Unbound & FreeBSD

(A story during the last days of November ’2013)
About me:

Pablo Carboni (42), from Buenos Aires, Argentina.

Worked as Unix Admin, DNS Admin, Net Admin, etc, the last 2 decades.

“Passionate” for DNS, FreeBSD, Network, RFC, and developer stuff related.

You can reach me at:

Twitter / Mastodon: @pcarboni / @pcarboni@bsd.network

Linkedin: linkedin.com/in/pcarboni

Disclaimer: This story has intentionally workplaces renamed, and sensitive info deleted.
How does this story start?

Five years ago, at work, while taking KPIs from some DNS hardware appliances, I’ve detected a bottle-neck on CPU usage and queries per second... (HW upgrade was planned in the meantime).

The numbers? **60% cpu usage - 20 kqps per physical box (40 kqps total)**.

(In parallel, - just for “fun” -, I’ve started to test Unbound on lab environment, because some people gave me good comments).

**So, put our hands on the keyboard!**
The tools (lab infrastructure, part #1/2)

- **Hardware**: Dell PowerEdge 1950 double Quadcore (2.0 Gigahertz)
- **OS**: FreeBSD 8.4 RELEASE/AMD64
- **DNS software**: Unbound 1.4.21 [*NLNet labs*], installed from ports directory -tree updated-, compiled with Libevent [*Niels Provos*].
  Just in case, I’ve used Libevent 1.4.14b (proven stable)
- **Measurement tools**: dnstop, from *Measurement factory*. 
The tools (lab infrastructure, part #2/2)

- **Stressing tools:** dnsperf tool, in particular resperf (plus query file sample) [Nominum - Now Akamai]

  *Query files taken from:*

- **A depth-in reading (essential, do not skip)** from the site: https://calomel.org

  (Specially, *Unbound DNS tutorial and FreeBSD Network performance tuning* )

**Note:** The site is highly recommended for several common tasks like fine tuning services, and *BSD OSes.
After FreeBSD was installed, one terminal was opened with `dnstop`. The other terminal was running `resperf`.

Why did I use dnstop?

- It’s a powerful tool for debugging queries and gathering dns stats.
- *When queries quantity was almost the same as the answers, it shows that maximum capacity was not reached (yet).*
- It doesn’t interfere with any DNS service.
- It’s very lightweight, available for several OSes
Why did I use resperf? (Seems that current dnsperf was enhanced)

○ It gave me the maximum QPS allowed by random queries by simulating a cache resolver and increasing queries quantity

○ At that time, it had better(objective) results vs dnsperf.

Note that resperf is an interesting tool for simulating random queries from a with certain maximum desired.
Some sample screenshots (taken from elsewhere)
The masterplan (The end)

- First tests were promising. Without tuning, I've got 10-15kqps
- By following Calomel’s hints about Unbound and FreeBSD, I’ve ended up by doing fine tuning on network card, OS (udp, sockets, ports range, etc), and Unbound config. (However, no DNSSEC was used)
- My dry (but real) tests were incredible: I've got > 54kqps!
- Yes, DNS service -with high load in mind- was under my control! :-)

I've got 10-15kqps
However, no DNSSEC was used
The deployment (The end of this story)

- It should be noted that a rapid deployment for this lab took place because of several factors (including dns performance).
- Final deployment lasted for 6 months; started from 80kqps, ending with 120kqps distributed on 3 physical servers.
- It’s worth to note that the queries were made from subscribers to the internet.

- And yes, the result was incredible!
Acknowledgements

- FreeBD project (https://www.freebsd.org)
- NLNet labs (https://www.nlnetlabs.nl/)
- Nominum (now part of Akamai) (https://www.akamai.com)
- The Measurement Factory (http://dns.measurement-factory.com/tools/)

Special ACKs to Adam Wolk (@mulander) because his search of a guest speaker for this BSD_PL meetup. ;-)
Thank you!