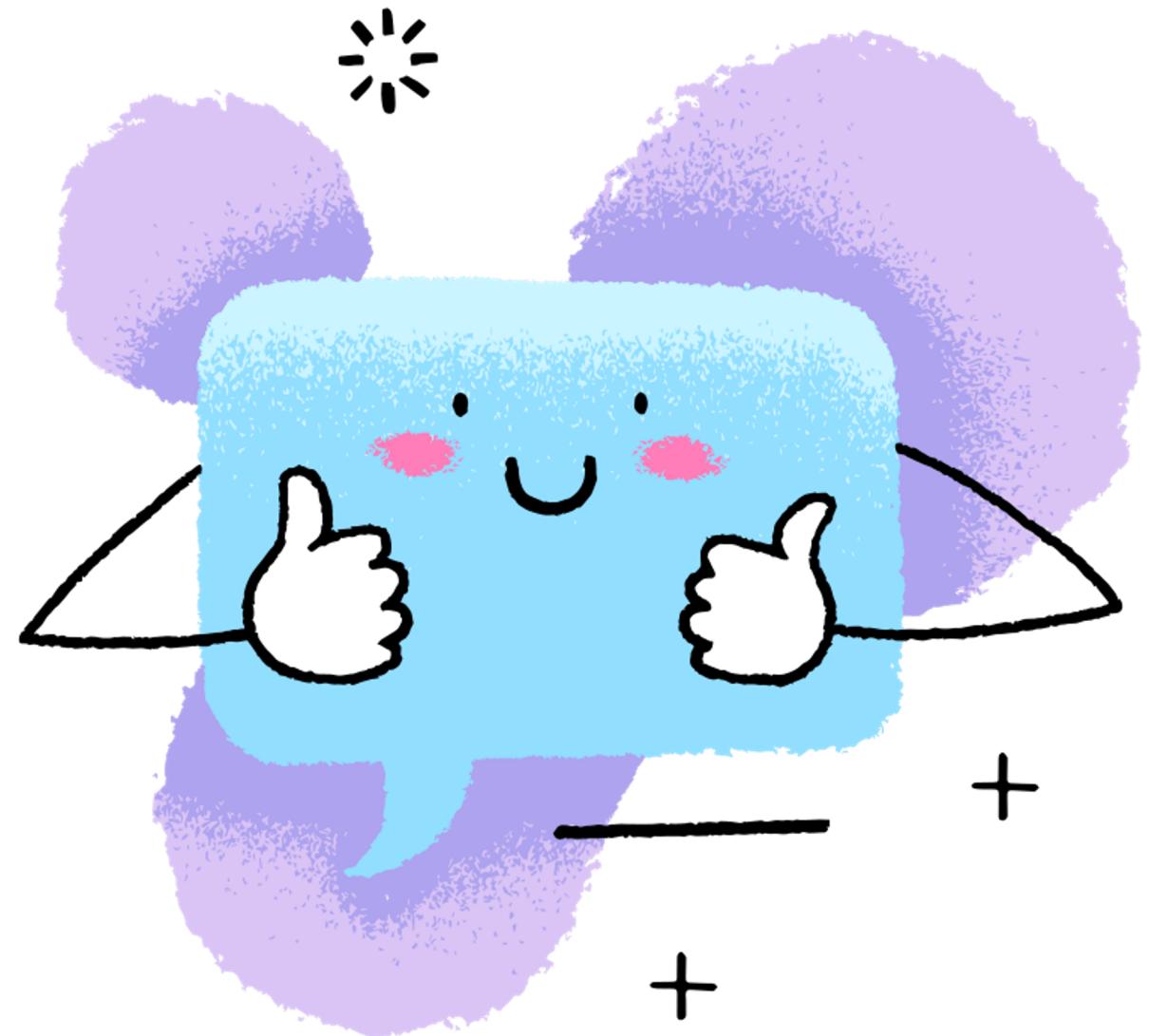


PySpark



Знакомство

Котельников Егор Александрович

sbermegamarket.ru: создание распределенного хранилища, реализация потоковой обработки данных из BigQuery, Google Analytics

Платформа ОФД: разработка хранилища, разработка аналитических продуктов на данных продаж не крупного бизнеса (50% всех чеков РФ)

МТС (инженер данных): создание витрин данных для DS, поиск закономерностей в геолокациях

Яндекс (стажер): стажировка разработчиком в Яндекс.Таланты

Образование: физфак МГУ, ШАД (один курс)



Что такое большие данные?

Ключевые факторы – это **объем, разнообразие и скорость обработки хранимых данных.**

Объём – не умещается на одной машине

Разнообразие – различные форматы, состав

Скорость обеспечивает актуальность аналитики

Вспомним: вертикальное vs горизонтальное масштабирование

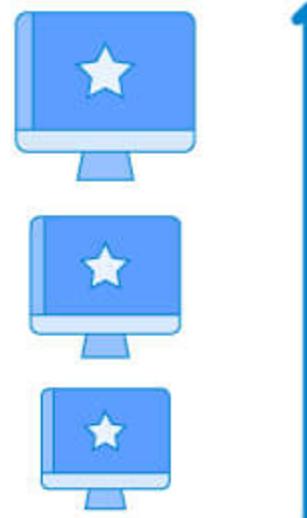
Горизонтальное масштабирование:

- дешевле
- легко масштабировать

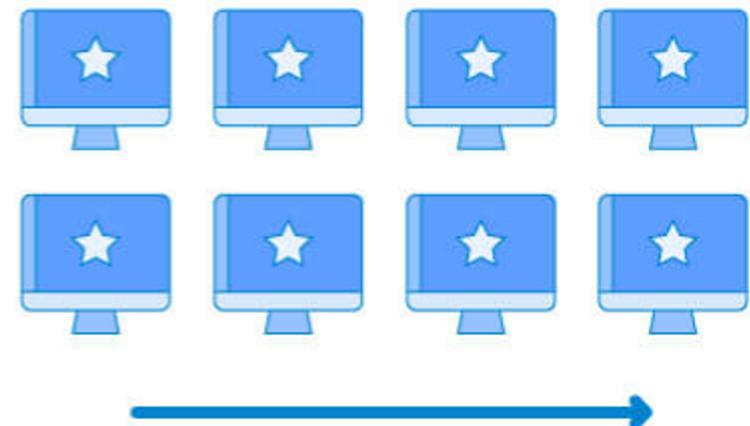
Особенности распределенной системы

1. Устойчивость к выбытию узлов
2. Код к данным, а не данные к коду

VERTICAL SCALING
Increase size of instance
(RAM, CPU etc.)

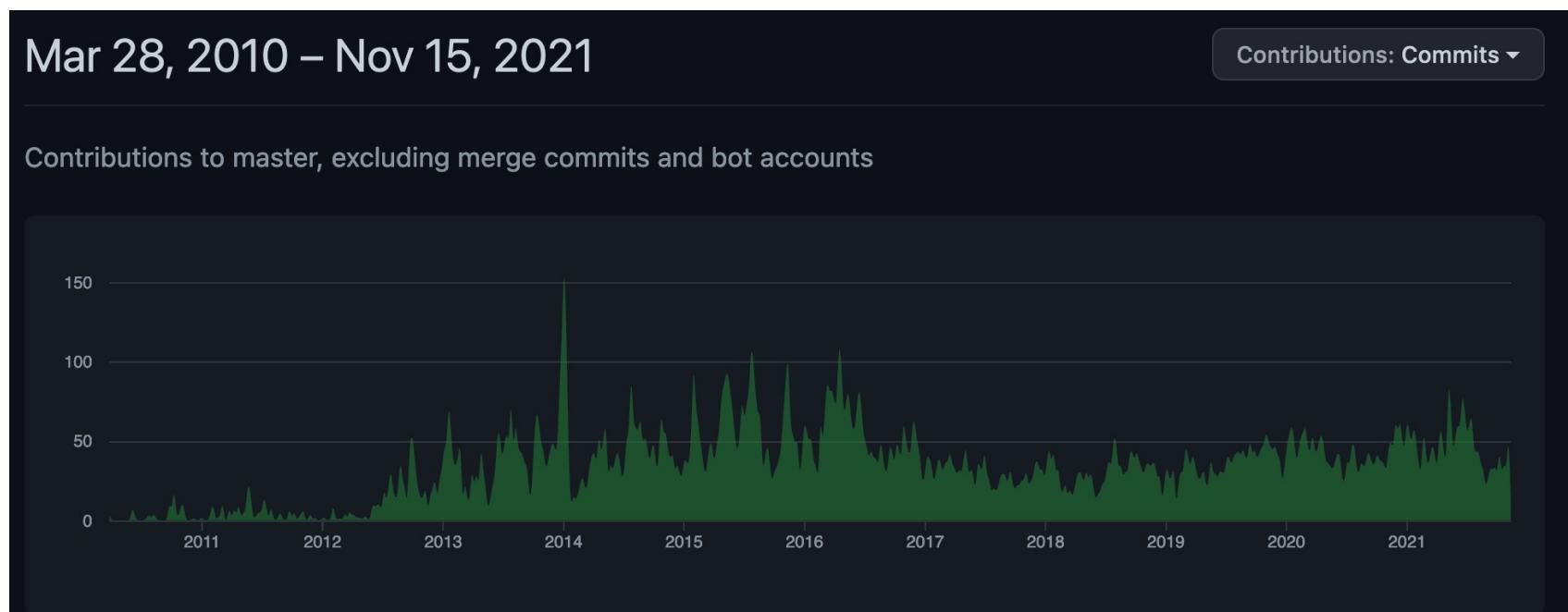


HORIZONTAL SCALING
(Add more instances)



Почему Spark

1. Spark есть везде: банки (масштабируемость), средний и маленький бизнес (бесплатно)
2. Молодая технология, следовательно, мало специалистов
3. Универсальный фреймворк: создание хранилища, запросы к данным, потоковая обработка, ML.
4. Активное развитие



Что такое Apache Spark?

Apache Spark – это универсальная платформа для быстрой обработки больших объёмов данных, разработанная на языке scala

- поддерживает распределенные вычисления (до сотен нод)
- in-memory, fault tolerant структуры данных
- API на языках scala, python, java, R, SQL
- открытый код

Spark -- Unified Engine for Big Data Processing -- объединил в себе возможности пакетной обработки, работы с графами, потоками и SQL

Spark SQL and
DataFrames +
Datasets

Spark Streaming
(Structured
Streaming)

Machine Learning
MLlib

Graph
Processing
Graph X

Spark Core and Spark SQL Engine

Scala

SQL

Python

Java

R

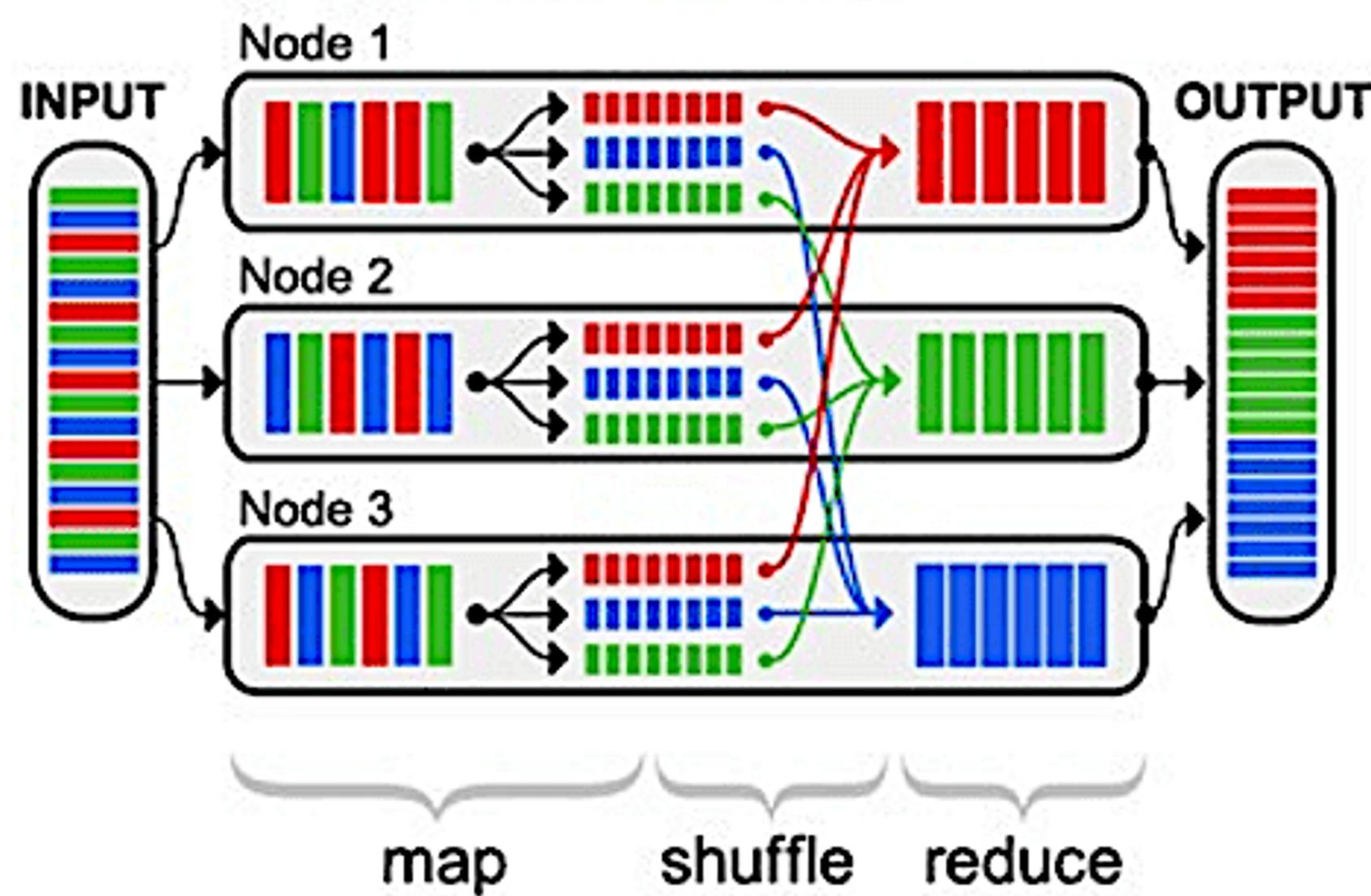
История возникновения

[Google File System \(2003\)](#): Scalable distributed file system for large distributed data-intensive applications

[Bigtable](#): scalable storage of structured data across GFS

[MapReduce \(MR\)](#) is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

Что такое map-reduce?



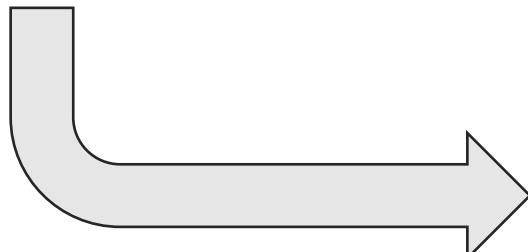
Найти строку максимальной длины

решение с помощью цикла:

```
def find_longest_string(list_of_strings):
    longest_string = None
    longest_string_len = 0

    for s in list_of_strings:
        if len(s) > longest_string_len:
            longest_string_len = len(s)
            longest_string = s

    return longest_string
```



```
mapper = len

def reducer(p, c):
    if p[1] > c[1]:
        return p
    return c

from multiprocessing import Pool

pool = Pool(8)

data_chunks = chunkify(large_list_of_strings, number_of_chunks=8)

#step 1:
mapped = pool.map(mapper, data_chunks)

#step 2:
reduced = reduce(reducer, mapped)

print(reduced)
```

MapReduce

Плюсы

1. Функциональный подход – легко тестировать
2. Автоматическая масштабируемость.

Минусы

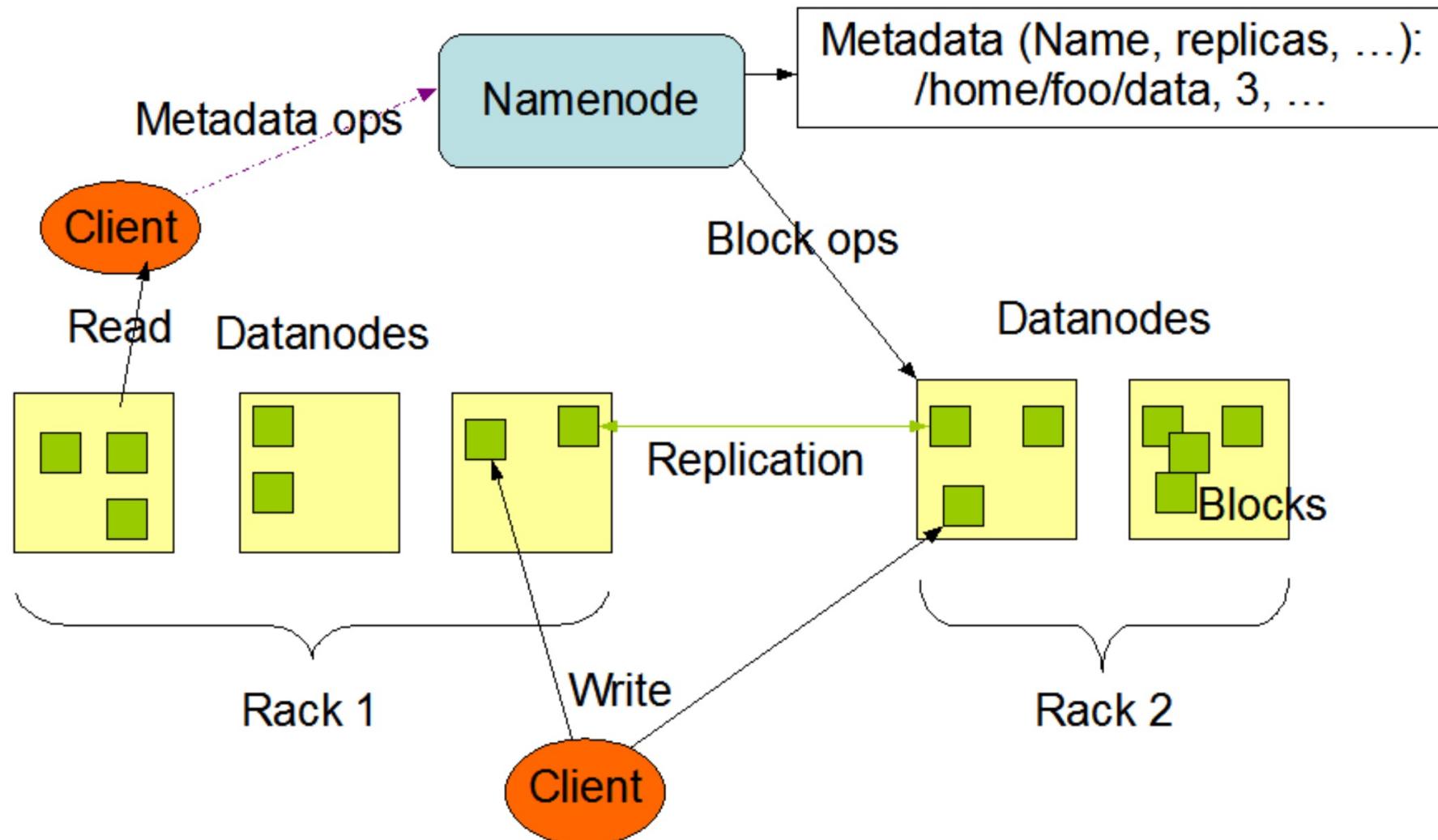
1. Нужно писать все руками (трансформации данных, джобы)
2. Пайплайн обработки тоже создается руками
3. Трудности с кешированием, переиспользованием
4. Записи промежуточных шагов на диск (медленно)

Логика сложнее джойна нескольких таблиц - дни разработки

Hadoop modules

<p>Others (For Data Processing)</p>	<p>MapReduce (For Data Processing)</p>
<p>YARN (Resource Management For Cluster)</p>	
<p>HDFS (A Reliable & Redundant Storage)</p>	

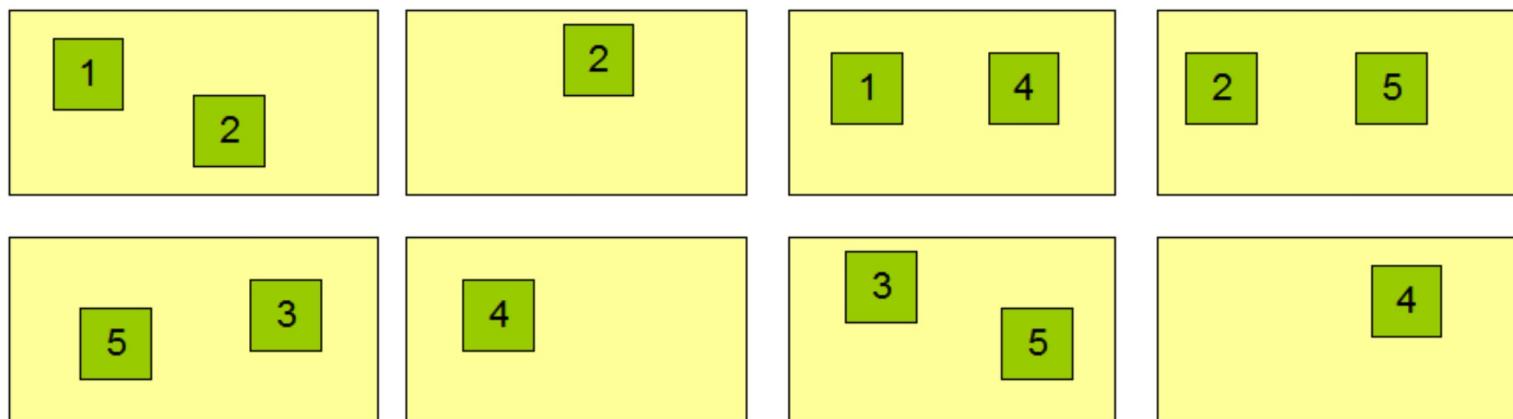
HDFS architecture



Data Replication

```
Namenode (Filename, numReplicas, block-ids, ...)  
/users/sameerp/data/part-0, r:2, {1,3}, ...  
/users/sameerp/data/part-1, r:3, {2,4,5}, ...
```

Datanodes



Недостаток Hadoop MapReduce:
промежуточные результаты последовательных операций
записываются на диск

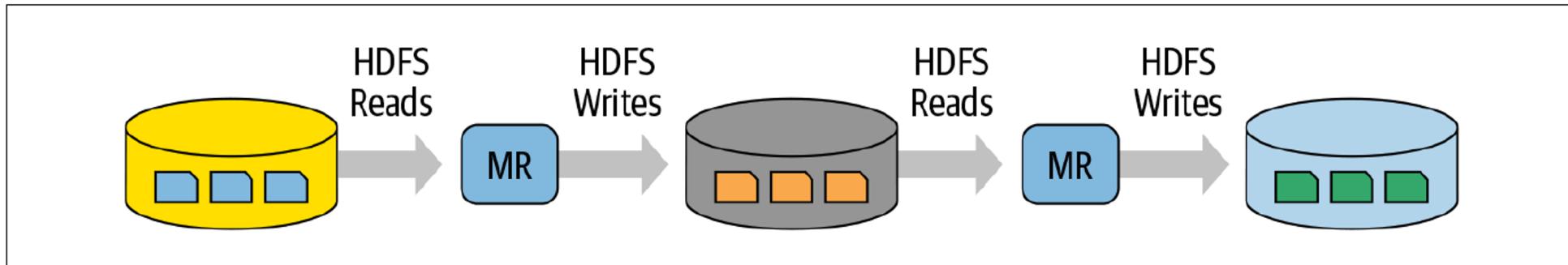
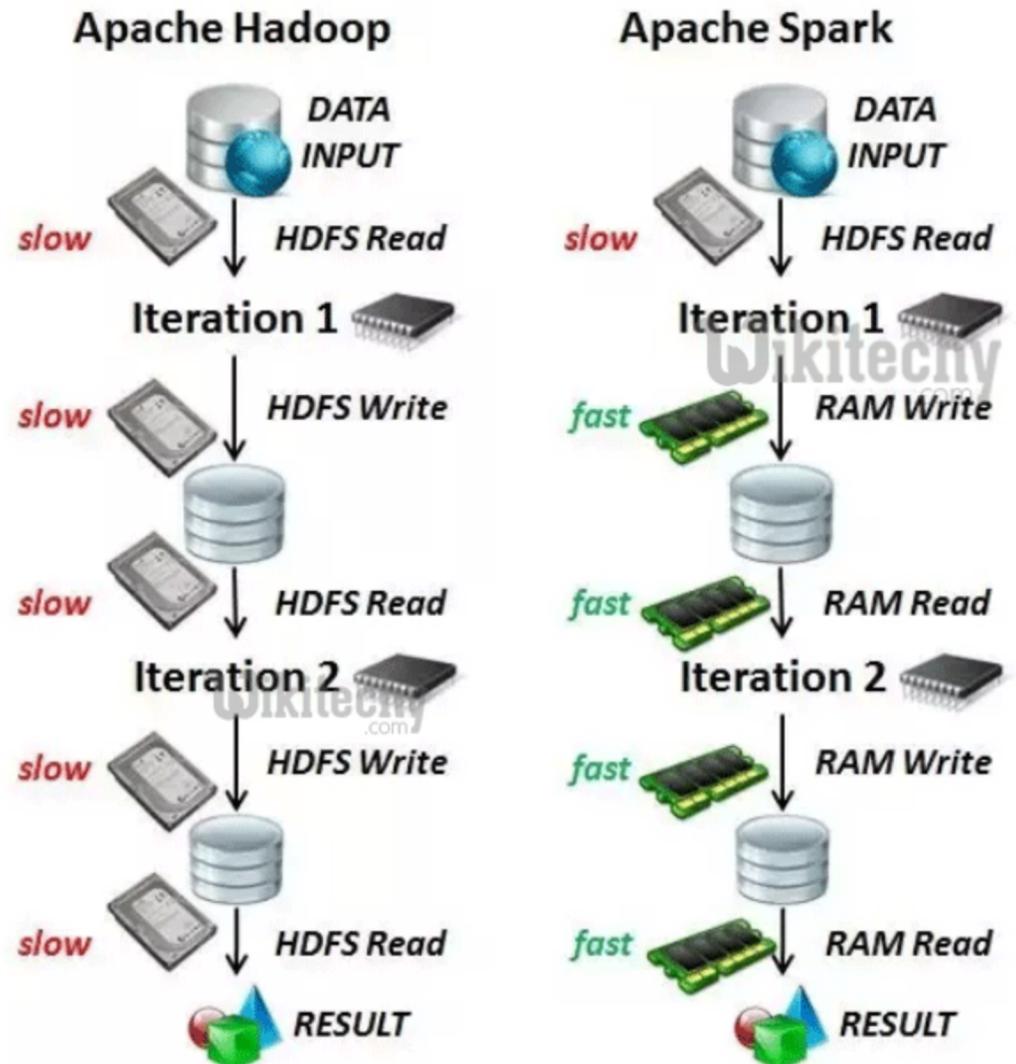


Figure 1-1. Intermittent iteration of reads and writes between map and reduce computations

Apache Hive (Tez), Apache Impala

Spark (2009) vs Hadoop MapReduce:

1. in-memory storage for intermediate results between iterative and interactive map and reduce computations
2. offer easy and composable APIs in multiple languages as a programming model

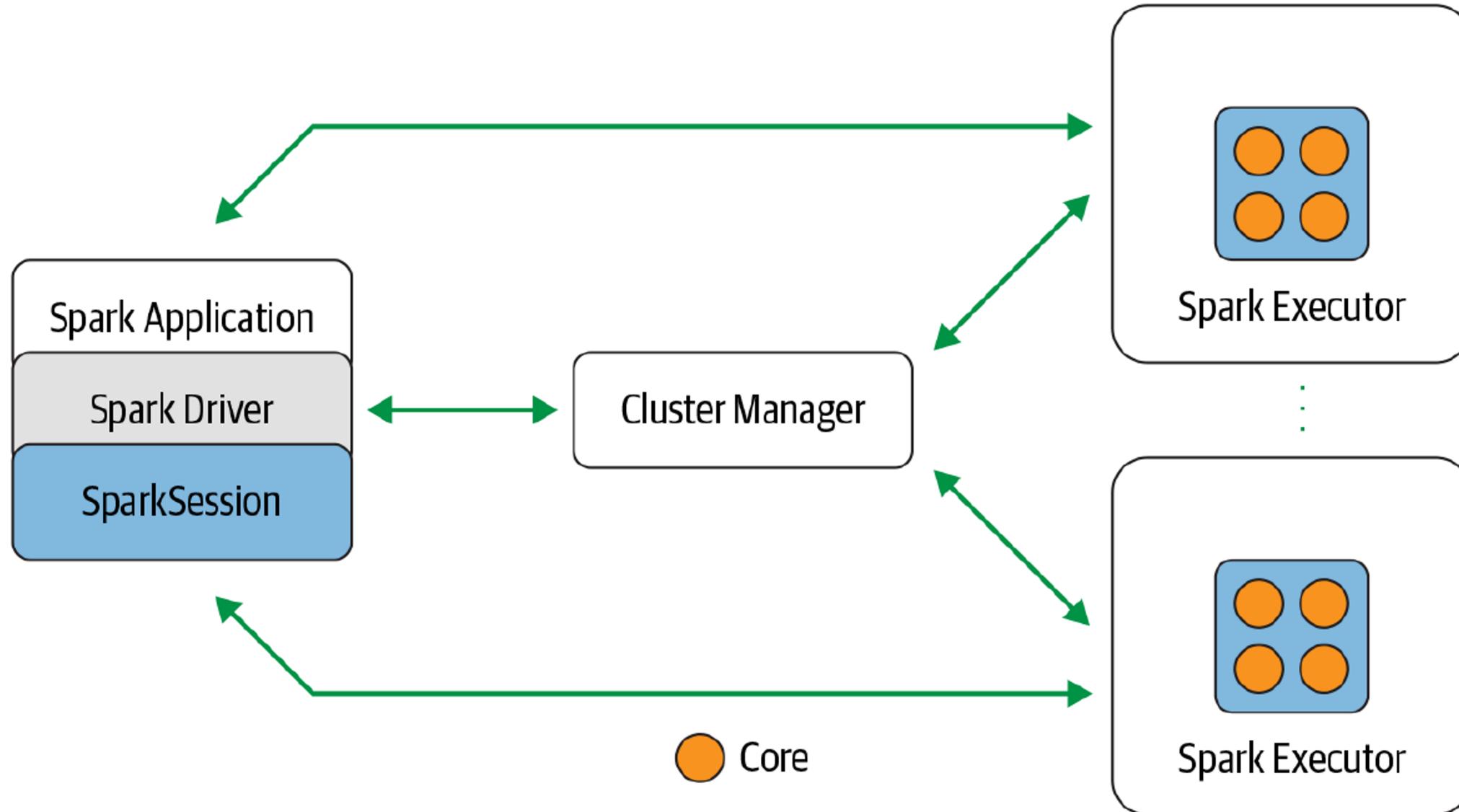


Причины популярности Spark:

Со времен появления Hadoop MapReduce железо стало лучше по числу ядер CPU и памяти. Вычисления не выполняются последовательно, а оптимизируются после формирования полного запроса.

Потребность в разделении хранилища от вычислений: Spark может принимать данные из любых источников и обрабатывать их в памяти, пакетами (jdbc for SQL) или потоком (Kafka)

Архитектура приложения Spark



Cluster manager

Распределяет ресурсы между spark приложениями

standalone cluster manager

FIFO исполнение приложений

Apache Hadoop YARN

стандартное решение распределяет, освобождает ресурсы между различными приложениями

Часть Hadoop

Apache Mesos

YARN-like, but большая изолированность процессов

поддерживает не-hadoop приложения

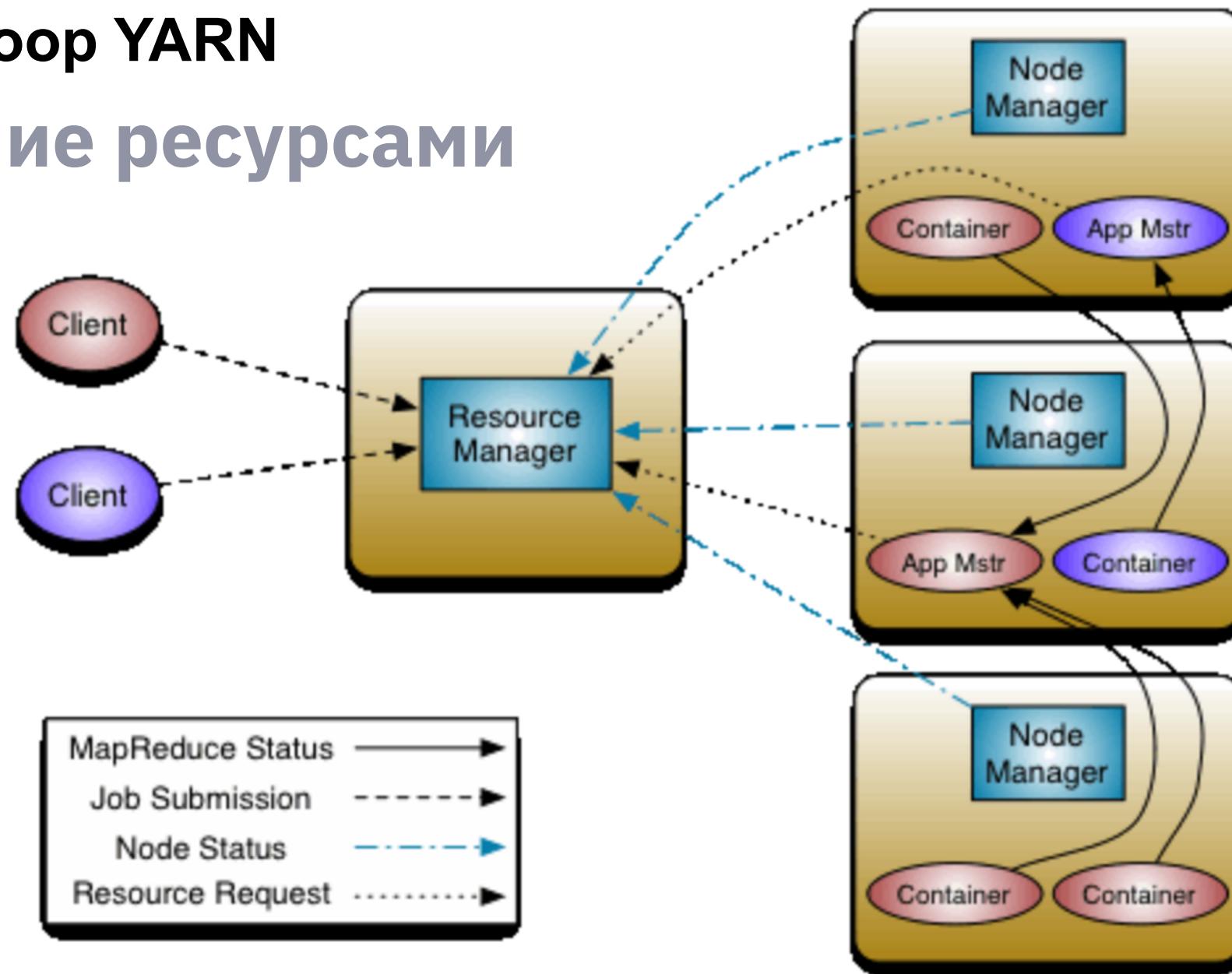
Kubernetes

запуск в контейнерах = абсолютная изолированность

существенно сложнее в поддержке

Apache Hadoop YARN

Управление ресурсами



Spark driver

Может
находиться вне
кластера

SparkSession

Точка входа в приложение

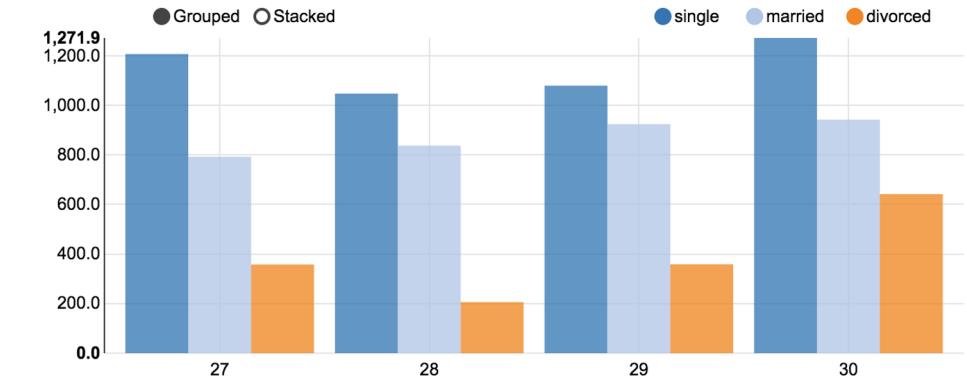
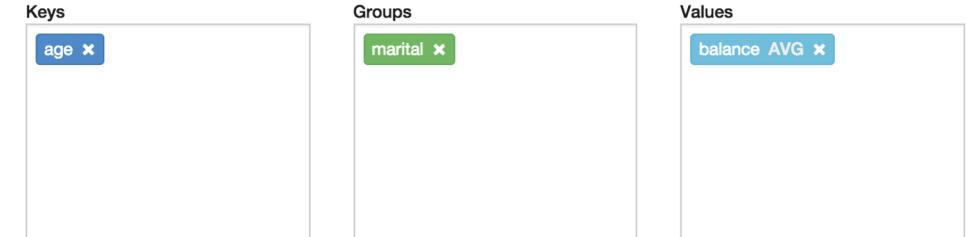
С его помощью можно:

- Управлять параметрами spark приложения
- читать и записывать данные в spark DataFrame
- исполнять SQL

```
1 from pyspark.sql import SparkSession  
2  
3 spark = SparkSession \  
4     .builder \  
5     .appName("Python Spark SQL basic example") \  
6     .config("spark.some.config.option", "some-value") \  
7     .getOrCreate()
```

Zppelin

- Встроенная визуализация
- Динамические формы
- Каждый параграф в своем интерпретере
- Плагины



```
%md Hello ${name=sun}
```

name
moon

Hello moon

FINISHED ▶ ✎ 📄 ⚙

%spark

FINISHED ▶ ✎ 📄 ⚙

```
println("Today is "+z.select("day", Seq(("Monday","1"),  
("Tuesday","2"),  
("Wednesday","3"),  
("Thursday","4"),  
("Friday","5"),  
("Saturday","6"),  
("Sunday","7"))))
```

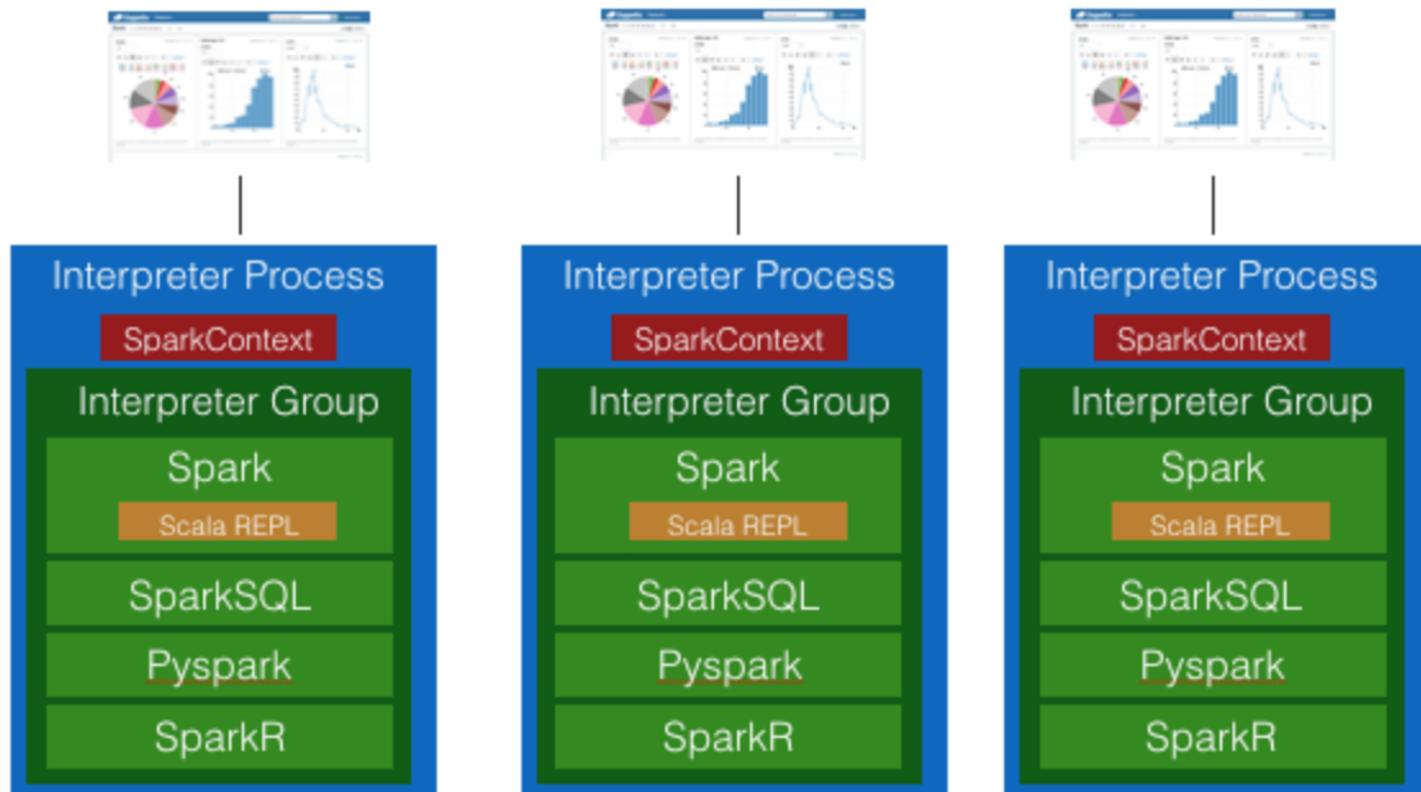
day

5

Today is Friday

Zeppelin Interpreter

In *Isolated* mode, each Note has its own SparkContext and Scala REPL.



Практика

Работа с DataFrame

тетрадки

Lecture 1

Spark SQL (PySpark)

Parquet vs CSV

Меньше размер, быстрее чтение

The following table compares the savings as well as the speedup obtained by converting data into Parquet from CSV.

Dataset	Size on Amazon S3	Query Run Time	Data Scanned	Cost
Data stored as CSV files	1 TB	236 seconds	1.15 TB	\$5.75
Data stored in Apache Parquet Format	130 GB	6.78 seconds	2.51 GB	\$0.01
Savings	87% less when using Parquet	34x faster	99% less data scanned	99.7% savings

Локальный запуск

<https://docs.google.com/document/d/1r1CVIwvcLpaSNrBHwh0L6MPHnZgzTAMsqYMelqyYtOo>

Полезные ресурсы

<https://sparkbyexamples.com>

[parquet doc](#)

[A Neanderthal's Guide to Apache Spark in Python](#)

[Matei Zaharia. Презентация от автора Spark](#)

<https://www.kaggle.com/mkechinov/ecommerce-behavior-data-from-multi-category-store/version/8>

Спасибо!
Каждый день
вы становитесь
лучше :)

