1.

Record Nr.	UNINA9910828281303321
Autore	Esposito Antonio
Titolo	Learning .NET high-performance programming : learn everything you need to know about performance-oriented programming for the .NET framework / / Antonio Esposito
Pubbl/distr/stampa	Birmingham, England ; ; Mumbai, [India] : , : Packt Publishing, , 2015 ©2015
Edizione	[1st edition]
Descrizione fisica	1 online resource (305 p.)
Collana	Professional expertise distilled
Disciplina	005.276
Soggetti	Microsoft .NET Framework
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index. "Professional Expertise Distilled"--Cover.
Nota di contenuto	Cover; Copyright; Credits; About the Author; About the Reviewers; www.PacktPub.com; Table of Contents; Preface; Chapter 1: Performance Thoughts; Understanding performance; Performance as a requirement; Performance engineering; Performance aspects; Latency; Throughput; Resource usage; Availability/reliability; Scalability; Efficiency; Class of applications; Case study: performance aspects of a desktop application; Case study: performance aspects of a mobile application; Case study: performance aspects of a server application; Performance concerns as time changes; Technical overview Multithreaded programmingParallel programming; Distributed computing; Grid computing; Summary; Chapter 2: Architecting High-performance .NET Code; Software architecture; Performance concerns about architecture; Object-oriented design principles; Single responsibility principle; The open-closed principle; The Liskov substitution principle; That interface segregation principle; The dependency inversion principle; Common designs and architectures; Layered architecture; Performance concerns; Model-View-Controller and ASP.NET MVC; Performance concerns; Model-View-ViewModel and XAML Performance concerns3-tier architecture; Performance concerns; Service-Oriented Architecture (SOA); Standardized service contract;

Service loose coupling; Service abstraction; Service reusability; Service autonomy; Service statelessness; Service discoverability; Service composability; Performance concerns; Architecture comparison; Common platform architectures; Architecting desktop applications; Architecting mobile applications; Architecting web applications; Architecting cloud web applications; Performance considerations; Caching, when and where; PLINQ everywhere; Inversion of Control (IoC)

Lazy loadingReusability of code; Agnostic versus idiom-powered implementation; Short coding; Remote computation; Cloud versus on-premise applications; Summary; Chapter 3: CLR Internals; Introduction to CLR; Memory management; Garbage collection; Large object heap; Collection tuning; Working with AppDomains; IDisposable interface; Threading; Multithreading synchronization; Locks; Signaling locks; Drawbacks of locks; Exception handling; Summary; Chapter 4: Asynchronous Programming; Understanding asynchronous programming; Asynchronous programming theory; Asynchronous Programming Model (APM)

Event-based Asynchronous Pattern (EAP)Task-based Asynchronous Pattern (TAP); Task creation; Task synchronization; Task exception handling; Task cancellation; Task continuation; Task factories; Task UI synchronization; Async/await; Summary; Chapter 5: Programming for Parallelism; Parallel programming; Task parallelism; Data parallelism; Task parallelism with TPL; Data parallelism with TPL; ThreadPool tuning; Parallel execution abortion; Partitions; Sliding parallel programming; Integrated querying with LINQ; Data parallelism with PLINQ; Partitioning optimization; Summary

Chapter 6: Programming for Math & Engineering

Sommario/riassunto	If you are a .NET developer with an understanding of application development, but want to learn how to optimize the performance of your applications, this is the book for you. Basic knowledge of C# is expected.