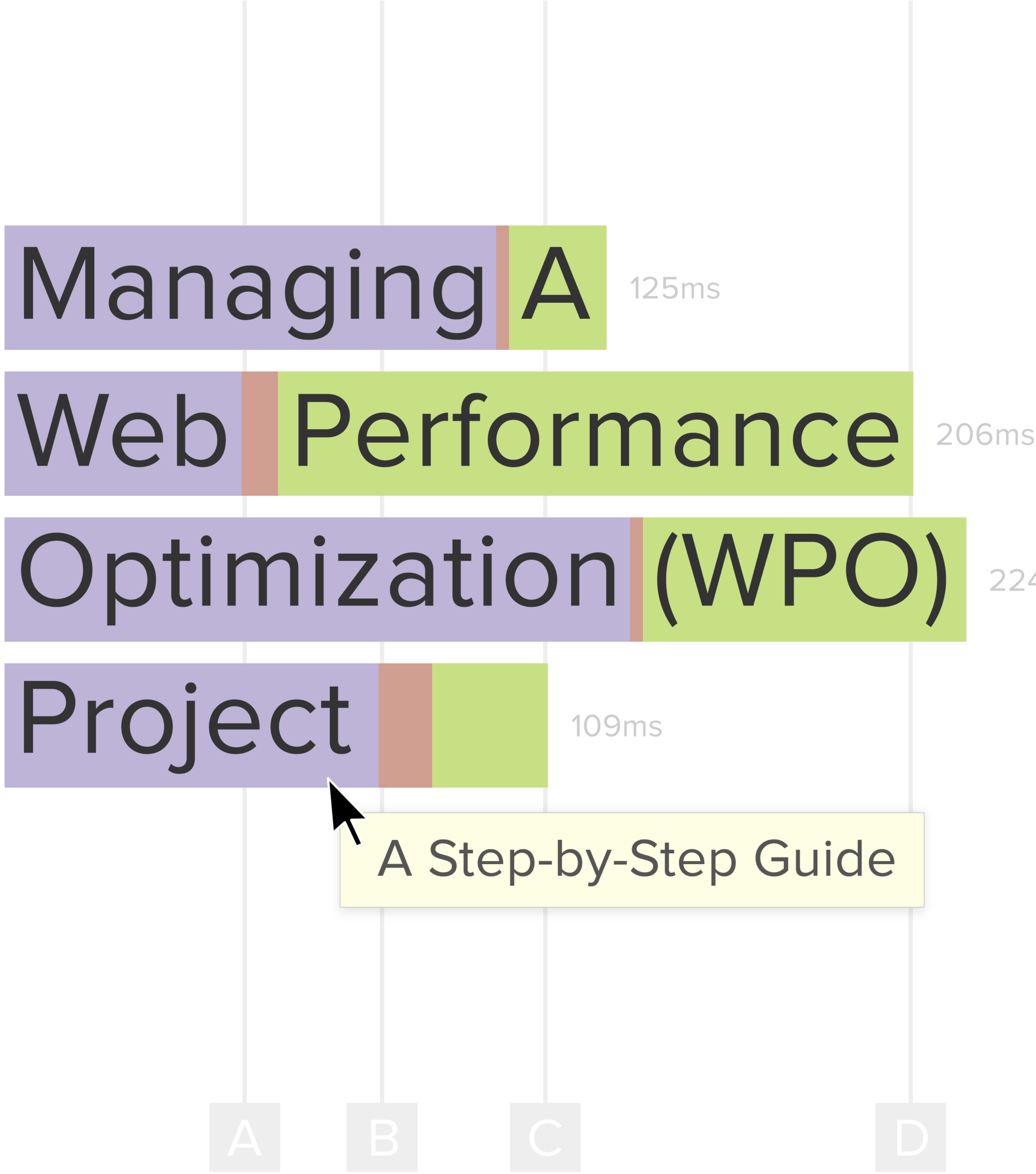


Managing A Web Performance Optimization (WPO) Project

A Step-by-Step Guide

yottaa

Intro: The Process

Web Performance Optimization (WPO) is the science of improving websites. This means making websites lighter, more scalable, and faster across key variables like browser, device, and geographic location.

Doing WPO "right" requires a thoughtful approach. It's easy to waste time with techniques that are supposed to speed up your pages, but may not have the intended effect. The only way of knowing whether you're using your time and resources efficiently is to follow a project outline that involves testing, planning, and continuous monitoring.

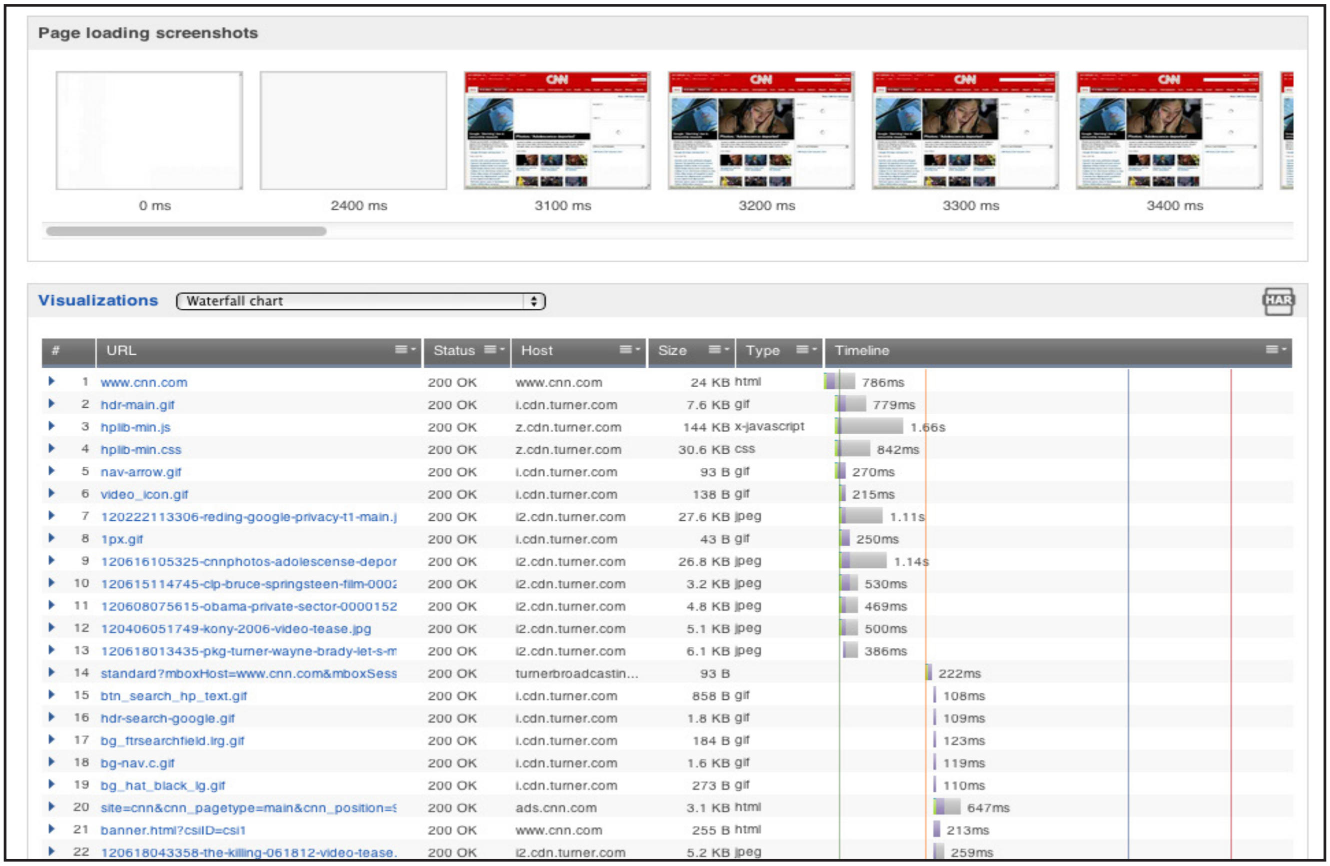
This eBook is a top-level guide to creating your own plan for a WPO project. The principles in this book can be applied to a full-scale website overhaul, but also to smaller projects such as "band-aid" fixes for poor performance.

WPO Tip: This book is an overview. For greater detail on each step of the process, check out our eBooks 11 Techniques to Make Your Website Rock and 17 Metrics You Should Care About. Buttress these resources with books like *High Speed Websites* and *Even Faster Websites* by Steve Souders. For a complete set of resources, check the last page of this eBook.

Table of Contents

1. Establish A Baseline

To succeed in WPO it's important to find out how your site performs around the world, on different browsers, and with different last-mile connectivities. In this chapter, we'll explain how to create a plan for thorough baseline testing.



#	URL	Status	Host	Size	Type	Timeline
1	www.cnn.com	200 OK	www.cnn.com	24 KB	html	786ms
2	hdr-main.gif	200 OK	i.cdn.turner.com	7.6 KB	gif	779ms
3	hplib-min.js	200 OK	z.cdn.turner.com	144 KB	x-javascript	1.66s
4	hplib-min.css	200 OK	z.cdn.turner.com	30.6 KB	css	842ms
5	nav-arrow.gif	200 OK	i.cdn.turner.com	93 B	gif	270ms
6	video_icon.gif	200 OK	i.cdn.turner.com	138 B	gif	215ms
7	120222113306-reding-google-privacy-t1-main.j	200 OK	i2.cdn.turner.com	27.6 KB	jpeg	1.11s
8	1px.gif	200 OK	i.cdn.turner.com	43 B	gif	250ms
9	120616105325-cnnphotos-adolescense-depor	200 OK	i2.cdn.turner.com	26.8 KB	jpeg	1.14s
10	120615114745-clp-bruce-springsteen-film-0002	200 OK	i2.cdn.turner.com	3.2 KB	jpeg	530ms
11	120608075615-obama-private-sector-0000152	200 OK	i2.cdn.turner.com	4.8 KB	jpeg	469ms
12	120406051749-kony-2006-video-tease.jpg	200 OK	i2.cdn.turner.com	5.1 KB	jpeg	500ms
13	120618013435-pkg-turner-wayne-brady-let-s-m	200 OK	i2.cdn.turner.com	6.1 KB	jpeg	386ms
14	standard?mboxHost=www.cnn.com&mboxSess	200 OK	turnerbroadcastin...	93 B		222ms
15	btn_search_hp_text.gif	200 OK	i.cdn.turner.com	858 B	gif	108ms
16	hdr-search-google.gif	200 OK	i.cdn.turner.com	1.8 KB	gif	109ms
17	bg_ftrsearchfield.lrg.gif	200 OK	i.cdn.turner.com	184 B	gif	123ms
18	bg-nav.c.gif	200 OK	i.cdn.turner.com	1.6 KB	gif	119ms
19	bg_hat_black_lg.gif	200 OK	i.cdn.turner.com	273 B	gif	110ms
20	site=cnn&cnn_pagetype=main&cnn_position=!	200 OK	ads.cnn.com	3.1 KB	html	647ms
21	banner.html?csiID=csi1	200 OK	www.cnn.com	255 B	html	213ms
22	120618043358-the-killing-061812-video-tease.	200 OK	i2.cdn.turner.com	5.2 KB	jpeg	259ms

An excerpt from the results page of Websitetest.com, showing the waterfall chart and page load screenshots

1a. Choose a tool

You will be revisiting the testing tool you choose throughout the WPO process. Dozens of free testing tools are available on the web. These are a few of our favorites:

- Websitetest.com
- Webpagetest.org
- YSlow
- Mobitest (for mobile)

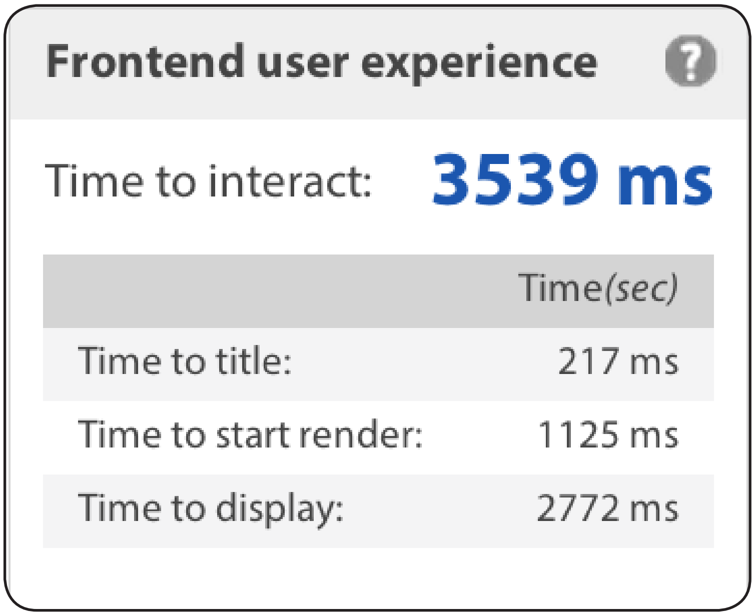
WPO Tip: Check out this page on WPO guru Steve Souders' site for a more complete list of available tools.

1b. Become Familiar with Performance Metrics

Performance metrics drive the WPO process. Conveniently, WPO metrics largely fall into four categories. The primary category, "Front end user experience," provides a direct picture of how your site performs; the secondary categories answer how and why the front-end metrics are they way they are.

Primary Performance Metrics:

"Front-end user experience" - These include the four metrics shown below (excerpted from WebsiteTest.com results). The front end focuses on how fast a given visitor's browser can render your website content. The front end has taken on a new level of importance as websites have become more complex. It is commonly said that **80-90% of a page's total load time now occurs on the front end.**



Frontend user experience

Time to interact: **3539 ms**

	Time*(sec)*
Time to title:	217 ms
Time to start render:	1125 ms
Time to display:	2772 ms

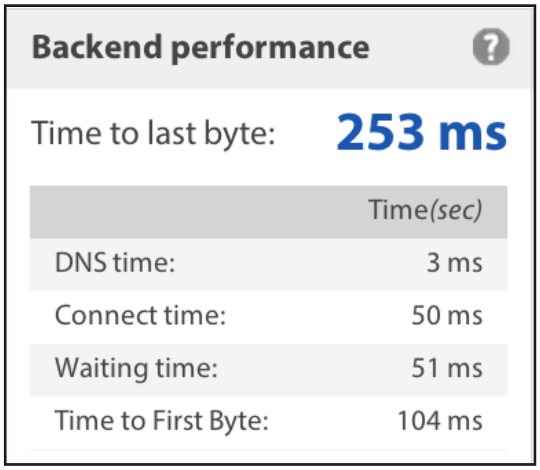
An important note on front-end metrics: visitors can actually see these stages as they happen. This means that the user's perception of your site's performance is informed not only by how long it takes the page to load overall, but how fast each stage is completed. (Staring at a blank page is annoying, even if the page ends up loading in a reasonable time).

In conclusion: be cognizant of each stage, not just the Time To Interact!

1b. Become Familiar with Performance Metrics (cont’d)

Secondary Performance Metrics:

(Example data is from results of a test of CNN.com on Websitetest.com)



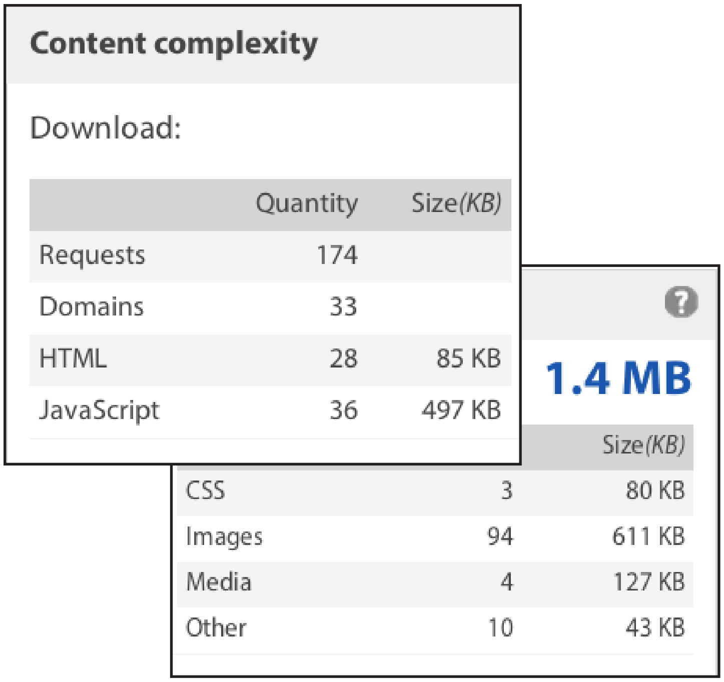
Backend performance

Time to last byte: **253 ms**

	Time*(sec)*
DNS time:	3 ms
Connect time:	50 ms
Waiting time:	51 ms
Time to First Byte:	104 ms

“Backend performance” – These metrics show your website’s data traveling from your server to a visitor’s browser. Performance in this category directly affects the metrics in the front end, since the entire front-end process is on hold until your website’s assets arrive at the browser.

“Content complexity” – These metrics quantify the content of your website, including how many assets you have, what types of assets those are (CSS, HTML, images, etc.), and how much of your site’s weight is comprised by each type of asset. Website content is a major factor in front-end performance.

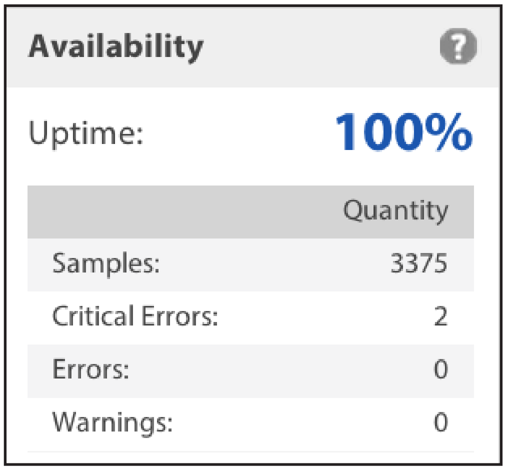


Content complexity

Download:

1.4 MB

	Quantity	Size*(KB)*
Requests	174	
Domains	33	
HTML	28	85 KB
JavaScript	36	497 KB
CSS	3	80 KB
Images	94	611 KB
Media	4	127 KB
Other	10	43 KB



Availability

Uptime: **100%**

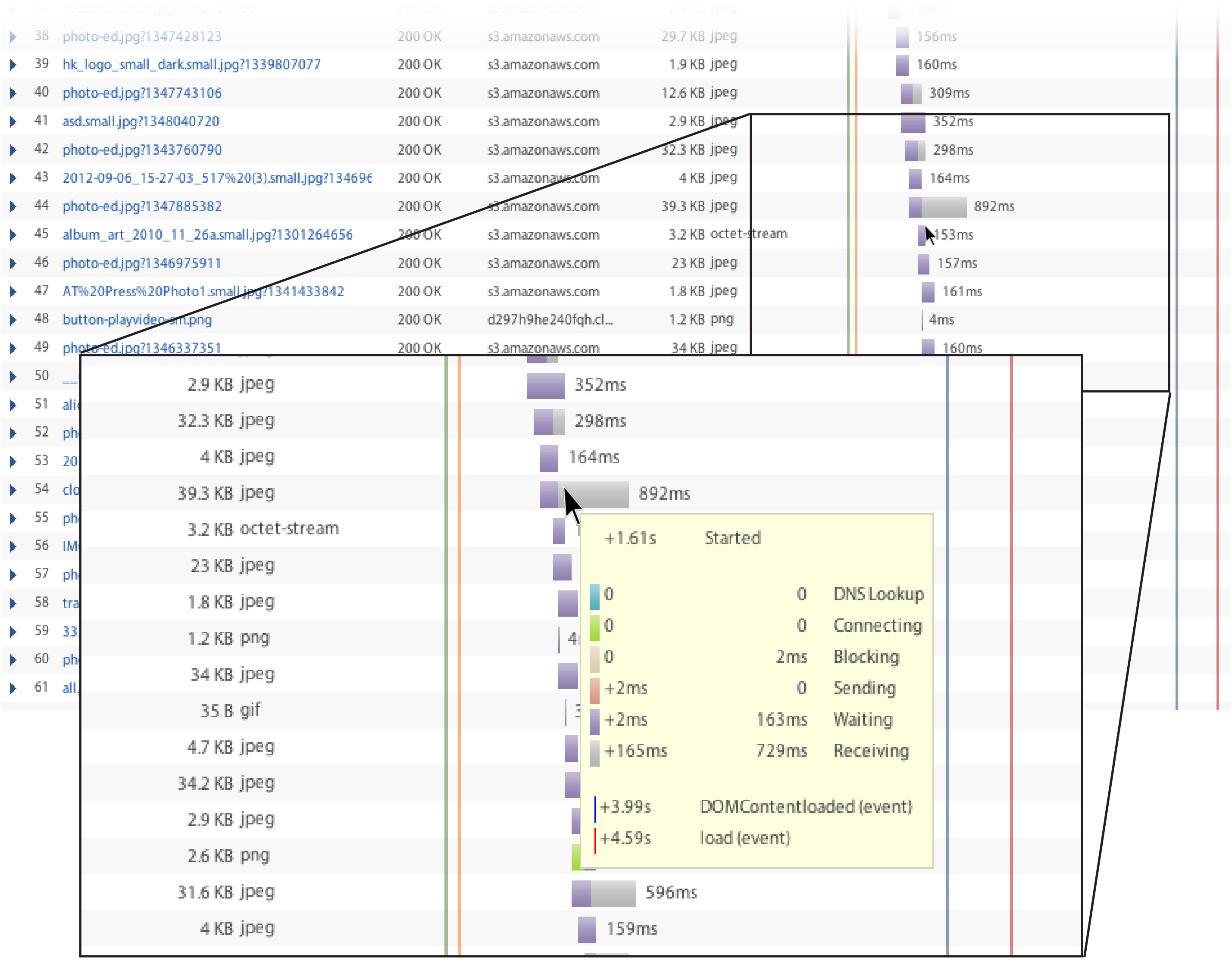
	Quantity
Samples:	3375
Critical Errors:	2
Errors:	0
Warnings:	0

“Availability” -- These metrics show how often your site is up and available to visitors. Since it requires constant collection over time, this information is less important for preliminary performance testing, but it will become important later when you are doing continuous site performance monitoring. Problems in this area indicate issues in the backend of your site.

1c. Become familiar with Performance Visualization

Web performance metrics offer generalities, but it's often necessary to use visualizations to dig deeper into how your site's load process appears to users in different conditions.

The **waterfall chart** is an indispensible tool for WPO. It displays basic data on all the assets of a web page visually, including the order they were downloaded by the browser, their weight, and how long they took to download. Using the waterfall chart is how you make the jump from numerical averages to actionable insights!



37	headshot.small.jpg?1342125100	200 OK	s3.amazonaws.com	2.4 KB jpeg	160ms
38	photo-ed.jpg?1347428123	200 OK	s3.amazonaws.com	29.7 KB jpeg	156ms
39	hk_logo_small_dark.small.jpg?1339807077	200 OK	s3.amazonaws.com	1.9 KB jpeg	160ms
40	photo-ed.jpg?1347743106	200 OK	s3.amazonaws.com	12.6 KB jpeg	309ms
41	asd.small.jpg?1348040720	200 OK	s3.amazonaws.com	2.9 KB jpeg	352ms
42	photo-ed.jpg?1343760790	200 OK	s3.amazonaws.com	32.3 KB jpeg	298ms
43	2012-09-06_15-27-03_517%20(3).small.jpg?13469€	200 OK	s3.amazonaws.com	4 KB jpeg	164ms
44	photo-ed.jpg?1347885382	200 OK	s3.amazonaws.com	39.3 KB jpeg	892ms
45	album_art_2010_11_26a.small.jpg?1301264656	200 OK	s3.amazonaws.com	3.2 KB octet-stream	153ms
46	photo-ed.jpg?1346975911	200 OK	s3.amazonaws.com	23 KB jpeg	157ms
47	AT%20Press%20Photo1.small.jpg?1341433842	200 OK	s3.amazonaws.com	1.8 KB jpeg	161ms
48	button-playvideo-sm.png	200 OK	d297h9he240fqh.cl...	1.2 KB png	4ms
49	photo-ed.jpg?1346337351	200 OK	s3.amazonaws.com	34 KB jpeg	160ms
50	__				
51	ali				
52	ph				
53	20				
54	clo				
55	ph				
56	IM				
57	ph				
58	tra				
59	33				
60	ph				
61	all				

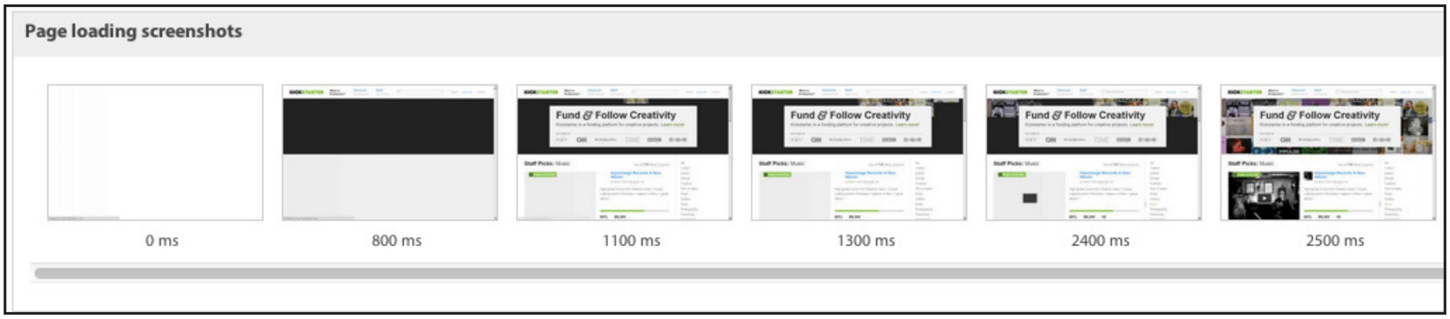
2.9 KB jpeg	352ms
32.3 KB jpeg	298ms
4 KB jpeg	164ms
39.3 KB jpeg	892ms
3.2 KB octet-stream	
23 KB jpeg	
1.8 KB jpeg	
1.2 KB png	4
34 KB jpeg	
35 B gif	3
4.7 KB jpeg	
34.2 KB jpeg	
2.9 KB jpeg	
2.6 KB png	
31.6 KB jpeg	596ms
4 KB jpeg	159ms

+1.61s		Started
0	0	DNS Lookup
0	0	Connecting
0	2ms	Blocking
+2ms	0	Sending
+2ms	163ms	Waiting
+165ms	729ms	Receiving
+3.99s		DOMContentloaded (event)
+4.59s		load (event)

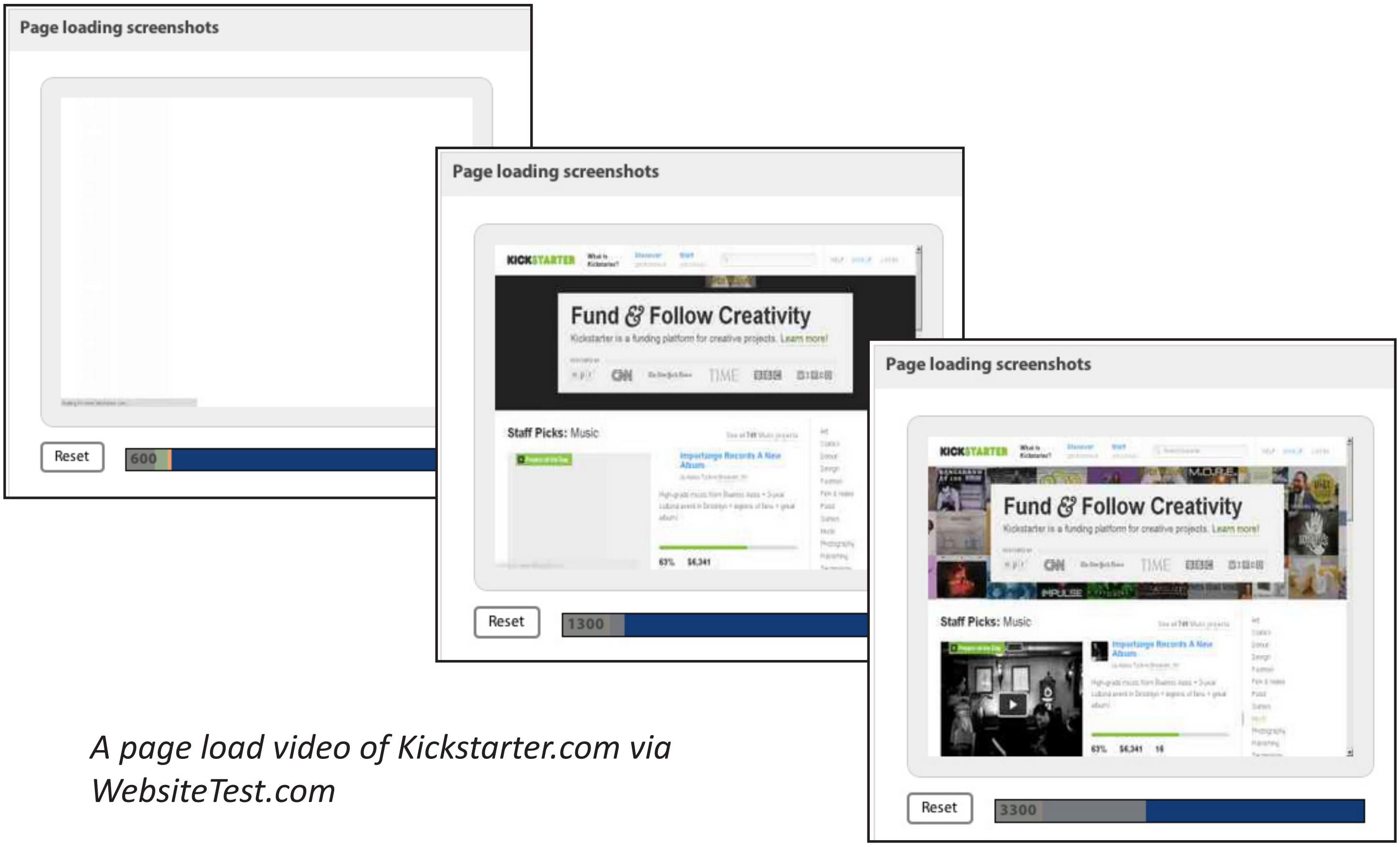
WPO Tip: The waterfall chart on WebsiteTest.com offers options for filtering and sorting these assets in order to easily isolate and identify trends or problematic assets.

1c. Become familiar with Performance Visualizations (cont'd)

After looking at all those metrics, sometimes you need a reality check. That's where screenshots and page load videos come in. **Screenshots and video give you a sense of what all the numbers and bars in the waterfall chart actually translate to for your users**. These tools offer a chance to make some subjective judgements in this otherwise scientific problem-solving process. Put yourself in the position of a visitor: is the page load process fast enough to keep you engaged and confident in the site's performance?



Page load screenshots of Kickstarter.com via WebsiteTest.com



A page load video of Kickstarter.com via WebsiteTest.com

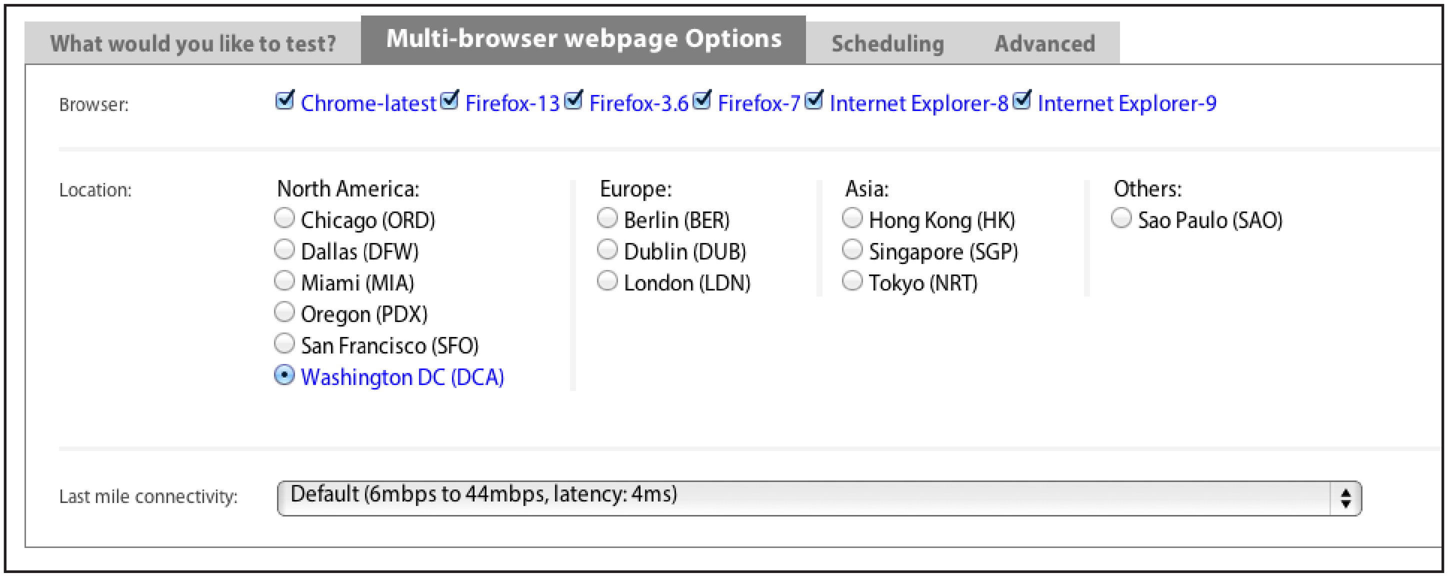
1d. Baseline Testing

*In order to gain a complete understanding of web performance it's a good idea to test your site's performance with as many key variables as possible. This is referred to as **multivariate testing**.*

The most common variables to test for are browser, geographic location, and last-mile connectivity (type and speed). Selecting multiple variables among these categories will help to identify weaknesses in performance based on delivery (geography), browser compatibility, and more.

There are a handful of uses for the data you collect at this stage. You could:

- Use the testing data to determine which techniques to use as you organize and prioritize your WPO project (Chapter 2).
- Use the data later, as a basis of comparison to gauge the success of your efforts in development and after going live (Chapters 3 and 4).
- Use the data to create benchmarks of your competitors (next page).



Browser: ☑ Chrome-latest ☑ Firefox-13 ☑ Firefox-3.6 ☑ Firefox-7 ☑ Internet Explorer-8 ☑ Internet Explorer-9

Location:

North America:
○ Chicago (ORD)
○ Dallas (DFW)
○ Miami (MIA)
○ Oregon (PDX)
○ San Francisco (SFO)
● Washington DC (DCA)

Europe:
○ Berlin (BER)
○ Dublin (DUB)
○ London (LDN)

Asia:
○ Hong Kong (HK)
○ Singapore (SGP)
○ Tokyo (NRT)

Others:
○ Sao Paulo (SAO)

Last mile connectivity: Default (6mbps to 44mbps, latency: 4ms)

WPO Tip: Make note of the exact locations, browsers, and connectivities you test by, as well as the frequency and time of day and week. The testing you'll do after implementing techniques should be roughly equivalent in order to provide an accurate control. Save your test data in a spreadsheet or other means, or use a built in save function if it is available in your testing tool.

1e. Benchmarks

Another way to make use of the results of your website tests is to ***compare your site's performance to your competitors'****.*

Return to the testing tool you used before, and this time plug in a sample of your competitors' websites. Find at least five, but an ideal sample is 10 to 15. Putting together a report on these test results will help you to make more informed decisions on how to improve your own site.

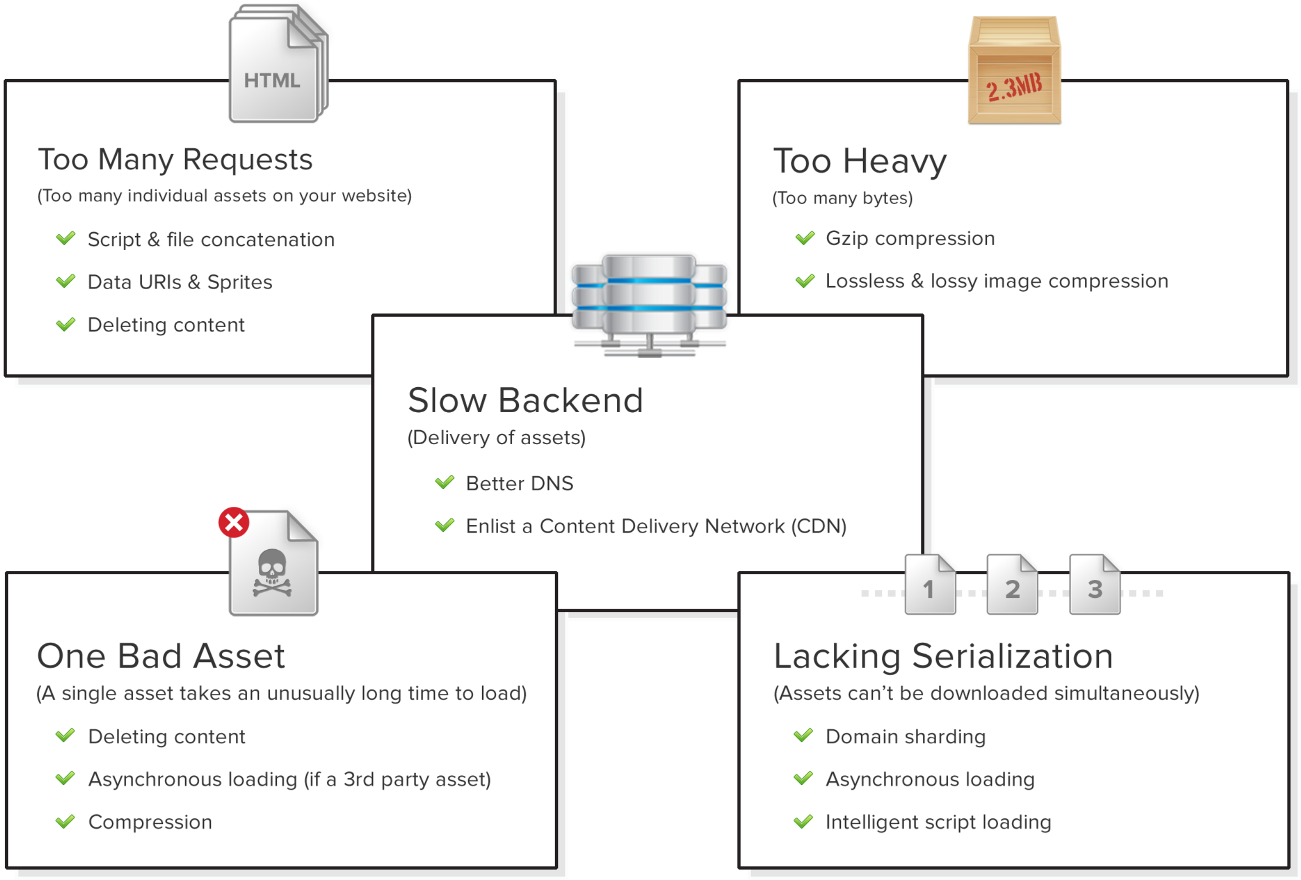
You can also compare your site to the figures in the table below. It contains performance data that Yottaa collected from over 15,000 websites. Find out how your site's metrics compare to the rest of the Web!

	50th Percentile	Bottom 20th PCT	Bottom 5th PCT
DNS Time (msec)	60	149	413
Connect Time (msec)	144	188	227
Time to First Byte (msec)	533	1,090	2,418
Time to Last Byte (msec)	721	1,412	3,105
Time to Title (msec)	1,259	2,368	4,647
Time to Render (msec)	2,420	4,071	7,160
Time to Display (msec)	5,168	8,780	15,254
Time to Interact (msec)	6,263	10,643	18,494
DOM elements	281	601	1224
Domains	6	17	33
Asset Count	47	84	152
Asset Weight (Bytes)	690,779	1,557,812	3,393,192
JS Count	8	19	33
JS Size (bytes)	119,378	346,852	586,442
CSS Count	3	8	16
CSS Size (Bytes)	18,134	60,265	120,000
Image Count	25	48	87
Image Size (Bytes)	318,788	895,948	2,056,237

2. Planning

2a. Define scope, Identify techniques

For any performance problem found on your site, there's at least one known technique to fix it. Here are some common problems and the performance techniques associated with them:



Too Many Requests

(Too many individual assets on your website)

- ✓ Script & file concatenation
- ✓ Data URIs & Sprites
- ✓ Deleting content

Too Heavy

(Too many bytes)

- ✓ Gzip compression
- ✓ Lossless & lossy image compression

Slow Backend

(Delivery of assets)

- ✓ Better DNS
- ✓ Enlist a Content Delivery Network (CDN)

One Bad Asset

(A single asset takes an unusually long time to load)

- ✓ Deleting content
- ✓ Asynchronous loading (if a 3rd party asset)
- ✓ Compression

Lacking Serialization

(Assets can't be downloaded simultaneously)

- ✓ Domain sharding
- ✓ Asynchronous loading
- ✓ Intelligent script loading

WPO Tip: Check out the last page of this eBook for helpful resources on choosing techniques to use in WPO!

2b. Prioritize

You simply can't do it all: there's always another way to make your website faster. That's why it's a good idea to prioritize your projects so that you are getting the most out of your time.

When approaching any website project, you must consider tradeoffs. As programmer Douglas Crockford notes in Steve Souders' *Even Faster Websites*: **each time you touch code, you trade off the potential for improvement with the possibility of breaking the code.** Keep this in mind.

Some WPO techniques are non-invasive. Compressing a single image, for instance, is unlikely to cause any issues with your site. However other techniques, like JavaScript minification and script concatenation, involve tinkering with the code itself, which is inherently risky. Take the nature of the techniques into account when you assess the risk of implementation.

In addition to assessing the risk, you should predict the amount of time each part of the project will take, and cross that with the potential impact. As is the case with any project with a fixed goal, at a certain point increasing time spent on the project will begin to outweigh the benefit of completion.

To prioritize, first figure out where your site falls short. Do you have too much JavaScript? Are your images too heavy? Does your site have a long time to first paint? Out of your site's shortcomings, find the areas with the largest potential benefit. For instance, a single image that loads slowly is easy to optimize, but doing so may have a minimal effect on the overall site load time. Conversely, if you have dozens and dozens of JavaScript assets, concatenating these scripts should be a high priority since the impact on your site's load time is likely to be significant.

2c. Find Tools

When you've identified some techniques to try, it's a good idea to do some Internet searches to see if any tools exist that can make those jobs easier. For some techniques, developers have created shortcuts. JavaScript minification, for instance, has at least three tools -- JSMin, Minify JS, and JS Packer -- to help automate the process of stripping out white space. **Tools like these aren't perfect (and should be used with caution) but they do have the potential to seriously cut down development time.**

3. Development and Testing

3a. Development

Using what you found in preliminary testing and benchmarking, it's time to take action. We've already shown some common problems and their associated solutions, but here's an overview of avenues you can take:

Sever side issues:

- Tune your database, improve business logic
- Load balancing, install firewall

Network (delivery of content)

- TCP/IP tuning
- CDN (Akamai, Yottaa)

Front end

- Reduce payload
- Reduce number of requests
- Promote parallel processing

3b. Test

Now the fruits of your preliminary baseline testing will come to bear. Before going live with your site changes, test the site in a staging environment, making sure sure that the server cache has surpassed its Time To Live (TTL). The TTL is point at which the cache refreshes to reflect any recent changes to the site content.

Test the front-end and backend performance data with multiple browsers, locations, and connection speeds. It's also a good idea to space tests out over time for a day or two in order to account for variations in the data that could be due to the daily and weekly vicissitudes of the Internet.

Cross-reference your baseline data with the new data looking for discrepancies. Compare the averages from before and after to see if your work has produced a net change, but also pay attention to outliers. This may involve drilling into the waterfall to see if any individual assets or groups of assets have been affected by your work.

- If the changes you find in the testing data match the intended effect of the technique - awesome!
- If you notice changes in your site for the worse, or in seemingly unrelated parts of your site, the technique you've implemented may have had unintended consequences. Figuring out why may involve some debugging.

3c. Tweak/Refine

Based on what you find in testing, make any refinements you can to improve the results.

Example

Say you initially discovered a third-party JavaScript asset that was loading *synchronously*, and was therefore holding up your site's load process. You entered a script that forces the JavaScript to load *asynchronously*, in hopes of improving site performance. Although you succeed in making the asset load asynchronously, post-implementation testing reveals that the asset is still hurting performance, this time because it is simply a heavy asset that loads slowly. You would then want to circle back and compress and/or minify that JavaScript to complete the WPO project.

4. Go Live!

*When you have implemented the techniques that you prioritized in step 2, and tested them thoroughly, **go live** with the changes on your main site.*

4a. Make sure nothing breaks!

If you've done web development before, you already know: as rigorously as you test in staging, things can still go wrong when your changes go out in the "real world." Leave yourself time to troubleshoot small problems, or (if bigger problems arise) to retract your changes and go back to square one.

4b. Ongoing monitoring

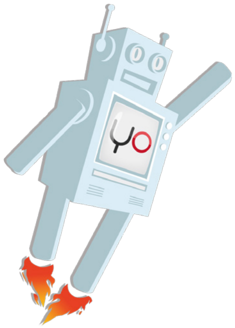
This is the final step, and it is crucial. In order to guarantee a return on the investment of time and effort you've made in web performance optimization, you must monitor website performance continuously over time. The web is constantly shifting and changing, and your website's performance can likewise change at any moment.

There are several options for site performance monitoring including Pingdom, Yottaa Site Monitor, and Keynote. Using these tools, set up your site to be monitored in real time and save data to show trends. If available, set up service level agreements (SLAs) to establish a threshold for performance goals, and further, set up alerts or notifications when performance emergencies or SLA violations arise.

If you take the time to set up monitoring properly, the data will become your best friend. You will be able to plan future WPO projects more accurately with extensive data on your site's performance at your disposal.

Resources

- *High Performance Websites* by Steve Souders
- *Even Faster Websites* by Steve Souders
- 11 Techniques To Make your Web Performance Rock (Yottaa ebook)
- 17 Metrics You Should Care About (Yottaa ebook)
- YSlow Best Practices for Speeding Up Your Web Site
- Yottaa Blog
- Performance Matters Blog by Pat Meenan
- AaronPeters.nl - Blog by Aaron Peters
- Google Developers Blog
- Book of Speed eBook by Stoyan Stefanov



About Yottaa

Yottaa is an easy-to-use cloud service that optimizes, protects, and monitors any website.

Check out our most popular blog posts:

- How To: Identify 10 Performance Patterns in 10 Seconds
- 8 Monitoring Tips to Keep Your Site Up & Performing Well
- Monitoring SaaS Performance Behind a Login Screen? No Sweat.

Download Other Useful eBooks:

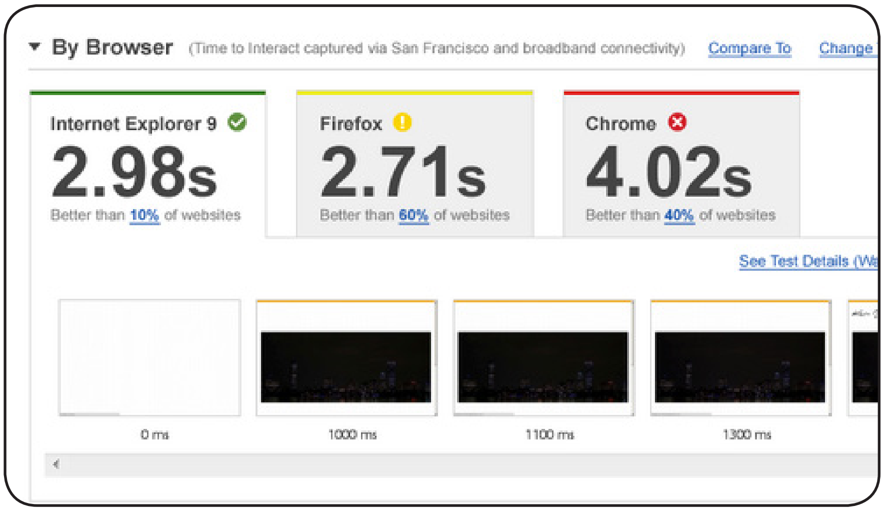
- 11 Techniques to Make Your Website Rock
- 17 Metrics You Should Care About

How Does Your Site Compare?

Request a Free Website Benchmarking Report to Get:

- **A Breakdown of site performance** from around the world, across browsers and connection speeds
- **Benchmarking** relative to your competitors
- **Identification** of the biggest issues impacting your site

Get Your Report Now →



▾ By Browser (Time to Interact captured via San Francisco and broadband connectivity) Compare To Change

Internet Explorer 9 ✓
2.98s
Better than 10% of websites

Firefox !
2.71s
Better than 60% of websites

Chrome ✕
4.02s
Better than 40% of websites

See Test Details (W

0 ms 1000 ms 1100 ms 1300 ms

Like this eBook? Share it!