RustNL GOSIM 2024 EUROPE

MAKE WEB UI EASY WITH LEPTOS

:%S/EASY/EASIER

BEN WISHOVICH | @BENWIS@HACHYDERM.IO

SOCIALS

My Blog
Mastodon: @benwis@hachyderm.io
LinkedIn

I've been a web developer for four years now, building high performance ecommerce and bespoke websites.

My first PR to Leptos was 15 months ago

Note: Leptos has existed for 16.5 months

SHOUTOUT TO GREG

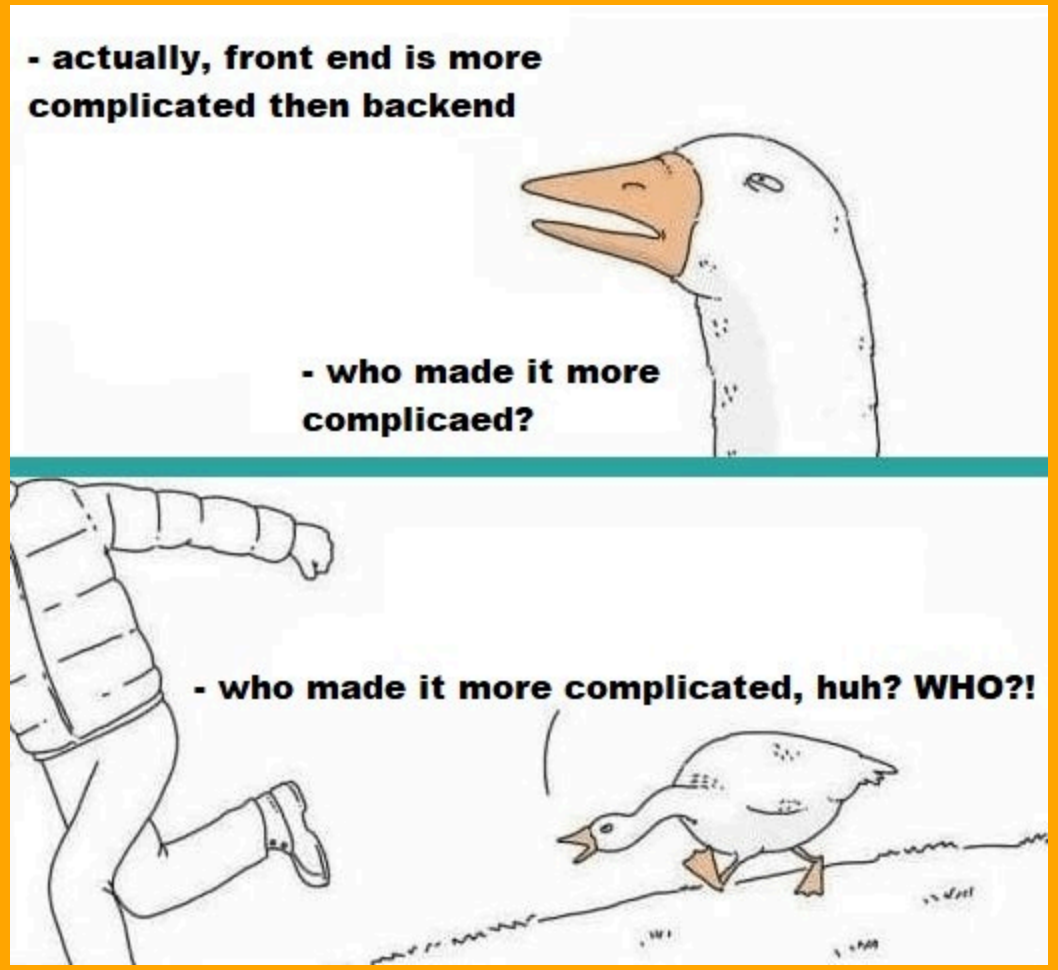
Creator of Leptos

Ben Wishovich | @benwis@hachyderm.io

"FRONTEND IS MORE COMPLICATED THAN BACKEND"

-- DEVELOPERS

Ben Wishovich | @benwis@hachyderm.io


- actually, front end is more complicated then backend
- who made it more complicaed?
- who made it more complicated, huh? WHO?!

Leptos

is a full-stack web framework that lets you leverage the power of Rust and fine-grained reactivity to deliver interactive, stable, and powerful web applications

Ben Wishovich | @benwis@hachyderm.io

LEPTOS DESIGN PRINCIPLES

1. Stability and performance
2. It should feel like a Rust app
3. Compile time, not run time
4. Keep it simple, but customizable
5. HTML-first + progressive enhancement

WEBASSEMBLY FRAMEWORK

- Anything that needs to happen in the browser will be compiled to Webassembly and run by the browser

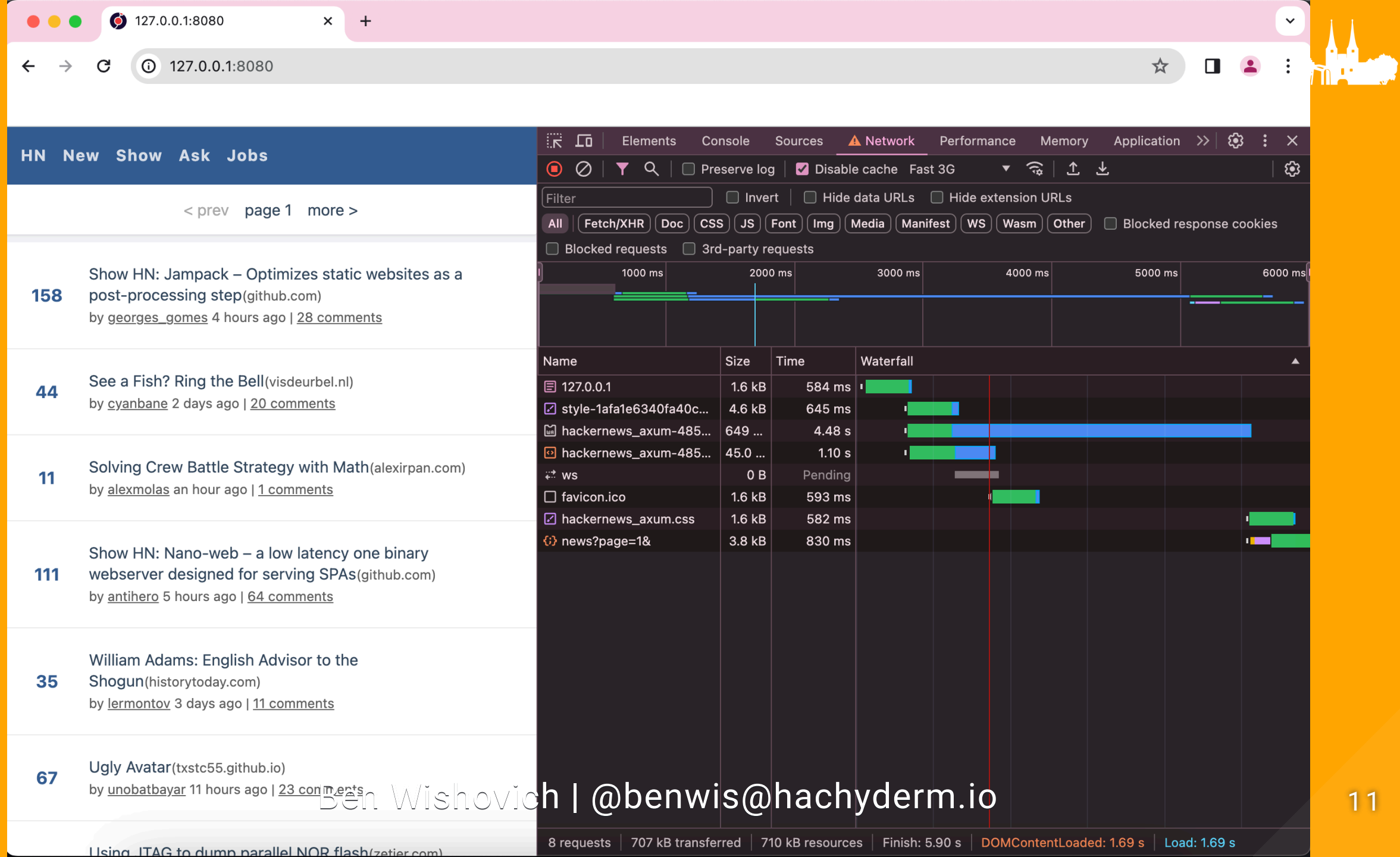
CLIENT SIDE RENDERING

Serve a basic html template and JS file that loads our webassembly, which builds the page and runs async functions

Navigation occurs on the client

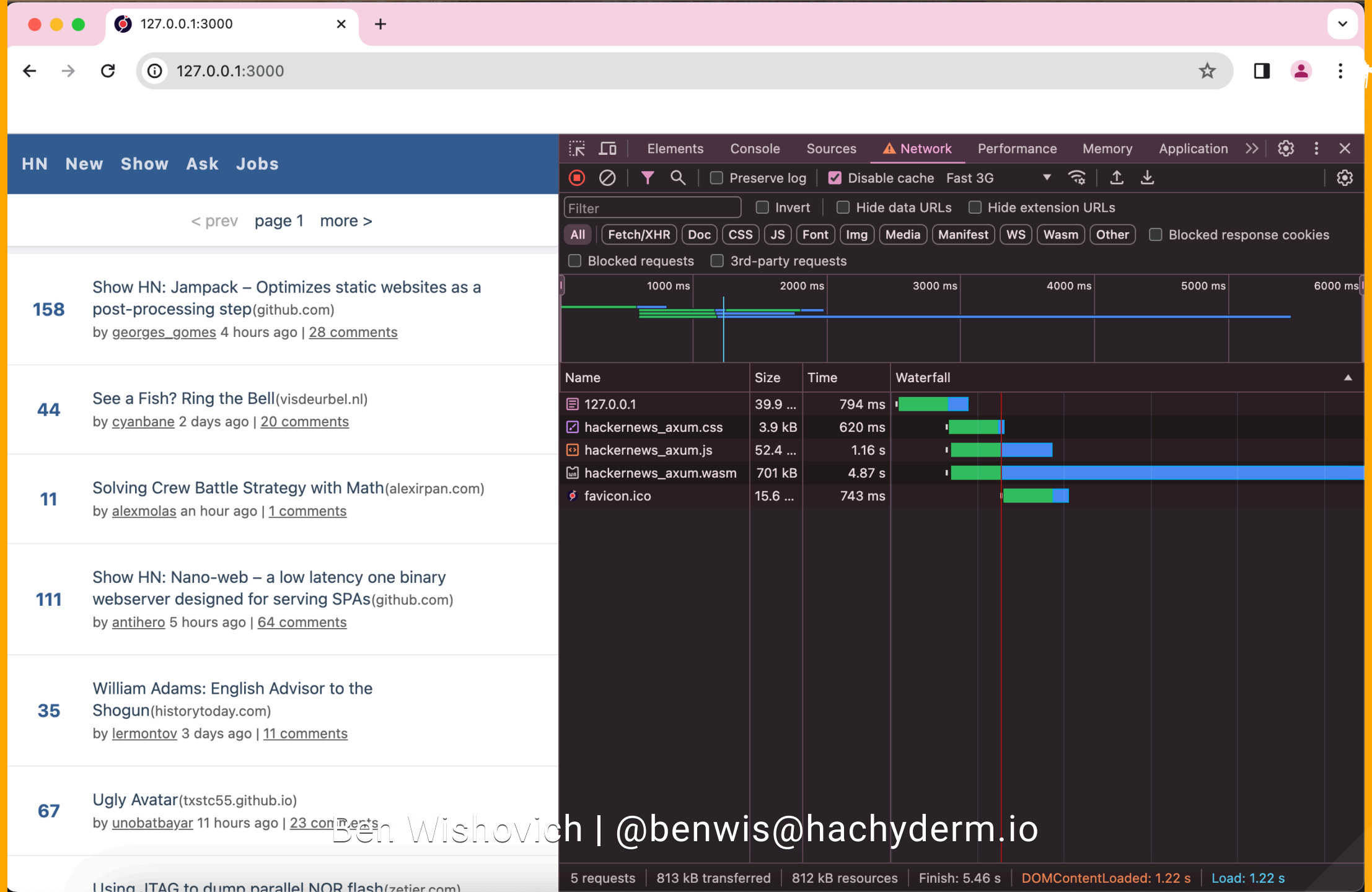
```
<!DOCTYPE html><html><head>
        <script type="module">import init from '/hackernews_axum-485cb1406e801850.js';init('/hackernews_axum-485cb1406e801850_bg.wasm');</script>
        <link rel="stylesheet" href="/style-1afa1e6340fa40c3.css">

<link rel="preload" href="/hackernews_axum-485cb1406e801850_bg.wasm" as="fetch" type="application/wasm" crossorigin="">
<link rel="modulepreload" href="/hackernews_axum-485cb1406e801850.js"></head>
    <!-- deleted Trunk reloading websocket code -->
    <body></body>
</html>
```


127.0.0.1:8080
127.0.0.1:8080
HN New Show Ask Jobs
< prev page 1 more >
158 Show HN: Jampack – Optimizes static websites as a post-processing step(github.com)
by georges_gomes 4 hours ago | 28 comments
44 See a Fish? Ring the Bell(visdeurbel.nl)
by cyanbane 2 days ago | 20 comments
11 Solving Crew Battle Strategy with Math(alexirpan.com)
by alexmolas an hour ago | 1 comments
111 Show HN: Nano-web – a low latency one binary webserver designed for serving SPAs(github.com)
by antihero 5 hours ago | 64 comments
35 William Adams: English Advisor to the Shogun(historytoday.com)
by lermontov 3 days ago | 11 comments
67 Ugly Avatar(txstc55.github.io)
by unobatbayar 11 hours ago | 23 comments
Using JTAG to dump parallel NOR flash(zetier.com)
Elements Console Sources Network Performance Memory Application
Preserve log Disable cache Fast 3G
Filter Invert Hide data URLs Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other Blocked response cookies
Blocked requests 3rd-party requests
1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms
Name Size Time Waterfall
127.0.0.1 1.6 kB 584 ms
style-1afa1e6340fa40c... 4.6 kB 645 ms
hackernews_axum-485... 649 ... 4.48 s
hackernews_axum-485... 45.0 ... 1.10 s
ws 0 B Pending
favicon.ico 1.6 kB 593 ms
hackernews_axum.css 1.6 kB 582 ms
news?page=1& 3.8 kB 830 ms
8 requests 707 kB transferred 710 kB resources Finish: 5.90 s DOMContentLoaded: 1.69 s Load: 1.69 s

SERVER SIDE RENDERING

Dynamically render HTML on the server and send it to the browser, populating it later with results from the server. Navigation typically occurs on the client


127.0.0.1:3000
127.0.0.1:3000
HN New Show Ask Jobs
< prev page 1 more >
158 Show HN: Jampack – Optimizes static websites as a post-processing step(github.com)
by georges_gomes 4 hours ago | 28 comments
44 See a Fish? Ring the Bell(visdeurbel.nl)
by cyanbane 2 days ago | 20 comments
11 Solving Crew Battle Strategy with Math(alexirpan.com)
by alexmolas an hour ago | 1 comments
111 Show HN: Nano-web – a low latency one binary webserver designed for serving SPAs(github.com)
by antihero 5 hours ago | 64 comments
35 William Adams: English Advisor to the Shogun(historytoday.com)
by lermontov 3 days ago | 11 comments
67 Ugly Avatar(txstc55.github.io)
by unobatbayar 11 hours ago | 23 comments
Using JTAG to dump parallel NOR flash(zetier.com)
Elements Console Sources Network Performance Memory Application
Preserve log Disable cache Fast 3G
Filter Invert Hide data URLs Hide extension URLs
All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other Blocked response cookies
Blocked requests 3rd-party requests
1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms
Name Size Time Waterfall
127.0.0.1 39.9 ... 794 ms
hackernews_axum.css 3.9 kB 620 ms
hackernews_axum.js 52.4 ... 1.16 s
hackernews_axum.wasm 701 kB 4.87 s
favicon.ico 15.6 ... 743 ms
5 requests 813 kB transferred 812 kB resources Finish: 5.46 s DOMContentLoaded: 1.22 s Load: 1.22 s

LET'S BUILD

A BASIC COUNTER APP WITH LEPTOS

Ben Wishovich | @benwis@hachyderm.io

STARTING OFF

- Even More Basic Starter

CONGRATULATIONS!

We built the Leptos Axum Starter

Ben Wishovich | @benwis@hachyderm.io

TEMPTING YOU

MYTHS OF LEPTOS AND WEBASSEMBLY

1. The bundle size is too big
2. The startup time is too slow
3. It's limited by a lack of direct DOM access for Webassembly
4. Compiling takes too long

PERFORMANCE

JS FRAMEWORK BENCH

Duration in milliseconds ± 95% confidence interval (Slowdown = Duration / Fastest)

Name Duration for...	vanillajs	svelte-v5.0.0-next.28	solid-v1.8.0	leptos-0.7-sledge-hammer-v0.7.0	leptos-0.7-v0.7.0	leptos-v0.6.9	vue-v3.4.3	angular-ngfor-v17.0.2	react-hooks-v18.2.0	alpine-v3.12.0
Implementation notes	772			1139	1139	1139				1139
Implementation link	code	code	code	code	code	code	code	code	code	code
create rows creating 1,000 rows (5 warmup runs).	33.9 ±0.2 (1.00)	35.3 ±0.2 (1.04)	35.4 ±0.1 (1.04)	36.8 ±0.1 (1.09)	39.8 ±0.3 (1.17)	42.7 ±0.1 (1.26)	41.3 ±0.5 (1.22)	42.6 ±0.2 (1.26)	43.1 ±0.4 (1.27)	100.8 ±0.3 (2.97)
replace all rows updating all 1,000 rows (5 warmup runs).	37.6 ±0.2 (1.00)	40.0 ±0.2 (1.06)	40.1 ±0.4 (1.07)	41.0 ±0.1 (1.09)	45.8 ±0.1 (1.22)	47.9 ±0.3 (1.27)	47.7 ±0.6 (1.27)	50.4 ±0.1 (1.34)	50.8 ±0.3 (1.35)	122.5 ±0.5 (3.26)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 4 x CPU slowdown.	16.1 ±0.3 (1.00)	16.0 ±0.1 (1.00)	16.1 ±0.1 (1.01)	16.3 ±0.4 (1.02)	16.8 ±0.3 (1.05)	16.6 ±0.2 (1.04)	19.5 ±0.2 (1.22)	16.7 ±0.2 (1.05)	20.4 ±0.5 (1.28)	21.4 ±0.2 (1.34)
select row highlighting a selected row. (5 warmup runs). 4 x CPU slowdown.	5.4 ±0.9 (1.04)	5.7 ±0.9 (1.09)	6.7 ±1.1 (1.28)	6.8 ±1.1 (1.31)	7.1 ±1.5 (1.36)	6.2 ±1.3 (1.19)	5.4 ±1.0 (1.02)	5.2 ±1.0 (1.00)	5.8 ±0.6 (1.12)	33.5 ±0.8 (6.41)
swap rows swap 2 rows for table with 1,000 rows. (5 warmup runs). 4 x CPU slowdown.	18.6 ±0.3 (1.00)	19.9 ±0.5 (1.07)	19.7 ±0.3 (1.06)	19.4 ±0.2 (1.04)	19.1 ±0.2 (1.03)	19.3 ±0.2 (1.04)	21.0 ±0.4 (1.13)	165.2 ±1.4 (8.89)	159.4 ±0.8 (8.58)	33.8 ±0.4 (1.82)
remove row removing one row. (5 warmup runs). 2 x CPU slowdown.	15.5 ±0.1 (1.00)	16.0 ±0.1 (1.03)	15.9 ±0.1 (1.03)	16.2 ±0.2 (1.04)	16.0 ±0.2 (1.03)	16.1 ±0.1 (1.04)	19.3 ±0.1 (1.25)	16.6 ±0.2 (1.07)	18.0 ±0.1 (1.16)	25.2 ±0.1 (1.63)
create many rows creating 10,000 rows. (5 warmup runs with 1k rows).	364.0 ±1.0 (1.00)	373.2 ±1.8 (1.03)	376.2 ±1.8 (1.03)	381.1 ±3.6 (1.05)	422.9 ±2.0 (1.16)	454.0 ±3.2 (1.25)	432.4 ±2.2 (1.19)	441.3 ±2.7 (1.21)	587.0 ±4.8 (1.61)	906.1 ±3.6 (2.49)
append rows to large table appending 1,000 to a table of 1,000 rows.	39.2 ±0.3 (1.00)	41.6 ±0.2 (1.06)	41.3 ±0.2 (1.05)	41.6 ±0.3 (1.06)	46.5 ±0.3 (1.19)	48.2 ±0.5 (1.23)	47.2 ±0.2 (1.20)	48.0 ±0.4 (1.22)	49.8 ±0.1 (1.27)	117.6 ±0.5 (3.00)
clear rows clearing a table with 1,000 rows. 4 x CPU slowdown. (5 warmup runs).	12.3 ±0.3 (1.00)	13.5 ±0.3 (1.10)	14.1 ±0.2 (1.15)	16.0 ±0.2 (1.31)	16.6 ±0.2 (1.35)	15.9 ±0.2 (1.30)	15.5 ±0.3 (1.26)	26.7 ±0.2 (2.17)	25.3 ±0.3 (2.06)	47.0 ±0.9 (3.83)
weighted geometric mean of all factors in the table compare: Green means significantly faster, red significantly slower	1.00	1.05	1.06	1.09	1.16	1.19	1.21	1.34	1.45	2.56
	compare	compare	compare	compare	compare	compare	compare	compare	compare	compare

BEN'S BLOG PERFORMANCE TEST

- Wrote my blog in two different web frameworks as similar as possible, Remix and Leptos
- Measure how long it takes to serve the home page using each framework, under differing levels of load

DETAILS

- Home page fetches 3 most recent posts from a sqlite database, displays post metadata
- Html/CSS/Logic as functionally similar as possible

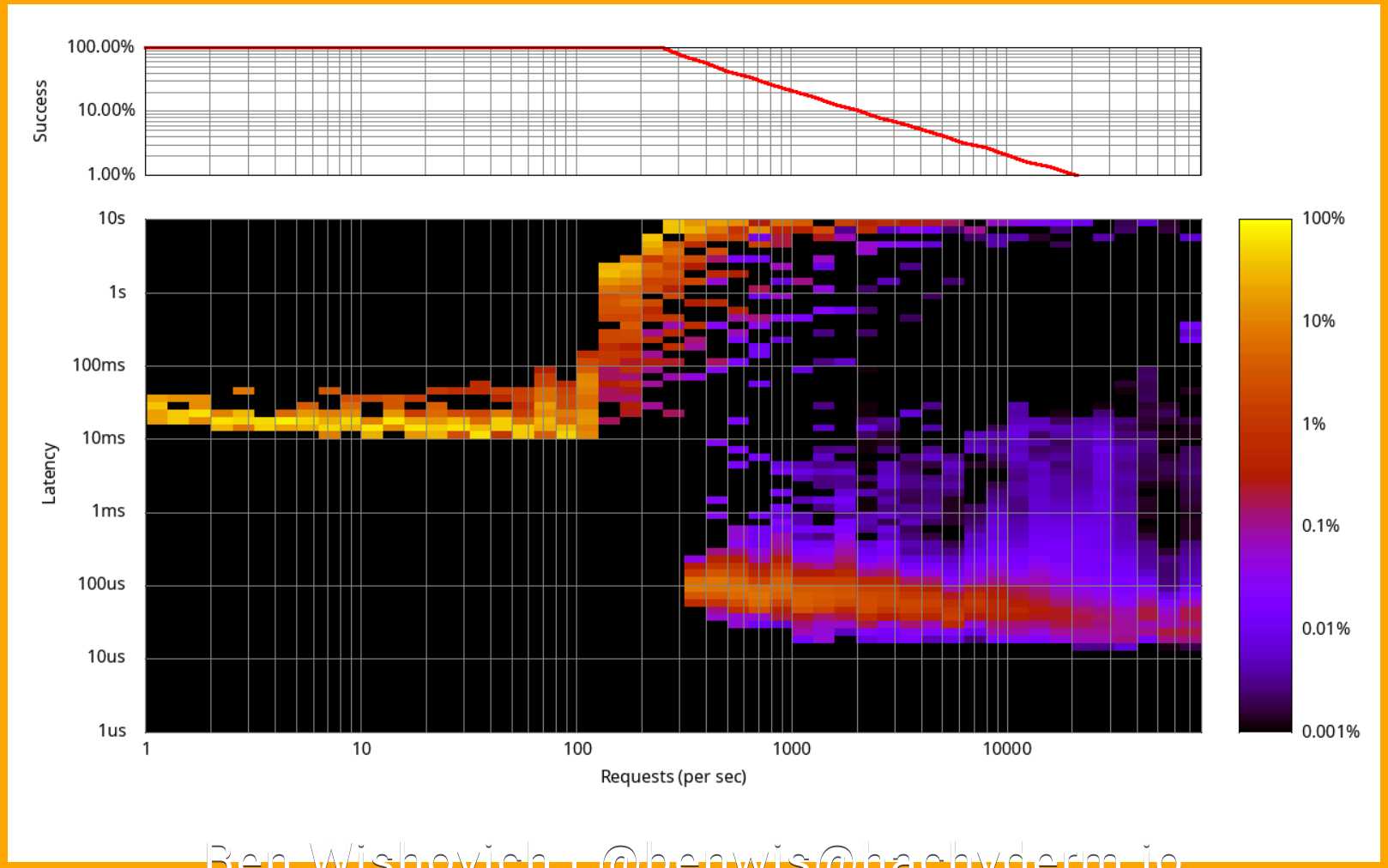
TEST HARDWARE

Web apps run on a Digital Ocean VM with:

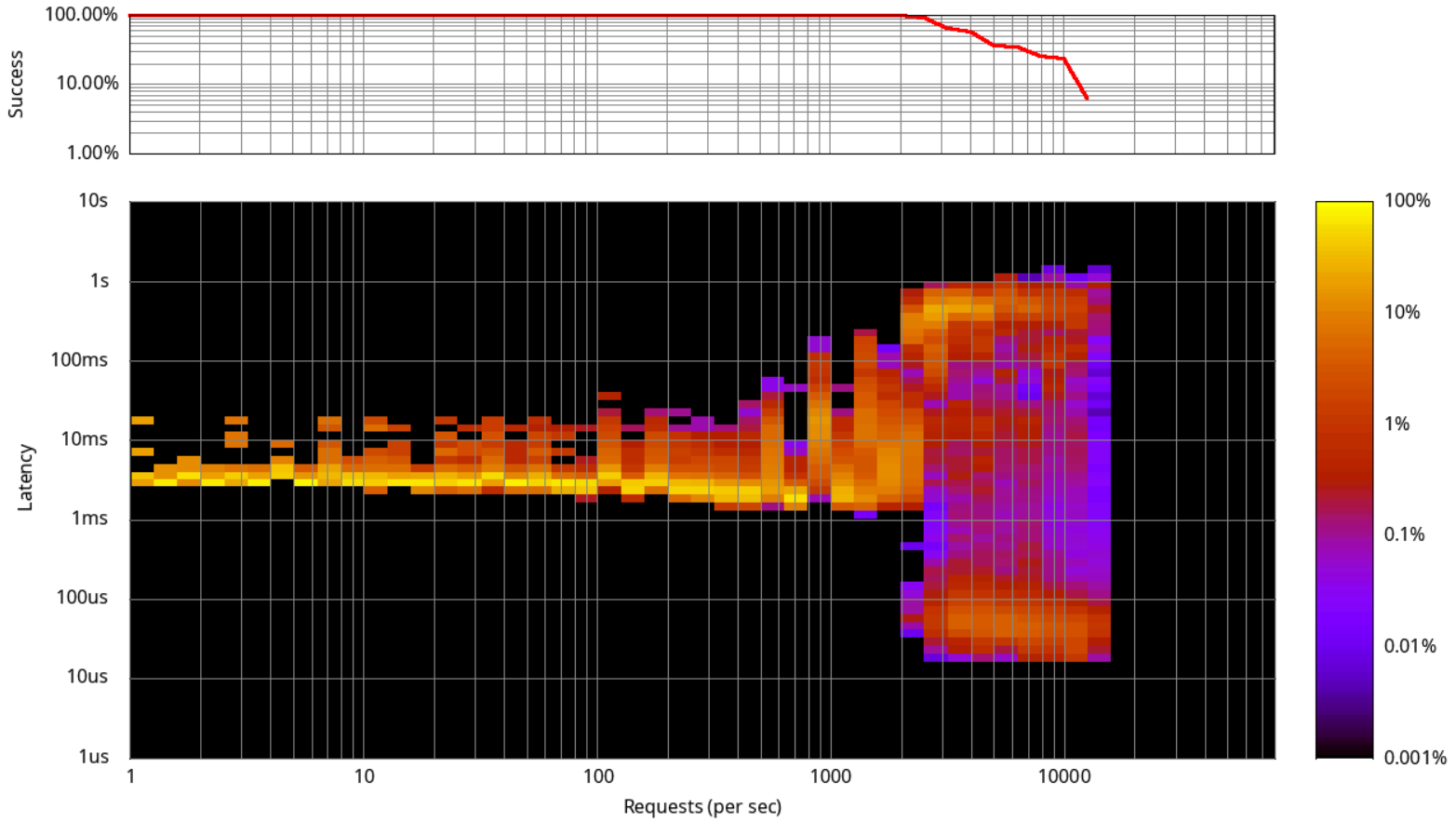
- Two dedicated AMD "vcpus"
- 4GB RAM
- 2Gbps bandwidth

Load tester run on equivalent VM, using vegeta

REMIX + EXPRESS.JS



LEPTOS V0.6 + AXUM



TAKEAWAYS

- ~50% reduction in page load time
- 3x-10x increase in traffic handled

RUST'S TYPE SYSTEM AND TOOLING

- Types across the boundary between Client and Server
- Types at compile time and run time
- Cargo, cargo-leptos, and rustfmt vs Eslint/Prettier/Npm/tsc/vite/etc.

DEVELOPER TIME

- The more work the tooling does, the less the programmer needs to keep things in their head
- Simplifies building and maintaing your codebase

> The rust part of it + reactivity brings amazing benefits to making sure that we spend (a little bit) more time building our application and almost no time debugging the version that's already running. So most of our time is spent on building new features and focusing on the product / user experience rather than fixing bugs and pushing patches. - Rakshith Ravi - VP Engineering, Patr

> Having a language built with a type system from the beginning combined with a framework that is competitive with modern JS frameworks and all of the use cases that implies means that I can build comparable sites to what I've done my entire career with far less cognitive overhead. - Chris Biscardi, Rust Adventure

> Leptos is essentially taking all the benefits of Rust and marrying them to all the benefits of Signals & SSR... I have done truly nothing to optimize yet and I already have top notch time to paint and time to reactive. Even on poor LTE beaches. - Alex, CBVA

ECOSYSTEM AND COMMUNITY

SHOUTOUT TO THE LEPTOS DISCORD

AWESOME LEPTOS

COMING SOON - LEPTOS 0.7

- Complete reactive system rewrite
- Make reactive system modular
- Ergonomic improvements for async data loading

TEMPTED YET?

1. Server functions
2. Performance
3. Infra costs
4. Rust's type system, error messages, and tooling
5. Reduced developer time vs app complexity

JOIN US

LEPTOS WEBSITE: HTTPS://LEPTOS.DEV

DISCORD

GITHUB

THE END

QUESTIONS?

PS: Come find me if you'd like to chat and/or if you'd like some Leptos stickers

Ben Wishovich | @benwis@hachyderm.io