AFTERWORDS - INTELLIGENT CUSTOMER ENGAGEMENT

Industry: Technology | **Location:** Florida, USA | **Solution:** ClickHouse Nested arrays

KEY HIGHLIGHTS

Challenges

- Many-to-many relationships were slow and complex
- Spending much development time on maintaining master tables

Solutions

- ClickHouse Nested arrays
- External data for query processing

Impact / Benefits

- Answering any "check basket" type of question with exemplary speed
- Ability to send multiple temporary tables together with Select statement in one roundtrip
- Create ready to use ClickHouse cluster of any size in the cloud, ClickHouse is very easy to configure and administrate.

CUSTOMER STORY

AfterWords

AfterWords is an intelligent customer engagement system that delivers questions based on transactional history. The mission of the company is simple – provide easy-to-use software, actionable data, and industry expertise that helps customers provide the best customer experience possible!

ClickHouse Usage

Www.surveyafterwords.com is a growing project, which is mostly targeted at big restaurant chains and allows to produce surveys tied to a check-level customer data.

For a database engine which powers all the reporting in our project, we chose ClickHouse and so far, we are very pleased. ClickHouse works like a Swiss watch scales up and out perfectly and has many unique features which add so much more appeal to it.

Instead of repeating other people about speed and scalability of ClickHouse (so much said about it), I am going to describe a few of those special ClickHouse features and illustrate why they are so useful for us. I would prefer not to repeat what other people said already, like "Our ClickHouse cluster can ingest 100M records in few seconds", it is true but not much informative since many other people already wrote about it.

Aside from the speed and scalability, most of ClickHouse features were liked by AfterWords.

Nested Structures

There are many good use cases for nested structures, but I'll describe just one.

Many-to-many relationships and their most frequent example in our data domain - so-called "check basket" problem. In essence, the question you ask your DBMS is like:

- "how many sales checks I have, where beer was sold together with fish?"
- or "how many checks I have, where there are a beer and steak sold and there is no fish?"
- or "how many surveys I have, where the customer gave answer X together with Answer Y but without Answer Z?".

In RDBMS world, answering such questions means expensive joins because pre-aggregating data for not known in advance questions is not feasible. In OLAP world, there are some solutions for many-to-many relationships, but they are slow and complex, and this is the problem, because, of course, the Marketing Department would want their reports to show summaries by region, by year and aggregated for thousands of stores. And within a second or two.

With ClickHouse Nested arrays, it is possible to put any number (there are some limits, but they are large enough) of child rows into one master row, so there is no need for those expensive joins I mentioned above. Functions like arrayCount, arraySum, hasAll, hasAny allow us to easily answer any "check basket" type of question with exemplary speed.

> "Surely, there are other SQL-like DBMS engines which support nested arrays, but none of them scales like ClickHouse, none of them is fast and free like ClickHouse. "
>
> *Oleg Khozyainov, AfterWords*

External Data For Query Processing

The ability to send multiple temporary tables together with Select statement in one roundtrip is a very convenient feature. We found that sometimes it is easier to send a few temp tables (to filter and group the results) together with a query, instead of spending much development time on maintaining master tables or dictionaries for ClickHouse. Amazingly, the query like

```
select count() from MyTable where StoreID in storelist and ItemID in itemlist
```

sent together with two tables, 100K rows for store list and 1M rows for item list finished in similar few milliseconds as a regular query sent without temp tables, so the overhead of sending temp tables together with a select query seems to be very small.

ClickHouse In The Cloud

It was relatively easy to write deployment script which creates ClickHouse cluster of any size completely ready to use. Our script has parameters like: number of Clickhouse nodes, number of Zookeeper nodes, size of each node, size of a disk for each node, etc. The script creates all the cloud environment like IP addresses, load balancers, nets, subnets, nodes, it installs ClickHouse and Zookeeper on those nodes, creates ClickHouse config files with cluster configuration inside, creates Zookeeper config files, configures SSL certificates, etc.

In a single click, we can create ready to use ClickHouse cluster of any size in the cloud, ClickHouse is very easy to configure and administrate.

> "As a final note, I would like to express our big gratitude to guys who created and continue to create ClickHouse. It is amazing software, with so many unique features, easy to use, fast and scalable like nothing else. Thank you! "
>
> *Oleg Khozyainov, AfterWords*

Written by Oleg Khozyainov who shares his experience of using ClickHouse as part of work done for www.surveyafterwords.com

SOLUTION COMPONENTS

A nested data structure is like a nested table. The parameters of a nested data structure – the column names and types – are specified the same way as in a CREATE query. Each table row can correspond to any number of rows in a nested data structure.

ClickHouse allows sending a server the data that is needed for processing a query, together with a SELECT query. This data is put in a temporary table and can be used in the query (for example, in IN operators).