HAProxy 1.5 and beyond

FRnOG 22 - 2014/04/04

Willy Tarreau <willy@haproxy.com>

HAProxy / ALOHA R&D
http://www.haproxy.com/

Quick history - major dates

- Project started in 2000 as a hack to rewrite HTTP headers during a benchmark
- **2001/12/16** : version 1.0 : deployed in emergency to off-load a failing load balancer
- **2002/03/10** : version 1.1.0 : basic LB, checks
- **2003/09/20** : version 1.1.23 with English Documentation marks the real take-off
- **2003/11/09** : version 1.2.0 : IPv6, beginning of performance improvements
- **2004/12/26** : version 1.2.3 : first external contrib (appsession)
- **2006/03/19** : version 1.2.10 : first use in a commercial product (ALOHA v1.0)
- **2006/06/29** : version 1.3.0 : focus on flexibility (frontends/backends)
- **2009/06/09** : start of 1.4-dev branch : new development model with stable/dev branches
- **2010/02/26** : version 1.4.0 : much better HTTP compliance, content analysis, ...
- **2010/05/23** : start of 1.5-dev branch

Change of goals over time

- Initially, focused on **simplicity** (was a tool subject to quick and dirty updates).
- Then focused on **CPU and memory** savings for mainstream OSes and hardware (Solaris 2.6 on UltraSparc 170 MHz, Linux 2.2 on Pentium2 450, with up to 128 MB of RAM).
- Focused on **reliability** when starting to be used in production in a large bank
- Focused on connection **scalability** as the usage grew
- Focused on dealing with **large configurations** as adoptions increased
- Started to focus on **maintainability** as critical sites adopted it
- Changed the development model to adopt **devel and stable** branches (*thanks Git*)
- Focused on **network bandwidth** as large sites adopted it (TCP splicing for 10+ Gbps)
- Focused on **modularity** as features started to grow and to share code
- Current focus is on ability to **contribute/debug/audit** to scale the project team

Goals which have not changed

- Reliability above anything else.
- When a user asks for a wrong feature, he has real needs that must be addressed (*eg: leastconn, server weight, SSL, compression, keep-alive, ...*)
- Long-term maintenance (1.3 still supported, 1.1 still alive in field)
- No config breakage, guide user through warnings and advices instead (1.5 loads 1.1 configs)

Current state of affairs

- 1.5-dev22 released on 2014/02/16.
- 1.5-final expected "soon" (*"soon" = "when it's ready"*)
- 1.4 currently is the mostly deployed version in numbers of sites
- 1.5-dev currently is the version deployed on the largest sites.
- Some sites using 1.4 have already replaced stunnel with 1.5-dev on the front

  ⇒ **1.5-dev still needs to be stable enough because large sites rely on it today.**

Why migrate from 1.4 to 1.5

- SSL : getting rid of Stunnel
  - Native OpenSSL inclusion, all SSL info available
  - Client and Server side
  - Supports SNI, NPN, ALPN
  - Multi-hosting, wildcards and crt-list
  - *Note: thanks to Bumptech for the immense help with Stud!*
- End-to-end HTTP Keep-Alive (static farms, NTLM)
- IPv6 : supported everywhere (server, ACLs, ...)
- HTTP Compression

Why migrate from 1.4 to 1.5 *(cont'd)*

- PROXY protocol : now adopted by many common products :

```
PROXY TCP4 192.168.0.1 192.168.0.11 56324 443\r\n
GET / HTTP/1.1\r\n
Host: 192.168.0.11\r\n
\r\n
```

  - Client-side : haproxy, stud, stunnel, exaproxy, ELB
  - Server-side : haproxy, stud, postfix, exim, nginx, varnish (in progress)
- More rulesets:
  - tcp-request connection,
  - tcp-response,
  - http-response
- More actions:
  - HTTP: add-header, set-header, redirect, tarpit
  - TCP: set-nice, set-log-level, set-tos, set-mark, close, expect-proxy, track-*

Why migrate from 1.4 to 1.5 *(cont'd)*

- Sample extraction from everything available (address, payload, cookie count, date, env, ...)
- Pipelined sample processing via various converters (eg: `"hdr(host),lower,map(to_cust.map)"`)
- ACLs can use any match method with any input sample
- Maps and dynamic ACLs updatable from the CLI
- Stick-tables and counters : track usage stats for any given key :
  - cumulated/concurrent connections, connection rate
  - total bytes in/out, in/out byte rates
  - total HTTP requests, HTTP errors, HTTP req rate, error rate

Why migrate from 1.4 to 1.5 *(cont'd)*

- Dynamic strings made from samples, usable at many places :
  - Custom log format
  - Custom unique-id insertion
  - HTTP header manipulation
  - Redirects
- Improved health checks :
  - All are SSL-compatible
  - Scriptable TCP checks
  - Check agent
  - Redis, PgSQL

Why migrate from 1.4 to 1.5 *(cont'd)*

- Many actions on the CLI :
  - frontend: enable/disable/shutdown
  - table/acl/map : add/del/show/search/clear entries
  - checks: enable/disable
  - limits: set maxconn, rate-limit on many settings
- Programmable actions on server state transition (`on-marked-down`...)
- Environment variables usable in all addresses
- More tunables (header counts, cookie length, ...)
- Configurable hash algorithms
- Configuration scalability to tens of thousands of backends

Why migrate from 1.4 to 1.5 *(cont'd)*

- Platform-specific features :
  - IPv6 transparent binding (Linux)
  - TCP Fast Open (Linux)
  - cpu-map (Linux)
  - tproxy (FreeBSD/OpenBSD)
- PCRE Jit

Focus for 1.6

- Config syntax update / removal of obsolete features (eg: reqsetbe)
- better multi-process / multi-thread integration
  - needed to maximize SSL & compression performance
  - requires better stats handling
  - health check synchronization
  - stick-table sharing ?
- RAM-based small objects cache
- DNS resolving on-the-fly / checks
- HTTP/2 gatewaying to 1.x

Focus for 1.6 (cont'd)

- Stateless gzip compression
- SSL : shared cache, CyaSSL
- Improved POST/body processing
- save / restore check states across reloads
- "wait on resource"
  - dynamic buffer allocation
  - multi-level traffic shaping
- More core developers for better scalability
- More: see ROADMAP file!

Commercial extensions to come by 2014

- Browser fingerprinting
  ⇒ *differenciate a real browser from a bot*
  ⇒ *avoid blocking search engines scraping your site*
- Bot Net stopper
  ⇒ prevent botnets from hammering your site
- APT protection
  ⇒ don't let attackers abuse HTTP to wake up backdoors or bypass filtering
- DDoS mitigation
  ⇒ 20 Gbps stateful line-rate **software-only** filter blocks invalid packets
  ⇒ System's network stack handles the TCP validation
  ⇒ HAProxy handles the HTTP validation and abuse prevention
- *... and more by the end of the year*

Thanks!

Questions ?