Benchmarking DNS resolvers
using realistic workload

Petr Špaček • petr.spacek@nic.cz • 2019-10-16
Talk outline

- Motivation
- Classic approach
- Classic pitfalls
- DNS Shotgun – tool for realistic benchmarking
Motivation

- Running DNS resolver ⇒ power, cooling
- Power, cooling ⇒ €€€
- Benchmarking ↔ optimization
  - ⇒ cost reduction
DNS resolving recap

- Client
- Resolver
- Cache
- Auth 1
- Auth 2
- Auth 3

- cache hit
- cache miss
Inside of a DNS resolver: Cache hit

- Query parsing
- Cache search
- Answer serialization
Inside of a DNS resolver: Cache miss

- Authoritative server selection – who to ask?
- Retransmit strategy
- DNSSEC validation
- Socket management – reuse? randomization?
- Policy engine
- Cache write & eviction
Classic benchmarking: QPS QPS QPS!

- $ man resperf
- Query list: tcpdump -> text
- Ramp-up query traffic
- Find max QPS
  - Response rate drops
Classic pitfalls 1/2

- No query timing
  - Ignores TTL ⇒ unrealistic cache hit rate
- Text query list
  - EDNS info lost ⇒ unrealistic TCP fallbacks
- QPS ramp-up
  - Waits for cache hit rate increase ⇒ unrealistic
  - Resolver restart!
Classic pitfalls 2/2

- Small # of clients
  - Affects workload distribution
- No fallback to TCP
  - Truncated bit
- No connection management
  - TCP, TLS, DoH!
- Over-focuses on QPS!
DNS Shotgun: Introduction

- New toolset
  - Based on dnsjit by DNS-OARC
  - https://www.dns-oarc.net/tools/dnsjit
- Realistic DNS benchmarking
- Open-source
DNS Shotgun: Client-based approach

- How many clients can the resolver handle?

- Performance depends on clients
  - IoT, mobile, desktop, mail server, ...
DNS Shotgun: Principle

- Phase 1: Analyze traffic patterns in PCAPs
- Phase 2: Simulate $N$ of your clients
DNS Shotgun: Traffic analysis

- Query stream for each IP/DNS client
  - IoT – mobile – desktop – mail server …
  - Beware! NAT!
- Pre-generate test data
  - $N$ clients with $S$ seconds
  - $S = 60$ seconds
  - $N = 100k, 200k, 300k, …, 1M
### DNS Shotgun: 3 => 6 clients – generation

<table>
<thead>
<tr>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client 1</td>
<td>Q11</td>
<td></td>
<td></td>
<td></td>
<td>Q15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>Q21</td>
<td>Q22</td>
<td>Q23</td>
<td>Q24</td>
<td>Q25</td>
<td></td>
<td>Q28</td>
<td></td>
</tr>
<tr>
<td>Client 3</td>
<td>Q31</td>
<td></td>
<td>Q33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Q37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client 1</td>
<td>Q11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 2</td>
<td>Q21</td>
<td>Q22</td>
<td>Q23</td>
<td>Q24</td>
</tr>
<tr>
<td>Client 3</td>
<td>Q31</td>
<td></td>
<td>Q33</td>
<td></td>
</tr>
<tr>
<td>Client 4</td>
<td>Q15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client 5</td>
<td>Q25</td>
<td></td>
<td>Q28</td>
<td></td>
</tr>
<tr>
<td>Client 6</td>
<td></td>
<td></td>
<td>Q37</td>
<td></td>
</tr>
</tbody>
</table>
DNS Shotgun: Client simulation

- Replay pre-generated traffic
- Socket/connection per query/client
- Keep $\pm$ 1 second query timing
  - Realistic cache hit rate
  - $\Rightarrow$ QPS varies over time
- Want higher "QPS"? Add clients!
DNS Shotgun: Performance testing

- Simulate $N$ clients
  - Analyze response rate + RCODEs
  - Monitor resource usage
- Increase $N$
  - ... as long as resolver can keep up
- $N = \text{maximum } \# \text{ of clients}$
  - for given input PCAP & connection parameters
DNS Shotgun: Experiment

- Input: anonymized traffic from a Czech university
- Empty cache
- **Measure response rate over 120 s**
- Monitor NOERROR/NXDOMAIN/SERVFAIL ratios
- Increase # of clients
- 4 CPUs, no qname minimization, same cache params
PowerDNS Recursor 4.2.0: defaults

Response Rate [%]

Time [s]

- 100 kC (38 k QPS)
- 200 kC (79 k QPS)
- 250 kC (100 k QPS)
- 300 kC (122 k QPS)
- 350 kC (145 k QPS)
- 400 kC (168 k QPS)
- 500 kC (209 k QPS)
- 600 kC (236 k QPS)
PowerDNS Recursor 4.2.0: max-mthreads?
PowerDNS Recursor 4.2.0: reuseport?
PowerDNS Recursor 4.2.0: reuseport?

Do not generalize!

Measure it yourself!

Use your traffic capture!
BIND 9.14.6: --tuning=?

![Graph showing response rate over time for different configurations of BIND 9.14.6.](image-url)

- **100 kC default (38 k QPS)**
- **100 kC large (38 k QPS)**
- **160 kC default (62 k QPS)**
- **160 kC large (62 k QPS)**

Response Rate [%] vs Time [s]
BIND 9.14.6: --tuning=default, synth-from-dnssec?
BIND 9.14.6: --tuning=default, synth-from-dnssec?

Do not generalize!

Measure it yourself!

Use your traffic capture!
Knot Resolver 4.2.2 defaults

![Graph showing response rate over time for different KC and QPS values.](image-url)
Knot Resolver 4.2.2 vs. to-be-4.3.0

![Graph comparing Knot Resolver 4.2.2 vs. to-be-4.3.0](image)

- **350 kC v4.2.2 (145 k QPS)**
- **400 kC v4.2.2 (168 k QPS)**
- **400 kC v4.3.0 alpha (168 k QPS)**
- **500 kC v4.3.0 alpha (209 k QPS)**
Knot Resolver 4.2.2 vs. to-be-4.3.0

Do not generalize!
Do not generalize!
Measure it yourself!
Use your traffic capture!
Do not generalize!
Measure it yourself!
Use your traffic capture!
DNS Shotgun: Limitations

- Requires **a lot** of PCAPs
  - 1 hour, 1k clients = 6 minutes, 10k clients (simulated)
- Results depend on input traffic capture
  - ... simulates **your own clients**
- TCP/TLS/DoH not supported *yet*
DNS Shotgun: Try it

- Very much work-in-progress
  - Here be dragons! :-)
- Try it anyway
  - https://gitlab.labs.nic.cz/knot/shotgun
- Sponsors needed!
  - TCP/TLS/DoH support
  - Configurable connection reuse (pipelining, keepalive)
Closing remarks

- DNS micro-benchmarks do not reflect real world
- HW & OS changes invalidate results
- Generalization is hard
  - Compare using *your config* and *your traffic*
- Interested in benchmarking? Get in touch
  - petr.spacek@nic.cz
  - https://gitlab.labs.nic.cz/knot/shotgun