

# Speeding Up JavaScript



Lori MacVittie, 2007-17-08

An interesting article on "[How JavaScript is Slowing Down the Web \(And What To Do About It\)](#)".

The basic premise is that Web 2.0 applications like blogs are using so much JavaScript to load widgets and perform other functions that it's causing the initial page load to be slow.

That's so true.

The author has many (okay, there are five) suggestions for improving performance, but one sticks out in my mind, probably because it's core to F5 and its products - and it's not entirely accurate.

### 3. Load-balance requests by generating different URLs

Most scripts these days give you back the same URL, for example: `www.mycoolwidget.com`. This is not easy to scale. As the widget becomes more wide-spread, the amount of concurrent requests going through the same server will be quite large. A better approach is to give out different URLs, like this: `server1.mycoolwidget.com`, `server2.mycoolwidget.com`, etc. Even though for now they can be pointing to the same domain, in the future there will be an option to split the traffic.

Let's dissect this, because there's a few things that need to be said about this particular suggestion and I'm going to say them.

#### 1. *This [a single URL] is not easy to scale*

Not true at all. The whole point of load balancing is that a single URL *can* scale if it's handled by a solution like [BIG-IP](#). A single URL is handled by the load balancer, which in turn intelligently determines which of X number of servers is best capable of handling the request at the time it is received. The *load* is therefore *balanced* across a pool of servers.

In fact, I'd argue that a single URL is the easiest thing in the world to scale, especially if you just want straight up, layer 4 load balancing. It's transparent, it's simple, and as long as you can add servers, you can scale to handle just about any load.

#### 2. *Give out different URLs*

That's not really load balancing, that's purely traffic management and architecture; it's a distributed architectural solution to an age old problem. What this *does* help with is that it gets around the browser imposed limitations on the number of concurrent connections that can be open to a server.

The issue with modifying the application is that the URLs (another misconception, the author is actually suggesting that you modify the *host*, not the entire URL) are now hard coded and that makes future architectural changes more difficult because you have to change the code everywhere the URL is hard-coded. A much better option for achieving this functionality is to use an application acceleration product like [BIG-IP WebAccelerator](#) because it transparently performs this type of functionality and doesn't require any changes to your code.

The author also forgets to mention caching and compression, both of which can help speed up the delivery of JavaScript. When load balancing is coupled with an application acceleration technology like [BIG-IP WebAccelerator](#) you get even better performance because it can intelligently cache those scripts and deliver them, effectively offloading the burden from the server. This means the server has more CPU cycles and resources to respond to requests that can't be cached, like requests for truly dynamic, data-driven content.

I'm not disagreeing that load balancing can help improve the performance of web applications, especially those laden with JavaScript. Quite the contrary! What I am disagreeing with is the author's understanding (and presentation) of load balancing, and the fact that while load balancing and application delivery solutions will certainly speed up the delivery of applications and its JavaScript components, load balancing in and of itself cannot improve the execution of JavaScript once it's on the client, which is really the point of the author's post in the first place.

If you're trying to get content to the client faster and more efficiently, then a load balancing and application acceleration solution (or as we say in the 21st century, application delivery network) is definitely the way to go. As for speeding up the execution of JavaScript on the client?

That's a whole new technology, one that has yet to be invented.

*Imbibing: Pink Lemonade (made with real lemons, or so says the can)*

Technorati tags: [F5](#), [BIG-IP](#), [MacVittie](#), [application delivery](#), [Web 2.0](#), [application acceleration](#)

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)