Speed up your web site

SEIDEN GROUP

seidengroup.com

Seiden Group and Club Seiden

Alan is a leader and expert in PHP on IBM i; leader, Zend's PHP Toolkit for IBM i: and "Performance guru of PHP on IBM i"

SEIDEN GROUP

Seiden Group is a team of experts available for mentoring/troubleshooting/project advice/development.

seidengroup.com, alan@seidengroup.com

Contact

Alan Seiden

alan@seidengroup.com

201-447-2437

www.SeidenGroup.com
twitter: @alanseiden

SEIDEN GROUP

Contact information

Alan Seiden

alan@alanseiden.com

201-447-2437

alanseiden.com
twitter: @alanseiden

Agenda for fast user experience

- **Fast user experience**
  - Beyond speed of PHP and server
- **Performance big picture**
- **Tools that show issues visually**
- **Tips and configurations**

Why I started to focus on web front end

- Clients called me in for performance help
- Assumed drag was on the server, PHP/DB2
- BUT many of the problems were in the front end (HTML, JS, CSS)

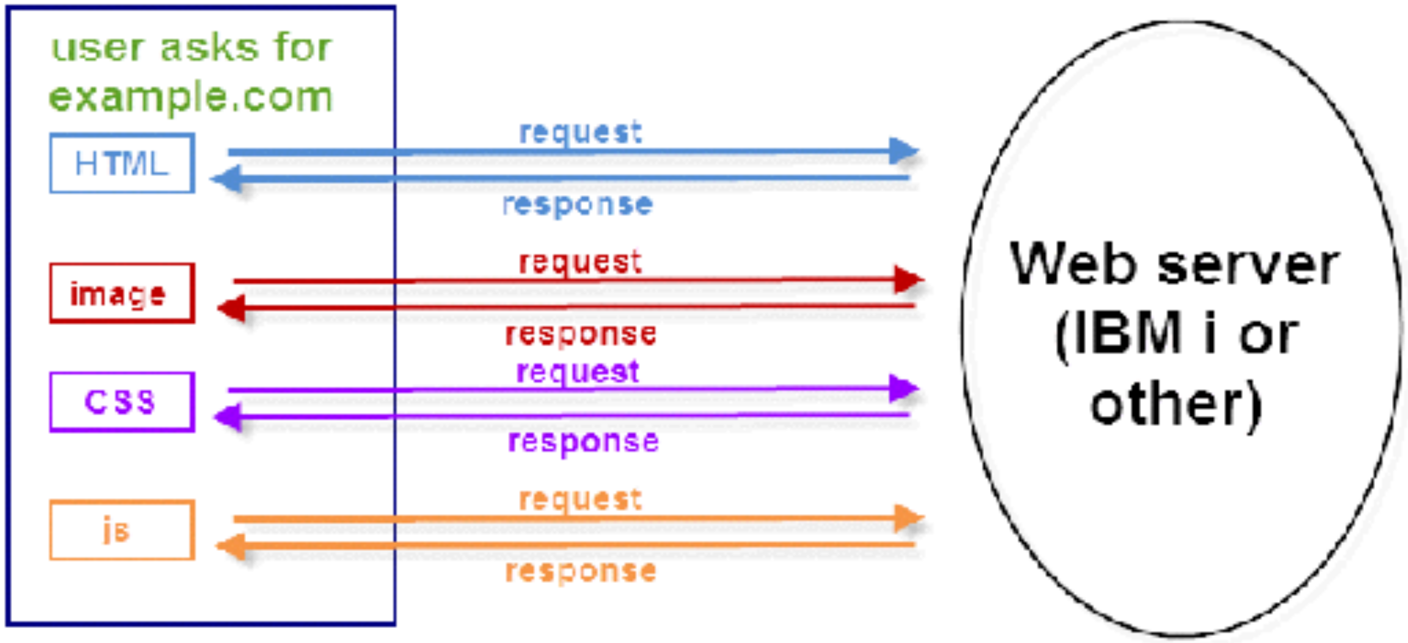
- Today's complex web and mobile applications require attention to front-end performance

Let's start with the basics

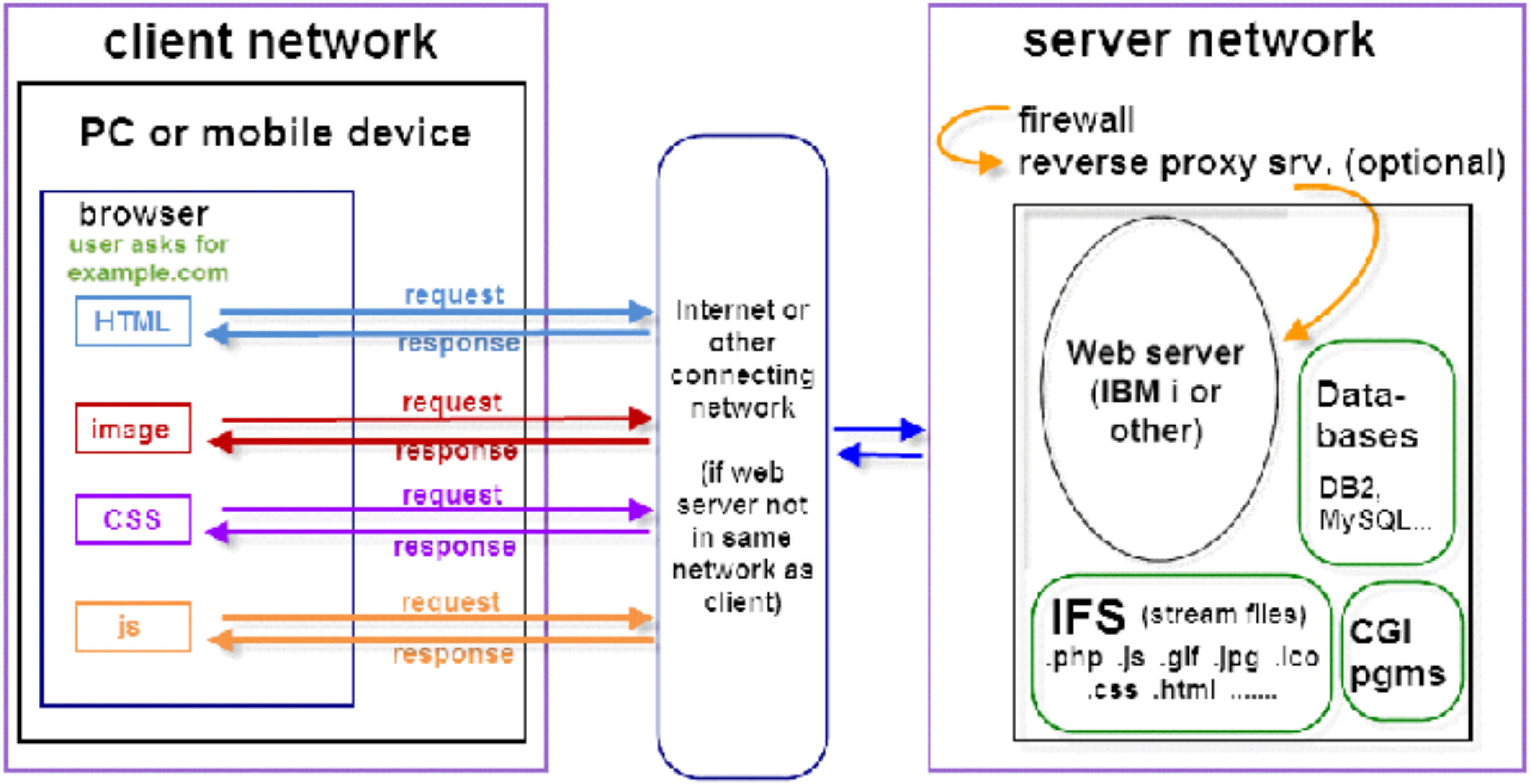
HTTP (web) flow

Request-response protocol

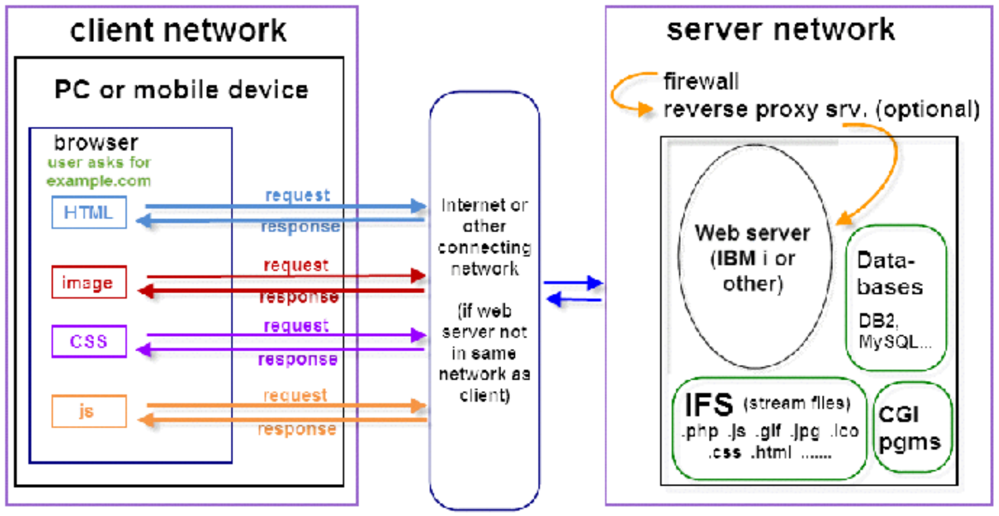
- Client (browser) requests a file; server responds
- One file at a time (at most 2–6 in parallel)
- Browser requests HTML file, then as it parses HTML, finds other file names to request (images, css, js...)



Each request passes through several layers


client network
PC or mobile device
browser
user asks for example.com
HTML
image
CSS
js
request
response
request
response
request
response
request
response
Internet or other connecting network
(if web server not in same network as client)
server network
firewall
reverse proxy srv. (optional)
Web server (IBM i or other)
Data-bases
DB2, MySQL...
IFS (stream files)
.php .js .gif .jpg .ico .css .html ....
CGI pgms

You might guess one top strategy


client network
PC or mobile device
browser
user asks for example.com
HTML
image
CSS
js
request
response
request
response
request
response
request
response
Internet or other connecting network
(if web server not in same network as client)
server network
firewall
reverse proxy srv. (optional)
Web server (IBM i or other)
Data-bases
DB2, MySQL...
IFS (stream files)
.php .js .gif .jpg .ico .css .html ......
CGI pgms

Each HTTP request travels through several layers

A common-sense performance strategy suggests itself

Reduce the number and size of HTTP requests

Perceived speed

When users say app is slow

- Watch them using the application
- Is slow page response the major problem?
- Or does the application not match their workflow?

- Can you help users get their job done with fewer clicks?

Tips for perceived speed

- **Users want to feel successful in achieving their tasks. You can:**
  - Provide feedback and status information
  - Give users a fast path through common tasks
  - Reduce users' anxiety by clearly labeling page elements, buttons, links, etc., using their own terminology
  - Run slow tasks asynchronously so users can cancel if desired
- **Old but interesting study: http://www.uie.com/events/roadshow/articles/download_time/**
  - "...when people accomplish what they set out to do on a site, they perceive that site to be fast."
- **Let users know that "something is happening"**
  - The spinning "waiting" graphic still works

Reduce HTTP requests

HTTP requests are costly

- **Each request makes a round trip to server**
- **Each HTTP request consumes bandwidth and CPU**
- **In-network tests do not measure end-user performance outside the network**
  - Users could have unpredictable DSL or mobile connections
  - Firewalls and proxy servers may sit between the web server and end user
    - I’ve seen convoluted network configurations

Can caching help?

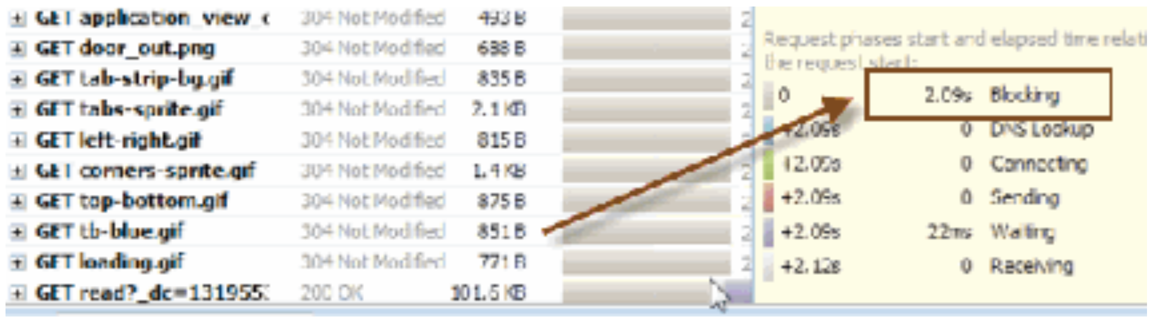
- **Browsers can cache most files**
- **Files won't have to be downloaded again till server has updated versions**

- **BUT browser must check for updates to each file**

- **Possible successful status codes:**
  - HTTP 200: Server delivered new, fresh version of file
  - HTTP 304: Server said "not modified." Use cached copy.
    - Faster, but still requres that request to check the file's timestamp

- **More about blocking and caching on next slide**

Requests cause “blocking” in browser

- Browsers typically limit themselves to 2–6 parallel requests to a given server
- File requests stack up, blocked by prev. requests
- 

```
GET application_view_c    304 Not Modified    493 B
GET door_out.png          304 Not Modified    688 B
GET tab-strip-bg.gif      304 Not Modified    835 B
GET tabs-sprite.gif       304 Not Modified    2.1 KB
GET left-right.gif        304 Not Modified    815 B
GET corners-sprite.gif    304 Not Modified    1.4 KB
GET top-bottom.gif        304 Not Modified    875 B
GET tb-blue.gif           304 Not Modified    851 B
GET loading.gif           304 Not Modified    771 B
GET read?_dc=131955       200 OK              101.5 KB
```

```
Request phases start and elapsed time relati
the request start:
0         2.09s   Blocking
+2.09s    0       DNS Lookup
+2.09s    0       Connecting
+2.09s    0       Sending
+2.09s    22ms    Waiting
+2.12s    0       Receiving
```

- Above, even “304 not modified” files caused blocking
- Solution: reduce number of images or improve caching via “Expires” headers
  - http://httpd.apache.org/docs/2.0/mod/mod_expires.html

Example: “Expires” headers (caching)

- **For aggressive caching, place these directives in Apache config file**
- **Can specify file types**

```
ExpiresActive On
# A2592000 means expire after a month in the client's cache
ExpiresByType text/css A2592000
ExpiresByType application/x-javascript A2592000
ExpiresByType application/javascript A2592000
ExpiresByType text/html A2592000
ExpiresByType image/png A2592000
ExpiresByType image/gif A2592000
ExpiresByType image/jpeg A2592000
```

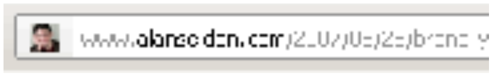
- **Many options:** http://httpd.apache.org/docs/2.0/mod/mod_expires.html

More ways to reduce “blocking”

- **If many .js or .css files are used:**
  - Combine them into fewer files
  - Move contents of smaller .js or .css files inline to your pages, eliminating those external files
  - Page Speed tool will help you decide

Create a favicon for your site

- **Browsers always look for a file called favicon.ico in your document root**
- **Those little icons that appear in the browser**



- **Once found, will be “remembered” by browser**
- **If not found, will be requested every time**
- **How to create a favicon:**
  - http://www.alanseiden.com/2007/05/25/brand-your-site-with-a-favicon/

Keep connections open

Keep HTTP connections alive

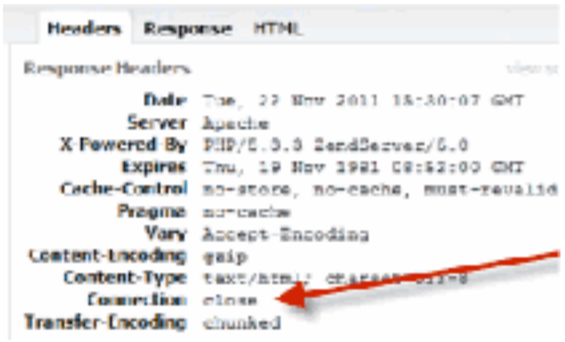
- **Enable “KeepAlive” setting in Apache**
- **The TCP connection will stay open, waiting for you**
  - Good when downloading many images, css, js files
  - You’ll reduce the number of three-way “handshakes” that establish a connection
  - Even more important with longer SSL handshakes

KeepAlive details

- **Configurable by number of seconds, number of files to be downloaded, before closing connection**
- **Recommended settings for average site**
  - KeepAlive On
  - KeepAliveTimeout 15
- **Details:**
  - http://httpd.apache.org/docs/2.0/mod/core.html#keepalive
- **Don't overdo it—you are locking out other users from that HTTP job while it's dedicated to you**

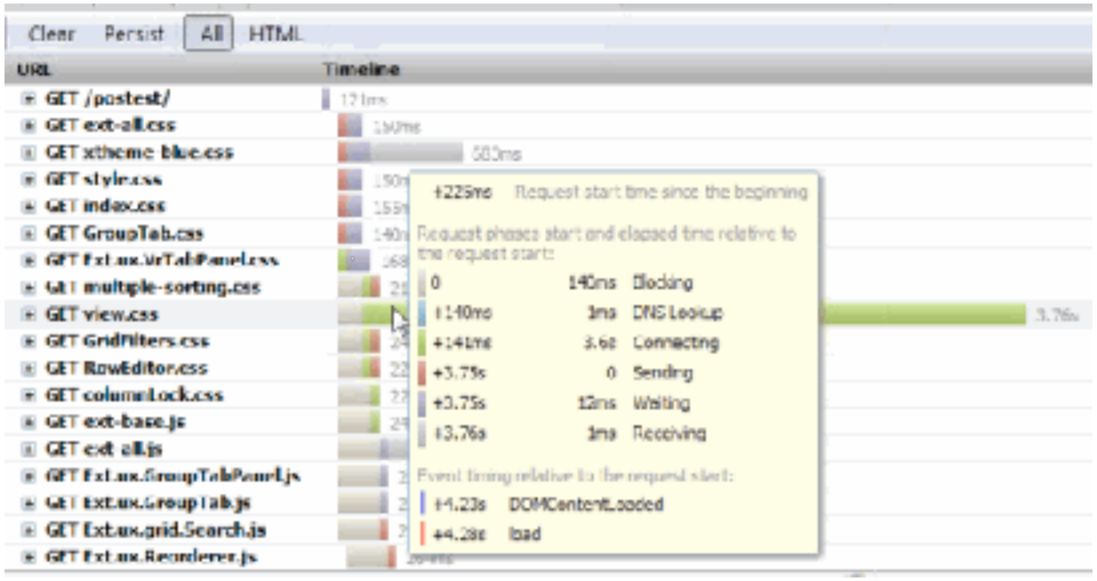
Connecting takes time

- **Clues that Keepalive is off**
  - “Connection: close”, “Connecting”
- **Example bottom right: 3.6 seconds “Connecting” (longer than average but it really happened)**



Response Headers

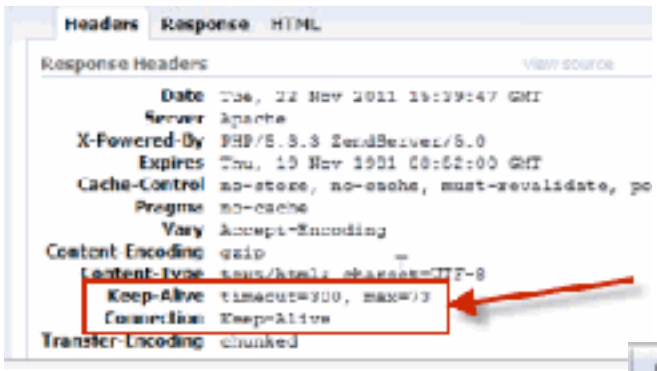
```
             Date  Tue, 22 Nov 2011 15:30:07 GMT
           Server  Apache
     X-Powered-By  PHP/5.3.3 ZendServer/5.0
          Expires  Thu, 19 Nov 1981 08:52:00 GMT
    Cache-Control  no-store, no-cache, must-revalid
           Pragma  no-cache
             Vary  Accept-Encoding
 Content-Encoding  gzip
     Content-Type  text/html; charset=utf-8
       Connection  close
Transfer-Encoding  chunked
```



```
URL                              Timeline
GET /postest/                    121ms
GET ext-all.css                  150ms
GET xtheme-blue.css              583ms
GET style.css                    150m
GET index.css                    155m
GET GroupTab.css                 140m
GET Ext.ux.VrTabPanel.css        168
GET multiple-sorting.css         21
GET view.css                     3.76s
GET GridFilters.css              24
GET RowEditor.css                22
GET columnLock.css               22
GET ext-base.js                  24
GET ext-all.js
GET Ext.ux.GroupTabPanel.js      2
GET Ext.ux.GroupTab.js           2
GET Ext.ux.grid.Search.js        2
GET Ext.ux.Reorderer.js          264ms

+225ms   Request start time since the beginning

Request phases start and elapsed time relative to the request start:
0           140ms   Blocking
+140ms        1ms   DNS Lookup
+141ms       3.6s   Connecting
+3.75s          0   Sending
+3.75s       12ms   Waiting
+3.76s        1ms   Receiving

Event timing relative to the request start:
+4.23s   DOMContentLoaded
+4.28s   load
```

What you see when Keep-alive is on

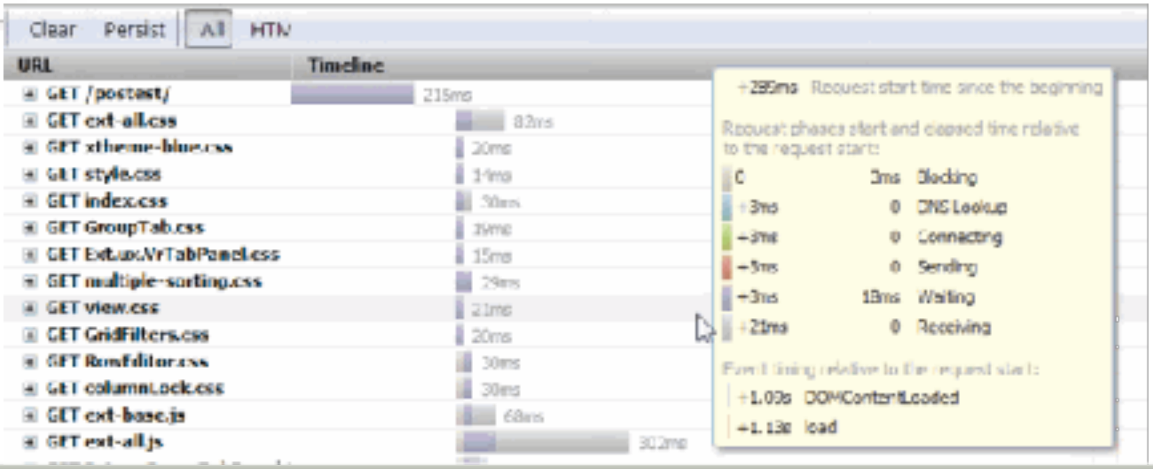
- **Firebug’s “Net” tab shows “Connection: Keep-Alive”, and, here, timeout=300 seconds (5 minutes)**



Headers | Response | HTML

Response Headers — View source

```
             Date  Tue, 22 Nov 2011 15:39:47 GMT
           Server  Apache
     X-Powered-By  PHP/5.3.3 ZendServer/5.0
          Expires  Thu, 19 Nov 1981 08:52:00 GMT
    Cache-Control  no-store, no-cache, must-revalidate, po
           Pragma  no-cache
             Vary  Accept-Encoding
 Content-Encoding  gzip
     Content-Type  text/html; charset=UTF-8
       Keep-Alive  timeout=300, max=73
       Connection  Keep-Alive
Transfer-Encoding  chunked
```



Clear | Persist | All | HTML

URL	Timeline
GET /postest/	215ms
GET ext-all.css	82ms
GET xtheme-blue.css	20ms
GET style.css	14ms
GET index.css	30ms
GET GroupTab.css	19ms
GET Ext.ux.VrTabPanel.css	15ms
GET multiple-sorting.css	29ms
GET view.css	21ms
GET GridFilters.css	20ms
GET RowEditor.css	30ms
GET columnLock.css	30ms
GET ext-base.js	68ms
GET ext-all.js	302ms

+285ms Request start time since the beginning

Request phases start and elapsed time relative to the request start:

0	3ms	Blocking
+3ms	0	DNS Lookup
+3ms	0	Connecting
+3ms	0	Sending
+3ms	18ms	Waiting
+21ms	0	Receiving

Event timing relative to the request start:

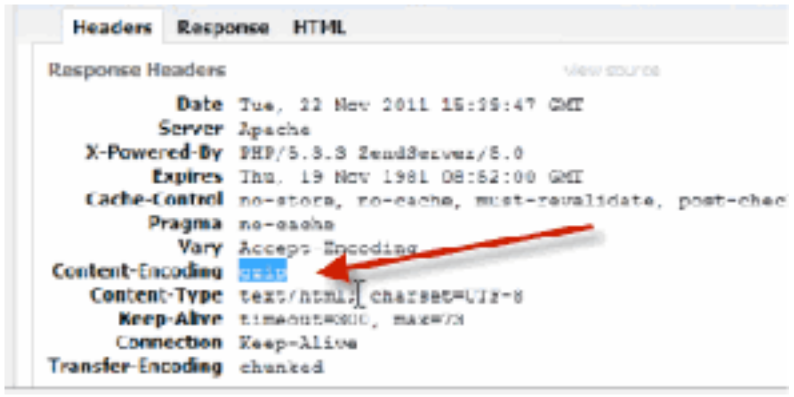
+1.03s	DOMContentLoaded
+1.13s	load

- **Zero seconds to connect**
- **Keep-alive is working!**

Use compression

Compression reduces file size

- Called gzip or mod_deflate, the same for our purposes



Response Headers

```
             Date Tue, 22 Nov 2011 15:55:47 GMT
           Server Apache
     X-Powered-By PHP/5.3.8 ZendServer/5.0
          Expires Thu, 19 Nov 1981 08:52:00 GMT
    Cache-Control no-store, no-cache, must-revalidate, post-chec
           Pragma no-cache
             Vary Accept-Encoding
 Content-Encoding gzip
     Content-Type text/html; charset=UTF-8
       Keep-Alive timeout=[illegible], max=[illegible]
       Connection Keep-Alive
Transfer-Encoding chunked
```

- Compresses, speeds up html, javascript, css, favicons, anything text-based

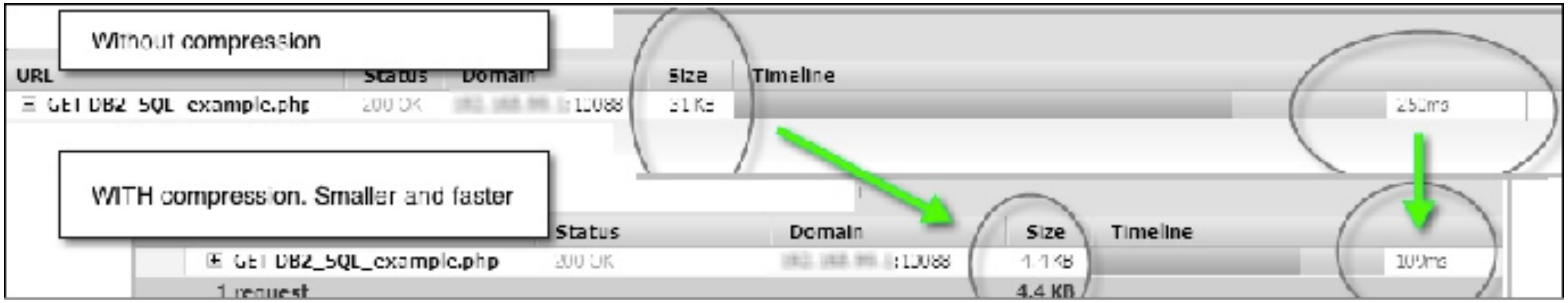
Netflix improved with gzip/deflate

- **Saw 13-25% performance improvement**
- **Cut outbound traffic in half**
  - That saves money for a busy site such as Netflix
- **Details:**
  - http://www.slideshare.net/billwscott/improving-netflix-performance-experience
- **It really works!**

My compression test

- http://your-server:10080/Samples/SQL_access/DB2_SQL_example.php

- Before compression: 31.0kb; loaded in 250ms
- After compression: 4.4kb; loaded in 109ms.
- That's 14% of the size and 50% of the time!



Without compression

URL	Status	Domain	Size	Timeline
⊞ GET DB2_SQL_example.php	200 OK	[redacted]:10088	31 KB	250ms

WITH compression. Smaller and faster

	Status	Domain	Size	Timeline
⊞ GET DB2_SQL_example.php	200 OK	[redacted]:10088	4.4 KB	109ms
1 request			4.4 KB	

Details of deflate/gzip compression

- **Apache directives (sample)**

```
# Load IBM i's module that performs compression
LoadModule deflate_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM

# Specify content types to compress
AddOutputFilterByType DEFLATE application/x-httpd-php application/
   json text/css application/x-javascript application/javascript
   text/html
```

- **Tutorial on my blog:**
  - http://www.alanseiden.com/2010/08/13/maximize-zend-server-performance-with-apache-compression/
- **Apache reference:**
  - http://httpd.apache.org/docs/2.0/mod/mod_deflate.html

Ajax: friend or foe?

AJAX=Asynchronous Javascript And XML

- **AJAX updates parts of a page without reloading the whole page**
- **Not always XML. These days, JSON too**
- **Requests and file sizes are generally small**
- **Meant to bring SPEED to the web**

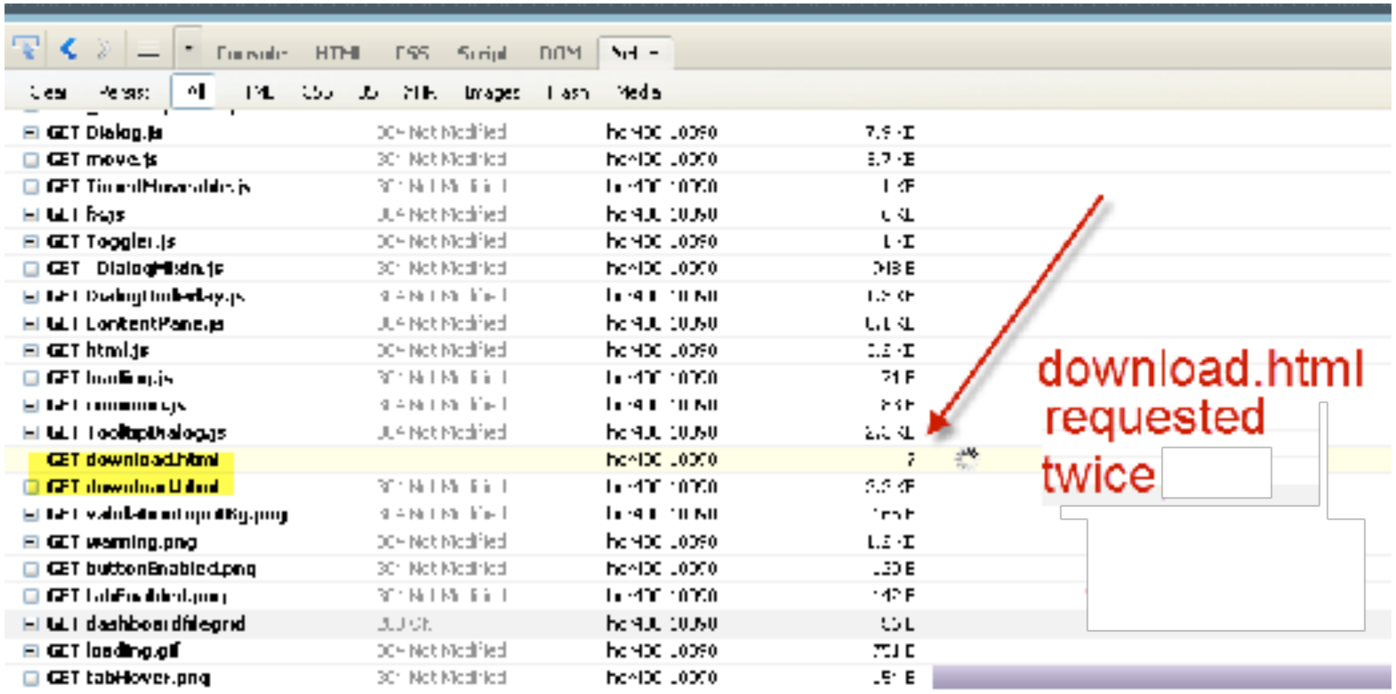
- **Potential problems if overused**

AJAX mistake #1

- **Too much of a good thing**
  - Requiring several AJAX requests to complete before the page itself can load fully
  - Too many HTTP requests at once
  - I’ve seen a situation where 4 AJAX requests were embedded in a page load
    - The AJAX doesn’t even start till the page loads
    - Causes “blocking” as the requests pile up, waiting for the previous ones to complete
    - Sessions may be shared by all AJAX calls, so locks can occur
  - Solution: when page first loads, embed AJAX content in the page
    - Re-use logic on the server side when building page
    - Subsequent updates can be done with AJAX

AJAX mistake #2

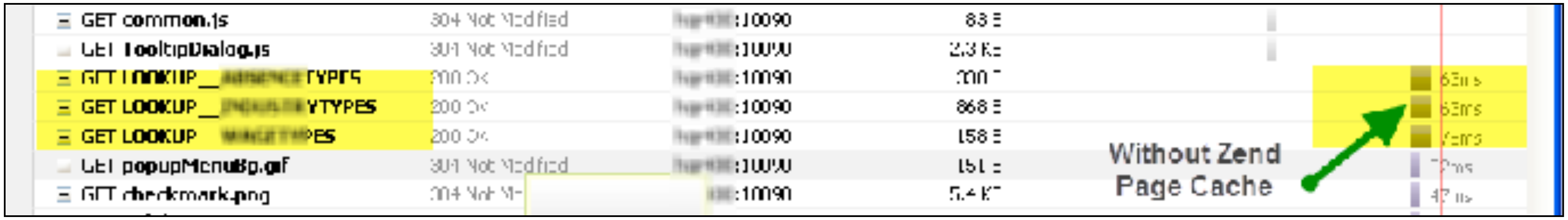
- **Duplicate requests**
  - Might go unnoticed with javascript library tools (Dojo, jQuery...)
  - Happens more than you would expect! Common, actually



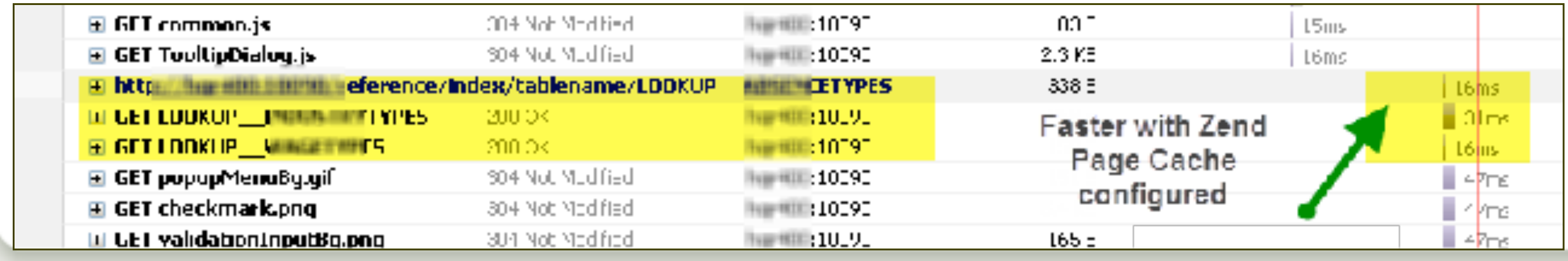
```
Console  HTML  CSS  Script  DOM  Net
Clear  Persist  All  HTML  CSS  JS  XHR  Images  Flash  Media
GET Dialog.js              304 Not Modified
GET move.js                304 Not Modified
GET [illegible]            [illegible]
GET [illegible]            304 Not Modified
GET Toggler.js             304 Not Modified
GET [illegible]            304 Not Modified
GET [illegible]            [illegible]
GET ContentPane.js         304 Not Modified
GET html.js                304 Not Modified
GET [illegible]            [illegible]
GET [illegible]            [illegible]
GET TooltipDialog.js       304 Not Modified
GET download.html
GET [illegible]            [illegible]
GET [illegible]            [illegible]
GET warning.png            304 Not Modified
GET buttonEnabled.png      304 Not Modified
GET [illegible]            [illegible]
GET dashboardfilegrid      [illegible]
GET loading.gif            304 Not Modified
GET tabHover.png           304 Not Modified
```

AJAX mistake #3

- **Dynamically generating static content (don't do that)**
  - Especially JSON to feed dropdown widgets
- **Solutions:**
  - Change to static files
  - Cache URLs (e.g. with Zend Page Cache if using PHP, or Apache caching) See example below, before and after caching
- **(Apologies for blurring: protecting confidentiality)**



```
GET common.js              304 Not Modified   [redacted]:10090    83 B
GET TooltipDialog.js       304 Not Modified   [redacted]:10090   2.3 KB
GET LOOKUP_[redacted]TYPES    200 OK             [redacted]:10090   200 B    63ms
GET LOOKUP_[redacted]YTYPES   200 OK             [redacted]:10090   868 B    63ms
GET LOOKUP_[redacted]PES      200 OK             [redacted]:10090   158 B    78ms
GET popupMenuBg.gif        304 Not Modified   [redacted]:10090   151 B    47ms
GET checkmark.png          304 Not M          [redacted]:10090   5.4 KB   47ms
```



```
GET common.js              304 Not Modified   [redacted]:10090    83 B    15ms
GET TooltipDialog.js       304 Not Modified   [redacted]:10090   2.3 KB   16ms
http[redacted]eference/index/tablename/LOOKUP_[redacted]CETYPES   338 B    16ms
GET LOOKUP_[redacted]TYPES    200 OK             [redacted]:10090            31ms
GET LOOKUP_[redacted]ES       200 OK             [redacted]:10090            16ms
GET popupMenuBg.gif        304 Not Modified   [redacted]:10090            47ms
GET checkmark.png          304 Not Modified   [redacted]:10090            47ms
GET validationInputBg.png  304 Not Modified   [redacted]:10090   165 B    47ms
```

Blocking from JS/CSS

Javascript is expensive for speed

- **Besides all the HTTP requests, JS must be parsed and run by your browser**
  - Even worse for mobile. Uses battery, CPU. Blocks UI
- **JS libraries (Dojo, jQuery) include dozens of JS files that you may not need**
  - Take a look with the tools shown later in this presentation. You may see 100+ JS files
  - Customize your JS library build to make distribution more compact
- **CSS (style sheets) are another area to examine. Cut down/consolidate if you can**

More tips for JS/CSS

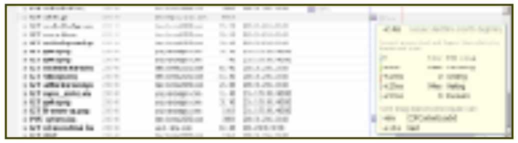
- **"Minify" if you can**
  - Strip out spaces/comments for production code
    - http://www.jsmini.com/
    - http://www.csscompressor.com/
    - Many other tools
  - Saves bandwidth; browser parses JS/CSS more quickly

- **Create a custom build of your JS library**
  - Tutorial to create custom build of jQuery
    - http://www.packtpub.com/article/building-a-custom-version-of-jquery

Live demos of front-end tools

Front-end tools demystify the web

- **Visualize HTTP requests**



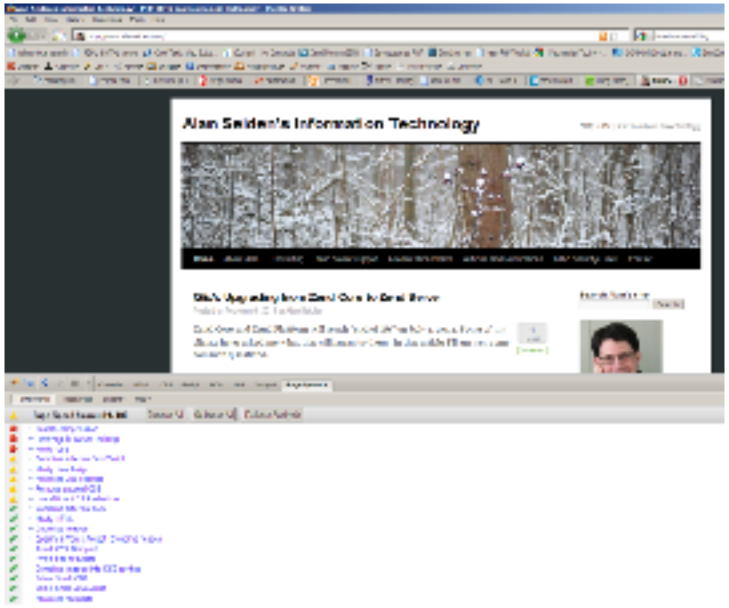
- **Find ways to eliminate requests or shrink responses**
- **Test more easily**
- **Capture "before and after" results**
  - For your own documentation
  - For a report to management

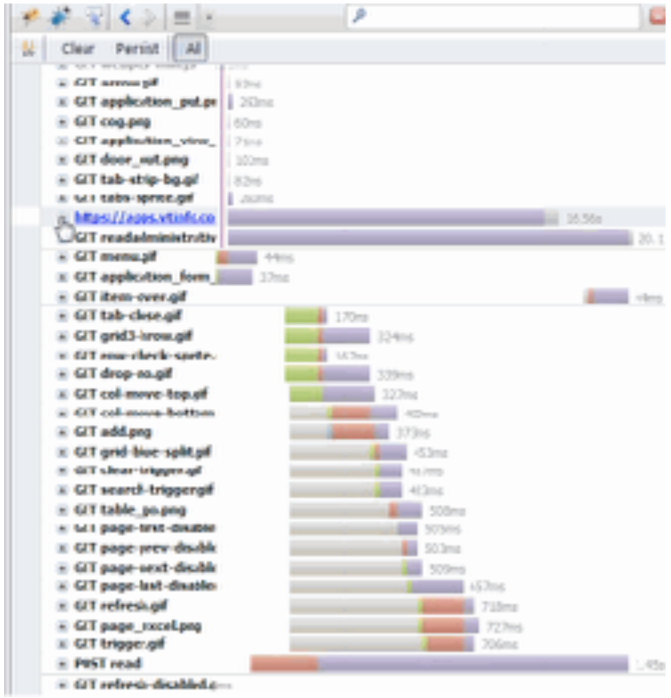
Favorite front-end performance tools

- **REDbot**
  - http://redbot.org
- **Firebug**
  - https://addons.mozilla.org/firefox/addon/firebug/
  - Even better with Page Speed add-on
    - http://code.google.com/speed/page-speed/
- **Page Speed Insights from Google**
  - https://developers.google.com/speed/pagespeed/insights
- **Web Page Test**
  - http://webpagetest.org

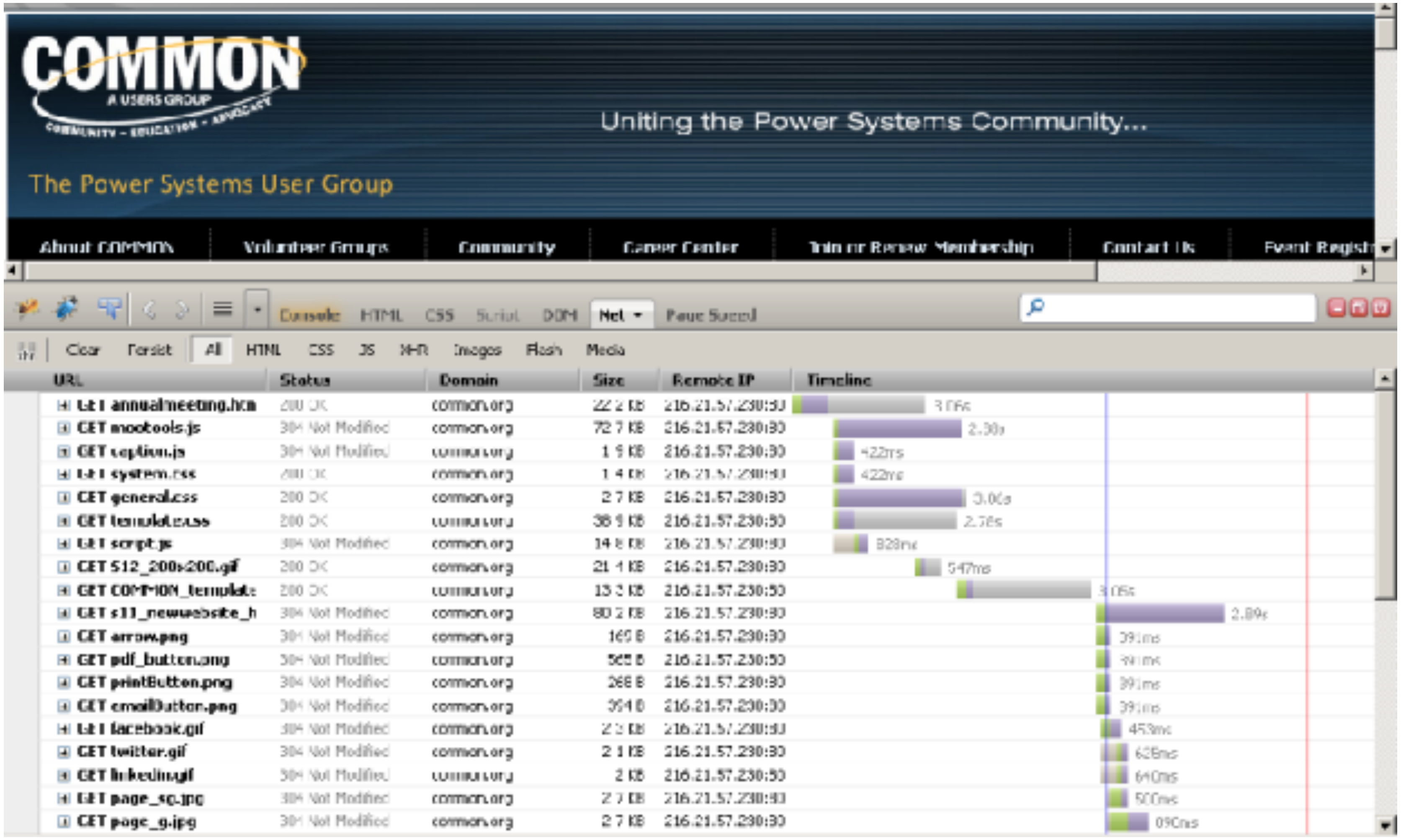
Firebug

- **Firebug**
  - https://addons.mozilla.org/firefox/addon/firebug/
  - Along with Page Speed, empowers anyone for performance



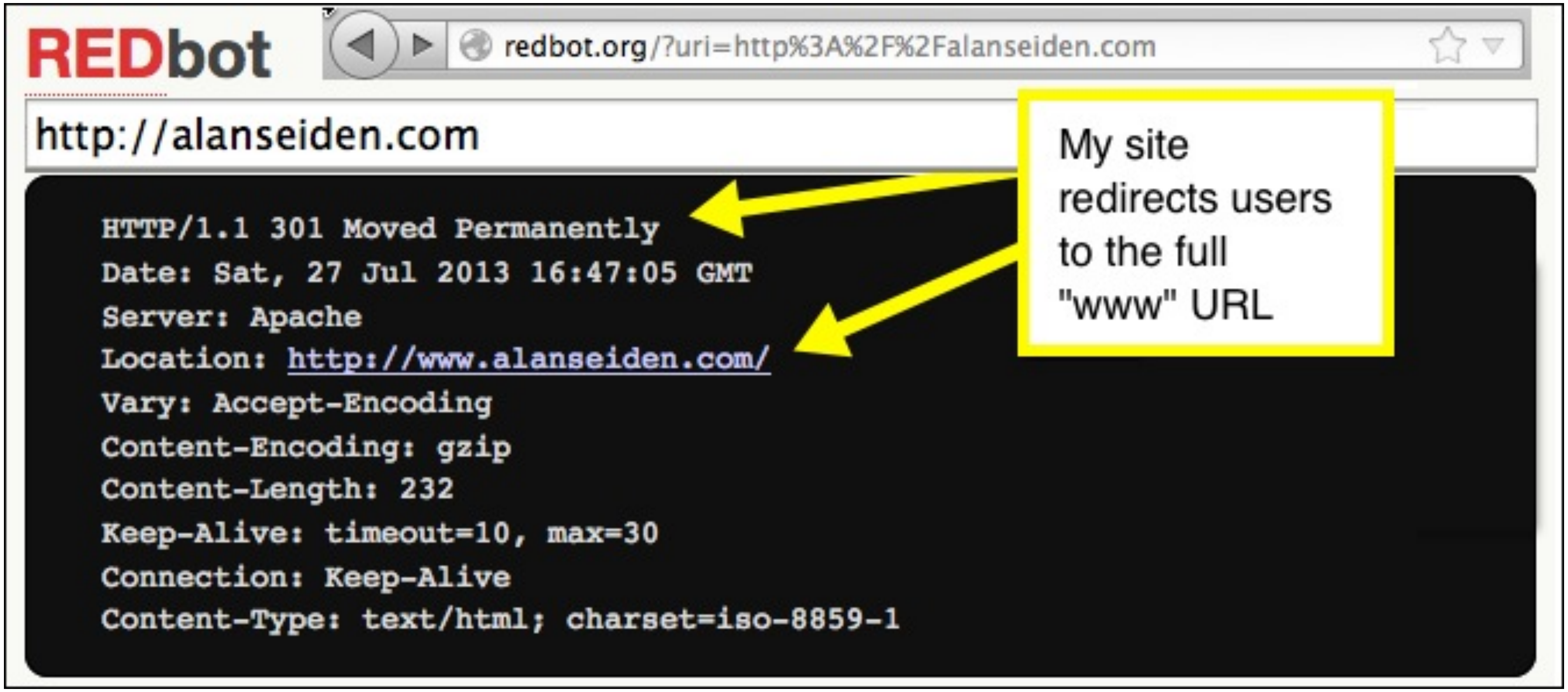


Firebug “Net” tab example



URL	Status	Domain	Size	Remote IP	Timeline
GET annualmeeting.htm	200 OK	common.org	22.2 KB	216.21.57.230:80	3.06s
GET mootools.js	304 Not Modified	common.org	72.7 KB	216.21.57.230:80	2.30s
GET caption.js	304 Not Modified	common.org	1.9 KB	216.21.57.230:80	422ms
GET system.css	200 OK	common.org	1.4 KB	216.21.57.230:80	422ms
GET general.css	200 OK	common.org	2.7 KB	216.21.57.230:80	3.06s
GET template.css	200 OK	common.org	38.9 KB	216.21.57.230:80	2.78s
GET script.js	304 Not Modified	common.org	14.8 KB	216.21.57.230:80	828ms
GET 512_200x200.gif	200 OK	common.org	21.4 KB	216.21.57.230:80	547ms
GET COMMON_template	200 OK	common.org	13.3 KB	216.21.57.230:80	3.06s
GET s11_newwebsite_h	304 Not Modified	common.org	80.2 KB	216.21.57.230:80	2.89s
GET arrow.png	304 Not Modified	common.org	169 B	216.21.57.230:80	391ms
GET pdf_button.png	304 Not Modified	common.org	565 B	216.21.57.230:80	391ms
GET printButton.png	304 Not Modified	common.org	268 B	216.21.57.230:80	391ms
GET emailButton.png	304 Not Modified	common.org	354 B	216.21.57.230:80	391ms
GET facebook.gif	304 Not Modified	common.org	2.3 KB	216.21.57.230:80	453ms
GET twitter.gif	304 Not Modified	common.org	2.1 KB	216.21.57.230:80	625ms
GET linkedin.gif	304 Not Modified	common.org	2 KB	216.21.57.230:80	640ms
GET page_sq.jpg	304 Not Modified	common.org	2.7 KB	216.21.57.230:80	500ms
GET page_g.jpg	304 Not Modified	common.org	2.7 KB	216.21.57.230:80	690ms

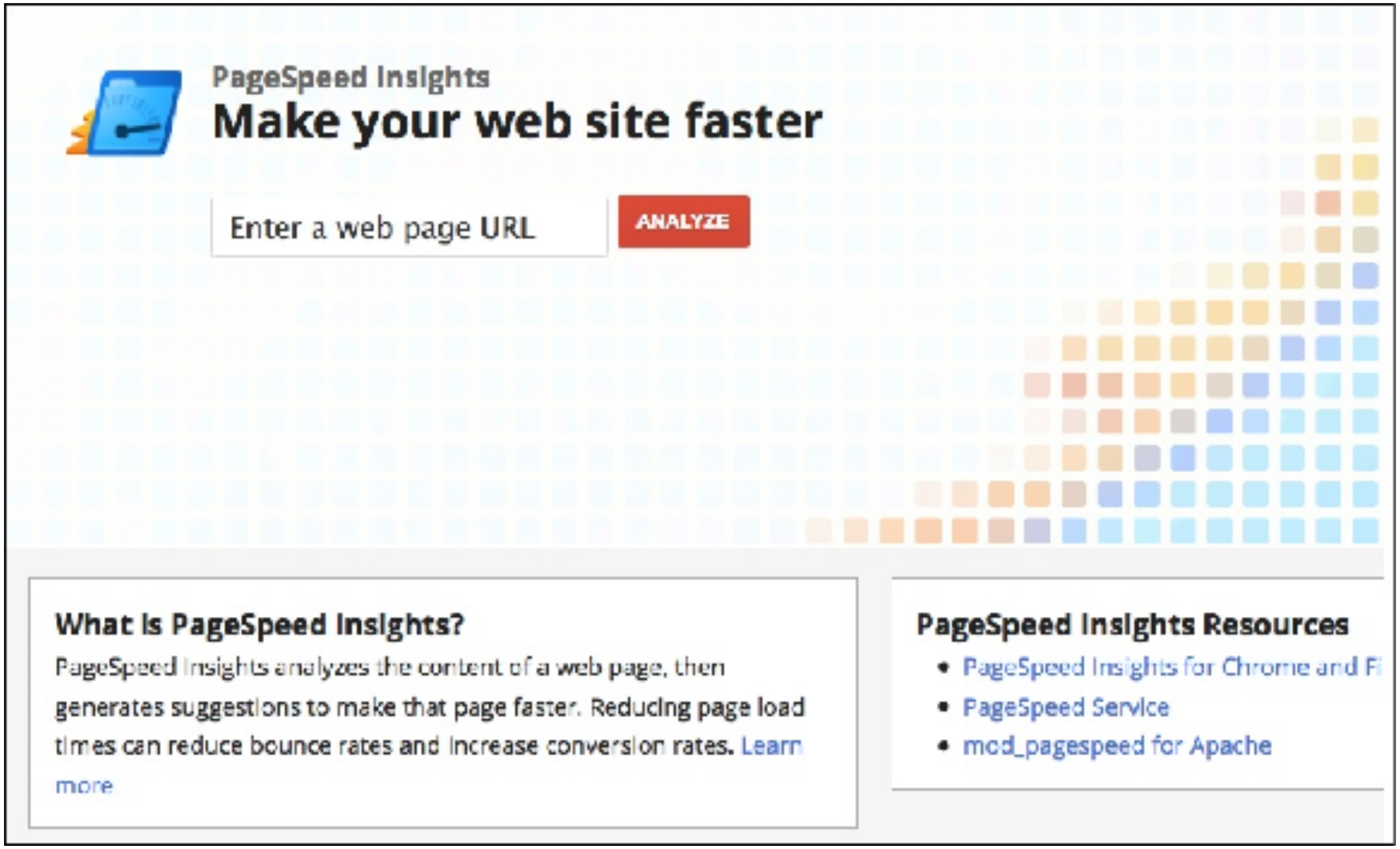
REDbot shows HTTP headers, codes


REDbot
redbot.org/?uri=http%3A%2F%2Falanseiden.com
http://alanseiden.com
My site redirects users to the full "www" URL

```
HTTP/1.1 301 Moved Permanently
Date: Sat, 27 Jul 2013 16:47:05 GMT
Server: Apache
Location: http://www.alanseiden.com/
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 232
Keep-Alive: timeout=10, max=30
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
```

- Visitors to alanseiden.com are redirected to www.alanseiden.com
- Although redirects can harm performance, this one (‘www’) helps search engines

Page Speed Insights by Google



PageSpeed Insights

Make your web site faster

Enter a web page URL

ANALYZE

What is PageSpeed Insights?

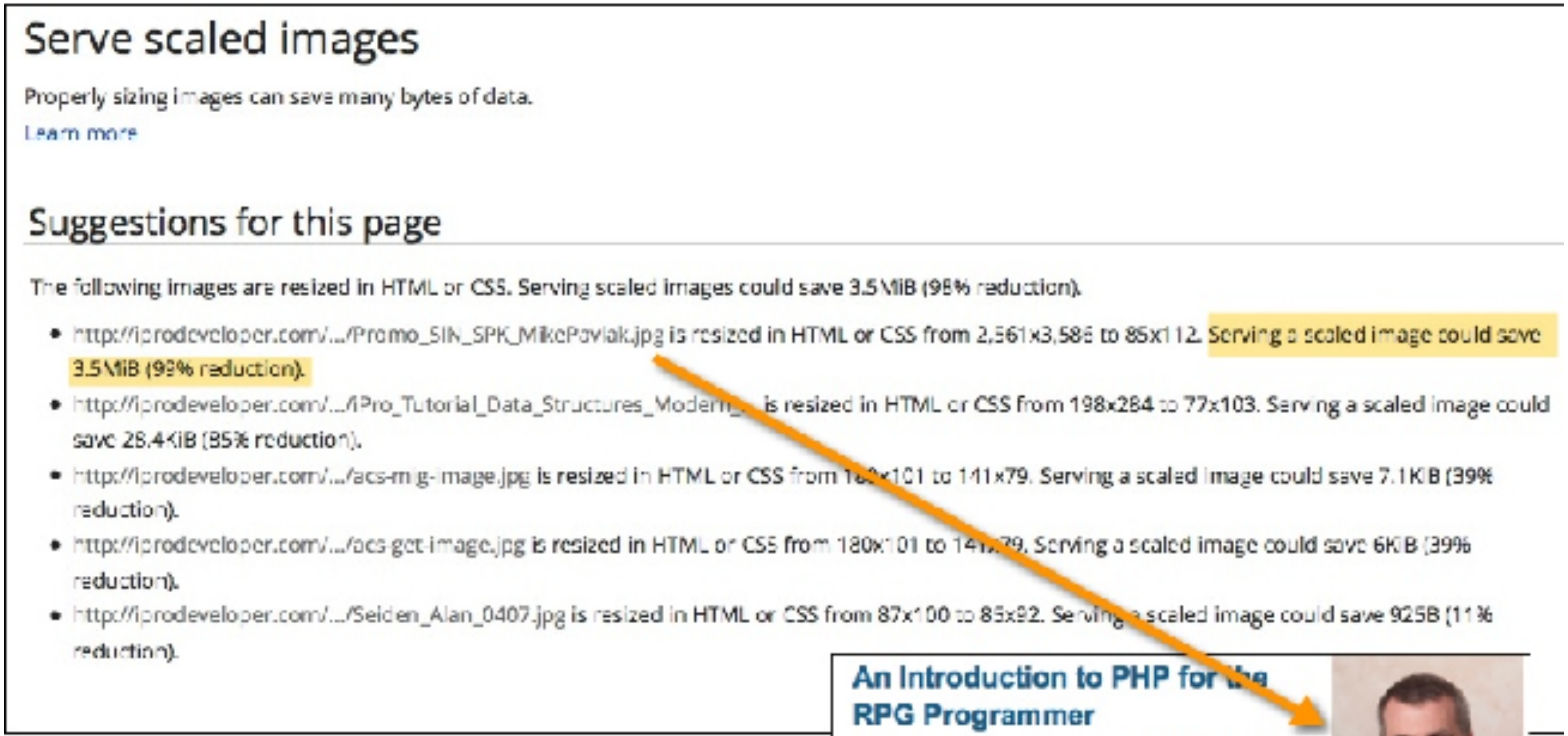
PageSpeed Insights analyzes the content of a web page, then generates suggestions to make that page faster. Reducing page load times can reduce bounce rates and increase conversion rates. Learn more

PageSpeed Insights Resources

- PageSpeed Insights for Chrome and Fi
- PageSpeed Service
- mod_pagespeed for Apache

https://developers.google.com/speed/pagespeed/insights

A tip from Page Speed Insights



Serve scaled images

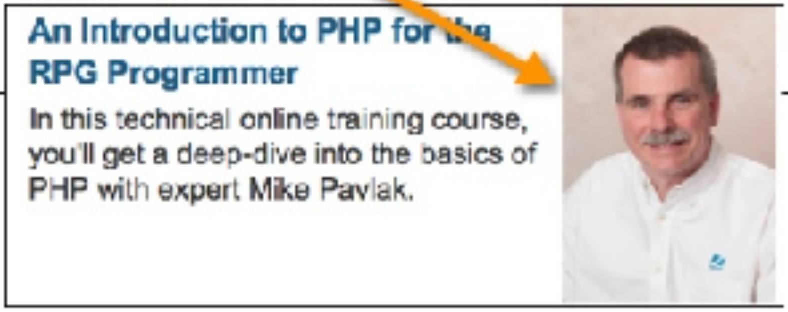
Properly sizing images can save many bytes of data.

Learn more

Suggestions for this page

The following images are resized in HTML or CSS. Serving scaled images could save 3.5MiB (98% reduction).

- http://iprodeveloper.com/.../Promo_SIN_SPK_MikePavlak.jpg is resized in HTML or CSS from 2,561x3,586 to 85x112. Serving a scaled image could save 3.5MiB (99% reduction).
- http://iprodeveloper.com/.../iPro_Tutorial_Data_Structures_Modern_... is resized in HTML or CSS from 198x284 to 77x103. Serving a scaled image could save 28.4KiB (85% reduction).
- http://iprodeveloper.com/.../acs-mig-image.jpg is resized in HTML or CSS from 180x101 to 141x79. Serving a scaled image could save 7.1KiB (39% reduction).
- http://iprodeveloper.com/.../acs-get-image.jpg is resized in HTML or CSS from 180x101 to 141x79. Serving a scaled image could save 6KiB (39% reduction).
- http://iprodeveloper.com/.../Seiden_Alan_0407.jpg is resized in HTML or CSS from 87x100 to 85x92. Serving a scaled image could save 925B (11% reduction).

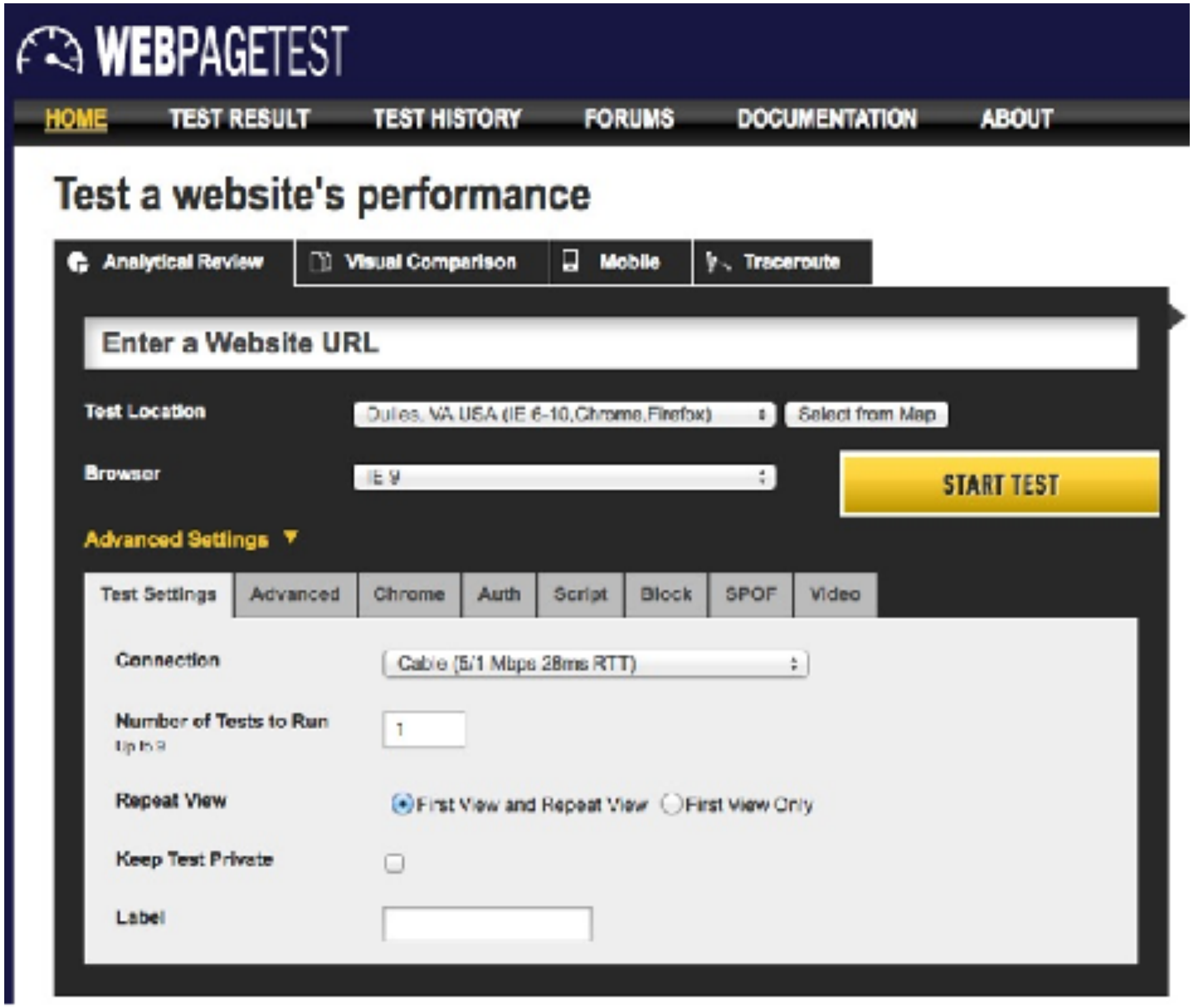


An Introduction to PHP for the RPG Programmer

In this technical online training course, you'll get a deep-dive into the basics of PHP with expert Mike Pavlak.

A large headshot was scaled to a small size. Better to use a smaller photo.

Web Page Test (webpagetest.org)

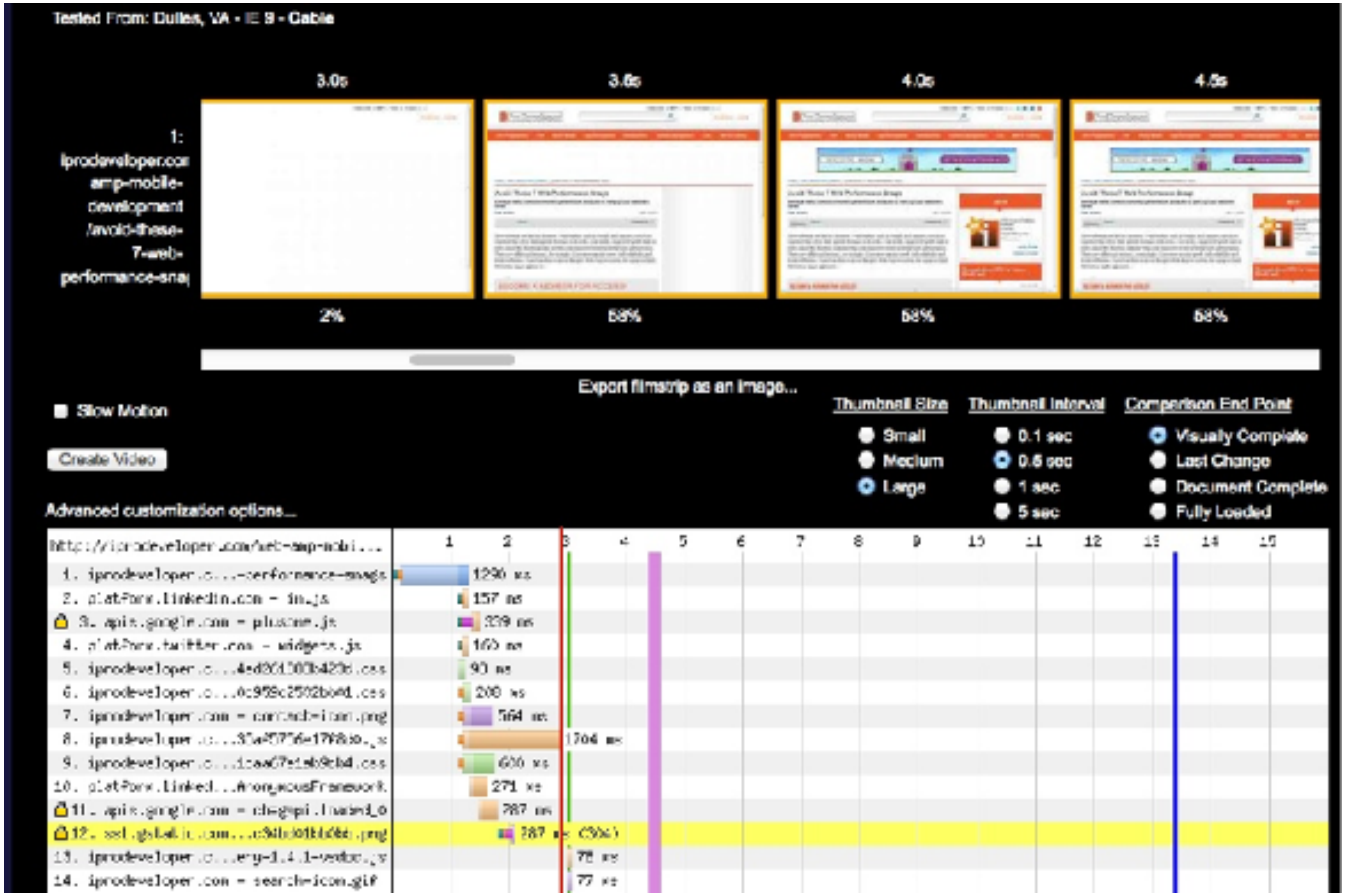


Test a website's performance

Enter a Website URL

Test Location	Dulles, VA USA (IE 6-10,Chrome,Firefox)
Browser	IE 9
Connection	Cable (5/1 Mbps 28ms RTT)
Number of Tests to Run Up to 9	1
Repeat View	[✓ First View and Repeat View] First View Only
Keep Test Private	☐
Label	

Webpagetest “Video/filmstrip” view



Tested From: Dulles, VA - IE 9 - Cable

1: iprodeveloper.com/amp-mobile-development/avoid-these-7-web-performance-sna

	3.0s	3.5s	4.0s	4.5s
	2%	58%	58%	58%

Export filmstrip as an image...

Slow Motion

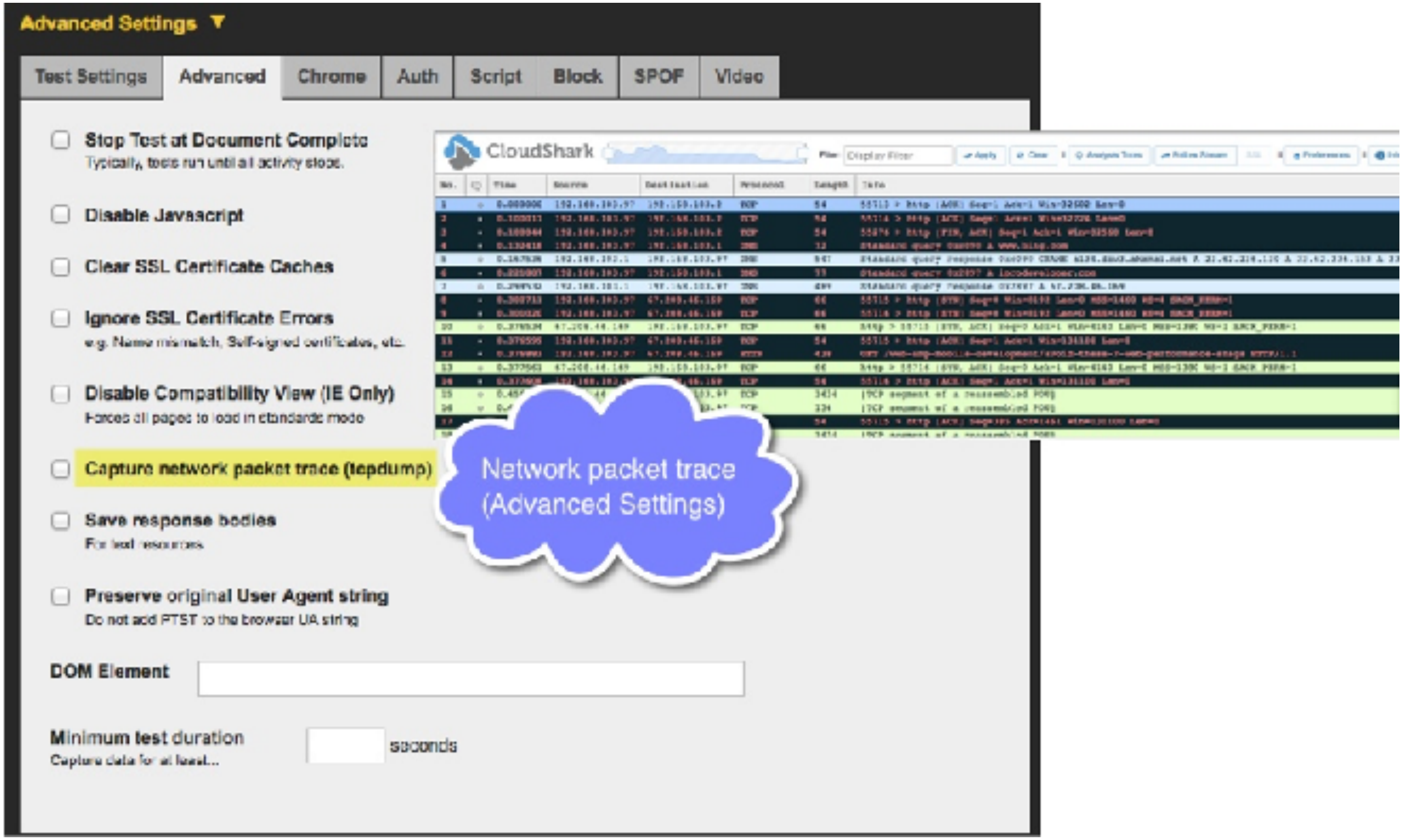
Create Video

Thumbnail Size	Thumbnail Interval	Comparison End Point
Small	0.1 sec	[✓ Visually Complete]
Medium	[✓ 0.5 sec]	Last Change
[✓ Large]	1 sec	Document Complete
	5 sec	Fully Loaded

Advanced customization options...

```
http://iprodeveloper.com/web-amp-mobi...
1. iprodeveloper.c...-performance-snaps      1290 ms
2. platform.linkedin.com - in.js             157 ms
3. apis.google.com - plusone.js              339 ms
4. platform.twitter.com - widgets.js         160 ms
5. iprodeveloper.c...4ed3d1303b423d.css      90 ms
6. iprodeveloper.c...0c959c2502b0d1.css      200 ms
7. iprodeveloper.com - contact-icon.png      564 ms
8. iprodeveloper.c...35a5756e17f8b0.js       1704 ms
9. iprodeveloper.c...1caa67e1eb9cb4.css      600 ms
10. platform.linked...AnonymousFramework     271 ms
11. apis.google.com - cb=gapi.loaded_0       787 ms
12. ssl.gstatic.com...c9dbd01bbd6e.png       287 ms (304)
13. iprodeveloper.c...ery-1.4.1-vendor.js    78 ms
14. iprodeveloper.com - search-icon.gif      77 ms
```

Advanced Settings of webpagetest



- [ ] Stop Test at Document Complete
  Typically, tests run until all activity stops.
- [ ] Disable Javascript
- [ ] Clear SSL Certificate Caches
- [ ] Ignore SSL Certificate Errors
  e.g. Name mismatch, Self-signed certificates, etc.
- [ ] Disable Compatibility View (IE Only)
  Forces all pages to load in standards mode
- [ ] Capture network packet trace (tcpdump)
- [ ] Save response bodies
  For text resources
- [ ] Preserve original User Agent string
  Do not add PTST to the browser UA string

DOM Element

Minimum test duration
Capture data for at least... seconds

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.103.97	198.150.103.2	TCP	54	55713 > http [ACK] Seq=1 Ack=1 Win=32500 Len=0
2	0.100011	192.168.103.97	198.150.103.2	TCP	54	55714 > http [ACK] Seq=1 Ack=1 Win=32725 Len=0
3	0.100044	192.168.103.97	198.150.103.2	TCP	54	55876 > http [FIN, ACK] Seq=1 Ack=1 Win=32550 Len=0
4	0.132418	192.168.103.97	192.168.103.1	DNS	72	Standard query 0x0090 A www.bing.com
5	0.167838	192.168.103.1	192.168.103.97	DNS	547	Standard query response 0x0090 CNAME a134.dsw3.akamai.net A [illegible]
6	0.225007	192.168.103.97	192.168.103.1	DNS	77	Standard query 0x2097 A [illegible]
7	0.266932	192.168.103.1	192.168.103.97	DNS	499	Standard query response 0x7887 A [illegible]
8	0.300713	192.168.103.97	67.208.46.150	TCP	66	55715 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
9	0.300026	192.168.103.97	67.208.46.150	TCP	66	55716 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
10	0.375534	67.208.46.149	192.168.103.97	TCP	66	http > 55715 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=1 SACK_PERM=1
11	0.375595	192.168.103.97	67.208.46.150	TCP	54	55715 > http [ACK] Seq=1 Ack=1 Win=131100 Len=0
12	0.376003	192.168.103.97	67.208.46.150	HTTP	439	GET /web-and-mobile-development/[illegible] HTTP/1.1
13	0.377563	67.208.46.149	192.168.103.97	TCP	66	http > 55716 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1380 WS=1 SACK_PERM=1
14	0.377608	192.168.103.97	67.208.46.150	TCP	54	55716 > http [ACK] Seq=1 Ack=1 Win=131100 Len=0
15	0.45	[illegible]	[illegible]	TCP	1434	[TCP segment of a reassembled PDU]
16	0.4	[illegible]	[illegible]	TCP	334	[TCP segment of a reassembled PDU]
17	[illegible]	[illegible]	[illegible]	TCP	54	55715 > http [ACK] Seq=386 Ack=1461 Win=131100 Len=0
18	[illegible]	[illegible]	[illegible]	[illegible]	1434	[TCP segment of a reassembled PDU]

Keep front-end performance in mind

Remember...

- **To provide an speedy overall user experience, use front-end performance techniques, such as to:**
  - Reduce or shrink file sizes when you can
  - Use gzip/deflate
  - Enable keepalive (in moderation)
  - Use a favicon
  - Keep an eye on AJAX performance
- **Let Firebug, Web Page Test, and Page Speed Insights assist you**
- **Get help when you need it**
- **To keep learning, see “Resources” slide, coming right up**

Resources

Resources for front-end performance

- **“Avoid These 7 Web Performance Snags”**
  - Alan’s article from June 2013 (subscription to iProDeveloper required)
    - http://iprodeveloper.com/web-amp-mobile-development/avoid-these-7-web-performance-snags
- **Performance Calendar (front-end performance articles)**
  - http://calendar.perfplanet.com/
- **Meetup groups and conferences: live and remote**
  - http://web-performance.meetup.com/
  - http://velocityconf.com/
- **Steve Souders (formerly Yahoo!, now Google)**
  - http://stevesouders.com
  - @souders
  - Books: High Performance Web Sites, Even Faster Web Sites

Contact and tips

Alan Seiden

Seiden Group

Ho-Ho-Kus, NJ

Free newsletter:
http://seidengroup.com/tips

SEIDEN GROUP

alan@seidengroup.com • 201-447-2437 • twitter: @alanseiden