

# ClickHouse для инженеров и архитекторов БД



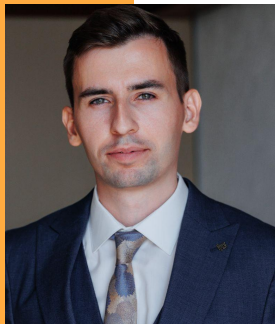
Проверить, идет ли запись

**Меня хорошо видно  
&& слышно?**



Тема вебинара

# Аналитические движки (СУБД) для работы с данными



**Алексей Железной**

**Senior Data Engineer/Architect**

Магистратура - ФКН ВШЭ

Руководитель курсов **DWH Analyst, ClickHouse** для инженеров и архитекторов  
**БД** в OTUS

Преподаватель курсов **Data Engineer, DWH Analyst, PostgreSQL** и пр. в OTUS

[LinkedIn](#)

# Правила вебинара



Активно  
участвуем



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Знакомство в чате группы

1. Где работаете? Чем занимаетесь?
2. Какие из тем знакомы? Уровень владения? На чем хотелось бы сделать акцент?
3. Ожидания от курса в целом?

# Маршрут вебинара



Аналитические СУБД как класс

Ключевые принципы устройства аналитических СУБД

Знакомство с ClickHouse

Подготовка инфраструктуры

Рефлексия

# Цели вебинара

1. Провести знакомство с курсом, платформой и планом работы
2. Разобрать основные принципы устройства аналитических движков и их использование на практике

# Аналитические СУБД как класс



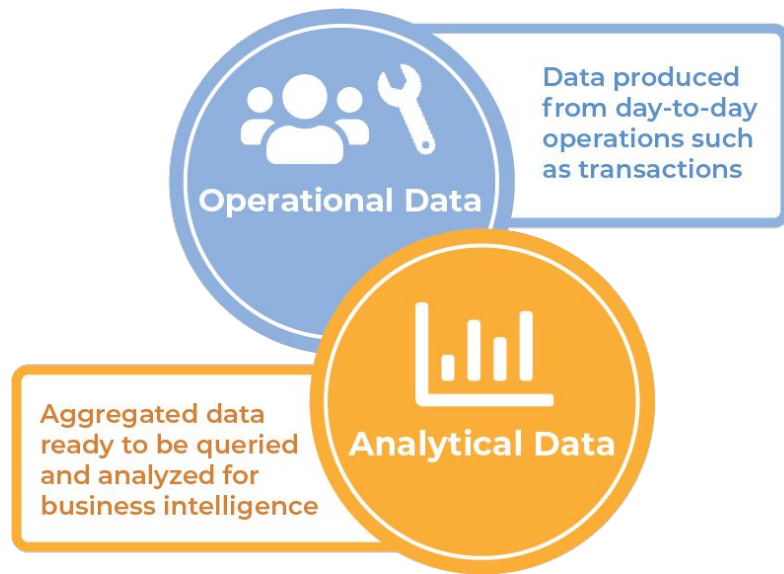
# Инфраструктура данных включает

## Operational Data Systems

- Операционные данные — это данные, которые производятся в ходе повседневной деятельности вашей организации
- Поддерживают доступ к OLTP

## Analytical Data Systems

- Аналитические данные используются для принятия бизнес-решений
- Аналитические данные лучше всего хранить в системе данных OLAP



# OLAP/OLTP

# OLTP

**OLTP = OnLine Transactional Processing** - обработка транзакций в реальном времени.

*Реляционные БД*

- Одновременное выполнение нескольких транзакций (экономических, финансовых, цифрового взаимодействия), таких как интернет-банкинг, покупки, ввод заказов или отправка текстовых сообщений
- Задача - ввод, редактирование, удаление данных в режиме онлайн и их хранение
- Больше про нормализацию

# Особенности OLTP

- Нормализованные данные;
- Высокая интенсивность добавления и изменения данных;
- Большое количество одновременно активных пользователей (\*)
- Внесение данных и расчеты осуществляют пользователи системы;
- Содержат актуальные данные (\*)
- OLTP использует транзакции, которые включают небольшие объемы данных.
- Индексированные данные в базе данных могут быть легко доступны
- Трехуровневая архитектура, которая обычно состоит из уровня представления, уровня бизнес-логики и уровня хранилища данных

# Примеры OLTP




# OLAP

**OLAP = OnLine Analytical Processing** = аналитическая обработка данных в реальном времени. *Многомерные БД*

O L A P

**Processing** = Обработываются некие исходные данные...

**Analytical** = ... чтобы получить какие-то аналитические отчеты или новые знания...



**OnLine** = ... в реальном времени, практически без задержек на обработку.

# Бизнес-смысл OLAP

**Залог успешного бизнеса с точки зрения Big Data:**

- +** Много данных (как фактовых, так и исторических)
  - +** Проработанные механизмы сбора и обработки данных
  - +** Крутые, мощные системы для их хранения, анализа
  - +** Наглядные BI-системы для отображения информации
- =** Принятие подготовленных, “правильных” бизнес-решений

*Сферы применения - в анализе тенденций, финансовой отчетности, прогнозировании продаж, бюджетировании и других целях планирования*

# Тех-смысл OLAP

- используется для построения отчетов на основе больших объемов накопленных исторических данных за огромные промежутки времени, но эти отчеты обновляются не слишком часто (\*)
- чаще всего это столбцовые СУБД (или поддерживающие column-orientation)
- выбирает данные быстро
- в центре находится таблица фактов, в которой находятся все показатели (сумма, кол-во) и ссылки на справочники (\*)
- чем больше столбцов, тем ниже скорость выполнения операций над строками (таких как добавление или изменение данных)
- больше про денормализацию



# Многомерные БД для OLAP



# Особенности OLTP

OnLine Transaction Processing (**OLTP**) - система обработки транзакций в реальном времени.

## Типичные операции:

- Создание (CREATE)
- Чтение (SELECT)
- Модификация (UPDATE)
- Удаление (DELETE)

**Пример:** Система бронирования авиабилетов

OnLine Analytical Processing (**OLAP**) - интерактивная аналитическая обработка данных. Данные суммируются, агрегируются.

## Типичные операции:

- Чтение
- Агрегация (GROUP BY, OVER PARTITION BY)

**Пример:** Система отслеживания объемов продаж авиабилетов

# Два типа использования данных

- Большой объем операций
- Высокая скорость обработки
- Много таблиц
- Данные нормализованы
- “Кто купил X?”



- Большой объем данных
- Низкая скорость обработки запросов
- Мало таблиц
- Данные денормализованы
- “Сколько человек купило X?”

# Вопросы?



Ставим "+",  
если вопросы есть



Ставим "-",  
если вопросов нет

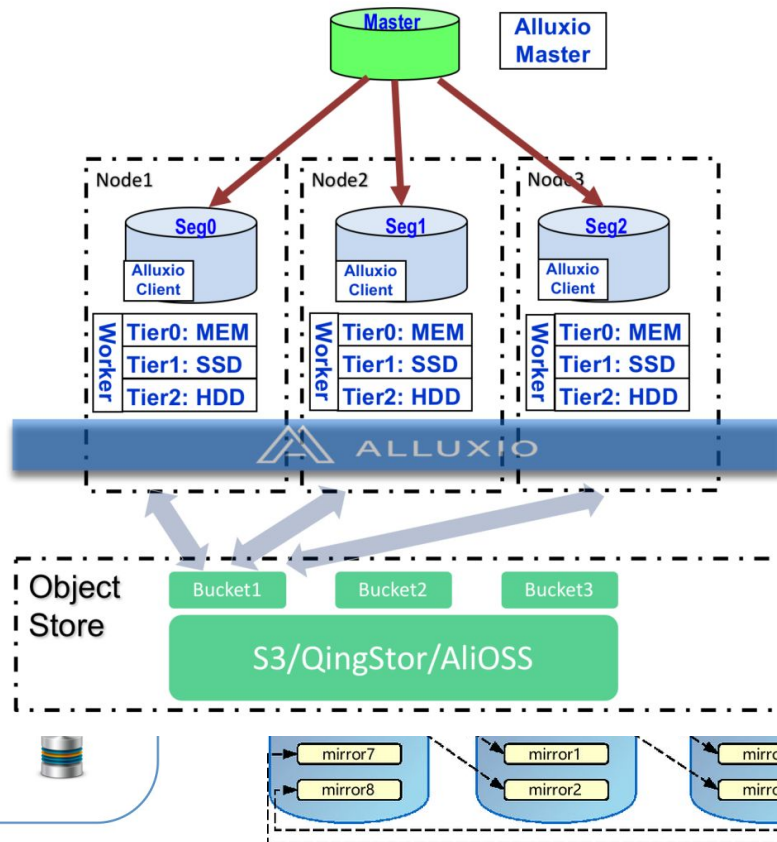
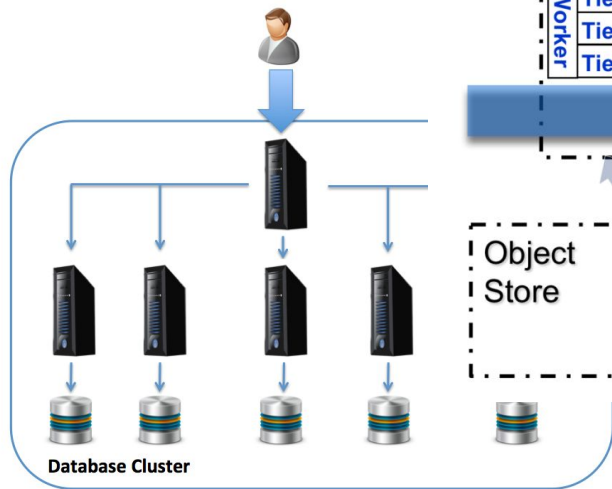


# Теория аналитических СУБД

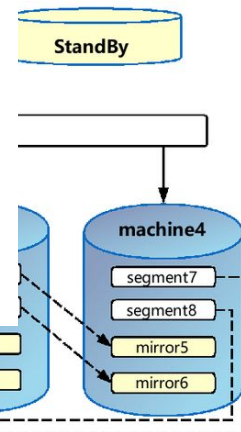
# Что такое MPP, и как это работает?

- Massive Parallel I/O
- Выделение необ...

## MPP System Architect

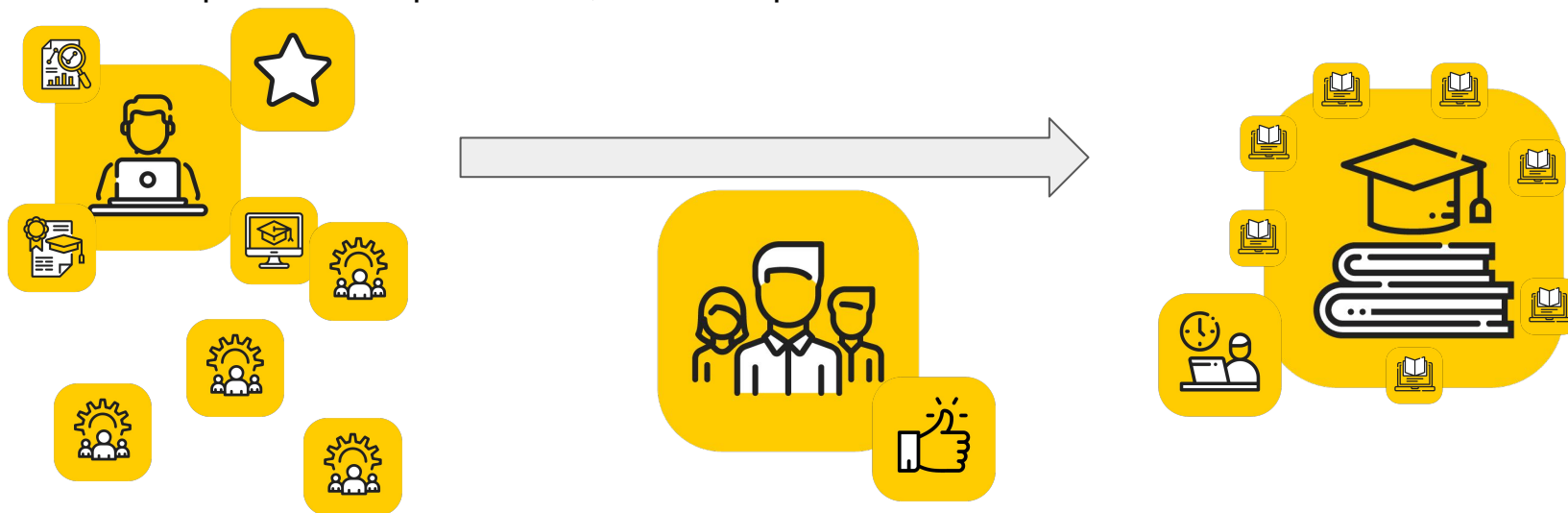


данных)  
им различным узлам



# Анализ больших данных: человеческий пример

Масштабировать по горизонтали, а не по вертикали



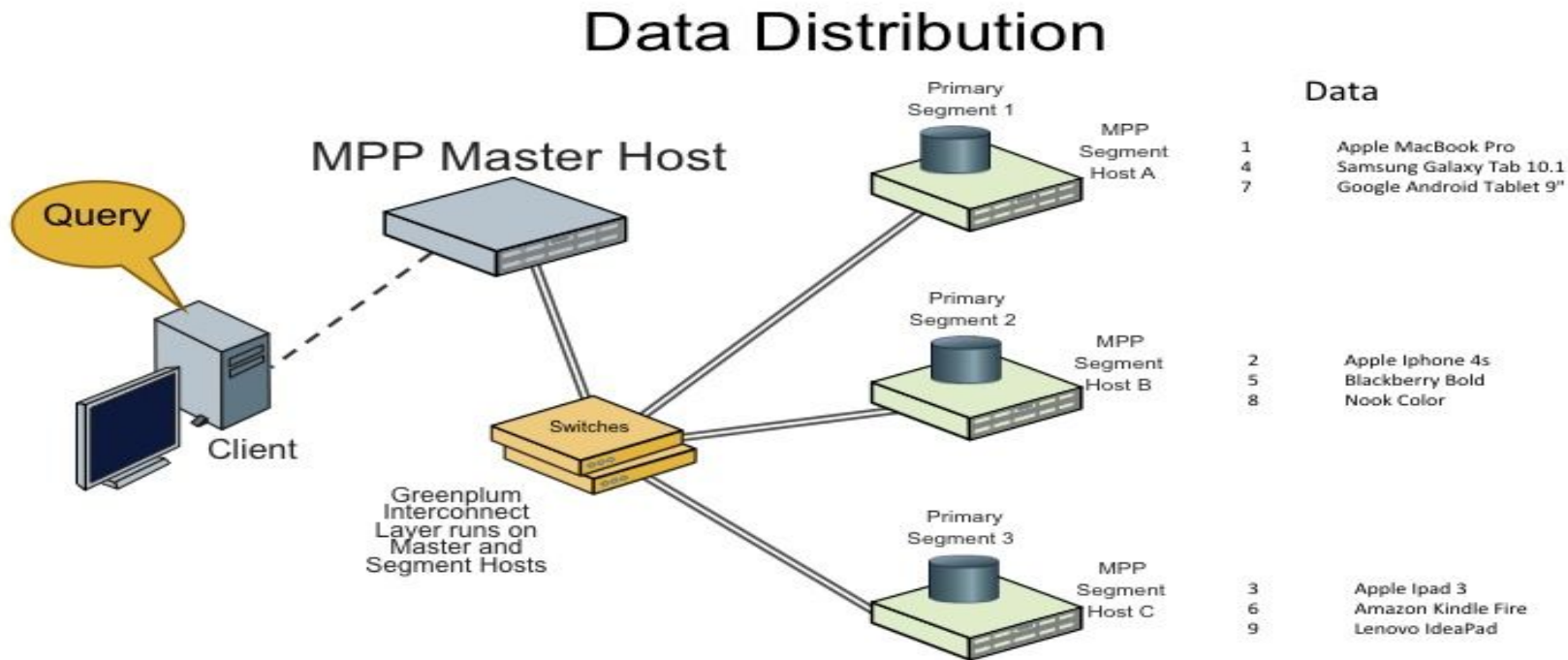
**Это массивно-параллельная обработка в действии, только с людьми, а не с компьютерами. Разделение простых, но больших задач на несколько сегментов и одновременная обработка этих сегментов будет намного быстрее, чем один человек, работающий в одиночку, независимо от того, насколько он квалифицирован.**

# Что такое MPP, и как это работает?

- **База данных MPP** — это тип базы данных или хранилища данных, в котором данные и вычислительная мощность распределяются между несколькими различными узлами (серверами) с одним ведущим узлом и одним или несколькими вычислительными узлами
- В MPP лидера (вас) называют ведущим узлом, работников библиотеки - вычислительными узлами
- Базы данных MPP можно масштабировать горизонтально, добавляя дополнительные вычислительные ресурсы (узлы)
- **Плюсы** — относительная быстрота обработки больших объемов данных (*Shared Nothing*), простота горизонтального масштабирования до сотен узлов, отказоустойчивость за счет зеркалирования и резервирования.
- **Минусы** — высокие требования к ресурсам, низкая производительность при большом объеме простых запросов, неоптимальное распределение сегментов

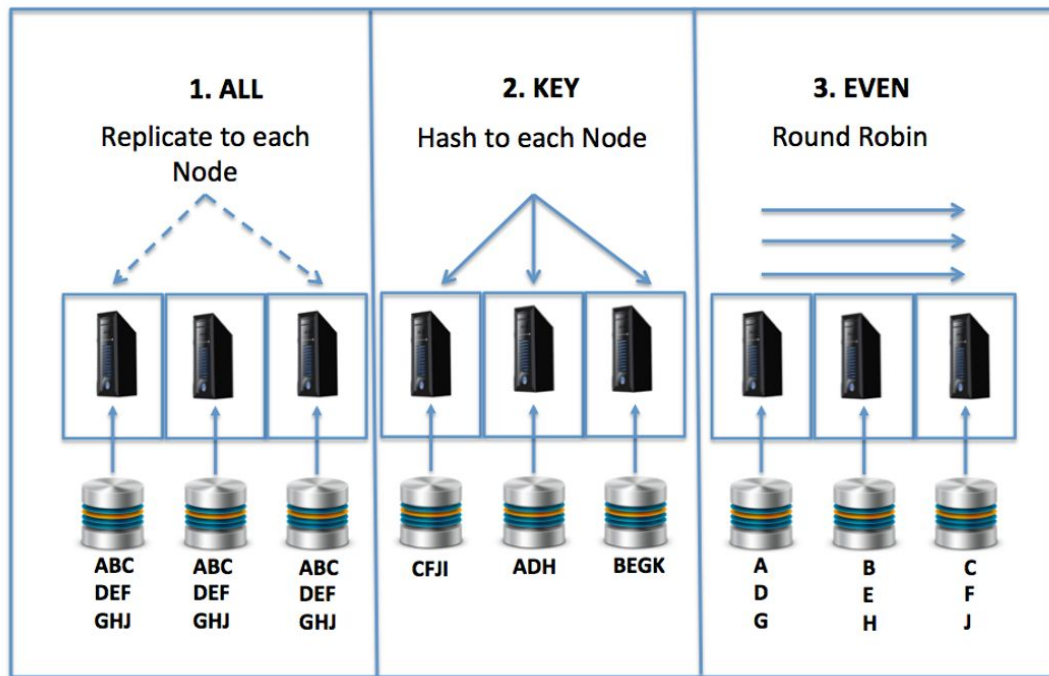


# Что такое MPP, и как это работает?

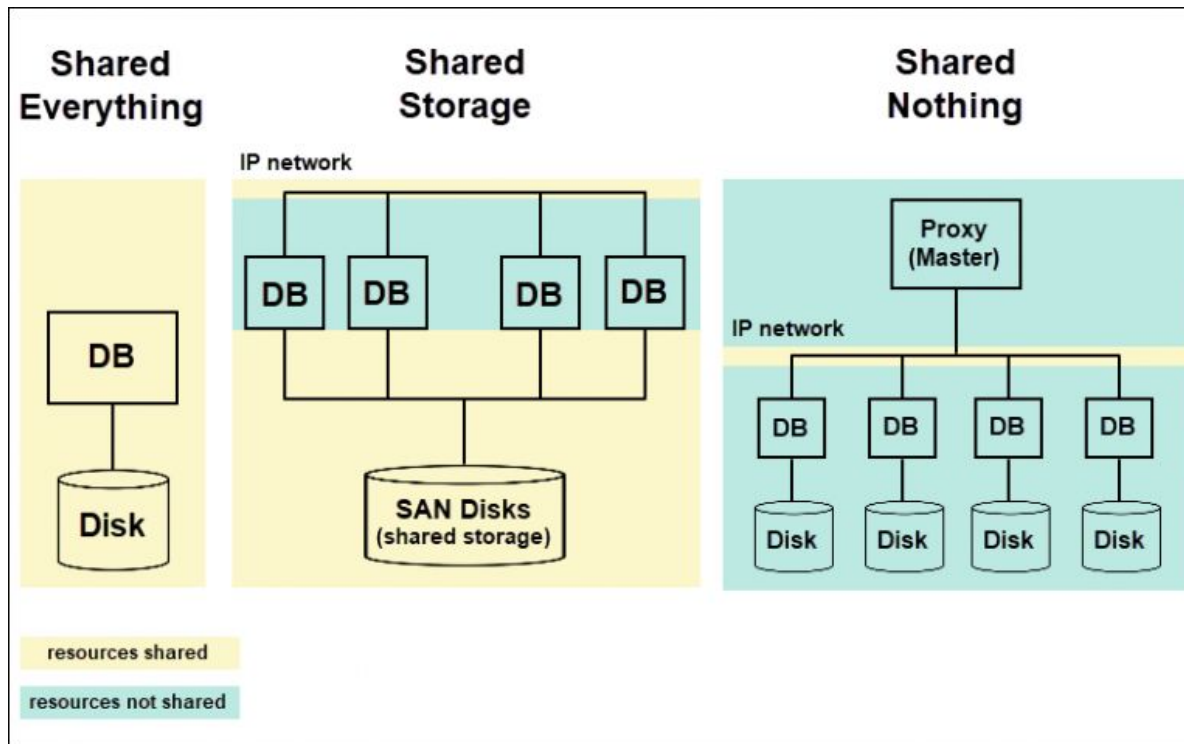


# MPP Data Distribution

## Three MPP Data Distribution Styles



# Shared-\* architecture



# Хранение данных по колонкам (не строкам!)

	Col A	Col B	Col C
Row 0	A0	B0	C0
Row 1	A1	B1	C1
Row 2	A2	B2	C2
Row 3	A3	B3	C3
Row 4	A4	B4	C4
Row 5	A5	B5	C5



A0	B0	C0	A1	B1	C1	A2	B2	C2
A3	B3	C3	A4	B4	C4	A5	B5	C5



A0	A1	A2	A3	A4	A5	B0	B1	B2
B3	B4	B5	C0	C1	C2	C3	C4	C5

# Сжатие данных (алгоритмы и кодеки)

## Modes

value	name	description
0x02	None	No compression, only checksums
0x82	LZ4	Extremely fast, good compression
0x90	ZSTD	Zstandard, pretty fast, best compression

Both LZ4 and ZSTD are made by same author, but with different tradeoffs.

name	ratio	encoding	decoding
zstd 1.4.5 -1	2.8	500 MB/s	1660 MB/s
lz4 1.9.2	2.1	740 MB/s	4530 MB/s

## Block

field	type	description
checksum	uint128	Hash of (header + compressed data)
raw_size	uint32	Raw size without header
data_size	uint32	Uncompressed data size
mode	byte	Compression mode
compressed_data	binary	Block of compressed data



# Представители MPP СУБД



Google BigQuery



# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет

# Показания к применению ClickHouse



# Что такое ClickHouse?

- Column Oriented
- MPP, shared nothing, distributed
- Аналитическая СУБД, которую можно развернуть в облаке, Kubernetes, Docker, локально
- Open Source (Apache 2.0)

## Зачем он нам нужен?

- интерактивные аналитические запросы
- на постоянно пополняемых данных



# История разработки ClickHouse

- 2008 год - первые коммиты в кодовой базе
- 2010 - начало исследований и разработок
- 2012 - использование в проде для решения некоторых задач;
- 2014 - использование в проде в качестве основной технологии сервиса;
- 2015 - широкое распространение внутри Яндекса;
- 2016 - выпущен в open-source!
- Сейчас - более 1000 компаний используют ClickHouse.

# Основной кластер Yandex.Metrica

- Ежедневно вставляется 120 млрд. строк
- 57 605 305 880 525 строк (по состоянию на 2020 год)
- Размер данных 66,41 Пбайт (без сжатия на 1 из 3 реплик)
- Размер данных 5,68 Пбайт (сжатые данные на 1 из 3 реплик)
- 360 серверов в одном кластере, 1 862 сервера в одном отделе

# Что ClickHouse предлагает из коробки?

- переваривает огромные объемы информации
- векторная обработка запросов
- эффективная компрессия данных
- мощный SQL диалект
- репликация и шардирование из коробки
- множество [типов данных](#) (json, IP, UUID, Map, Array, etc)
- работа с геоданными, машинным обучением
- поддержка RabbitMQ, Kafka, MySQL, PostgreSQL ([интеграции](#))

И многое другое

# Что ClickHouse не может предложить?

- [транзакции](#) (\*)
- [изменение данных](#) (update/delete) (\*)
- быстрый доступ по ключу (\*)
- полноценный оптимизатор запросов

# Зачем нужен ClickHouse?

- Производительность.
  - ClickHouse очень быстрый, как и аналитика на нем
- Стоимость.
  - И в первую очередь стоимость масштабирования (антиподы - Vertica, Oracle, Microsoft). ClickHouse open-source
- Операционная стоимость.
  - Это подход чуть-чуть с другой стороны. Примеры - RedShift, Google BigQuery.

# Где используется ClickHouse сейчас?

- Аналитика веб-приложений
- AdTech компании
- Анализ операционных логов с разных источников.
- Мониторинг логов безопасности. Загрузка, хранение и визуализация в BI
- Финансовый анализ
- Телекоммуникационные компании
- Мониторинга производственных процессов. Пример - тестирование электроники и анализ получившейся продукции
- Блокчейн-аналитика

# Реальные примеры. Object Storage

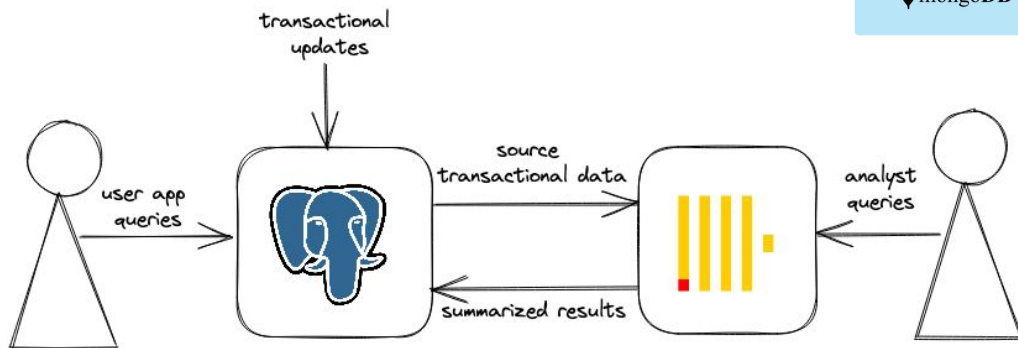
```
SELECT *  
FROM  
s3('https://datasets-documentation.s3.eu-west-3.amazonaws.com/ny  
c-taxi/trips_*.gz', 'TabSeparatedWithNames')  
LIMIT 10;
```

```
CREATE TABLE s3_engine_table (name String, value UInt32)  
    ENGINE = S3(path, [aws_access_key_id,  
aws_secret_access_key,] format, [compression])  
    [SETTINGS ...]
```

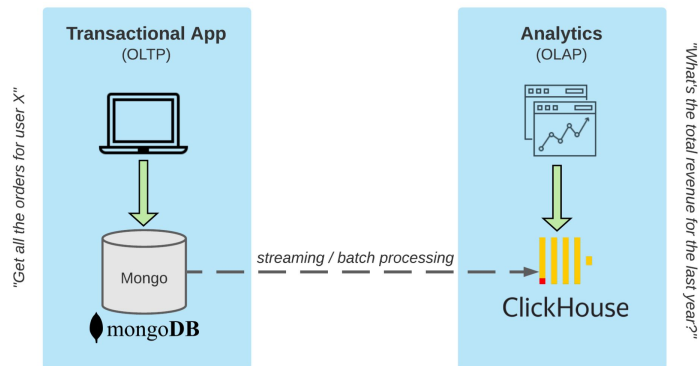


# Реальные примеры. Интеграции с БД

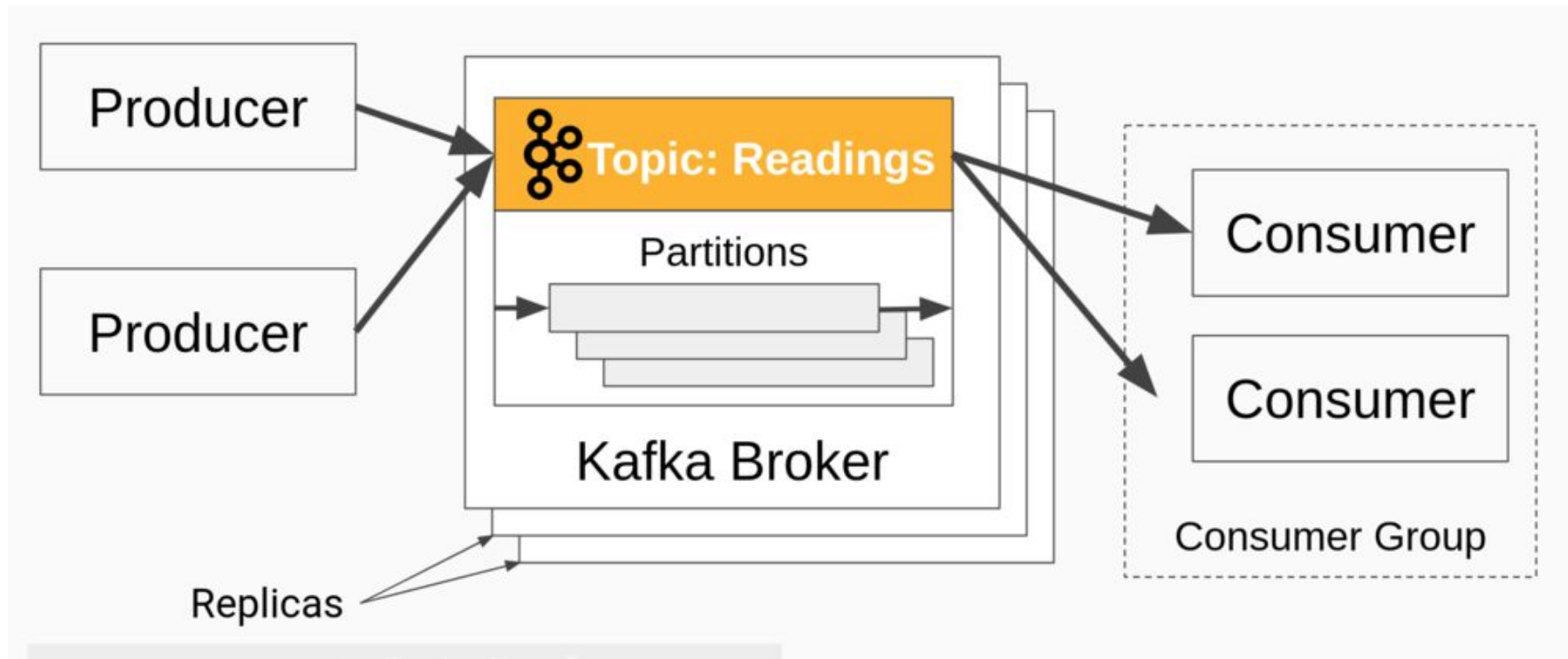
- [Postgres](#) (Greenplum)
- MongoDB
- MySQL
- Redis



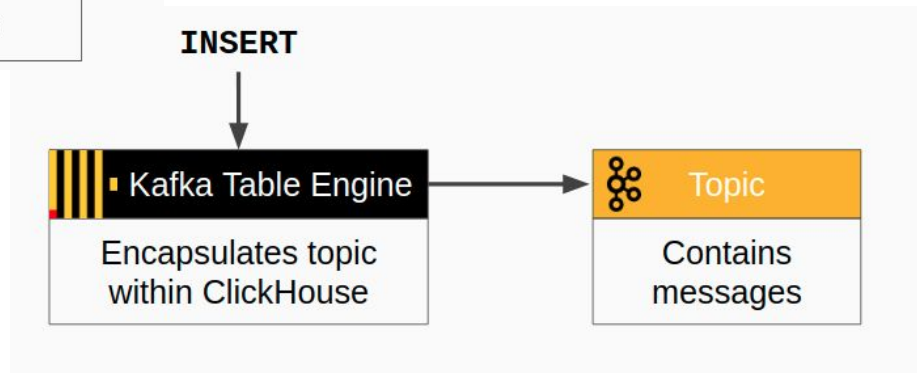
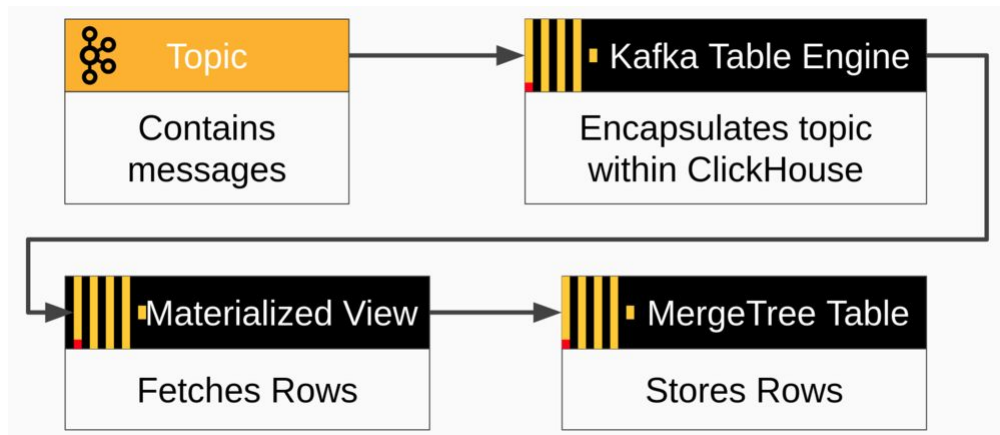
OLTP (Mongo) vs OLAP (ClickHouse)



# Реальные примеры. Apache Kafka



# Реальные примеры. Apache Kafka



# Реальные примеры. Визуализация данных

Многие из популярных BI-инструментов и средств визуализации подключаются к ClickHouse - как "из коробки", так и через установку коннектора.



# Вопросы?



Ставим “+”,  
если вопросы есть



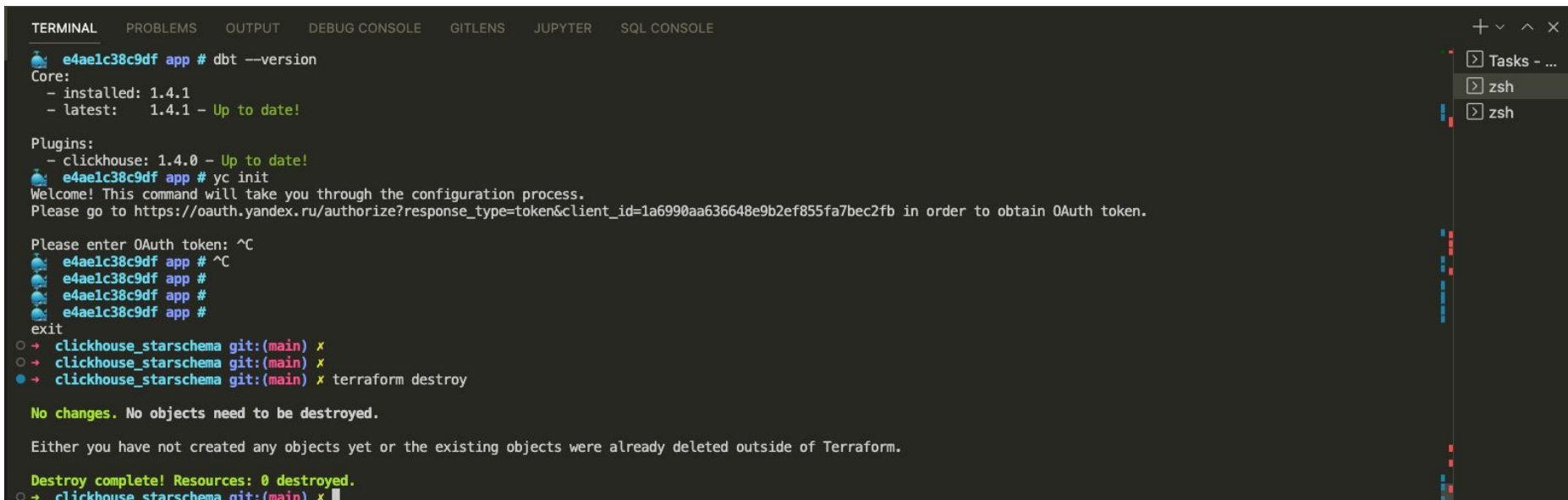
Ставим “-”,  
если вопросов нет

# Инструментарий разработки

# Какой IDE пользуетесь? С чем имели опыт?



# Built-in terminal (ctrl + `)



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  GITLENS  JUPYTER  SQL CONSOLE

e4a1c38c9df app # dbt --version
Core:
- installed: 1.4.1
- latest: 1.4.1 - Up to date!

Plugins:
- clickhouse: 1.4.0 - Up to date!
e4a1c38c9df app # yc init
Welcome! This command will take you through the configuration process.
Please go to https://oauth.yandex.ru/authorize?response_type=token&client_id=1a6990aa636648e9b2ef855fa7bec2fb in order to obtain OAuth token.

Please enter OAuth token: ^C
e4a1c38c9df app # ^C
e4a1c38c9df app #
e4a1c38c9df app #
e4a1c38c9df app #
exit
○ → clickhouse_starschema git:(main) x
○ → clickhouse_starschema git:(main) x
● → clickhouse_starschema git:(main) x terraform destroy

No changes. No objects need to be destroyed.

Either you have not created any objects yet or the existing objects were already deleted outside of Terraform.

Destroy complete! Resources: 0 destroyed.
○ → clickhouse_starschema git:(main) x
```





# Z shell (zsh)

- [Installing zsh](#)

# Shell commands basics

- pwd, ls, cd, mkdir, rm
- hotkeys: editing, history (ctrl+r), interrupt (ctrl+c)
- [Linux tutorial](#)
- [Cheat sheet](#)

# git basics

- Essential git commands: status, log, checkout, add, commit, push, fetch, [rebase](#), merge
- [Atlassian tutorials](#)
- [Collaborating with git](#)
- [Making a Pull Request](#)
- [Comparing Git Workflows](#)

# Docker

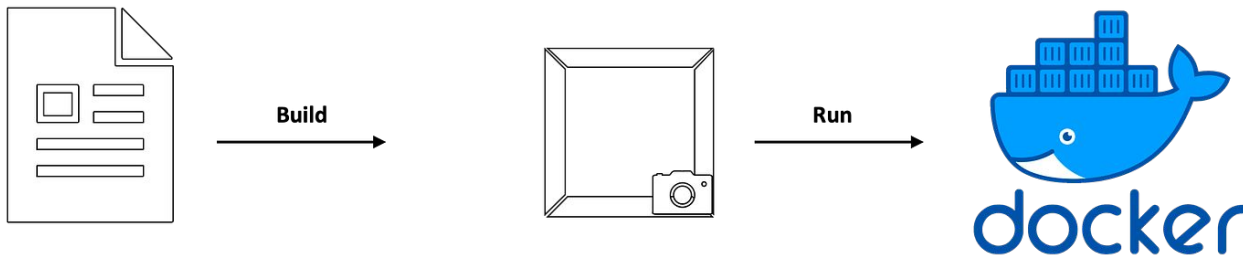
**Docker** - инструмент для создания и управления контейнерами

- **Dockerfile** содержит набор инструкций для создания образа Docker.
- Образ Docker (**Docker Image**) служит шаблоном для создания контейнеров Docker.
  - Содержит весь необходимый код, среду выполнения, системные инструменты, библиотеки и настройки, необходимые для запуска программного приложения.
- **Dockerfile используется для создания Docker Image, который затем используется в качестве шаблона для создания одного или нескольких контейнеров Docker.**

1. Dockerfile

2. Docker Image

3. Docker Container(s)



# Пару слов

- Docker - это компьютерное программное обеспечение, используемое для виртуализации с целью обеспечения работы нескольких операционных систем на одном хосте.
- Docker - это клиент-серверный тип приложения, то есть у нас есть клиенты, которые передают данные на сервер.
- Образы Docker - это "исходный код" для наших контейнеров; мы используем их для сборки
- Dockerfile имеет два типа реестров 1) публичный и 2) приватный реестры
- Контейнеры - это организационные единицы тома Docker. Проще говоря, образ - это шаблон, а контейнер - копия этого шаблона. Можно иметь несколько контейнеров (копий) одного и того же образа.

# Yandex Cloud CLI

- [Install yc CLI](#)
- [yc CLI reference](#)
- [Yandex Object Storage CLI \(AWS CLI\)](#)

# Вопросы?



Ставим “+”,  
если вопросы есть



Ставим “-”,  
если вопросов нет

# Список материалов для изучения

1. [Shared Nothing Architecture Explained](#)
2. [Clickhouse vs. Greenplum. Какую MPP-базу данных выбрать? // Демо-занятие курса «Data Engineer»](#)
3. [What is an MPP Database? Intro to Massively Parallel Processing](#)
4. [Row vs Column Oriented Databases](#)
5. <https://clickhouse.tech/>
6. <https://www.altinity.com/blog>



# Рефлексия

# Рефлексия



С какими впечатлениями уходите с вебинара?

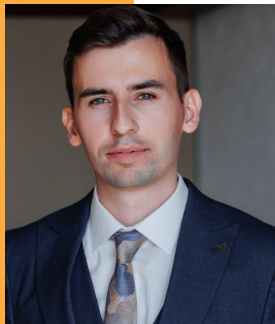


Как будете применять на практике то, что узнали на вебинаре?

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Приходите на следующие вебинары



**Алексей Железной**

**Senior Data Engineer/Architect**

Магистратура - ФКН ВШЭ

Руководитель курсов **DWH Analyst, ClickHouse** для инженеров и архитекторов  
**БД** в OTUS

Преподаватель курсов **Data Engineer, DWH Analyst, PostgreSQL** и пр. в OTUS

[LinkedIn](#)

