DNS Survival Guide

Artyom Gavrichenkov <ag@qrator.net>
A bit of a history: **DNS**

1983:

```c
(int32)*host_str;
```
A bit of a history: **DNS**

1983:

```c
(int32)*host_str;
```

1997-2017:

- load balancing
- geobalancing
- ASN policies
A bit of a history: DNS

1983:
(int32)*host_str;

1997-2017:
• load balancing
• geobalancing
• ASN policies
• failover
• EDNS0
A bit of a history: **DNS**

1983:

`(int32)*host_str;`

1997-2017:

- load balancing
- geobalancing
- ASN policies
- failover
- EDNS0
- AAAA
- DNSSEC
- DANE, CAA, ...
Problem statement

How should an Internet company maintain its DNS infrastructure?

- In-house?
- Outsourcing?
Problem statement

How should an Internet company maintain its DNS infrastructure?

• In-house
  • How to choose a software product?

• Outsourcing
  • How to choose a service provider?
1. How to choose a software product?

Naïve approach:

a) It must be scalable

b) It should support *features*
DNS benchmarks, 2013

- Knot (1.2.0 & 1.3.0-RC5)
- Yadifa (1.0.2)
- NSD3 (3.2.15)
- NSD4 (4.0.0b4)
- PowerDNS (3.3)
- TinyDNS (1.05)
- Unbound (1.4.16)
- Pdnsd (1.2.8)

- Server:
  Dual Xeon E5-2670
  32Gb RAM DDR3 1333Mhz
  Intel X520-DA2 10Gbit

- Generator:
  Single Xeon E5-2670
  32Gb RAM DDR3 1333Mhz
  Intel X520-DA2 10Gbit

- Gentoo Linux 3.7.9
DNS benchmarks, 2013. Setup

• Vanilla DNS software!
• Purpose:
  purely academic (who runs better codebase)
• Authoritative:
  300 zones
• Caching:
  Same amount of data in cache
DNS benchmarks, 2013.

https://www.slideshare.net/ximaera/dns-server-benchmarking
DNS benchmarks, 2013.

https://www.slideshare.net/ximaera/dns-server-benchmarking

...WAIT. Where’s BIND?
DNS benchmarks, 2017
DNS benchmarks, 2017

Queries, K/s

Responses, K/s

OH HERE IT IS
Response Rate – Percentage

Linux 4.10.0, Hosting (1M), Intel X710 10GbE, (2017–08–26)

- Without DNSSEC

Response Rate [%]
- BIND 9.12.0-pre-alpha
- BIND 9.11.2
- Knot DNS 2.5.3
- NSD 4.1.17
- PowerDNS 4.0.4

Queries per second:
- 0 500k
- 1 000k
- 1 500k
- 2 000k
- 2 500k

---
[Image 0x11 to 793x540]
Response Rate – Percentage
Linux 4.10.0, Hosting (1M) DNSSEC, Intel X710 10GbE, (2017–08–27)
This is not good.

The de-facto standard software doesn’t scale well
This is not good.

The de-facto standard software doesn’t scale well

• Yes, a balancer (Nginx) with a soccer field full of BIND servers will do.
• Definite overkill for a small task
This is not good.

The de-facto standard software doesn’t scale well

What scales well causes concern in other areas

- Maintainability?
- Reliability?
- Support?
- Backward compatibility?
- Patches and security?
- Features?
Back to the requirements.

Naïve approach:

a) It must be scalable – how scalable?

b) It should support *features* – what features do we really want?
DNS lookup
DNS lookup

ximaera@nostromo:~$ sudo tcpdump -qni any tcp > /dev/null
 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
 listening on any, link type LINUX_SLL (Linux cooked), capture size 65535 bytes
 ^C
 792 packets captured
 794 packets received by filter
 0 packets dropped by kernel

ximaera@nostromo:~$ sudo tcpdump -qni any port 53 > /dev/null
 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
 listening on any, link type LINUX_SLL (Linux cooked), capture size 65535 bytes
 ^C
 104 packets captured
 156 packets received by filter
 0 packets dropped by kernel

ximaera@nostromo:~$
### DNS lookup

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Action</th>
<th>Response</th>
<th>Status</th>
<th>Additional Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00:34.510826 IP</td>
<td>192.168.1.5.63097 &gt; 8.8.8.8.53:9508+</td>
<td>A? facebook.com.</td>
<td>(30)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00:34.588632 IP</td>
<td>8.8.8.8.53 &gt; 192.168.1.5.63097:9508 1/0/0</td>
<td>facebook.com. A</td>
<td>31.13.72.36</td>
<td>(45)</td>
<td></td>
</tr>
</tbody>
</table>
DNS lookup

• Apparently, not rocket science?

• Well, it’s not – for the \((\text{int32})*\text{host\_str}\) feature.
More to it?

- Geobalancing
MaxMind GeoIP database

GeolP2 City Database Demo

IP Addresses

8.8.8.8

Enter up to 25 IP addresses separated by spaces or commas. You can also test your own IP address.

Submit

GeolP2 City Results

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Country Code</th>
<th>Location</th>
<th>Postal Code</th>
<th>Approximate Coordinates*</th>
<th>Accuracy Radius</th>
<th>ISP</th>
<th>Organization</th>
<th>Domain</th>
<th>Metro Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.8.8.8</td>
<td>US</td>
<td>United States, North America</td>
<td></td>
<td>37.751, -97.822</td>
<td>1000</td>
<td>Google</td>
<td>Google</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MaxMind GeoIP database

### GeoIP2 City Database Demo

**IP Addresses**

8.8.8.8

Enter up to 25 IP addresses separated by spaces or commas. You can also test your own IP address.

**Submit**

---

### GeoIP2 City Results

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Country Code</th>
<th>Location</th>
<th>Postal Code</th>
<th>Approximate Coordinates</th>
<th>Accuracy Radius</th>
<th>ISP</th>
<th>Organization</th>
<th>Domain</th>
<th>Metro Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.8.8.8</td>
<td>US</td>
<td>United States, North America</td>
<td></td>
<td>37.751, -97.822</td>
<td>1000</td>
<td>Google</td>
<td>Google</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Sorry, this is wrong!**
MaxMind GeoIP database

Has its “owner location vs actual location” dilemma. Generally unreliable for anything except statistics.

MaxMind GeoIP database

Has its “owner location vs actual location” dilemma. Generally unreliable for anything except statistics.

• There’s no geography on the Internet, just network topology.

• There are no countries, just autonomous systems and their relations.
ASN and prefix targeting: example

https://ns1.com/solutions/technical-solutions/filter-chain

- Filters are like little programs that run inline for every DNS query.
- They are attached directly to RFC-compliant DNS records

<table>
<thead>
<tr>
<th>NETFENCE ASN</th>
<th>Restrict to answers where Autonomous System (AS) of requester IP matches AS list</th>
</tr>
</thead>
<tbody>
<tr>
<td>NETFENCE PREFIX</td>
<td>Restrict to answers where requester IP matches prefix list</td>
</tr>
</tbody>
</table>
Contemporary DNS server requirements

• Latency reduction: geobalancing *prefix targeting*
Dynamic configuration

08:35 < j***k> and also VERY FUNNY PEOPLE
08:35 < m***k> j***k likes us \o/
08:35 < s****k> we like j***k
08:36 < d****n> DNS DEPLOY
08:36 < j***v> What is this "DNS DEPLOY" thingy you guys keep screaming?
08:36 < d****n> j***v, when we deploy new dns content
08:36 < j***k> http://i.qkme.me/364h55.jpg
08:36 < j***v> Alerarting eachother?
08:36 < d****n> yup
08:37 < j***v> d****n: Why?
08:37 < d****n> in case there's problems and I guess also as a locking mechanism

Dynamic configuration

DNS is not a static config anymore, this is essentially an API for configuration management systems and applications:

- Provisioning
- Stats
- Policy management

Enterprises will want this sooner or later. Treating DNS not as an API is error-prone.
Contemporary DNS server requirements

• Latency reduction: geobalancing, prefix targeting
• Dynamic configuration
• Failover
Failover, TTL 120s
Contemporary DNS server requirements

- Latency reduction: geobalancing \texttt{prefix targeting}
- Dynamic configuration
- Failover
- Vulnerability intelligence
- DDoS attacks
DNS DDoS

- Volumetric attacks: effective line rate challenges/handshake
- Water Torture and so on: query analysis, statistics and blacklists
- Anycast is necessary
Contemporary DNS server requirements

• Latency reduction: geobalancing prefix targeting
• Dynamic configuration
• Failover
• DDoS attacks
• DNSSEC, TLS, etc. More than 180 RFCs
Contemporary DNS server requirements

• Latency reduction: geobalancing prefix targeting
• Dynamic configuration
• Failover
• DDoS attacks
• DNSSEC, TLS, etc. More than 180 RFCs

Okay, now this is rocket science 😞
What about service providers?

Thousands out there!

• Dyn
• NS1
• Route 53
• Name.com
• Azure DNS
• Google Cloud DNS
• Cloudflare
• ... (sorry for not putting your favorite provider in the list)
What about service providers?

Thousands out there!

• How to choose?
What about service providers?

Thousands out there!
• How to choose?
• Well, why?
SRTT: Smoothed Round Trip Time

- A mechanism intended to help to run a lot of nameservers simultaneously for a zone
- Deployed in most SOHO and enterprise networks
- NS1 study suggests up to 90% Internet traffic being serviced by SRTT-enabled resolvers
“Boxplot”
SRTT
SRTT

https://blog.serverfault.com/2017/01/09/surviving-the-next-dns-attack/
SRTT

R53

R53 + Google
How to choose a service provider

• The more you have, the better
  • Up to 4-6 will be fine
• Easy to compare and replace the underperforming ones
• Helps also with maintenance windows and downtime issues
• AXFR doesn’t support a lot of features
  • Prefer providers with nice API
Q&A