H2O — a fast HTTP server

Frederik Deweerdt

Overview

- Kazuho Oku main developer
- github.com/h2o/h2o
- MIT license
- In development since 2014
- In production at Fastly since July 2016

Features

Features

- HTTP/1.0, HTTP/1.1
- HTTP/2
- C
- Multi-threaded
- Asynchronous (epoll, kevent, libuv)
- Reverse proxy
- Ruby scriptable
- TLS with OpenSSL/LibreSSL/BoringSSL
- TLS 1.3
- Can be used as a library

H2O libraries

- picohttpparser (HTTP/1.x parser)
- picotls (TLS 1.3 implementation)
- neverbleed: isolates private keys
- server starter (seamless restart)

Test suite

- Extensive test suite with two main components:
    - C unit tests
    - Perl integration tests
- Run in Travis CI for every github PR
- At Fastly we run builds under ASAN

YAML config file

```
listen: 8080
listen:
  port: 8081
  ssl:
    certificate-file: examples/h2o/server.crt
    key-file: examples/h2o/server.key
hosts:
  "127.0.0.1.xip.io:8080":
    paths:
      /:
        file.dir: examples/doc_root
    access-log: /dev/stdout
  "alternate.127.0.0.1.xip.io:8081":
    listen:
      port: 8081
      ssl:
        certificate-file: examples/h2o/alternate.crt
        key-file: examples/h2o/alternate.key
    paths:
      /:
        file.dir: examples/doc_root.alternate
    access-log: /dev/stdout
```

Performance

HTTP/2

- Supports pushes via Link: headers
- Cache aware pushes via CASPER
- MIME aware prioritization
- O(1) stream scheduler
- Supports early metadata for async pushes
- TCP responsiveness

Numbers

```
$ wrk -s pipeline.lua -t 1 -c 400 -d 10 https://127.0.0.1:18081/ -- 80
Running 10s test @ https://127.0.0.1:18081/
  1 threads and 400 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency     7.38ms    0.92ms  12.16ms   84.00%
    Req/Sec   115.23k    16.14k  137.47k    53.61%
  771447 requests in 10.00s, 158.18MB read
Requests/sec:  77140.92
Transfer/sec:     15.82MB
```

```
$ wrk -s pipeline.lua -t 6 -c 400 -d 10 https://127.0.0.1:18081/ -- 80
Running 10s test @ https://127.0.0.1:18081/
  6 threads and 400 connections
  Thread Stats   Avg      Stdev     Max   +/- Stdev
    Latency     1.74ms  524.05us  30.82ms   81.53%
    Req/Sec    95.89k    16.11k  132.67k    68.05%
  5046422 requests in 9.93s, 1.01GB read
Requests/sec: 508084.52
Transfer/sec:    104.18MB
```

Memory allocation

- Use per request memory pools
  - `h2o_mem_pool_t *pool`
  - memory is freed at the end of the request
- Per thread memory allocator acts as a free list
- The allocator has a refcounting API for shared references
- When the request is closed, the pool is destroyed

Modules

Source

- handler
  - generates content
  - file, proxy
- filter
  - modifies content
  - compression, chunking
  - can be chained
- logger
  - Also used for stats

Source

- Protocols: `lib/http1.c` and `lib/http2/*`
- HTTP/2
    - The main entry point is `lib/http2/connection.c`
    - `handle_xyz_frame()`

hello world handler

https://github.com/deweerdt/h2o-exercise