

counters and ETW events Use the latest and greatest .NET features Build a performance-minded team ...and so much more *Optimize Your C# Applications* CreateSpace

Build the skills to apply Microsoft .NET collections effectively Put .NET collections to work—and manage issues with GUI data binding, threading, data querying, and storage. Led by a data collection expert, you'll gain task-oriented guidance, exercises, and extensive code samples to tackle common problems and improve application performance. This one-stop reference is designed for experienced Microsoft Visual Basic and C# developers—whether you're already using collections or just starting out. Discover how to: Implement arrays, associative arrays, stacks, linked lists, and other collection types Apply built in .NET collection classes by learning their methods and properties Add enumerator, dictionary, and other .NET collection interfaces to your classes Query collections by writing simple to complex Microsoft LINQ statements Synchronize data across threads using built in .NET synchronization classes Enhance your custom collection classes with serialization support Use simple data binding to display collections in Windows Forms, Microsoft Silverlight, and Windows Presentation Foundation

[Build faster web apps using Node.js, Svelte.js, and WebAssembly](#) "O'Reilly Media, Inc."

Your Python code may run correctly, but you need it to run faster. Updated for Python 3, this expanded edition shows you how to locate performance bottlenecks and significantly speed up your code in high-data-volume programs. By exploring the fundamental theory behind design choices, *High Performance Python* helps you gain a deeper understanding of Python's

implementation. How do you take advantage of multicore architectures or clusters? Or build a system that scales up and down without losing reliability? Experienced Python programmers will learn concrete solutions to many issues, along with war stories from companies that use high-performance Python for social media analytics, productionized machine learning, and more. Get a better grasp of NumPy, Cython, and profilers Learn how Python abstracts the underlying computer architecture Use profiling to find bottlenecks in CPU time and memory usage Write efficient programs by choosing appropriate data structures Speed up matrix and vector computations Use tools to compile Python down to machine code Manage multiple I/O and computational operations concurrently Convert multiprocessing code to run on local or remote clusters Deploy code faster using tools like Docker

[Developer's Guide to Collections in Microsoft .NET](#) No Starch Press

The Go Workshop takes you from being a novice Go programmer to a confident developer who can leverage the key features of the language to build real-world applications. This book helps you cut through excessive theory and delve into the practical features and techniques that are commonly applied to design performant, scalable applications.

High Performance Web Sites "O'Reilly Media, Inc."

Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This

book does not just teach you how the CLR works-it teaches you exactly what you need to do now to obtain the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to: Choose what to measure and why Use many amazing tools, freely available, to solve problems quickly Understand the .NET garbage collector and its effect on your application Use effective coding patterns that lead to optimal garbage collection performance Diagnose common GC-related issues Reduce costs of JITting Use multiple threads sanely and effectively, avoiding synchronization problems Know which .NET features and APIs to use and which to avoid Use code generation to avoid performance problems Measure everything and expose hidden performance issues Instrument your program with performance counters and ETW events Use the latest and greatest .NET features Ensure your code can run on mobile devices without problems Build a performance-minded team ...and much more.

[The Bulgarian C# Book](#) Cambridge University Press

Advanced .NET IL Assembler is a comprehensive drill-down into the inner workings of the .NET Framework. Acknowledged runtime expert and Microsoft insider Serge Lidin steps through the internal structures and operations that take place when .NET code is executed, showing how the syntax and grammar of the coding language is broken down into low-level units that can be expressed through the ILAsm language that runs behind the

scenes in .NET. By reading this book you will develop the skills you need to write tighter, faster, .NET code; to debug complex error handling situations; and to oversee multi-language and multi-platform projects with confidence.

Hands-On High Performance with Go "O'Reilly Media, Inc."

Use this in-depth guide to correctly design benchmarks, measure key performance metrics of .NET applications, and analyze results. This book presents dozens of case studies to help you understand complicated benchmarking topics. You will avoid common pitfalls, control the accuracy of your measurements, and improve performance of your software. Author Andrey Akinshin has maintained BenchmarkDotNet (the most popular .NET library for benchmarking) for five years and covers common mistakes that developers usually make in their benchmarks. This book includes not only .NET-specific content but also essential knowledge about performance measurements which can be applied to any language or platform (common benchmarking methodology, statistics, and low-level features of modern hardware). What You'll Learn Be aware of the best practices for writing benchmarks and performance tests Avoid the common benchmarking pitfalls Know the hardware and software factors that affect application performance Analyze performance measurements Who This Book Is For .NET developers concerned with the performance of their applications

[High Performance JavaScript](#) Packt Publishing Ltd

Writing High-Performance .Net Code

F# High Performance Packt Publishing Ltd

Covers topics such as integrating multiple .NET technologies, cross-language integration, versioning, database and monitoring

tools for application development, accessing data, and COM+.

Delphi High Performance Packt Publishing

Just as Strunk and White's "The Elements of Style" provides rules of usage for writing in the English language, this text furnishes a set of rules for writing in C#.

.NET IL Assembler Packt Publishing Ltd

Getting started with Django is easy. There are tutorials and books that literally walk you through the process of getting your first site up and running. Taking that code from your laptop to the real world is like opening Pandora's box. Should I use Apache, Unicorn, uWSGI or something else? Where should I use caching to make things faster? How do I know if my database has the right indexes or if it needs more resources? Do I need a NoSQL database like MongoDB? The site runs great on my laptop. Why is it so slow in production? How many servers does my site need? How big should they be? What is the 20% effort that will solve 80% of my performance problems? If you've asked yourself any of these questions, you're like most Django developers. Heck, we were asking some of the same questions when we started working with Django 7 years ago at Lincoln Loop. Since then we've built, deployed, and maintained a lot of Django sites. Everything from realtime applications to large-scale CMSes with tons of traffic. Quite frankly, we made a lot of mistakes, but we learned a lot too. High Performance Django is the book we wish we had when we got started. It will give you a repeatable blueprint for building and deploying fast, scalable Django sites. More information and ebook formats available at <https://highperformancedjango.com>.

Concurrency in C# Cookbook Packt Publishing Ltd

Why does the word "legacy" with synonyms like heritage and birthright now describe difficult software? What anchors our code making it rigid and unyielding? How do we identify those anchors? How do we write code that is less painful and more resilient? Leonard is a software architect and .NET specialist who has spent his career asking and answering these questions. He has developed a list of maxims that serve as reminders on how to build systems that are easier to maintain, adapt, and grow. When encountering difficult code, it is easy to want to tear it all down and start fresh. If we choose to do that, how do we ensure our successors will not want to do the same? What if we didn't have to tear it all down? What if we could identify the pain points in the current system and abstract them? This book is full of examples. For example, the open/closed principle, the second of five well-known SOLID principles, says our code should be open for extension and closed for modification, but what does it look like when our code is closed for extension or open for modification? Each chapter of this book will focus on one of Leonard's code maxims which will highlight either some aspect of code design or the software development lifecycle. Through this book, you will learn how to identify those things anchoring your code to the past. You will learn concepts that make testing and maintainability easy. Your code will be more resilient. When confronted with difficult code or changing business requirements, you will become more resilient.

Mastering JavaScript High Performance Packt Publishing Ltd

Learn how to use Python to create efficient applications About This Book Identify the bottlenecks in your applications and solve them using the best profiling techniques Write efficient numerical

code in NumPy, Cython, and Pandas Adapt your programs to run on multiple processors and machines with parallel programming Who This Book Is For The book is aimed at Python developers who want to improve the performance of their application. Basic knowledge of Python is expected What You Will Learn Write efficient numerical code with the NumPy and Pandas libraries Use Cython and Numba to achieve native performance Find bottlenecks in your Python code using profilers Write asynchronous code using Asyncio and RxPy Use Tensorflow and Theano for automatic parallelism in Python Set up and run distributed algorithms on a cluster using Dask and PySpark In Detail Python is a versatile language that has found applications in many industries. The clean syntax, rich standard library, and vast selection of third-party libraries make Python a wildly popular language. Python High Performance is a practical guide that shows how to leverage the power of both native and third-party Python libraries to build robust applications. The book explains how to use various profilers to find performance bottlenecks and apply the correct algorithm to fix them. The reader will learn how to effectively use NumPy and Cython to speed up numerical code. The book explains concepts of concurrent programming and how to implement robust and responsive applications using Reactive programming. Readers will learn how to write code for parallel architectures using Tensorflow and Theano, and use a cluster of computers for large-scale computations using technologies such as Dask and PySpark. By the end of the book, readers will have learned to achieve performance and scale from their Python applications. Style and approach A step-by-step practical guide filled with real-

world use cases and examples

Resilient Code Packt Publishing Ltd

An example-driven guide covering modern web app development techniques and emerging technologies such as WebAssembly, Service Workers, and Svelte.js to build faster, secure, and scalable apps Key FeaturesDiscover effective techniques for accessing DOM, minimizing painting, and using a V8 engine to optimize JavaScriptUnderstand what makes the web tick and create apps that look and feel like native desktop applicationsExplore modern JavaScript frameworks like Svelte.js for building next-gen web appsBook Description High-performance web development is all about cutting through the complexities in different layers of a web app and building services and APIs that improve the speed and performance of your apps on the browser. With emerging web technologies, building scalable websites and sustainable web apps is smoother than ever. This book starts by taking you through the web frontend, popular web development practices, and the latest version of ES and JavaScript. You'll work with Node.js and learn how to build web apps without a framework. The book consists of three hands-on examples that help you understand JavaScript applications at both the server-side and the client-side using Node.js and Svelte.js. Each chapter covers modern techniques such as DOM manipulation and V8 engine optimization to strengthen your understanding of the web. Finally, you'll delve into advanced topics such as CI/CD and how you can harness their capabilities to speed up your web development dramatically. By the end of this web development book, you'll have understood how the JavaScript landscape has evolved, not

just for the frontend but also for the backend, and be ready to use new tools and techniques to solve common web problems. What you will learn Explore Vanilla JavaScript for optimizing the DOM, classes, and modules, and querying with jQuery Understand immutable and mutable code and develop faster web apps Delve into Svelte.js and use it to build a complete real-time Todo app Build apps to work offline by caching calls using service workers Write C++ native code and call the WebAssembly module with JavaScript to run it on a browser Implement CircleCI for continuous integration in deploying your web apps Who this book is for This JavaScript book is for web developers, C/C++ programmers, and anyone who wants to build robust web applications using advanced web technologies. This book assumes a good grasp of Vanilla JavaScript and an understanding of web development tools, such as Chrome Developer tools or Mozilla's developer tools.

Programming F# 3.0 Apress

If you're like most developers, you rely heavily on JavaScript to build interactive and quick-responding web applications. The problem is that all of those lines of JavaScript code can slow down your apps. This book reveals techniques and strategies to help you eliminate performance bottlenecks during development. You'll learn how to improve execution time, downloading, interaction with the DOM, page life cycle, and more. Yahoo! frontend engineer Nicholas C. Zakas and five other JavaScript experts—Ross Harmes, Julien Lecomte, Steven Levithan, Stoyan Stefanov, and Matt Sweeney—demonstrate optimal ways to load code onto a page, and offer programming tips to help your JavaScript run as efficiently and quickly as possible. You'll learn

the best practices to build and deploy your files to a production environment, and tools that can help you find problems once your site goes live. Identify problem code and use faster alternatives to accomplish the same task Improve scripts by learning how JavaScript stores and accesses data Implement JavaScript code so that it doesn't slow down interaction with the DOM Use optimization techniques to improve runtime performance Learn ways to ensure the UI is responsive at all times Achieve faster client-server communication Use a build system to minify files, and HTTP compression to deliver them to the browser

Hands-On JavaScript High Performance Faber Publishing

This book will help you understand what "programming for performance" means, and use effective coding patterns and techniques to optimize your .NET applications. You will begin by understanding what "high performance coding" means, and the different performance concerns. You will see how CLR works and get an understanding of concepts such as memory management, garbage collection, and thread life cycles. You will proceed to learn about the theoretical and practical concepts of PLINQ programming. You will also see what Big Data is, and how to architect a Big Data solution to manipulate large datasets. Finally, you will learn how to launch and analyze a profile session and execute tests against a code block or application for performance analysis. By the end of this book, you will have a complete understanding of efficient programming using high-performance techniques, and will be able to write highly optimized applications.

Boost and optimize the performance of your Golang

applications at scale with resilience Apress

Build powerful and fast applications with F# About This Book Explore the advanced concurrency support in F# and .NET TPL Covers major optimization techniques in F# to improve the performance of applications Use Struct, Class and Record model, Interop with C# and VB without sacrificing performance. Who This Book Is For This book is for F# developers who want to build high-performance applications. Knowledge of functional programming would be helpful. What You Will Learn Understand how the execution of functions in F# works Identify common performance bottlenecks Implement best practices to optimize performance Use the available tooling to help measure performance Combine the best practice of asynchronous and synchronous Optimize further using various F# language constructs In Detail F# is a functional programming language and is used in enterprise applications that demand high performance. It has its own unique trait: it is a functional programming language and has OOP support at the same time. This book will help you make F# applications run faster with examples you can easily break down and take into your own work. You will be able to assess the performance of the program and identify bottlenecks. Beginning with a gentle overview of concurrency features in F#, you will get to know the advanced topics of concurrency optimizations in F#, such as F# message passing agent of MailboxProcessor and further interoperation with .NET TPL. Based on this knowledge, you will be able to enhance the performance optimizations when implementing and using other F# language features. The book also covers optimization techniques by using F# best practices and F# libraries. You will

learn how the concepts of concurrency and parallel programming will help in improving the performance. With this, you would be able to take advantage of multi-core processors and track memory leaks, root causes, and CPU issues. Finally, you will be able to test their applications to achieve scalability. Style and approach This easy-to-follow guide is full of hands-on examples of real-world multithreading tasks. Each topic is explained and placed in context, and for the more inquisitive, there are also more in-depth details of the concepts used.

Clean Code in C# Apress

Do you want your .NET code to have the absolute best performance it can? This book demystifies the CLR, teaching you how and why to write code with optimum performance. Learn critical lessons from a person who helped design and build one of the largest high-performance .NET systems in the world. This book does not just teach you how the CLR works--it teaches you exactly what you need to do now to obtain the best performance today. It will expertly guide you through the nuts and bolts of extreme performance optimization in .NET, complete with in-depth examinations of CLR functionality, free tool recommendations and tutorials, useful anecdotes, and step-by-step guides to measure and improve performance. Among the topics you will learn are how to:- Choose what to measure and why- Use many amazing tools, freely available, to solve problems quickly- Understand the .NET garbage collector and its effect on your application- Use effective coding patterns that lead to optimal garbage collection performance- Diagnose common GC-related issues- Reduce costs of JITting- Use multiple threads sanely and effectively, avoiding synchronization problems- Know

which .NET features and APIs to use and which to avoid- Use code generation to avoid performance problems- Measure everything and expose hidden performance issues- Instrument your program with performance counters and ETW events- Use the latest and greatest .NET features- Ensure your code can run on mobile devices without problems- Build a performance-minded team...and much more.

Network Programming for the Microsoft .NET Framework Pearson Education

Apache Spark is amazing when everything clicks. But if you haven't seen the performance improvements you expected, or still don't feel confident enough to use Spark in production, this practical book is for you. Authors Holden Karau and Rachel Warren demonstrate performance optimizations to help your Spark queries run faster and handle larger data sizes, while using

fewer resources. Ideal for software engineers, data engineers, developers, and system administrators working with large-scale data applications, this book describes techniques that can reduce data infrastructure costs and developer hours. Not only will you gain a more comprehensive understanding of Spark, you'll also learn how to make it sing. With this book, you'll explore: How Spark SQL's new interfaces improve performance over SQL's RDD data structure The choice between data joins in Core Spark and Spark SQL Techniques for getting the most out of standard RDD transformations How to work around performance issues in Spark's key/value pair paradigm Writing high-performance Spark code without Scala or the JVM How to test for functionality and performance when applying suggested improvements Using Spark MLlib and Spark ML machine learning libraries Spark's Streaming components and external community packages