

Mayur Ramgir, Nick Samoylov

Java 9 High Performance

Practical techniques and best practices for optimizing Java applications through concurrency, reactive programming, and more



Packt>

Java 9 High Performance

Finally, a book that focuses on the practicalities rather than theory of Java application performance tuning. This book will be your one-stop guide for optimization of your Java application.

We will begin by looking at the new features and APIs of JDK 9. You will then be taught the techniques and practical solutions of JVM tuning, how to make the best use of the garbage collector, and find out how to measure the performance with microbenchmarking. Moving ahead, you will be introduced to multithreading and learning about concurrent programming with Java 9 to build highly elastic and resilient software system. You will learn how to fine-tune your code for the best results. You will discover techniques on how to improve performance and reduce various bottlenecks in your data processing. We'll also cover the best practices of Java programming that will help you improve the quality of your codebase.

By the end of the book, you will be armed with the knowledge to build and deploy efficient, scalable, and responsive applications in Java.

Things you will learn:

- Work with JIT compilers
- Understand the use of profiling tools
- Generate JSON with code examples
- Leverage the command-line tools to speed up application development
- Build microservices in Java 9
- Explore the use of APIs to improve application code
- Speed up your application with reactive programming and concurrency

Java 9 High Performance

Mayur Ramgiri, Nick Samoylov



Java 9 High Performance

Practical techniques and best practices for optimizing Java applications through concurrency, reactive programming, and more

Mayur Ramgir
Nick Samoylov

Packt>

BIRMINGHAM - MUMBAI

Java 9 High Performance

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: November 2017

Production reference: 1251017

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78712-078-5

www.packtpub.com

Credits

Authors

Mayur Ramgir
Nick Samoylov

Reviewer

Aristides Villarreal Bravo

Commissioning Editor

Kunal Parikh

Acquisition Editor

Denim Pinto

Content Development Editor

Rohit Kumar Singh

Technical Editor

Ruvika Rao

Copy Editors

Pranjali Chury
Karuna Narayanan
Laxmi Subramanian

Project Coordinator

Vaidehi Sawant

Proofreader

Safis Editing

Indexer

Francy Puthiry

Graphics

Jason Monteiro

Production Coordinator

Shraddha Falebhai

About the Authors

Mayur Ramgir has more than 16 years of experience in the software industry, working at various levels. He is a Sun Certified Java Programmer and Oracle Certified SQL Database Expert. He completed an MS in computational science and engineering at Georgia Tech, USA (rank 7th in the world for computer science), and an M.Sc. in multimedia application and virtual environments at University of Sussex, UK. He has also attended various universities for other degrees and courses, such as MIT for applied software security, and University of Oxford for system and software security.

He is the CEO of a software company, Zonopact, Inc. headquartered in Boston, USA, which specializes in bringing innovative applications based on AI, robotics, big data, and more. He has single-handedly developed Zonopact's flagship product, Clintra (B2B-integrated AI-assisted business management software). He is also the inventor of two patent pending technologies, ZPOD (an automated cloud-based medical kiosk system) and ZPIC (an AI-enabled robotic in-car camera system). Apart from this, he is also a prolific business writer who has authored two international award-winning books, *Unbarred Innovation: A Pathway to Greatest Discoveries* and *Evolve Like a Butterfly: A Metamorphic Approach to Leadership*.

He was featured on the TV and in print media, including Fox News, NBC News, CBS News, Fox Business, Bloomberg International TV, Forbes, Inc. magazine, Daily Mirror, and The Huffington Post. He is also a contributing author of New York Daily Newspaper, the Software Development Times magazine, Newsmax Finance, AlleyWatch, Singapore's top entrepreneurship magazine *Young Upstarts*, and several more. He is frequently invited as a guest lecturer at various technical and management schools. He has also been invited as a judge at an international innovation challenge competition (Living Talent) in Dubai in December 2017.

Nick Samoylov was born in Moscow, raised in Ukraine, and lived in the Crimea. He graduated as an engineer-physicist from Moscow Institute of Physics and Technologies and has even worked as a theoretical physicist. He has learned programming as a tool for testing his mathematical models using FORTRAN and C++.

After the demise of the USSR, Nick created and successfully ran a software company, but was forced to close it under the pressure of governmental and criminal rackets. In 1999, with his wife, Luda, and two daughters, he emigrated to the USA and has been living in Colorado since then.

Nick adopted Java in 1997 and used it for his work as a software developer-contractor for a variety of companies, including BEA Systems, Warner Telecom, and Boeing. For Boeing, he and his wife, also a Java programmer, developed a system to load application data to an airplane via the internet.

Nick's current projects are related to machine learning and developing a highly scalable system of microservices using non-blocking reactive technologies, including Vert.x, RxJava, and RESTful webservices on Linux deployed in a cloud.

Nick and Luda have two daughters who graduated from Harvard and Tufts universities, respectively. One of their daughters has also received a doctoral degree from Brown University and now works as a professor in the University of California in Chico. Their other daughter is an executive director of the investment bank JPMorgan, in Madrid, Spain.

In his free time, Nick likes to read (mostly non-fiction), write (fiction novels and blogs), and hike the Rocky Mountains.

About the Reviewer

Aristides Villarreal Bravo is a Java developer, a member of the NetBeans Dream Team, Java User Groups, and a developer of the jMoordb framework. He lives in Panamá. He has organized and participated in various conferences and seminars related to Java JavaEE, NetBeans, NetBeans Platform, free software, and mobile devices, nationally and internationally. He is a CEO of Javscz Software Developers.

He has done the technical reviews of the books, *Primefaces BluePrints*; *Apache Hive CookBook*; *Dart By Example*; *ExpressJS BluePrints*; *Java 9 CookBook*; *Angular UI Development with PrimeNG*; *Java EE Development with Eclipse, Second Edition*; *Mastering Google App Engine*; and *PrimeFaces Cookbook, Second Edition*; all published by Packt.

Thanks to Llelat for believing in dreams.

www.PacktPub.com

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www.packtpub.com/mapt>

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at <https://www.amazon.com/dp/1787120783>.

If you'd like to join our team of regular reviewers, you can e-mail us at customerreviews@packtpub.com. We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

Table of Contents

Chapter 1: Learning Java 9 Underlying Performance Improvements	7
Introducing the new features of Java 9	8
Modular development and its impact	8
Quick introduction to modules	13
String operations performance	17
Compact string	17
The world of heap	18
Why bother compressing strings?	18
What did they do?	19
What is the escape route?	19
What is the performance gain?	19
Indify String Concatenation	20
Invokedynamic	22
Storing interned strings in CDS archives	24
Concurrency performance	25
Compiler improvements	26
Tiered Attribution	27
Ahead-of-Time compilation	27
Security manager improvements	28
Graphics rasterizers	30
Summary	30
Chapter 2: Identifying Performance Bottlenecks	31
CPU infrastructure	32
Memory utilization	36
Java heap	36
Java stack	38
Java heap versus stack memory	40
Escape analysis	41
Common memory problems	45
OutOfMemory	45
Mismanagement of object life cycle: java.lang.OutOfMemoryError: Java heap space	46
java.lang.OutOfMemoryError: GC Overhead limit exceeded	47
java.lang.OutOfMemoryError: Permgen space	48
java.lang.OutOfMemoryError: Metaspace	49
java.lang.OutOfMemoryError: Unable to create new native thread	49

java.lang.OutOfMemoryError: request size bytes for reason	50
java.lang.OutOfMemoryError: Requested array size exceeds VM limit	51
Out of memory: kill process or sacrifice child	51
StackOverflow	52
Database operations	55
I/O operations	58
Network operations	62
WebSocket Client API	64
Summary	65
Chapter 3: Learning How to Troubleshoot Code	67
Measuring performance	68
Performance checklist	68
Basic principles of troubleshooting	69
Why some developers find troubleshooting difficult?	70
Setting up the environment for troubleshooting	70
Importance of error handling	71
A basic try-catch-finally block and its usages	72
What are try-catch-finally blocks?	72
Usage of try-catch-finally blocks	72
Mistakes programmers make when using error handling	73
Why swallowing exception is bad?	74
Use of logging	75
Logging vs Debugging	75
What are the main components?	76
How to do logging?	76
What logging frameworks are available?	78
Java Util Logging	78
Log4j 2	80
Logback	81
Performance	84
Support	84
Ease of Use	85
Best use case for logging	85
Analyzing thread dump	86
What is a thread dump?	86
Benefits of thread dump analysis	87
How to collect a thread dump	87
How to analyze a thread dump	88
Best practices	89
Analyzing heap dump	89
What is a heap dump	89
Benefits of analyzing heap dumps	90

When do you analyze a heap dump	90
How to collect heap dump	91
How to analyze heap dump	91
Best practices	92
Summary	92
Chapter 4: Learning How to Use Profiling Tools	93
Introducing profiling	94
Why a developer should use a profiling tool	95
The art of profiling	99
Profiling areas	100
CPU profiling	100
Memory profiling	101
Thread profiling	103
The different types of Java profiling tool	104
Jmap	104
VisualVM	106
JProfiler	120
XRebel	134
YourKit	135
JRockit	136
BTrace	137
AQtime	137
Reporting issues	138
Summary	138
Chapter 5: Understanding Garbage Collection and Making Use of It	141
Understanding the Java 9 garbage collector for top optimization	142
Garbage Collection	142
The principle of GC	143
Benefits and demerits of garbage collection	143
Unpredictability	144
GC methods	144
Escape analysis	144
Reference counting	145
Space reduction	145
Cycles	145
Atomic tasks	146
GC in Java	146
Understanding the current Parallel GC	147
Generations	148
Heap sizing	149

Parallelism	150
Why parallelism?	152
The new G1 collector	153
A brief history	154
Impact on programming	154
Deep insight on G1GC	155
Comparing G1 collector to ParallelGC	156
Common points	156
The qualities of ParallelGC	157
The qualities of G1GC	157
Concurrent mode failures	158
Understanding memory errors	160
Summary	161
Chapter 6: Optimizing Code with Microbenchmarking	163
The concept of microbenchmarking	164
Microbenchmarks in Java	165
Understanding the JMH	169
Setting up the JMH	170
Building up in the JMH	172
Benchmark modes in the JMH	173
Throughput measure	173
Time-based measures	174
Applying the JMH in Java projects	175
What is Maven?	176
Why use microbenchmarking?	179
The challenges associated with microbenchmarks	181
Wrong solutions from microbenchmarks	182
Wrong component use	183
Wrong hierarchy	183
Wrong memory use	183
Use of a specific environment	184
The benefits of using the JMH	184
Understanding the limitations of the JMH	188
Summary	189
Chapter 7: Speeding Up JSON Generation	191
A transparent understanding of JSON in Java	192
A brief history of JSON and examples	198
Understanding JSON generation	200
What is Javax.json?	201
JSON object model	201
JSON stream	202

What are events?	202
How does an event work?	203
Types in JSON for object modeling	204
Types in JSON for streaming API	204
JSON trees	207
Understanding the tree model	212
Converting CSV to JSON	213
Converting Excel to JSON	215
Understanding value trees	217
The benefits of using JSON	220
Summary	221
Chapter 8: Tools for Higher Productivity and Faster Application	223
The JShell tool usage	223
Creating a JShell session and setting context	224
JShell commands	229
Ahead-of-Time (AOT)	239
Static versus dynamic compilation	240
The AOT commands and procedures	241
Summary	244
Chapter 9: Multithreading and Reactive Programming	245
Prerequisites	246
Thread pools	253
Monitoring threads	260
Sizing thread pool executors	272
Thread synchronization	274
Tuning JVM	282
Reactive programming	286
Summary	293
Chapter 10: Microservices	295
Why microservices?	296
Building microservices	299
Container-less deployment	322
Self-contained microservices	324
In-container deployment	325
Summary	327
Chapter 11: Making Use of New APIs to Improve Your Code	329
Filtering streams	330
Basic filtering	330

Using other stream operations for filtering	335
Stack-walking APIs	342
Stack analysis before Java 9	342
New better way to walk the stack	345
Convenience factory methods for collections	349
Why new factory methods?	350
The new factory methods in action	357
CompletableFuture in support of asynchronous processing	360
The CompletableFuture API overview	361
The CompletableFuture API enhancements in Java 9	363
The problem and the solution using Future	364
The solution with CompletableFuture	369
Other useful features of CompletableFuture	371
Stream API improvements	374
Summary	375
Index	377
