2.– 5. September 2013
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

X ways to improve your web application's performance

Eduard Tudenhöfner

adesso AG

Why is performance important?

A page that was **2 seconds** slower results in a **4.3%** drop in revenue/user (***Bing***)

400 ms delay cause **0.59%** drop in searches/user

(***Google***)

400 ms slowdown cause **5-9%** drop in full-page traffic

(***Yahoo***)

Introducing gzip compression resulted in **13-25%** speedup and cut outbound network traffic by **50%** (***Netflix***)

Source: www.stevesouders.com

Investing in Performance really pays off

What to expect?

Backend Performance

- Memory Optimizations & Java GC Tuning
- App Server Performance Improvements
- DB & Persistence Layer Tuning
- …
- Might require:
  - redesigning app architecture
  - adding/modifying HW
  - distributing databases

Frontend Performance

- Reducing number of requests
- Reducing transferred data
- …
- Frontend improvements:
  - require less time and resources
  - easier applicable
  - can have significant outcomes

What to expect?

Backend Performance

- Memory Optimizations & Java GC Tuning
- App Server Performance Improvements
- DB & Persistence Layer Tuning
- …
- Might require:
  - redesigning app architecture
  - adding/modifying HW
  - distributing databases

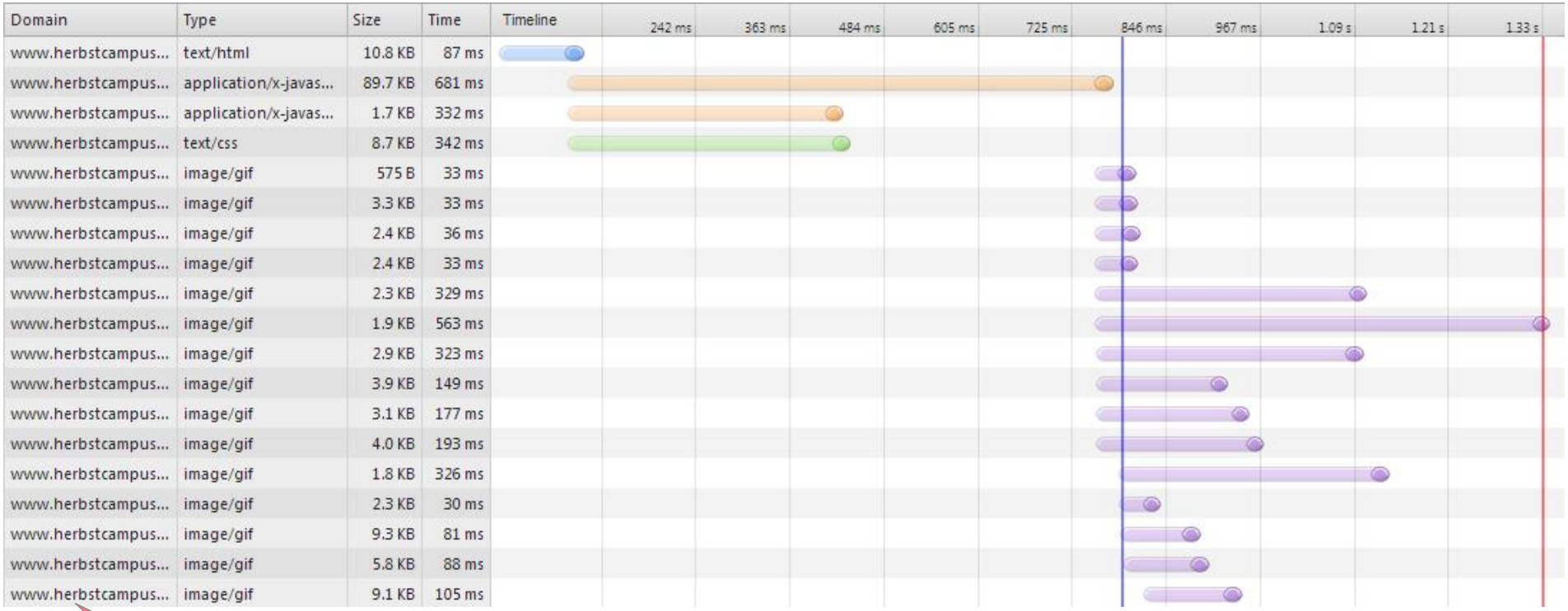
Frontend Performance

- Reducing number of requests
- Reducing transferred data
- …
- Frontend improvements:
  - require less time and resources
  - easier applicable
  - can have significant outcomes

adesso business. people. technology.

Where does the time go?

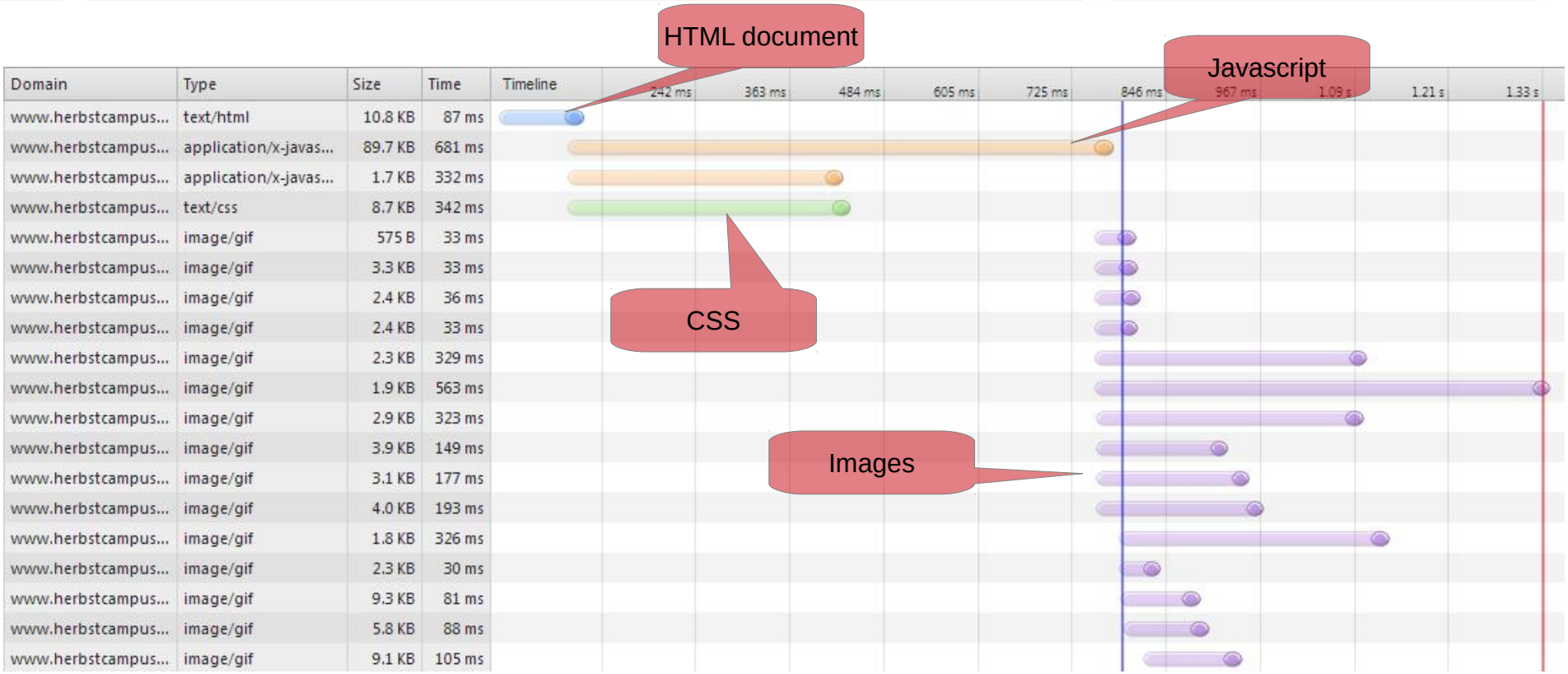
Where does the time go?



Domain	Type	Size	Time
www.herbstcampus...	text/html	10.8 KB	87 ms
www.herbstcampus...	application/x-javas...	89.7 KB	681 ms
www.herbstcampus...	application/x-javas...	1.7 KB	332 ms
www.herbstcampus...	text/css	8.7 KB	342 ms
www.herbstcampus...	image/gif	575 B	33 ms
www.herbstcampus...	image/gif	3.3 KB	33 ms
www.herbstcampus...	image/gif	2.4 KB	36 ms
www.herbstcampus...	image/gif	2.4 KB	33 ms
www.herbstcampus...	image/gif	2.3 KB	329 ms
www.herbstcampus...	image/gif	1.9 KB	563 ms
www.herbstcampus...	image/gif	2.9 KB	323 ms
www.herbstcampus...	image/gif	3.9 KB	149 ms
www.herbstcampus...	image/gif	3.1 KB	177 ms
www.herbstcampus...	image/gif	4.0 KB	193 ms
www.herbstcampus...	image/gif	1.8 KB	326 ms
www.herbstcampus...	image/gif	2.3 KB	30 ms
www.herbstcampus...	image/gif	9.3 KB	81 ms
www.herbstcampus...	image/gif	5.8 KB	88 ms
www.herbstcampus...	image/gif	9.1 KB	105 ms

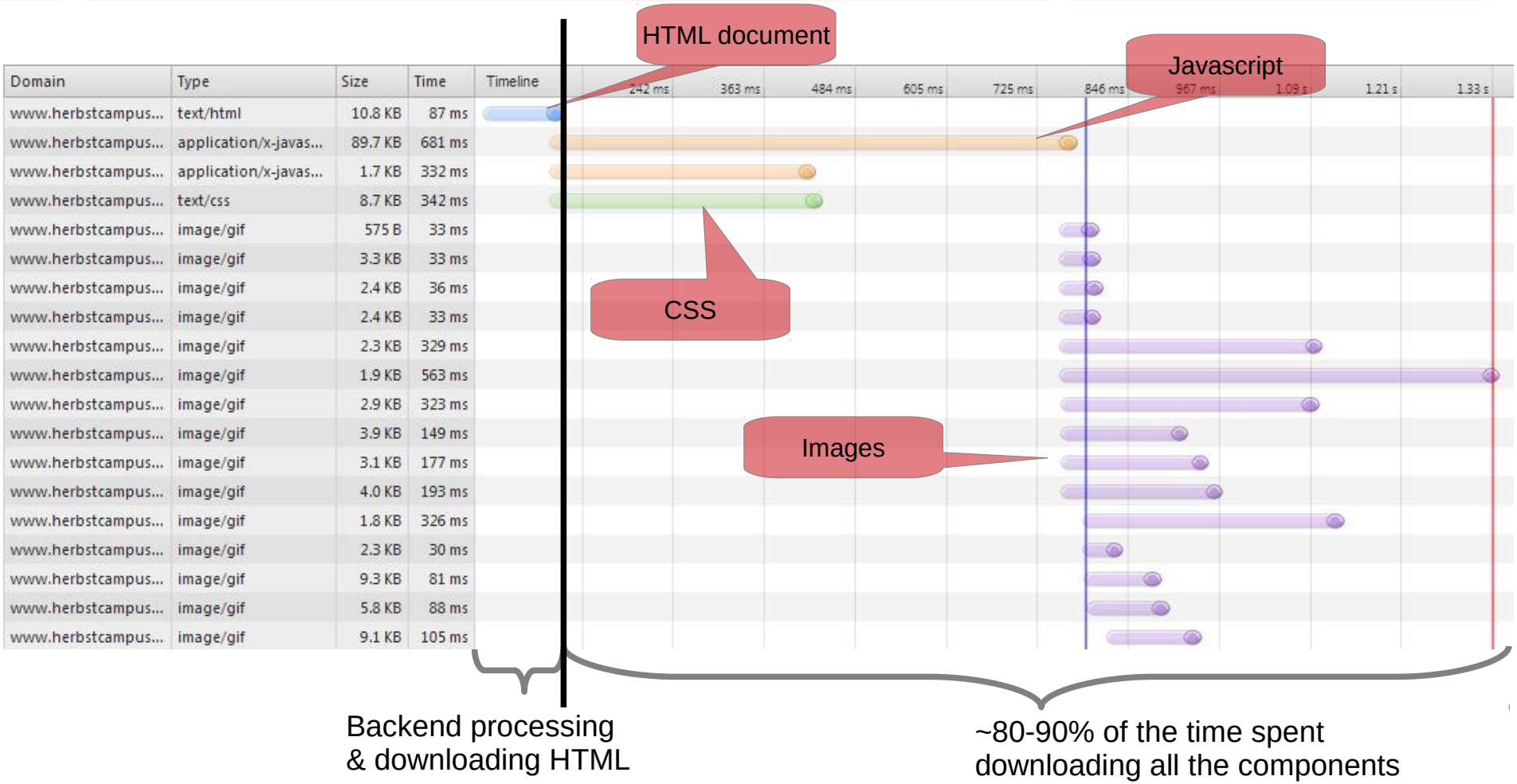
Main Herbstcampus Page

Where does the time go?



Domain	Type	Size	Time	Timeline
www.herbstcampus...	text/html	10.8 KB	87 ms	
www.herbstcampus...	application/x-javas...	89.7 KB	681 ms	
www.herbstcampus...	application/x-javas...	1.7 KB	332 ms	
www.herbstcampus...	text/css	8.7 KB	342 ms	
www.herbstcampus...	image/gif	575 B	33 ms	
www.herbstcampus...	image/gif	3.3 KB	33 ms	
www.herbstcampus...	image/gif	2.4 KB	36 ms	
www.herbstcampus...	image/gif	2.4 KB	33 ms	
www.herbstcampus...	image/gif	2.3 KB	329 ms	
www.herbstcampus...	image/gif	1.9 KB	563 ms	
www.herbstcampus...	image/gif	2.9 KB	323 ms	
www.herbstcampus...	image/gif	3.9 KB	149 ms	
www.herbstcampus...	image/gif	3.1 KB	177 ms	
www.herbstcampus...	image/gif	4.0 KB	193 ms	
www.herbstcampus...	image/gif	1.8 KB	326 ms	
www.herbstcampus...	image/gif	2.3 KB	30 ms	
www.herbstcampus...	image/gif	9.3 KB	81 ms	
www.herbstcampus...	image/gif	5.8 KB	88 ms	
www.herbstcampus...	image/gif	9.1 KB	105 ms	

Where does the time go?



Backend processing
& downloading HTML

~80-90% of the time spent
downloading all the components

Clearly visible where the time does **NOT** go:

does not go into downloading the HTML document incl. backend processing

Bandwidth / Latency

Bandwidth / Latency

Bandwidth

- Important, but is not the only factor in performance
- Higher Bandwidth good for:
  - Audio/video streaming
  - Large downloads

Latency

- primarily determined by the distance a request must travel
- Physics get in our way

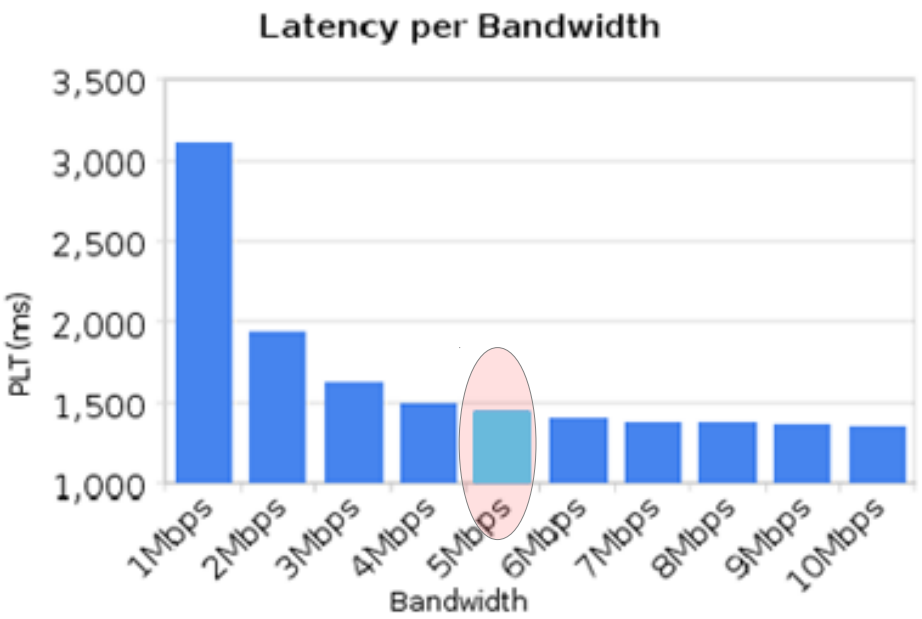
Bandwidth / Latency

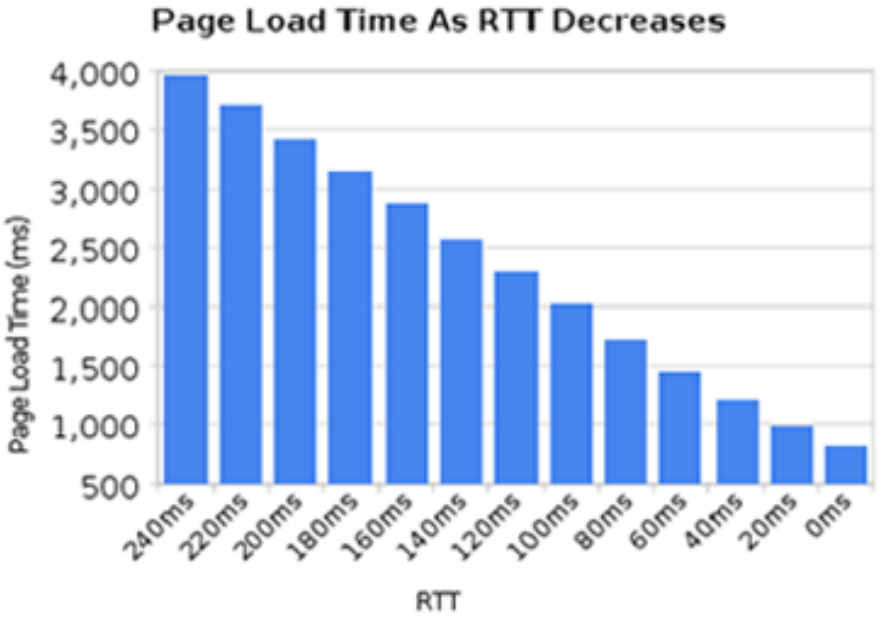
Bandwidth

- Important, but is not the only factor in performance
- Higher Bandwidth good for:
  - Audio/video streaming
  - Large downloads

Latency

- primarily determined by the distance a request must travel
- Physics get in our way





Source: More Bandwidth Doesn't Matter (much), Mike Belshe

Bandwidth / Latency

What is more important?

- Bandwidth is important, but is not the only factor

- HTTP uses short, bursty connections (for downloading web content)
  - **RTT (round-trip-time) dominates** performance more than bandwidth does

- Faster browsing experience → **reduce RTT**

How to improve?

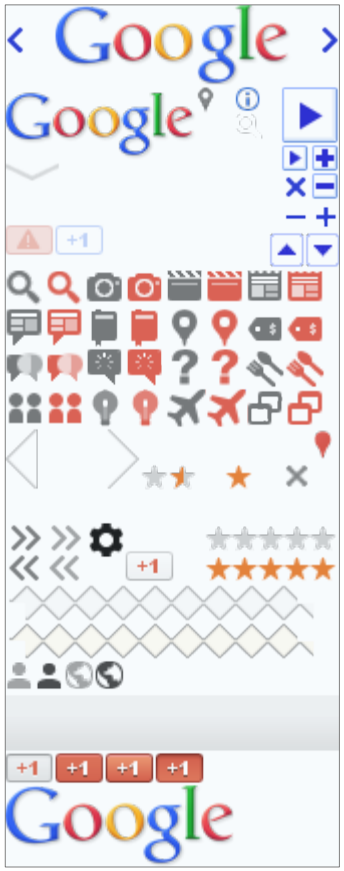
How to improve? → Fewer HTTP Requests

Fewer HTTP Requests

- Simple Rule: **less components to download = less round trips**
- But: we don't want to make tradeoffs between performance and design
- What to do?
  - Image Sprites
  - Combine JS / CSS
  - Improve caching (more to come in own chapter)

How to improve? → Fewer HTTP Requests

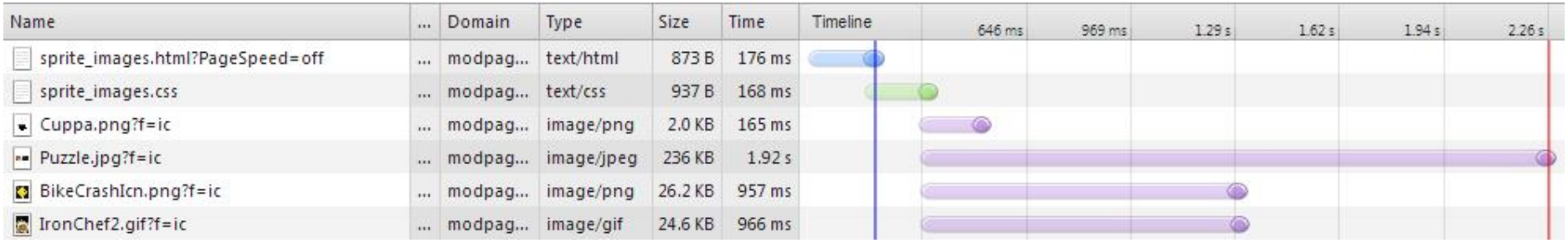
CSS Image Sprites



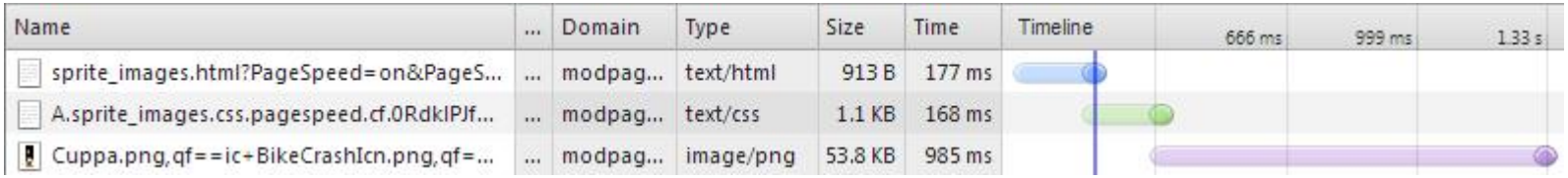
How to improve? → Fewer HTTP Requests

CSS Image Sprites

Before



Name	...	Domain	Type	Size	Time	Timeline
sprite_images.html?PageSpeed=off	...	modpag...	text/html	873 B	176 ms	
sprite_images.css	...	modpag...	text/css	937 B	168 ms	
Cuppa.png?f=ic	...	modpag...	image/png	2.0 KB	165 ms	
Puzzle.jpg?f=ic	...	modpag...	image/jpeg	236 KB	1.92 s	
BikeCrashIcn.png?f=ic	...	modpag...	image/png	26.2 KB	957 ms	
IronChef2.gif?f=ic	...	modpag...	image/gif	24.6 KB	966 ms	



Name	...	Domain	Type	Size	Time	Timeline
sprite_images.html?PageSpeed=on&PageS...	...	modpag...	text/html	913 B	177 ms	
A.sprite_images.css.pagespeed.cf.0RdkIPJf...	...	modpag...	text/css	1.1 KB	168 ms	
Cuppa.png,qf==ic+BikeCrashIcn.png,qf=...	...	modpag...	image/png	53.8 KB	985 ms	

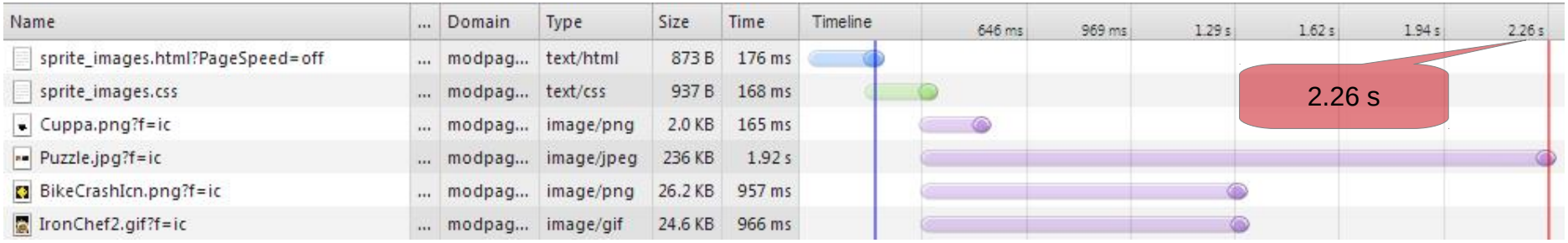
After

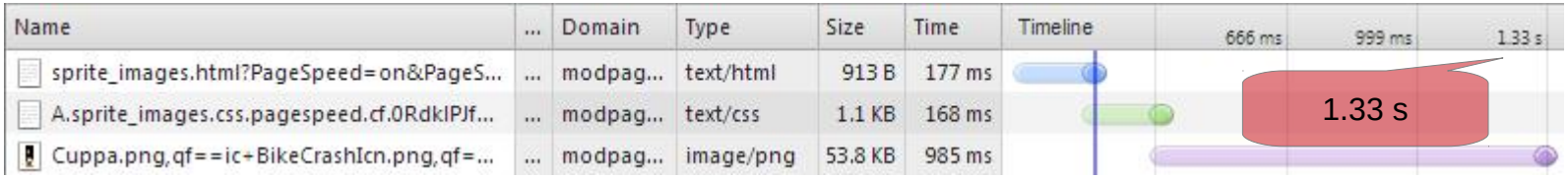
Examples implemented at: http://modpagespeed.com

How to improve? → Fewer HTTP Requests

CSS Image Sprites

Before





After

Savings: ~41%

Examples implemented at: http://modpagespeed.com

How to improve? → Fewer HTTP Requests

Combine Javascript

Before

Name	...	Domain	Type	Size	Time	Timeline	191 ms	286 ms	381 ms	476 ms	572 ms	667 ms
combine_javascript.html?PageSpeed=off	...	modpag...	text/html	565 B	331 ms							
combine_javascript1.js	...	modpag...	applicatio...	551 B	166 ms							
combine_javascript2.js	...	modpag...	applicatio...	441 B	334 ms							

Name	...	Domain	Type	Si...	T...	Timeline	174 ms	261 ms	348 ms
combine_javascript.html?ModPagespeed=on&ModPagespeedFilters=combine_javascript	...	www.m...	text/html	6...	1...				
combine_javascript1.js+combine_javascript2.js.pagespeed.jc.zYiUaxFS8I.js	...	www.m...	applicatio...	7...	1...				

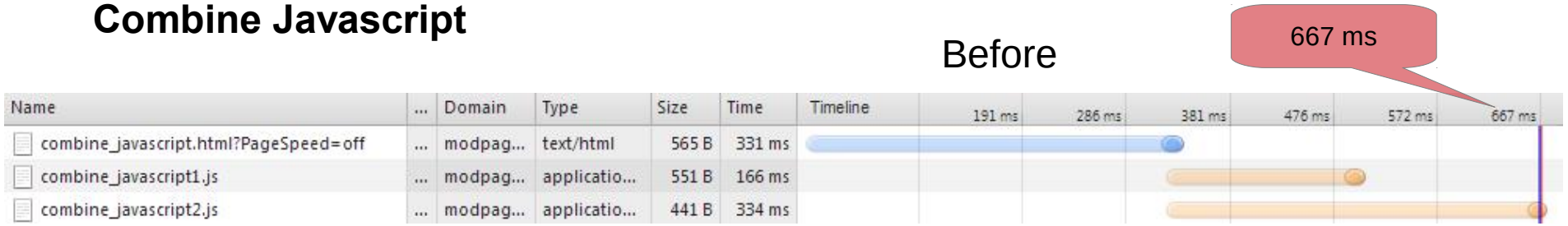
After

Examples implemented at: http://modpagespeed.com

How to improve? → Fewer HTTP Requests

Combine Javascript

Before



After

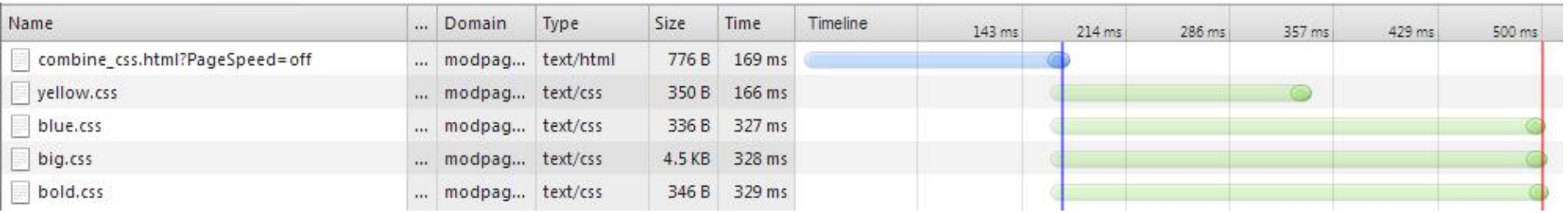
Savings: ~48%

Examples implemented at: http://modpagespeed.com

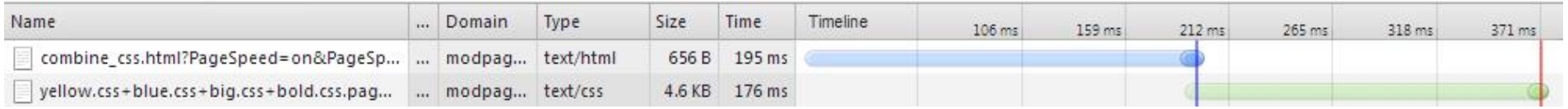
How to improve? → Fewer HTTP Requests

Combine CSS

Before



Name	...	Domain	Type	Size	Time	Timeline
combine_css.html?PageSpeed=off	...	modpag...	text/html	776 B	169 ms	
yellow.css	...	modpag...	text/css	350 B	166 ms	
blue.css	...	modpag...	text/css	336 B	327 ms	
big.css	...	modpag...	text/css	4.5 KB	328 ms	
bold.css	...	modpag...	text/css	346 B	329 ms	



Name	...	Domain	Type	Size	Time	Timeline
combine_css.html?PageSpeed=on&PageSp...	...	modpag...	text/html	656 B	195 ms	
yellow.css+blue.css+big.css+bold.css.pag...	...	modpag...	text/css	4.6 KB	176 ms	

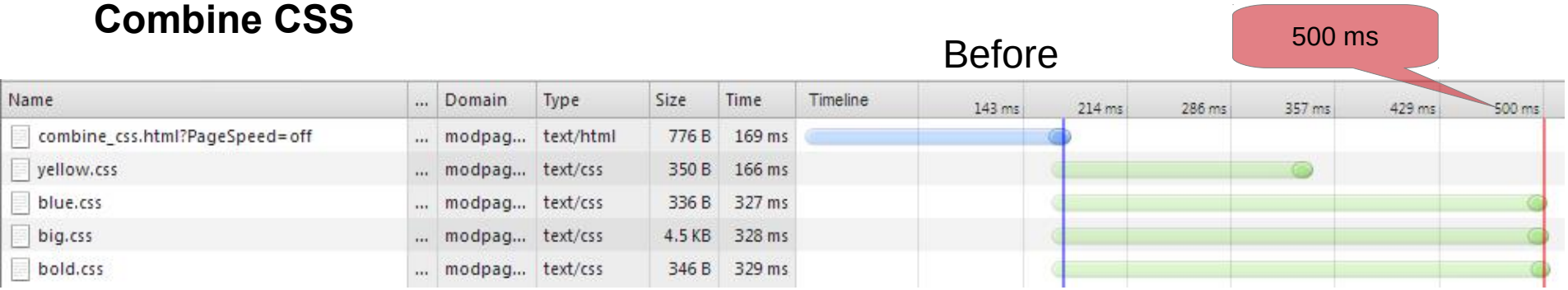
After

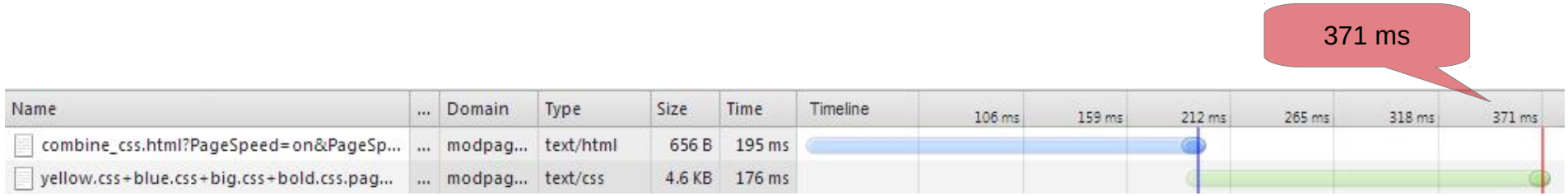
Examples implemented at: http://modpagespeed.com

How to improve? → Fewer HTTP Requests

Combine CSS

Before





After

Savings: ~26%

Examples implemented at: http://modpagespeed.com

How to improve? → Use a CDN

CDN (Content Delivery Network)

- Simple Rule: **content closer to the user = lower latency**

- to implement geographically dispersed content:
  - we want to bring **static** content closer to the user
  - we don't want to redesign our web app to work in a distributed way (clustering, …)

- dispersing content is much easier than dispersing an entire application

- nice benefit → spikes in traffic during peak load times can be absorbed

- CDN Providers (taken from http://goo.gl/l4UmJC)
  - Akamai
  - Limelight Networks
  - CacheFly
  - CloudFare
  - MaxCDN

How to improve? → Improve Caching

Caching

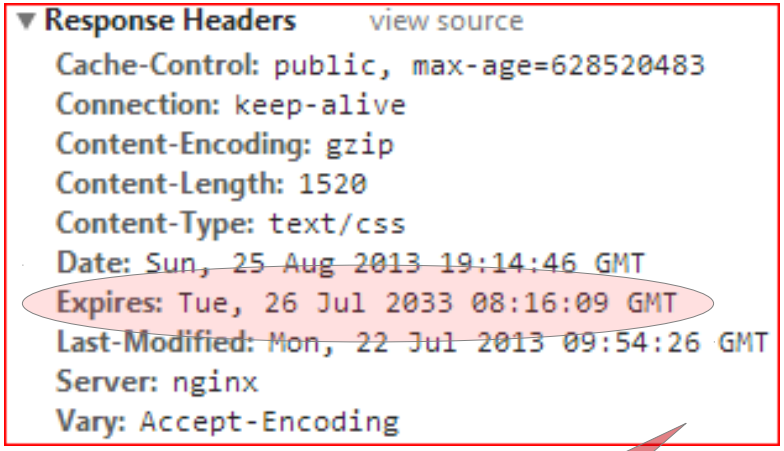
- We want to maximize the browser's caching capabilities
- First-time visitor might have to make much more # of requests than a returning user
- What to cache?
  - Images, Scripts, Stylesheets, Flash, …
- How to handle updates to cached components?
  - Rename them (e.g. use version numbers)
- How to cache?
  - add Expires / Cache-Control Header
  - configure ETags

How to improve? → Improve Caching

Expires Header

- Tells the browser that this response won't be stale until a given date/time
- # of requests is reduced by one
- mod_expires

```
ExpiresDefault "modification plus 10 years"
```

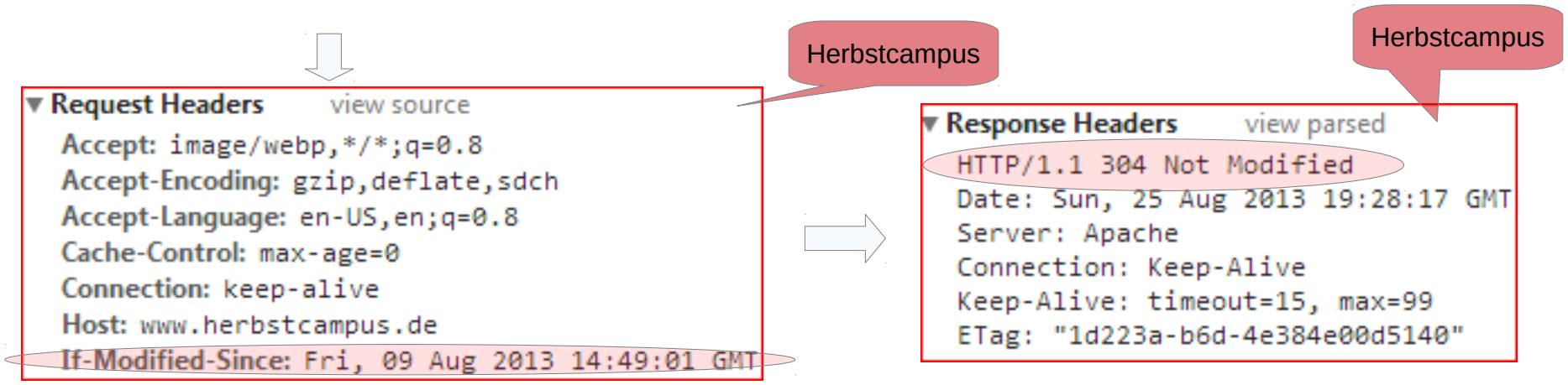


Amazon

How to improve? → Improve Caching

Expires Header

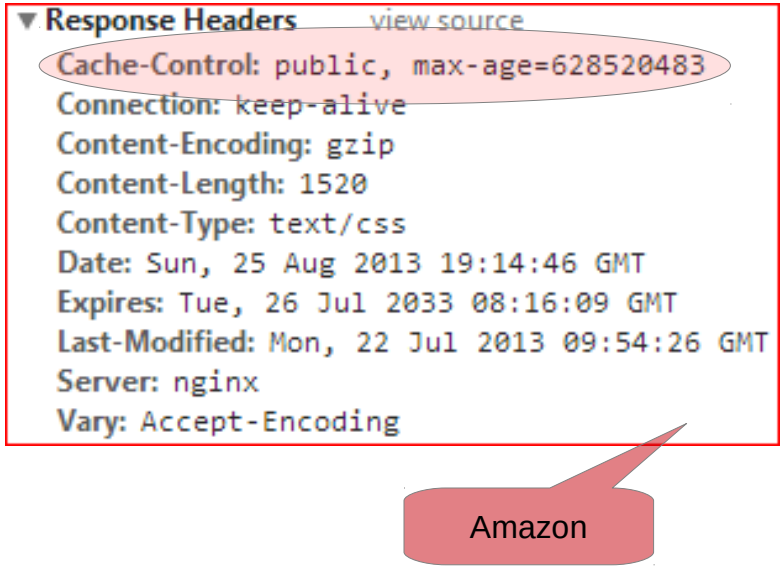
- What if Expires header is not set?
  - Component is stored in the browser's cache
  - **Conditional request** is required



How to improve? → Improve Caching

Cache Control Header (since HTTP 1.1)

- introduced to overcome limitations of Expires header
  - clock synchronization



How to improve? → Improve Caching

INode
The file's i-node number will be included in the calculation

MTime
The date and time the file was last modified will be included

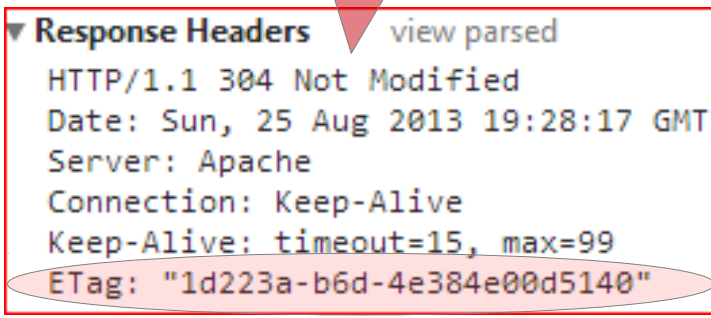
Size
The number of bytes in the file will be included

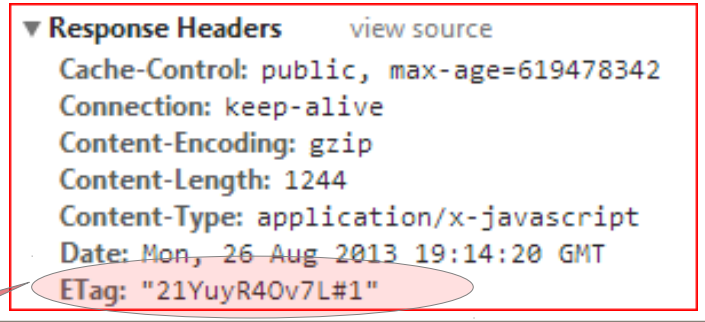
Source: http://httpd.apache.org/docs/2.2/mod/core.html

Configure/Remove ETags (Entity Tags)

- Uniquely identifies a **specific version** of a resource
- Apache 2.x ETag format
  - **<inode-timestamp-size>**
  - Should be changed for clustered environments
- Problem
  - Inode might be different for 2 servers
  - http://www.apacheweek.com/issues/02-01-18
- Example
  - 10 servers in our cluster
  - Probability 1/10 = **10%** that user will get a 304 Code
  - **90%** → wasteful 200 Code

Herbstcampus





Amazon

How to improve? → Gzip Components

Enable Gzip Compression

- Simple rule: **less data to transmit = transfer time decreases**

- Easiest of all techniques & has biggest impact

YouTube

Request Headers view parsed

```
GET / HTTP/1.1
Host: www.youtube.com
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.3
Referer: http://www.youtube.com/
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
```

Response Headers view source

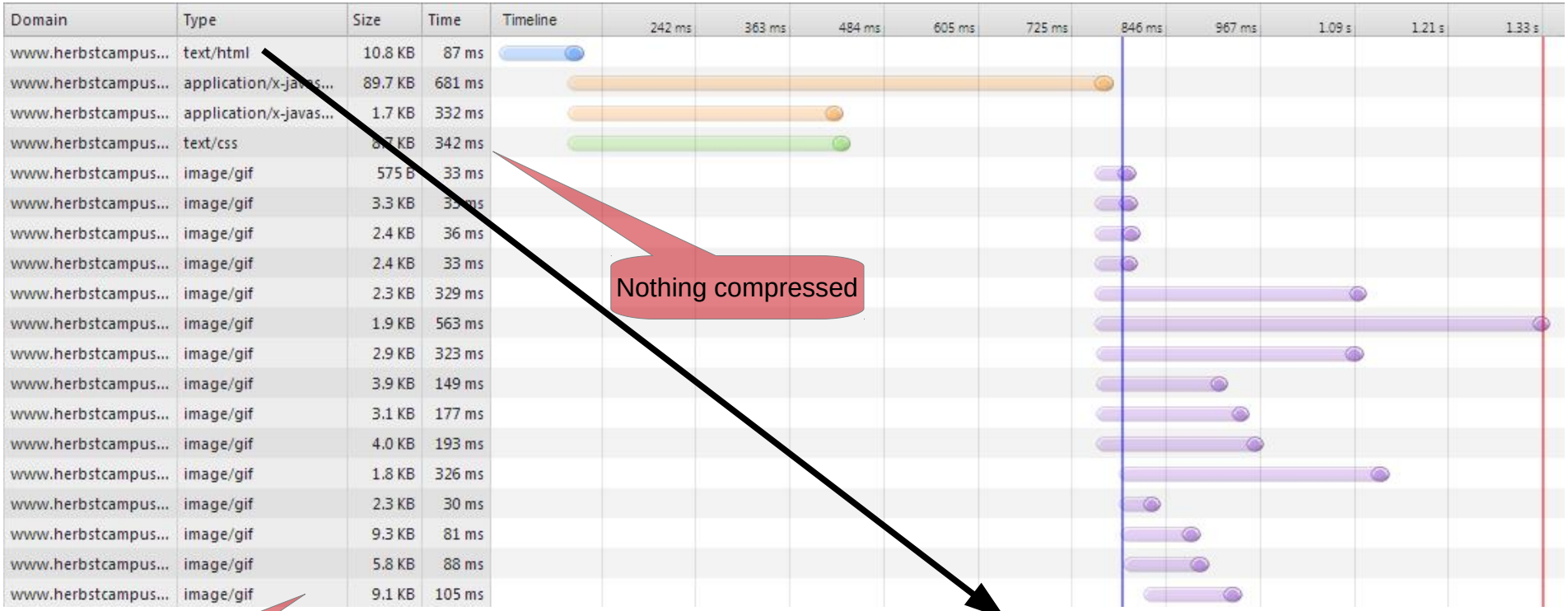
```
Alternate-Protocol: 80:quic
Cache-Control: no-cache
Content-Encoding: gzip
Content-Length: 30748
Content-Type: text/html; charset=utf-8
Date: Mon, 26 Aug 2013 19:02:19 GMT
Expires: Tue, 27 Apr 1971 19:44:06 EST
```

- What to compress?
  - Any text response (HTML, Scripts, CSS, XML, JSON)
  - Not necessary to compress images, PDFs (see http://goo.gl/7WYx1l)

- How? → Apache mod_deflate

```
AddOutputFilterByType DEFLATE text/html text/plain text/xml
```

How to improve? → Gzip Components



Nothing compressed

Main Herbstcampus Page

Nothing compressed

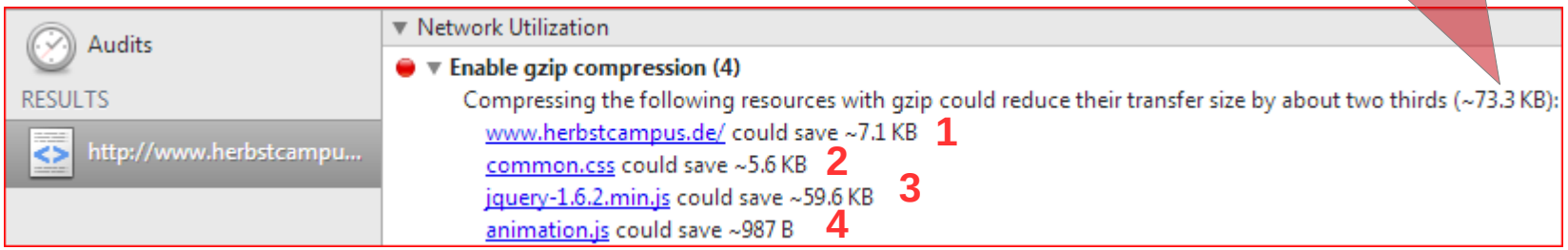
Response Headers view parsed

```
HTTP/1.1 200 OK
Date: Wed, 28 Aug 2013 19:33:52 GMT
Server: Apache
Last-Modified: Mon, 26 Aug 2013 15:28:54 GMT
ETag: "1d229d-2ac6-4e4db69fcf980"
Accept-Ranges: bytes
Content-Length: 10950
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
```

How to improve? → Gzip Components

Enable Gzip Compression

Total Possible Savings: ~73.3 KB

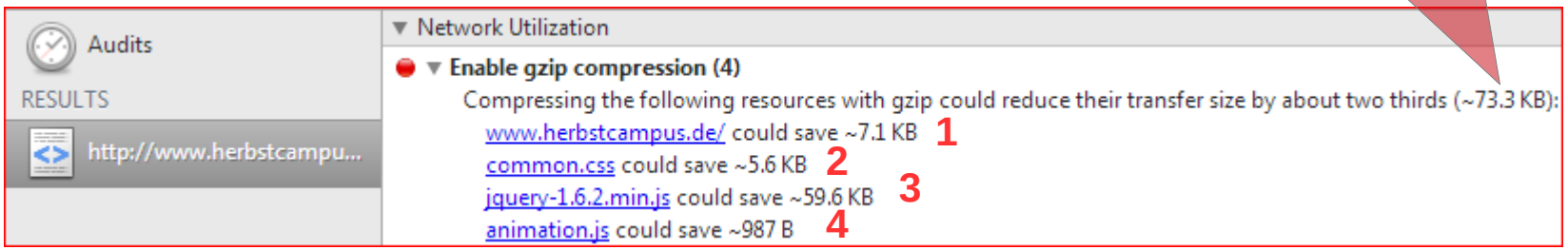


Name	Type	Size	Time
www.herbstcampus.de 1	text/html	11.0 KB	2.91 s
common.css 2	text/css	8.7 KB	75 ms
animation.js 4	application/x-javascript	1.7 KB	57 ms
jquery-1.6.2.min.js 3	application/x-javascript	89.7 KB	356 ms

How to improve? → Gzip Components

Enable Gzip Compression

Total Possible Savings: ~73.3 KB



Name	Type	Size	Time
www.herbstcampus.de 1	text/html	11.0 KB	2.91 s
common.css 2	text/css	8.7 KB	75 ms
animation.js 4	application/x-javascript	1.7 KB	57 ms
jquery-1.6.2.min.js 3	application/x-javascript	89.7 KB	356 ms

File	Size	Sav. in KB	Sav. in %
HTML (1)	11.0 KB	7.1 KB	**64.5 %**
CSS (2)	8.7 KB	5.6 KB	**64.4 %**
JS (3)	89.7 KB	59.6 KB	**66.4 %**
JS (4)	1.7 KB	987 B	**58.1 %**
	111.1 KB	**73.3 KB**	**~65 %**

How to improve? → Minification

Minify JS / CSS

- Simple rule: **less data to transmit = transfer time decreases**

- Minification = process of removing unnecessary characters

How to improve? → Minification

Minify JS / CSS

- Simple rule: **less data to transmit = transfer time decreases**

- Minification = process of removing unnecessary characters

Compressed Code

```
(function(a){a.fn.slider=function(d,j){var l=d*2;var f=0;var i=0;var g=null;var b=null;var e=false;var
h=false;function c(){if(e||f==g){return}e=true;b.children().eq(0).animate({top:"-="+i},d,function()
{e=false;f=f+1})}function k(){e=true;b.children().eq(0).animate({top:"0"},d,function(){e=false;f=0})}return
this.each(function(){b=a(this);i=b.children().eq(0).children().eq(0).height();b.css("height",
(i*j));g=b.children().eq(0).children().size()-j;setInterval(function(){if(!h){if(f==g)
{k()}else{c()}}},l);b.mouseenter(function(){h=true});b.mouseleave(function(){h=false})})})})
(jQuery);$(document).ready(function(){$("#slider").slider(500,3)});
```

Herbstcampus JS File

before	1547
after compression	647
compression ratio	58%
after compression and gzip	337
compression and gzip ratio	78%

Savings: 58%

Code minified with YUI Compressor

How to improve? → Minification

Minify JS / CSS

- Simple rule: **less data to transmit = transfer time decreases**

Herbstcampus CSS File

Compressed Code

```
html{height:100%;max-
height:100%;padding:0;margin:0;border:0;background:#fff;/*\*/overflow:hidden;/**/}body{height:100%;ma
x-
height:100%;overflow:hidden;padding:0;margin:0;border:0;background:#e1dfd9}#head{position:absolute;
margin:0;top:0;left:0;display:block;width:100%;height:181px;z-index:5;text-align:center;border-
bottom:1px solid #555;background:white}#head_brown{background-
color:#685c53;position:relative;margin:0;top:0;left:0;display:block;width:100%;height:60px}#foot{positio
n:absolute;margin:0;bottom:0;left:0;display:block;width:100%;height:60px;z-index:5;background-
color:#685c53}#breadcrumb{padding-top:10px;padding-left:20px;left:0;top:10px;width:90%;font-
size:1em;background:#e1dfd9;overflow:hidden;height:80px}#menu,#container,#container_wide,#sponso
rs{position:absolute;left:0;top:181px;bottom:60px;width:25%;font-size:1em;z-
index:4;background:#e1dfd9;overflow:hidden}* html #menu,* html #container,* html #container_wide,*
html #sponsors{height:100%;top:0;bottom:0;border-top:181px solid #fff;border-bottom:60px solid
#fff}#menu{left:0;background:#d7d2cb;border-right:1px solid
#555}#container{left:26%;width:50%}#container_wide{left:26%;width:74%}#sponsors{left:77%;width:23
%}#content{margin:auto;padding-left:20px;padding-right:0;margin-
top:10px;left:0;top:100px;bottom:20px;width:90%;position:absolute;overflow:auto}* html
#content{top:0;border-top:80px solid #e1dfd9;border-bottom:20px solid #e1dfd9;height:100%;z-
index:-1}#divider{width:90%;border-bottom:1px solid #555;margin-left:20px;margin-top:0}* html
#divider{margin-top:-20px}*{color:#444;font-family:Georgia,"Times New
```

before	9165
after compression	6841
compression ratio	25%
after compression and gzip	1474
compression and gzip ratio	84%

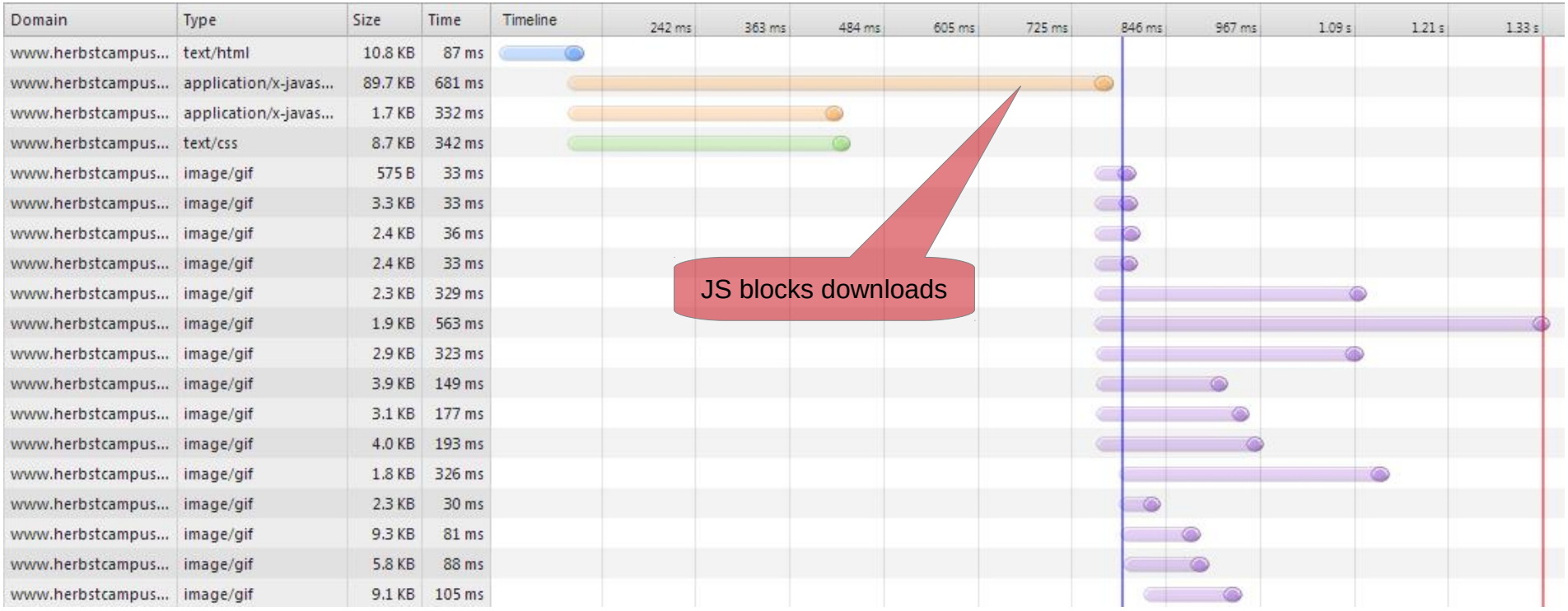
Savings: 25%

Code minified with YUI Compressor

How to improve? → Correct Placement of Files

Stylesheets at Top / JS at Bottom

- browser should start rendering as early as possible (user perceives a faster loading page)
- anything below the script is blocked from rendering and downloading until after the script is loaded (even when threads are available)→ **entire page is delayed**



How to improve? → Correct Placement of Files

Stylesheets at Top / Scripts at Bottom

Script at the TOP

Name	Method	Status	Type	Initiator	Size	Time	Timeline
js-top.php	GET	200	text/html	Other	9.1 KB	408 ms	
sleep.cgi?type=js&sleep=10&e...	GET	200	applicat...	js-top.php:3	579 B	10.23 s	
utils.js	GET	200	applicat...	js-top.php:3	2.4 KB	353 ms	
sleep.cgi?type=gif&sleep=2&e...	GET	200	image/gif	js-top.php:3	1.3 KB	2.24 s	
sleep.cgi?type=gif&sleep=2&e...	GET	200	image/gif	js-top.php:3	1.5 KB	2.44 s	
sleep.cgi?type=gif&sleep=2&e...	GET	200	image/gif	js-top.php:3	1.7 KB	2.41 s	
greyhound_a_32x32.jpg	GET	200	image/j...	js-top.php:3	1.3 KB	356 ms	
sleep.cgi?type=gif&sleep=2&e...	GET	200	image/gif	js-top.php:3	740 B	2.41 s	

Timeline: 4.36 s | 6.55 s | 8.73 s | 10.91 s | 13.09 s

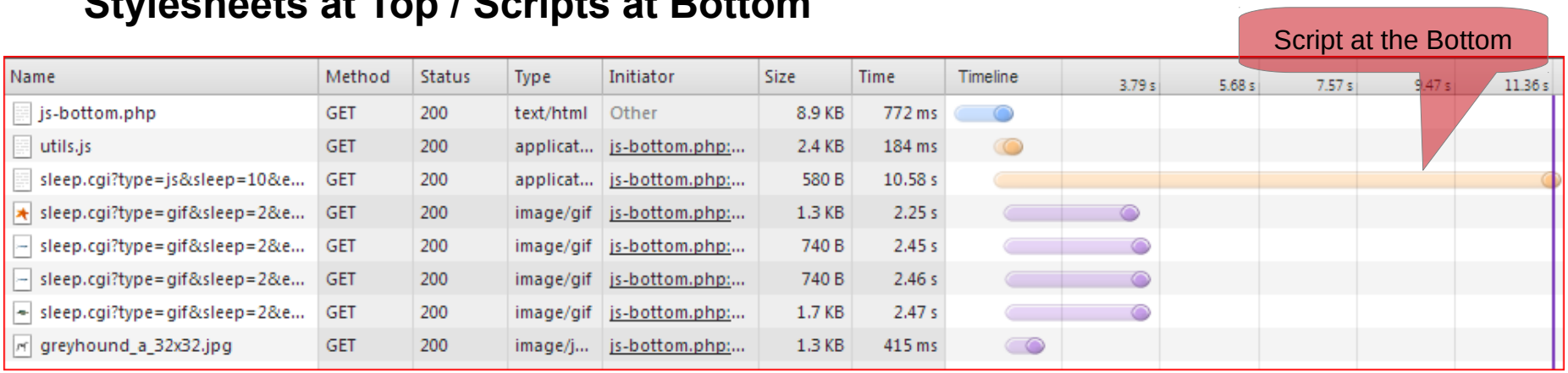
```
▼ <html>
  ▼ <head>
      <script src="/bin/sleep.cgi?type=js&sleep=10&expires=-1&last=0"></script>
      <title>Scripts at the Top</title>
    ▶ <style>…</style>
      <script src="utils.js"></script>
    ▶ <script>…</script>
    </head>
  ▼ <body>
    ▶ <div class="header">…</div>
    ▶ <div class="subheader">…</div>
    ▶ <div class="content">…</div>
      <!-- close content div -->
    ▶ <cite>…</cite>
    </body>
  </html>
```

CGI Script that sleeps for 10s

Example implemented at: http://stevesouders.com/examples/rule-js-bottom.php

How to improve? → Correct Placement of Files

Stylesheets at Top / Scripts at Bottom




```
▼ <html>
  ▼ <head>
      <title>Scripts at the Bottom</title>
    ▶ <style>…</style>
      <script src="utils.js"></script>
    ▶ <script>…</script>
    </head>
  ▼ <body>
    ▶ <div class="header">…</div>
    ▶ <div class="subheader">…</div>
    ▶ <div class="content">…</div>
      <!-- close content div -->
    ▶ <cite>…</cite>
      <script src="/bin/sleep.cgi?type=js&sleep=10&expires=-1&last=0"></script>
    </body>
  </html>
```

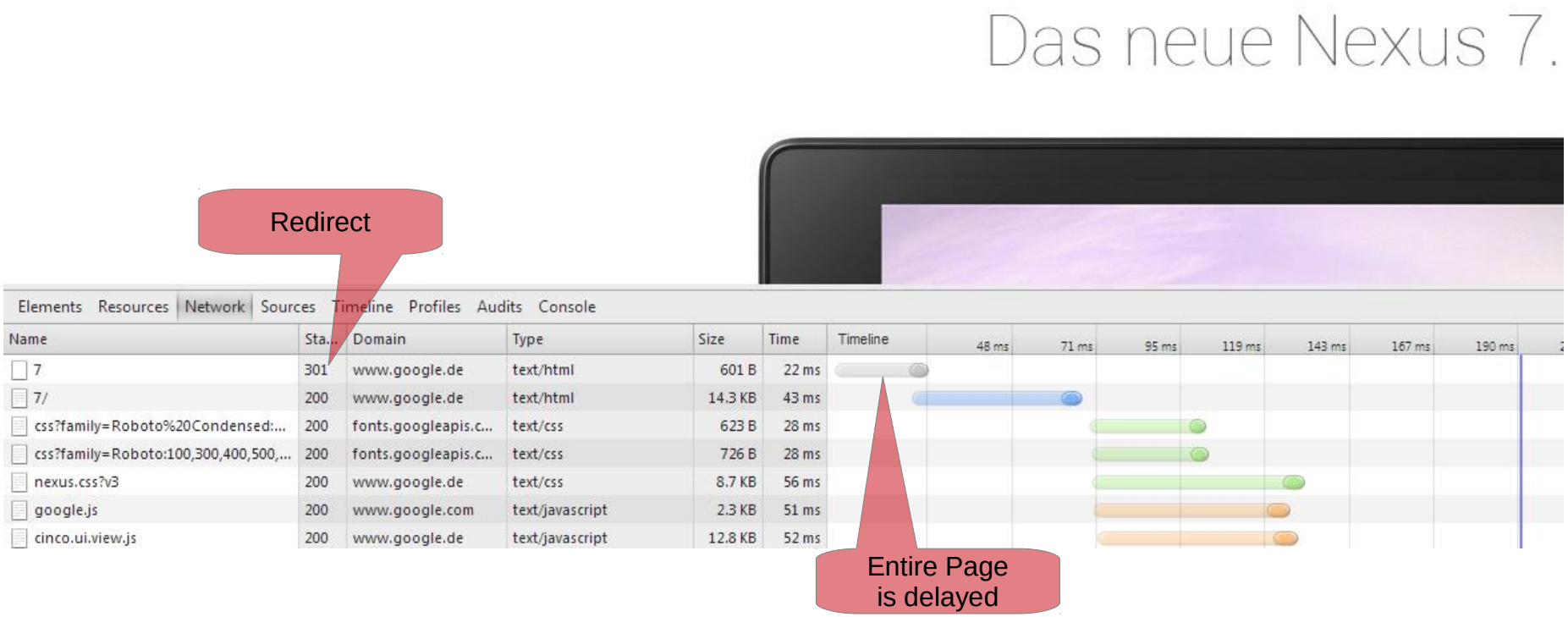
Example implemented at: http://stevesouders.com/examples/rule-js-bottom.php

How to improve? → Reduce Redirects

Reduce Redirects

3xx Redirection:
*"This class of status code indicates that **further** action needs to be taken by the **user agent** to fulfil the request."*

From Wikipedia

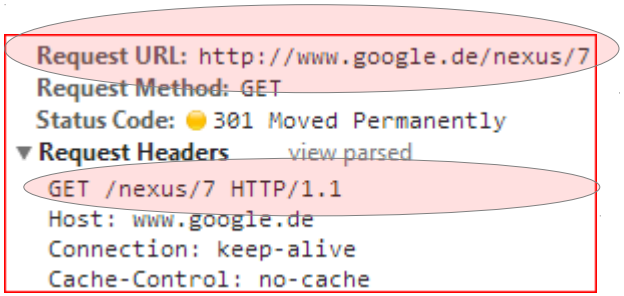


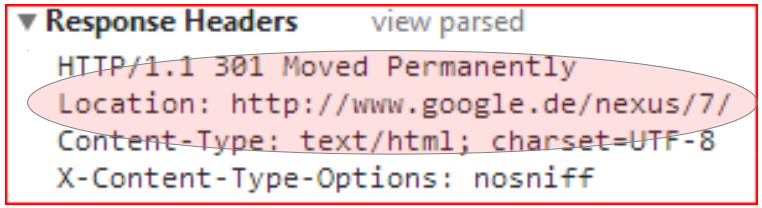
How to improve? → Avoid Redirects

Reduce Redirects

- Redirect blocks entire page loading (worse than putting Scripts at the TOP)

- Most wasteful redirect is the missing trailing '/'
  - www.google.de/nexus/7 → redirect to: www.google.de/nexus/7/





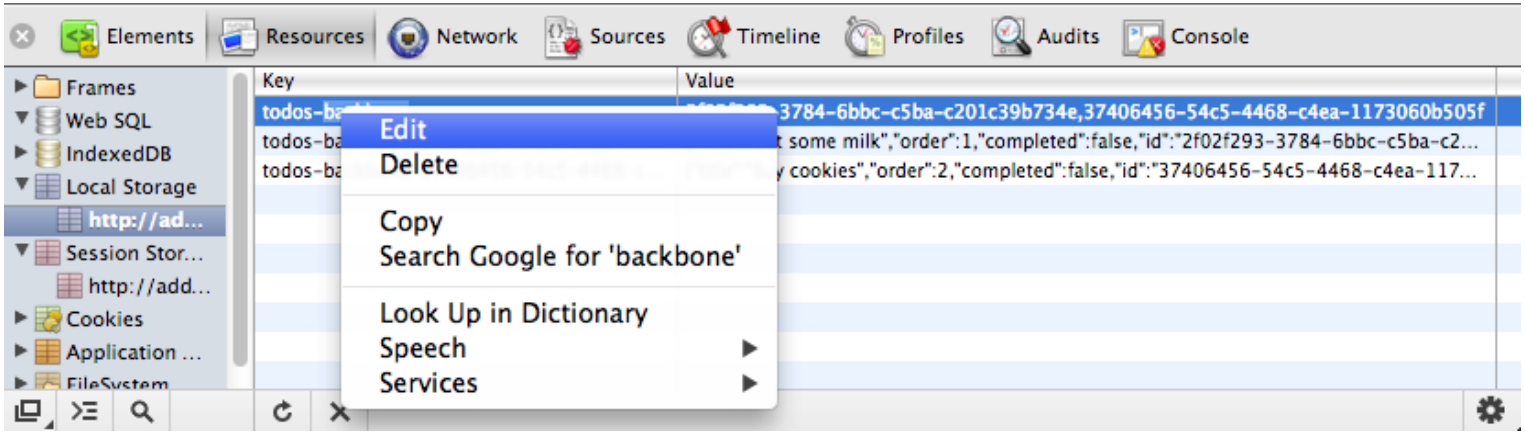
- Workaround?
  - Apache Alias → Alias /myurl /usr/local/apache/...
  - Apache mod_rewrite
  - Note: both do not solve the problem of finding URLs relative to the current directory

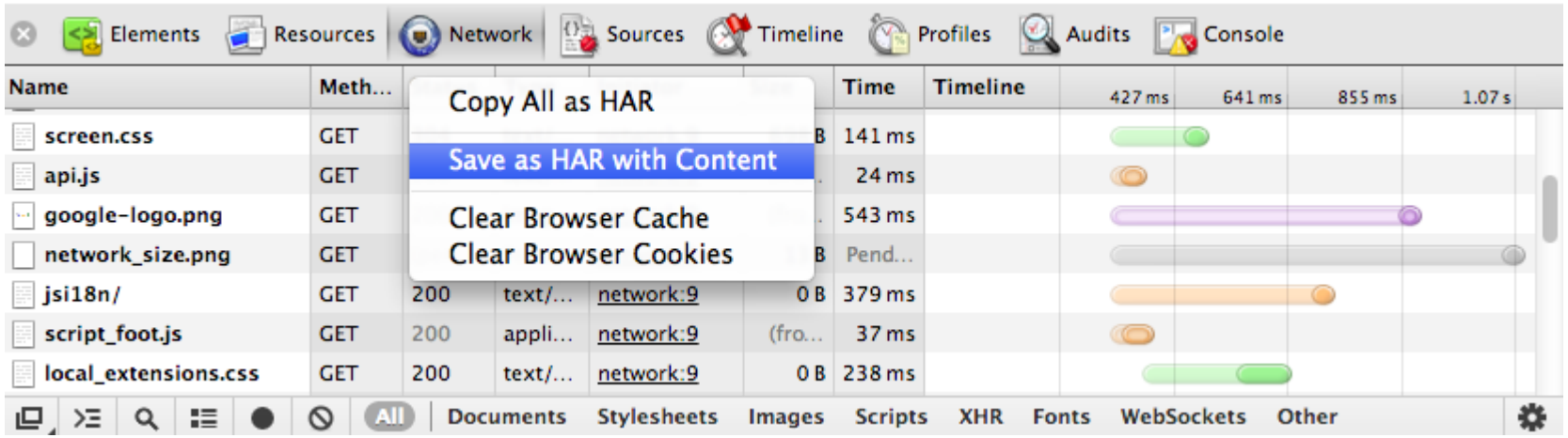
Tools (just a few)

Which Tools to Use?

Tools

- Chrome Developer Tools (https://developers.google.com/chrome-developer-tools/)



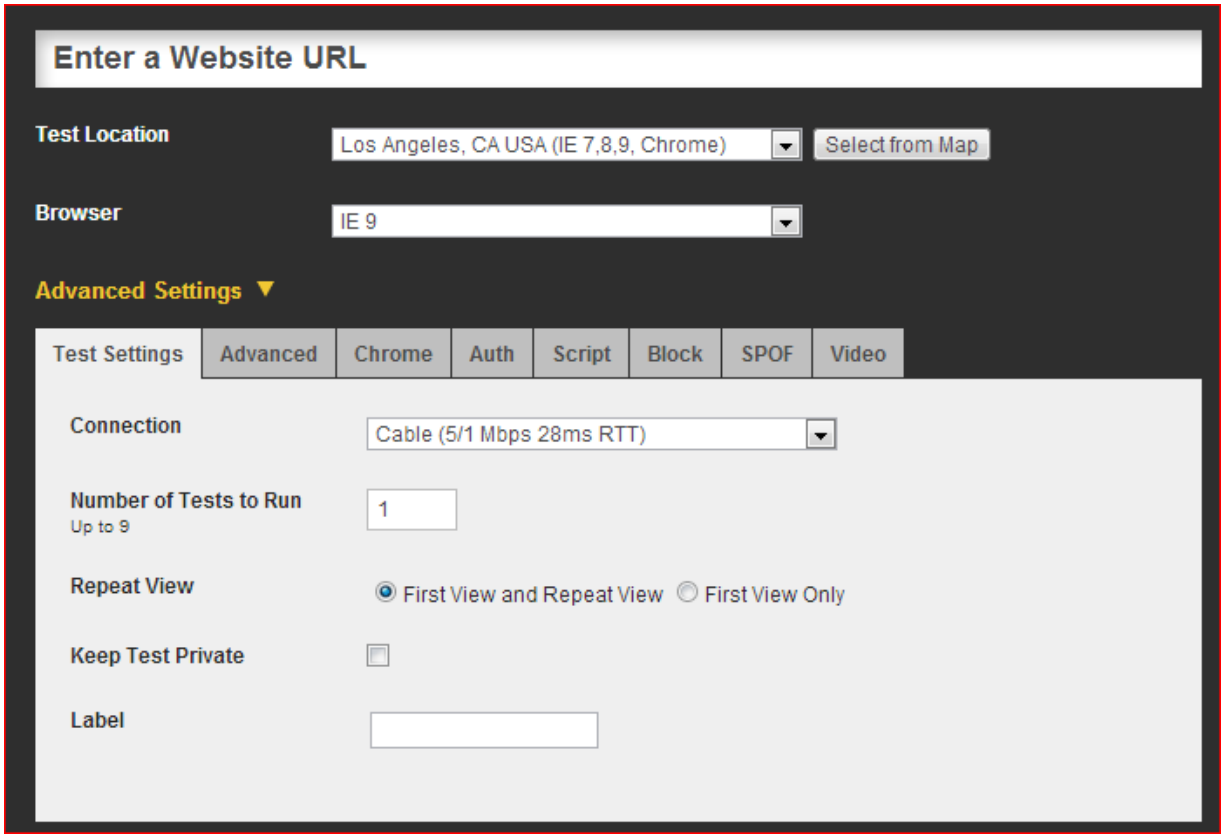


Source: https://developers.google.com/chrome-developer-tools/

Which Tools to Use?

Tools

- http://www.webpagetest.org/
  - Allows to simulate different RTTs / Browsers / Geographic Locations / Bandwidths



Which Tools to Use?

Tools

- JAWR (jawr.java.net)
  - Built-in minification
  - Enforced caching
  - Bundling of resources
  - CSS image sprite generation
  - Can be used with (JSF, Spring MVC, Wicket, Grails, …)

- mod_pagespeed (modpagespeed.com)
  - Apache module for rewriting web pages to reduce latency and bandwidth
  - Automatic website and asset optimization
  - 40+ configurable optimization filters

adesso business. people. technology.

Thank you for your Attention!

eduard.tudenhoefner@adesso.de
www.adesso.de

Noteworthy Literature

Literature & Sources

- High Performance Web Sites, Steve Souders
- Even Faster Web Sites, Steve Souders
- High Performance Browser Networking, Ilya Grigorik
- Google's „Make the Web Faster“, https://developers.google.com/speed/
- Web Performance Optimization, http://goo.gl/4xjs
- Improve the performance of your web applications, IBM developerWorks, http://goo.gl/UD5Ksj