radware

STATE OF THE UNION

Ecommerce Page Speed & Web Performance

Spring 2014

SHARE THIS WHITEPAPER

in f t g+

Table of Contents

Executive Summary

While the 2014 holiday shopping season is months away, online retailers are already poised to begin planning their online strategy, making this report a timely reminder to consider the impact of page growth and complexity on page speed and the user experience.

Looking at the trend over time, the months of September, October, and November have traditionally been a time of significant page growth as site owners race to implement new features, third-party scripts, and ever-richer content before Black Friday. An unfortunate side effect of this page growth has, historically, been incremental page slowdowns – a slow and steady degradation of the user experience.

While this deterioration in retail web performance has been consistent over the past four years, since the inception of our benchmarking research, it is not necessarily inevitable. This report outlines the current state of retail web pages and offers an overview of performance best practices that site owners should consider implementing for their web sites.

Why Page Speed Matters

The speed with which a page renders in a visitor's browser affects every conceivable business metric, including page views, bounce rate, conversions, customer satisfaction, return visits, and of course revenue. These effects are felt at companies of all sizes – from internet giant Yahoo!, which found that making pages just 400 milliseconds faster resulted in a 9% traffic increase[1], to online auto parts retailer AutoAnything, which cut its page load times in half and experienced a 13% increase in sales.[2]

The Purpose of This Research

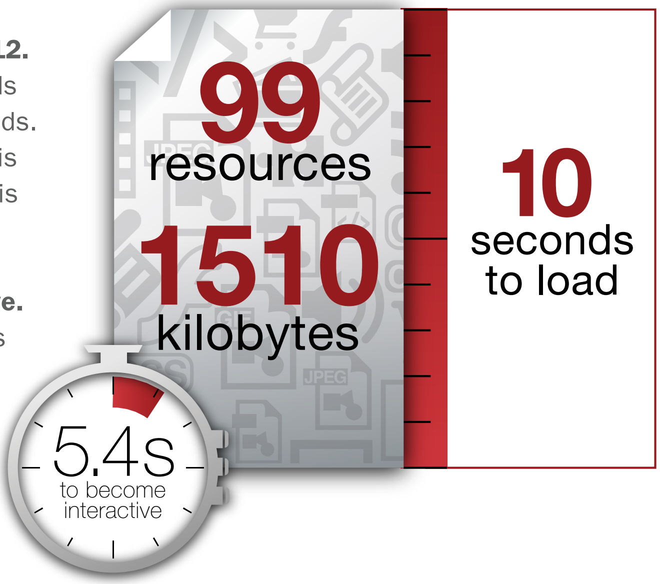
Just a few seconds – and sometimes even fractions of a second – can make the difference between online success and failure, yet it can be difficult for site owners to gain a real-world understanding of their web site's performance. Since 2010, we have measured and analyzed the performance and page composition of the world's top 500 ecommerce sites in order to obtain ongoing visibility into how web sites perform for real users under real browsing conditions.

This report answers the following questions:

- Given the assumption that page speed is an increasingly urgent issue for online retailers, has this urgency translated into faster pages over time?
- How do pages actually render in real-world scenarios?
- Does the use of a content delivery network (CDN) correlate to faster web performance?
- How quickly are retailers acting to adopt performance best practices?

Except where otherwise noted, the results discussed in this report are for pages tested in Chrome 33. At the time of conducting this research, Chrome was the most widely used browser in the United States, with a market share of 32.64%.[3]

Key Findings



1. The median web page has slowed down by 47% since 2012.
The median top 500 ecommerce home page takes 10 seconds to load. In spring 2012, the median page loaded in 6.8 seconds. This represents a 47% slowdown in just two years. Page size is partly to blame: the median page contains 99 resources and is 1510 KB in size.

2. The median page takes 5.4 seconds to become interactive.
“Time to interact” (TTI) refers to how long it takes for a page’s primary content to load and become usable. For the median ecommerce page, visitors must wait 5.4 seconds for the page’s primary content to render and become usable. The majority of online shoppers will abandon a page after waiting 3 seconds for it to load.[4]

3. The top 100 sites are slower than the top 500.
Higher profile sites did not fare better than their lesser-known counterparts. Among the top 100 ecommerce sites, the median load time is 10.7 seconds – 7% slower than the median load time for the top 500 sites.

4. Sites that use a CDN were slower than sites that do not.
Pages that use a CDN took a full second longer to become usable than pages that do not use a CDN. The time to interact for pages that use a CDN was 5.7 seconds, compared to a TTI of 4.7 seconds for pages that do not use a CDN.

5. Most sites fail to leverage best practices for optimizing images.
Despite the fact that images represent one of the single greatest performance challenges (and opportunities), 34% of pages failed to properly implement image compression, and 76% failed to take advantage of progressive image rendering. Each of these techniques has been proven to deliver faster real and perceived performance.

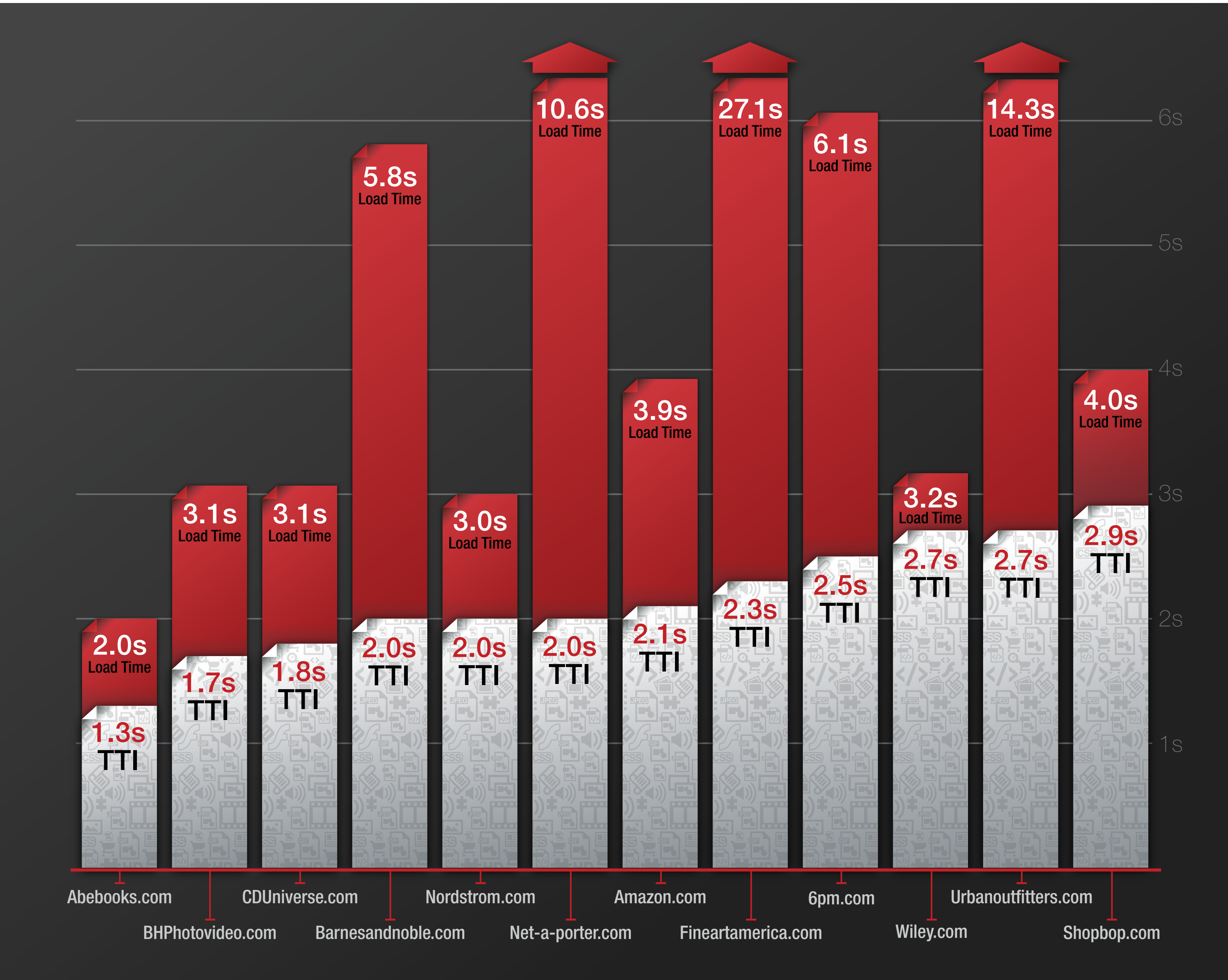
Who Was Fastest?

All times are indicated in seconds.

The 12 Fastest Sites

Among the top 100 sites, these were the fastest in terms of their ability to display meaningful, interactive content (e.g., feature banners with functional call-to-action buttons). This metric is known as “time to interact” (TTI) and is distinct from the better-known “load time” metric. (Load time indicates when all of a page’s resources – from images to third-party-party scripts – have downloaded and rendered.) From a user experience perspective, TTI is considered a more meaningful performance metric than load time, as it indicates when a page begins to be usable.

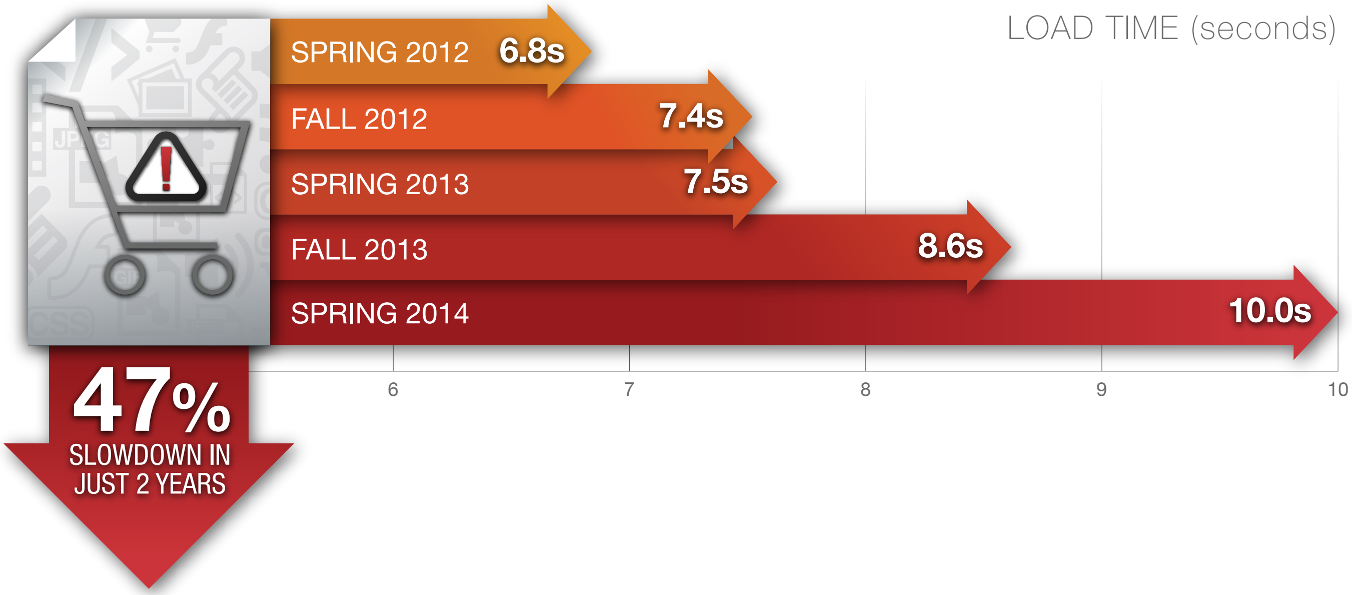
We have provided the time to interact alongside each page’s full load time in order to give perspective into the distinction between the two metrics, and to illustrate that **load time is not always the most meaningful measure of a site’s performance**. For example, while Fineartamerica.com has a load time of 27.1 seconds, it has a time to interact of 2.3 seconds; the TTI indicates that this site delivers a satisfactory user experience.



Finding #1: The Median Web Page Takes 10 Seconds to Load

The median top 500 ecommerce home page takes 10 seconds to load. In spring 2012, the median page loaded in 6.8 seconds. This represents a 47% slowdown in just two years.

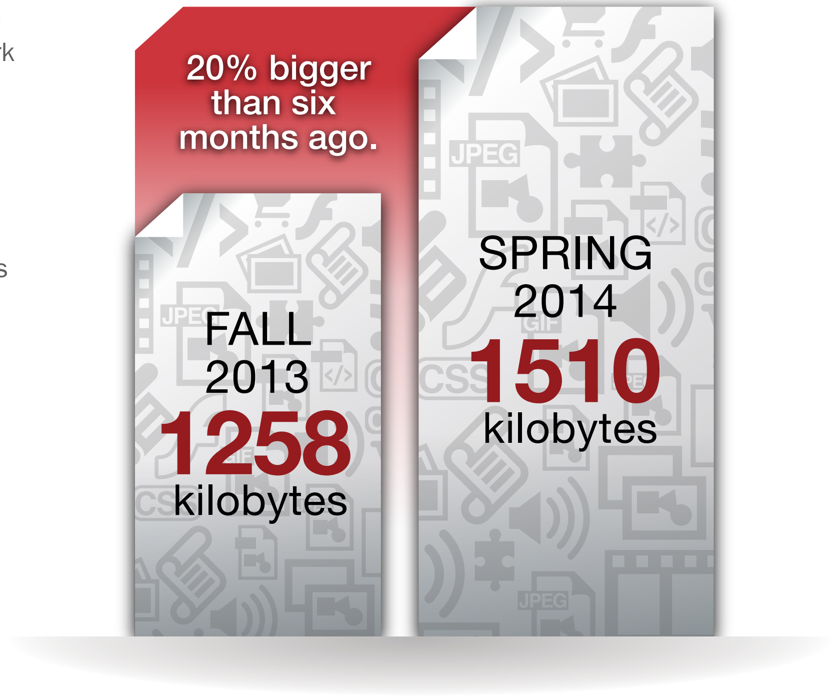
This finding also illustrates that sites are moving further away from the load-time target of 3 seconds or less: 57% of online consumers will abandon a site after waiting 3 seconds for pages to render.[5]



Page Size and How It Affects Load Time

The median page contains 99 resources and is 1510 KB in size. In other words, a typical page is 20% bigger than it was just six months ago.

For most sites, one of the greatest performance drains is the need to complete dozens of network round-trips to retrieve resources such as style sheets, scripts, and images. Each of these resources makes an individual round trip from the user's browser, which requests the file from the host server, which in turn delivers the file to the browser. Each of those requests experiences at least 20-30 milliseconds of latency. Typically, latency is in the 75-140 millisecond range, even for sites that use a content delivery network to cache resources closer to end users. These numbers add up quickly when a typical page contains almost 100 resources.



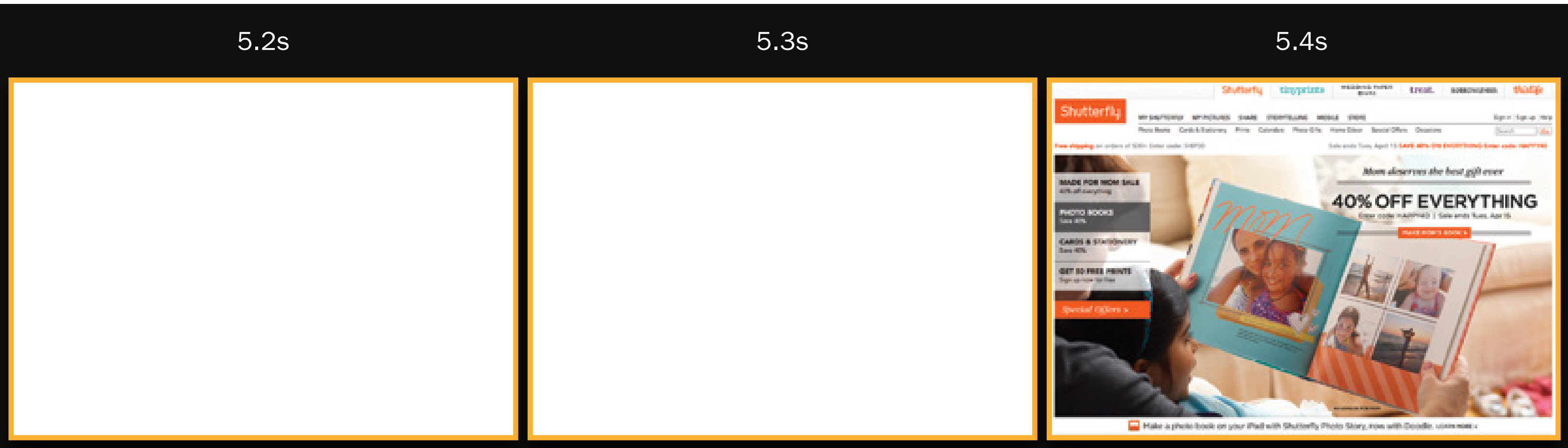
Finding #2: The Median Page Takes 5.4 Seconds to Become Interactive

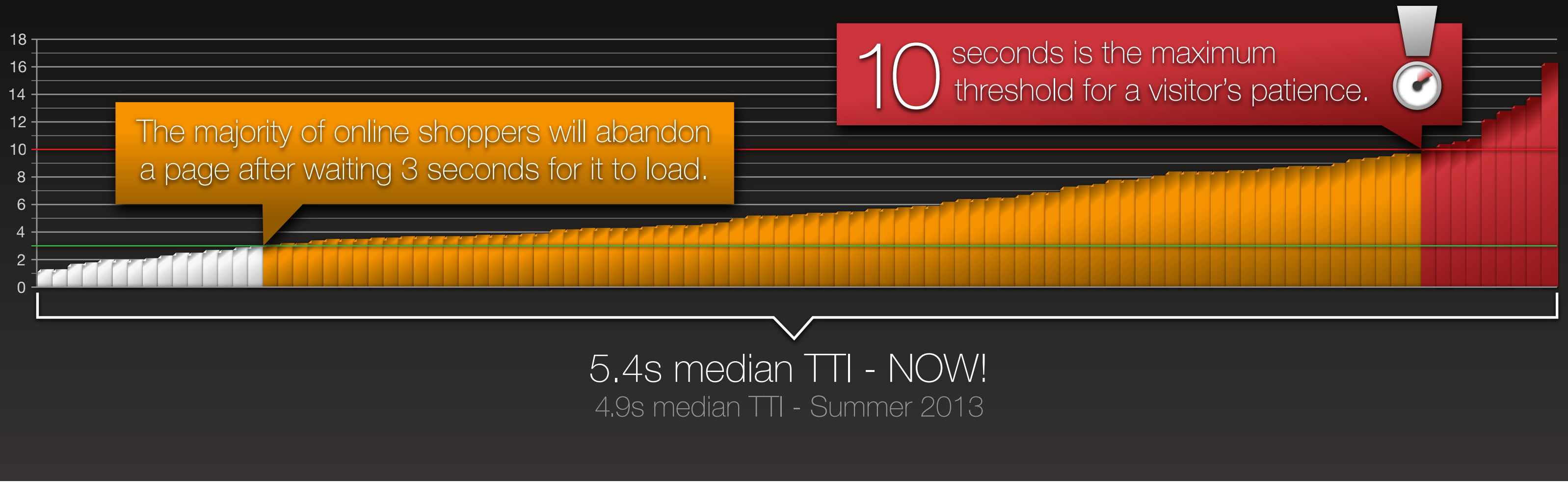
Time to interact (TTI) is the point at which a page displays its primary interactive content (e.g., feature banners with functional call-to-action buttons). For the median ecommerce page, visitors wait 5.4 seconds for the page's primary content to render and become usable.

The majority of online shoppers report that they will abandon a page that takes longer than 3 seconds to load.[6] Ten seconds is the maximum threshold for most visitors' patience.[7]

To identify the time to interact (TTI) for each page, we generated a timed filmstrip view of the median page load for each of the top 100 sites. We found the following:

- The median TTI was 5.4 seconds.
- Only 12% of the top 100 retail sites loaded in under 3 seconds.
- 9% of pages took 10 seconds or longer to become interactive.

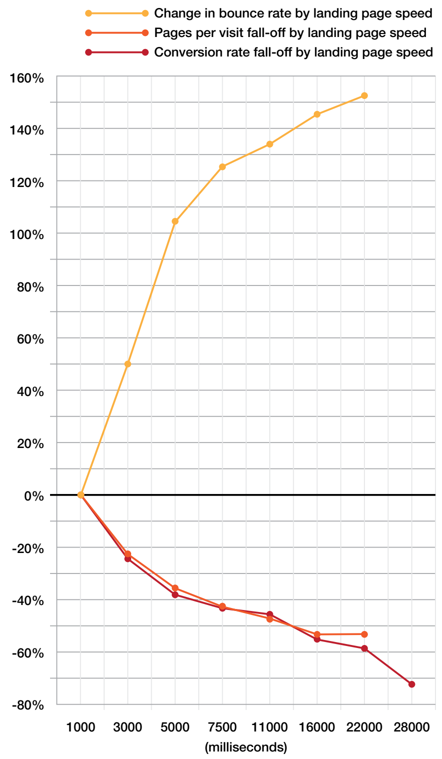




We began calculating time to interact in our Summer 2013 State of the Union. At that time, the median TTI was 4.9 seconds. In other words, the median page takes 10% longer to become interactive today than it did nine months ago.

Why Pages Should Be Interactive in 3 Seconds or Less

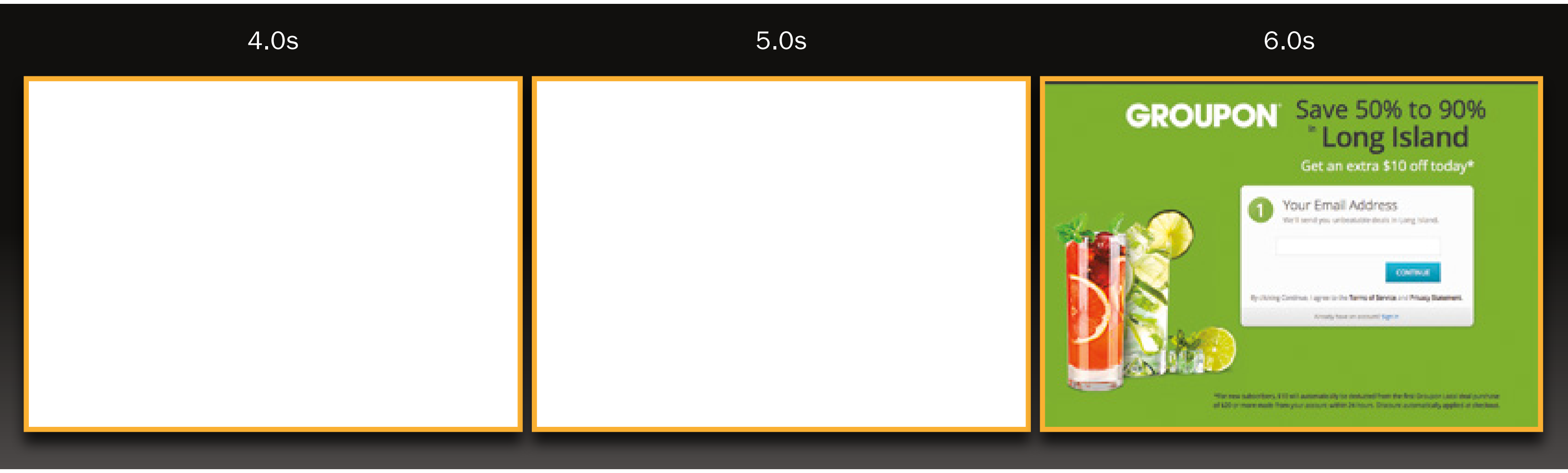
- A site that loads in 3 seconds experiences 22% fewer page views and a 50% higher bounce rate than a site that loads in 1 second. Impact on conversions: -22%.
- A site that loads in 5 seconds experiences 35% fewer page views and a 105% higher bounce rate. Impact on conversions: -38%.
- A site that loads in 10 seconds experiences 46% fewer page views and a 135% higher bounce rate. Impact on conversions: -42%.[8]



4 Common Types of Interactivity Roadblock

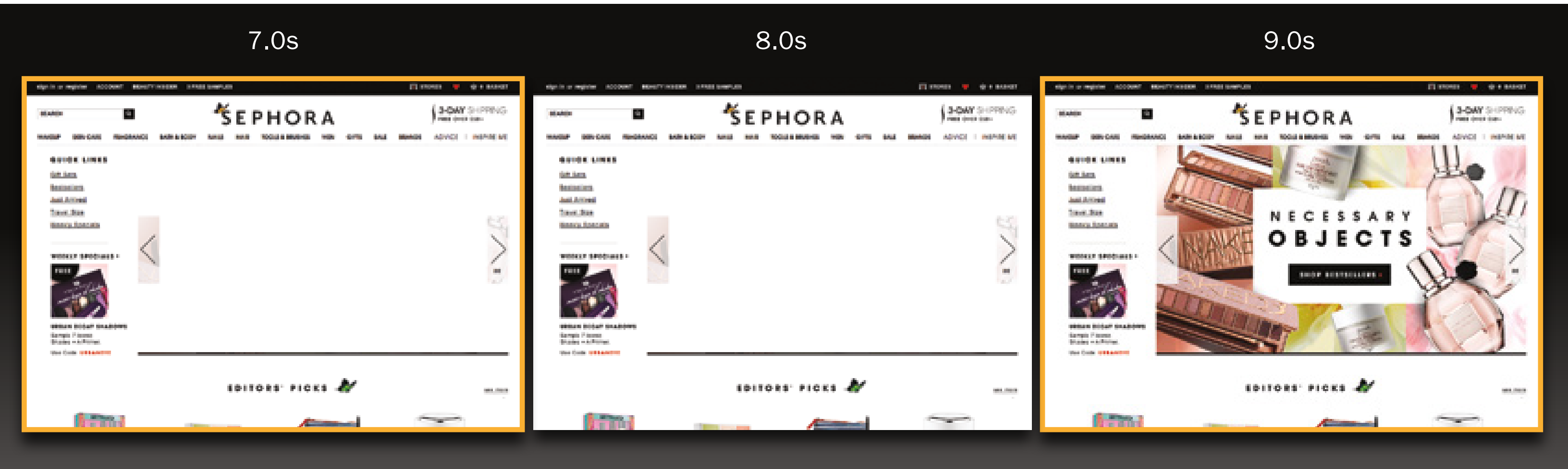
1. Pages that are blank in the browser for several seconds, then suddenly populate.

Site owners should be aware of the usability consequence of delaying the rendering of feature content: **a user who endures an 8-second download delay spends only 1% of their total viewing time looking at the featured space on a page.** In contrast, a user who receives instantaneous page rendering spends 20% of their viewing time within the feature area of a page.[9]

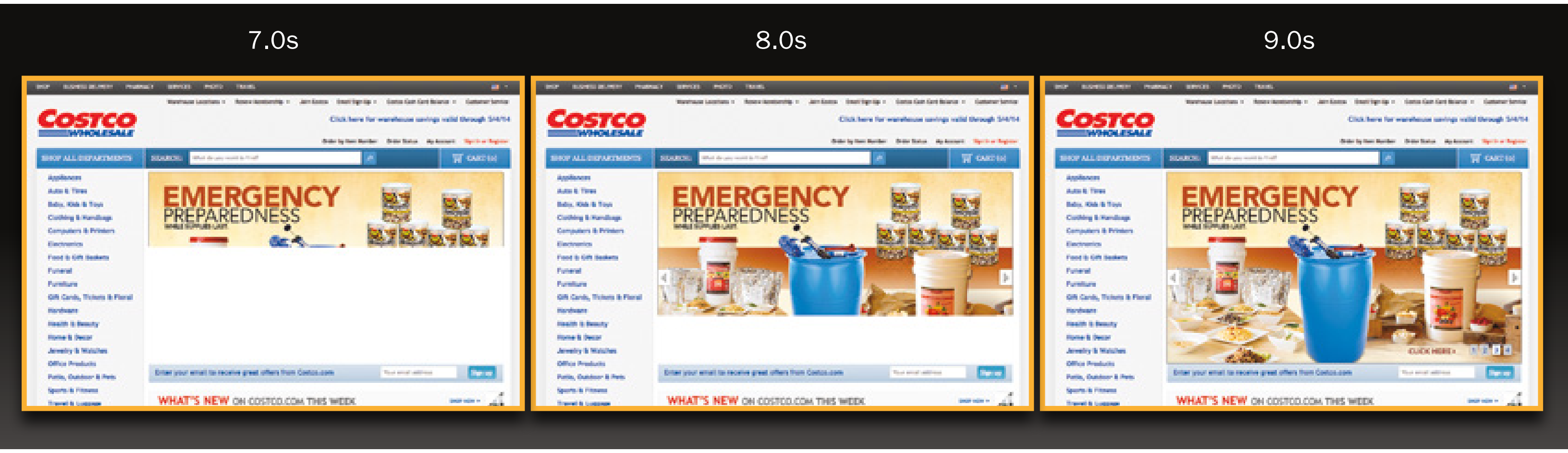


2. Pages in which the navigation elements load early, and the feature content loads last.

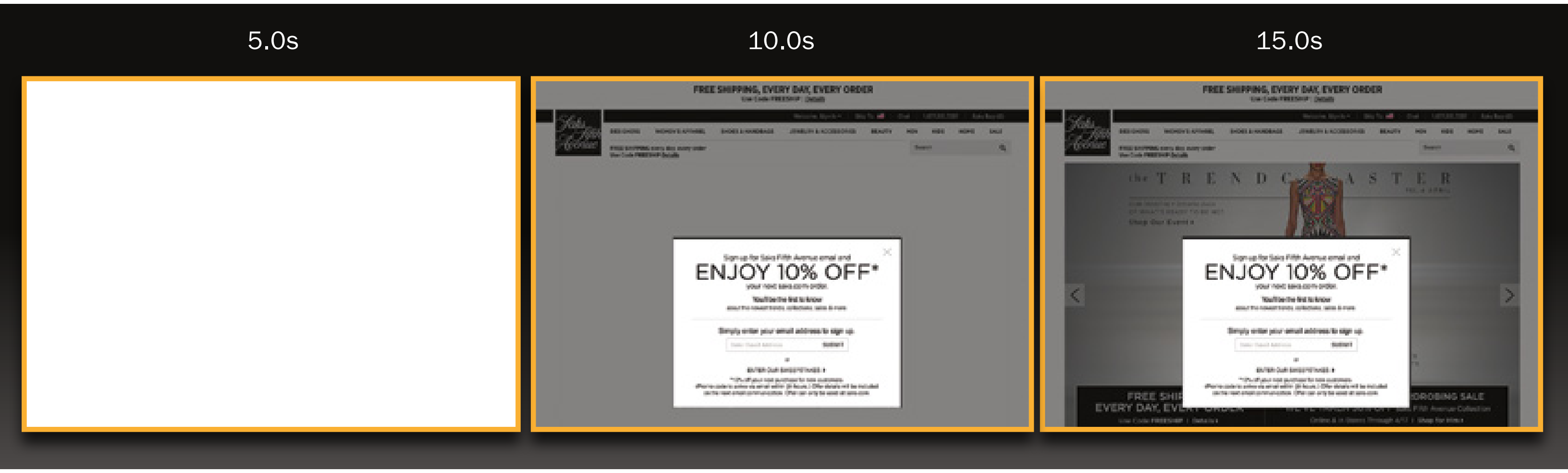
While this usability issue isn't as concerning as the one above, the consequence remains the same: the user will spend only 1% of their viewing time on the page's most important content.



3. Pages that use baseline feature images, so the image loads in chunks, rendering the call to action last.
Placing the call to action (CTA) at the bottom of feature banners is a widely followed design convention, yet it frequently incurs a significant performance penalty, particularly in pages that use baseline images (i.e., images that load line by line or in "chunks"). In many of the pages we studied, it was noted that the CTA – arguably the most critical page element – was often the last visible element to render.

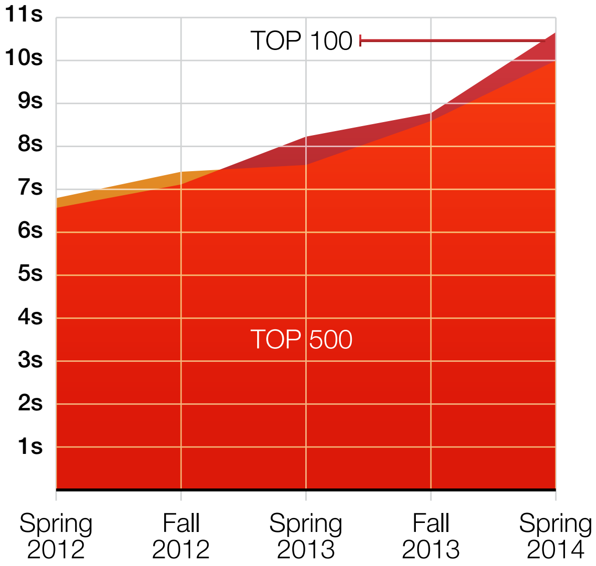


4. Pages in which a popup blocks the main page before it finishes rendering.
This was a recurring usability issue on many sites: within moments of arriving at the home page, users are served with a pop-up. There are some use cases that support on-arrival pop-ups, such as requiring a user to identify their location in order to serve accurate item and shipping costs; however, in many cases the pop-ups noted in our study were for newsletters, surveys, and other opt-in marketing campaigns. Not only do these pop-ups act as a usability irritant, they slow down or block the rendering of the main page content.



Finding #3: The Top 100 Sites were Slower Than the Top 500

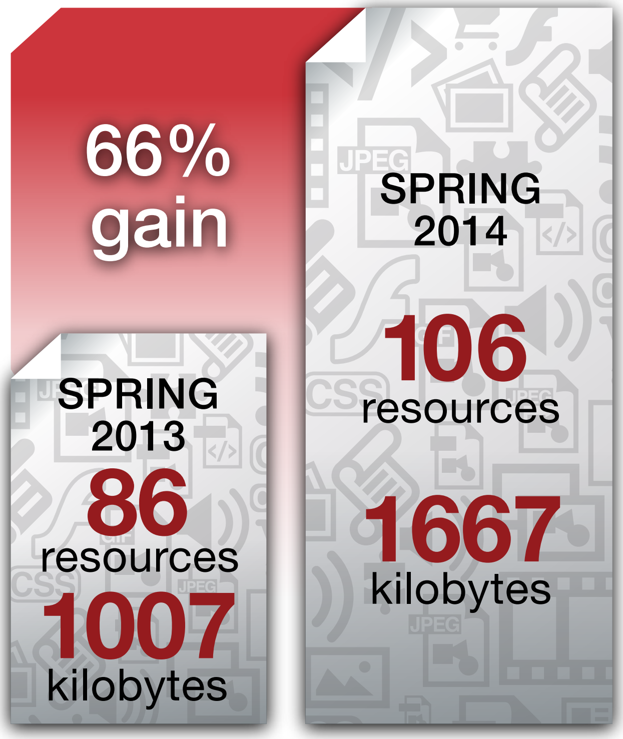
Looking at the top 100 sites, the median site took 10.7 seconds to load, compared to 6.6 seconds in spring 2012. This represents a 62% slowdown in two years.



"Page Bloat" Is Even More Pronounced Among the Top 100 Sites

A year ago, the median number of resource requests for a Top 100 site was 86. That number has grown by 23% to 106 resources.

Not only has the number of resources grown, the weight of those individual resources has also increased dramatically. In early 2013, the median Top 100 page was 1007 KB. Today, the median page carries a payload of 1667 KB – a 66% gain.



Why Are These Findings Significant?

There is a general assumption that leading web sites must have fewer performance-related issues than other sites, due to their larger development and operations budgets. In fact, often the opposite is true. For example:

- **Leading sites are more likely to incorporate large, high-resolution images and other rich content** in their design, thereby increasing the number of resources and total page size.

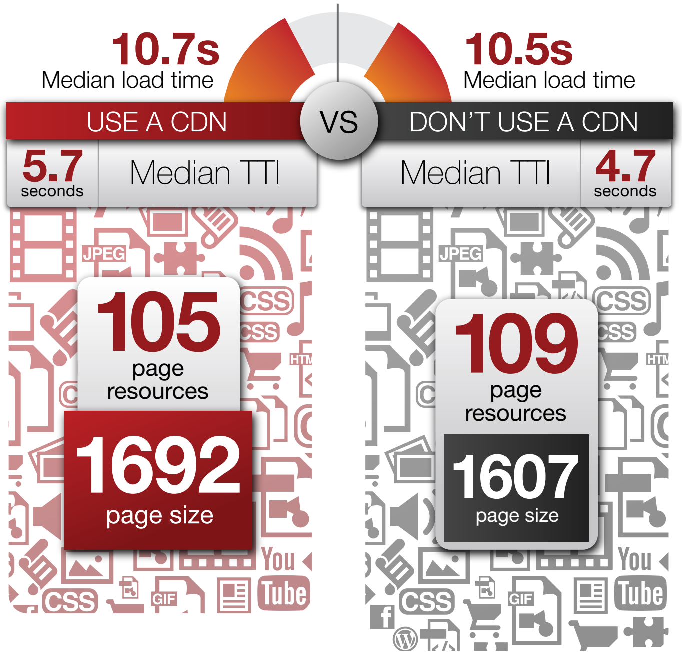
- **Retail leaders also tend to take a more aggressive approach to incorporating third-party marketing tools**, such as trackers and social analytics. Third-party scripts can have a significant impact on performance. Poorly implemented scripts can delay page render, and non-functional scripts can prevent a page from loading.

Finding #4: Sites That Use a CDN Performed Worse Than Sites That Do Not

A content delivery network (CDN) is a web performance solution that addresses the problem of latency: the amount of time it takes for the host server to receive, process, and deliver on a request for a page resource (images, CSS files, etc.). Latency depends largely on how far away the user is from the server, and it is compounded by the number of resources a web page contains. A CDN caches static page resources in distributed servers (AKA edge caches, points of presence, or PoPs) across a region or worldwide, thereby bringing resources closer to users and reducing round trip time.

75% of the top 100 retail sites use a content delivery network (CDN), yet this performance best practice does not correlate to faster pages.

While the load times for pages that do and do not use a CDN were very similar (10.7 seconds versus 10.5 seconds, respectively), the median time to interact (TTI) was not. Pages that use a CDN took a full second longer to become usable than pages that do not use a CDN: TTI for pages that use a CDN was 5.7 seconds, compared to a TTI of 4.7 seconds for pages that do not use a CDN.



A one-second delay can have a significant negative impact on key performance indicators:

- 8.3% increase in bounce rate
- 3.5% decrease in conversions
- 2.1% decrease in cart size
- 9.4% decrease in page views

How to Interpret These Findings

This finding should not be interpreted as a criticism of the efficacy of content delivery networks. Caching page resources closer to end users is a proven performance technique that shortens server round trips and reduces latency.

To understand why using a CDN may not always correlate to faster pages, we must consider the pages themselves.

The first consideration is page size. As the graph in this section indicates, both the pages that use a CDN and those that do not are comparable in terms of page size and total number of resources. Pages that use a CDN tend to use slightly fewer, but fatter, resources, which results in somewhat larger pages overall. But this small difference in size arguably cannot account for the fact that pages that use a CDN become interactive a full second behind those sites that do not use a CDN.

This suggests that the issue is not solely about getting resources to the user faster: it is about how the pages themselves are built. While it was not within the scope of this research to perform a deep analysis of each web site in this study, it is feasible to conjecture that sites that use a CDN may experience the same performance issues that were discussed in the previous section of this report ("Finding #3: The Top 100 Sites Performed Worse Than the Top 500"), including:

- Increased likelihood of incorporating large, high-resolution images and other rich content.
- Increased likelihood of implementing third-party marketing tools, such as trackers and social analytics, which can have a significant impact on performance.

Finding #5: Most Sites Do Not Take Advantage of Opportunities to Optimize Images

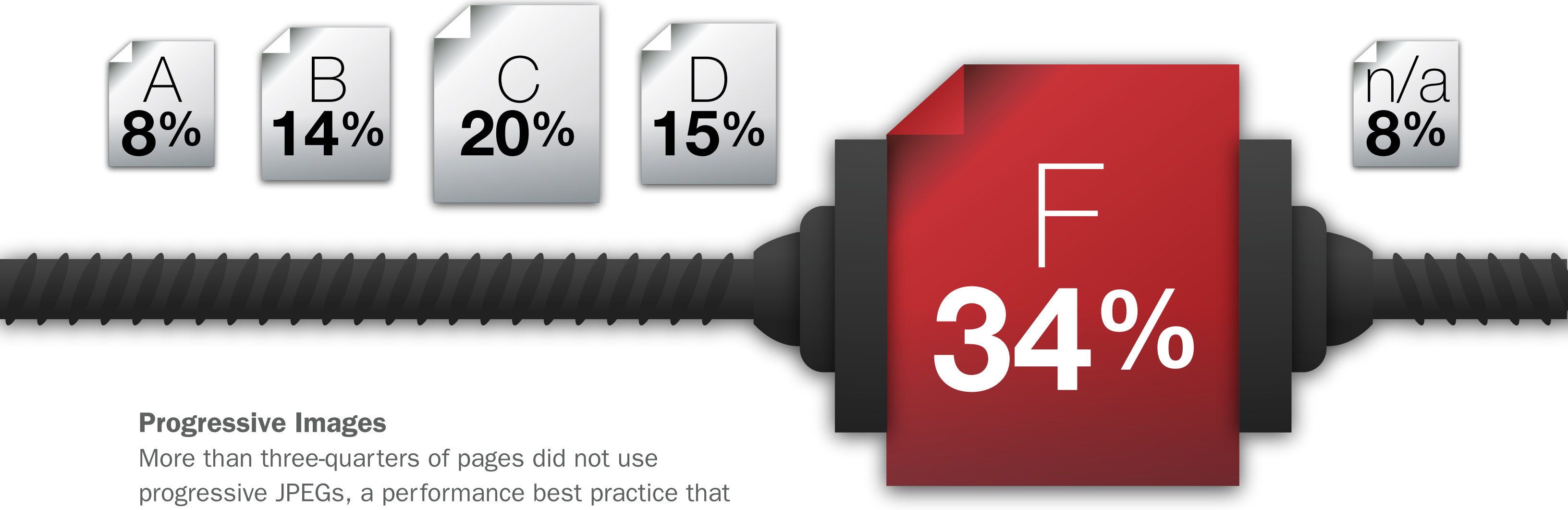
There are a number of reasons why web pages have slowed down, chief of which are images, which account for at least half of a typical page's total size. According to the HTTP Archive, images are responsible for 61% of the page weight for the top 100 sites.[11]) What's worse, many images are not optimized to render quickly in the browser. While there are techniques available for optimizing images for performance, most sites fail to fully exploit them.

Image Compression

Image compression is a performance technique that minimizes the size (in bytes) of a graphics file without degrading the quality of the image to an unacceptable level. Reducing an image's file size has two benefits:

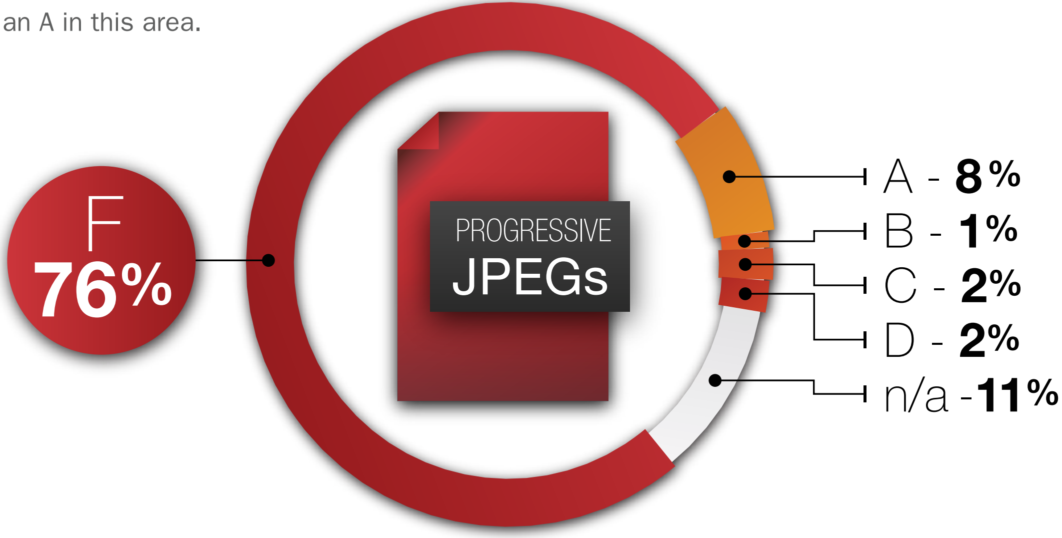
- lessening the amount of time required for images to be sent over the internet or downloaded, and
- increasing the number of images that can be stored in the browser cache, thereby improving page render time on repeat visits to the same page.

Despite the benefits of image compression, we found that 34% of pages failed to implement this technique, while only 8% scored an A.



Progressive Images

More than three-quarters of pages did not use progressive JPEGs, a performance best practice that can improve perceived page speed. In one study of the top 2,000 retail sites, progressive JPEGs improved median load time by 15%.[12] Despite the performance benefits of using progressive images, only 8% of the pages we tested scored an A in this area.



12 Things You Can Do to Cure Your Web Site's Performance Pains

There are a number of best practices site owners can implement in order to improve both the real and perceived user experience for online shoppers.

1. Consolidate JavaScript and CSS

Consolidating JavaScript code and CSS styles into common files that can be shared across multiple pages should be a common practice. This technique simplifies code maintenance and improves the efficiency of client-side caching. In JavaScript files, be sure that the same script isn't downloaded multiple times for one page. Redundant script downloads are especially likely when large teams or multiple teams collaborate on page development.

2. Sprite Images

Spriting is a CSS technique for consolidating images. Sprites are simply multiple images combined into a rectilinear grid in one large image. The page fetches the large image all at once as a single CSS background image and then uses CSS background positioning to display the individual component images as needed on the page. This reduces multiple requests to only one, significantly improving performance.

3. Compress Text and Images

Compression technologies such as gzip reduce payloads at the slight cost of adding processing steps to compress on the server and decompress in the browser. These operations are highly optimized, however, and tests show that the overall effect is a net improvement in performance. Text-based responses, including HTML, XML, JSON (JavaScript Object Notation), JavaScript, and CSS, can all be reduced in size by as much as 70%.

4. Defer Rendering "Below the Fold" Content

Assure that the user sees the page quicker by delaying the loading and rendering of any content that is below the initially visible area, sometimes called "below the fold." To eliminate the need to reflow content after the remainder of the page is loaded, replace images initially with placeholder <img> tags that specify the correct height and width.

5. Ensure That Feature Images Are Optimized to Load Early and Quickly

As discussed earlier in this report ("Finding #2: The Median Page Takes 5.4 Seconds to Become Interactive"), site owners should be aware of the usability consequence of delaying the rendering of feature content: a user who experiences instantaneous page rendering spends 20% of their viewing time within the feature area of a page, whereas a user who endures an 8-second download delay spends only 1% of their total viewing time looking at the featured space on a page.

6. Rethink the Design and Location of Call-to-Action Links in Feature Graphics

While the accepted design convention has been to position CTA buttons at the bottom of feature banners, this convention does not always serve the best interests of users or site owners, as shoppers must wait for the image to fully render before taking their next action on the page. Alternative solutions include repositioning the CTA, or using progressive images, which display the CTA earlier. (More on this practice below.)

7. Reformat Images

Inappropriate image formatting is an extremely common performance culprit. An image that is saved to the wrong format can be several times larger than it would be if saved to the optimal format. Images with unnecessarily high resolution waste bandwidth, processing time, and cache space.

As a general rule of thumb, these are the optimal formats for common image types:

- **Photos** – JPEG, PNG-24
- **Low complexity with transparency** – GIF, PNG-8
- **Line art** – SVG
- **Low complexity (few colors)** – GIF, PNG-8
- **High complexity with transparency** – PNG-24

8. Use Progressive JPEGs

Progressive JPEGs are not a new innovation. They were widely used in the 1990s, but fell out of favor due to performance issues caused by slow connection speeds and crudely rendered JPEGs; watching a progressive image load pixel by pixel was a painful experience. Now that connection speeds have improved and progressive JPEGs are more sophisticated, this technique is feasible again and is returning as a newly heralded performance best practice. In one study of the top 2,000 retail sites, progressive JPEGs improved median load time by 15%.[13] (Note: While all popular browsers will render progressive images, Safari, Mobile Safari, Opera and Internet Explorer 8 render them only as baseline JPEGs, meaning there is no performance benefit.)

9. Minify Code

Minification, which is usually applied to scripts and style sheets, eliminates non-essential characters such as spaces, newline characters, and comments. A correctly minified resource is used on the client without any special processing, and file-size reductions average about 20%. Script and style blocks within HTML pages can also be minified. There are many good libraries available to perform minification, often along with services to combine multiple files into one, which additionally reduces requests.

10. Defer Loading and Executing Non-Essential Scripts

Many script libraries aren't needed until after a page has finished rendering. Downloading and parsing these scripts can safely be deferred until after the onload event. For example, scripts that support interactive user behavior, such as drag and drop, can't possibly be called before the user has even seen the page. The same logic applies to script execution. Defer as much as possible until after onload instead of needlessly holding up the initial rendering of the important visible content on the page.

The script to defer could be your own or, often more importantly, scripts from third parties. Poorly optimized scripts for advertisements, social media widgets, or analytics support can block a page from rendering, sometimes adding precious seconds to load times.

11. Use of AJAX for Progressive Enhancement

AJAX (Asynchronous JavaScript and XML) is a technique for using the XHR (XMLHttpRequest) object to fetch data from a Web server without refreshing the page where the code is running. AJAX enables a page to display updated data in a section of a page without reconstructing the entire page. This is often used to respond to user interaction, but it can also enable your application to load a bare-bones version of a page quickly, and then to fill in more detailed content while the user is already viewing the page.

12. Implement an Automated Web Performance Optimization Solution

While many of the performance techniques outlined above can be performed manually by developers, hand-coding pages for performance is specialized, time-consuming work. It is also a never-ending task, particularly on highly dynamic sites, as both browser requirements and page requirements continue to develop. Automated performance optimization solutions, such as Radware FastView, apply a range of performance techniques that deliver faster pages consistently and reliably across the entire site.

Takeaways

1. Ecommerce sites continue to experience performance degradation, due in part to the trend toward increasingly large, complex pages.

As site owners rise to the challenge of making their sites more attractive to users by offering richer, more dynamic content, they ignore performance at their peril. Page speed is a critical usability feature that can make the difference between a shopper who stays on the page and a shopper who abandons the site in favor of a faster competitor.

2. Leading retailers are not immune to this degradation.

Perhaps due to the fact that they are more inclined to implement rich content and third-party scripts, top ecommerce sites experience poorer performance than the rest of the pack.

3. Many sites commit the same four performance/usability mistakes.

These mistakes include rendering the most critical page content last and serving non-essential popups that block primary content before it renders. Focusing solely on metrics such as start render and load time will not always reveal these issues. To better understand how visitors see a site – and to identify usability issues that might otherwise be missed – it is crucial to scrutinize how pages perform frame by frame, and over a real-world connection.

4. A content delivery network (CDN) will not cure every performance pain.

A CDN is an effective tool for solving latency issues: shortening the amount of time it takes for the host server to receive, process, and deliver on a request for a page resource (images, CSS files, etc.). However, for ecommerce sites, two of the biggest performance issues are third-party content and server-side processing, which a CDN does not help with. To get the best acceleration results, ecommerce site owners should use a combination of solutions: CDN, front-end optimization (FEO), application delivery controller (ADC), and in-house engineering.

5. Images represent a wealth of untapped optimization opportunities.

While images comprise more than half of a typical page's total payload, most site owners are not currently fully leveraging image optimization techniques to decrease this payload and speed up page rendering.

Methodology

The tests in this study were conducted using an online tool called WebPagetest – an open-source project primarily developed and supported by Google – which simulates page load times from a real user's perspective using real browsers. Radware tested the home page of every site in the Alexa Retail 500 nine consecutive times. The system automatically clears the cache between tests. The median test result for each home page was recorded and used in our calculations.

The tests were conducted on March 24, 2014, via the WebPagetest.org server in Dulles, VA, using Chrome 33 on a DSL connection.

In very few cases, WebPagetest rendered a blank page or an error in which none of the page rendered. These instances were represented as null in the test appendix. Also, in very few cases, WebPagetest.org rendered a page in more then 60 seconds (the default timeout for webpagetest.org). In these cases, 60 seconds was used for the result instead of null.

To identify the time to interact (TTI) for each page, we generated a timed filmstrip view of the median page load for each site in the Alexa Retail 100. Time to interact is defined as the moment that the featured page content and primary call-to-action button or menu was rendered in the frame.

About Radware

Radware (NASDAQ: RDWR), is a global leader of application delivery and application security solutions for virtual and cloud data centers. Its award-winning solutions portfolio delivers full resilience for business-critical applications, maximum IT efficiency, and complete business agility. Radware's solutions empower more than 10,000 enterprise and carrier customers worldwide to adapt to market challenges quickly, maintain business continuity, and achieve maximum productivity while keeping costs down. For more information, please visit www.radware.com.

Sources

[1] YSlow, Stoyan Stefanov, Velocity (2008)
[2] AutoAnything Cuts Page Load Time in Half and Revs Up Sales by 13%
[3] StatCounter Global Browser Stats, March 2014
[4] PhoCusWright, Consumer Response to Travel Site Performance, June 2010
[5] Consumer Response to Travel Site Performance
[6] Consumer Response to Travel Site Performance
[7] Jakob Nielsen, Website Response Times, June 2010
[8] Strangeloop Networks, The 90-Minute Optimization Life Cycle, June 2010
[9] Website Response Times
[10] Strangeloop Networks, The impact of HTML delay on mobile business metrics, November 2011
[11] HTTP Archive, March 1, 2014
[12] Patrick Meenan, Progressive JPEGs FTW!, June 2013
[13] Progressive JPEGs FTW!