




Zoompf's Web Performance Blog



Note: Archived Content

This is the archived version of the Zoompf blog. Since our acquisition by Rigor (</blog/2015/10/zoompf-gets-acquired>), all our new research and posts on web performance are being published on The Rigor Blog (<http://rigor.com/blog>)

What Bagging Groceries Can Teach You About Web Performance

 Zoompf Performance (</contact>) on May 6, 2015. Category: Optimization (</blog/category/optimization>)



(/wp-content/uploads/2015/04/grocery_cart.jpg)

Image courtesy of Pixabay.

While you're looking at your web browser, let's go grocery shopping. Just in your mind. OK, what's your grocery list look like? How many times per week do you go to the grocery store? Are you a *planner* who makes a thorough list and only goes shopping once a week? Or are you a little more haphazard and end up visiting the grocery store 3 or 4 times each week for just a few items each time?

While everyone has their patterns and preferences, there's no arguing over which method is most efficient. Make a list, and go to the store once a week to get it all. Very efficient. Now let's think for a minute how you check out and get your groceries to your car (assuming you drive to the store). Do you take each individual item in your hand and take it to your car, then come back in and get one more item?

Of course not! That would just be crazy talk. You'd spend all day going back and forth to your car!. Nobody would ever take just one item at a time out of the store to their vehicle. But many websites force your web browser to do just that.

Yes, we just compared web browsing your favorite web sites and apps to grocery shopping. The store is the website you're visiting. Each "item" is an HTML request to the server. The desktop or mobile browser GUI is your "vehicle". How many trips you make is, unfortunately, determined by the developers of the website or web app, not the user. How would you like it if your local grocery store stopped you when you were about to walk out the door, and told you that you can only take one item at a time out with you?!? That's what we thought you'd say.

Instead, you'd put several items in each grocery bag, and several grocery bags in a cart. If you shop once a week and have a big family, you might have a few dozen bags and a couple of shopping carts. THAT's how you'd get your stuff out of the store efficiently, and you would hope for a talented check out clerk at the store who can pack your grocery bags efficiently, putting cold

items in the same bag, lighter items in the same bag and on the top of the cart, loading items that are too large to fit in bags on the bottom of the cart, and making sure none of your bags are too heavy.

The grocery store is the developer of the web site you're using. It's up to them to make sure your bags (HTTP requests) are optimized for travel efficiency (to your browser).

The average web site, as of mid-2014, consisted of 96 objects and was 1622K in size (<http://www.websiteoptimization.com/speed/tweak/average-web-page/>). Each object requires an HTTP request, which is the equivalent of taking home 96 grocery items one at a time. When the web site is in development, it's good practice to break your code up into different libraries and files; however, you should start thinking about web performance from the beginning of design, and prepare to combine as many HTTP requests as you can (</blog/2015/01/automatically-optimize-css-javascript-asp-net>), without making each request too large.

Using this page as an example, Google Chrome's tools show the following results from loading the page.

43 requests | 11.8 KB transferred | Finish: 188 ms | DOMContentLoaded: 315 ms | Load: 342 ms

Compared to the PGA Tour website, which is very media heavy:

160 requests | 1.7 MB transferred | Finish: 21.49 s | DOMContentLoaded: 1.97 s | Load: 3.33 s

It's simple to see the many differences, starting with more than 4X the number of HTTP requests on the PGA Tour website, in addition to the total size of the elements in the site. Each request takes time, even if only a few milliseconds, and that time is crucial in keeping visitors and shoppers on your site. Just like at a grocery store, if the service is slow and inefficient, you may not return there to do business.

In addition to reducing image sizes through lossless or lossy optimization, website developers and designers should plan from the beginning of the specifications process for excellent performance by reducing the number of HTTP requests that any client browser will have to make in order to use the page or app. Combining CSS and JavaScript files together is a great way to start (</blog/2015/01/automatically-optimize-css-javascript-asp-net>).

If you are interested in reducing HTTP requests to speed up website performance, you will love Zoompf. We have a number of free tools that help you detect and correct frontend performance issues on your website: check out our free report ([/free](#)) to analyze your website for common performance problems, and if you like the results consider signing up for our free alerts ([/alerts](#)) to get notified when changes to your site slow down your performance.

Next Post (</blog/2015/05/announcing-the-new-face-of-zoompf>)

Earlier Post (</blog/2015/04/web-performance-at-the-speed-of-light>)

Comments