KubeCon | CloudNativeCon

North America 2018

Switching the Engine

CoreDNS ♡ Kubernetes

KubeCon | CloudNativeCon

North America 2018

Jake Sanders

Contributor: Kubernetes

SWE: GKE Security

@dekkagaijin

Michael Grosser

Maintainer: CoreDNS | Contributor: K8s

Founder: Okkur Labs & Rekkur Solutions

@stp_ip @stp-ip

Agenda

- Background
- NodeLocal Caching
- Local Benchmarks
- Cluster Benchmarks
- e2e Application Benchmarks
- Outlook
- Questions

Background

Kube-dns History

- Bundle of dnsmasq + SkyDNS
- SkyDNS written in Go by Miek Gieben and others*
- Partly maintained

*Erik st. Martin, Brian Ketelsen, Michael Crosby

CoreDNS History

- Authored by Miek Gieben
- Based on Caddy (Golang webserver)
- Plugin-based architecture for extensibility++
- GA in 1.11, default in 1.13

Protocol support

- UDP(RFCs: 1034, 1035, etc.)
- TCP (RFC 7766)
- TLS (DoT) (RFCs: 7858, 8310)
- HTTPS (DoH) (RFC 8484)
- GRPC (DoG?) (RFCs: none)

NodeLocal caching

Huge thanks:

Pavithra Ramesh, GKE Networking

Internet Considered Harmful

“Aaaand it’s gone”
- South Park

- Standards from 70s & 80s
- Assumption of most DNS records being ~static
- Congestion + availability > consistency + reliability
- Old decisions can’t keep up w/ new usage patterns

DNS 1.0 (RFCs: 1034, 1035)

- Requests generally occur over **UDP**, except under special circumstances. Per RFC1035 4.2:
    - "The DNS assumes that messages will be transmitted as datagrams (UDP) or in a byte stream carried by a virtual circuit (TCP). While virtual circuits can be used for any DNS activity, **datagrams are preferred for queries due to their lower overhead and better performance.**"
    - "Depending on how well connected the client is to its expected servers, the minimum retransmission interval should be **2-5 seconds**."

/etc/resolv.conf, <resolv.h>

timeout:n

Sets the amount of time the resolver will wait for a response from a remote name server before retrying the query via a different name server … Measured in **seconds**, the default is RES_TIMEOUT (**currently 5**, see <resolv.h>) ...

attempts:n

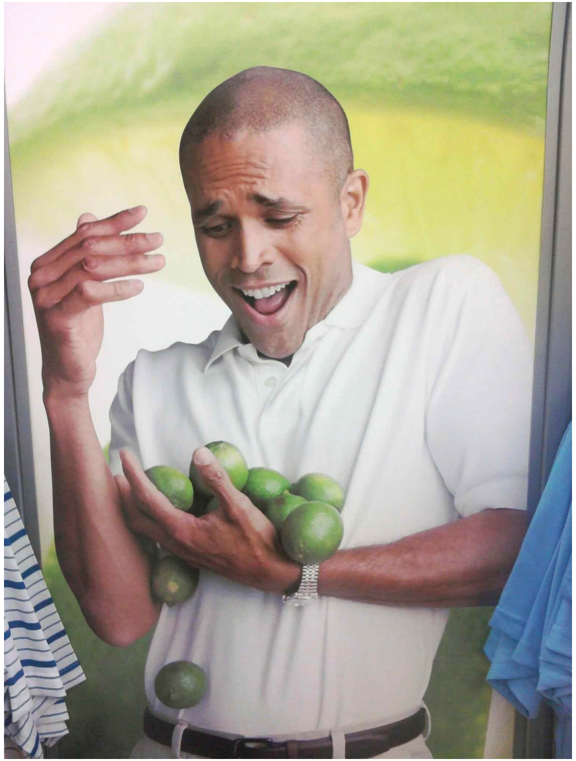
Sets the number of times the resolver will send a query to its name servers before giving up ... The default is RES_DFLRETRY (**currently 2**, see <resolv.h>).

conntrack limits & races

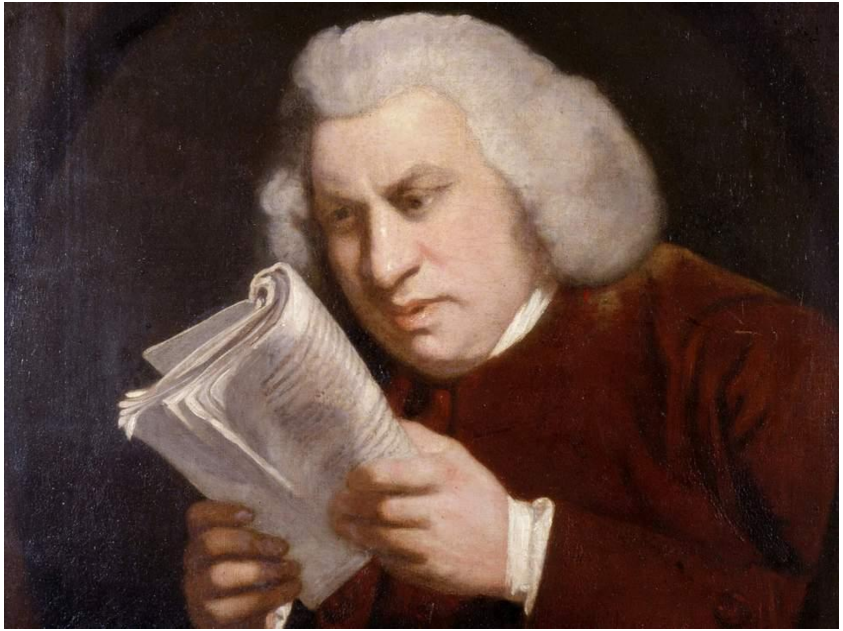
- Cluster DNS is a k8s **Service**
- DNAT rules used to translate ClusterIP to Pod IP
- conntrack table usually limited to **65536 entries**
  => **dropped packets**
- Multiple conntrack table entries per 'connection' (including UDP)
  - No UDP 'close' → entries persist long after they're useful

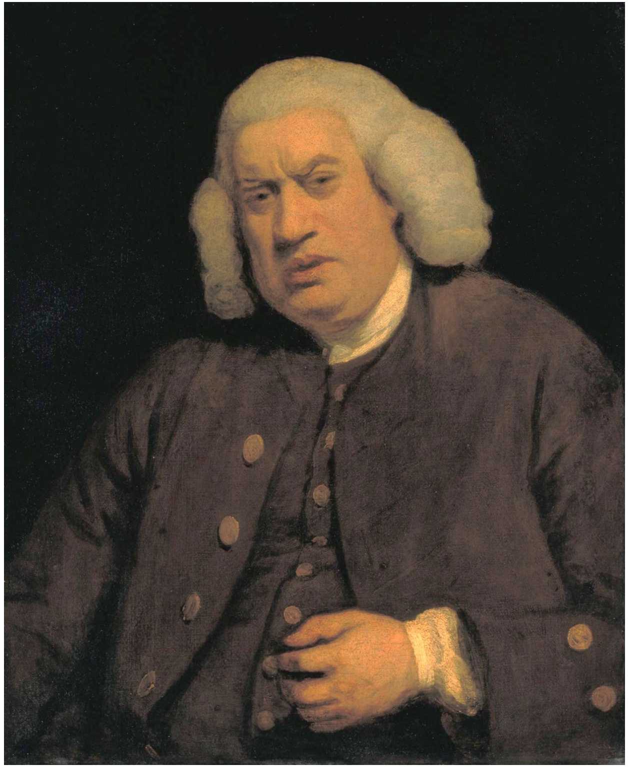
conntrack limits & races

- multiple UDP reqs from the same ip:port can results in **race conditions**
  => **dropped packets**
- Races aggravated by **parallelized** reqs for different record types (e.g. A, AAAA)
- N search paths mean **N times** more requests for failed queries



We’ve All Been There





nf_conntrack: table full, dropping packet

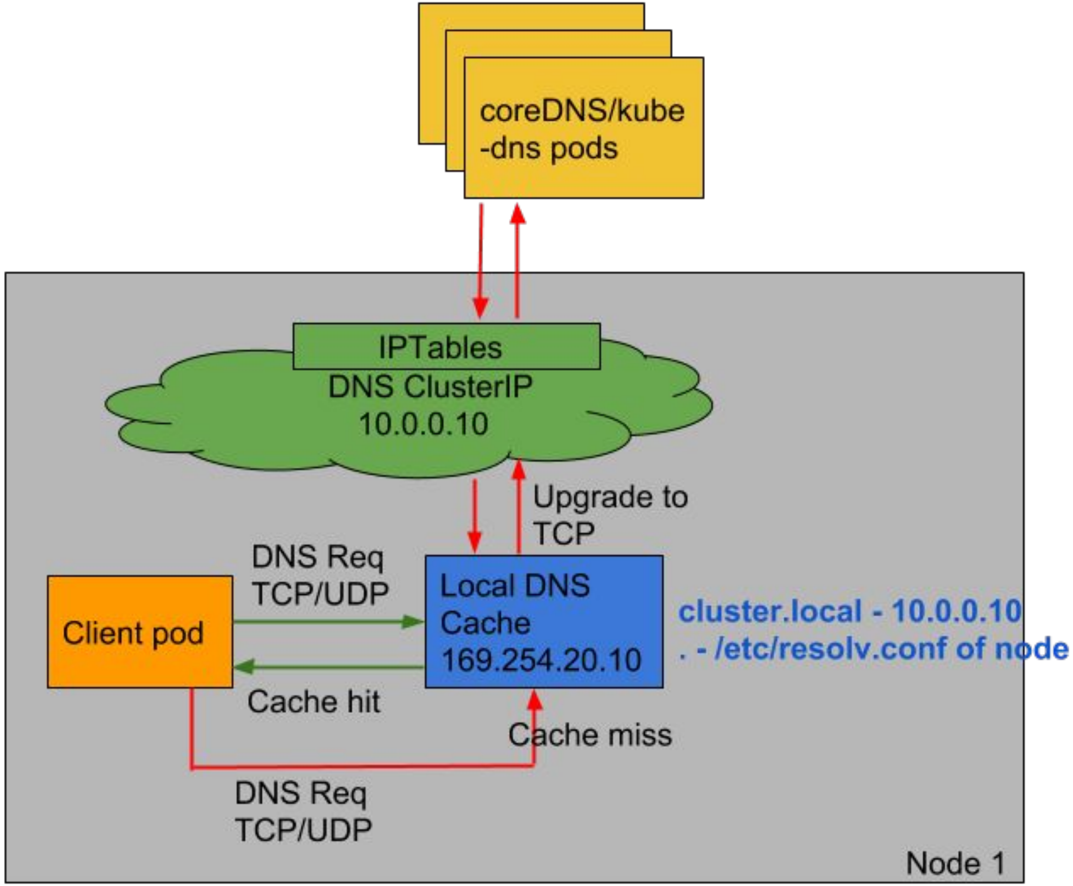
Enter: NodeLocal DNS Cache

Coming soon (optionally) to a 1.13 cluster near you!

Runs on every node, serves DNS for pods that are using cluster DNS.

- Improve latency by reducing communication over the network
- **Skips `conntrack`** for pod-cache connection
  - Less dropped packets!
- Proxy queries over TCP (and preserves the connection)
  - DARPA-grade reliability & consistency!
  - Even less pressure on the cluster DNS's conntrack tables (see above)
- Node-level DNS metrics

Special thanks: Pavithra Ramesh, GKE

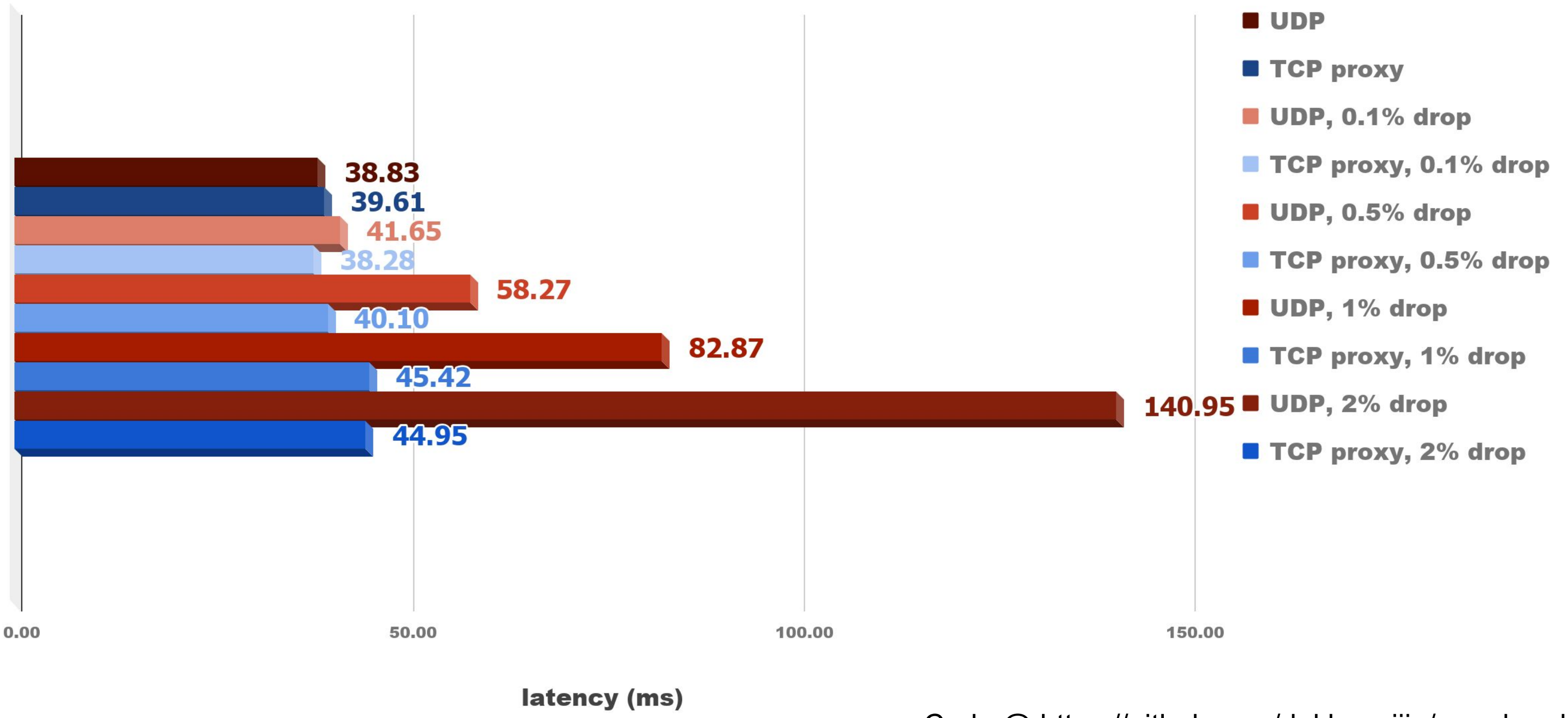

coreDNS/kube -dns pods
IPTables
DNS ClusterIP
10.0.0.10
Upgrade to TCP
DNS Req TCP/UDP
Local DNS Cache
169.254.20.10
cluster.local - 10.0.0.10
. - /etc/resolv.conf of node
Client pod
Cache hit
Cache miss
DNS Req TCP/UDP
Node 1

KubeCon | CloudNativeCon

North America 2018

Local Benchmarks

localhost -> 8.8.8.8 (mean latency)

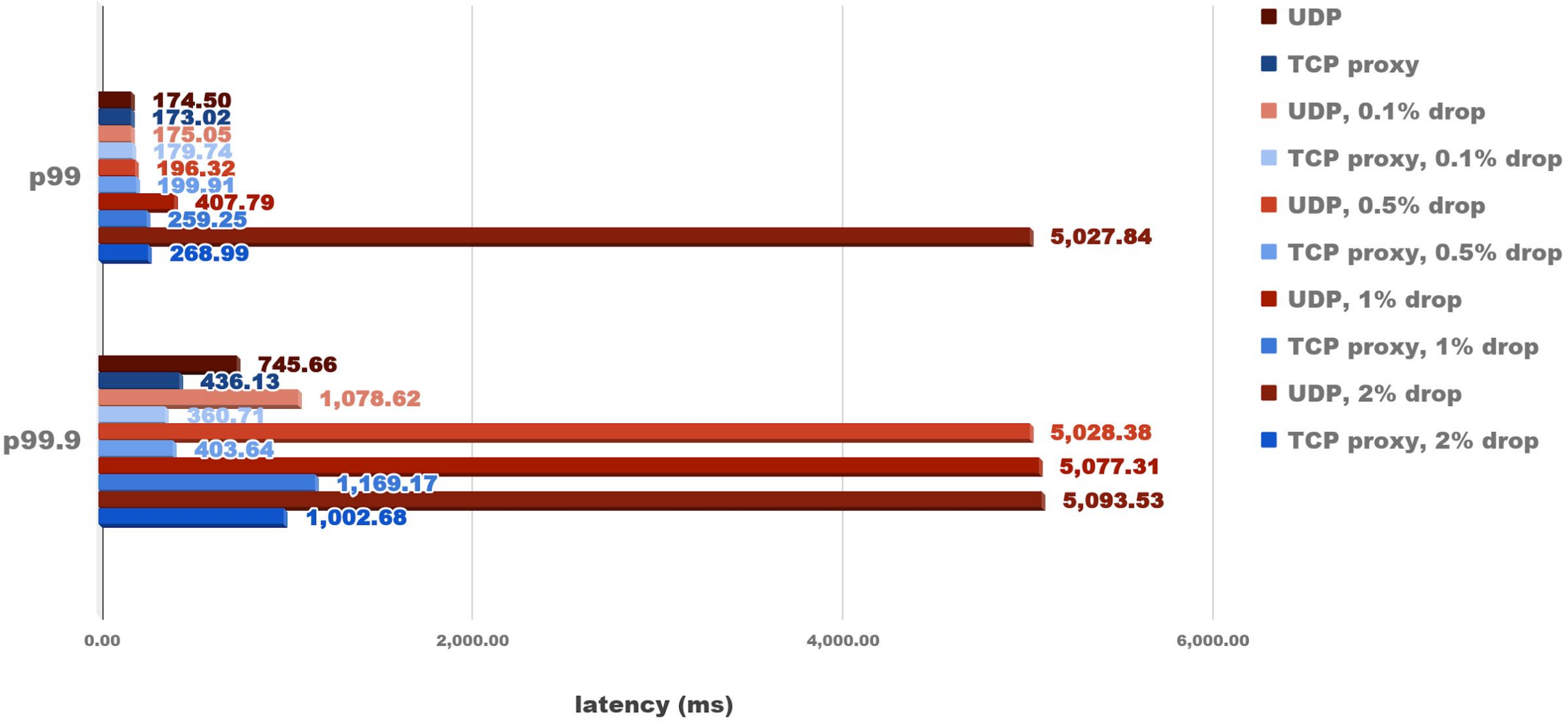


Code @ https://github.com/dekkagaijin/coredns-demo/

localhost -> 8.8.8.8 (tail latency)

KubeCon | CloudNativeCon

North America 2018



KubeCon | CloudNativeCon

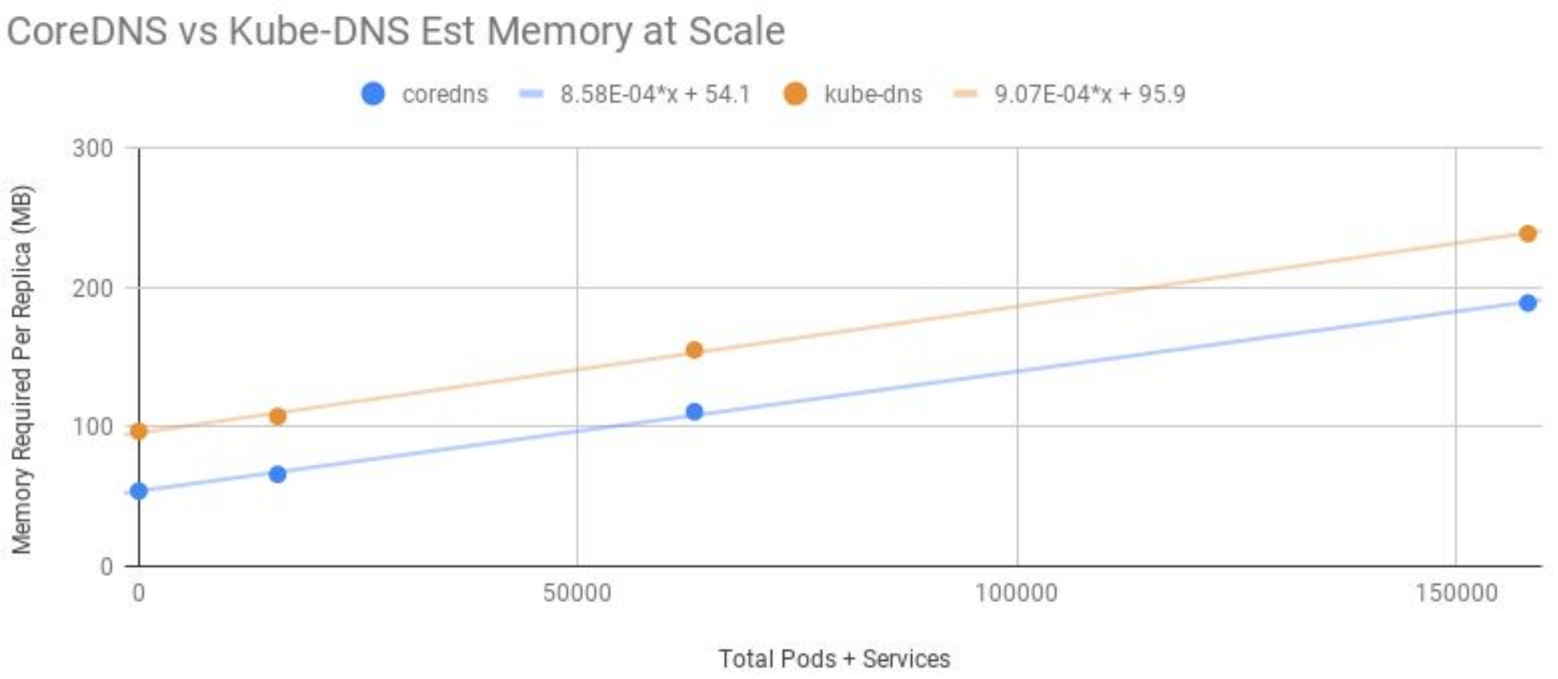
North America 2018

Cluster DNS Benchmarking

Huge thanks:

Chris O'Haver, Infoblox

CoreDNS vs Kube-DNS: Memory



Credit: Chris O'Haver, Infoblox

CoreDNS vs Kube-DNS: Queries

DNS Server	Query Type	QPS	Avg Latency (ms)
CoreDNS	external	6733	12.02
CoreDNS	internal	33669	2.608
Kube-dns	external	2227	41.585
Kube-dns	internal	36648	2.639

Credit: Chris O'Haver, Infoblox

KubeCon | CloudNativeCon

North America 2018

Application Benchmarks

Our application: TXTDirect

- DNS TXT record-based redirects
- Control over your entrypoint and data
- Open Source based on Caddy
- Does a lot of DNS requests

TXTDirect: Request flow

“GET” kubernetes.opensourcesoftware.rocks

“A/AAAA/CNAME” for **kubernetes.opensourcesoftware.rocks**

TXT for **_redirect.kubernetes.opensourcesoftware.rocks**

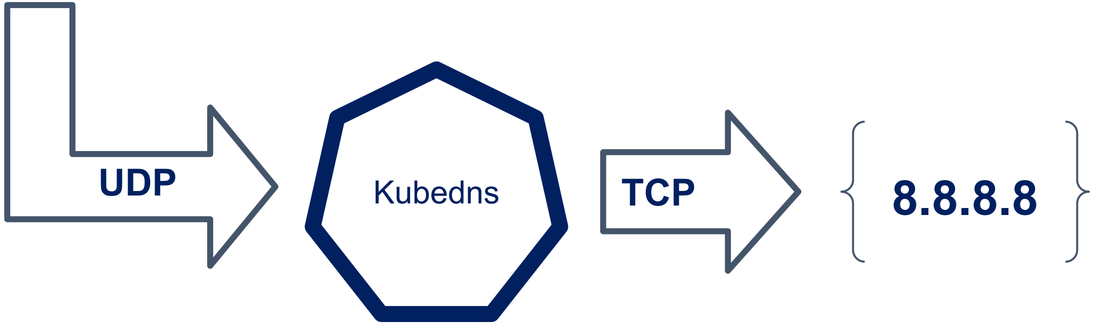
“v=txtv0;type=host;to=https://kubernetes.io"

⇨ **kubernetes.io**

Learn more on txtdirect.org

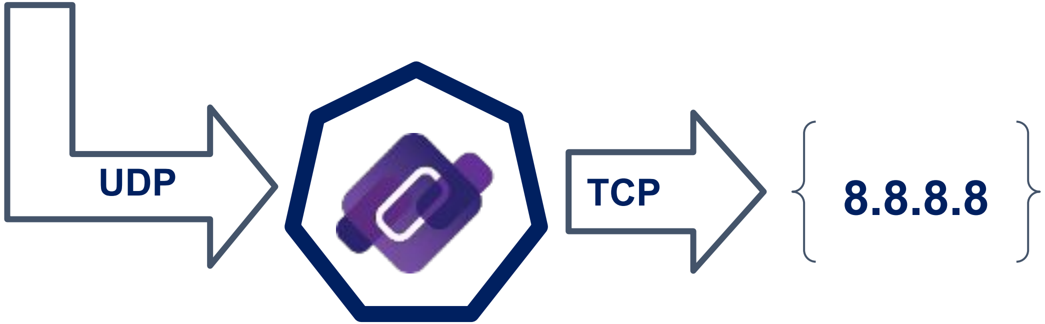
Setup: Standard Kube-dns

TEXTDirect



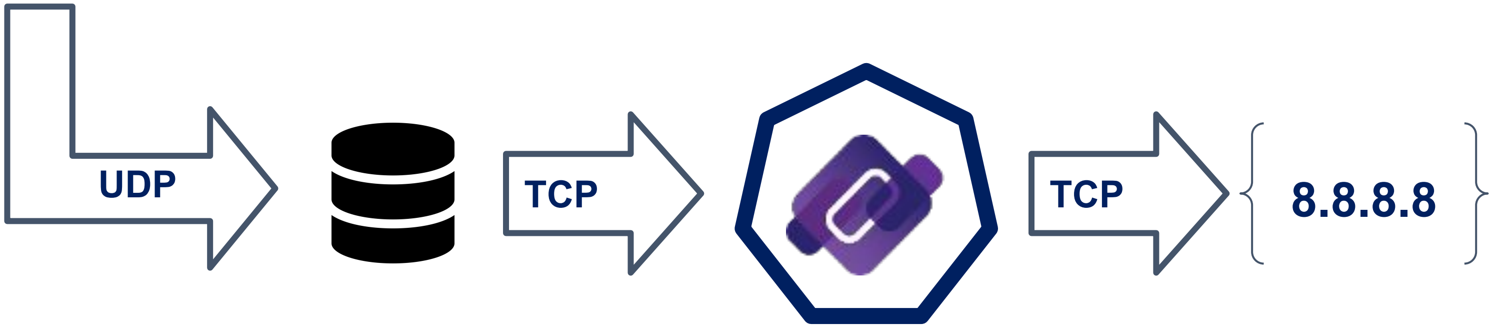
Pod, UDP → Kube-dns

Setup: Standard CoreDNS



Pod, UDP → CoreDNS, TCP → 8.8.8.8

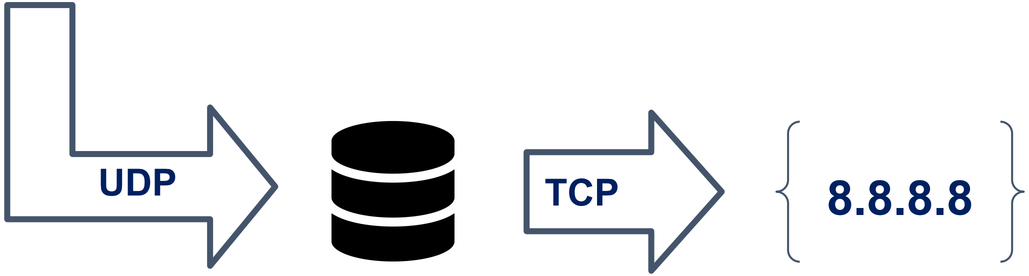
Setup: NodeLocal cluster



Pod, UDP → NL, TCP → CoreDNS, TCP → 8.8.8.8

Setup: NodeLocal direct

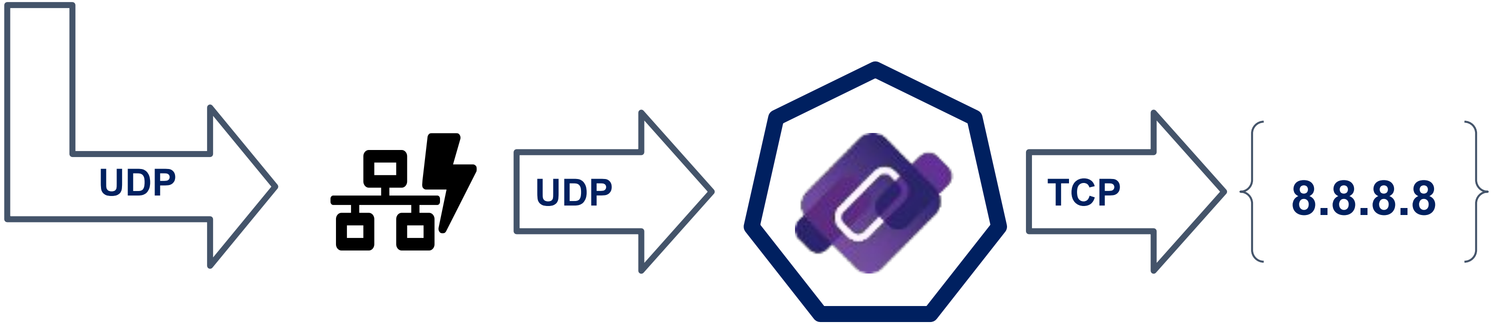
TEXTDirect



Pod, UDP → NL, TCP → 8.8.8.8

Setup: Inducing Chaos

TEXTDirect



UDP

UDP

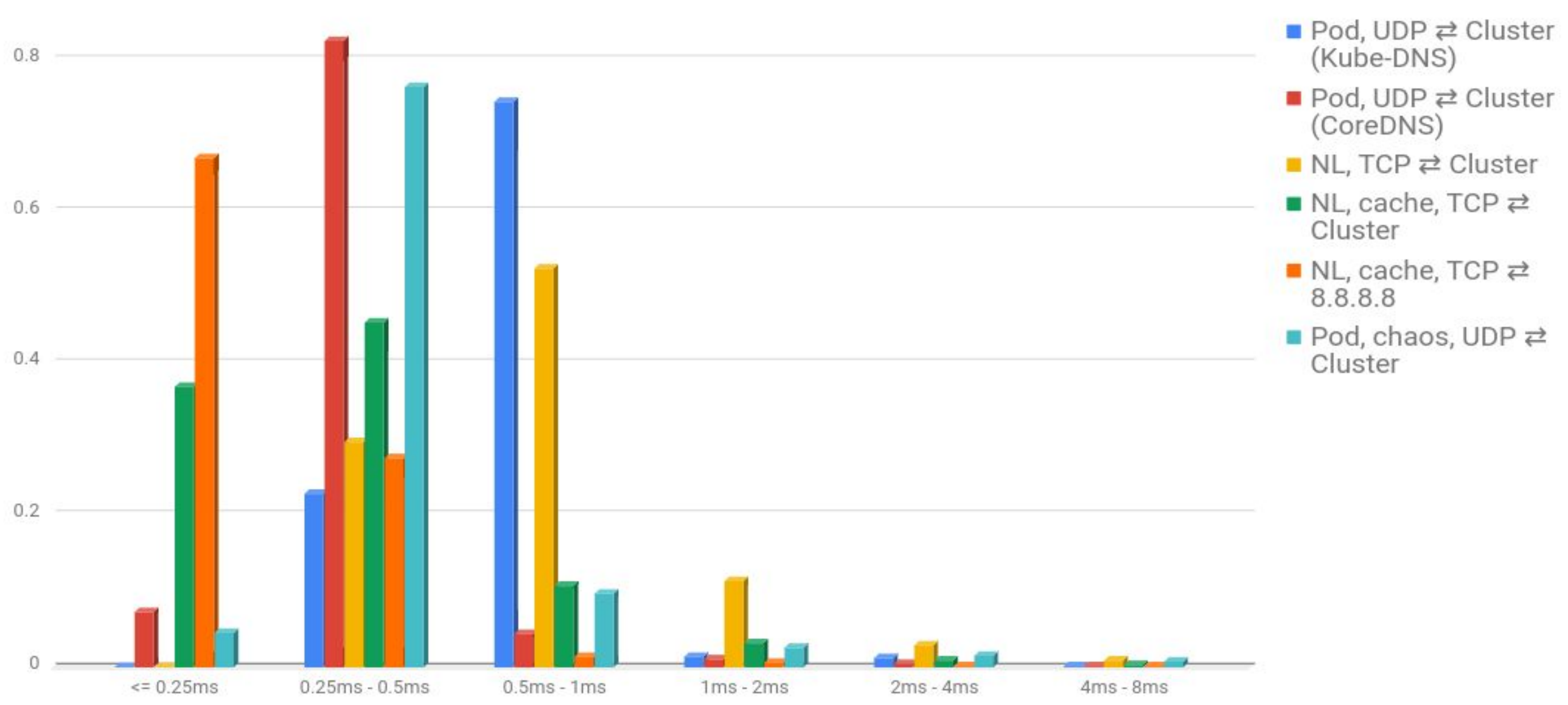
TCP

8.8.8.8

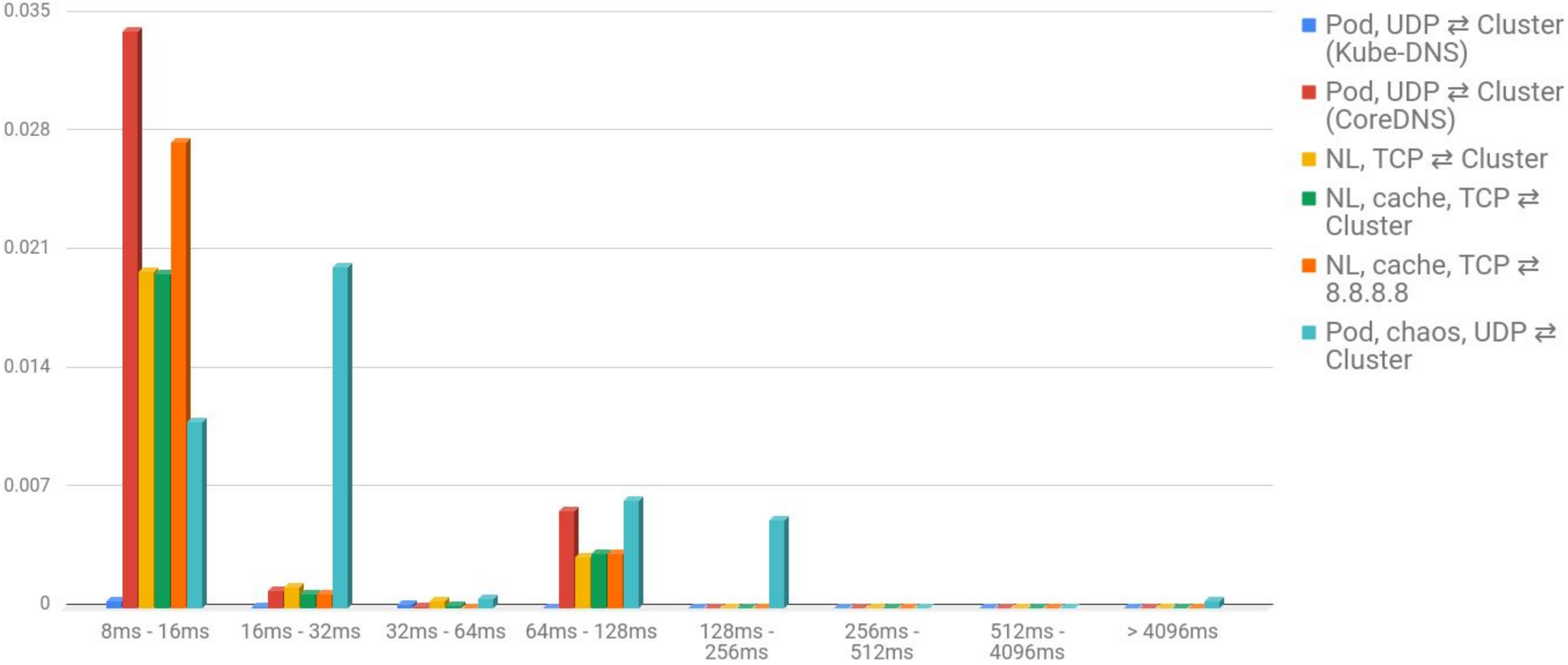
Pod, chaos, UDP → CoreDNS

Show me the numbers!

Probability Distribution (Latency)



Probability Distribution (Tail Latency)



What we learned

- TCP does what it’s supposed to
- TCP forwarding improves the performance of traditional (UDP) clients, with less variance, and without incurring a ton of overhead
- CoreDNS’s plugins make it a good fit for special use-cases

KubeCon | CloudNativeCon

North America 2018

Outlook

Future Work

- Native DNS over GRPC
- Watch based DNS records
- Performance and reliability improvements
- Ideas? Let us know after the talk!

Q&A

Thank You!

Michael Grosser

@stp_ip

[stp-ip.net](stp-ip.net)

Jake Sanders

@dekkagaijin

jsand@google.com

Drinks...