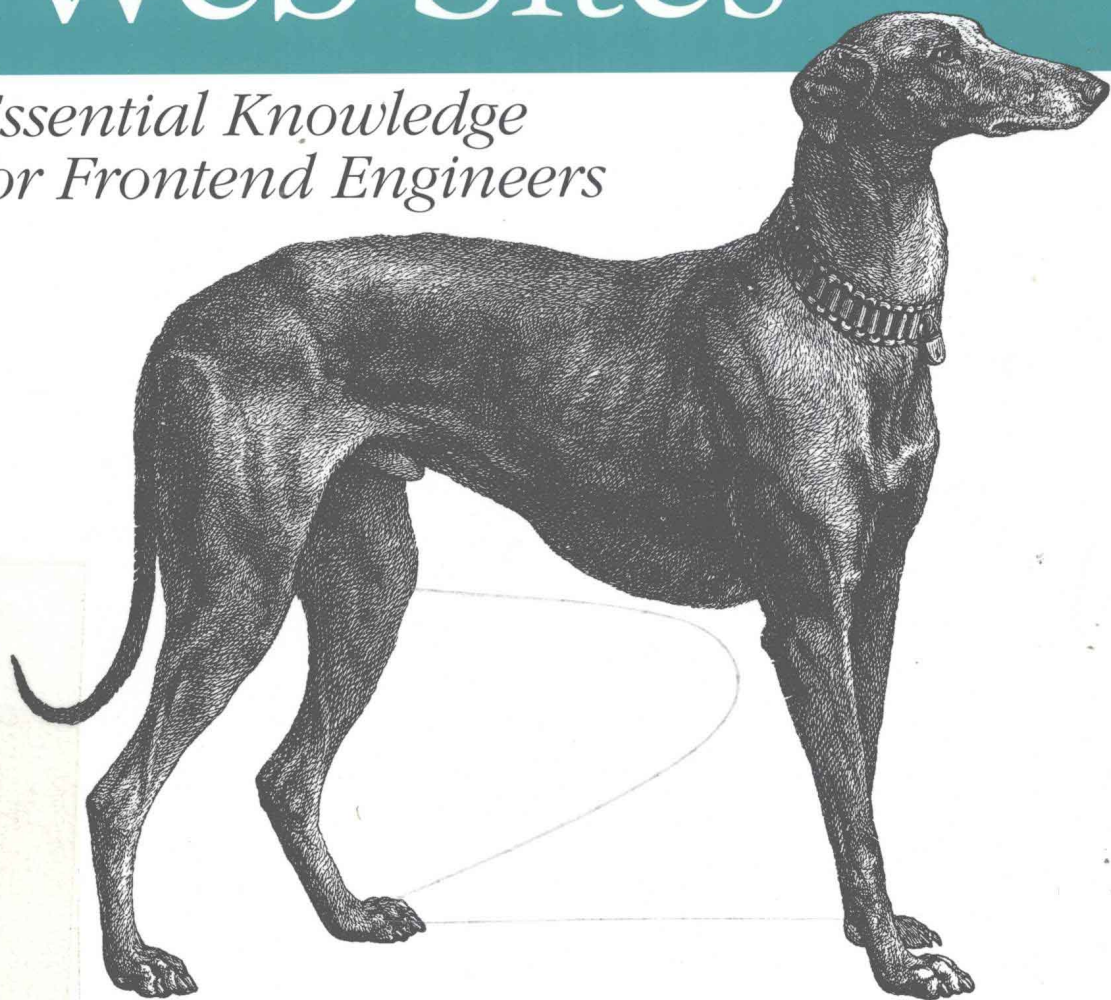高性能网站（影印版）

High Performance Web Sites

Essential Knowledge for Frontend Engineers
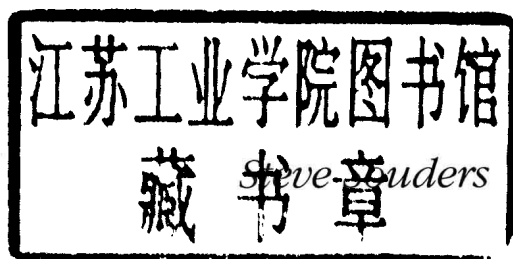
Steve Souders 著

Nate Koechley 序

高性能网站(影印版)
High Performance Web Sites

Steve Souders

O'REILLY®

Praise for *High Performance Web Sites*

"If everyone would implement just 20% of Steve's guidelines, the Web would be a 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

"As the stress and performance test lead for Zillow.com, I have been talking to all of the developers and operations folks to get them on board with the rules Steve outlined in this book, and they all ask how they can get a hold of this book. I think this should be a mandatory read for all new UE developers and performance engineers here."

> —Nate Moch, *www.zillow.com*

"*High Performance Web Sites* is an essential guide for every web developer. Steve offers straightforward, useful advice for making virtually any site noticeably faster."

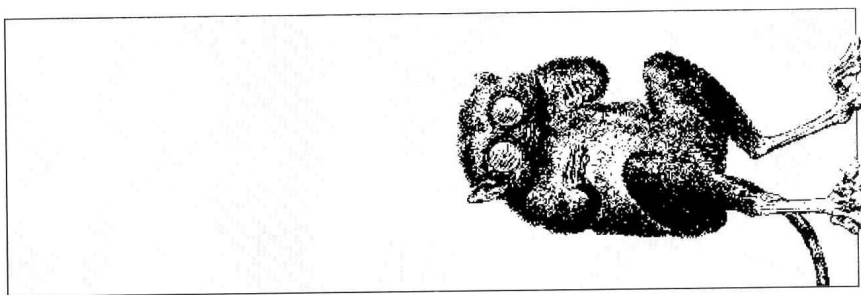> —Tony Chor, Group Program Manager, Internet Explorer team, Microsoft Corporation

高性能网站（影印版）

High Performance Web Sites

Foreword

You're lucky to be holding this book. More importantly, your web site's *users* are lucky. Implement even a few of the 14 techniques Steve shares in this groundbreaking book and your site will be faster immediately. Your users will thank you.

Here is why it matters. As a frontend engineer, you hold a tremendous amount of power and responsibility. You're the users' last line of defense. The decisions you make directly shape their experience. I believe our number one job is to take care of them and to give them what they want—quickly. This book is a toolbox to create happy users (and bosses, too). Best of all, once you put these techniques in place—in most cases, a one-time tweak—you'll be reaping the rewards far into the future.

This book will change your approach to performance optimization. When Steve began researching performance for our Platform Engineering group at Yahoo!, I believed performance was mainly a backend issue. But he showed that frontend issues account for 80% of total time. I thought frontend performance was about optimizing images and keeping CSS and JavaScript external, but the 176 pages and 14 rules you're holding in your hand right now are proof that it's much more.

I've applied his findings to several sites. Watching already-fast sites render nearly twice as quickly is tremendous. His methodology is sound, his data valid and extensive, and his findings compelling and impactful.

The discipline of frontend engineering is still young, but the book in your hands is an important step in the maturation of our craft. Together we'll raise expectations about the Web by creating better and faster (and therefore more enjoyable) interfaces and experiences.

Cheers to faster surfing!

<div align="right">

—Nate Koechley
Senior Frontend Engineer
Yahoo! User Interface (YUI) Team, Platform
Engineering, Yahoo! Inc.
San Francisco, August, 2007

</div>

Preface

In eighth grade, my history class studied the efficiency experts of the Industrial Revolution. I was enthralled by the techniques they used to identify and overcome bottlenecks in manufacturing. The most elegant improvement, in my mind, was the adjustable stepstool that afforded workers of different heights the ability to more easily reach the conveyor belt—a simple investment that resulted in improved performance for the life of the process.

Three decades later, I enjoy comparing the best practices in this book to that 19th-century stepstool. These best practices enhance an existing process. They require some upfront investment, but the cost is small—especially in comparison to the gains. And once these improvements are put in place, they continue to boost performance over the life of the development process. I hope you'll find these rules for building high performance web sites to be elegant improvements that benefit you and your users.

How This Book Is Organized

After two quick introductory chapters, I jump into the main part of this book: the 14 performance rules. Each rule is described, one per chapter, in priority order. Not every rule applies to every site, and not every site should apply a rule the same way, but each is worth considering. The final chapter of this book shows how to analyze web pages from a performance perspective, including some case studies.

Chapter A, *The Importance of Frontend Performance* explains that at least 80 percent of the time it takes to display a web page happens after the HTML document has been downloaded, and describes the importance of the techniques in this book.

Chapter B, *HTTP Overview* provides a short description of HTTP, highlighting the parts that are relevant to performance.

Chapter 1, *Rule 1: Make Fewer HTTP Requests* describes why extra HTTP requests have the biggest impact on performance, and discusses ways to reduce these HTTP requests including image maps, CSS sprites, inline images using data: URLs, and combining scripts and stylesheets.

Chapter 2, *Rule 2: Use a Content Delivery Network* highlights the advantages of using a content delivery network.

Chapter 3, *Rule 3: Add an Expires Header* digs into how a simple HTTP header dramatically improves your web pages by using the browser's cache.

Chapter 4, *Rule 4: Gzip Components* explains how compression works and how to enable it for your web servers, and discusses some of the compatibility issues that exist today.

Chapter 5, *Rule 5: Put Stylesheets at the Top* reveals how stylesheets affect the rendering of your page.

Chapter 6, *Rule 6: Put Scripts at the Bottom* shows how scripts affect rendering and downloading in the browser.

Chapter 7, *Rule 7: Avoid CSS Expressions* discusses the use of CSS expressions and the importance of quantifying their impact.

Chapter 8, *Rule 8: Make JavaScript and CSS External* talks about the tradeoffs of inlining your JavaScript and CSS versus putting them in external files.

Chapter 9, *Rule 9: Reduce DNS Lookups* highlights the often-overlooked impact of resolving domain names.

Chapter 10, *Rule 10: Minify JavaScript* quantifies the benefits of removing whitespace from your JavaScript.

Chapter 11, *Rule 11: Avoid Redirects* warns against using redirects, and provides alternatives that you can use instead.

Chapter 12, *Rule 12: Remove Duplicate Scripts* reveals what happens if a script is included twice in a page.

Chapter 13, *Rule 13: Configure ETags* describes how ETags work and why the default implementation is bad for anyone with more than one web server.

Chapter 14, *Rule 14: Make Ajax Cacheable* emphasizes the importance of keeping these performance rules in mind when using Ajax.

Chapter 15, *Deconstructing 10 Top Sites* gives examples of how to identify performance improvements in real-world web sites.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, file extensions, pathnames, directories, Unix utilities, and general emphasis.

`Constant width`

Indicates computer code in a broad sense. This includes commands, options, switches, variables, attributes, keys, functions, types, classes, namespaces, methods, modules, properties, parameters, values, objects, events, event handlers, XML tags, HTML tags, macros, the contents of files, and the output from commands.

HTTP requests and responses are designated graphically as shown in the following example.

`GET / HTTP/1.1` is an HTTP request header

`HTTP/1.1 200 OK` is an HTTP response header

Code Examples

Online examples can be found on this book's companion web site:

http://stevesouders.com/hpws

Examples are included in each chapter in the context in which they are discussed. They are also listed here for easy review.

No Image Map (Chapter 1)
http://stevesouders.com/hpws/imagemap-no.php

Image Map (Chapter 1)
http://stevesouders.com/hpws/imagemap.php

CSS Sprites (Chapter 1)
http://stevesouders.com/hpws/sprites.php

Inline Images (Chapter 1)
http://stevesouders.com/hpws/inline-images.php

Inline CSS Images (Chapter 1)
http://stevesouders.com/hpws/inline-css-images.php

Separate Scripts (Chapter 1)
http://stevesouders.com/hpws/combo-none.php

Combined Scripts (Chapter 1)
http://stevesouders.com/hpws/combo.php

CDN (Chapter 2)
http://stevesouders.com/hpws/ex-cdn.php

No CDN (Chapter 2)
http://stevesouders.com/hpws/ex-nocdn.php

No Expires (Chapter 3)
http://stevesouders.com/hpws/expiresoff.php

Far Future Expires (Chapter 3)
http://stevesouders.com/hpws/expireson.php

Nothing Gzipped (Chapter 4)
http://stevesouders.com/hpws/nogzip.html

HTML Gzipped (Chapter 4)
http://stevesouders.com/hpws/gzip-html.html

Everything Gzipped (Chapter 4)
http://stevesouders.com/hpws/gzip-all.html

CSS at the Bottom (Chapter 5)
http://stevesouders.com/hpws/css-bottom.php

CSS at the Top (Chapter 5)
http://stevesouders.com/hpws/css-top.php

CSS at the Top Using @import (Chapter 5)
http://stevesouders.com/hpws/css-top-import.php

CSS Flash of Unstyled Content (Chapter 5)
http://stevesouders.com/hpws/css-fouc.php

Scripts in the Middle (Chapter 6)
http://stevesouders.com/hpws/js-middle.php

Scripts Block Downloads (Chapter 6)
http://stevesouders.com/hpws/js-blocking.php

Scripts at the Top (Chapter 6)
http://stevesouders.com/hpws/js-top.php

Scripts at the Bottom (Chapter 6)
http://stevesouders.com/hpws/js-bottom.php

Scripts Top vs. Bottom (Chapter 6)
http://stevesouders.com/hpws/move-scripts.php

Deferred Scripts (Chapter 6)
http://stevesouders.com/hpws/js-defer.php

Expression Counter (Chapter 7)
http://stevesouders.com/hpws/expression-counter.php

One-Time Expressions (Chapter 7)
http://stevesouders.com/hpws/onetime-expressions.php

Event Handler (Chapter 7)
http://stevesouders.com/hpws/event-handler.php

Inlined JS and CSS (Chapter 8)
http://stevesouders.com/hpws/inlined.php

External JS and CSS (Chapter 8)
http://stevesouders.com/hpws/external.php

Cacheable External JS and CSS (Chapter 8)
http://stevesouders.com/hpws/external-cacheable.php

Post-Onload Download (Chapter 8)
http://stevesouders.com/hpws/post-onload.php

Dynamic Inlining (Chapter 8)
http://stevesouders.com/hpws/dynamic-inlining.php

Small Script Normal (Chapter 10)
http://stevesouders.com/hpws/js-small-normal.php

Small Script Minified (Chapter 10)
http://stevesouders.com/hpws/js-small-minify.php

Small Script Obfuscated (Chapter 10)
http://stevesouders.com/hpws/js-small-obfuscate.php

Large Script Normal (Chapter 10)
http://stevesouders.com/hpws/js-large-normal.php

Large Script Minified (Chapter 10)
http://stevesouders.com/hpws/js-large-minify.php

Large Script Obfuscated (Chapter 10)
http://stevesouders.com/hpws/js-large-obfuscate.php

XMLHttpRequest Beacon (Chapter 11)
http://stevesouders.com/hpws/xhr-beacon.php

Image Beacon (Chapter 11)
http://stevesouders.com/hpws/redir-beacon.php

Duplicate Scripts—Not Cached (Chapter 12)
http://stevesouders.com/hpws/dupe-scripts.php

Duplicate Scripts—Cached (Chapter 12)
http://stevesouders.com/hpws/dupe-scripts-cached.php

Duplicate Scripts—10 Cached (Chapter 12)
http://stevesouders.com/hpws/dupe-scripts-cached10.php

In general, you may use the code in this book and these online examples 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 Web Sites* by Steve Souders. Copyright 2007 Steve Souders, 978-0-596-52930-7."

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*.

Comments and Questions

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 and any additional information. You can access this page at:

http://www.oreilly.com/catalog/9780596529307

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

http://www.oreilly.com

Safari® Books Online

Safari
Books Online

When you see a Safari® Books Online icon on the cover of your favorite technology book, that means the book is available online through the O'Reilly Network Safari Bookshelf.

Safari offers a solution that's better than e-books. It's a virtual library that lets you easily search thousands of top tech books, cut and paste code samples, download chapters, and find quick answers when you need the most accurate, current information. Try it for free at *http://safari.oreilly.com*.

Acknowledgments

About the Author

Steve Souders holds down the job of Chief Performance Yahoo! at Yahoo! He's been at Yahoo! since 2000, working on many of the platforms and products within the company. He ran the development team for My Yahoo! before reaching his current position.

As Chief Performance Yahoo!, he has developed a set of best practices for making web sites faster. He builds tools for performance analysis and evangelizes these best practices and tools across Yahoo!'s product teams.

Prior to Yahoo!, Steve worked at several small to mid-size startups, including two companies he cofounded: Helix Systems and CoolSync. He also worked at General Magic, WhoWhere?, and Lycos. In the early 1980s, Steve caught the Artificial Intelligence bug and worked at a few companies doing research on Machine Learning. He received a B.S. in Systems Engineering from the University of Virginia and an M.S. in Management Science and Engineering from Stanford University.

Steve's interests are varied. He sits on the board of Freehand Systems and Fremont Hills Country Club, and he teaches Sunday School. He's played basketball with several NBA and WNBA players, but has recently retired and switched to Ultimate Frisbee. He was a member of the Universal Studios Internet Task Force, has rebuilt a 90-year-old carriage house, and participated in setting a Guinness world record. He has a wonderful wife and three daughters.

Colophon

The animal on the cover of *High Performance Web Sites* is a greyhound.

The fastest dog in the world, a greyhound can reach speeds of up to 45 miles per hour, enabled by its streamlined, narrow body; large lungs, heart, and muscles; double suspension gallop (two periods of a gait when all four feet are off the ground); and the flexibility of its spine. Although greyhounds are incredibly fast, they are actually low-energy dogs and lack endurance, requiring less exercise time than most dogs. For this reason, they're often referred to as "45-mile-per-hour couch potatoes" because when not chasing smaller prey (such as rabbits and cats), they are content to spend their days sleeping.

Greyhounds are one of the oldest breeds of dogs, appearing in art and literature throughout history. In ancient Egypt, greyhounds were often mummified and buried with their owners, and hieroglyphics from 4000 B.C.E. show a dog closely resembling the modern greyhound. In Greek and Roman mythology, greyhounds were often depicted with gods and goddesses. Greyhounds appeared in the writings of Homer, Chaucer, Shakespeare, and Cervantes, and they are the only type of dog mentioned in the Bible. They've long been appreciated for their intelligence, graceful form, athleticism, and loyalty.

During the early 1920s, modern greyhound racing was introduced into the United States. Smaller and lighter than show greyhounds, track greyhounds are selectively bred and usually stand between 25–29 inches tall and weigh 60–70 pounds. These dogs instinctively chase anything that moves quickly (as they are sighthounds, not bloodhounds), hence the *lure*—the mechanical hare they chase around the track. Greyhound racing is still a very popular spectator sport in the United States and, like horse racing, is enjoyed as a form of parimutuel gambling.

Greyhound racing is very controversial as the dogs experience little human contact and spend most of their non-racing time in crates. Once greyhounds are too old to race (somewhere between three and five years of age), many are euthanized, though there are now many rescue programs that find homes for retired racers. Because greyhounds are naturally docile and even-tempered, most adjust well to adoption and make wonderful pets.

The cover image is from *Cassell's Natural History*. The cover font is Adobe ITC Garamond. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condensed.

O'Reilly Media, Inc.介绍

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的50本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商 —— 每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体 —— 他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。