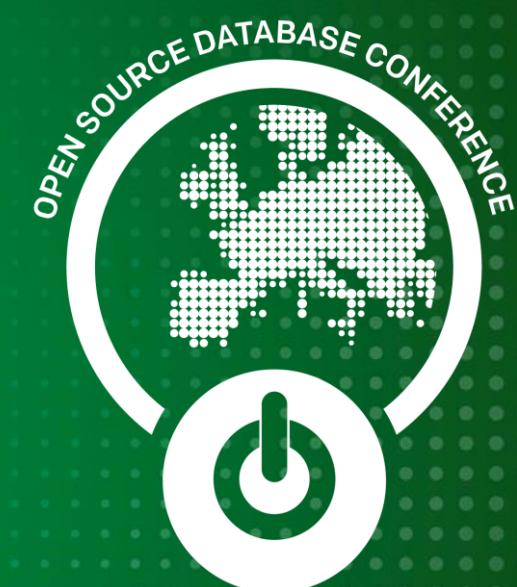


ClickHouse 2018

**How to stop waiting for your queries
to complete and start having fun**

Alexander Zaitsev

Altinity



**PERCONA
LIVE EUROPE
FRANKFURT**

Who am I

M.Sc. In mathematics from Moscow State University

Software engineer since 1997

Developed distributed systems since 2002

Focused on high performance analytics since 2007

Director of Engineering in LifeStreet

Co-founder of Altinity – ClickHouse Service Provider

.. and I am not Peter's brother :)

What Is ClickHouse?

NoSQL Databases



NewSQL Databases



Graph Databases



MPP Databases



Cloud EDW



Data Transformation



Data Integration



PERCONA
LIVE

ClickHouse DBMS is

Column Store

MPP

Realtime

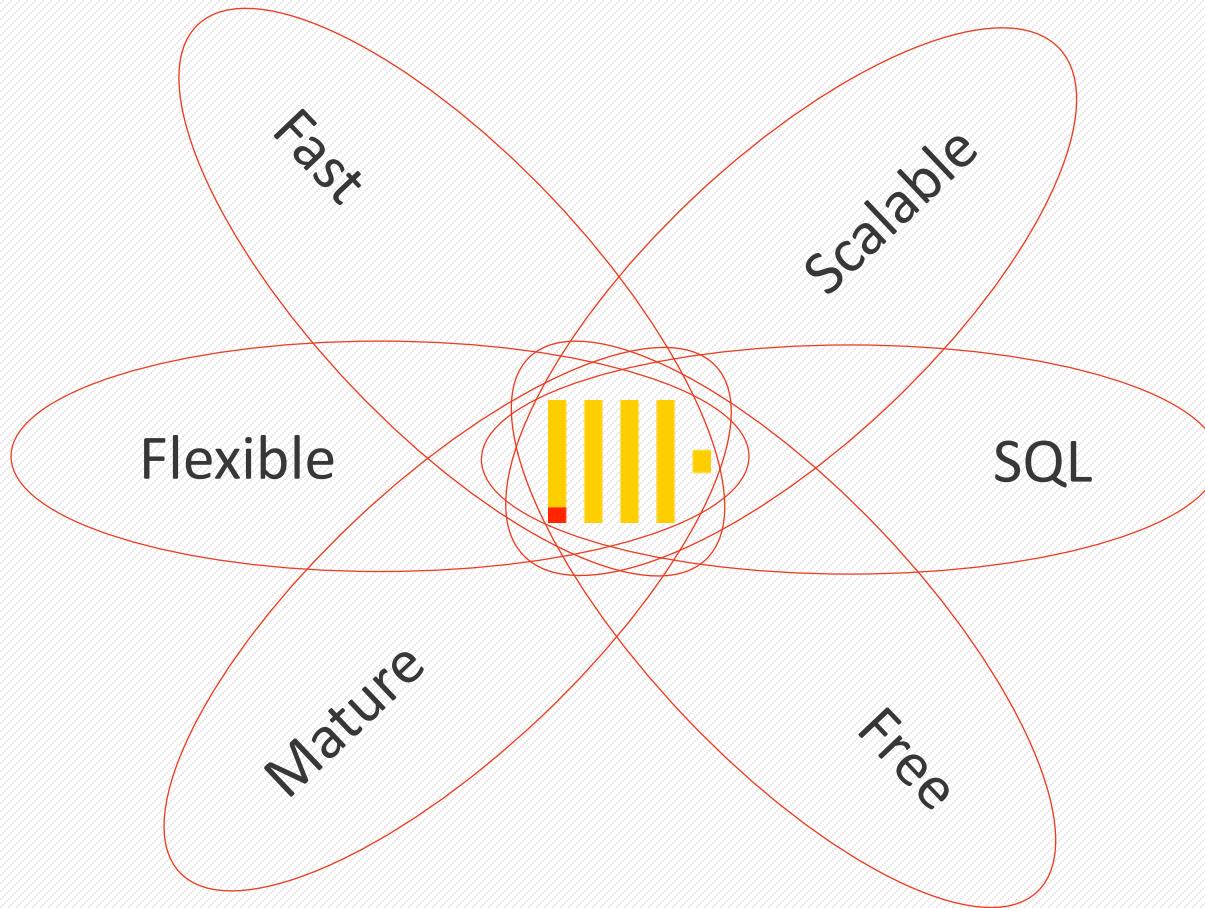
SQL

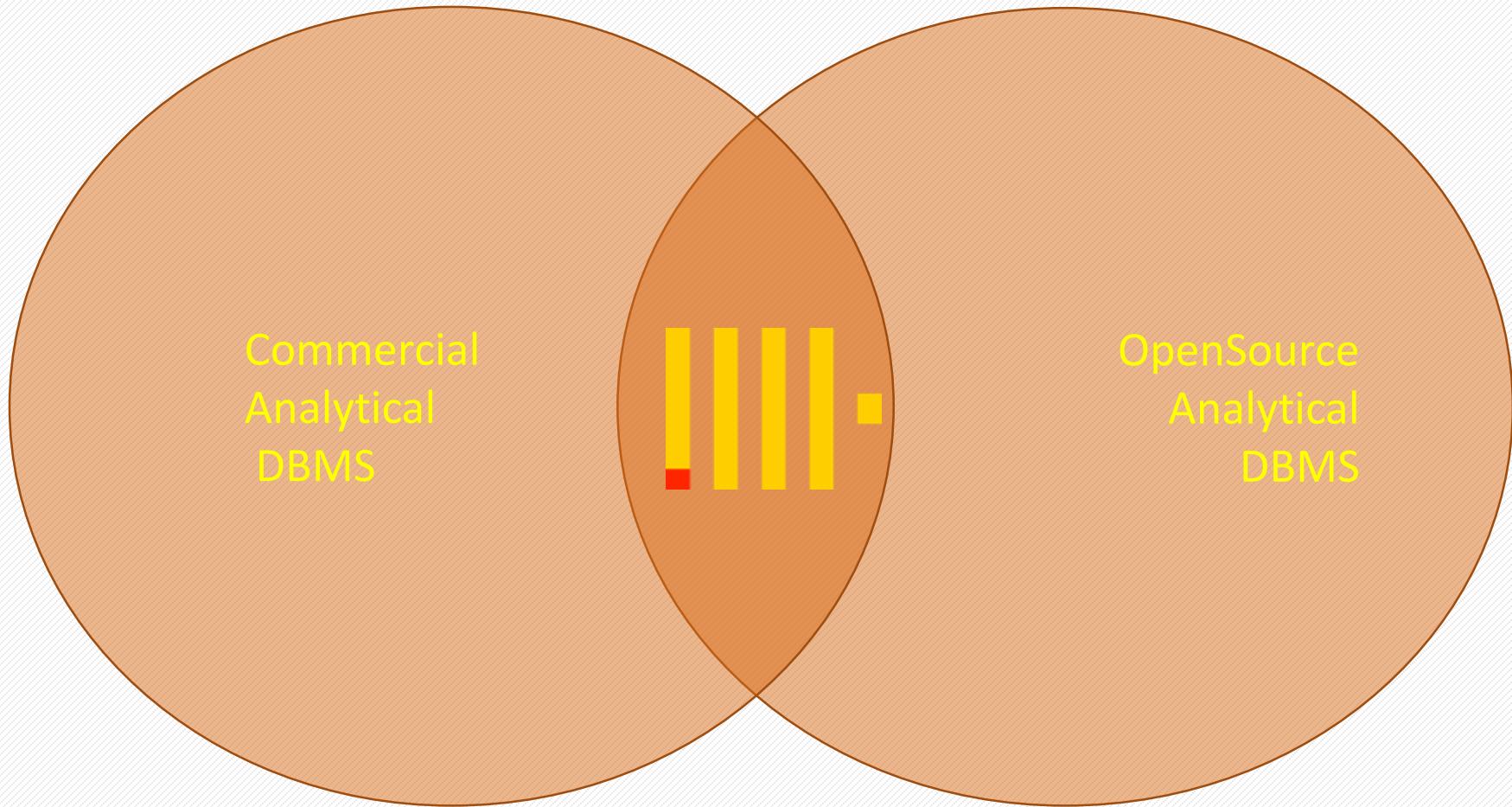
Open Source

<http://clickhouse.yandex>

- Developed by Yandex for Yandex.Metrica
 - *Yandex (NASDAQ: YNDX) – “Russian Google” (50% market share in search, 50+ b2b and b2c products)*
 - *Yandex.Metrica – world 2nd largest web analytics platform*
- Open Source since June 2016 (Apache 2.0 license)
- 200+ companies using in production today
- Several hundred experimenting, doing POC etc.
- Dozens of contributors to the source code

Why Yet Another DBMS?





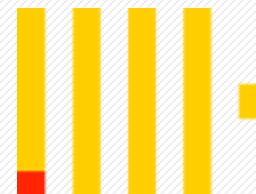
ClickHouse

Fast!

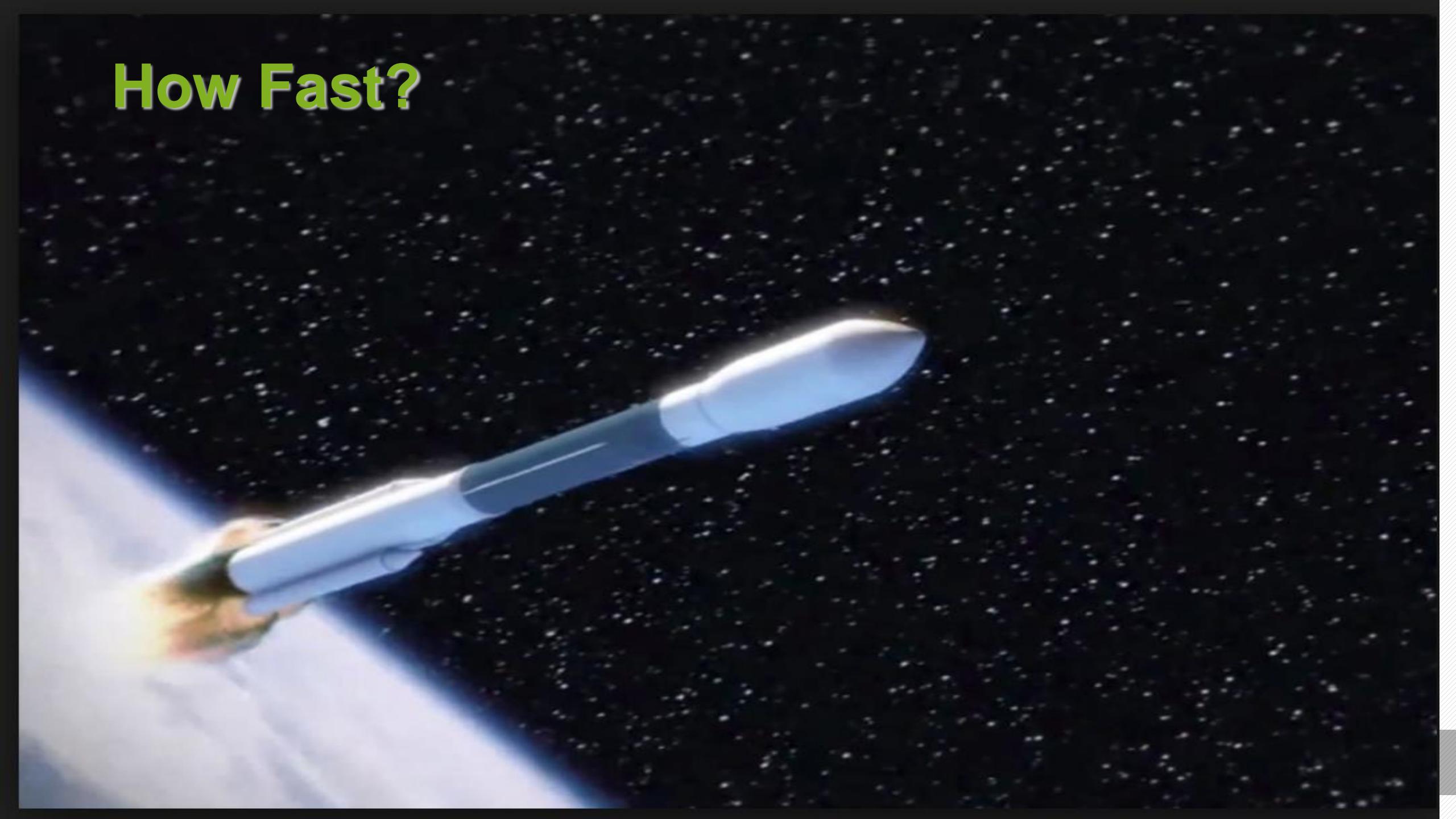
Flexible!

Free!

Fun!



How Fast?



1+ trillion rows table

```
:) select count(*) from dw.ad8_fact_event;
```

```
SELECT count(*)
FROM dw.ad8_fact_event
```

```
count()
1261705085657
```

```
1 rows in set. Elapsed: 3.552 sec. Processed 1.26 trillion rows, 1.26 TB (355.22 billion
rows/s., 355.22 GB/s.)
```

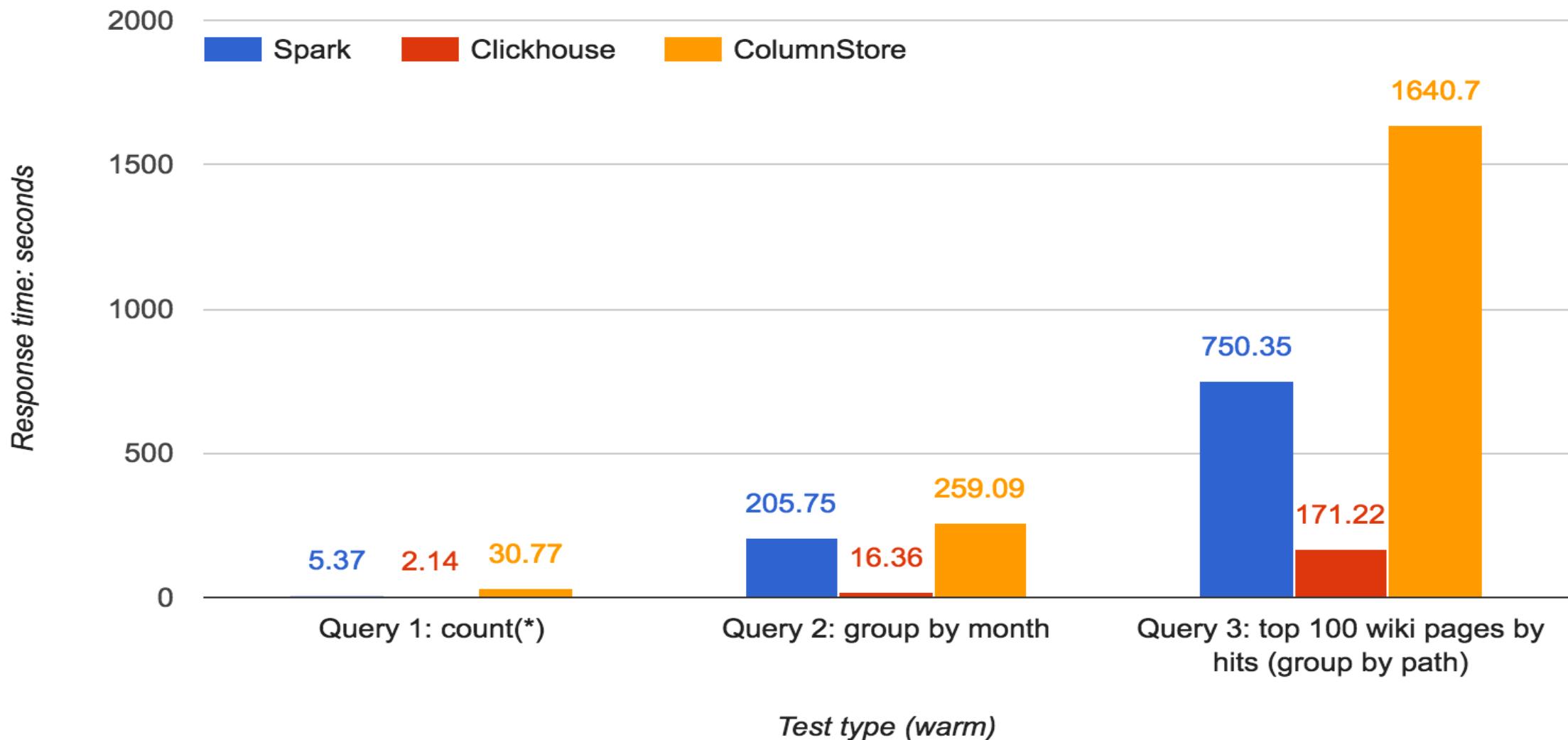
1+ trillion rows table

```
:) select sum(price_cpm) from dw.ad8_fact_event where access_day=today()-1 and event_key=-2;  
  
SELECT sum(price_cpm)  
FROM dw.ad8_fact_event  
WHERE (access_day = (today() - 1)) AND (event_key = -2)
```

```
sum(price_cpm)  
87579.09035192338
```

```
1 rows in set. Elapsed: 0.168 sec. Processed 161.89 million rows, 2.91 GB (961.83 million  
rows/s., 17.31 GB/s.)
```

Spark, Clickhouse and ColumnStore



WikiStat data, 28B rows.

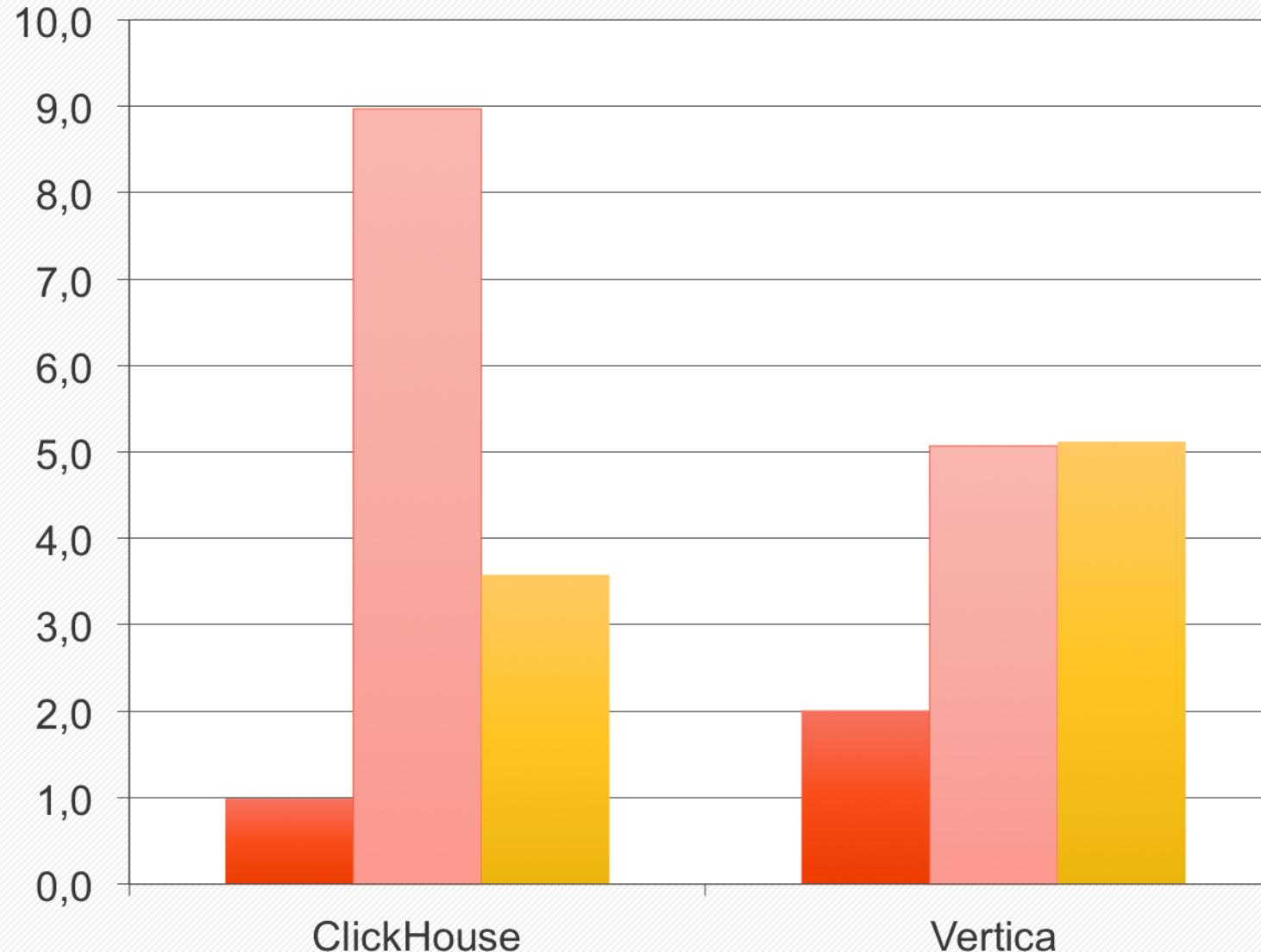
<https://www.percona.com/blog/2017/03/17/column-store-database-benchmarks-mariadb-columnstore-vs-clickhouse-vs-apache-spark/>

“1.1 Billion Taxi Rides Benchmarks”

<http://tech.marksblogg.com/benchmarks.html>

Query 1	Query 2	Query 3	Query 4	Setup
0.034	0.061	0.178	0.498	MapD & 2-node p2.8xlarge cluster
0.051	0.146	0.047	0.794	kdb+/q & 4 Intel Xeon Phi 7210 CPUs
-	2.415	3.599	4.962	ClickHouse at Altinity demo server
0.762	2.472	4.131	6.041	BrytlytDB 1.0 & 2-node p2.16xlarge cluster
1.034	3.058	5.354	12.748	ClickHouse, Intel Core i5 4670K
1.56	1.25	2.25	2.97	Redshift, 6-node ds2.8xlarge cluster
2	2	1	3	BigQuery
6.41	6.19	6.09	6.63	Amazon Athena
8.1	18.18	n/a	n/a	Elasticsearch (heavily tuned)
14.389	32.148	33.448	67.312	Vertica, Intel Core i5 4670K
22	25	27	65	Spark 2.3.0 & single i3.8xlarge w/ HDFS
35	39	64	81	Presto, 5-node m3.xlarge cluster w/ HDFS
152	175	235	368	PostgreSQL 9.5 & cstore_fdw

2016 LifeStreet benchmark
(unpublished)



- 19 queries, 1200M rows table, 3-node clusters

■ Single Table SELECT GROUP BY
■ JOIN t USING t_key WHERE <t.smth>?
■ JOIN (SELECT * FROM t WHERE <smth>) USING t_key

Time Series benchmarks (first time today!)

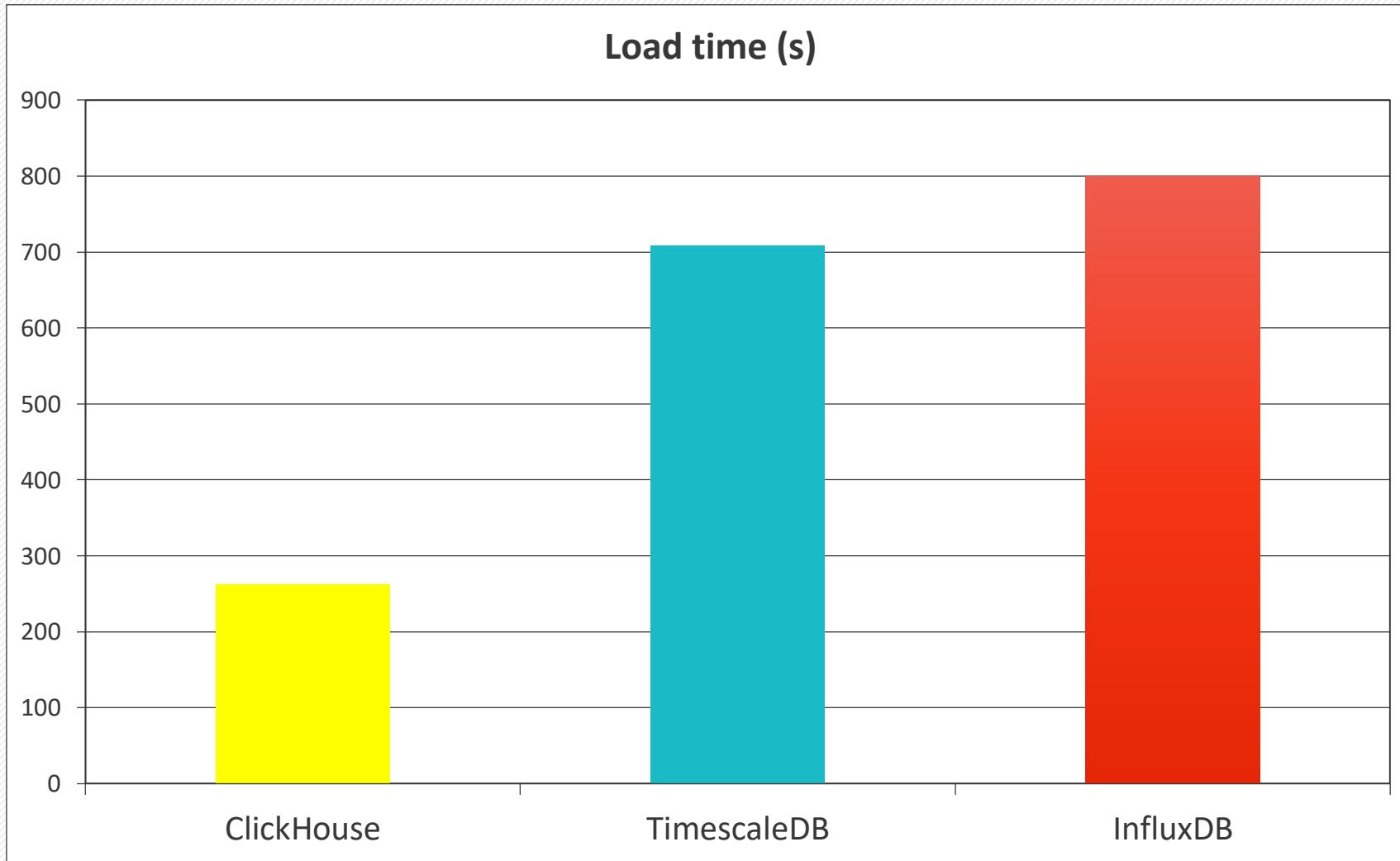
<https://github.com/timescale/tsbs>

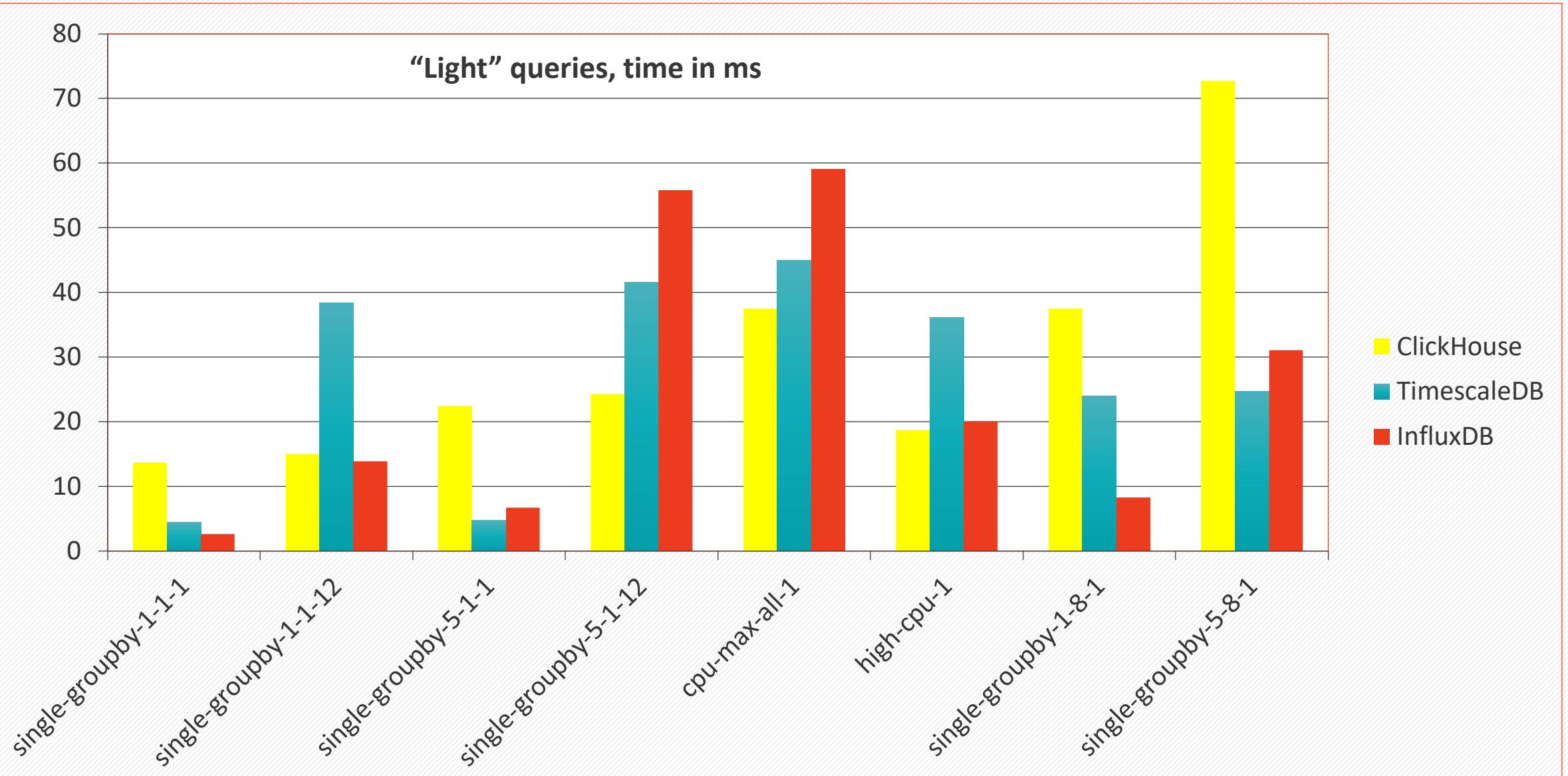
Benchmark suite to automate testing

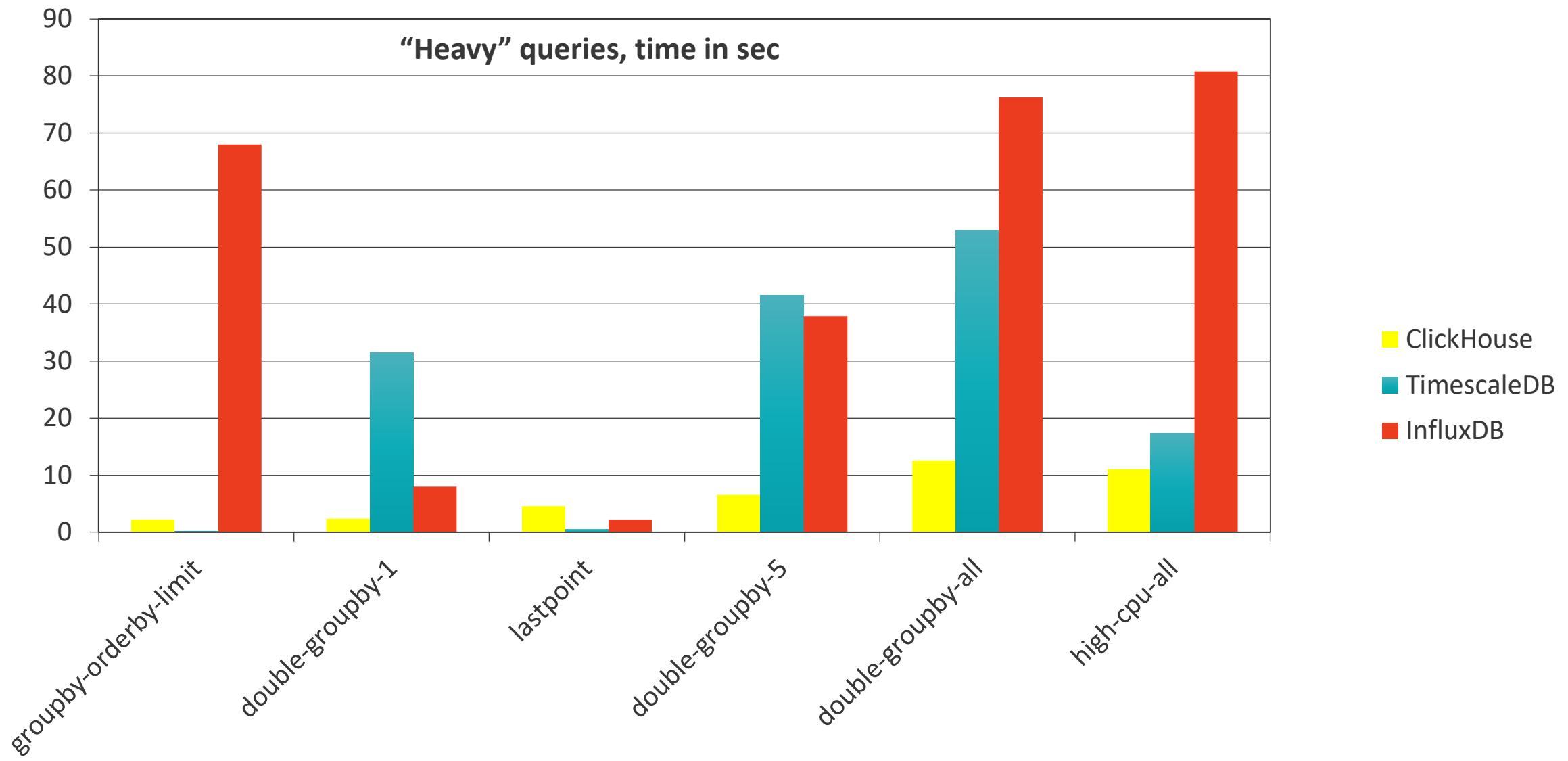
Loads 103M rows, 10 metrics per row

Runs 15 queries, 1000 runs each in 8 parallel threads

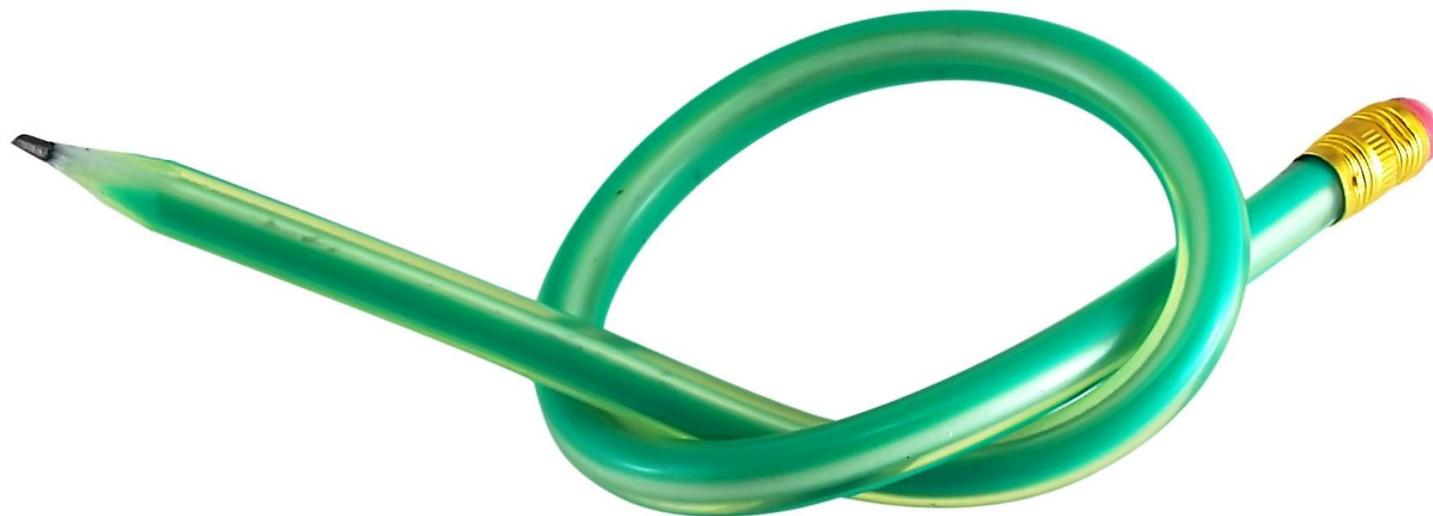
Supports TimescaleDB, InfluxDB, Cassandra, MongoDB and ClickHouse
(Altinity PR is submitted)







How flexible?



ClickHouse runs at

Bare metal (any Linux)

Amazon

Azure

VMware, VirtualBox

Docker, K8s

ClickHouse solves business problems at:

Mobile App and Web analytics

AdTech bidding analytics

Operational Logs analytics

DNS queries analysis

Stock correlation analytics

Telecom

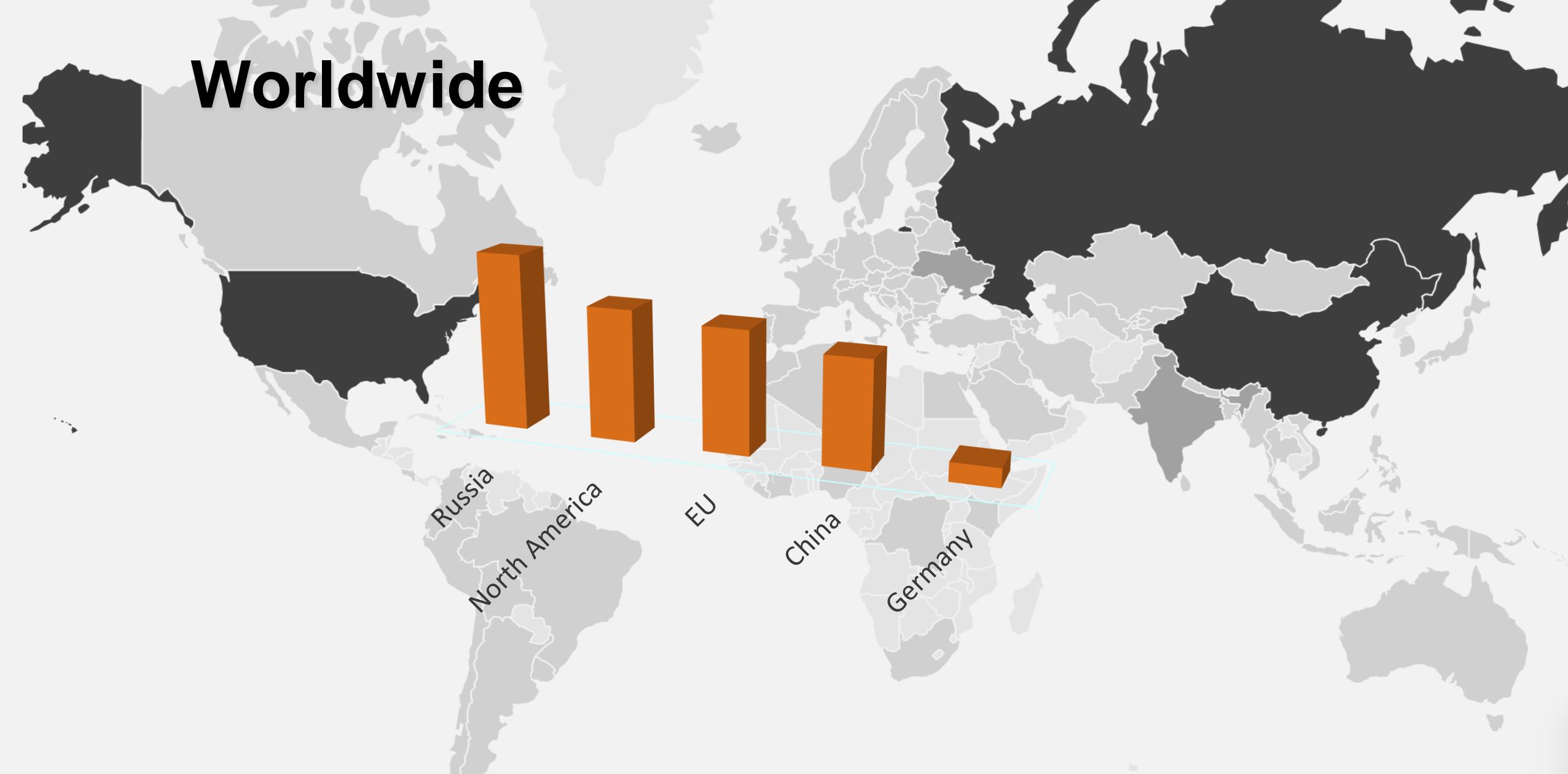
Security audit

Fintech SaaS

Manufacturing process control

BlockChain transactions analysis

Worldwide



www.altinity.com visits in 2013

Size does not matter



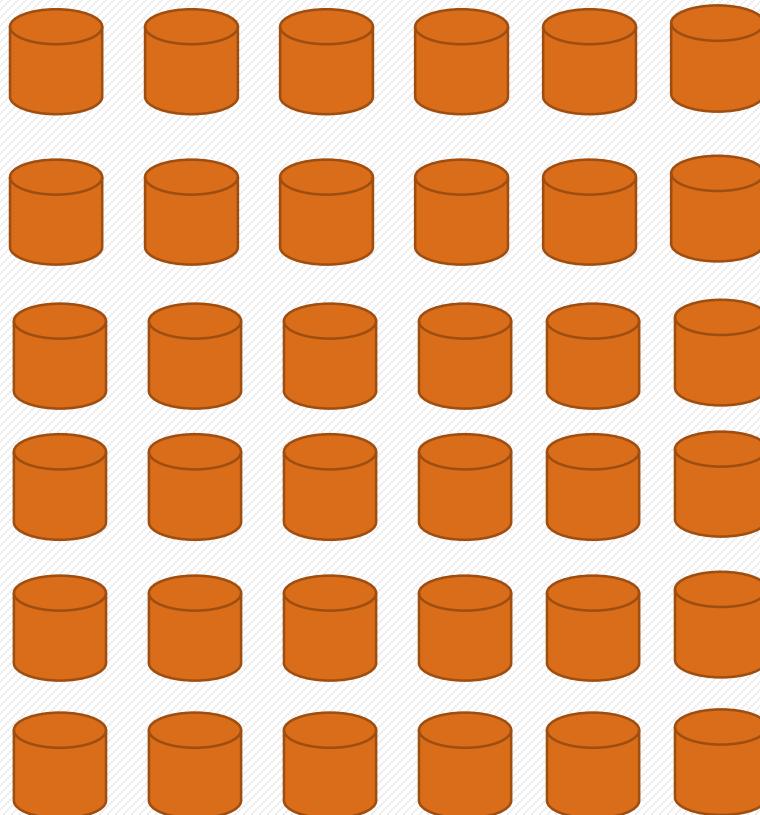
Yandex: 500+ servers, 25B rec/day

LifeStreet: 60 servers, 75B rec/day

CloudFlare: 36 servers, 200B rec/day

Bloomberg: 102 servers, 1000B rec/day

Toutiao: 400 servers, moving to 1000 this month



How fun 😊

```
life←{↑1 ω∨.∧3 4=+/ ,¬1 0 1°.⊖¬1 0 1°.∅⊂ω}
```

```
with (select groupArray(C) from C) as Ca
select id,
       groupArray(S) Sa, groupArray(V) Va, groupArray(D) Da,
groupArray(P) Pa,
       arrayMap(c -> arrayFirstIndex(s -> s > c, Sa)-1, Ca) Ka,
       arrayMap((c,k) -> Va[k] + (Va[k+1] - Va[k])/(Sa[k+1] -
Sa[k])*(c-Sa[k]),Ca,Ka) Ta,
       arrayMap(s -> arrayFirstIndex(c -> c>s, Ca)>0 ?
arrayFirstIndex(c -> c>s, Ca)-1 : toInt32(length(Ca)), Sa) Ja,
       arrayMap(i -> Ta[i], Ja) Ra,
       arrayMap((v,r) -> v - r, Va, Ra) ARa,
       arraySum((x,y,z) -> x*y*z, ARa, Da, Pa) result
from T group by id
```

What's new in 2018

- Table functions mysql/odbc/file/http
- clickhouse-copier
- Predicate pushdown for views/subselects
- LowCardinality datatype
- Decimal datatype
- JOIN enhancements
- ALTER TABLE UPDATE/DELETE
- WITH ROLLUP

... and tons of performance improvements and small features

More user friendly than ever!

- GDPR compliance – thanks to UPDATE/DELETE
- Easier BI integration – thanks to SQL compatibility changes and improvements in ODBC driver
- Easier cluster operation – thanks to clickhouse-copier, distributed DDL
- Easier integration with other systems. Thanks to:
 - Table functions
 - Kafka storage engine
 - Logs integration with Logstash, ClickTail
 - clickhouse-mysql for migration from MySQL

Case Study. Ivinco jump on to ClickHouse

Supports mature boardreader system

A lot of data collected from different sources

A lot of operational data (performance monitoring)

200TB in MySQL!

Operational problems

Hard to scale

Hard to make HA solution

Performance issues:

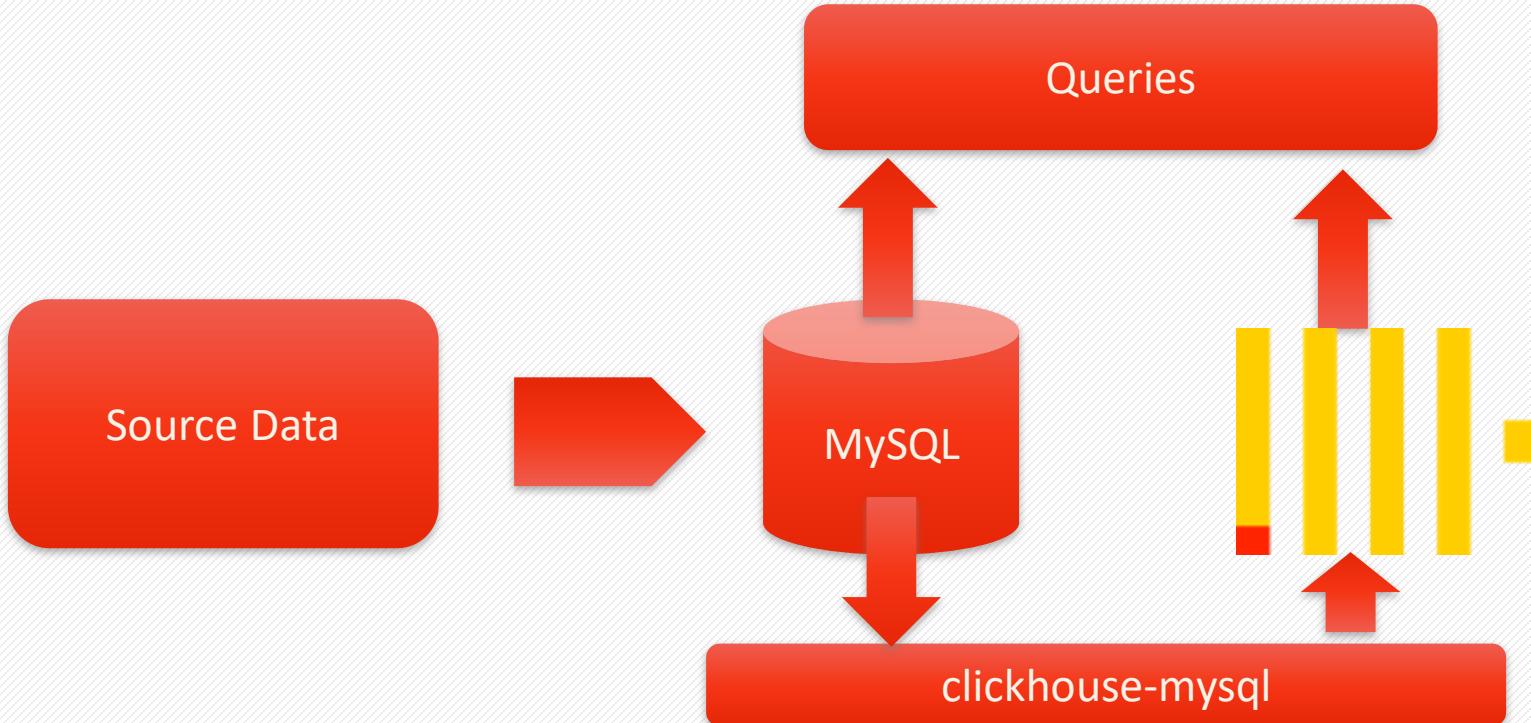
- ‘Manual’ partitioning and sharding
- Dozens of indexes per table etc.

Organizational problems

No development resources to rewrite

Minimal changes to current system are allowed

Binary log replication from MySQL to ClickHouse



See details at:

<https://www.altinity.com/blog/2018/6/30/realtime-mysql-clickhouse-replication-in-practice>

Results

Seamless integration of ClickHouse into the current system
No developers/coding involved, project is done with DevOps
Easy to test performance side by side
ClickHouse is 100 times faster
Now ready to re-write main system

More ways to integrate with MySQL

- mysql() table function
- MySQL table engine
- MySQL external dictionaries
- ProxySQL

mysql() table function

- Easiest and fastest way to get data from MySQL
- Load to CH table and run queries much faster

```
select * from mysql('host:port', database, 'table', 'user', 'password');
```

<https://www.altinity.com/blog/2018/2/12/aggregate-mysql-data-at-high-speed-with-clickhouse>

MySQL table engine

CREATE TABLE ...

```
Engine = MySQL('host:port', 'database', 'table', 'user', 'password'[, replace_query,  
'on_duplicate_clause']);
```

- SELECTs and INSERTs!
- No caching, data is queried from remote server

https://clickhouse.yandex/docs/en/operations/table_engines/mysql/

MySQL external dictionaries

- Makes data from mysql database accessible in ClickHouse queries
- Stores in memory
- Updates when the source data changes

```
SELECT dictGetString('dim_geo', 'country_name', geo_key) country_name,  
    sum(imps)  
FROM T  
GROUP BY country_name;
```

Accessing ClickHouse from MySQL



ProxySQL

ClickTail

- Log ingest based on honeycomb.io
- Understands Nginx Access Log, MySQL Slow Log, MySQL Audit Logs, MongoDB and Regex Custom Format
- Easily extensible with other formats

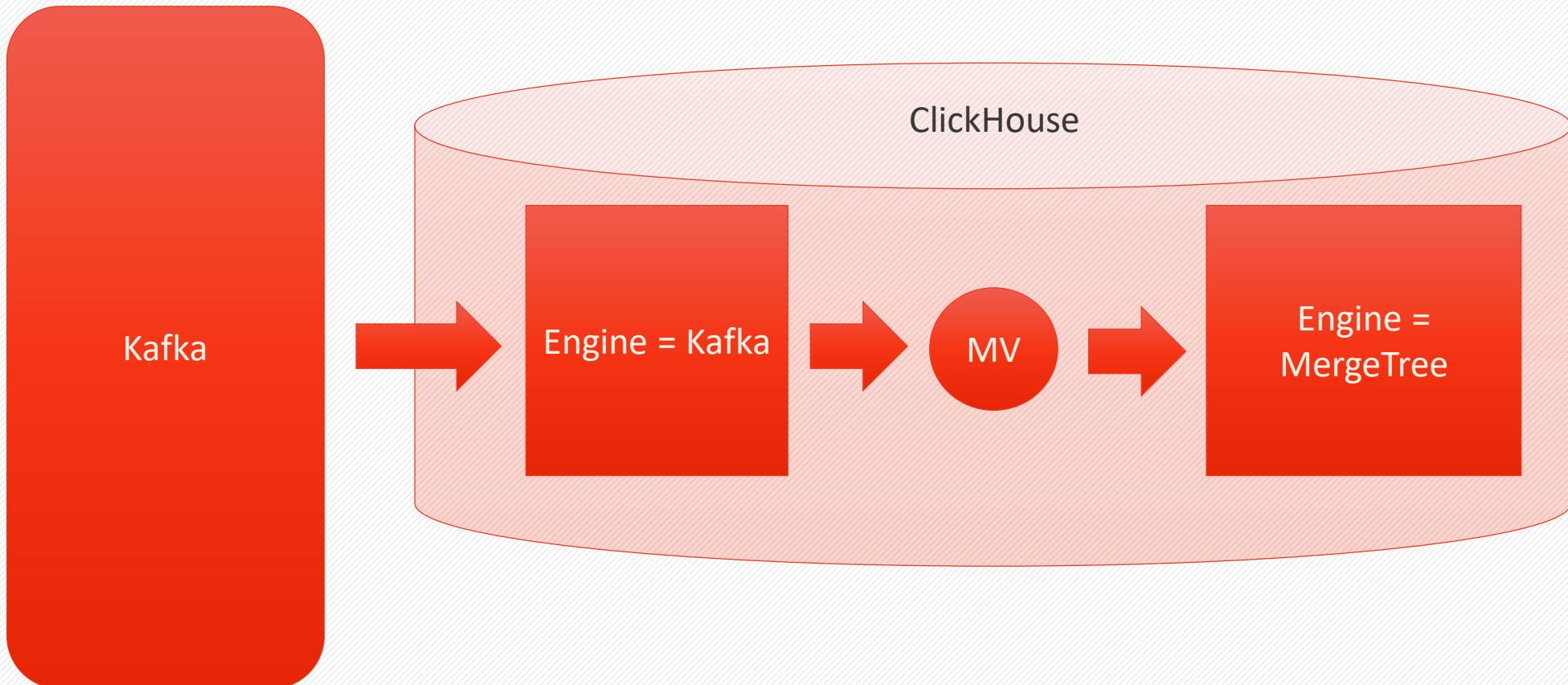
<https://github.com/Altinity/clicktail>

<https://www.altinity.com/blog/2018/3/12/clicktail-introduction>

<https://www.percona.com/blog/2018/02/28/analyze-raw-mysql-query-logs-clickhouse/>

<https://www.percona.com/blog/2018/03/29/analyze-mysql-audit-logs-clickhouse-clicktail/>

Kafka Engine



https://clickhouse.yandex/docs/en/operations/table_engines/kafka/

“Secret” Roadmap disclosed

ANSI SQL JOIN support:

- Multi-table joins – Q1/2019
- merge joins – Q2/2019

Protobuf/Parquet formats - Q4/2018

Per column compression/encoding settings – Q4/2018

Dictionary DDLs – Q1/2019

Secondary indexes – Q2/2019

LDAP integration, security enhancements -- Q2/2019

ClickHouse Today

Mature Analytic DBMS. Proven by many companies

2+ years in Open Source

Constantly improves – new cool features were added recently

Many community contributors

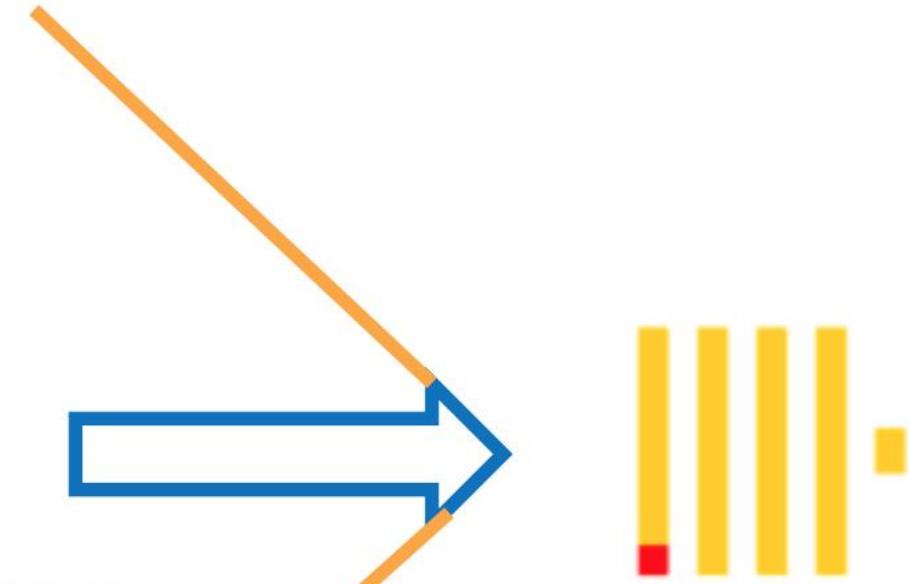
Emerging eco-system

Support from Altinity

ClickHouse Today



elasticsearch





Q&A

Contact me:

alexander.zaitsev@lifestreet.com

alz@altinity.com

skype: alex.zaitsev

telegram: @alexanderzaitsev