
Яндекс

ClickHouse

ClickHouse для аналитиков

Мария Мансурова, аналитик Яндекс.Метрики

План на сегодня

- эволюция инструментов аналитики в Яндексе
- аналитика про аналитиков
- топ-10 лайфхаков при использовании ClickHouse

Кто такие аналитики?

В каких случаях стоит использовать ClickHouse?

Сфера применения ClickHouse

Аналитика событий

- несколько слабо-связанных таблиц с большим количеством столбцов
- хранятся небольшие значения: числа, строки
- простой сценарий обновления данных
- результат выполнения запроса существенно меньше исходных данных

Данные Яндекс.Метрики

большинство аналитиков используют CH для работы с Метрикой

- 2 основные таблицы с данными о сессиях и просмотрах на сайтах
- более 200-500 колонок
- суммарный объем 4 петабайта данных

Альтернативы ClickHouse для данных Метрики

- **Старый подход OLAP cubes**
  - недостаточно гибко
- **MapReduce (YT, YQL)**
  - можно посчитать все, что угодно
  - не всегда быстро

ClickHouse

Особенности

- быстро
- SQL на «стероидах»

Задачи

- RT monitoring
- регулярные отчеты
- ad-hoc задачи

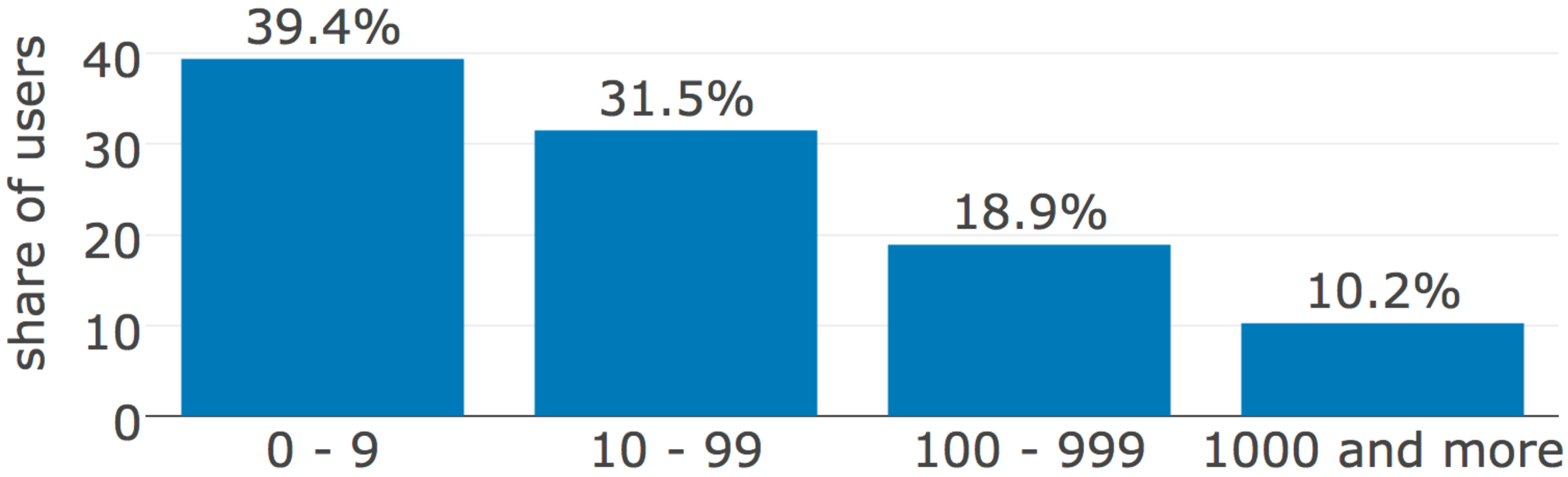
Аналитика
про аналитиков

Общая картина

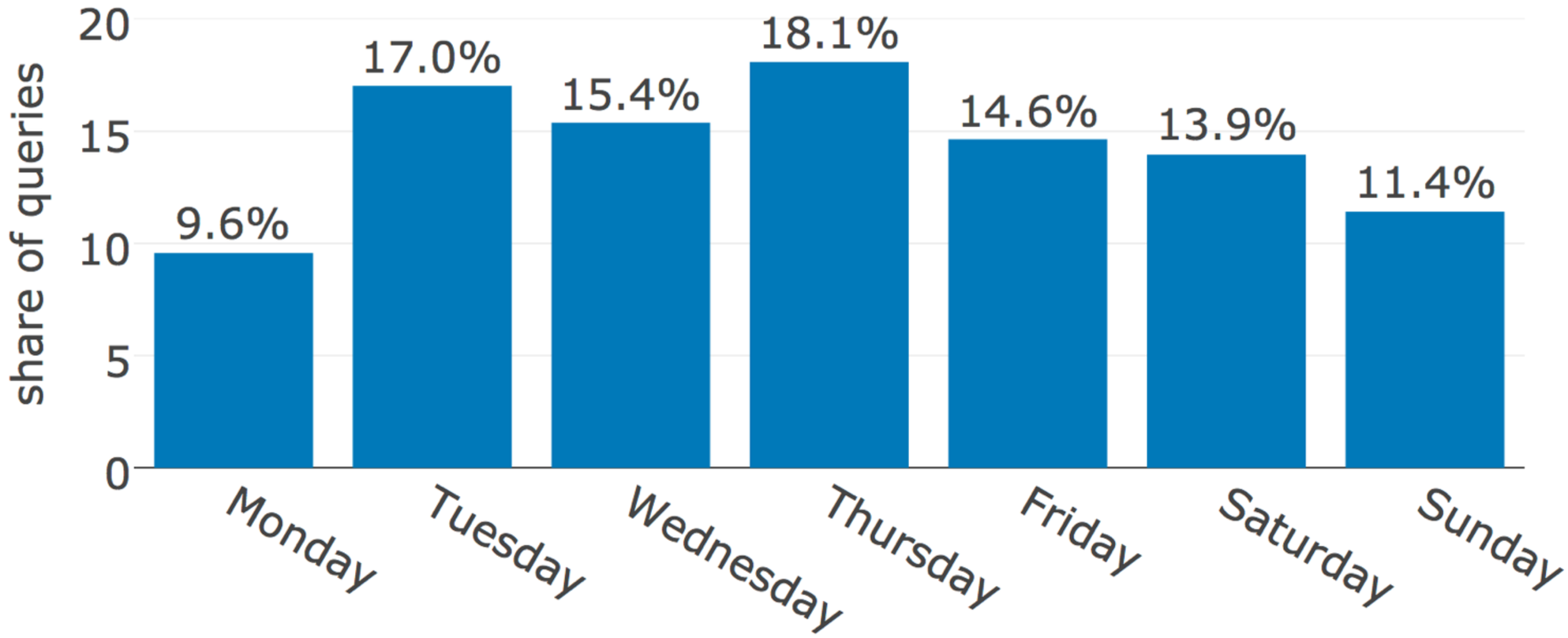
- 4 недели
- 130 уникальных логинов
- 1.5 миллиона запросов
- суммарное время выполнения запросов 145 дней
- 8 терабайт данных было получено и 1 терабайт записан, база данных при этом обработала 70 петабайт

Сколько запросов в день вы задаете?

Сколько запросов задают в день аналитики?



Запросы по дням недели



Использование семплирования

№	Семплирование	Доля в трафике
1	100 %	59.4 %
2	10 %	11.4 %
3	1 %	8.4 %
4	0.0001 %	7.2 %
5	0.1 %	3.0 %

Какая самая распространенная функция в ClickHouse?


sum
toDate
transform
toStartOfFiveMinute
toStartOfMinute
arrayElement
regionToName
uniqExact
OSToRoot
toUInt64
toInt64
uniq
count
indexOf
cityHash64
toDateTime
regionToCity
splitByChar
IPv6NumToString
concat
toUInt32
intDiv
anyLast
yesterday
toFixedString
toString
substring
extractURLParameter
arrayJoin
arrayCount
max
min
now
match
today
arrayFirst
arrayExists
arrayFilter
regionHierarchy
regionToArea
regionToPopulation
dictGetString
toUnixTimestamp
lengthUTF8
groupArray
toMonday
replaceAll
arrayMap
toStartOfMonth
toStartOfHour
position
round
length
lower
modulo
replaceRegexpAll
replaceOne
quantiles
quantile
extract
toMinute
toHour
reverse
avg
like
toInt32
halfMD5
bitmaskToArray
timeSlot
timeSlots
toFloat32
toFloat64
toUInt8
toUInt16
toInt16
reinterpretAsUInt64
reinterpretAsString
toUInt32OrZero
toFloat32OrZero
positionCaseInsensitive
groupUniqArray
arrayStringConcat
regionIn
OSIn
SEIn
roundAge
path
pathFull
log2
argMax
CAST
dictGetUInt32OrDefault
visitParamExtractFloat
cutToFirstSignificantSubdomain

Распространенные ошибки

Exception	Доля
Quota for user has been exceeded	50.7 %
Syntax error	38.5 %
Cannot parse string as date	3.3 %
Query is executing too slow	2.6 %
Too much simultaneous queries for user	1.5 %
Double-distributed IN/JOIN subqueries is	1.5 %
Unknown identifier	0.9 %
Memory limit (for query) exceeded	0.5 %
Table doesn't exist	0.1 %

Top-10 things I wish I knew when I started using ClickHouse

#1 Парсим ответ от базы данных

```
q = '''SELECT
    count() as visits,
    Date as date
FROM visits_table
GROUP BY date'''
```

```
items = []
for line in data.split('\n'):
    queries, date =
    line.split('\t')
    items.append({
        'queries': queries,
        'date': date
      })
```

#1 Парсим ответ от базы данных оптимальнее

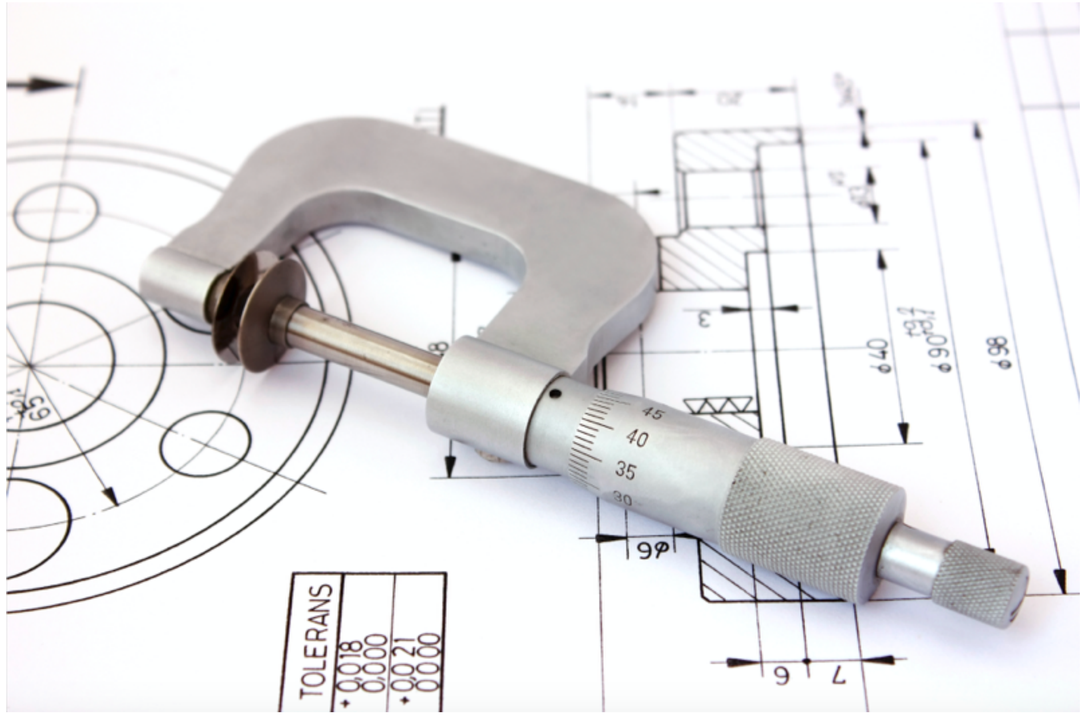
```
q = '''SELECT
    count() as visits,
    Date as date
FROM visits_table
GROUP BY date
FORMAT TabSeparatedWithNames
'''
```

```
df = pandas.read_table(
    StringIO.StringIO(data)
)
```

#2 Узнаем как все устроено

```
SHOW CREATE TABLE visits_table

CREATE TABLE visits_table
    (UserID UInt64, …)  ENGINE =
    MergeTree(EventDate,
    intHash32(UserID),
    (EventDate, StartURLDomain,
    intHash32(UserID)), 8192)
```



#3 «Слишком» большие данные

Ошибка: «Memory limit (for query) exceeded»

〉 Вариант 1: используем SAMPLE

```
SELECT
    count()*10 as visits,
    sum(PageViews)*10 as hits,
    uniq(UserID)*10 as users,
    URL as url
FROM visits_table SAMPLE 1/10
GROUP BY url
```

#3 Хочется еще и точно...

- Ошибка: «Memory limit (for query) exceeded»
  - Вариант 1: используем SAMPLE
  - Вариант 2: используем SAMPLE и OFFSET

```
SELECT
    count() as visits,
    sum(PageViews) as hits,
    uniq(UserID) as users,
    URL as url
FROM visits_table SAMPLE 1/10
   OFFSET {i}/10
GROUP BY url

i = 0, 1, 2, .. 9
```

#3 Таблица не семплирована

- Ошибка: «Memory limit (for query) exceeded»
  - Вариант 1: используем SAMPLE
  - Вариант 2: используем SAMPLE и OFFSET
  - Вариант 3: делаем SAMPLE и OFFSET своими руками

```
SELECT
    count() as visits,
    sum(PageViews) as hits,
    uniq(UserID) as users,
    URL as url
FROM visits_table
WHERE
    intHash32(UserID) % 10 = i
GROUP BY url

i = 0, 1, 2, .. 9
```

#4 Временная таблица

```
$ head -n 3 user_ids
123456789
34012347
129078999

$ clickhouse-client --query="SELECT count() FROM
visits_table WHERE UserID IN users" --external
--file=user_ids --name=users --types=UInt64
```

#5 Модификаторы функции -If, -Array

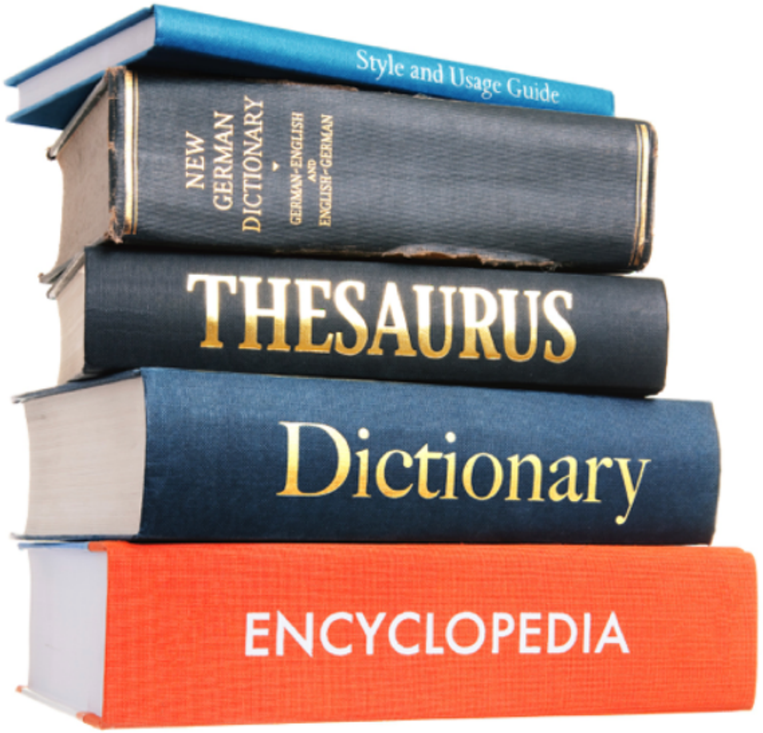
```
SELECT
    count() as visits,
    sum(PageViews) as hits,
    countIf(IsMobile) as mobile_visits,
    sumIf(PageViews, IsMobile) as mobile_hits,
    mobile_visit/visits as mobile_visits_share,
    mobile_hits/hits as mobile_hits_share
FROM visits_table
FORMAT TabSeparatedWithNames
```

#5 Модификаторы функции -If, -Array

```
SELECT
    sumArray(ProductsPrices) as revenue
FROM visits_table
FORMAT TabSeparatedWithNames
```

#6 Расшифровываем IDs

- Подключаем внешний словарь и расшифровываем поля на уровне БД `dictGetString('my_dict', 'description', id)`



#7 Разработка в консольном клиенте

`clickhouse-client --multiline`

```
miptgirl@miptgirl-ml:~$ clickhouse-client -m
ClickHouse client version 1.1.54112.
Connecting to localhost:9000.
Connected to ClickHouse server version 1.1.54112.

:) SELECT
:-]     count() as total_queries,
:-]     countIf(exception='') as success_queries,
:-]     uniqExact(user) as users,
:-]     sum(query_duration_ms)/(1000*60*60*24) as query_duration_days,
:-]     sum(query_duration_ms)/(1000) as query_duration_secs,
:-]     sum(result_bytes) as total_result_bytes,
:-]     sum(read_bytes) as total_read_bytes,
:-]     sum(written_bytes) as total_written_bytes,
:-]     total_result_bytes/pow(10, 12) as result_tbytes,
:-]     total_read_bytes/pow(10, 12) as read_tbytes,
:-]     total_written_bytes/pow(10, 12) as written_tbytes
:-] FROM query_log;
```

#7* Есть еще и GUI

https://github.com/smi2/clickhouse-frontend

8 Системные таблицы

- **system.query_log** - логи пользователей
- **system.functions** - полный список функций
- **system.settings** - текущие настройки

#9 argMin, argMax

```
-- обычный SQL
SELECT PurchaseID
FROM visits_table
WHERE DateTime = (SELECT
    min(DateTime) FROM
    visits_table)
```

```
-- ClickHouse
SELECT argMin(PurchaseID,
    DateTime)
FROM visits_table
```

#10 Сила массивов

Массивы - это очень полезно

- `arrayJoin` - развернуть массив в строки
- `groupArray` - аггрегатная функция, которая создает массив
- функции высшего порядка `arrayMap, arrayFilter` и т.д. + `lambda-функции`

#10 Выручка от проданных книг

```
SELECT
 sumArray(
        arrayFilter(
            price, name -> name LIKE %book%,
            ProductsPrices, ProductsNames)
)
FROM visits_table
```

#10 Выручка от проданных книг v.2

```
SELECT
    sum(price)
FROM visits_table
ARRAY JOIN
    ProductsNames as name,
    ProductsPrices as price
WHERE name LIKE %book%’
```

Пример аналитической задачи

- Код: bit.ly/ch_example
- Вебинар: bit.ly/ch_webinar

Задачи

- расчет продуктовых метрик retention и rolling retention
- построение кастомных моделей атрибуции

Самый главный совет

- Прочитать документацию … хотя бы оглавление :)

  https://clickhouse.yandex/

Контакты

- **Сайт:** https://clickhouse.yandex/
- **Google группы:** https://groups.google.com/forum/#!forum/clickhouse
- **MailList:** clickhouse-feedback@yandex-team.ru
- **Telegram:** https://telegram.me/clickhouse_en и https://telegram.me/clickhouse_ru (уже 505 подписчиков)
- **GitHub:** http://github.com/yandex/ClickHouse


Яндекс