Observations and Measurements of HTTP/2 During Large-Scale Web Crawls

MAPRG, IETF 124 Montréal

Thom Vaughan <thom@commoncrawl.org>
Principal Engineer, Common Crawl Foundation

COMMON CRAWL

About Common Crawl

Common Crawl is a non-profit organization which regularly crawls a significant sample of the web

We make the data accessible free to everyone interested in running analysis of large-scale web data

Over 10 PiB archived since 2007

Cited in over 10,000 research papers

At present, we crawl every month between 2 billion and 4 billion web pages

The data is hosted in the Amazon cloud as part of the AWS Open Data Sponsorship program

COMMON
CRAWL

More About Common Crawl

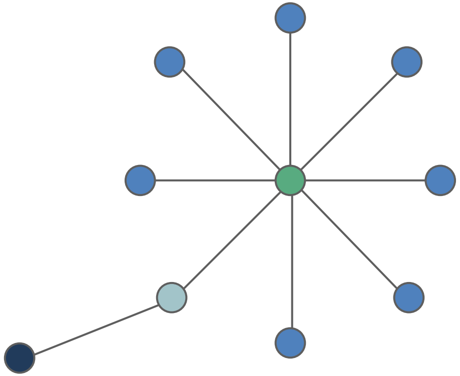
Our crawler is a fork of Apache Nutch™ (which does not speak HTTP/3... yet)

Our crawl is steered by domain-based Harmonic Centrality rankings, constructed by aggregating host-level data at the pay-level domain (PLD) level, using the public suffix list maintained on publicsuffix.org

We publish WARC, WAT, and WET data, along with Web Graphs (BVGraph) on a monthly basis

We have two indexes, one traditional CDX and one Apache Parquet™ (columnar) format

We preserve `robots.txt` fetches too



$$H(v) = \sum_{u \neq v} \frac{1}{d(v, u)}$$

Where $H(v)$ is the **Harmonic Centrality** of vertex v,
and $d(v,u)$ is the shortest path distance between vertices v and u.

About HTTP2

HTTP/2 was introduced in 2015 as the second major version of the Hypertext Transfer Protocol

HTTP/2 addresses several issues with HTTP/1.1 and focuses on performance, specifically low latency when loading a web page in web browsers and efficient use of network and server resources

The current standard is defined in RFC 9113

HTTP/2 continues to use the same building blocks as HTTP/1.1 – methods, status codes, header fields, and URIs, but the transport layer between client and server comes with several improvements

Example HTTP/2 WARC Capture

```
WARC/1.0
WARC-Type: request
WARC-Date: 2025-02-17T13:30:42Z
...
WARC-Target-URI: https://en.wikipedia.org/wiki/Saturn
WARC-Protocol: h2
WARC-Protocol: tls/1.3
WARC-Cipher-Suite: TLS_AES_128_GCM_SHA256
GET /wiki/Saturn HTTP/1.1
User-Agent: CCBot/2.0 (https://commoncrawl.org/faq/)
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: br,gzip
Host: en.wikipedia.org
Connection: Keep-Alive
```

```
WARC/1.0
WARC-Type: response
WARC-Date: 2025-02-17T13:30:42Z
...
WARC-Target-URI: https://en.wikipedia.org/wiki/Saturn
WARC-Protocol: h2
WARC-Protocol: tls/1.3
WARC-Cipher-Suite: TLS_AES_128_GCM_SHA256
...
HTTP/1.1 200
date: Sun, 16 Feb 2025 22:21:15 GMT
...
<!DOCTYPE html>
...
<title>Saturn - Wikipedia</title>
...
```

COMMON CRAWL

The `WARC-Protocol` field was the only option in order not to break downstream user code.

As shown in the example, we decided to repeat the `WARC-Protocol` header to encode both the HTTP and TLS version. Of course, this requires that downstream tools reading the WARC files are able to deal with the repeated headers.

WARC Reader Backward-Compatibility

HTTP/2 does not define a “reason phrase”; e.g., in HTTP/1.1 it would be “`Ok`” for the status code 200.

We decided not to include an artificial reason phrase, however, a whitespace is added after the status code to ensure that WARC readers can split the status line into three parts at space characters:

```
HTTP/1.1␣200␣\r\n
```

In 2020, HTTP/2 was accidentally enabled. Tests run in the aftermath showed that many WARC readers complain or fail on a HTTP status line with an unknown HTTP version number:

```
GET /index.html HTTP/2
```

We switched on h2 (deliberately) during the **CC-MAIN-2024-30** crawl of July 2024.

For the purposes of this analysis we are looking at the crawl **CC-MAIN-2025-05** which was performed in January of this year. The sample size is 3.0 billion web pages from 49 million hosts or 39 million registered domains.

61%

of all fetches in January were over HTTP/2 (including ones that failed).

COMMON CRAWL

See `s3://commoncrawl-dev/http2-experiments/` to get this data.

More specifically...

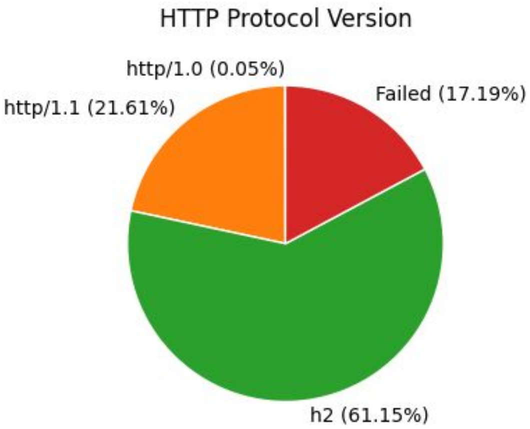
There are fetches which either failed before a HTTP connection could be established or were even not tried for various reasons, e.g. disallowed by `robots.txt`

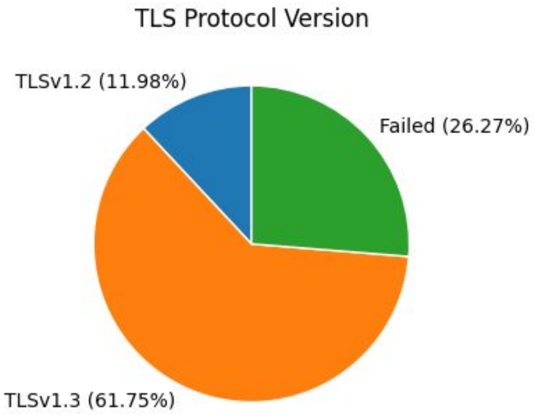
HTTP Protocol Version

```
     2204813      0.05%
http/1.0
   913279676     21.61%
http/1.1
  2584512235     61.15%   h2
```

TLS Protocol Version

```
   506399897     11.98%   TLSv1.2
  2609583284     61.75%   TLSv1.3
```

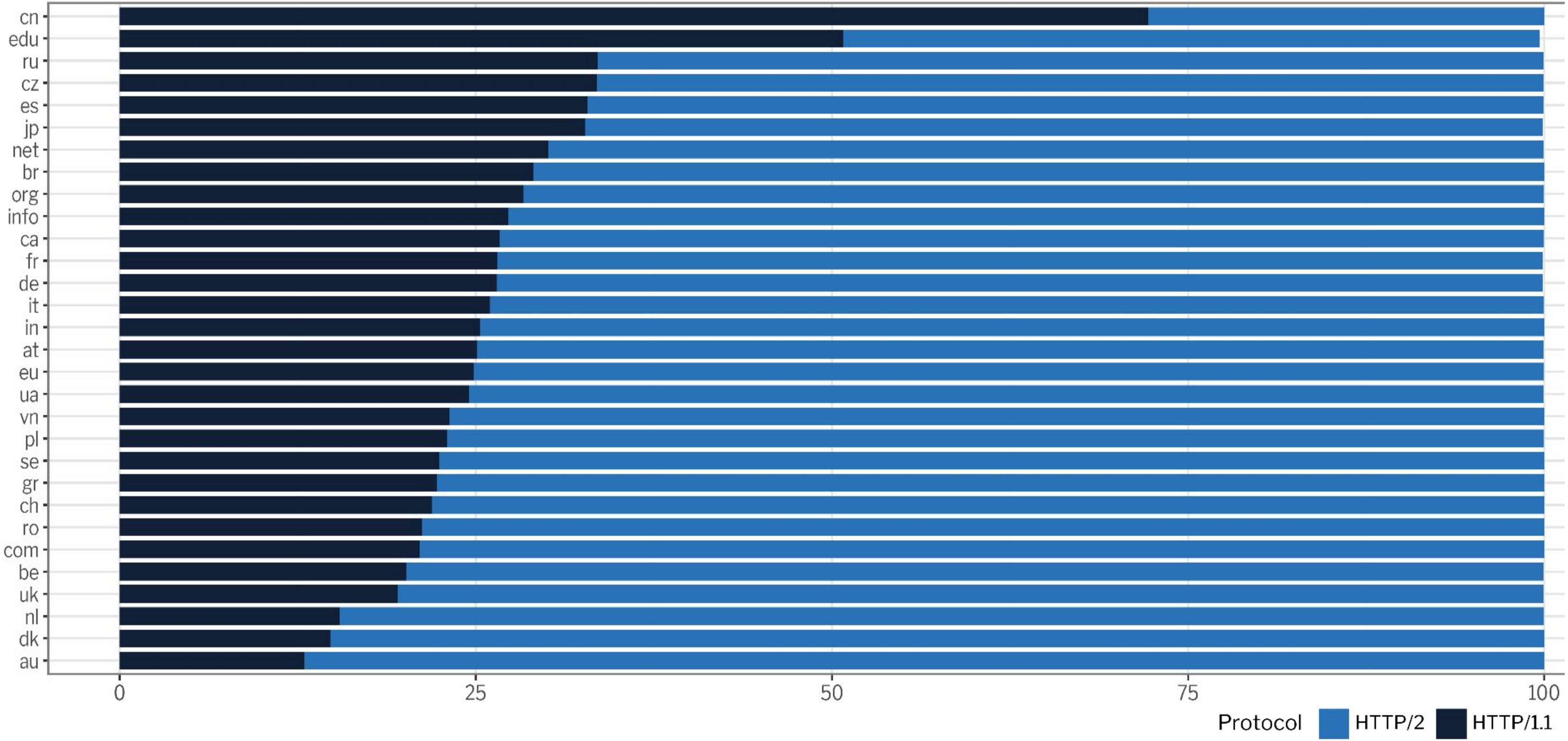


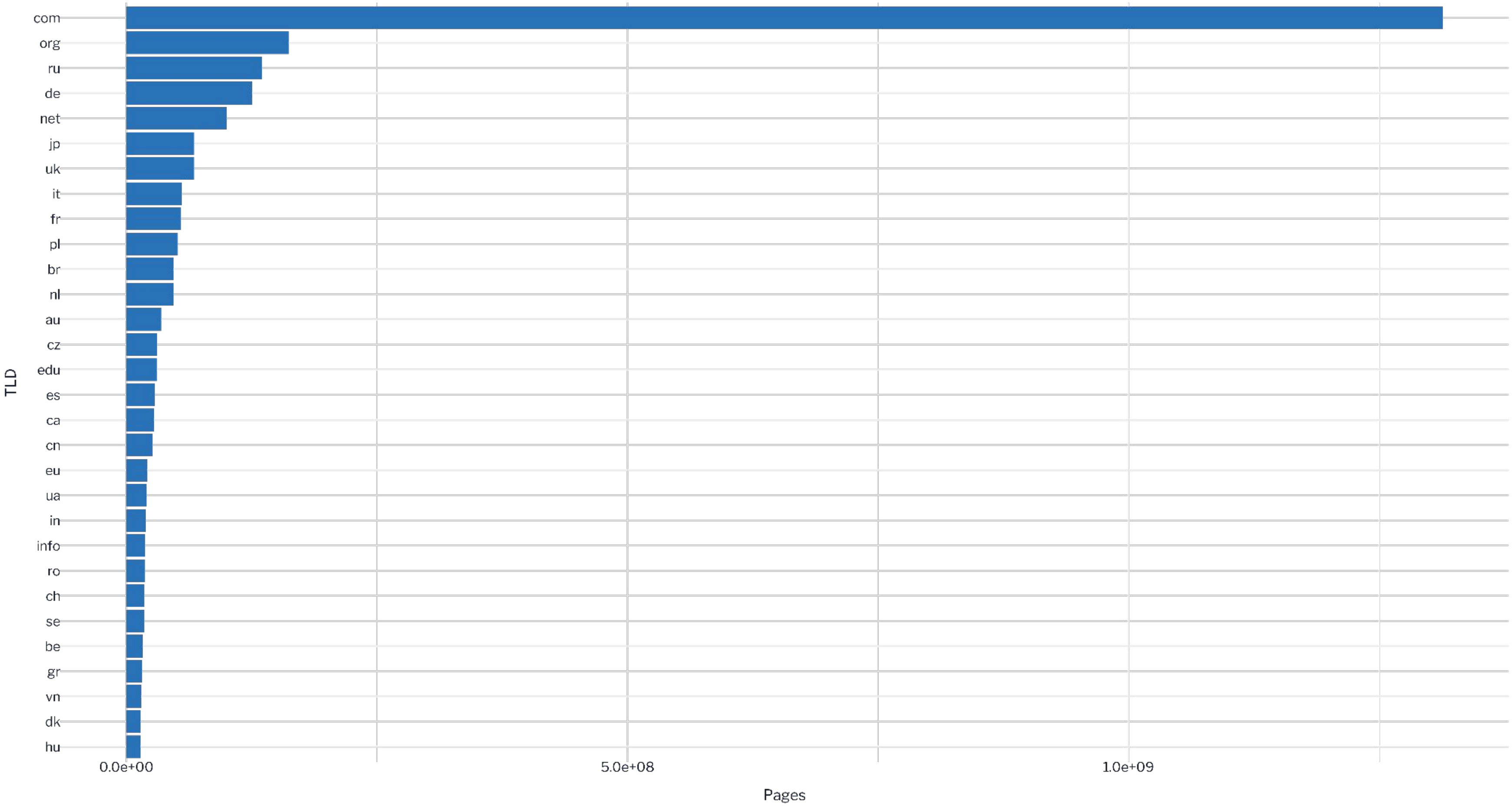


COMMON CRAWL

Adoption Rates of HTTP2 per TLD (CC-MAIN-2025-05)

Top 30 TLDs by total volume, ordered by HTTP/2 share





Top 30 TLDs for CC-MAIN-2025-05

Page Fetch Time

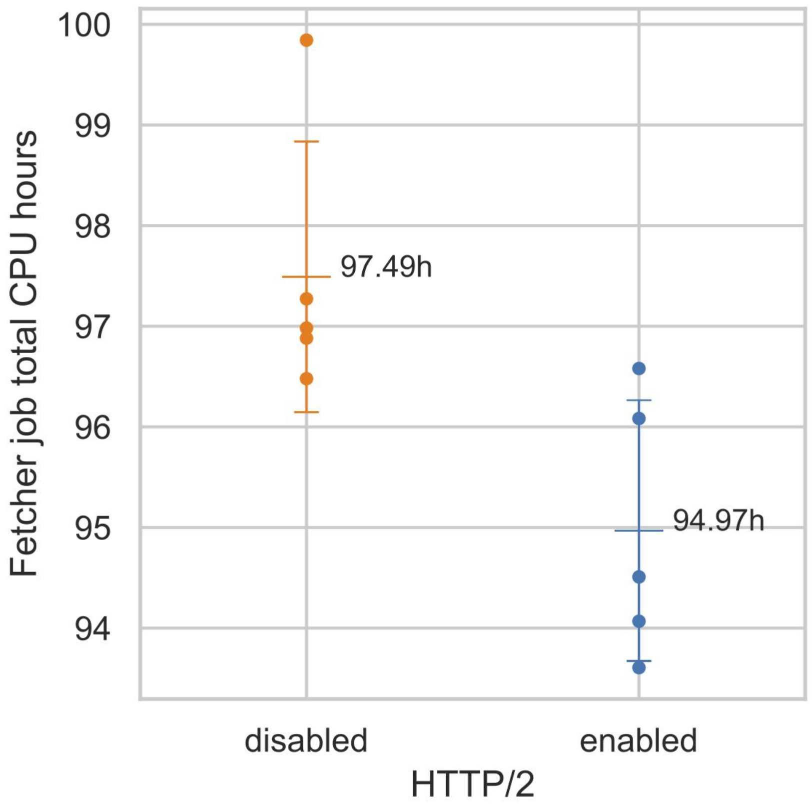
The average time spent to fetch a web page is slightly reduced with HTTP/2 enabled: from 1038 to 1020 ms.

A speed-up is also visible for the most common combination of HTTP/2 and TLS 1.3 in comparison to HTTP/1.1 and TLS 1.3.

HTTP/2	HTTP	TLS	CAPTURES	CAPT %	AVG FETCH TIME (MS)
disabled	*	*	**128,841,301**	**100.00**	**1038.29**
	http/1.0	-	15,166	0.01	1011.45
		tls/1.2	13,122	0.01	1168.32
		tls/1.3	21,471	0.02	1708.62
	http/1.1	-	14,681,800	11.40	900.04
		tls/1.2	20,781,482	16.13	1155.79
		tls/1.3	93,328,260	72.44	1033.71
enabled	*	*	**128,445,578**	**100.00**	**1020.31**
	http/1.0	-	15,892	0.01	1464.92
		tls/1.2	12,070	0.01	847.56
		tls/1.3	12,797	0.01	1846.70
	http/1.1	-	14,590,516	11.36	913.69
		tls/1.2	9,425,891	7.34	1186.34
		tls/1.3	11,410,878	8.88	1169.11
	h2	tls/1.2	11,327,309	8.82	1113.72
		tls/1.3	81,650,225	63.57	986.26

COMMON CRAWL

We observed a **2.5% reduction in CPU time** spent for the entire fetcher job.

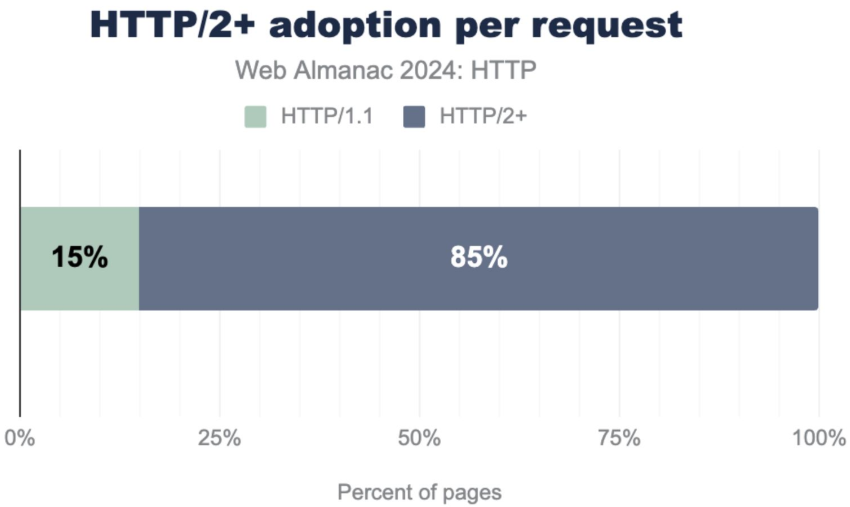


COMMON
CRAWL

What does the Web Almanac say about this?

In 2024 the adoption of HTTP/2+ is around 85%, this is "[...] for all requests (including subresources and third-parties) [...] because many websites will load additional resources from a variety of third-party domains (for example, analytics, plugins/tags, and social media integrations).

Due to their scale, these external services often either provide support for HTTP/2+ themselves or make use of a so-called CDN or Content Delivery Network (for example Akamai, Cloudflare or Fastly) that does it for them."



https://almanac.httparchive.org/en/2024/http#http-version-adoption

Summary

HTTP/2+ now dominates large-scale web traffic (61% of fetches)

It improves fetch times and reduces CPU load slightly

Implementation required careful handling of WARC compatibility

Common Crawl data enables reproducible, open research on real-world web technology adoption

Collaboration is most welcome!

Questions?