ClickHouse Summer Meetup

July 3, 2018. Berlin

Agenda:

7:00pm: ClickHouse introduction - Alexander Zaitsev (*Altinity*)

7:30pm: Using ClickHouse for experimentation metrics at Spotify - Gleb Kanterov (*Spotify*)

8:20pm: Deep dive into ClickHouse internals - Aleksey Milovidov (*Yandex*)

ClickHouse Analytical DBMS

Introduction

Alexander Zaitsev, Altinity

Delivery Hero, Berlin, 3 Jul 2018

What Is ClickHouse?

ClickHouse DBMS is

- Column Store
- MPP
- SQL
- Open Source

ClickHouse Timeline

- Developed in Yandex in 2012-2015
- Open Sourced June 2016
- First non-Yandex deployments Q4 2016
- Hundreds of companies by Q2 2018

Why Yet Another DBMS?

Vertica
MemSQL
Actian
SnowFlake
RedShift

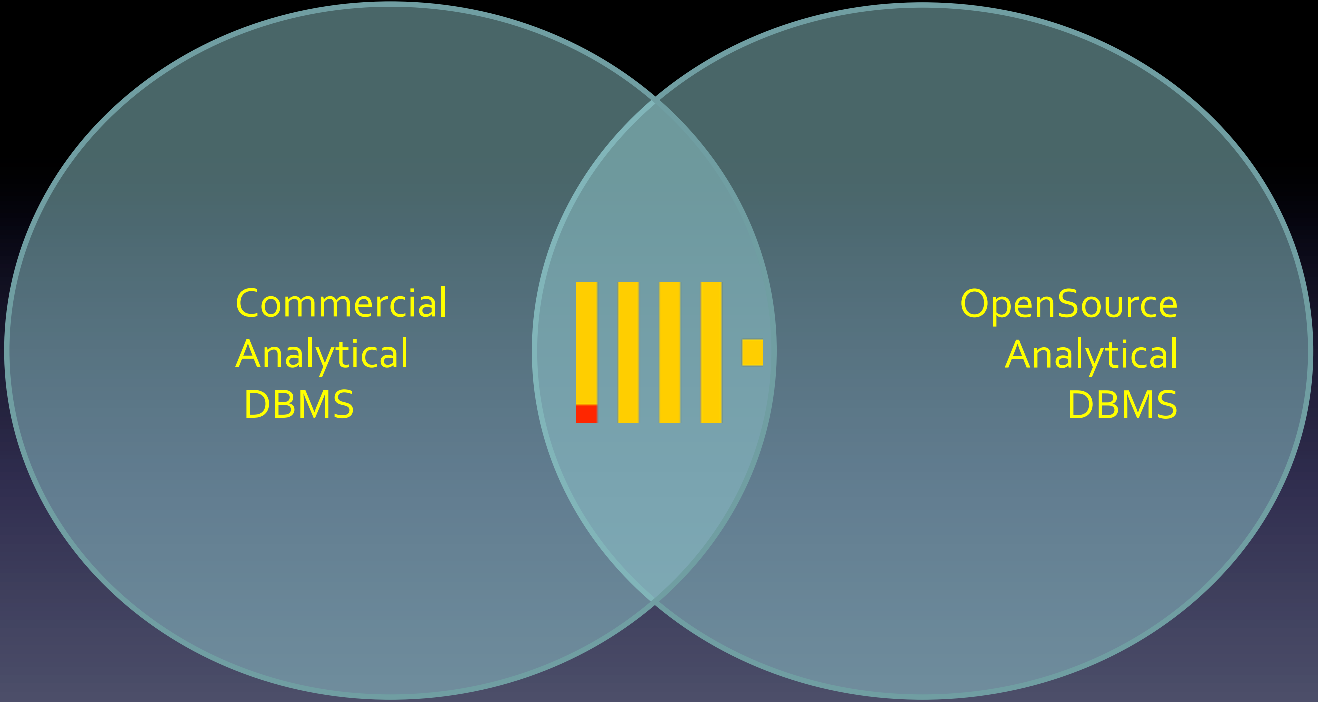
EXPENSIVE

InfiniDB (MariaDB cs)
InfoBright
MonetDB
GreenPlum
Spark

SLOW or LIMITED

Commercial
Analytical
DBMS

OpenSource
Analytical
DBMS



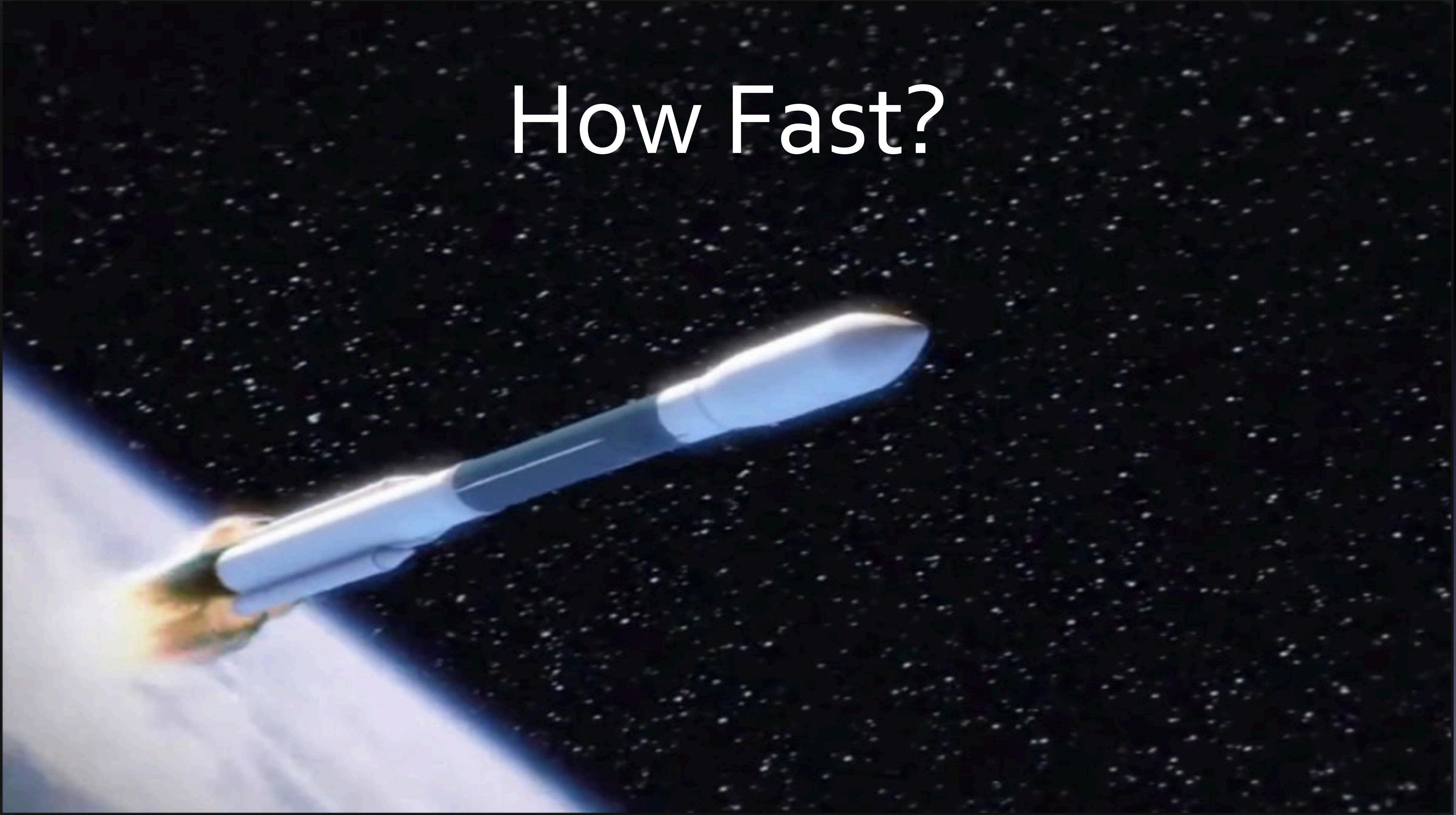
Commercial
Analytical
DBMS

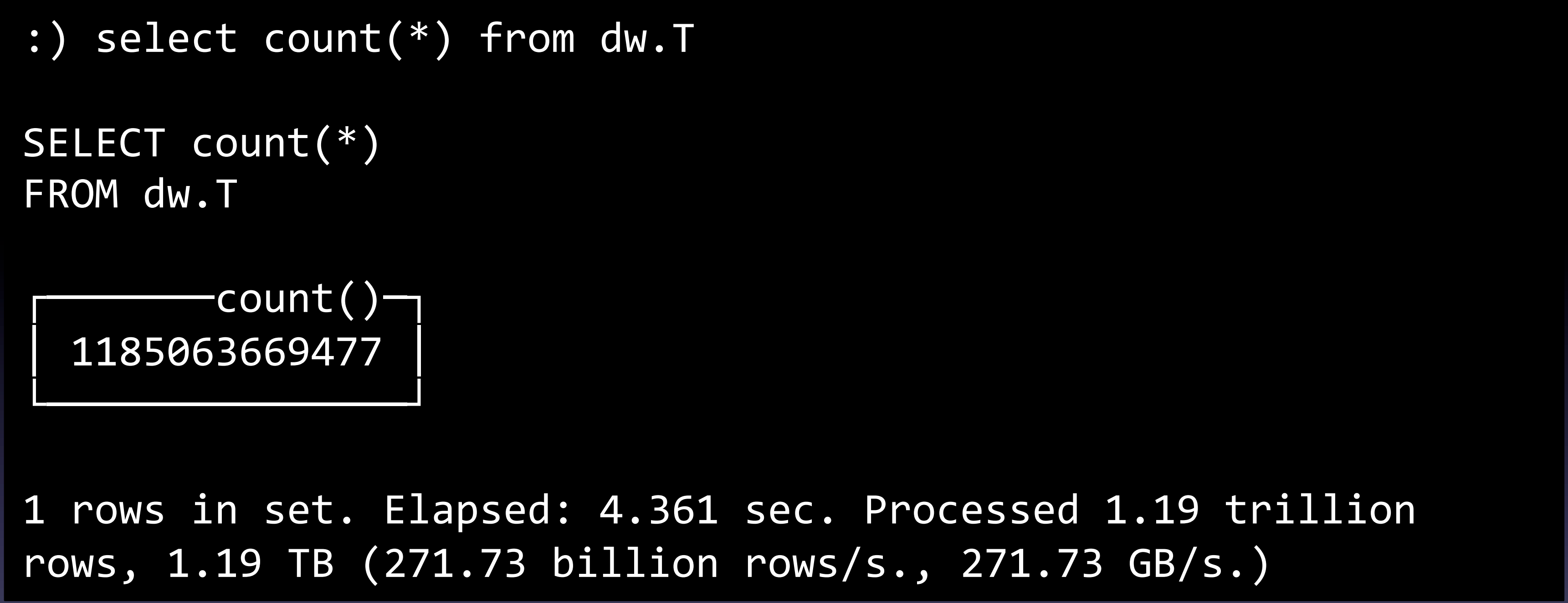
OpenSource
Analytical
DBMS

ClickHouse

- Fast!
- Flexible!
- Free!

How Fast?





```
:) select count(*) from dw.T

SELECT count(*)
FROM dw.T

┌───────count()─┐
│ 1185063669477 │
└───────────────┘


1 rows in set. Elapsed: 4.361 sec. Processed 1.19 trillion
rows, 1.19 TB (271.73 billion rows/s., 271.73 GB/s.)
```

“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
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

"1.1 Billion Taxi Rides Benchmarks"

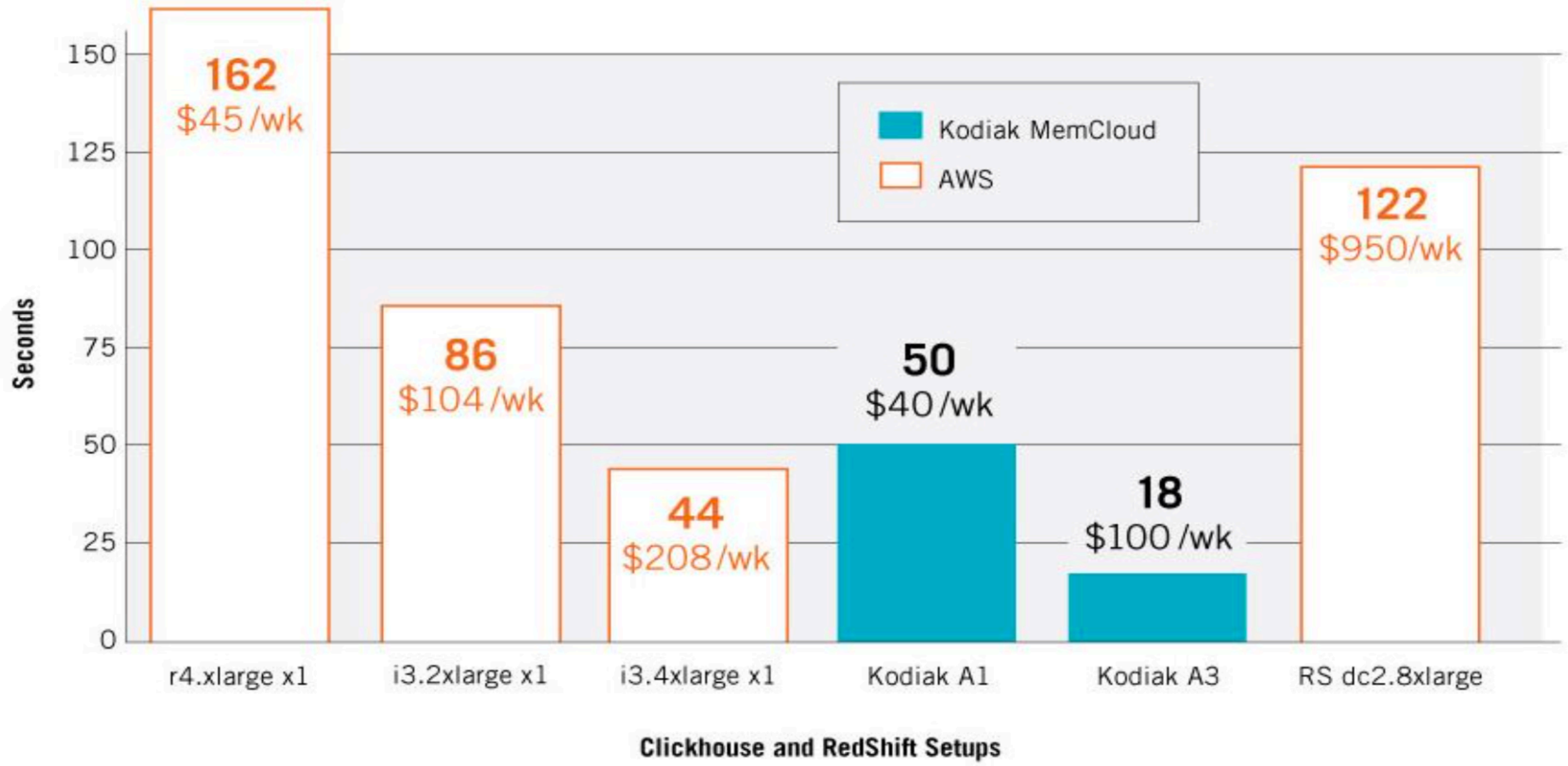
http://tech.marksblogg.com/benchmarks.html

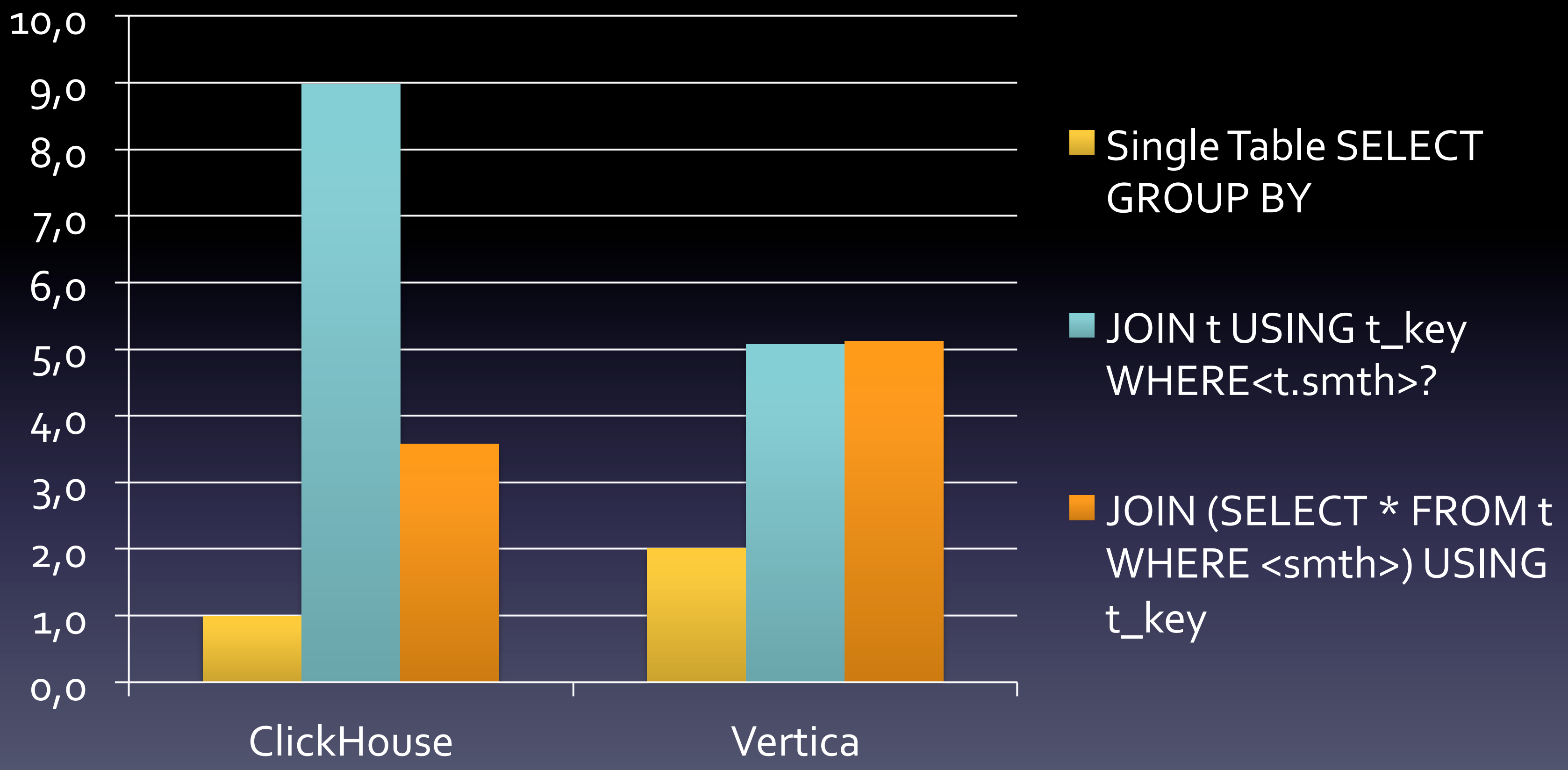
Query 1	Query 2	Query 3	Query 4	Setup
0.034	0.061	0.178	0.498	MapD & 2-node p2.8xlarge cluster
0.051	0.146	0.047	0.794	kdb+/q & 4 Intel Xeon Phi 7210 CPUs
-	**2.415**	**3.599**	**4.962**	**ClickHouse at Kodiak Data server**
0.762	2.472	4.131	6.041	BrytlytDB 1.0 & 2-node p2.16xlarge cluster
1.034	3.058	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

ClickHouse runs at

- Bare metal (any Linux)
- Amazon
- Azure
- Kubernets, VM Ware etc.
- Kodiak Data cloud

Total Query Time (For different ClickHouse and RedShift setups, less is better)



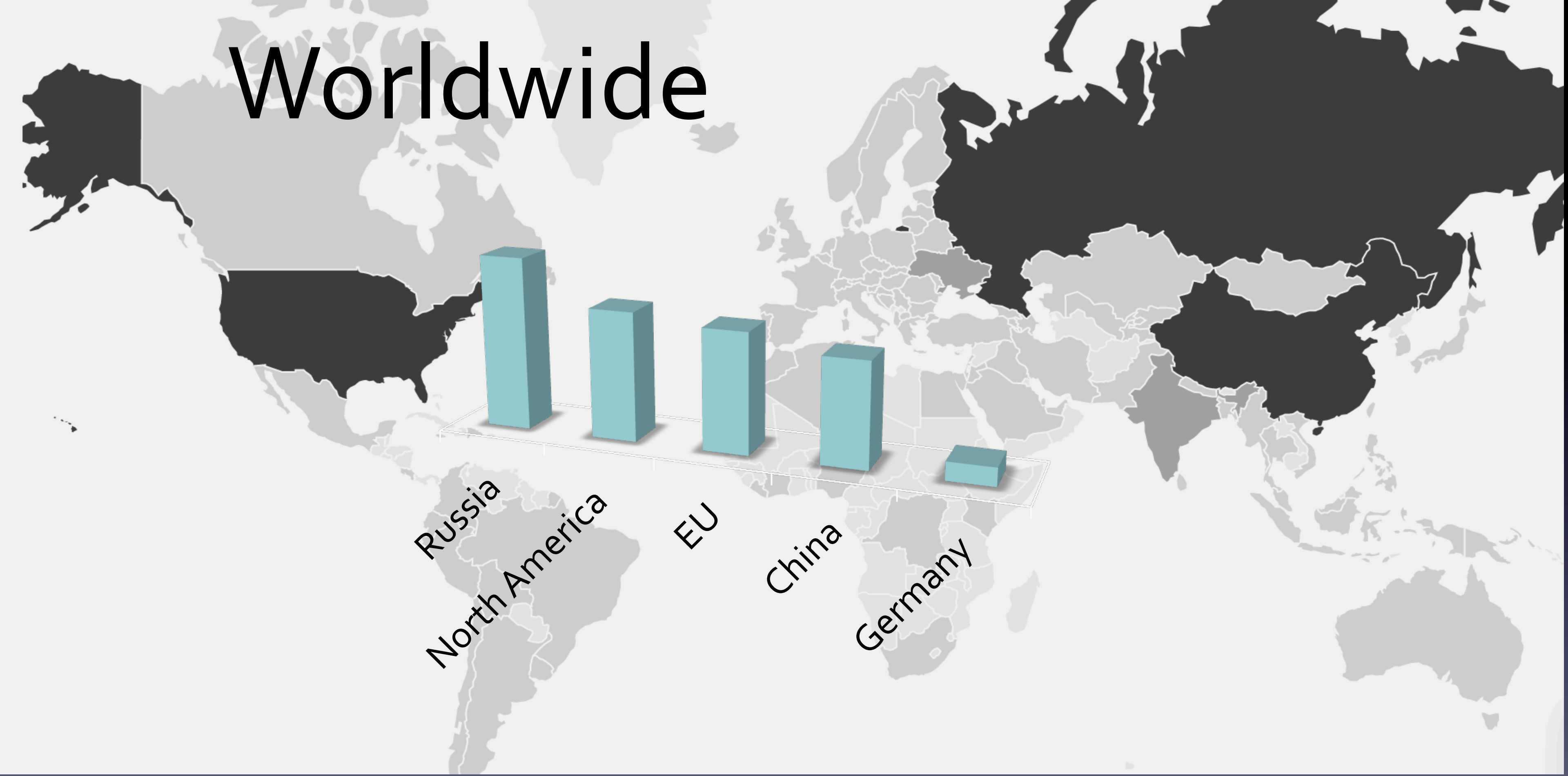


- 19 queries, 1200M rows table, 3-node clusters

Real companies are using ClickHouse for:

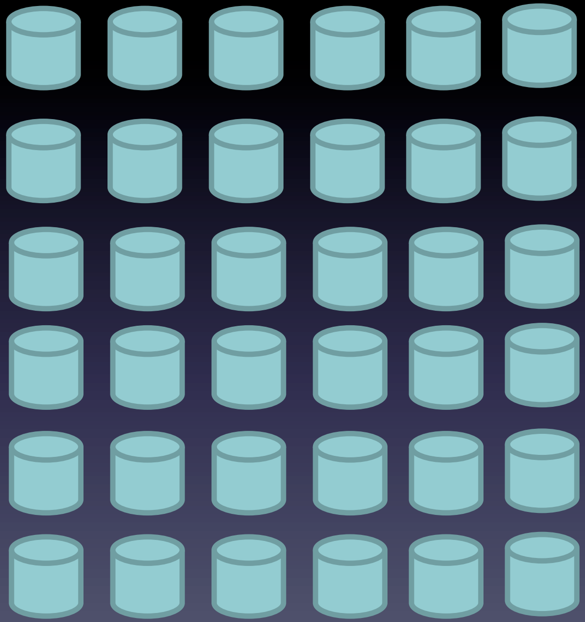
- Mobile App and Web analytics
- AdTech bidding analytics
- Operational Logs analytics
- DNS queries analysis
- Stock correlation analytics
- Telecom
- Security audit
- Fintech SaaS
- Manufactoring process control
- BlockChain transactions analysis

Worldwide



* www.altinity.com visits in 2018

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

Happy Migrations!

- From MySQL/InfoBright/ PostreSQL/Spark to ClickHouse → SPEED!

- From Vertica/RedShift to ClickHouse → COST! VENDOR UN-LOCKING!

03.07	19:00	CLICKHOUSE	GATE 2	boarding
03.07	19:30	CLICKHOUSE	GATE 3	
03.07	20:00	CLICKHOUSE	GATE 4	

Few Case Studies

LIFESTREET

- Ad Tech (ad exchange, ad server, RTB, DMP etc.)
- Ad Optimization, programmatic bidding
- A lot of data:
  - 10,000,000,000+ events/day
- A lot of queries: users and algorithms

Used Vertica, but needed to move

- Data sizes constantly grow
- Estimated PBs
- Vertica license would be too expensive

… migration was not easy

* More details at October 2017 Berlin Meetup

Major Design Decisions

- Dictionaries for star-schema design
- Extensive use of Arrays
- SummingMergeTree for realtime aggregation
- Smart query generation
- Multiple shards and replicas

Results

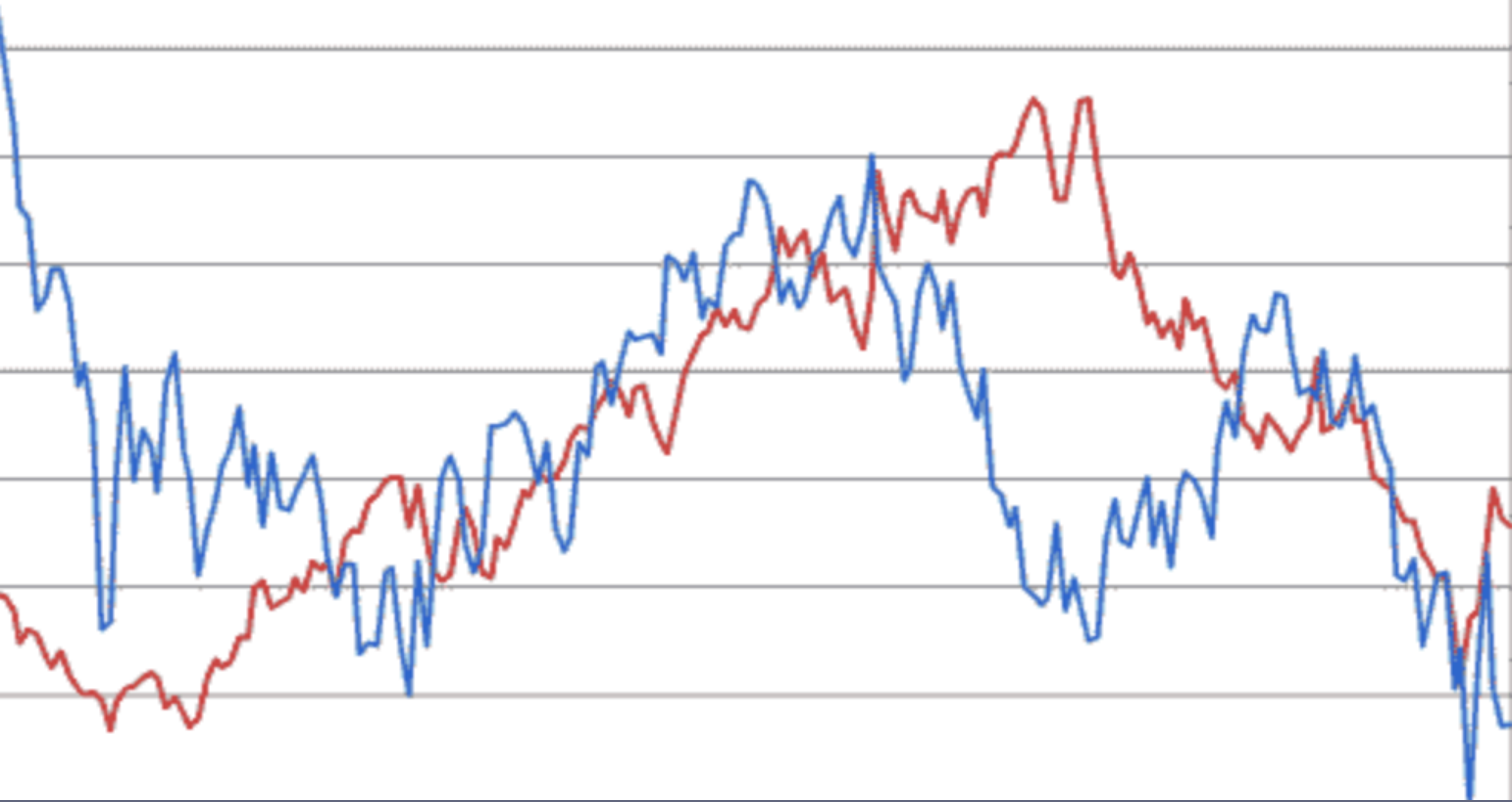
- Successful migration, 1y+ in production
- Better performance and flexibility
  - 75B rows/day
  - 1M rows/sec in peak hours
  - 1.3M SQL queries /day
- 30% hardware cost reduction (less expensive storage):
- No license cost and limits:
  - 3PB of raw data
  - 6,000 billion rows

Powered by:

Case 2. Fintech Company

- Stock Symbols Correlation Analysis
- 5000 Symbols
- 10 years of data

100B data points



Challenge

- (time, symbol, price) – 100 billion
- log_return = runningDifference(log(price)) – 100 billion times
- corr(s1,s2) = corr(log_return(s1),log_return(s2))

For every pair (s1,s2) from 5000 s(i), 12.5M pairs overall

- Group by hours

$$\frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}}$$

Calculate 12,500,000 times
For every hour!

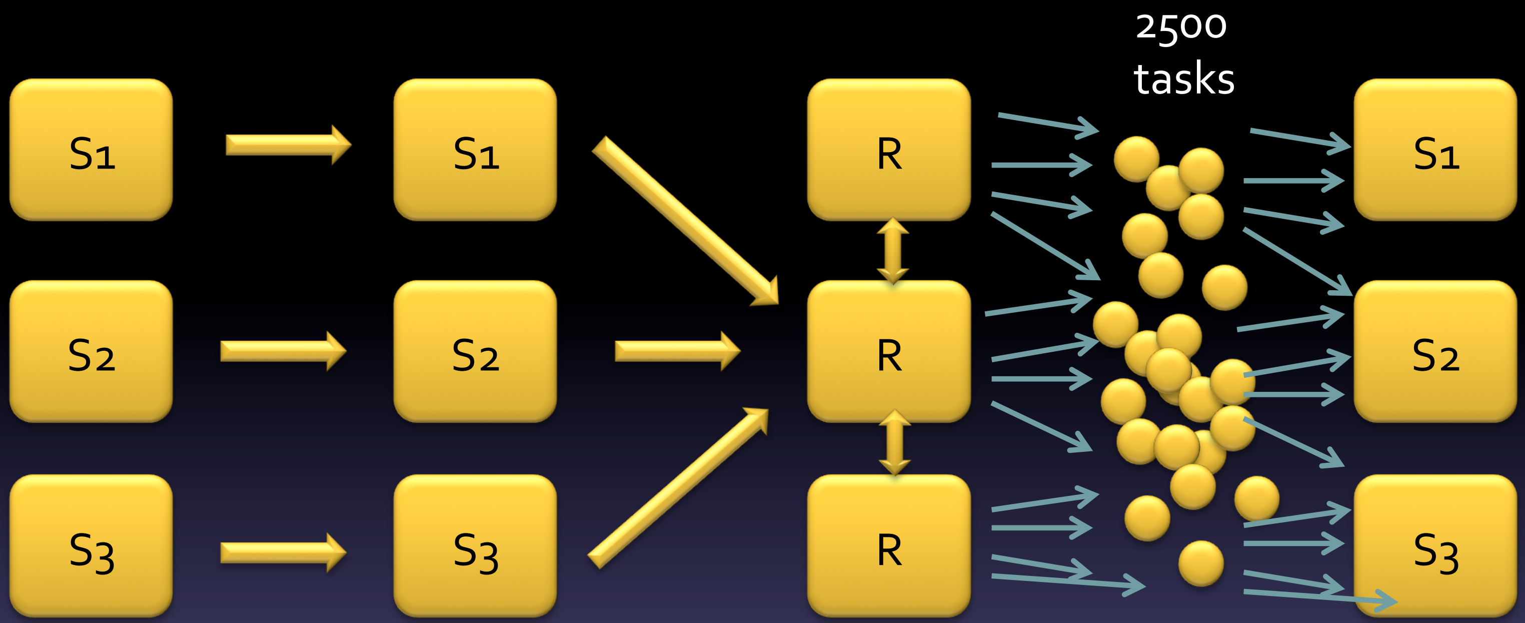
Very slow

Tried…

- Hadoop
- Spark
- Greenplum

Slow
(weeks to complete)
Expensive to scale

ClickHouse



time
symbol
price

time
symbol
logReturn(price)

time
groupArray(symbol)
groupArray(logRet..)

date+hour
corr(S(i),S(j))

POC Performance Results

- 3 servers setup
- 2 years, 5000 symbols:
  - log_return calculations: ~1 h (distributed)
  - Converting to arrays: ~ 1 h (almost distributed)
  - Correlations: ~50 hours (also distributed)
    - 12,5M/50h = 70/sec

Distributed => it scales easily!

Case 3. Ivinco

- Mature boardreader system
- A lot of data collected from different sources
- A lot of operational data (performance monitoring)

200TB in MySQL!

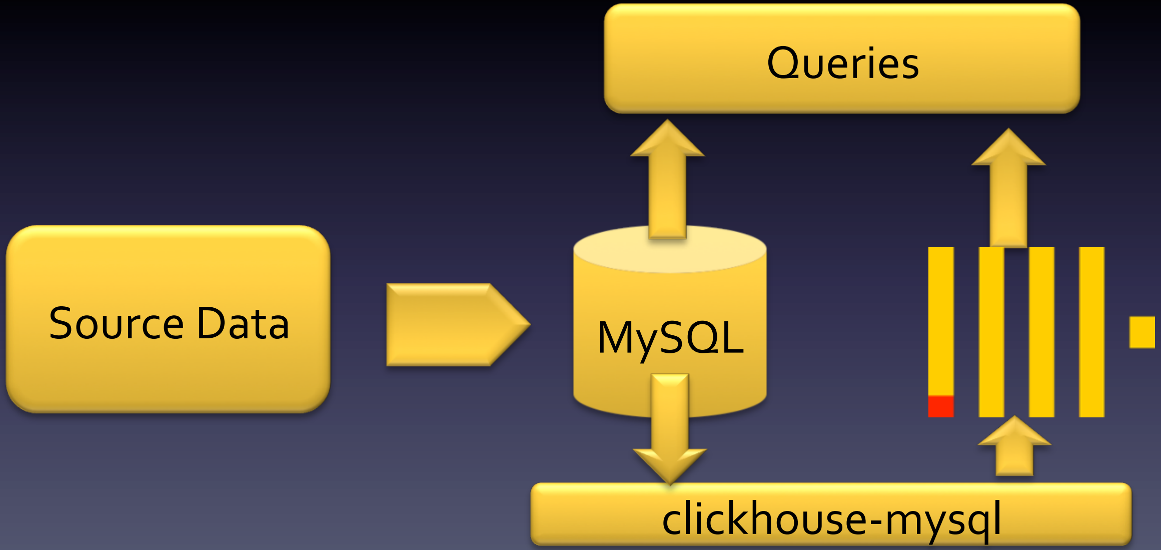
Operational problems

- 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



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

ClickHouse Today

- Mature Analytic DBMS. Proven by many companies
- 2+ years in Open Source
- Transparent development roadmap
- Many community contributors
- Emerging eco-system (tools, drivers, integrations)
- Support and Consulting from Altinity

Q&A

Contact me:

alexander.zaitsev@lifestreet.com
alz@altinity.com
skype: alex.zaitsev
telegram: @alexanderzaitsev

Altinity