

Tuning NGINX for high performance

Nick Shadrin

nick@nginx.com

All links on one page

shadrin.org/talks/

Twitter:

[@shadrin](https://twitter.com/shadrin)

[@nginx](https://twitter.com/nginx)

[@nginxorg](https://twitter.com/nginxorg)





About me

- Nick Shadrin
- Product Manager at NGINX
- Based in San Francisco
- 16 years experience with web tech
- nick@nginx.com || [@shadrin](https://twitter.com/shadrin) || <https://shadrin.org>



Agenda

- A basic NGINX configuration
- NGINX performance optimizations:
- Operating system-level optimizations
- Networking-level optimizations
- NGINX core optimizations
- Conclusions and questions

NGINX



“... when I started NGINX, I focused on a very specific problem – how to handle more customers per single server.”

- Igor Sysoev, NGINX creator & our founder

About NGINX, Inc.

- Company founded in 2011, NGINX Plus started in 2013
- VC-backed by enterprise software industry leaders
- HQ in San Francisco, offices in US and Europe
- 800+ commercial customers
- 120+ employees

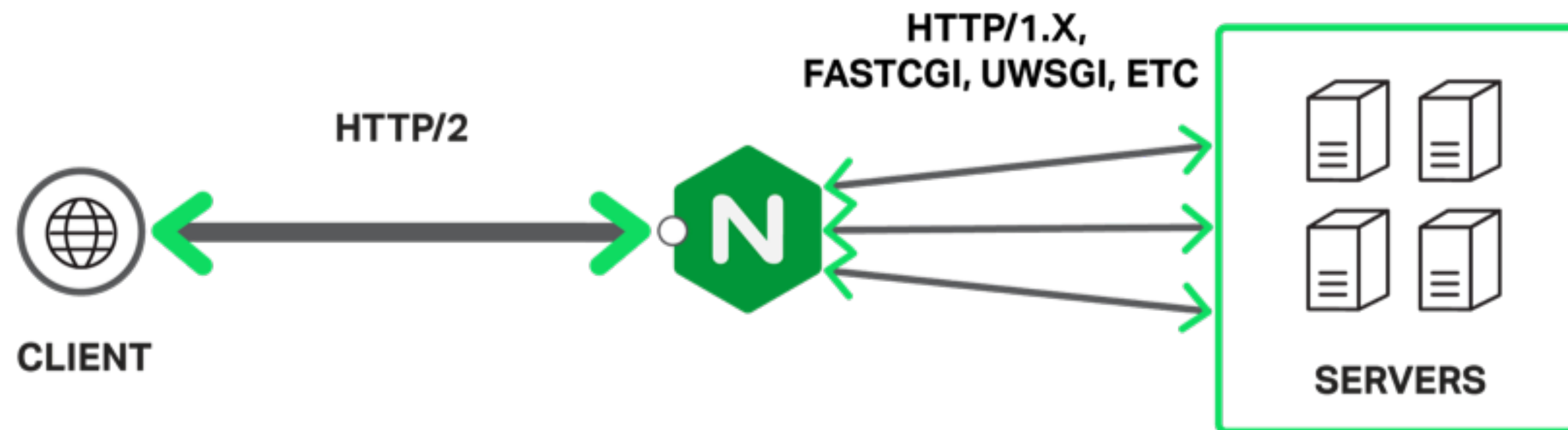


Web Scale

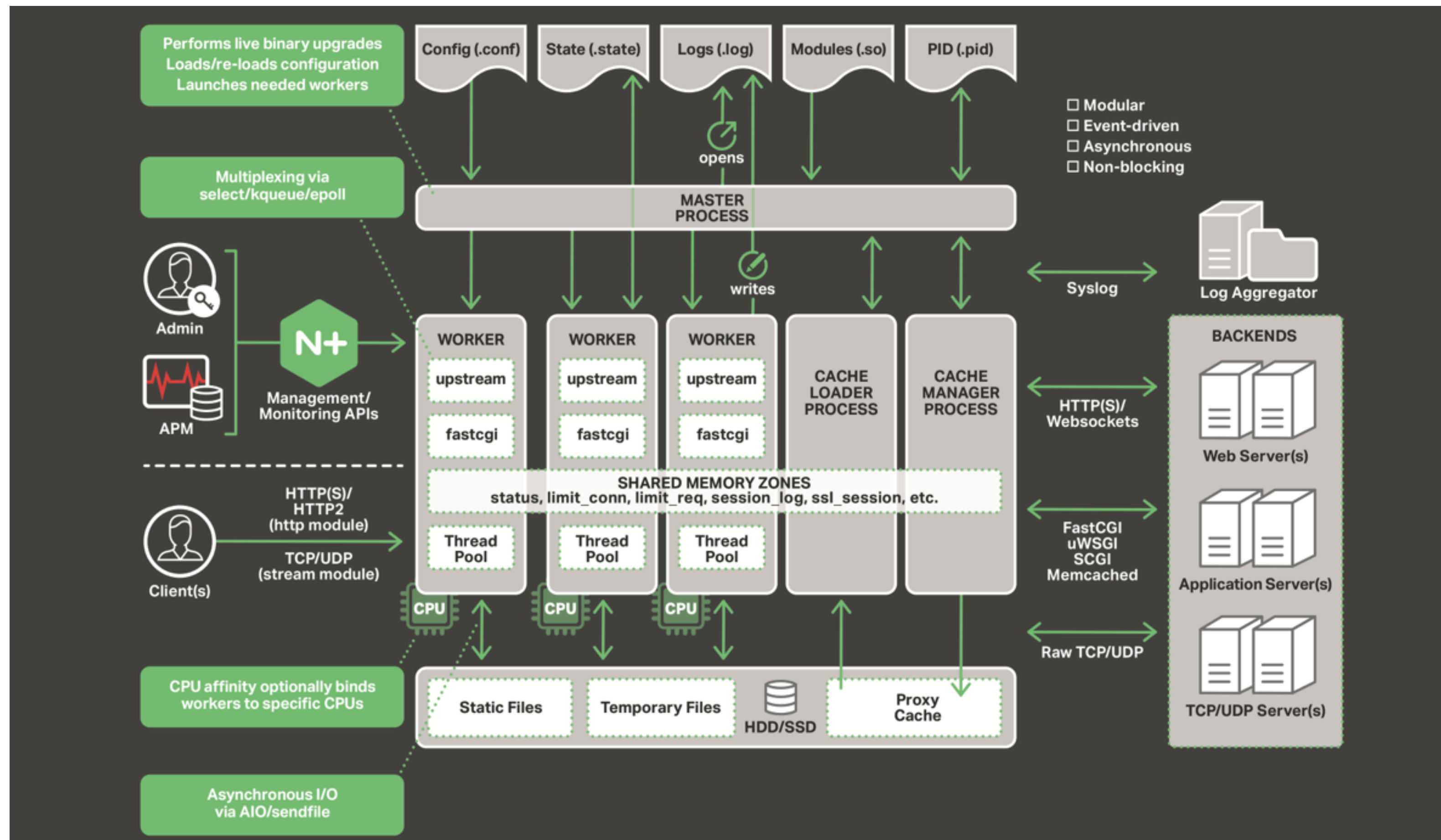
Architecture approach

- Design for scaling
- Segment microservices out
- Use caching and microcaching

Basic NGINX placement

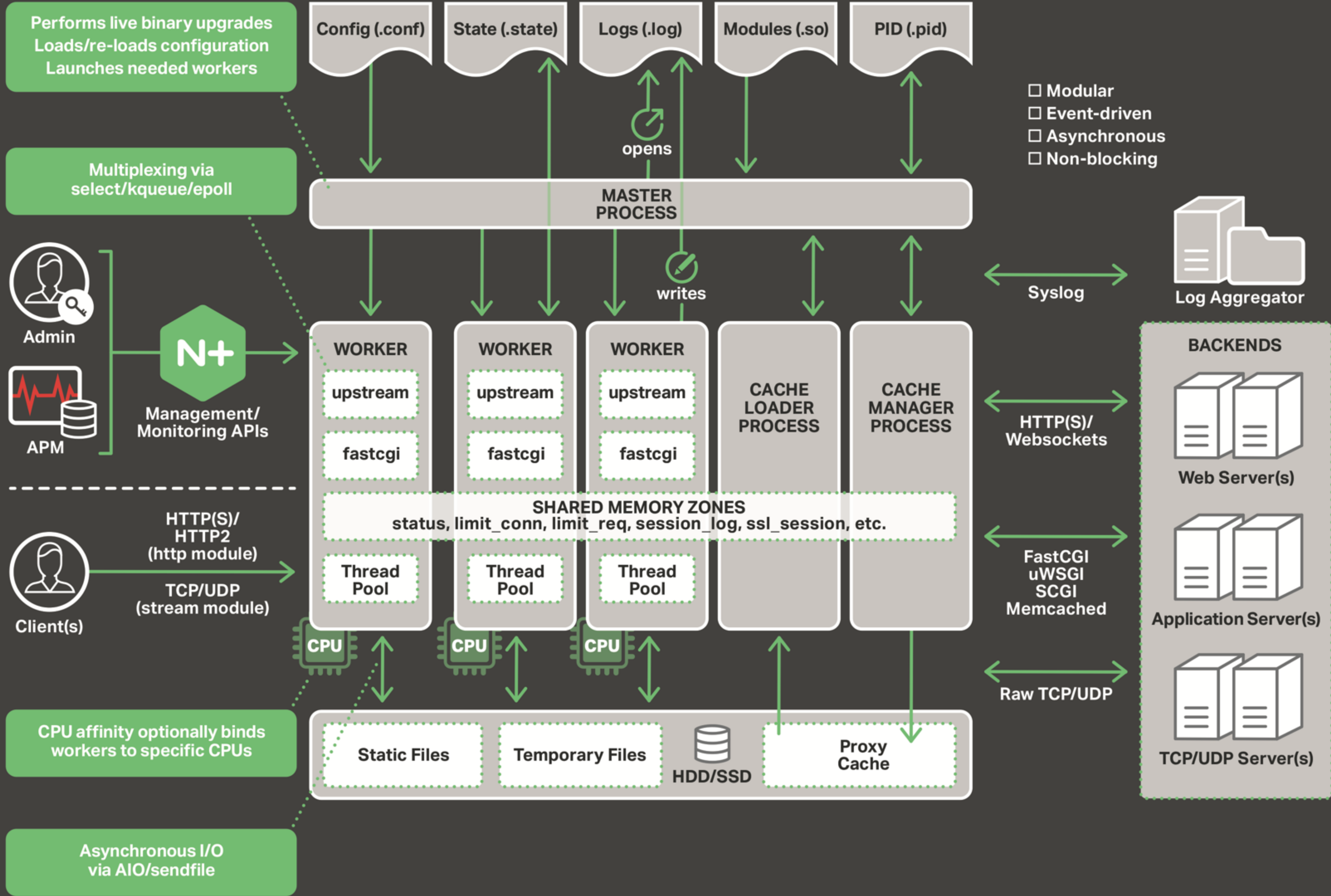


Inside NGINX



<https://www.nginx.com/blog/inside-nginx-how-we-designed-for-performance-scale/>

<http://www.aosabook.org/en/nginx.html>



OS tuning

- `net.core.somaxconn`
- `net.core.netdev_max_backlog`
- `net.ipv4.ip_local_port_range`
- `sys.fs.file_max`
- `/etc/security/limits.conf`, nofile setting

See <https://www.nginx.com/blog/tuning-nginx/>



Overcoming ephemeral port exhaustion

- Increase local port range
- Split traffic across multiple IPs
- NGINX since 1.11.2 uses `IP_BIND_ADDRESS_NO_PORT` socket option when available

<https://www.nginx.com/blog/overcoming-ephemeral-port-exhaustion-nginx-plus/>

Minimal NGINX configuration

```
events {}

http {

    server {
        listen 80;
        location / {
            proxy_pass http://backend;
        }
    }

    upstream backend {
        server backend1.example.com:8080;
        server backend2.example.com:8080;
    }
}
```



NGINX Performance features



NGINX Core features

- Use correct number of **worker_processes**
 - auto
 - # of available CPU cores
- Increase **worker_connections**
- Increase **worker_rlimit_nofile**

NGINX Core Features (cont'd)

- Turn off accept_mutex:
accept_mutex off;
- Turn on Sendfile
sendfile on;
- Use thread pools if I/O needs offloading:
aio threads;

<https://www.nginx.com/blog/thread-pools-boost-performance-9x/>



Changes with nginx 1.11.3

26 Jul 2016

*) Change: now the "accept_mutex" directive is turned off by default.

[skip]

<http://nginx.org/en/CHANGES>

NGINX Core Features (cont'd)

- Turn off accept_mutex:
accept_mutex off;
- Turn on Sendfile
sendfile on;
- Use thread pools if I/O needs offloading:
aio threads;

<https://www.nginx.com/blog/thread-pools-boost-performance-9x/>



HTTP Keep alive

- Keepalive connections allow to reuse the same TCP connection for multiple HTTP requests.
- For HTTP/1.1, no need to define anything, it's enabled by default on the frontend.
- Keepalives provide major performance benefit when used over SSL/TLS connections.

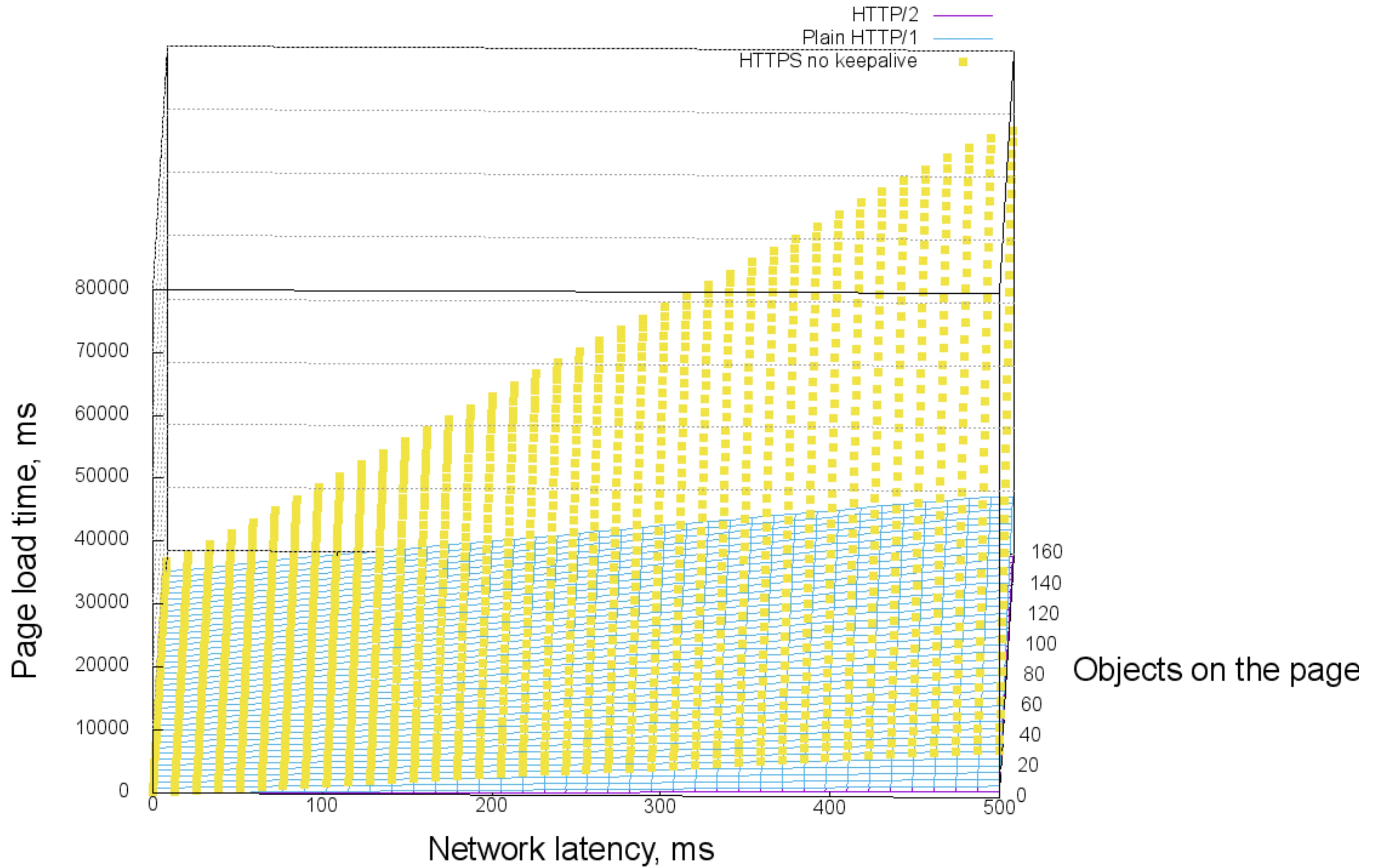


HTTP Keepalive: benchmark

- HTTPS with NO keepalive (worst setup)
- Plain HTTP
- HTTP/2 with SSL



HTTP/2 vs HTTP/1 vs HTTPS





HTTP Keepalive

- Keepalive on the Frontend:
keepalive_requests 100;
keepalive_timeout 75s;

HTTP Keepalive on the backend

Keepalive on the Backend:

```
server {  
    location / {  
        proxy_pass http://backend;  
        proxy_http_version 1.1;  
        proxy_set_header Connection "";  
    }  
}  
...  
upstream backend {  
    server example.com;  
    keepalive 32;  
}
```



HTTP Caching

- Microcaching with NGINX:
<https://www.nginx.com/blog/benefits-of-microcaching-nginx/>
- Cache placement strategies:
<https://www.nginx.com/blog/cache-placement-strategies-nginx-plus/>



HTTP/2

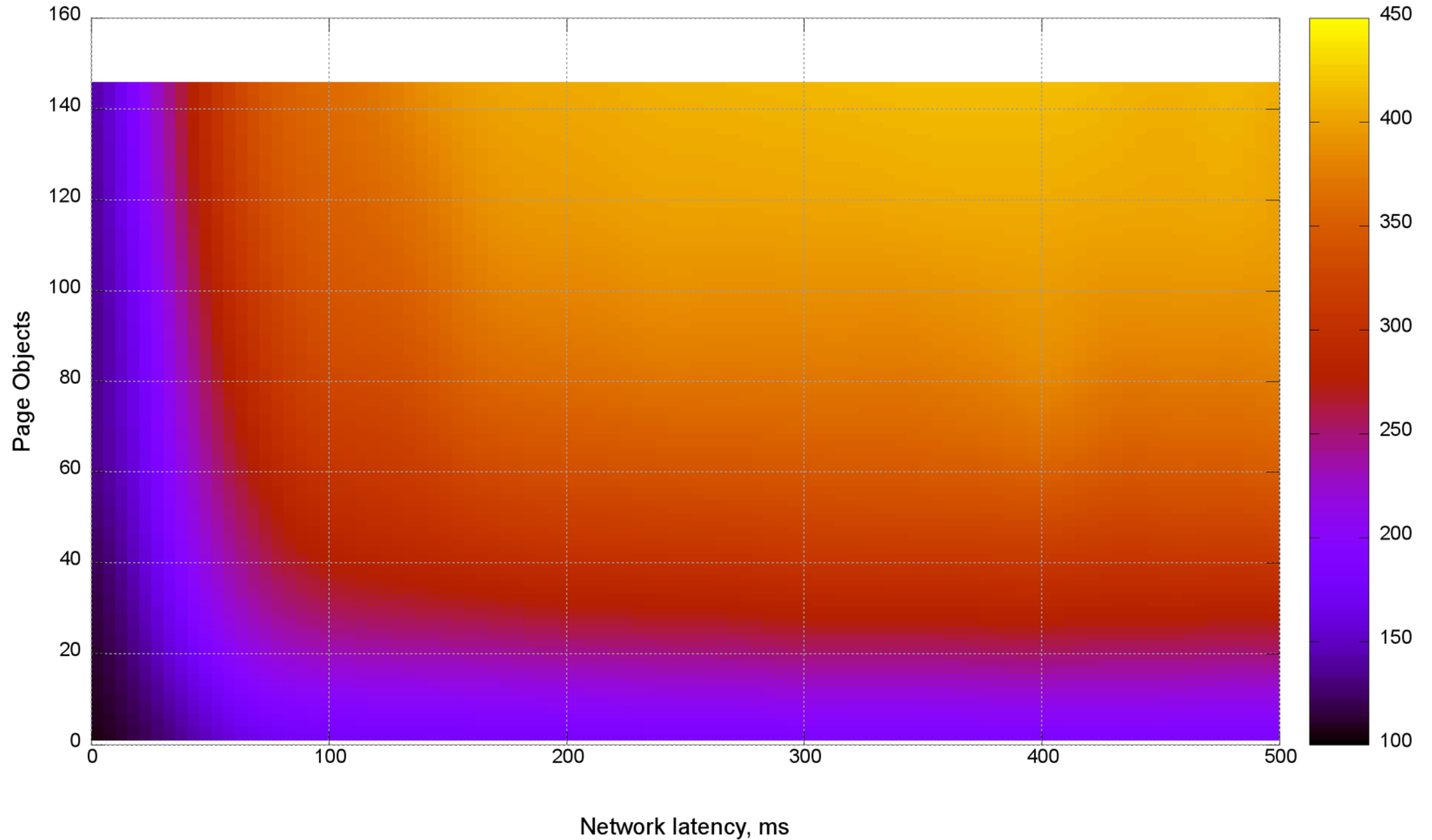
- Introduced in 2015 as a standard
- Based on Google's SPDY
- Includes major changes compared to HTTP/1:
 - Binary headers with HPACK
 - Multiple streams
 - Prioritization
 - Server Push





HTTP/2 benchmark

- NGINX 1.10.0
- Ubuntu 16.04
- Openssl 1.0.2
- Chrome Web browser
- Measuring full page reload

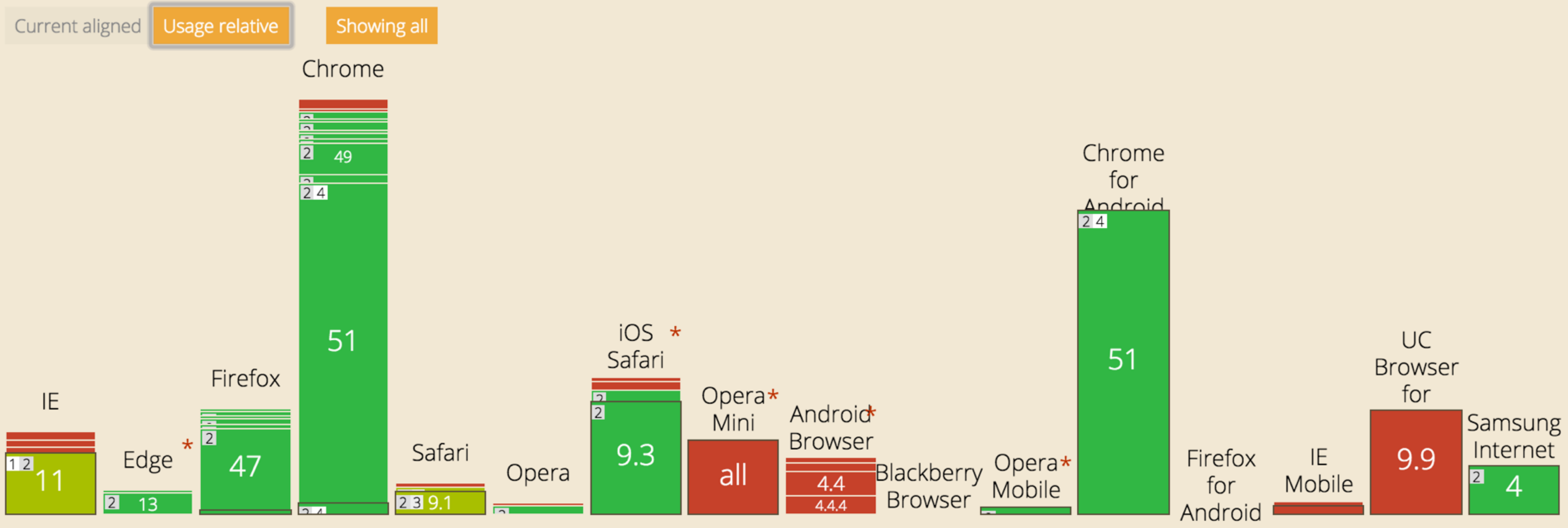
HTTP/2 vs HTTP/1/SSL, percentage performance increase



Some numbers

- 40ms / 50 objects:
HTTP/1: **510ms**
HTTP/2: **250ms**  **~2 times faster**
- 200ms / 100 objects:
HTTP/1: **4.0s**
HTTP/2: **1.1s**  **~4 times faster**

Networking protocol for low-latency transport of content over the web. Originally started out from the SPDY protocol, now standardized as HTTP version 2.



Screenshot: 2016-08-23, caniuse.com

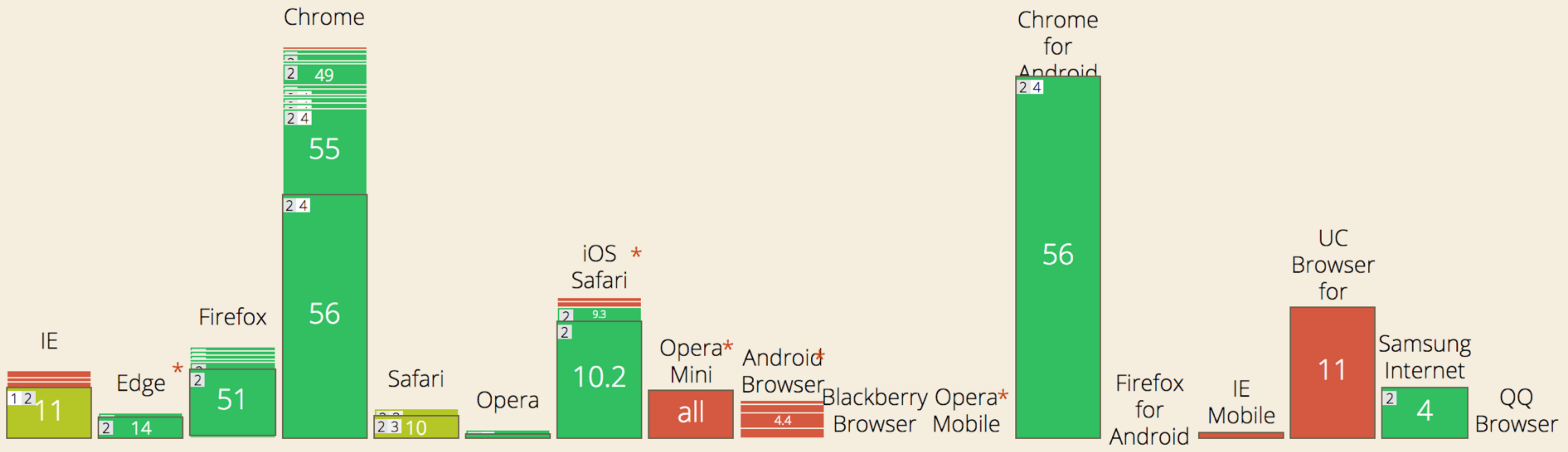
HTTP/2 protocol - OTHER

Global

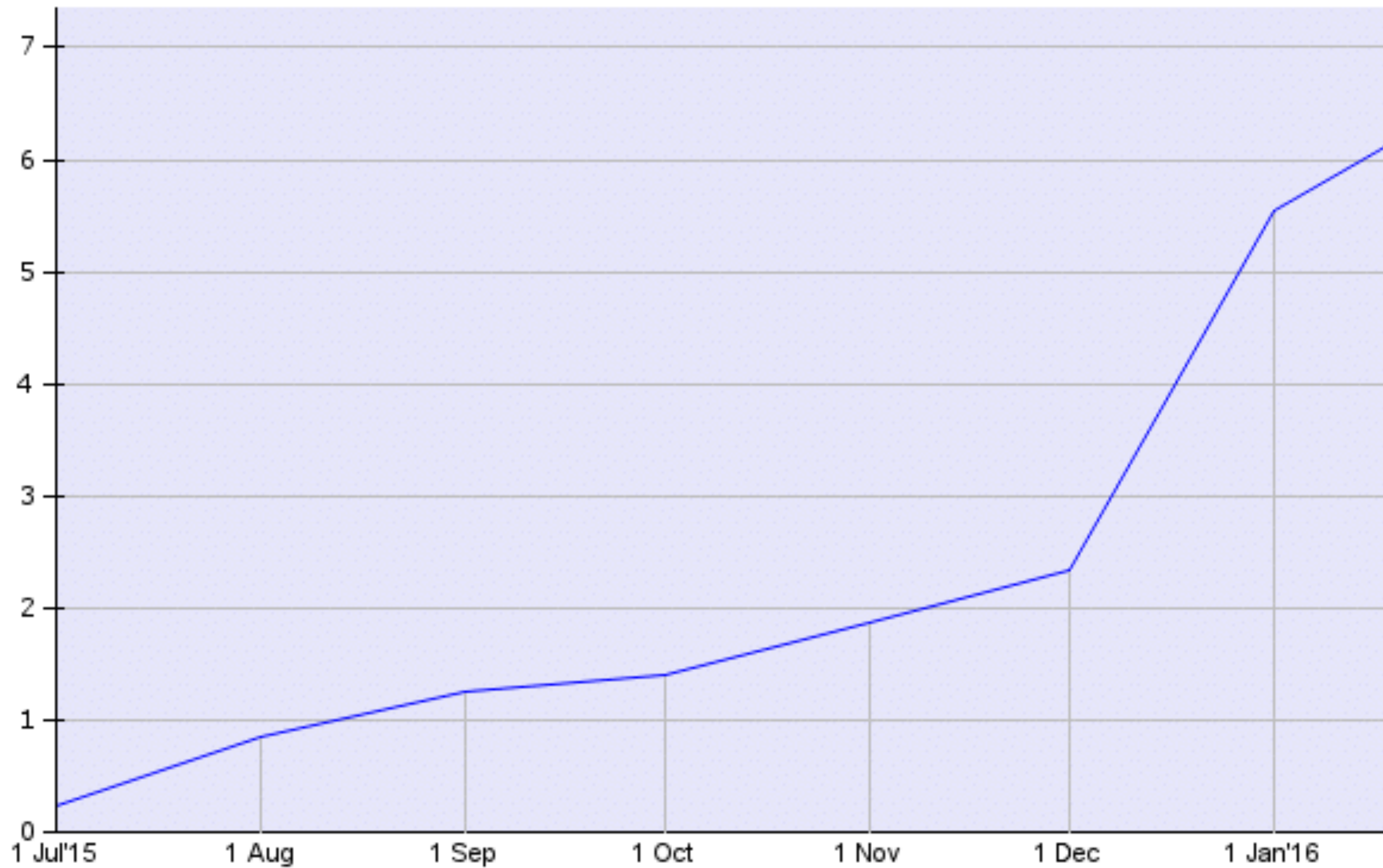
73.91% + 5.65% = 79.56%

Networking protocol for low-latency transport of content over the web. Originally started out from the SPDY protocol, now standardized as HTTP version 2.

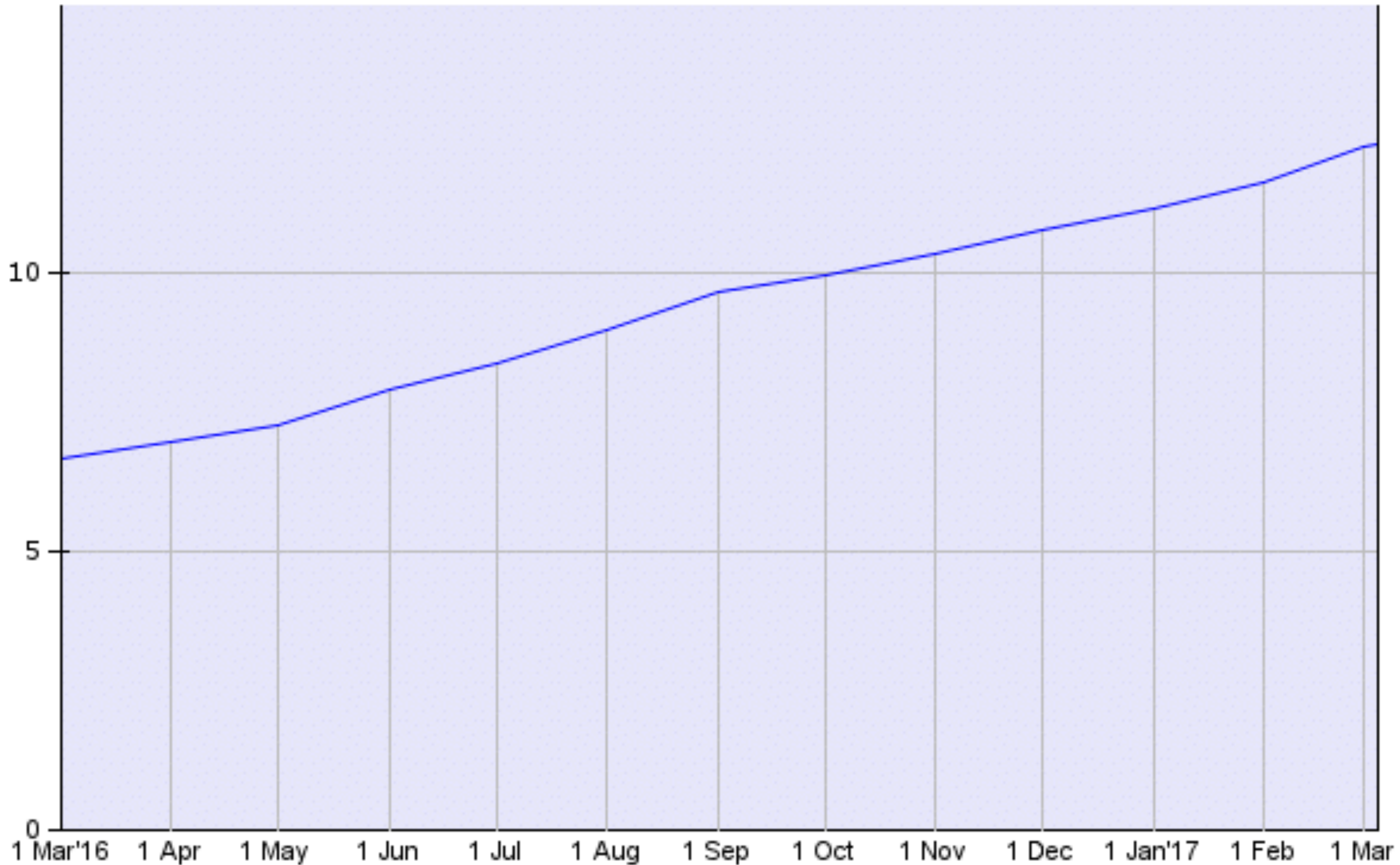
Current aligned Usage relative Date relative Showing all



Screenshot: 2017-03-05, caniuse.com



Usage of HTTP/2 for websites, 18 Jan 2016, W3Techs.com



Usage of HTTP/2 for websites, 5 Mar 2017, W3Techs.com

Measure your results

- NGINX provides extensive logs with custom variables. Configure **log_format** with:
\$upstream_response_time
\$request_time
\$upstream_cache_status
- NGINX has simple set of metrics with stub_status module. Configure **stub_status**
- **NGINX Plus** provides more extensive metrics with Extended Status module
- **NGINX Amplify** is a free monitoring SaaS solution.

Sign up at amplify.nginx.com

The screenshot displays the NGINX Amplify dashboard for a system named 'NGINX-PLUS-R11'. The interface includes a navigation bar with 'NGINX Amplify beta', 'Graphs', 'Dashboards', 'Analyzer', and 'Alerts'. A left sidebar lists several systems, with 'prod-nginxplus-lb01' (nginx-plus-r11) highlighted. The main area contains six performance graphs for the selected system, all showing data from 11:30 to 12:15. A vertical green line at 12:15 marks a specific event, with a callout box indicating a value of 62.22 at that time.

- NGINX CONNECTIONS/S:** Line graph showing 'nginx.http.conn.accepted' (blue) and 'nginx.http.conn.dropped' (black). A peak of 62.22 is noted at 12:15.
- NGINX REQUESTS/S:** Line graph showing 'nginx.http.request.count' (blue).
- NGINX CURRENT CONNECTIONS:** Line graph showing 'nginx.http.conn.current' (blue), 'nginx.http.conn.active' (black), and 'nginx.http.conn.idle' (green).
- NGINX CURRENT REQUESTS:** Line graph showing 'nginx.http.request.current' (blue), 'nginx.http.request.reading' (black), and 'nginx.http.request.writing' (green).
- NGINX HTTP ERRORS:** Bar chart showing 'nginx.http.status.4xx' (blue), 'nginx.http.status.5xx' (black), and 'nginx.http.status.discarded' (green).
- NGINX HTTP VERSION:** Bar chart showing 'nginx.http.v1_0' (blue), 'nginx.http.v1_1' (black), and 'nginx.http.v2' (green).

At the bottom left, there is a 'New System' button and a copyright notice: 'Copyright © 2015 NGINX, Inc. Terms of Service.' A chat icon is visible in the bottom right corner.

Sign up at amplify.nginx.com

The screenshot displays the NGINX Amplify web interface. The browser address bar shows 'amplify.nginx.com'. The navigation bar includes 'NGINX Amplify beta', 'Graphs', 'Dashboards', 'Reports', and 'Alerts'. On the left, a 'SYSTEMS' sidebar lists several systems, with 'prod-nginxplus-lb01' highlighted in a green box. The main content area is titled 'NGINX CONFIGURATION REPORT' and shows the configuration for 'nginx-plus-r10 @ prod-nginxplus-lb01'. The 'Static analysis' tab is active, displaying two warnings. The first warning states: 'Warning – There should be normally a unique `server_name` directive per `server`. Please ensure that a `server_name` contains all possible virtual server names for your site(s)'. It lists the following files to check: `/etc/nginx/amplify.conf`, line 27; `/etc/nginx/caches.conf`, lines 14, 20, 26, 38, 52; `/etc/nginx/demo.conf`, lines 92, 100. The second warning states: 'Warning – It is recommended that a `proxy_pass` always includes header definitions (at least `Host` header has to be explicitly defined). Consider adding the following additional directives to your nginx configuration:'. Below this, a text box contains the directive: `proxy_set_header Host $host;`. At the bottom, it says 'and optionally the following:' followed by a text box containing: `proxy_set_header X-Real-IP $remote_addr;`. The footer includes the NGINX logo, copyright information, and a social media icon.

NGINX Amplify ^{beta}

Graphs Dashboards **Reports** Alerts

Q SYSTEMS

- dev-nodejs-api01
nginx 1.10.0
- dev-nodejs-api02
nginx 1.10.0
- dev-nodejs-api03
nginx 1.10.0
- prod-nginxplus-lb01**
nginx-plus-r10
- prod-rails-web01
nginx 1.10.0
- prod-rails-web02
nginx 1.10.0

+ New System

Copyright © 2015 NGINX, Inc. Terms of Service.

NGINX CONFIGURATION REPORT

nginx-plus-r10 @ prod-nginxplus-lb01

Version Overview Virtual servers SSL **Static analysis**

Warning – There should be normally a unique `server_name` directive per `server`. Please ensure that a `server_name` contains all possible virtual server names for your site(s).

Check the following file(s):

- `/etc/nginx/amplify.conf`, line 27
- `/etc/nginx/caches.conf`, line 14
- `/etc/nginx/caches.conf`, line 20
- `/etc/nginx/caches.conf`, line 26
- `/etc/nginx/caches.conf`, line 38
- `/etc/nginx/caches.conf`, line 52
- `/etc/nginx/demo.conf`, line 92
- `/etc/nginx/demo.conf`, line 100

Warning – It is recommended that a `proxy_pass` always includes header definitions (at least `Host` header has to be explicitly defined). Consider adding the following additional directives to your nginx configuration:

```
proxy_set_header Host $host;
```

and optionally the following:

```
proxy_set_header X-Real-IP $remote_addr;
```



Conclusions

- Plan for scalability early
- Tune low level operating system
- Configure Keepalive
- Configure caching
- Enable HTTP/2
- Measure your results



How to Contribute

- hg.nginx.org
- github.com/nginx
- nginx.org/mailman

Thank You

All links in one page:
<https://shadrin.org/talks>

Twitter: @shadrin

nick@nginx.com

