

# СИСТЕМЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

## Разработка приложений



К.Т.Н.  
Папулин Сергей Юрьевич  
*[papulin\\_bmstu@mail.ru](mailto:papulin_bmstu@mail.ru)*

# Программа курса



- Hadoop: Hadoop Distributed File System (HDFS)
- Управление ресурсами и приложениями (YARN), платформа MapReduce
- Apache Spark. Распределенная координация с Zookeeper
- Системы потоковой обработки. Apache Storm. Spark Streaming. Kafka
- Системы обработки графов: Giraph, Spark GraphX, Spark GraphFrames
- Docker и Kubernetes. Управление контейнеризированными приложениями

- HDFS и MapReduce
  - Spark. Основные операции над RDD
  - Spark. Основные операции над Dataframe
  - Spark. Взаимодействие с HDFS, Parquet, Avro
  - Поточковая обработка. Разработка приложений под Spark Streaming и Structured Streaming
  - Обработка графов. Разработка приложений под Spark GraphFrame
  - Развертывание Spark на Docker и Kubernetes
- PK1,2

- MapReduce
- Spark + Kafka
- Spark + MLlib
- Spark GraphFrame

**cloudera**



Java, Scala, Python

<https://github.com/bigdataprocsystems>

**Модуль 1**

Д31:

- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$PK1 = 35/20 \cdot (K1.1 \cdot \text{Д31.Часть 1} + K1.2 \cdot \text{Д31.Часть 2})$$

$$PK1 = K1.1 \cdot \text{Д31.Часть 1} + K1.2 \cdot \text{Д31.Часть 2} + \text{Вопросы (15 баллов)}$$

$K$  – коэффициент (1; 0.85; 0.7)

**Модуль 2**

Д32:

- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$PK2 = 35/20 \cdot (K2.1 \cdot \text{Д32.Часть 1} + K2.2 \cdot \text{Д32.Часть 2})$$

$$PK2 = K2.1 \cdot \text{Д32.Часть 1} + K2.2 \cdot \text{Д32.Часть 2} + \text{Вопросы (15 баллов)}$$

# Рабочая машина



# Лекция 1. Концепция Больших Данных



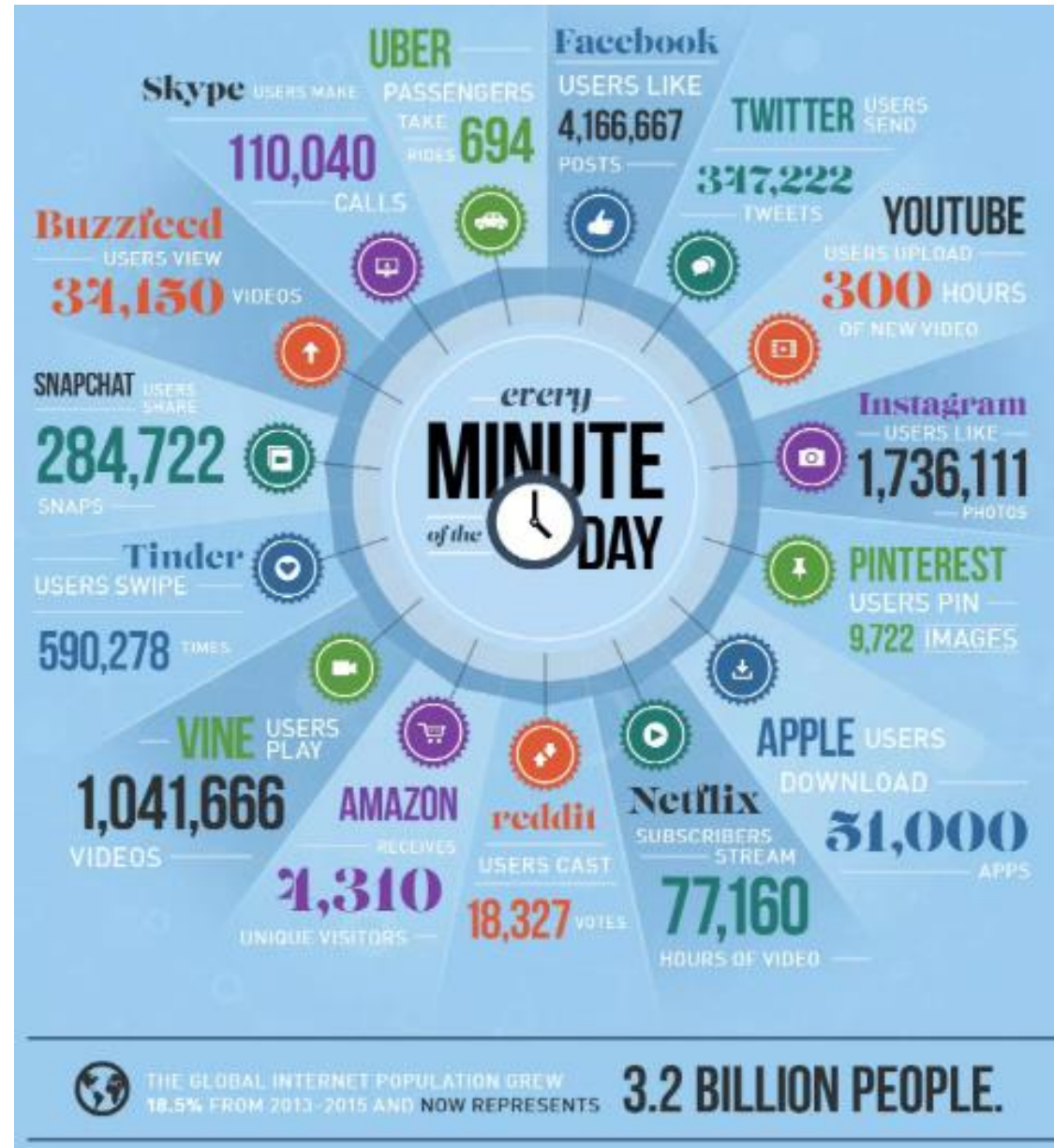


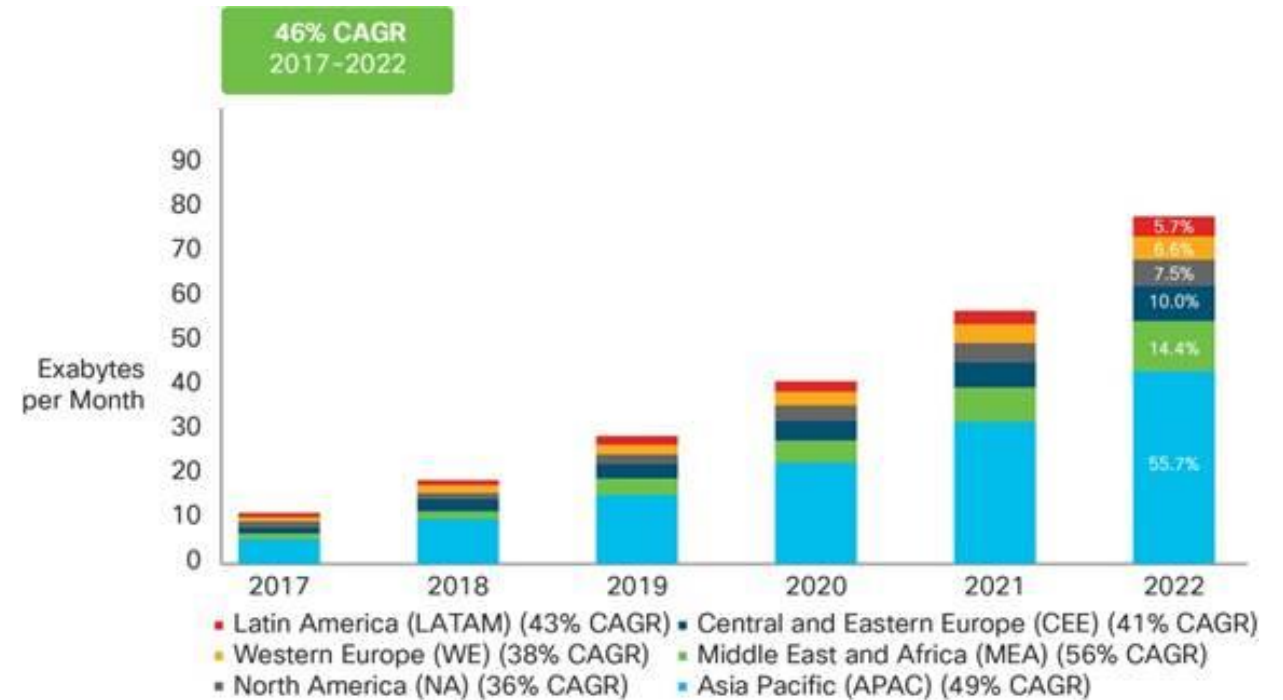
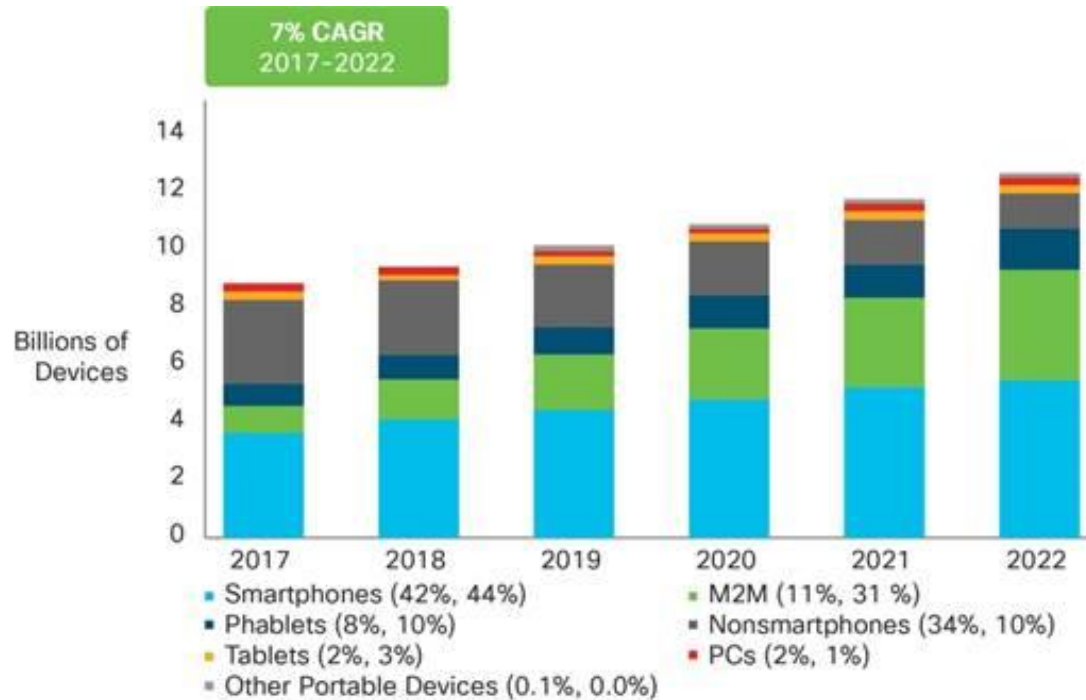
- Большие данные – 4V
- Параллельные и распределенные вычисления
- Системы обработки и хранения больших данных
- стек технологий
- Облачные ресурсы

# Системы обработки больших данных

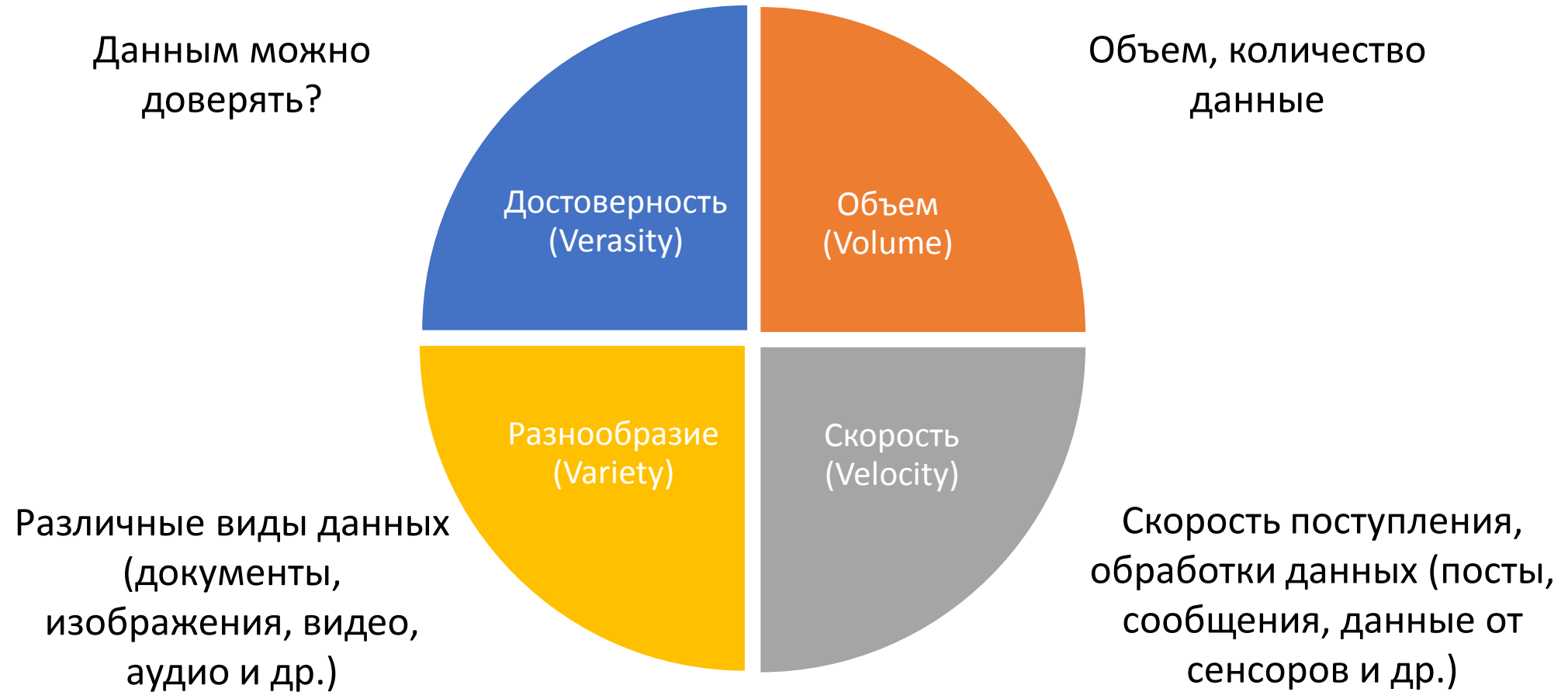
## Пример, Facebook

- 2.23 млрд. активных пользователей в месяц (2018)
- 90,032 постов в день (2018)





# Большие Данные – 4V



- Чем больше данных у нас есть, тем больше знаний мы можем извлечь, лучшее решение можем принять
- Чем быстрее обрабатываются поступающие данные, тем быстрее можно начать анализ
- Чем более разнообразные источники данных (социальные сети, история просмотров, покупок и пр.), тем лучше можно составить портрет клиента
- Чем более достоверные данные, тем точнее можно составить портрет клиента

# Источники данных

## Публичные данные

Экономические  
Перепись  
Гео-информация  
Погода  
Открытые данные

## Коммерческие данные

Бизнес-информация  
Исследования рынка  
Кредитное бюро

## Социальные сети

Сообщества  
Блоги  
Twitter, Facebook, LinkedIn, Tumblr

## Операционные данные

Сенсоры  
GPS  
Транзакции

## Корпоративные данные

Взаимодействия с клиентами  
Отчеты  
Логи  
Контакты

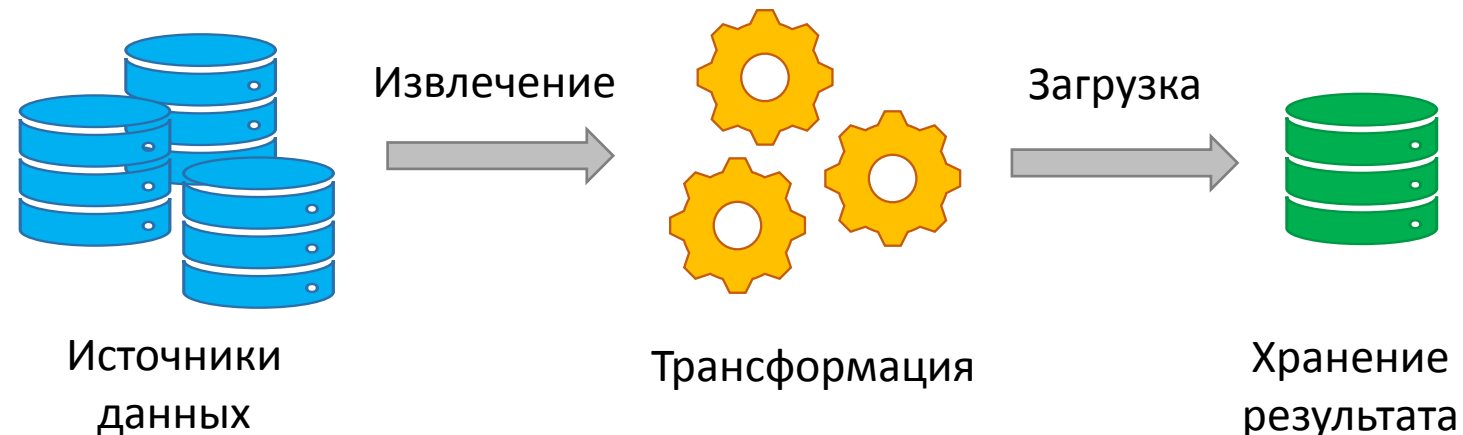
- ETL (Extract, Transform, Load)
- Поиск информации
- Анализ данных
- Машинное обучение



# Extract Transform Load

# ETL. Общая схема

- Извлечение данных из различных внешних источников (БД, приложения, системы)
- Преобразование и очистка данных в соответствии с целями обработки
- Загрузка результата в хранилище данных

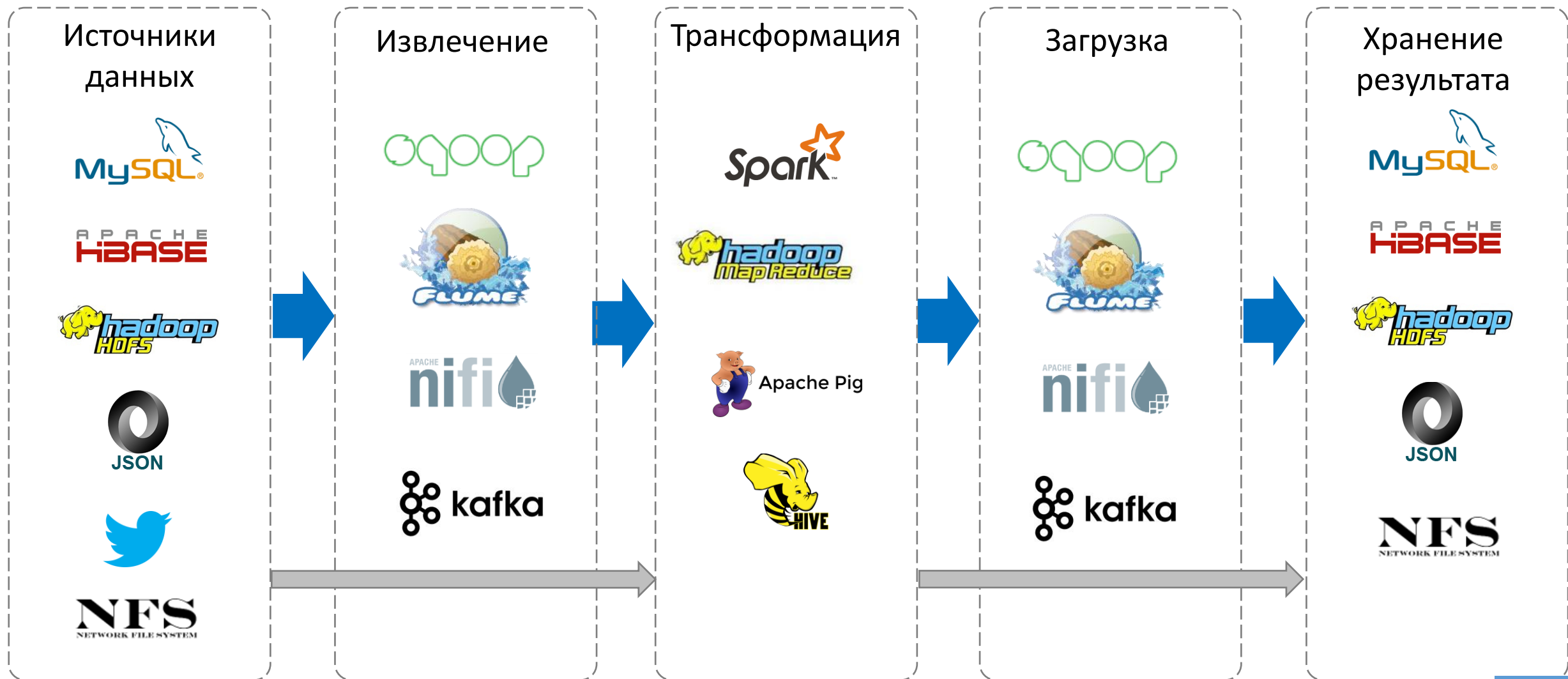


# ETL. Базовые трансформации

- **Очистка (cleaning)**  
обработка null значений, пропущенных данных, аномальных значений и пр.)
- **Преобразование формата (format revision)**  
преобразование формата даты, времени, единиц измерения и пр.)
- **Преобразование структуры данных (restructuring)**
- **Дедубликация (deduplication)**  
удаление повторных записей

- **Фильтрация** (filtering)
- **Получение производных данных** (derivation)
- **Агрегирование** (aggregation)
- **Обобщение/резюме** (summarization)  
сводные данные на разных уровнях, например, страна, регион, город и пр.
- **Слияние** (merging)
- **Разделение** (splitting)  
(например, одного столбца на несколько)  
  
и др.

# ETL. Общая схема



# Поиск информации



## Индексирование текстовых документов

- Формирование вектора термов документа (предобработка, токенизация, стемминг и пр.)
- Формирование инвертированного индекса



## Поиск по текстовому запросу

- Формирование вектора термов запроса (предобработка, токенизация, стемминг и пр.)
- Отбор документов по инвертированному индексу
- Ранжирование отобранных документов по релевантности

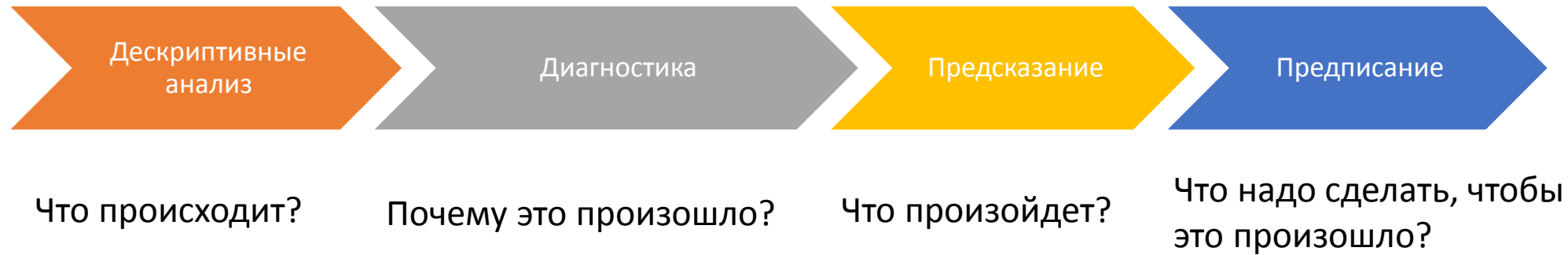


## Оптимизация



# Анализ данных и машинное обучение



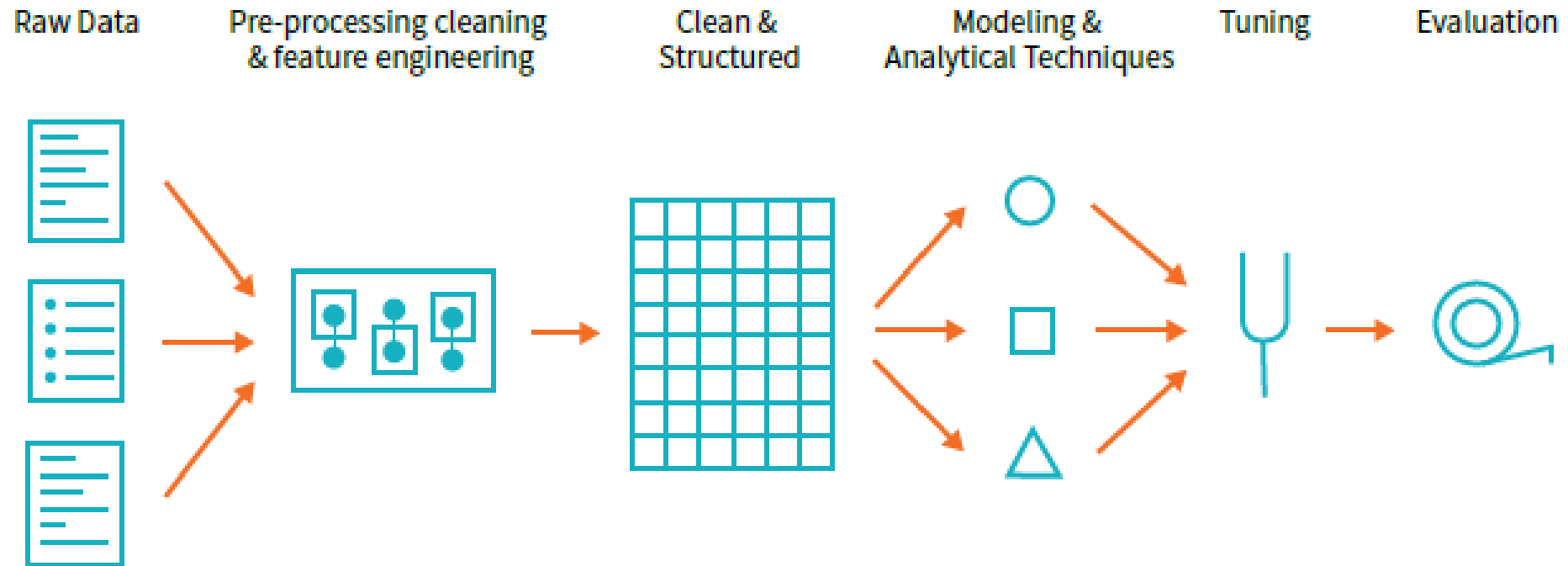


- Дескриптивный анализ
- Предобработка данных (очистка, отбор признаков, преобразование признаков)
- Обучение с учителем (регрессия, классификация)
- Обучение без учителя (кластеризация, уменьшение размерности, выявление аномалий, тематическое моделирование)
- Рекомендательные системы (рекомендации товаров на основе поведения и предпочтений)
- Анализ графов (выявление взаимосвязей различных структур в графе)
- Глубокое обучение

# Решаемые задачи

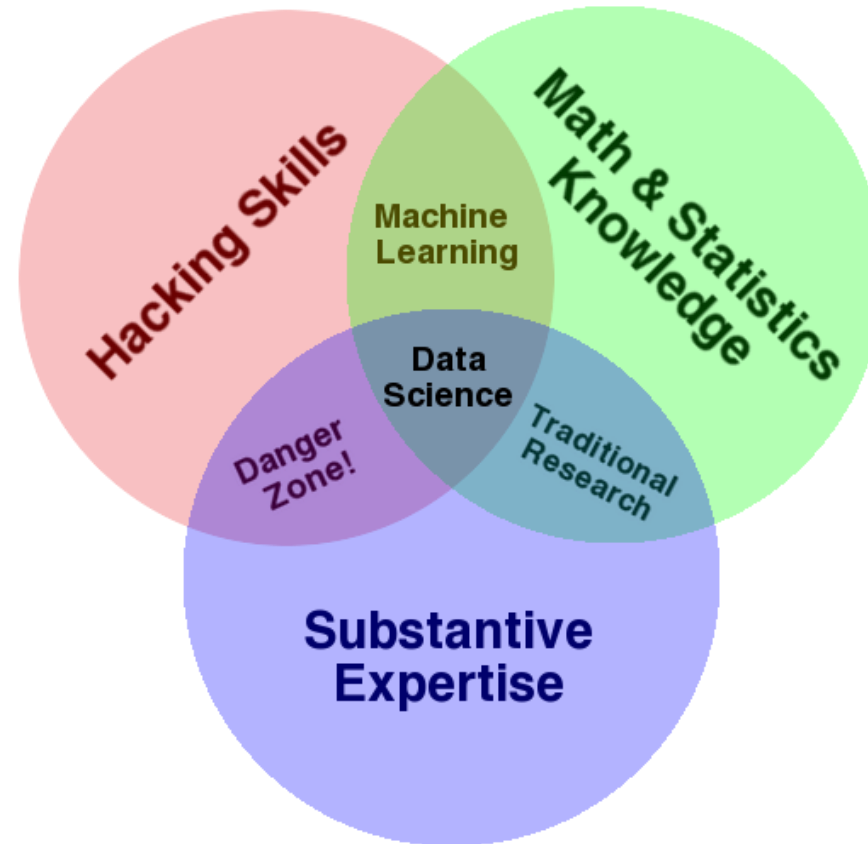


# Анализ данных и машинное обучение



Spark: The Definitive Guide By Bill Chambers & Matei Zaharia (book)

# Наука о данных (Data Science)

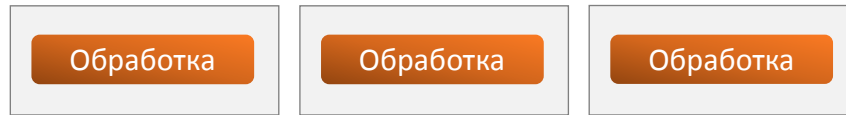


# Архитектура систем обработки больших данных

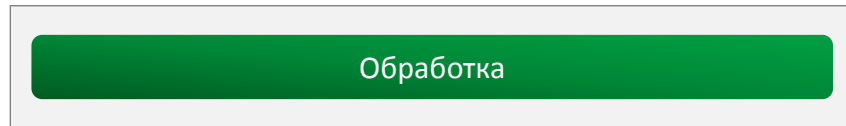
# Вычислительные ресурсы

## Общие данным

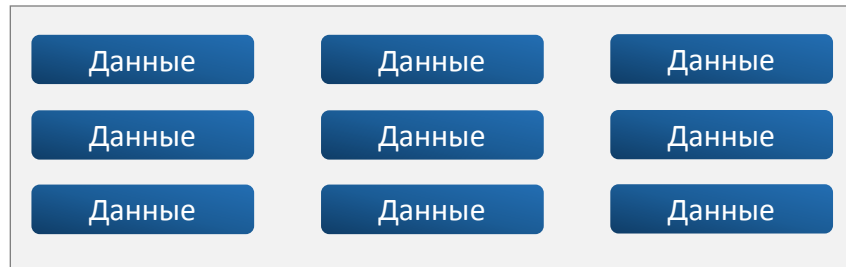
Приложения



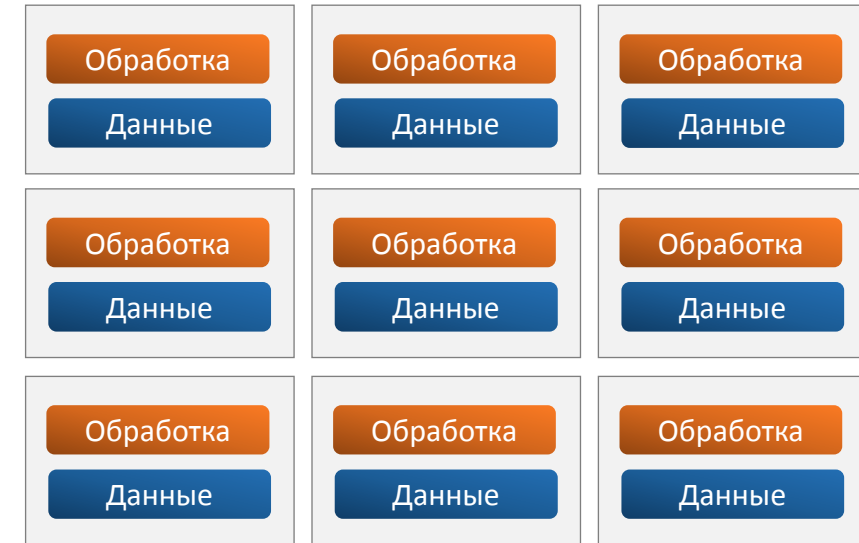
СУБД



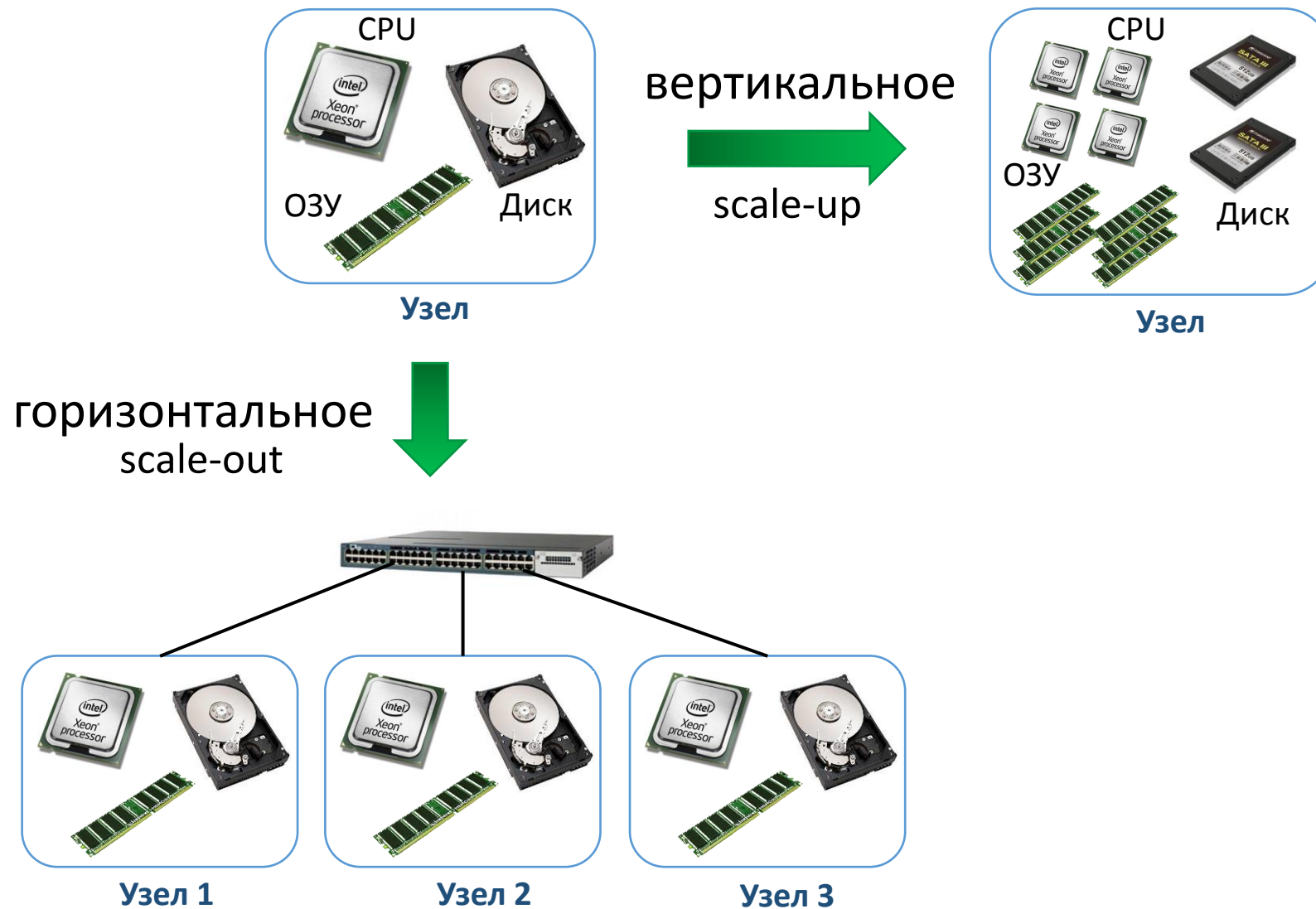
SAN/NAS



## Данные обрабатываются там же, где они хранятся



# Наращивание производительности





# Кластер

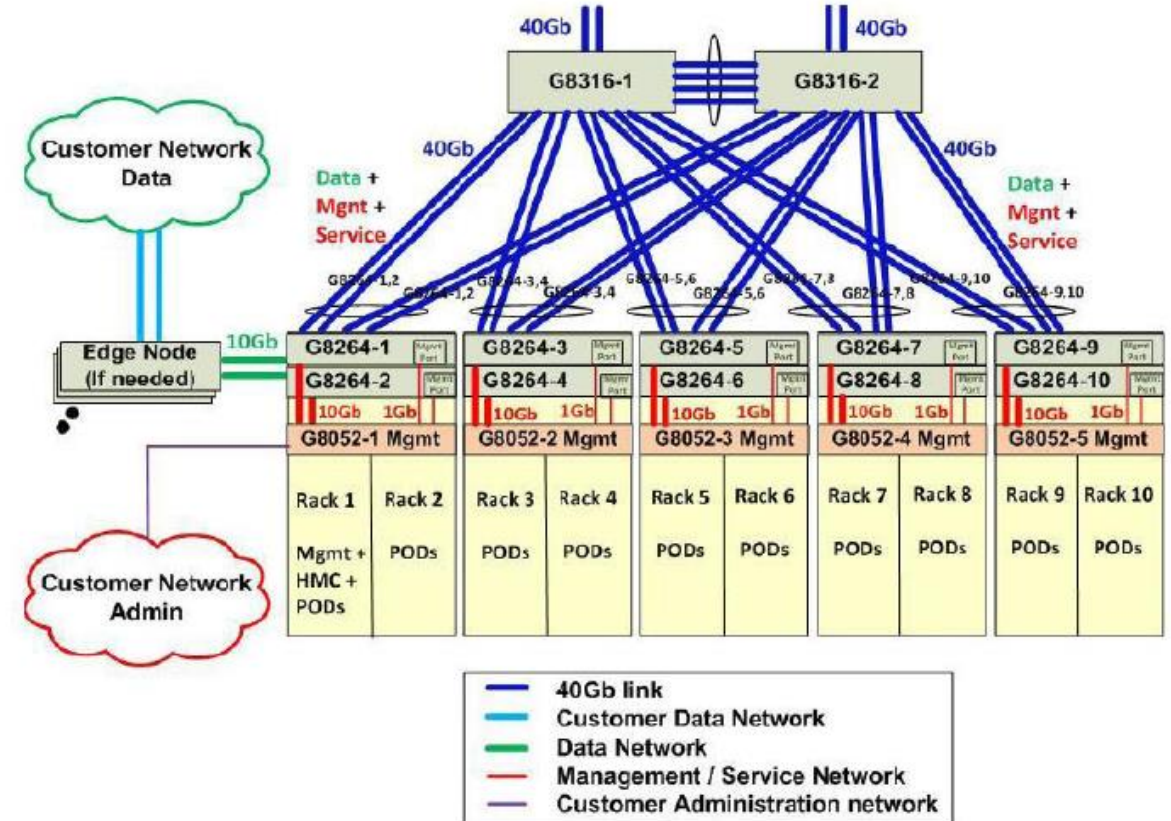
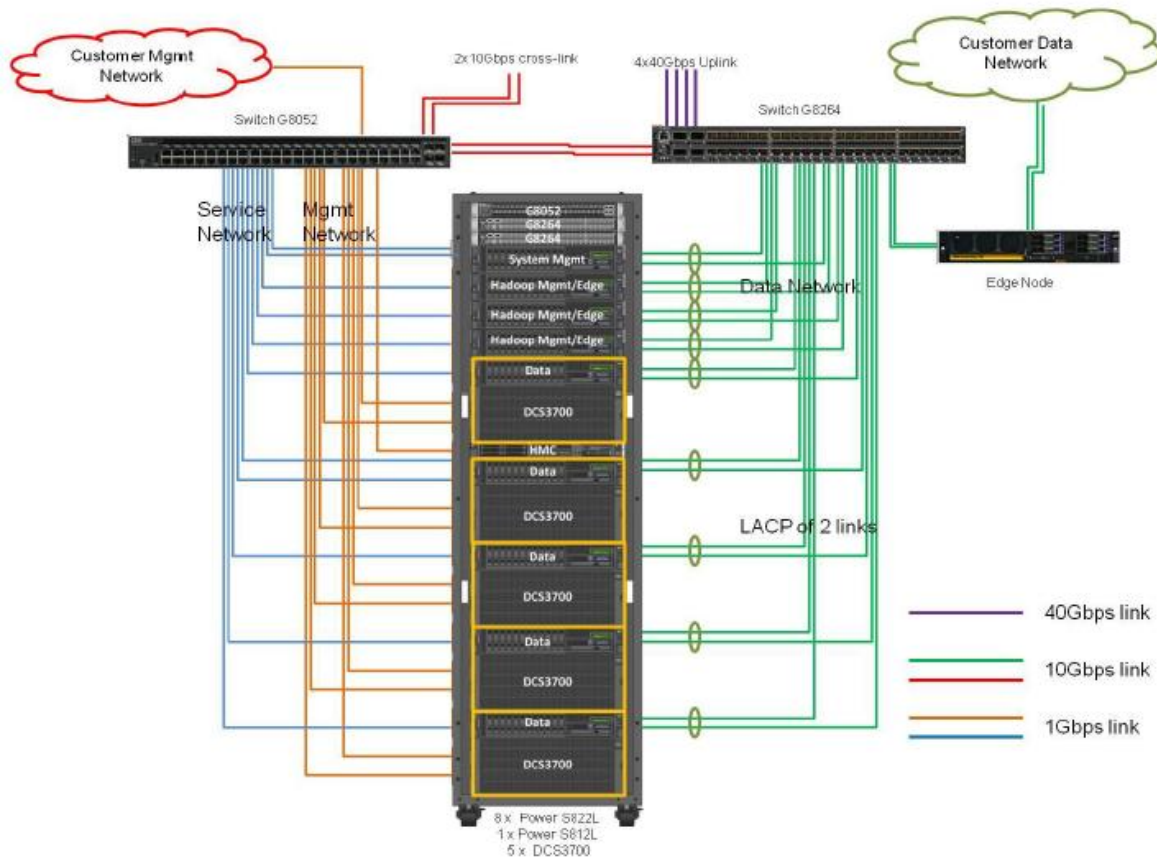
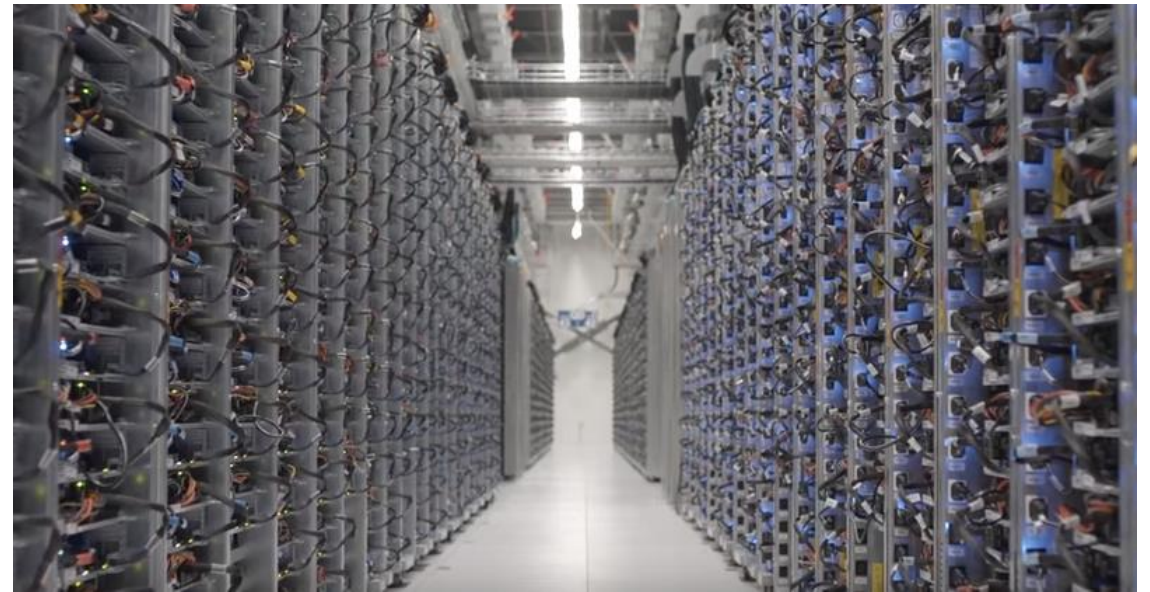
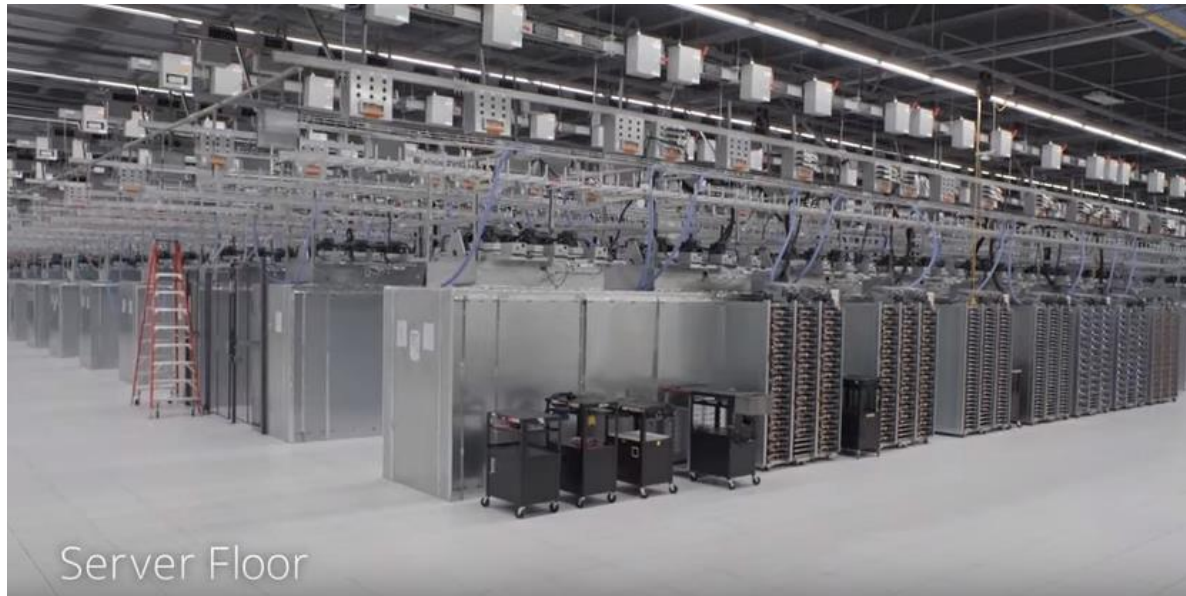
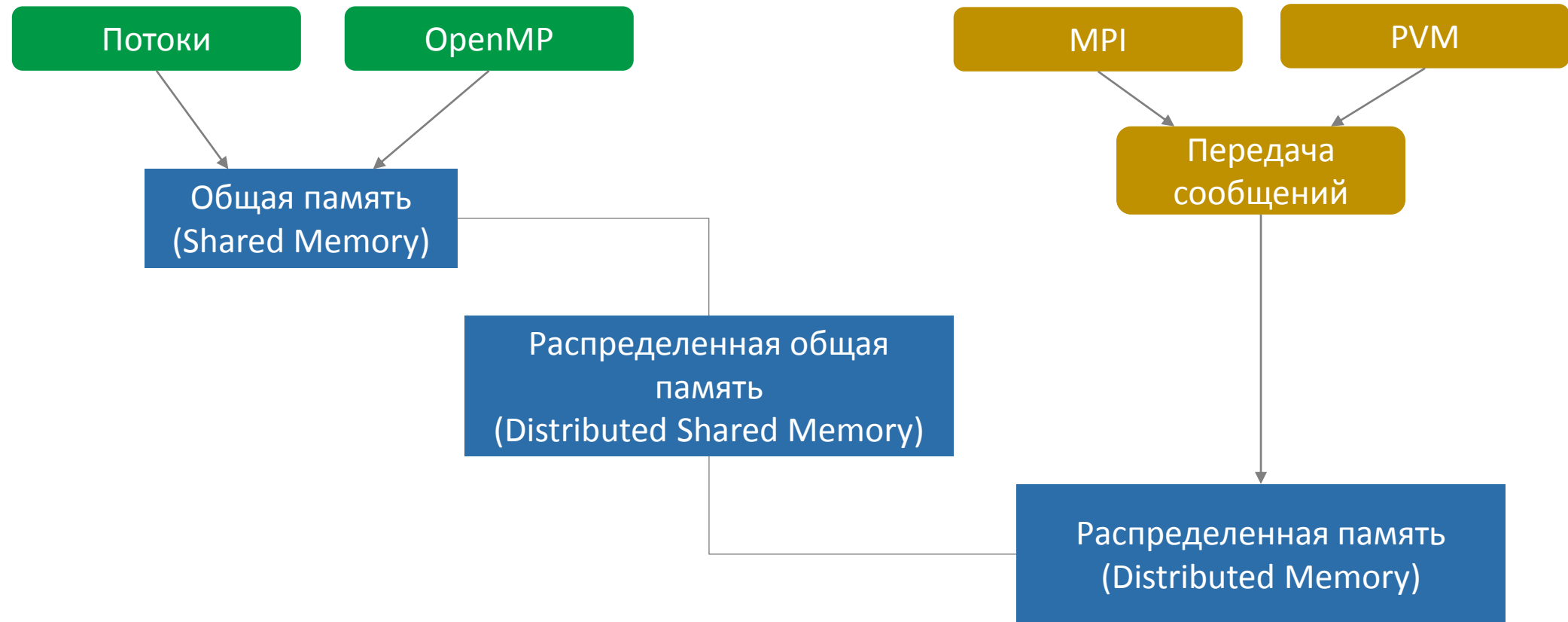


Figure 6: Cross-rack networking



# Параллельные и распределенные вычисления

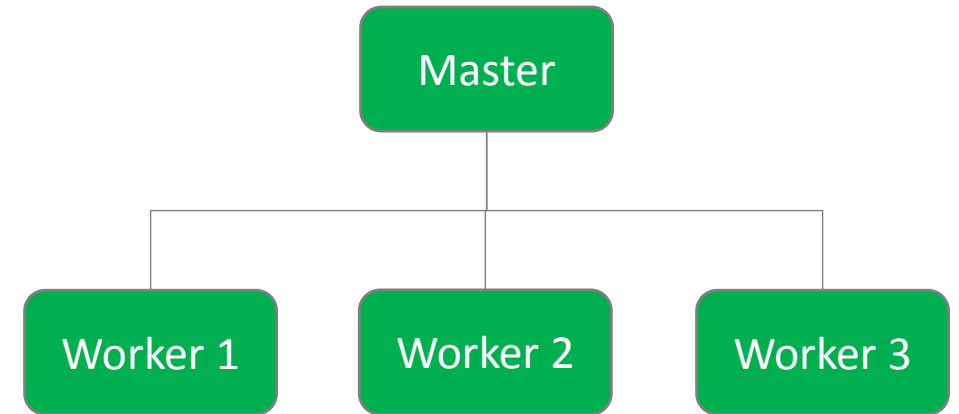
# Параллельное программирование



# Распределенные приложения

# Что необходимо учитывать

- Сеть не обязательно надежна
- Существуют задержки
- Пропускная способность ограничена
- Сеть может быть небезопасной
- Топология может изменяться
- Может быть несколько администраторов
- Существуют транспортные затраты
- Сеть может быть гетерогенной





## Синхронная

HTTP REST    THRIFT



## Асинхронная

WebSocket



kafka



redis



RabbitMQ



APACHE  
ZooKeeper™



## ➤ Выбор мастера (Master election)

Назначение задач worker'ам

## ➤ Обнаружение отказа (Crash detection)

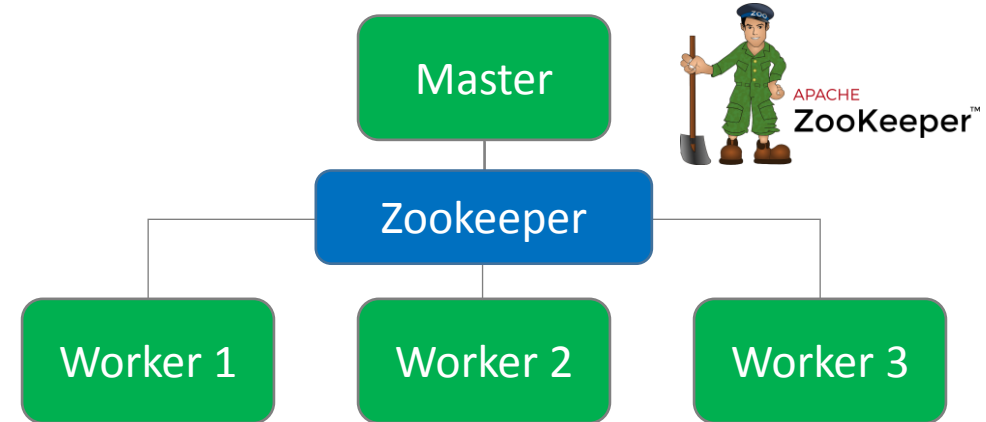
Мастер должен уметь определять, что worker вышел из строя или с ним потеряно соединение

## ➤ Управление группами (Group membership management)

Мастер должен знать, какие worker'ы доступны для выполнения задач

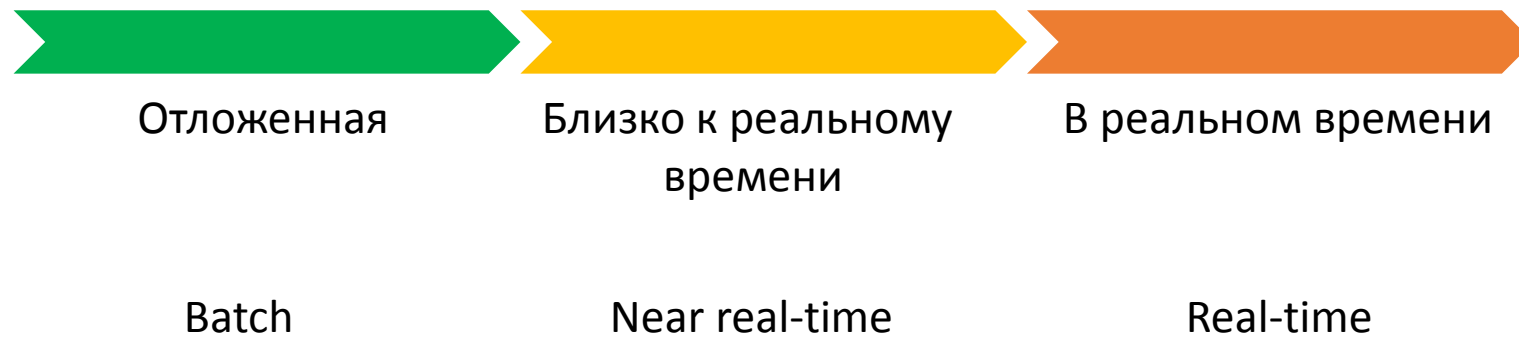
## ➤ Управление метаданными (Metadata management)

Мастер и worker'ы должны хранить задания и статусы выполнения надёжным способом





# Системы обработки и хранения больших данных



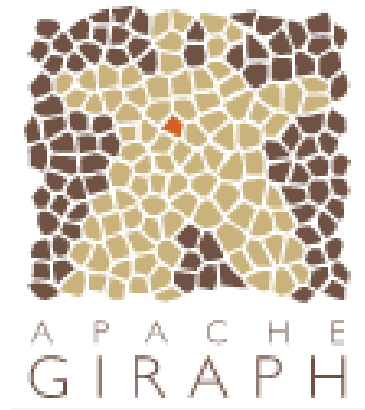
## Принципы выполнения вычислений



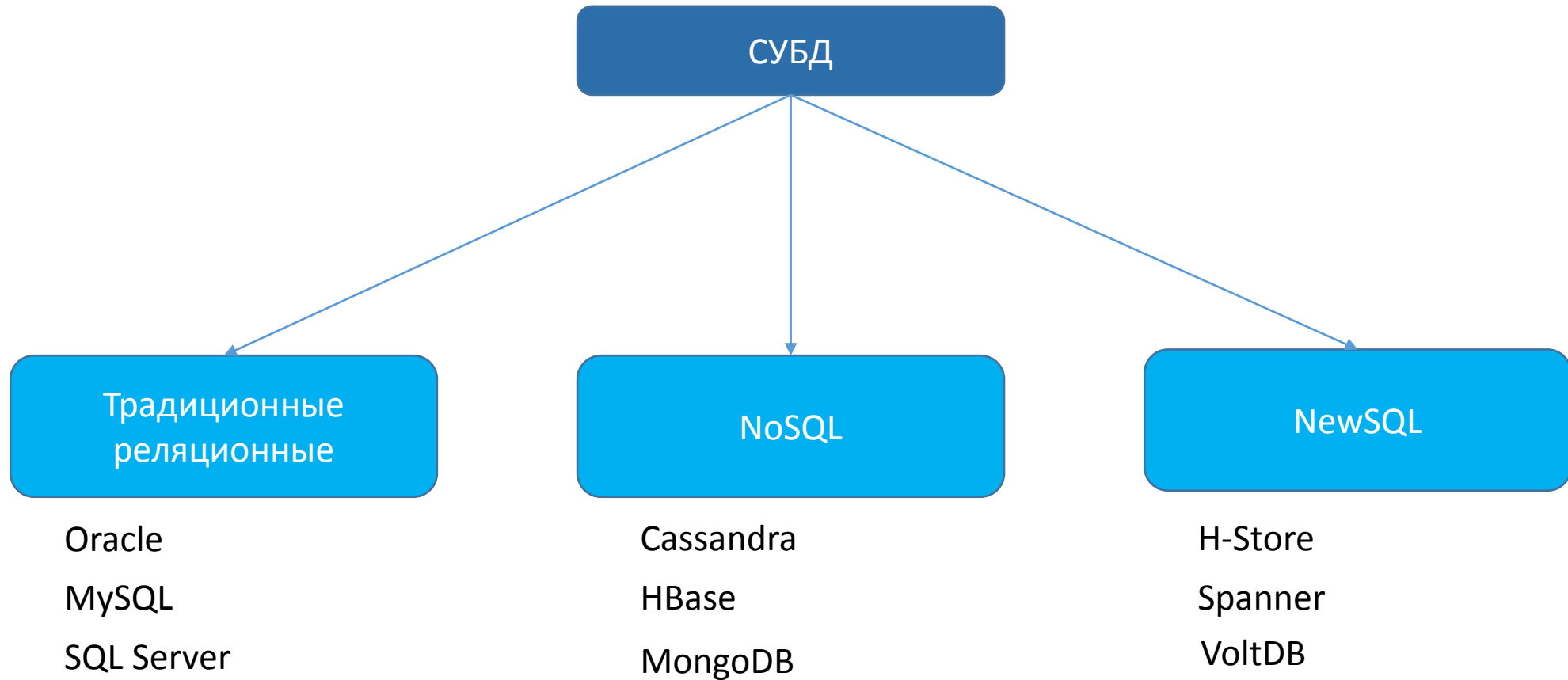
# Обработка коллекций данных (batch processing)







# Классификация NoSQL СУБД



## ACID

Атомарность (Atomicity)

Согласованность (Consistency)

Изолированность (Isolation)

Долговечность (Durability)

## BASE

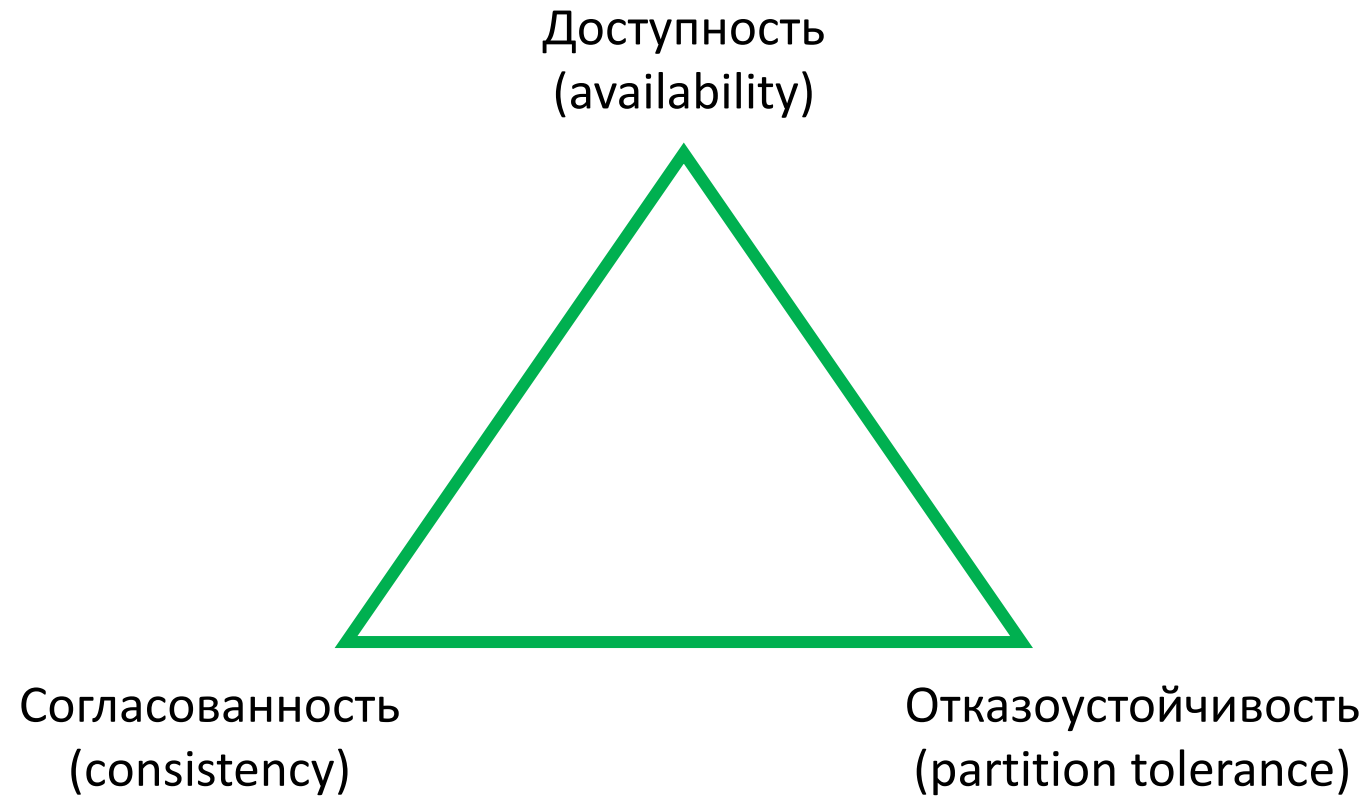
Basic Availability

Soft-state

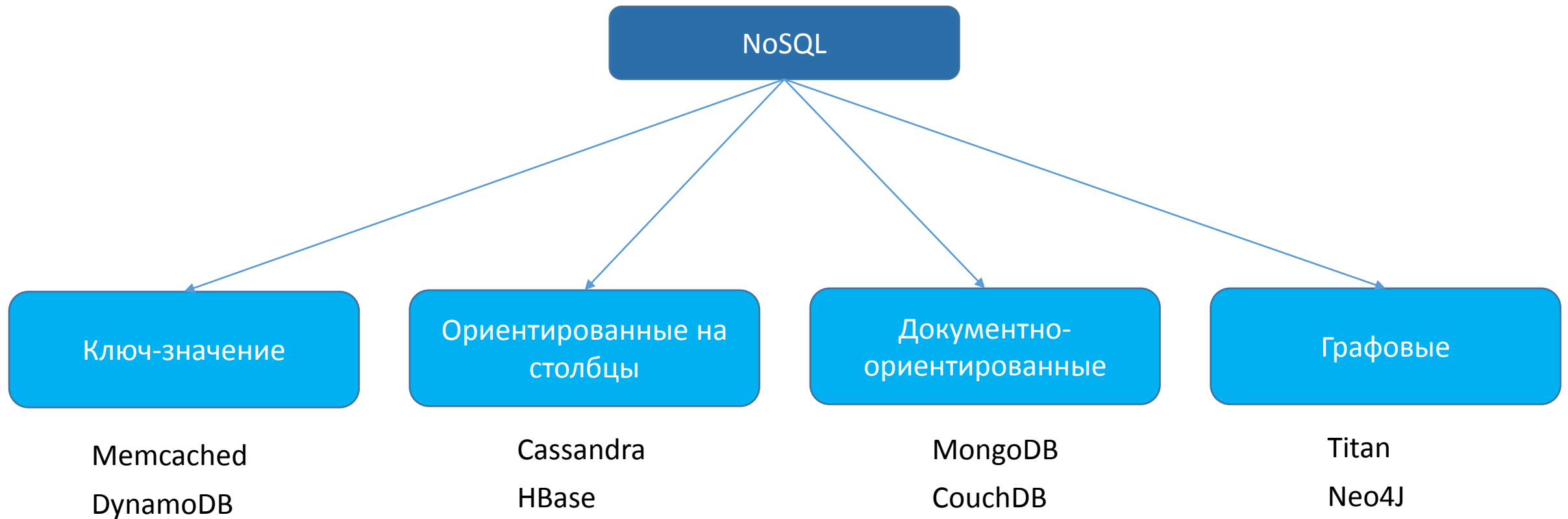
Eventual consistency



# CAP



# Классификация NoSQL СУБД



# Примеры СУБД

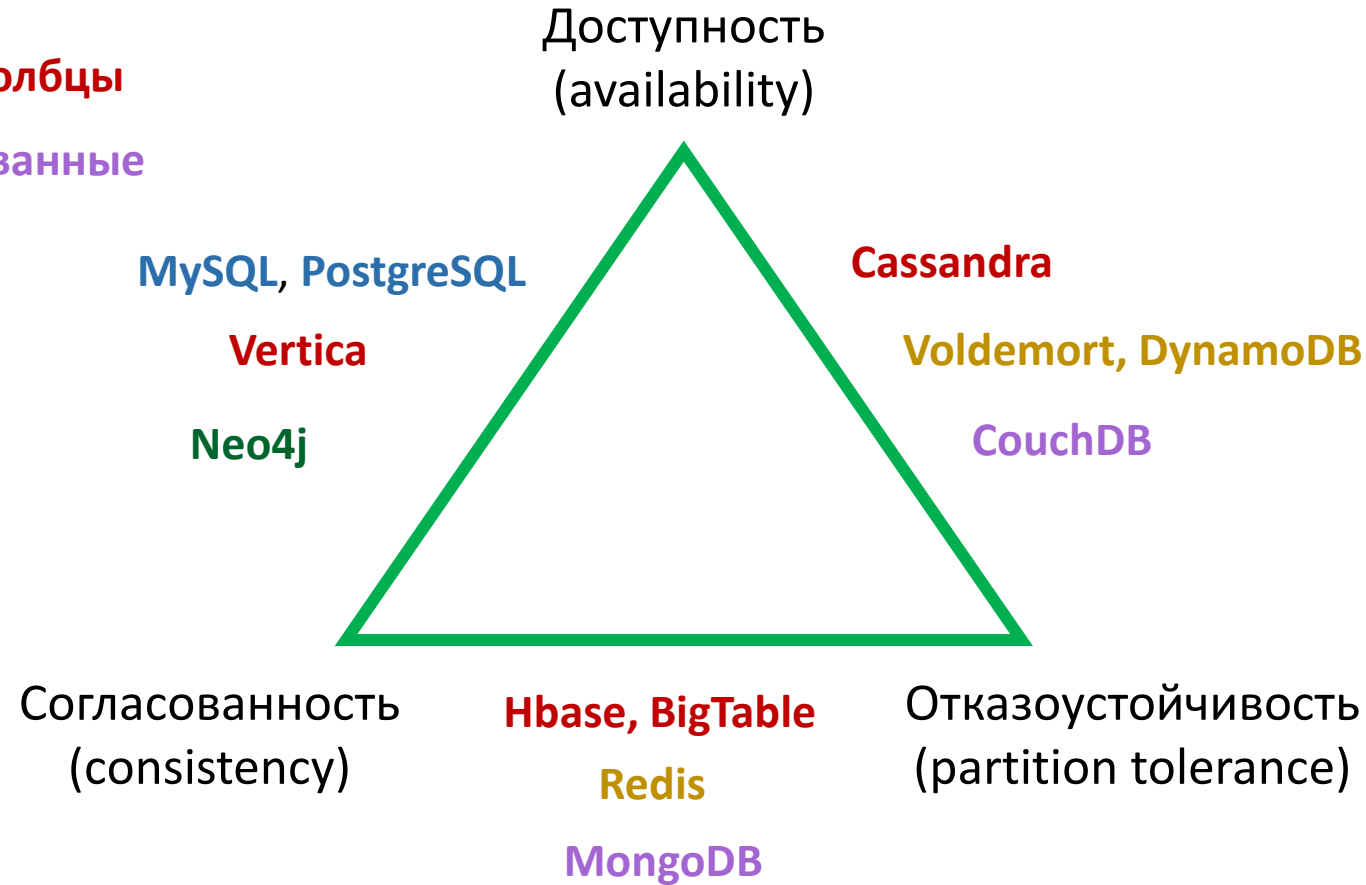
Реляционные

Ключ-значение

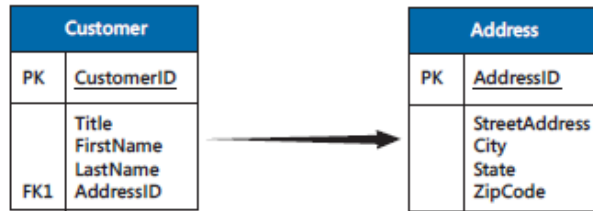
Ориентированные на столбцы

Документно-ориентированные

Графовые



# Примеры СУБД



Customer Table

CustomerID	Title	FirstName	LastName	AddressID
1	Mr	Mark	Hanson	500
2	Ms	Lisa	Andrews	501
3	Mr	Walter	Harp	500

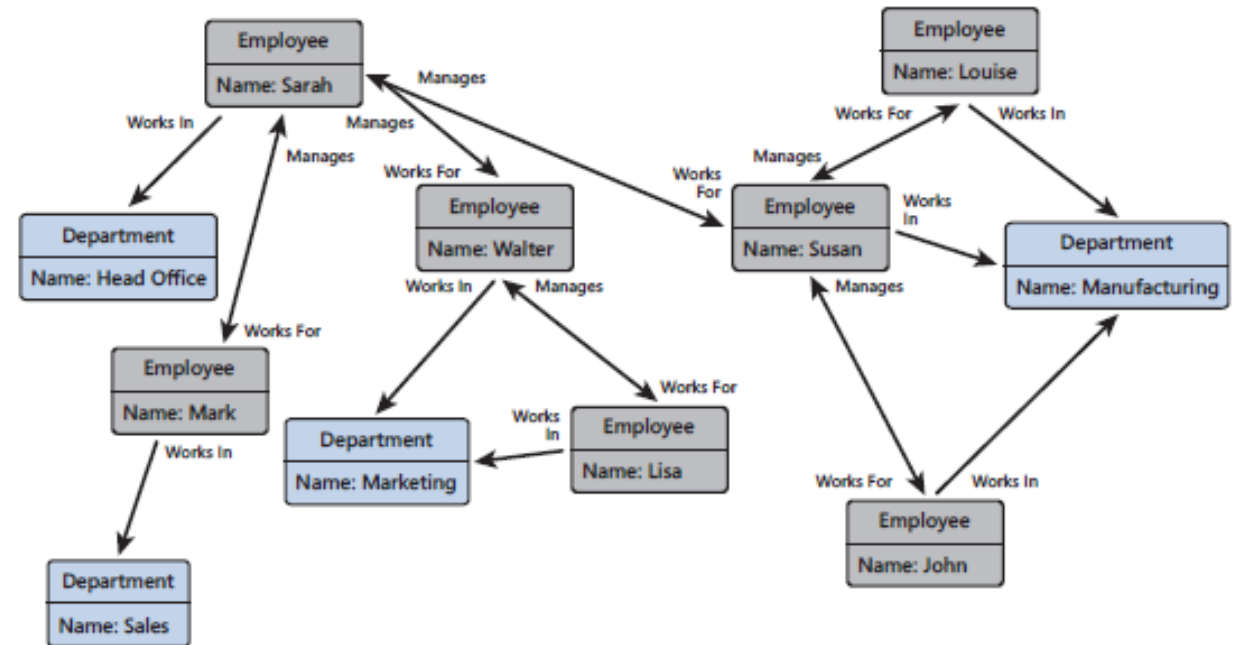
Address Table

AddressID	StreetAddress	City	State	ZipCode
500	999 500th Ave	Bellevue	WA	12345
501	888 W. Front St	Boise	ID	54321

Key	Value (blob)
AAAAA	110100100100111101001001001
AABAB	000110100111100100011110010
DFA766	01011001100100110011111001011
FABCC4	1111000011001010010110011001

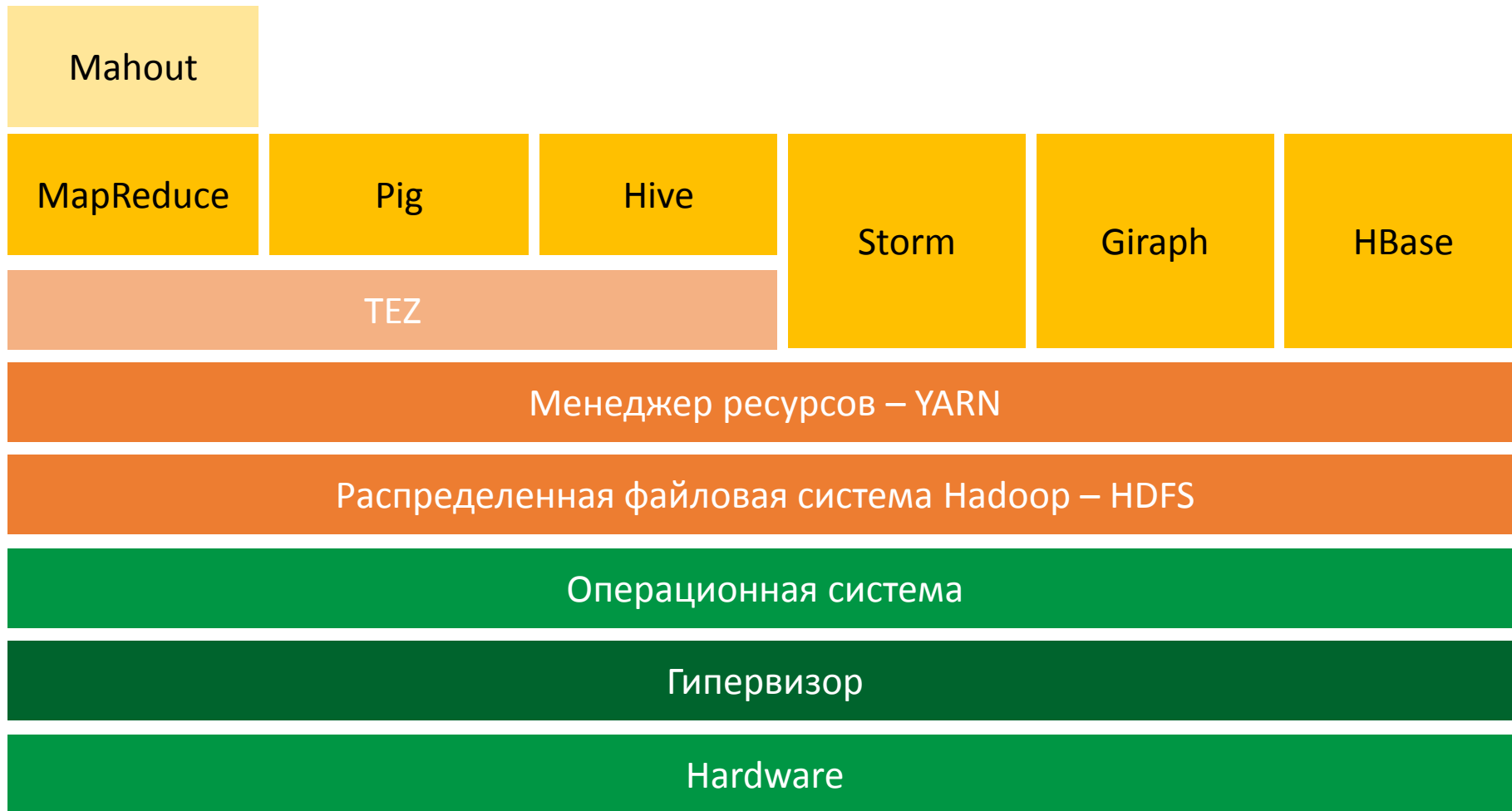
Row Key	Column Families	
CustomerID	CustomerInfo	AddressInfo
1	CustomerInfo:Title Mr CustomerInfo:FirstName Mark CustomerInfo:LastName Hanson	AddressInfo:StreetAddress 999 Thames St AddressInfo:City Reading AddressInfo:County Berkshire AddressInfo:PostCode RG99 922
2	CustomerInfo:Title Ms CustomerInfo:FirstName Lisa CustomerInfo:LastName Andrews	AddressInfo:StreetAddress 888 W. Front St AddressInfo:City Boise AddressInfo:State ID AddressInfo:ZipCode 54321
3	CustomerInfo:Title Mr CustomerInfo:FirstName Walter CustomerInfo:LastName Harp	AddressInfo:StreetAddress 999 500th Ave AddressInfo:City Bellevue AddressInfo:State WA AddressInfo:ZipCode 12345

Row Key	Document
1001	<p>OrderDate: 06/06/2013  OrderItems: ProductID: 2010  Quantity: 2  Cost: 520</p> <p>ProductID: 4365  Quantity: 1  Cost: 18</p> <p>OrderTotal: 1058  Customer ID: 99  ShippingAddress: StreetAddress: 999 500th Ave  City: Bellevue  State: WA  ZipCode: 12345</p>
1002	<p>OrderDate: 07/07/2013  OrderItems: ProductID: 1285  Quantity: 1  Cost: 120</p> <p>OrderTotal: 120  Customer ID: 220  ShippingAddress: StreetAddress: 888 W. Front St  City: Boise  State: ID  ZipCode: 54321</p>

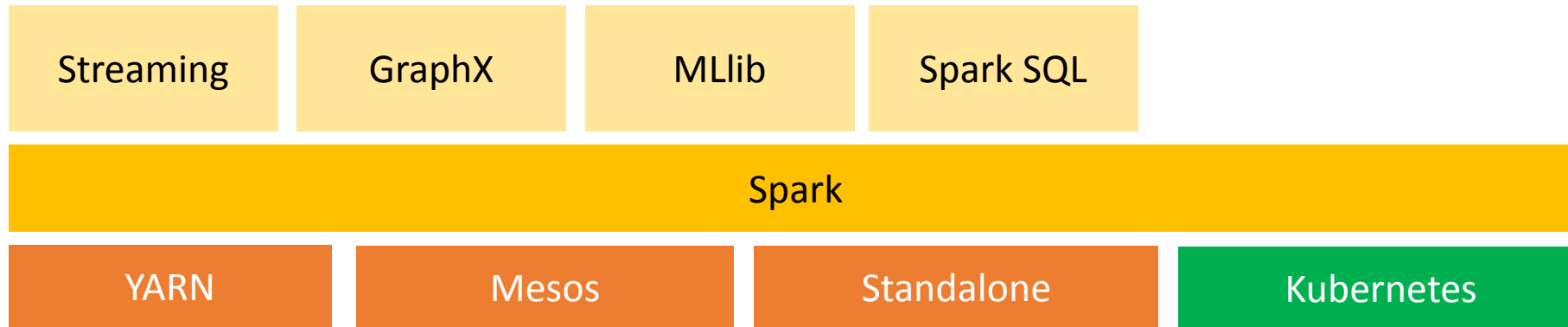


# Стек технологий Hadoop и Spark

# Стек Hadoop

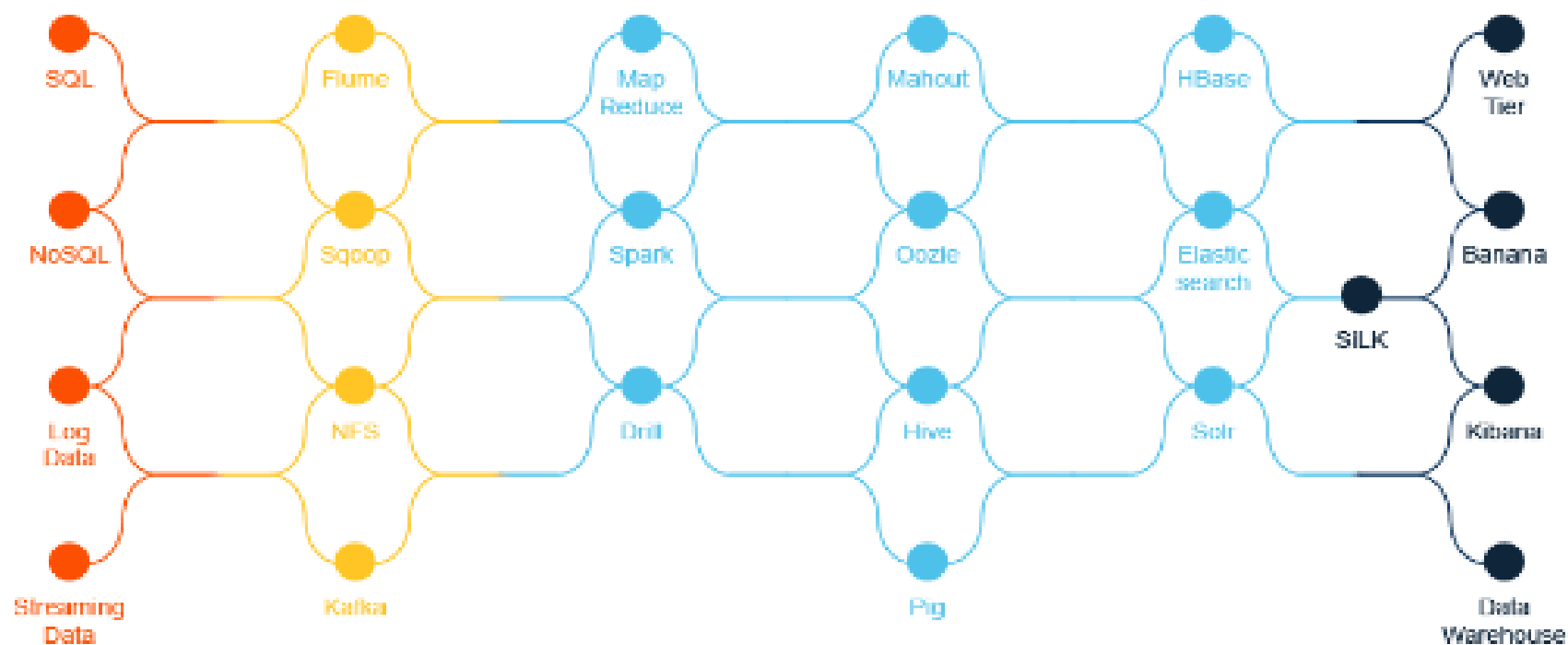


# Стек Spark



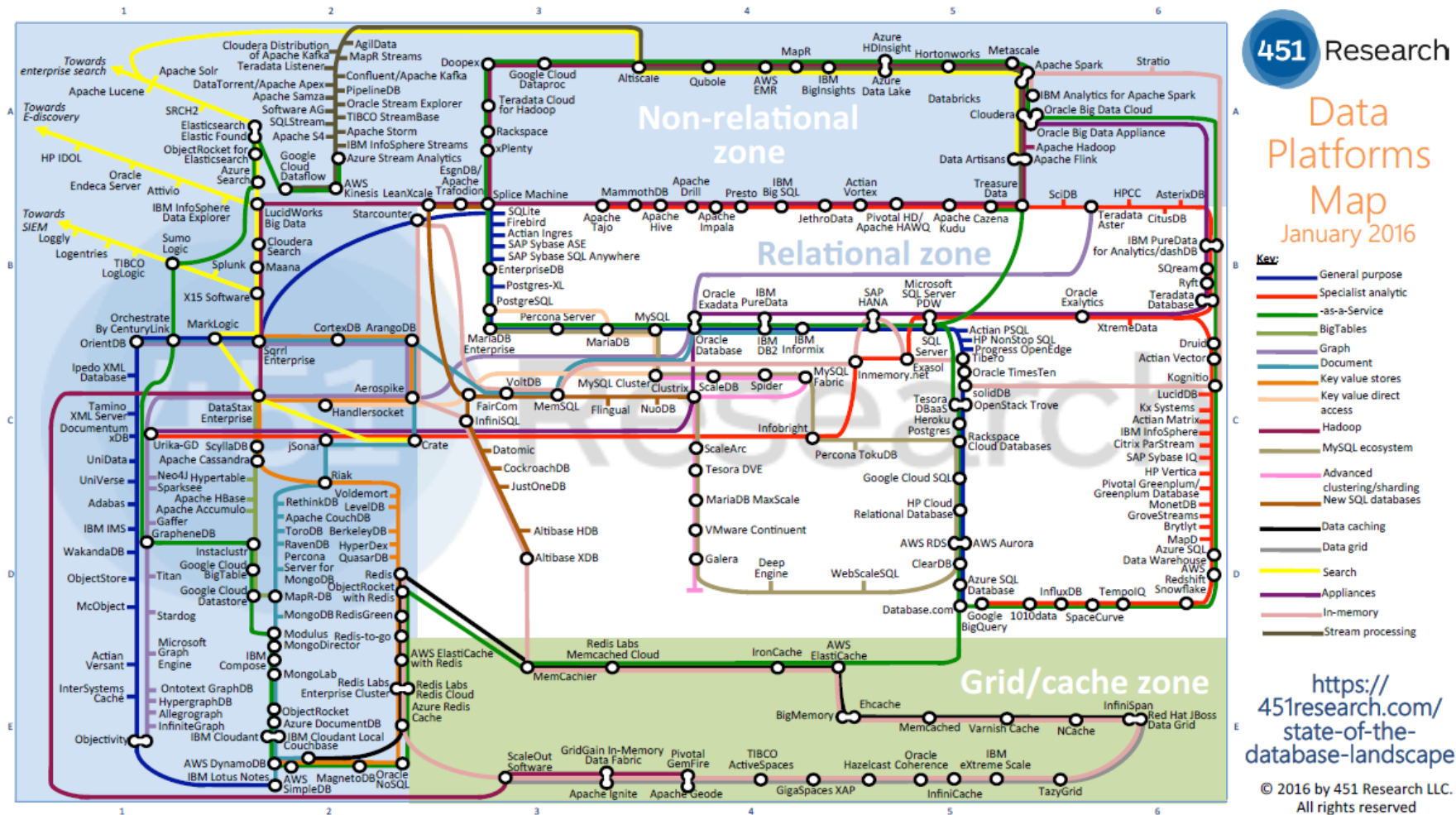


# Стек технологий



MapR

# Карта технологий



# Облачные ресурсы

# Облачные ресурсы

Magic Quadrant for Cloud Infrastructure as a Service



Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services





## Инфраструктуры анализа больших данных

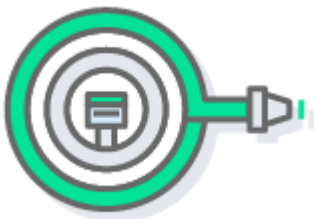


Hadoop и Spark  
**Amazon EMR**



Elasticsearch  
**Сервис Amazon Elasticsearch**

## Анализ больших данных в режиме реального времени



**Amazon Kinesis Firehose**



**Amazon Kinesis Streams**



**Amazon Kinesis Analytics**

## Хранилища и базы больших данных



Объектное хранилище  
**Amazon S3**



NoSQL  
**Amazon DynamoDB**



HBase в Amazon EMR



Реляционные базы данных  
**Amazon RDS**



Графовые базы данных  
**Amazon DynamoDB для БД Titan**