

ClickHouse в SEMrush: 5 историй с продакшена

Алексей Бойко,
Технический руководитель направления



Немного о SEMrush

SEMrush-all-in-one платформа для профессионалов в сфере digital-маркетинга



Работаем с
2008 года



2,000,000+
пользователей



450+
сотрудников

**4 офиса на двух
континентах:**

Филадельфия (США)

Прага (Чехия)

Санкт-Петербург (Россия)

Лимассол (Кипр)

Best SEO Software Suite 2017:

**US 2017
SEARCH AWARDS**

**UK 2017
SEARCH AWARDS**

**EUROPEAN 2017
SEARCH AWARDS**

Опыт применения

- 8 продуктов **успешно** построены с использованием **ClickHouse**
- **2** готовятся
- Инфраструктура: от **1** до **80** серверов для **хранения** данных
- Количество записей в самой **большой** таблице - **1.8T** записей и **растет**
- Размер самой **большой** таблицы в **сжатом** виде - **500Tb (1.2Pb)**
- Полтора года в **продакшене** (первый запуск в апреле 2017)



История 1: Преагрегация данных

История 1

Проблема: оперативная **память** на запрос **> 25 GB**

- Размер кластера: **30** серверов
- Количество сырых данных: **30B** записей (30 столбцов)
- Размер выборки после GROUP BY: **400M** (10 столбцов)
- GROUP BY по UInt64, плюс **any** для **длинных** строк (url)
- *distributed_group_by_no_merge* нельзя из-за **ключа шардирования**
- Сортировки по **любым** столбцам
- Множественные фильтры по **любым** столбцам
- Нагрузка: **100-1000** rps на отчет
- Нагрузка на систему: **100-1000** rps

Решение 1

- Создать **новую** таблицу
- Упростить до **простого** SELECT (без GROUP BY) путем **преагрегации** данных по частям
- Перешардировать для **равномерного** распределения данных

Результат: потребление оперативки **200MB**



История 2: Морфологический поиск

История 2

Проблема: средняя скорость **выполнения** запроса на другой СУБД **~10сек**

- Размер кластера: **3** сервера
- Поиск происходит по строке размером до **12** слов. Каждое слово до **30** символов.
- Поиск: точное соответствие, совпадение части слова, **морфологический** поиск
- Количество **сырых** данных: **7B** записей (20 столбцов)
- Максимальный **размер** выборки - **15M** записей (15 столбцов)
- Сортировки по **любым** столбцам
- Множественные фильтры по **любым** столбцам
- Нагрузка: **50-500** rps

Решение 2

- Используем **внешний** сервис
- Разбиваем **строку** на **слова**
- Каждое **слово** преобразуем с помощью **nlp алгоритмов** до **морфемы**
- Берем **хэш** от морфемы
- В итоге получаем **массив хэшей** от морфем
- Дублируем строку **столько** раз, сколько получилась **длина массива**, каждый раз записывая в **отдельную колонку** значение **итерации** по массиву

ClickHouse **не тормозит**: скорость выполнения самого медленного запроса **~750мс**

Пример

- **Строка:** *I want to start using clickhouse*
- **Морфемы:** *[want, start, us, clickhouse]*
- **Запись в базу:** 4 строки

want	I want to start using clickhouse	[want,start,us,clickhouse]
start	I want to start using clickhouse	[want,start,us,clickhouse]
us	I want to start using clickhouse	[want,start,us,clickhouse]
clickhouse	I want to start using clickhouse	[want,start,us,clickhouse]



История 3: Борьба с JOIN

История 3

Проблемы:

оперативная **память** на запрос **> 20GB**

время **выполнения** запроса **~15 сек**

- Размер кластера: **30** серверов
- Количество сырых данных: **30B** записей (**30** столбцов)
- Размер **выборки** 1: **500M** (**15** столбцов)
- Размер **выборки** 2: **150M** (**12** столбцов)
- Пересечение: **GLOBAL LEFT JOIN**
- Сортировки по **любым финальным** столбцам
- Множественные фильтры по **любым финальным** столбцам
- Нагрузка: **100-1000** rps

Решение 3

- Немного **денормализуем** данные
- Сначала **копим** инкремент из разных источников
- INSERT SELECT JOIN в **отдельную** таблицу
- Если нужно, добавляем **колонку** с типом источника данных

ClickHouse **перестает** тормозить: **~1сек и 500MB**



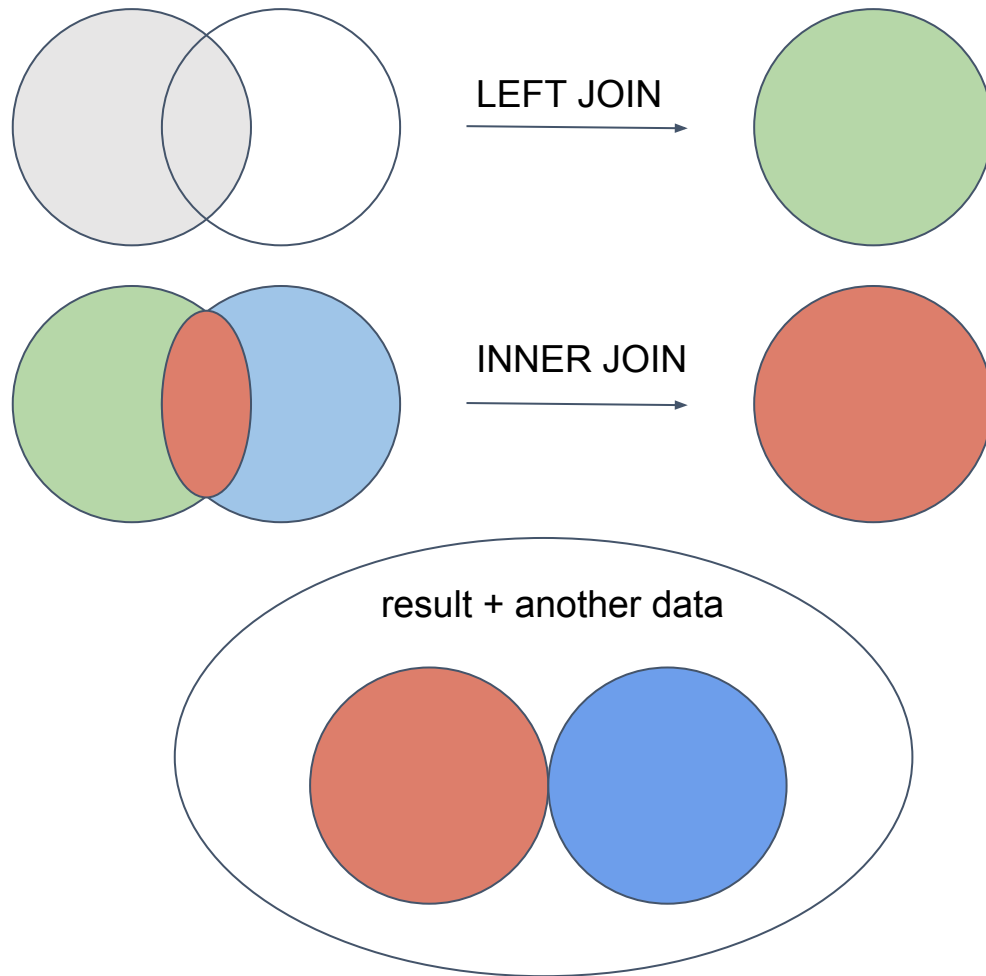
История 4: Как нас Merge движок спас

История 4

Проблема: технические сложности реализации подобных требований

- Есть **три** таблицы, данные по которым нужно пересекать **особым** образом
- Во все три таблицы данные пишут **разные** системы
- Размер кластера: **30** серверов
- Количество сырых данных: **30В** записей (30 столбцов)
- Размер выборки может достигать до **1М** записей (**12** столбцов)
- Сортировки по **любым финальным** столбцам
- Множественные фильтры по **любым финальным** столбцам
- Нагрузка: **10-500** rps

История 4



Решение 4

- Просим **совет** у **Алексея** Миловидова :)
- Создаем **Merge** таблицу, которая смотрит на **локальную** таблицу
- Поверх создаем **Distributed** таблицу
- Пишем **хитрый** запрос

ClickHouse **решает**: продолжаем работать как с одной таблицей!



История 5: Как нас секция SETTINGS спасала

История 5

- Не все знают, что в конце SELECT запросов можно писать **SETTINGS** (а часто даже нужно)
- **SETTINGS** можно также использовать и в **подзапросах**
- Настройки применяются только для **конкретного** запроса

Пример

```
SELECT
    ?ArgTransitsCount as transits,
    ?ArgIntersects as intersects
FROM (
    SELECT
        ?ArgMinPositionsFor [REDACTED]
    FROM g [REDACTED]
    WHERE 1=1
        ?ByDatabase
        ?ByMonth
        ?ByD [REDACTED]
        ?ByExcludeLost
        ?ByPhrase
        ?ByV [REDACTED]
        ?ByC [REDACTED]
        ?ByResults
        ?ByCo [REDACTED]
        ?ByTables
    GROUP BY phrase
    HAVING 1=1
        ?ByPositions1
        ?ByPositions2
        ?ByPositions3
        ?ByPositions4
        ?ByPositions5
    SETTINGS distributed_group_by_no_merge = 1, max_threads = 4
)
```

Решение 5

Какие настройки использовали **чаще** всего:

- **max_threads**
 - снижение количества тредов при небольших выборках может ускорить запрос
- **distributed_product_mode**
 - используя разные значения в подзапросах и основном запросе может получиться очень интересный результат
- **distributed_group_by_no_merge**
 - при верном выборе шардинга дает огромный прирост в скорости
- **distributed_aggregation_memory_efficient**
 - может спасти, когда данных ооочень много. иногда дает наоборот негативный эффект



Проблема чаще всего не в системе. Но это не точно :)

Экспериментируйте, учитесь, спрашивайте

Контакты

- Адрес: <https://semrush.com>
- Github: <https://github.com/semrush/>
- Telegram: @playboyko
- Email: a.boyko@semrush.com