Let your PHP apps fly on IBM i: High Performance PHP

Part 3

Ensuring a fast user experience

www.SeidenGroup.com

Speaker: Alan Seiden

IBM CHAMPION

- **Principal at Seiden Group**
  - Mentor CIOs and development teams
  - Deliver modern technical solutions
- **Host and sponsor of <u>CIO Summit</u>**
- **Club Seiden: The next generation**
- **Open source advocate, contributor**

Seiden Group

- **Team of experts who:**
  - Mentor IBM i teams, IT Directors, CIOs
  - Consult on projects and process
  - Develop applications
  - Troubleshoot

    www.seidengroup.com

Club Seiden

Forum for younger developers with a passion for open source, IBM i and collaboration

Session 3 on peformance

Session 1: How fast? Find the bottlenecks!

(measure, measure, measure)

Session 2: Accelerate your application

(speed up PHP and the server back end)

Session 3: Ensuring a fast user experience *

(speed up the front end)

*** You are here**

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/RPG
- BUT many of the problems were in the front end (HTML, JS, CSS)

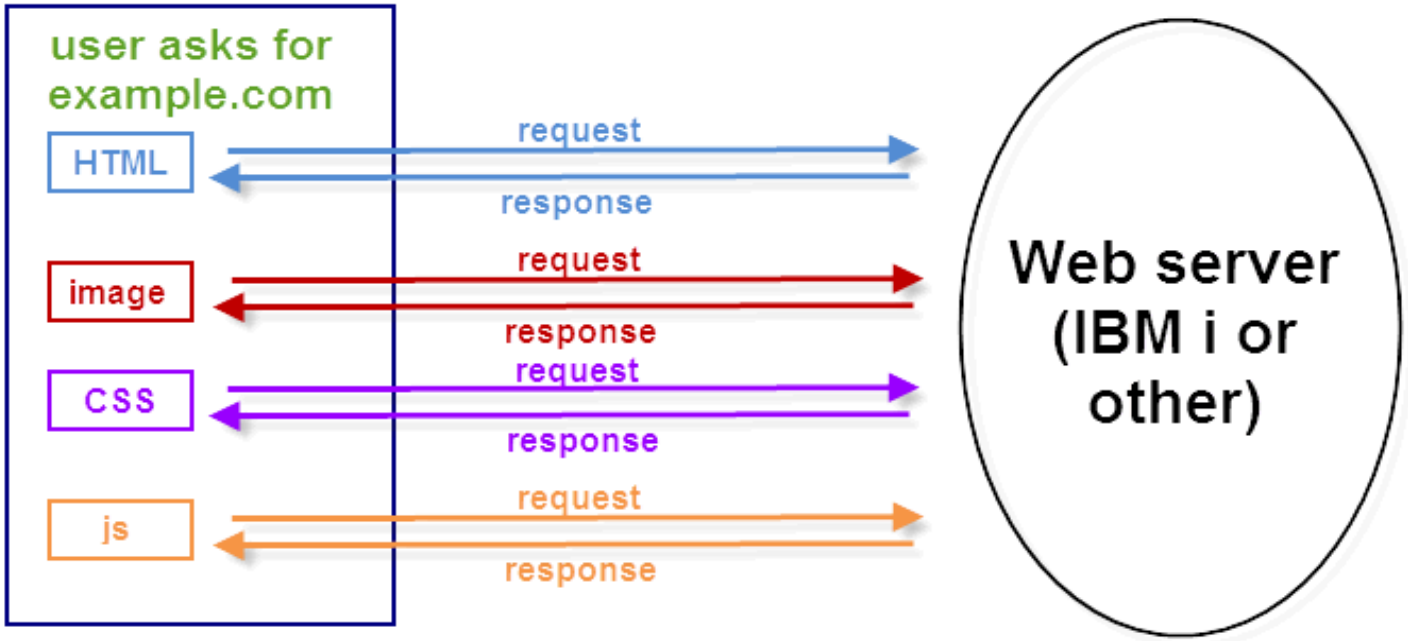
- Today's complex web and mobile applications require attention to front-end performance

Let's start with the basics

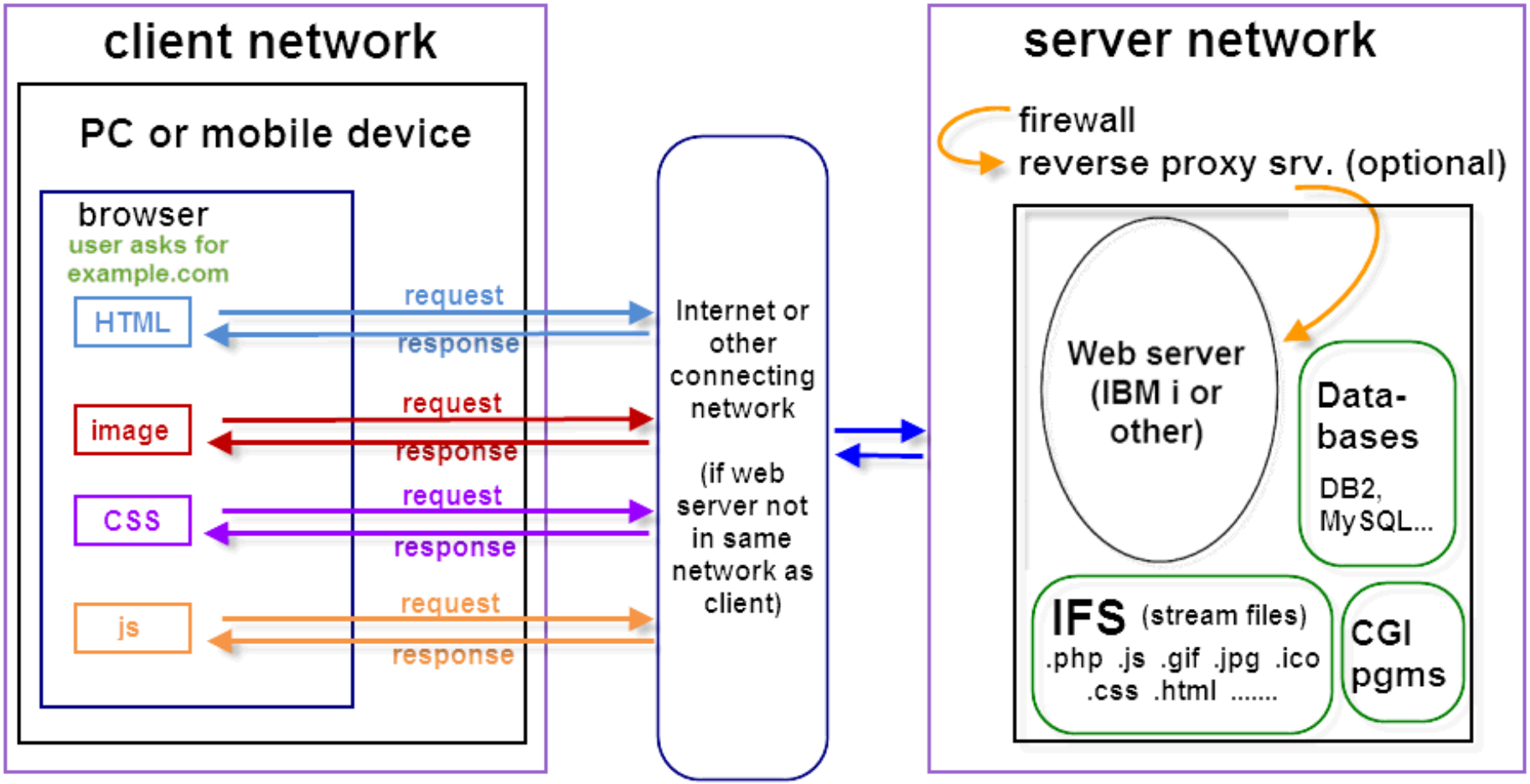
HTTP (web) flow

Request-response protocol

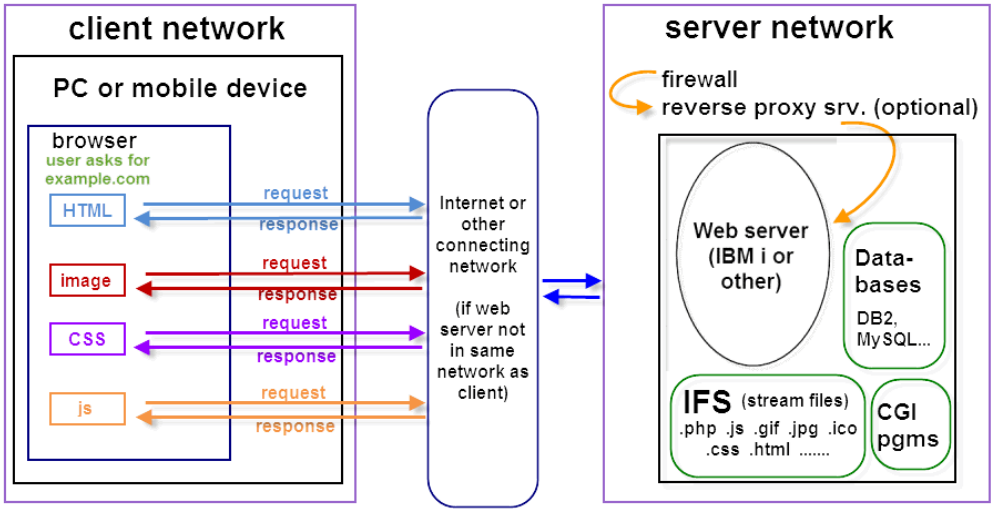
- Client (browser) requests a file; server responds
- One file at a time (at most 2–6 in parallel)
- Browser requests HTML file, then as it parses HTML, finds other file names to request (images, css, js...)



Each request passes through several layers



You might guess one top strategy



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

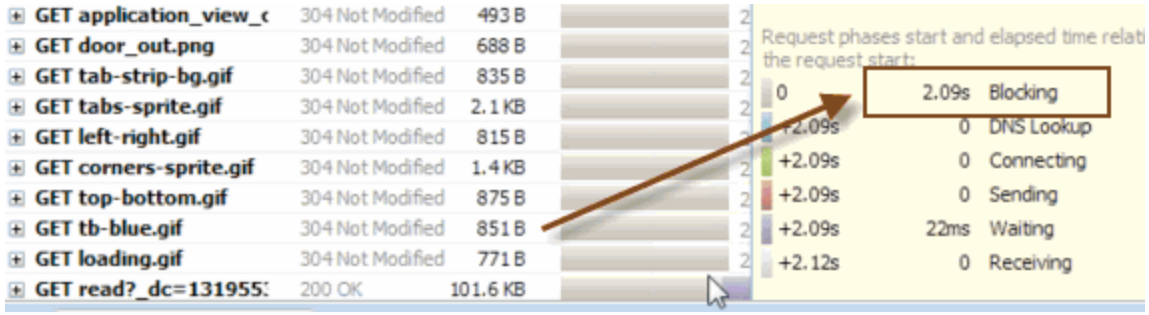
HTTP requests are costly

- **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**
- 

- **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:** httpd.apache.org/docs/current/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

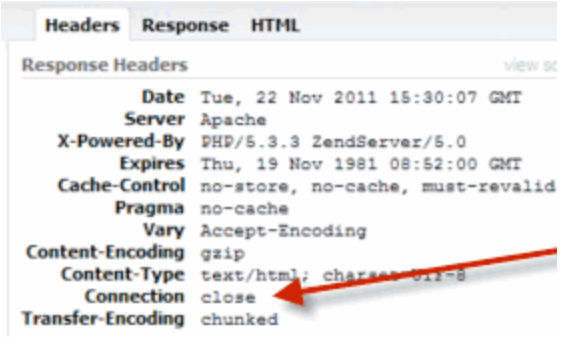
- **Enable “KeepAlive” setting in Apache**
- **The TCP connection will stay open, waiting for you**
  - Good when downloading many images, css, js files
  - You’ll reduce the number of three-way “handshakes” that establish a connection
  - Even more important with longer SSL handshakes

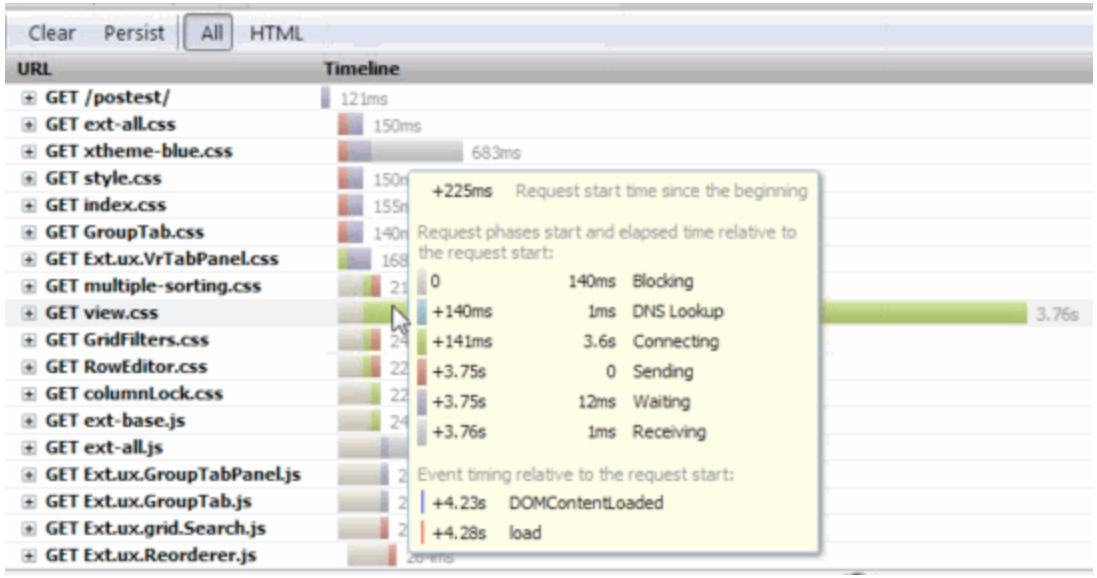
KeepAlive details

- **Configurable by number of seconds, number of files to be downloaded, before closing connection**
- **Recommended settings for average site**
  - KeepAlive On
  - KeepAliveTimeout 3
- **Details:**
  - https://httpd.apache.org/docs/2.4/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)**





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

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

Clear | Persist | All | HTM

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

+289ms 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

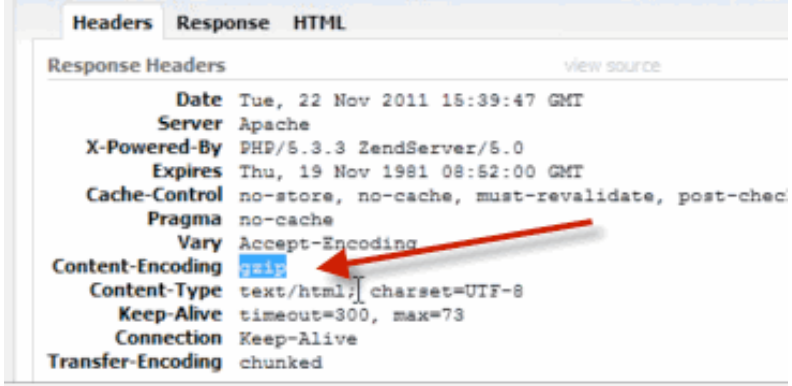
Event timing relative to the request start:

+1.09s	DOMContentLoaded
+1.13s	load

Use compression

Compression reduces file size

- **Called gzip or mod_deflate, the same for our purposes**



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

Netflix improved with gzip/deflate

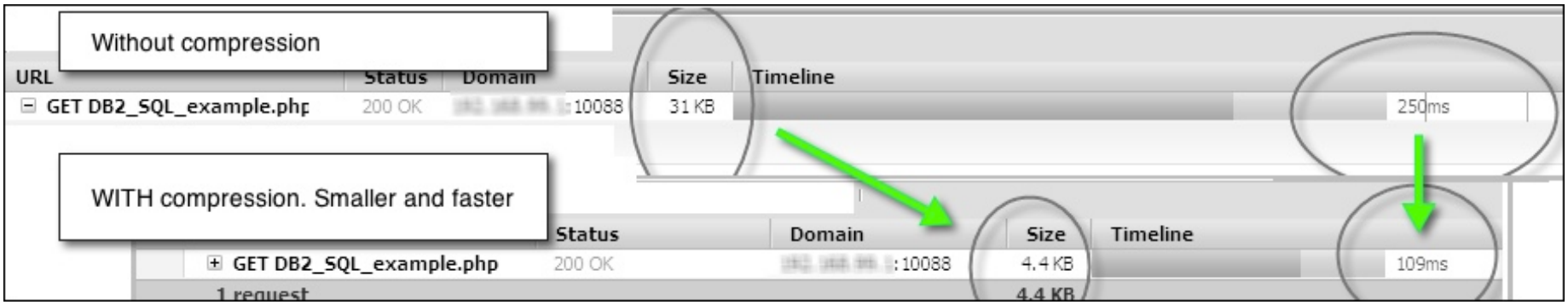
- **Saw 13-25% performance improvement**
- **Cut outbound traffic in half**
  - That saves money for a busy site such as Netflix
- **Details:**
  - http://www.slideshare.net/billwscott/improving-netflix-performance-experience

- **It really works!**
  - CNX IBM i case study: https://www.cnxcorp.com/blog/speed-your-valence-instances-apache-compression
    "jaw-dropping 68% reduction in the initial load time of newly created Valence apps"

My compression test

- http://your-server:10088/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!



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.seidengroup.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

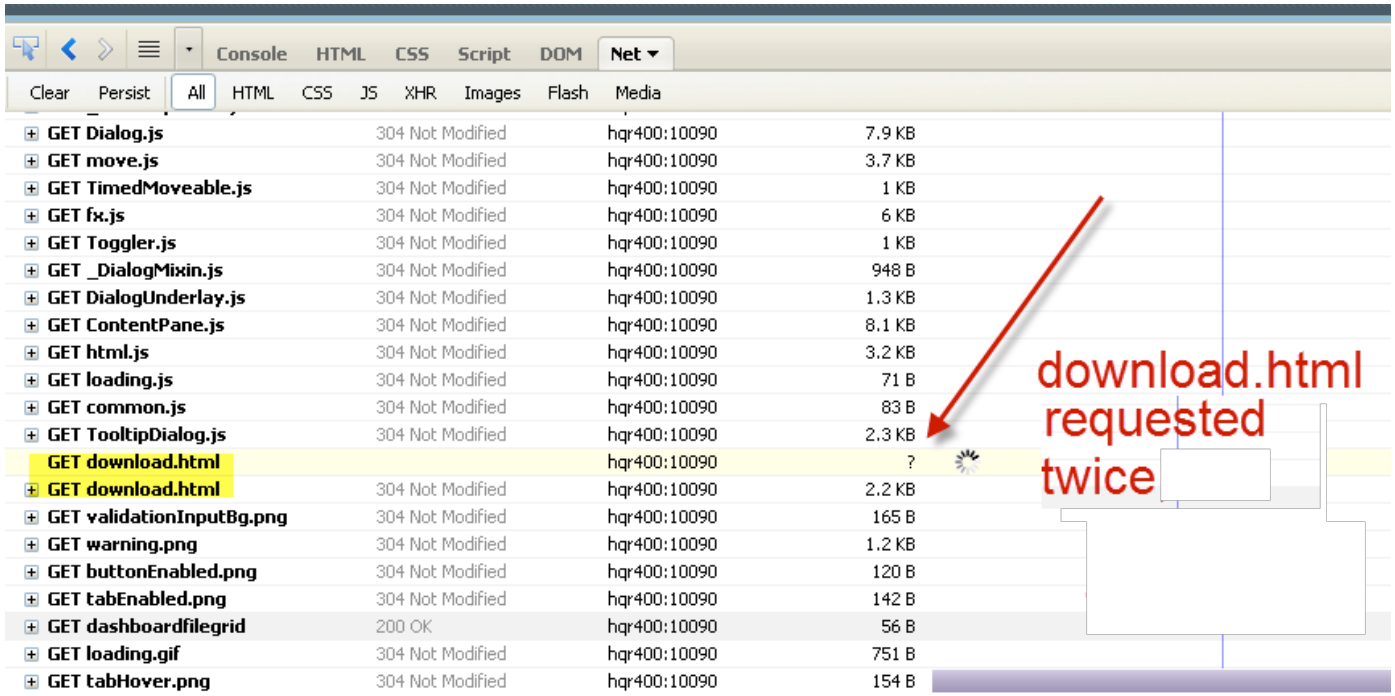
- Potential problems if overused

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

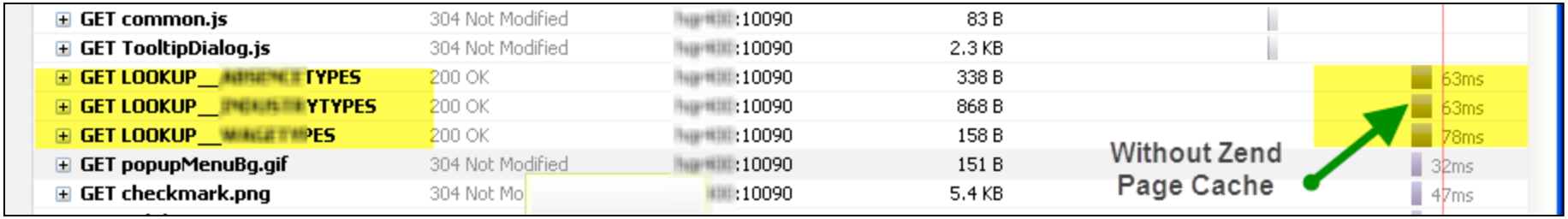
AJAX mistake #2

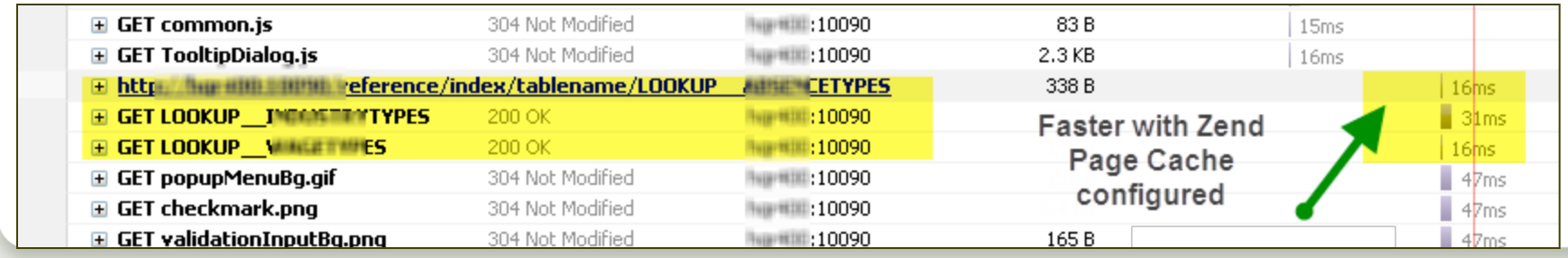
- **Duplicate requests**
  - Might go unnoticed with javascript library tools (Dojo, jQuery...)
  - Happens more than you would expect! Common, actually



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)**





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 (e.g. 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**

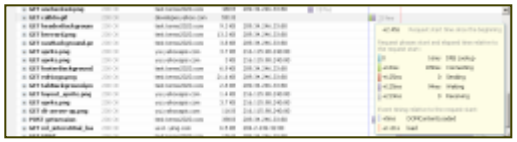
More tips for JS/CSS

- **"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

Front-end tools (demos if time)

Front-end tools demystify the web

- **Visualize HTTP requests**

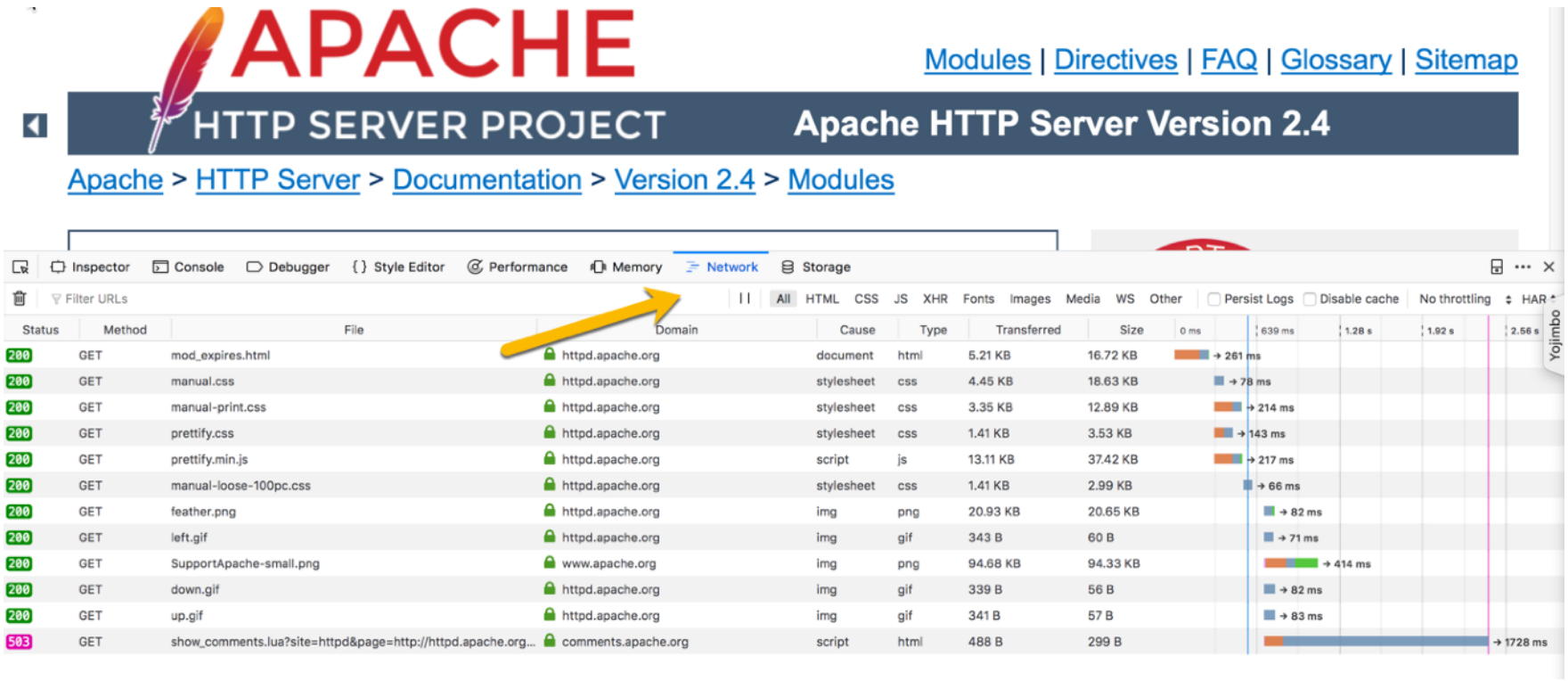


- **Find ways to eliminate requests or shrink responses**
- **Test more easily**
- **Capture "before and after" results**
  - For your own documentation
  - For a report to management

All major browsers now contain tools

- **They go by many names…most open via F12 key**
  - Chrome
    - Developer Tools
  - Firefox
    - Developer Tools
  - Internet Explorer
    - Developer Toolbar
  - Opera
    - Dragonfly
  - Safari
    - Developer Tools (turned off by default)

Network tab shows waterfall chart



Postman for APIs shows timing

POST echo.getpostman.com/post?token=abcdef Params Send Save

Authorization Headers Body Pre-request Script Tests Generate Code

```
var jsonData = JSON.parse(responseBody);
postman.setEnvironmentVariable("token", jsonData.args.token);
```

SNIPPETS

- Clear a global variable
- Clear an environment variable
- Response body: Contains string
- Response body: Convert XML body to a JSON Object
- Response body: Is equal to a string
- Response body: JSON value check
- Response headers: Content-Type header check
- Response time is less than 200ms

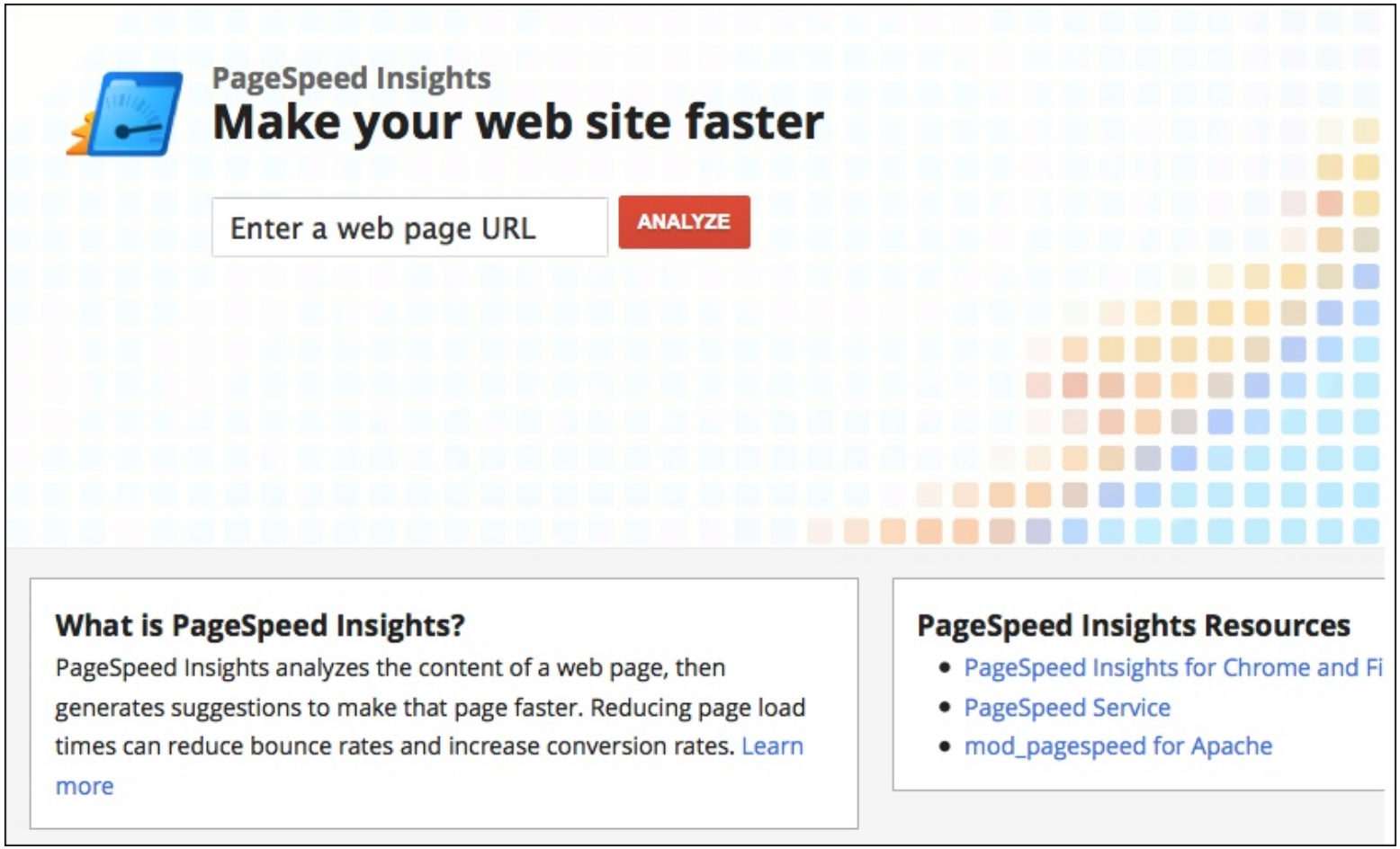
Body Cookies Headers (13) Tests Status: 200 OK Time: 332 ms

Pretty Raw Preview JSON

```
{
  "args": {
    "token": "abcdef"
  },
```

https://getpostman.com

Page Speed Insights by Google



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

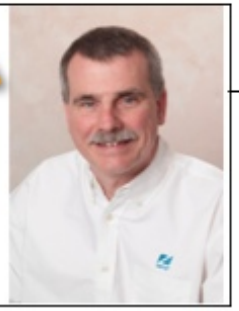
Suggestions for this page

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)

WEBPAGETEST

HOME | TEST RESULT | TEST HISTORY | FORUMS | DOCUMENTATION | ABOUT

Test a website's performance

Analytical Review | Visual Comparison | Mobile | Traceroute

Enter a Website URL

Test Location: Dulles, VA USA (IE 6-10,Chrome,Firefox) | Select from Map

Browser: IE 9

START TEST

Advanced Settings

Test Settings | Advanced | Chrome | Auth | Script | Block | SPOF | Video

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

eck]

Webpagetest "Video/filmstrip" view

Tested From: Dulles, VA - IE 9 - **Cable**

3.0s 3.5s 4.0s 4.5s

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

2% 58% 58% 58%

Export filmstrip as an image...

Slow Motion

Create Video

Advanced customization options...

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

http://iprodeveloper.com/web-amp-nobi...

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

1. iprodeveloper.c...-performance-snags 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...4ed261883b423d.css 98 ms
6. iprodeveloper.c...0c959c2502bb01.css 208 ms
7. iprodeveloper.com - contact-icon.png 564 ms
8. iprodeveloper.c...35af5756e17f8d0.js 1704 ms
9. iprodeveloper.c...1caa67e1ab9cb4.css 600 ms
10. platform.linked...AnonymousFramework 271 ms
11. apis.google.com - cb=gapi.loaded_0 287 ms
12. ssl.gstatic.com...c34bd01bb0bb.png 287 ms (304)
13. iprodeveloper.c...ery-1.4.1-vsdoc.js 78 ms
14. iprodeveloper.com - search-icon.gif 77 ms

Advanced Settings of webpagetest

Advanced Settings

Test Settings | Advanced | Chrome | Auth | Script | Block | SPOF | Video

- [ ] **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 seconds
Capture data for at least...

CloudShark

Network packet trace (Advanced Settings)

GTMetrix (gtmetrix.com)

https://gtmetrix.com/reports/www.seidengroup.com/KQsaewGW

GTmetrix | Features | Resources | Blog | GTmetrix PRO | Log in | Sign Up

Latest Performance Report for:

http://www.seidengroup.com/

Report generated: Sat, Nov 4, 2017, 7:07 PM -0700
Test Server Region: Vancouver, Canada
Using: Chrome (Desktop) 58.0.3029.96, PageSpeed 1.15-gt1, YSlow 3.1.8

Looks like you're running WordPress
Have a look at our WP optimization tips »

Looks like you might not be using a CDN
Why should I use a CDN? »

Re-Test
Compare
Page Settings
Monitor
Set Up Alerts
Download PDF
Share This Report

Performance Scores

PageSpeed Score	YSlow Score
B (86%)	B (84%)

Page Details

Fully Loaded Time	Total Page Size	Requests
2.7s	473KB	26

PageSpeed | YSlow | Waterfall | Timings | Video | History

RECOMMENDATION	GRADE	TYPE	PRIORITY
Defer parsing of JavaScript	E (57)	JS	HIGH
Enable Keep-Alive	D (67)	SERVER	HIGH
Avoid bad requests	C (75)	CONTENT	HIGH
Optimize the order of styles and scripts	A (92)	CSS/JS	HIGH
Leverage browser caching	A (92)	SERVER	HIGH
Optimize images	A (94)	IMAGES	HIGH
Minify JavaScript	A (99)	JS	HIGH

What do my scores mean?

Rules are sorted in order of impact upon score
Optimizing rules at the top of the list can greatly improve your overall score.

Not every recommendation will apply to your page
The recommendations are meant to be generic, best practices; some things will be out of your control (eg. external resources) or may not apply to your page.

Learn more about PageSpeed/YSlow scores and how they affect performance.

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 your developer tools (or Firebug), and tools such as Web Page Test, Page Speed Insights, and GTMetrix 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 SpeedCurve)**
  - http://stevesouders.com
  - @souders
  - Books: High Performance Web Sites, Even Faster Web Sites

Please provide feedback on this talk

https://joind.in/talk/9b47f

Slides will be posted there by tomorrow

Contact and tips

Alan Seiden

Seiden Group

Ho-Ho-Kus, NJ

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

CIO Summit information:
RSVP@seidengroup.com

SEIDEN GROUP

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