Web Performance Metrics 101

AlertSite by SMARTBEAR

Web Performance Metrics 101

Contents

Preface

Slow response time has been the most common complaint of site users since the inception of the Web.[1] Just when we thought broadband and quad core processors would solve all our problems, mobile devices and Wi-Fi hotspots set us back again. The struggle against latency remains an ongoing battle, but the first step towards a faster Web begins with accurately measuring and optimizing the factors that make up response time and page load time.

In 2006, Amazon reported that for every 100ms improvement in response time, they received a 1% increase in revenue.[2] In 2008, Shopzilla reduced page load times from 7 seconds to 2 seconds and saw a 7% – 12% increase in conversion rate.[3] In 2010 Mozilla shaved 2.2 seconds off its landing pages and increased download conversions by 15.4% — resulting in an additional 60 million downloads.[4] While these stats clearly quantify the value of optimization, it doesn't tell us how fast is fast enough. Clearly there must be diminishing returns in optimizing for response and page load time. We'll, get to that, but the first question really needs to be — what exactly do we mean when we refer to response time?

In 2006, Amazon reported that for every 100ms improvement in response time, they received a 1 % increase in revenue.

Technically, response time is the time it takes for a user to send a command (for example, a page request) and the browser to finish loading the related HTML. Simple enough, but when you consider how a modern page is designed, with so many additional objects, response time doesn't tell you very much about the user's experience.

[1] www.websiteoptimization.com

[2] Make Data Useful, Greg Linden Amazon, 2006

[3] http://velocityconf.com/velocity2009/public/schedule/detail/7709

[4] FireFox and Pageload Speed – Part II,

A somewhat better measure is page load time. Page load time measures how long it takes for the browser to finish loading the page and all referenced objects after the user sends a command.

Like response time, page load time is not one thing, but many. It's a series of unfolding steps that should be monitored individually, as each step can help tell the story of where a problem lies — but let's focus on response time to start.

Response Time

Response time consists of the following components:

- DNS resolution time
- TCP connection time
- HTTP redirect time
- Time to first byte
- HTML content time
- Full page object load time

Finding the exact cause of a slowdown requires knowing how these components operate — both individually and together.

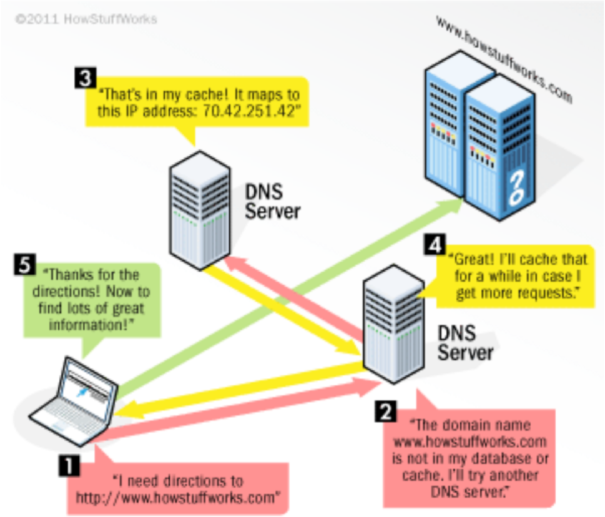
DNS Resolution Time

The DNS lookup time measures how long it takes to resolve the website's hostname to a certain IP address. Most people tend to think that DNS resolution is either working or not, but it's not that simple.

You may experience more subtle problems, like long response times, timeouts, and corrupt caches. In these cases a query can get through — it just takes a lot longer.

Usually, if the DNS lookup time is high, it means that you or your hosting provider has a problem with their DNS servers. Remember that

DNS resolution time goes up significantly with the distance separating the DNS name server and the site — particularly for international web sites — and DNS resolution time goes down considerably for cached resources.


©2011 HowStuffWorks
www.howstuffworks.com
3
"That's in my cache! It maps to this IP address: 70.42.251.42"
DNS Server
4
"Great! I'll cache that for a while in case I get more requests."
5
"Thanks for the directions! Now to find lots of great information!"
DNS Server
2
"The domain name www.howstuffworks.com is not in my database or cache. I'll try another DNS server."
1
"I need directions to http://www.howstuffworks.com"

A graphic representation the DNS process from HowSuffWorks.com

TCP Connection Time

Once the URL has been resolved to an IP address, TCP connection time shows how long it takes to establish a connection to your server.

Monitoring TCP connection time helps you get ahead of your users in discovering network latency, routing issues, and server bandwidth problems.

For example, if your server's bandwidth is insufficient for its workload, clients will usually become aware of this before the server does. Client requests to the server might be rejected or time out, or the response might be delayed. On the server side, the indicators are less clear because the server continues to establish connections, receive requests, and transmit data.

HTTP Redirect Time

HTTP redirection, also called URL forwarding, is a web technique for making a web page available under more than one URL address. When a web browser attempts to open a URL that has been redirected, a page with a different URL is opened.

HTTP Redirect Time starts when the TCP connection time is completed. It represents the amount of time between sending the initial notification to redirect and fully receiving the final object to which the browser is redirected. If there are no redirections, redirection time is zero.

URL redirection can be used for URL shortening, to prevent broken links when web pages are moved, or to allow multiple domain names to refer to a single web site.

HTTP Redirect Time includes all of the response time metrics of the server the user is redirected to, including DNS resolution time, TCP connection time, and others.

You never want more than one redirect to get to any of your resources. Since your webpage isn't just loading HTML, you need to know what resources your page is calling as it loads. There may be redirects in CSS files, images, external scripts or other objects.

Time to First Byte

When we think about site optimization, we tend to think about optimizing content — combining files, optimizing multimedia, properly caching and compressing files. But some response time slowdowns require server optimization.

One of the best indicators of a slow server is time to first byte. The first byte time shows how long it takes from the moment a connection is created until the first byte is received by the browser. The time to perform any negotiations with the server and the time needed for the server to calculate the result are also included.

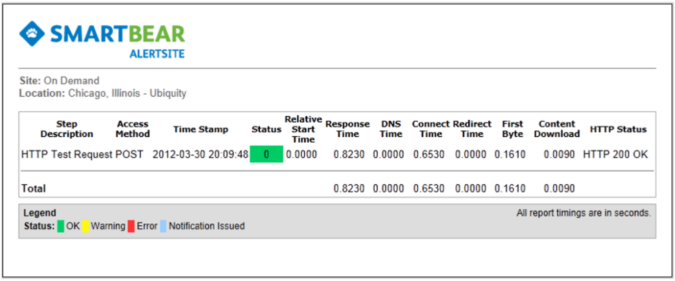
Common server problems indicated by time to first byte include memory leaks, programs that spawn too many processes — and fail to shut them down on completion — inefficient SQL queries, and calls to busy external resources like Google and Facebook.

HTML Content Time

Once the first byte of HTML has been received, the Web server continues to send the HTML that represents the layout of the web page, including CSS and Java Script. The content time is directly related to the size of the HTML.

The overall content time value is calculated by measuring the time elapsed between the first content load activity and the end of the last content load activity for all documents on the page. An event may load a page that contains multiple documents, and each document may have its own content time. So, HTML content time includes time to last byte for all documents loaded with the HTML.

HTML content time is generally considered a good bandwidth indicator, but be careful. As you've probably guessed by now, it doesn't tell you much about the user's experience since so much content is provided by other objects that load after the last byte of HTML content.



SMARTBEAR ALERTSITE

Site: On Demand
Location: Chicago, Illinois - Ubiquity

Step Description	Access Method	Time Stamp	Status	Relative Start Time	Response Time	DNS Time	Connect Time	Redirect Time	First Byte	Content Download	HTTP Status
HTTP Test Request	POST	2012-03-30 20:09:48	0	0.0000	0.8230	0.0000	0.6530	0.0000	0.1610	0.0090	HTTP 200 OK
Total					0.8230	0.0000	0.6530	0.0000	0.1610	0.0090	

Legend
Status: 🟩 OK 🟨 Warning 🟥 Error 🟦 Notification Issued

All report timings are in seconds.

Response Time Monitoring Example from AlertSite

Full Page Object Load Time

This brings us to Full Page Object Load Time. As soon as the HTML content is completely retrieved, the browser analyzes the HTML to determine what additional objects require retrieval.

Full Page Object Load Time starts with the last byte of HTML and ends when all page objects are fully loaded. These objects include all Web page references to images, JavaScript, CSS, Flash objects, RSS feeds and JavaScript files.

Measuring full-page object load time is particularly useful for monitoring the effects of third-party content, like ads, however, it doesn't take into account what your users are actually seeing. For example, it doesn't tell you whether a slow-loading piece of third-party content exists above or below the fold. So while Full-page object load time by itself may alert you to problems, they may not actually affect the user's experience of the page.

Page Load Time and the User Experience

Now that we've explored each component of response time, you should have a better understanding of how to use these metrics to identify the source of your problem (if you have one) — but the question of how to measure and optimize the user experience still remains. While these network side metrics can tell you where to look if you have a problem, they don't necessarily uncover the affect of the problem on your users.

When talking about speed improvements that translate into revenue, we're talking about user perception — how users are actually experiencing your website and applications. While these backend metrics can provide insight around specific technical problems that may or may not be affecting overall page load and response times, there is no direct connection between their measurements and the user's experience.

While this continues to be an evolving area for performance metrics there are several newer metrics that you should be leveraging in order to monitor your users' perceptions of performance and overall experience. These include:

- Page Load Time
- DOM Load Time
- First paint Time
- Above-The-Fold Time

Technically, Page Load Time and DOM Load Time are browser timings, not user timings, but they help set the stage for what the user perceives.

Page Load Time

Page Load Time represents the entire time elapsed between the user command and the completed loading of the page and all referenced objects by the browser. It is the be-all, end-all of response time metrics as it includes all of the page's documents, all referenced objects and scripts, style-sheets, and images. While page load time may seem important, it isn't as valuable as it appears because it includes everything — whether visible to the user or not. As we'll see later, what the users don't know won't hurt them.

DOM Load Time

Dom Load Time represents the time it takes to finish parsing the page's documents beginning with the first byte, but not including referenced style-sheets, images and sub-frames.

It is the browser's job to piece together each page element and load external files into memory. The DOM, or Document Object Module, defines a standard way of structuring and parsing the elements of an HTML page, including text, I-frames, and the like so they can be logically assembled before being loaded.

The "DOMContentLoaded" event is fired when the document has been completely loaded and parsed without waiting for stylesheets, images, and subframes to finish loading. That means DOM load time is concerned with the structure of the page rather than its content. If, after a while, a website begins to bog down, it's often because front-end blocks of code in the DOM become exaggerated or locked or refer to hidden bits.

A high number of DOM elements can be a sign that the markup of the page needs fixing without necessarily removing content. For example, throwing in extra <div>s only to fix layout issues can create unnecessary complexity over time. Increased complexity means slower DOM access in JavaScript. Having to loop through 1000 DOM elements versus 100 can materially slow the DOM load time.

> A high number of DOM elements can be a sign that the markup of the page needs fixing without necessarily removing content.

When simplifying a complex page, take note that differing parsing engines sequence the elements for assembly differently. Chrome, Safari, Firefox, and Opera may all load your page in different orders, which can make a difference to the user's perception.

First paint Time

A lot has been written about the impatience of Internet users, but nothing frustrates them more than being flat-out ignored. first paint Time measures how long it takes for the browser to display the first burst of visual activity after a user has entered a URL.

First paint tells the user that the site is responding to their action. How long will a user wait? About as long as it takes to blink. According to a New York Times article, if users don't get that acknowledgement in 250 milliseconds — literally the time it takes to blink — they will abandon the site. [5]

Once the first paint begins, the order in which elements load becomes critically important. Web users spend 80% of their time looking at information above the page fold. Although users do scroll, they only allocate 20% of their attention below the fold.

Above the fold Time

Above the fold is where users focus their attention — meaning you need to focus on isolating and optimizing above-the-fold performance throughout your website.

Above-The-Fold Time measures the time it takes for the visible page's content to reach its final rendering, with enough intelligence to adapt for animated GIFs, streaming video, rotating ads, etc.

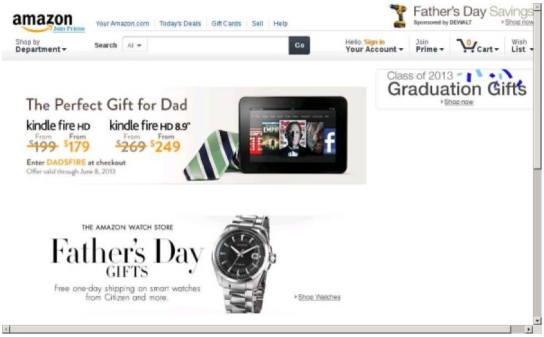
[5] The New York Times, "For Impatient Web Users, an Eye Blink is Just Too Long to Wait," by Steve Lohr, 2/29/2012

The first thing above the fold time tells us is that you shouldn't set your site to load all the images when the page starts. A site's assets should load in roughly the order of importance to the user, loading content and images that are above the fold first to optimize the user experience.

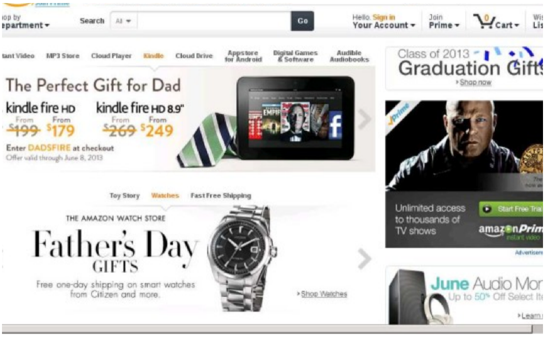
AlertSite Visual User Experience Metrics

When looking at these newer user experience metrics it's important to note that many Web Performance Monitoring solutions rely on implied metrics from within the browser to report first paint and above the fold timings — which can lead to misleading results. For example, if the first object displayed on a page happens to be a white image on a white background it wouldn't necessarily resonate with users as the first burst of visual activity. Similarly, many times above-the-fold page content appears stable to a user before it actually is according to the data from the browser.

AlertSite by SmartBear goes a step further when delivering these timings by video processing your page as it renders to report when the user's eye would be able to see the first burst of activity on the page and again when the user would visually perceive above-the-fold stability. As a result, AlertSite first paint and above the fold measurements will not always match the same browser-implied metrics for a given page — but in reality, AlertSite is capturing the true experience of your users much more accurately. Additionally, AlertSite provides screenshots of exactly what your users see at these two critical moments in the page load process.



AlertSite First Paint Screenshot for Amazon.com — 1.1883 Seconds



AlertSite Above the Fold Screenshot for Amazon.com — 3.3303 Seconds

Conclusion

In the end, web performance monitoring is like politics — perception is reality. While traditional backend metrics can provide insight around specific technical problems, they don't provide much insight into the thing that matters most: the user's experience. By incorporating the user-centric metrics we've discussed into your WPM plan you can quickly identify optimization opportunities that will directly translate into a better experience for your visitors.

About AlertSite

AlertSite, a part of SmartBear Software, is a global leader in Web, API and mobile performance monitoring solutions that continuously improve the Web user experience. AlertSite uniquely provides both technical performance measurement as well as visual user experience measurement from more than 80 locations around the globe and from within your environment with InSite, a private monitoring location. AlertSite's services measure basic availability, Web performance using real instances of IE, Firefox and Chrome browsers, API performance and mobile website performance. Gain a real-time view of service quality in terms of availability, performance, consistency and the user experience.

Sign up for your complimentary 30-day trial here.

About SmartBear Software

More than one million developers, testers and operations professionals use SmartBear tools to ensure the quality and performance of their APIs, desktop, mobile, Web and cloud-based applications. SmartBear products are easy to use and deploy, are affordable and available for trial at the website. Learn more about the company's award-winning tools or join the active user community at http://www.smartbear.com, on Facebook or follow us on Twitter @smartbear and Google+.

SMARTBEAR

SmartBear Software, Inc. 100 Cummings Center, Suite 234N   Beverly, MA 01915
+1 978.236.7900 www.smartbear.com