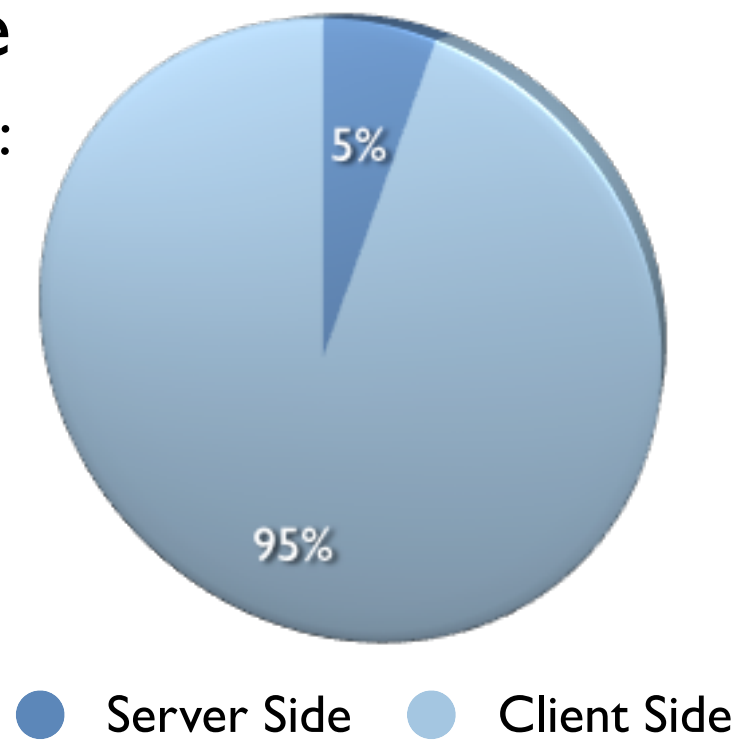
High Performance Web Pages

Real World Examples: Netflix Case Study

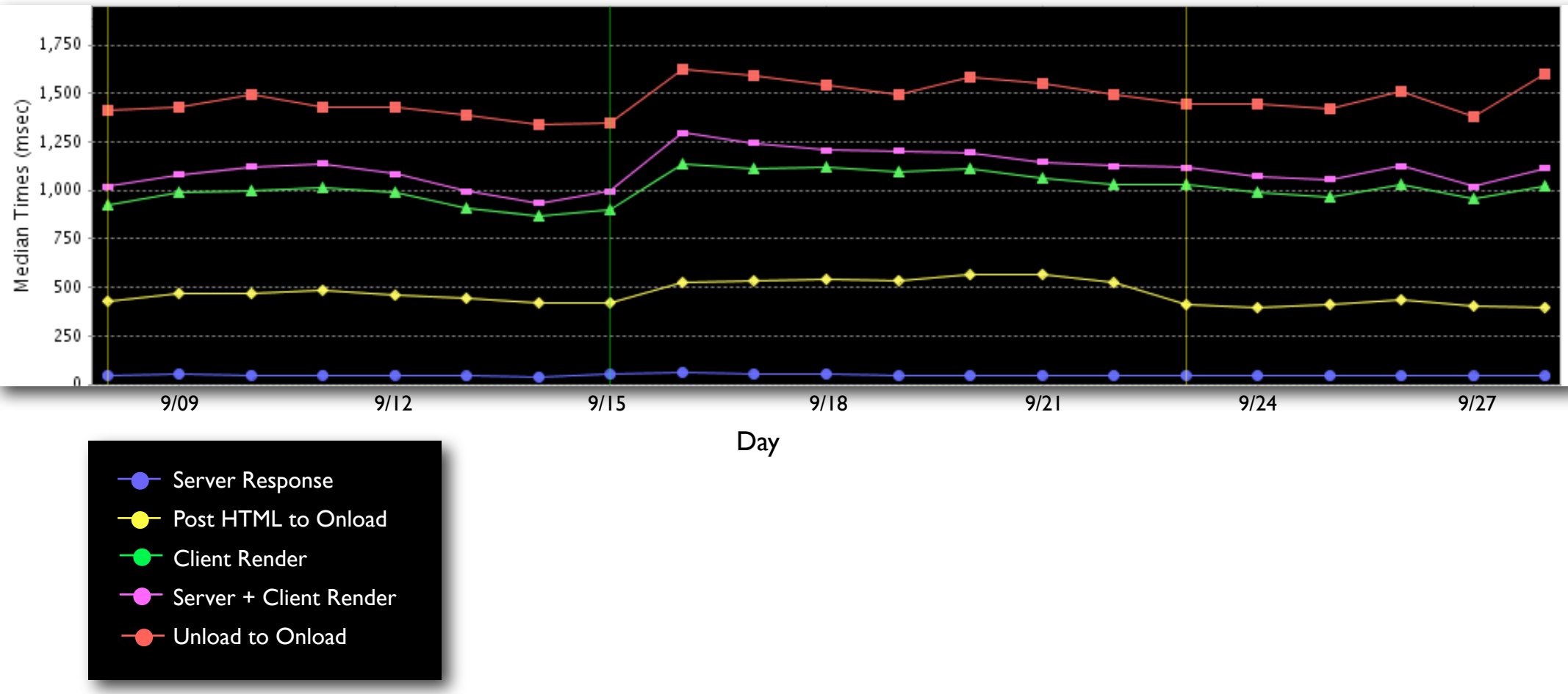
Bill Scott
CS193H. Fall 2008.
Stanford University

The Situation

- All attention was focused on server side
  - Most pages 200 - 300 ms
  - Savings of 20 ms celebrated
- Server side is a tiny fraction of the performance pie
  - Typical Netflix page:



Typical Performance



Same Profile on Member Home

- Total response/render times:
  - 75% of customers experience less than10 seconds (which means 25% are experiencing greater than 10 seconds)
  - 42% less than 5 seconds
  - 29% less than 3 seconds
- Server response times:
  - Appear to be only at a maximum **4% of total response/ render time**
  - 75% of server response times are less than 1.15 seconds
- Conclusion: Lots of room for client side improvement

Goal

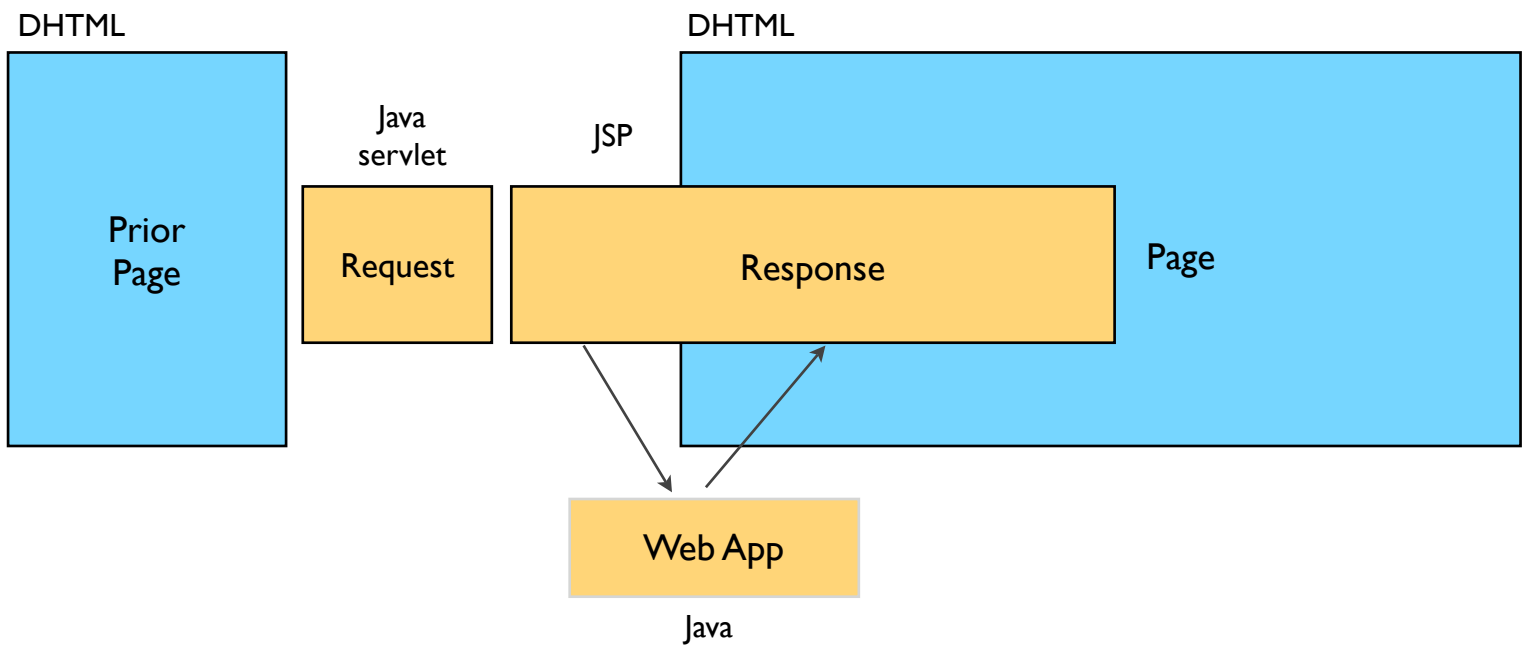
- Measure request-to-response cycle
  - Gets at what the user actually experiences
- Improve end user performance
  - Implement Steve Souder's performance rules

The Plan

- Metrics Capture (round trip tracing)
- Apache - gzip, ffe, etag configuration, Proxy cache configuration
- Integrate new Starbar into website
- Better minification for JS & CSS (yui minifier)
- Sprite Bob Graphic Images
- Sprite Header Graphics
- Sprite Queue graphic images
- Configure CDN image assets to use FFE & create image version push system
- Move JS to the bottom, CSS to the top, better JS/CSS packaging
- Switch from graphics based buttons to CSS/Text based buttons
- Switch vignettes from graphics based to CSS/Text based graphic
- Lighter weight BOB, QACL, Menu Nav
- Reduce number of CDN cnames in use

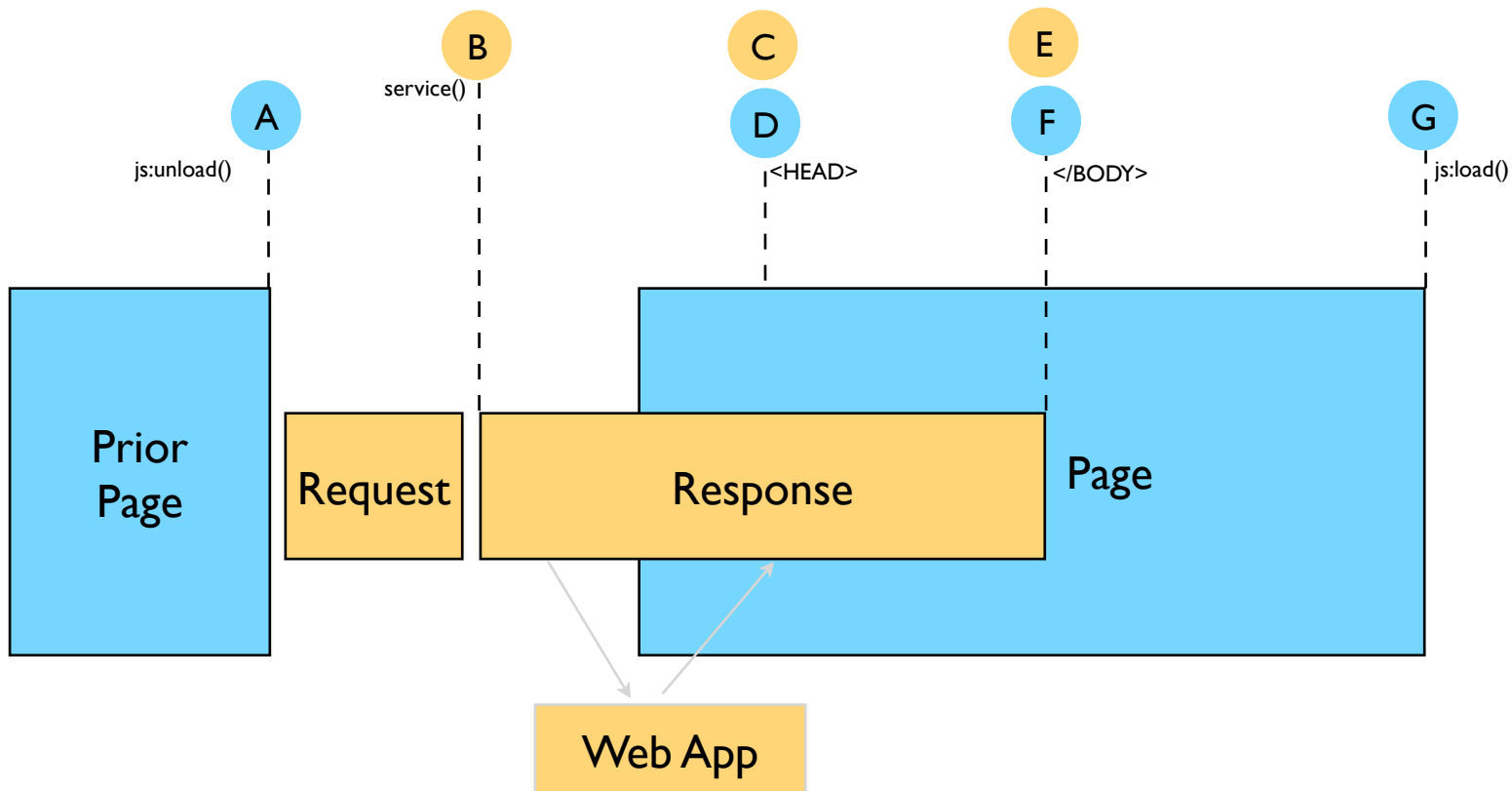
Metrics Capture

Round Trip Tracers

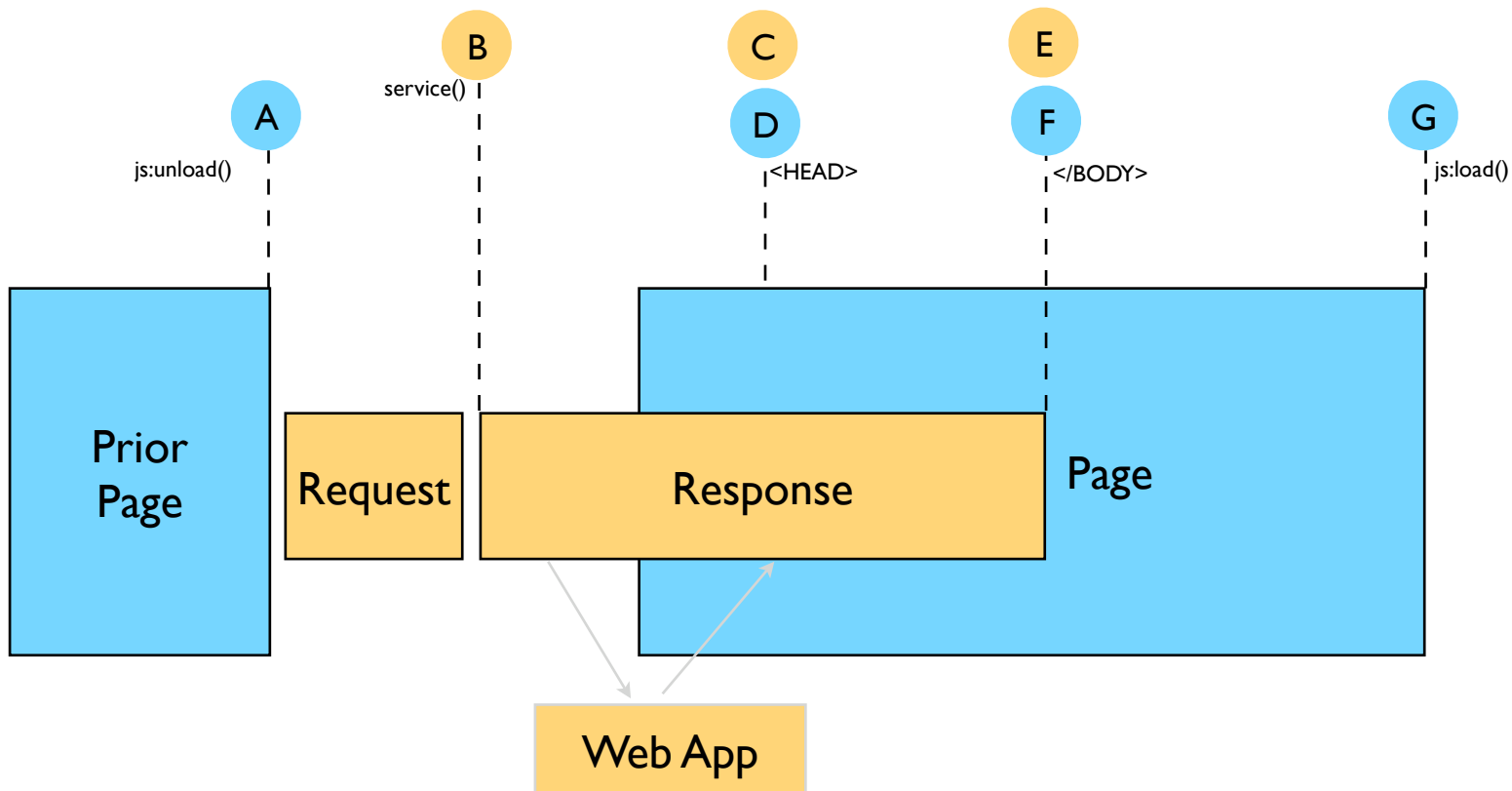


- Client side javascript + server side Java captures time points
- Allows us to measure roundtrip time from request to render

Measurement Points

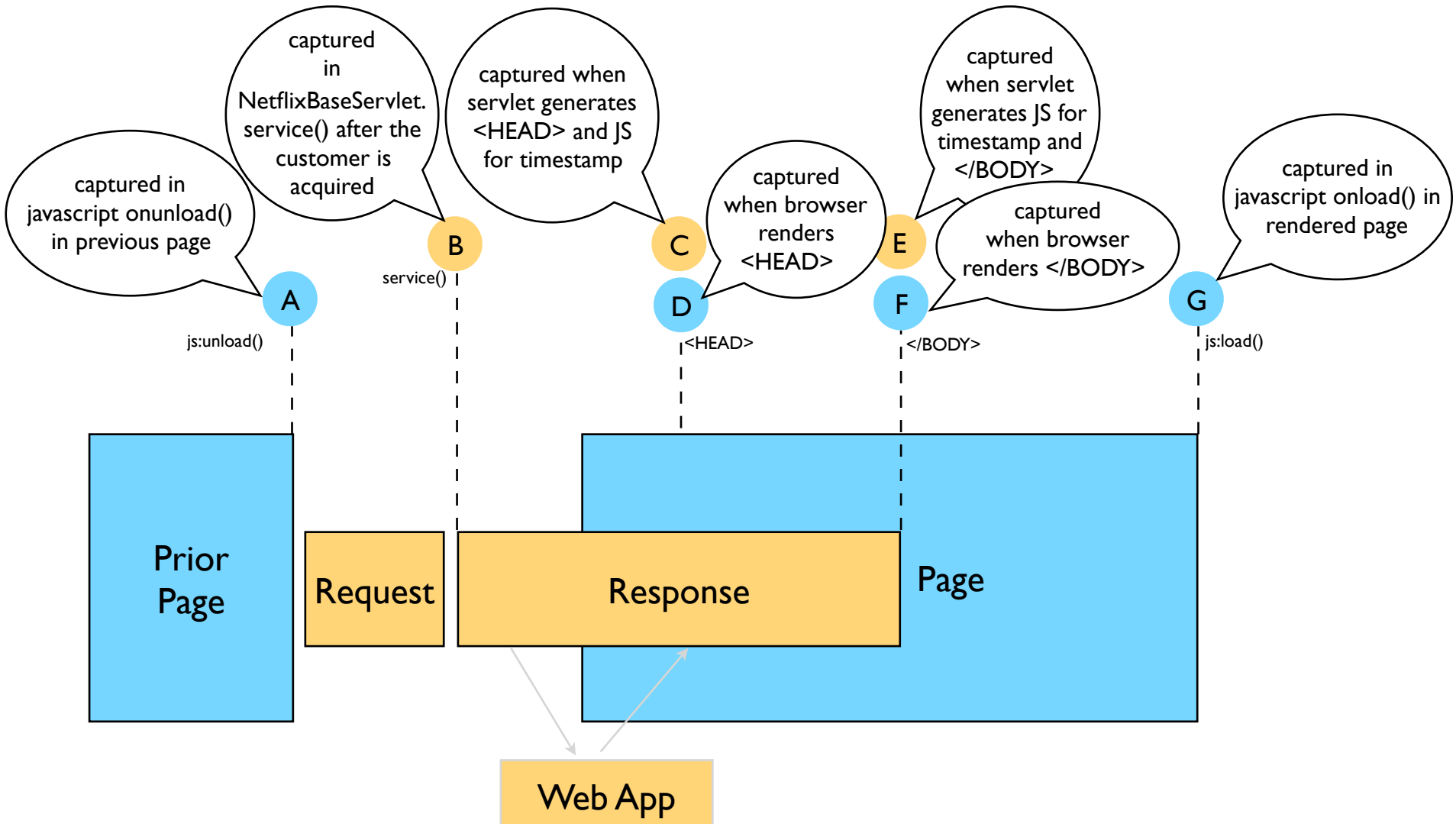


Measurement Points

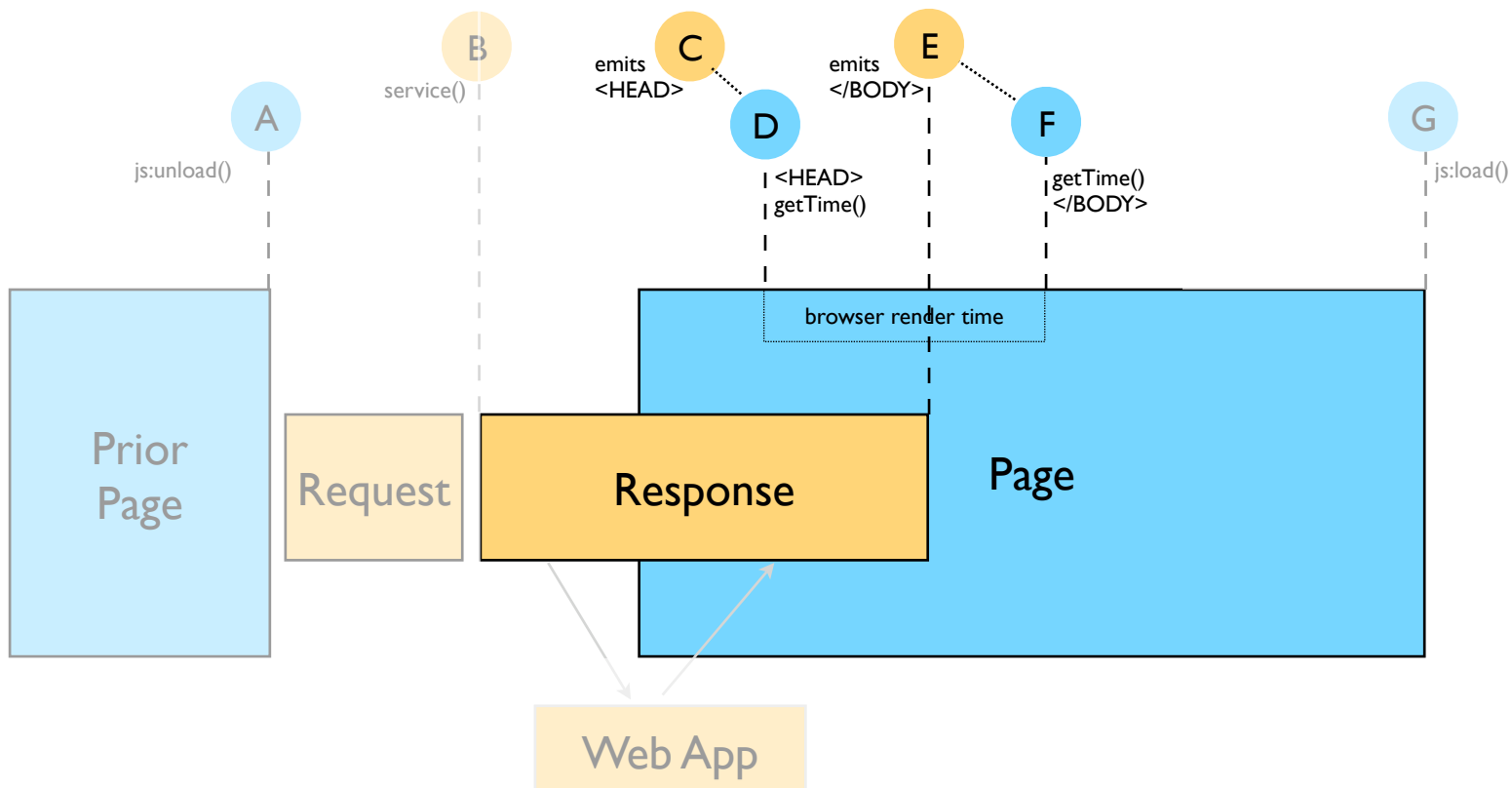


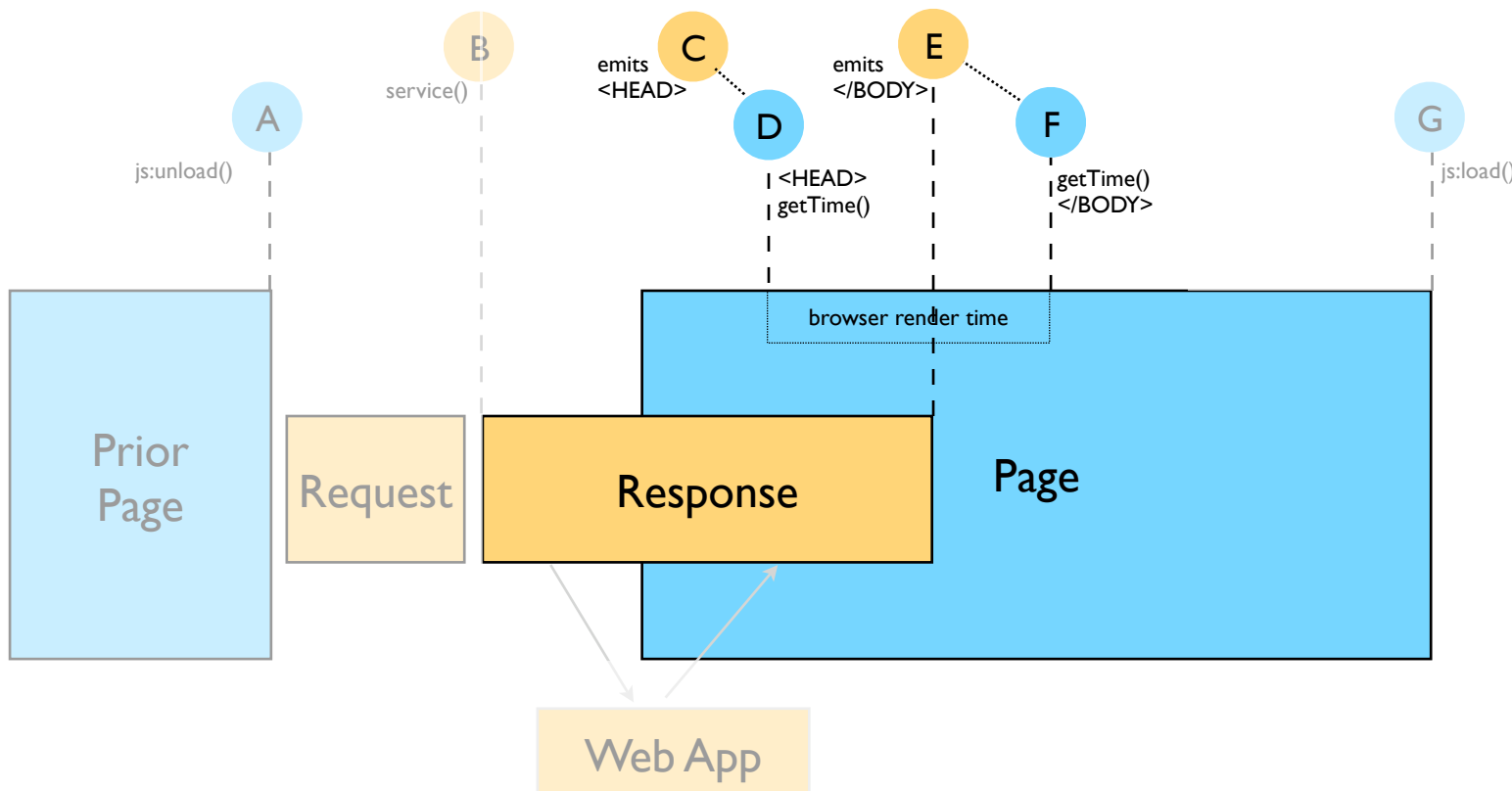
- 8 time stamps
  - 4 client, 3 server

Measurement Points

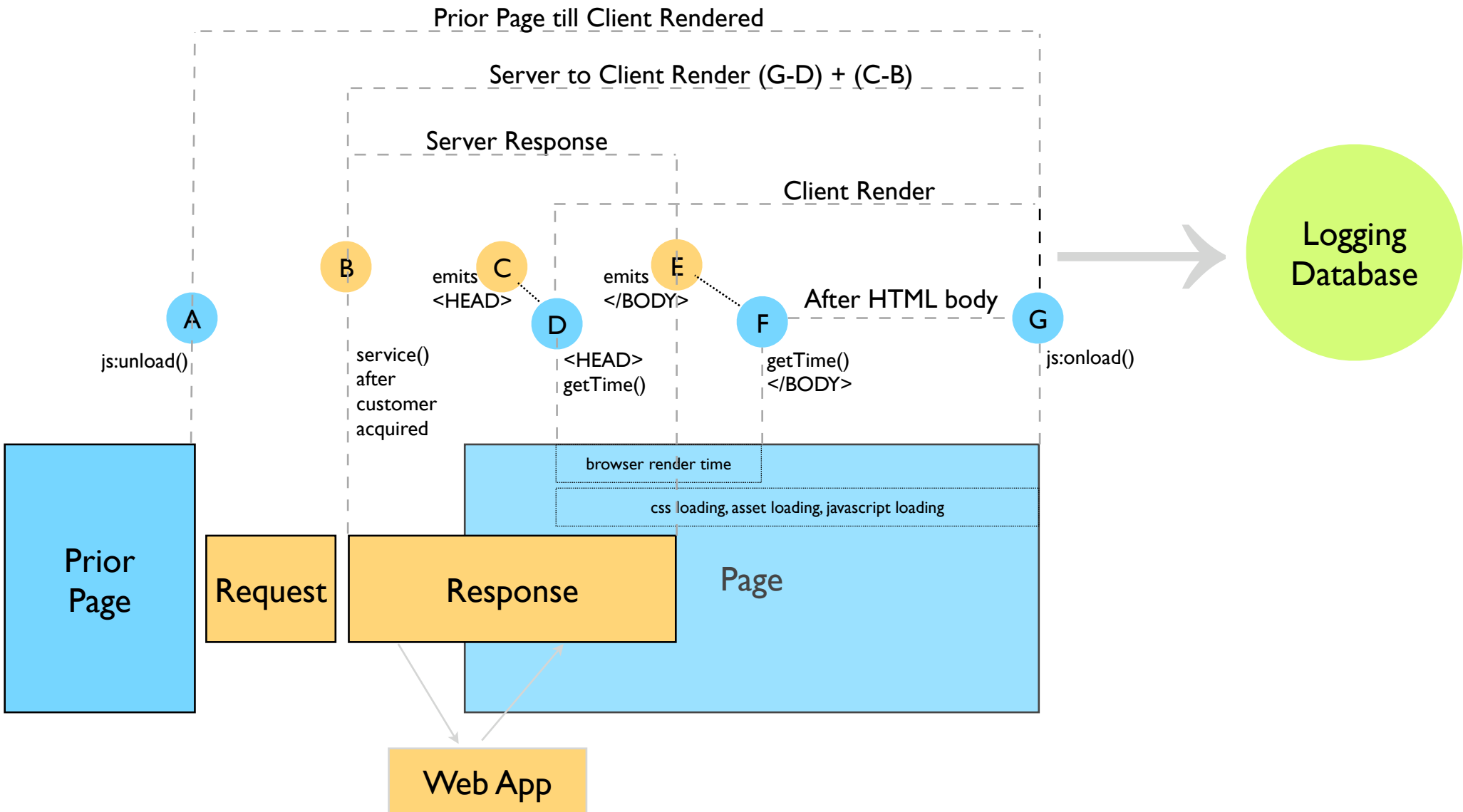


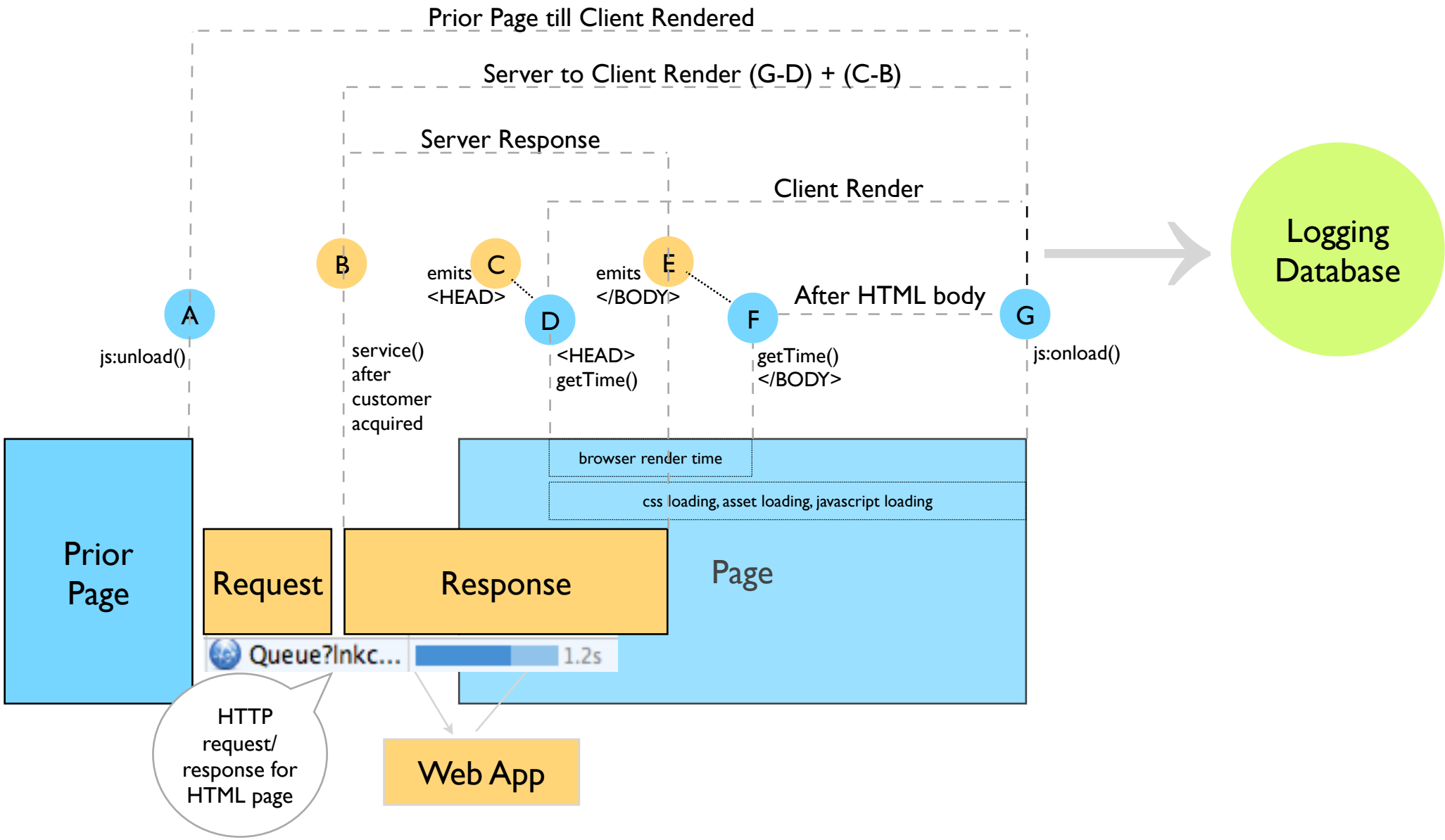
- 8 time stamps
  - 4 client, 3 server

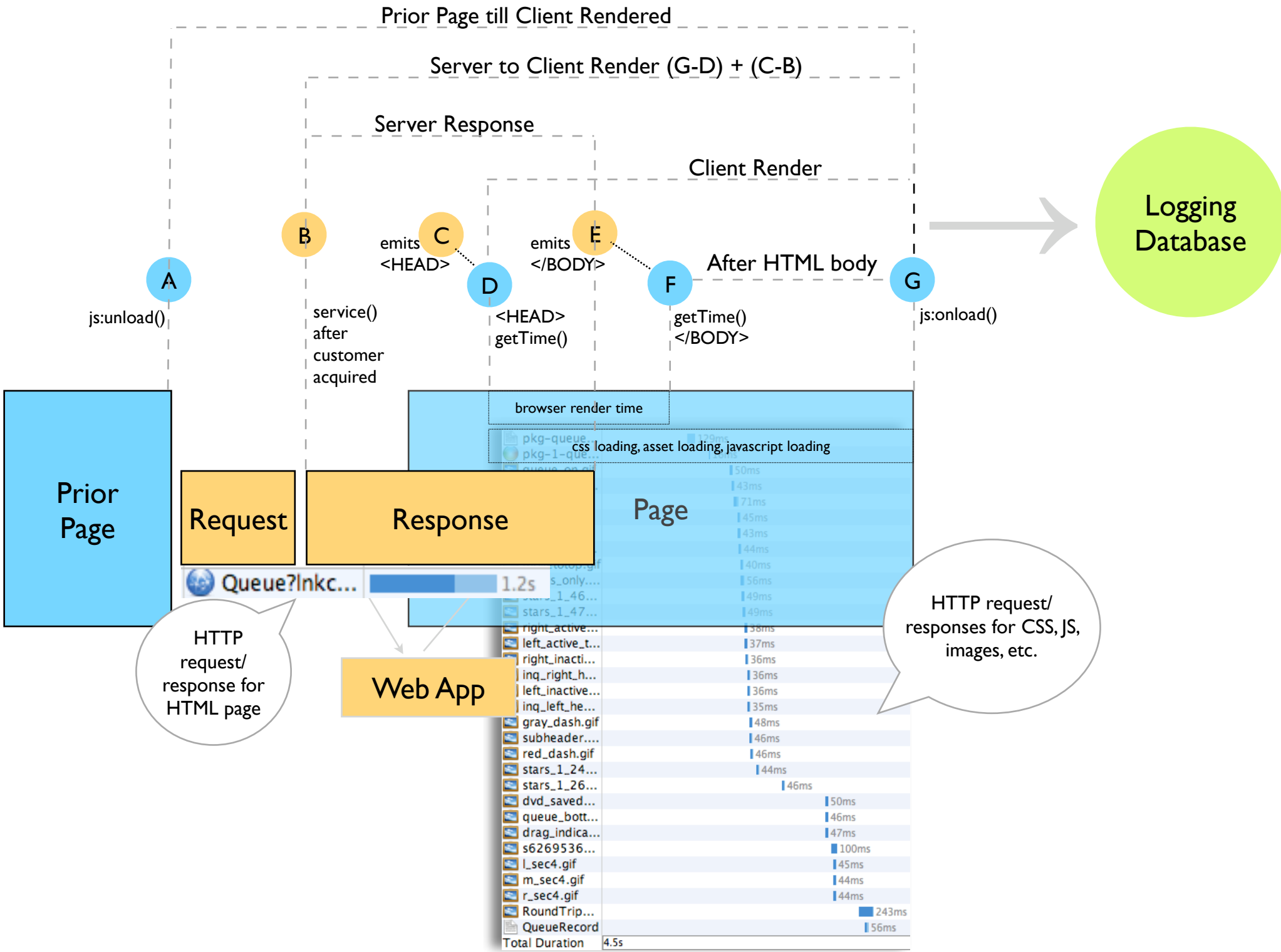

A
B
C
D
E
F
G
service()
emits
<HEAD>
emits
</BODY>
js:unload()
<HEAD>
getTime()
getTime()
</BODY>
js:load()
browser render time
Prior
Page
Request
Response
Page
Web App

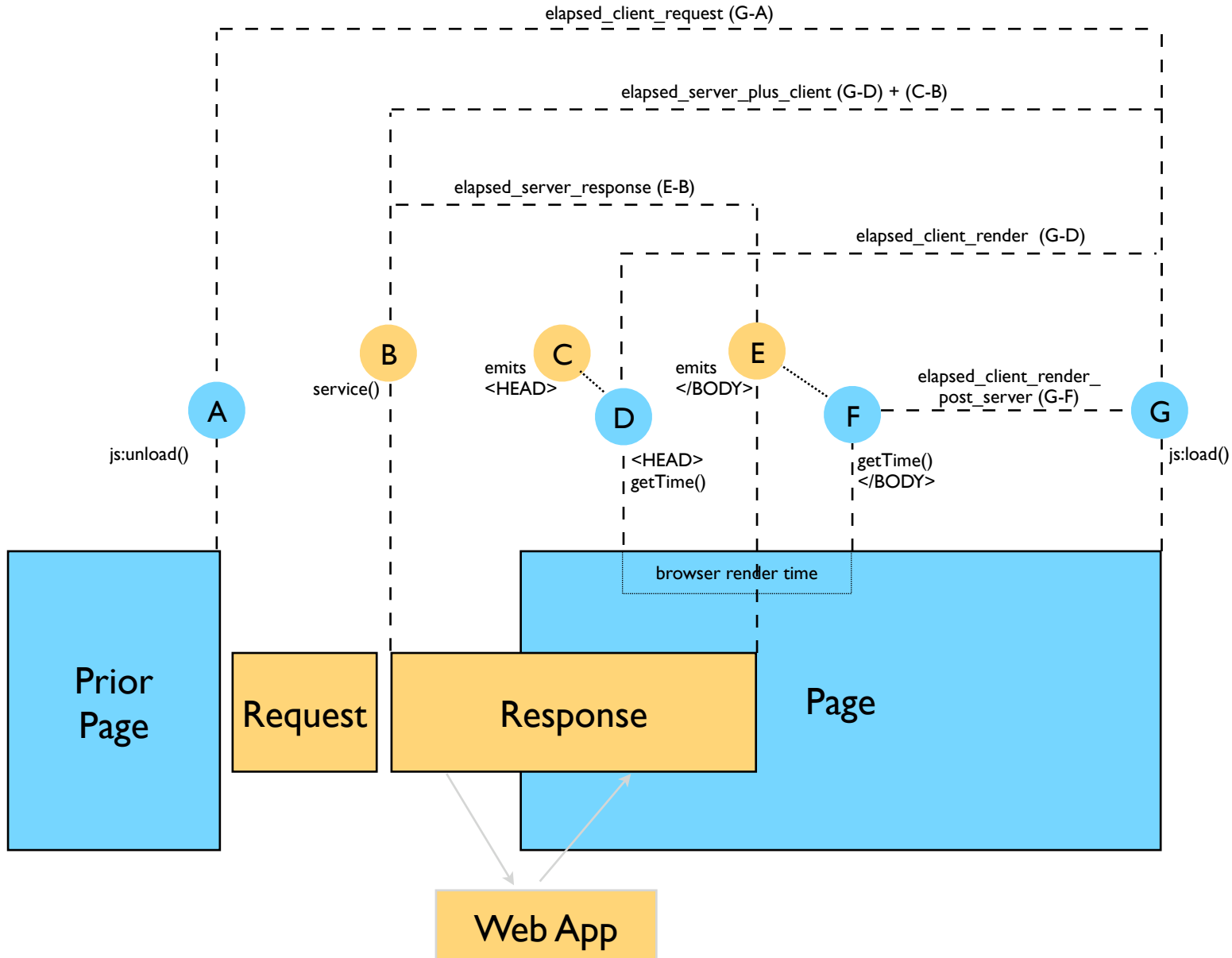


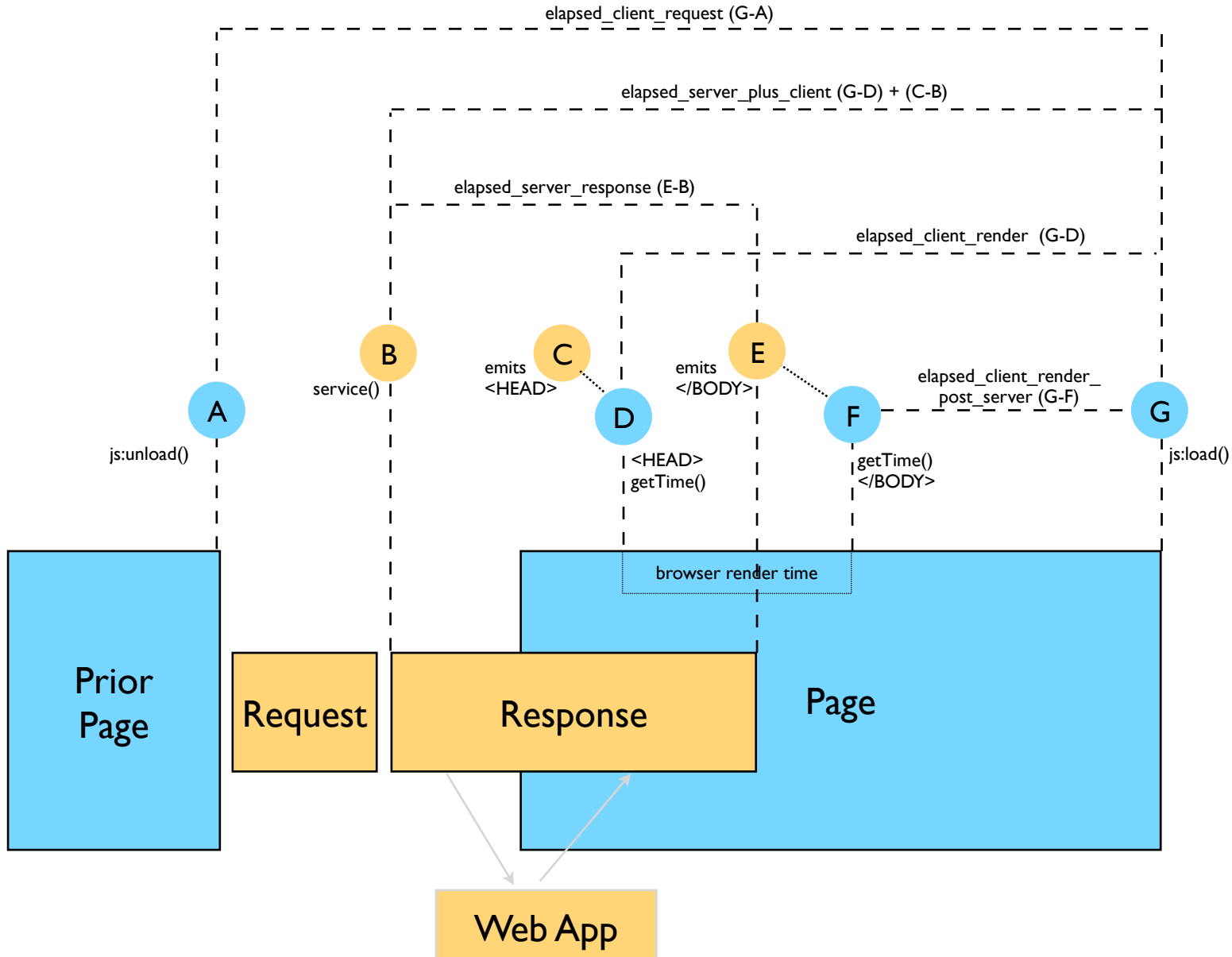
- C&D and E&F are don't happen together
  - servlet generation time vs browser render time


Prior Page till Client Rendered
Server to Client Render (G-D) + (C-B)
Server Response
Client Render
A
B
C
D
E
F
G
emits
<HEAD>
emits
</BODY>
After HTML body
js:unload()
service()
after
customer
acquired
<HEAD>
getTime()
getTime()
</BODY>
js:onload()
Logging
Database
browser render time
css loading, asset loading, javascript loading
Prior
Page
Request
Response
Page
Web App


Prior Page till Client Rendered
Server to Client Render (G-D) + (C-B)
Server Response
Client Render
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
After HTML body
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
Logging Database
browser render time
css loading, asset loading, javascript loading
Prior Page
Request
Response
Page
Queue?lnkc...
1.2s
HTTP request/ response for HTML page
Web App

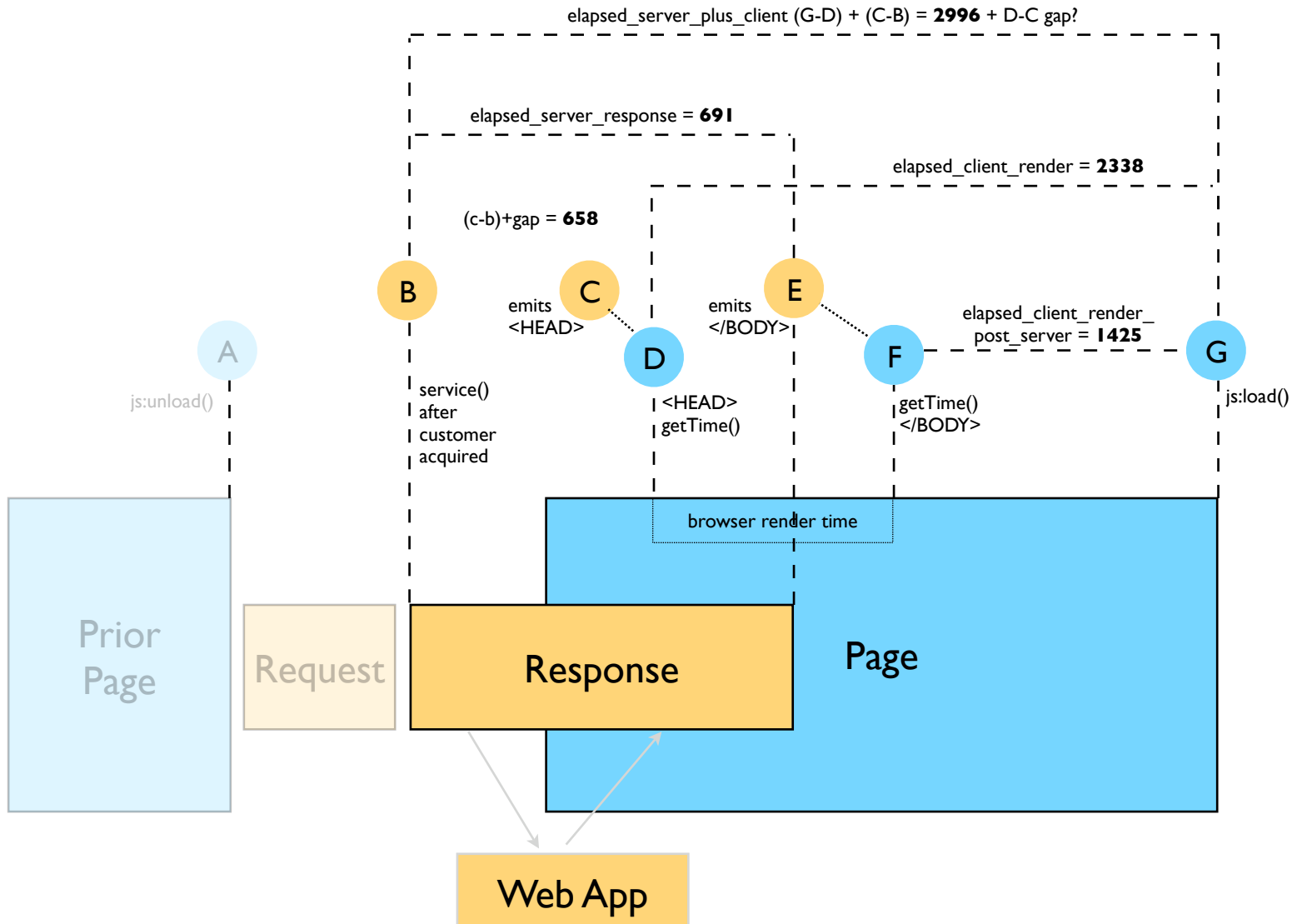

Prior Page till Client Rendered
Server to Client Render (G-D) + (C-B)
Server Response
Client Render
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
After HTML body
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
Logging Database
Prior Page
Request
Response
Page
browser render time
css loading, asset loading, javascript loading
Queue?lnkc...
1.2s
HTTP request/ response for HTML page
Web App
HTTP request/ responses for CSS, JS, images, etc.
pkg-queue
left_active_t...
right_inacti...
inq_right_h...
left_inactive...
inq_left_he...
gray_dash.gif
subheader....
red_dash.gif
stars_1_24...
stars_1_26...
dvd_saved...
queue_bott...
drag_indica...
s6269536...
l_sec4.gif
m_sec4.gif
r_sec4.gif
RoundTrip...
QueueRecord
Total Duration
4.5s


elapsed_client_request (G-A)
elapsed_server_plus_client (G-D) + (C-B)
elapsed_server_response (E-B)
elapsed_client_render (G-D)
A
B
C
D
E
F
G
js:unload()
service()
emits
<HEAD>
<HEAD>
getTime()
emits
</BODY>
getTime()
</BODY>
elapsed_client_render_
post_server (G-F)
js:load()
browser render time
Prior
Page
Request
Response
Page
Web App

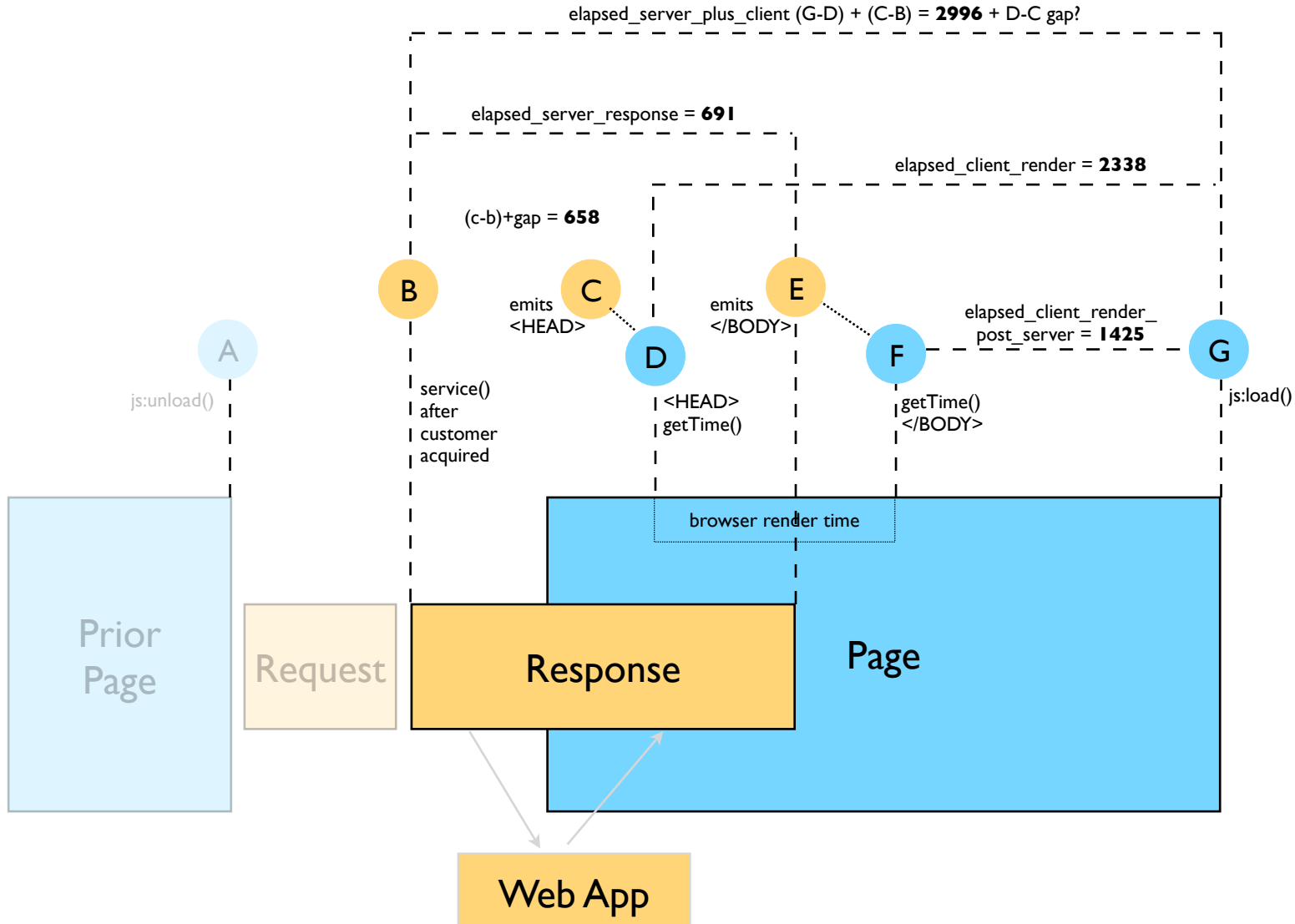


- 5 metric values; 2 on client, 2 on server, 1 timed on both (G-D) + (C-B)... there is a gap however

/MemberHome

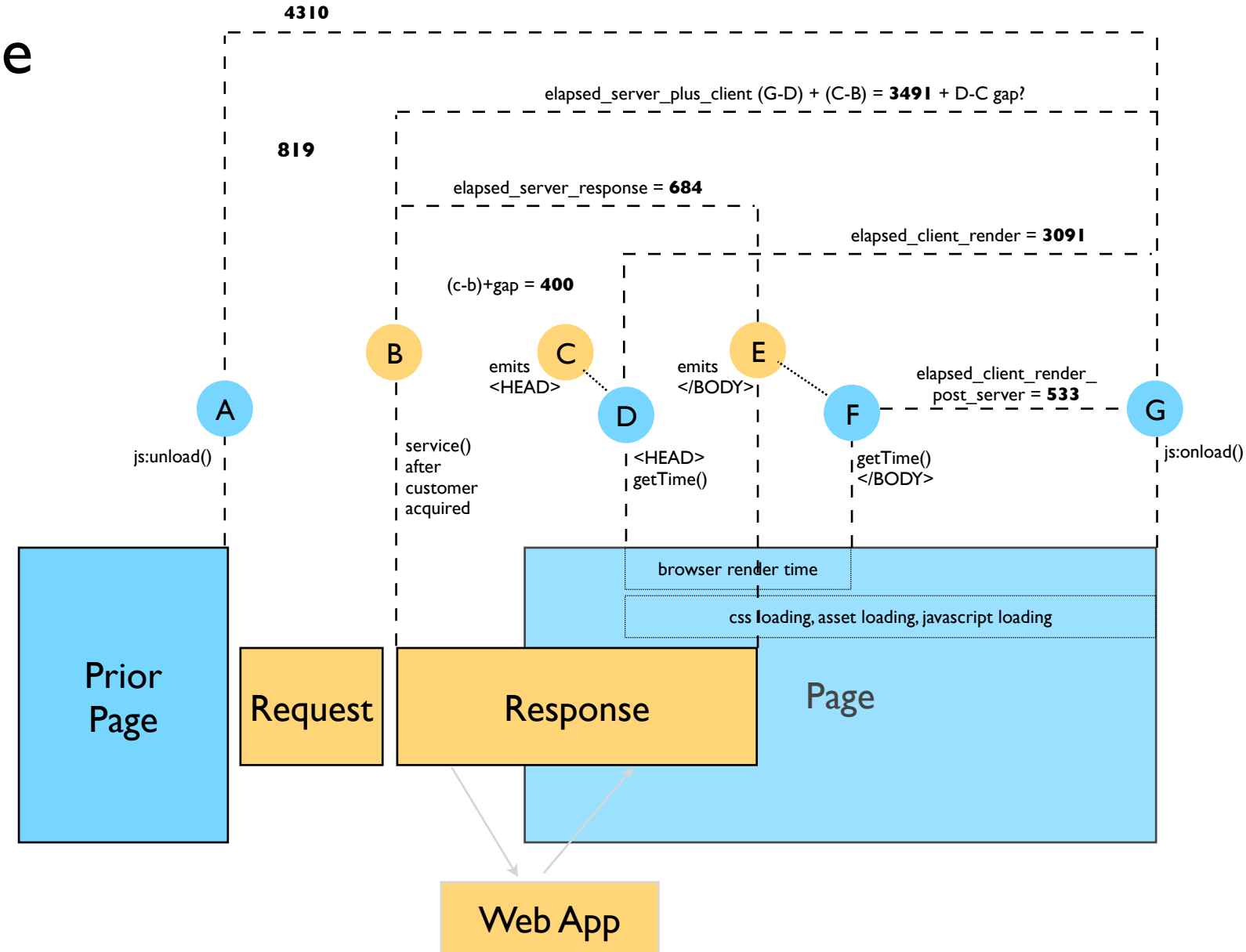

elapsed_server_plus_client (G-D) + (C-B) = 2996 + D-C gap?
elapsed_server_response = 691
elapsed_client_render = 2338
(c-b)+gap = 658
B
C
emits
<HEAD>
D
E
emits
</BODY>
F
elapsed_client_render_
post_server = 1425
G
A
js:unload()
service()
after
customer
acquired
<HEAD>
getTime()
getTime()
</BODY>
js:load()
browser render time
Prior
Page
Request
Response
Page
Web App

/MemberHome

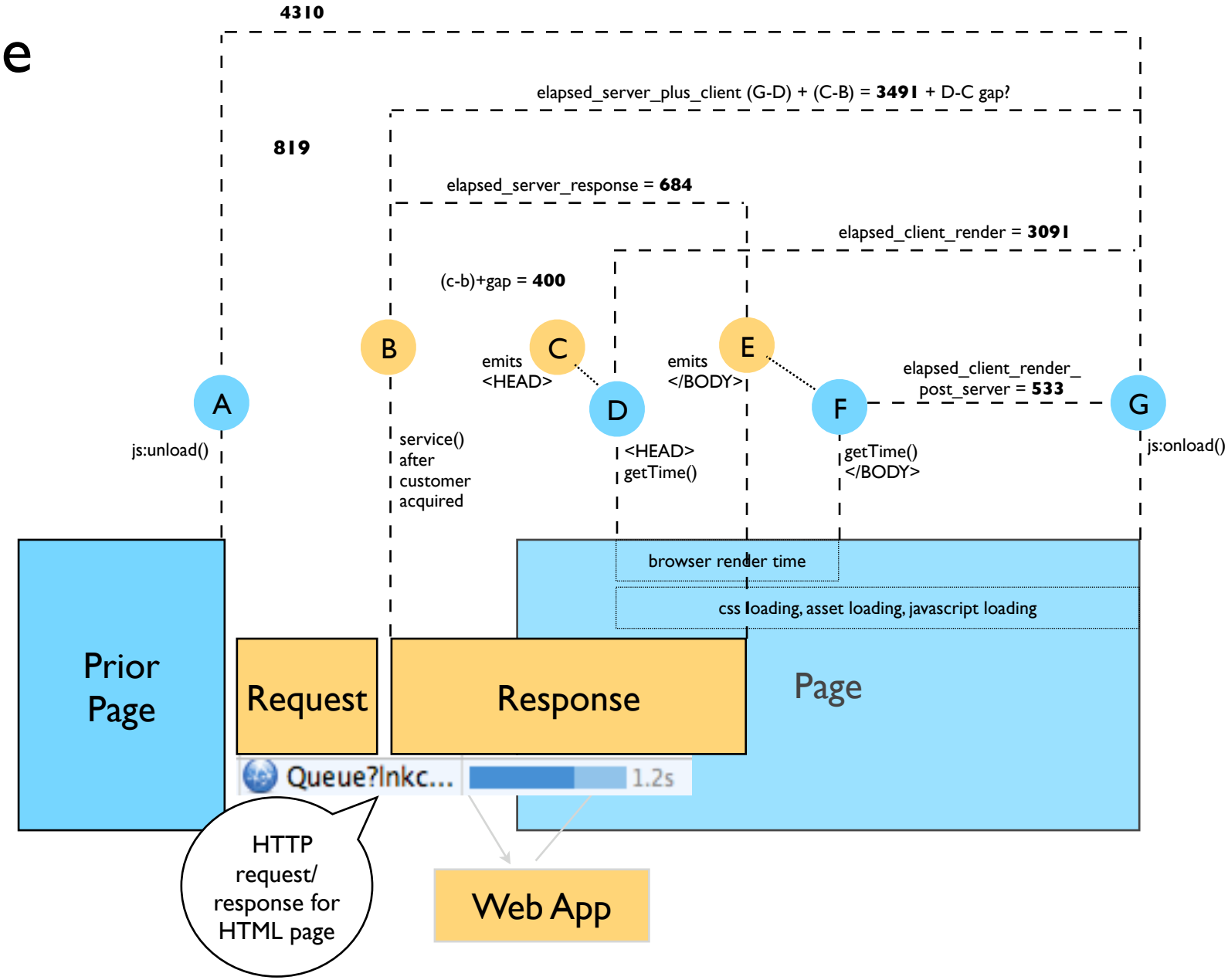


- Navigated directly from another site (no G-A) available

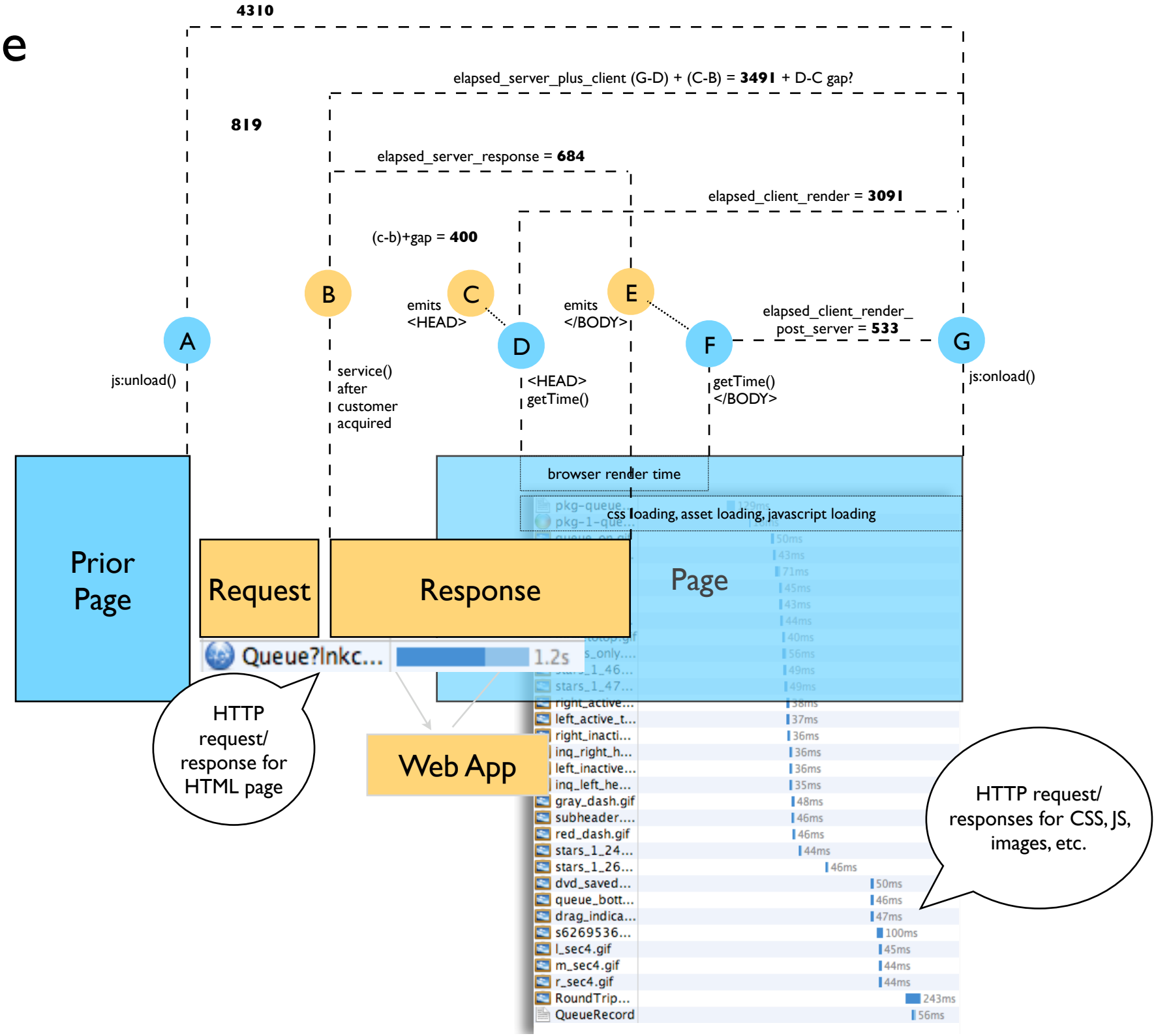
/Queue


4310
elapsed_server_plus_client (G-D) + (C-B) = 3491 + D-C gap?
819
elapsed_server_response = 684
elapsed_client_render = 3091
(c-b)+gap = 400
A
B
C
D
E
F
G
emits
<HEAD>
emits
</BODY>
elapsed_client_render_
post_server = 533
js:unload()
service()
after
customer
acquired
<HEAD>
getTime()
getTime()
</BODY>
js:onload()
browser render time
css loading, asset loading, javascript loading
Prior
Page
Request
Response
Page
Web App

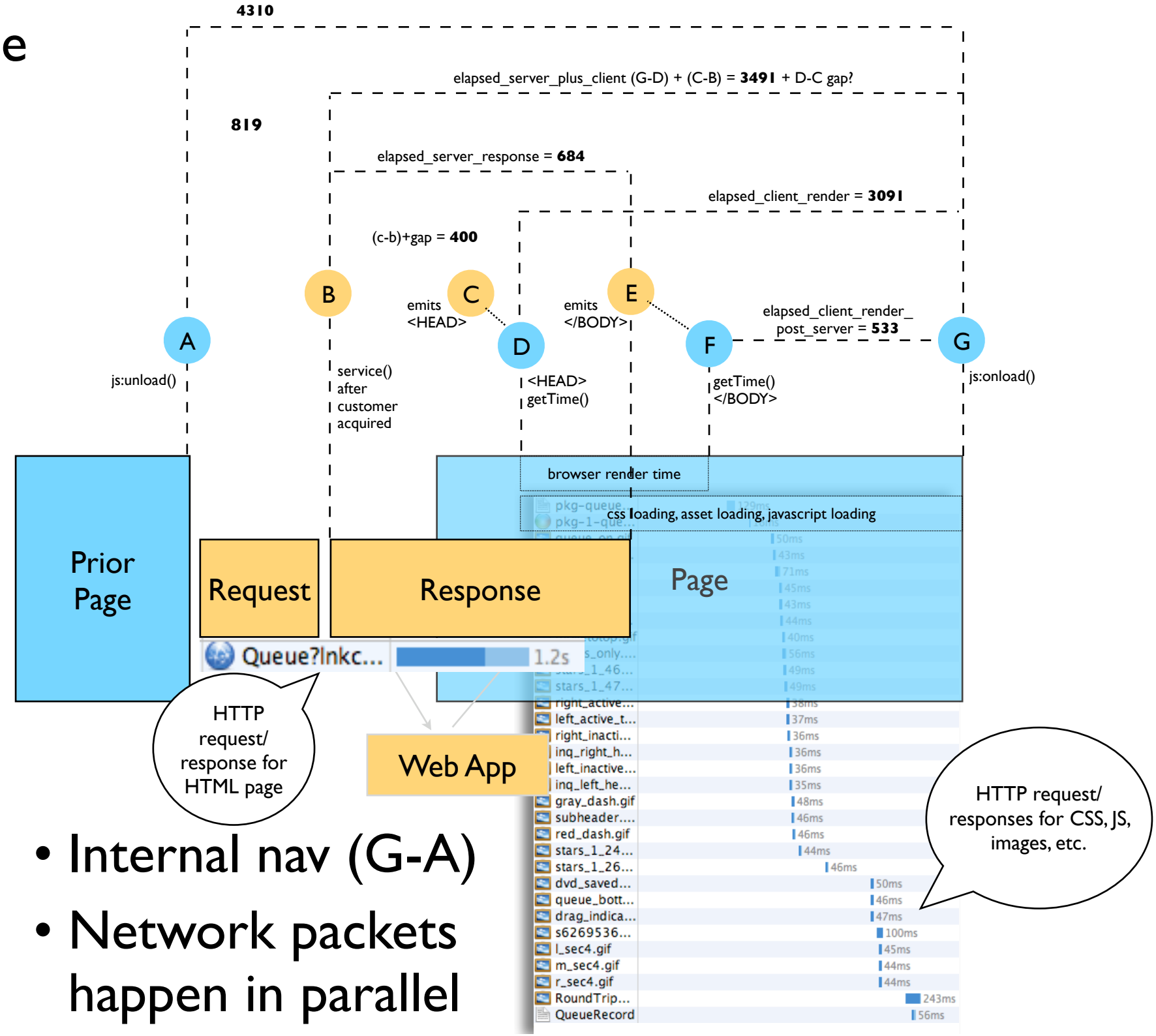
/Queue


4310
elapsed_server_plus_client (G-D) + (C-B) = 3491 + D-C gap?
819
elapsed_server_response = 684
elapsed_client_render = 3091
(c-b)+gap = 400
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
elapsed_client_render_ post_server = 533
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
browser render time
css loading, asset loading, javascript loading
Prior Page
Request
Response
Page
Queue?lnkc...
1.2s
HTTP request/ response for HTML page
Web App

/Queue


4310
elapsed_server_plus_client (G-D) + (C-B) = 3491 + D-C gap?
819
elapsed_server_response = 684
elapsed_client_render = 3091
(c-b)+gap = 400
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
elapsed_client_render_ post_server = 533
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
browser render time
css loading, asset loading, javascript loading
Prior Page
Request
Response
Page
Queue?lnkc...
1.2s
HTTP request/ response for HTML page
Web App
HTTP request/ responses for CSS, JS, images, etc.
pkg-queue...
pkg-1-que...
stars_1_46...
stars_1_47...
left_active_t...
right_inacti...
inq_right_h...
left_inactive...
inq_left_he...
gray_dash.gif
subheader....
red_dash.gif
stars_1_24...
stars_1_26...
dvd_saved...
queue_bott...
drag_indica...
s6269536...
l_sec4.gif
m_sec4.gif
r_sec4.gif
RoundTrip...
QueueRecord

/Queue



- Internal nav (G-A)
- Network packets happen in parallel

What to Capture

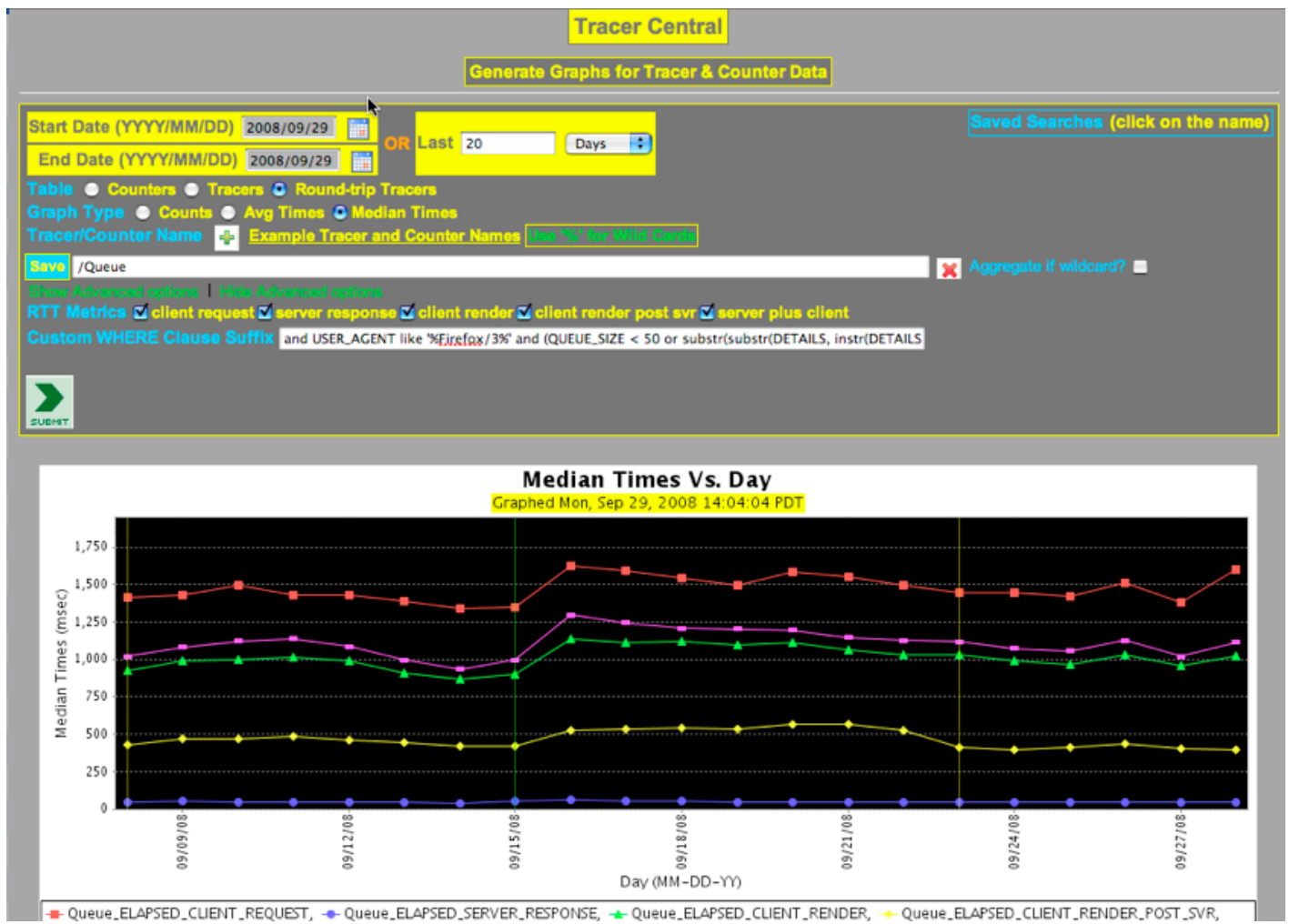
- Time-stamp
- Customer ID
- Page (logical name)
- URL
- Referrer
- Full round trip request time
- Server Reponse time
- Client render time
- Time from HTML processing to Onload
- Server Response + Client Render

What to Capture

- Details
- IP address
- User agent
- Connection type
- Server name
- Browser
- OS
- Bandwidth test

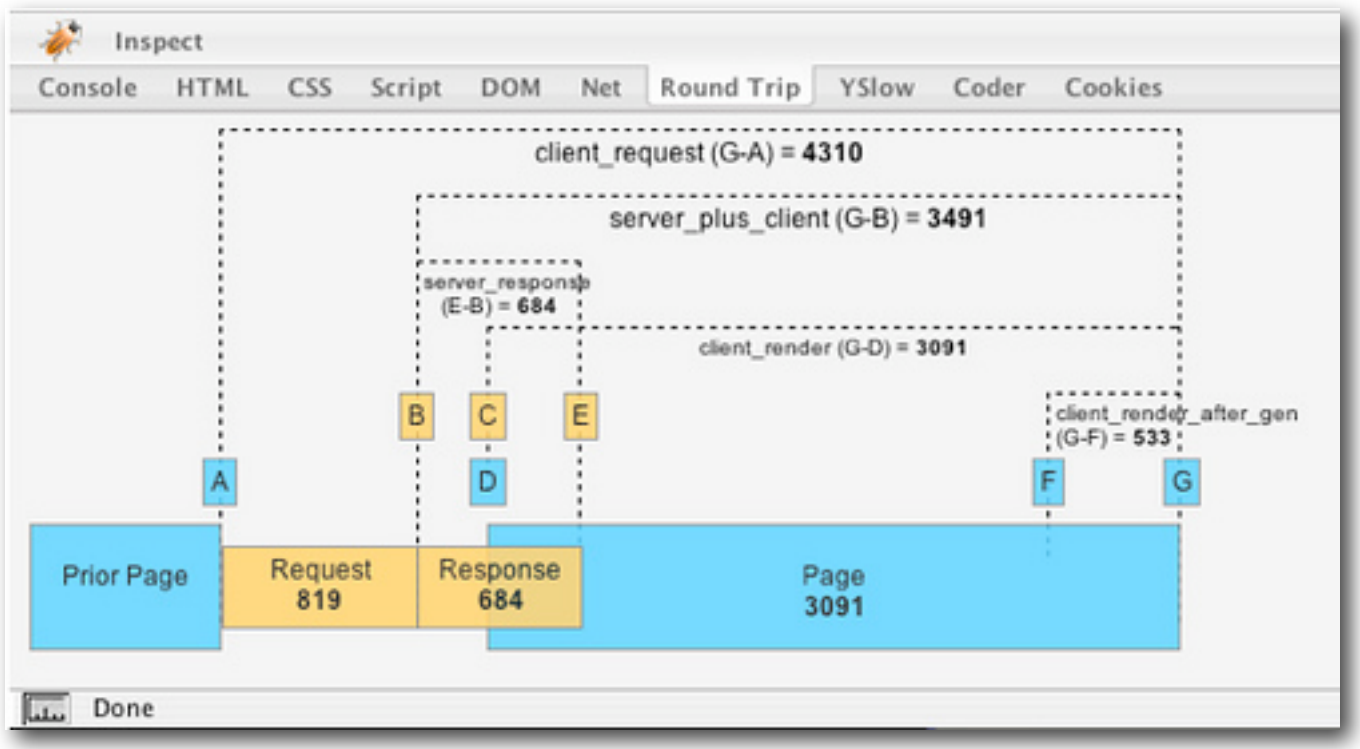
Logged Metrics

- Logged to database
- Self-Service Portal for plotting metrics

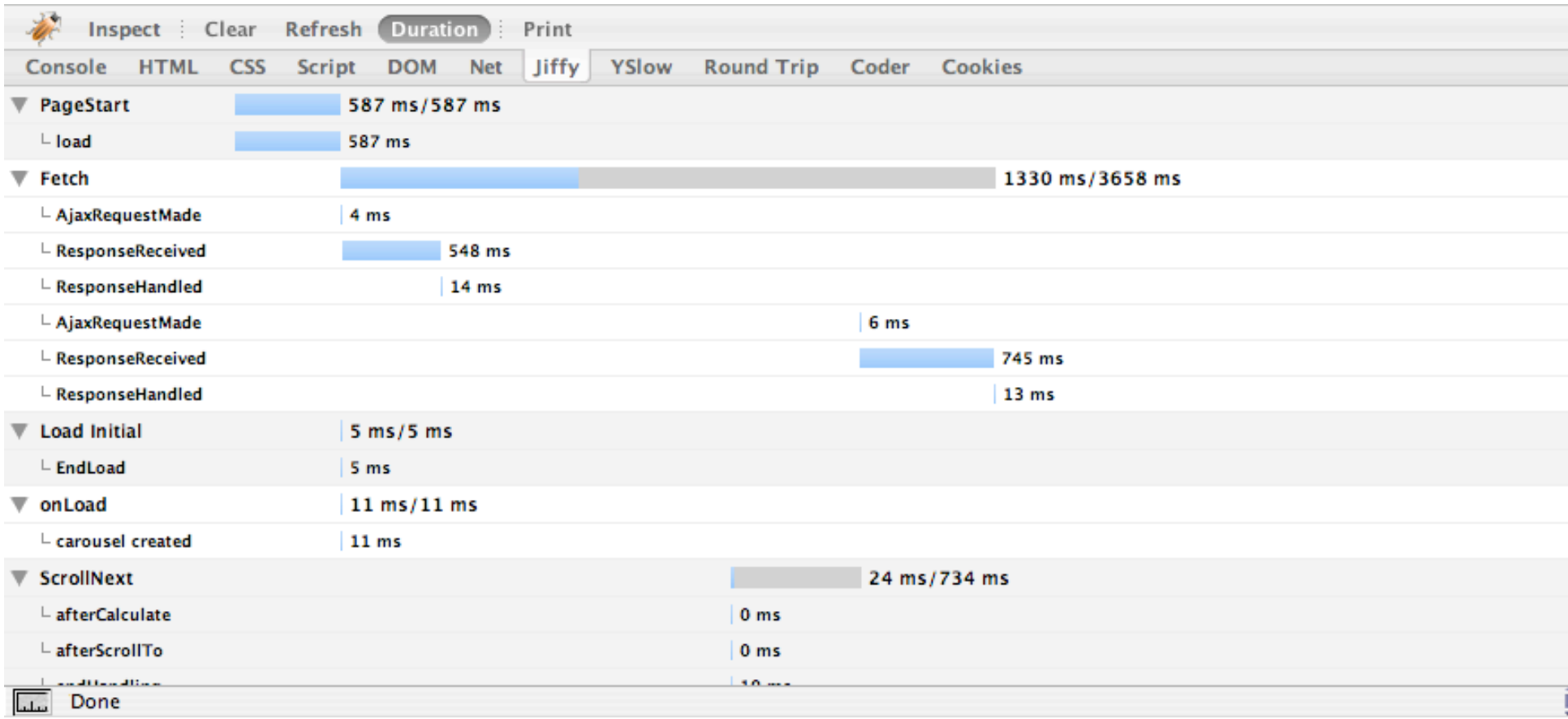


Firebug Extension

- Firebug extension that shows real-time performance measures for a given page



Jiffy Firebug Extension



Performance Improvements

Performance Steps

- GZIP HTML, Javascript and CSS
  - (exception old netscape browsers and IE6 gets only compressed HTML)
- Far future expires header for Javascript & CSS
- Turn off etags
  - These often force unnecessary requests, we don't use them so turning them off can help performance.
- Proxy Cache Configuration
  - For browsers behind a proxy cache we tell the proxy server that we are handling this. This prevents some errors that might occur when a proxy cache server tries to serve up cached content for a browser that doesn't expect it (already served the URL to one that did).

GZIP

NETFLIX

Bill Scott | Your Account | Buy / Redeem Gift | Help

Browse DVDs | Watch Instantly | Your Queue | Movies You'll ♥ | Friends & Community | DVD Sale $5.99

Movies, actors, directors, genres — Search

Welcome, Bill Scott

Your Queue

Show all DVD and Instant activity

Rate Recently Viewed Movies

DVD (266) | Instant (287)

DVDs At Home

	Movie Title	Instant	Star Rating	Shipped	Est. Arrival	
1.	August Rush		★★★★☆	08/25/08	08/26/08	Report Problem
+	The Kite Runner		★★★★½	05/27/08	05/28/08	Report Problem

DVD Queue (257)

See Queue tips — Update DVD Queue

List Order		Movie Title	Instant	Star Rating	Genre	Expected Availability	Remove
1		Battlestar Galactica: The Miniseries		★★★★★	Television	Now	
2	TOP ▲	Battlestar Galactica: Season 1: Disc 2		Series	Television	Now	
3	TOP ▲	Battlestar Galactica: Season 1: Disc 3		Series	Television	Now	
4	TOP ▲	Battlestar Galactica: Season 1: Disc 4		Series	Television	Now	
5	TOP ▲	Battlestar Galactica: Season 1: Disc 5		Series	Television	Now	
6	TOP ▲	Saints and Soldiers		★★★½	Drama	Now	
7	TOP ▲	MST3K: Mixed-Up Zombies		★★★★	Television	Long Wait	

253	TOP	Mongol		Foreign	Releases on DVD Oct 14, 2008	
254	TOP	Spider-Man 3		Action & Adventure	Now	
255	TOP	Arrested Development: Season 1: Disc 1		Television	Now	
256	TOP	Arrested Development: Season 1: Disc 2	Series	Television	Now	
257	TOP	Arrested Development: Season 1: Disc 3	Series	Television	Now	
		Enter Movie Title Here				

Update DVD Queue

Saved DVDs (9)

Movie Title	Instant	Star Rating	Genre	Availability	Remove
Get Smart			Comedy	Nov 2008	
Chronicles of Narnia: Prince Caspian			Children & Family	Dec 2008	
12			Foreign	Unknown	
Aladdin: Platinum Edition			Children & Family	Unknown	
Dexter: Season 3			Television	Unknown	
Katyn			Foreign	Unknown	
The Marx Brothers in a Nutshell			Documentary	Unknown	
The Spanish Prisoner			Drama	Unknown	
VeggieTales: Lyle the Kindly Viking			Children & Family	Unknown	

Update Saved DVDs

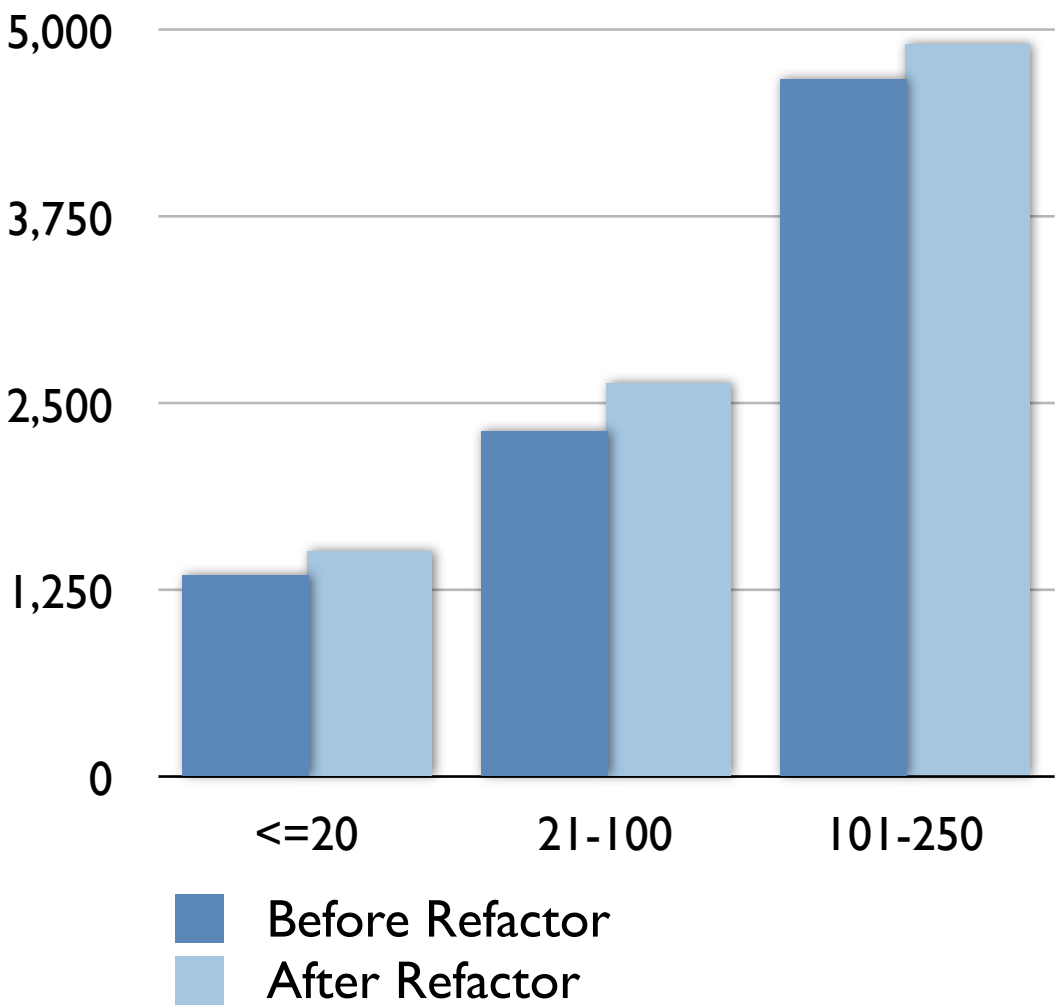
Questions? Visit our FAQ section.

Rewriting Queue

- Queue was completely re-written from scratch
  - Changed from Java generated HTML to JSP generated HTML
  - Migrated to Struts 2 Framework
- Idea was to radically clean up old crusty code

Queue Performance Degraded

- Side effect: Larger payload due to whitespace in JSP



Queue - Payload Sizes

Current Queue (in production)

Empty Cache	Primed Cache
592.5K 1HTML/Text	592.5K 1HTML/Text
231.3K 4JavaScript Files	0.0K 4JavaScript Files
70.3K 2Stylesheet Files	0.0K 2Stylesheet Files
22.4K23CSS Images	0.0K23CSS Images
48.5K46Images	0.0K46Images
965.2K**Total size**	592.5K**Total size**
76HTTP requests	**76HTTP requests**

New Queue (for 4/24)

Empty Cache	Primed Cache
804.8K 1HTML/Text	804.8K 1HTML/Text
284.5K 3JavaScript Files	284.5K 3JavaScript Files
73.4K 2Stylesheet Files	73.4K 2Stylesheet Files
8.0K20CSS Images	0.0K20CSS Images
55.0K60Images	0.0K60Images
1225.8K**Total size**	1162.8K**Total size**
86HTTP requests	**86HTTP requests**

Gzip Components

- Can Gzip scripts, stylesheets, JSON, XML, etc.
- Reduces response size by about 70%
- 90% of all traffic is handled by browsers that support gzip
- For those browsers that don't support gzip, apache supports Vary response headers automatically
- Requires apache configuration (mod_gzip or mod_deflate)

GZIP: Apache Configuration

```
<Proxy *>
    SetOutputFilter DEFLATE
</Proxy>

<Location />
    # GZIP COMPRESSION.
    # For all browsers turn on html, css and javascript gzip compression
    # For old browsers turn OFF all gzip compression
    # For IE6 gzip html only

    # Allow gzip compression for html, css, and javascript
    AddOutputFilterByType DEFLATE text/html text/javascript text/css application/x-javascript

    # Netscape 4.x has some problems...
    BrowserMatch ^Mozilla/4 gzip-only-text/html

    # Netscape 4.06-4.08 have some more problems
    BrowserMatch ^Mozilla/4\.0[678] no-gzip

    # MSIE masquerades as Mozilla, but it is fine
    BrowserMatch \bMSIE\s7 !no-gzip !gzip-only-text/html

    # Turn off gzip for images, pdf, zips and swfs
     SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
     SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary
     SetEnvIfNoCase Request_URI \.zip$ no-gzip dont-vary
     SetEnvIfNoCase Request_URI \.swf$ no-gzip dont-vary

 </Location>
```

FFE, Cache Control, eTags

```
<LocationMatch "\.(css$|js$)">
    # Far Future Expires for Javascript and CSS
    Header set Expires "Thu, 15 Apr 2020 20:00:00 GMT"
</LocationMatch>


<Location />
    Header unset ETag
    FileETag None

    #Header append Vary User-Agent env=!dont-vary
    Header set Cache-Control "private"

 </Location>
```

GZIP, FFE, eTags, Cache: Results

- Queue Payload improvement

- Network outbound traffic cut in half

GZIP, FFE, eTags, Cache: Results

- Queue Payload improvement

Empty Cache		Primed Cache	
804.8K	1HTML/Text	804.8K	1HTML/Text
284.5K	3JavaScript Files	284.5K	3JavaScript Files
73.4K	2Stylesheet Files	73.4K	2Stylesheet Files
8.0K	20CSS Images	0.0K	20CSS Images
55.0K	60Images	0.0K	60Images
1225.8K	**Total size**	1162.8K	**Total size**
86	**HTTP requests**	**86**	**HTTP requests**

- Network outbound traffic cut in half

GZIP, FFE, eTags, Cache: Results

- Queue Payload improvement

Empty Cache	Primed Cache
804.8K 1HTML/Text	804.8K 1HTML/Text
284.5K 3JavaScript Files	284.5K 3JavaScript Files
73.4K 2Stylesheet Files	73.4K 2Stylesheet Files
8.0K20CSS Images	0.0K20CSS Images
55.0K60Images	0.0K60Images
1225.8K**Total size**	1162.8K**Total size**
86HTTP requests	**86HTTP requests**

Empty Cache	Primed Cache
51.6K 1HTML/Text	
0.0K 1XMLHttpRequest	
63.9K 3JavaScript Files	51.6K 1HTML/Text
15.4K 2Stylesheet Files	0.0K 1XMLHttpRequest
21.9K23CSS Images	0.0K23CSS Images
56.3K61Images	0.0K61Images
209.3K**Total size**	51.6K**Total size**
91HTTP requests	**86HTTP requests**

- Network outbound traffic cut in half

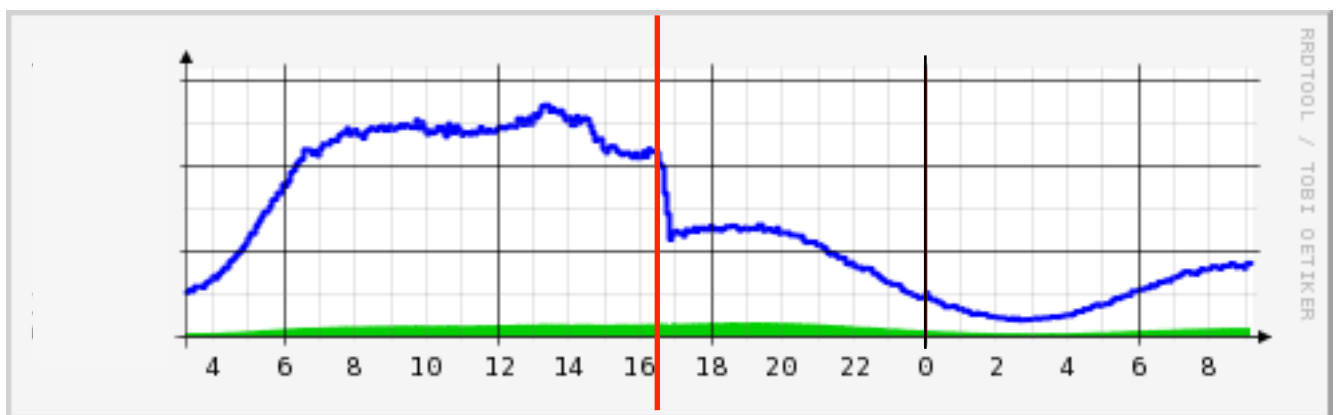
GZIP, FFE, eTags, Cache: Results

- Queue Payload improvement

Empty Cache	Primed Cache
804.8K 1HTML/Text	804.8K 1HTML/Text
284.5K 3JavaScript Files	284.5K 3JavaScript Files
73.4K 2Stylesheet Files	73.4K 2Stylesheet Files
8.0K20CSS Images	0.0K20CSS Images
55.0K60Images	0.0K60Images
1225.8K**Total size**	1162.8K**Total size**
86HTTP requests	**86HTTP requests**

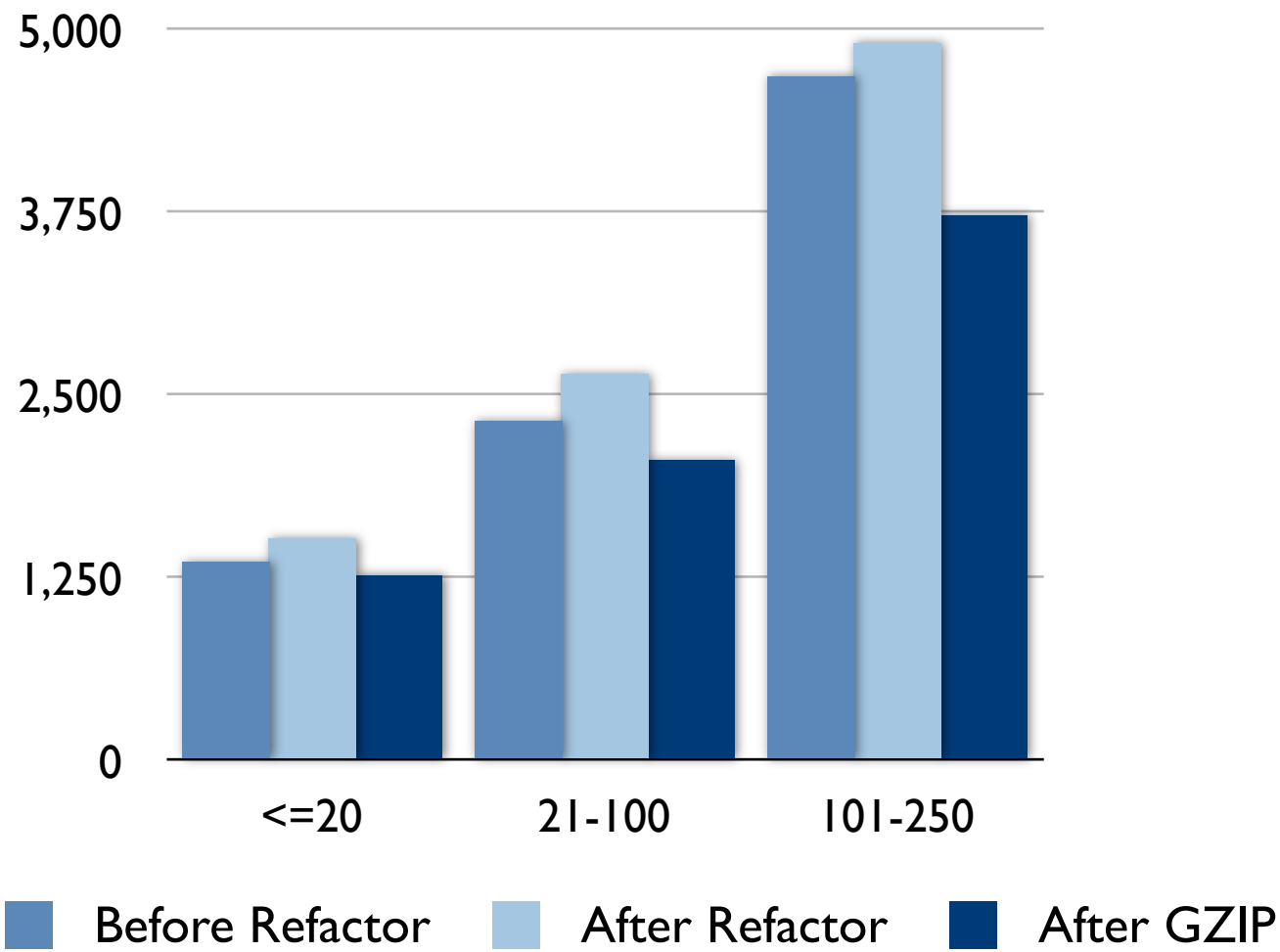
Empty Cache	Primed Cache
51.6K 1HTML/Text	
0.0K 1XMLHttpRequest	
63.9K 3JavaScript Files	51.6K 1HTML/Text
15.4K 2Stylesheet Files	0.0K 1XMLHttpRequest
21.9K23CSS Images	0.0K23CSS Images
56.3K61Images	0.0K61Images
209.3K**Total size**	51.6K**Total size**
91HTTP requests	**86HTTP requests**

- Network outbound traffic cut in half



GZIP, FFE, eTags, Cache: Results

- Performance improvement: 13-25%



Far Future Expires

- Avoids unnecessary HTTP requests
- Requires version naming of files (which we do for some files)
- Only aids those with an primed cache
- What % come with empty cache?
- At Yahoo! it averaged 40-60%

How Browser Handles Expires

- With Empty Cache (component not cached)
  - GET Request for component
- With Primed Cache (component is cached)
  - Has Far Futures Expires Header
    - Browser finds in cache
    - Determines not stale
    - Reads from local cache
  - No FFE
    - Does NOT have Far Future Expires Header
    - Browser finds in cache
    - Determines it is “stale” (expired)
    - Makes a Conditional GET
    - If it has not changed
      - Reads from Local Cache
    - If it has changed
      - Performs a GET Request

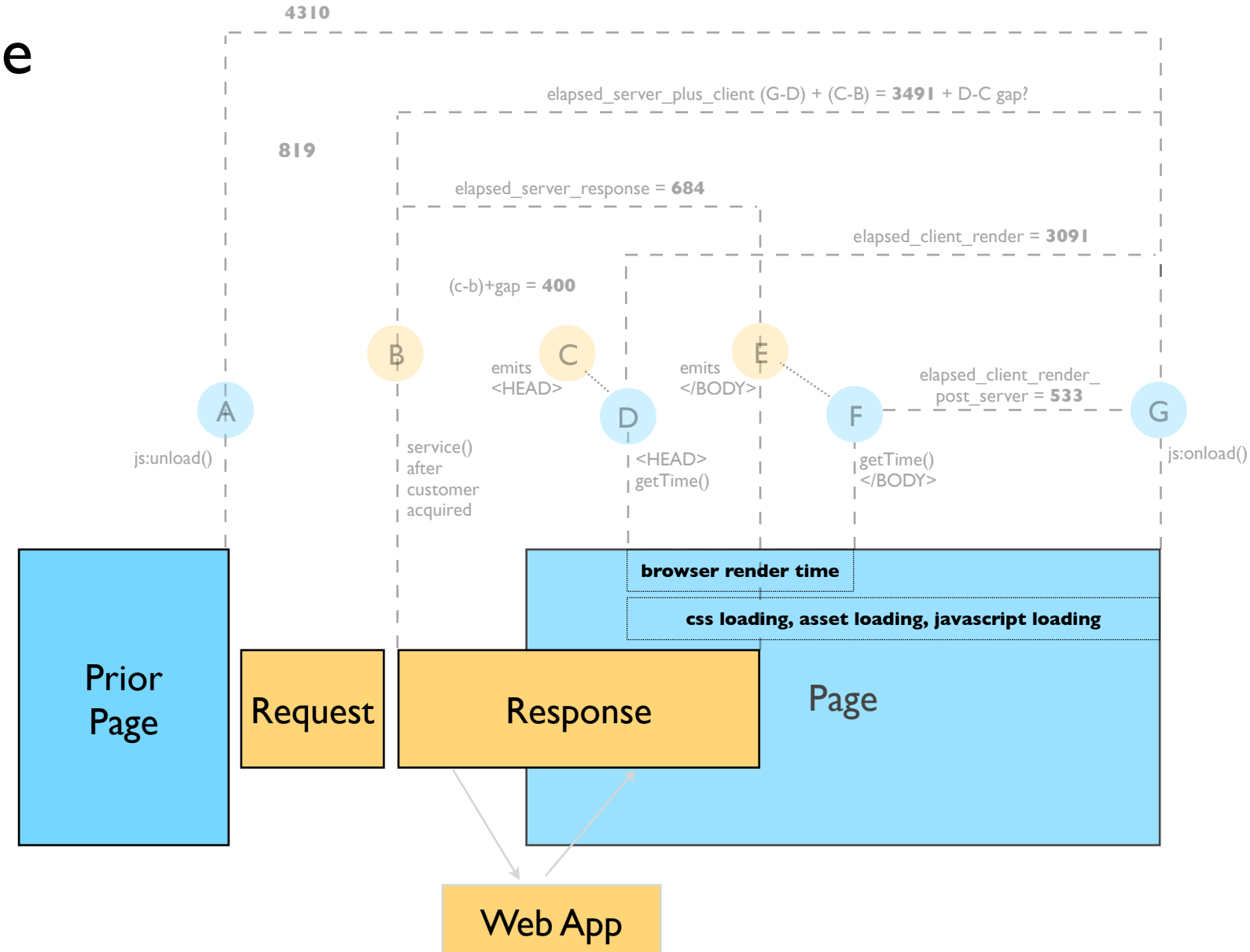
There was an issue (Safari)

- For a given HTTP request, the server may respond with a HTTP status of 204. This means 'no content' changed.
  - Used throughout Netflix site
  - Started in 2000 with first star bars on the web!
- Once we turned on gzip, Apache for some reason attempted to compress NO CONTENT responses
- Some builds of Safari 3 choked on this

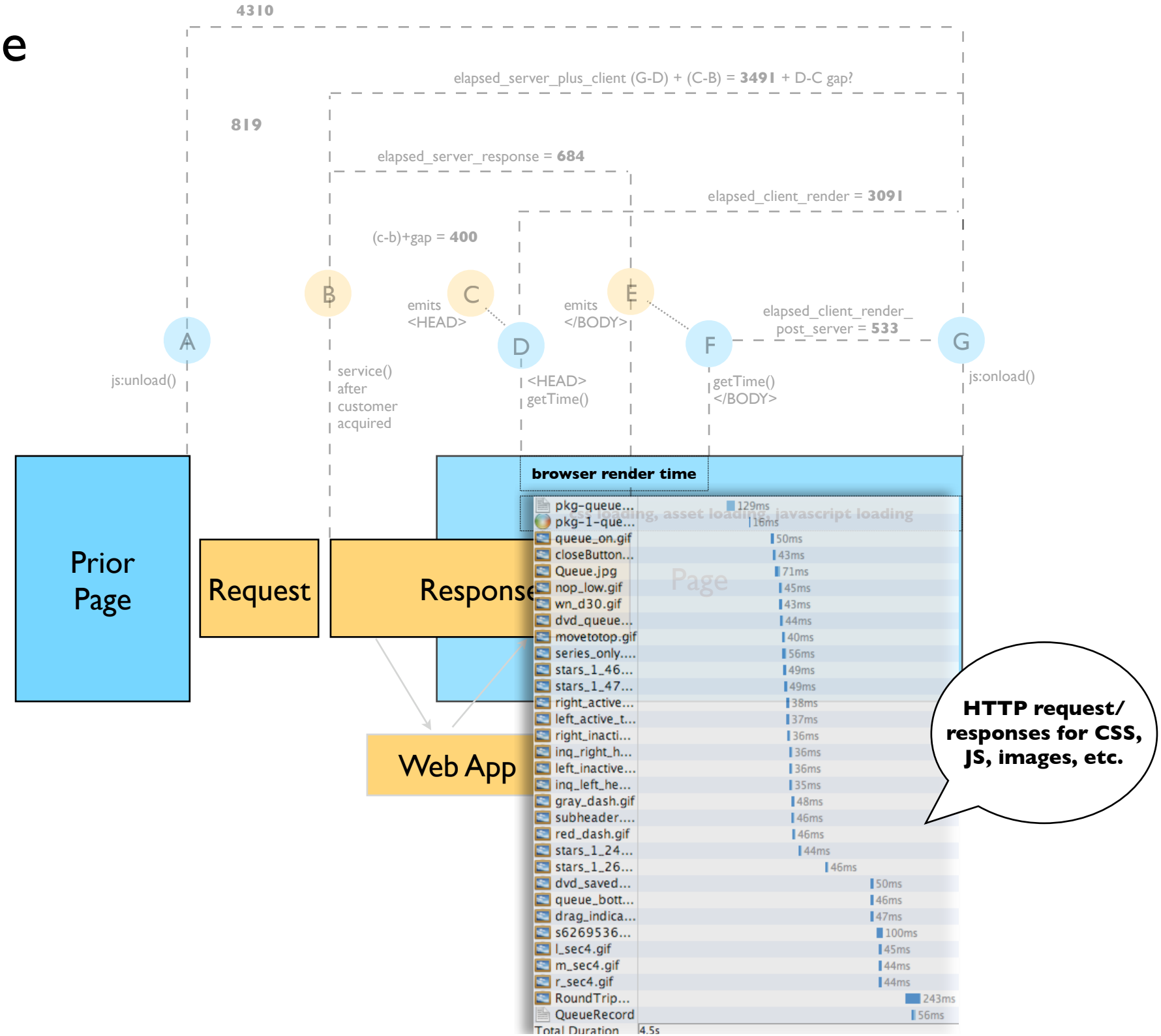
Star Bars

Star Bars everywhere

/Queue


4310
elapsed_server_plus_client (G-D) + (C-B) = 3491 + D-C gap?
819
elapsed_server_response = 684
elapsed_client_render = 3091
(c-b)+gap = 400
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
elapsed_client_render_ post_server = 533
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
browser render time
css loading, asset loading, javascript loading
Prior Page
Request
Response
Page
Web App

/Queue


4310
elapsed_server_plus_client (G-D) + (C-B) = 3491 + D-C gap?
819
elapsed_server_response = 684
elapsed_client_render = 3091
(c-b)+gap = 400
A
B
C
D
E
F
G
emits <HEAD>
emits </BODY>
elapsed_client_render_post_server = 533
js:unload()
service() after customer acquired
<HEAD> getTime()
getTime() </BODY>
js:onload()
Prior Page
Request
Response
Web App
browser render time
pkg-queue... 129ms
pkg-1-que... 16ms
queue_on.gif 50ms
closeButton... 43ms
Queue.jpg 71ms
nop_low.gif 45ms
wn_d30.gif 43ms
dvd_queue... 44ms
movetotop.gif 40ms
series_only.... 56ms
stars_1_46... 49ms
stars_1_47... 49ms
right_active... 38ms
left_active_t... 37ms
right_inacti... 36ms
inq_right_h... 36ms
left_inactive... 36ms
inq_left_he... 35ms
gray_dash.gif 48ms
subheader.... 46ms
red_dash.gif 46ms
stars_1_24... 44ms
stars_1_26... 46ms
dvd_saved... 50ms
queue_bott... 46ms
drag_indica... 47ms
s6269536... 100ms
l_sec4.gif 45ms
m_sec4.gif 44ms
r_sec4.gif 44ms
RoundTrip... 243ms
QueueRecord 56ms
Total Duration 4.5s
HTTP request/ responses for CSS, JS, images, etc.

What HTTP requests?

Where did the time go (180 item Q)			
Images	**57.1%**	3873	2500
Javascript	**7.3%**	493	300
CSS	**1.3%**	91	
HTML	**23.9%**	1624	
Redirect	**5.2%**	356	
Other	**5.1%**	346	
		6783	
Everthing Else	74.6%		
Star bars	25.4%	1723	

Where did the time go (180 item Q)		
Images	**31.7%**	3873
Star bars	25.4%	1723
Javascript	**7.3%**	493
CSS	**1.3%**	91
HTML	**23.9%**	1624
Redirect	**5.2%**	356
Other	**5.1%**	346
		6783
Everthing Else	74.6%	

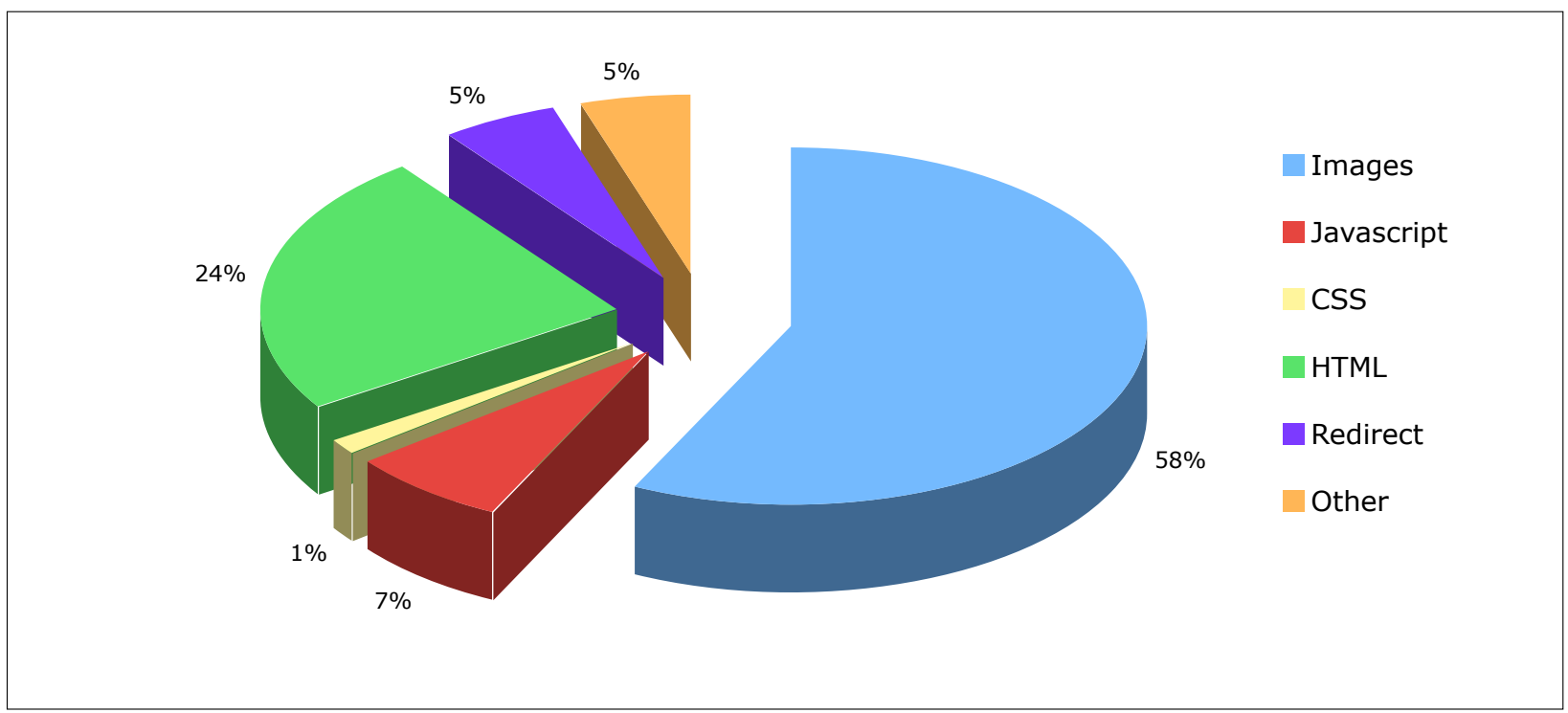
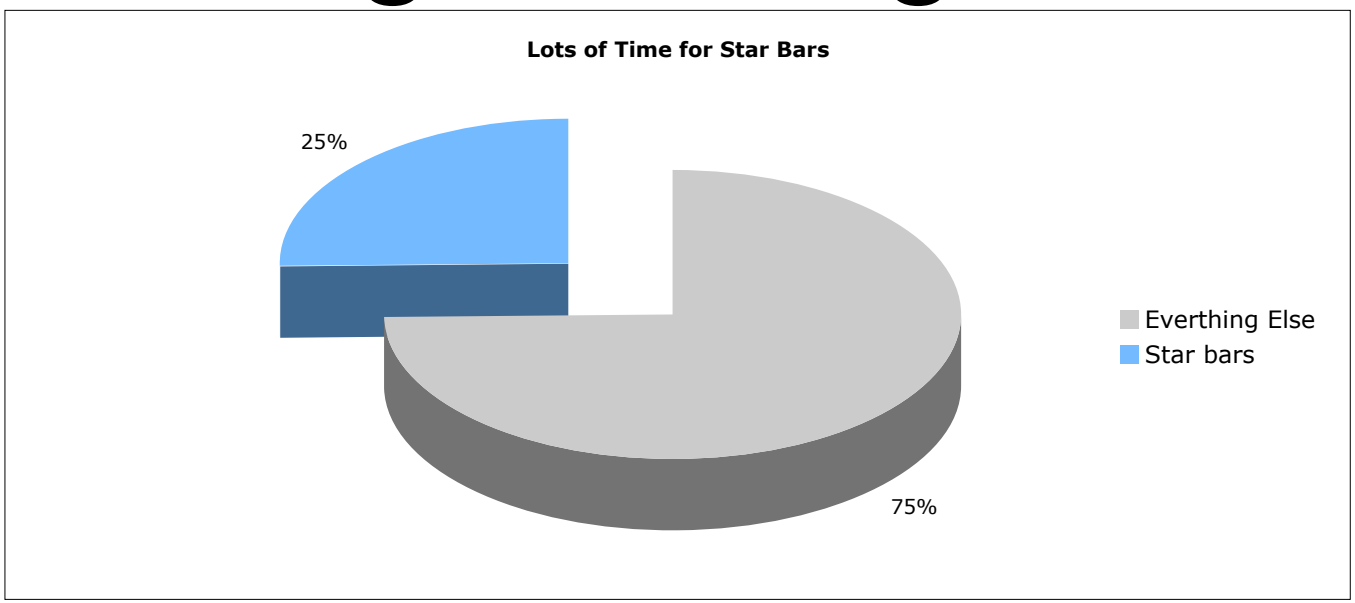
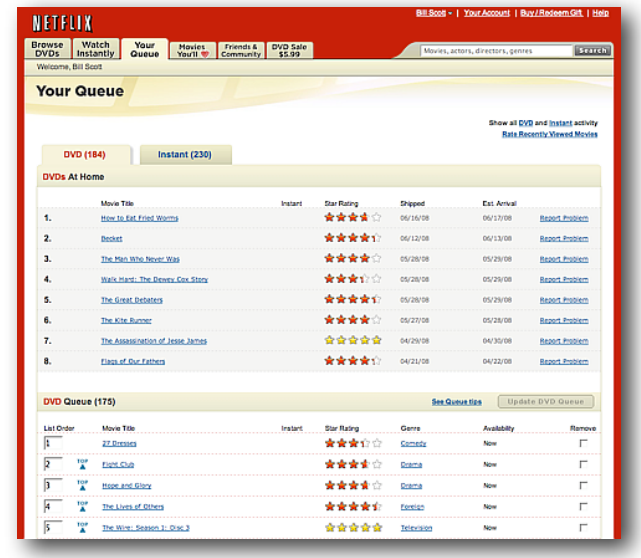
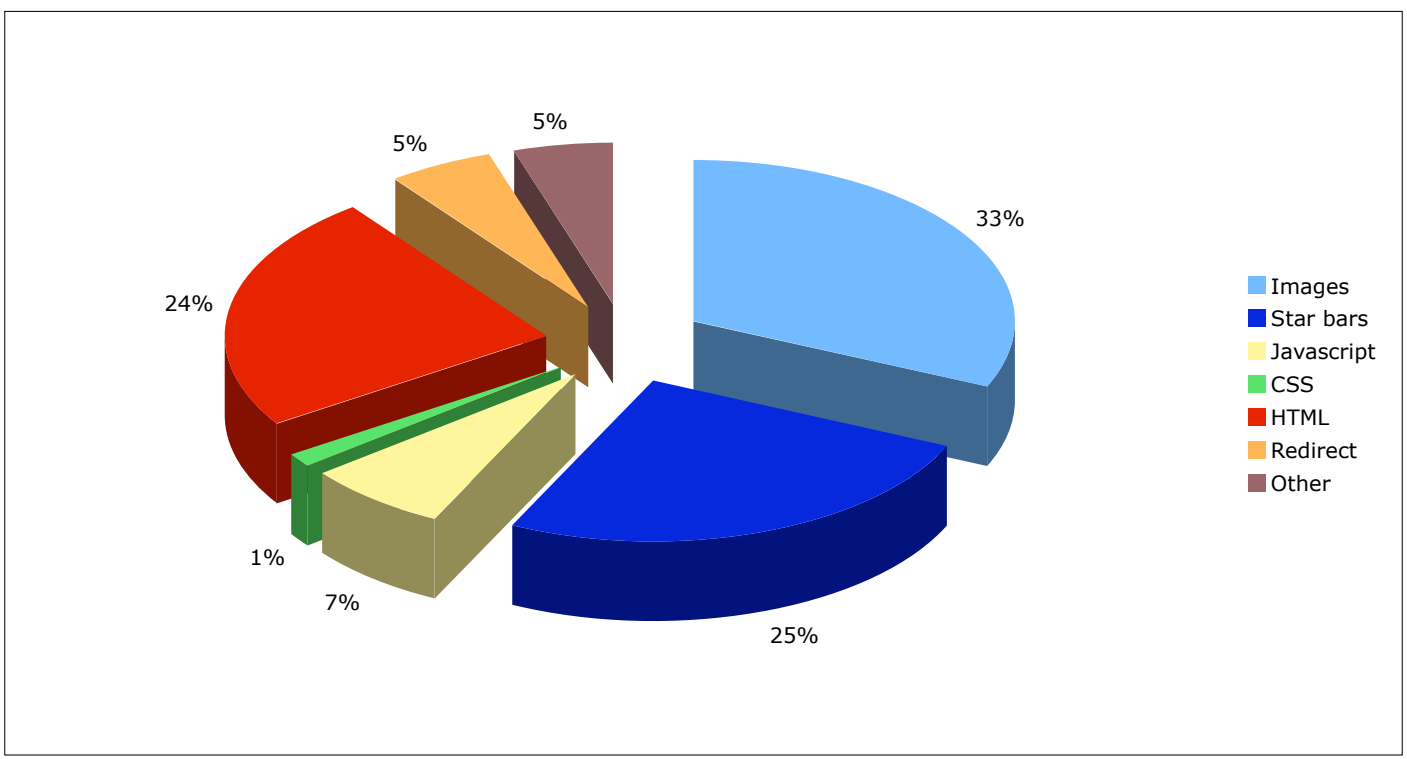


Image Fetching is Costly

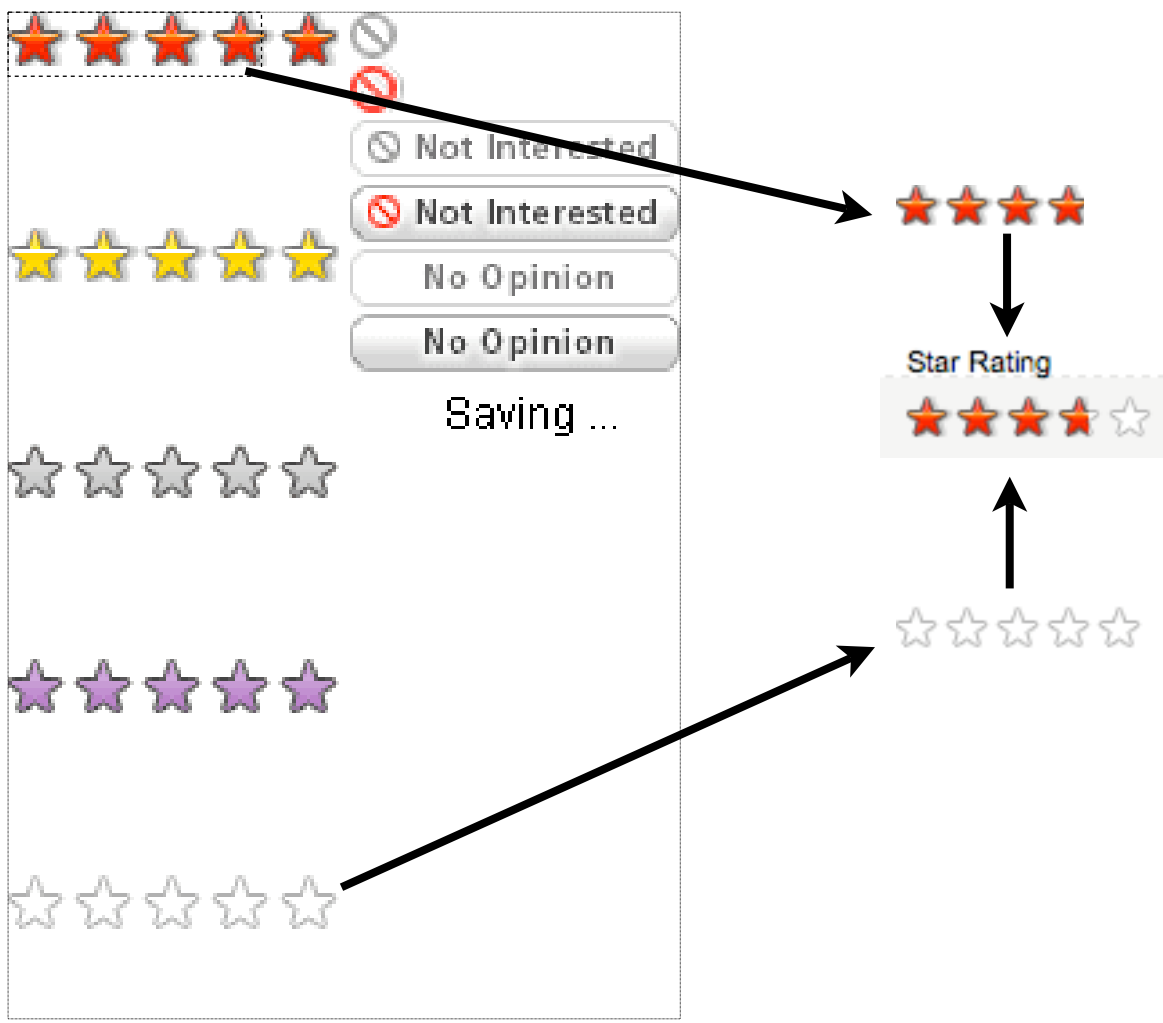


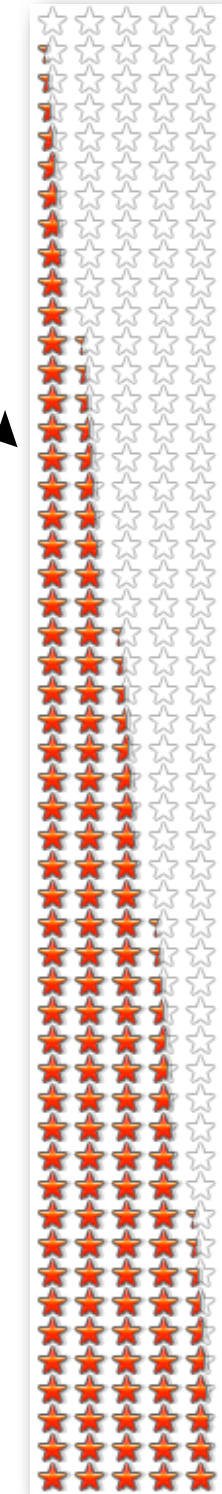




Spriting Star Bars

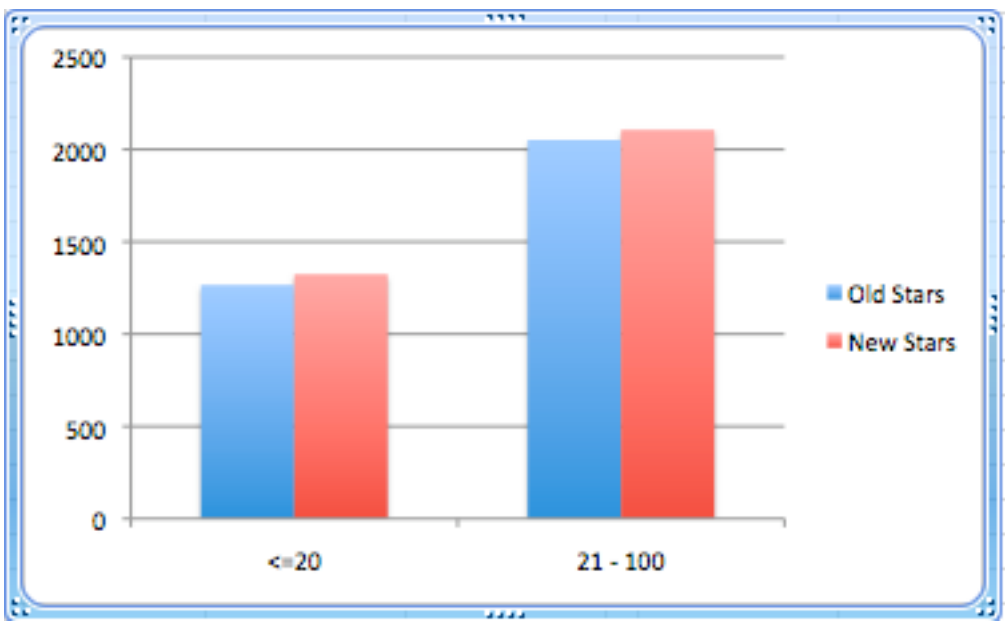
- Originally 51 separate images; not sprited
- New version single sprite for all star bars





Surprising performance hit

		11-Jun		4-Jun		
		Client	Server+Clien	Client	Server+Client	
Queue	<=20	1422	1502	1374	1433	48
	21 - 100	2511	2611	2156	2234	355
	101 - 250	4922	5122	4006	4135	916
	251 - 500	11812	12060	8670	8890	3142
		2688	2787	2359	2458	329

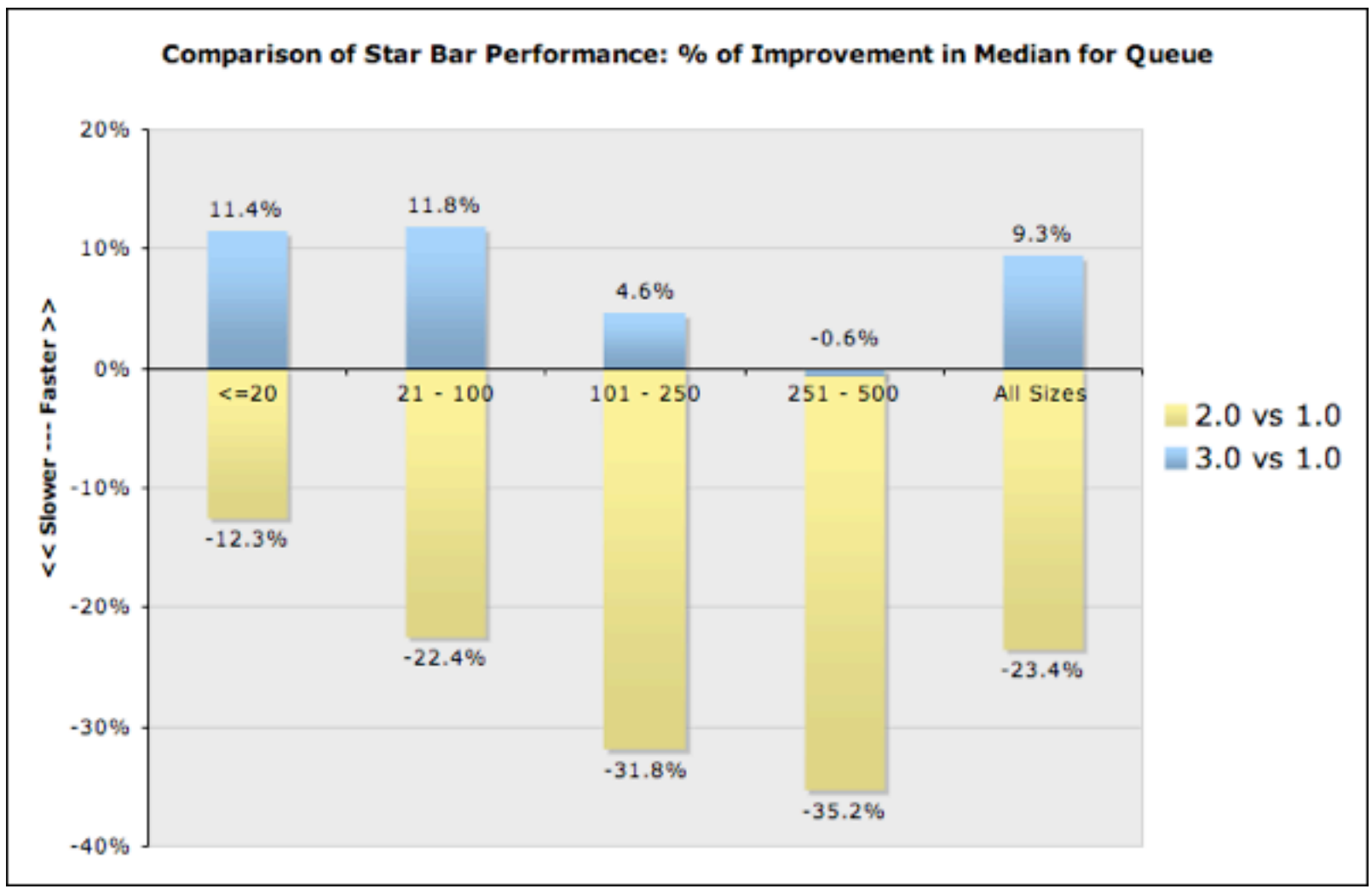


What went wrong?

- Old star bars did inline event attachment
  - onmouseover=””
  - generated with the page
- New starbars attach events on DomReady
  - with hundreds of events to attach this can cause a slow down
- Solutions (3)
  - generate inline events (yuck)
  - change to container based event model (lots of global work)
  - use a flyweight pattern of a single interactive star bar shared across the page
- Solution: inline events

After inline events: improved

- Most members experienced another 10% improvement



Other Challenges

Large Table

- IE7 & IE6 suck when rendering large tables
- Some solutions
  - Break large table into smaller chunks
    - Rendering is faster since triggering re-render of smaller table is faster than re-rendering large table
  - Used fixed layout for table to prevent re-rendering
  - Use progressive loading
    - Either a manual approach to load additional
    - Or dynamically load in the background
    - Or a combination of both

JS at Bottom & CSS at Top

- Scripts scattered throughout the page
  - Causes browsers to block while script code is executed
- Queue Example
  - To speed up **perceived** page rendering time you can pre-load background images specified in CSS
  - However, this has to be near the top
  - On IE7 & IE6 this caused significant delays (5-10 seconds on large queue)
  - Removing the performance hack decreased page load time!

```
<script>
if (document.images) {
   img1 = new Image();
   img2 = new Image();
   img1.src = "../path/to/image-01.gif";
   img2.src = "../path/to/image-02.gif";
}
</script>
```

Drag and Drop

- The sheer number of rows (up to 500) can cause a page to croak for adding drag & drop

- Issues
  - Exploding number of event handlers (use container events)
  - Dynamic cursors in IE6 (avoid)
  - Class switching (instead use style switching)
  - Extra calculations for drop targets at drag start (e.g., 500 rows)
    - Don't measure everything. Measure prototypical row. Flag exceptions.

jQuery Optimization

- Need: find drop targets dynamically
  - Normal way was ok

```
$(“td.dtc em”, “#dvd-queue”)
```

  - But more obtuse way is faster

```
$("em", "#dvd-queue")
  .filter(function() {
      return this.parentNode.className === 'dtc';
  })
```

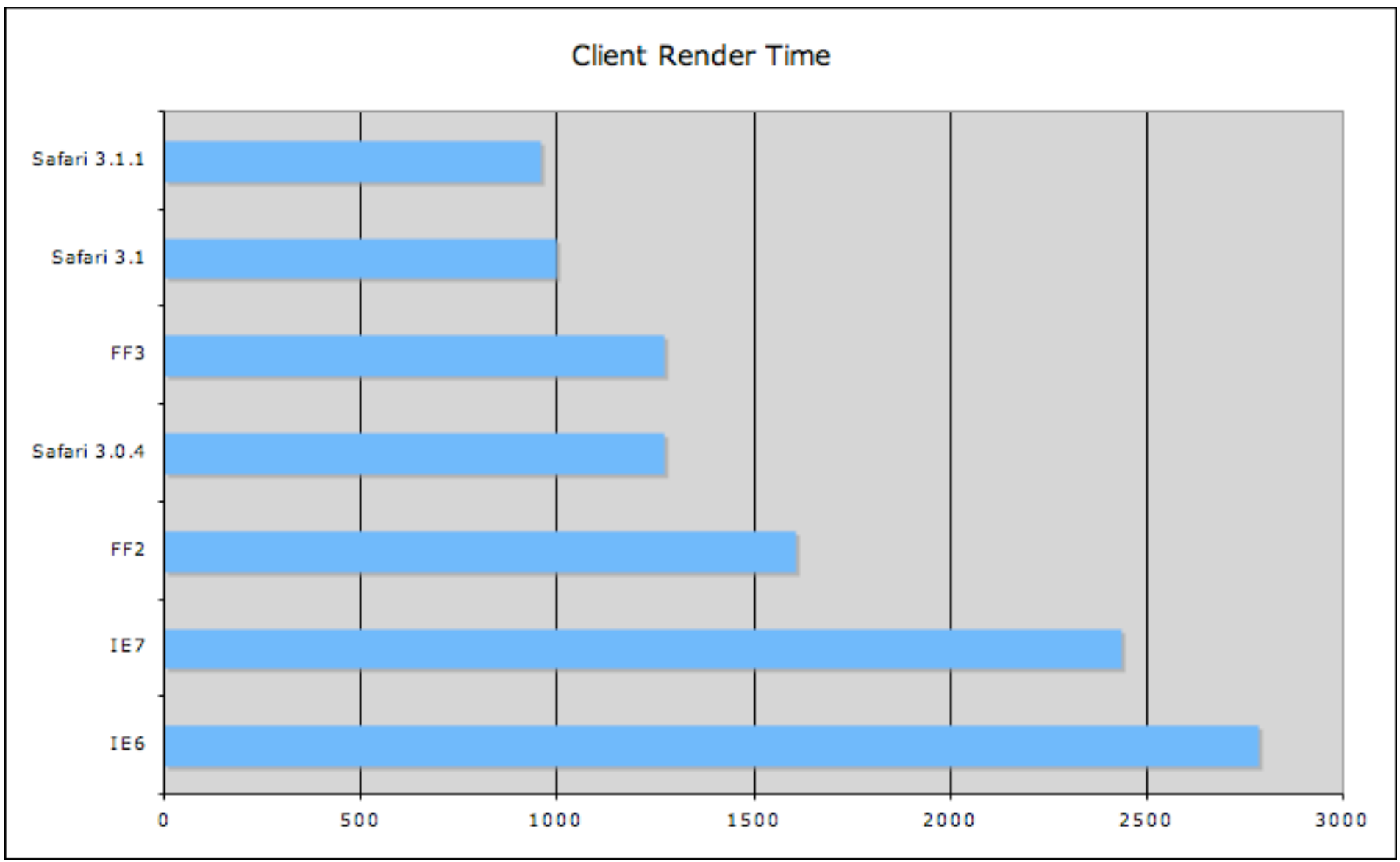
Browser Variances

Browser Variance

- Easy to forget that the experience varies greatly per browser
- Browser share
  - IE7 50%
  - IE6 20%
  - FF3 22%
  - Safari 6%

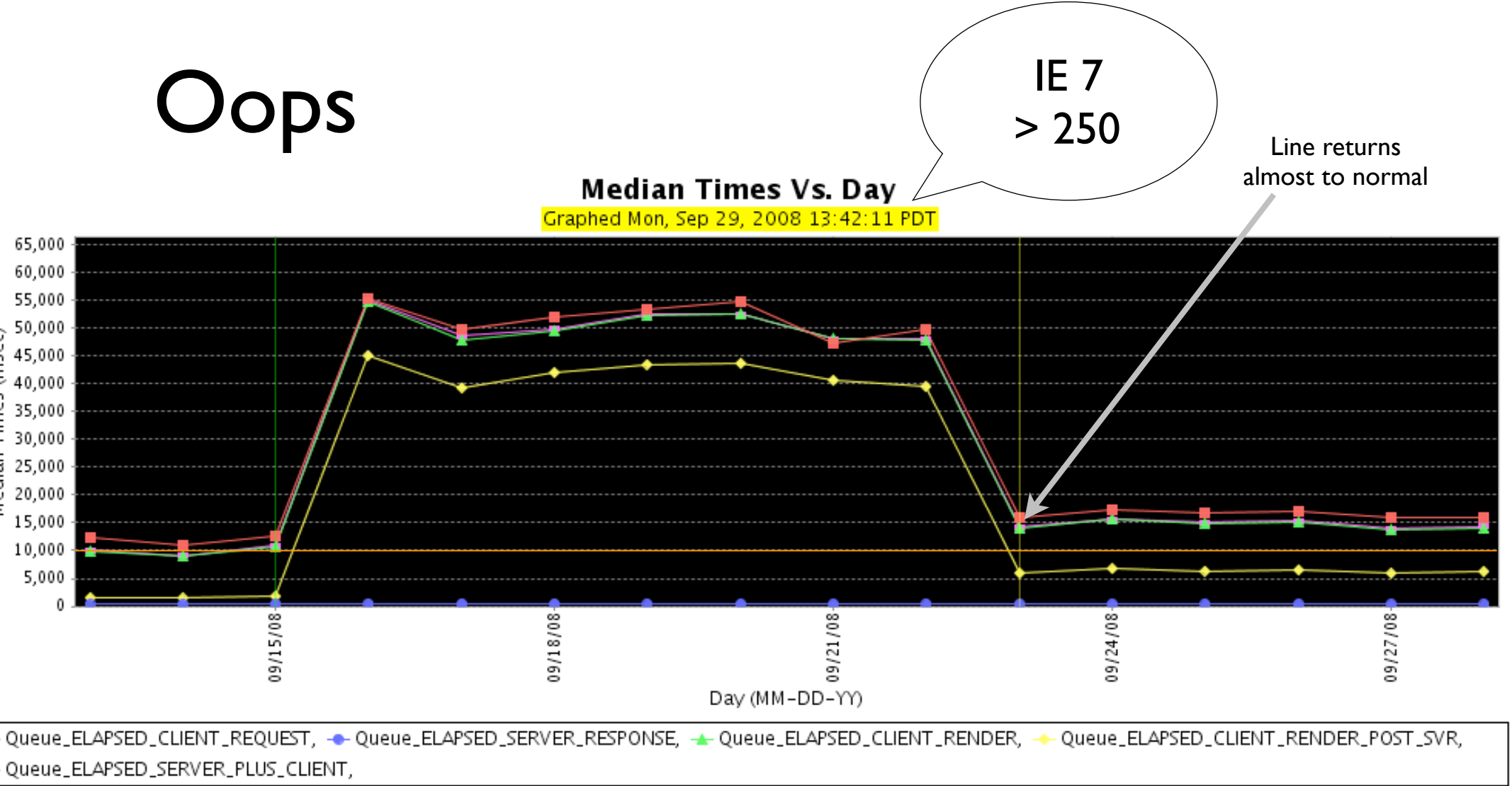
Safari much faster

- Queue median times all sizes



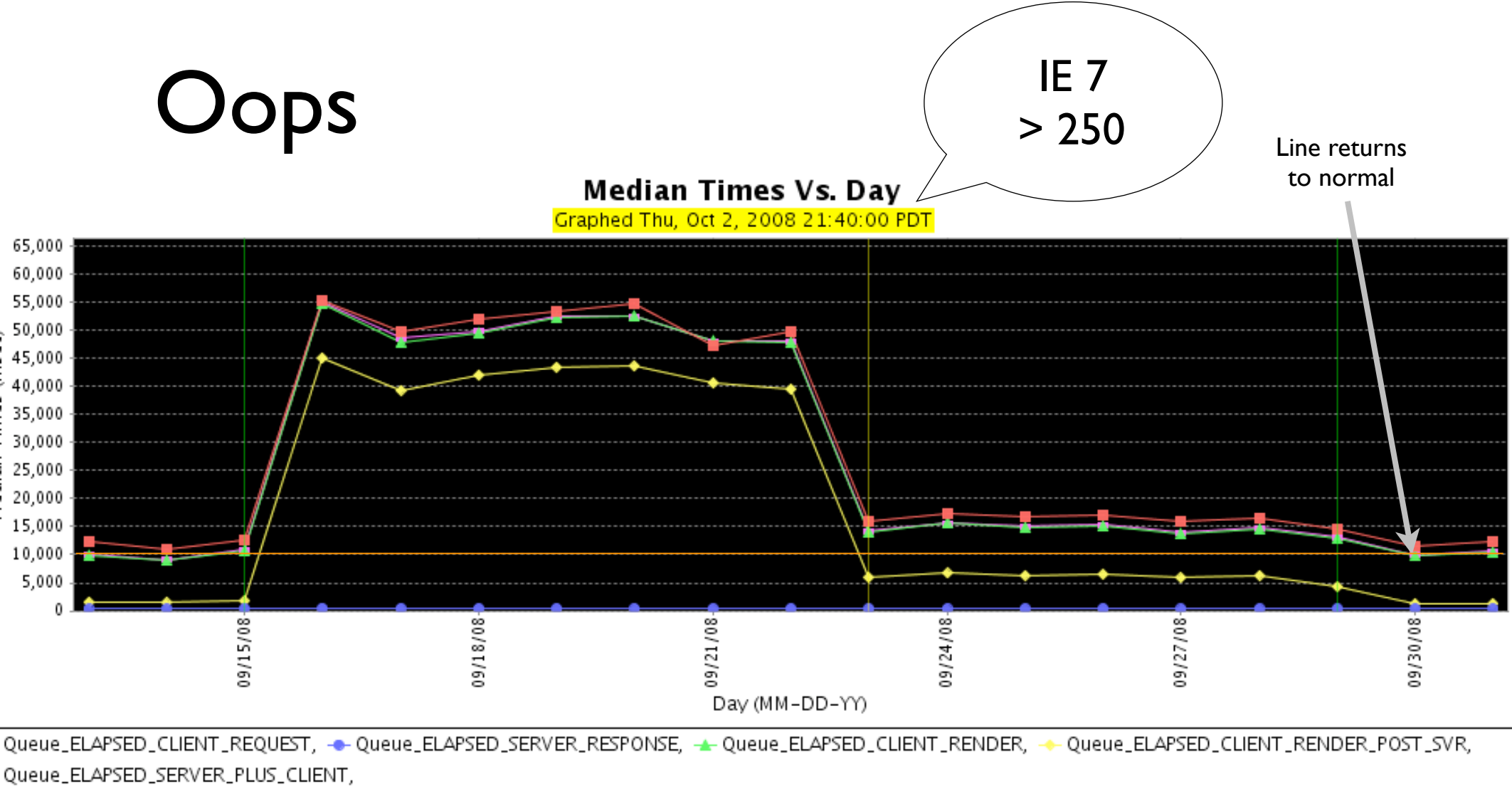
Crazy Mistakes

Oops



- Script running got triggered by the loss of a boolean on the page
- Notice slightly elevated (before moving JS to bottom)

Oops



- Simply removed a <STYLE> block in the middle of the page

Lessons Learned

Lessons

- Most of the Yahoo! recommendations are a sure bet (far futures, gzip, etag, etc.)
- The easiest win is gzip
- Not all pages are created equal
- A lot of time is spent fetching images
- Nothing beats simple design, separation of concerns, clean architecture
- It's the little stuff multiplied that you have to watch out for

Lessons

- The surest way to improve performance (and keep improving performance) is to measure, measure, measure
  - Use some way to capture the full user experience time
  - Be able to log & graph trends
  - Median works best
  - Make sure your numbers are statistically significant
- Don’t be surprised by performance degradation when you expected performance improvement

Lessons

- Browsers are not created equal
  - Some browsers will eat your lunch (read IE)
- Tools are essential
  - Firebug, round trip tracing, self-service dashboard, HammerHead, yslow, Toad, Excel, Jiffy Extension, Round Trip Extension, and old-fashioned instrumentation.
- Science is messy
  - E.O. Wilson