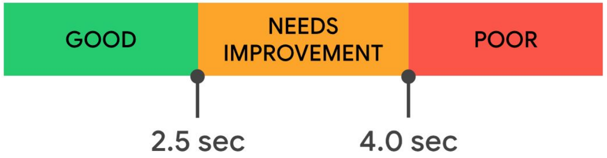
Speculative Loading

Domenic Denicola & Jeremy Roman
Google Chrome @ TPAC 2023

LCP

Largest Contentful Paint



GOOD

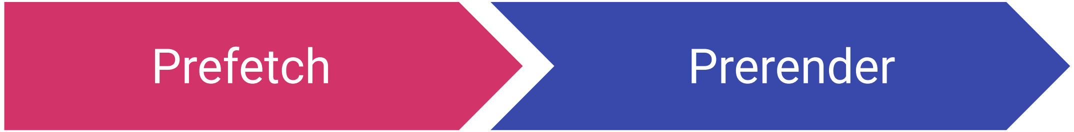
NEEDS IMPROVEMENT

POOR

2.5 sec

4.0 sec

Prefetch & prerender



Prefetch

- Easiest to adopt
- Lowest overhead
- Limited to time downloading the resource

Prerender

- Instant loads (ideally)
- More work to adopt
- Requires longer head start to fully benefit

The technology menu

Speculative loading

Same-site prefetch

Cross-site prefetch

Same-site prerender

Browser UI speculation

URL bar, new tab page, bookmarks, …

Triggers the technologies (in "same-site" mode)

Sites, analytics libraries, etc. need to be aware this can happen

Speculation rules

A declarative JSON syntax for telling the browser what to speculatively load, when

More powerful than <link>s or <a rel="">s

The story so far

Throughout 2022

Ship URL bar prerendering

Ship speculation rules prefetch and prerender

Expand platform coverage

Fix rough edges

2023 to-date

Focus on ecosystem adoption

Ship good DevTools

Respond to API gaps found by partners

Expand URL bar prefetching and prerendering

Ongoing OTs & plans

No-Vary-Search

Document rules

Header-based delivery

Improved document rules heuristics (ML?!)

Let's see some numbers…

web.dev

–200 ms

75th percentile LCP

for prefetch

–700 ms

75th percentile LCP

for prerender

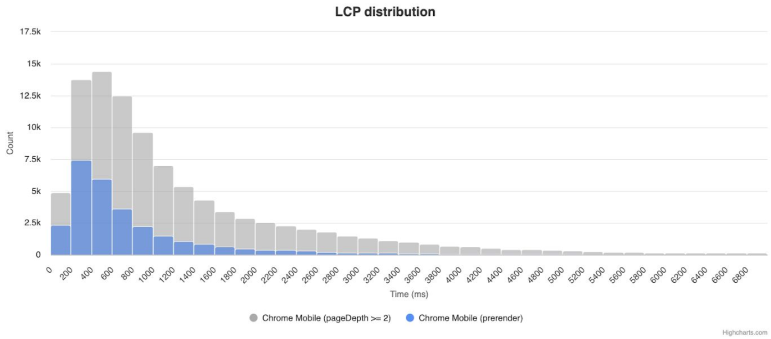
```
{
  "prerender": [{
    "source": "document",
    "where": {
      "and": [
        {"href_matches": "/*\\?*"},
        {"not": {"href_matches": "/patterns/*\\?*"}}
      ]
    }
  }]
}
```

Mobile

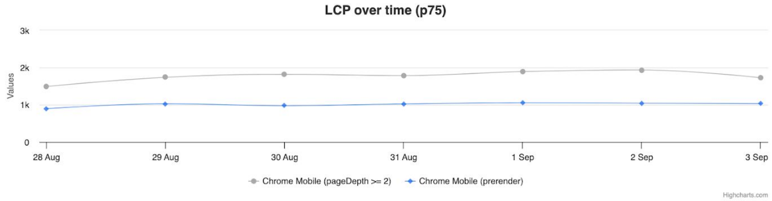
LCP

Largest Contentful Paint *(p75)*

Chrome Mobile (pageDepth >= 2) **1765** Chrome Mobile (prerender) **997**



LCP distribution



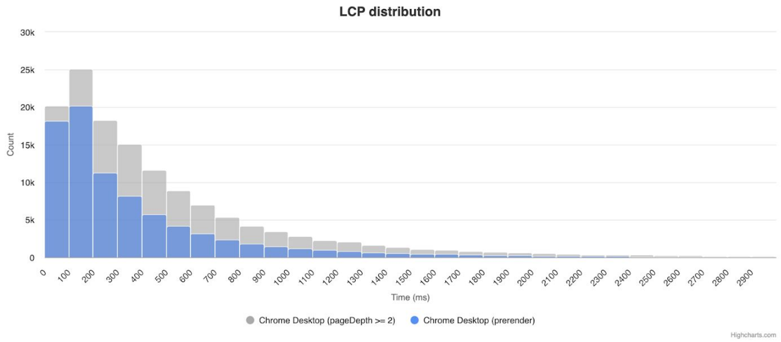
LCP over time (p75)

Desktop

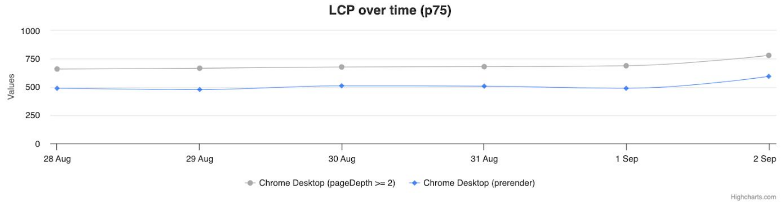
LCP

Largest Contentful Paint *(p75)*

Chrome Desktop (pageDepth >= 2) **680** Chrome Desktop (prerender) **502**



LCP distribution



LCP over time (p75)

NitroPack

−168 ms

75th percentile LCP

prerender vs. <link rel="prefetch">

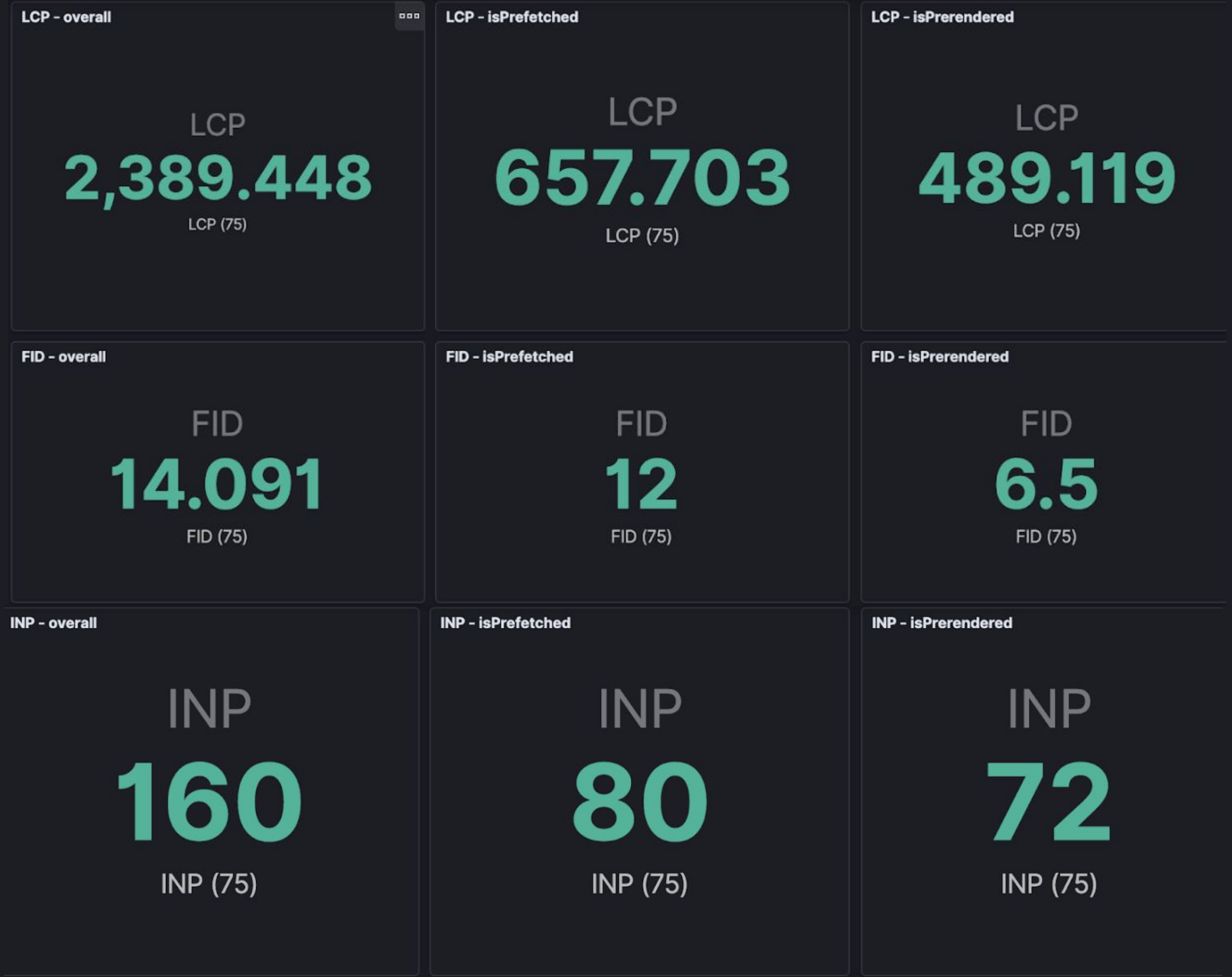
−1900 ms

75th percentile LCP

prerender vs. no speculative loading

```
{
  "prerender": [{
    "source": "list",
    "url": "..."
  }]
}
```

inserted on hover



LCP - overall

LCP

2,389.448

LCP (75)

LCP - isPrefetched

LCP

657.703

LCP (75)

LCP - isPrerendered

LCP

489.119

LCP (75)

FID - overall

FID

14.091

FID (75)

FID - isPrefetched

FID

12

FID (75)

FID - isPrerendered

FID

6.5

FID (75)

INP - overall

INP

160

INP (75)

INP - isPrefetched

INP

80

INP (75)

INP - isPrerendered

INP

72

INP (75)

chrome

−602 ms

75th percentile LCP

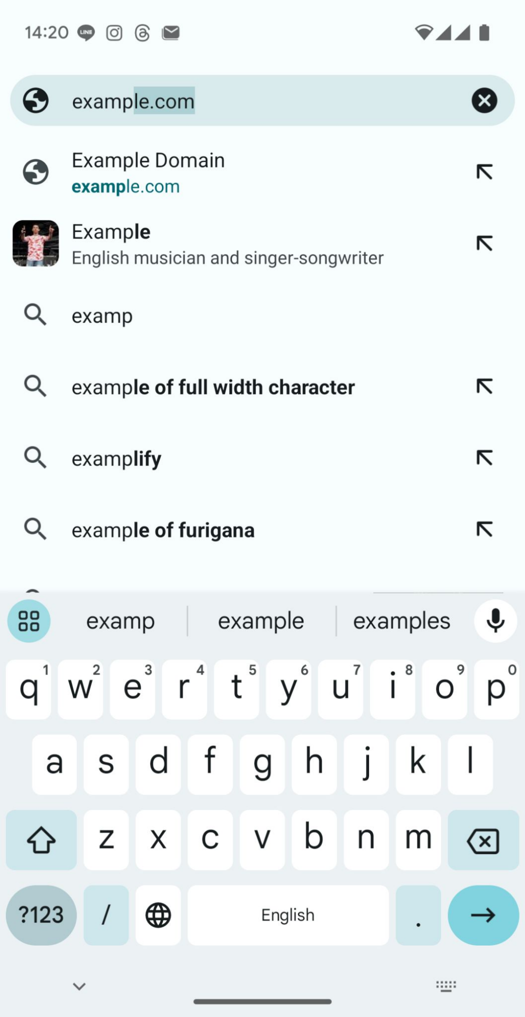
2614 ms → 2012 ms

−9 ms

global average LCP impact

(not just URL bar navigations)

Desktop: -485 ms (2971 ms → 2486 ms)



14:20

example.com

Example Domain
example.com

Example
English musician and singer-songwriter

examp

example **of full width character**

examp**lify**

example **of furigana**

Speculation rules impact

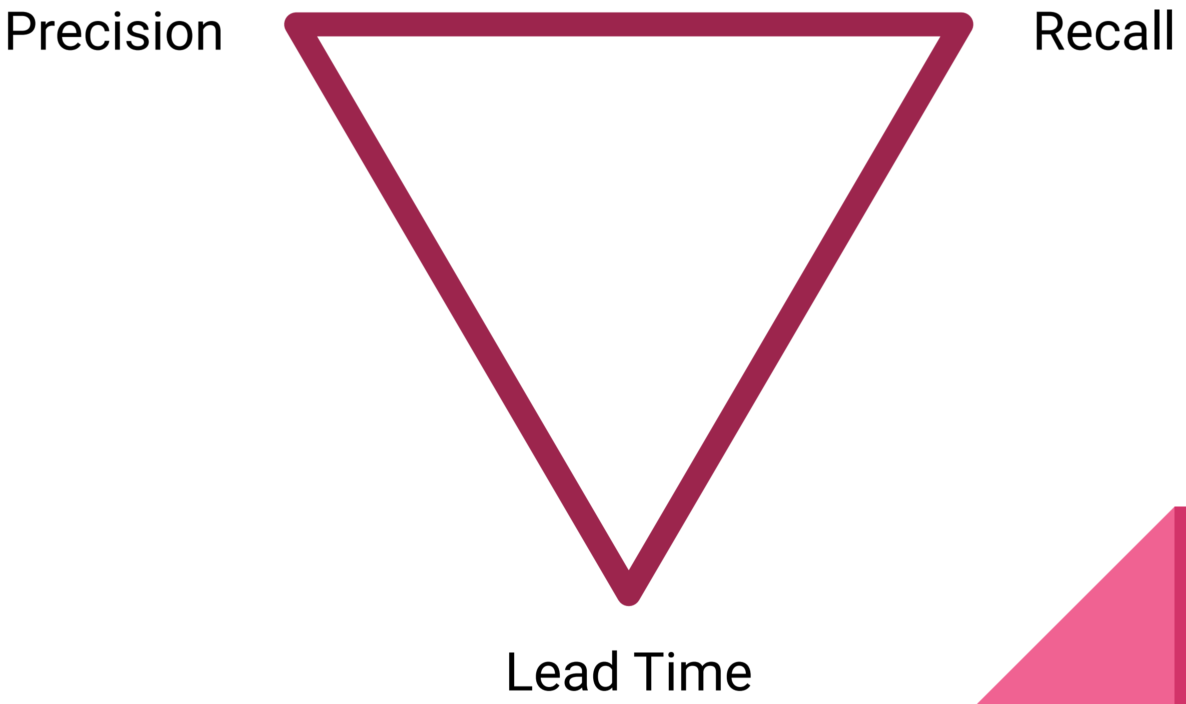
Prefetch

- Android: -278 ms (2879 ms → 2601 ms)
- Desktop: (not enough data)
- -10 ms global average LCP movement

Prerender

- Android: -541 ms (1938 ms → 1397 ms)
- Desktop: -340 ms (1917 ms → 1577 ms)
- No significant global LCP movement (yet!)

The core speculation tradeoff



Precision

Recall

Lead Time

Clearing hurdles to adoption

Prefetch complications

Query Parameters | User State | Injection

Some search parameters do not affect the semantic meaning of the resource (but carry analytics or server processing instructions).

```
No-Vary-Search: key-order, params=("utm_source" "req_pri")
```

Since we wait for prefetch responses we expect to match, to avoid sending a duplicate request it's useful to hint to the browser what policy it should expect.

```
"expects_no_vary_search":
    "key-order, params=(\"utm_source\" \"req_pri\")"
```

Prefetch complications

Query Parameters | **User State** | Injection

Prefetches, if they are not cacheable, are intentionally short-lived to reduce the likelihood of stale state.

However, it's still possible for a user to log in (or out) and see an old prefetched page.

Ideally, these could be discarded when this changes (indicated, e.g., by a particular cookie changing).

API possibilities:

- HTTP Variants proposal (or similar)
- Vary-Cookie: "user_type"
- "discard_if": [{"cookie_changes": "user_type"}]
- Something else?

Prefetch complications

Query Parameters | User State | **Injection**

For cases where the origin server or a reverse proxy is authored or operated by a different group than the document, it can be easier to deploy via a response header.

This allows middleware and service providers to more easily add speculation rules when the site owner wants, without needing to modify document markup.

```
Speculation-Rules: "/speculationrules.json"
```

Identifying prefetched and prerendered pages

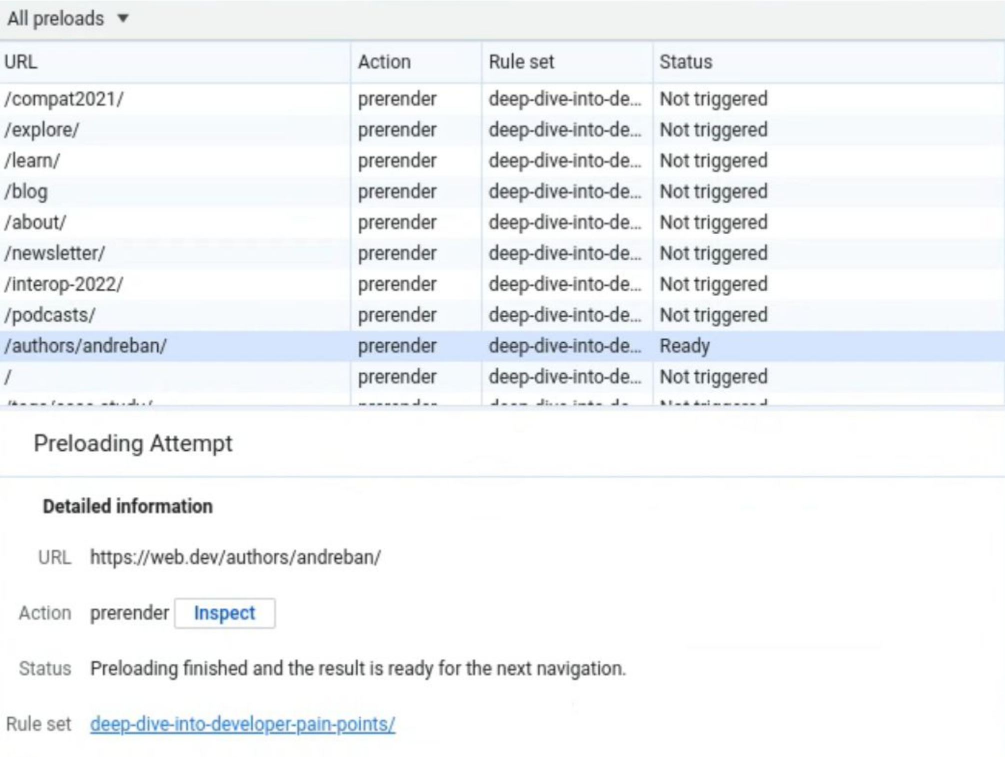
```
performance.getEntriesByType('navigation')[0].deliveryType ===
'navigational-prefetch'
```

```
document.prerendering
document.onprerenderingchange = …
performance.getEntriesByType('navigation')[0].activationStart > 0
```

On the server side:

```
Sec-Purpose: prefetch
Sec-Purpose: prefetch;prerender
```

DevTools support



URL	Action	Rule set	Status
/compat2021/	prerender	deep-dive-into-de...	Not triggered
/explore/	prerender	deep-dive-into-de...	Not triggered
/learn/	prerender	deep-dive-into-de...	Not triggered
/blog	prerender	deep-dive-into-de...	Not triggered
/about/	prerender	deep-dive-into-de...	Not triggered
/newsletter/	prerender	deep-dive-into-de...	Not triggered
/interop-2022/	prerender	deep-dive-into-de...	Not triggered
/podcasts/	prerender	deep-dive-into-de...	Not triggered
/authors/andreban/	prerender	deep-dive-into-de...	Ready
/	prerender	deep-dive-into-de...	Not triggered
/tags/case-study/	prerender	deep-dive-into-de...	Not triggered

Preloading Attempt

Detailed information

URL https://web.dev/authors/andreban/

Action prerender Inspect

Status Preloading finished and the result is ready for the next navigation.

Rule set deep-dive-into-developer-pain-points/

Outro

Call to action

Browser vendors: consider implementing! Your users will thank you!

Web developers and platforms: add speculation rules! You can start small, and you'll see a big boost.

RUM and analytics platforms: make sure you update to account for prerender! It's a reasonable chunk of traffic, and will only grow in the future.

Questions?