Automating Performance with PageSpeed

network, compute, and render...

Ilya Grigorik
igrigorik@google.com

Video @ bit.ly/io-pagespeed
● Web applications are becoming more powerful
● Web applications are becoming more complex
● Web applications are becoming more ambitious

... and speed is a feature, among many others.
Our applications are complex, and growing...

<table>
<thead>
<tr>
<th>Content Type</th>
<th>Desktop</th>
<th>Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg # of requests</td>
<td>Avg size</td>
</tr>
<tr>
<td>HTML</td>
<td>10</td>
<td>56 KB</td>
</tr>
<tr>
<td>Images</td>
<td>56</td>
<td>856 KB</td>
</tr>
<tr>
<td>Javascript</td>
<td>15</td>
<td>221 KB</td>
</tr>
<tr>
<td>CSS</td>
<td>5</td>
<td>36 KB</td>
</tr>
<tr>
<td>Total</td>
<td>86+</td>
<td>1169+ KB</td>
</tr>
<tr>
<td>Delay</td>
<td>User reaction</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>0 - 100 ms</td>
<td>Instant</td>
<td></td>
</tr>
<tr>
<td>100 - 300 ms</td>
<td>Slight perceptible delay</td>
<td></td>
</tr>
<tr>
<td>300 - 1000 ms</td>
<td>Task focus, perceptible delay</td>
<td></td>
</tr>
<tr>
<td>1 s+</td>
<td>Mental context switch</td>
<td></td>
</tr>
<tr>
<td>10 s+</td>
<td>I'll come back later...</td>
<td></td>
</tr>
</tbody>
</table>

The 1000 ms "time to glass" challenge.

- Simple user-input must be acknowledged within ~100 milliseconds.
- To keep the user engaged, the task must complete within 1000 milliseconds.

Ergo, our pages must render with 1000 milliseconds.
What's the impact of slow sites?

Sure, fast is "good", but really... does it matter?
### Server Delays Experiment

- **50ms**
  - Distinct Queries/User: -
  - Query Refinement: -
  - Revenue/User: -
  - Any Clicks: -
  - Satisfaction: -
  - Time to Click: 500ms

- **200ms**
  - Distinct Queries/User: -
  - Query Refinement: -
  - Revenue/User: -0.3%
  - Any Clicks: -0.4%
  - Satisfaction: -
  - Time to Click: 1200ms

- **500ms**
  - Distinct Queries/User: -
  - Query Refinement: -0.6%
  - Revenue/User: -1.2%
  - Any Clicks: -1.0%
  - Satisfaction: -0.9%
  - Time to Click: 1900ms

- **1000ms**
  - Distinct Queries/User: -0.7%
  - Query Refinement: -0.9%
  - Revenue/User: -2.8%
  - Any Clicks: -1.9%
  - Satisfaction: -1.6%
  - Time to Click: 3100ms

- **2000ms**
  - Distinct Queries/User: -1.8%
  - Query Refinement: -2.1%
  - Revenue/User: -4.3%
  - Any Clicks: -4.4%
  - Satisfaction: -3.8%
  - Time to Click: -

- **2000 ms delay reduced per user revenue by 4.3%!**

- **Strong negative impacts**
- **Roughly linear changes with increasing delay**
- **Time to click changed by roughly double the delay**

---

**Performance Related Changes and their User Impact**
Impact of **1000 millisecond delay**...

A 1-SECOND DELAY =

- **7%** loss in conversions
- **11%** fewer page views
- **16%** decrease in customer satisfaction

In dollar terms, this means that if your site typically earns $100,000 a day, this year you could lose **$2.5 MILLION** in sales.

Source: Aberdeen Group

www.strangeloopnetworks.com
Web Performance Optimization (WPO)

- 86+ requests
- 1+ MB transferred

- Can we download less?
- Can we execute faster?
- Can we render faster?
If you care about performance, then...

- Image compression & resizing
- Minify CSS, JavaScript and HTML
- Inline small images, CSS, and JavaScript
- Cache all static assets
- Defer JavaScript
- Combine CSS and JavaScript
- Domain sharding
- ...

Rinse, lather, repeat...
Why aren’t all websites fast?

**SPEED**
- Long cache lifetimes
- Inlined / sprites / minification
- Exploit latest browser features
- Track latest WPO techniques

**EASE OF MAINTENANCE**
- Simple development & deployment
- Ability to rapidly deploy changes
- Support all browsers
- Focus on content

Use automated tools
To deliver fast and optimized applications, we must **invest into tools and workflows** which will help us **identify performance bottlenecks**, and **resolve them**.

*Performance is not a checklist, it's a [continuous process]*.
1. Identifies performance problems
2. Provides advice and guidance
3. **Automates** site optimization

PageSpeed
by Google™
What can I do to optimize my site?

Can you automate WPO best practices?

PageSpeed Insights

mod_pagespeed

ngx_pagespeed

PageSpeed Service
Analysis with PageSpeed Insights

What can I do to optimize my site?
Performance diagnostics in your browser with 30+ optimization rules

Install from Chrome Store
- Optimizing the following images could reduce their size by **4.9MB (51% reduction)**.
- Compressing resources with gzip could reduce their transfer size by **244.1KB (70% reduction)**.
- Minifying the following JavaScript resources could reduce their size by **105.1KB (40% reduction)**.

- Expiration not specified for 42 resources.
- Images should be combined into as few images as possible using CSS sprites.
- Defer parsing JavaScript to reduce blocking of page rendering.
- External CSS files were included after an external JavaScript file.

---

**Optimize images**

Properly formatting and compressing images can save many bytes of data. 

**Suggestions for this page**

Optimizing the following images could reduce their size by **4.9MB (52% reduction)**.
PageSpeed Insights

- Same functionality available in an online tool!
  - [https://developers.google.com/speed/pagespeed/insights](https://developers.google.com/speed/pagespeed/insights)
require 'net/https'
require 'json'

uri = URI.parse('https://www.googleapis.com/pagespeedonline/v1/runPagespeed')
http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true

params = { :key => 'API KEY', :url => 'http://mysite.com/',
           :strategy => 'desktop', :rules => '...' }

uri.query = URI.encode_www_form(params)
req = Net::HTTP::Get.new(uri.request_uri)
res = http.request(req)

jj JSON.parse(res.body)
Quick perf review: PageSpeed Insights
Local development: Chrome + PageSpeed
Performance monitoring: PageSpeed API

Follow the "learn more" link in each recommendation, to learn about the why and how of each criteria!
PageSpeed Optimization

If you can tell me what to optimize, and how, can you just do it for me?
"PageSpeed Optimization Libraries (PSOL) are a set of C++ classes that automatically optimize web pages and resources they use, using a server-independent framework."

https://developers.google.com/speed/pagespeed/psol
PageSpeed is a performance JIT

- **400,000+ sites** using server-side PageSpeed optimization
  - Open-source (free) and hosted versions

- **40+ optimization filters**
  - Single server, cluster, and CDN friendly
  - HTML, CSS, JS, and image optimization, all in one!
Optimizing the waterfall...

PageSpeed OFF

PageSpeed ON

http://www.webpagetest.org/result/121004_KP_CFM/3/details/

Same site, with respective waterfalls before and after *mod_pagespeed* optimization.

http://www.webpagetest.org/result/121004_0H_3C8/
With PageSpeed, you can...

- Keep your current workflow
- Eliminate additional compression and build steps
- Stop bugging designers and users to optimize images
- Get the benefit of dynamic UA optimization (e.g. WebP)
- ...

Automatically combining multiple CSS files

<link rel="stylesheet" href="styles/yellow.css">
<link rel="stylesheet" href="styles/blue.css">
<link rel="stylesheet" href="styles/big.css">
<link rel="stylesheet" href="styles/bold.css">

<div class="blue yellow big bold">Hello, mod_pagespeed!</div>

Combined file Served with 1-year TTL
Makes CDNs more effective

<link rel="stylesheet" href="styles/yellow.css+blue.css+big.css+bold.css.pagespeed.cc.HASH.css">

<div class="blue yellow big bold">Hello, mod_pagespeed!</div>

#protip: ModPagespeedCssInlineMaxBytes {max bytes}
Server-side image rewriting and optimization

```
<img src="photos/awesome_cat.png" width="800">
```

```
awesome_cat.png       350 KB
awesome_cat.jpg       80  KB
awesome_cat.webp      60  KB
```

Does the client support WebP?
(UA or Accept check)

Is awesome cat 800 px wide?
Nope, he is **8000 px** wide!

Resize the image

```
<img src="photos/800x450wawesome_cat.png.pagespeed.icHASH.webp" width="800">
```
### 40+ Optimization filters

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rewrite_javascript</td>
<td>Rewrites Javascript files to remove excess whitespace and comments.</td>
</tr>
<tr>
<td>combine_javascript</td>
<td>Combines multiple script elements into one.</td>
</tr>
<tr>
<td>inline_css</td>
<td>Inlines small CSS files into the HTML document.</td>
</tr>
<tr>
<td>inline_javascript</td>
<td>Inlines small JS files into the HTML document.</td>
</tr>
<tr>
<td>rewrite_images</td>
<td>Optimizes images, re-encoding them, removing excess pixels, and inlining small images.</td>
</tr>
<tr>
<td>convert_jpeg_to_webp</td>
<td>Generates webp rather than jpeg images for browsers that support webp.</td>
</tr>
<tr>
<td>lazyload_images</td>
<td>Loads images when they become visible in the client viewport.</td>
</tr>
<tr>
<td>resize_images</td>
<td>Implied by rewrite_images. Resizes images when the corresponding <code>&lt;img&gt;</code> tag specifies a smaller width and height</td>
</tr>
</tbody>
</table>

- **Core filters** are safe and enabled by default
- **Optional filters** must be enabled by site owner

[https://developers.google.com/speed/docs/mod_pagespeed/config_filters](https://developers.google.com/speed/docs/mod_pagespeed/config_filters)
Server performance with PageSpeed

- All optimization is performed on demand, results are cached
  - First request may serve unoptimized asset (for speed)
  - Optimization is done in the background (images, etc)

- For best performance....
  - Optimize the size of local cache (default 100MB)
  - Use a shared cache (memcached) for multi-server deployments
  - Configure fetch timeouts, number of optimization threads, ...

Lots of great performance tips in our documentation: developers.google.com/speed
Large and fast growing list of PageSpeed users...
Chrome Data Proxy is using PageSpeed!

Chrome Data Proxy is using PageSpeed: 50% data compression!
- Image optimization: convert all files to WebP
- Rewrites HTML, CSS, JavaScript

Give it a try: Settings > Bandwidth management > Reduce data usage.

https://developers.google.com/chrome/mobile/docs/data-compression
Aol. + mod_pagespeed

gadling.com
40% PLT improvement!

stylelist.com
20% PLT improvement!
"... up to a 75% reduction in page sizes and a 50% improvement in page rendering speeds."

https://www.zippykid.com/2013/04/23/partnership-with-google-to-deliver-fast-wordpress-sites/
Getting started with PageSpeed...

- modpagespeed.com
  - [mod-pagespeed-discuss](#)
    - `$> rpm -U mod-pagespeed-*.rpm`
    - `$> dpkg -i mod-pagespeed-*.deb && apt-get -f install`

- ngxpagespeed.com
  - [ngx-pagespeed-discuss](#)
    - `$> ./configure --add-module=$HOME/ngx_pagespeed`

- Community developed...
  - [IISSpeed](#) for Microsoft IIS server
  - PageSpeed for Apache Traffic Server
Wouldn't it be nice if...

- The optimization was done **automagically**
- We didn't need to modify or update our servers
- And we had an **all-in-one solution** for...
  - Optimization, CDN, DoS protection, ...
PageSpeed Service hosted by Google

PageSpeed optimization is performed on and by Google servers

1. Sign up at: https://developers.google.com/speed/pagespeed/service
2. CNAME www.your-site.com to pagespeed.googlehosted.com
3. Visitor hits the Google server
   ○ Google requests the resource from your origin server
   ○ Page is optimized and cached in Google CDN!

https://developers.google.com/speed/pagespeed/service
Optimize, CSS, JavaScript, Images... check.

Configure any and all filters from the Google API console!
PageSpeed Service on AppEngine!

www.html5rocks.com/app.yaml

ebidel a month ago Making H5R more awesome

4 contributors

<table>
<thead>
<tr>
<th>file</th>
<th>231 lines (183 sloc)</th>
<th>6.094 kb</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>application: html5rocks-hrd</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>version: master</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>runtime: python27</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>threadsafe: yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>api_version: 1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>pagespeed:</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>enabled_rewriters:</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>- MinifyCss</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>- InlineImages</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>- CollapseWhitespace</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>- ImageAddDimensions</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>- RemoveComments</td>
<td></td>
</tr>
</tbody>
</table>

313,495 of pageviews sent page load sample

- Avg. Page Load Time (sec): -11.91%
- Avg. Redirection Time (sec): -11.66%
- Avg. Server Connection Time (sec): -30.41%
- Avg. Page Download Time (sec): -29.47%

Updated .yaml file, made site **10% faster!**
Automating WPO offers big wins because...

- **Performance is a continuous process**
- **Minimizes mundane optimization work**
- **Dynamic optimization offers more opportunities**
  - User-Agent customization - e.g. WebP
  - Automagic HTTP 2.0 and SPDY enhancements
- **Allows you to focus on your application and users**
Fin. Questions?

+Ilya Grigorik
igrigorik@google.com

Video @ bit.ly/io-pagespeed