

Яндекс

Как устроено API в AppMetrica

Ефим Пышнограев

Цели доклада



Вы узнаете:

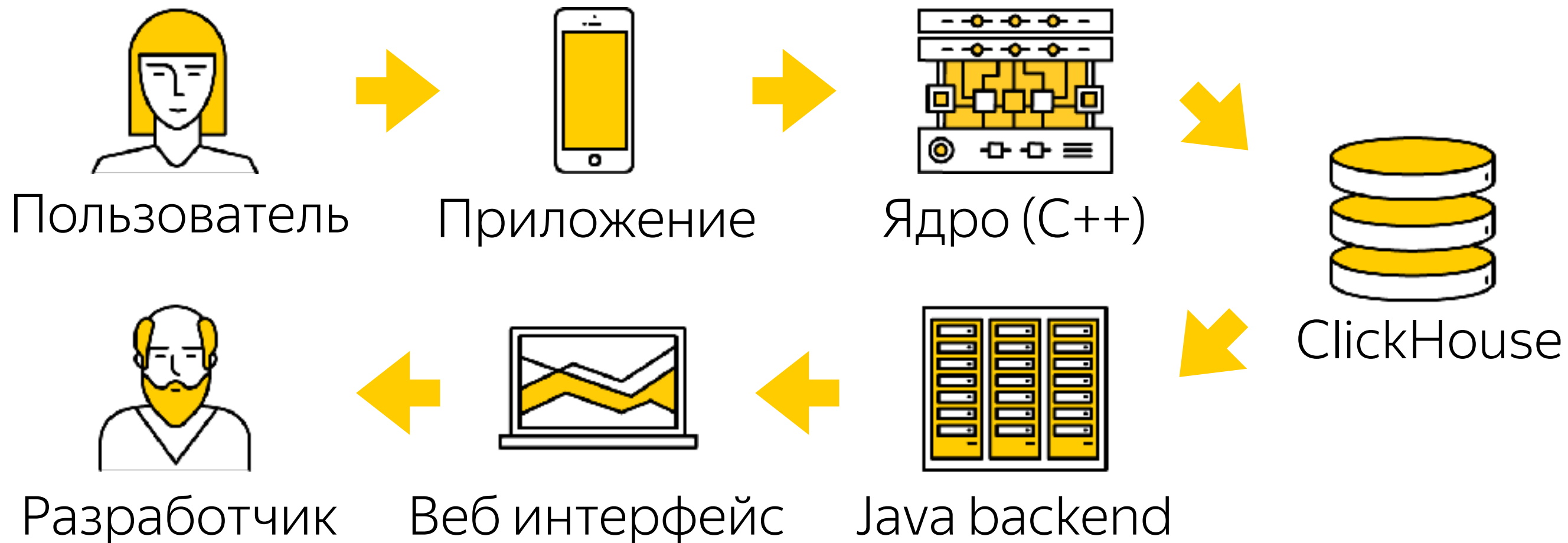
- › как устроена крупная система аналитики
- › какие проблемы возникают при построении отчетов
- › как выглядит Reporting API и зачем он нужен

О сервисе



- › Система мобильной аналитики
- › На рынке с 2013 года
- › Более 4 тыс. активных приложений
- › Мы установлены более чем на 250 млн. устройств
- › Более 12 млрд. событий в сутки

Архитектура сервиса



Хранение данных



Агрегированные данные

Что хорошо

- › Все данные есть, отчет строится мгновенно

Что плохо

- › Только predetermined отчеты

Сырые данные

- › Можно построить отчет произвольной сложности

- › Скорость агрегации становится критичной

Хранение данных

Таблица фактов в ClickHouse

- › Open source column-oriented DBMS
- › Запросы на диалекте SQL
- › 48 машин
- › 490 Tb сжатых данных

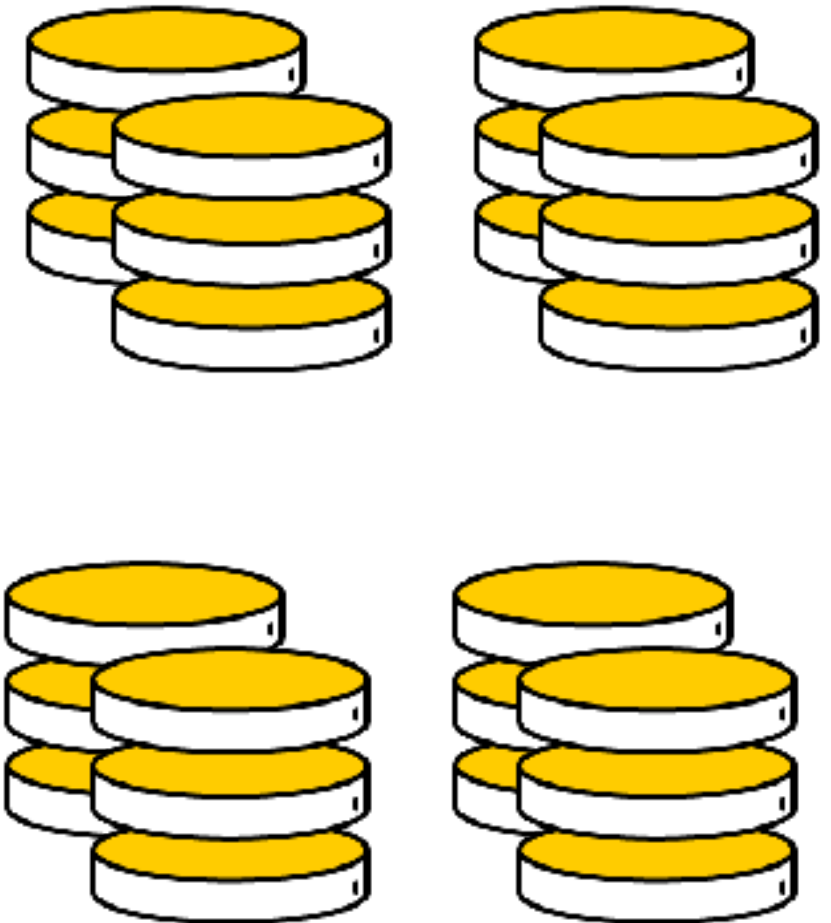


Вспомогательная информация в MySQL

Кластер ClickHouse



- › Каждый хост реплицирован в разных ДЦ
- › Весь кластер разделен на партиции
- › Данные одного приложения находятся на одной партиции



Распределенные таблицы



ClickHouse поддерживает таблицы с движком Distributed

Запрос в локальную таблицу:

```
SELECT count() FROM events
```

Запрос в распределенную таблицу:

```
SELECT count() FROM events_partition
```


Пример таблицы фактов



~140 столбцов

<u>Date</u>	<u>AppID</u>	DeviceID	AppPlatform	CountryID	...
2017-06-01	46	268927106	android	6	
2017-06-01	46	1292856823	android	43	
2017-06-01	87	5341985450	iOS	0	
2017-06-02	22	876028365	windows	43	

Эволюция хранения данных



Раньше:

- › Одна таблица на все события со столбцом EventType

Сейчас:

Таблица под каждый тип события

- › Общие события
- › Клиентские события
- › Атрибуция — клики и установки
- › Рассылки push уведомлений

Язык запросов



Используется интерфейсом и клиентами напрямую

Формат языка запросов должен позволять

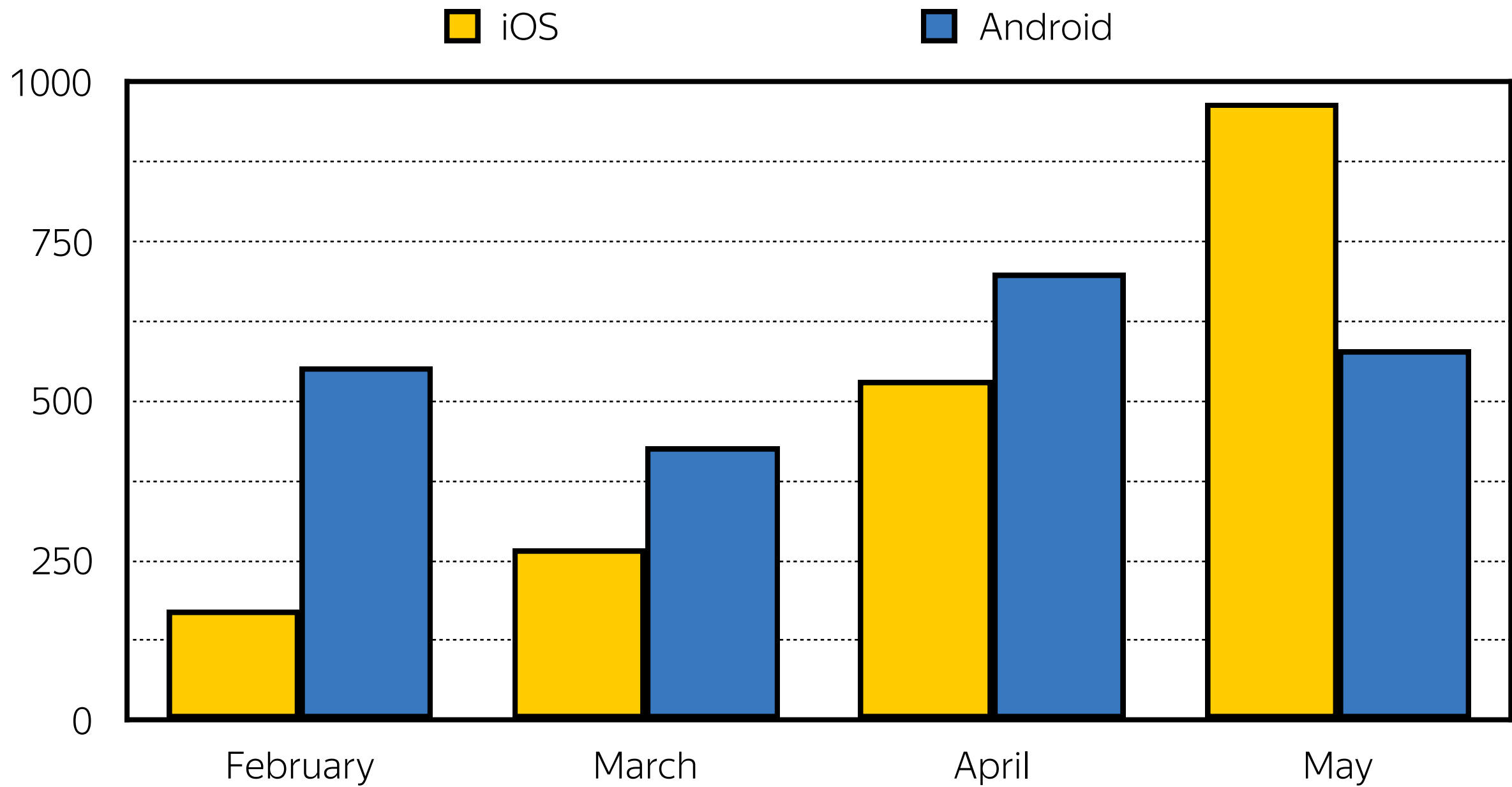
- › Покрывать большинство полезных отчетов
- › Скрывать сложность SQL запросов
- › Скрывать реальную модель данных
- › Эффективно использовать ClickHouse

Reporting API



- › Написано на Java
- › 6 машин в 3 ДЦ
- › Более 200 тыс. запросов в сутки
- › Каждый запрос обрабатывает в среднем 100 млн. строк

Пример отчета



Количество пользователей приложения с системами iOS и Android с февраля по май 2017 года

Количество пользователей



id	date1	date2	metrics
46	2017-06-01	2017-06-07	e:users

SELECT

uniq(DeviceID) **AS** users

FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

Несколько метрик



id	date1	date2	metrics
46	2017-06-01	2017-06-07	e:users,e:events

SELECT

uniq(DeviceID) **AS** users,

count() **AS** events

FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

Пользователи по странам



metrics

dimensions

e:users

e:country

SELECT

CountryID **AS** country,

uniq(DeviceID) **AS** users

FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

GROUP BY country

Фильтрация данных



metrics	dimensions	filters
e:users	e:country	os == 'android'

SELECT

CountryID **AS** country,
uniq(DeviceID) **AS** users

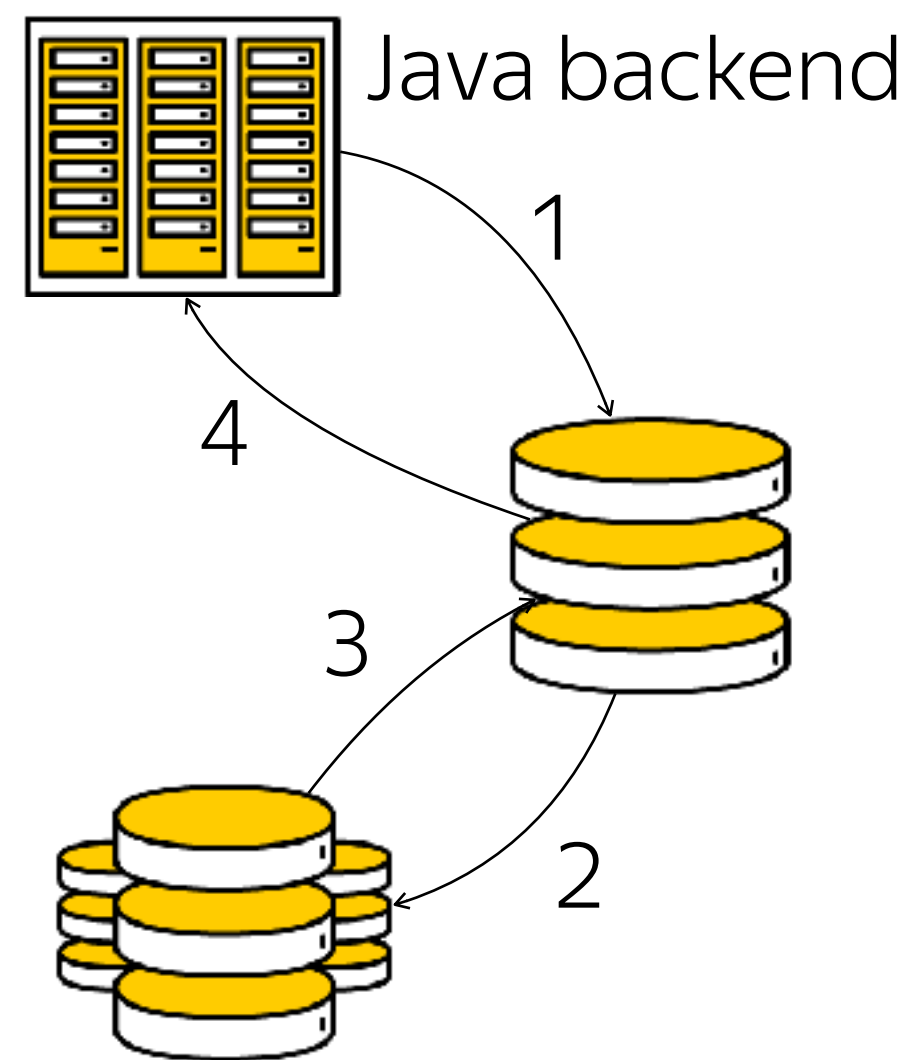
FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')
AND AppPlatform = 'android'

GROUP BY country

Запрос в распределенную таблицу

1. Запрос отправляется на произвольный хост партиции
2. Хост отправляет запрос на все локальные таблицы партиции
3. Все хосты возвращают частичные агрегаты из своих локальных данных
4. Первый хост соединяет эти агрегаты и возвращает результат



Запрос в распределенную таблицу

- › Партиция линейно масштабируется
- › Работают очень эффективно
- › Нельзя строить сложные сегменты

Несколько таблиц



metrics	dimensions	filters
e:users	e:country	os == 'android' and exists pc:device with (campaign == 'BlackFriday')

SELECT

CountryID **AS** country,
uniq(DeviceID) **AS** users

FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

AND AppPlatform = 'android' **AND** DeviceID **GLOBAL IN** (

SELECT DeviceID

FROM push_campaigns_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

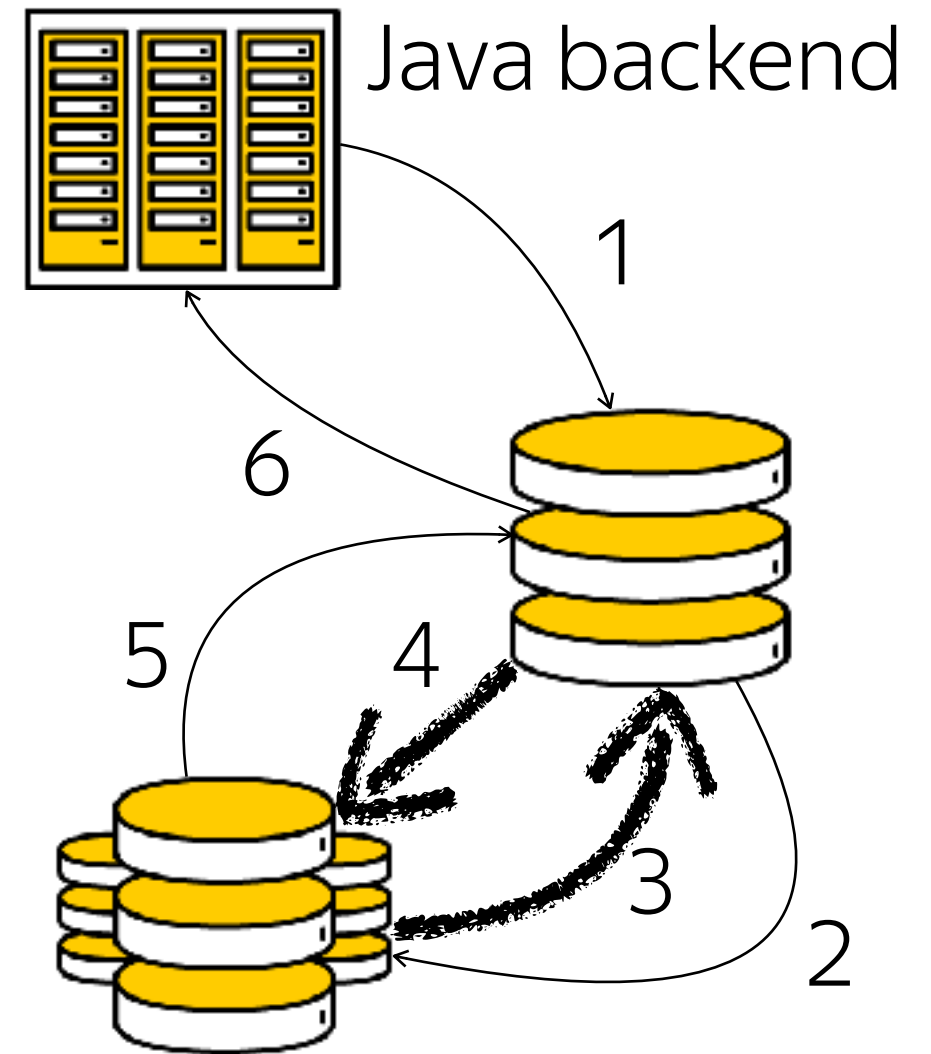
AND CampaignID = 888

)

GROUP BY country

Запрос с GLOBAL IN

1. Запрос отправляется на произвольный хост нужной партии
- 2-3. Хост вычисляет внутренний подзапрос в распределенную таблицу
4. Отправляет внешний запрос на все хосты вместе с результатом подзапроса
- 5-6. Получает частичные агрегаты и соединяет их. Возвращает результат



Запрос с GLOBAL IN



- › При расширении кластера загружается сеть
- › Работают менее эффективно
- › Можно строить сегменты любой сложности

Локальность данных



metrics	dimensions	filters
e:users	e:country	os == 'android' and exists pc:device with (campaign == 'BlackFriday')

SELECT

CountryID **AS** country,
uniq(DeviceID) **AS** users

FROM events_partition

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

AND AppPlatform = 'android' **AND** DeviceID **IN** (

SELECT DeviceID

FROM push_campaigns

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

AND CampaignID = 888

)

GROUP BY country

Локальность данных



- › Замена глобальных IN и JOIN на локальные ускоряет запросы и разгружает сеть
- › Имеет смысл если почти все наши IN и JOIN происходят по полю DeviceID
- › Но при этом нужно уметь делать перешардирование

Семплирование в отчетах



Что делать если крупное приложение хочет посмотреть сложный отчет за несколько лет?

- › Движок таблицы MergeTree поддерживает семплирование
- › Ключ семплирования должен быть равномерно распределен

Запросы без семплирования и с семплированием

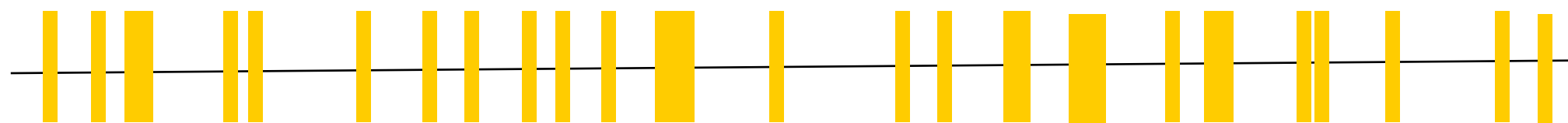
```
SELECT count() FROM events
```

```
SELECT count() * 10 FROM events SAMPLE 1/10
```

Семплирование в отчетах



`SELECT count() * 3 FROM events WHERE DeviceID % 3 = 0`



`SELECT count() * 3 FROM events SAMPLE 1/3`



Семплирование в отчетах



- › Ключ должен быть одинаковым во всех таблицах
- › Ключ должен быть столбцом, по которому делается IN

```
SELECT count() * 10
FROM events
SAMPLE 1/10
WHERE DeviceID IN (
    SELECT DeviceID
    FROM push_campaigns
    SAMPLE 1/10
```

Адаптивное семплирование



accuracy	metrics
medium	e:users

SELECT

$\text{uniq}(\text{DeviceID}) * (42/13)$ **AS** users

FROM events_partition

SAMPLE 13/42

WHERE AppID = 46 **AND** (Date **BETWEEN** '2017-06-01' **AND** '2017-06-07')

Адаптивное семплирование



- › Оценить сколько строк будет обработано запросом
- › Выставить коэффициент семплирования так, чтобы обработанных строк стало не больше N
- › Делать это быстро

Разбиение запроса на части



Если запрос не помещается в память

```
SELECT ... FROM events GROUP BY DeviceID
```



```
SELECT ... FROM events SAMPLE 1/3 OFFSET 0 GROUP BY DeviceID
```

```
SELECT ... FROM events SAMPLE 1/3 OFFSET 1/3 GROUP BY DeviceID
```

```
SELECT ... FROM events SAMPLE 1/3 OFFSET 2/3 GROUP BY DeviceID
```

Заключение



- › Мы храним данные в ClickHouse и строим отчеты на лету
- › Мы сделали свой DSL для описания отчетов
- › Не все отчеты можно выразить через Reporting API
- › Есть возможность выгрузки сырых логов

Спасибо за внимание!

Ефим Пышнограев

Разработчик

 graev@yandex-team.ru

 [badgersow](https://t.me/badgersow)