ClickHouse In Real Life
Case Studies and Best Practices

Alexander Zaitsev, LifeStreet/Altinity
Percona Live 2018
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
ClickHouse is

• Fast
• Flexible
• Scalable

How does it work in real?
What Is It For?
What Is It For?

• Fast analytical queries
• Low latent data ingestion/aggregation
• Distributed computations
• Fault-tolerant data warehousing

All scaled from 1 to 1000s servers
Who Is It For?
Who Is It For?

• Analysts/Developers/DevOps
  • who need analyze huge amounts of data
• Startups
  • build high performance analytics with low investment
• Companies
  • having performance problems with current systems
  • paying too much for license or infrastructure
Successful Production Deployments

- DNS queries analytics (CloudFlare)
- AdTech (multiple companies worldwide)
- Operational logs analytics (multiple companies worldwide)
- Stock correlation analytics, investor tools (Canadian company)
- Hotel booking analytics SaaS (Spanish company)
- Security audit (Great Britain, USA)
- Fintech SaaS (France)
- Mobile App and Web analytics (multiple companies worldwide)
Evaluating/implementing:

- Telecom companies
- Satellite data processing
- Search engine ranking analytics
- Blockchain platform analysis
- Manufacturing process control
Happy Transitions!

• From MySQL/InfoBright/PostgreSQL/Spark to ClickHouse

• From Vertica/RedShift to ClickHouse

SPEED!

COST!

VENDOR UN-LOCKING!

24.04 16:50 CLICKHOUSE GATE 2 boarding
24.04 19:30 CLICKHOUSE GATE 3
24.04 20:00 CLICKHOUSE GATE 4
Case Studies

• Migration from Vertica to ClickHouse
• Distributed Computations and Analysis of Financial Data
• Blockchain Platform Analytics
• ClickHouse with MySQL
Case 1.

- Ad Tech (ad exchange, ad server, RTB, DMP etc.)
- Creative optimization, programmatic bidding
- A lot of data:
  - 10,000,000,000+ bid requests/day
  - 2-3K event record (300+ dimensions)
  - 90-120 days of detailed data

\[10^B \times 3K \times [90-120] = [2.7-3.6]PB\]
Business Requirements

• Ad-hoc analytical reports on 3 months of detail data
• Low data and query latency
• High Availability
• Tried/used/evaluated:
  • MySQL (TokuDB, ShardQuery)
  • InfiniDB
  • MonetDB
  • InfoBright EE
  • Paraccele (now RedShift)
  • Oracle
  • Greenplum
  • Snowflake DB
  • Vertica
Main Migration Challenges

- Efficient star-schema for OLAP
- Reliable data ingestion
- Sharding and replication
- Client interfaces
Sharding and Replication

Table 1

S1  S2  S3  S4  Sn

Replica 1

S1  S2  S3  S4  Sn

Replica 2

S1  S2  S3  S4  Sn

Replica 3
Major Design Decisions

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

• Successful migration and cost reduction
• Increased performance and flexibility
• 60 servers in 3 replicas
• 2-3PB of data
• 6,000B+ rows in fact and aggregate tables (50B+ daily load)
• 1M+ SQL-queries/day

Powered by: [Link]
Case 2. Fintech Company

• Stock Symbols Correlation Analysis
• 5000 Symbols
• 100ms granularity
• 10 years of data

100B data points
Main Challenge

• Symbols $S(1) .. S(5000)$
• Time points $T(1) ... T(300M)$
• $\log_{\text{return}}(n)(m) = \text{runningDifference}(\log(\text{price}(n)))$
• $\text{corr}(n1,n2) = \text{corr}(\log_{\text{return}}(n1),\log_{\text{return}}(n2))$
• For every tuple $(n1,n2)$, 12.5M tuples altogether

$$\frac{\sum(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum(X - \bar{X})^2 \sum(Y - \bar{Y})^2}}.$$ calculate 12,500,000 times!
Tried...

- Hadoop
- Spark
- Greenplum

ClickHouse

Slow (weeks to complete)
Expensive to scale
Distributed Computations

• Distribute data across N servers
• Calculate log_return for every symbol at every server using Arrays:
  • (timestamp, Array[String], Array[Float32])
• Distribute correlation computations across all servers
  • Batch planning
POC Performance Results

• 3 servers setup
• 2 years, 5000 symbols:
  • log_return calculations: ~1 h
  • Converting to arrays: ~1 h
  • Correlations: ~50 hours
    • 12,5M/50h = 70/sec

And is scales easily!
Case 3. Bloxy.info - Etherium network analysis

- 450M transactions
- Transaction level interactive reports
- Transaction graph navigation
- Aggregate reports
- Rich visualization
Tried

- MySQL

ClickHouse
Main Challenge:

ClickHouse is bad for point queries!
Main Design Decisions

• Encode transaction IDs to binary
• ClickHouse MergeTree with low index_granularity
• Materialized Views for different sort orders
• Apache SuperSet for visualization
### Top ETH Senders today

<table>
<thead>
<tr>
<th>Sender address</th>
<th>Total amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x876eabf441b2ee5b5b0554fd502a8e0600950cfa</td>
<td>678k</td>
</tr>
<tr>
<td>0xf4b51b14b9ee30dc37ec970b50a486f37686e2a8</td>
<td>470k</td>
</tr>
<tr>
<td>0xc0d5c4d8bc3294ce7ab2e37a2c85fa03a506e570</td>
<td>220k</td>
</tr>
<tr>
<td>0x3f5ce5fbeb3e9af3971dd833d26ba9b5c936f0be</td>
<td>204k</td>
</tr>
<tr>
<td>0x008024771614f4290696b63ba3dd3a1ceb34d4d9</td>
<td>123k</td>
</tr>
<tr>
<td>0xef54f559b5e3b55b783c7bc59850f83514b6149c</td>
<td>102k</td>
</tr>
<tr>
<td>0xde30e7c5855ed92ace9160deaa331b50f2e6fdbf</td>
<td>88.7k</td>
</tr>
<tr>
<td>0xb518106e3eb59ba13bb4707c62215f64cbc992d</td>
<td>77.0k</td>
</tr>
<tr>
<td>0xc257274276a4e539741ca11b590b9447b26a8051</td>
<td>70.4k</td>
</tr>
<tr>
<td>0xd24400ae8bfebb18ca49be86258a3c749cf46853</td>
<td>62.8k</td>
</tr>
</tbody>
</table>
http://stat.bloxy.info/superset/dashboard/today/?standalone=true
Mystical Mixer

Histogram of addresses, receiving ETH from N unique senders

Histogram of addresses, sending ETH to N unique receivers (last 100 days)
And more: http://bloxy.info

• Etherium Mixer Analysis
• Token Dynamics
• Token Distribution
• ERC721 Token and Collectibles
• ICO Analysis and Trends
• Smart Contract Events and Methods
• Etherium Mining
• DAO Efficiency Analytics
Case 4. ClickHouse with MySQL

• Accessing MySQL from ClickHouse
• Accessing ClickHouse from MySQL
• Streaming data from MySQL to ClickHouse
• Analyzing MySQL logs with ClickHouse
Accessing MySQL from ClickHouse

- External dictionaries from MySQL table
  - Map mysql table to in-memory structure
- Mysql() function

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

Accessing ClickHouse from MySQL
Streaming Data from MySQL to ClickHouse

https://github.com/Altinity/clickhouse-mysql-data-reader
Combine together MySQL and ProxySQL.
Analyzing MySQL logs with ClickHouse

- MySQL Logs may grow large

Main Lessons

• Schema is the most important
  • Proper data types
  • Arrays
  • Dictionaries
• Summing/Aggregating MergeTree for realtime aggregation
• Materialized Views if one key is not enough
• Reduce Index granularity for point queries
• Distribute data and load as uniform as possible
• Integrate smartly
ClickHouse is

• Fast
• Flexible
• Scalable

And it really works!
Q&A

Contact me:

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