Website Acceleration with mod_pagespeed

Joshua Marantz
Google
June 15, 2011

@jmarantz
www.modpagespeed.com
© 2011 Google, Inc. All rights reserved.
Velocity 2011: Faster By Default
wget https://dl-ssl.google.com/dl/linux/direct/\mod-pagespeed-beta_current_amd64.deb

sudo dpkg -i mod-pagespeed-*.deb

sudo apt-get -f install

sudo /etc/init.d/apache2 restart
What is mod_pagespeed?

Automatically rewrite web sites to make them load and display faster

Delivered as an Open-source Apache Module

“Just In Time Compilation for the web”
Why aren’t all web-sites fast?

**SPEED**
- Inlined / sprited / minified resources
- Long cache lifetimes
- Exploit features in modern browsers
- Track latest WPO techniques

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

Solution: Use Automated Tools
Why use mod_pagespeed?

Optimize Your Web-Site without changing it

Do not change your content generation flow
Don’t change your HTML, CSS, or JavaScript
Don’t manually run compression on all your photos

Get a machine to do that for you.
What does mod_pagespeed do?

- Automatic Image Compression & Resizing
- Minify CSS, JavaScript and HTML
- Inline small images, CSS, and JavaScript
- Cache Extension
- CSS/Javascript Combining
- Domain Mapping
- Domain Sharding
Example: Combining multiple CSS files

```html
<head>
  <link rel="stylesheet" type="text/css" href="styles/yellow.css">
  <link rel="stylesheet" type="text/css" href="styles/blue.css">
  <link rel="stylesheet" type="text/css" href="styles/big.css">
  <link rel="stylesheet" type="text/css" href="styles/bold.css">
</head>
<body>
  <div class="blue yellow big bold">Hello, mod_pagespeed!</div>
</body>
```

md5 sum of combined CSS file

Combined file Served with 1-year TTL

Makes CDNs more effective
mod_pagespeed Integration Flow

code.google.com/p/modpagespeed

Source (svn) .rpm .deb

Existing Apache Httpd Server

Improved latency Reduced bandwidth Longer Cache with rapid changes

edit pagespeed.conf

Existing Content Generation Flow

Existing Content

No Changes to Content No Changes to Flow
mod_pagespeed Architecture

General Purpose Rewriting Engine

Rewriting Filters

HTML Parsing & Rewriting Framework

HTTP Caching & Header Manipulation

URL Parsing and Reconstruction

HTTP Resource Management

C++ STL jsmin libpng libjpeg zlib OpenCV Chromium PageSpeed Google CSS Parser

Apache Module Gasket

HTTP Content Cache

Metadata Cache

HTTP Fetch

Threading System

Logging

Configuration
Page Speed Automatic integration

Apache Module Gasket
modpagespeed.so

Experimental nginx Gasket
Open-source contribution

varnish Gasket
(looking for volunteers)

Proprietary Gaskets

Asynchronous HTTP Fetcher
Asynchronous Cache
mod_pagespeed Results:
How much faster is the web now?
mod_pagespeed adoption on the web

Launch Date: November 3, 2010
DreamHost support: November 5, 2010
Go Daddy support: January 27, 2011
Lines of C++: 60K + 350K
Number of Sites: 60K
mod_pagespeed site latency improvements: 28 sites

Sites with mod_pagespeed installed – median page-load time over 50 runs
Effectiveness of mod_pagespeed Image Optimizations

% Reduction in size (right is better)

Number of images

mod_pagespeed image size reduction on Alexa 1000
Page Speed Score Improvements via mod_pagespeed
androidacademy.com timeline (median of 50 runs)

First View Timeline (ms)

Repeat View Timeline (ms)
androidacademy.com waterfall from webpagetest.org

mod_pagespeed off

mod_pagespeed on

304 (not modified) responses
First View Load Time (ms) over 50 Samples

Load-Time Samples (ms)

- mps off
- mps on

Repeat View Load Time (ms) over 50 Samples

Load-Time Samples (ms)

- mps off
- mps on

Measurement variance, but conclusive results
Web Performance Anti-Patterns

• Vary: User-Agent : Prefer separate resource URLs.
• Incorrect Content-Type (internal to Apache)
• Incorrect Content-Encoding (internal to Apache)
• Incorrect DOCTYPE (XHTML vs HTML)
• Excessive Combining of resources
Lessons learned building mod_pagespeed

• Don’t Break The Web….Validate

• Don’t start the race 200ms behind: minimize TTFB

• Lots of Measurement – visually confirm stable & consistent results via scatter plots, statistics, Firebug…

• Load-Testing: *guaranteed* to be a learning experience
mod_pagespeed roadmap

• Make more of the web faster: 60,000 sites is a start…
  – High-volume partners (Hosting Providers, CDNs)
  – Leveraged integrations with servers, proxies, caches, etc
  – Extreme Support

• Make the web more fast
  – More User-Agent-sensitive optimization: Mobile, WebP
  – Image spriting
  – Deferring JavaScript execution

• Measure Better
  – Which filters have the highest impact?
  – Help users evaluate mod_pagespeed benefits
mod_pagespeed: Automatically Make Your Web Sites Faster

<table>
<thead>
<tr>
<th>Activity</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read About it</td>
<td><a href="http://code.google.com/speed/page-speed/docs/module.html">http://code.google.com/speed/page-speed/docs/module.html</a></td>
</tr>
<tr>
<td>Try it</td>
<td><a href="http://www.modpagespeed.com">http://www.modpagespeed.com</a></td>
</tr>
<tr>
<td>Download it</td>
<td><a href="http://code.google.com/speed/page-speed/download.html">http://code.google.com/speed/page-speed/download.html</a></td>
</tr>
<tr>
<td>Discuss it</td>
<td><a href="http://groups.google.com/group/mod-pagespeed-discuss">http://groups.google.com/group/mod-pagespeed-discuss</a></td>
</tr>
<tr>
<td>Build From source</td>
<td><a href="http://code.google.com/p/modpagespeed/wiki/HowToBuild">http://code.google.com/p/modpagespeed/wiki/HowToBuild</a></td>
</tr>
</tbody>
</table>
Google Booth Presentations

Wednesday, June 15, 2011

• 10:00 am – 10:10 am  Perfect Resource Caching OR Multiple iFrames, One Request
• 10:15 am – 10:25 am  TCP Fast Open and Loss Recovery
• 3:30 pm – 3:40 pm  Effectively Using WebPagetest to Analyze and Optimize a Site’s Performance
• 3:50 pm – 4:00 pm  How to Configure mod_pagespeed to Work Well on Your Site

Thursday, June 16, 2011

• 10:15 am – 10:25 am  Efficient DOM Manipulation with Declarative HTML Templates and AngularJS
• 3:30 pm – 3:40 pm  Building Tools with the Page Speed Online API
• 3:50 pm – 4:00 pm  http streaming