

---

# Writing High Performance Net Code

---

C# 7 and .NET Core 2.0 High Performance  
 Clean Code in C#  
 C++ High Performance  
 High Performance Web Sites  
 Pro .NET Memory Management  
 Clojure High Performance Programming  
 Programming C#  
 Improving .NET Application Performance and Scalability  
 Network Programming for the Microsoft .NET Framework  
 Write Great Code, Volume 1  
 Metaprogramming in .NET  
 C# and .NET Core Test-Driven Development  
 Hands-On High Performance with Go  
 Functional Programming in C#, Second Edition  
 Hands-On JavaScript High Performance  
 Expert C# 5.0  
 Delphi High Performance  
 High-level Synthesis  
 Michael Abrash's Graphics Programming Black Book  
 High Performance JavaScript  
 Learning .NET High-performance Programming  
 Performance Tuning and Optimizing ASP.NET Applications  
 Network Programming with Go  
 Pro .NET Performance  
 Writing High-Performance .Net Code  
 The Elements of C# Style  
 F# High Performance  
 Concurrency in .NET  
 R High Performance Programming  
 Programming C# 8.0  
 Writing High-Performance .NET Code  
 CLR Via C#  
 High-performance Java Platform Computing  
 Writing High-Performance .NET Code, 2nd Edition  
 High Performance Python  
 Writing Secure Code  
 Pro .NET Benchmarking  
 Microsoft .NET Web Performance (O'Reilly) (O'Reilly)  
 High-Performance Programming in C# and .NET  
 Performance Analysis and Tuning on Modern CPUs

*Writing High  
 Performance Net Code*

*Downloaded from  
<ftp.wtvq.com> by guest*

---

## COLON MOYER

---

C# 7 and .NET Core 2.0 High Performance  
 No Starch Press  
 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 Clean Code in C# Understand .NET memory management internal workings, pitfalls, and techniques in order to effectively avoid a wide range of performance and scalability problems in your software. Despite automatic memory

management in .NET, there are many advantages to be found in understanding how .NET memory works and how you can best write software that interacts with it efficiently and effectively. Pro .NET Memory Management is your comprehensive guide to writing better software by understanding and working with memory management in .NET. Thoroughly vetted by the .NET Team at Microsoft, this book contains 25 valuable troubleshooting scenarios designed to help diagnose challenging memory problems. Readers will also benefit from a multitude of .NET memory management "rules" to live by that introduce methods for writing memory-aware code and the means for avoiding common, destructive pitfalls. What You'll Learn Understand the theoretical underpinnings of automatic memory management Take a deep dive

into every aspect of .NET memory management, including detailed coverage of garbage collection (GC) implementation, that would otherwise take years of experience to acquire. Get practical advice on how this knowledge can be applied in real-world software development. Use practical knowledge of tools related to .NET memory management to diagnose various memory-related issues. Explore various aspects of advanced memory management, including use of Span and Memory types. Who This Book Is For .NET developers, solution architects, and performance engineers.

*C++ High Performance* "O'Reilly Media, Inc."

Integrate proven performance and scalability techniques throughout the .NET application life cycle--and gain an edge in building better-performing products. This guide presents a robust framework organized by task and role, helping developers, architects, testers, and administrators prioritize and implement the best options at the appropriate time. It offers focused, end-to-end guidance--including processes for modeling performance and techniques for measuring, testing, and fine-tuning your applications. You'll also get tips direct from Microsoft development teams for improving the performance and scalability of managed code; Microsoft ASP.NET, ADO.NET, and SQL Server; Web services; .NET Remoting; XML; and more. The book features a "How To" section that details the steps for a number of specific performance-related tasks, such as adding performance counters and using the common language runtime (CLR) profiler. PATTERNS & PRACTICES guides are reviewed and approved by Microsoft engineering teams, consultants, partners, and customers--delivering accurate, real-world information that's been technically validated and tested.

*High Performance Web Sites* Pearson Education

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.

*Pro .NET Memory Management* Xlibris Corporation

Want your web site to display more quickly? This book presents 14 specific rules that will cut 25% to 50% off response time when users request a page. Author Steve Souders, in his job as Chief Performance Yahoo!, collected these best practices while optimizing some of the most-visited pages on the Web. Even sites that had already been highly optimized, such as Yahoo! Search and the Yahoo! Front Page, were able to benefit from these surprisingly simple performance guidelines. The rules in *High Performance Web Sites* explain how you can optimize the performance of the Ajax, CSS, JavaScript, Flash, and images that you've already built into your site -- adjustments that are critical for any rich web application. Other sources of information pay a lot of attention to tuning web servers, databases, and hardware, but the bulk of display time is taken up on the browser side and by the communication between server and browser. *High Performance Web Sites* covers every aspect of that process. Each performance rule is supported by specific examples, and code snippets are available on the book's companion web site. The rules include how to: Make Fewer HTTP Requests Use a Content Delivery Network Add an Expires Header Gzip Components Put Stylesheets at the Top Put Scripts at the Bottom Avoid CSS Expressions Make JavaScript and CSS External Reduce DNS Lookups Minify JavaScript Avoid Redirects Remove Duplicates Scripts Configure

ETags Make Ajax Cacheable If you're building pages for high traffic destinations and want to optimize the experience of users visiting your site, this book is indispensable. "If everyone would implement just 20% of Steve's guidelines, the Web would be dramatically better place. Between this book and Steve's YSlow extension, there's really no excuse for having a sluggish web site anymore." - Joe Hewitt, Developer of Firebug debugger and Mozilla's DOM Inspector "Steve Souders has done a fantastic job of distilling a massive, semi-arcane art down to a set of concise, actionable, pragmatic engineering steps that will change the world of web performance." -Eric Lawrence, Developer of the Fiddler Web Debugger, Microsoft Corporation  
*Clojure High Performance Programming* Apress

The programming language C# was built with the future of application development in mind. Pursuing that vision, C#'s designers succeeded in creating a safe, simple, component-based, high-performance language that works effectively with Microsoft's .NET Framework. Now the favored language among those programming for the Microsoft platform, C# continues to grow in popularity as more developers discover its strength and flexibility. And, from the start, C# developers have relied on *Programming C#* both as an introduction to the language and a means of further building their skills. The fourth edition of *Programming C#*--the top-selling C# book on the market--has been updated to the C# ISO standard as well as changes to Microsoft's implementation of the language. It also provides notes and warnings on C# 1.1 and C# 2.0. Aimed at experienced programmers and web developers, *Programming C#, 4th Edition*, doesn't waste too much time on the basics. Rather, it focuses on the features and programming patterns unique to the C# language. New C# 2005 features covered in-depth include: Visual Studio 2005 Generics Collection interfaces and iterators Anonymous methods New ADO.NET data controls Fundamentals of Object-Oriented Programming Author Jesse Liberty, an acclaimed web programming expert and entrepreneur, teaches C# in a way that experienced programmers will appreciate by grounding its applications firmly in the context of Microsoft's .NET platform and the development of desktop and Internet applications. Liberty also incorporates reader suggestions from previous editions to help create the most consumer-friendly guide possible.

**Programming C#** Packt Publishing Ltd

Become an expert at writing fast and high performant code in Clojure 1.7.0 About This Book Enhance code performance by using appropriate Clojure features Improve the efficiency of applications and plan their deployment A hands-on guide to designing Clojure programs to get the best performance Who This Book Is For This book is intended for intermediate Clojure developers who are looking to get a good grip on achieving optimum performance. Having a basic knowledge of Java would be helpful. What You Will Learn Identify performance issues in Clojure programs using different profiling tools Master techniques to achieve numerical performance in Clojure Use Criterion library to measure latency of Clojure expressions Exploit Java features in Clojure code to enhance performance Avoid reflection and boxing with type hints Understand Clojure's concurrency and state-management primitives in depth Measure and monitor performance, and understand optimization techniques In Detail Clojure treats code as data and has a macro system. It focuses on programming with immutable values and explicit progression-of-time constructs, which are intended to facilitate the development of more robust programs, particularly multithreaded ones. It is built with performance, pragmatism, and simplicity in mind. Like most general purpose languages, various Clojure features have different performance characteristics that one should know in order to write high performance code. This book shows you how to evaluate the performance implications of various Clojure abstractions, discover their underpinnings, and apply the right approach for optimum performance in real-world programs. It starts by helping you classify various use cases and the need for them with respect to performance and analysis of various performance aspects. You will also learn the performance vocabulary that experts use throughout the world and discover various Clojure data structures, abstractions, and their performance characteristics. Further, the book will guide you through enhancing performance by using Java interoperability and JVM-specific features from Clojure. It also highlights the importance of using the right concurrent data structure and Java concurrency abstractions. This book also sheds light on performance metrics for measuring, how to measure, and how to visualize and monitor the collected data. At the end of the book, you will learn to run a performance profiler, identify bottlenecks, tune performance, and

refactor code to get a better performance. Style and approach An easy-to-follow guide full of real-world examples and self-sufficient code snippets that will help you get your hands dirty with high performance programming with Clojure. *Improving .NET Application Performance and Scalability* Microsoft Press Network Programming with Go teaches you how to write clean, secure network software with the programming language designed to make it seem easy. Build simple, reliable, network software Combining the best parts of many other programming languages, Go is fast, scalable, and designed for high-performance networking and multiprocessing. In other words, it's perfect for network programming. Network Programming with Go will help you leverage Go to write secure, readable, production-ready network code. In the early chapters, you'll learn the basics of networking and traffic routing. Then you'll put that knowledge to use as the book guides you through writing programs that communicate using TCP, UDP, and Unix sockets to ensure reliable data transmission. As you progress, you'll explore higher-level network protocols like HTTP and HTTP/2 and build applications that securely interact with servers, clients, and APIs over a network using TLS. You'll also learn: Internet Protocol basics, such as the structure of IPv4 and IPv6, multicasting, DNS, and network address translation Methods of ensuring reliability in socket-level communications Ways to use handlers, middleware, and multiplexers to build capable HTTP applications with minimal code Tools for incorporating authentication and encryption into your applications using TLS Methods to serialize data for storage or transmission in Go-friendly formats like JSON, Gob, XML, and protocol buffers Ways of instrumenting your code to provide metrics about requests, errors, and more Approaches for setting up your application to run in the cloud (and reasons why you might want to) Network Programming with Go is all you'll need to take advantage of Go's built-in concurrency, rapid compiling, and rich standard library. Covers Go 1.15 (Backward compatible with Go 1.12 and higher)

#### **Network Programming for the Microsoft .NET Framework**

Independently Published Are you an RTL or system designer that is currently using, moving, or planning to move to an HLS design environment? Finally, a comprehensive guide for designing hardware using C++ is here. Michael Fingeroff's High-Level Synthesis

Blue Book presents the most effective C++ synthesis coding style for achieving high quality RTL. Master a totally new design methodology for coding increasingly complex designs! This book provides a step-by-step approach to using C++ as a hardware design language, including an introduction to the basics of HLS using concepts familiar to RTL designers. Each chapter provides easy-to-understand C++ examples, along with hardware and timing diagrams where appropriate. The book progresses from simple concepts such as sequential logic design to more complicated topics such as memory architecture and hierarchical subsystem design. Later chapters bring together many of the earlier HLS design concepts through their application in simplified design examples. These examples illustrate the fundamental principles behind C++ hardware design, which will translate to much larger designs. Although this book focuses primarily on C and C++ to present the basics of C++ synthesis, all of the concepts are equally applicable to SystemC when describing the core algorithmic part of a design. On completion of this book, readers should be well on their way to becoming experts in high-level synthesis.

*Write Great Code, Volume 1* No Starch Press

Today's programmers are often narrowly trained because the industry moves too fast. That's where Write Great Code, Volume 1: Understanding the Machine comes in. This, the first of four volumes by author Randall Hyde, teaches important concepts of machine organization in a language-independent fashion, giving programmers what they need to know to write great code in any language, without the usual overhead of learning assembly language to master this topic. A solid foundation in software engineering, The Write Great Code series will help programmers make wiser choices with respect to programming statements and data types when writing software. *Metaprogramming in .NET* Apress 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.

*C# and .NET Core Test-Driven*

*Development* Packt Publishing Ltd

Enhance your applications' performance using best practices for benchmarking, application profiling, asynchronous programming, designing responsive UIs, gRPC communication, and distributed applications

**Key Features**

- Make the best use of performance enhancements in C# 10.0 and .NET 6
- Boost application performance by identifying hardware bottlenecks and common performance pitfalls
- Get to grips with best practices and techniques for improving the scalability of distributed systems

**Book Description** Writing high-performance code while building an application is crucial, and over the years, Microsoft has focused on delivering various performance-related improvements within the .NET ecosystem. This book will help you understand the aspects involved in designing responsive, resilient, and high-performance applications with the new version of C# and .NET. You will start by understanding the foundation of high-performance code and the latest performance-related improvements in C# 10.0 and .NET 6. Next, you'll learn how to use tracing and diagnostics to track down performance issues and the cause of memory leaks. The chapters that follow then show you how to enhance the performance of your networked applications and various ways to improve directory tasks, file tasks, and more. Later, you'll go on to improve data querying performance and write responsive user interfaces. You'll also discover how you can use cloud providers such as Microsoft Azure to build scalable distributed solutions. Finally, you'll explore various ways to process code synchronously, asynchronously, and in parallel to reduce the time it takes to process a series of tasks. By the end of this C# programming book, you'll have the confidence you need to build highly resilient, high-performance applications that meet your customer's demands. What you will learn

  - Use correct types and collections to enhance application performance
  - Profile, benchmark, and identify performance issues with the

codebase

- Explore how to best perform queries on LINQ to improve an application's performance
- Effectively utilize a number of CPUs and cores through asynchronous programming
- Build responsive user interfaces with WinForms, WPF, MAUI, and WinUI
- Benchmark ADO.NET, Entity Framework Core, and Dapper for data access
- Implement CQRS and event sourcing and build and deploy microservices

Who this book is for This book is for software engineers, professional software developers, performance engineers, and application profilers looking to improve the speed of their code or take their skills to the next level to gain a competitive advantage. You should be a proficient C# programmer who can already put the language to good use and is also comfortable using Microsoft Visual Studio 2022.

*Hands-On High Performance with Go* Packt Publishing Ltd

Learn how to apply a test-driven development process by building ready C# 7 and .NET Core applications. **Key Features** Create tests to quickly detect and resolve issues when writing portable code Uncover code integration issues that improve code quality using continuous integration Set up and use data-driven unit testing to verify your code **Book Description** This book guides developers to create robust, production-ready C# 7 and .NET Core applications through the practice of test-driven development process. In *C# and .NET Core Test-Driven Development*, you will learn the different stages of the TDD life cycle, basics of TDD, best practices, and anti-patterns. It will teach you how to create an ASP.NET Core MVC sample application, write testable code with SOLID principles and set up a dependency injection for your sample application. Next, you will learn the xUnit testing framework and learn how to use its attributes and assertions. You'll see how to create data-driven unit tests and mock dependencies in your code. You will understand the difference between running and debugging your tests on .NET Core on LINUX versus Windows and Visual Studio. As you move forward, you will be able to create a healthy continuous integration process for your sample application using GitHub, TeamCity, Cake, and Microsoft VSTS. By the end of this book, you will have learned how to write clean and robust code through the effective practice of TDD, set up CI build steps to test and build applications as well as how to package application for deployment on NuGet. What you will learn Write flexible, maintainable, and verifiable

code for .NET Core Write testable code using SOLID principles and dependency injections Recognize the characteristics of a good unit test Structure and group your unit test Use mock objects to handle dependencies Set up an end-to-end continuous integration process Who this book is for This book is for .NET developers who would like to build efficient applications by implementing principles of test-driven development. C# programming and working knowledge of VS is assumed. [Functional Programming in C#, Second Edition](#) Ben Watson

The authors make performance issues the central topic, with very in-depth discussion and examples.

*Hands-On JavaScript High Performance* Packt Publishing Ltd

No one has done more to conquer the performance limitations of the PC than Michael Abrash, a software engineer for Microsoft. His complete works are contained in this massive volume, including everything he has written about performance coding and real-time graphics. The CD-ROM contains the entire text in Adobe Acrobat 3.0 format, allowing fast searches for specific facts.

*Expert C# 5.0* Packt Publishing Ltd

Learn the best ways to exploit the networking APIs in the .NET Framework—and deliver greater flexibility, interoperability, and power to your network applications. Three network programming specialists from Microsoft demonstrate how to use the System.Net namespace, which contains the core classes for network development, across a range of scenarios—from writing your first socket-based application to developing high performance N-tier Web applications. Whether you're a Win32-based network programmer moving to the .NET Framework or you're an XML Web Services or .NET Remoting developer looking for a better understanding of how these technologies relate to the network, this singular reference delivers the code and instruction you need. Discover how to: Use serialization techniques—binary, XML, and SOAP—to package complex data Enable concurrent programming—and increase application flexibility—with threads and asynchronous I/O Resolve URIs without writing protocol-specific code Communicate over IP networks using DNS, IPv4, and IPv6 Learn core to advanced socket programming techniques for both client and server operations Employ the .NET Web-related classes to retrieve HTTP content—proxy servers, cookies, credentials, and more Fine-tune XML Web services in your network programs and customize the underlying HTTP protocol

for optimal efficiency Write a .NET Remoting custom channel to communicate over any data transmission medium Utilize code access security, encryption technology, and HTTP authentication techniques Boost application performance and scalability by streamlining resources *Delphi High Performance* Simon and Schuster

### 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

### High-level Synthesis "O'Reilly Media, Inc."

Proven methodologies and concurrency techniques that will help you write faster and better code with Go programming Key FeaturesExplore Go's profiling tools to write faster programs by identifying and fixing bottlenecksAddress Go-specific performance issues such as memory allocation and garbage collectionDelve into the subtleties of concurrency and discover how to successfully implement it in everyday applicationsBook Description Go is an easy-to-write language that is popular among developers thanks to its features such as concurrency, portability, and ability to reduce complexity. This Golang book will teach you how to construct idiomatic Go code that is reusable and highly performant. Starting with an introduction to performance concepts, you'll understand the ideology behind Go's performance. You'll then learn how to effectively implement Go data structures and algorithms along with exploring data manipulation and organization to write programs for scalable software. This book covers channels and goroutines for parallelism and concurrency to write high-performance code for distributed systems. As you advance, you'll learn how to manage memory effectively. You'll explore the compute unified device architecture (CUDA) application programming interface (API), use containers to build Go code, and work with the Go build cache for quicker compilation. You'll also get to grips with profiling and tracing Go code for detecting bottlenecks in your system. Finally, you'll evaluate clusters and job queues for performance optimization and monitor the application for performance regression. By the end of this Go programming book, you'll be able to improve existing code and fulfill customer requirements by writing efficient programs. What you will

learnOrganize and manipulate data effectively with clusters and job queuesExplore commonly applied Go data structures and algorithmsWrite anonymous functions in Go to build reusable appsProfile and trace Go apps to reduce bottlenecks and improve efficiencyDeploy, monitor, and iterate Go programs with a focus on performanceDive into memory management and CPU and GPU parallelism in GoWho this book is for This Golang book is a must for developers and professionals who have an intermediate-to-advanced understanding of Go programming, and are interested in improving their speed of code execution. [Michael Abrash's Graphics Programming Black Book](#) Packt Publishing Ltd C# is undeniably one of the most versatile programming languages available to engineers today. With this comprehensive guide, you'll learn just how powerful the combination of C# and .NET can be. Author Ian Griffiths guides you through C# 8.0 fundamentals and techniques for building cloud, web, and desktop applications. Designed for experienced programmers, this book provides many code examples to help you work with the nuts and bolts of C#, such as generics, LINQ, and asynchronous programming features. You'll get up to speed on .NET Core and the latest C# 8.0 additions, including asynchronous streams, nullable references, pattern matching, default interface implementation, ranges and new indexing syntax, and changes in the .NET tool chain. Discover how C# supports fundamental coding features, such as classes, other custom types, collections, and error handling Learn how to write high-performance memory-efficient code with .NET Core's Span and Memory types Query and process diverse data sources, such as in-memory object models, databases, data streams, and XML documents with LINQ Use .NET's multithreading features to exploit your computer's parallel processing capabilities Learn how asynchronous language features can help improve application responsiveness and scalability *High Performance JavaScript* Packt Publishing Ltd

10, ACT