Deep Dive into Browser Performance

Ilia Alshanetsky
@iliaa

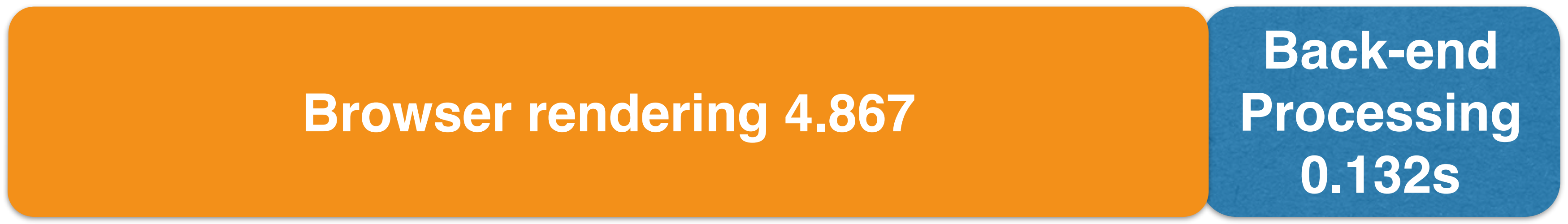
Me, Myself and I

PHP Core Developer

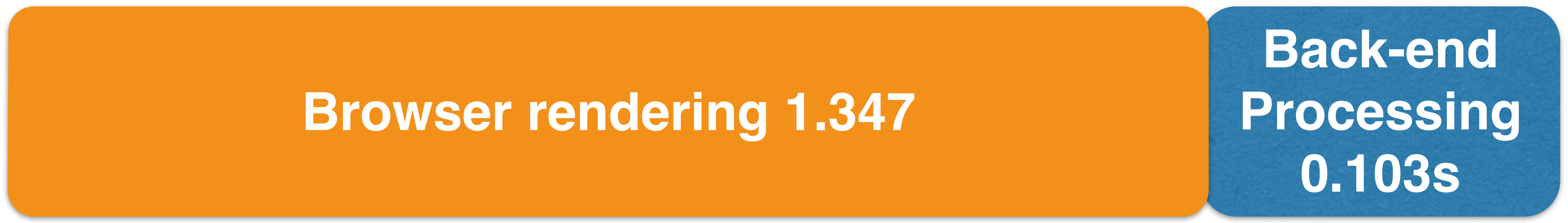
Author of
Guide to PHP Security

CIO at Centah Inc.

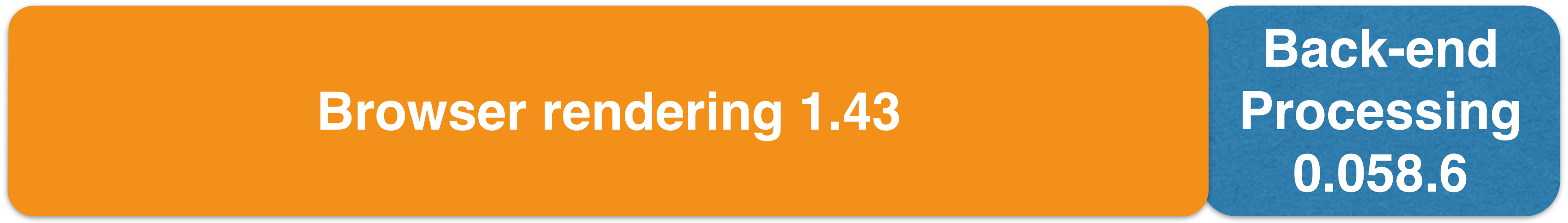
Why Browser Performance Matters?



ConFoo.ca - Total Page Load - 4.99s*



PHP.net - Total Page Load - 1.45s

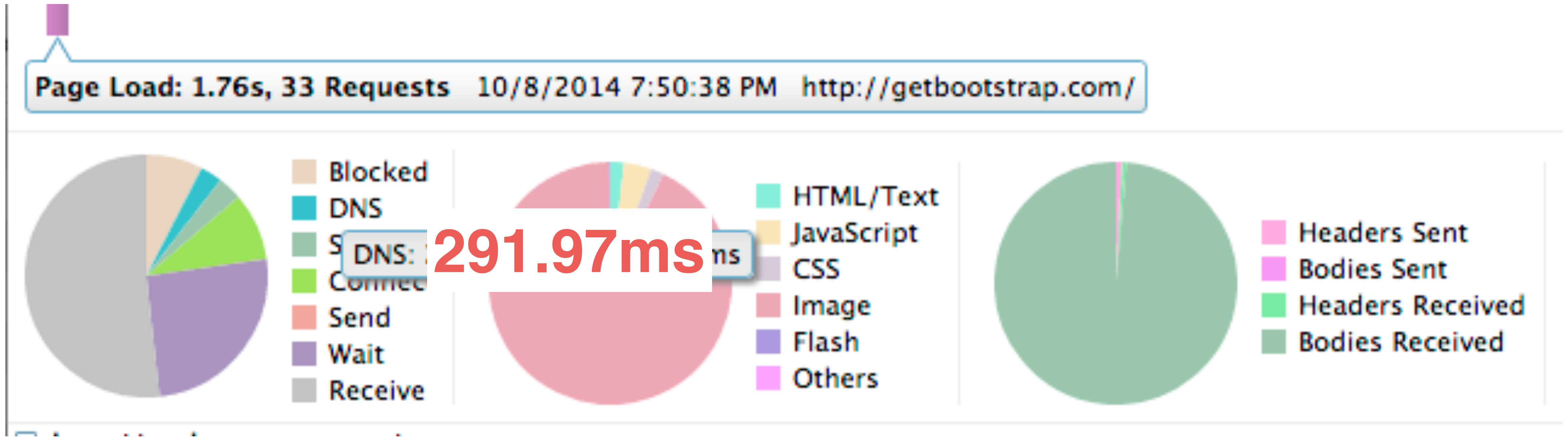


Github - Total Page Load - 1.43s

What Takes All This Time?

1. **DNS**
2. **HTTP + SSL Negotiation**
3. **JavaScript Processing**
4. **CSS Rendering**
5. **Image Processing**
6. **DOM Rendering**

DNS



DNS may take up-to 20% of 1st page load!

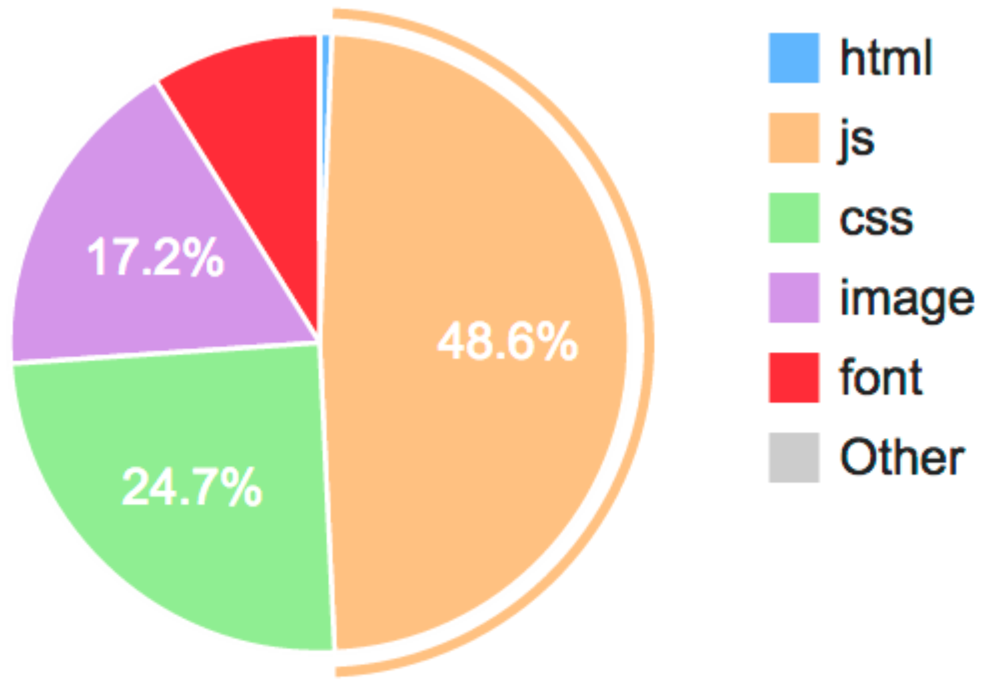
DNS Based Optimizations

1. **Use Embedded images via *data:image***
2. **Limit image requests via use of *sprites***
3. **Defer loading of external resources**
4. **Avoid multi-domain CDNs**
5. **Single-page applications for the win!**

Profile Page Loading

- Use Your Browser
  - **Developer Tools or Equivalent**
- Do Remote Tests
  - **http://www.webpagetest.org/**
  - **https://developers.google.com/speed/pagespeed/**
  - **https://www.modern.ie/en-us**
- Actual User Profiling
  - **http://www.lognormal.com/boomerang/doc/**
  - **Use Web-Timing API directly**

Compression For The Win!



1,394 KB
59 requests,
4.63 seconds to load

Compression Reduces size by >50% and makes page loads in 2.1 seconds!

Use gzip compression

965.8 KB total in compressible text, savings = 695.2 KB

Compress Images

171.6 KB total in images,savings = 51.8 KB

Confoo.ca via http://www.webpagetest.org

			Document Complete			Fully Loaded		
	Load Time	First Byte	Time	Requests	Bytes In	Time	Requests	Bytes In
First View	5.086s	0.390s	5.086s	62	1,082 KB	5.241s	63	1,093 KB
Repeat View	3.294s	0.274s	3.294s	6	36 KB	3.294s	6	117 KB

Cache, Cache, Cache

Set **max-age** or **expires** headers

Value should be at least **30 days**

To prevent stale content, **use unique file names** on new deployments for changed files.

Your goal is that 2nd page load only asks the server for the dynamic content!

Unique Filename Solutions

Nginx Re-write Trick

```
rewrite ^/static/[a-z0-9]+/(.*)$ /static/$1 break;
```

Version Based IDs

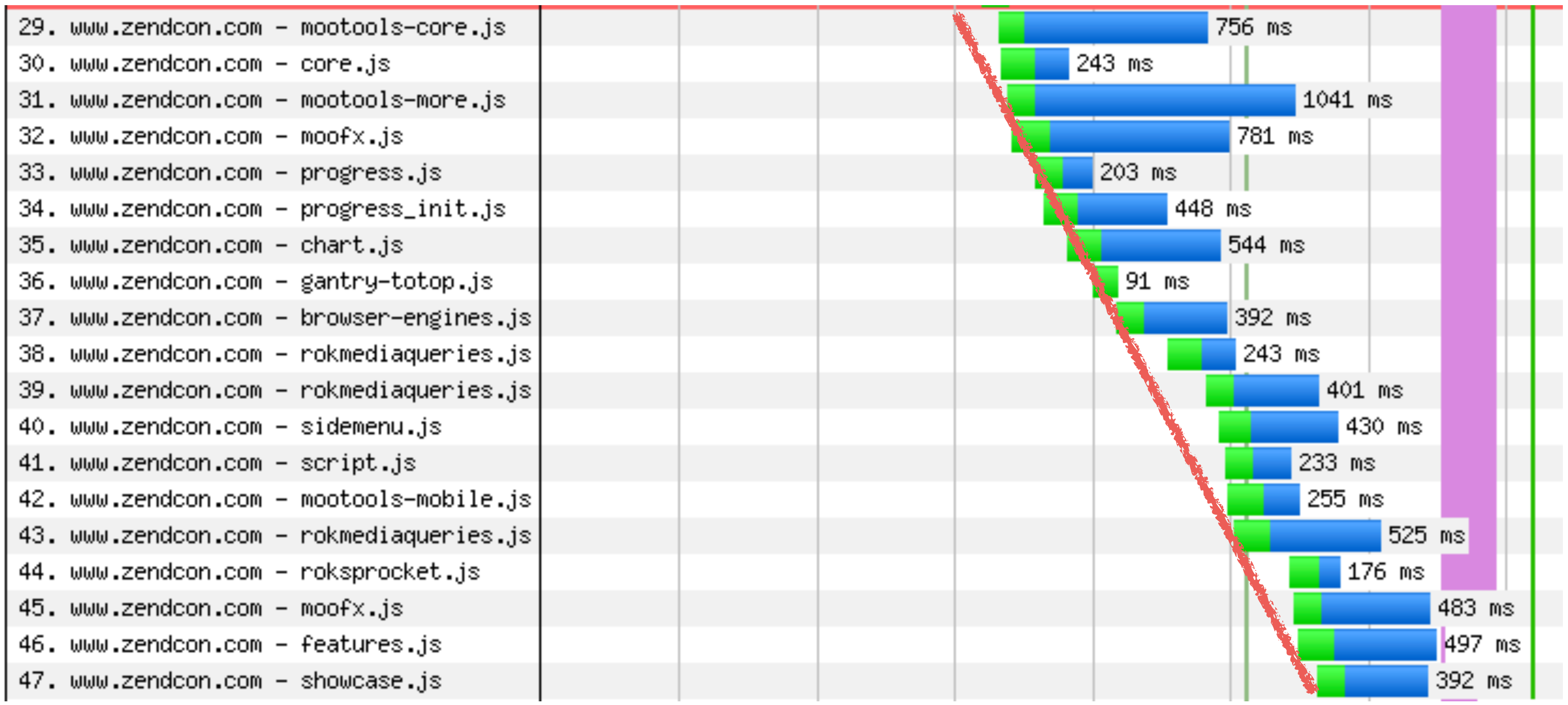
```
<?php $version_id = dechex(crc32(VERSION_STRING)); ?>

<script type="text/javascript"
  src="//static/<?=$version_id; ?>/my.js"></script>
```

Checksum Based IDs

```php
if (!($statics = my::cache("statics"))) {
    $statics = array(
        "js" => array(
            md5_file("/web/static/file.js") => "/static/file.js"
        ),
        "css" => array(
            md5_file("/web/static/file.css") => "/static/file.css"
        )
    );
    my::cache("statics", $statics);
}
```

JavaScript



29. www.zendcon.com - mootools-core.js	756 ms
30. www.zendcon.com - core.js	243 ms
31. www.zendcon.com - mootools-more.js	1041 ms
32. www.zendcon.com - moofx.js	781 ms
33. www.zendcon.com - progress.js	203 ms
34. www.zendcon.com - progress_init.js	448 ms
35. www.zendcon.com - chart.js	544 ms
36. www.zendcon.com - gantry-totop.js	91 ms
37. www.zendcon.com - browser-engines.js	392 ms
38. www.zendcon.com - rokmediaqueries.js	243 ms
39. www.zendcon.com - rokmediaqueries.js	401 ms
40. www.zendcon.com - sidemenu.js	430 ms
41. www.zendcon.com - script.js	233 ms
42. www.zendcon.com - mootools-mobile.js	255 ms
43. www.zendcon.com - rokmediaqueries.js	525 ms
44. www.zendcon.com - roksprocket.js	176 ms
45. www.zendcon.com - moofx.js	483 ms
46. www.zendcon.com - features.js	497 ms
47. www.zendcon.com - showcase.js	392 ms

JavaScript is loaded synchronously, so compact your files into a single compressed file!

JavaScript

Combination & minifying of JS files is best achieved with:

- Closure Compiler
  - http://goo.gl/8MVOIJ
- YUI Compressor
  - http://refresh-sf.com/yui/
  - http://yui.github.io/yuicompressor/
- PHP Based
  - https://github.com/tedious/JShrink

JavaScript

Don’t over-do combining of JS Files!

- Unnecessary data loading
- Decompression Overhead
- Extra JS Compilation

Micro-Case Study: SlashDot.org

One “BIG” JavaScript file

- 71kb compressed, 251kb actual size
- 199ms to receive
- 37ms to process

21.3% of total page load, 16% of total page size

- **< 10% of loaded JS code is executed**

JavaScript

Only load up-front what you absolutely need

Defer loading of everything else via RequireJS

```
<head>
   <script src="scripts/require.js"></script>
</head>
```

```
require.config({
    baseUrl: 'js/lib',
    paths: { jquery: ‘jquery-1.11.1' }
});
```

```
define(['lib/jquery'], function ($) {...});
```

http://requirejs.org/

If you can’t win, cheat!

```
$(document).ready(function() {
  setTimeout(function() {
   $.get( “your-file.js” );
  }, 2000);
};
```

General JS Tips

1. **Avoid Xpath, reference/search by ID**
2. **Setup events pre-load as opposed to post-load**

```
onkeyup="js_function()" vs $("input").each(function() {});
```

3. **For Grids only load the data to be displayed**
4. **innerHTML is not always faster than DOM**

http://jsperf.com/dom-vs-innerhtml/37

General JS Tips

- Most browsers leak memory with JS, avoid the common culprits:
    - ✦ Avoid passing objects (can result in circular references)
    - ✦ Avoid global variables
    - ✦ Use closures

General JS Tips

Help browser to make use of multiple CPUs by using iFrames to embed complex components such as grids.

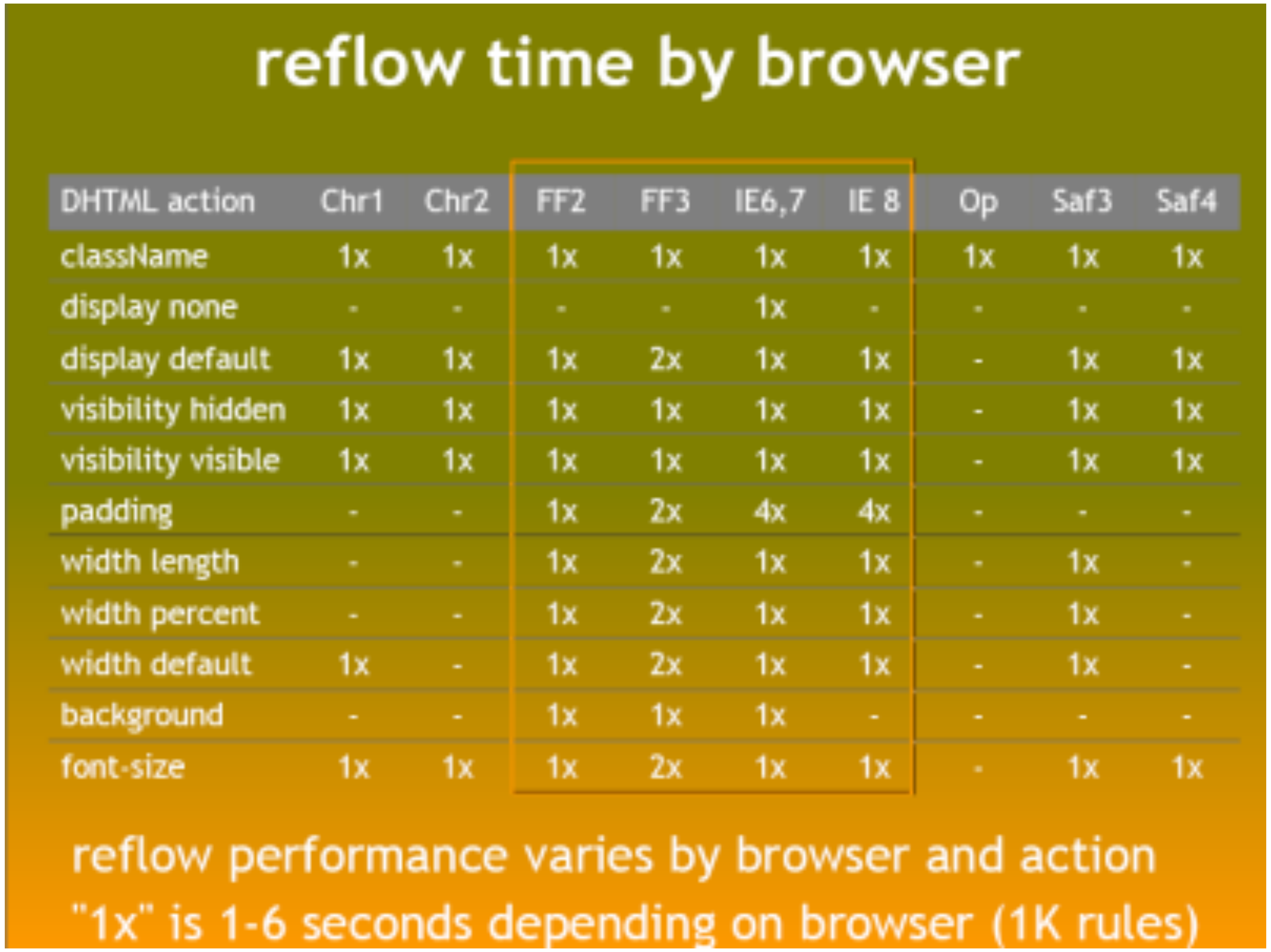
YO DAWG WE HEARD YOU LIKE IFRAMES
SO WE PUT AN IFRAME IN YOUR IFRAME SO YOU CAN LOAD PAGES WHILE YOU LOAD PAGES WHILE YOU LOAD PAGES
quickmeme.com

CSS

- Minimize
- Combine
- Compress

- Don’t fear <style> (inlined) CSS

Avoid Repaints & Reflows



reflow time by browser

DHTML action	Chr1	Chr2	FF2	FF3	IE6,7	IE 8	Op	Saf3	Saf4
className	1x	1x	1x	1x	1x	1x	1x	1x	1x
display none	-	-	-	-	1x	-	-	-	-
display default	1x	1x	1x	2x	1x	1x	-	1x	1x
visibility hidden	1x	1x	1x	1x	1x	1x	-	1x	1x
visibility visible	1x	1x	1x	1x	1x	1x	-	1x	1x
padding	-	-	1x	2x	4x	4x	-	-	-
width length	-	-	1x	2x	1x	1x	-	1x	-
width percent	-	-	1x	2x	1x	1x	-	1x	-
width default	1x	-	1x	2x	1x	1x	-	1x	-
background	-	-	1x	1x	1x	-	-	-	-
font-size	1x	1x	1x	2x	1x	1x	-	1x	1x

reflow performance varies by browser and action
"1x" is 1-6 seconds depending on browser (1K rules)

https://developers.google.com/speed/articles/reflow

- Changes to DOM nodes
- Hiding DOM nodes
- Actions that extend the page (causes scroll)
- Changes to colour, background and outline properties

Merge Style Changes

```
// slowest
el.style.left = "10px";
el.style.top  = "10px";

// getting better
el.className += " top-left-class";

// best
el.style.cssText += "; left: 10px; top: 10px;";
```

Peekaboo Trick

```
var me = $("#el");
me.hide();

// make various changes to DOM/Content

me.show();
```

Dolly Trick

```
var $dolly = el.clone();

// make changes to the copy

el.replaceWith($dolly);
```

Don’t Abuse Computed Styles

```
// nein, nein, nein!!!!
for (var i = 0; i < 100; i++) {
    el[i].style.left = el.offsetLeft + "10px";
    el[i].style.top  = el.offsetTop  + "10px";
}

// Wunderbar
for (
var left = el.offsetLeft, top = el.offsetTop, i = 0;
i < 100;
i++, top+=10, left+=10) {
  el[i].style.cssText += "; left: " + left +
                            "px; top: " + top + "px;";
}
```

Good Reference Points

http://www.phpied.com/rendering-repaint-reflowrelayout-restyle/

http://www-archive.mozilla.org/newlayout/doc/reflow.html

https://developers.google.com/speed/articles/reflow

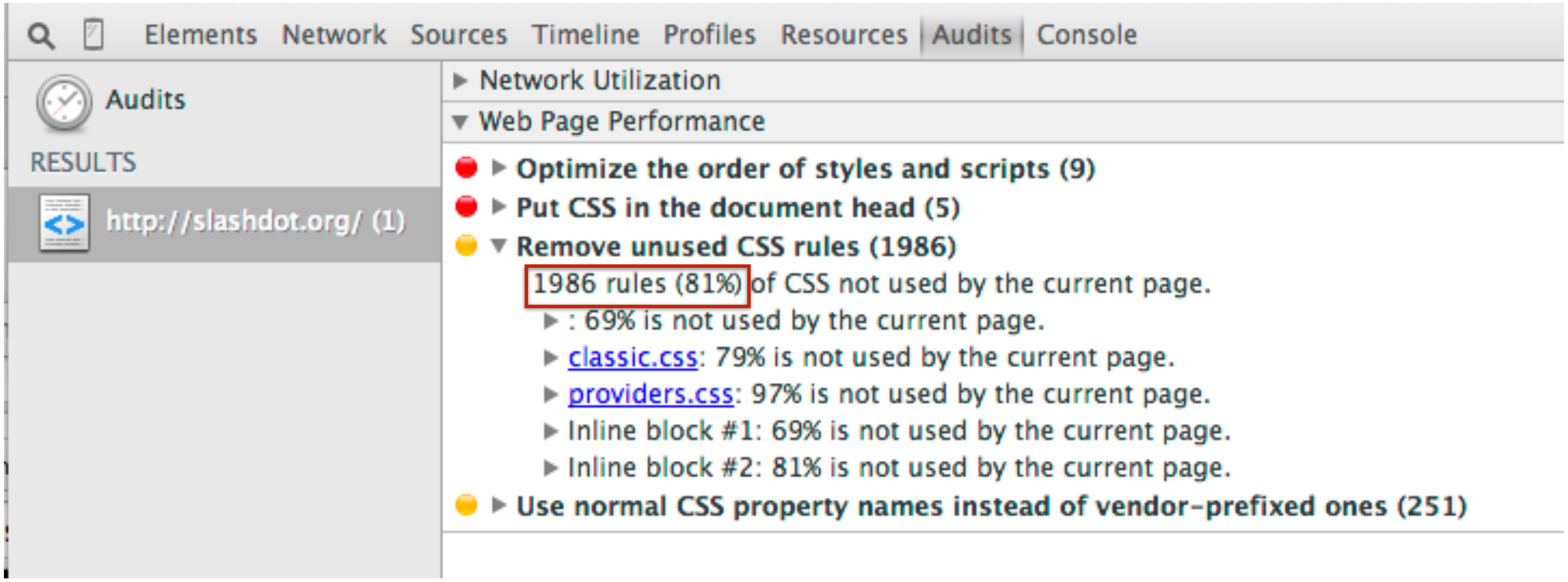
More CSSery

- **Reference by element ID**
- **Be specific, but avoid child selectors**
- **Avoid @import()**
- **Avoid multi-class css rule (.foo.bar.baz)**

More CSSery

- Pseudo selectors are slow
- Name space attribute selectors (type="…" vs input[type="…"])
- Eliminate un-used rules
- Avoid browser specific extensions (-webkit, -opera, -moz, etc…)

Micro-Case Study: SlashDot.org



▶ Network Utilization
▼ Web Page Performance

● ▶ Optimize the order of styles and scripts (9)
● ▶ Put CSS in the document head (5)
● ▼ Remove unused CSS rules (1986)

1986 rules (81%) of CSS not used by the current page.

▶ : 69% is not used by the current page.
▶ classic.css: 79% is not used by the current page.
▶ providers.css: 97% is not used by the current page.
▶ Inline block #1: 69% is not used by the current page.
▶ Inline block #2: 81% is not used by the current page.

● ▶ Use normal CSS property names instead of vendor-prefixed ones (251)

1986 rules (81% unused)

CSS Tools

https://github.com/Cerdic/CSSTidy

PHP

http://devilo.us/

Web-based

Slides: http://ilia.ws

@iliaa

Please leave feedback @ https://joind.in/13255