

---

# Writing High Performance Net Code

---

Reactive Applications with Akka.NET  
 Functional Programming in C#, Second Edition  
 High Performance JavaScript  
 Learning .NET High-performance Programming  
 Julia High Performance  
 Network Programming for the Microsoft .NET Framework  
 Clojure High Performance Programming  
 Hands-On JavaScript High Performance  
 Writing High-Performance .NET Code  
 Code Like a Pro in C#  
 Modern Software Engineering  
 CLR Via C#  
 Michael Abrash's Graphics Programming Black Book  
 High Performance Python  
 C# 7 and .NET Core 2.0 High Performance  
 Fundamentals of Computer Programming with C#  
 Writing High-Performance .Net Code  
 Performance Tuning and Optimizing ASP. Net Applications  
 Writing High-Performance .NET Code, 2nd Edition  
 Hands-On High Performance with Go  
 Pro .NET Performance  
 Crafting Interpreters  
 Improving .NET Application Performance and Scalability  
 The Elements of C# Style  
 Metaprogramming in .NET  
 High Performance Django  
 Concurrency in .NET  
 C++ High Performance  
 Programming C# 8.0  
 High-performance Java Platform Computing  
 Pro .NET Memory Management  
 Performance Analysis and Tuning on Modern CPUs  
 Delphi High Performance  
 C# 5.0 Unleashed  
 Writing Secure Code  
 IPython Interactive Computing and Visualization Cookbook  
 Pro .NET Benchmarking  
 Entity Framework Core in Action  
 Microsoft .NET Web Performance (Book) (Paperback)  
 Expert C# 5.0

Writing High Performance Net Code

Downloaded from [archive.imba.com](http://archive.imba.com) by guest

---

## MALIK ADRIENNE

---

Reactive Applications with Akka.NET Faber Publishing  
 10, ACT  
 Functional Programming in C#, Second Edition Packt Publishing Ltd

### 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 Performance JavaScript* Simon and Schuster

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 Features: Discover effective techniques for accessing DOM, minimizing painting, and using a V8 engine to optimize JavaScript. Understand what makes the web tick and create apps that look and feel like native desktop applications. Explore modern JavaScript frameworks like Svelte.js for building next-gen web apps. Book 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.

**Learning .NET High-performance Programming** Genever Benning

PLEASE PROVIDE COURSE INFORMATION PLEASE PROVIDE

[Julia High Performance](#) Independently Published

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* Simon and Schuster

Build fast, scalable, and high performing applications with Delphi. Key Features: Build efficient and concurrent applications in Delphi with focused examples. Identify performance bottlenecks and apply the correct algorithm to increase the performance of applications. Delve into parallel programming and memory management to optimize your code. Book Description: Delphi is a cross-platform Integrated Development Environment (IDE) that supports rapid application development for Microsoft Windows, Apple Mac OS X, Google Android, iOS, and now Linux with RAD Studio 10.2. This book will be your guide to build efficient high performance applications with Delphi. The book begins by explaining how to find performance bottlenecks and apply the correct algorithm to fix them. It will teach you how to improve your algorithms before taking you through parallel programming. You'll then explore various tools to build highly concurrent applications. After that, you'll delve into improving the performance of your code and master cross-platform RTL improvements. Finally, we'll go through memory management with Delphi and you'll see how to leverage several external libraries to write better performing programs. By the end of the book, you'll have the knowledge to create high performance applications with Delphi. What you will learn: Find performance bottlenecks and easily mitigate them. Discover different approaches to fix algorithms. Understand parallel programming and work with various tools included with Delphi. Master the RTL for code optimization. Explore memory managers and their implementation. Leverage external libraries to write better performing programs. Who this book is for: This book is for Delphi developers who would like to build high performance applications with Delphi. Prior knowledge of Delphi is assumed.

*Clojure High Performance Programming* Simon and Schuster

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.

*Hands-On JavaScript High Performance* Simon and Schuster  
Dig deep and master the intricacies of the common language runtime (CLR) and the .NET Framework. Written by a highly regarded programming expert and consultant to the Microsoft .NET team, this guide is ideal for developers building any kind of application--including Microsoft ASP.NET, Windows Forms, Microsoft SQL Server, Web services, and console applications. You'll get hands-on instruction and extensive code C# code samples to help you tackle the tough topics and develop high-performance applications. Discover how to: Build, deploy, administer, and version applications, components, and shared assemblies Design types using constants, fields, constructors, methods, properties, and events Work effectively with the CLR's special types including enumerators, arrays, and strings Declare, create, and use delegates to expose callback functions Define and employ re-usable algorithms with interfaces and generics Define, use, and detect custom attributes Use exception handling to build robust, reliable, and security-enhanced components Manage memory automatically with the garbage collector and work with native resources Apply CLR Hosting, AppDomains, assembly loading, and reflection to build dynamically extensible applications PLUS--Get code samples on the Web

*Writing High-Performance .NET Code* Simon and Schuster  
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 Criterium 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.

*Code Like a Pro in C#* "O'Reilly Media, Inc."

Despite using them every day, most software engineers know little about how programming languages are designed and implemented. For many, their only experience with that corner of computer science was a terrifying "compilers" class that they suffered through in undergrad and tried to blot from their memory as soon as they had scribbled their last NFA to DFA conversion on the final exam. That fearsome reputation belies a field that is rich with useful techniques and not so difficult as some of its practitioners might have you believe. A better understanding of how programming languages are built will make you a stronger software engineer and teach you concepts and data structures you'll use the rest of your coding days. You might even have fun. This book teaches you everything you need to know to implement a full-featured, efficient scripting language. You'll learn both high-level concepts around parsing and semantics and gritty details like bytecode representation and garbage collection. Your brain will light up with new ideas, and your hands will get dirty and calloused. Starting from main(), you will build a language that features rich syntax, dynamic typing, garbage collection, lexical scope, first-class functions, closures, classes, and inheritance. All packed into a few thousand lines of clean, fast code that you thoroughly understand because you wrote each one yourself.

*Modern Software Engineering* Microsoft Press

Howard and LeBlanc (both are security experts with Microsoft) discuss the need for security and outline its general principles before outlining secure coding techniques. Testing, installation, documentation, and error messages are also covered.

Appendices discuss dangerous APIs, dismiss pathetic excuses, and provide security checklists. The book explains how systems can be attacked, uses anecdotes to illustrate common mistakes, and offers advice on making systems secure. Annotation copyrighted by Book News, Inc., Portland, OR.

**CLR Via C#** 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

**Michael Abrash's Graphics Programming Black Book** Apress Expert C# 5.0 is a book about getting the best from C#. It's based on the principle that to write good, high-performance, robust applications you need to understand what's going on deep under the hood. If you are already experienced with writing managed applications and want to learn more about how to get the best from the language at an advanced level, then this is the book for you. Expert C# 5.0 discusses the familiar C# language in forensic detail. Examining familiar elements closely to reveal how they really work. Key language features that you are already familiar with, such as Enums, Strings and Collections, are teased apart and examined under the twin microscopes of MSIL (Intermediate Language) and the Windbg debugger to see what's really going on behind the scenes as your code is compiled and passed to the CLR. This unparalleled depth of explanation will help you to become a true master of the C# language and architect better crafted applications that work in the most efficient and reliable way possible. It will also give you the insight you need to rapidly identify and fix the stubborn coding faults that others may be unable to diagnose.

#### **High Performance Python** [O'Reilly](#)

Improve the speed of your code and optimize the performance of your apps Key Features Understand the common performance pitfalls and improve your application's performance Get to grips with multi-threaded and asynchronous programming in C# Develop highly performant applications on .NET Core using microservice architecture Book Description While writing an application, performance is paramount. Performance tuning for realworld applications often involves activities geared toward finding bottlenecks; however, this cannot solve the dreaded problem of slower code. If you want to improve the speed of your code and optimize an application's performance, then this book is for you. C# 7 and .NET Core 2.0 High Performance begins with an introduction to the new features of what?explaining how they help in improving an application's performance. Learn to identify the bottlenecks in writing programs and highlight common performance pitfalls, and learn strategies to detect and resolve these issues early. You will explore multithreading and asynchronous programming with .NET Core and learn the importance and efficient use of data structures. This is followed with memory management techniques and design guidelines to increase an application's performance. Gradually, the book will show you the importance of microservices architecture for building highly performant applications and implementing resiliency and security in .NET Core. After reading this book, you will learn how to structure and build scalable, optimized, and robust applications in C#7 and .NET. What you will learn Measure application performance using BenchmarkDotNet Explore the techniques to write multithreaded applications Leverage TPL and PLinq libraries to perform asynchronous operations Get familiar with data structures to write optimized code Understand design techniques to increase your application's performance Learn about memory management techniques in .NET Core Develop a containerized application based on microservices architecture Learn tools and techniques to monitor application performance Who this book is for This book is for .NET developers looking at improving the speed of their code or simply wanting to take their skills to the next level. Basic C# knowledge is assumed.

#### **C# 7 and .NET Core 2.0 High Performance** Apress

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 [Fundamentals of Computer Programming with C#](#) Packt Publishing Ltd

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#.

*Writing High-Performance .Net Code* Cambridge University Press Design and develop high performing programs with Julia About This Book Learn to code high reliability and high performance programs Stand out from the crowd by developing code that runs faster than your peers' codes This book is intended for developers who are interested in high performance technical programming. Who This Book Is For This book is for beginner and intermediate Julia programmers who are interested in high performance technical computing. You will have a basic familiarity with Julia syntax, and have written some small programs in the language. What You Will Learn Discover the secrets behind Julia's speed Get a sense of the possibilities and limitations of Julia's performance Analyze the performance of Julia programs Measure the time and memory taken by Julia programs Create fast machine code using Julia's type information Define and call functions without compromising Julia's performance Understand number types in Julia Use Julia arrays to write high performance code Get an overview of Julia's distributed computing capabilities In Detail Julia is a high performance, high-level dynamic language designed to address the requirements of high-level numerical and scientific computing. Julia brings solutions to the complexities faced by developers while developing elegant and high performing code. Julia High Performance will take you on a journey to understand the performance characteristics of your Julia programs, and enables you to utilize the promise of near C levels of performance in Julia. You will learn to analyze and measure the performance of Julia code, understand how to avoid bottlenecks, and design your program for the highest possible performance. In this book, you will also see how Julia uses type information to achieve its performance goals, and how to use multiple dispatch to help the compiler to emit high performance machine code. Numbers and their arrays are obviously the key structures in scientific computing - you will see how Julia's design makes them fast. The last chapter will give you a taste of Julia's distributed computing capabilities. Style and approach This is a hands-on manual that

will give you good explanations about the important concepts related to Julia programming.

Performance Tuning and Optimizing ASP. Net Applications  
"O'Reilly Media, Inc."

Summary Concurrency in .NET teaches you how to build concurrent and scalable programs in .NET using the functional paradigm. This intermediate-level guide is aimed at developers, architects, and passionate computer programmers who are interested in writing code with improved speed and effectiveness by adopting a declarative and pain-free programming style. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Unlock the incredible performance built into your multi-processor machines. Concurrent applications run faster because they spread work across processor cores, performing several tasks at the same time. Modern tools and techniques on the .NET platform, including parallel LINQ, functional programming, asynchronous programming, and the Task Parallel Library, offer powerful alternatives to traditional thread-based concurrency. About the Book Concurrency in .NET teaches you to write code that delivers the speed you need for performance-sensitive applications. Featuring examples in both C# and F#, this book guides you through concurrent and parallel designs that emphasize functional programming in theory and practice. You'll start with the foundations of concurrency and master essential techniques and design practices to optimize code running on modern multiprocessor systems. What's Inside The most important concurrency abstractions Employing the agent programming model Implementing real-time event-stream processing Executing unbounded asynchronous operations Best concurrent practices and patterns that apply to all platforms About the Reader For readers skilled with C# or F#. About the Book Riccardo Terrell is a seasoned software engineer and Microsoft MVP who is passionate about functional programming. He has over 20 years' experience delivering cost-effective technology solutions in a competitive business environment. Table of Contents PART 1 - Benefits of functional programming applicable to concurrent programs Functional concurrency foundations Functional programming techniques for concurrency Functional data structures and immutability PART 2 - How to

Related with Writing High Performance Net Code:

- 3 5 Minute Speeches : [click here](#)

approach the different parts of a concurrent program The basics of processing big data: data parallelism, part 1 PLINQ and MapReduce: data parallelism, part 2 Real-time event streams: functional reactive programming Task-based functional parallelism Task asynchronicity for the win Asynchronous functional programming in F# Functional combinators for fluent concurrent programming Applying reactive programming everywhere with agents Parallel workflow and agent programming with TPL Dataflow PART 3 - Modern patterns of concurrent programming applied Recipes and design patterns for successful concurrent programming Building a scalable mobile app with concurrent functional programming Writing High-Performance .NET Code, 2nd Edition O'Reilly Media 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

**Hands-On High Performance with Go** Prentice Hall

Intended to anyone interested in numerical computing and data science: students, researchers, teachers, engineers, analysts, hobbyists... Basic knowledge of Python/NumPy is recommended. Some skills in mathematics will help you understand the theory behind the computational methods.