Observing and Optimizing Interaction to Next Paint at Scale

New York UX Speed and Web Performance Group Meetup
June 5, 2024

Speed Team at DDM

Ben Cochran - Product Owner, VP of Engineering

Brian Piccione - Senior Director, Brand Platform Engineering

Russell Brown - Manager, Performance Engineering

Joseph Freeman - Principal Engineer

Tyron Goldschmidt - Software Engineer

Bayan Zhumanova - Senior QA Analyst

Hao Li - QA Software Developer

Dotdash Meredith

America’s Largest Print / Digital Publisher


People


Investopedia


BYRDIE


health


allrecipes

Dotdash Meredith Playbook

Best Content, **Fastest Sites,** Fewest Ads

Agenda

- What is INP?
- How do we *observe* INP?
- How do we *optimize* INP?
- How do we *test* INP?

INP

Interaction to Next Paint

Background

A new method of measuring the performance of the web

Core Web Vitals

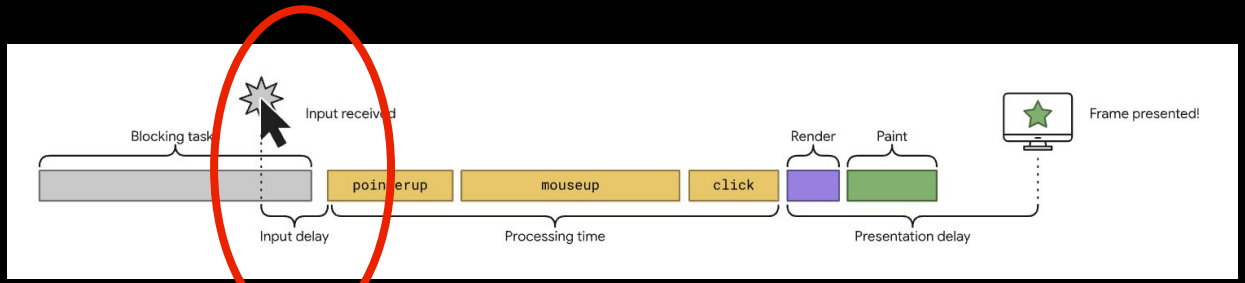
- Web vitals: metrics for user experience
  - FCP, TTI, TBT, Speed Index

- Core Web Vitals: SEO affecting web vitals
  - Loading
    - LCP
  - Stability
    - CLS
  - Interactivity

Dotdash meredith

First things First: FID

- First Input Delay measures first impression of *interactivity*
  - The time the user first interacts to the time the browser is able to respond to the interaction
- Rationale
  - First impressions last
  - Worst interactivity problems often occur during page load

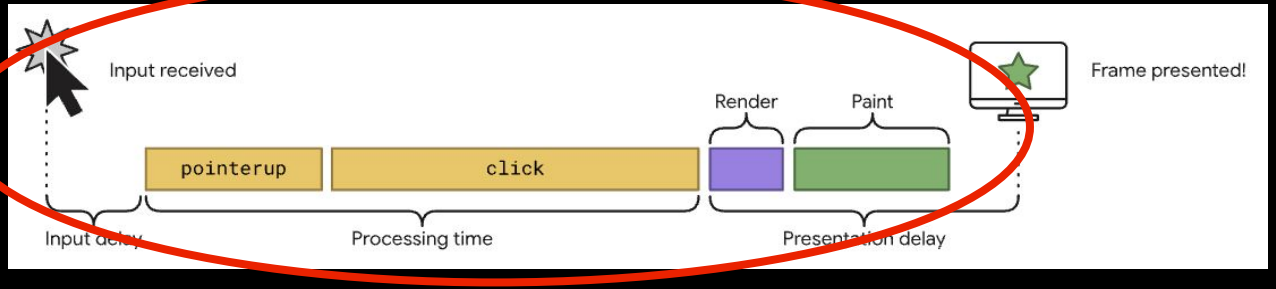


Source: https://web.dev/articles/inp

Dotdash meredith

The New Kid on the Block: INP

- **Interaction to Next Paint replaces FIP as a Core Web Vital**
- **Measures *overall* responsiveness to interactions**
  - The longest duration from an interaction to the browser painting the next frame
- **Rationale**
  - A more complete picture of the interactivity throughout the lifecycle of the page



Source: https://web.dev/articles/inp

Dotdash meredith

FID	INP
Time from first interaction to when main thread next idle	Time from interaction to next visual update
First impression	Overall impression
	
Easier to diagnose problems	Trickier to diagnose problems

Dotdash meredith

How can we observe INP?

Monitoring INP

🎶 *Go chasing waterfalls* 🎶

Dotdash meredith

Chase Waterfalls

- Monitoring infrastructure triggers an alert
- A daily report indicates metrics increase

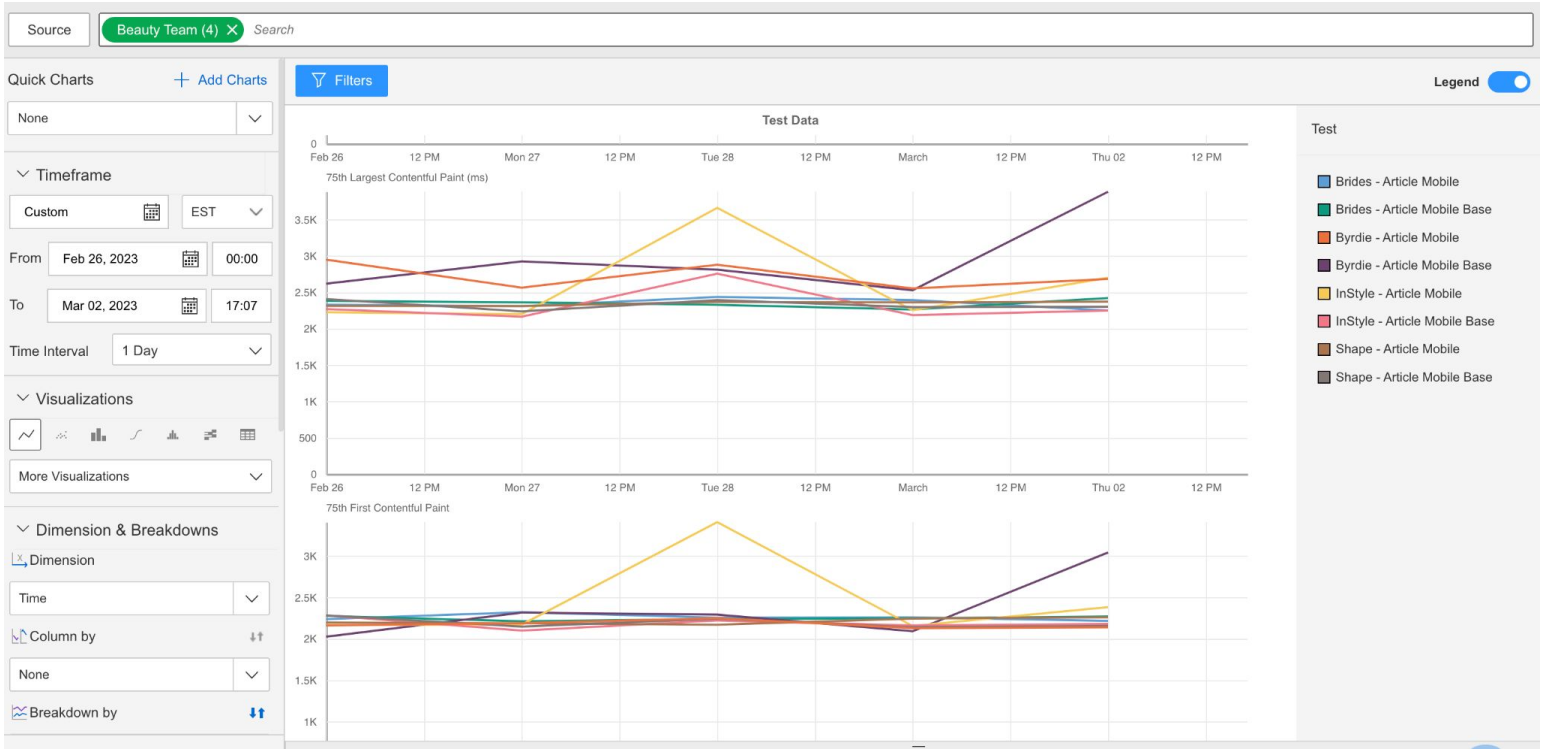
- Check for changes across all infrastructure to rule things out
- Dive into greater detail on the specific metric, comparing types of averages
- Look at individual record charts

- Based on chart graphs, identify problematic threshold(s)
- Understand corresponding metric
- Formulate hypothesis

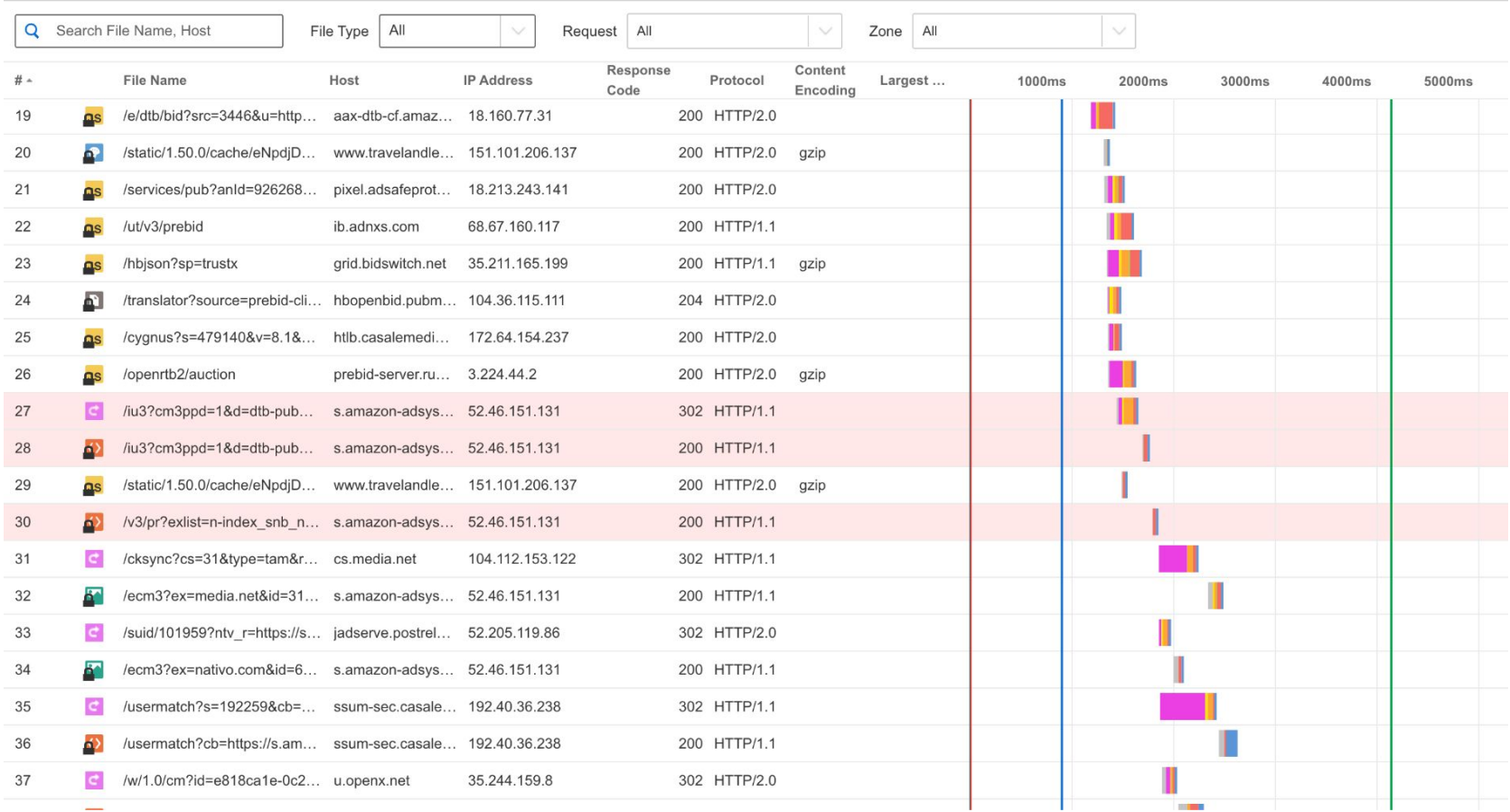
- Inspect the waterfalls for corresponding records
- Identify problem via comparisons or unusual activity

Dotdash meredith

Catchpoint Explorer Chart

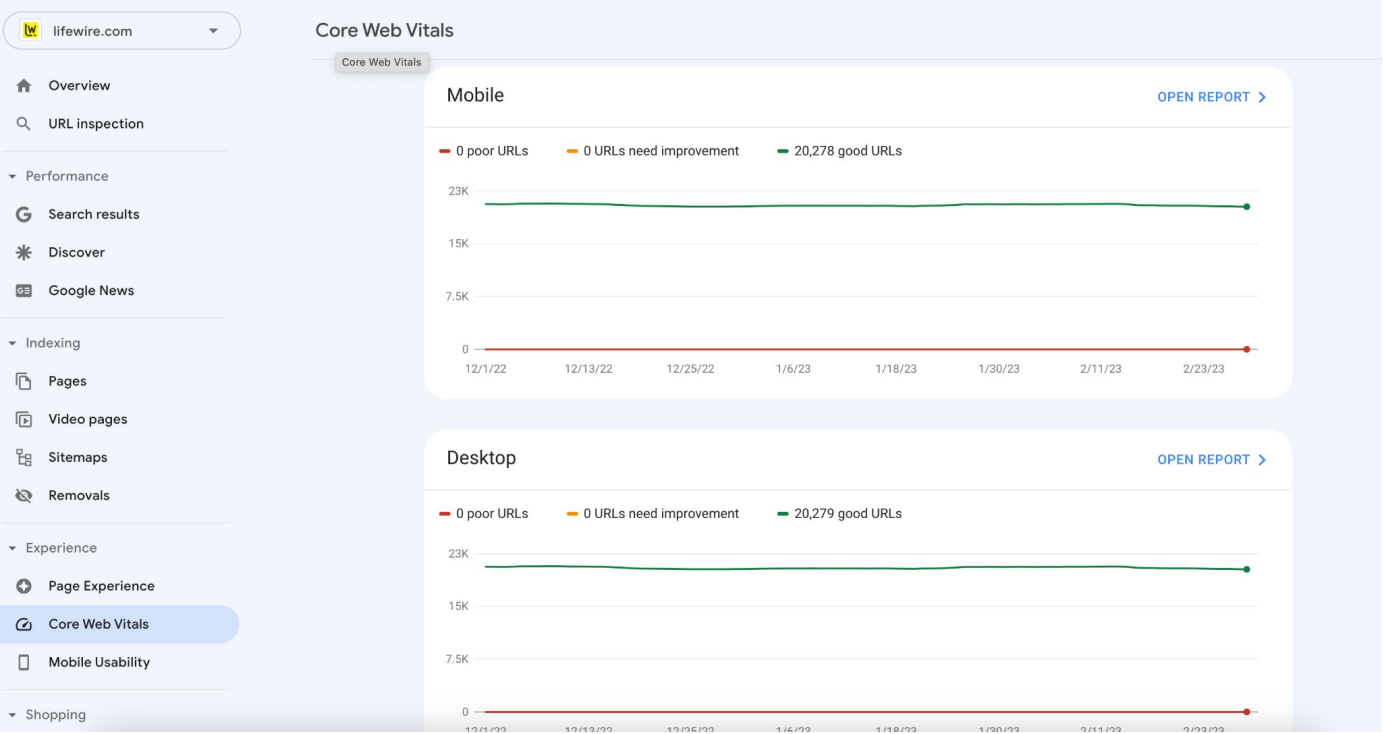


Catchpoint Explorer Record View



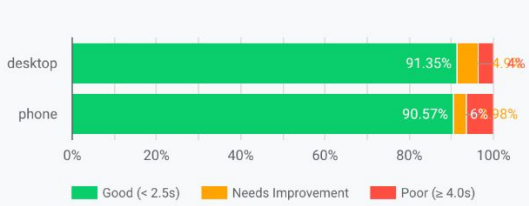
SEO

Google Search Console



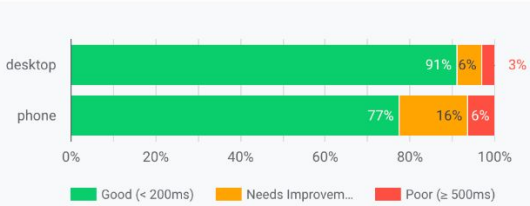
Largest Contentful Paint (LCP)

LCP reports the render time of the largest content element that is visible within the viewport.
web.dev/lcp



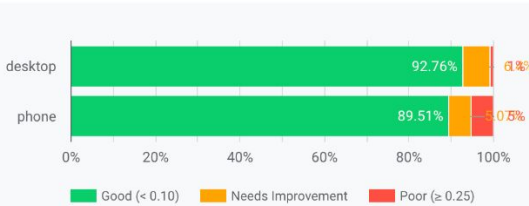
Interaction to Next Paint (INP)

INP reports the overall responsiveness to user interactions, as measured by the longest time from an interaction until the next frame is presented with visual feedback, ignoring outliers.
web.dev/inp

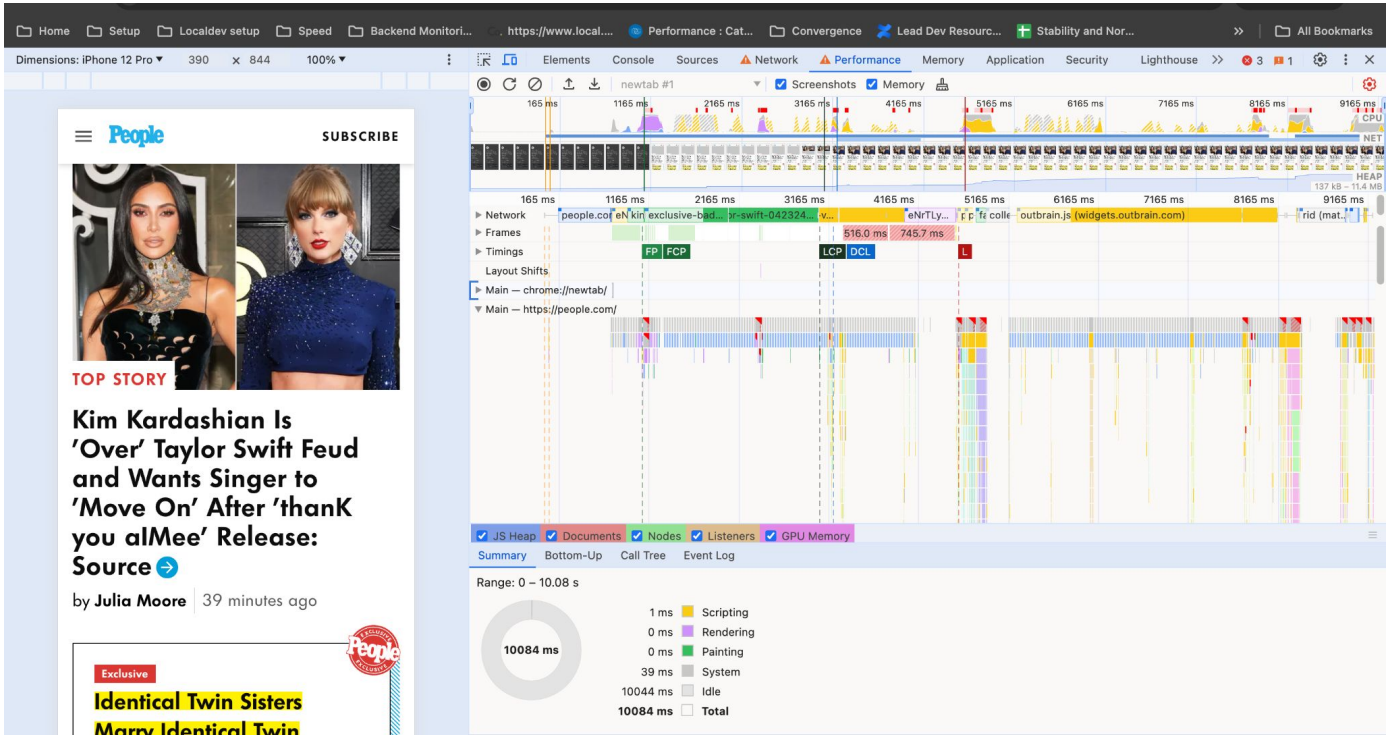


Cumulative Layout Shift (CLS)

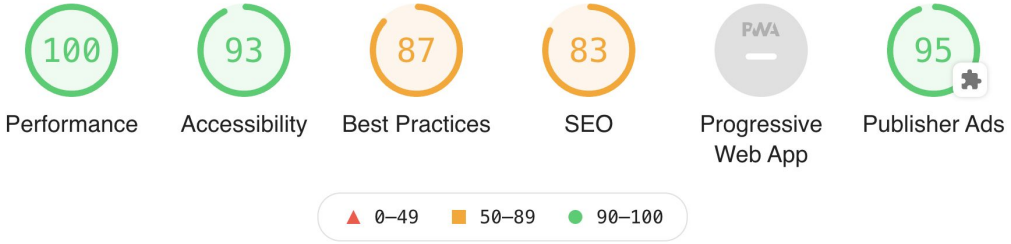
CLS measures the sum total of all individual layout shift scores for every unexpected layout shift that occurs during the entire lifespan of the page.
web.dev/cls



Local Development - Performance Chrome Tab



Engineering - Lighthouse



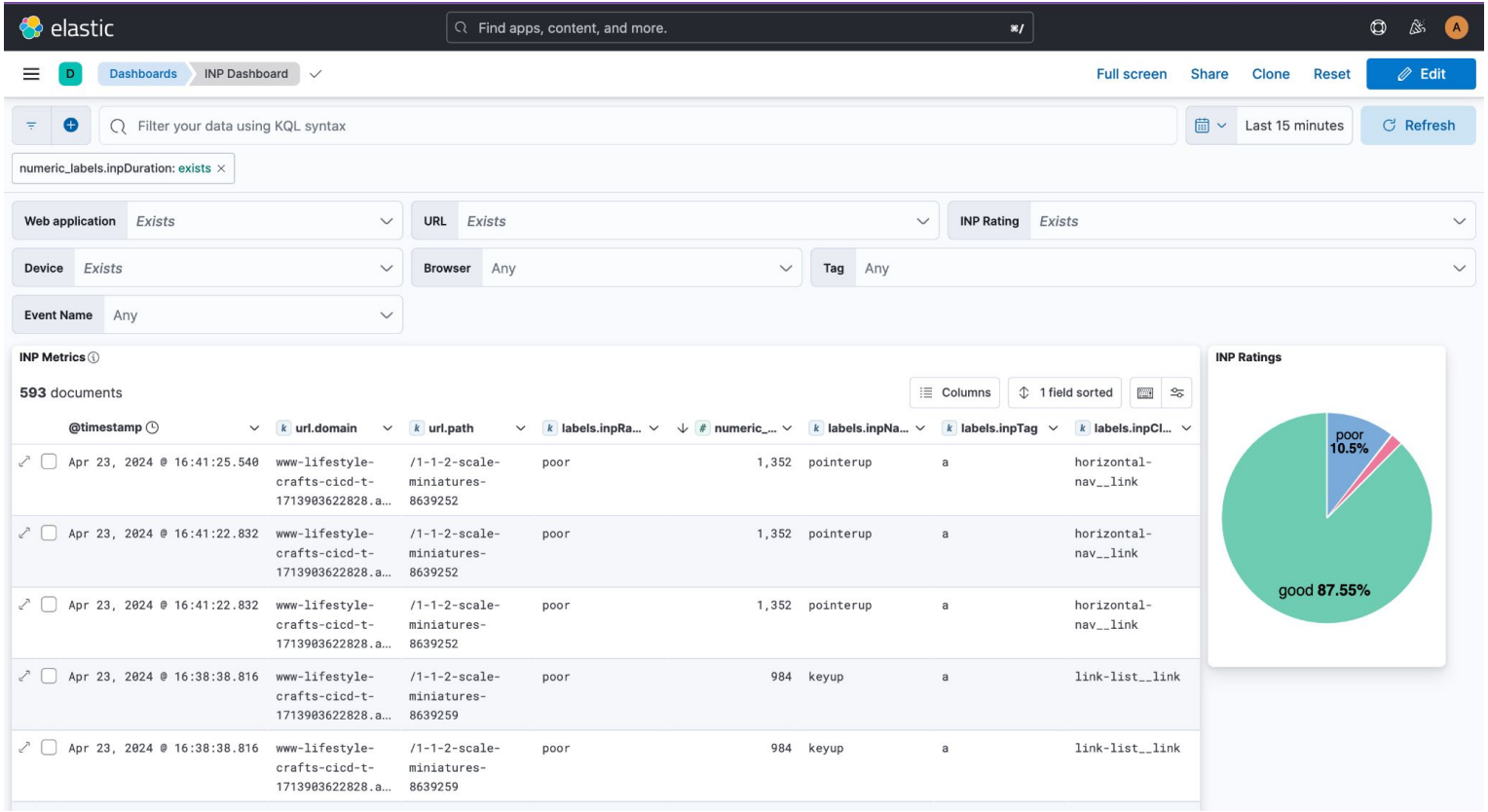
There were issues affecting this run of Lighthouse:

- **No ads were rendered when rendering this page.**
- **No ads were rendered when rendering this page.**

Performance

Metrics

First Contentful Paint	0.3 s	Time to Interactive	0.3 s
Speed Index	0.4 s	Total Blocking Time	0 ms
Largest Contentful Paint	0.5 s	Cumulative Layout Shift	0


elastic
Find apps, content, and more.
Dashboards
INP Dashboard
Full screen
Share
Clone
Reset
Edit
Filter your data using KQL syntax
Last 15 minutes
Refresh
numeric_labels.inpDuration: exists
Web application Exists
URL Exists
INP Rating Exists
Device Exists
Browser Any
Tag Any
Event Name Any
INP Metrics
593 documents
Columns
1 field sorted
@timestamp
url.domain
url.path
labels.inpRa...
numeric_...
labels.inpNa...
labels.inpTag
labels.inpCl...
Apr 23, 2024 @ 16:41:25.540 www-lifestyle-crafts-cicd-t-1713903622828.a... /1-1-2-scale-miniatures-8639252 poor 1,352 pointerup a horizontal-nav__link
Apr 23, 2024 @ 16:41:22.832 www-lifestyle-crafts-cicd-t-1713903622828.a... /1-1-2-scale-miniatures-8639252 poor 1,352 pointerup a horizontal-nav__link
Apr 23, 2024 @ 16:41:22.832 www-lifestyle-crafts-cicd-t-1713903622828.a... /1-1-2-scale-miniatures-8639252 poor 1,352 pointerup a horizontal-nav__link
Apr 23, 2024 @ 16:38:38.816 www-lifestyle-crafts-cicd-t-1713903622828.a... /1-1-2-scale-miniatures-8639259 poor 984 keyup a link-list__link
Apr 23, 2024 @ 16:38:38.816 www-lifestyle-crafts-cicd-t-1713903622828.a... /1-1-2-scale-miniatures-8639259 poor 984 keyup a link-list__link
INP Ratings
poor 10.5%
good 87.55%

How can we improve INP?

Optimizing for INP at Scale

Organization buy-in

- Internal advocacy
- Education & training
- Monitoring & data
- Collaborate with vendors

Optimizing for INP - Strategy

The Key is “interaction to next **paint**”

Asynchronicity

- promises
- async/await,
- setTimeout
- scheduler.yield
- (rAF & rIC)

In-depth

- Re-think classic optimization
- Break down libraries/code
- UI vs background
  - Web workers

Optimizing for INP

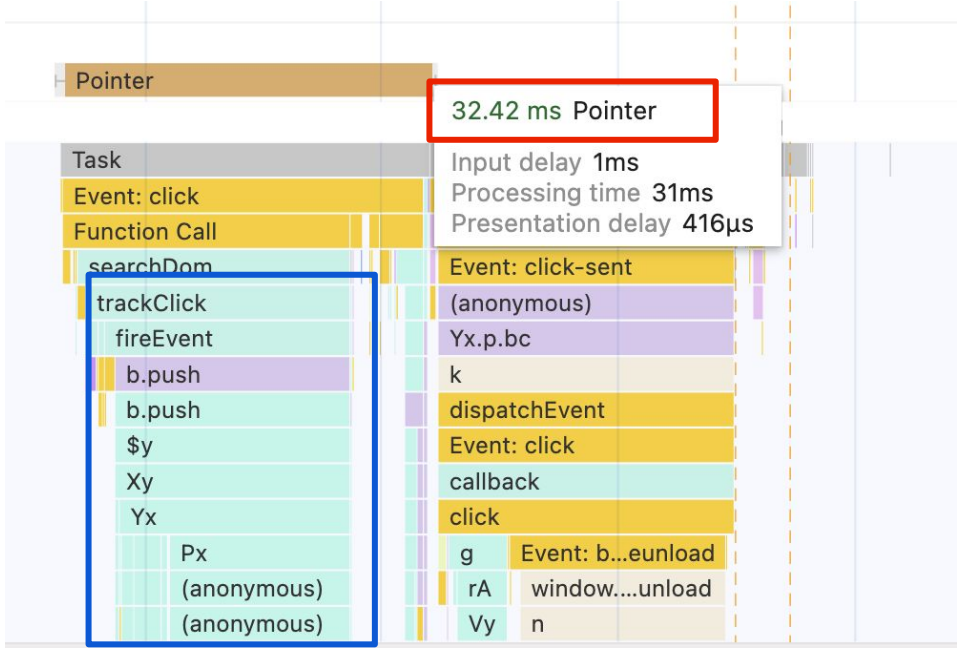
INP utilities

- Callback
- Events

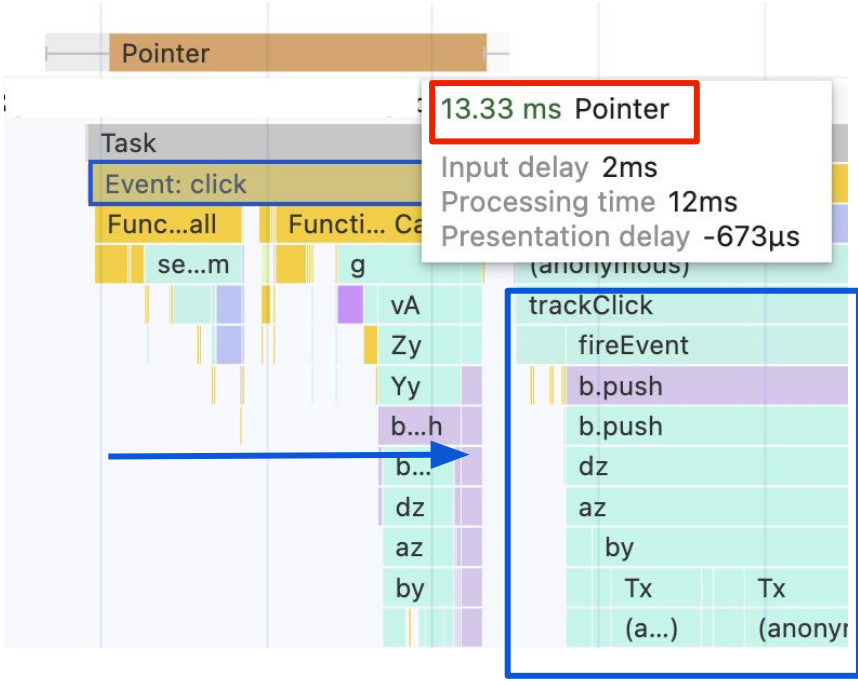
```
addEventHandler('click', (e) => {})

addEventHandler('click',
                util.async((e) => {})
)
```

Example

Before



After



How can we test INP optimization?

Transforming Testing Strategies to Support INP Optimization

Automation Testing to Support INP Optimization

Example of Click Tracking

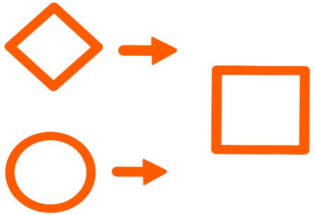
Before (Synchronous)

-> Click -> Track Click (Tracking Date created) -> Fire Event -> Event Handled

After (Asynchronous)

-> Click -> Event handled

-> Async Track Click (Tracking Date created) -> Fire Event



With INP, click tracking events are optimized to execute asynchronously

Automation Testing to Support INP Optimization

Difficulties

1. Asynchronous operations uncertainty

   e.g. expected event is not found
2. Handling of rapid page transitions

   Element loaded / unloaded?

Strategies

1. Add waits/delays for asynchronous operations to complete
2. Monitor network requests and page state

thanks.

Q&A