Dynamic Web Optimization Works!
Experiences with mod_pagespeed and Pagespeed Automatic

Jan-Willem Maessen
Software Engineer,
Pagespeed Automatic
jmaessen@google.com
Impact of Load Time on Effectiveness

- Amazon.com: -100 ms latency -> +1% revenue*
- Edmunds.com: -50% load time -> +17% page views per session**
- Stats***

---

* Source: Mashable
** Source: Edmunds.com Technology blog
*** Source: Strangeloop Networks
But It's a Balancing act

**SPEED**
- Inlined / sprited / minified resources
- Long cache lifetimes
- Exploit features in modern browsers
- Track latest optimization 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
Drop in usage in Apache

Wget \
https://dl-ssl.google.com/dl/linux/direct/\nmod-pagespeed-beta_current_amd64.deb

sudo dpkg -i mod-pagespeed-*_.deb

sudo apt-get -f install

sudo /etc/init.d/apache2 restart

Must not break the web
Effectiveness: androidacademy.com

http://www.youtube.com/watch?feature=player_profilepage&v=1FPUzUPnLR8
Webpagetest.org waterfall for Androidacademy.com

mod_pagespeed off

mod_pagespeed on

304 (not modified) responses
The problem of consistency: androidacademy.com

First View Load Time (ms) over 50 Samples

Repeat View Load Time (ms) over 50 Samples

Measurement variance, but conclusive results
extend_cache in action

Normal behavior with Cache-Control: 5 min
extend_cache in action

Normal behavior with Cache-Control: 1 year
If we edit a file, the user may take 1 year to notice!
Short cache lifetimes are used to ensure that users see new versions of content.
extend_cache in action

After extend_cache, resource urls contain a content hash. Updates are seen immediately.
Pagespeed Automatic Optimizations

- **Cache extension** (Sign URL with content hash, increase lifetime to 1 year)

- **CSS optimizations**
  - Outlining
  - Minification
  - Combining
  - Inlining
  - Move CSS to head
  - Sprite multiple background images

- **Image optimizations**
  - Re-encode
  - Resize based on context
  - Add dimensions to HTML
  - Inline using data: URLs
  - Convert gif → png, jpg → webp

- **Javascript optimizations**
  - Outlining
  - Minification
  - Combining (string constants + eval)
  - Inlining

- **HTML optimizations**
  - Combine multiple HEADs
  - Remove attribute quotes
  - Remove unused attributes
  - Remove comments
  - Left trim urls
Streaming Architecture

HTML rewrite path: no blocking fetches -- only rewrite resources in cache

Resource serving path: blocking fetches only when serving resources from a different server

Instaweb Apache

Browser

Rewritten HTML
Rewritten URLs
Rewritten Resources

Rewritten HTML

Rewriter1
Rewriter2

HTML Writer

HTML Event

HTML

CSS
JPEG
Javascript

Cache
File System
Evaluating mod_pagespeed 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
Page Speed Score improvements
Latency improvements on 28 sites

Sites with mod_pagespeed installed – median page-load time over 50 runs

First View
Repeat View
Do Not Break the Web

• Validate mod_pagespeed on thousands of web pages
• Assume that bugs will visibly alter at least one page
• Turn off optimizations that don't validate
• JavaScript can still break any optimization we perform

• Webmasters can explicitly enable unsafe optimizations
Looking Ahead

• Current optimizations are simple, safe, and effective
• Plenty more proxy optimizations in the literature
• Browser-dependent optimizations (Ex: webp, HTML5)
• Do all our browsing through a mod_pagespeed proxy
• Work even better with multi-server setups
Dynamic Web Optimization Works!

Thank you!