



Speedometer 3

Bas Schouten

Table of contents

1. [A Brief History of Browser Benchmarks](#)
2. [Starting Speedometer 3](#)
3. [Defining New Workloads](#)
4. [Using Speedometer to Improve Firefox](#)

A Brief History of Browser Benchmarks

01

‘The second browser wars’

Benchmarks emerged



The Universal Browser Test

DROMAEO

JAVASCRIPT PERFORMANCE TESTING

Octane

Kraken



JetStream2



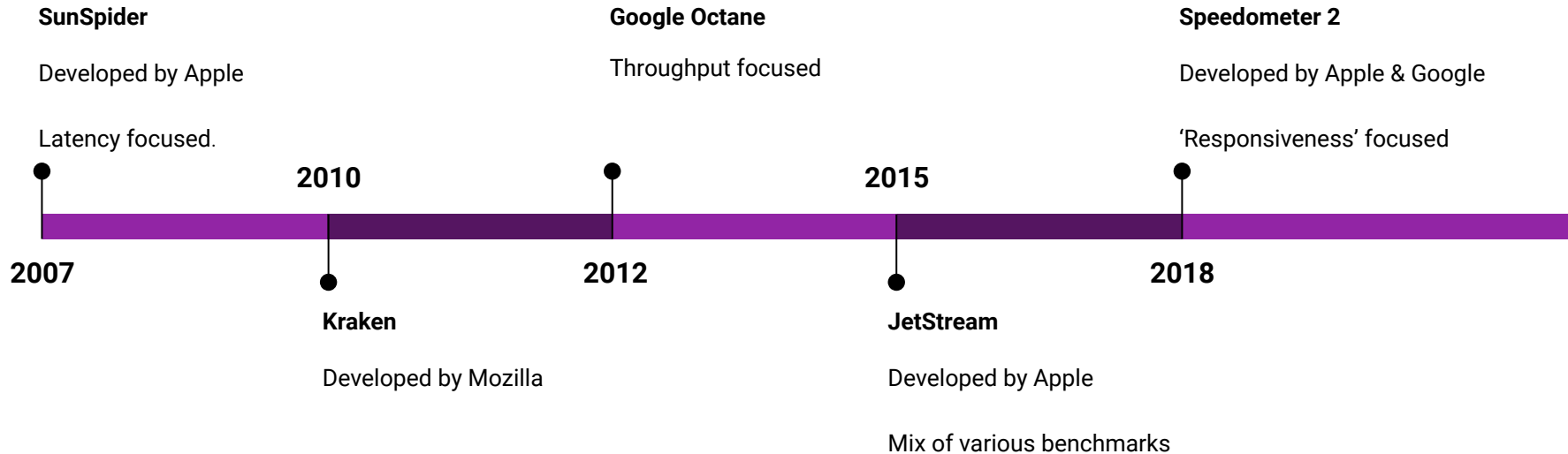
SunSpider

speedometer



Popular Benchmarks

Benchmarks became the source of truth on browser performance



Some Challenges

These benchmarks help us quantify performance, but ...

Representativity

- Focus on the **easily measurable**
- Strong **JS** emphasis
- **Ever-changing** web environment

Cross-browser inconsistency

- **Scheduling** differences
- **Fingerprinting** Resistance
- **API** availability

Subjectivity

- Developed by a **single vendor**
- **Marketing** interests



What makes a browser fast?

This question has no simple answer.

SOLVED: What is the Fastest Browser... In Real World Use

Published by Ian Matthews on June 22, 2012

Browser	Avg. Page Load Time (sec)	Pageviews ↓
<input type="checkbox"/> 1. Internet Explorer	7.92	37,954
<input type="checkbox"/> 2. Firefox	12.29	23,663
<input type="checkbox"/> 3. Chrome	12.78	22,289
<input type="checkbox"/> 4. Safari	30.86	5,126



(Source: <https://www.urtech.ca/2012/06/solved-what-is-the-fastest-browser-in-real-world-use/>)



Starting Speedometer 3

02

Advantages of a Benchmark

Despite RUM, benchmarks are needed

Proactive vs Reactive

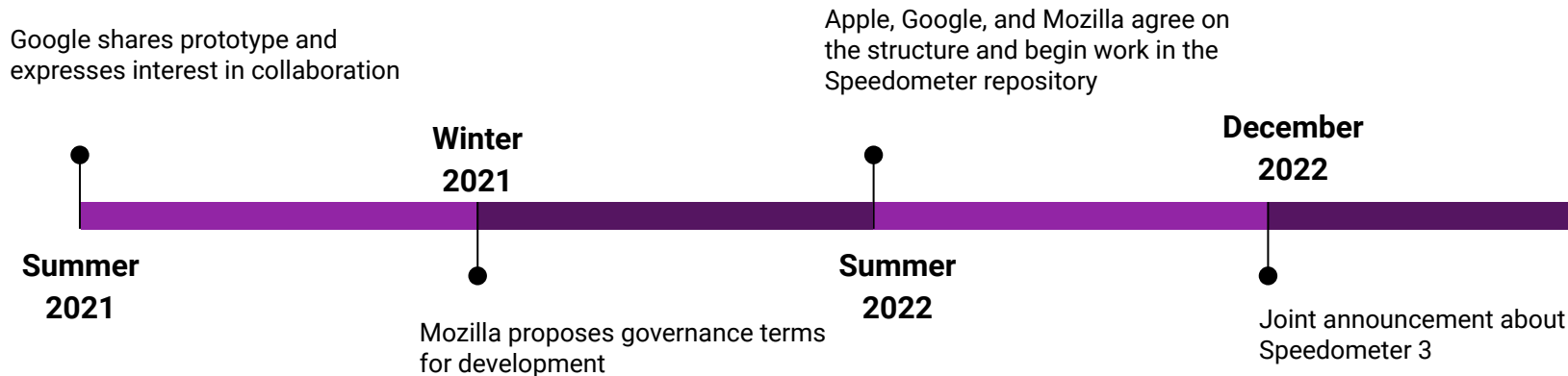
- Run in **continuous integration**
- Allow local **experimentation**
- Guide **decision making**

Competitive Analysis

- **Consistent** environments
- **Shared** understanding
- Improve the **web platform**



How it started



The Goals

Why are we doing this?

Performant Experiences

- **Alignment** on what should be fast
- **Shared** definition
- A **performant web** for all

Effective Optimizations

- Higher score = **faster web**
- No **overfitting**
- Avoid **microbenchmarks**



The Principles

What makes Speedometer 3 unique?

Joint Governance

- **Major changes** require **consensus**
- Browser **engine** projects

For the Real Web

- Workloads based on **real applications**
- **Community** contributions
- **Evolving** over time

Simplicity

- **Trivial** to run
- **Short** duration
- One **authoritative score**

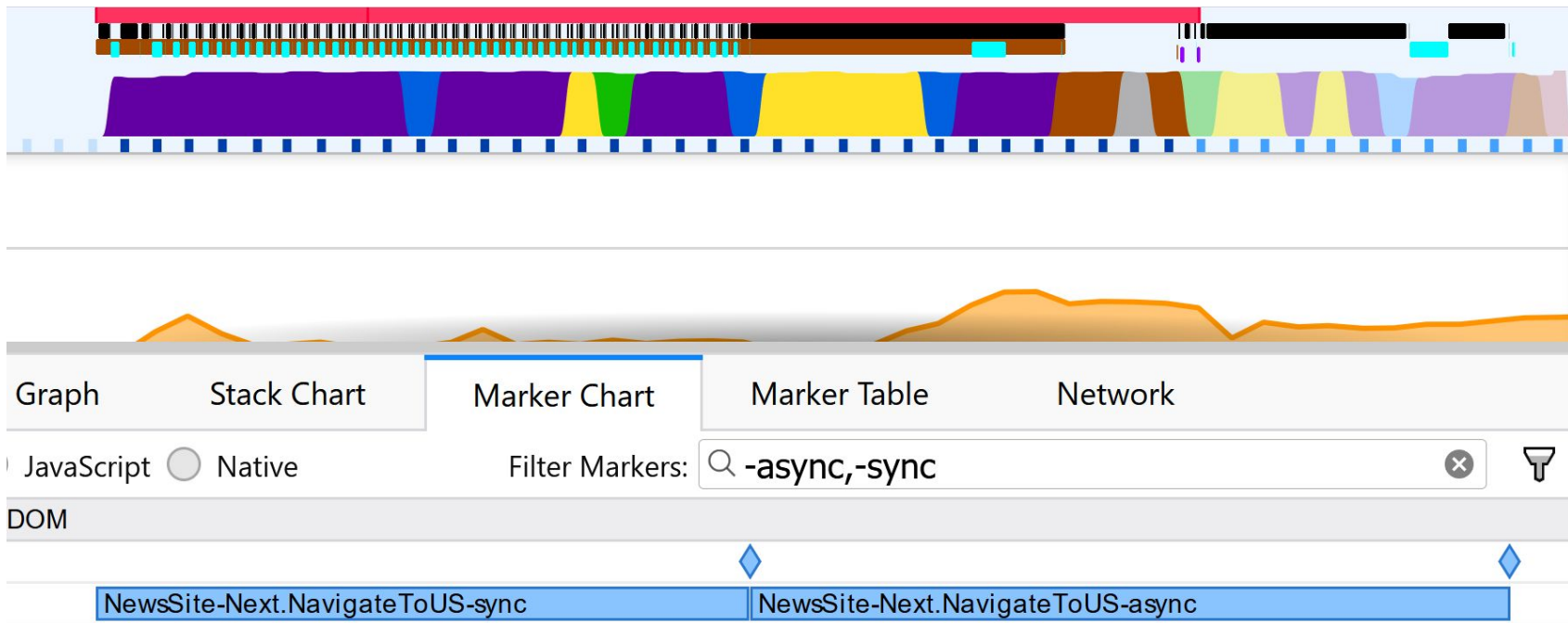


Defining New Workloads

03

The Methodology

What does Speedometer measure?



Where to begin...

How do you start a new workload?

Technologies

- Define technologies
- Form application
- Easily becomes unrealistic

Existing Applications

- Take an existing application
- Reduce into a measurable workload
- Often closed source

User Journey

- Define a user journey
- Build a realistic application
- Labor-intensive



Todo-MVC

todos

What needs to be done?

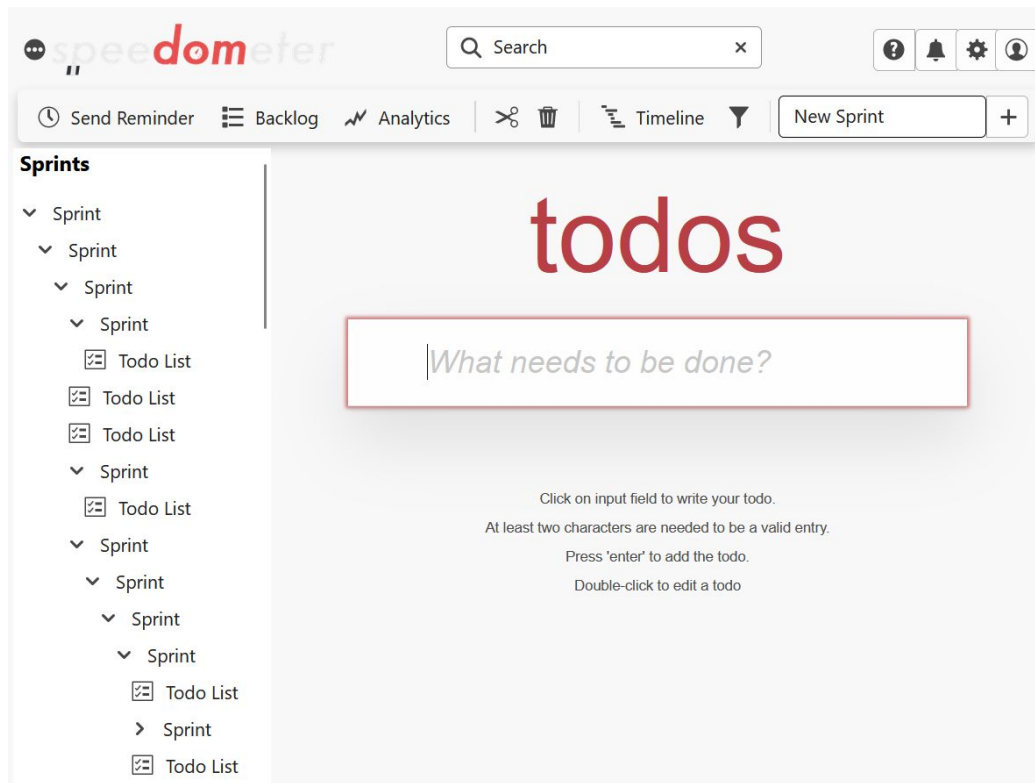
Double-click to edit a todo

Created by Sindre Sorhus

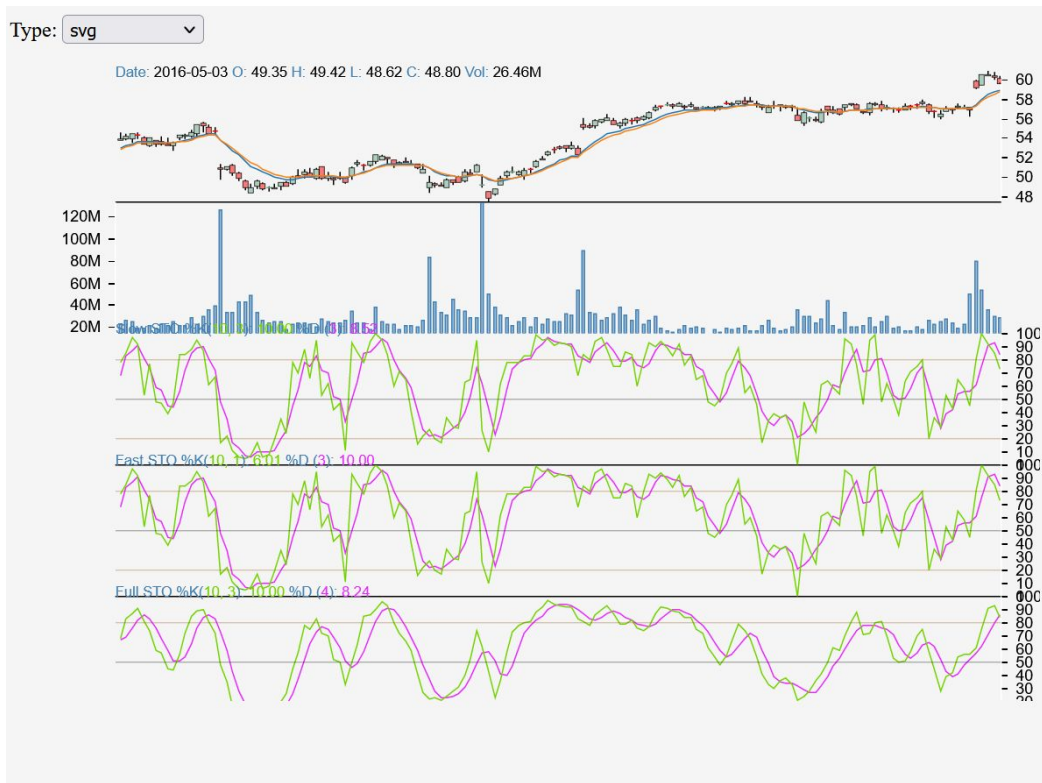
Part of TodoMVC



Todo-MVC (Complex DOM)



Charts



Editor



The Project Gutenberg eBook of Du côté de chez Swann

This ebook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this ebook or online at www.gutenberg.org. If you are not located in the United States, you'll have to check the laws of the country where you are located before using this eBook.

Title: Du côté de chez Swann

Author: Marcel Proust

Release date: May 1, 2001 [eBook #2650]

Most recently updated: August 12, 2021

Language: French

*** START OF THE PROJECT GUTENBERG EBOOK DU CÔTÉ DE CHEZ SWANN ***

MARCEL PROUST

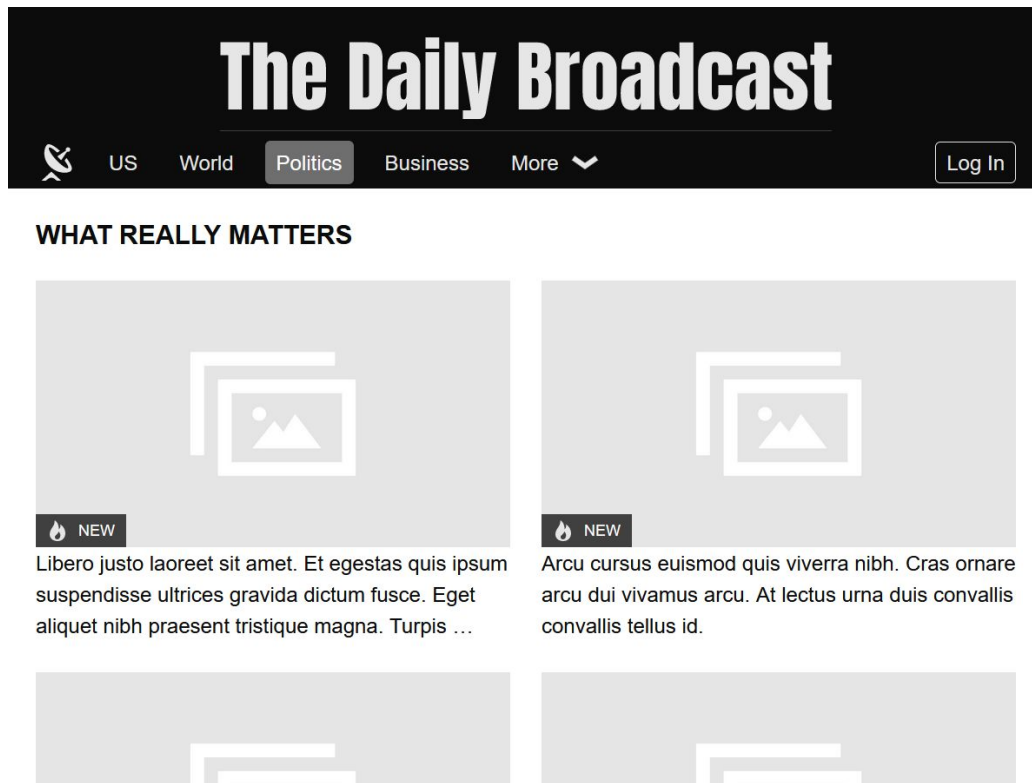
A LA RECHERCHE DU TEMPS PERDU

TOME I

Du Côté de Chez Swann



News Sites



The List

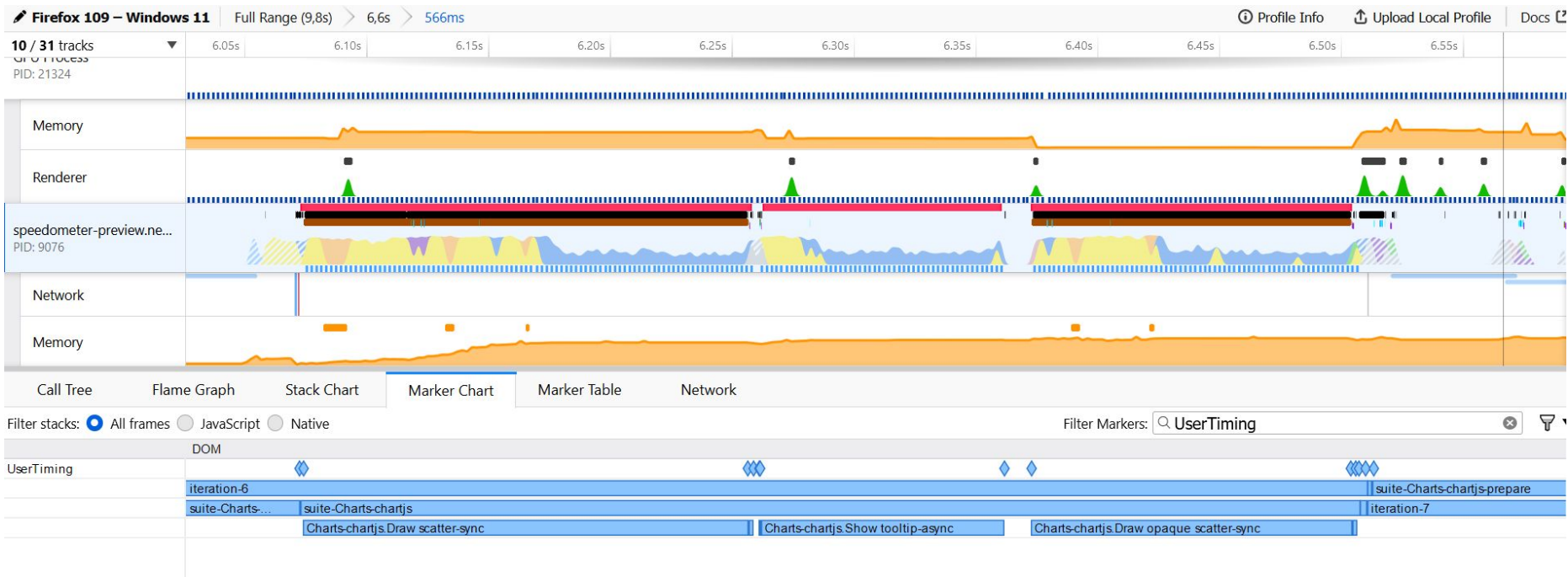
- **TodoMVC**
 - Angular
 - Backbone
 - JavaScript-ES5
 - JavaScript-ES6-Webpack
 - Lit
 - React
 - Preact
 - Svelte
 - Vue
 - Webcomponents
 - jQuery
- **Editors**
 - CodeMirror
 - TipTap
- **Charts**
 - ChartJS
 - Observable-Plot
 - React-Stockcharts
 - Perf-Dashboard
- **News Sites**
 - Next
 - Nuxt



Using Speedometer to Improve Firefox

04

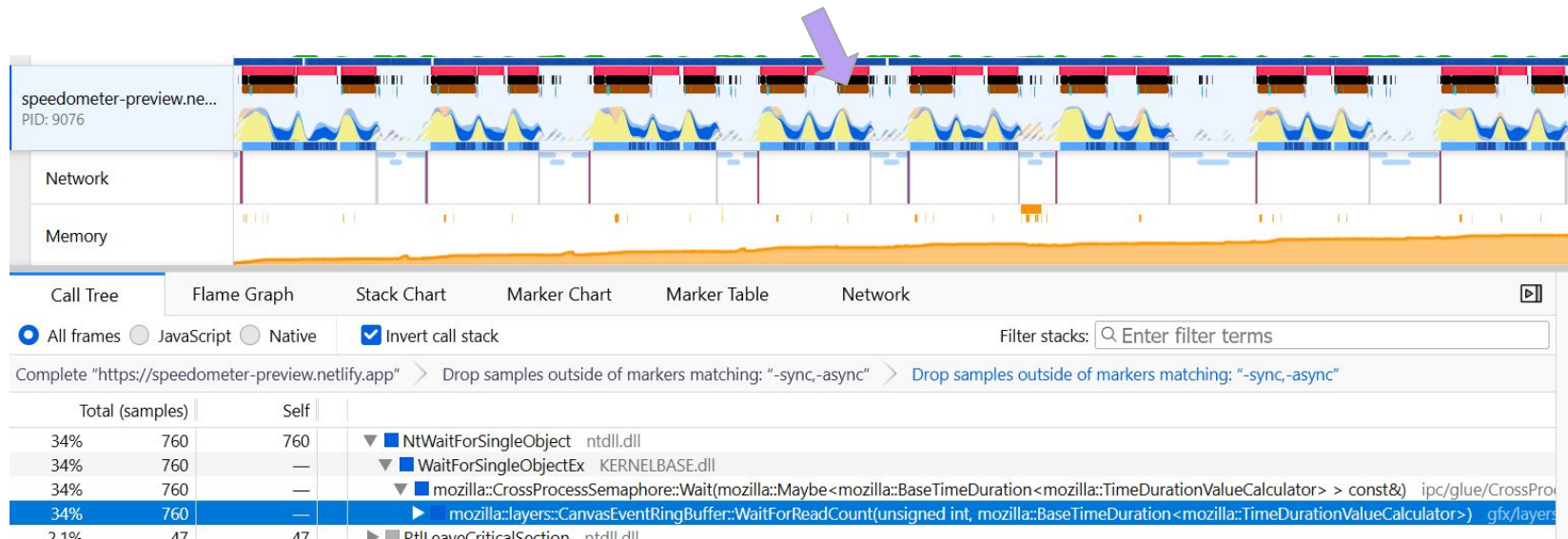
Profiling Speedometer



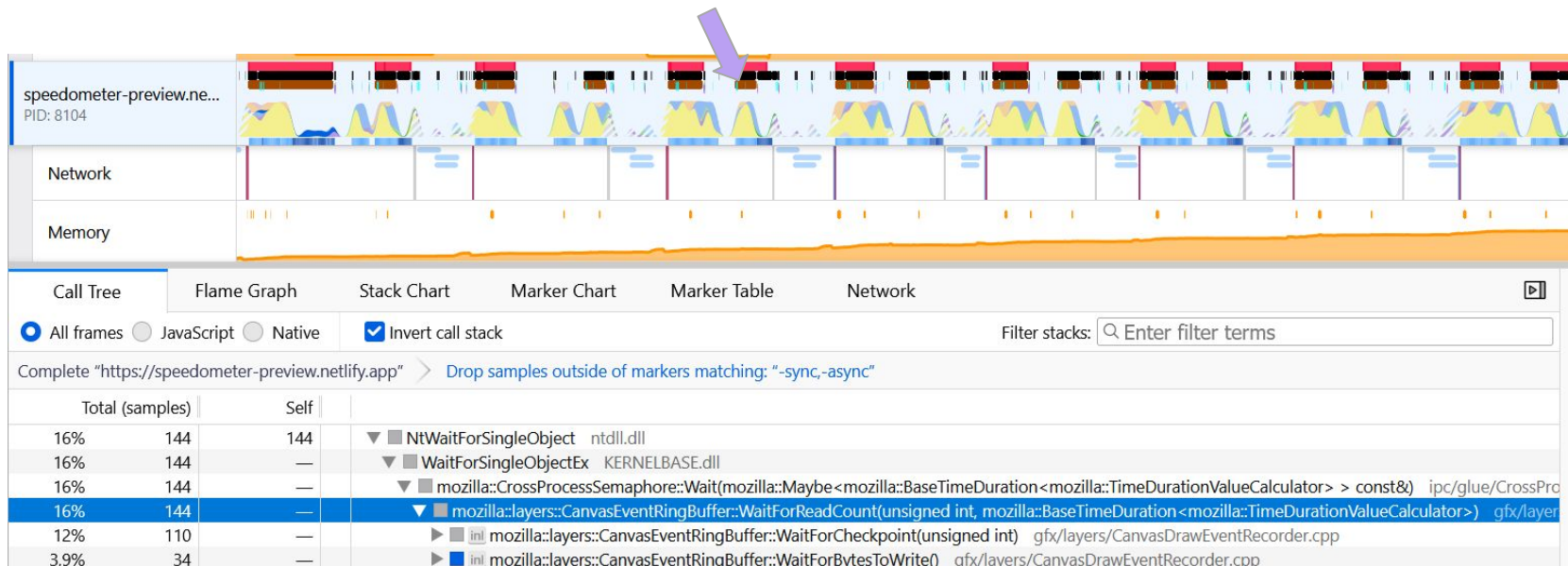
Public 24



Profiling Speedometer

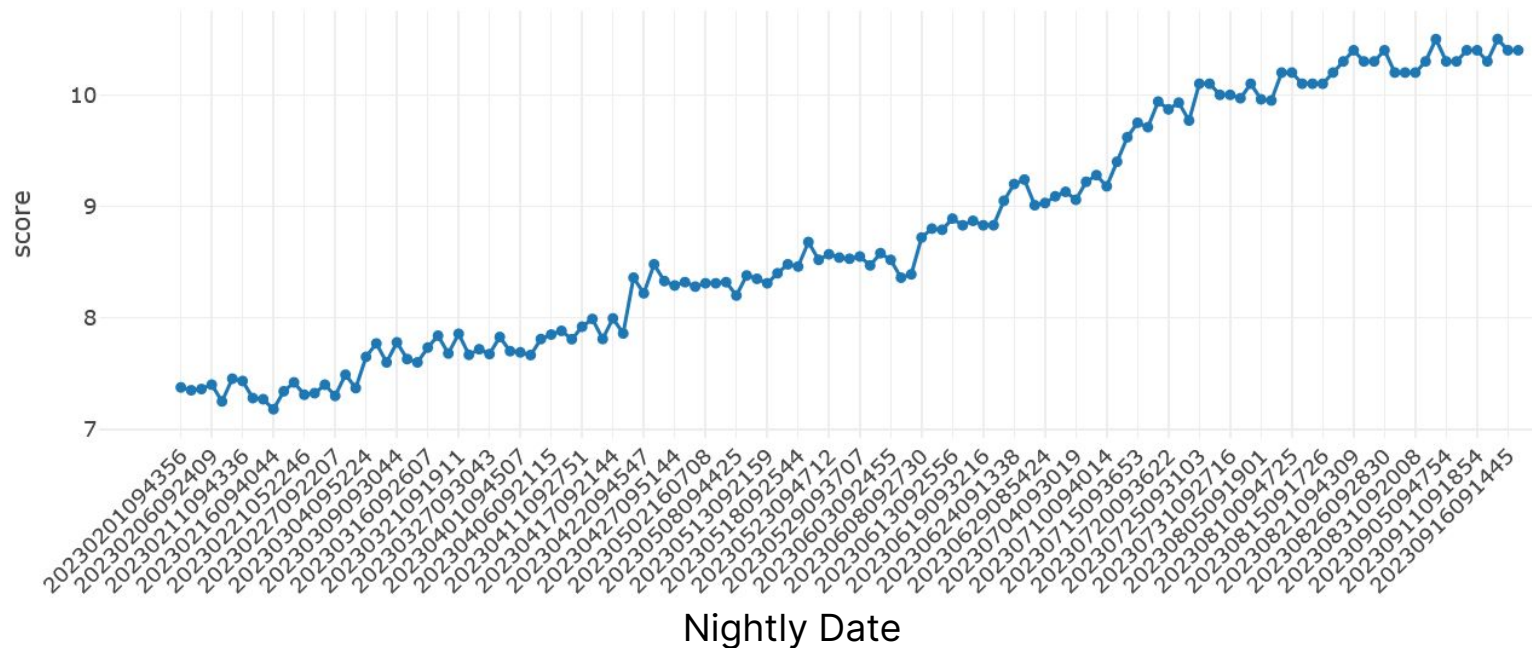


Profiling Speedometer



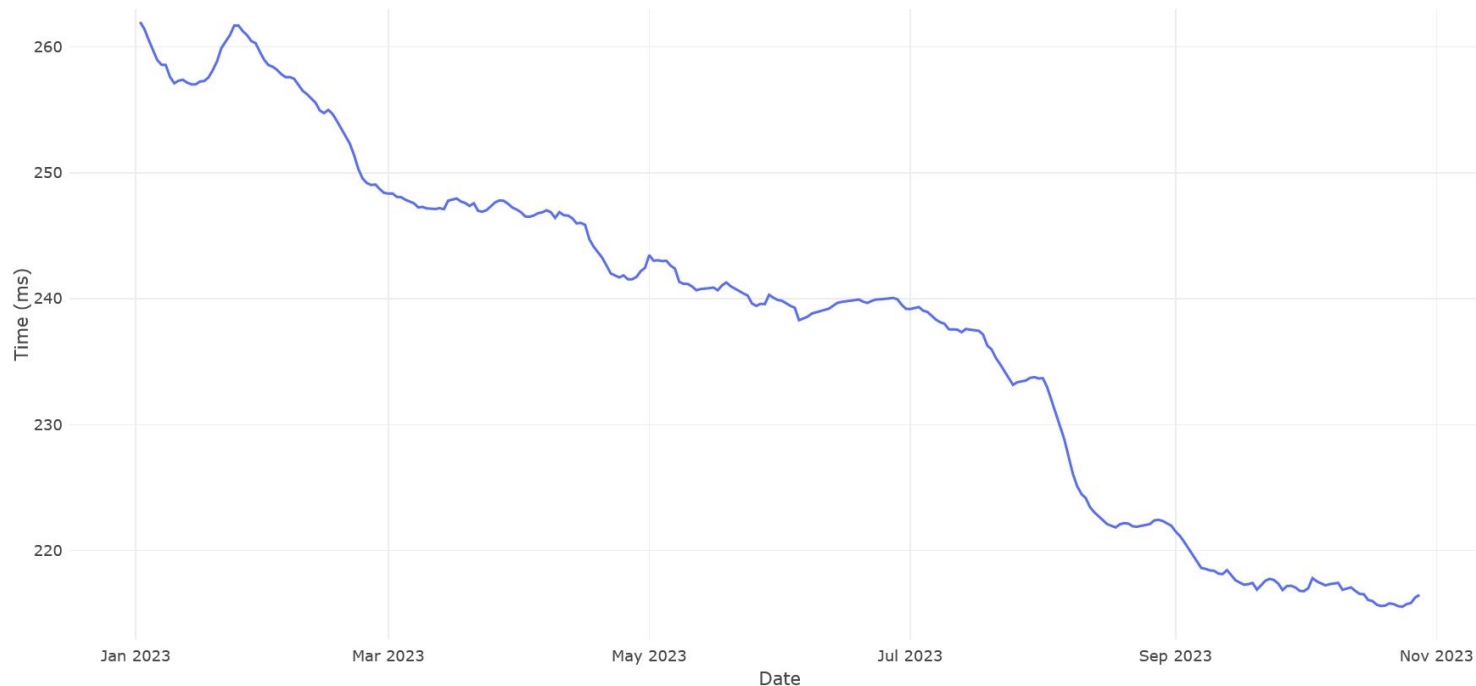
Benchmark Score

Higher is better



RUM

Median First Contentful Paint (ms - 15% improvement) - Lower is better



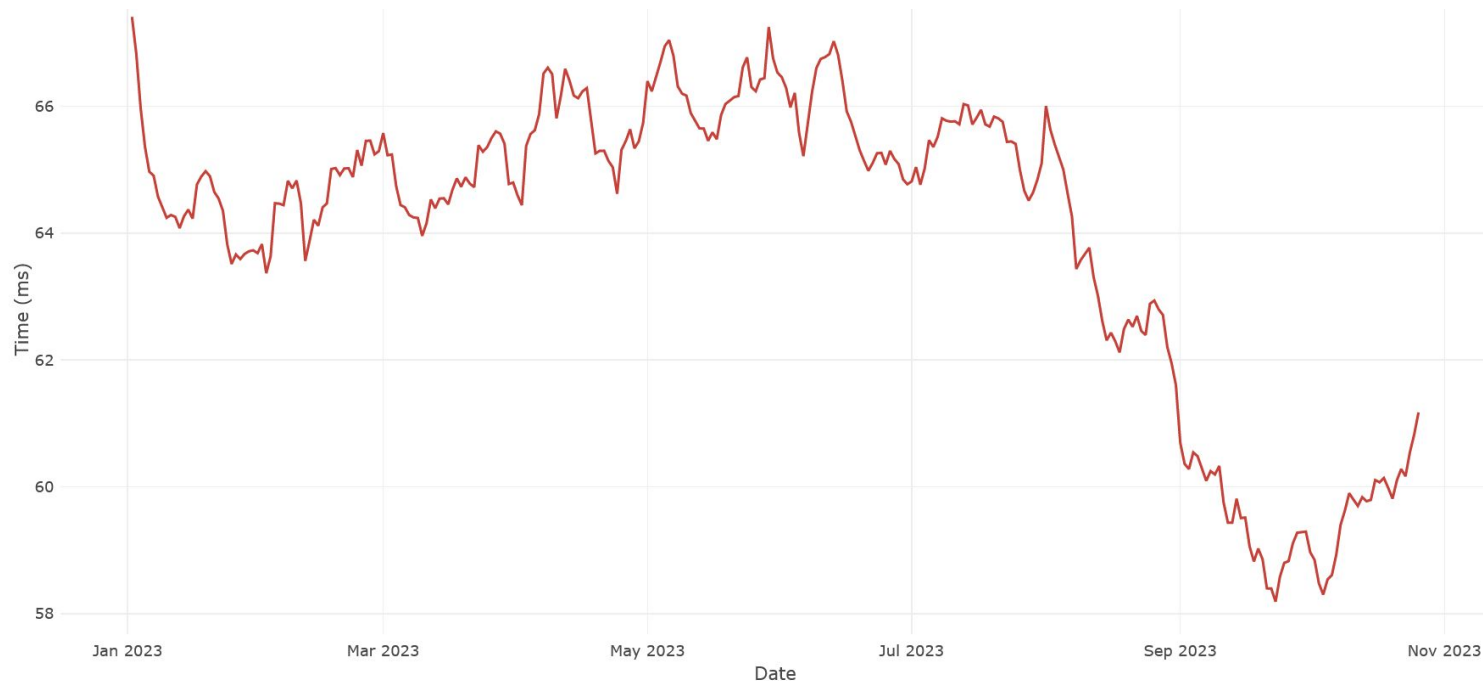
RUM

95th Percentile Pageload JS Execution (ms - 20% improvement) - Lower is better



RUM

95th Percentile - Keypress Present Latency (ms - 10% improvement) - Lower is better



Thank you!