DNS Benchmarking 101: Essentials and Common Pitfalls

Petr Špaček

2024-02-09

pspacek@isc.org

First try: BIND, 1 vs 4 CPU

- 1 CPU thread

```
server$ named -n 1
client$ yes '. A' | dnsperf -l 5
> Queries per second:   24599.530893
```

- 4 CPU thread

```
server$ named -n 4
client$ yes '. A' | dnsperf -l 5  👀
> Queries per second:   29581.661614
```

Stop it!

- We measured
  - *something*
  - *somewhere*
- Result is *a number*
  - … larger the better? 🤔 🙏

How-to

- Test design
  - Resolver ≠ authoritative ⚠
  - Data!
  - Tools
- Test environment validation
- Monitoring
- Evaluation

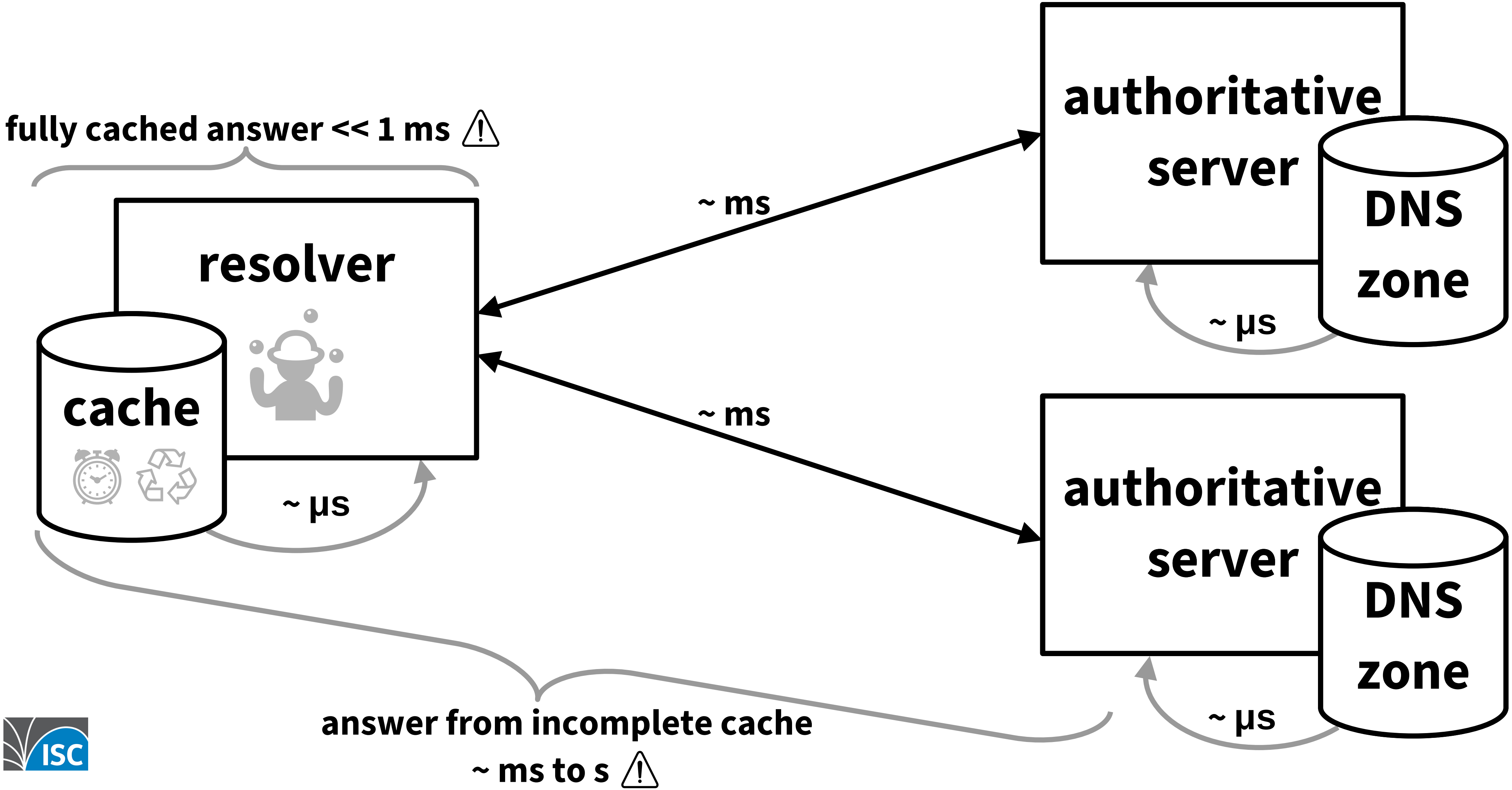
Test design

🍐 ≠ 🍎 ≠ 🍍

Test design

- Resolver ≠ authoritative ⚠

Resolver ≠ authoritative ⚠



Test design: data

- Different queries – different costs 🪙 💰
- Normal operation
  - **Real** traffic samples
  - Resolvers – include **timing** ⚠
- DoS
  - Most expensive queries 💸

Test design

- Resolver ≠ authoritative ⚠
- DoS ≠ normal operations 🐉 ≠ 🐎 ≠ 🐄
- DoS + normal operations ≠ ∑
  - Nonlinear effects
- Management?
  - Zone update, filtering rule updates …

Tools

🐎≠🐄≠🐉

Tool	input	# clients	model	auth	resolver	DoS	RCODE	latency	UDP	TCP	DoT	DoH	DoQ	IPv6
dnsgen	binary	< 48 000	🔫/⚖	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗
dnsmeter	text, PCAP	∞	🔫	✓	✗	✓	✓	😞	✓	✗	✗	✗	✗	✗
dnsperf	text, binary	~ 1000	⚖	✓	✗	😞	✓	✓	✓	😞	😞	😞	✗	✓
flamethrower	gener, text	~ 1000	🔫	😞	✗	✓	✓	😞	✓	✓	✓	😞	✗	✗
kxdpgun	text	∞^*	🔫	✓	✗	✓	✓	✗	✓	✓	✗	✗	✓	✓
resperf	text	< 65 535	⚖	✗	✗	✓	✓	😞	✓	😞	😞	😞	✗	✓
shotgun	PCAP	~ 1 M	🔫	✗	✓	✗	✓	✓	✓	✓	✓	✓	🏗	✓

😞 supported but … don't use it

⚖ answers affect request stream

🔫 answers do not affect request stream

Tools: authoritative servers

- dnsperf
  - easy to use
  - incremental results 🔬
  - latency measurement ⏱
- kxdpgun
  - extreme throughput 💪
  - suitable (also) for DoS

Tools: resolvers

- ~~resperf~~
  - Avoid it – wrong methodology☣
- shotgun
  - ~ only realistic option
  - See RIPE 79 DNS WG:
    Benchmarking and Optimizing DNS Resolvers
    on the ISP level

Test environment validation

🔬👀🤬

Validation #1

- Results …
  - 24 vs 29 kQPS
  - 1 vs 4 CPU
    - mere + 20 % QPS
    - "BIND does not scale!"
- Really?

Validation: Echo server, 1 vs 4 CPU

- 1 process

```
server$ dumdumd -r -R 53
client$ yes '. A' | dnsperf -l 5
> Queries per second:   25148.146148
```

- 4 processes

```
server$ for _ in $(seq 1 4)
        do dumdumd -r -R 53 & done
client$ yes '. A' | dnsperf -l 5
> Queries per second:   29717.900739
```

⅓ 🤔

Validation #2

- `server$ top -H`

```
 PID %CPU  COMMAND
 721 57.9  dumdumd
   1  0.0  systemd
```

🤔

- `server$ tcpdump -n`

```
IP6 2600:...0.37276 > 2600:...1.53: 0+ A? .
IP6 2600:...0.37276 > 2600:...1.53: 1+ A? .
IP6 2600:...0.37276 > 2600:...1.53: 2+ A? .
```

💡

- `$ sudo ethtool -k ens5 | grep hash`

```
receive-hashing: on
```

Validation #3

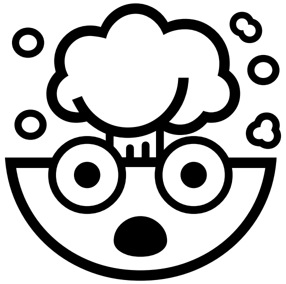
-
  ```
  client$ yes '. A' | dnsperf -l 5 -c 128
  Queries per second:   29843.365414
  ```

-
  ```
  server$ tcpdump -n
  ```

  ```
  IP6 2600:...0.47786 > 2600:...1.53: 0+ A? .
  IP6 2600:...0.58158 > 2600:...1.53: 1+ A? .
  IP6 2600:...0.34970 > 2600:...1.53: 2+ A? .
  ```

-
  ```
  server$ top -H
  ```

  ```
   PID %CPU  COMMAND
   801 15.7  dumdumd
   799 15.8  dumdumd
   800 14.9  dumdumd
   802 13.9  dumdumd
  ```



Validation #4

-
  ```
  client$ sudo dmesg
  …
  ------------[ cut here ]------------
  WARNING: CPU: 0 PID: 140231 …
  RSP: 002b:00007f2e781cb060 …
  RAX: ffffffffffffffda …
  RDX: 000000000000002c …
  …
  ```

- Network driver bug!
  - OS kernel update follows …

Validation #5: new kernel

- 1 process

```
server$ dumdumd -r -R 53
client$ yes '. A' | dnsperf -l 5 -c 128
> Queries per second:   157201.098311
```

- server$ top -H

```
 PID %CPU   COMMAND
 846 99.0   dumdumd
       ✔
```

😌

Validation #6: new kernel

- **4 processes**

```
server$ for _ in $(seq 1 4)
            do dumdumd -r -R 53 & done
client$ yes '. A' | dnsperf -l 5 -c 128
> Queries per second:   312712.494497
```

½🤔

- server$ top -H

```
 PID %CPU  COMMAND
 852 84.0  dumdumd
 851 73.0  dumdumd
 853 72.0  dumdumd
 854 64.0  dumdumd
```

⚠

Validation #7: more ports & threads

- **4 server processes, 2x2 client threads**

```
server$ for _ in $(seq 1 4)
        do dumdumd -r -R 53 & done
client$ yes '. A' | dnsperf -l 5 -c 256 -T2
> Queries per second:   338916.274148
```

- `client$ top -H`

```
  PID %CPU  COMMAND
 1961 94.0  perf-recv-0001
 1959 92.0  perf-recv-0000
 1960 57.0  perf-send-0000
 1962 57.0  perf-send-0001
```

½ 🤔

⚠

Validation #8: different client

- **4 server processes**

```
client$ echo '. A' > query.list
client$ sudo kxdpgun -t 5 -Q 600000 -i query.list
   total queries:     3000030 (600006 pps)
   total replies:     1933893 (386778 pps)
```

- server$ top -H

```
  PID   %CPU   COMMAND
 1005   82.2   dumdumd
 1006   82.2   dumdumd
 1004   81.2   dumdumd
 1007   81.2   dumdumd
```

⅔🤔

Validation #9: queues

-
  ```
  $ sudo ethtool -l ens5
  Channel parameters for ens5: …
  Current hardware settings:
  RX:   n/a
  TX:   n/a
  Other:    n/a
  Combined: 8
  ```

-
  ```
  $ sudo ethtool -L ens5 combined 4
  ```

Validation #10: queues

- **4 procesy**

```
client$ echo '. A' > query.list
client$ sudo kxdpgun -t 5 -Q 600000 -i query.list
   total queries:      3000040 (600008 pps)
   total replies:      2808476 (561695 pps)
```

- server$ top -H

```
  PID   %CPU   COMMAND
 1005   99.9   dumdumd
 1006   99.9   dumdumd
 1007   99.9   dumdumd
 1004   95.0   dumdumd
```

↑

~ 93 %

🤨

Takeaways

- Test environment validation 🔬👀🤬
- Resolver ≠ authoritative ⚠
- DoS + normal operation ≠ ∑
- Beyond QPS
  - Outliers 🔬🤯
  - See DNS-OARC 40:
    Detecting latency spikes in DNS server implementation(s)

Thank you!

- Main website: https://www.isc.org
- Software downloads: https://www.isc.org/download or https://downloads.isc.org
- Presentations: https://www.isc.org/presentations
- Main GitLab: https://gitlab.isc.org