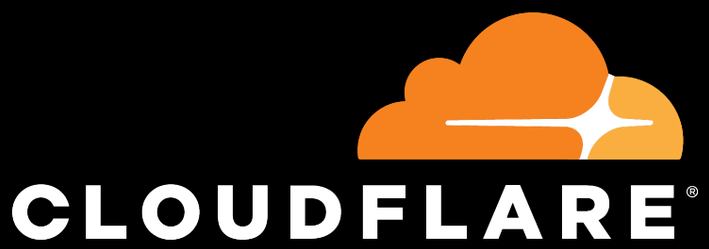


How we analyze over O(100B) DNS requests a day

Marek Vavruša & Ólafur Guðmundsson

Data Analytics at scale is hard

Data Analytics at scale is hard

Cloudflare has tried 5 times to do DNS analytics and failed 4 times

CLOUDFLARE

Data Analytics at scale is hard

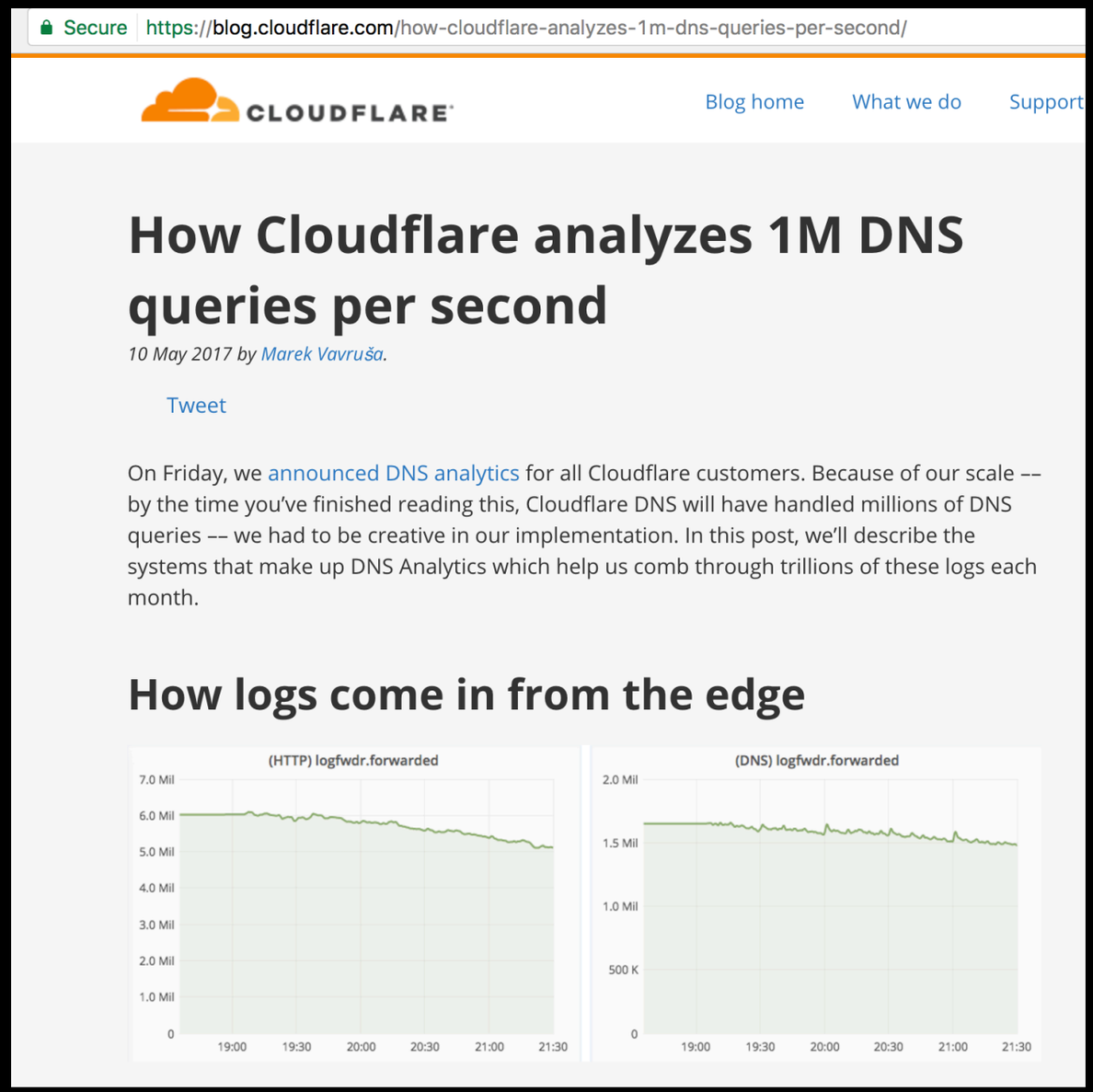
Cloudflare has tried 5 times to do DNS analytics and failed 4 times

[This is about the success story]

Data Analytics at scale is hard

Cloudflare has tried 5 times to do DNS analytics and failed 4 times

[This is about the success story]



CLOUDFLARE

Scale, resolution, cost

Scale, resolution, cost

O(M) queries/sec,
O(K) metals,
O(100) sites,
O(10) weeks data retention

Scale, resolution, cost

O(M) queries/sec,
O(K) metals,
O(100) sites,
O(10) weeks data retention

CLOUDFLARE

Scale, resolution, cost

Scale, resolution, cost

Fast lookups

Scale, resolution, cost

Fast lookups
Analyze at query level

Scale, resolution, cost

Fast lookups
Analyze at query level
Do not make assumptions about analytics

Scale, resolution, cost

Fast lookups
Analyze at query level
Do not make assumptions about analytics

Easy to integrate different viewing “platforms”

Scale, resolution, cost

Fast lookups
Analyze at query level
Do not make assumptions about analytics
Easy to integrate different viewing “platforms”
Must not be DNS specific

Scale, resolution, cost

Scale, resolution, cost

To meet all goals with prior solutions required lots of hardware

Scale, resolution, cost

To meet all goals with prior solutions required lots of hardware

OR

CLOUDFLARE

Scale, resolution, cost

To meet all goals with prior solutions required lots of hardware

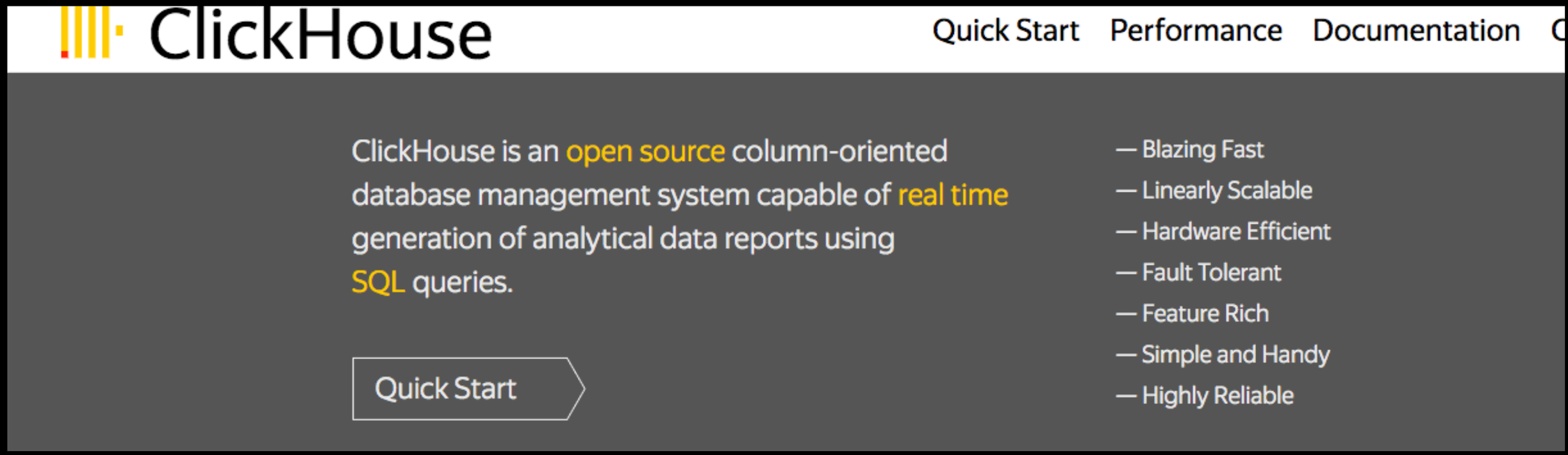
OR

compromises on what can be queried

ClickHouse Clicked

Yandex Open Sourced in 2016

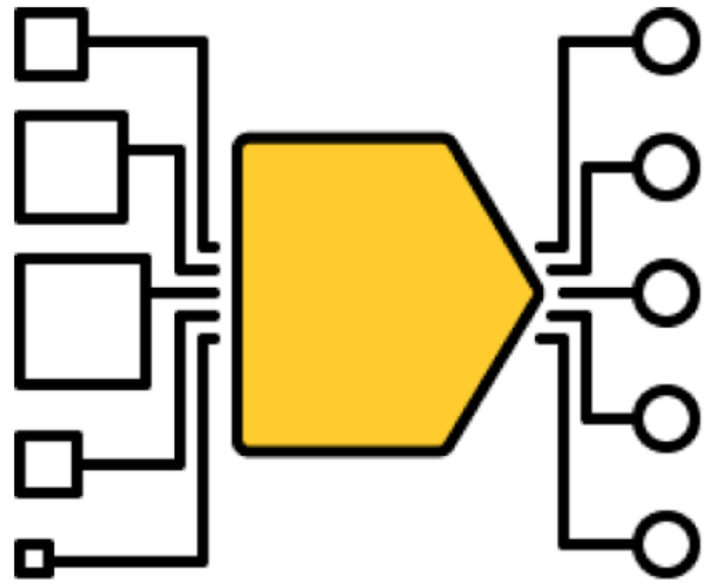
https://clickhouse.yandex/



CLOUDFLARE

Requirements

- Fast. Really fast
- Data processing in real time
- Capable of storing petabytes of data
- Fault-tolerance in terms of datacenters
- Flexible query language

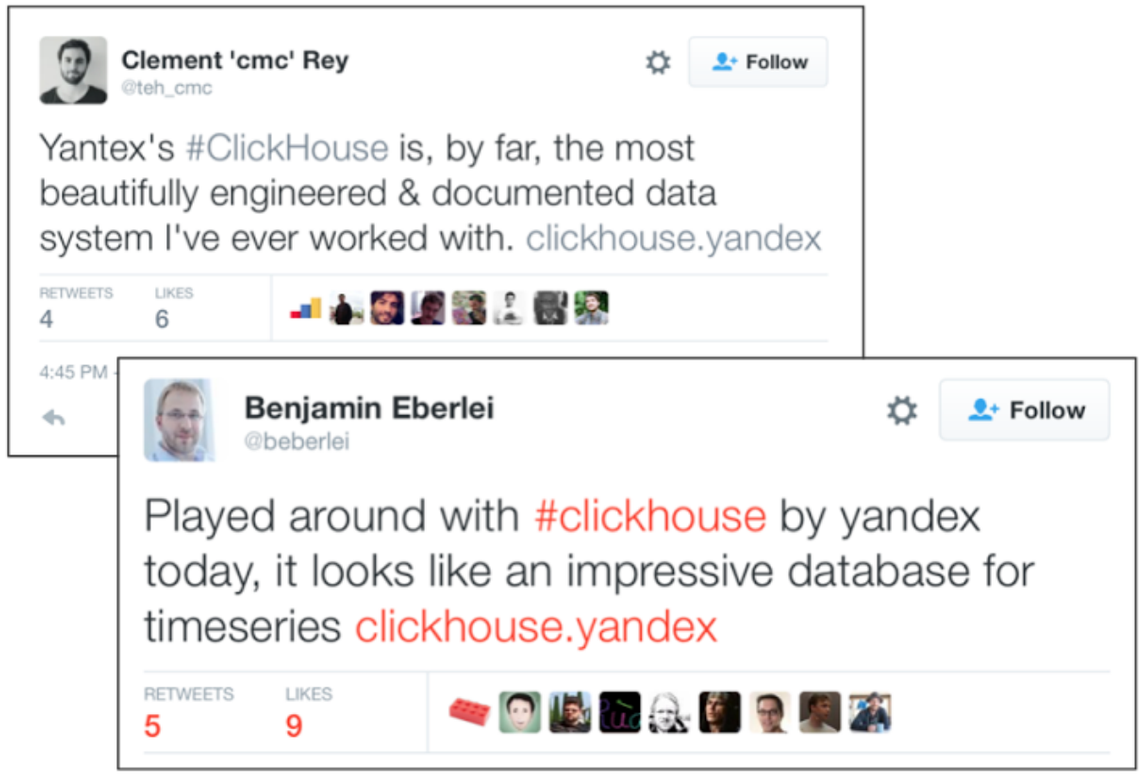


The main ideas behind ClickHouse

- SQL
- Linearly scalable
- Focused on fast query execution
- Realtime
- Column-oriented

ClickHouse today

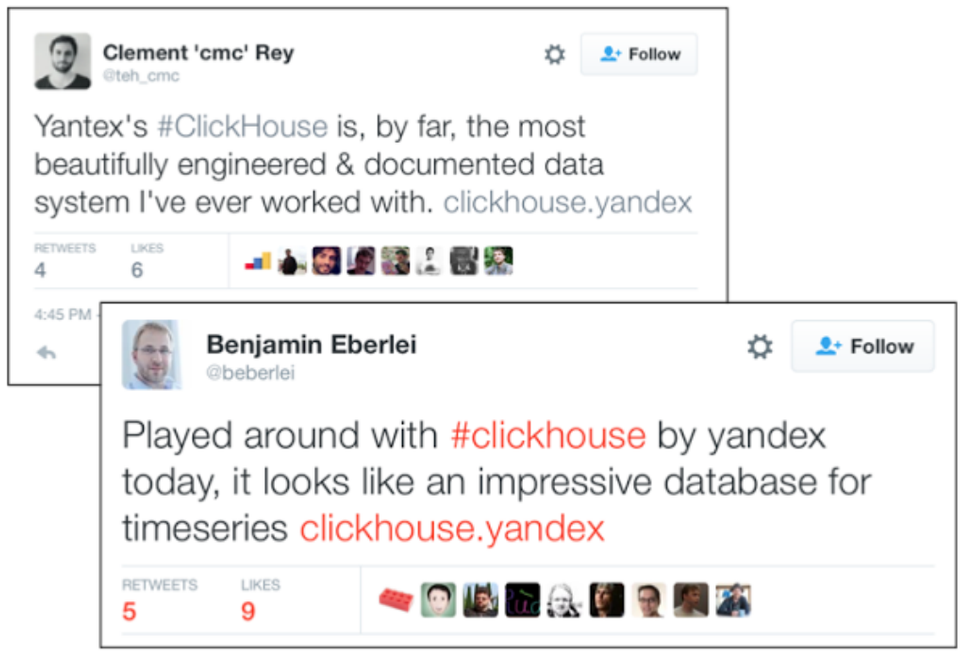
- Open-source, Apache 2.0
- 100+ companies outside Yandex
- Strong community
- Active development



X

Community

- 700+ people in Telegram chats, active every day
- 102 side projects on GitHub: drivers, clients, interfaces etc.
- Tabix: web interface over ClickHouse
- Integrations:
  Grafana
  Redash
  Apache Zeppelin
  Superset
  Power BI



X

Querying

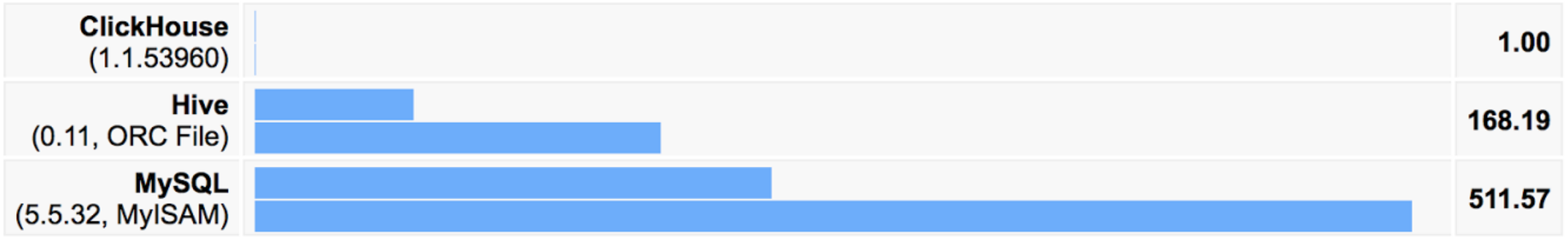
- SQL dialect + extensions
- Additional features: approximate functions, URI functions and more
- Arrays, nested data types
- Distributed out of the box
- Pluggable external key-value dictionaries

Performance

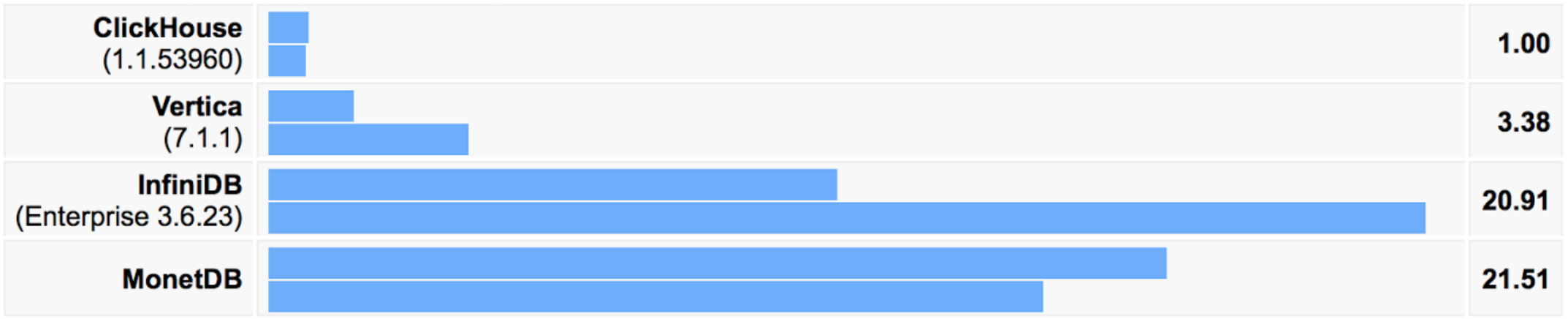
- Sub-second query latency
- >100x faster than Hadoop,
  >100x faster than typical DBMS
- Up to a few billion rows/second
  per single node
- Up to 2 terabytes per second
  on clustered setup of 400 nodes

X

Relative query processing time (lower is better):



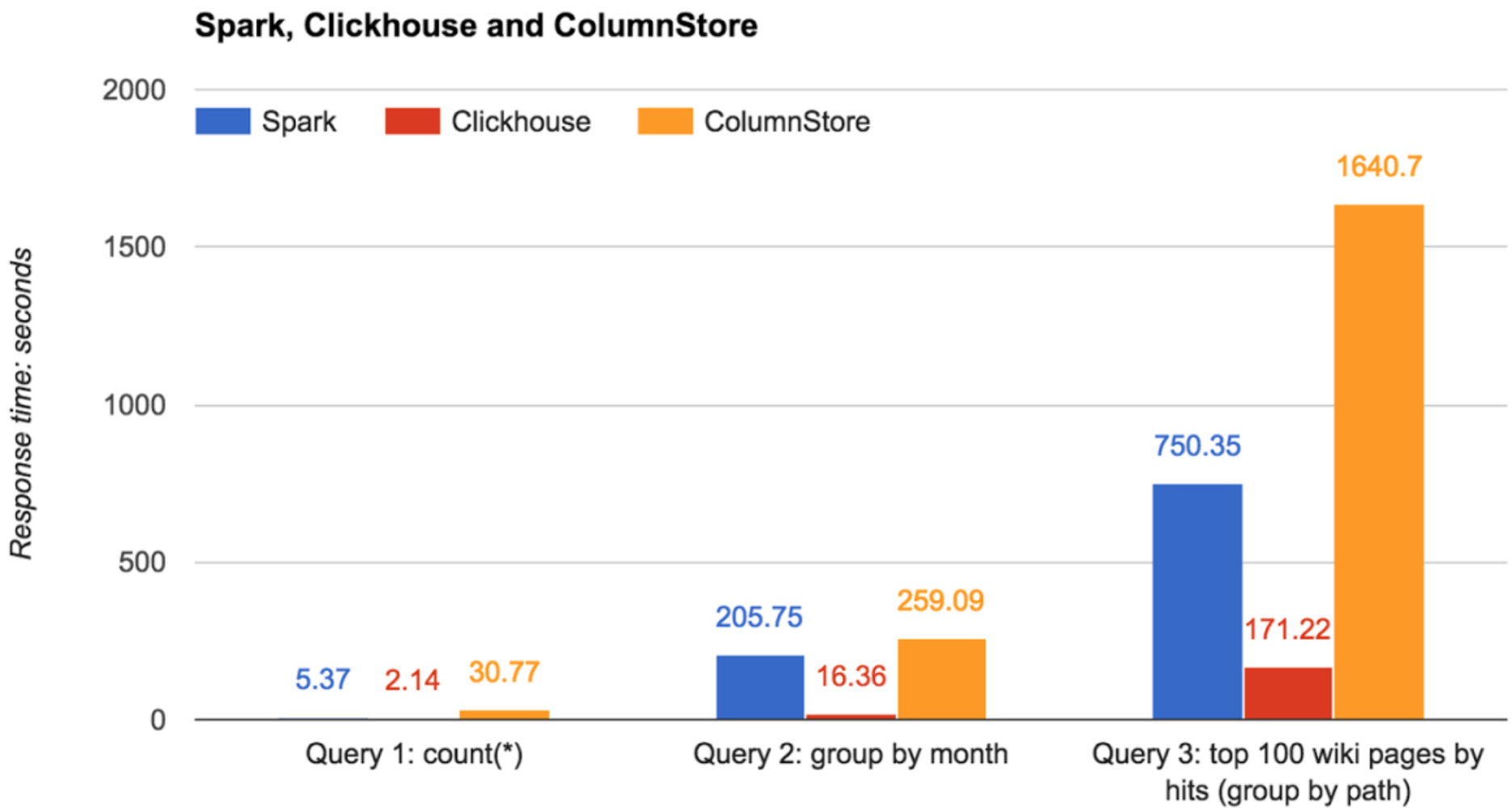
Relative query processing time (lower is better):



More info: https://clickhouse.yandex/benchmark.html

x

ClickHouse vs. Spark and MariaDB



http://bit.ly/2oINPsJ

X

Interfaces

- Console client
- HTTP
- JDBC Driver, ODBC Driver in beta
- Clients for:
  Python, PHP, Go, Node.js
  Perl, Ruby, R, C++
  .NET, Scala, Julia

What we log

What we log

One log entry for [query[+answer]]

What we log

One log entry for [query[+answer]]
Metal, Time, IP’s, Ports, Transport
Q-trinity, Header bits, EDNS

What we log

One log entry for [query[+answer]]
Metal, Time, IP's, Ports, Transport
Q-trinity, Header bits, EDNS
Response code, size, counts, cached y/n, signed,
<special> logic, Zone information,
Special sections: DNS Firewall, CnameFlatten, etc.

What we log

One log entry for [query[+answer]]
Metal, Time, IP's, Ports, Transport
Q-trinity, Header bits, EDNS
Response code, size, counts, cached y/n, signed,
<special> logic, Zone information,
Special sections: DNS Firewall, CnameFlatten, etc.
  Most log entries <150 bytes

What we log

One log entry for [query[+answer]]
Metal, Time, IP's, Ports, Transport
Q-trinity, Header bits, EDNS
Response code, size, counts, cached y/n, signed,
<special> logic, Zone information,
Special sections: DNS Firewall, CnameFlatten, etc.
  Most log entries <150 bytes
Why Log inside server?

What we log

One log entry for [query[+answer]]
Metal, Time, IP's, Ports, Transport
Q-trinity, Header bits, EDNS
Response code, size, counts, cached y/n, signed,
<special> logic, Zone information,
Special sections: DNS Firewall, CnameFlatten, etc.
  Most log entries <150 bytes
Why Log inside server?
    Strong binding between Q&A

What we log

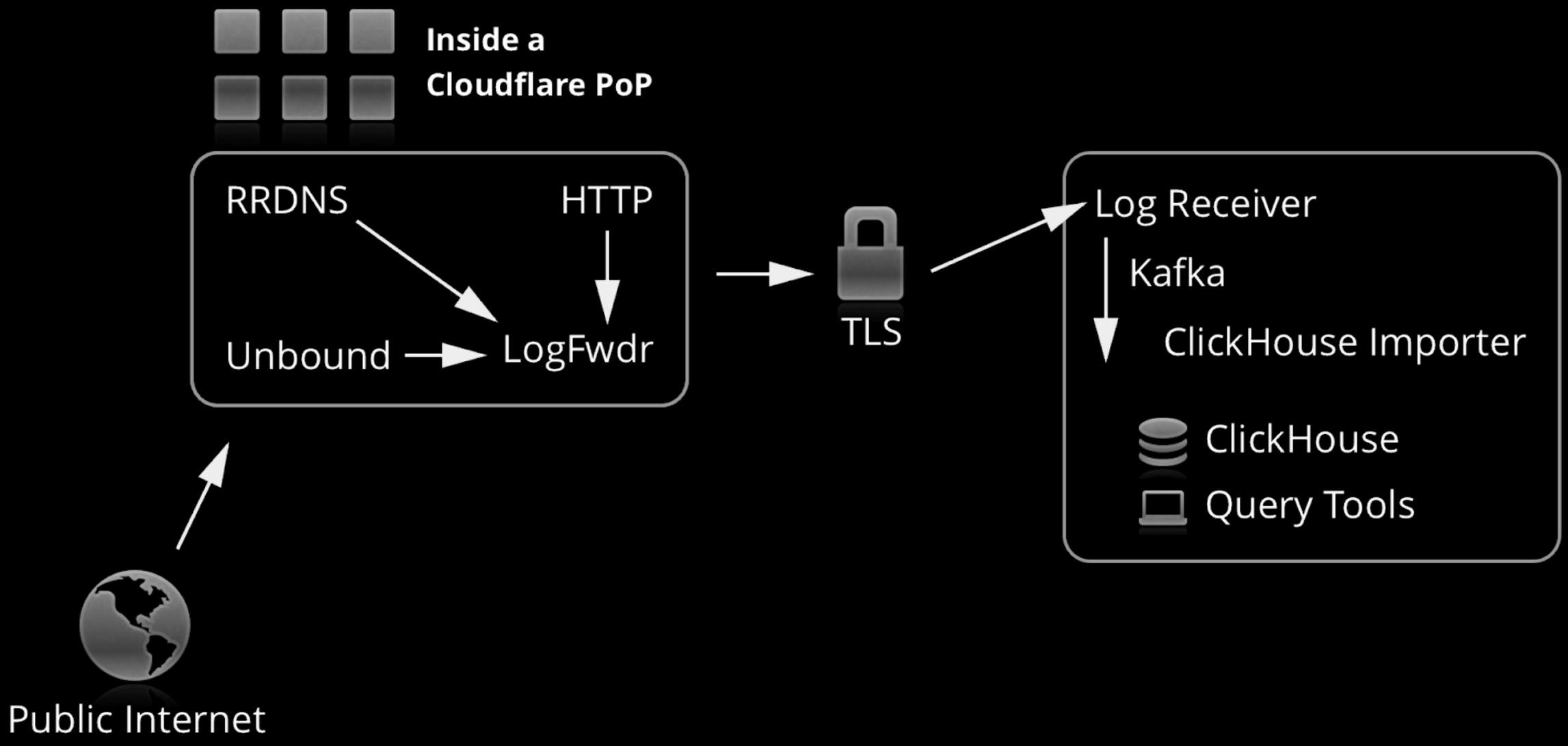
One log entry for [query[+answer]]
Metal, Time, IP's, Ports, Transport
Q-trinity, Header bits, EDNS
Response code, size, counts, cached y/n, signed,
<special> logic, Zone information,
Special sections: DNS Firewall, CnameFlatten, etc.
  Most log entries <150 bytes
Why Log inside server?
    Strong binding between Q&A
    Know which source answer is derived from

Collecting the data

Multiple DNS processes on each metal

One way to get logs: LogFwdr
One format: Cap'n'Proto
Processes write to Sockets
LogFwdr sends on;
- handles network issues

Simplified illustration



Mapping into ClickHouse

Mapping into ClickHouse

Read the documentation and experiment
Careful choices help performance

Mapping into ClickHouse

Read the documentation and experiment
Careful choices help performance

Mapping into ClickHouse

Read the documentation and experiment
Careful choices help performance

A small program reads from Kafka and exports to ClickHouse

Mapping into ClickHouse

Read the documentation and experiment
Careful choices help performance

A small program reads from Kafka and exports to ClickHouse

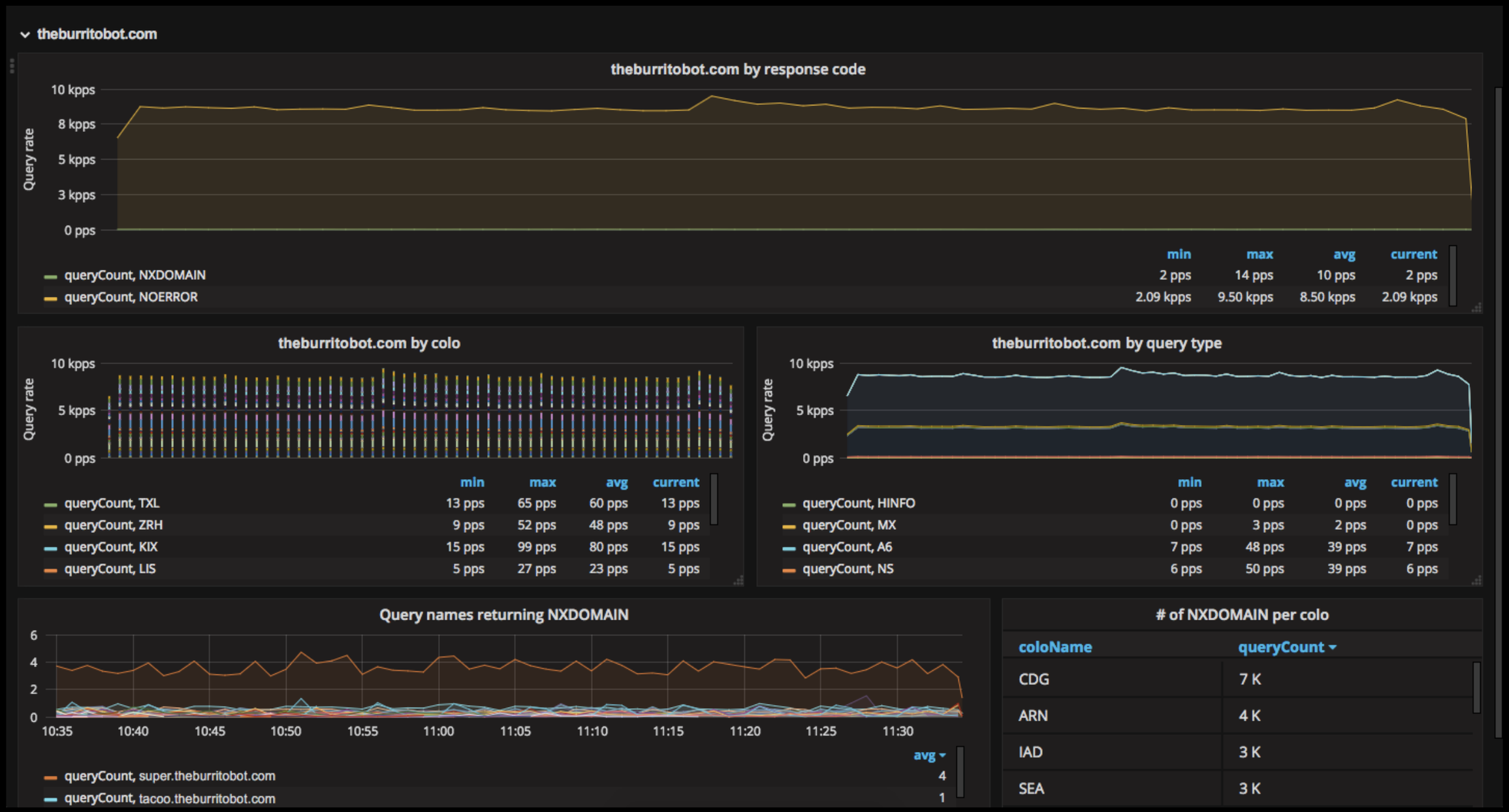
Mapping into ClickHouse

Read the documentation and experiment
Careful choices help performance

A small program reads from Kafka and exports to ClickHouse

Kafka enables multiple consumers of data

Grafana dashboard for customers with API



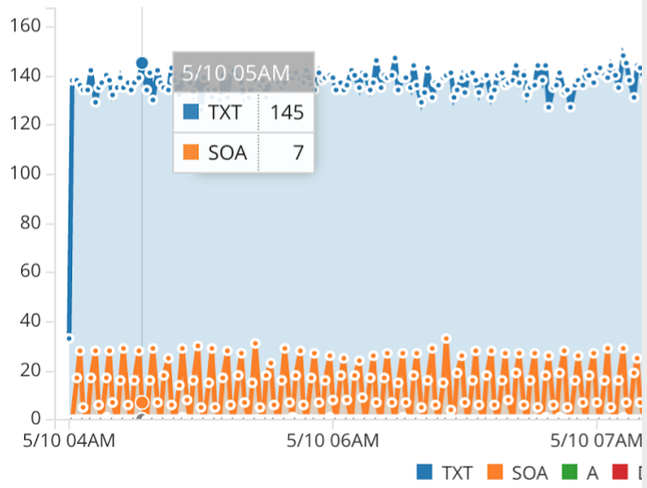
What can customers see in their dashboard

Queries, types, response codes, locations…

DNS Traffic

Queries by Response Code

Queries by Record Type

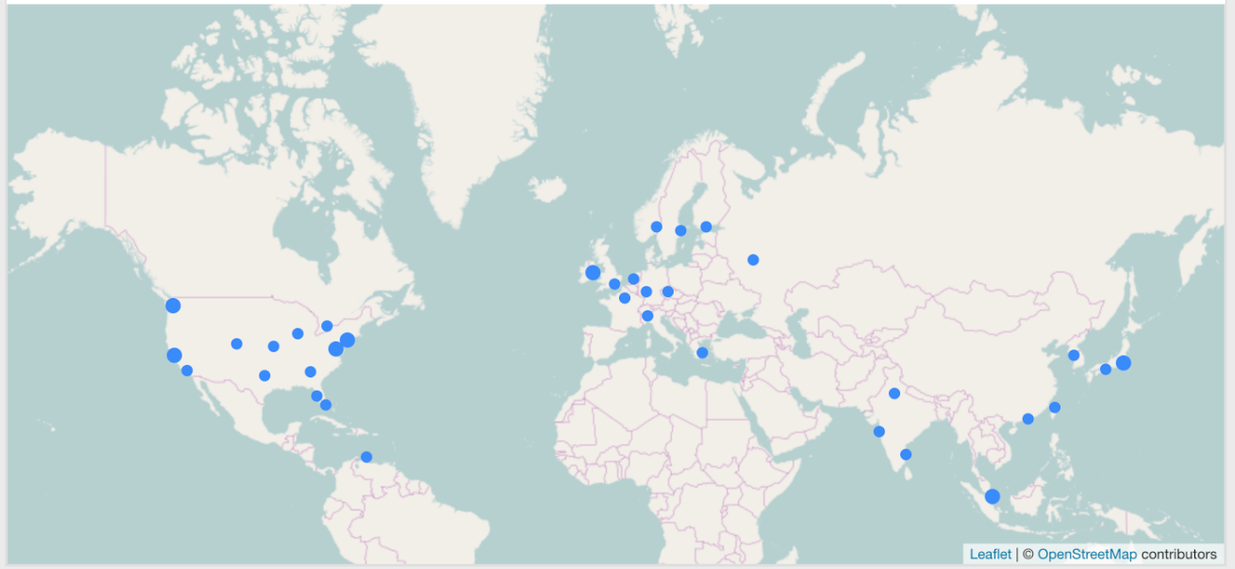


Geography

Last 24 hours

Web | DNS

DNS Traffic | NXDOMAIN



Top Data Centers for DNS Traffic

Last 24 hours

Top	City	Queries
1	Washington, United States (IAD)	18,942
2	San Jose, United States (SJC)	14,050
3	Newark, United States (EWR)	13,901
4	Seattle, United States (SEA)	13,599
5	Sydney, Australia (SYD)	13,248

Top Data Centers for NOERROR

Last 24 hours

Top	City	Queries
1	Washington, United States (IAD)	18,942
2	San Jose, United States (SJC)	14,050
3	Newark, United States (EWR)	13,901
4	Seattle, United States (SEA)	13,591
5	Sydney, Australia (SYD)	13,248

Top Data Centers for NXDOMAIN

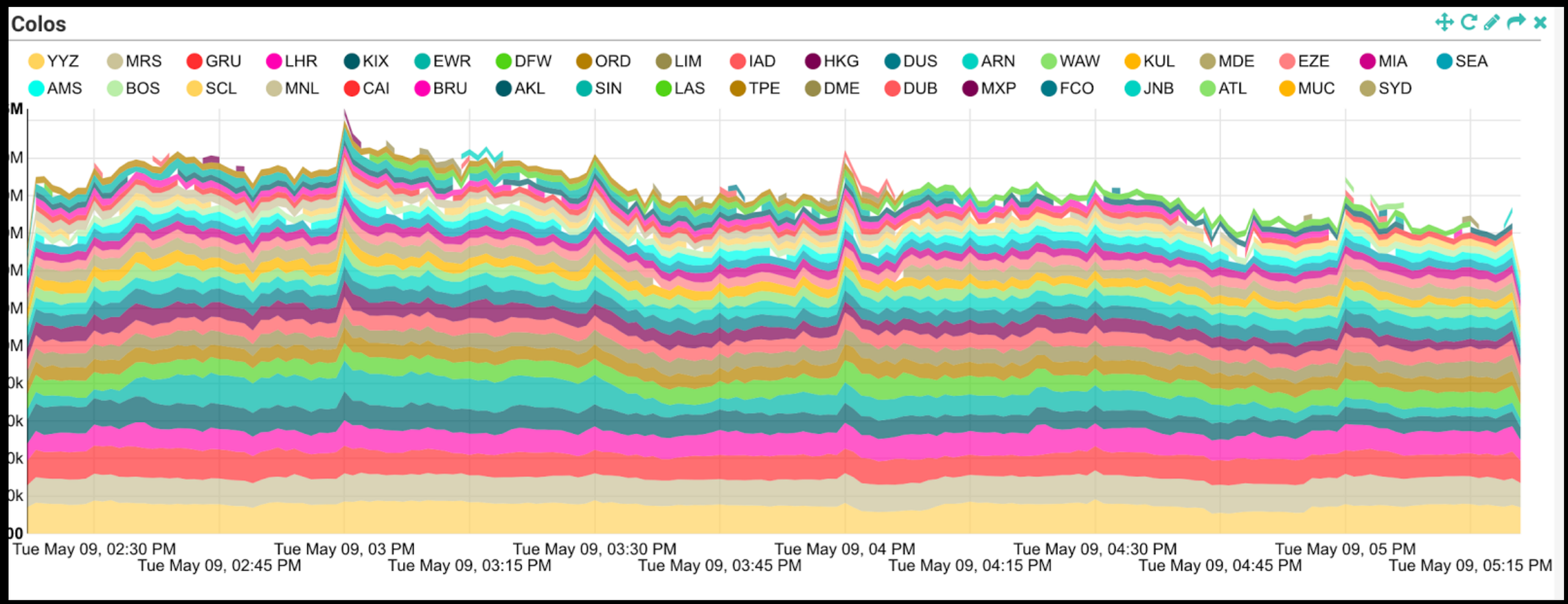
Last 24 hours

Top	City	Queries
1	Seattle, United States (SEA)	8
2	Dallas-Fort Worth, United States (DFW)	1

Help ▸

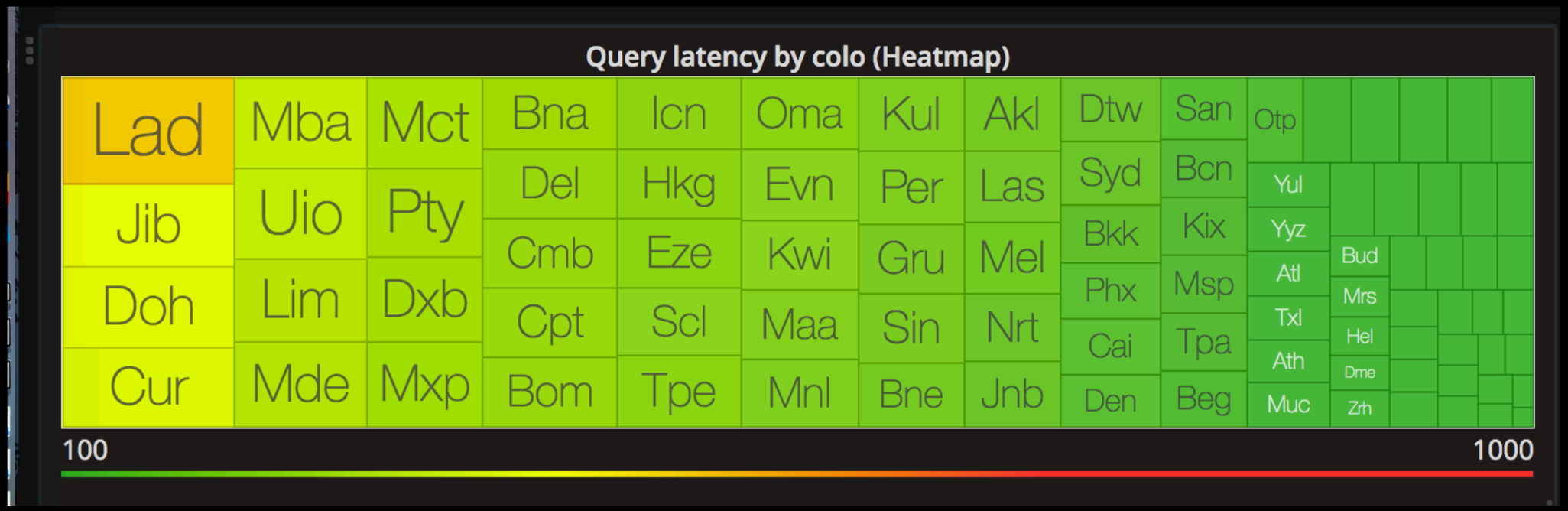
CLOUDFLARE

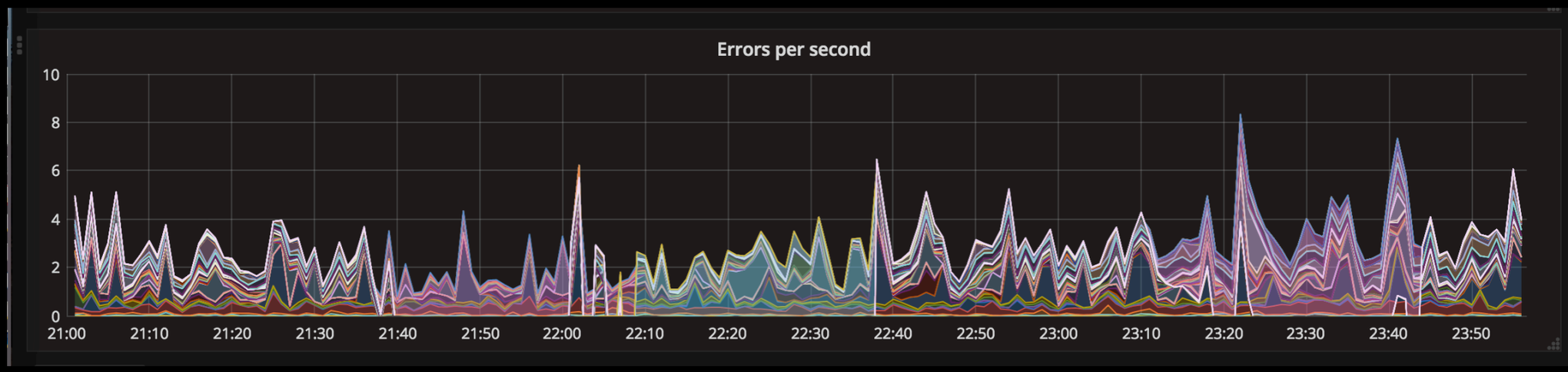
How do sites stack up
Internal view (superset)



CLOUDFLARE

Internal views: Grafana





CLOUDFLARE

What has ClickHouse enabled

- Lots of new dashboards
- People from all departments dive into data to help the with “today’s issue”
- Problems/questions answered

CLOUDFLARE

Example: How is NS traffic distributed?

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world
In April 2016

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world
In April 2016

- Added second IPv4 addresses to NS6 and NS7

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world
In April 2016

- Added second IPv4 addresses to NS6 and NS7
- Later added glue for NS6

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world
In April 2016

- Added second IPv4 addresses to NS6 and NS7
- Later added glue for NS6

Monitored the deployment:

Example: How is NS traffic distributed?

Cloudflare.com has 5 NS all equal around the world
In April 2016

- Added second IPv4 addresses to NS6 and NS7
- Later added glue for NS6

Monitored the deployment:

With pcap's from the sflow samples sent from edges to core DDoS defense system.

Distribution of queries: April/May 2016

Distribution of queries: April/May 2016

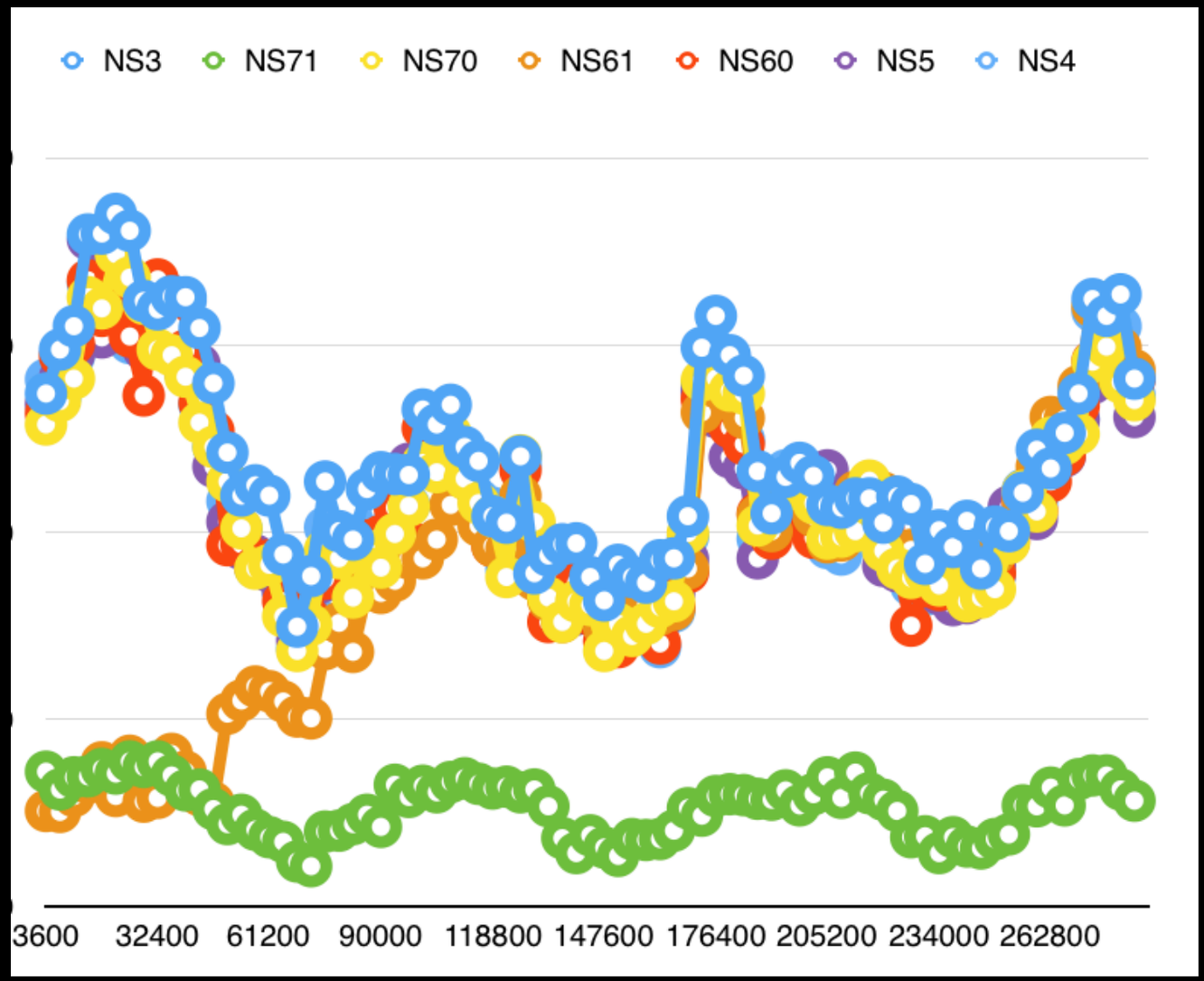
Graph shows effects

- Glue less address
- GLUE record added

Distribution of queries: April/May 2016

Graph shows effects

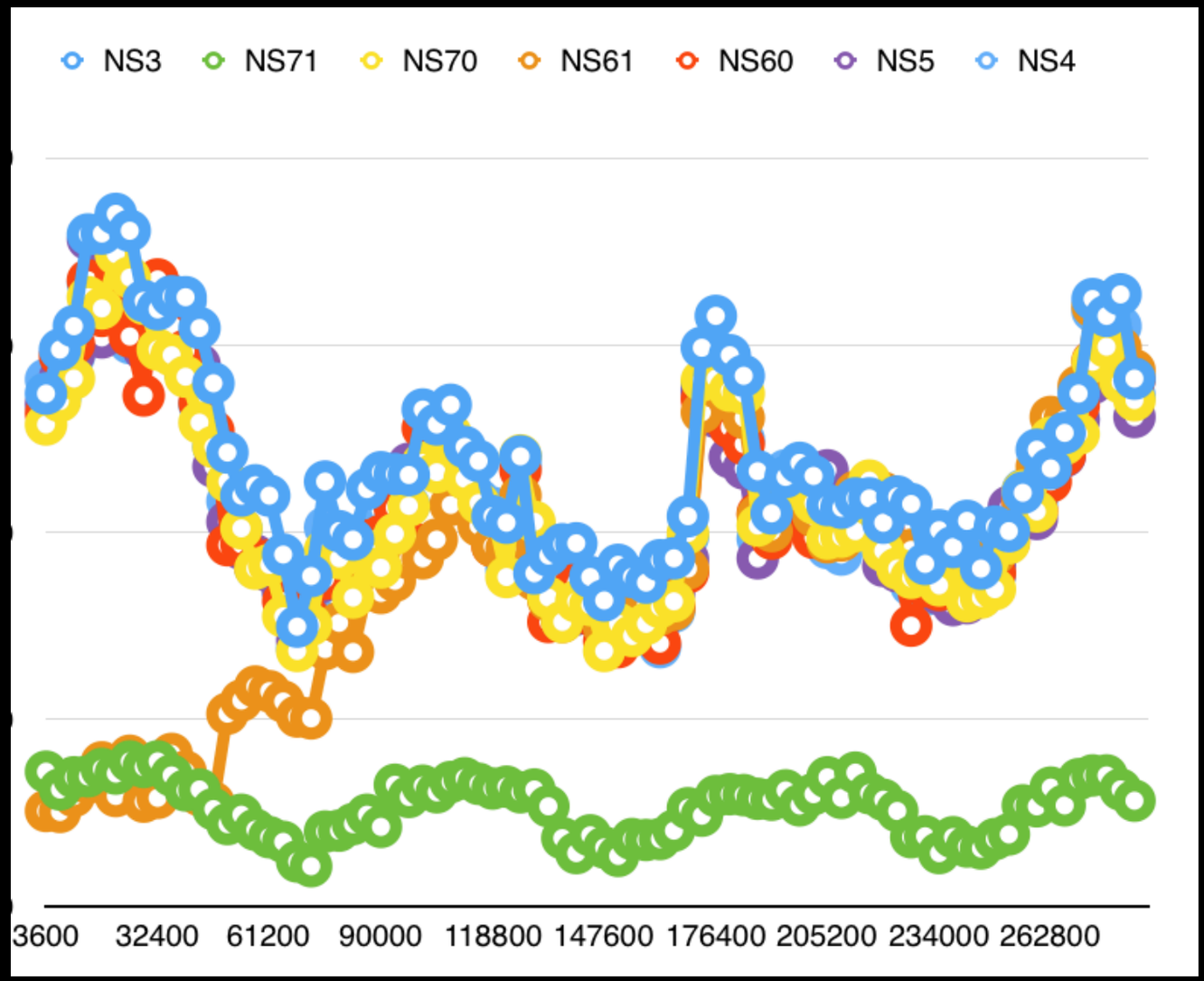
- Glue less address
- GLUE record added



Distribution of queries: April/May 2016

Graph shows effects

- Glue less address
- GLUE record added



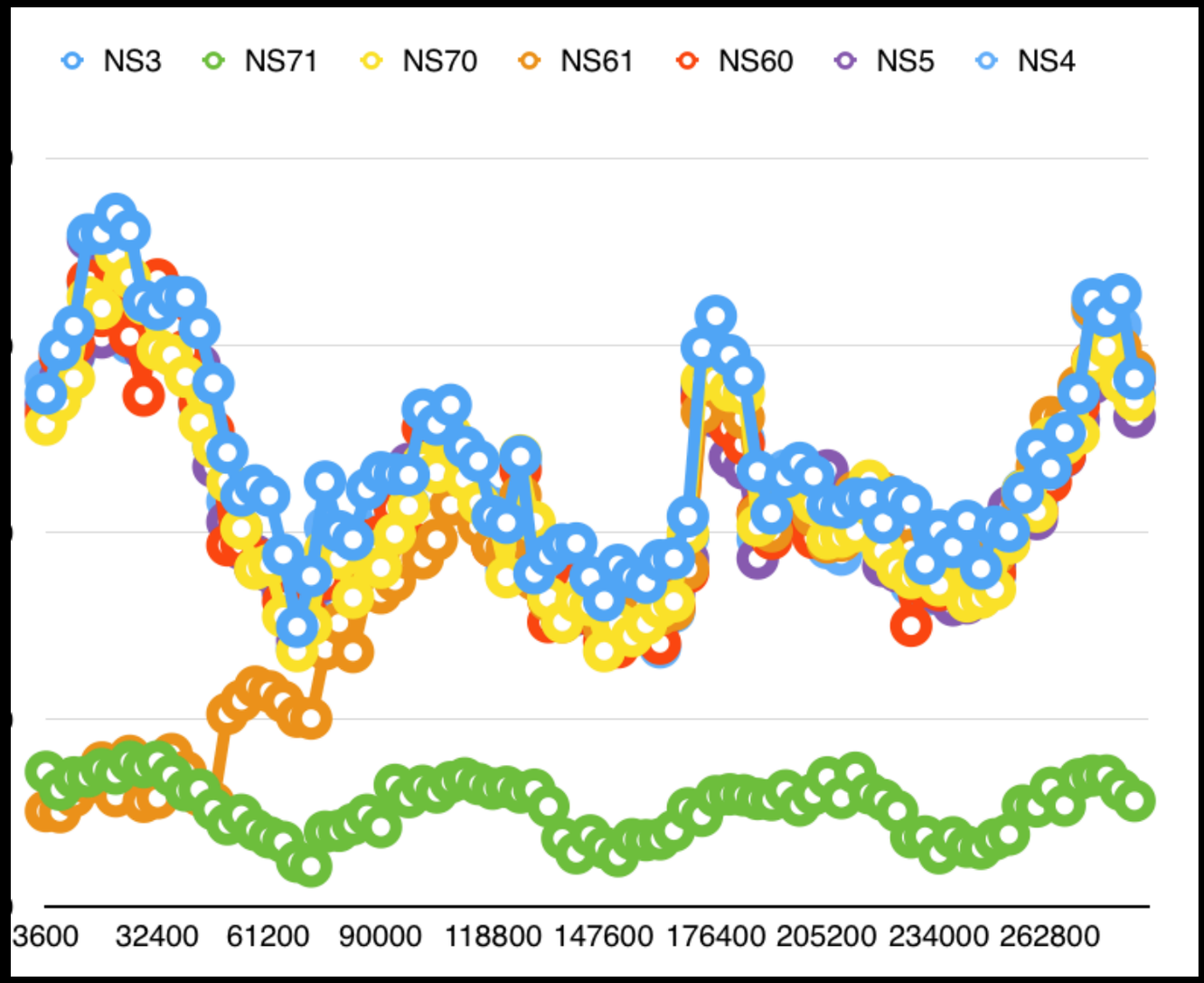
Monitoring not scaleable
Samples are “skewed”

Distribution of queries: April/May 2016

Graph shows effects

- Glue less address
- GLUE record added

Monitoring not scaleable
Samples are “skewed”



Added second v4 and v6 address to those NS’s

CLOUDFLARE

ClickHouse query

ClickHouse query

```
SELECT IPv4NumToString(dstIPv4) AS  DstAddr, count() AS cc
```

ClickHouse query

```
SELECT IPv4NumToString(dstIPv4) AS  DstAddr, count() AS cc
From dnslogs
```

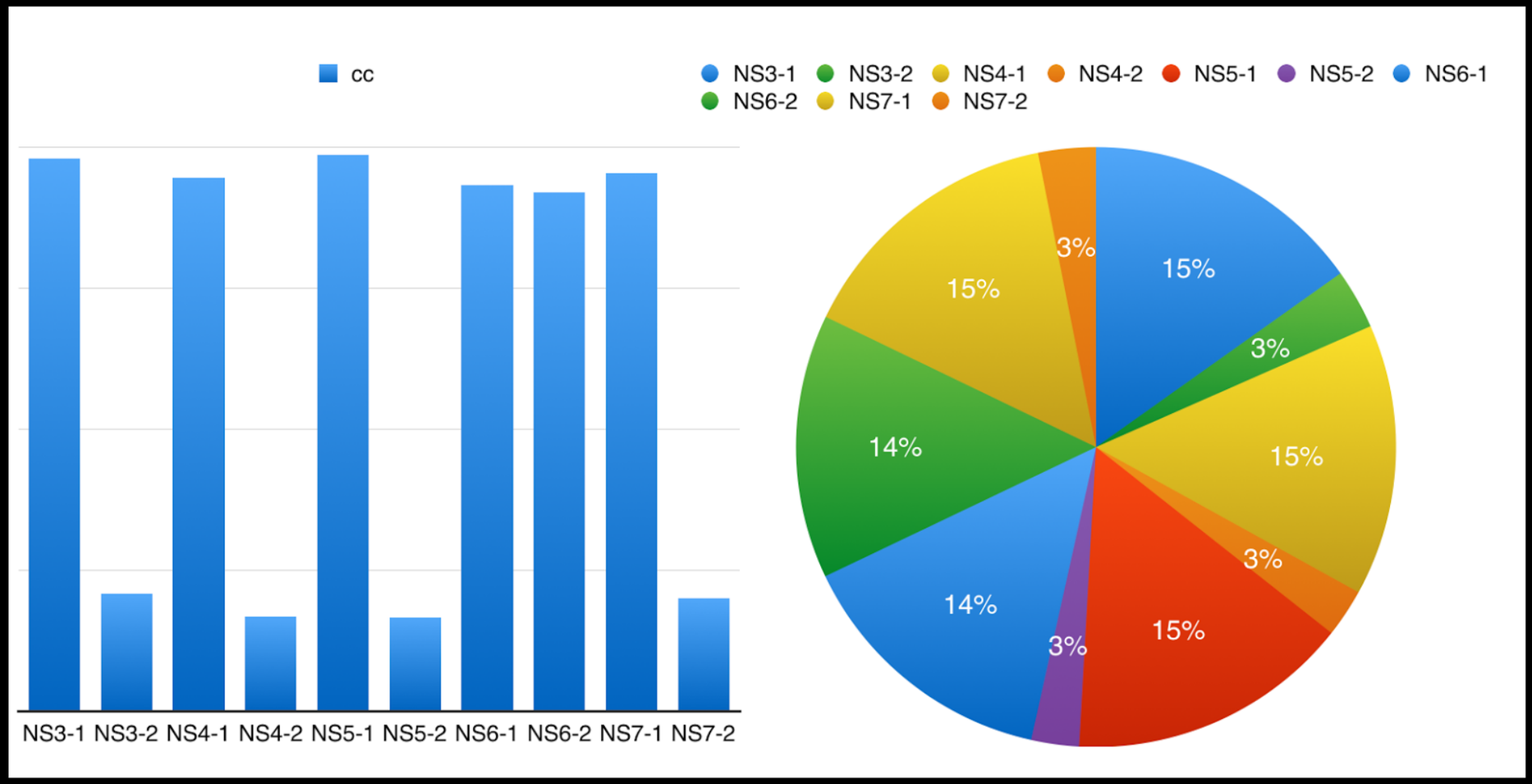
ClickHouse query

```
SELECT IPv4NumToString(dstIPv4) AS  DstAddr, count() AS cc
From dnslogs
WHERE date = yesterday() AND zoneId = 42
```

ClickHouse query

```
SELECT IPv4NumToString(dstIPv4) AS  DstAddr, count() AS cc
From dnslogs
WHERE date = yesterday() AND zoneId = 42
group by dstIPv4 ORDER by count() DESC LIMIT 15
```

ClickHouse result for IPv4



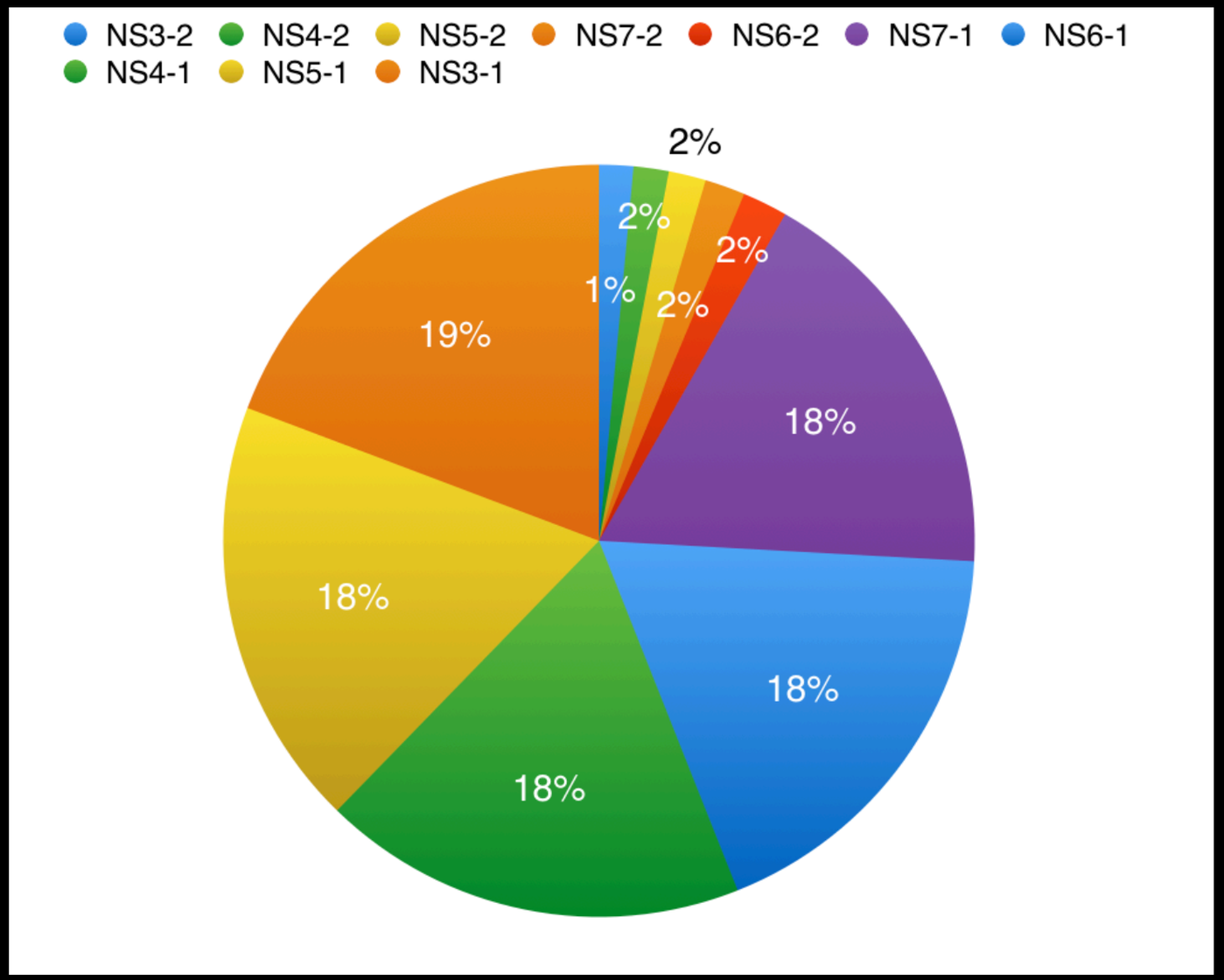
Glue addresses get similar volume traffic
Glue less addresses get 1/4 of traffic

CLOUDFLARE

ClickHouse results: IPv6

- Traffic is 17% of v4
- Much worse glue ratio:
  9-13x not 4x

- Why???



Glue summary

Glue summary

- Glue matters as most resolvers are:
  parent glue centric

Glue summary

- Glue matters as most resolvers are:
  parent glue centric
- Resolvers select name servers to query based on:
  Addresses not Names

Glue summary

- Glue matters as most resolvers are:
  parent glue centric
- Resolvers select name servers to query based on:
  Addresses not Names
- IPv6 traffic is dominated by resolvers

Glue summary

- Glue matters as most resolvers are:
  parent glue centric
- Resolvers select name servers to query based on:
  Addresses not Names
- IPv6 traffic is dominated by resolvers
- V4 has more “diversity” in what is querying

Whats next

- Cloudflare is adopting ClickHouse as the default data analytics
- Improve service
- Make customers happier
- More tooling
- More interesting research projects

Q/A

Before asking a question:
guess the size of our ClickHouse cluster

blog.cloudflare.com
clickhouse.yandex
Grafana
Superset