Now I Know My HTML5

Next Time Won’t You Build with Me

Hands on Deep Dive into HTML5 Ads

Agenda

- Introduction
- Building your HTML5 Ad
- Why Web Fonts is a Good Idea
- Optimize and Package
- Performance in the WWW

iab.

Tools and Platforms

Tools and Platforms used in this presentation are for illustration purpose only

Tools shown here are selected by Speakers based on their preferences and comfort level to show live example of building an HTML5 Ad

IAB does not endorse any tool over another

Speakers

Kenji Baheux
Product Manager, Google

Cory Hudson
Senior Creative Director, AOL

John Percival
Head of Design Technology, Amazon

Vladimir Levantovsky
Sr. Technology Strategist, Monotype

Shailley Singh
Director, Mobile & Ad Products, IAB

Build HTML5 Ads

Tooling Options

Questions & confusion over appropriate tooling options

- **Adobe Flash Pro CC**
  - Familiarity
  - Small learning curve
  - Future facing technologies
  - Solid output, CDN hosted
  - Flexible workflow

- **CreateJS**
  - Robust documentation
  - Dedicated support
  - Extensive functionality

CREATEJS

Build HTML5 Ads

- **Emerging IAB Standards**

  - 200kb is the “new 40kb”
  - File size calculation
  - Asset packaging
  - CDN hosted libraries
  - Max 15 server requests (5-8 creative level)
  - 30% max CPU usage
  - Fallback ad experience
  - Pre-loader

iab.

INTERACTIVE ADVERTISING BUREAU – DISPLAY & MOBILE ADVERTISING CREATIVE FORMAT GUIDELINES

2015 ADVERTISING CREATIVE GUIDELINES FOR DISPLAY & MOBILE – UPDATED FOR HTML5

Build HTML5 Ads

- **SWF to HTML5 Conversions**

  - **Google Swiffy**
    Extension (Flash CS6) & web service
    FLA vs. SWF
    Output
    Suitable usage

  - **Flash Pro CC**
    If working from a source FLA
    Output
    Facilitates a true transition to HTML5 & JS

Process: https://blogs.adobe.com/flashpro/converting-your-flash-ads-to-html5-canvas/

Build HTML5 Ads

- **AS to JS Basics**

Whitepaper - HTML5 Banner Ads With CreateJS:
http://createjs.com/html5ads/

EaselJs:
http://www.createjs.com/docs/easeljs/modules/EaselJS.html

Build HTML5 Ads

- **HTML5 Ad Creation Checklist**

  - ✓ **Beginning a Project (AS3 & converting vs. Canvas Doc type)**
  - ✓ **Text Support**
  - ✓ **Retina / High DPI Support**
  - ✓ **Filters and Effects**
  - ✓ **Spritesheet creation & implementation**
  - ✓ **Image optimization**
  - ✓ **Publish Settings**
  - ✓ **Implement AdHelper.js**

Build HTML5 Ads

Text Support

Flash Pro currently only supports the usage of dynamic text fields
In order to maintain the visual integrity of your styled text you may want to take one of the following approaches:

- Start your banner ad project as a Flash ActionScript 3 document type (not an HTML5 Canvas FLA) and then break apart all static text before converting the file over to HTML5 Canvas.
- Maintain a separate AS3 FLA for the styling of all text and then paste the text into your working HTML5 Canvas document after it has been converted to vector outlines.
- Design all of your text in Photoshop or Flash Pro and then integrate the text as bitmaps. This approach is not recommended as it removes the benefits of vector based text which would be small file size and scalability.

Build HTML5 Ads

- **Text Support**

  - Create a movieclip of your broken apart text and set it to cache as bitmap. This will improve performance as the vector drawing calculations won't have to be repeatedly processed on every stage tick.
  - CacheAsBitmap for complex vectors
  - Be sure to maintain an editable version of your text by creating a "guided out" backup layer containing a static text field. This ensures that edits can be easily made in the future.
  - Web fonts
  - Make sure text is on even pixels!!!
  - Pre-release feature: “Publish Text as Outlines”

Build HTML5 Ads

Retina / High DPI Support

- In order to ensure that bitmaps are displayed crisply on high DPI screens, they should be created at double the display size and then scaled down accordingly within Flash Pro.
- Resize your stage to double size
- Resize appropriate image assets to double size
- Scale contents of symbols to double size, but be careful not to scale stage instances
- Publish HTML, and uncheck “Overwrite HTML” (very important publish useful setting!)

```
<canvas id="canvas" width="600" height="500" style="background-
color:#720E10; width: 300px; height: 250px;"></canvas>
```

- Test on actual devices
- AdHelper.js simplifies and optimizes this process for us

Build HTML5 Ads

Image Handling

- Need to double size images accordingly in order to support retina
- Need to compress with ImageOptim or imageAlpha
- Publish Setting: Overwrite Images
- Unused images are not published

- Should combine similar images into sprite sheets
  - There should generally be no need for more than 2 sprite sheets.

- Flash Pro's built-in features:
  - Generate Sprite Sheet. Available via the Flash Pro library panel and requires manual placement and masking for each displayed image.
  - Export All Bitmaps As Sprite Sheets. Available via the Flash Pro publish settings which automatically handles sprite sheet creation for you however currently outputs a single sprite sheet for all asset types.

Build HTML5 Ads

Using Filters and Color Effects

- Flash Pro CC supports most filters and color effects
- Shadow Effects
- Filter Effects
- Numerous effects can be converted to images and contained within a single sprite sheet
- Animated filters can be converted to image sequences if needed

Build HTML5 Ads

- **Animation Control**
  - **No more FPS**
  - **GreenSock Animation Library**
    Familiarity
    Robustness
    Performance
    CDN hosted
  - **Move transformation point to coincide with registration mark**
  - **AdHelper**
    Stopping at 15 secs (includes GSAP tweens)
    Monitoring and managing performance
    Time synched animations
  - **CreateJs Ticker, RAF, RAF_SYNCHED**

Build HTML5 Ads

- **Fall back Experience**

CreateJs and Canvas are supported in all modern browsers
The only unsupported browser with notable market share is IE8

```
<canvas>
        <script>
         if (!window.CanvasRenderingContext2D) {
         document.write("<a href='url'><img src='image.jpg' alt='text'></a>");
         }
        </script>
    </canvas>
```

The CreateJS AdHelper class simplifies setting up fallback content, and makes the code above unnecessary.

Build HTML5 Ads

- **Create JS Ad Helper**

  - An in-depth white paper on building HTML5 advertising with CreateJS and Flash Pro. Includes supporting materials, sample banner ad, and helper classes: https://github.com/createjs/html5ads/

  - Sleep / Wake
  - Visibility
  - Alternative / Fallback Content
  - Time Synch
  - High DPI Support
  - Performance Monitoring

Build HTML5 Ads

Designer/ Developer Workflow

- There are many valid workflows for developing HTML5 Canvas content using Flash Pro.
- You can do everything within Flash Pro for simpler projects.
- As projects grow in complexity, it is generally wise to create a separation between presentation and logic.
- Benefits: productivity, quality control, reuse and robustness.
- Can leverage more robust code editors
- Designer owns the FLA, Developer owns the code
- Make sure to set linkage IDs in your Flash library!

Build HTML5 Ads

- **Debugging and Testing**
  - Test on actual devices whenever possible
  - Console.log()
  - Flash Output Panel
  - Devtools (console, file size, CPU profiling, asset loading times, identifying bottlenecks, GPU usage, browser repaints, measure FPS, etc.)
  - PreLoadJs
  - AdHelper provides performance monitoring capabilities for you
  - HTML5 Canvas rendering performance utilities: https://blogs.adobe.com/flashpro/behind-the-scenes-of-html5-canvas-rendering-performance/

Build HTML5 Ads

Where to Go from Here

- HTML5 Banner Ads With CreateJS
  http://createjs.com/html5ads/
- GitHub Repo With AdHelper
  https://github.com/createjs/html5ads/
- CreateJs / EaselJs
  http://createjs.com/easeljs
- HTML5 For Digital Advertising
  http://www.iab.com/guidelines/html5-for-digital-advertising-1-0-guidance-for-ad-designers-creative-technologists/
- IAB Display Advertising Guidelines
- http://www.iab.com/guidelines/iab-display-advertising-guidelines/
- IAB HTML5 Wiki
  http://www.iab.net/wiki/index.php/HTML5_for_Digital_Advertising_Resources

Web Fonts

And Why Using Them Is a Good Idea

Web Fonts

Why Using them is a good idea

Text is meant to be readable

- on any screen, on any device, by anyone (human or a screen reader);
- devices and screen sizes vary greatly;
- screen resolutions vary greatly;
- images with pre-rendered text are only readable if you can see them!

Fonts determine the visual identity of text content

- size and layout;
- look and feel;
- in written communication - the font you chose is your voice!

The visual identity of text communicates

- your brand;
- advertized products or services;
- whether the ad can be trusted!

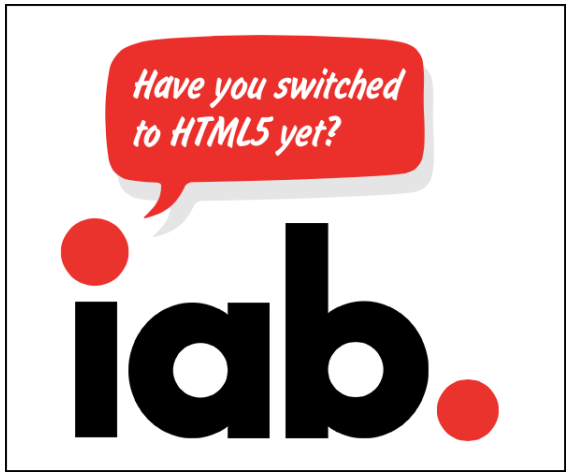
Web Fonts preserve visual identity!

Web Fonts

- **Which Ad Would You Trust and Identify**

Web Fonts

- **What's the catch** (warning! Tech talk below)
  - **Anything that used to happen automatically in Flash now needs to be a conscious effort!**
    - Nothing happens by itself in HTML5 - you need to make it happen
  - **Fonts are standalone resources that need to be**
    - Defined (both web fonts as primary choice and fallback fonts if something goes wrong)
    - Loaded (which takes time and bandwidth)
    - Minimized in size for optimal usage (e.g. subsetted to only include glyphs that are actually used)
  - **Fonts can be cached and shared**
    - Repeated ad impressions won't require repeat font loads - better user experience
    - Between multiple ad units - optimized ads delivery and creation workflow
  - **Fonts need to be licensed for a particular use**



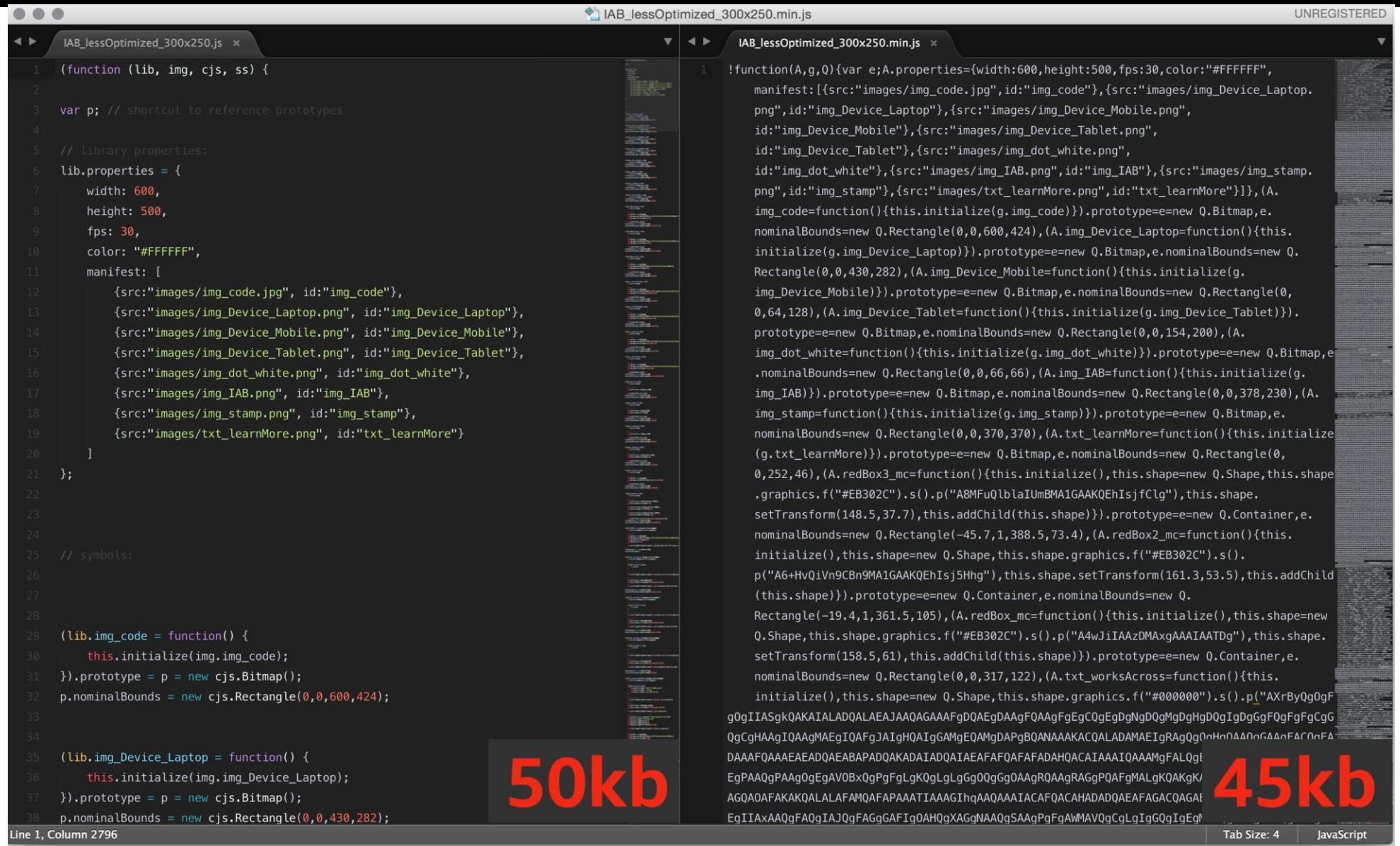
Optimize & Package

Ad Packaging and Tag Prep

Asset Optimization

- **Text-Based Files**
  - **Minify (HTML/CSS/JS/JSON)**
  - **Example: Flash Output of Ad Script.js - visual of the pre vs. post of minification, concatenation of text**
  - **Concatenate (Reduce HTTP requests)**
  - **CDN Hosting (Typically done by your ad server)**
  - **Tools to use (Gulp, Grunt, Online Tools)**

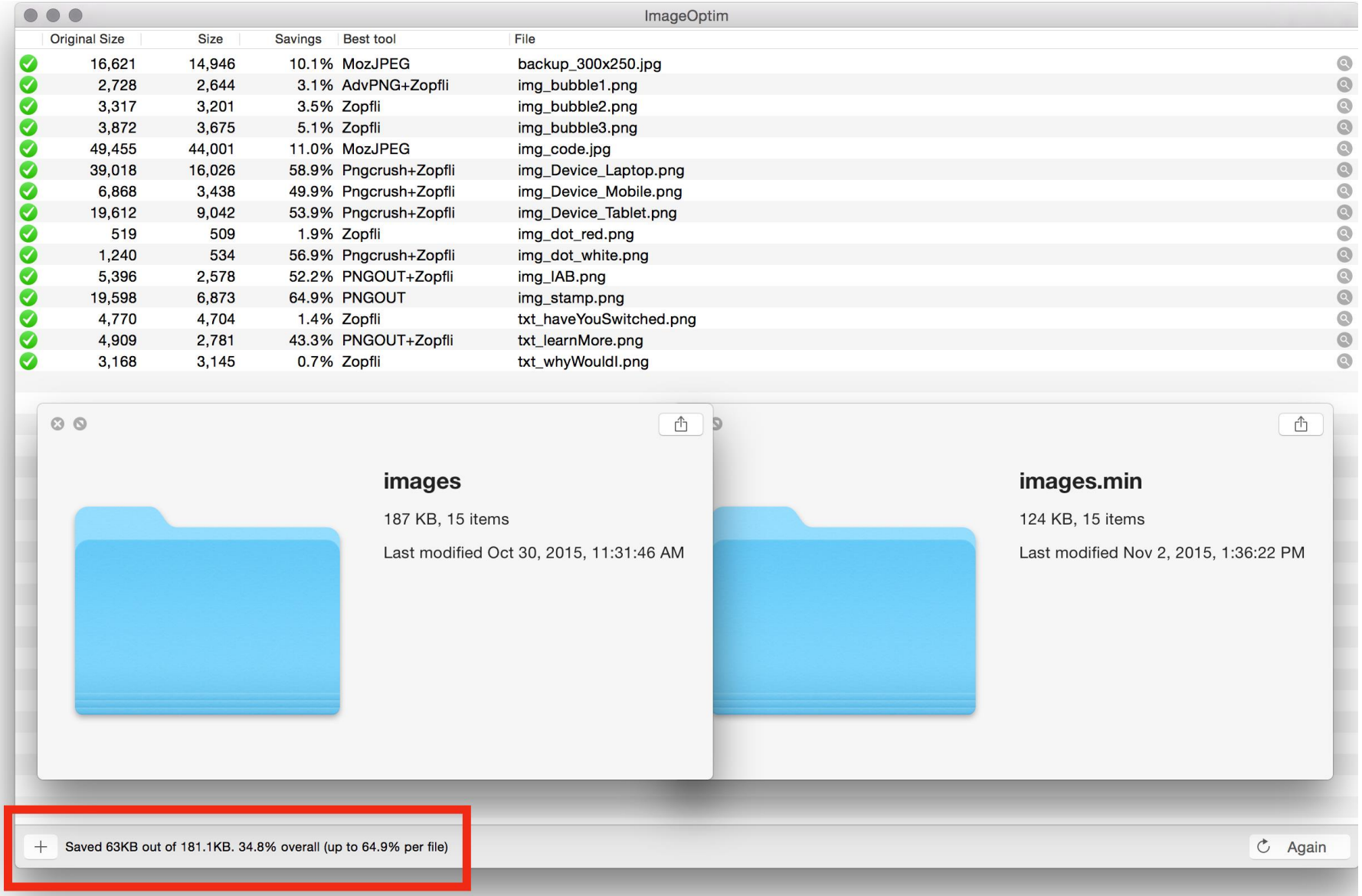
Asset Optimization



Asset Optimization

- **Images**
  - **Reference Cory’s sprite**
  - **Compress (ImageOptim) - show before and after**

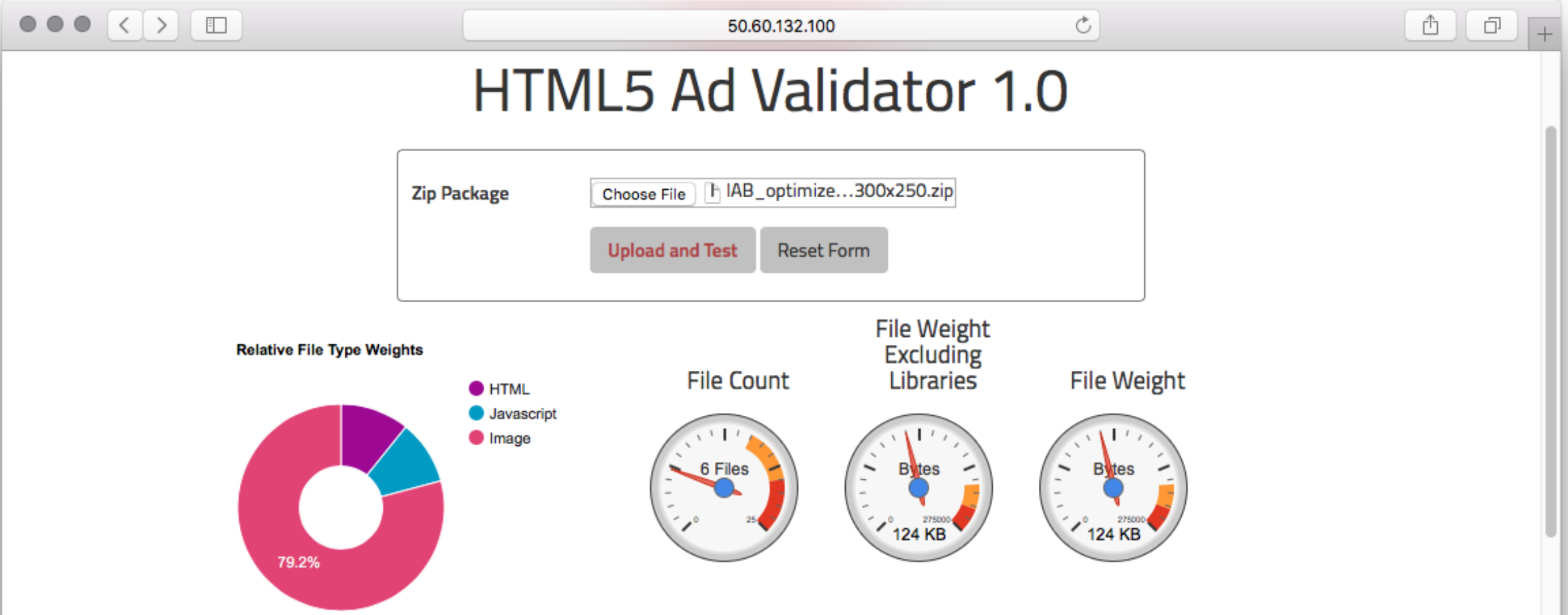
Asset Optimization



Ad Packaging

- **ClickTags**
  - Install tracking and install clickTags
  - window.open(window.clickTag, “_blank”, opts);
  - install clickTag based on the ad server macros (Reach out to your ad ops specialist/trafficker)

- **Zipping Up!**
  - Zip it up for ad server ingestion
  - Ad server tag creation for publisher/network trafficking



IAB HTML5 Ad Validator (http://html5.iabtechlab.com)

Package Details

Package	IAB_optimized_300x250.zip
Size	124 KB
Size Excluding Libraries	124 KB

Ad Creative Performance in the WWW

https://www.flickr.com/photos/70251312@N00/8659643962

Performance in the WWW

- **What is Slow**

If there was one topic that deserve the “most prolific” award, it would be Performance.
Not only do we have no shortage of performance advice, the topic shows no sign of drying up.
The problem is that everything comes with caveats and disclaimers, and sometime one piece of advice seem to actively contradict another.

General “truths” like “The DOM is slow” or “Always use CSS animations” make for great headlines, but the reality is often far more nuanced.

“So,” you ask, “is changing the DOM slow? What about loading a <script> in the <head>?
Also, does a 20-millisecond operation take too long? What about 0.5 seconds? 10 seconds?”

Performance in the WWW

- **What Actually Matters**

#perfmatters

but

#contextmatters

even more

It’s actually hard to say objectively whether something is slow or fast without the context of when it’s happening.

For different context, different answers.
Is your code running during idle time? Or in a touch handler? Or in the hot path of a game loop?

iab.

Performance in the WWW

- **For Whom and How It Matters**

performance

expectations

context

user

Put another way, the people using the website have the answers in the form of performance expectations for each of those contexts.
We build products for our users, and what they perceive is what matters the most. In fact, number one on Google's ten things we know to be true is "Focus on the user and all else will follow."

Performance in the WWW

- **What to Aim for?**

Desired Performance for a Good User experience ?

Then, what is the desired performance for a good user experience.

Performance in the WWW

- **The RAIL Performance Model**

RAIL

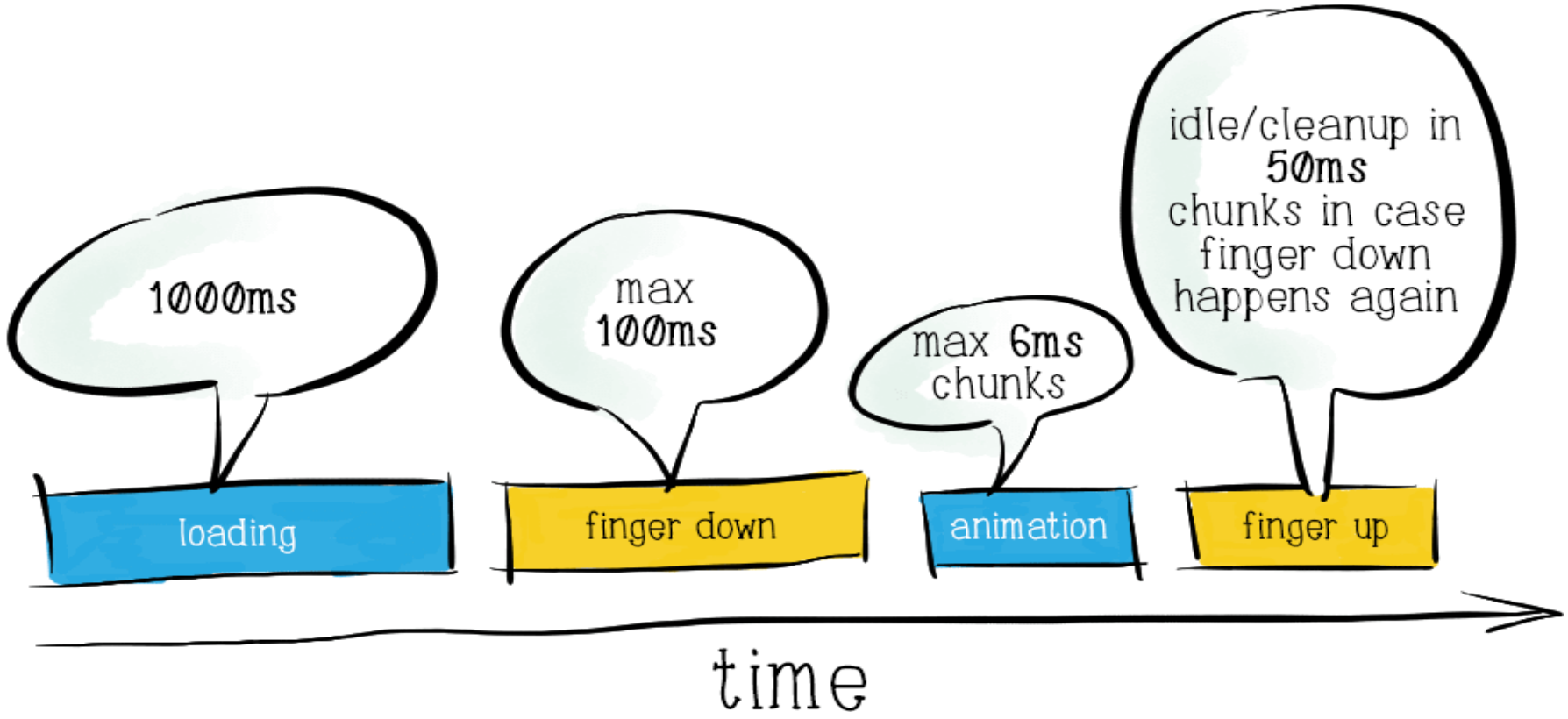
Response
Animation
Idleness
Loading

The Chrome team believes that a good user experience can be explained by specific budgets for each type of interactions.
We call it RAIL.
It stands for Response, Animation, Idleness and Loading.
It's based on existing user research, you can learn more at https://goo.gl/454PXY

iab.

Performance in the WWW

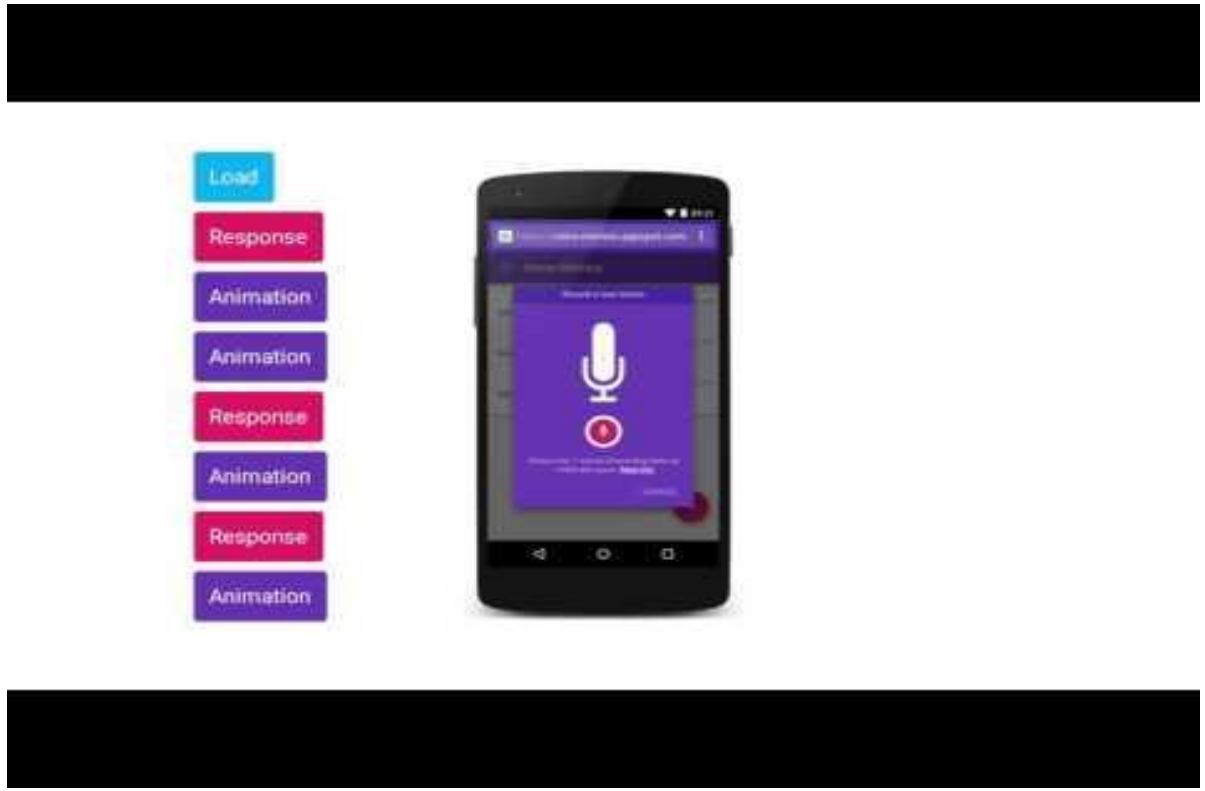
- **RAIL Budget**



http://www.smashingmagazine.com/2015/10/rail-user-centric-model-performance

Performance in the WWW

- **RAIL Example**



Here is a video illustrating the RAIL performance model.

Performance in the WWW

- **Shared Responsibility**

On the web, performance is a collaborative effort. There is only one main thread and other scarce resources such as network bandwidth, memory and battery which are shared by everyone, from first party to (many) third parties.

Relationships

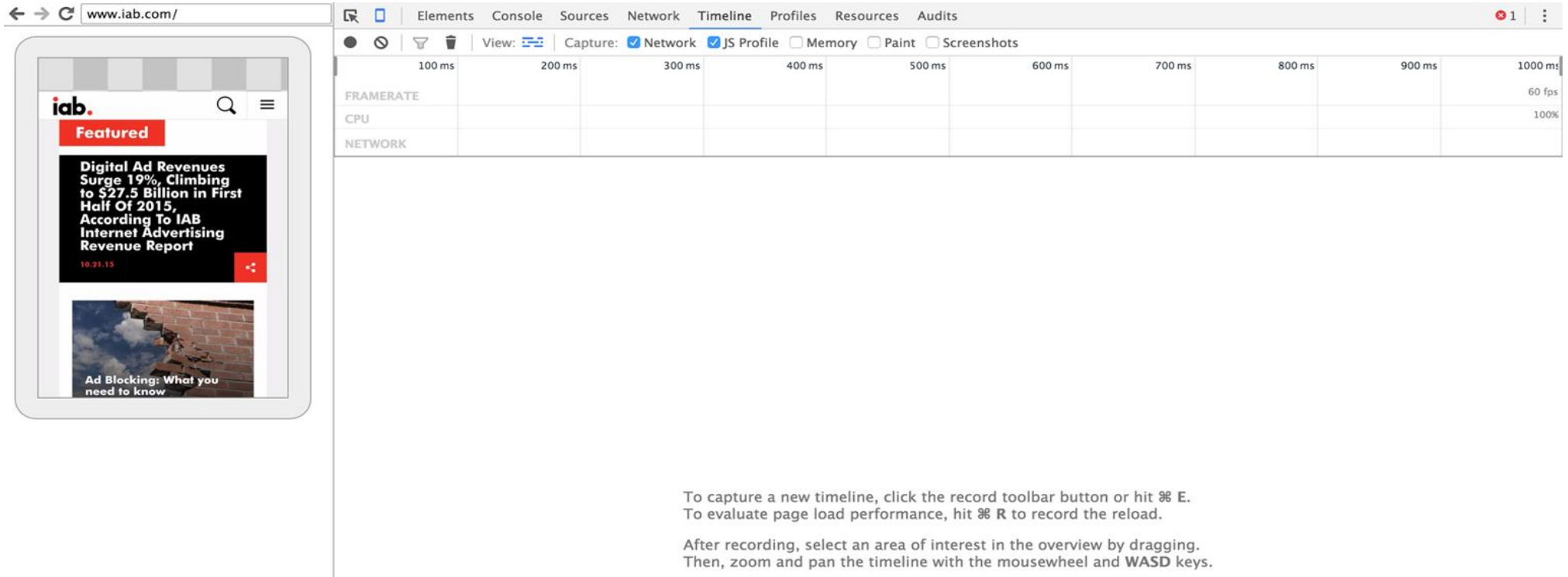
Photo by Brian Gratwicke

As an advertiser, you have 2 competing goals:

1. Get the user attention with an inventive creative which in turns help the publisher thrive
2. While avoiding impacting negatively the overall user experience.

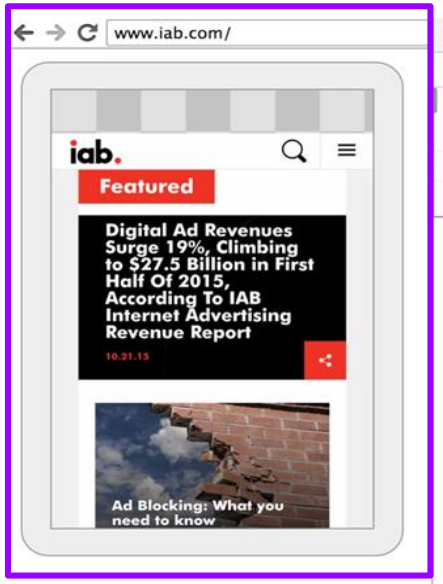
In the former case, there is a symbiosis relationship between you and the publisher while in the later case, the relationship between you and performance is one of commensalism: your presence should not hurt performance.

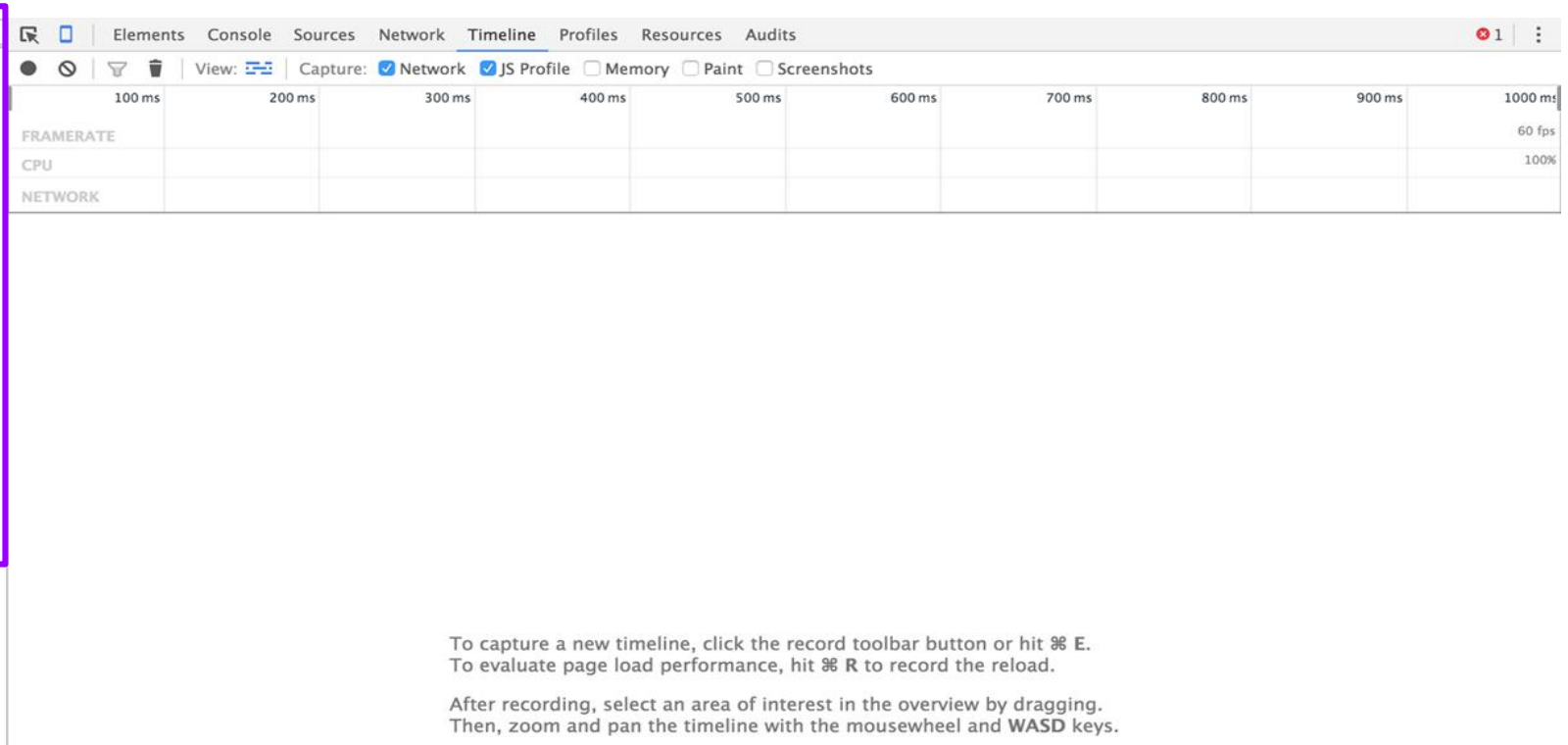
Performance: Dev Tool Overview



The right way to check for the performance of your creative on an actual mobile device via a tool like Chrome's devtools.
Let's take a look at the main UI elements that you will typically use.
(more details on how to get started can be found in the links section at the end of the presentation)

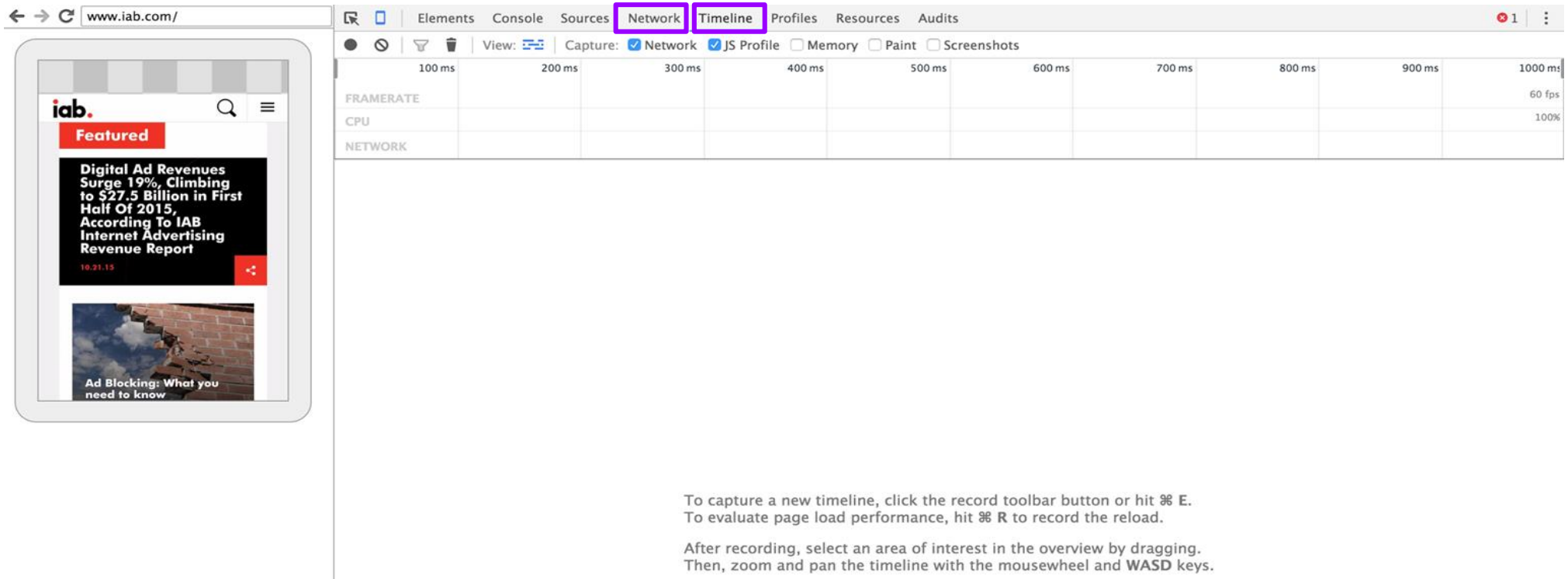
Performance: Dev Tool Overview





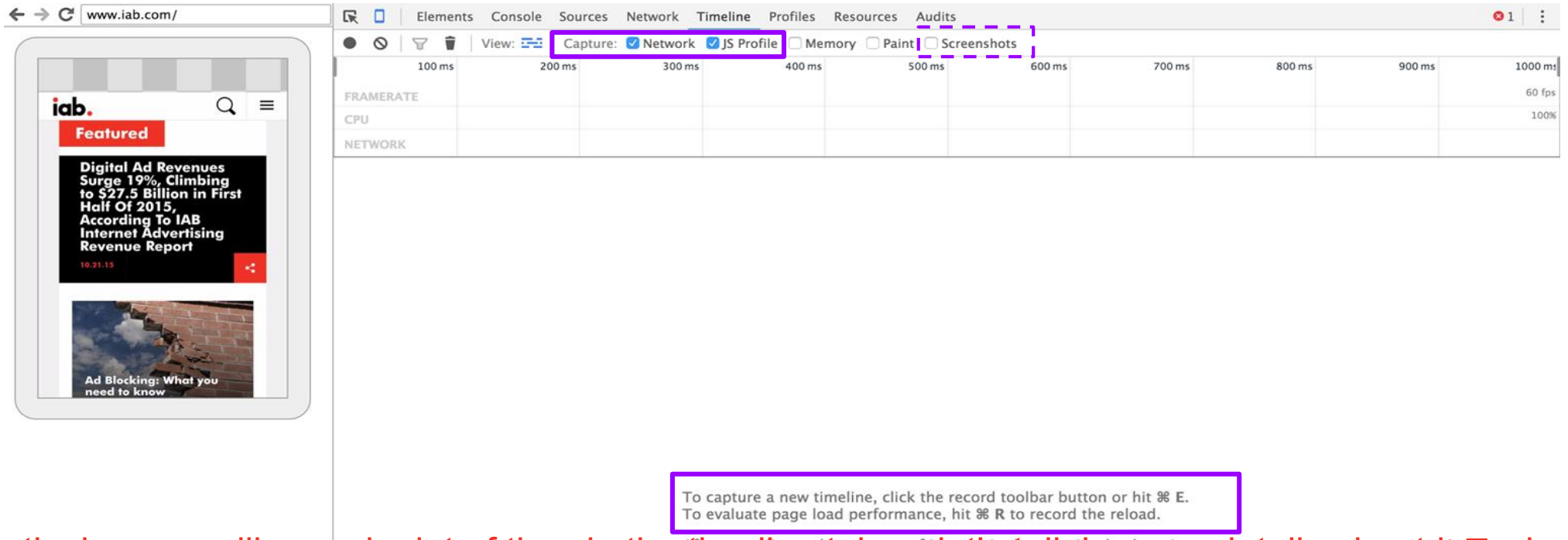
On the left hand side, you can see an overview of what's on your mobile device connected via USB.

Performance: Dev Tool Overview



DevTools provides a lot of features but you will mostly use 2 categories: network and timeline.
Network shows network activity while timeline provides an all-in-one view of all activities (main thread, network...).

Performance: Dev Tool Overview

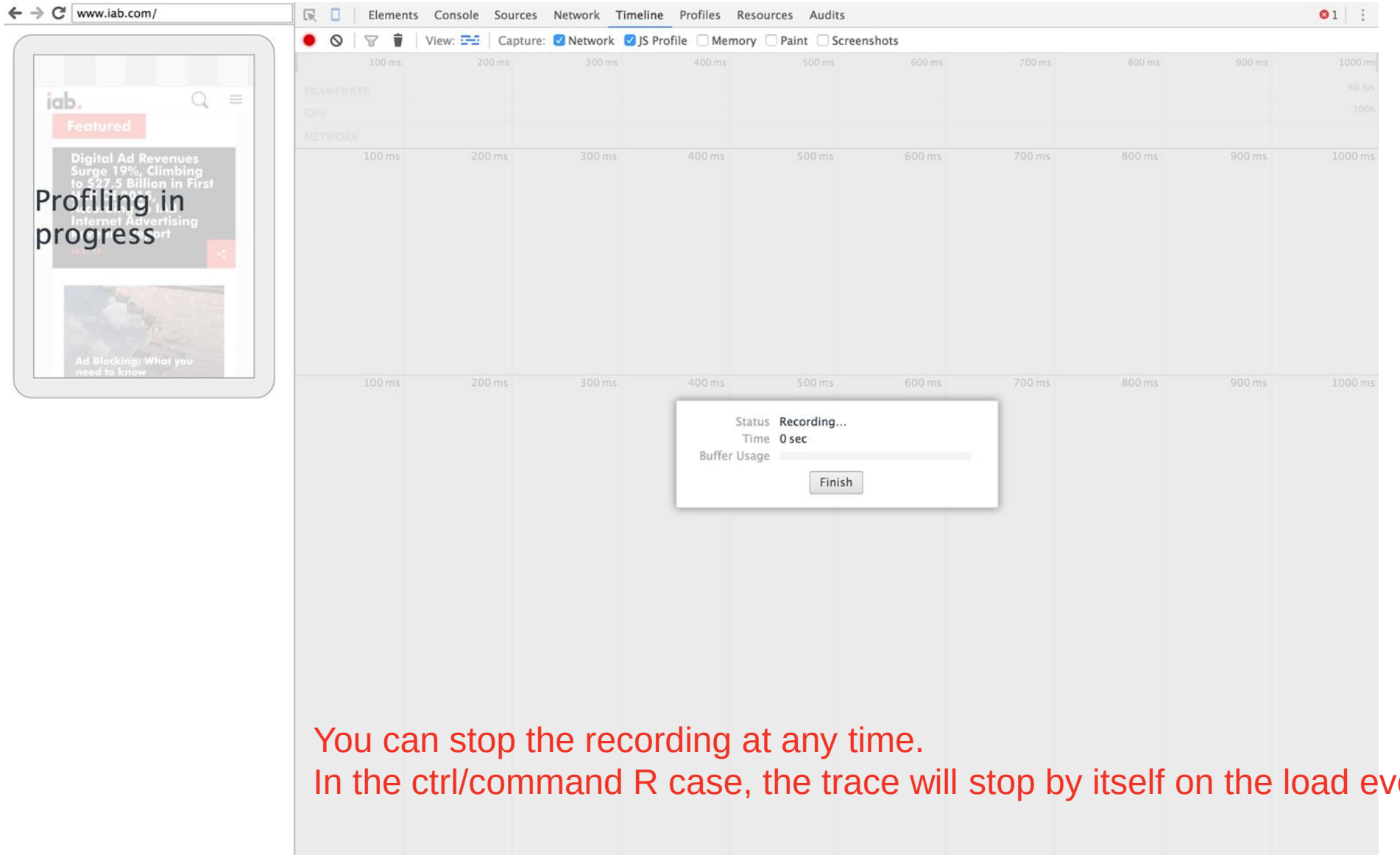


In particular, you will spend a lot of time in the timeline tab so let's talk in more details about it.Typically, you will want to check network (see the links section for how to enable this option) & js profile and optionally screenshots (note: has some overhead).

There are 2 ways to capture timeline traces:

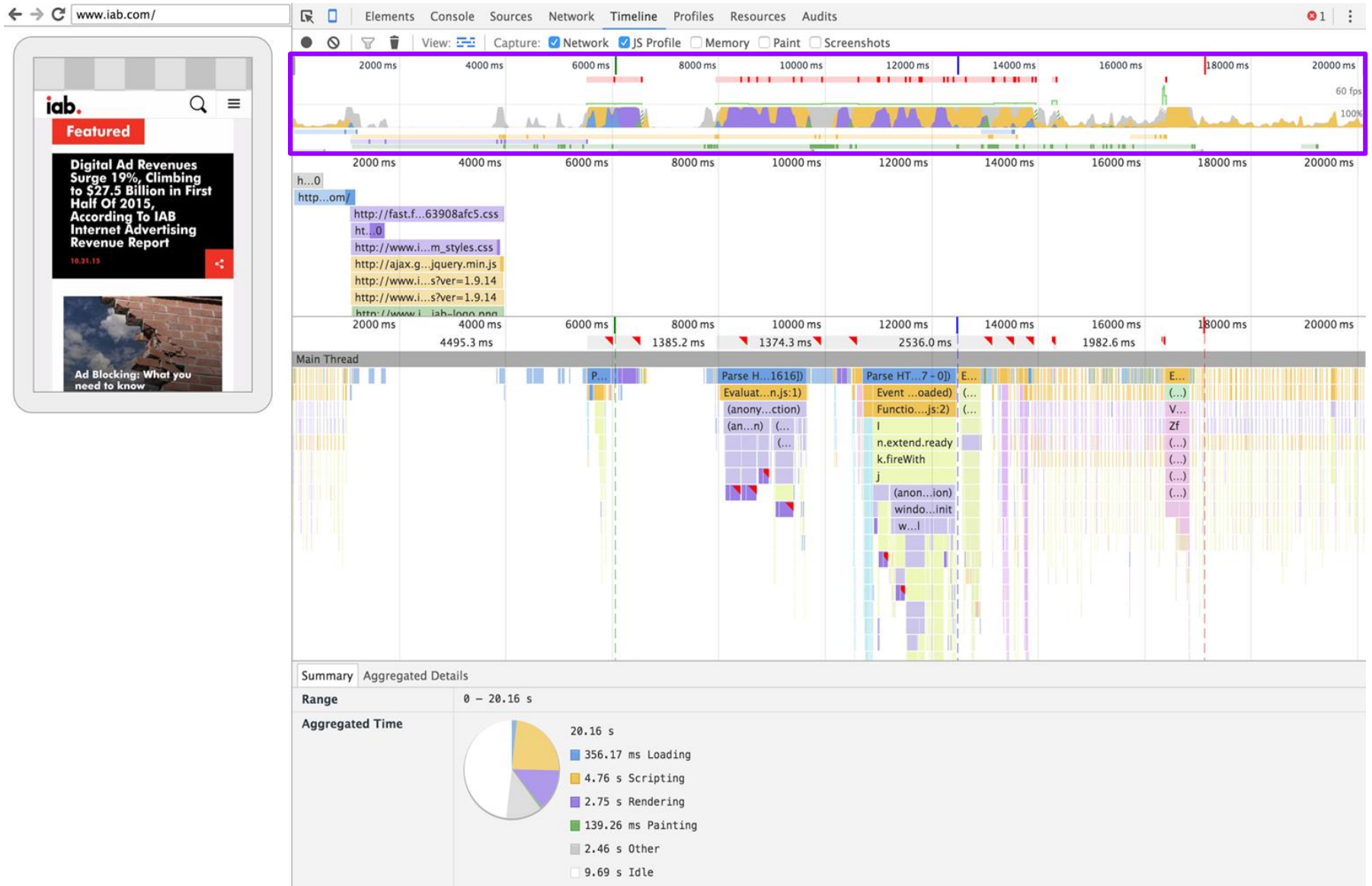
1. ctrl/command e to start a recording: which you would use to confirm the RAI aspect of the RAIL performance model.
2. or ctrl/command R to record a reload of the page: which you would use to confirm the L aspect of the RAIL performance model.

Performance: Dev Tool Overview



You can stop the recording at any time.
In the ctrl/command R case, the trace will stop by itself on the load event.

Performance: Dev Tool Overview



Let's look at other parts of the UI.
On the top, there is an overview for the whole timeline.

Performance in WWW

- **Dev Tool Overview**

Let's take a closer look. The overview UI consists of:

1. a timeline
2. a Jankiness bar showing Janky frames
3. frames per second graph (the higher the better)

Performance in WWW

- **Dev Tools Overview**

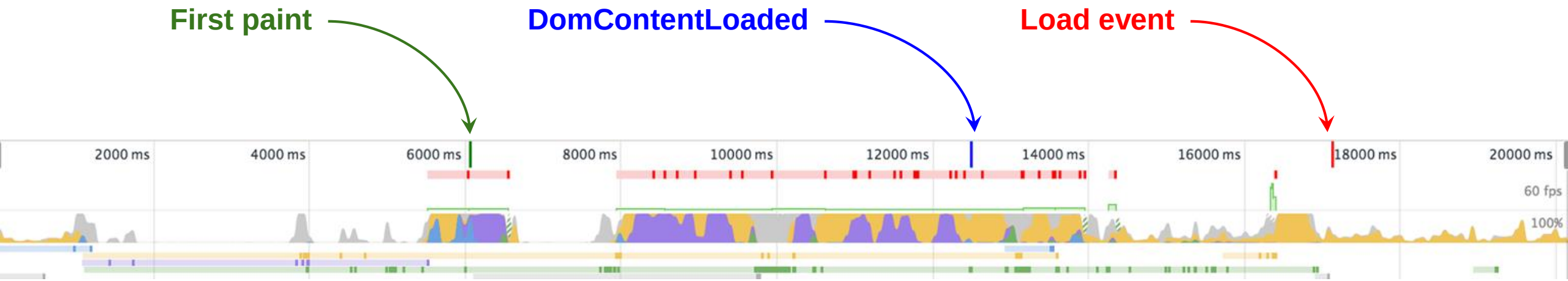
Main thread	Network (examples)
Loading, parsing	HTML
Scripts	JS
Rendering	CSS
Painting	Images
Other	xhr...

(continued)The overview UI consists of:

1. main thread overview
2. network overview
3. with color coded blocks as explained on this slide.

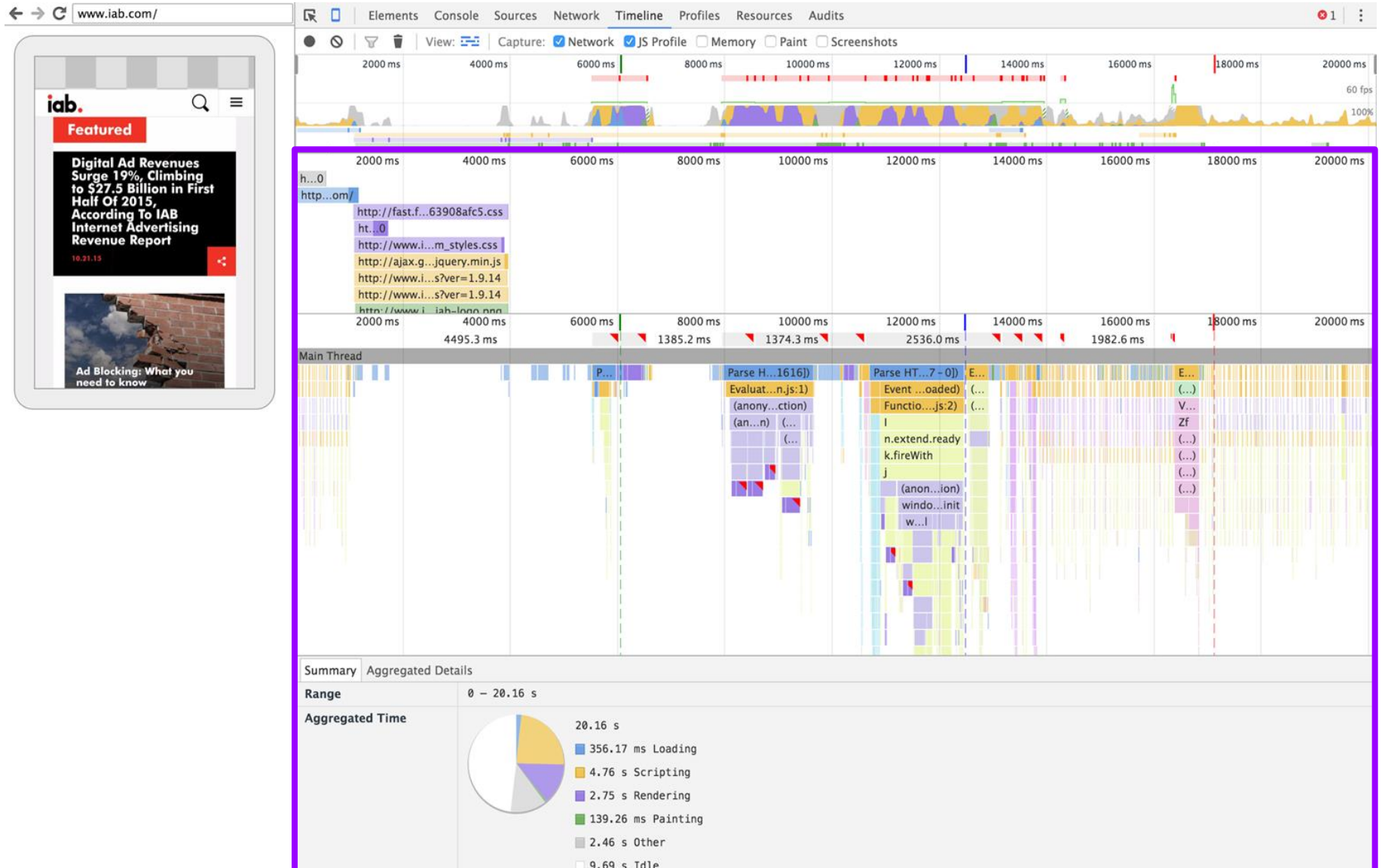
Performance in WWW

- **Dev Tools Overview**



On the timeline you might see some markers such as First paint, Dom Content Loaded, Load event. However, none of these really matter in a RAIL performance model.

Performance: Dev Tool Overview



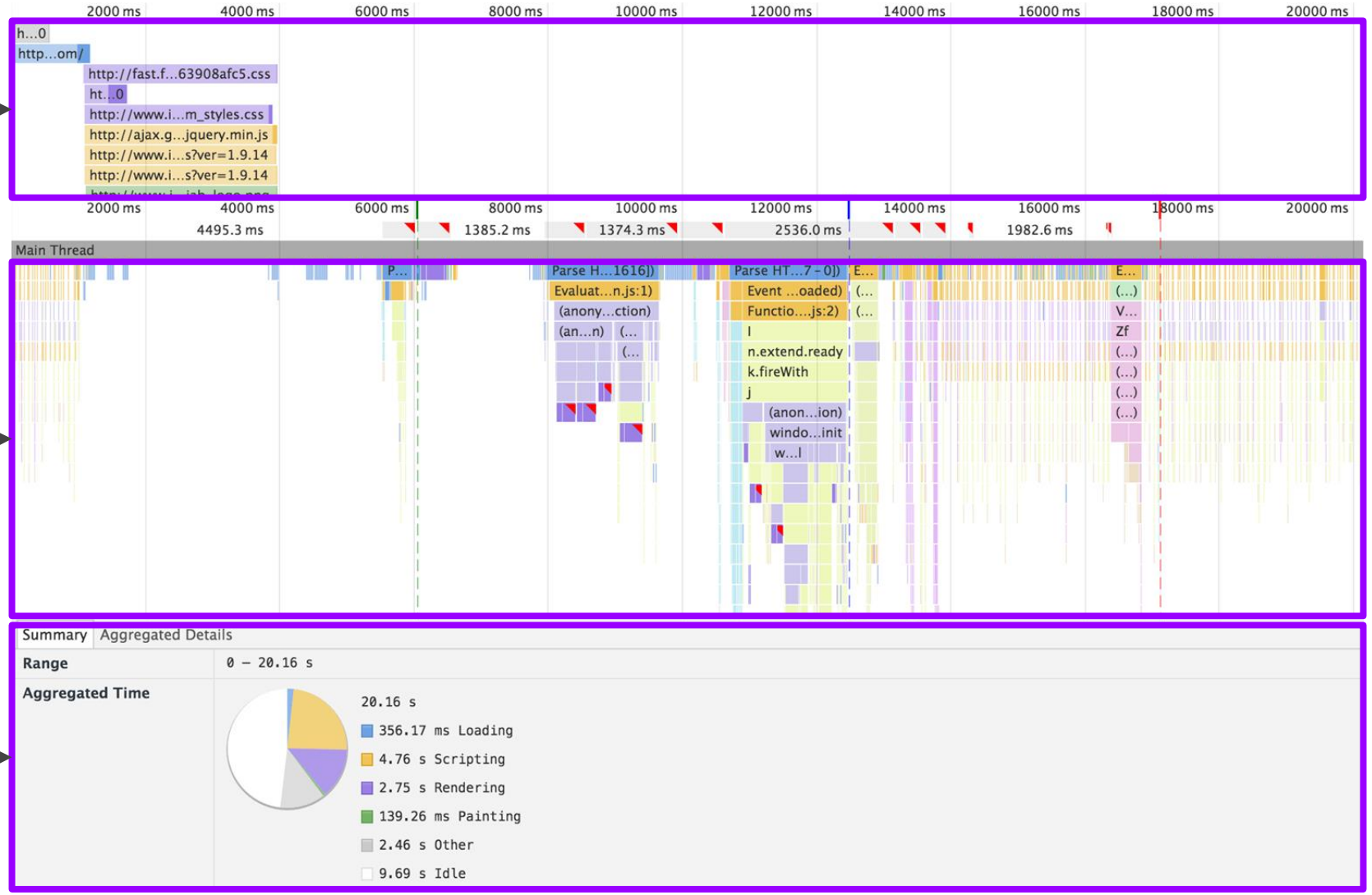
The second section shows more details for a section of the timeline.

Performance: Dev Tool Overview

Network activity

Main thread (flame chart)

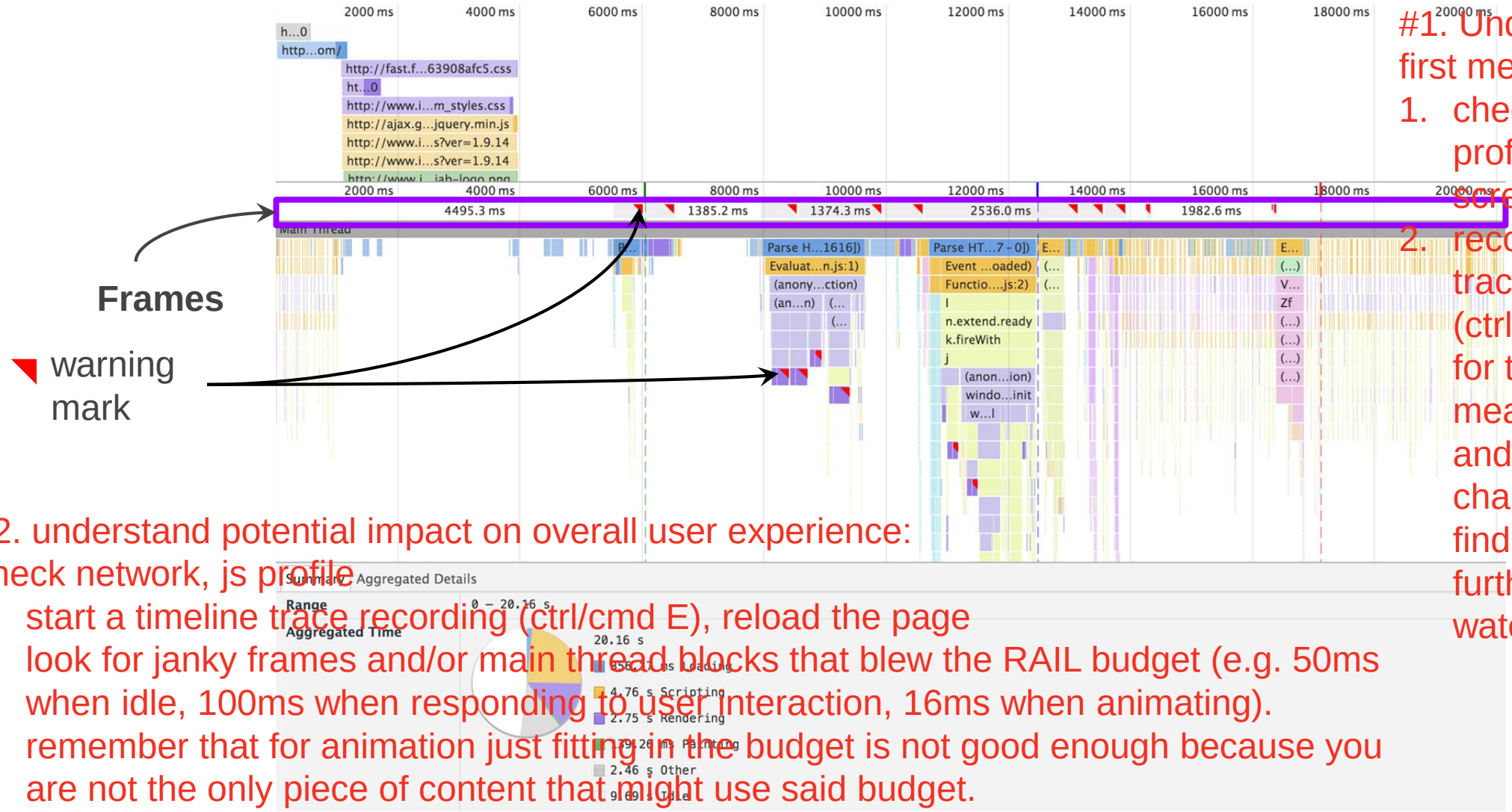
Overview for the range



It's composed of network activity, main thread flame chart and an overview panel for the selected range.

Performance: Dev Tool Overview

The main thread timeline has indicators for frames and warning marks to look for.



#1. Understand time to first meaningful paint:

1. check network, js profile and screenshots
2. record a timeline trace for a reload (ctrl/cmd R), look for the first meaningful paint and walk back the chain of events to find out ways to further optimize the waterfall.

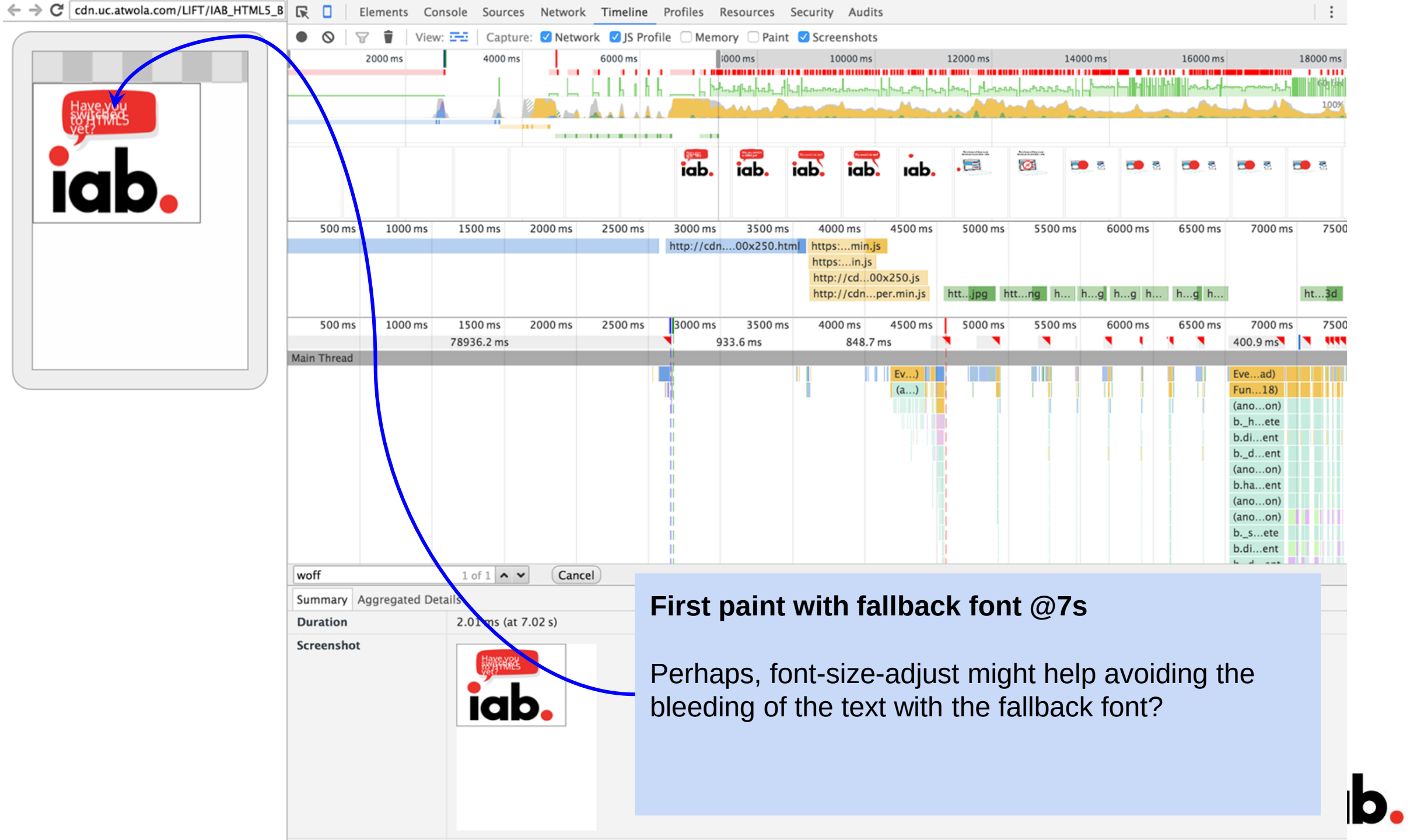
#2. understand potential impact on overall user experience: check network, js profile

1. start a timeline trace recording (ctrl/cmd E), reload the page
2. look for janky frames and/or main thread blocks that blew the RAIL budget (e.g. 50ms when idle, 100ms when responding to user interaction, 16ms when animating).
3. remember that for animation just fitting in the budget is not good enough because you are not the only piece of content that might use said budget.

Demo Time or what could possibly go wrong

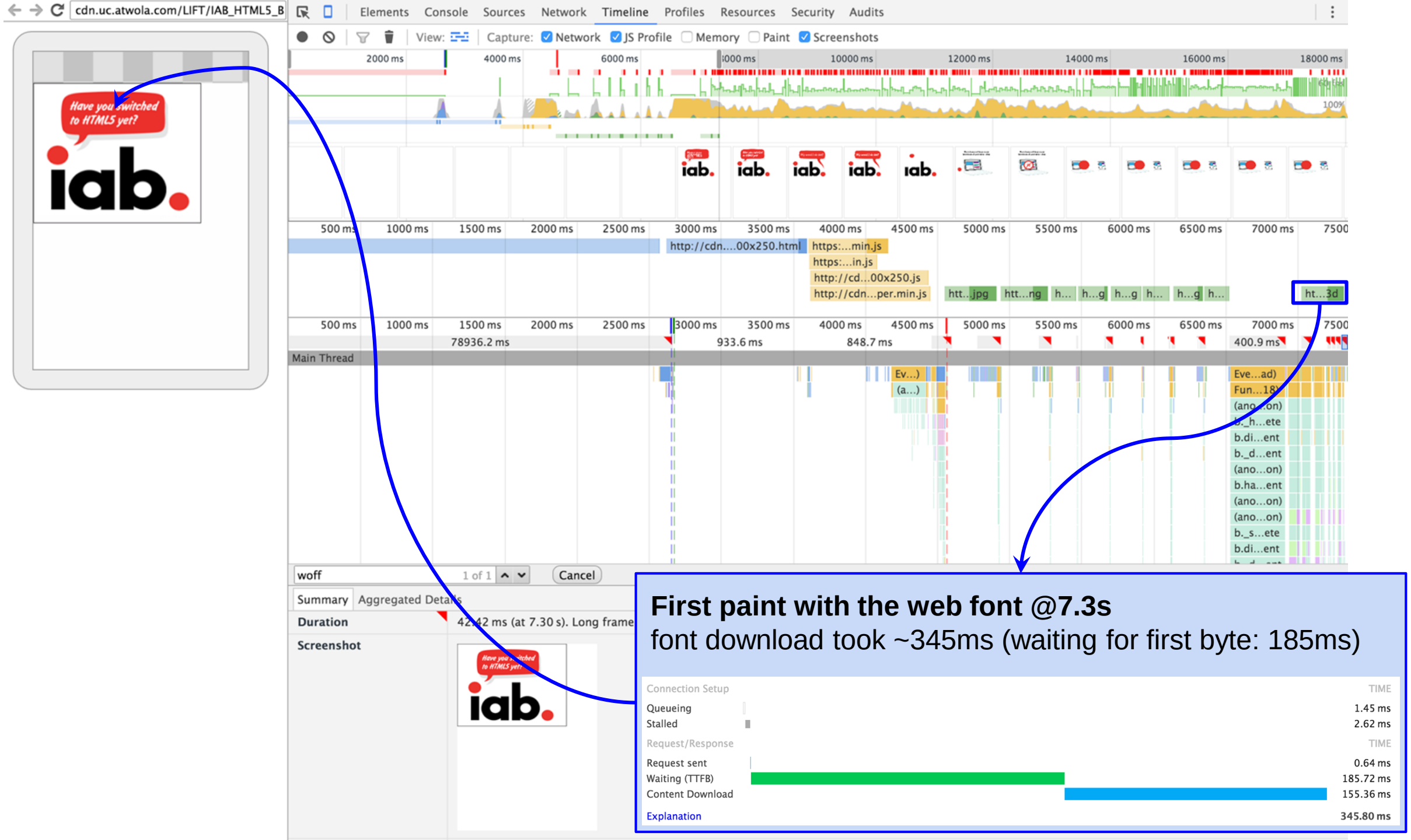
Photo by Lep

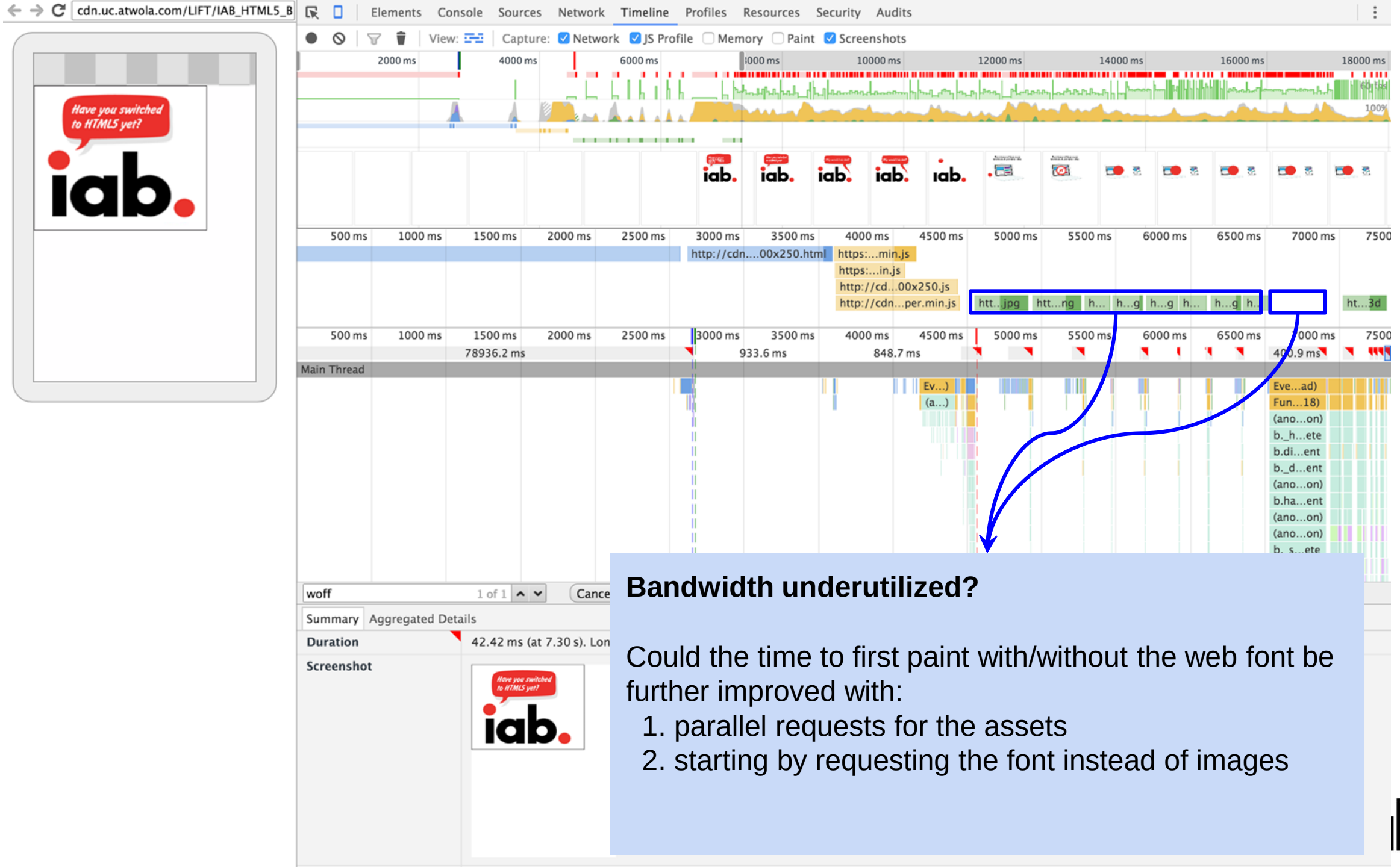
Non Optimized Version



First paint with fallback font @7s

Perhaps, font-size-adjust might help avoiding the bleeding of the text with the fallback font?

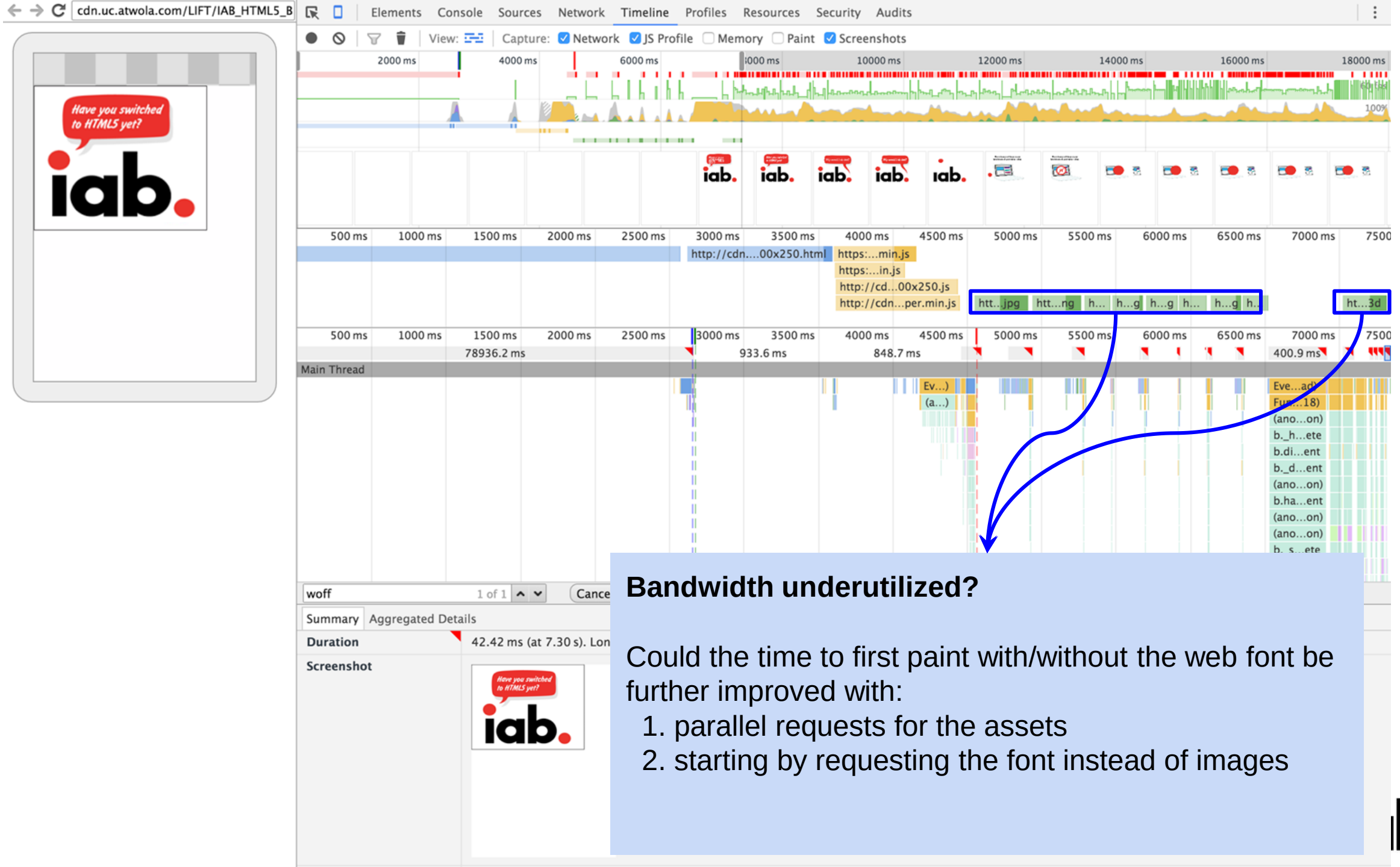

cdn.uc.atwola.com/LIFT/IAB_HTML5_B
Have you switched to HTML5 yet?
iab.
Elements Console Sources Network Timeline Profiles Resources Security Audits
View: Capture: Network JS Profile Memory Paint Screenshots
2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms 14000 ms 16000 ms 18000 ms
100%
500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms 4000 ms 4500 ms 5000 ms 5500 ms 6000 ms 6500 ms 7000 ms 7500
http://cdn....00x250.html
https:...min.js
https:...in.js
http://cd...00x250.js
http://cdn...per.min.js
htt...jpg
htt...ng
h...
h...g
ht...3d
78936.2 ms
933.6 ms
848.7 ms
400.9 ms
Main Thread
Ev...)
(a...)
Eve...ad)
Fun...18)
(ano...on)
b._h...ete
b.di...ent
b._d...ent
(ano...on)
b.ha...ent
(ano...on)
(ano...on)
b._s...ete
b.di...ent
woff
1 of 1
Cancel
Summary
Aggregated Details
Duration
42.42 ms (at 7.30 s). Long frame
Screenshot
First paint with the web font @7.3s
font download took ~345ms (waiting for first byte: 185ms)
Connection Setup
TIME
Queueing
1.45 ms
Stalled
2.62 ms
Request/Response
TIME
Request sent
0.64 ms
Waiting (TTFB)
185.72 ms
Content Download
155.36 ms
Explanation
345.80 ms



Bandwidth underutilized?

Could the time to first paint with/without the web font be further improved with:

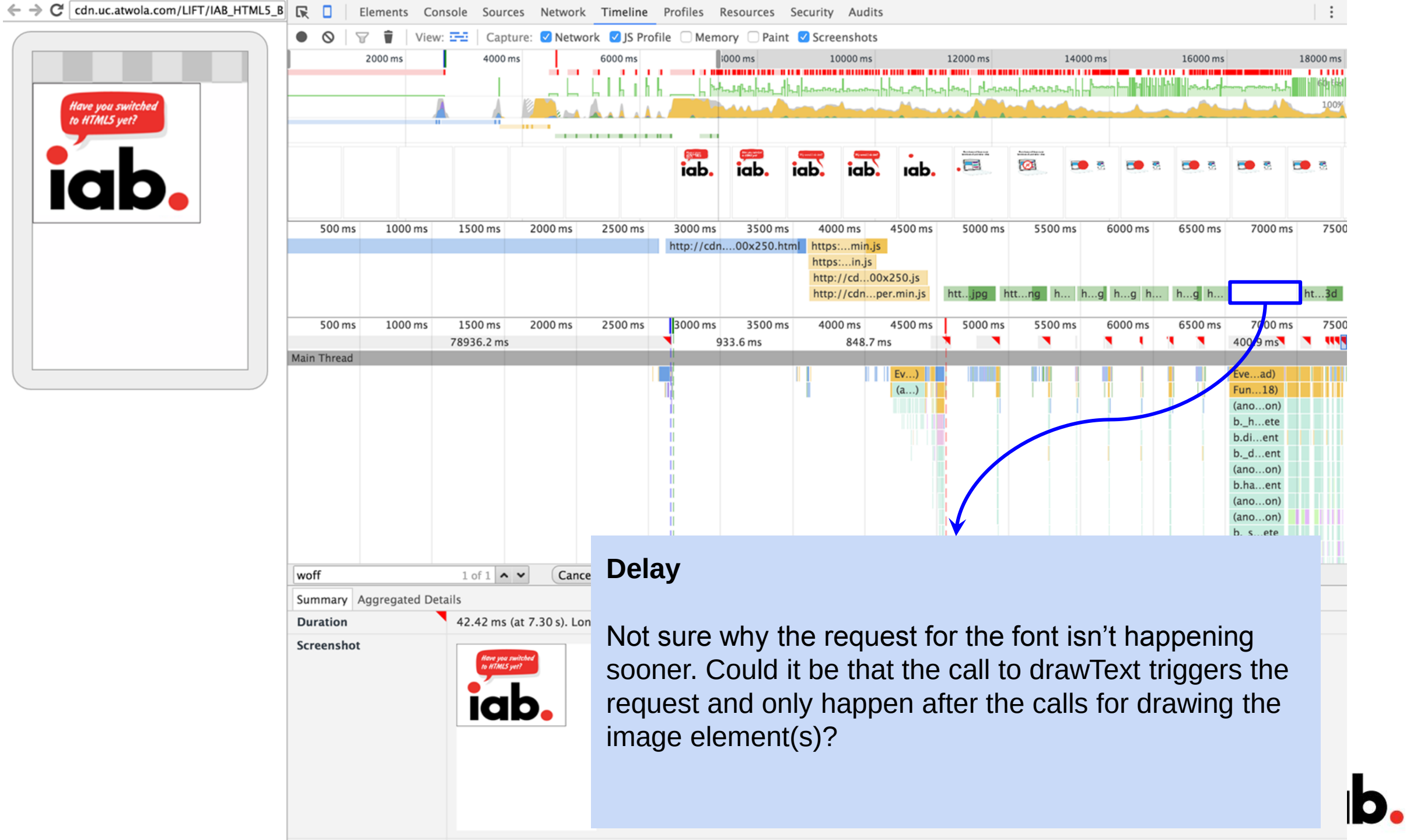
1. parallel requests for the assets
2. starting by requesting the font instead of images



Bandwidth underutilized?

Could the time to first paint with/without the web font be further improved with:

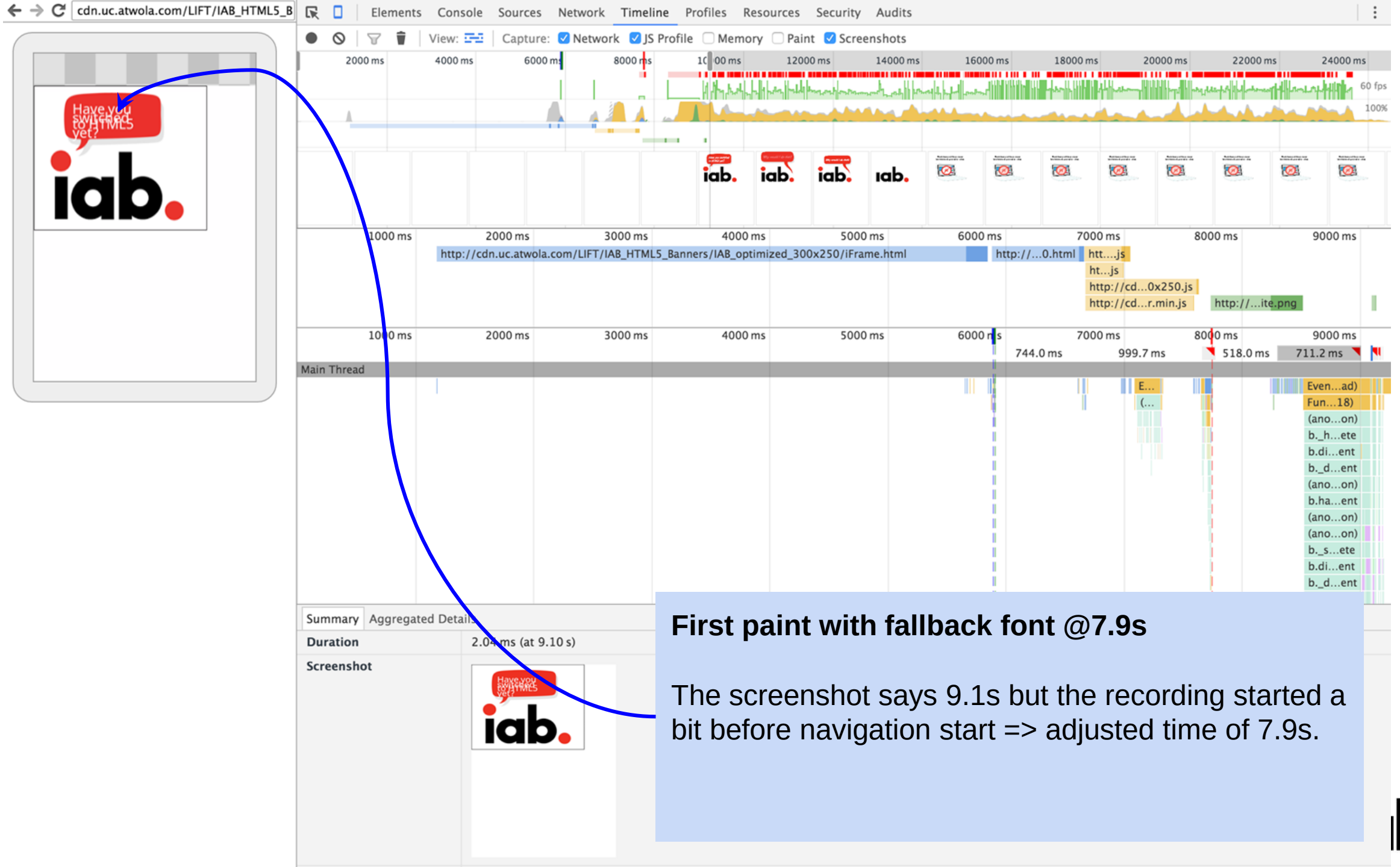
1. parallel requests for the assets
2. starting by requesting the font instead of images



Delay

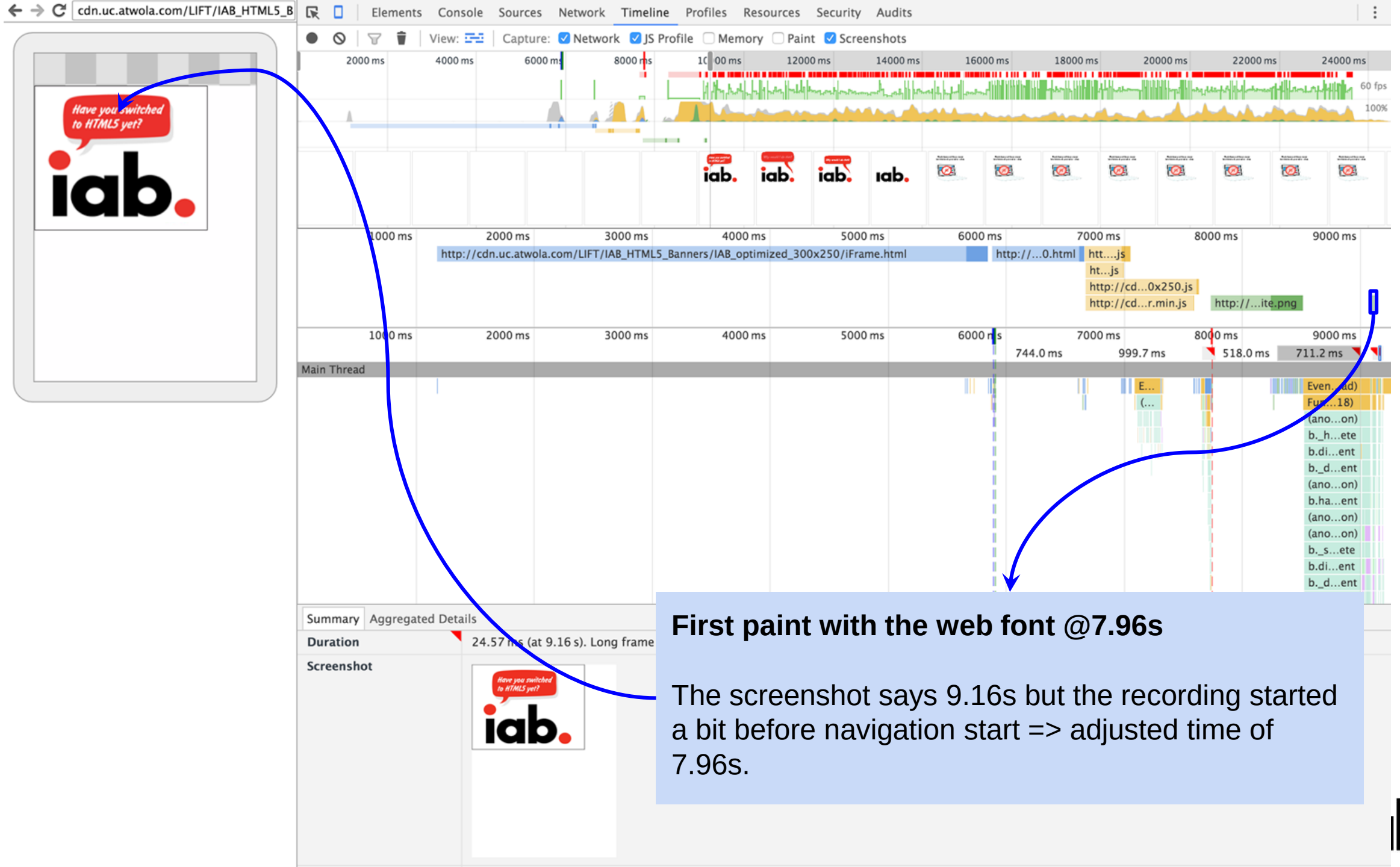
Not sure why the request for the font isn't happening sooner. Could it be that the call to drawText triggers the request and only happen after the calls for drawing the image element(s)?

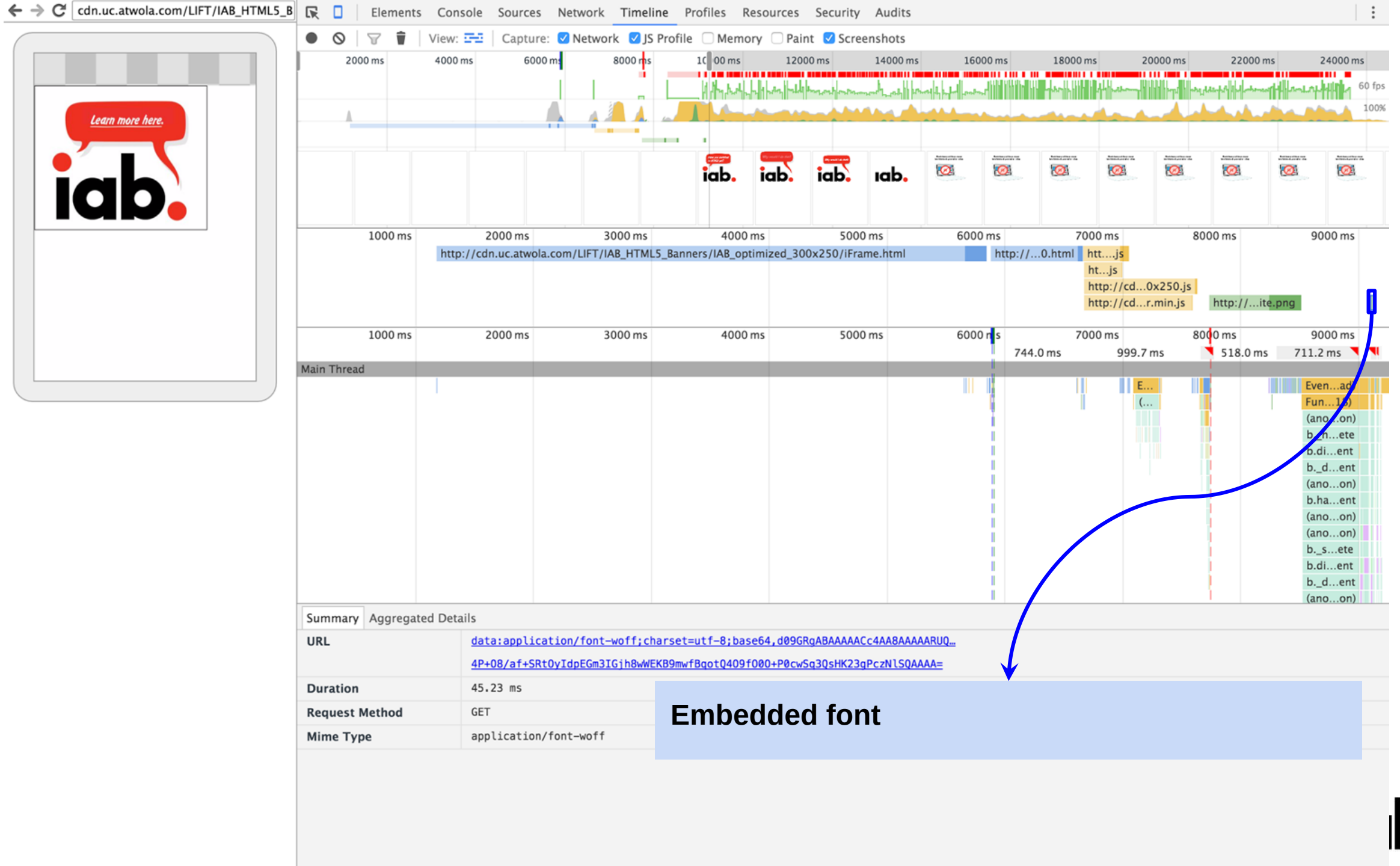
Optimized Version

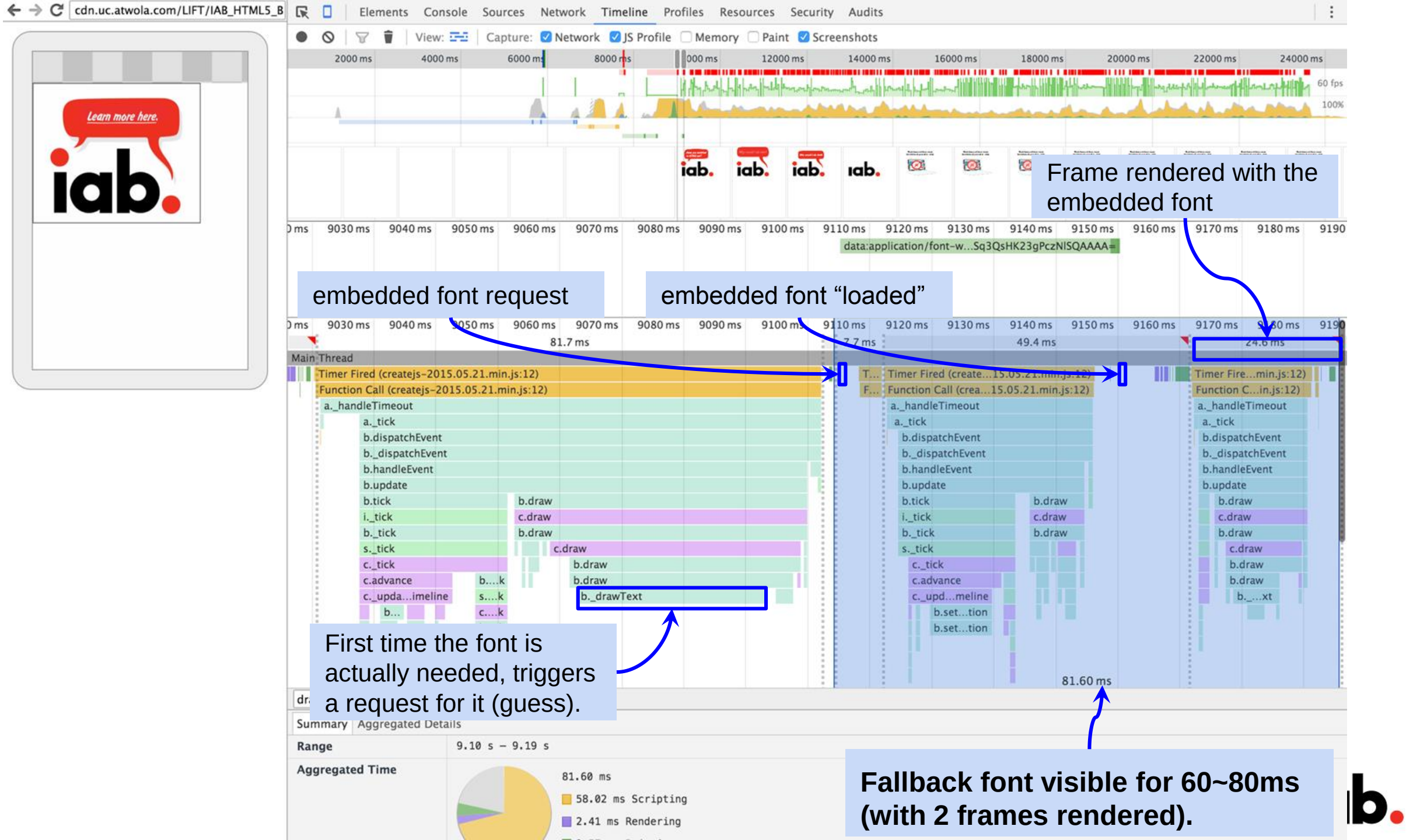


First paint with fallback font @7.9s

The screenshot says 9.1s but the recording started a bit before navigation start => adjusted time of 7.9s.

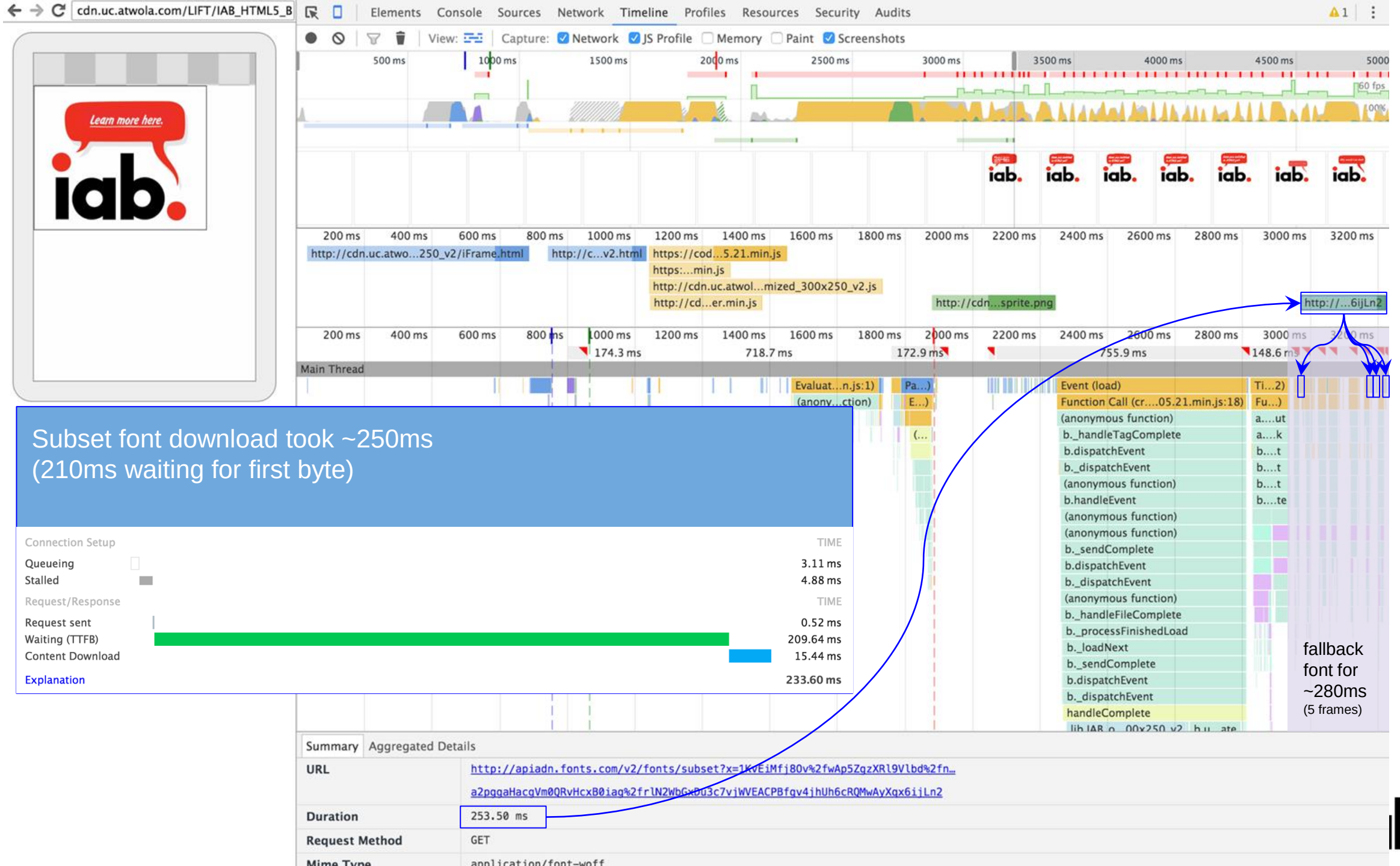

cdn.uc.atwola.com/LIFT/IAB_HTML5_B
Have you switched to HTML5 yet?
iab.
Elements
Console
Sources
Network
Timeline
Profiles
Resources
Security
Audits
View:
Capture:
Network
JS Profile
Memory
Paint
Screenshots
2000 ms
4000 ms
6000 ms
8000 ms
10000 ms
12000 ms
14000 ms
16000 ms
18000 ms
20000 ms
22000 ms
24000 ms
60 fps
100%
1000 ms
2000 ms
3000 ms
4000 ms
5000 ms
6000 ms
7000 ms
8000 ms
9000 ms
http://cdn.uc.atwola.com/LIFT/IAB_HTML5_Banners/IAB_optimized_300x250/iFrame.html
http://...0.html
htt....js
ht...js
http://cd...0x250.js
http://cd...r.min.js
http://...ite.png
744.0 ms
999.7 ms
518.0 ms
711.2 ms
Main Thread
E...
(...
(ano...on)
b._h...ete
b.di...ent
b._d...ent
b.ha...ent
b._s...ete
Summary
Aggregated Details
Duration
Screenshot
First paint with the web font @7.96s
The screenshot says 9.16s but the recording started a bit before navigation start => adjusted time of 7.96s.


cdn.uc.atwola.com/LIFT/IAB_HTML5_B
Learn more here.
iab.
Elements
Console
Sources
Network
Timeline
Profiles
Resources
Security
Audits
View:
Capture:
Network
JS Profile
Memory
Paint
Screenshots
2000 ms
4000 ms
6000 ms
8000 ms
10000 ms
12000 ms
14000 ms
16000 ms
18000 ms
20000 ms
22000 ms
24000 ms
60 fps
100%
1000 ms
2000 ms
3000 ms
4000 ms
5000 ms
6000 ms
7000 ms
8000 ms
9000 ms
http://cdn.uc.atwola.com/LIFT/IAB_HTML5_Banners/IAB_optimized_300x250/iFrame.html
http://...0.html
htt....js
ht...js
http://cd...0x250.js
http://cd...r.min.js
http://...ite.png
744.0 ms
999.7 ms
518.0 ms
711.2 ms
Main Thread
Summary
Aggregated Details
URL
data:application/font-woff;charset=utf-8;base64,d09GRgABAAAAACc4AA8AAAAARUQ…
4P+O8/af+SRtOyIdpEGm3IGjh8wWEKB9mwfBqotQ4O9fO00+P0cwSq3QsHK23qPczNlSQAAAA=
Duration
45.23 ms
Request Method
GET
Mime Type
application/font-woff
Embedded font


cdn.uc.atwola.com/LIFT/IAB_HTML5_B
Learn more here.
iab.
Elements Console Sources Network Timeline Profiles Resources Security Audits
View: Capture: Network JS Profile Memory Paint Screenshots
2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms 14000 ms 16000 ms 18000 ms 20000 ms 22000 ms 24000 ms
60 fps
100%
Frame rendered with the embedded font
9030 ms 9040 ms 9050 ms 9060 ms 9070 ms 9080 ms 9090 ms 9100 ms 9110 ms 9120 ms 9130 ms 9140 ms 9150 ms 9160 ms 9170 ms 9180 ms 9190
data:application/font-w...Sq3QsHK23gPczNlSQAAAA=
embedded font request
embedded font “loaded”
81.7 ms
49.4 ms
24.6 ms
Main Thread
Timer Fired (createjs-2015.05.21.min.js:12)
Function Call (createjs-2015.05.21.min.js:12)
a._handleTimeout
a._tick
b.dispatchEvent
b._dispatchEvent
b.handleEvent
b.update
b.tick
i._tick
b._tick
s._tick
c._tick
c.advance
c._upda...imeline
b.draw
c.draw
b._drawText
b.set...tion
First time the font is actually needed, triggers a request for it (guess).
81.60 ms
Summary Aggregated Details
Range 9.10 s – 9.19 s
Aggregated Time
81.60 ms
58.02 ms Scripting
2.41 ms Rendering
Fallback font visible for 60~80ms (with 2 frames rendered).

Subset Font Version

subset font

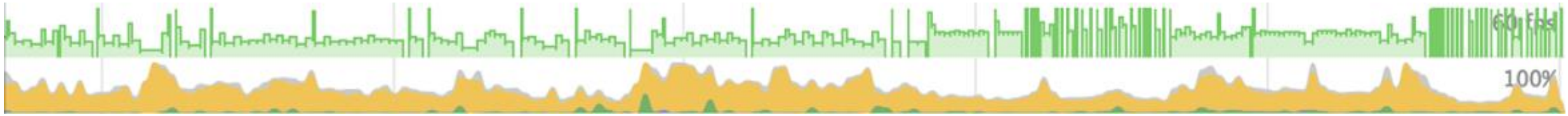


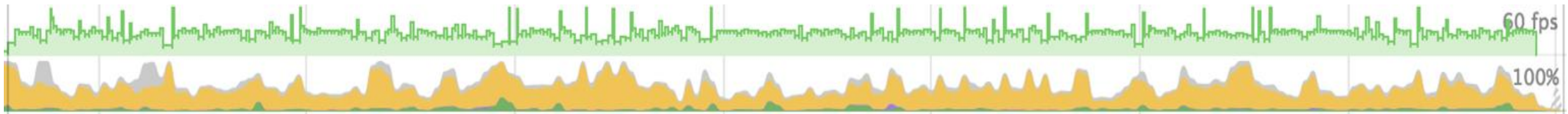
Subset font download took ~250ms
(210ms waiting for first byte)

Connection Setup	TIME
Queueing	3.11 ms
Stalled	4.88 ms
Request/Response	**TIME**
Request sent	0.52 ms
Waiting (TTFB)	209.64 ms
Content Download	15.44 ms
Explanation	233.60 ms

fallback font for ~280ms
(5 frames)

Timers vs. Synched rAF


60 fps
100%


60 fps
100%

HTML5 Ad Checklist

Build

- ✓ Beginning a Project (AS3 & converting vs. Canvas Doc type)
- ✓ Text Support
- ✓ Retina / High DPI Support
- ✓ Filters and Effects
- ✓ Spritesheet creation
- ✓ Image optimization
- ✓ Publish Settings
- ✓ Implement AdHelper.js

Optimize & Test

- ✓ Click Tags
- ✓ Zip Up
- ✓ RAIL Performance Model
- ✓ Browser Dev Tools

Useful Links

HTML5 Banner Ads With CreateJS
http://createjs.com/html5ads/

GitHub Repo With AdHelper
https://github.com/createjs/html5ads/

CreateJs / EaselJs
http://createjs.com/easeljs

HTML5 For Digital Advertising
http://www.iab.com/guidelines/html5-for-digital-advertising-1-0-guidance-for-ad-designers-creative-technologists/

IAB Display Advertising Guidelines
http://www.iab.com/guidelines/iab-display-advertising-guidelines/

IAB HTML5 Wiki
http://www.iab.net/wiki/index.php/HTML5_for_Digital_Advertising_Resources

RAIL/ Dev Tool
https://goo.gl/jfBMjN

Thank You