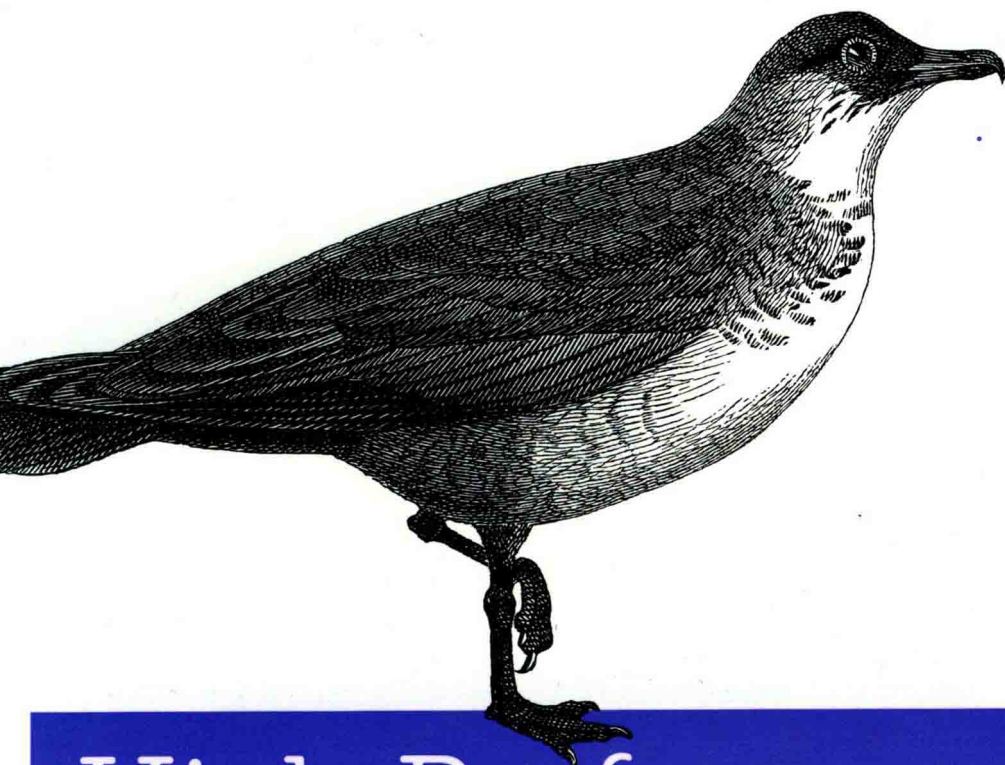


O'REILLY®



High Performance iOS Apps

高性能iOS应用 (影印版)

东南大学出版社

Gaurav Vaish 著

高性能iOS应用 (影印版)

High Performance iOS Apps

Gaurav Vaish 著

常州大学图书馆
藏书章

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目(CIP)数据

高性能 iOS 应用:英文/(印)高瑞夫·维希(Gaurav Vaish)著. —影印本. —南京:东南大学出版社,2017.2
书名原文:High Performance iOS Apps
ISBN 978-7-5641-7002-8

I. ①高… II. ①高… III. ①移动终端—应用程序—程序设计—英文 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 008145 号
图字:10-2015-260 号

© 2016 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2017. Authorized reprint of the original English edition, 2015 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2016。

英文影印版由东南大学出版社出版 2017。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有,未得书面许可,本书的任何部分和全部不得以任何形式重制。

高性能 iOS 应用(影印版)

出版发行:东南大学出版社

地 址:南京四牌楼 2 号 邮编:210096

出 版 人:江建中

网 址:<http://www.seupress.com>

电子邮件:press@seupress.com

印 刷:常州市武进第三印刷有限公司

开 本:787 毫米×980 毫米 16 开本

印 张:29

字 数:568 千字

版 次:2017 年 2 月第 1 版

印 次:2017 年 2 月第 1 次印刷

书 号:ISBN 978-7-5641-7002-8

定 价:92.00 元

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

Table of Contents

This book is dedicated to Agryav Vaish, my son.

Preface	xi
---------	----

Part I. Getting Started

1. Performance in Mobile Apps	1
Defining Performance	2
Performance Metrics	3
Battery	3
Power Consumption	4
Installation Time	4
Execution Speed	5
Responsiveness	5
Local Storage	6
Interoperability	6
Network Condition	7
Bandwidth	7
Data Retention	8
Multitouch Support	8
Single Sign-on	9
Security	9
Camera	10
App Profiling	10
Logging	11
Instrumentation	11
Measurement	12
Profiling and Heap Dump	12
Crash Reporting Setup	13

Preface

You may already have an amazing iOS app, or be developing one. Everything looks great except for some kinks that prohibit users from giving the app that final fifth star or prevent you from releasing it.

Issues like a jitter when the user goes to the n th item in table view or the app hogging the network or consuming a lot battery are some of the visible symptoms that the user will be aware of. However, the problems lie somewhere deeper.

Tuning an app for performance is a never-ending task, especially with new app features, OS versions, third-party libraries, and device configurations. These are only some of the things that keep developers on their toes about app performance.

According to a study, about a quarter of users abandon an app if it does not load within 3 seconds, and about a third of users will tell others about their poor experience.

Users want apps that are fast, responsive, and do not hog resources. This book takes a look at various aspects that negatively impact use and outlines how you can tune *your* app for performance.

Who Should Read This Book

If you have been writing iOS apps and publishing them to the App Store, you have an implicit goal to make them better, faster, and more fluid—not to mention loved by your users. And if you are continuously looking for ways to achieve that goal, then this book is just for you.

You should already have working knowledge of Objective-C and iOS. This book is not about jump-starting with Objective-C or iOS, though the fundamentals will be revisited for completeness when necessary.

Why I Wrote This Book

iPhone OS 1 and iPhone 1 were launched in June 2007. During their early releases, developers were busy flushing out code and releasing apps for a wider audience. As the hardware, OS, networks, and overall ecosystem advanced over time—and as new user interface (UI) and engineering design patterns emerged—apps matured in functionality, stability, and performance.

More often than not, performance is an afterthought. And I can, to some degree, agree with that approach. After all, it is important to first get the functionality right, rather than worrying about performance. Thinking about optimizing early in the development lifecycle is more commonly referred to as *premature optimization*. But when poor performance hits, it hits hard.

The primary goal of this book is to show you how to write optimal code from an engineering perspective.

This book is not about theoretical computer science, data structures, and algorithms for faster execution of a task. You can find tons of literature on those topics. It covers best practices for implementing an app in such a manner that even in non-ideal conditions (low storage space, poor network, low battery, etc.), users still can use your app effectively and love using it. Generally, it will not be possible to optimize across all the parameters, but you'll be able to achieve the best trade-off considering the available options.

Navigating This Book

This book is organized in five parts. Each part is comprised of one or more chapters that detail the theme of the particular part. Each chapter begins with a brief abstract outlining the topics that will be covered.

Part I provides an overview of how to measure performance. Chapter 1 discusses the aspects that we want to optimize and outlines the parameters we want to measure as part of tracking app performance.

Part II reviews the key optimizations that are core to any app. Chapter 2 addresses memory management. It describes the memory management model and object reference types. It also discusses best practices for design patterns that impact memory consumption, namely singleton and dependency injection.

Chapter 3 covers energy and techniques for minimizing consumption. Chapter 4 provides an overview of concurrent programming. It describes the various options available and gives a comparative analysis.

Part III covers performance optimization techniques specific to iOS app development. Chapter 5 provides in-depth coverage of the application lifecycle. It details how lifecycle events can be utilized to ensure resources are used effectively.

Chapter 6 reviews optimization techniques specifically for the user interface. Chapter 7 and Chapter 8 deal with networking and data sharing, respectively. Chapter 9 provides an in-depth review of security. It describes how enhanced security can slow down an app and discusses what can be done for an effective trade-off.

Part IV reviews the non-code aspects of performance. Chapter 10 covers testing, and performance testing in particular. It also discusses continuous integration and automation.

Chapter 11 provides an overview of the tools you can use to measure performance during development. Chapter 12 discusses instrumentation and analytics, and how it can be used to collect performance-related data from apps in production.

Part V focuses on iOS 9. Chapter 13 provides an outline of the changes in this release and how they impact the code you write from the perspective of performance.

Code snippets are provided throughout the book where applicable. Several of these snippets can be used either as is or with minimal modifications in your app. Other snippets may need to be further adapted to suit your app.

Each chapter also provides a set of best practices associated with the topic. It may not always be possible to follow all of the best practices in a single app. The decisions of which ones you want to trade off against others will be based on your app's requirements.

Online Resources

This book refers to several online blogs, articles, tutorials, and other references. The links to those references are provided where applicable. Should you feel any reference has been missed out accidentally, feel free to contact the publisher or the author.

This book also references screenshots from several apps. The apps are copyright of their respective owners. The screenshots have been added for educational and illustrative purposes only.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user. It also highlights new code in program listings as compared to an earlier example that may have been shown.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a general note.



This icon signifies a tip or suggestion.



This icon indicates a warning or a caution.

Using Code Examples


Supplemental material (code examples, exercises, etc.) is available for download at <https://github.com/gvaish/high-performance-ios-apps>.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*High Performance iOS Apps* by Gaurav Vaish (O’Reilly). Copyright 2016 Gaurav Vaish, 978-1-491-91100-6.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online

 **Safari**® *Safari Books Online* is an on-demand digital library that delivers expert content in both book and video form from the world’s leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government, education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O’Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at <http://bit.ly/hp-ios-apps>.

To comment or ask technical questions about this book, send email to bookquestions@oreilly.com.

For more information about our books, courses, conferences, and news, see our website at <http://www.oreilly.com>.

Find us on Facebook: <http://facebook.com/oreilly>

Follow us on Twitter: <http://twitter.com/oreillymedia>

Watch us on YouTube: <http://www.youtube.com/oreillymedia>

Acknowledgments

Even though it might seem so at times, nobody writes a book alone.

I am grateful to have worked closely with Daryl Low, Distinguished Architect at Yahoo!, in developing the monetization SDK from the ground up. He provided guidance on several sections in the book. It has always been fun and exciting to work with him on prototypical apps just to test performance extremes or identify the root cause of an error.

I would like to thank Walter Pezzini, who heads mobile DevOps engineering at Yahoo. He provided key insights on my understanding of continuous integration and a delivery pipeline, and what it takes to build a high-quality system.

As an author, it is easy to assume a few things to be known by the reader that may not always be the case. A big thanks goes to Chris Devers, Laura Savino, and Niklas Saers for their reviews, which identified areas that I needed to clarify. Thanks for also providing feedback that helped elevate overall content quality.

I would like to acknowledge Renu Chandel, my wife, for her continuous push to force me to complete this book. It would not have been finished if not for her. Thanks for all the coffee!

Last, but not the least, thanks to O'Reilly Media for publishing this book.

This book would not have been complete without collective efforts from all of you. Thanks, team.

Table of Contents

Preface.....	xiii
--------------	------

Part I. Getting Started

1. Performance in Mobile Apps.....	3
Defining Performance	4
Performance Metrics	4
Memory	4
Power Consumption	4
Initialization Time	5
Execution Speed	5
Responsiveness	6
Local Storage	6
Interoperability	7
Network Condition	8
Bandwidth	9
Data Refresh	9
Multiuser Support	9
Single Sign-on	11
Security	11
Crashes	11
App Profiling	12
Sampling	12
Instrumentation	12
Measurement	12
Project and Code Setup	13
Crash Reporting Setup	14

Instrumenting Your App	15
Logging	20
Summary	23

Part II. Core Optimizations

2. Memory Management	27
Memory Consumption	28
Stack Size	28
Heap Size	29
Memory Management Model	32
Autoreleasing Objects	34
Autorelease Pool Blocks	35
Automatic Reference Counting	39
Rules of ARC	41
Reference Types	42
Variable Qualifiers	43
Property Qualifiers	44
Getting Your Hands Dirty	45
Photo Model	45
Storyboard Update	46
Method Implementations	47
Output Analysis	49
Zombies	50
Rules of Memory Management	51
Retain Cycles	52
Rules to Avoid Retain Cycles	53
Common Scenarios for Retain Cycles	54
Observers	67
Returning Errors	70
Weak Type: id	71
Solution to the Problem	72
Object Longevity and Leaks	74
Singletons	74
Finding Mystery Retains	77
Best Practices	78
Memory Usage in Production	79
Summary	80
3. Energy	81
CPU	81

Network	83
Location Manager and GPS	87
Optimal Initialization	89
Turn Off Inessential Features	90
Use Network Only If Essential	92
Background Location Services	92
NSTimers, NSThreads, and Location Services	93
Restart After App Kill	93
Screen	94
Animation	94
Video Play	94
Multiple Screens	94
Other Hardware	99
Battery Level and State-Aware Code	100
Profiling for Energy Use	102
Best Practices	103
Summary	106
4. Concurrent Programming	107
Threads	107
The Cost of Threads	108
Kernel Data Structures	108
Stack Size	108
Creation Time	109
GCD	109
Operations and Queues	110
Thread-Safe Code	112
Atomic Properties	112
Synchronized Blocks	113
Locks	115
Use Reader–Writer Locks for Concurrent Reads and Writes	121
Use Immutable Entities	123
Have a Central State Updater Service	128
State Observers and Notifications	134
Prefer Async over Sync	139
Summary	141
<hr/>	
Part III. iOS Performance	
5. Application Lifecycle	145
App Delegate	145

Application Launch	147
First Launch	150
Cold Start	158
Warm Launch	166
Launch After Upgrade	169
Push Notifications	169
Remote Notifications	169
Local Notifications	173
Background Fetch	174
Summary	177
6. User Interface	179
View Controller	181
View Load	183
View Hierarchy	184
View Visibility	187
View	189
UILabel	191
UIButton	192
UIImageView	194
UITableView	195
UIWebView	199
Custom Views	203
Auto Layout	210
Size Classes	212
New Interaction Features in iOS 8	217
Interactive Notifications	217
App Extensions	219
Summary	222
7. Network	223
Metrics and Measurement	223
DNS Lookup Time	224
SSL Handshake Time	226
Network Type	227
Latency	234
Networking API	237
App Deployment	238
Servers	238
Request	239
Data Format	240
Tools	241

Network Link Conditioner	241
AT&T Application Resource Optimizer	243
Charles	245
Summary	248
8. Data Sharing.....	251
Deep Linking	252
Pasteboards	258
Sharing Content	261
Document Interaction	262
Activities	271
iOS 8 Extensions	274
Configuration for Action and Share Extensions	276
Action Extensions	277
Share Extensions	278
Document Provider Extension	282
App Groups	287
Summary	289
9. Security.....	291
App Access	292
Anonymous Access	292
Authenticated Access	294
Network Security	299
Use HTTPS	299
Use Certificate Pinning	300
Local Storage	305
Data Sharing	312
Security and App Performance	312
Checklist	313
Summary	315

Part IV. Beyond Code

10. Testing and Release.....	319
Test Types	319
Definitions	320
Unit Testing	321
Setup	321
Writing Unit Tests	323
Code Coverage	325

Asynchronous Operations	330
Xcode 6 Bonus: Performance Unit Tests	331
Dependency Mocking	333
Other Frameworks	337
Functional Testing	338
Setup	339
Writing Functional Tests	341
Project Structure	344
Dependency Isolation	345
Testing and Component Design	347
Continuous Integration and Automation	349
Best Practices	351
Performance Testing	352
Summary	355
11. Tools	357
Accessibility Inspector	358
Xcode Accessibility Inspector	359
iOS Accessibility Inspector	360
Instruments	363
Using Instruments	364
Activity Monitor	367
Allocations	369
Leaks	372
Network	374
Time Profiler	374
Xcode View Debugger	375
PonyDebugger	377
Charles	382
Summary	388
12. Instrumentation and Analytics	389
Vocabulary	390
Instrumentation	391
Planning	392
Implementation	395
Deployment	398
Analytics	398
Real User Monitoring	399
Analytics Versus Real User Monitoring	399
Using RUM	399
Summary	400

Part V. iOS 9

13. iOS 9	403
Application Lifecycle	403
Universal Links	404
Search	407
Search Best Practices	412
User Interface	414
UIKit Framework	414
Safari Services Framework	418
Extensions	421
Content Blocker Extension	422
Spotlight Index Extension	426
App Thinning	426
Slicing	427
On Demand Resources	427
Bitcode	431
Summary	432
Index	433