

Интеграция с множественным доступом с ClickHouse для обработки больших данных

Н.А. Лукашев¹, В.В. Давыдов²

¹Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, проспект Большевиков 22, Санкт-Петербург, Россия, 193232

²Санкт-Петербургский политехнический университет Петра Великого, Политехническая 29, Санкт-Петербург, Россия, 194064

Аннотация

При росте объема данных статистики на кластере со временем выгрузка результата перестает укладываться в таймауты на бэкенде. Скрипт для подсчёта статистики на MySQL не отвечает требованиям расширяемости, поэтому было принято решение реинтеграции статистики с колоночной СУБД в режиме множественного доступа. Настроенный кластер показал значительное улучшение производительности при росте метрик и росте периода выборки в сравнении со строчными СУБД.

Ключевые слова

Большие данные, СУБД, высокая нагрузка, рещардинг

1. Введение

Хранение больших объемов статистики существенно во многих отраслях, в том числе непосредственно с ИТ не связанных, порой приходится хранить действия каждого конкретного пользователя на периодах в несколько месяцев и более. При организации доступа в подобную базу для множества пользователей с выгрузкой статистики часто возникают проблемы с дублированием и актуальностью данных [1], так как объем данных статистики постоянно растет, модифицируется и запрашивается на чтение множеством пользователей, каждый из которых должен получить актуальные данные в адекватный временной период [2]. Процесс миграции Больших Данных на иную архитектуру всегда представляет повышенный интерес, новая же колоночная СУБД ClickHouse на данный момент испытывает особенный подъем, но явно ощущается недостаток каких-либо расширенных данных по ней кроме официальной документации, из чего следует что тематика данной работы особенно актуальна.

2. Проблема, первоначальный подход и решение

Статистика для клиента представляет собой формирование и вывод табличных отчетов с заранее определёнными метриками (количество обращений, средняя длительность обращений, средняя длина сообщения и т.д.) за заданный период с дополнительными фильтрами. С определённого момента формирование таких отчетов перестало укладываться в 30 секунд даже для небольших периодов.

Первоначально для решения возникшей проблемы был написан скрипт для вызова раз в сутки по CRON на подсчёт статистики. В рамках каждого вызова для прошедших суток считаются все метрики для всех возможных сочетаний фильтров. Полученные значения с значениями соответствующих фильтров пишутся в единую глобальную таблицу. Первоначально это позволило остаться в рамках единой СУБД MySQL и формировать отчеты в пределах 30 секунд. Такую скорость можно было считать условно удовлетворительной. Однако данный подход сам по себе создал ряд ограничений.

Глобальная таблица имеет тенденцию разрастаться до многих гигабайт данных, большая часть которых никогда не используется. Время же подсчёта метрик имеет тенденцию

расти к бесконечности с ростом фильтров, так как необходимо считать M-метрик * D-отделов * L-локалей * C-категорий. В итоге, мы уже оказываемся в ситуации, когда предподсчёт не укладывается даже за сутки. Это уже вынуждает включать фильтры по категориям только по отдельному запросу. Текущая схема не позволяет оперативно добавлять метрики, так как реализовать их досчёт за предыдущие периоды затратно по времени и ресурсам.

Исправление любых ошибок в расчётах метрик требует дополнительно сбрасывать уже посчитанные данные и начинать длительную процедуру пересчёта. Это дорогая операция как с точки зрения нагрузки на системы, так и затрат по времени. Для решения возникающих проблем с реализацией статистики на базе предподсчёта мы возвращаемся к началу - расчёт метрик и формирование отчётов по запросу. Чтобы не возникало проблем с производительностью, мы будем писать необходимые для метрик данные в одну из специальных аналитических колоночных СУБД, выбор был в пользу Яндекс.Clickhouse в первую очередь ввиду доступности, масштабируемости и отличных показателей по производительности пока не используются точечные запросы.

3. Интеграция и множественный доступ

Острее всего стояла проблема работы с асинхронными операциями и с доступом в базе для большого числа клиентов одновременно. Важной частью интеграции было внедрение рейт лимитера и кеша промежуточных результатов (так как полагаться на кеш операционной системы нельзя ввиду постоянной десериализации и сжатия данных).

Важно учесть, что если во время вставки разорвалось соединение, то мы не знаем, вставлены данные или нет, единственный вариант — это повторить вставку снова [3]. Если данные на самом деле были вставлены, и после вставка была проведена повторно, имеется дедубликация блоков. Она нужна во избежание дубликатов.

Для работы с материализованными представлениями были использованы 2 подтаблицы-представления с сырыми данными, чтобы хранить одну актуальную, в первом представлении происходят периодические дублирования, вторая – slave, по таймеру происходит пересчет данных, добор без дубликатов и запись материализованных представлений. Реализация этого средствами одного ClickHouse была нереалистична, API бэкенда должно следить за датой последнего обновления таблицы, совершить запросы к подтаблицам и агрегировать полученные данные по вышеуказанному паттерну.

Для контроля множественного доступа, как выяснилось, необходимо верно настроить OS thread priority на кластерах [4], execution speed для контроля выполнения операций и квоту на данные в единицу времени.

4. Заключение

После настройки кластера финальное сравнение производительности с MySQL показало что при росте количества метрик и периода выборки рост времени исполнения для ClickHouse значительно более пологий и с учетом планируемого рещардинга лучше всего подходит для дальнейшей поддержки кластера.

5. Литература

- [1] Network for Multi-User Access [Электронный ресурс]. – Режим доступа: <http://www.accessrepairrecovery.com/blog/ways-to-share-access-database-on-network> (30.12.2020).
- [2] Дейт К.Дж. Введение в системы баз данных. – Вильямс, 2018. – 1328 с.
- [3] Официальная документация ClickHouse [Электронный ресурс]. – Режим доступа: <https://clickhouse.tech/docs/en/> (03.01.2021).
- [4] Materialized View Concepts and Architecture [Электронный ресурс]. – Режим доступа: https://docs.oracle.com/cd/B10501_01/server.920/a96567/repview.htm (25.12.2021).