Video @ bit.ly/io-pagespeed

Automating Performance with PageSpeed

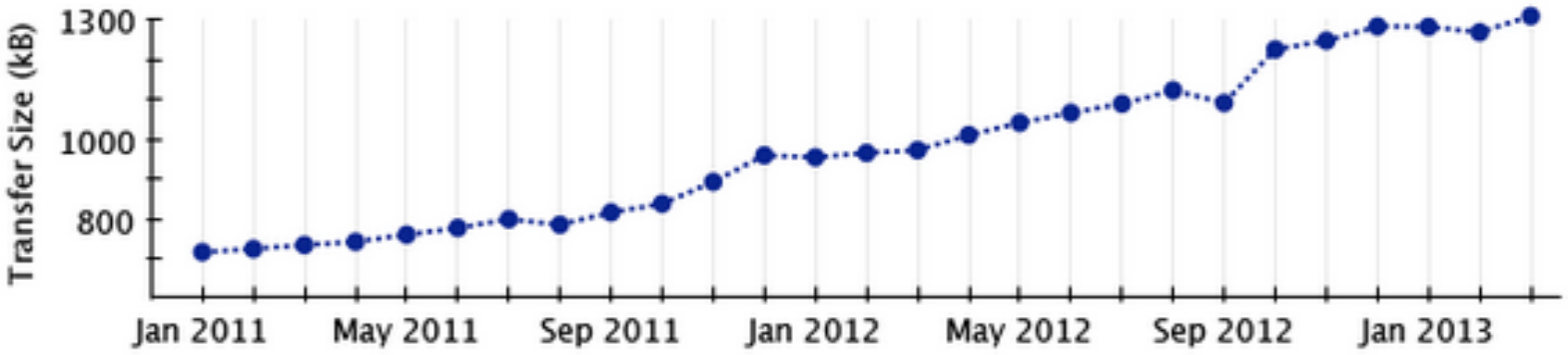
network, compute, and render…

Ilya Grigorik igrigorik@google.com

- **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...



Content Type	Desktop		Mobile	
	Avg # of requests	Avg size	Avg # of requests	Avg size
HTML	10	56 KB	6	40 KB
Images	**56**	**856 KB**	**38**	**498 KB**
Javascript	**15**	**221 KB**	**10**	**146 KB**
CSS	5	36 KB	3	27 KB
Total	**86+**	**1169+ KB**	**57+**	**711+ KB**

Ouch!

HTTP Archive

Delay	User reaction
0 - 100 ms	Instant
100 - 300 ms	Slight perceptible delay
300 - 1000 ms	Task focus, perceptible delay
1 s+	Mental context switch
10 s+	I'll come back later...

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?

Google & bing server delays experiment

	Distinct Queries/User	Query Refinement	Revenue/User	Any Clicks	Satisfaction	Time to Click (increase in ms)
50ms	-	-	-	-	-	-
200ms	-	-	-	-0.3%	-0.4%	500
500ms	-	-0.6%	-1.2%	-1.0%	-0.9%	1200
1000ms	-0.7%	-0.9%	-2.8%	-1.9%	-1.6%	1900
2000ms	-1.8%	-2.1%	-4.3%	-4.4%	-3.8%	3100

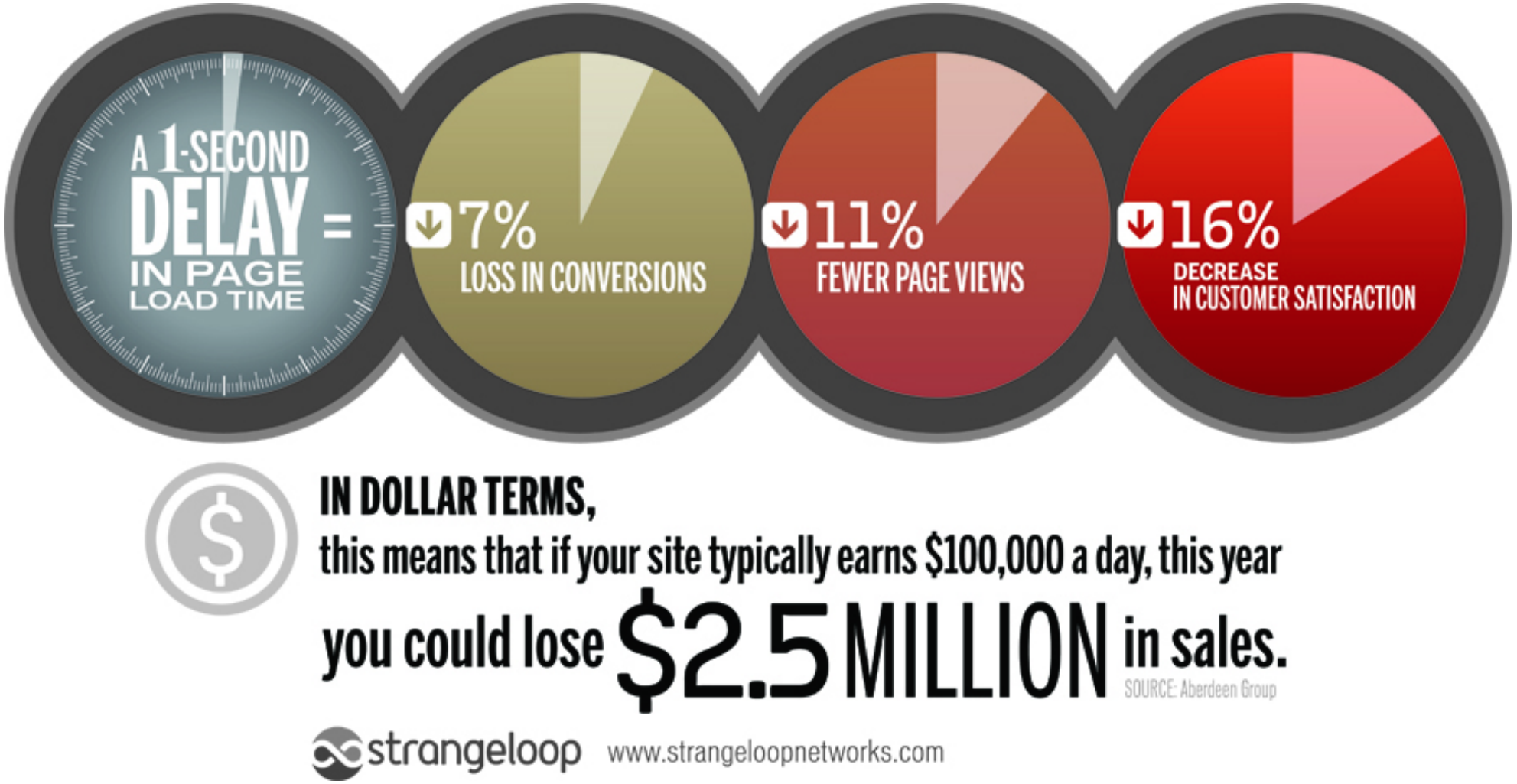
- Means no statistically significant change

"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

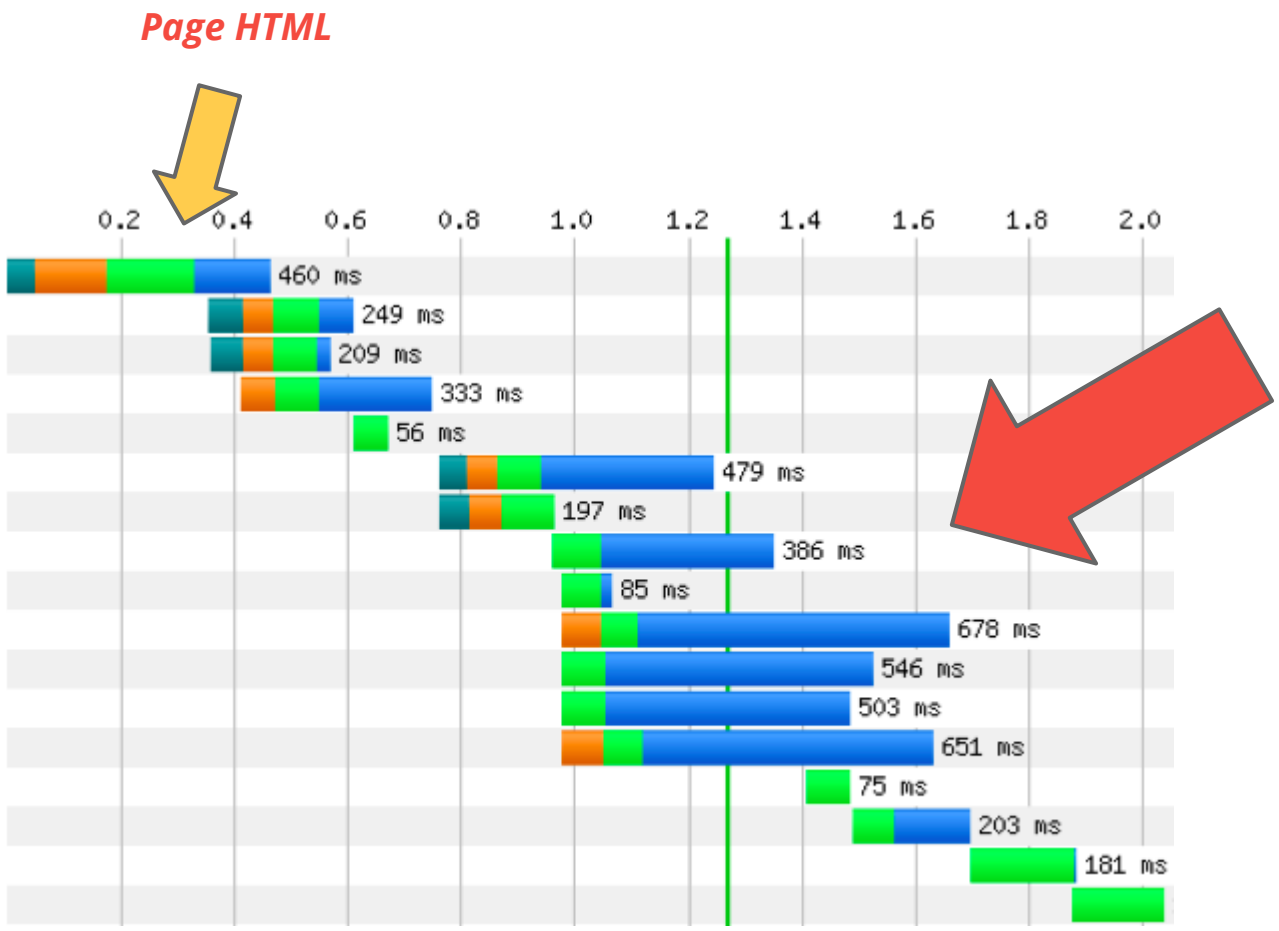
[Performance Related Changes and their User Impact]

Impact of 1000 millisecond delay...



[Impact of 1-second delay](#) - Strangeloop

Web Performance Optimization (WPO)

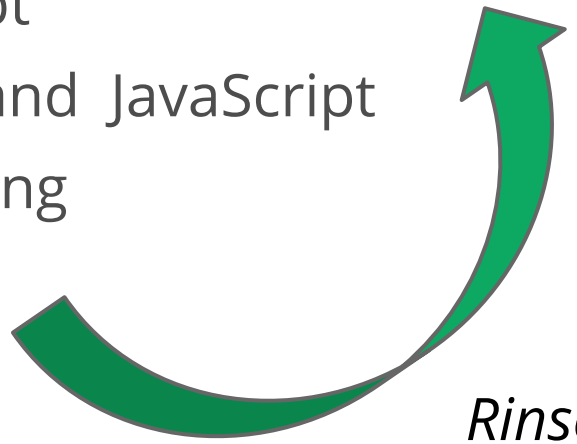


- **86+** requests
- **1+ MB** transferred

- **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
- …



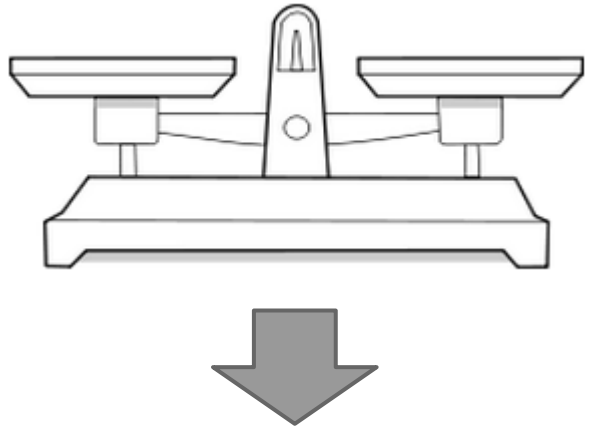
Rinse, lather, repeat…

Rinse, lather, repeat…

Rinse, lather, repeat…

Why aren’t all websites fast?

SPEED	***EASE OF MAINTENANCE***
Long cache lifetimes	Simple development & deployment
Inlined / sprites / minification	Ability to rapidly deploy changes
Exploit latest browser features	Support all browsers
Track latest WPO techniques	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 process.

continuous

PageSpeed

by Google

1. **Identifies performance problems**
2. **Provides advice and guidance**
3. **Automates site optimization**

PageSpeed

Analysis

What can I do to optimize my site?

PageSpeed Insights

Google apis

chrome

Optimization

Can you automate WPO best practices?

Apache mod_pagespeed

NGINX ngx_pagespeed

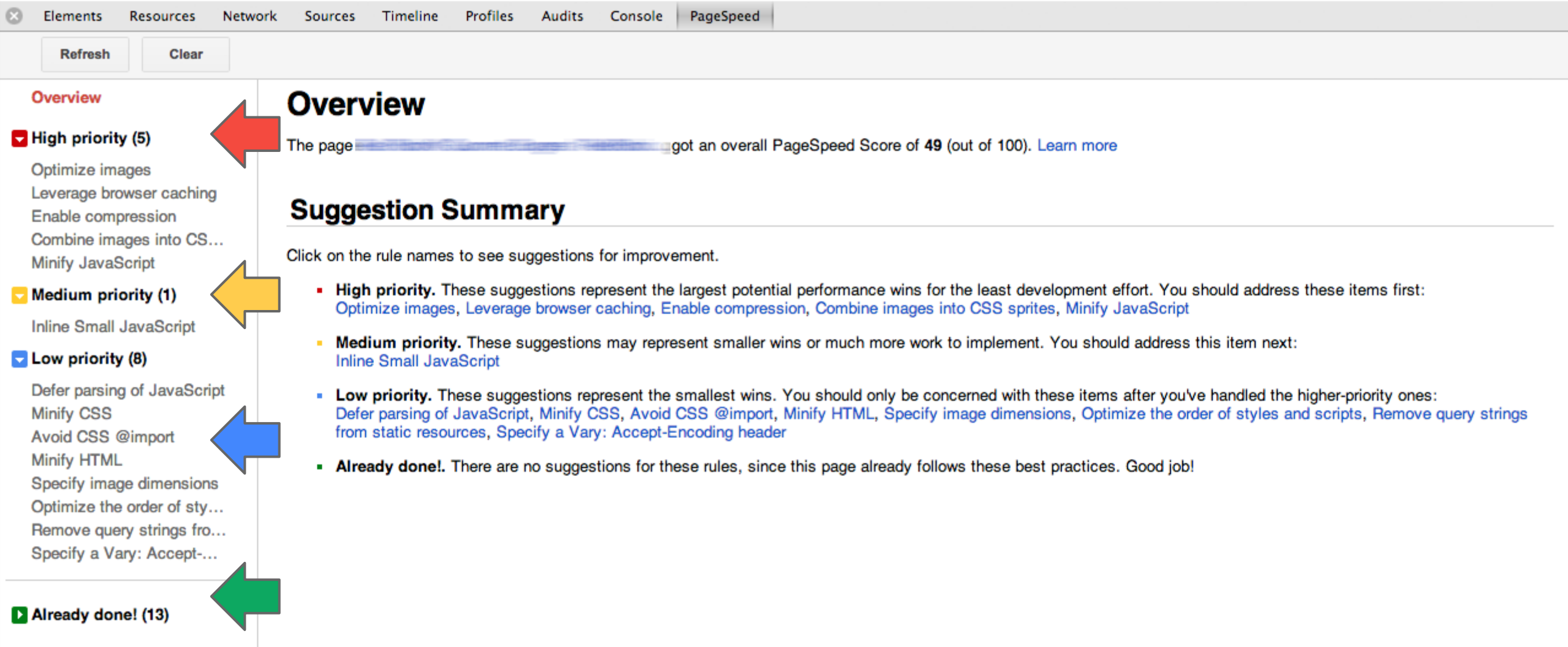
Google PageSpeed Service

Analysis with PageSpeed Insights

What can I do to optimize my site?

PageSpeed Insights + chrome

- Performance diagnostics in your browser **with 30+ optimization rules**
- Install from Chrome Store



PageSpeed Insights + chrome

- Overview
- **High priority (5)**
  - Optimize images
  - Leverage browser caching
  - Enable compression
  - Combine images into CS…
  - Minify JavaScript
- **Medium priority (1)**
  - Inline Small JavaScript

- 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.

- Overview
- **High priority (5)**
  - Optimize images
  - Leverage browser caching
  - Enable compression
  - Combine images into CS…
  - Minify JavaScript

Optimize images

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

Suggestions for this page

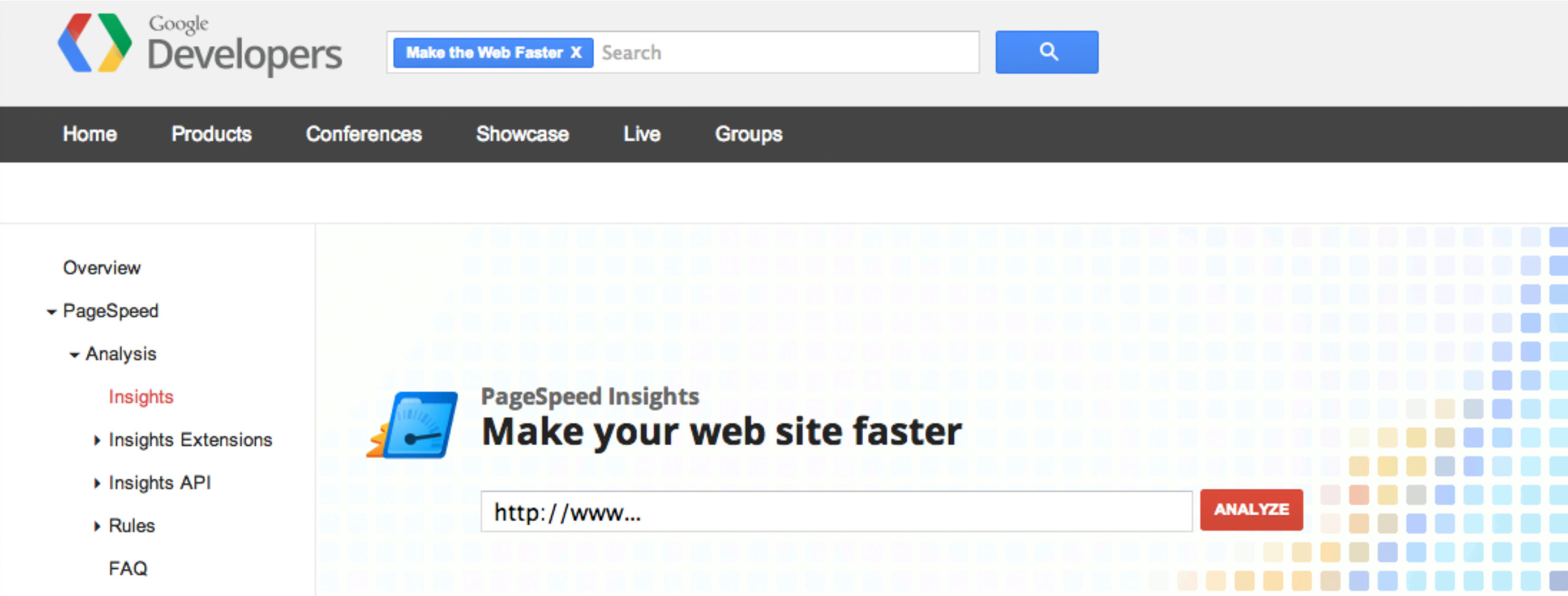
Optimizing the following images could reduce their size by 4.9MiB (52% reduction).

4.9MB reduction!

[PageSpeed Insights for Google Chrome]

PageSpeed Insights

- Same functionality available in an online tool!
  - https://developers.google.com/speed/pagespeed/insights



PageSpeed Insights

```ruby
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)
```

Online PageSpeed API for CI builds, catching performance regressions, reporting, ...

API documentation

PageSpeed Insights

- 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 Insights Rules

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!

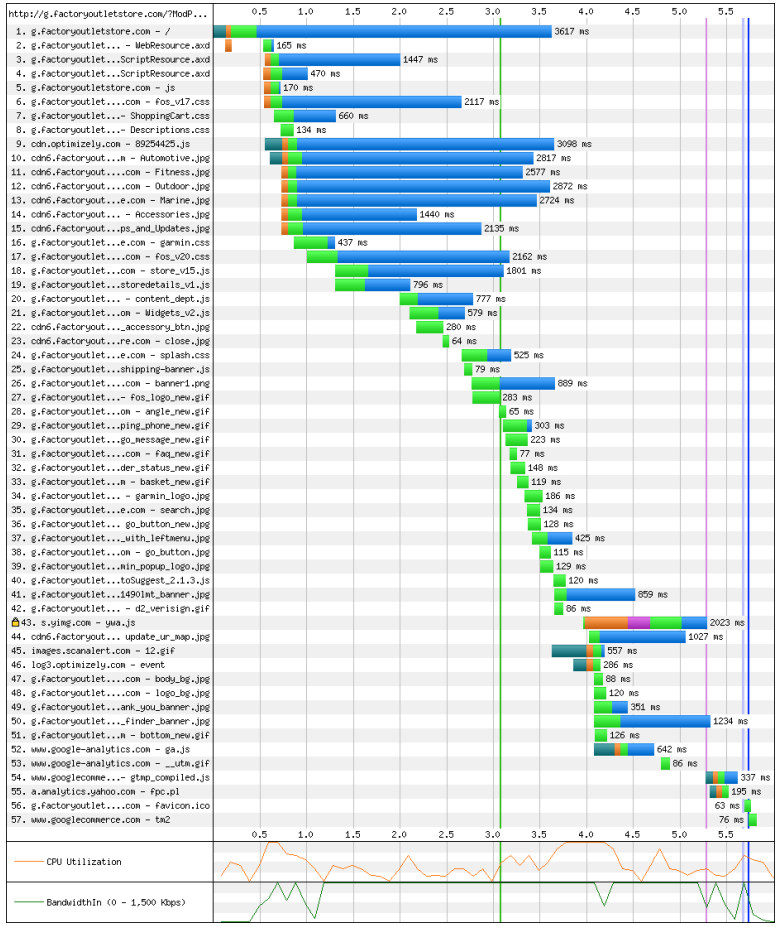
Apache mod_pagespeed

NGINX ngx_pagespeed

Google PageSpeed Service

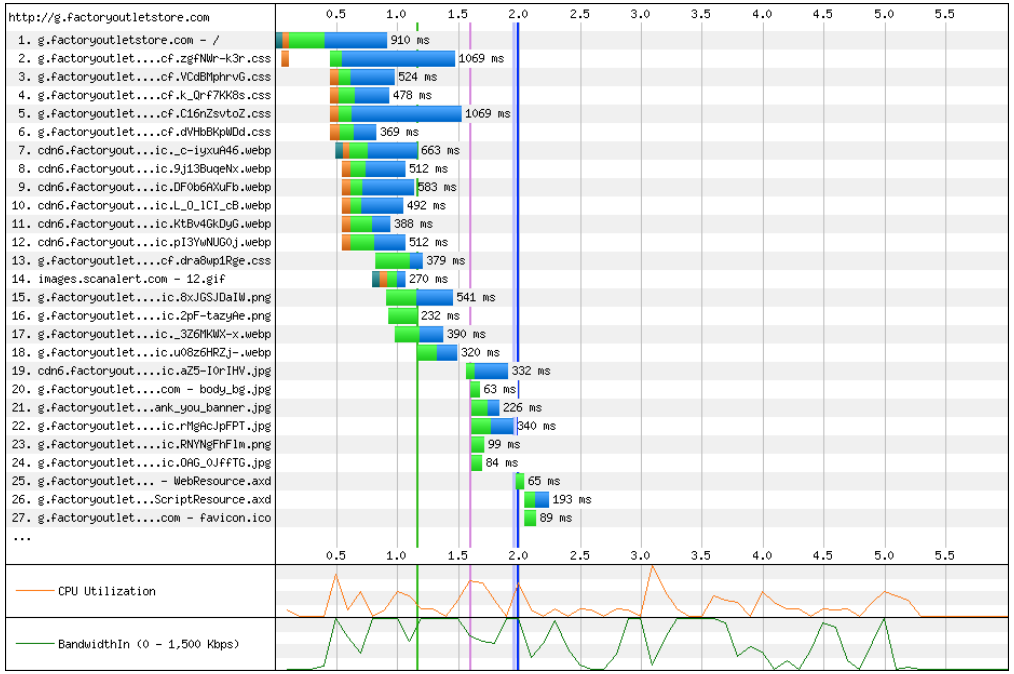
Optimizing the waterfall...

PageSpeed OFF



http://www.webpagetest.org/result/121004_0H_3C8/

PageSpeed ON



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

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

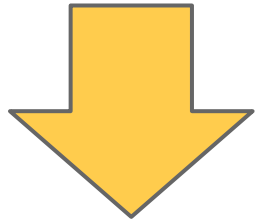
With PageSpeed, you can...

- *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



MD5 sum of combined CSS file

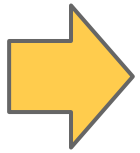
```
<link rel="stylesheet"
   href="styles/yellow.css+blue.css+big.css+bold.css.pagespeed.cc.HASH.css”>

<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.ic.HASH.webp" width="800">
```

40+ Optimization filters

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

- **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

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

Aol.

edgecast

DreamHost

maxCDN
CONTENT DELIVERY NETWORK

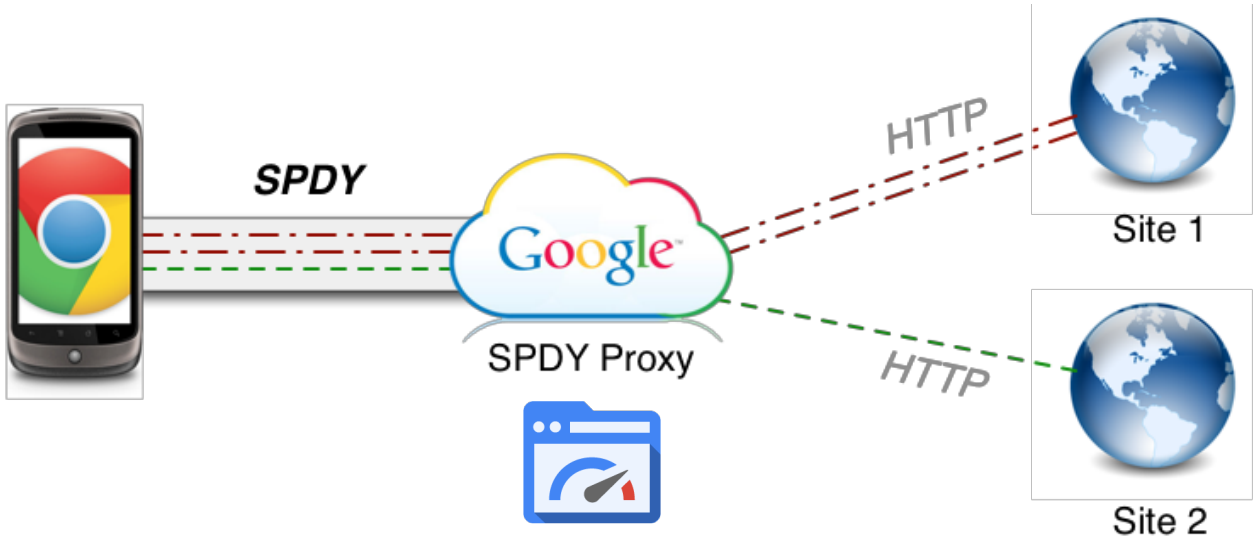
SiteGround

LinkedIn

淘宝网
Taobao.com

Large and fast growing list of PageSpeed users ...

Chrome Data Proxy is using PageSpeed!



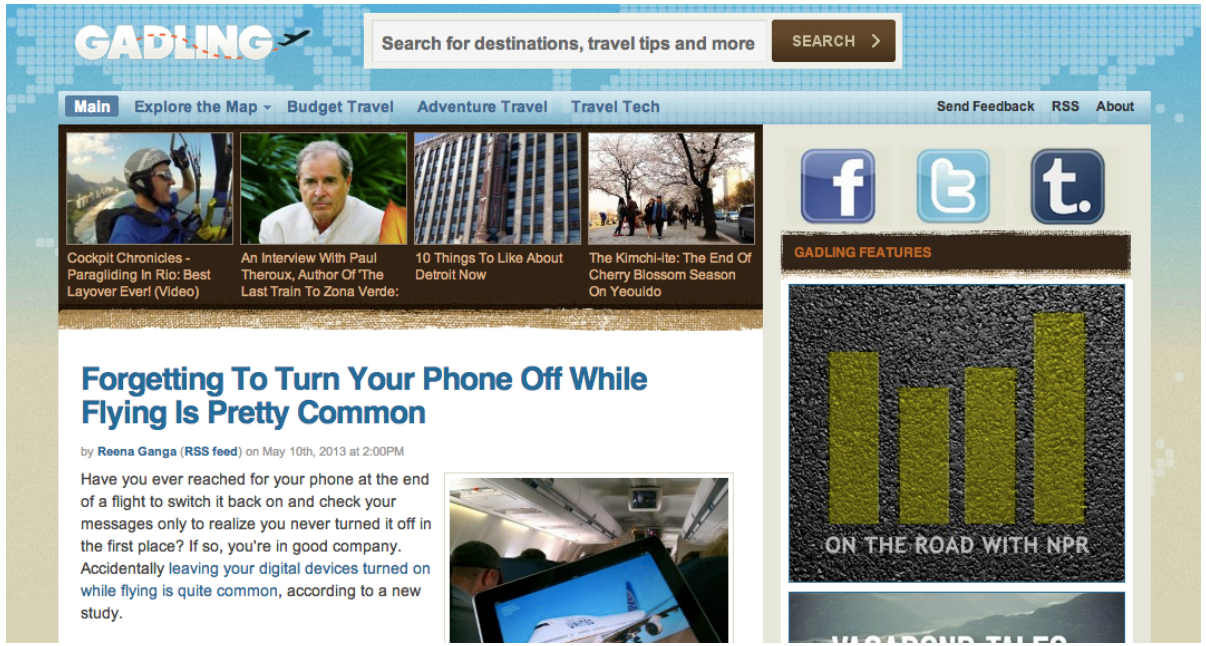
Chrome Data Proxy is using PageSpeed: 50% data compression!

- Image optimization: convert all files to WebP
- Rewrites HTML, CSS, JavaScript

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

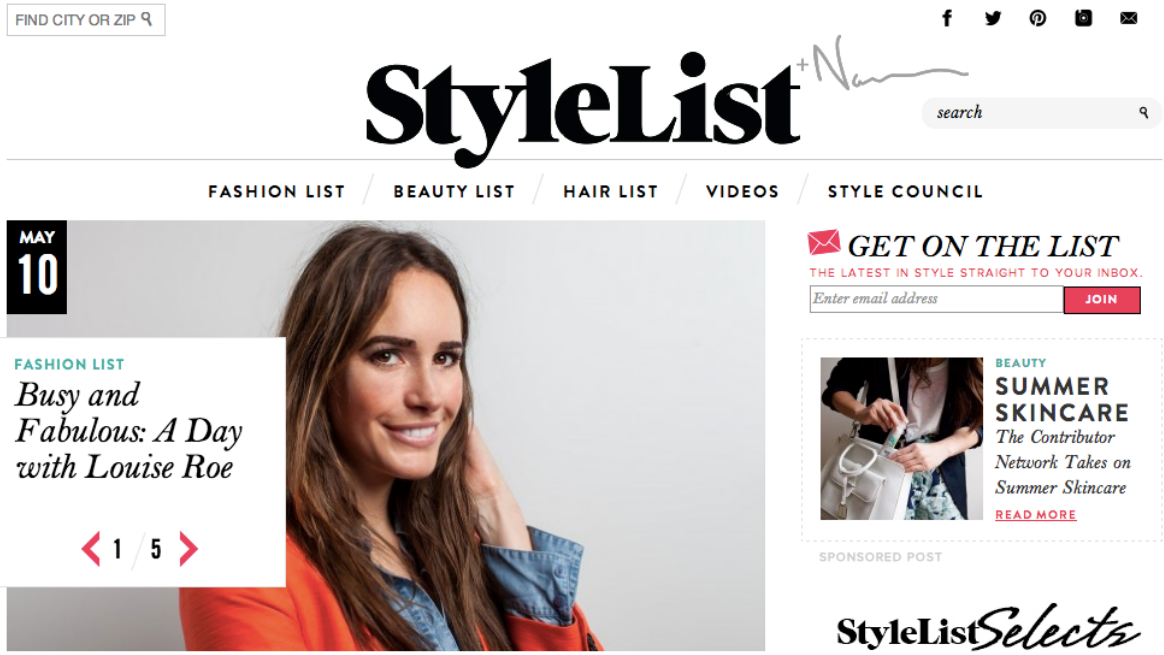
https://developers.google.com/chrome/mobile/docs/data-compression

Aol. + mod_pagespeed



gadling.com

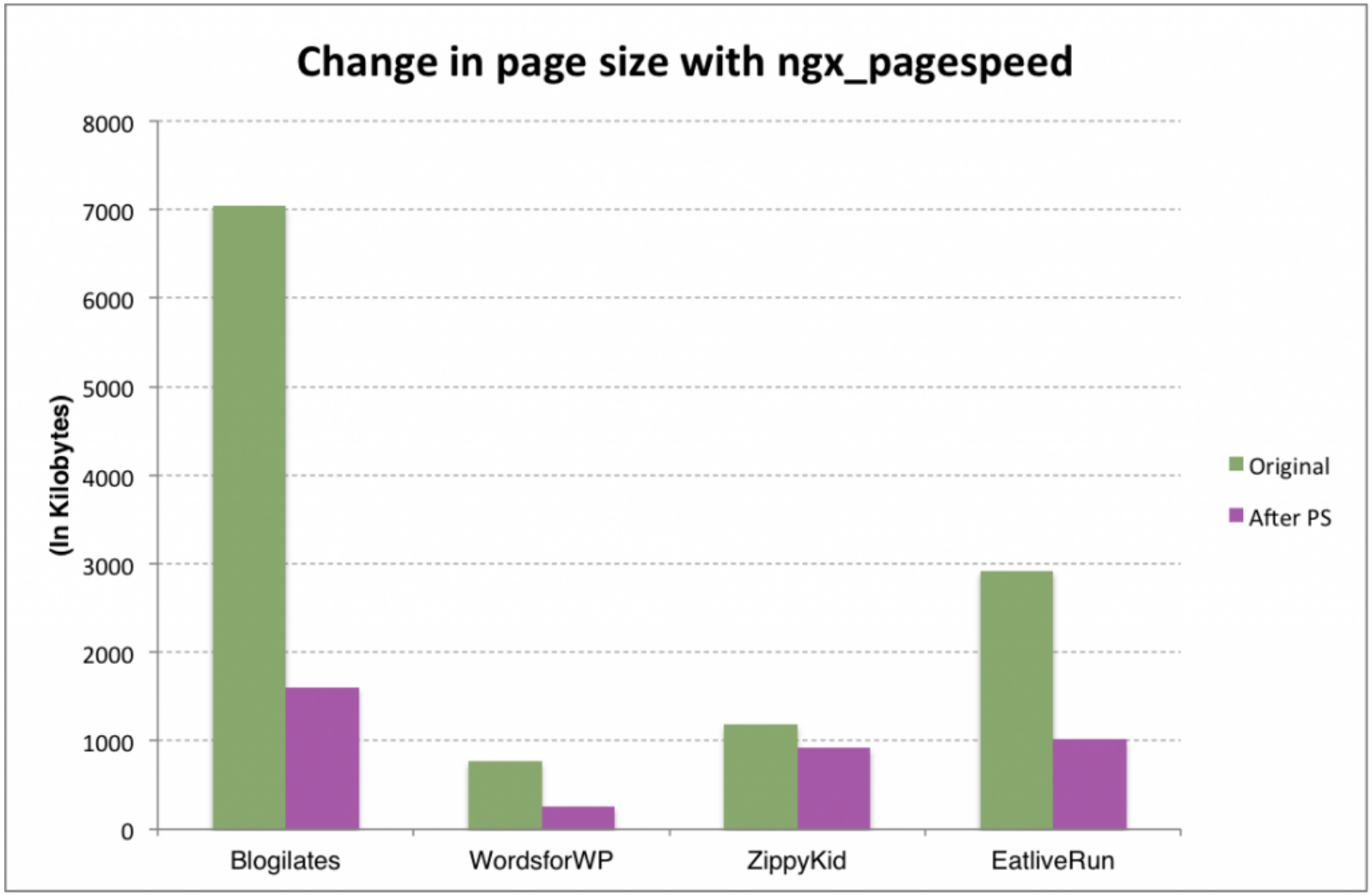
40% PLT improvement!



stylelist.com

20% PLT improvement!

zippykid + ngx_pagespeed



*"... 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...
  - **IISpeed** for Microsoft IIS server
  - PageSpeed for Apache Traffic Server

Wouldn't it be nice if...

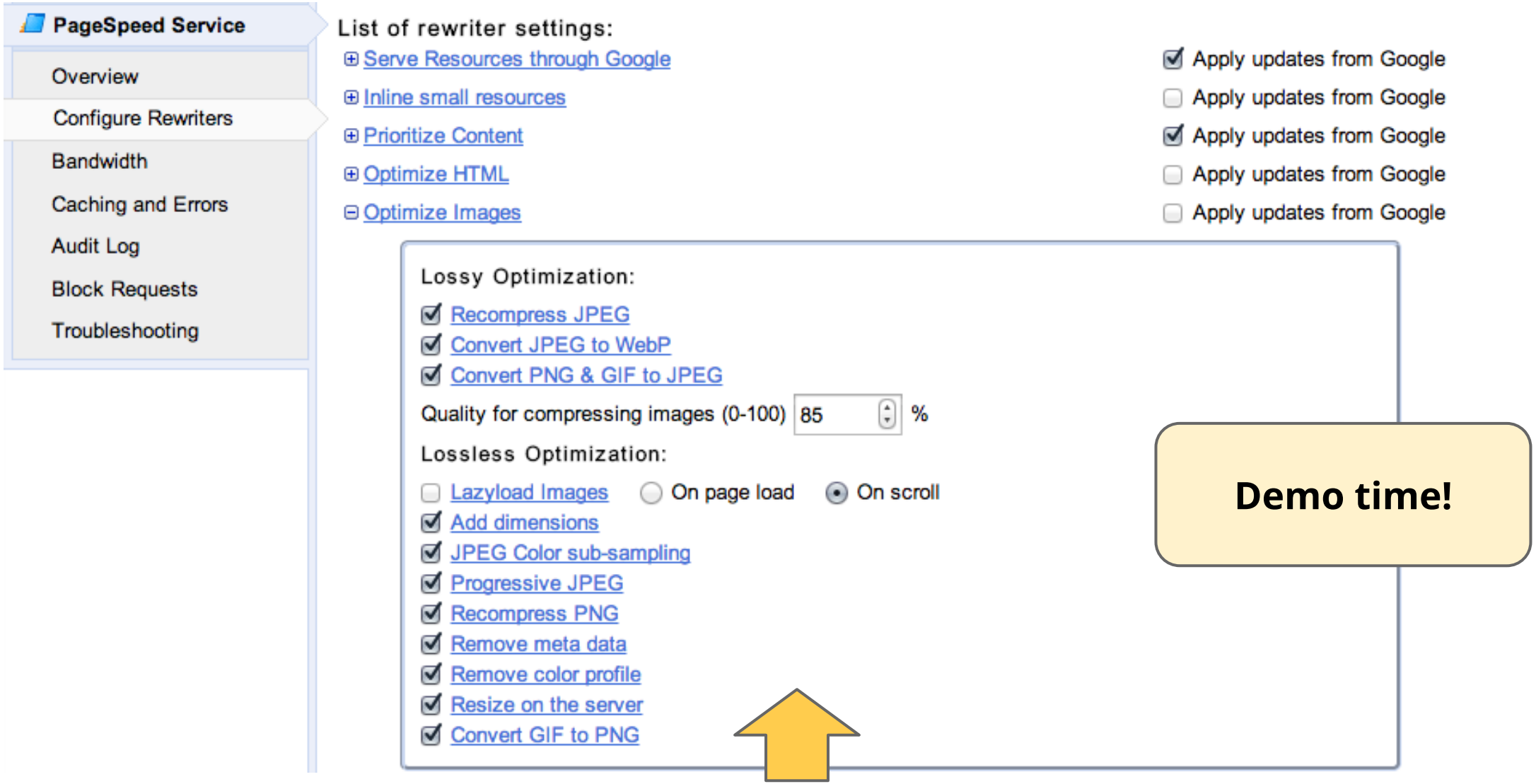
- *The optimization was done **automagically***
- *We didn't need to modify or update our servers*
- *And we had an **all-in-one solution** for...*
  - *Optimization, CDN, DoS protection, ...*

PageSpeed Service hosted by Google (BETA)

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!

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

file | 231 lines (183 sloc) | 6.094 kb

```
1  application: html5rocks-hrd
2  version: master
3  runtime: python27
4  threadsafe: yes
5  api_version: 1
6
```

```
22 pagespeed:
23   enabled_rewriters:
24   - MinifyCss
25   - InlineImages
26   - CollapseWhitespace
27   - ImageAddDimensions
28   - RemoveComments
```

Google Analytics

313,495 of pageviews sent page load sample

Avg. Page Load Time (sec): -11.91%
7.33 vs 8.32

Avg. Redirection Time (sec): -11.66%
0.19 vs 0.21

Avg. Server Connection Time (sec): -30.41%
0.04 vs 0.06

Avg. Page Download Time (sec): -29.47%
0.34 vs 0.48

*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**

Video @ bit.ly/io-pagespeed

Fin. Questions?

+Ilya Grigorik
igrigorik@google.com