Start-up Performance of Web Applications

Qi Sun

CONTENTS

PART ONE

Background

Why start-up performance matters

- 53% of mobile site visits are abandoned if the pages take longer than **3 seconds** to load.

- The average load time for mobile sites is **19 seconds** on 3G connections.

——Webpagetest.org, February 2016

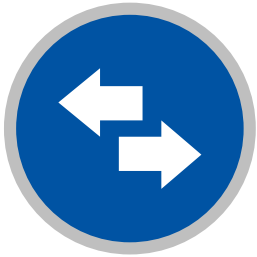# How to measure performance: RAIL

5

PART TWO

Facts & Strategies

Facts that influence start-up performance

Network

Fetch files and resources from server to user.

Scripting

Parse, compile & execute JavaScript.

rendering

Display items on user's screen.

Network

- 45% of mobile connections occurred over 2G worldwide.

- 75% of connections occurred on either 2G or 3G.

——GSMA, https://www.gsma.com/mobileeconomy/

Strategies

- Minify your code. (Uglify)

- Compress the files heavily. (Gzip)

- Remove unused code. (code-coverage tool: Google Chrome's Developer Tool, uncss)

- Preload all resources in the main document, or even better, preload with HTTP/2.0.
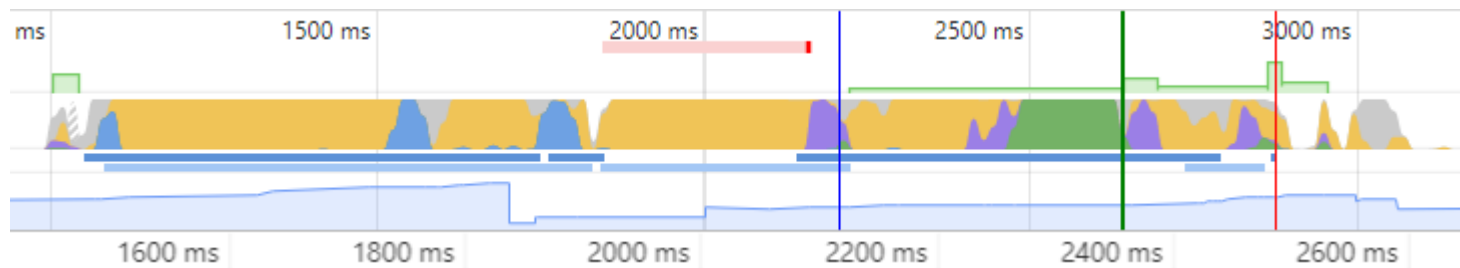
Scripting

- Test on Course Website with Chrome's Developer Tools

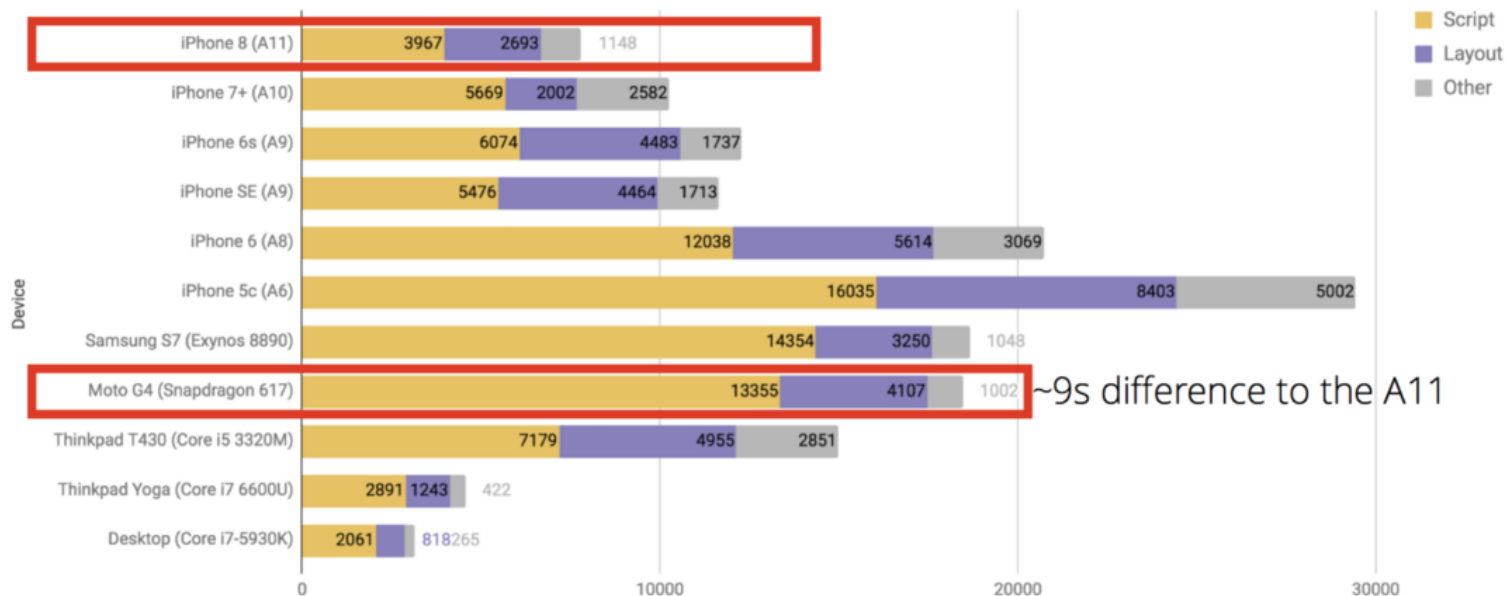| Self Time		Total Time		Activity
1287.3 ms	73.7 %	1287.3 ms	73.7 %	▶ ■ Scripting
175.3 ms	10.0 %	175.3 ms	10.0 %	▶ ■ Painting
157.9 ms	9.0 %	157.9 ms	9.0 %	▶ ■ Rendering
125.7 ms	7.2 %	125.7 ms	7.2 %	▶ ■ Loading

The most time-costing task for your browser!

Scripting

- Loading time is acceptable, but how about other devices?

  - global average cost this year is only 245.1$

Mobile cnn.com browser main thread time (Safari and Chrome)

Device	Script	Layout	Other
iPhone 8 (A11)	3967	2693	1148
iPhone 7+ (A10)	5669	2002	2582
iPhone 6s (A9)	6074	4483	1737
iPhone SE (A9)	5476	4464	1713
iPhone 6 (A8)	12038	5614	3069
iPhone 5c (A6)	16035	8403	5002
Samsung S7 (Exynos 8890)	14354	3250	1048
Moto G4 (Snapdragon 617)	13355	4107	1002
Thinkpad T430 (Core i5 3320M)	7179	4955	2851
Thinkpad Yoga (Core i7 6600U)	2891	1243	422
Desktop (Core i7-5930K)	2061	818	265

~9s difference to the A11

https://www.statista.com/statistics/484583/global-average-selling-price-smartphones/
https://medium.com/dev-channel/the-cost-of-javascript-84009f51e99e

Scripting

- Loading time is acceptable, but how about other devices?

  - Test of course website, on simulated Nexus 5X with Chrome

Strategies
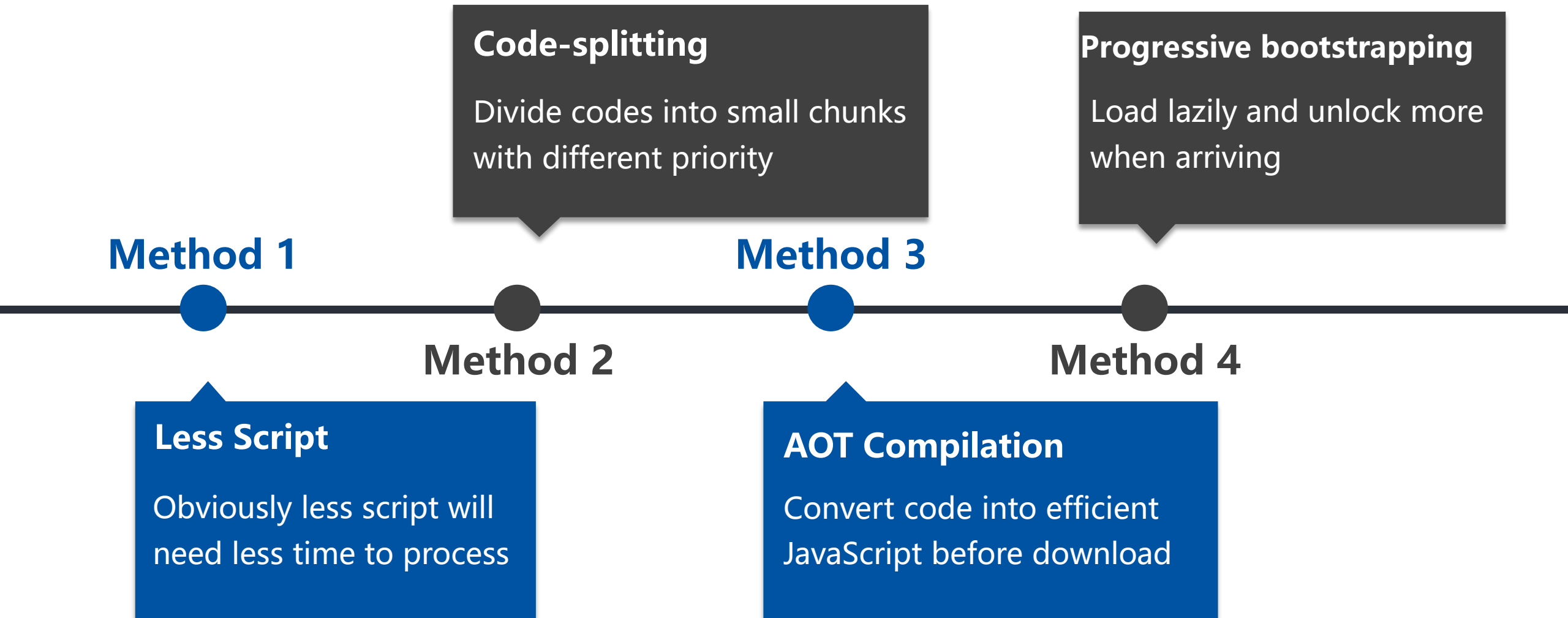
Code-splitting

Divide codes into small chunks with different priority

Progressive bootstrapping

Load lazily and unlock more when arriving

Method 1

Method 3

Method 2

Method 4

Less Script

Obviously less script will need less time to process

AOT Compilation

Convert code into efficient JavaScript before download

Rendering: Server-side VS Client-side

- Server Side Rendering

  - Good choice for static, less interactive pages

  - Convenient for search engine to parse

- Client Side Rendering

  - More efficient for webpages with complex features

  - Hard to handle: bundling, transpiling, linting, cache bursting……
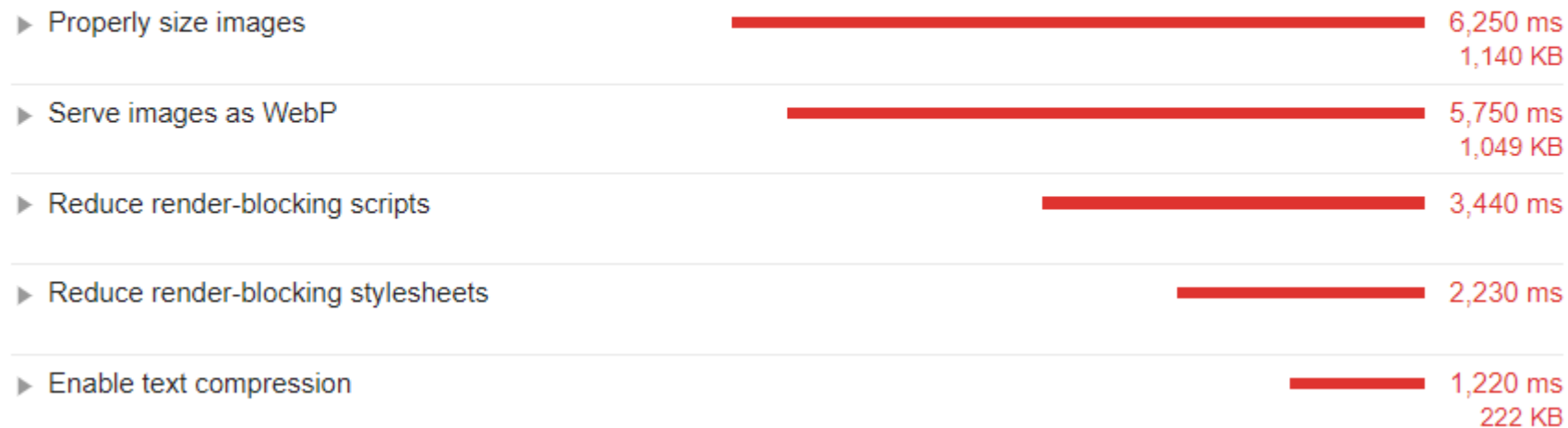
  - No worry, frameworks will do it for us!!

PART THREE

Summary

Summary

- Network, Scripting, Rendering

Opportunities
These are opportunities to speed up your application by optimizing the following resources.

▶ Properly size images	6,250 ms 1,140 KB
▶ Serve images as WebP	5,750 ms 1,049 KB
▶ Reduce render-blocking scripts	3,440 ms
▶ Reduce render-blocking stylesheets	2,230 ms
▶ Enable text compression	1,220 ms 222 KB

THANKS