O'REILLY®

High Performance Responsive Design

BUILDING FASTER SITES ACROSS DEVICES

Tom Barker

O'REILLY®

High Performance Responsive Design

Yes, you *can* use responsive web design to create high performance, compelling websites. With this practical book, author Tom Barker demonstrates that responsive design is not just a frontend-only approach, but also a philosophy for taking advantage of the entire web stack. Responsive design patterns and anti-patterns, derived from heavily used real-world sites, are guiding principles throughout the book.

Ideal for frontend-focused web developers, this book shows you how to incorporate responsiveness and performance into your project plan, use Node.js for device-specific functionality on the backend, and write automated tests for a continuous integration environment. You'll explore many useful tools and responsive frameworks, and gain useful insights from Barker's own experience with responsive design along the way.

- Get a primer on web performance concepts, web runtime performance, and performance tracking tools
- Write functionality with Node.js that serves up a device-specific experience to the client
- Explore client-side solutions, such as lazy loading entire sections of a page—including images, styling, and content
- Validate service level agreements (SLAs) by writing automated tests with PhantomJS
- Examine several responsive frameworks, including the author's server-side framework, Ripple

Tom Barker, a software engineer, engineering manager, and solutions architect, is Director of Software Engineering and Development at Comcast, and an adjunct professor at Philadelphia University.

DESIGN/WEB DESIGN

US $34.99 **CAN $36.99**

ISBN: 978-1-491-94998-6

9 781491 949986 53499

Twitter: @oreillymedia
facebook.com/oreilly

High Performance Responsive Design

Building Faster Sites Across Devices

Tom Barker

O'REILLY® Beijing · Cambridge · Farnham · Köln · Sebastopol · Tokyo

High Performance Responsive Design
by Tom Barker

Copyright © 2015 Tom Barker. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editors: Mary Treseler and Nick Lombardi
Production Editor: Melanie Yarbrough
Copyeditor: Octal Publishing Services
Proofreader: Jasmine Kwityn
Indexer: Deadline Driven Publishing

Cover Designer: Eleanor Volkhausen
Interior Designers: Ron Bilodeau and Monica Kamsvaag
Illustrators: Rebecca Demarest
Compositor: Melanie Yarbrough

November 2014: First Edition.

Revision History for the First Edition:

2014-11-04 First release

See *http://oreilly.com/catalog/errata.csp?isbn=0636920033103* for release details.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *High Performance Responsive Design* and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

Although the publisher and author have used reasonable care in preparing this book, the information it contains is distributed "as is" and without warranties of any kind. This book is not intended as legal or financial advice, and not all of the recommendations may be suitable for your situation. Professional legal and financial advisors should be consulted, as needed. Neither the publisher nor the author shall be liable for any costs, expenses, or damages resulting from use of or reliance on the information contained in this book.

978-1-491-94998-6

[TI]

[contents]

[*Preface*]

Even though responsive design is a fairly ubiquitous term at this point, it is still considered mainly a frontend concern. In the minds of most developers, *responsive design* is also tightly coupled with media queries. With this book, however, I propose that responsive design is more of a philosophy rather than a technology: an ideal that can be approached from many different angles, from the traditional frontend-only approach, but also that there is enough information passed to the web server in each HTTP request to be responsive on the backend. And, in some cases, it is a better performing solution to push our responsiveness to the backend.

I originally intended to write this book because although I was seeing designers and engineers around me running with the ideas of producing responsive websites, I also saw business and product owners souring from the idea because they were keenly aware of the web performance costs even when we weren't always. By focusing only on the responsiveness of the client side and not looking for more performant options, we were slowly disillusioning our stakeholders on the benefits of responsiveness, and even our own effectiveness.

As I got under way with this book, it began to take on a life of its own. After we are paying attention to the performance of our responsive websites, how do we plan for that in our grooming sessions? If we are creating service-level agreements (SLAs) for the performance of our pages, how do we test that performance during development, in a *continual integration* environment?

I look to answer each of those questions in this book.

INTENDED AUDIENCE

I wrote this book specifically with web developers in mind, specifically frontend-focused web developers who might not have ventured onto the backend yet. It's for this reason why I didn't rehash all of the existing frontend performance best practices for CSS that you can find anywhere else. That is also the reason I kept JavaScript as the primary language used in the book, especially NodeJS for all of the backend code samples.

With that said, there are enough introductory materials and explanatory notes that designers, technology leaders, and developers of every experience level and specialization should be able to benefit from the information within this book.

CHAPTER DESCRIPTIONS

In Chapter 1, I use the top 50 most trafficked sites as a sample dataset to derive common design patterns and anti-patterns in use for responsive design. These patterns and anti-patterns will be guiding principles for us throughout the book. We also look at the idea of *mdot* sites, and discuss their pros and cons.

Chapter 2 presents a primer on web performance concepts, web runtime performance, as well as tools to track performance. This is intended as an introduction if you aren't already familiar with web performance concepts. It's also a good refresher on concepts that aren't talked about as frequently, such as memory consumption on the client side.

Chapter 3 explores incorporating responsiveness, specifically an SLA for specifying performance of our responsive websites, into the planning and grooming phases of our projects.

Chapter 4 looks at implementing performance-responsive concepts to the backend. We use NodeJS to write functionality that serves up a device-specific experience to the client. We also look at using third-party device libraries to give greater context of client capabilities rather than just examining the User Agent string and deriving device capabilities ourselves.

In Chapter 5, we look at frontend solutions to implement the performance design patterns that we identified in Chapter 1. We look at the picture element, and the secret attribute to only load device-specific

images. We also look at the concept of lazy loading both images and whole chunks of a page based on client capabilities. Finally, we explore client-side device library APIs to determine form factor.

Chapter 6 uses PhantomJS to write automated tests to validate our performance SLAs and integrate these tests into a Jenkins continuous integration environment.

We close out the book with Chapter 7, in which we look at and evaluate the current frameworks available to build responsive web pages, using such criteria as how easy they are to use, what patterns and anti-patterns they use, what dependencies they have, and how much they add to the overall page payload. We also walk through Ripple, the server-side boilerplate framework that I open sourced based on the code examples from Chapter 4.

NOTES

When writing any technology book, the pace of technology will always be faster than the pace at which we can write, edit, and publish to scale—though I have to say that O'Reilly does a great job of getting the content of their books in reader's hands as quickly as possible with their Early Access program. That said, the case study of the Alexa top 50 sites in the United States presented in Chapter 1 was conducted back in December of 2013, and since then, there are new sites in the Alexa list, the remaining sites have updated their pages, and several browser iterations with updated handling of resource loading and preloading have come out. The same is true for any proposed standards that I talk about; by the time you read this, they might have been updated or altered before being finalized.

That progress occurs is an inevitability; however, the ideas and concepts behind the tactical implementations are what are most important.

ACKNOWLEDGMENTS

I want to thank my beautiful wife, Lynn, for her patience with me as I spent the majority of a year writing this book at night and over weekends. The same goes for my children—I tried to only write late at night when they were asleep, but I wasn't always successful with that, and so I appreciate their patience and understanding.

I want to thank Mary Treseler for giving the book a chance and for her feedback. I want to express my gratitude to Colleen Lobner, Nick Lombardi, Melanie Yarbrough, and Dianne Russell for help getting it over the finish line. I also want to thank Ilya Grigorik, Lara Swanson, Clarissa Peterson, and Jason Pamental; their feedback was vital to the completion of the book.

[1]

State of the Industry of Responsive Design

The Problem with Responsive Design

I WAS SITTING IN A ROADMAP PLANNING SESSION WITH ONE OF MY TEAMS AND OUR PRODUCT PERSON, and we were discussing a redesign of our video section when my team lead started talking about how we were planning to make the video experience for our website responsive. We described having one page that would load our default HTML5 video player but would resize and load assets and playlists of different video types depending on what devices our users used to view the page. It was going to be beautiful, all encompassing, and open our video viewership up to a range of devices that had previously been locked out of the video experience that we offered.

Our product owner wrinkled her nose and said, "Well about that, we have somewhat of a bad taste about the idea of responsiveness after how the responsive home page turned out."

That took me by surprise. What was wrong with our responsive home page? I started doing some research.

The impression from the product team was that it was heavy and slow to load. When it was demonstrated for them on developer laptops, it looked great, but when they tried to show it on actual devices for their executives, it took a long time to load—too long.

I took a look at *waterfall charts*[1] for both the desktop and the smartphone rendering of the home page. What I saw was something that in time I began noticing in a lot of other websites when I became aware of what to look for.

The smartphone rendering loaded all of the same assets as the desktop version, plus an additional CSS and sprite file. Figure 1-1 illustrates that this made the payload of the smartphone rendering slightly larger than the desktop version (1.2 MB versus 952 KB), and it added two additional HTTP requests.

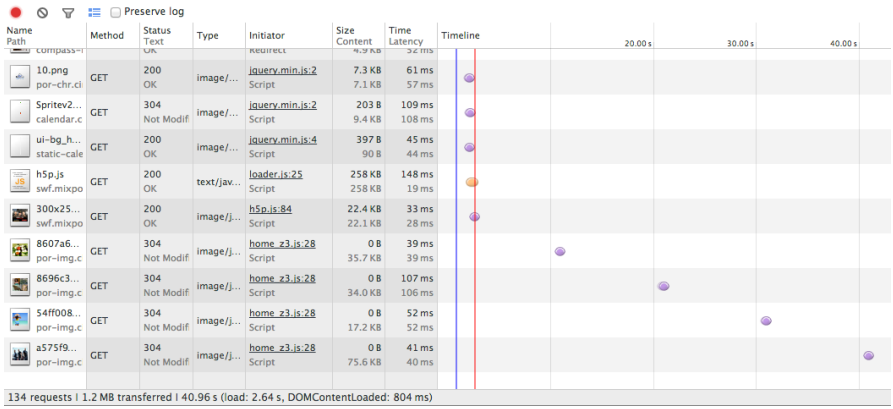

FIGURE 1-1

A waterfall chart of the home page, rendered for smartphone

Notice in Figure 1-1 that the total payload transferred is 1.2 MB from 134 HTTP requests. But this is the smartphone version; it should be a smaller payload. And yet it's not, as illustrated in Figure 1-2.

Observe how the total payload for the desktop is 952 KB from 132 HTTP requests. Clearly the smartphone version is loading all of the same content as the desktop version, plus an additional two files. It goes without saying that this is not responsive to the bandwidth concerns of the mobile experience.

This is completely contrary to our intention in creating a mobile page.

1 Waterfall charts are data visualizations that show the HTTP requests, the time it took to load the resources requested, and the payload or file size of each request that make up a web page. A much more in-depth discussion of waterfall charts concepts is presented in Chapter 2.

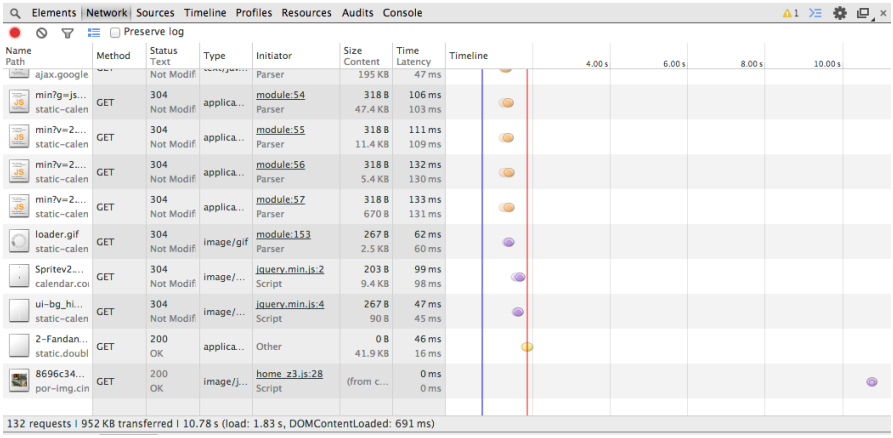

FIGURE 1-2

A waterfall chart of the home page rendered for desktop

And we weren't alone. I opened up a browser on my laptop and consulted HTTPWatch on my iPhone, and I went through the *Alexa.com* top 50 sites to do some competitive analysis. What I found was that 30% of the websites had a larger mobile payload than their desktop equivalent—technology companies, banks, and retailers alike.

Beyond my own research, a number of notable reports also reflected similar results. The Search Agency (a global digital marketing agency) analyzed the top 100 retail sites as well as the Fortune 100 companies' sites and produced the following reports:

- "Multichannel Retailers" (*http://bit.ly/1vqYUPh*)
- "Fortune 100 Companies" (*http://bit.ly/1r1SDlA*)

[TIP]

To access these reports, you will need to give The Search Agency your email address, and it will then send the reports to you.

Among its results is the chart in Figure 1-3, which shows that websites that used (or more accurately, misused) responsive design took an average of 1.91 seconds longer to load than plain, vanilla desktop websites. Most egregious of all, these same websites took 10.74 seconds longer than dedicated mobile sites.

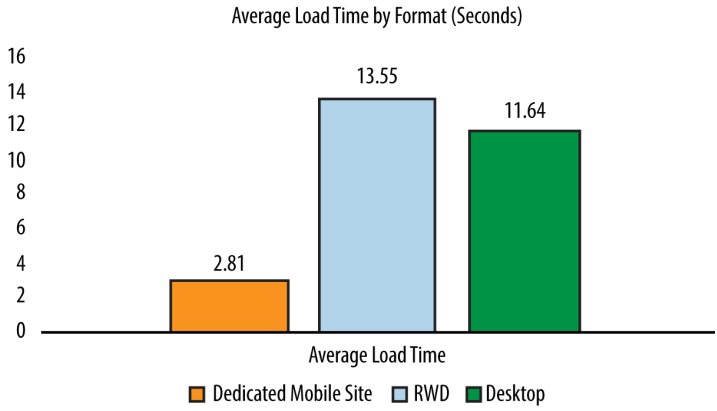

FIGURE 1-3

The Search Agency's comparison of average load times for responsive sites versus dedicated mobile and dedicated desktop sites

Guy Podjarny, CTO at Akamai, also wrote up a piece on his blog detailing his findings from running similar tests. He compared page sizes across a number of resolutions and found little difference between them. You can find his write-up at *http://bit.ly/1tBv6cT.*

Were we all missing the point of creating a responsive experience?

OBSERVATIONS FROM COMPETITIVE ANALYSIS

My own observations from the Alexa list yielded some interesting data, as well. Among other things, I noticed the following:

- Of the top websites for the United States, 47% still used dedicated *mdot sites.*[2] Think about that number for a minute. These are the most trafficked websites on the Internet, arguably the leaders of their respective industries, with members including YouTube, eBay, and Target, and they are foregoing a responsive site in favor of a standalone segmented site.

2 An mdot site is a dedicated website created just for the mobile experience that has its own URL, usually following the convention of using "m" as a subdomain (e.g., *m.comcast.net* or *m.homedepot.com*). There are even more recent derivations of the mdot for tablets, for which "t" is a subdomain for a dedicated tablet experience (e.g., *t.homedepot.com*).

- On average, these dedicated sites were 55% smaller than responsive sites. The mean size of the subset that used mdots was 383 KB, whereas the responsive sites had a mean size of 851 KB (see Figure 1-4). This speaks to a gross discrepancy between intention and implementation.

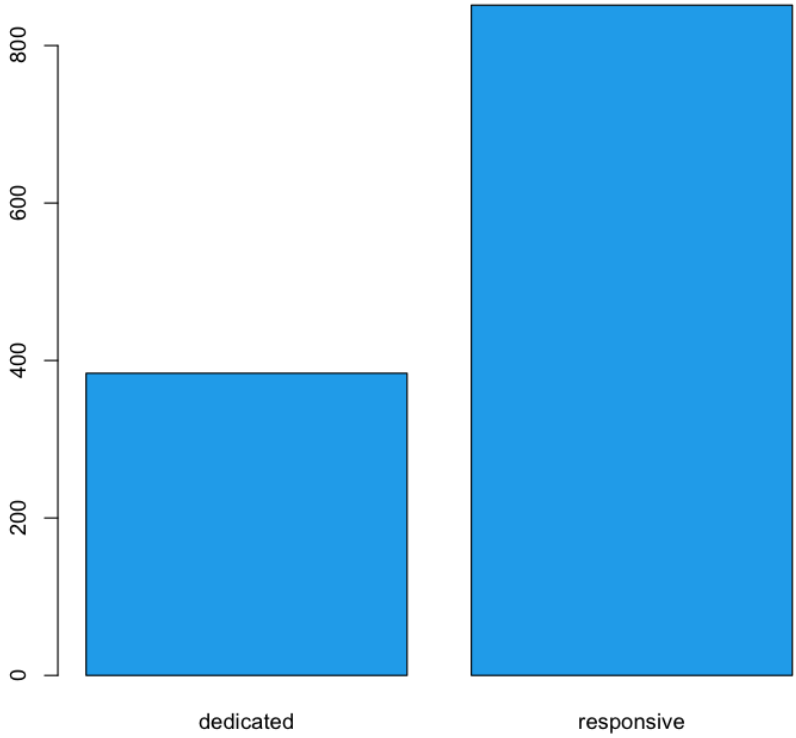

FIGURE 1-4
Mean file size for dedicated versus responsive websites (in KBs)

- The payload of responsive websites has a long-tailed distribution that stretches out into 4 MB, whereas mdot sites are all distributed across ranges less than 1 MB. In fact, mdots are most thickly grouped into the 0 to 200 KB and 200 to 400 KB ranges. I created histograms to look at the distribution of file sizes between mdot sites and responsive sites, which you can see in Figures 1-5 and 1-6.

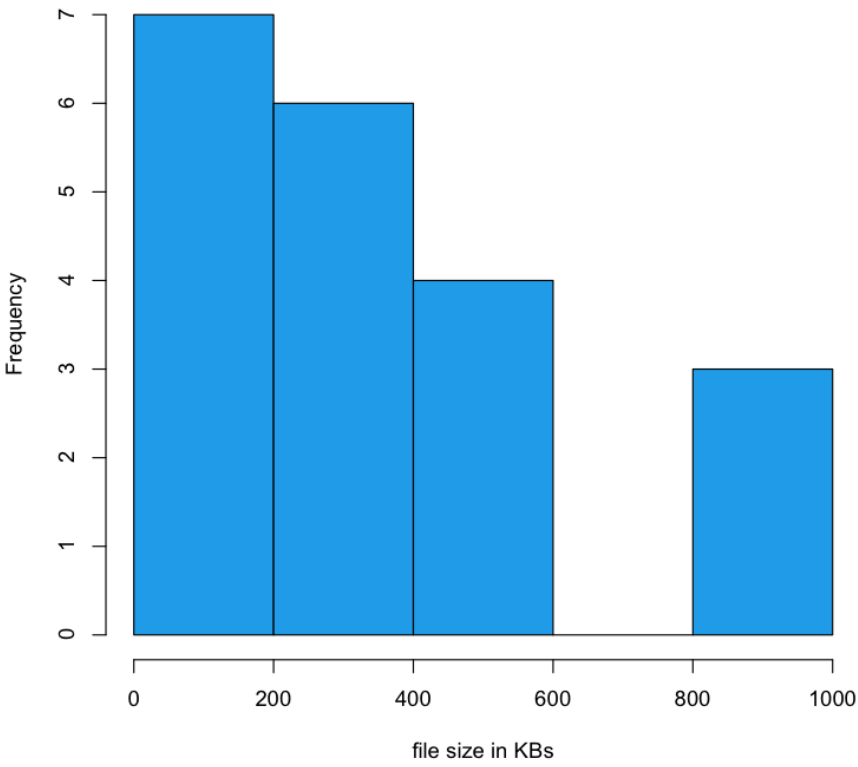

FIGURE 1-5
Distribution of file sizes for dedicated mobile sites (in KBs)

Note the scale of the x-axis in each histogram. The three outliers for the dedicated experiences were up against 1 MB. For the responsive sites, 1 MB is the second largest grouping and the tail keeps going out to 4 MB.

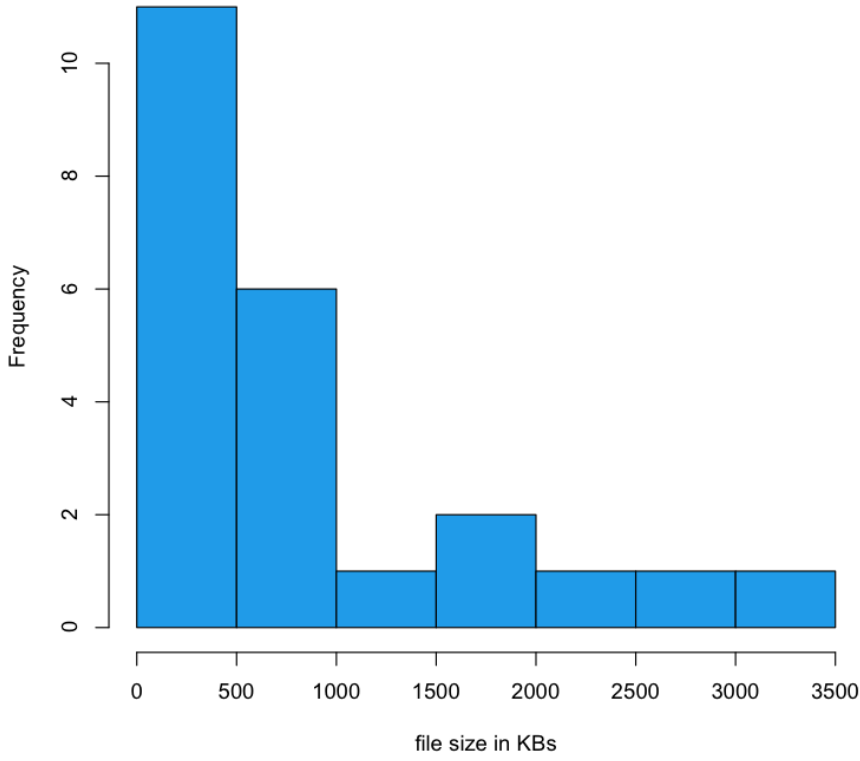

FIGURE 1-6

Distribution of file sizes for responsive sites (in KBs)

- Of the responsive websites, 43 percent had nearly the same or slightly more HTTP requests for their smartphone experiences compared to their desktop experiences. Contrast this to the 1.5 percent of the dedicated sites that had the same or higher HTTP requests for their smartphone experience compared to their desktop experience. Figures 1-7 and 1-8 depict this breakdown.

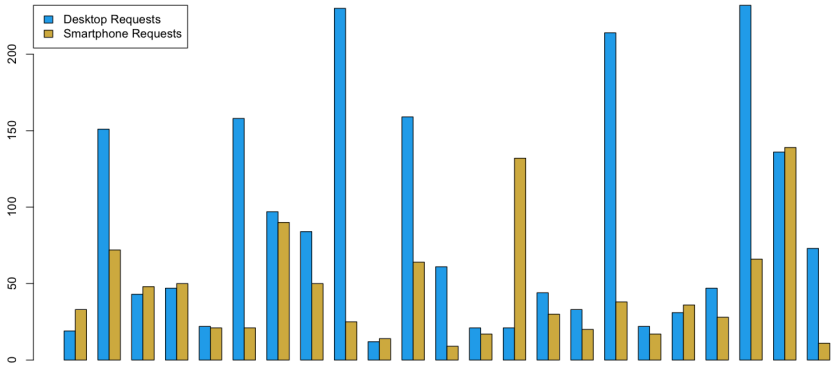

FIGURE 1-7

Grouped bar chart of HTTP requests for desktop and smartphone experiences on responsive sites

In Figure 1-7, notice that in each grouping, the blue bar represents the number of HTTP requests for a page served for the desktop experience, whereas the yellow bars represent the number of HTTP requests served for the same page served to a smartphone.

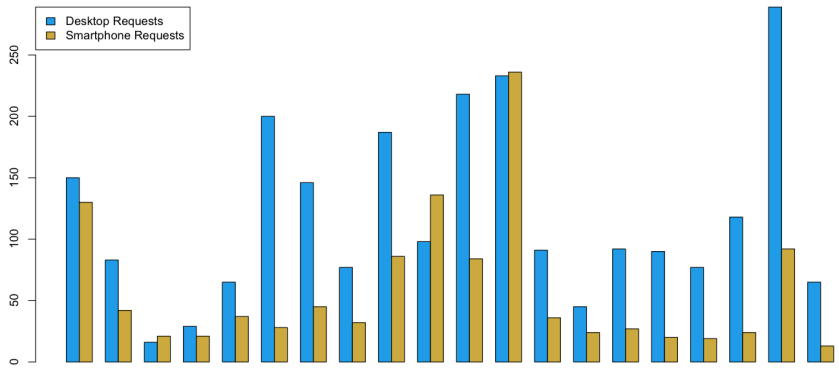

FIGURE 1-8

Grouped bar chart of HTTP requests for desktop and smartphone experiences on dedicated mdot sites

Again, note that for each grouping, the blue bar represents the number of HTTP requests for a page served for the desktop experience; the yellow bars show the number of HTTP requests for a smartphone.