

BREAKING THE SPEED LIMIT: GOOGLE'S PERFORMANCE GURU ON ACCELERATING WEB PERFORMANCE

A conversation with Web Performance Expert Steve Souders

Steve Souders is the author of *High Performance Web Sites*, *Essential Knowledge for front-end Engineers*, and creator of the *YSlow* front-end performance tool. He spent a number of years at Yahoo!, where he was the Chief Performance Yahoo! Currently, Steve is a member of the technical staff at Google, and a frequent speaker on Web performance topics. Benchmark talked to Steve to get his perspective and tips on optimizing front-end Web site performance.

Benchmark: You've been on both sides of the performance equation—the back-end and the front-end—and you have an expert point of view on how each impacts the ultimate user experience. Where do you get the best performance gain for your effort?

Steve Souders: I break performance into two main areas: efficiency and response time. On the efficiency side, we're looking to build Web application architectures that can scale, that can handle a huge number of users and requests and large amounts of data in an efficient way, while also bringing in our hardware costs and data centers, space costs, power costs, of course maintenance on all that hardware. This is all obviously on the back-end.

Until I got more focused on it, the thought was that this was also the place for improving the response time—making the user experience faster. Prior to starting my work in the performance area about four years ago, I was running large Web applications and people complained about response times. So I would look what could be done to the back-end architecture to try to bring up a 200 millisecond page creation time to 150 milliseconds or something like that.

But it turns out that, if you view a Web page loading from the end user's perspective, we're talking about *thousands* of milliseconds, and so this back-end timesaving of 100 or 200 or 500 milliseconds really isn't much of a factor when you look at the overall load time of a page from the browser's perspective, from the end user's perspective.

So if what you are really trying to do is deal with scalability issues, the place to focus is the back-end. If you're having a huge spike in traffic or a large increase in the amount of data or back-end calculations that you need to do, then the place of focus is on the back-end.

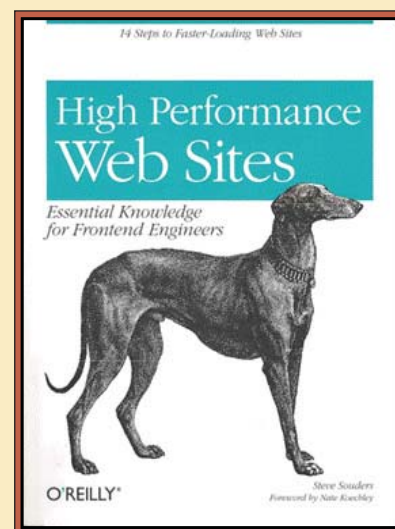
But if your objective is making user experience faster, the place of focus is the front-end.

Benchmark: And that's a big enough topic you could probably write a book about it—as you did. Are you seeing more people implementing better front-end practices?

Steve Souders: I think we're getting greater awareness out there. In the last year since I've been out talking, even though the size and complexity of pages has increased, the speed at which, for example, the Alexa top 10 are triggering that first page is improving, so I think that there is attention on the front-end and it is paying off.

Benchmark: An important result of your front-end performance work was the development of your YSlow tool, which is now integrated with the popular Firebug Web development tool. Can you tell us what it is, how it works, and why Web developers should be using it?

Steve Souders: I mentioned about four years ago I started in this role of really looking at performance from the end user's perspective and trying to find ways to speed up the time it takes pages to load from the client side.



Steve Souders' book outlines the best practices he developed and incorporated into the YSlow tool. It is available at amazon.com.

I started doing research and working with different development teams and Web sites and trying different things to find out what works, what has the biggest impact, what has the biggest bang for the buck. I started collecting those and making a list of performance best practices so that they could be evangelized from one development team to another. That list of best practices held to be pretty accurate.

But there were so many teams to do this evangelism and consulting with, that it was clear I just didn't have enough hours in the day to reach out to each one of them myself. So I wanted to try to codify this knowledge. I started my career working in artificial intelligence. So what I tried to do was to look at these best practices to see if any of them could be done in an automated fashion. It turns out that all of them can, and so I started building YSlow, which ultimately became part of Firebug.

"Firebug quickly caught on at Yahoo! and became the preeminent tool for Web developers to use, so it made sense to integrate YSlow with Firebug. A lot of the best practices that I talk about for improving the user experience are now followed by these front-end developers."

Benchmark: So what does YSlow show you? How does it help developers to build faster sites?

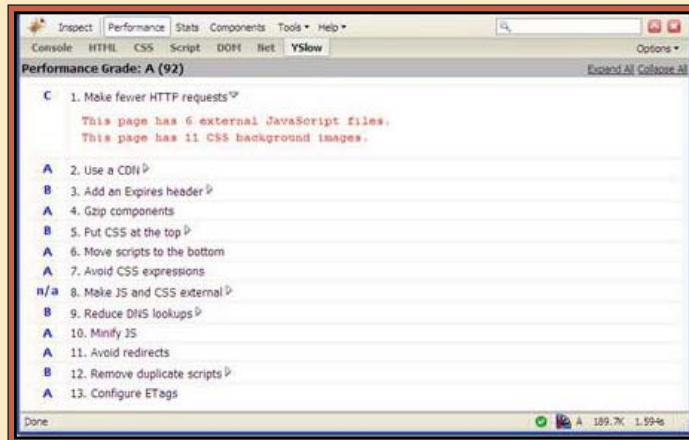
Steve Souders: You open it inside Firebug and run it on any Web page and it generates a grade from 0 to 100 that tells you how well your page is doing, based on these performance best practices, and then provides the specific details of where you're deviating from best practices and what's needed to fix those differences. It's free, it's real easy to use, it produces clear actionable output.

YSlow has created a simple vocabulary for people to talk about performance across the entire organization. Even people outside of engineering understand a YSlow score of 92 is better than 73. Or you can also translate that into letter grades – A, B, C, D, E – and people always want to look for As.

So when you show people that they are getting a C or a D and you tell them exactly what they need to do to pull up their grade, it really empowers them to make those improvements. And it lets people across the organization, from marketing to products to executives to the engineering team, track the progress and talk about that and prioritize improvements.

Benchmark: So you get your YSlow score—you're a C or a B, a 77 or an 83, say—now what do you do with it? YSlow shows you the details, which practices need fixing. Do you start rebuilding the site?

Steve Souders: It's nice that a lot of these performance improvements are actually pretty low-cost and easy to implement, especially when you compare them to some of the back-end architectural projects that companies undertake. For example, turning on compression or g-zipping.



YSlow evaluates any Web site and grades it according to the 14 rules described in Steve Souders' High Performance Web Sites. The report includes an overall grade, grades on each of the 14 performance areas, and specific recommendations for improving performance.

Typically in Apache, I believe, compressing the HTML that a server generates is on by default, but that's not the case for other forms of ASCII text responses like script, style sheet, XML responses—but anything that's actually text can be compressed.

And all it takes is one, two, or maybe three lines of configuration added to a Web server's configuration file. Yet some of these changes can have a big impact. You can reduce the number of bytes transferred over the wire by 70% if you turn on compression. And you can do that just by adding a few lines to your Web server's configuration file.

It's nice that even when you have to make those trade-offs between performance improvement and features, a lot of these performance improvements come with a very low cost.

So you can walk through the list and prioritize them by the investment, and tackle some of the low-cost, big-bang-for-your-buck improvements first.

Benchmark: People on the business side are getting more and more exposed to Web 2.0 technology and are starting to demand more of it for their Web sites. What's your overall take on the performance impact of these trends to make the content rich and active and personalized, and what's your advice to people who are looking at those things?

Steve Souders: Well, I certainly think it's true that as you add more content to your page, it's going to make it slower. That's just physics. If you have more bytes to download, it's just going to take more time.

But if that work of enriching a Web page is done hand-in-hand with improving the way that the page is built, including sensitivity to performance best practices—the “performance golden rule”—then you could actually end-up with a richer page that is even faster than the previous, simpler version of the page.

What I encourage people is to do those two hand-in-hand. As you're going back to revamp a site, to enhance the aesthetic quality of it, to add more features—do it in a way that follows performance best practices.

And at the end, most likely it's going to be faster than what you had before, even though it has more features and more content on the page.

Benchmark: What are some things you're seeing developers doing that are dealing a negative hit to performance? Bad practices vs. Best practices?

Steve Souders: JavaScript is something that I've been talking about lately and doing a lot of work on. There are painful characteristics about how browsers handle JavaScript, where it blocks all of the downloads. So when the browser is parsing and executing and downloading JavaScript, it basically stops all other activities. Doing things in parallel is a great way to improve how the browser loads a page, and the default ways of dealing with JavaScripts prevent that.

Benchmark: And yet it's getting to be that everyday, more and more sites are using JavaScript to create a richer or more interesting user experience.

Steve Souders: The trend is definitely the adoption of more and more JavaScript. There are also now a lot of JavaScript frameworks out there, and sometimes they can be pretty large, you know, hundreds of K of JavaScript.

We're even seeing sites where they want some of the features of one JavaScript framework and some from another and they're loading both of those in their pages. The way that the browser views this JavaScript blocks a lot of other actions, and so it can have a big delay in how long it takes for a browser to load a page.

It's pretty alarming to see this, the huge performance impact that JavaScript can have in a page, and also see that JavaScript is growing by leaps and bounds.

But I think it's important to keep in mind that this is an exciting

technology. With Ajax and Web 2.0, we really can do some amazing things in the browser. But we don't want to push the browser to its limits so much that we produce Web applications that have a really negative user experience and drive users away.

I know a lot of Web 2.0 applications have an option to switch back to an HTML only mode. I think we want to see that go away. We want to see where users don't have to choose between a rich full-featured application and one that loads fast. We want to have both of those things. That's what we need to work towards going forward—how to deal with this explosion in the use of JavaScript and still serve those Web 2.0 applications very quickly.

Benchmark: OK, now on the flip side, what are some of the more exciting, positive trends you're seeing in Web performance?

Steve Souders: I think everyone would agree the most exciting movement we're seeing there are the browser wars between Opera, WebKit, Safari, Firefox, and Internet Explorer. They're all touting performance as a speed differentiator.

There are JavaScript benchmarks, you know, load time—all of them are putting these statistics forward as a way to help promote their browser as the one that users should prefer.

That competition is really going to improve the experience for users. WebKit and IE8 are the first ones to be able to support downloading JavaScript in parallel, something that previous browsers don't do. More than any of the other actual optimizations of the JavaScript engine, that one change alone, being able to download scripts in parallel, is going to be the biggest improvement for JavaScript going forward.

Benchmark: Your *High Performance Web Sites* book describes 14 rules for improving front-end performance, but now we understand you have 10 more rules. Can you give us some highlights?

Steve Souders: Sure. The first three all deal with JavaScript, how to optimize the JavaScript that's downloaded to the browser and how to improve, through parallelization, the way that browsers deal with JavaScript.

So for example, the first new rule is that, if you have a large amount of JavaScript, load that so that you only download exactly what's needed for the initial page first, and then load all the rest of it later after the page is already displayed for the user.

That technique is going to work across all browsers. But then the second new rule that I'm talking about is ways that you can load those external scripts in parallel with other downloads in the page. That's not the default, but advanced techniques for downloading scripts exist where you can get parallelization. In some cases those advanced techniques differ in how well they work and in

their behavior across browsers.

So in some cases you can adopt these best practices and they work across all browsers. In other cases, you might need to special-case what you do depending on the browser.

Benchmark: Sounds like some very effective techniques in your new rules. So, to wrap things up and come full circle—your YSlow tool gives developers best practice benchmarks for how their sites are built. But what about measuring the actual end-user experience? Of course, we’re wondering how Keynote fits into your thinking on Web performance.

Steve Souders: Well, I tell teams that, even before they start working on any specific performance improvements, the first thing that’s put in place is measurement. It’s really important that you have an idea of what your performance is now and that you quantify that—so that as you make improvements, you can be sure you’re moving in the right direction, and you can also gauge the impact of your work, that you make sure that you’re investing resources wisely.

Keynote is a key for that part of getting started on Web performance, being able to very quickly gather specifics on page load times for your own Web site pages as well as your competitors. Then being able to track those trends over time on an objective platform that’s distributed worldwide is really key. Keynote has services that support that, and tools like the new release of Kite make that easy for companies to adopt.

Benchmark: You prefer getting real measurements from the field vs. simulations?

Steve Souders: Yes. As you said, there are ways to try to simulate some of those network and geographic distribution variables into the performance equation, but those simulations always fall short along one dimension or another. I really like being able to get data from the actual region, using an Internet connection that’s typical of users that are located there, and also doing that in a real browser. So the fact you can do that through Keynote is very powerful.

Benchmark: Thank you, Steve. And best of luck at Google!



Steve Souders

Web Performance Expert
Google

Steve works at Google on Web performance and open source initiatives. His book *High Performance Web Sites* explains his best practices for performance along with the research and real-world results behind them. Steve is the creator of YSlow, the performance analysis extension to Firebug. He is the co-chair of Velocity 2008, the Web performance and operations conference from O’Reilly, and is co-founder of the Firebug Working Group. He is teaching at Stanford, and frequently speaks at conferences including OSCON, The Ajax Experience, Rich Web Experience, and Web 2.0 Expo.