

*Афанасьев Г.И.*

*кандидат технических наук*

*доцент кафедры «Системы обработки информации и управления»*

*МГТУ им. Н.Э.Баумана*

*Россия, г. Москва*

*Белоногов И.Б.*

*старший преподаватель кафедры «Системы обработки информации и*

*управления»*

*МГТУ им. Н.Э.Баумана*

*Россия, г. Москва*

*Булатова И.Г.*

*доцент кафедры «Системы обработки информации и управления»*

*МГТУ им. Н.Э.Баумана*

*Россия, г. Москва*

**СРАВНЕНИЕ «СТРОКОВОЙ» СУБД POSTGRESQL И  
«СТОЛБЦОВОЙ» СУБД CLICKHOUSE НА НАБОРЕ  
СТАТИСТИЧЕСКИХ ВРЕМЕННЫХ ДАННЫХ В РАМКАХ OLAP**

*Аннотация:* В данной статье рассматриваются результаты эксперимента по сравнению быстродействия работы СУБД PostgreSQL и СУБД ClickHouse на примере реализации аналитических запросов в реальном масштабе времени на наборе статистических временных данных

*Ключевые слова:* СУБД, PostgreSQL, ClickHouse, OLAP, базы данных

*Abstract:* In this article it is considered the results of an experiment comparing the performance of the PostgreSQL DBMS and the ClickHouse DBMS with the example of implementing OLAP queries on a set of statistical time data

*Keywords:* DBMS, PostgreSQL, ClickHouse, OLAP, databases

## **Введение**

В настоящее время широкое распространение получили реляционные системы управления базами данных (СУБД), одним из основных логических элементов которых являются таблицы. Как известно, таблицу можно рассматривать как множество либо строковых данных, или также, как и множество столбцовых данных. Наиболее широкое распространение в настоящее время имеют так называемые «строковые» СУБД, одним из популярных представителей которых является СУБД PostgreSQL. Однако в ряде случаев при обработке данных, например для OLAP (online analytical processing) обработки, наиболее целесообразным способом представления и хранения данных является представление данных в виде столбцовых данных. Одной из набирающих популярность СУБД для организации OLAP обработки данных является появившееся не так давно на рынке СУБД ClickHouse от российской компании Yandex. В связи с этим подчас возникают вопросы связанные с тем, что на каких наборах данных и для каких задач целесообразнее использовать тот или иной тип СУБД

### **«Столбцовая» СУБД ClickHouse**

И так, что же такое СУБД ClickHouse? Это так называемая «столбцовая» система управления базами данных (СУБД) для аналитической обработки запросов реального времени (online analytical processing OLAP) [1].

Основные особенности ClickHouse:

- Все столбцы хранятся в отдельных файлах, что позволяет обрабатывать только те столбцы, к которым есть аналитический запрос. В случае, когда в таблице больше 10 столбцов, а запрос строится по 5 из них, то количество операций ввода/вывода уменьшается в минимум в 2 раза.
- СУБД использует так называемый векторный движок, который позволяющий производить обработку данных по фрагментам столбцов.
- ClickHouse строго использует значения постоянной длины, чтобы очистить данные от мусора в виде специальных символов. Что

приводит к уменьшению нагрузки на CPU и повышает пропускную способность.

- ClickHouse СУБД является многопоточным, и обработка запросов распараллеливаются естественным образом.
- Первичный ключ обязателен и должен является датой. Каждый пулл данных с одинаковым первичным ключём ClickHouse хранит в одном образе.
- Существует возможность в реализации репликации и шардировании данных.

Для работы репликации (хранение метаданных и координация действий) требуется ZooKeeper [2]. ClickHouse будет самостоятельно обеспечивать консистентность данных на репликах и производить восстановление после сбоев.

В рамках той достаточно узкой ниши, в которой находится ClickHouse, у него до сих пор нет достойных альтернатив. В рамках более широкой области применения, ClickHouse может оказаться выгоднее других систем с точки зрения скорости обработки запросов, эффективности использования ресурсов и простоты эксплуатации.

### **Топология кластеров для СУБД ClickHouse и PostgreSQL**

Для реализации минимально возможного хранилища с использованием СУБД ClickHouse, отвечающего по надежности, высокой доступности и скорости обработки, необходимо порядка 5 серверов [3]. Данные были разделены на 2 части, и каждая часть хранится в двух экземплярах. Минимальные требования к серверу: 512MB RAM, 1 CPU, 2GB SWAP, Ubuntu 14.04 x64.

На рисунке 1 изображена топология кластера с использованием СУБД ClickHouse.

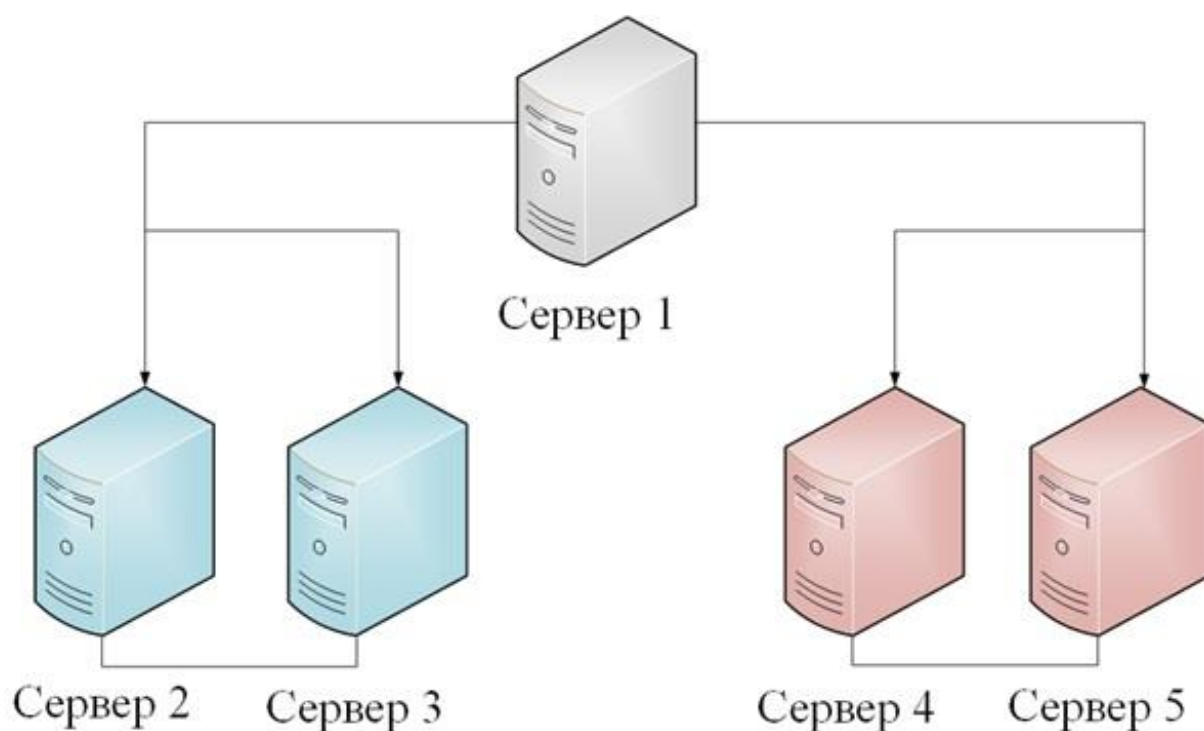


Рисунок 1. Топология кластера с использованием СУБД ClickHouse

Сервер 1 обеспечивает обработку данных и занимается распараллеливанием задач между репликами. Он собирает данные присланные от подчиненных серверов и формирует ответ на запрос. На этом сервере данные не хранятся.

Серверы 2,3 хранят первую половину данных. Эти сервера хранят одинаковые данные и постоянно синхронизируются. Они образуют 1-ый шард.

Серверы 4,5 хранят вторую половину данных. Оба сервера хранят одинаковые данные и постоянно синхронизируются. Они образуют 2-ой шард.

В качестве сравнения, взят кластер из 4 серверов, практически идентичных по мощности с установленным PostgreSQL. Конфигурация каждой машины следующая: 1GB RAM, 2GB SWAP, 1 CPU, Ubuntu 14.04 x64.

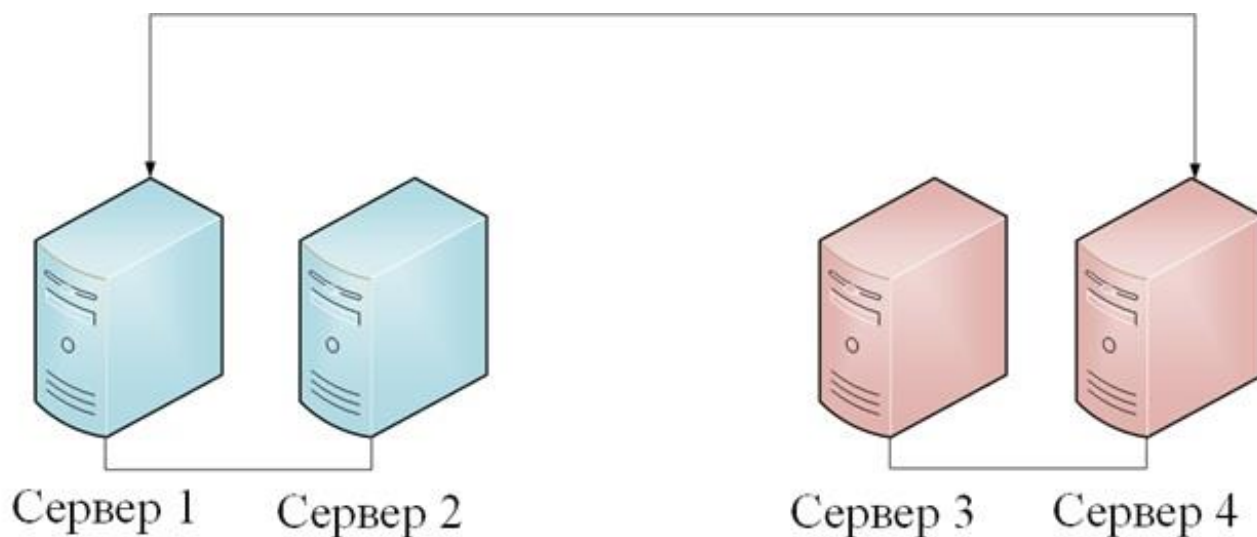


Рисунок 2. Топология кластера с использованием СУБД PostgreSQL

Сервер 1 и сервер 2 хранят первую половину одних и тех же данных, образуя первый шард.

Сервер 3 и сервер 4 хранят вторую половину одних и тех же данных, образуя второй шард.

### Проведение эксперимента

В качестве данных для тестирования взята база данных звонков одного из колл-центров. В среднем каждый день в этом колл-центре регистрируется где-то 10000 записей о звонках. Специально для тестирования выбран годовой диапазон или примерно 3,6 миллиона записей. В таблице 1 приведено описание полей, содержащихся в базах данных.

Таблица 1. Сравнительное описание типов данных в СУБД PostgreSQL и ClickHouse

Имя столбца	PostgreSQL тип данных	ClickHouse тип данных
session_id	character varying(64)	String
project_id	character varying(64)	String
created_date	character varying(20)	String
created_hour	integer	Int64
created_min_slot	integer	Int64
created_ts	timestamp without time zone	DateTime
queued_ts	timestamp without time zone	DateTime
unblocked_ts	timestamp without time zone	DateTime
connected_ts	timestamp without time zone	DateTime
ended_ts	timestamp without time zone	DateTime
is_unblocked	boolean	UInt8
incoming_asa	integer	UInt8
incoming_rt	integer	UInt8
incoming_tt	integer	UInt8
hold_duration	integer	UInt8
is_processed	boolean	UInt8
is_processed_after_threshold	boolean	UInt8
is_abandoned	boolean	UInt8
is_abandoned_after_threshold	boolean	UInt8
is_shortly_abandoned	boolean	UInt8
is_abandoned_on_hold	boolean	UInt8
is_abandoned_on_ivr	boolean	UInt8
agent_connected	character varying(256)	String

has_agent_leg	boolean	UInt8
created_ts	-	Date

Столбец **created\_ts** присутствует только лишь в базе данных среды ClickHouse. Данное поле нужно было для создания реплицируемых таблиц на серверах 2,3 и серверах 4,5 с движками типа MergeTree.

В качестве задачи было составлено три аналитических запроса на выборку данных.

Первый запрос это запрос, сгруппированный по дню.

На входе исходная требуемая дата, в данном примере это '2017-12-21' На выходе: проект (project\_id), кол-во звонков (quantity\_of\_calls), обслужен или нет (is\_processed), дата (created\_ts\_date)

```
SELECT project_id, COUNT() AS quantity_of_calls,is_processed,
created_ts_date
FROM inbound cell data
WHERE created_ts_date = '2017-12-21'
GROUP BY created_ts_date,project_id, is_processed
ORDER BY created_ts_date, project_id DESC;
```

Рисунок 1. Синтаксис запроса 1

Второй запрос это запрос, выходные данные которого будут сгруппированы по часам.

На входе также исходная требуемая дата, в данном примере это '2017-12-21' На выходе: проект (project\_id), кол-во звонков (quantity\_of\_calls), обслужен или нет (is\_processed), дата (created\_ts\_date)

```
SELECT project_id, COUNT() AS quantity_of_calls,is_processed,
toDateTime(toUInt32(toStartOfHour(created_ts)) + 3600) AS
created_ts_hours
FROM inbound cell data
WHERE created_ts_time = "2017-12-21"
```

```
GROUP BY created_ts_hours,project_id, is_processed  
ORDER BY created_ts_hours, project_id DESC;
```

Рисунок 2. Синтаксис запроса 2

Третий запрос это запрос, выходные данные которого будут сгруппированы по 15-ти минутным интервалам.

На входе также исходная требуемая дата, в данном примере это ‘2017-12-21’

На выходе: проект (project\_id), кол-во звонков (quantity\_of\_calls), обслужен или нет (is\_processed), дата (created\_ts\_date)

```
SELECT project_id, COUNT() AS quantity_of_calls,is_processed,  
toDateTime(toUInt32(created_ts) - toUInt32(created_ts) % 900 + 900)  
AS created_ts_15_min  
FROM inbound cell data  
WHERE created_ts_time = “2017-12-21”  
GROUP BY created_ts_15_min,project_id, is_processed  
ORDER BY created_ts_15_min, project_id DESC;
```

Рисунок 3. Синтаксис запроса 3

В таблице 2 приведены результаты выполнения указанных выше запросов

Таблица 2. Время выполнения OLAP запросов в СУБД PostgreSQL и ClickHouse

	PostgreSQL сек	ClickHouse сек
Запрос с группировкой по дню	1.6	0.108
Запрос с группировкой по часу	1.66	0.11
Запрос с группировкой по 15 минутному интервалу	1.9	0.126



## **Выводы**

Исходя из результатов проведенного эксперимента видно, что скорость обработки запросов в СУБД ClickHouse и СУБД PostgreSQL может значительно различаться в зависимости от набора исходных данных и типа выполняемых операций. Поэтому ключевую роль в выборе той или иной СУБД, в данном случае PostgreSQL или ClickHouse, играет набор исходных данных, а также требования, предъявляемые к скорости получения результатов аналитических запросов.

### **Использованные источники**

1. Руководство Yandex ClickHouse. Официальный сайт. [Электронный ресурс]. Режим доступа: [https://clickhouse.yandex/reference\\_ru.html](https://clickhouse.yandex/reference_ru.html) (дата обращения: 12.01.2018).
2. Руководство Apache ZooKeeper. Официальный сайт. [Электронный ресурс]. Режим доступа: <https://zookeeper.apache.org/doc/r3.5.2-alpha/> (дата обращения: 12.01.2018).
3. Flavio Junqueira, Benjamin Reed. ZooKeeper Distributed Process Coordination. USA, O'Reilly Media, 2014, 246 p