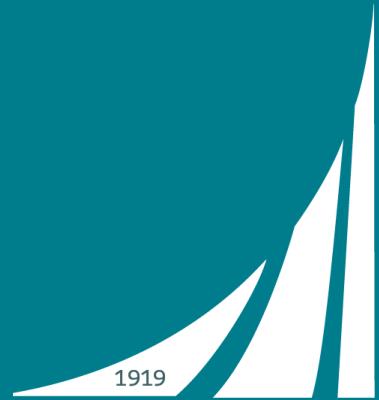


Большие данные. Инструменты работы с большими данными



ФИНАНСОВЫЙ
УНИВЕРСИТЕТ
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ

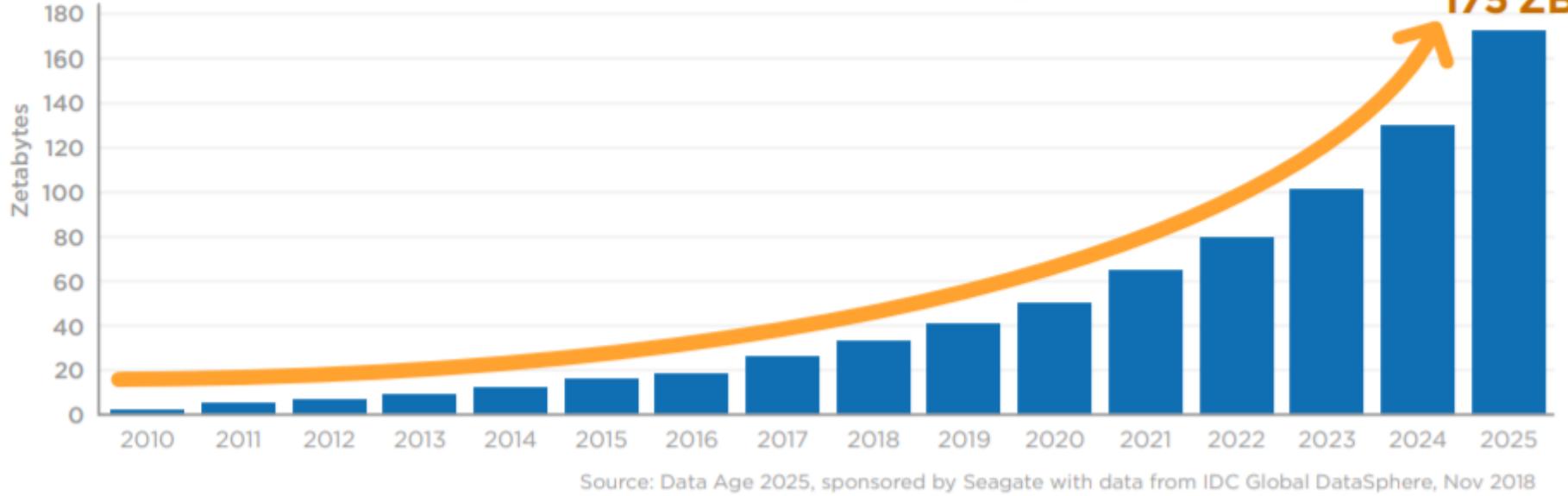


Большие данные (англ. *big data*) — обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами, появившимися в конце 2000-х годов и альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

В широком смысле о «больших данных» говорят как о социально-экономическом феномене, связанном с появлением технологических возможностей анализировать огромные массивы данных, в некоторых проблемных областях — весь мировой объём данных, и вытекающих из этого трансформационных последствий



Annual Size of the Global Datasphere



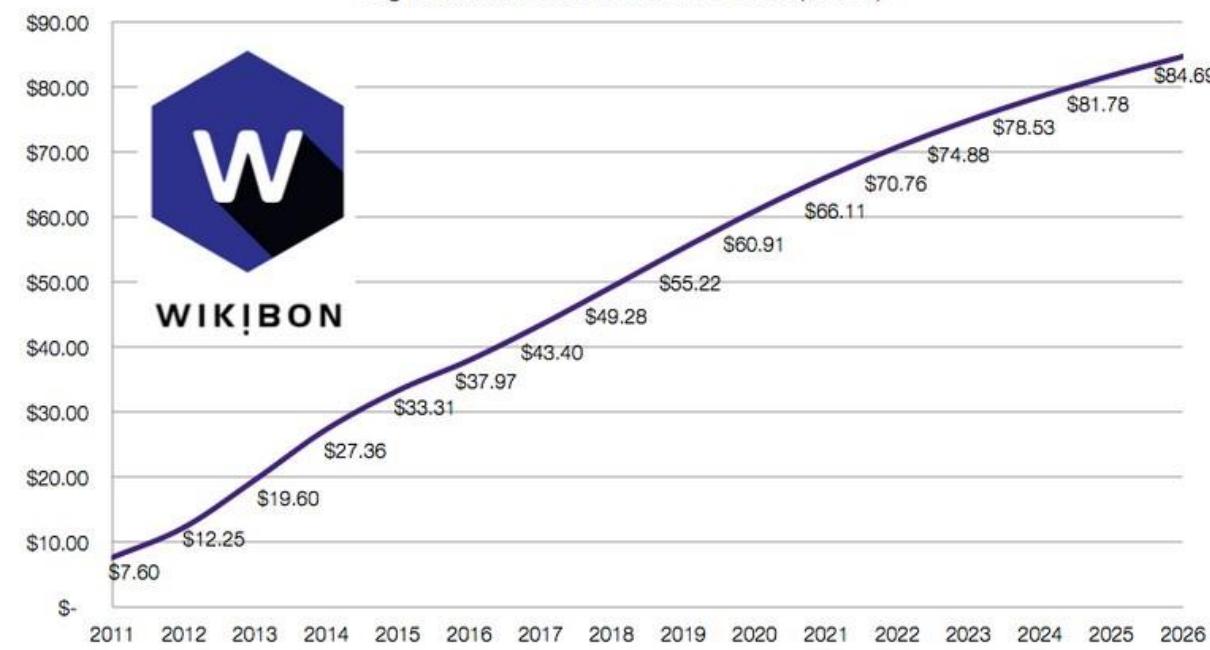
- По данным компании IBS, к 2003 году мир накопил 5 эксабайтов данных (1 ЭБ = 1 млрд гигабайтов). К 2008 году этот объем вырос до 0,18 зеттабайта (1 ЗБ = 1024 эксабайта), к 2011 году – до 1,76 зеттабайта, к 2013 году – до 4,4 зеттабайта. В мае 2015 года глобальное количество данных превысило 6,5 зеттабайта.
- В 2020 году, человечество сформировало 40-44 зеттабайтов информации. А к 2025 году этот объем вырастет в 10 раз, говорится в докладе аналитиков компании IDC. В докладе отмечается, что большую часть данных генерировать будут сами предприятия, а не обычные потребители.
- К 2025 году общий объем данных во всем мире составит 163 зеттабайта (ЗБ), прогнозирует аналитическая компания IDC. Для сравнения: в 2016 году на планете было в 10 раз меньше данных – 16 ЗБ, а 2006 году – всего 0,16 ЗБ.





BIG DATA & ANALYTICS

Big Data Market Forecast, 2011-2026 (\$US B)



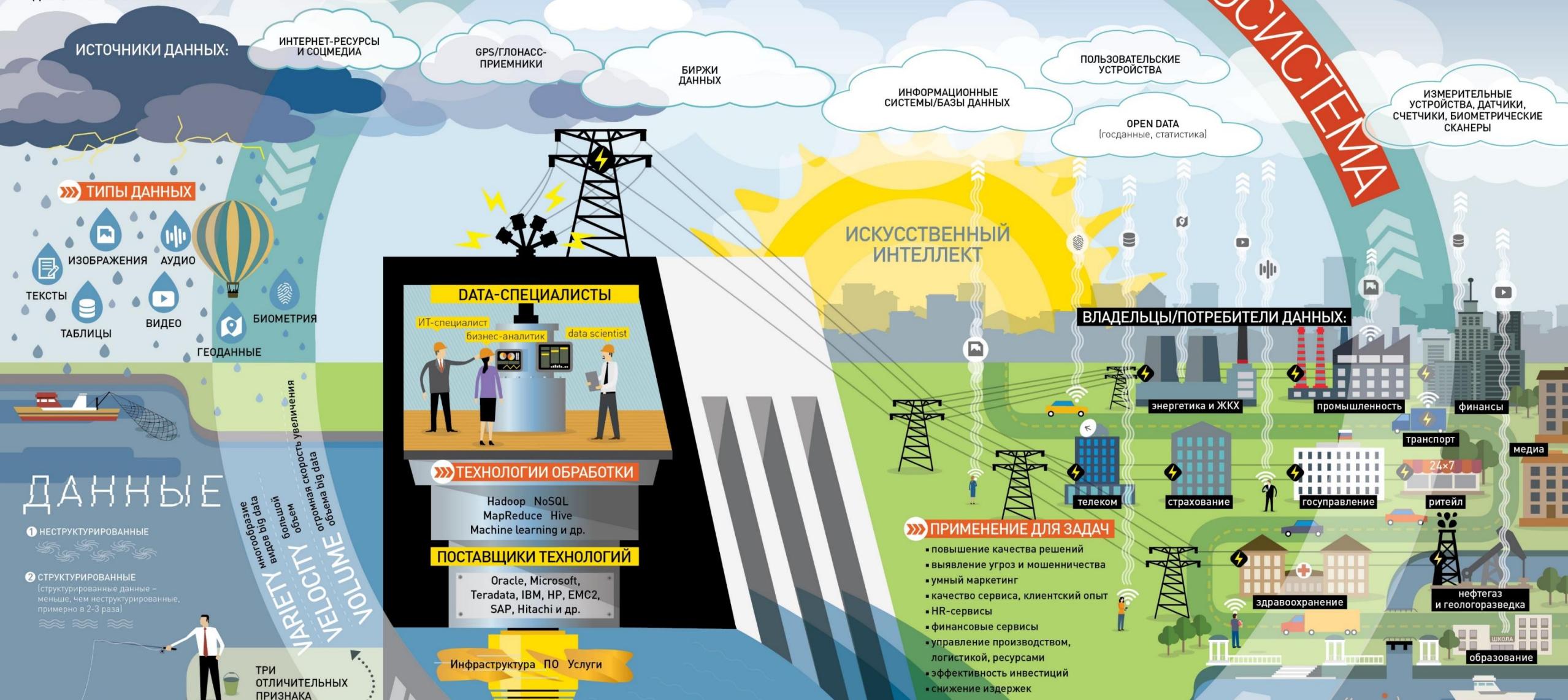
- Сейчас более 3,2 миллиарда интернет-пользователей, и число активных сотовые телефоны превысили 7,6 миллиарда. В настоящее время сотовых телефонов больше, чем на планете людей (7,4 миллиарда).
- Twitter обрабатывает 7 ТБ данных каждый день, и 600 ТБ данных обрабатываются Facebook каждый день.
- более 500 миллионов твитов, 90 миллиардов электронных писем, 65 миллионов сообщений WhatsApp - все за один день.
- 4 петабайта данных генерируются только на Facebook за 24 часа.
- Интересно, что около 80% этих данных неструктурированы.
- С таким огромным объемом данных компаниям требуется быстрое, надежное и глубокое понимание данных.
- Поэтому решения для больших данных на основе Hadoop и другого аналитического программного обеспечения становятся все более актуальным.



ПОСТОЯННО РАСТУЩИЙ ОБЪЕМ ДАННЫХ – ВАЖНЫЙ АКТИВ ОРГАНИЗАЦИИ, КОТОРЫЙ МОЖЕТ ПРИНЕСТИ ЕЙ ДОПОЛНИТЕЛЬНУЮ ЦЕННОСТЬ. ТЕХНОЛОГИИ BIG DATA – ЭТО НАБОР ИНСТРУМЕНТОВ И МЕТОДОВ, КОТОРЫЕ ПОЗВОЛЯЮТ ОБРАБАТЫВАТЬ И АНАЛИЗИРОВАТЬ ОГРОМНЫЕ МАССИВЫ ДАННЫХ, ИЗ РАЗНЫХ ИСТОЧНИКОВ, В РЕАЛЬНОМ ВРЕМЕНИ, ИЗВЛЕКАЯ ИЗ НИХ ПОЛЬЗУ. СФЕР ПРИМЕНЕНИЯ BIG DATA СТАНОВИТСЯ ВСЕ БОЛЬШЕ: СЕГОДНЯ ЭТИ ТЕХНОЛОГИИ ВОСТРЕБОВАНЫ ПОЧТИ ВО ВСХ ОТРАСЛЯХ, ПОЗВОЛЯЯ ИСПОЛЬЗОВАТЬ МАКСИМАЛЬНОЕ КОЛИЧЕСТВО ИНФОРМАЦИИ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ.

BIG DATA

«БОЛЬШИЕ ДАННЫЕ ОБЪЕДИНЯЮТ ТЕХНИКИ И ТЕХНОЛОГИИ, КОТОРЫЕ ИЗВЛЕКАЮТ СМЫСЛ ИЗ ДАННЫХ НА ЭКСТРЕМАЛЬНОМ ПРЕДЕЛЕ ПРАКТИЧНОСТИ»
FORRESTER



Каждые 2 года объем данных увеличивается
почти в два раза.

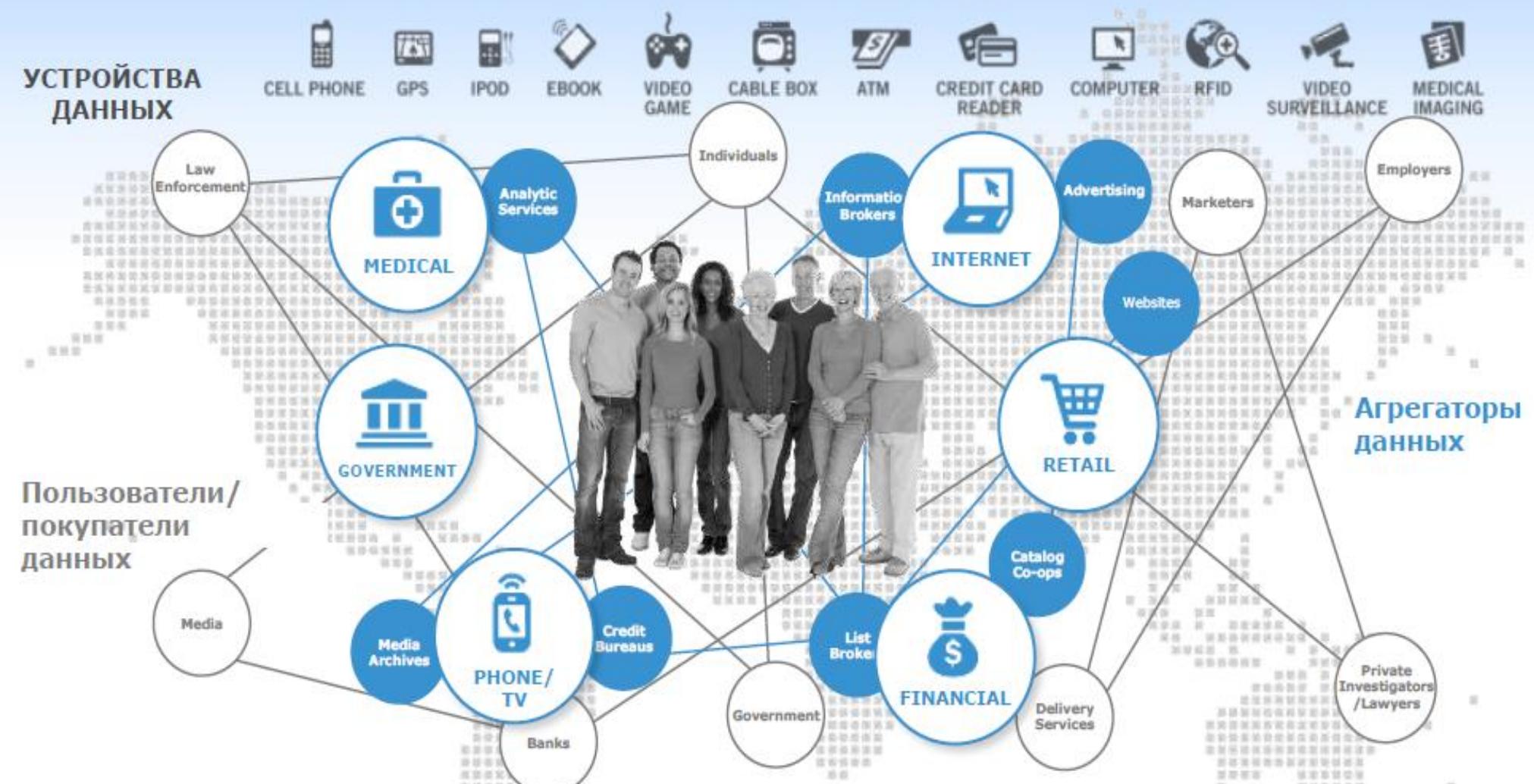
90%

2.5+

всех данных мира были созданы за
последние два года.

экрабайт данных создается
ежедневно

Данные стали разными

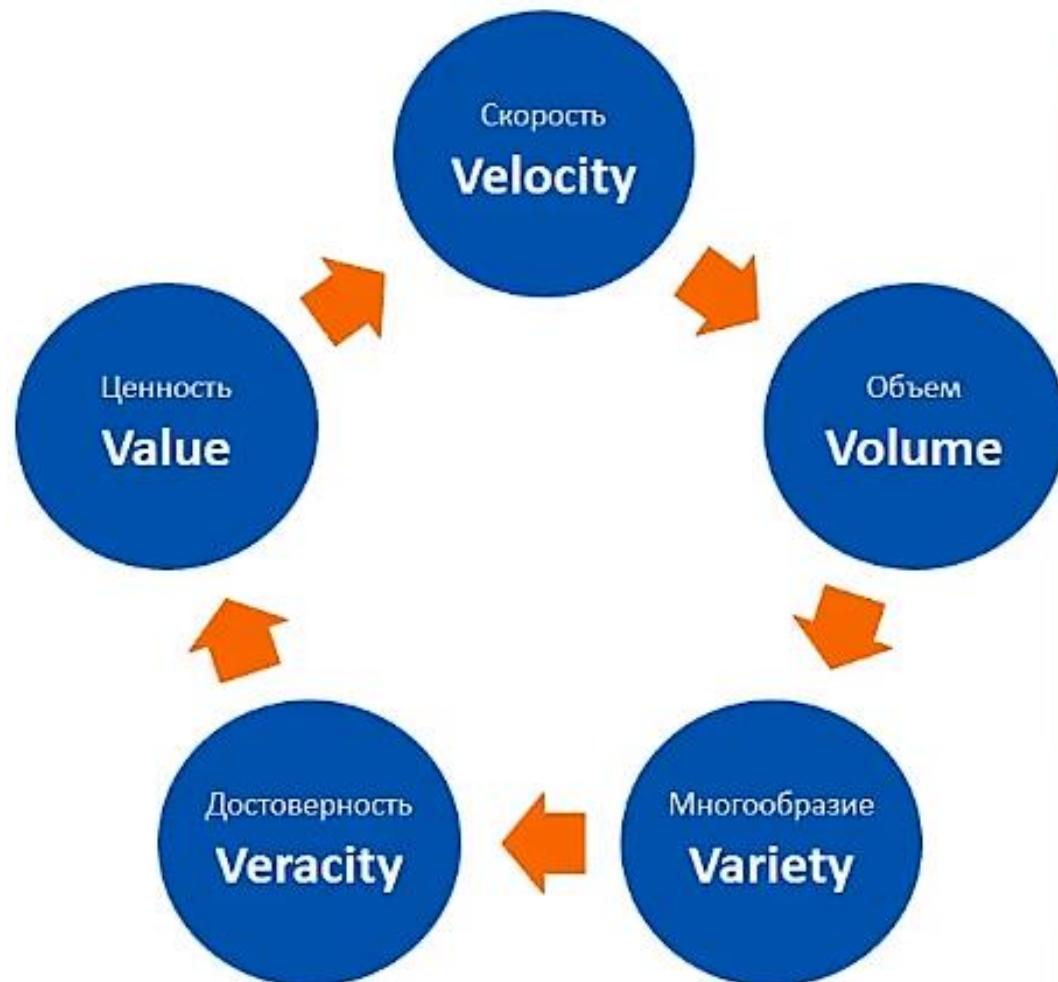


Откуда берутся большие данные:

- Социальные сети и их данные
- Данные от измерительных устройств
- Данные от RFID
- Журналы доступа пользователей веб-сайтов
- Сенсорные сети
- Тексты и документы из Интернета
- Научные данные (астрономия, геном человека, исследования атмосферы, биохимия, биология)
- Данные министерства обороны
- Медицинские наблюдения
- Фото- и видео-архивы
- Данные электронной коммерции



Характеристики больших данных



Характеристика	Традиционная база данных	База Больших Данных
Объем информации	От гигабайт до терабайт	От петабайт до эксабайт
Способ хранения	Централизованный	Децентрализованный
Структурированность данных	Структурирована	Полуструктурирована или неструктурирована
Модель хранения и обработки данных	Вертикальная модель	Горизонтальная модель
Взаимосвязь данных	Сильная	Слабая

"Big data is the term for a collection of data sets so **large** and **complex** that it becomes **difficult** to process using on-hand database management tools or traditional data processing applications"



1. **Объем (Volume)**. Базовая характеристика
 2. Скорость обновления (Velocity). Чем быстрее данные поступают, тем быстрее их надо обрабатывать
 3. Разнообразие (Variety). Множество форм и форматов данных усложняет их обработку.
 4. Достоверность (Veracity). Необходимо проверять, насколько можно доверять данным
 5. Ценность (Value). Сколько пользы данные могут принести для организации и общества.
- Различные сочетания этих характеристик порождает множество задач хранения и обработки больших данных.**

Зачем нужны большие данные:

- Более точное определение групп потенциальных клиентов
- Сегментация клиентской базы
- Более точная оценка бизнеса
- Выявление возможностей рынка
- Автоматизация решений в процессах реального времени
- Выяснение поведения клиентов
- Выявление угроз
- Использование и возврат инвестиций в сбор данных
- Вычисление рисков
- Отслеживание рыночных тенденций
- Понимание изменений в бизнесе
- Улучшение планирования и прогнозов
- Выявление ключевой причины издержек
- Понимание поведения клиентов по анализу посещения сайта
- Улучшение результатов производства
- Другое



Инструменты работы с большими данными

Облачные провайдеры



Фреймворки для обработки



Хранилища данных



Языки программирования



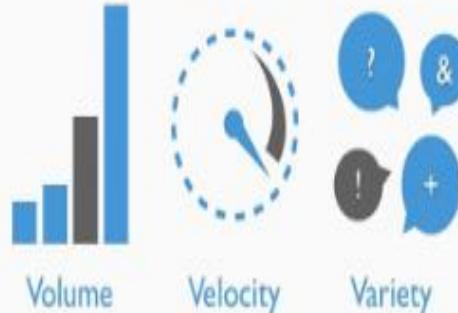
Компоненты решения Big Data

Современное Big Data решение состоит из нескольких блоков, требующих совместной работы команд с разными компетенциями и интеграции набора Open-source и проприетарных программных компонентов:

1. Техническое решение по сбору, хранению и обработке больших объемов данных, обозначенное на схеме как Big Data Tools. Такое решение, как правило, строится на основе стека Hadoop, так как он представляет хороший баланс между стоимостью, надёжностью и функциональностью.
2. Продвинутый анализ данных с использованием методов науки о данных (Data Science) и алгоритмов машинного обучения
3. Визуализация больших данных, а также создание интерактивных отчетов для руководства компании, сотрудников и клиентов (Business Intelligence). При этом используемая аналитическая платформа должна быть совместима со стеком Hadoop

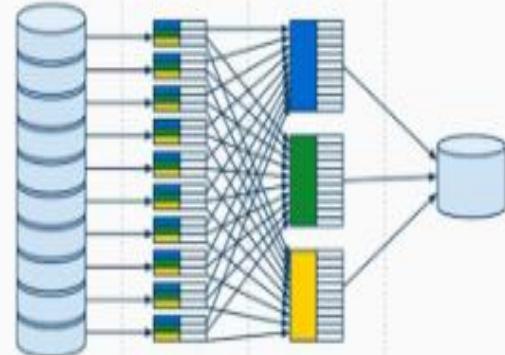
Аналитика больших данных: компоненты решения

BIG DATA



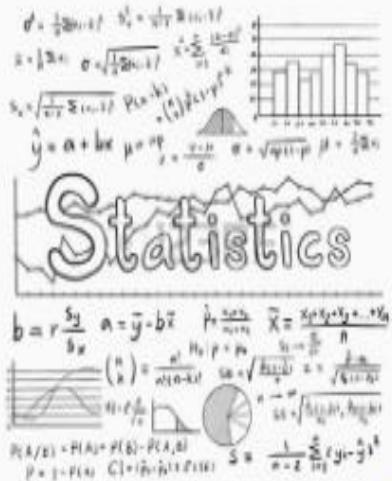
Данные, которые слишком велики для обычных средств хранения и слишком сложны для традиционных средств аналитики

BIG DATA TOOLS



Инструменты для сбора, хранения и анализа данных, распределенных по кластерам серверов

DATA SCIENCE



Математические и алгоритмические методы, оптимизированные для эффективного выявления сложных закономерностей

BUSINESS INTELLIGENCE



Визуализация и публикация данных для простого использования конечными бизнес-пользователями

Apache Hadoop

Каждые 2 года объем данных увеличивается
почти в два раза.

90%

всех данных мира были созданы за
последние два года.

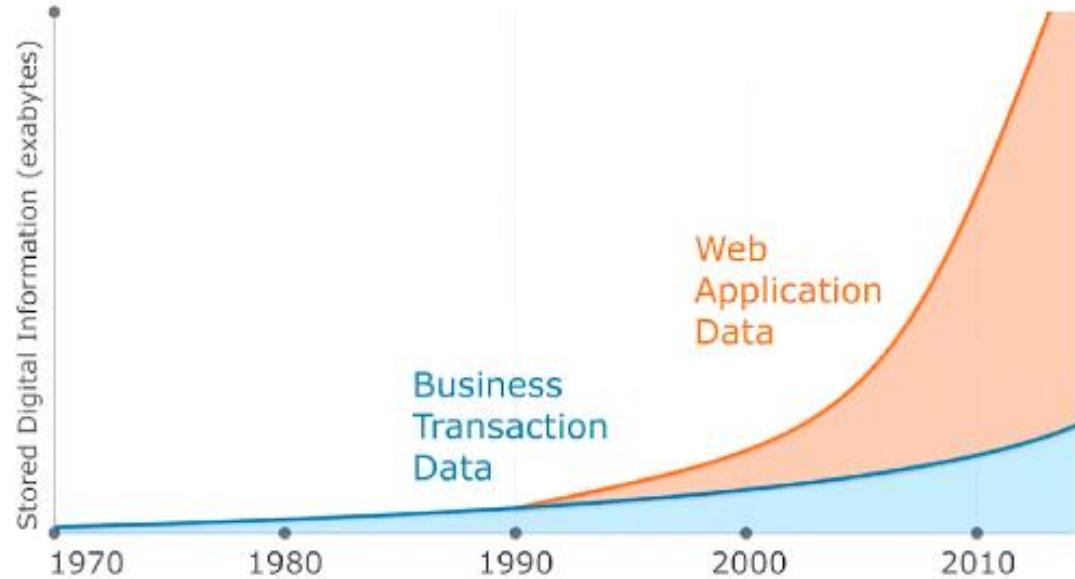
2.5+

экрабайт данных создается
ежедневно

Большие данные - это данные, которые слишком велики для обработки традиционными методами. Это ставит новые задачи, когда речь идет о хранении, обработке, извлечении и анализе больших данных.

Apache Hadoop - одна из самых популярных технологий, которая создает основу для анализа больших данных. Hadoop – проект фонда Apache с открытым исходным кодом. Это фреймворк, написанный на Java, разработанный Дугом Каттингом, который назвал его в честь игрушечного слона своего сына.





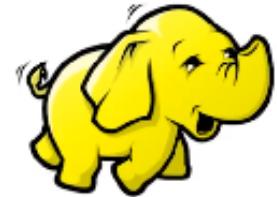
- Проект "The Apache™ Hadoop™" разрабатывает open-source ПО для отказоустойчивых, масштабируемых и распределенных вычислений
- Особенности:
 - Работает с BigData на обычных серверах
 - Сильное open-source сообщество
 - Много различных продуктов и средств используют Hadoop

Кто использует Hadoop



История Hadoop

- Начинался как подпроект в Apache Nutch
 - Nutch – это открытый Web Search Engine
 - OpenSource альтернатива Google
 - Начинал его Doug Cutting
- В 2004 году Google публикует статьи про GFS и MapReduce
- Doug Cutting и команда Nutch реализовала свой фреймворк на основе этих статей
- В 2006 Yahoo! Нанимает Doug Cutting для работы над Hadoop в своей команде
- В 2008 Hadoop становится Apache Top Level Project





Большие данные - это данные, которые слишком велики для обработки традиционными методами. Это ставит новые задачи, когда речь идет о хранении, обработке, извлечении и анализе больших данных.



Open Source



Apache Hadoop - одна из самых популярных технологий, которая создает основу для анализа больших данных. Hadoop – проект фонда Apache с открытым исходным кодом.

Это фреймворк, написанный на Java, разработанный Дугом Каттингом, который назвал его в честь игрушечного слона своего сына.



Проект Apache™ Hadoop® разрабатывает программное обеспечение с открытым исходным кодом для надежных, масштабируемых, распределенных вычислений.

Программная библиотека Apache Hadoop представляет собой среду, которая позволяет распределенную обработку больших наборов данных по кластерам компьютеров с использованием простых моделей программирования. Он предназначен для масштабирования от отдельных серверов до тысяч машин, каждый из которых предлагает локальные вычисления и хранилище. Вместо того, чтобы полагаться на аппаратное обеспечение для обеспечения высокой доступности, сама библиотека предназначена для обнаружения и обработки сбоев на уровне приложений, поэтому предоставляет высокодоступную службу поверх кластера компьютеров, каждый из которых может быть подвержен сбоям.

Проект включает в себя следующие модули:

Общее ПО Hadoop : Общие утилиты, которые поддерживают другие модули Hadoop.

Hadoop Distributed File System (HDFS™): распределенная файловая система, обеспечивающая высокопроизводительный доступ к данным.

Hadoop YARN: платформа для планирования заданий и управления ресурсами кластера.

Hadoop MapReduce: система на основе YARN для параллельной обработки больших наборов данных.

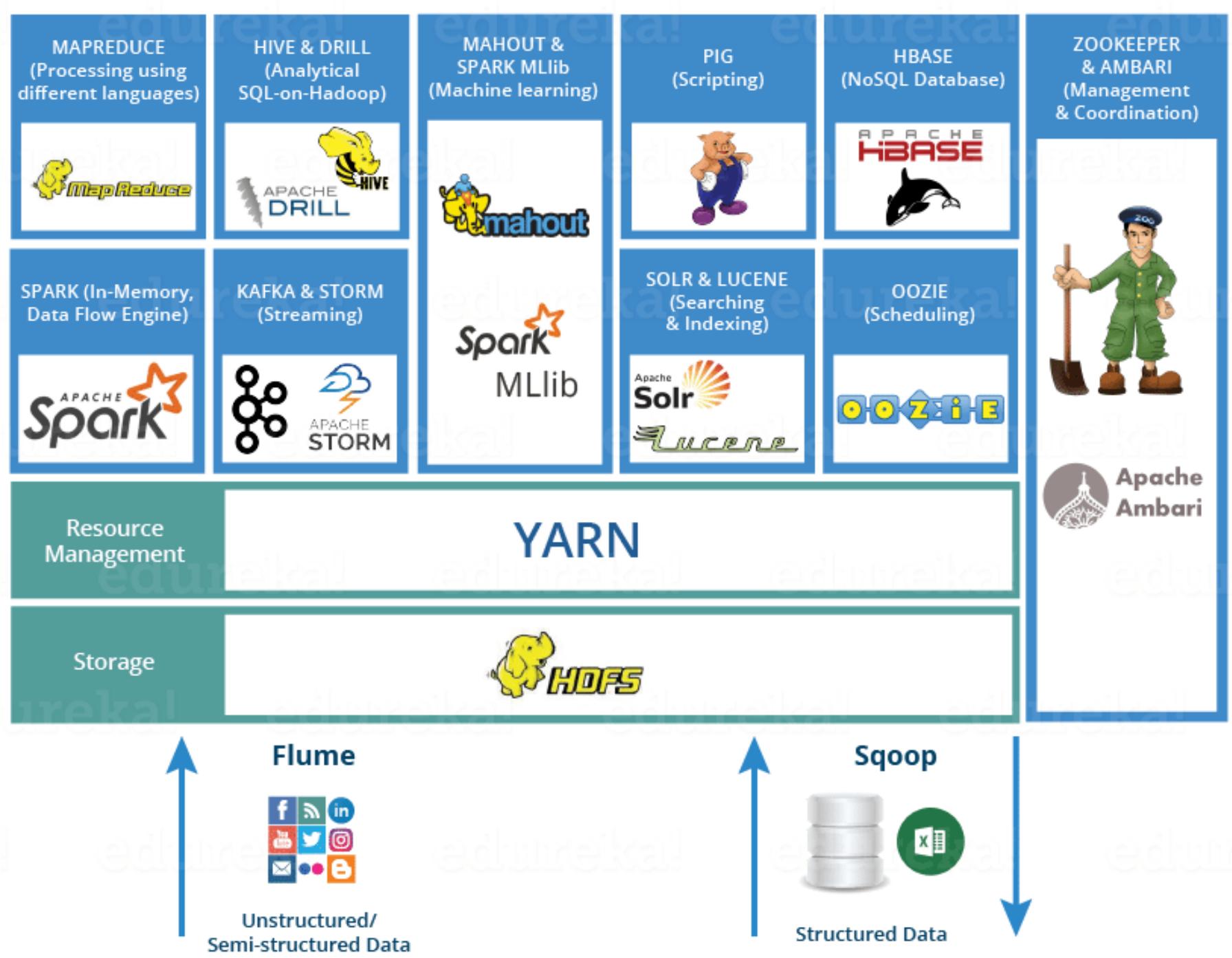
Hadoop Ozone: хранилище объектов для Hadoop.

Hadoop Submarine: механизм машинного обучения для Hadoop.

HADOOP ECOSYSTEM

Hadoop Ecosystem не является ни языком программирования, ни сервисом, это платформа, которая решает проблемы больших данных. Можно рассматривать его как набор, который включает в себя ряд сервисов (прием, хранение, анализ и обслуживание) внутри него.





Ниже приведены компоненты Hadoop, которые вместе образуют экосистему Hadoop:

- Spark -> Обработка данных в памяти
- PIG , HIVE -> Сервисы обработки данных с использованием запросов (в стиле SQL)
- HBase -> База данных NoSQL
- Mahout , Spark MLlib -> Машинное обучение
- Apache Drill -> SQL на Hadoop
- Zookeeper -> Управляющий кластер
- Oozie -> Планирование работы
- Flume , Sqoop -> Сервисы загрузки данных
- Solr & Lucene -> Поиск и индексирование
- Ambari -> Предоставление, мониторинг и поддержка кластера



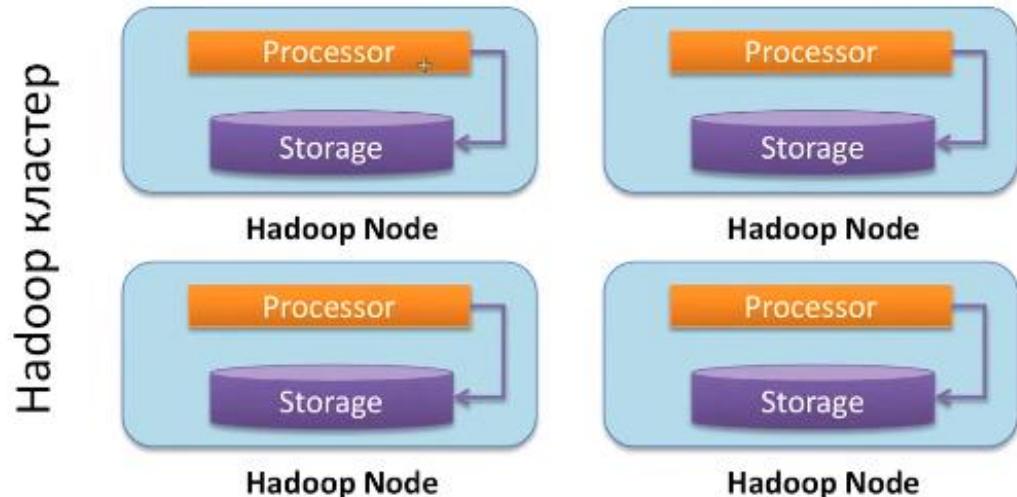
Системные принципы Hadoop

- Горизонтальное (Scale-Out) масштабирование вместо вертикального (Scale-Up)
- Отправляем код к данным
- Уметь обрабатывать падения нод и отказы оборудования
- Инкапсуляция сложности работы распределенных и многопоточных приложений

Масштабирование

- Вертикальное:
 - Добавить дополнительные ресурсы к существующему железу (CPU, RAM)
 - Если нельзя улучшить железо, то надо покупать более мощное новое
 - Закон Мура не успевает за ростом объема данных
- Горизонтальное:
 - Добавить больше машин к существующему кластеру
 - Приложение поддерживает добавление/удаление серверов
 - Просто масштабироваться “вниз”

Код к данным



Надоор кластер

Данные к коду



Инкапсуляция сложности реализации

- Hadoop скрывает многие сложности распределенных и многопоточных систем
- Освобождает разработчика от заботы о проблемах системного уровня
 - Race conditions, ожидание данных
 - Организация передачи данных, распределение данных, доставка кода и т.д.
- Позволяет разработчику фокусироваться на разработке приложения и реализации бизнес-логики

Отказы оборудования

- Чем больше количество машин, тем чаще будут отказы железа
- Hadoop разрабатывался с учетом отказов железа
 - Репликация данных
 - Перезапуск тасков

128MB	128MB	128MB	66MB
-------	-------	-------	------

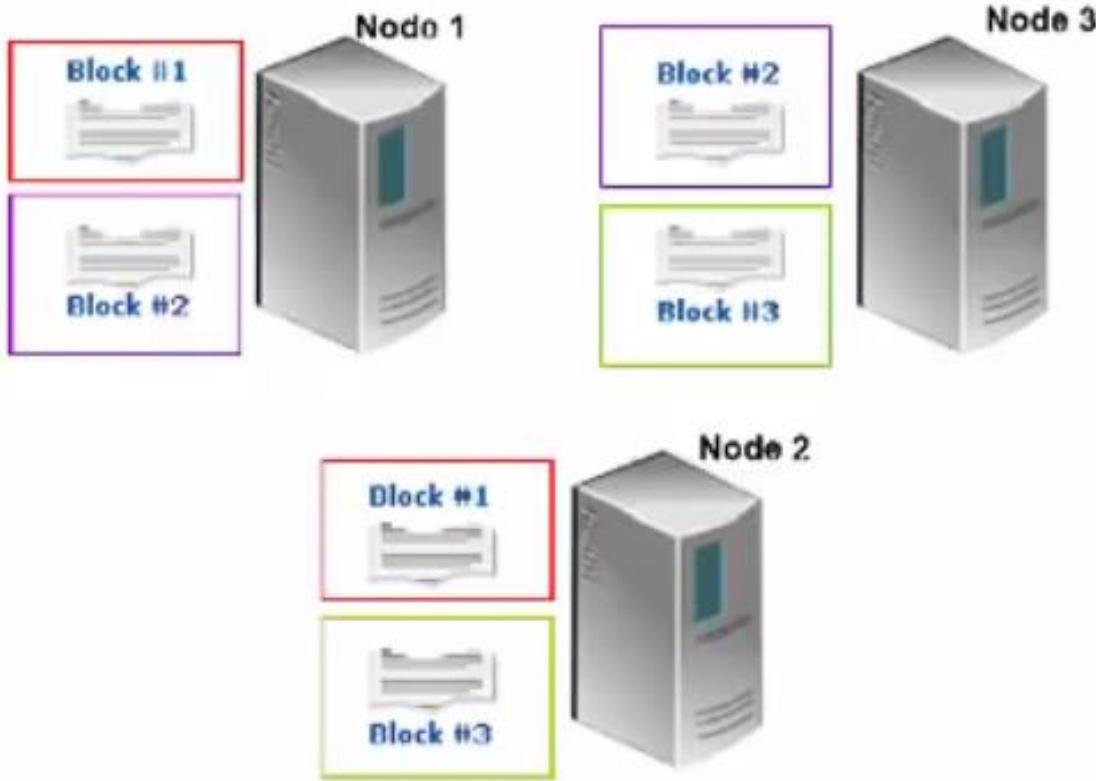
Hadoop использует блоки для хранения файла или частей файла.

Блок Hadoop - это файл в файловой системе. Поскольку базовая файловая система хранит файлы в виде блоков, один блок Hadoop может состоять из множества блоков в базовой файловой системе. Блоки большие. Они по умолчанию 64 мегабайта каждый и большинство систем работают с размером блока 128 мегабайт или больше. Блоки имеют несколько преимуществ:

- во-первых, они имеют фиксированный размер, что позволяет просто подсчитать, сколько можно уместить блоков на диске;
- во-вторых, будучи составленным из блоков, которые могут быть распределены по нескольким узлам, файл может быть больше, чем любой отдельный диск в кластере.

Если файл не кратен размеру блока, блок, содержащий остаток не занимает пространство всего блока. Как показано на рисунке, файл размером 450 мегабайт с размером блока 128 мегабайт занимает четыре блока, но четвертый блок не использует полный 128 мегабайт.

Наконец, блоки хорошо подходят для репликации, что позволяет HDFS быть отказоустойчивой и доступной на обычном оборудовании.

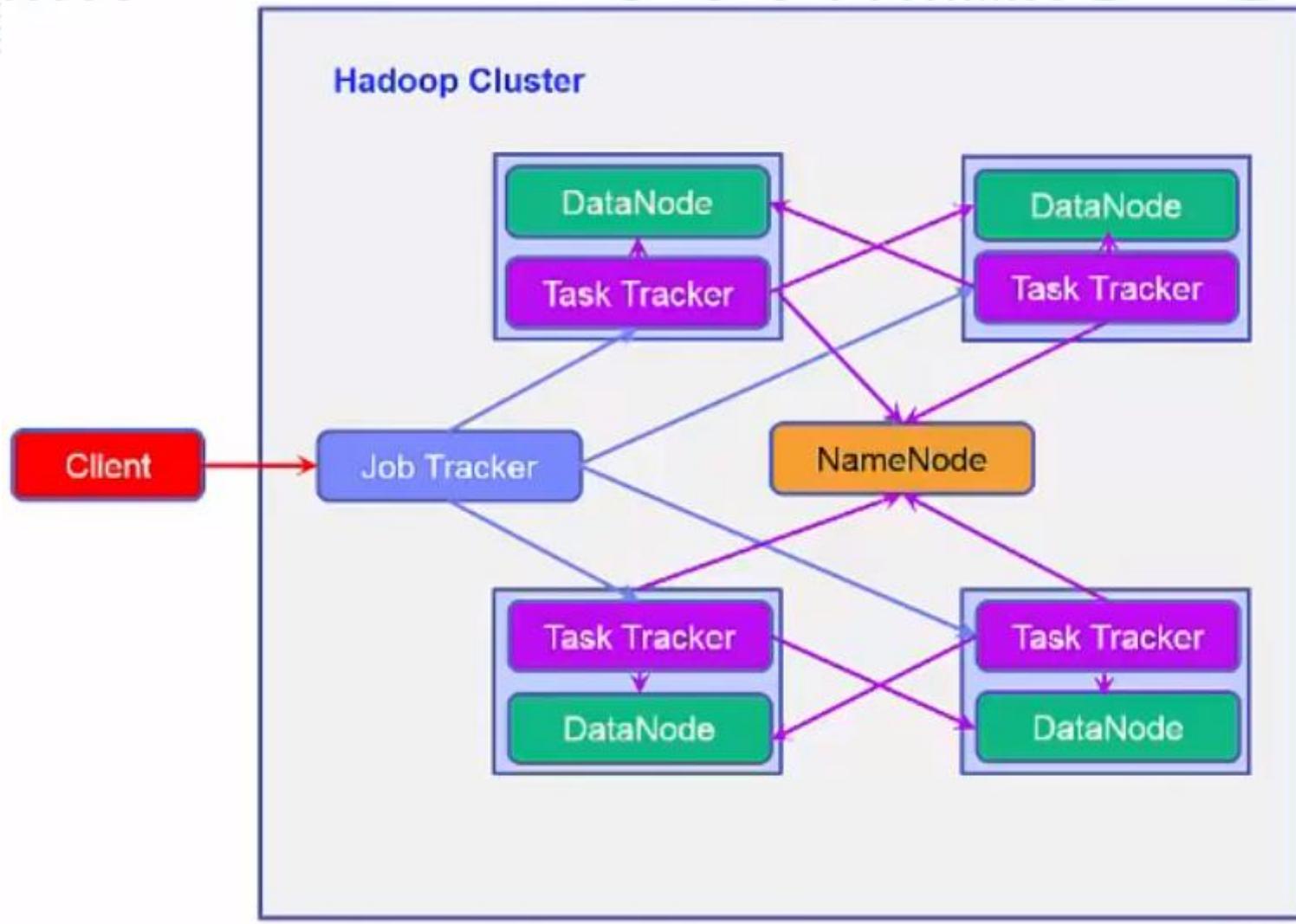


Каждый блок реплицируется на несколько узлов.

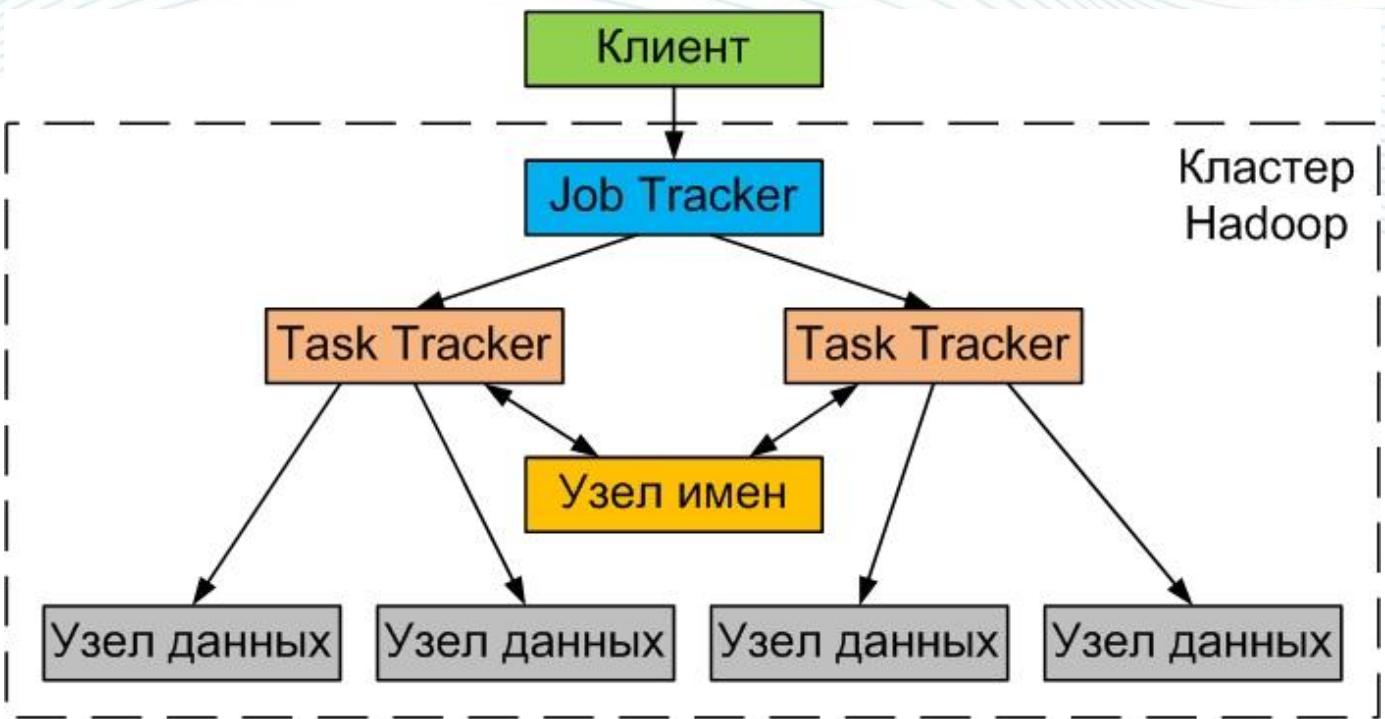
Например, блок 1 хранится на узле 1 и узле 2. Блок 2 хранится на узле 1 и узле 3. А блок 3 хранится на узле 2 и узле 3.

Это позволяет при сбое узла не потерять данные. Если узел 1 падает, узел 2 все еще работает и имеет данные блока 1.

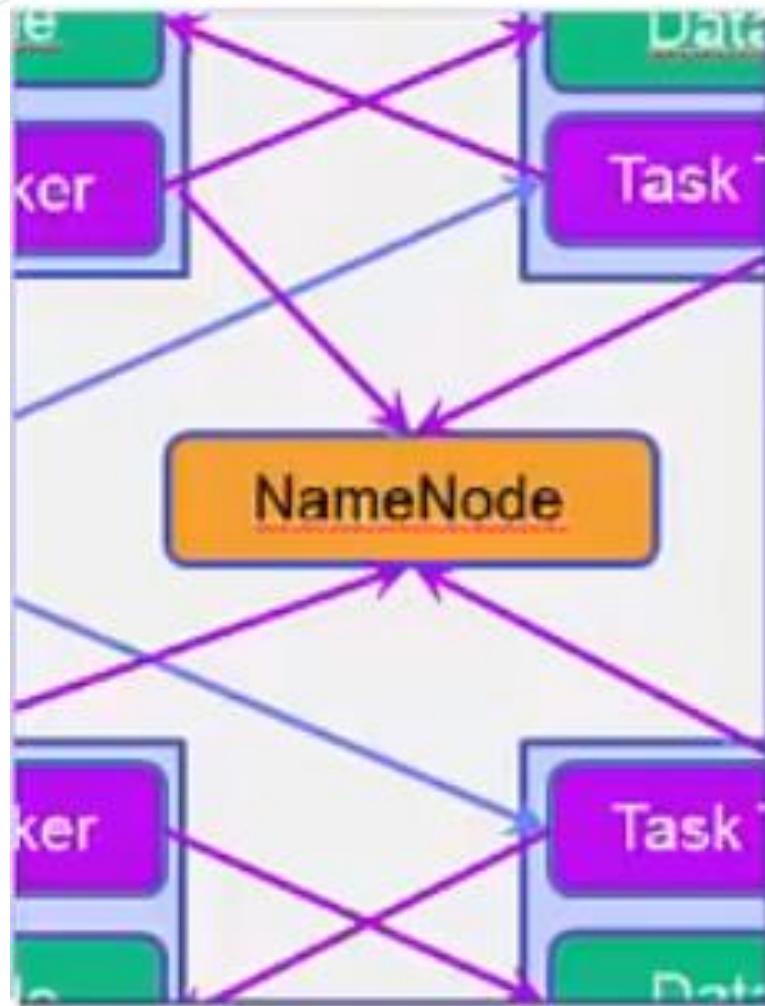
В этом примере мы только копируем данные на двух узлах, но можно настроить репликацию на множество других узлов, изменив настройки Hadoop или даже установить коэффициента репликации для каждого отдельного файла.



Основные типы узлов в Hadoop. Они разделяются на HDFS и MapReduce. Для узлов HDFS у нас есть NameNode и DataNodes. Для MapReduce у нас есть узлы JobTracker и TaskTracker. Эта диаграмма показывает связи между различными типами узлов в системе. Клиент показан связанным JobTracker. Он также может общаться с NameNode и с любым DataNode.

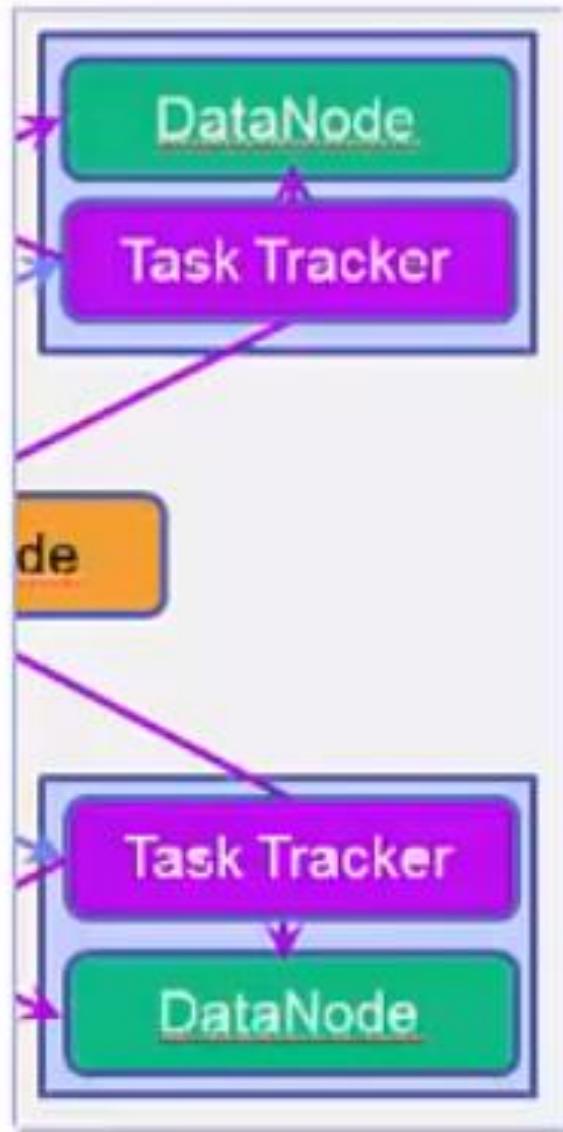


В кластере есть только один NameNode. В то время как данные, которые составляют файл, хранятся в блоках на узлах данных, метаданные для файла хранятся в NameNode. NameNode также отвечает за пространство имен файловой системы. Чтобы компенсировать тот факт, что существует только один NameNode, необходимо настроить NameNode для записи копии его состояния информации в нескольких местах, например, на локальном диске. NameNode также должен иметь как можно больше оперативной памяти, поскольку он сохраняет метаданные всей файловой системы в памяти.



Типичный кластер HDFS имеет много узлов данных. Узлы данных хранят блоки данных, один и тот же DataNode можно хранить блоки из разных файлов.

Когда клиент запрашивает файл, клиент узнает из NameNode, какие DataNodes хранят блоки, которые составляют файл и клиент непосредственно читает блоки из отдельных узлов данных. Каждый DataNode также периодически сообщает NameNode список блоков, которые он хранит.



Узлы DataNode не требует дорогостоящего корпоративного оборудования или репликации на аппаратном уровне. Узлы данных предназначены для работы на обычном оборудовании, и репликация предоставляется на программный уровень.

Узел JobTracker управляет заданиями **MapReduce**. В кластере есть только один JobTracker, он получает задания, представленные клиентом. Он планирует фазы **Map** и **Reduse** на соответствующие TaskTrackers, где находятся данные, в стойке и он отслеживает любые сбойные задачи, которые необходимо перенести на другой TaskTracker. Чтобы добиться параллелизма фазы **Map** и уменьшить количество задач, существует множество TaskTrackers в кластере Hadoop. Каждый TaskTracker порождает виртуальные машины Java для шага **Map** или шага **Reduse**. Он связывается с JobTracker и читает блоки из узлов данных.

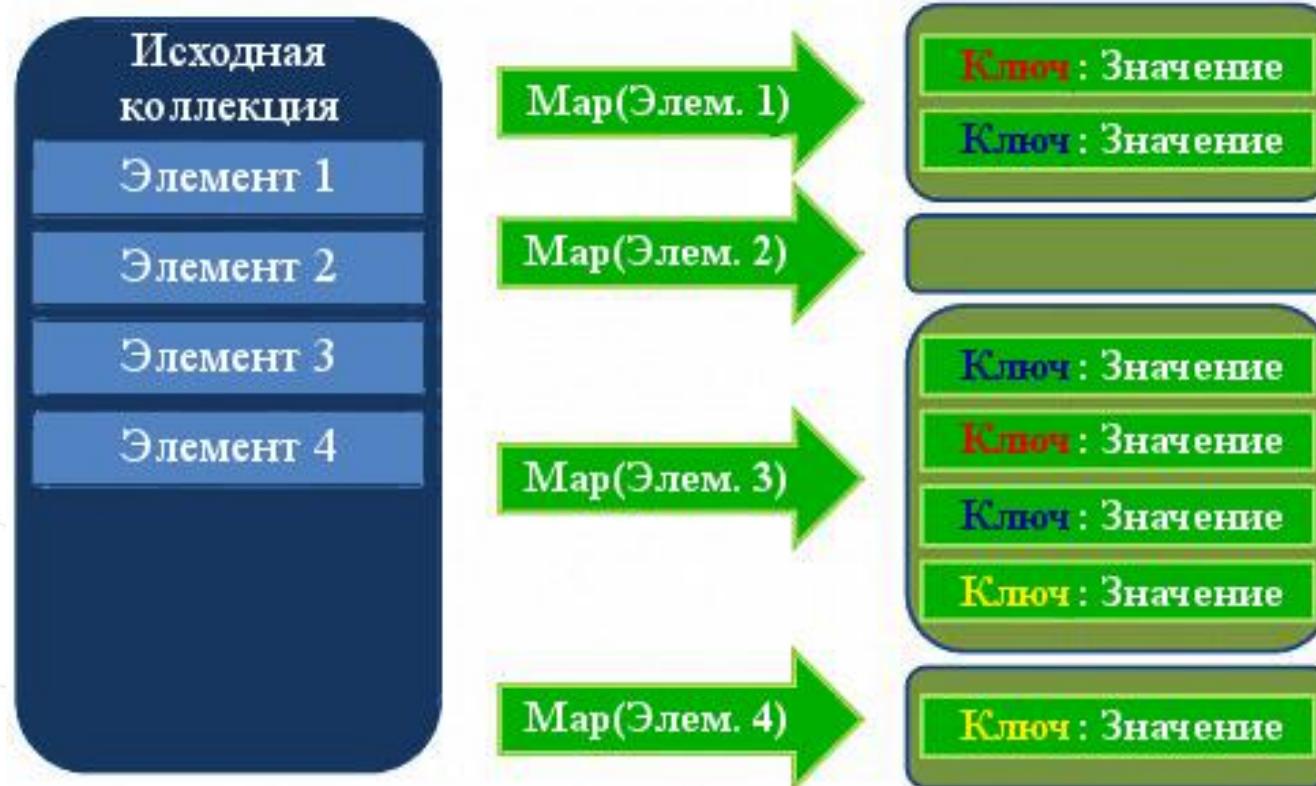
- MapReduce – программная модель предназначенная для параллельной обработки больших объемов данных за счет разделения задачи на независимые части
- MapReduce придумали в Google:
 - Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. 2004.
- Цель: упрощение индексации Web для поисковой системы
- В MapReduce к входному списку применяются последовательно функции Map и Reduce
- Списки состоят из пар: ключ-значение



Функция Мар

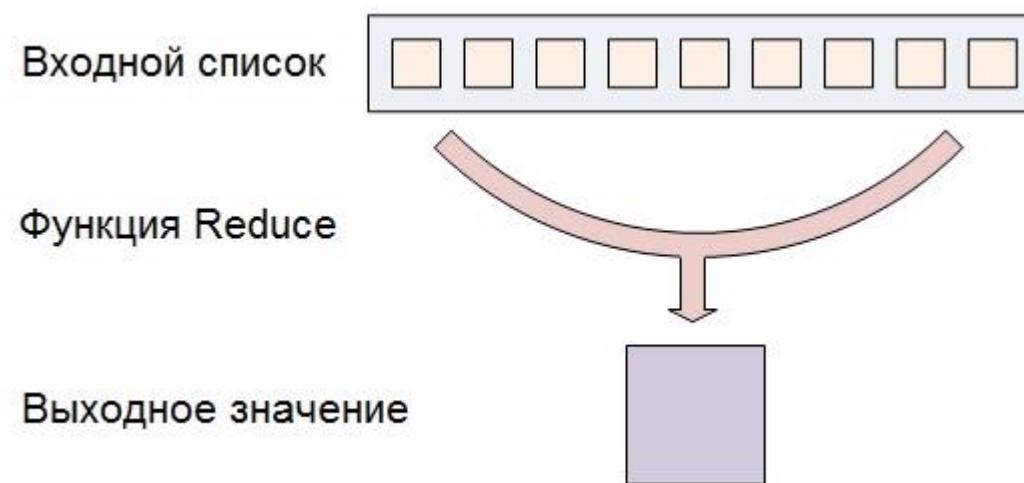
Функция Мар - принимает на вход список и некую функцию, затем применяет данную функцию к каждому элементу списка и возвращает новый список

Шаг Мар



Функция Reduce

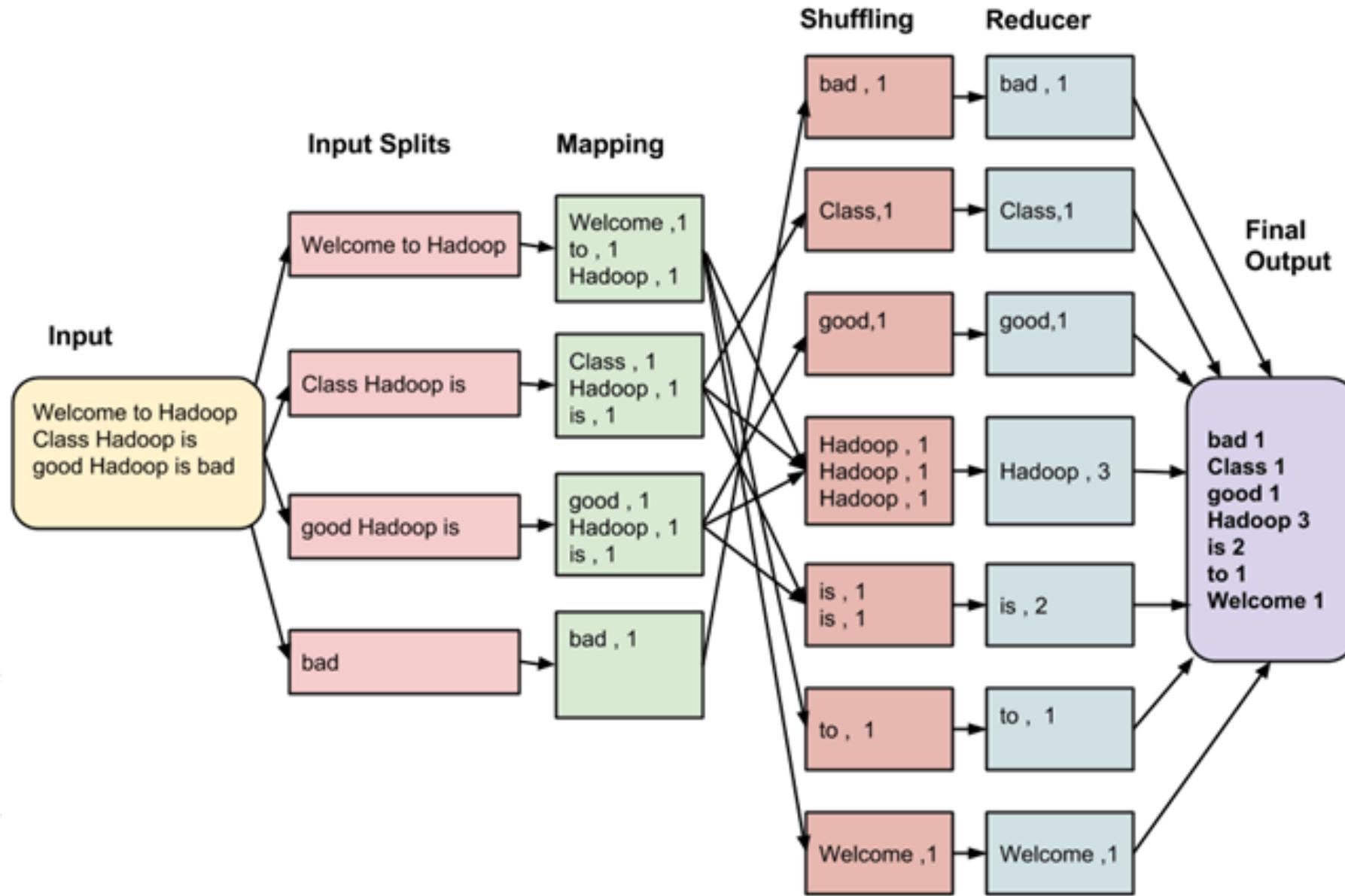
- Ставит в соответствие списку одно значение



- Пример функции Reduce: +
- Входной список также не меняется

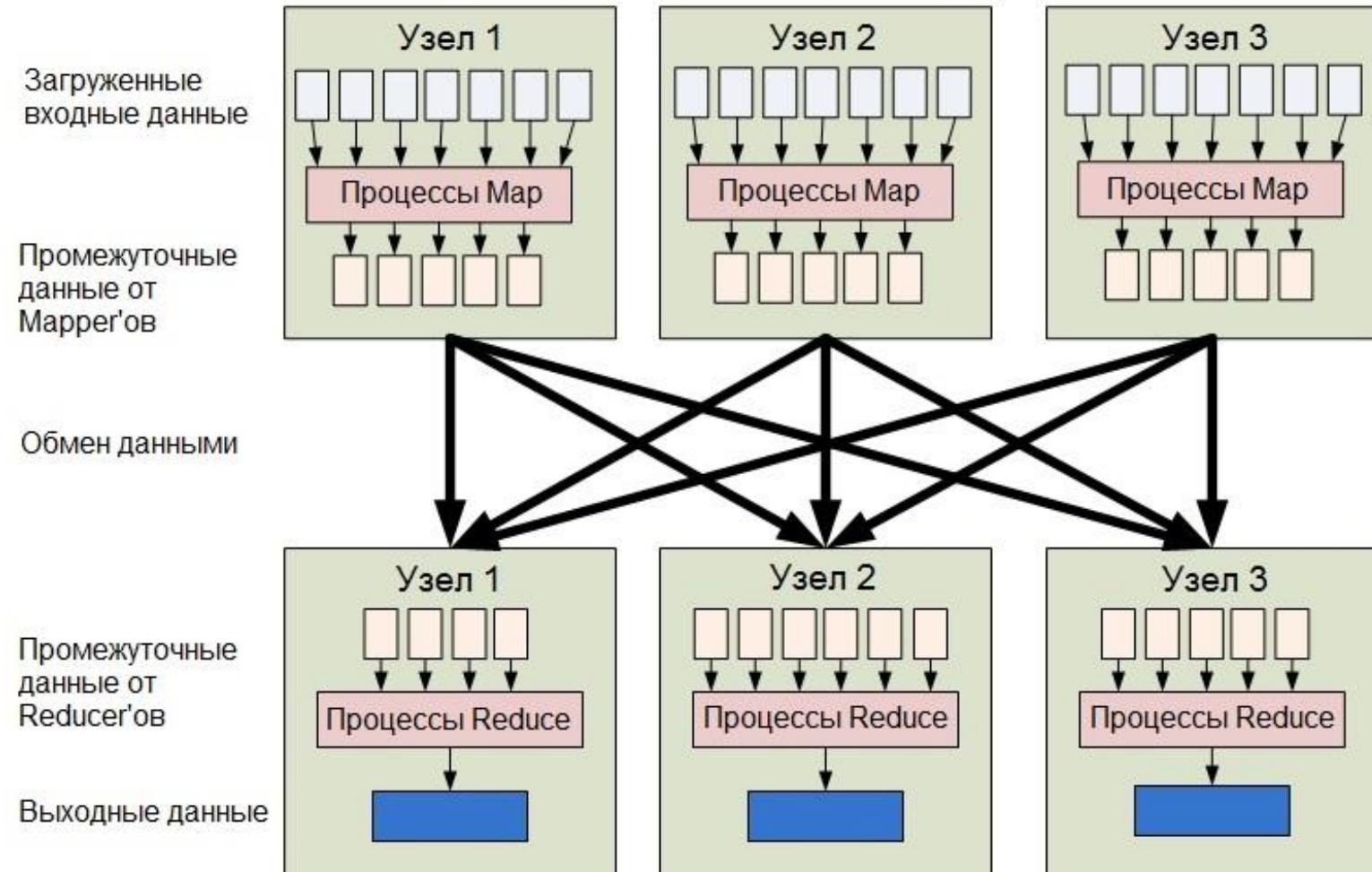


Процесс выполнения MapReduce для Wordcount



Поток данных в MapReduce

- Входные файлы загружаются в HDFS и распределяются по всем узлам
- На каждом узле запускаются процессы Map и обрабатывают входные файлы
 - Любой процесс Map может обрабатывать любой файл
- Процессы Map генерируют промежуточные списки пар ключ-значение



- Пары ключ-значение передаются по сети для обработки Reduce
- Все значения с одинаковым ключом передаются одному процессу Reduce
- Выходные данные в виде файлов записываются в HDFS



Поток данных в MapReduce

- Управление потоком данных MapReduce производится автоматически
- Программист не может менять поток данных
- Отдельные компоненты задачи не обмениваются данными между собой
 - В противном случае невозможно автоматическое восстановление после сбоя
- В случае сбоя узла процессы Map и Reduce автоматически перезапускаются на другом узле





Hadoop оптимизирован для обработки больших объемов данных, которые могут быть структурированными, неструктурными или полуструктурными, с использованием аппаратного обеспечения, то есть относительно недорогих компьютеров.

Эта массовая параллельная обработка выполняется с высокой производительностью. Hadoop копирует свои данные в разные компьютеры, так что если один выходит из строя, данные обрабатываются на одном из реплицированных компьютеров.

Hadoop **не подходит** для рабочих нагрузок OnLine Transaction Processing, где необходим произвольный быстрый доступ к структурированным данным, таким как реляционная база данных.

Кроме того, Hadoop не подходит для рабочих нагрузок OnLine Analytical Processing или Decision Support System.

Достоинства и недостатки Hadoop

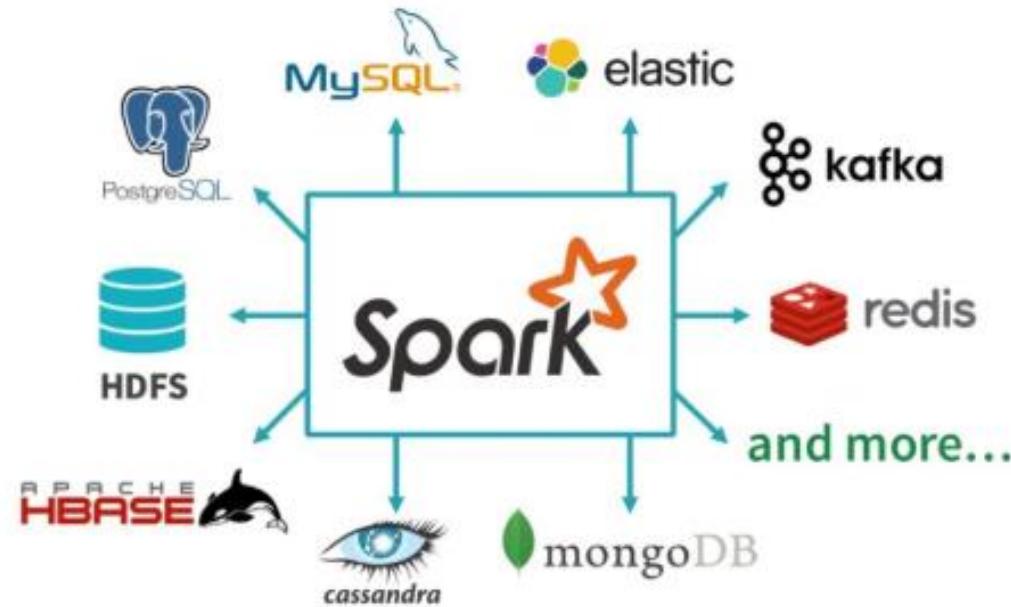
- Достоинства:
 - Простая модель программирования Map Reduce
 - Работа с большим объемом данных
 - Автоматизация распараллеливания
 - Защита от сбоев
- Недостатки
 - Длительный цикл разработки (реализация Mapper, Reducer, Driver, компиляция, упаковка в архив jar, копирование на кластер, запуск и т.д.)
 - Фиксированная последовательность обработки, нет workflow
 - Обязательная запись данных на диск после обработки
 - Сложно реализовать join
 - Только пакетная обработка
 - Чтение всех данных

Apache Spark

Технологические аспекты обработки больших данных

Apache Spark - платформа для обработки больших данных

Большинство организаций, сталкивающихся с необходимостью обработки больших объемов данных, используют для этих целей свободные проекты экосистемы Apache Hadoop. Основой для создания Apache Hadoop послужила разработанная компанией Google парадигма параллельного программирования MapReduce. Основные достоинства MapReduce - масштабируемость, простота использования, устойчивость к сбоям. Однако реализация MapReduce в Hadoop обладает рядом недостатков, основным из которых является низкая производительность при решении итеративных алгоритмов (например, машинного обучения).



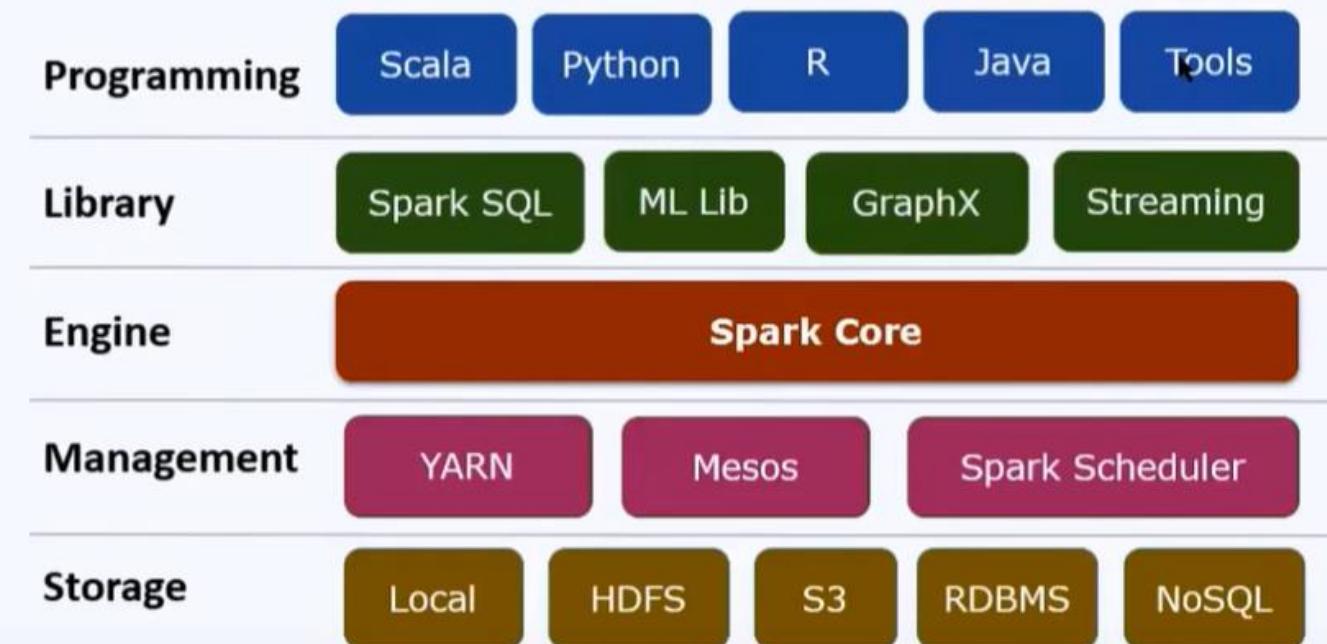
Крупнейший проект с открытым исходным кодом в обработке данных.

С момента выпуска Apache Spark , движок унифицированной аналитики, быстро внедряется предприятиями в широком спектре отраслей. Такие мощные интернет-компании, как Netflix, Yahoo и eBay, развернули Spark, совместно обрабатывая несколько петабайт данных в кластерах из более чем 8000 узлов. Spark быстро стал крупнейшим сообществом открытых данных в области больших данных, в котором приняли участие более 1000 участников из более 250 организаций.

Преимущества Spark

- Следующая ступень в обработке BigData:
 - Итеративные задачи
 - Интерактивная аналитика
- Может работать с разными типами данных (текст, графы, базы данных)
- Может обрабатывать данные по частям (batch) и в потоке (streaming)
- Имеет 80 высокоровневых функций для обработки данных (кроме map и reduce)

Spark Framework



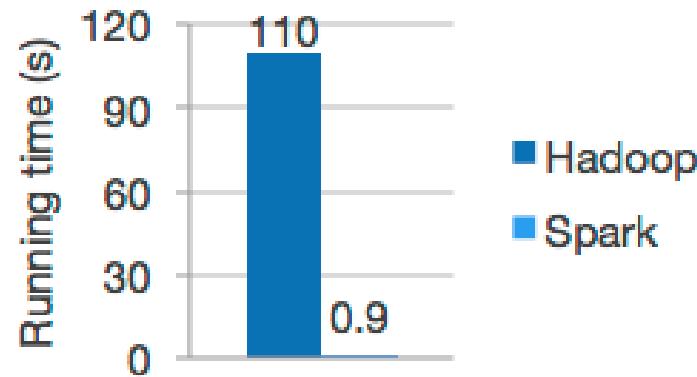


Apache Spark™

Apache Spark™ - это унифицированный аналитический механизм для обработки больших объемов данных.

В отличие от классического обработчика из ядра Hadoop, реализующего двухуровневую концепцию MapReduce с дисковым хранилищем, Spark использует специализированные примитивы для рекуррентной обработки в оперативной памяти, благодаря чему позволяет получать значительный выигрыш в скорости работы для некоторых классов задач, в частности, возможность многократного доступа к загруженным в память пользовательским данным делает библиотеку привлекательной для алгоритмов машинного обучения.





Logistic regression in Hadoop and Spark

СКОРОСТЬ

Выполнять рабочие нагрузки в 100 раз быстрее.

Apache Spark обеспечивает высокую производительность как для пакетных, так и для потоковых данных, используя современный планировщик DAG, оптимизатор запросов и механизм физического выполнения.

Простота использования

Быстрое написание приложений на Java, Scala, Python, R и SQL.

Spark предлагает более 80 операторов высокого уровня, которые облегчают создание параллельных приложений. И вы можете использовать его в *интерактивном режиме* из оболочек Scala, Python, R и SQL.

```
df = spark.read.json("logs.json")
df.where("age > 21")
    .select("name.first").show()
```



Apache Spark



всеобщность

Объедините SQL, потоковую и сложную аналитику.

Spark поддерживает стек библиотек, включая [SQL](#) и [DataFrames](#), [MLlib](#) для машинного обучения, [GraphX](#) и [Spark Streaming](#). Вы можете легко объединить эти библиотеки в одном приложении.

Работает везде

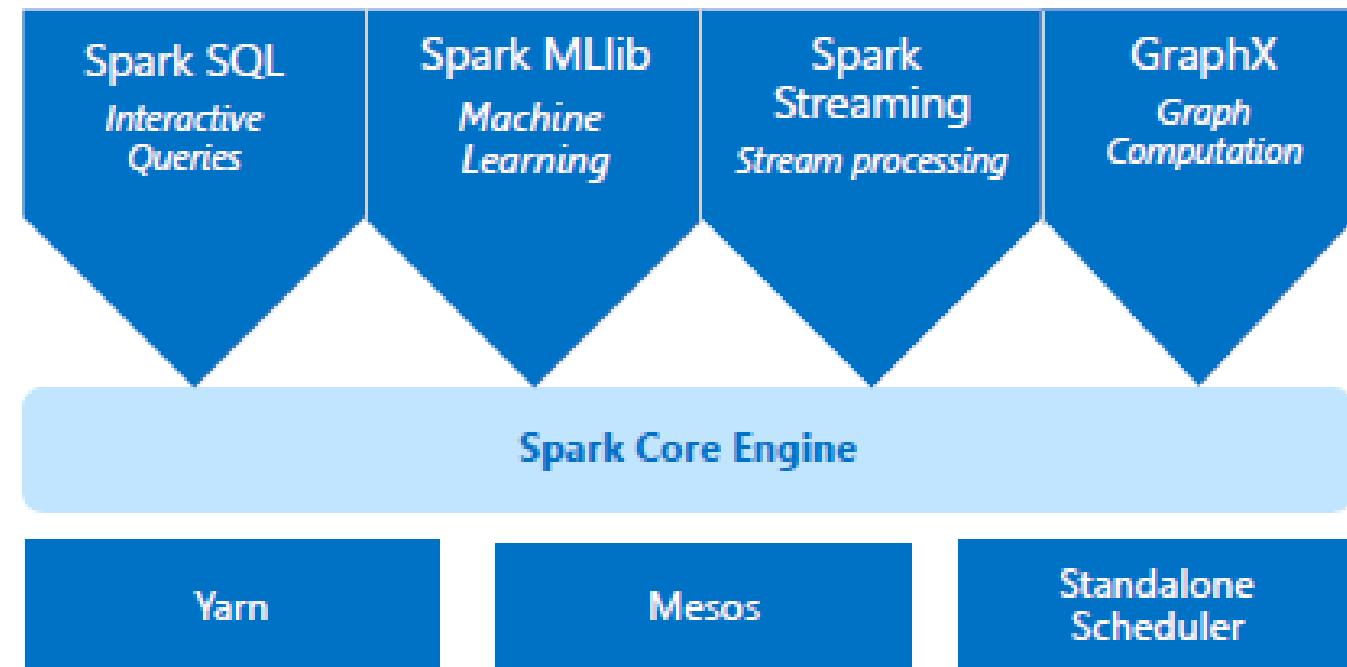
Spark работает на Hadoop, Apache Mesos, Kubernetes, в автономном режиме или в облаке. Он может получить доступ к различным источникам данных.

Вы можете запустить Spark в автономном кластерном режиме, в [EC2](#), в [Hadoop YARN](#), в [Mesos](#) или в [Kubernetes](#). Доступ к данным в [HDFS](#), [Alluxio](#), [Apache Cassandra](#), [Apache HBase](#), [Apache Hive](#) и сотнях других источников данных.

это унифицированная платформа, параллельной обработки и анализа больших данных с открытым исходным кодом

Объединяет в себя

- ✓ Пакетную обработку
- ✓ Интерактивный SQL
- ✓ Обработку в реальном времени
- ✓ Машинное обучение
- ✓ Глубокое обучение
- ✓ Движок работы с графиками



Производительность системы

- ✓ Spark существенно быстрее чем Hadoop благодаря вычислениям в оперативной памяти.
- ✓ Может использоваться для пакетной обработки данных
- ✓ Может использоваться для обработки данных в режиме реального времени

Унифицированная платформа

- ✓ Интегрированный фреймворк включает в себя высокоуровневые библиотеки для интерактивных запросов SQL, потоковой аналитики, машинного обучения и обработки графов.
- ✓ В одном приложении можно объединить все типы обработки данных.

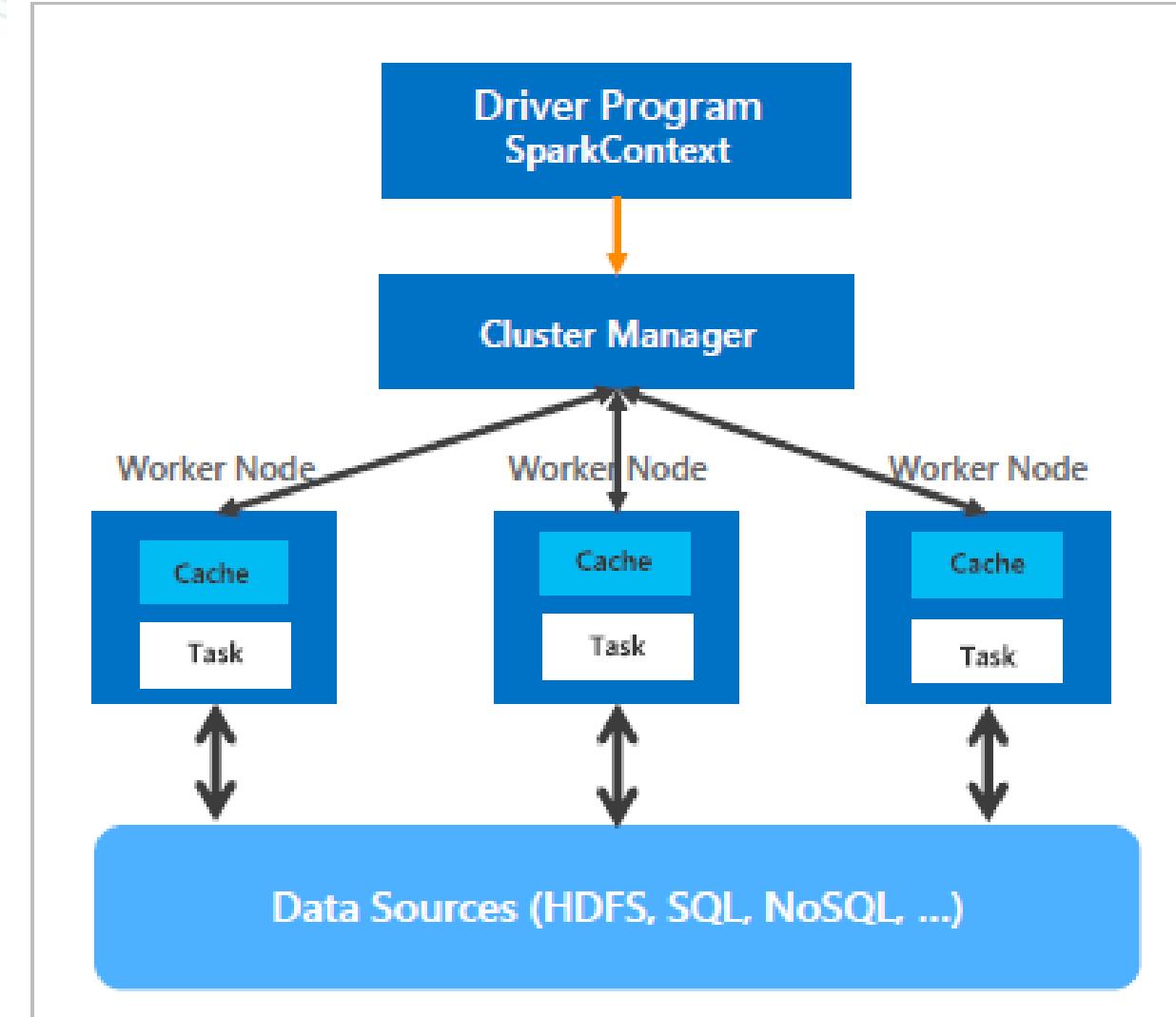
Продуктивность разработчиков

- ✓ Простые API для обработки больших массивов данных
- ✓ Содержит более 100 операторов для трансформации данных

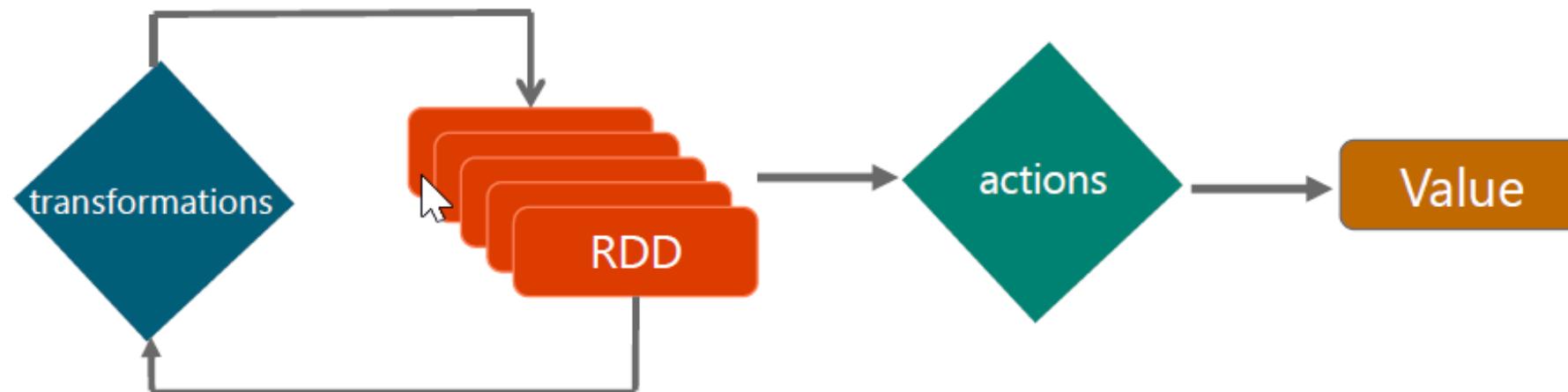
Экосистема

- ✓ В Spark есть встроенная поддержка для многих источников данных, богатая экосистема приложений независимых разработчиков ПО и активное сообщество разработчиков.
- ✓ Решение доступно на многих облачных платформах – Azure, AWS, Google и на локальных решениях

1. Драйвер запускает `main` функцию пользователя и выполняет различные параллельные операции на рабочих узлах
2. Результаты операций собираются драйвером
3. Рабочие узлы считывают и записывают или считывают данные из источников данных, включая HDFS.
4. рабочий узел также кэширует преобразованные данные в памяти в виде устойчивых распределенных наборов данных (en: RDD - Resilient Data Sets).
5. Рабочие узлы и узел драйвера выполняются как виртуальные машины в публичных облаках (Azure, AWS или Google)



- Устойчивые (эластичные) распределенные наборы данных - набор отказоустойчивых элементов, хранящихся в памяти или на диске, с которыми можно работать параллельно.
- RDD поддерживает два типа операций: Преобразование (Transformation) и Действия (Actions).
- Преобразование создает новый набор данных из существующего набора данных.
 - Все преобразования ленивы: они не вычисляют свои результаты сразу. Преобразования вычисляются только тогда, когда действие требует, чтобы результат был возвращен программе драйвера. Очевидно, что это не относится к сохраняемым на диск RDDs.
- Действия возвращают значение драйверу после выполнения вычисления с набором данных.

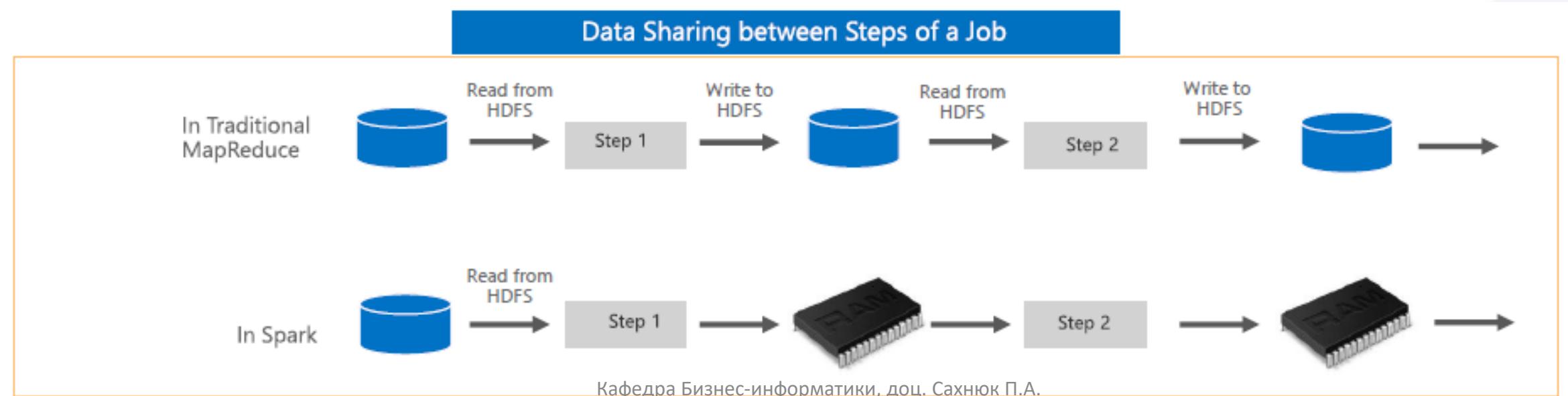


Кластерные вычисления в памяти

- ✓ Spark предоставляет примитивы для кластерных вычислений в памяти. Задание Spark может загружать и кэшировать данные в память и запрашивать их повторно (итеративно) намного быстрее, чем системы основанные на вводе выводе на диск.

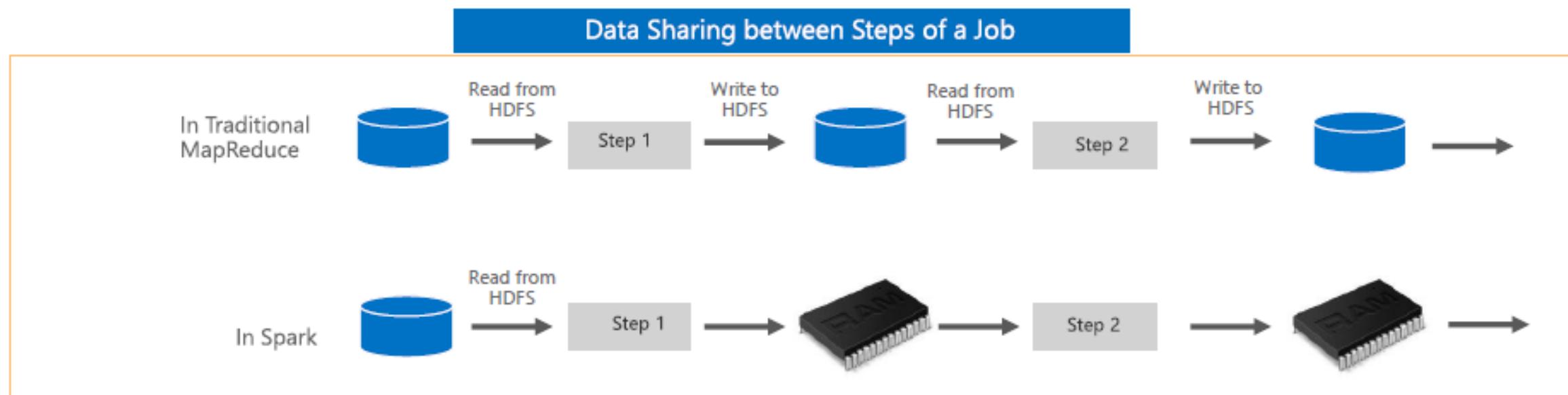
Интеграция со Scala

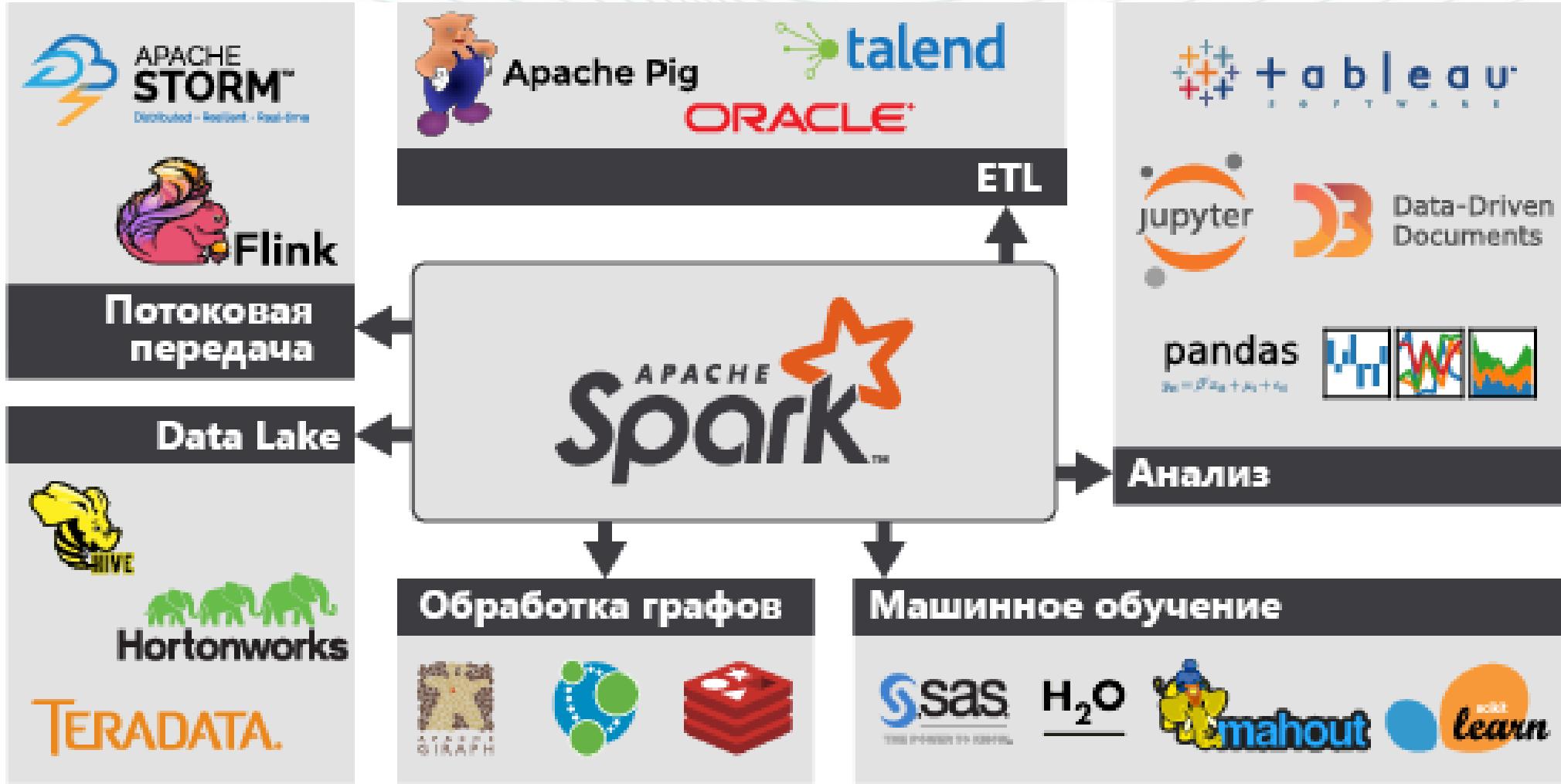
- ✓ Spark интегрируется с языком программирования Scala. Это позволяет работать с распределёнными наборами данных как с локальными коллекциями.
- ✓ Нет необходимости структурировать все операции как операции Map/Reduce.



Быстрый обмен данными

- ✓ Обмен данными между операторами происходит быстрее, так как данные находятся в памяти.
- ✓ в (традиционном) Hadoop данные передаются через HDFS, что долго: HDFS необходимо сделать три реплики прежде чем продолжить операции.
- ✓ Spark хранит данные в памяти без какой-либо репликации.

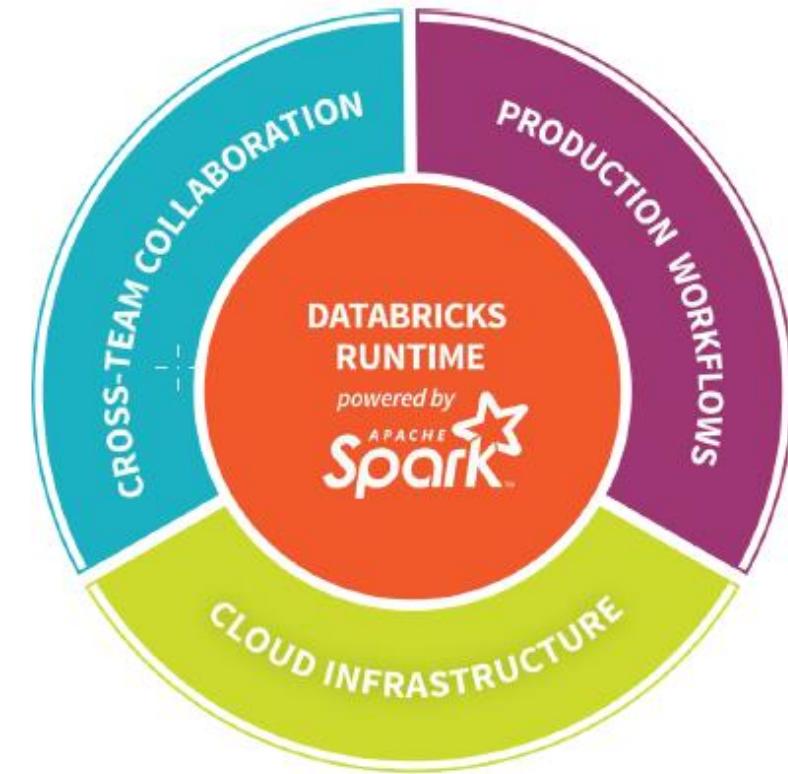




Spark предлагает систему вычислений и соединители практически для любого источника данных. Используя легко масштабируемую инфраструктуру и обращаясь к данным в месте хранения, Spark решает основные потребности приложений больших данных.

DATABRIKS

- Основана в 2013 году
- Создатели Apache Spark. Команда из amplab.
- amplab - лаборатория Калифорнийского университета в Беркли, специализирующаяся на аналитике больших данных. Название расшифровывается как «Алгоритмы, машины и люди»
- Крупнейший контрибутор в Apache Spark
- 2/3 проектов поддерживают партнеры (*крупнейшие партнеры: Hortonworks, MapR, DataStax*)
- Производит [сертификацию](#) по таким направлениям как: Databricks Certified Application, Databricks Certified Distribution and Databricks Certified Developer
- Собственный продукт: [Unified Analytics Platform](#)
- В октябре представлен продукт: [Databricks Delta](#)



Все ваши данные, аналитика и искусственный интеллект на единой платформе данных



Надежная
инженерия данных
→



SQL Analytics для
всех ваших данных
→



Совместная наука о
данных →



Производственное
машинное обучение
→

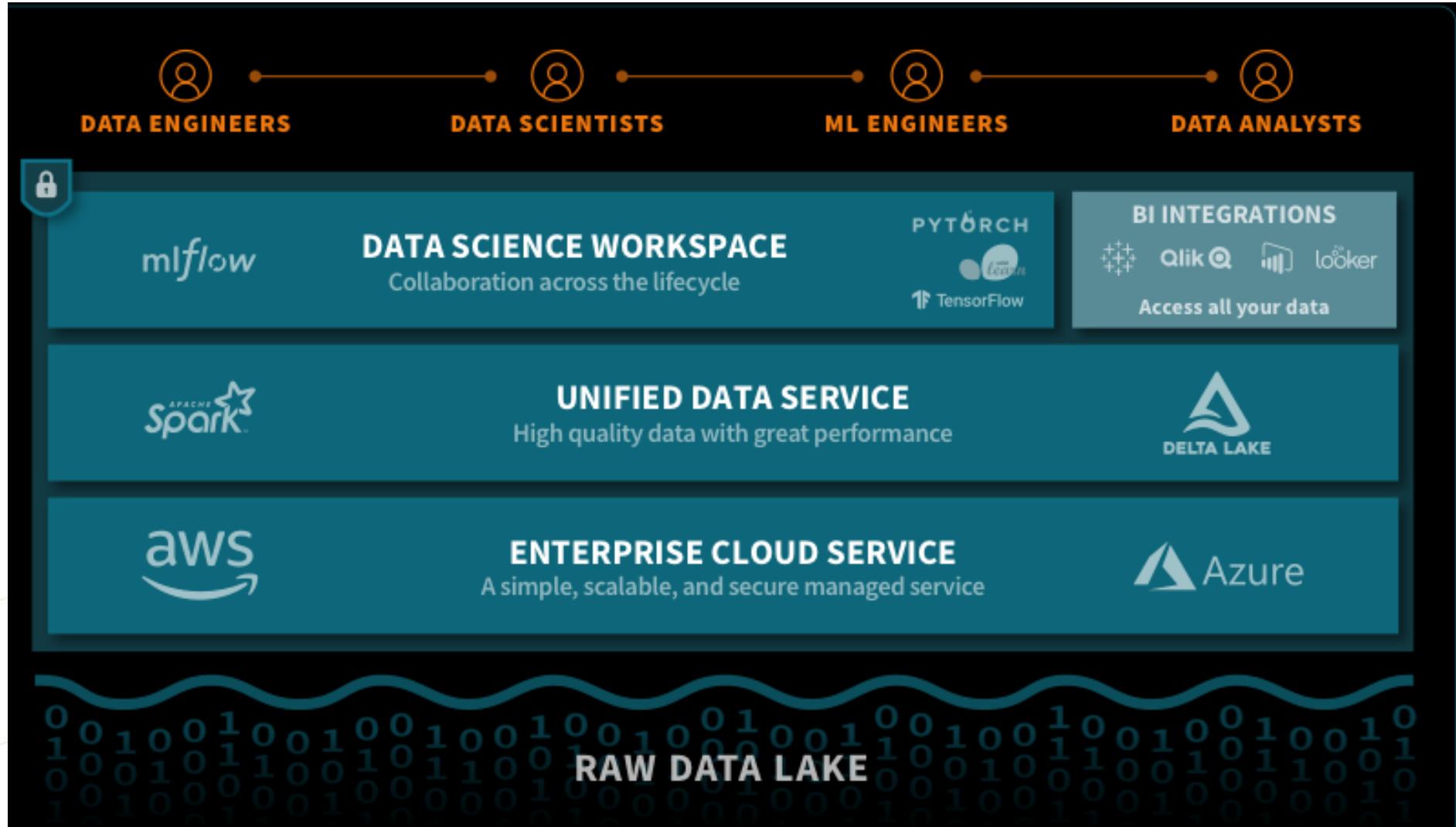


Databricks, основанная на Delta Lake, объединяет лучшие хранилища данных и озера данных в архитектуру Lakehouse, предоставляя вам единую платформу для совместной работы над всеми вашими данными, аналитикой и рабочими нагрузками AI.

На основе открытого исходного кода Создатели Apache Spark™, Delta Lake и MLflow, считают, что будущее данных и ИИ зависит от программного обеспечения с открытым исходным кодом и миллионов разработчиков, которые ежедневно вносят в него свой вклад. Создайте свой бизнес на открытой платформе, не зависящей от облака.

Единая платформа аналитики данных

Одна облачная платформа для крупномасштабного проектирования данных и совместной работы над данными



Data Science Workspace

Сотрудничество на протяжении всего жизненного цикла данных и машинного обучения

Единая платформа данных Databricks

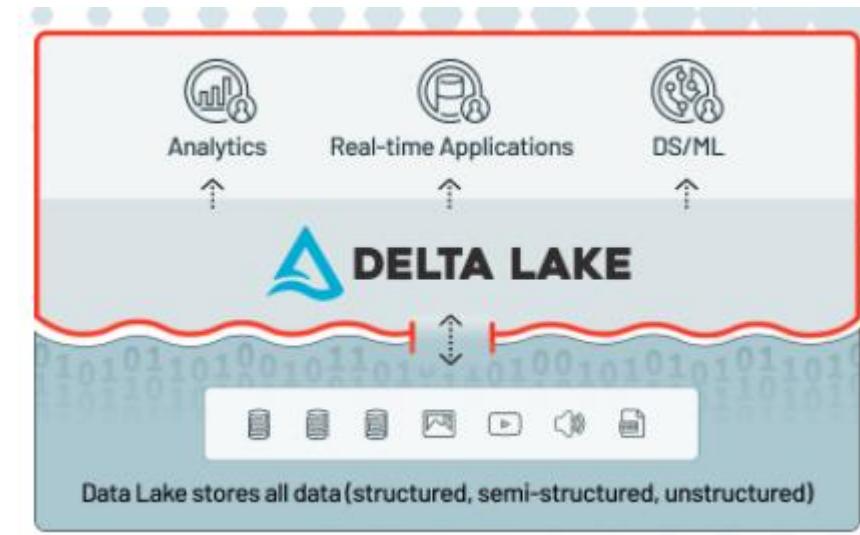
Одна открытая и простая платформа для хранения и управления всеми вашими данными для всех ваших аналитических рабочих нагрузок

Дельта Лейк

Надежность и производительность для озер данных

Надежные озера данных

Delta Lake обеспечивает надежность и масштабируемость данных в вашем существующем озере данных с помощью уровня хранения транзакций с открытым исходным кодом, предназначенного для полного жизненного цикла данных.



Быстрые и эффективные конвейеры данных

Простая обработка данных в инфраструктуре автомасштабирования. Работает на высокооптимизированном Apache Spark™ для увеличения производительности до 50 раз.

Наука о данных и машинное обучение

Сотрудничество на протяжении всего жизненного цикла науки о данных и машинного обучения



Блокноты для совместной работы

Быстрый доступ к данным и их изучение, поиск новых идей и обмен ими, а также совместное создание моделей с использованием языков и инструментов по выбору.



Оптимизированные среды машинного обучения

Доступ в один клик к предварительно настроенным средам машинного обучения для расширенного машинного обучения с использованием современных и популярных фреймворков машинного обучения.



Оптимизированные среды машинного обучения

Полный жизненный цикл машинного обучения
Отслеживайте эксперименты и делитесь ими, воспроизводите прогоны и управляйте моделями совместно из центрального репозитория, от экспериментов до производства.

Бизнес-аналитика

Более полные и актуальные данные для анализа каждой команды



Запросы озер данных с помощью SQL

Выполняйте рабочие нагрузки SQL непосредственно в озере данных, чтобы запрашивать и анализировать самые свежие данные с соотношением цена / производительность до 9 раз лучше, чем у традиционных облачных хранилищ данных.



Визуализируйте и делитесь идеями

Быстро и легко визуализируйте результаты запросов и организуйте визуализацию на многофункциональных панелях мониторинга, чтобы делиться информацией в реальном времени с вашей командой с автоматическими предупреждениями о критических изменениях.



Широкая интеграция инструментов бизнес-аналитики

Используйте предпочтаемые вами инструменты бизнес-аналитики, такие как Tableau и Microsoft Power BI, с оптимизированными соединителями, которые обеспечивают высокую производительность, низкую задержку и высокий уровень одновременного доступа пользователей к вашему озеру данных.

SQL Analytics

Цена/производительность для бизнес-аналитики и аналитики до 9 раз выше, чем у традиционных облачных хранилищ данных

SQL Analytics позволяет клиентам использовать многооблачную архитектуру «lakehouse architecture», которая обеспечивает производительность хранилища данных при экономичности озера данных и обеспечивает до 9 раз лучшее соотношение цены и производительности для рабочих нагрузок SQL по сравнению с традиционными облачными хранилищами данных.

SQL Analytics интегрируется с инструментами бизнес-аналитики, такими как Tableau и Microsoft Power BI, которые вы используете сегодня для запроса самых полных и последних данных в вашем озере данных. Дополните существующие инструменты бизнес-аналитики встроенным интерфейсом SQL, который позволяет аналитикам и специалистам по обработке данных запрашивать данные озера данных непосредственно в Databricks.

Делитесь аналитическими данными о запросах с помощью расширенных визуализаций и панелей мониторинга с возможностью перетаскивания с автоматическим оповещением о важных изменениях в ваших данных.

Обеспечьте надежность, качество, масштабируемость, безопасность и производительность вашего озера данных для поддержки традиционных рабочих нагрузок аналитики с использованием ваших самых последних и полных данных.

Преимущества

Собственный интерфейс SQL
В дополнение к поддержке существующих инструментов бизнес-аналитики SQL Analytics предлагает полнофункциональный редактор запросов на базе SQL, который позволяет аналитикам данных писать запросы в знакомом синтаксисе и легко исследовать схемы таблиц [Delta Lake](#). Регулярно используемый код SQL можно сохранить в виде фрагментов для быстрого повторного использования, а результаты запросов можно кэшировать, чтобы сократить время выполнения.

The screenshot shows the Databricks SQL interface. On the left is a sidebar with icons for dashboards, queries, endpoints, history, and alerts. The main area has a title "Average Rating by Company Location" and a sub-section "chocolate". It displays a shared endpoint dropdown set to "Shared Endpoint", a table dropdown set to "chocolate", and a "chocolate.flavors_of_cacao" table schema. The schema includes columns: Company (Maker-if known) STRING, Specific Bean Origin or Bar Name STRING, REF INT, Review Date INT, Cocoa Percent STRING, Company Location STRING, Rating DOUBLE, Bean Type STRING, and Broad Bean Origin STRING. Below the schema is a query editor with the following SQL code:

```
1 SELECT
2     Company_Location,
3     avg(Rating) AS avgRating
4 FROM chocolate.reviews
5 WHERE Review_Date = {{Year}}
6 GROUP BY Company.Location
7 ORDER BY avgRating ASC
```

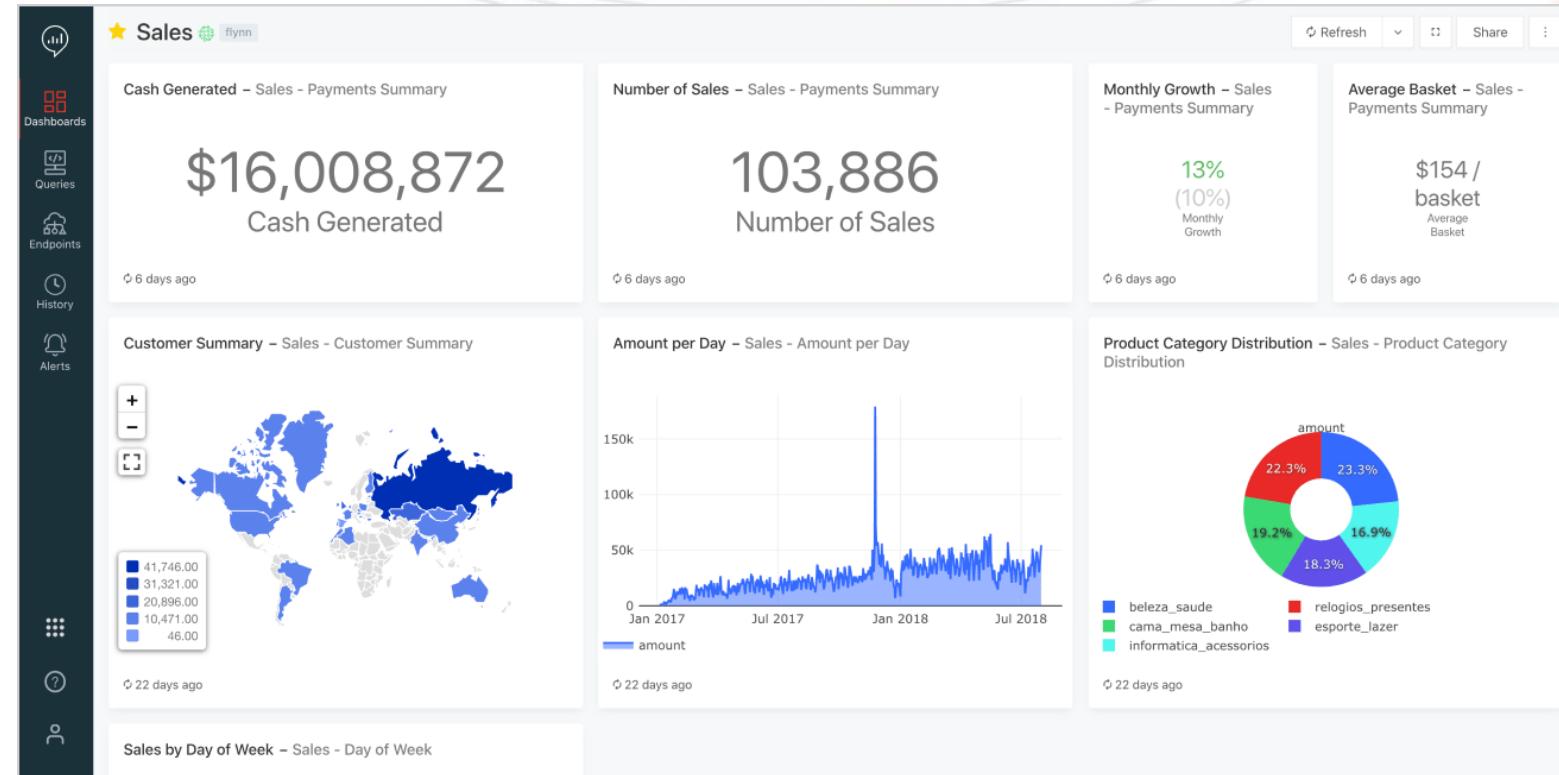
Below the query editor is a "Year" filter set to "2019". The results are displayed in a table view titled "Table" and a map visualization titled "Map (Choropleth)". The table view shows the following data:

Company_Location	avgRating
India	2.50
Israel	2.75
Puerto Rico	2.75
South Africa	2.75
Portugal	2.75
Venezuela	2.88

At the bottom right, there are navigation buttons for pages 1, 2, and 3, and a note "30 rows 3 seconds runtime". The status bar at the bottom right says "Refreshed just now".

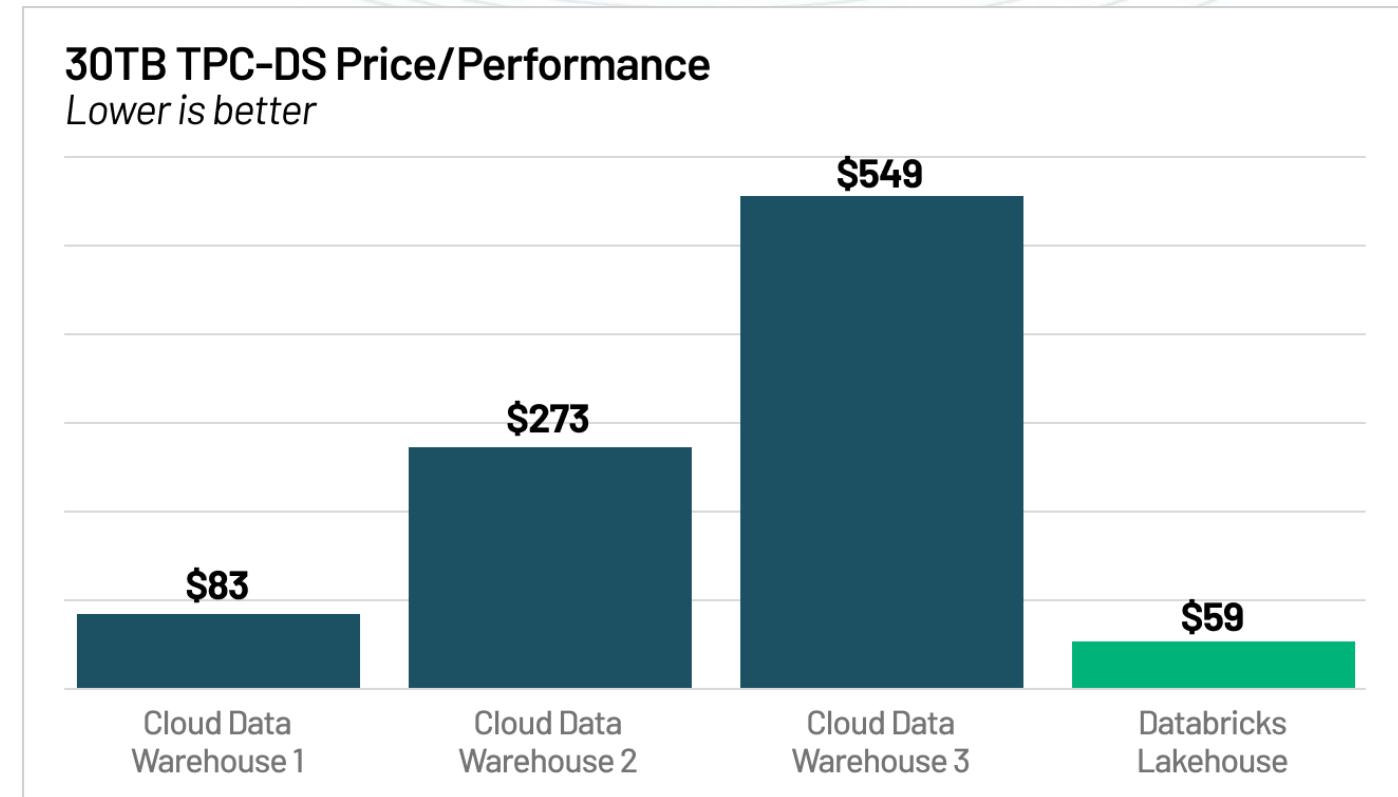
Преимущества

Легко создавайте визуализации и делитесь информационными панелями. После создания запросов SQL Analytics также позволяет аналитикам анализировать результаты с помощью широкого набора разнообразных визуализаций. Визуализации можно быстро организовать в панели мониторинга с помощью интуитивно понятного интерфейса перетаскивания. Панели мониторинга можно легко предоставить заинтересованным лицам как внутри организации, так и за ее пределами, через веб-браузер. Чтобы все были в курсе, информационные панели можно настроить на автоматическое обновление, а также на оповещение команды о значимых изменениях данных.



Преимущества

Повышение цены / производительности до 9 раз для бизнес-аналитики
SQL Analytics позволяет клиентам получить производительность хранилища данных в экономике озера данных с помощью оптимизированных для SQL вычислительных кластеров на базе [Delta Engine](#), которые напрямую запрашивают таблицы Delta Lake в озерах данных. Благодаря SQL Analytics запросы обеспечивают до 9 раз лучшее соотношение цены и производительности по сравнению с другими облачными хранилищами данных. SQL Analytics автоматически и прозрачно распределяет нагрузку между запросами в нескольких кластерах, чтобы обеспечить высокий уровень параллелизма и малую задержку без сбоев.



Data Science Workspace

Сотрудничество на протяжении всего жизненного цикла данных и машинного обучения



Совместные тетради
Быстрый доступ к данным и их изучение, поиск новых идей и обмен ими, а также совместное создание моделей с использованием языков и инструментов.



Оптимизированная среда ML
Доступ одним щелчком мыши к предварительно сконфигурированным средам ML для расширенного машинного обучения с использованием самых современных и популярных платформ ML.



Полный жизненный цикл ML
Отслеживайте и делитесь экспериментами, воспроизводите прогоны и совместно управляйте моделями из центрального хранилища, от экспериментов до производства.

Совместные тетради

Совместная наука о данных со знакомыми языками и инструментами

Выполняйте быстрое исследование данных или создавайте модели машинного обучения, используя совместные записные книжки, которые поддерживают несколько языков, встроенные средства визуализации данных, автоматическое управление версиями и ввод в эксплуатацию с заданиями.



advertising-analytics-click-prediction-ml-gbt (Python)

Detached File View: Code Permissions Run All Clear Schedule Comments Runs

You are viewing a notebook revision from Jul 18 2018, 9:41 AM PDT. [Exit](#)

Features by weight

```

1 import json
2 features = map(lambda c: str(json.loads(json.dumps(c))['name']), \
3                 predictions.schema['features'].metadata.get('ml_attr').get('attrs').values()[0])
4 # convert numpy.float64 to str for spark.createDataFrame()
5 weights=map(lambda w: '%.10f' % w, model.featureImportances)
6 weightedFeatures = sorted(zip(weights, features), key=lambda x: x[1], reverse=True)
7 spark.createDataFrame(weightedFeatures).toDF("weight", "feature").createOrReplaceTempView('wf')

```

Command took 0.12 seconds -- by tony.cruz@dataricks.com at 7/18/2018, 5:25:46 AM on unknown cluster

```

1 %sql
2 select feature, weight
3 from wf
4 order by weight desc

```

feature	weight (%)
C21_idx	44%
C19_idx	2%
site_category_idx	2%
app_category_idx	7%
hr_idx	5%
banner_pos_idx	5%
C18_idx	4%
C16_idx	2%
device_conn_type_idx	1%
C1_idx	1%
C15_idx	1%

Runs

- Jul 18 2018, 10:11 AM PDT denny.lee@dataricks.com
- Jul 18 2018, 10:09 AM PDT denny.lee@dataricks.com
- Jul 18 2018, 9:57 AM PDT denny.lee@dataricks.com
- Jul 18 2018, 9:46 AM PDT denny.lee@dataricks.com
- Jul 18 2018, 9:41 AM PDT Tony Cruz, denny.lee@dataricks.com
 - Restore this revision
- Jul 18 2018, 5:32 AM PDT Tony Cruz
- Jul 18 2018, 5:24 AM PDT Tony Cruz
- Jul 18 2018, 5:04 AM PDT Tony Cruz
- Jul 18 2018, 4:50 AM PDT Tony Cruz
- Jul 18 2018, 4:43 AM PDT Tony Cruz
- Jul 16 2018, 21:30 PM PDT denny.lee@dataricks.com
- Jul 16 2018, 21:22 PM PDT denny.lee@dataricks.com
- Jul 16 2018, 21:13 PM PDT denny.lee@dataricks.com
- Jul 16 2018, 17:16 PM PDT denny.lee@dataricks.com
- Jul 16 2018, 14:36 PM PDT denny.lee@dataricks.com

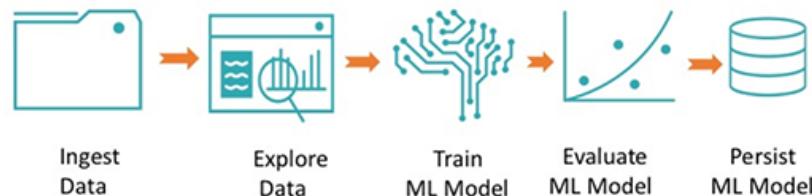
РАБОТАТЬ ВМЕСТЕ
Делитесь записными
книжками и
работайте со
сверстниками на
нескольких языках
(R, Python, SQL и
Scala) и библиотеках
по выбору.
Совместное
редактирование в
реальном времени,
комментирование и
автоматическое
управление
версиями упрощают
совместную работу,
оставаясь под
контролем.

Pipeline-1

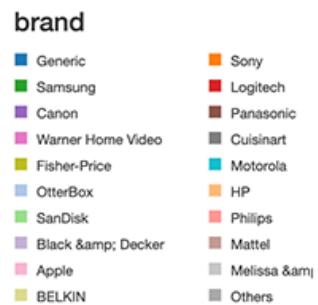
amazon-model : /dbfs/mnt/databricks-derr

amazon-stream-input : /dbfs/mnt/databricks-derr

Data Engineering Pipeline



Show Various Brands



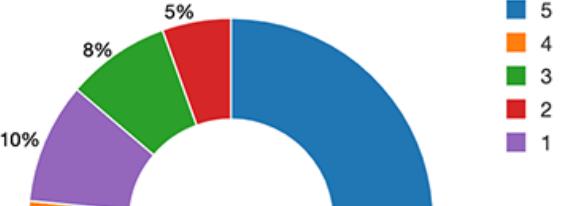
Step 1-3: Ingest Data

This table is distillation and derivation of [Amazon's public data](#) for their product offering on [Amazon](#), made up rating, reviews, brand, title, time, etc. Once the data is read, parsed, and cleansed, we created a table with the desired columns for further analysis.

You can use the Python script on this [page](#) to download the gzipped files, parse the files, save them as Parquetfiles, and create an SQL table from it.

(Because the data is fairly clean—last time we ingested it—we will

Aggregate Amazon Product Ratings



ДЕЛИТЬСЯ
Быстро открывайте
новые идеи с
помощью встроенных
интерактивных
визуализаций или
любой библиотеки,
такой как matplotlib
или ggplot.
Экспортируйте
результаты и
записные книжки в
формате html или
ipynb или создавайте
и публикуйте
информационные
панели, которые
всегда остаются
актуальными.

The screenshot shows the Databricks interface for a job named '04-Visualizations Dashboards'. The left sidebar contains navigation links for Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area displays the job details: Job ID: 18561, Task: Notebook at /_AWS_Day/_OnSite/00_StartHere/04-Visualizations Dashboards - Edit / Remove, Cluster: Driver: i3.xlarge, Workers: i3.xlarge, 8 workers, On-Demand and Spot, fall back to On-Demand, 5.5 LTS (includes Apache Spark 2.4.3, Scala 2.11) Edit, Schedule: None Edit, and Advanced options. Below this, the 'Active runs' section shows a table with columns: Run, Run ID, Start Time, Launched, Duration, Spark, and Status. A link 'Run Now / Run Now With Different Parameters' is provided. The 'Completed in past 60 days' section shows a table of completed runs with the same columns. The latest successful run is listed as Run 6, started on 2019-08-30 07:19:40 PDT, launched manually, and succeeded. Other completed runs are listed as Run 5, Run 4, Run 3, and Run 2.

Run	Run ID	Start Time	Launched	Duration	Spark	Status
Run 6	2381398	2019-08-30 07:19:40 PDT	Manually	3m 1s	Spark UI / Logs / Metrics	Succeeded
Run 5	2381268	2019-08-23 07:00:00 PDT	By scheduler	3m 5s	Spark UI / Logs / Metrics	Succeeded
Run 4	2381266	2019-08-23 06:48:00 PDT	By scheduler	3m 5s	Spark UI / Logs / Metrics	Succeeded
Run 3	2381264	2019-08-23 06:36:00 PDT	By scheduler	3m 5s	Spark UI / Logs / Metrics	Succeeded
Run 2	2381262	2019-08-23 06:24:00 PDT	By scheduler	3m 4s	Spark UI / Logs / Metrics	Succeeded

РАБОТАЕТ В БОЛЬШИХ МАСШТАБАХ
Запланируйте записные книжки для автоматического запуска машинного обучения и конвейеров данных в масштабе и создайте многоступенчатые конвейеры, используя рабочие процессы записных книжек.
Настройка оповещений и быстрый доступ к журналам аудита для простого мониторинга и устранения неполадок.

Доступ к данным : быстрый доступ к доступным наборам данных или подключение к любым источникам данных, локальным или облачным.

Поддержка нескольких языков: исследуйте данные с помощью интерактивных записных книжек с поддержкой нескольких языков программирования в одной записной книжке, включая R, Python, Scala и SQL.

Интерактивные визуализации: визуализируйте идеи с помощью широкого ассортимента визуализаций «укажи и щелкни». Или используйте мощные скриптовые опции, такие как matplotlib, ggplot и D3.

Соавторство в реальном времени : Работайте на одном ноутбуке в режиме реального времени, отслеживая изменения с подробной историей изменений.

Комментарии: Оставьте комментарий и сообщите коллегам из общих записных книжек.

Автоматическое управление версиями: отслеживание изменений и управление версиями происходит автоматически, так что вы можете продолжить с того места, на котором остановились, или отменить изменения.

Управление версиями в Git: интеграция с Git для обеспечения гибких и расширенных возможностей управления версиями.

Боковая панель прогонов: автоматически регистрирует эксперименты, параметры и результаты из записных книжек непосредственно в MLflow в виде прогонов, а также быстро просматривает и загружает предыдущие прогоны и версии кода с боковой панели.

Панели мониторинга: делитесь мнениями с коллегами и клиентами или позволяйте им выполнять интерактивные запросы с панелями управления на основе Spark.

Запускать записные книжки как рабочие места. Превратите записные книжки или JAR-файлы в устойчивые производственные задания одним щелчком мыши или вызовом API.

Планировщик заданий: выполнение заданий для производственных конвейеров по заданному расписанию.

Ноутбук Workflows: Создание многоступенчатых конвейеров с контрольными структурами исходного языка программирования.

Уведомления и журналы. Настройте оповещения и получите быстрый доступ к журналам аудита для удобного мониторинга и устранения неполадок.

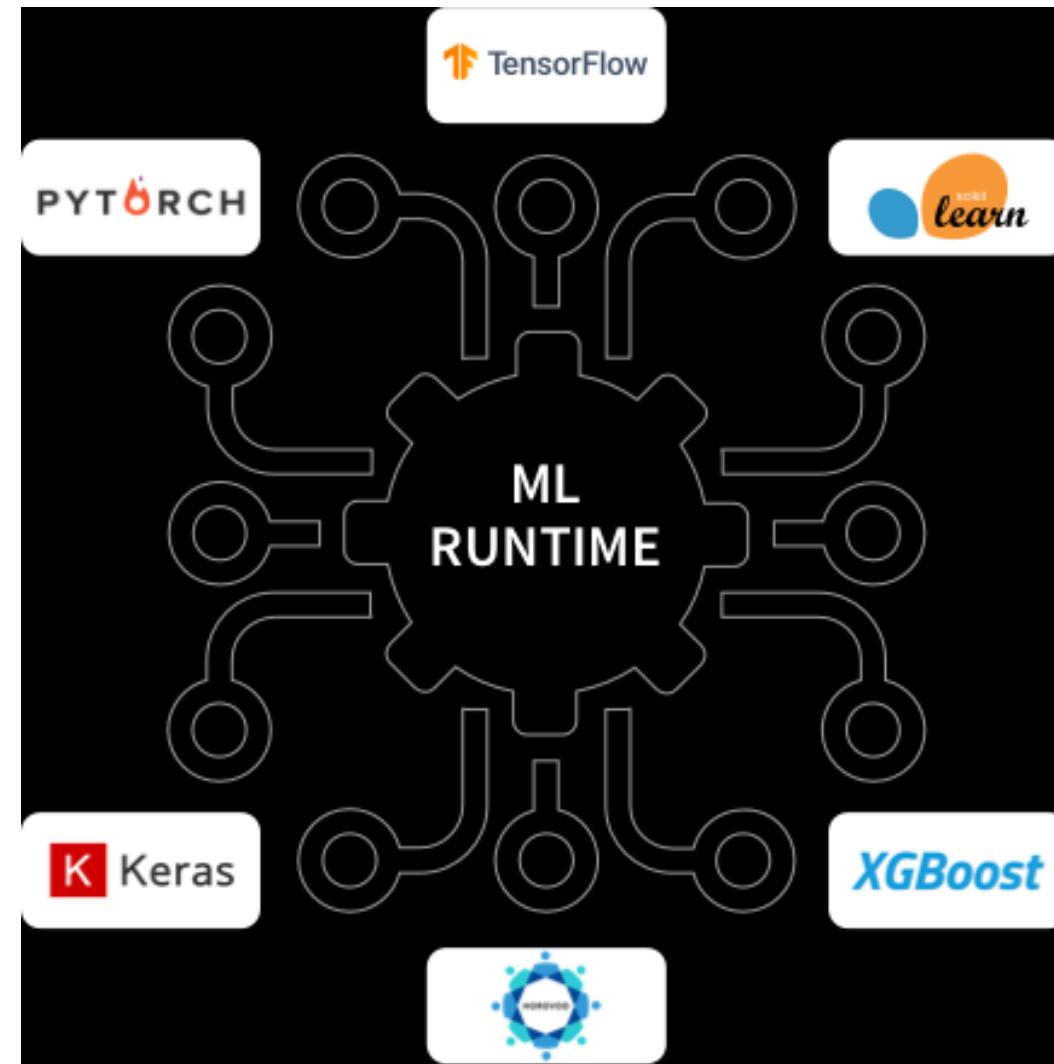
Управление разрешениями. Быстрое управление доступом к каждой отдельной записной книжке или коллекции записных книжек и экспериментами с помощью одной общей модели безопасности.

Кластеры. Быстрое подключение ноутбуков к автоматически управляемым кластерам для эффективного и экономичного масштабирования вычислений в беспрецедентном масштабе.

Интеграция: подключение к Tableau, Looker, Power BI, RStudio, SnowFlake и др., позволяя ученым и инженерам по обработке данных использовать знакомые инструменты.

Machine Learning Runtime

Готовая к использованию и оптимизированная среда машинного обучения



Среда машинного обучения (MLR) предоставляет ученым и специалистам по ML данные масштабируемые кластеры, включающие популярные платформы, встроенный AutoML и оптимизации для обеспечения непревзойденной производительности.

ОСНОВЫ ВЫБОРА

Фреймворки ML развиваются в очень быстром темпе, и практикующие должны управлять в среднем 8 библиотеками. ML Runtime обеспечивает доступ одним щелчком к надежному и производительному дистрибутиву наиболее популярных сред ML и пользовательских сред ML через предварительно собранные контейнеры или Conda.

ДОПОЛНИТЕЛЬНОЕ ОБУЧЕНИЕ МАШИНЫ

Ускорьте машинное обучение от подготовки данных к выводу с помощью встроенных возможностей AutoML, включая настройку гиперпараметров и поиск моделей с использованием Hyperopt и MLflow.

Упрощенное масштабирование

Легко переходите от небольших данных к большим с помощью автоматически управляемой и масштабируемой кластерной инфраструктуры. Среда машинного обучения также включает уникальные улучшения производительности для самых популярных алгоритмов, а также HorovodRunner, простой API для распределенного глубокого обучения.

ОСНОВЫ ВЫБОРА

Conda Managed Runtime: преимущества интеграции Conda для управления пакетами Python. Все пакеты Python устанавливаются в одной среде.

Фреймворки ML: Самые популярные библиотеки и фреймворки ML предоставляются из коробки, включая TensorFlow, Keras, PyTorch, MLflow, Horovod, GraphFrames, scikit-learn, XGboost, numpy, MLeap и Pandas.

AUGMENTED ML

Автоматическое отслеживание экспериментов: отслеживайте, сравнивайте и визуализируйте сотни тысяч экспериментов с использованием открытого источника или управляемого MLflow и функции построения параллельных координат.

Автоматический поиск моделей (для одноузлового ML): Оптимизированный и распределенный условный поиск гиперпараметров по нескольким модельным архитектурам с улучшенной Hyperopt и автоматическим отслеживанием MLflow.

Автоматическая настройка гиперпараметров для одноузлового машинного обучения: Оптимизированный и распределенный поиск гиперпараметров с улучшенным Hyperopt и автоматическим отслеживанием до MLflow.

Автоматическая настройка гиперпараметров для распределенного машинного обучения: глубокая интеграция с перекрестной проверкой PySpark MLlib для автоматического отслеживания экспериментов MLlib в MLflow.

Оптимизирован для упрощенного масштабирования

Оптимизированный TensorFlow. Воспользуйтесь преимуществами оптимизированной для TensorFlow версии CUDA для кластеров графических процессоров и оптимизированного пакета Intel MKL-DNN TensorFlow для процессоров Intel для максимальной производительности.

HorovodRunner: быстрая миграция учебного кода глубокого обучения на одном узле для запуска в кластере Databricks с HorovodRunner, простым API, который устраняет сложности, возникающие при использовании Horovod для распределенного обучения.

Оптимизированные MLlib логистическая регрессия и классификаторы деревьев: наиболее популярные оценщики были оптимизированы как часть среды выполнения Databricks для ML, чтобы обеспечить вам ускорение до 40% по сравнению с Apache Spark 2.4.0.

Оптимизированные GraphFrames: запуск GraphFrames в 2-4 раза быстрее и ускорение запросов Graph в 100 раз, в зависимости от рабочих нагрузок и перекоса данных.

Оптимизированное хранилище для рабочих нагрузок глубокого обучения. Используйте высокопроизводительные решения в Azure и AWS для загрузки данных и определения контрольных точек моделей, которые имеют решающее значение для рабочих нагрузок глубокого обучения.

MLflow



Платформа с открытым исходным кодом для жизненного цикла машинного обучения



РАБОТАЕТ С ЛЮБОЙ
БИБЛИОТЕКОЙ ML,
ЯЗЫКОМ И
СУЩЕСТВУЮЩИМ КОДОМ



РАБОТАЕТ ТАК ЖЕ В
ЛЮБОМ ОБЛАКЕ



ПРЕДНАЗНАЧЕН ДЛЯ
МАСШТАБИРОВАНИЯ ОТ 1
ПОЛЬЗОВАТЕЛЯ ДО
БОЛЬШИХ ОРГАНИЗАЦИЙ



МАСШТАБИРУЕТСЯ НА
БОЛЬШИЕ ДАННЫЕ С
APACHE SPARK™

MLflow - это платформа с открытым исходным кодом для управления жизненным циклом ML, включая эксперименты, воспроизводимость и развертывание. В настоящее время он предлагает три компонента:

Отслеживание MLflow

Запись и запрос экспериментов: код, данные, настройки и результаты.

Проекты MLflow

Формат упаковки для воспроизведения воспроизводится на любой платформе.

Модели MLflow

Общий формат для отправки моделей в различные инструменты развертывания.

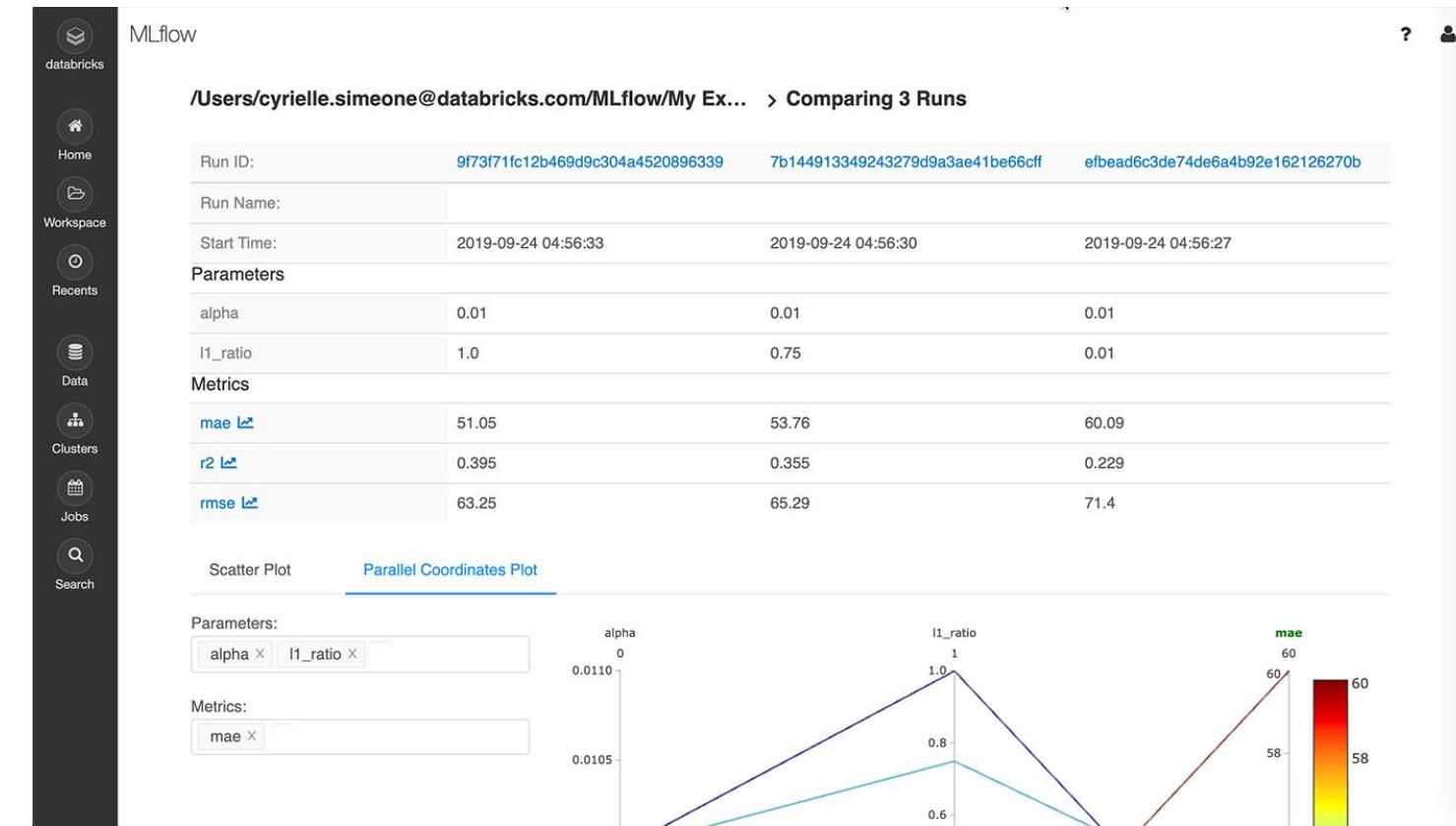
Встроенные интеграции:



Amazon SageMaker



Управляемый MLflow построен на основе [MLflow](#), платформы с открытым исходным кодом, разработанной Databricks для управления полным жизненным циклом машинного обучения с надежностью, безопасностью и масштабом предприятия.



ОТСЛЕЖИВАНИЕ ЭКСПЕРИМЕНТА

Проводите эксперименты с любой библиотекой ML, структурой или языком и автоматически отслеживайте параметры, метрики, код и модели из каждого эксперимента. С Managed MLflow для Databricks вы можете безопасно отслеживать, совместно использовать, визуализировать и управлять экспериментами из рабочей области Databricks и ноутбуков.



Registered Models

Name	Latest Version	Staging	Production	Last Modified
AaronModel	Version 5	—	Version 1	2019-10-11 15:30:02
Airline_Delay_Scikit	Version 3	—	Version 1	2019-10-11 12:41:43
Airline_Delay_SparkML	Version 5	—	Version 5	2019-10-11 12:45:15
BertlsLarge	Version 1	—	—	2019-10-11 15:18:05
holland-forecast-model	Version 1	—	Version 1	2019-10-07 15:38:27

search model name

< 1 2 3 >

МОДЕЛЬ УПРАВЛЕНИЯ
Используйте одно централизованное место для обмена моделями ML, сотрудничайте в том, чтобы перенести их из экспериментов в онлайн-тестирование и производство, интегрировать с рабочими процессами утверждения и управления, а также отслеживать развертывания ML и их производительность.

MLflow Quick Start Notebook: Packaging and Deploying Models (Python)

Cmd 8

```
1 import mlflow.pyfunc
2 pyfunc_udf = mlflow.pyfunc.spark_udf(spark, "model", run_id="7f38cd1ca0ea4660804b17c4575c53cf")
```

Command took 0.69 seconds -- by cyrielle.simeone@databricks.com at 9/24/2019, 4:56:44 AM on ben-ml

Cmd 9

```
1 predicted_df = dataframe.withColumn("prediction", pyfunc_udf(
2     'age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'))
3 display(predicted_df)
```

▶ (3) Spark Jobs

predicted_df: pyspark.sql.dataframe.DataFrame

age	sex	bmi	bp	s1	s2
0.0380759064334241	0.0506801187398187	0.0616962065186885	0.0218723549949558	-0.0442234984244464	-0.0348207628376986
-0.00188201652779104	-0.044641636506989	-0.0514740612388061	-0.0263278347173518	-0.00844872411121698	-0.019163339748222
0.0852989062966783	0.0506801187398187	0.0444512133365941	-0.00567061055493425	-0.0455994512826475	-0.0341944659141195
-0.0890629393522603	-0.044641636506989	-0.0115950145052127	-0.0366564467985606	0.0121905687618	0.0249905933641021
0.00538306037424807	-0.044641636506989	-0.0363846922044735	0.0218723549949558	0.00393485161259318	0.0155961395104161
-0.0926954778032799	-0.044641636506989	-0.0406959404999971	-0.0194420933298793	-0.0689906498720667	-0.0792878444118122

Schedule job + New

MLflow Quick Start Not... Idle

None
Last successful run: none

ГИБКИЙ РАЗМЕЩЕНИЕ
Быстрое развертывание
производственных моделей для
пакетного вывода в Apache
SparkTM или в виде REST API с
помощью встроенной интеграции с
контейнерами Docker, Azure ML
или Amazon SageMaker. Используя
управляемый MLflow для блоков
данных, вы можете
операционализировать и
отслеживать производственные
модели с помощью планировщика
заданий Databricks и автоматически
управляемых кластеров для
масштабирования по мере
необходимости в соответствии с
потребностями бизнеса.

Отслеживание MLFLOW

Отслеживание MLflow: автоматическая регистрация параметров, версий кода, метрик и артефактов для каждого запуска с использованием Python , REST , R API и Java API

Сервер отслеживания MLflow: быстро начинайте работу со встроенным сервером отслеживания, чтобы регистрировать все опыты и эксперименты в одном месте. Никакой конфигурации не требуется на Databricks.

Управление экспериментами: создание, защита, организация, поиск и визуализация экспериментов из рабочей области с контролем доступа и поисковыми запросами.

Боковая панель прогона MLflow: автоматически отслеживает прогоны из записных книжек и делает снимок вашей записной книжки для каждого прогона, чтобы вы всегда могли вернуться к предыдущим версиям кода.

Ведение журнала данных с помощью прогонов: регистрируйте параметры, наборы данных, метрики, артефакты и многое другое в виде прогонов в локальные файлы, в базу данных, совместимую с SQLAlchemy, или удаленно на сервер отслеживания.

Интеграция с Delta Lake: отслеживайте крупномасштабные наборы данных, которые снабжают ваши модели снимками Delta Lake.

Хранилище артефактов: храните большие файлы, такие как корзины S3, общую файловую систему NFS и модели, в Amazon S3, хранилище BLOB-объектов Azure, облачное хранилище Google, SFTP-сервер, NFS и локальные пути к файлам.

MLFLOW ПРОЕКТЫ

Проекты MLflow: Проекты MLflow позволяют указать программную среду, которая используется для выполнения вашего кода. В настоящее время MLflow поддерживает следующие среды проекта: среда Conda, среда контейнера Docker и среда системы. Любой репозиторий Git или локальный каталог можно рассматривать как проект MLflow.

Дистанционный режим исполнения: Запуск MLflow Проекты из Git или локальных источников на удаленном Databricks кластеров с помощью CLI Databricks быстро масштабировать свой код.

MLFLOW МОДЕЛЬ РЕГИСТРАЦИИ

Центральный репозиторий: Зарегистрируйте модели MLflow в Реестре моделей MLflow. Зарегистрированная модель имеет уникальное имя, версию, этап и другие метаданные.

Управление версиями моделей: автоматическое отслеживание версий зарегистрированных моделей при обновлении.

Стадия модели: назначьте предустановленные или настраиваемые этапы для каждой версии модели, например «Стадия» и «Производство», для представления жизненного цикла модели.

Интеграция рабочего процесса CI / CD: Записывайте переходы этапов, запрашивайте, просматривайте и одобряйте изменения как часть конвейеров CI / CD для лучшего контроля и управления.

Переходы на этапе модели: регистрируйте новые события или изменения регистрации как действия, которые автоматически регистрируют пользователей, изменения и дополнительные метаданные, такие как комментарии.

MLFLOW МОДЕЛИ

Модели MLflow: стандартный формат для моделей машинного обучения, который может использоваться в различных последующих инструментах, например, в режиме реального времени, через API REST или пакетный вывод в Apache Spark.

Настройка модели: используйте пользовательские модели Python и пользовательские разновидности для моделей из библиотеки ML, которые явно не поддерживаются встроенными разновидностями MLflow.

Варианты встроенных моделей: MLflow предоставляет несколько стандартных вариантов, которые могут быть полезны в ваших приложениях, такие как функции Python и R, H2O, Keras, MLeap, PyTorch, Scikit-learn, Spark MLlib, TensorFlow и ONNX. Встроенные инструменты развертывания. Быстрое развертывание на блоках данных через Apache Spark UDF for, локальный компьютер или несколько других производственных сред, таких как Microsoft Azure ML, Amazon SageMaker и создание образов Docker для развертывания .

MLflow - это легкий набор API и пользовательских интерфейсов, которые можно использовать с любой платформой ML в рамках всего процесса машинного обучения. Он включает в себя четыре компонента:

Отслеживание MLflow : запись и запрос экспериментов: код, данные, конфигурация и результаты.

Проекты MLflow : формат упаковки для воспроизводимых прогонов на любой платформе.

MLflow Models : Общий формат для отправки моделей в различные инструменты развертывания.

Реестр моделей MLflow : централизованное хранилище для совместного управления моделями MLflow на протяжении всего жизненного цикла.

Управляемый MLflow для Databricks - это полностью управляемая версия MLflow, предоставляющая практическим специалистам воспроизводимость и управление экспериментами через записные книжки, задания и хранилища данных, а также надежность, безопасность и масштабируемость платформы Unified Data Analytics .

Access All Your Data



Интеграции

В дополнение к встроенным средствам запросов и визуализации SQL Analytics обеспечивает поддержку всех ваших существующих приложений бизнес-аналитики. Настроить надежные подключения к таблицам Delta Lake просто, и вы можете интегрировать существующее решение аутентификации.

Бизнес-аналитика

Доступ к более полным и своевременным данным для понимания бизнеса

Полные данные

Запрашивайте данные непосредственно для получения наиболее полной информации.

Последние данные

Получение потоковых данных для аналитики в реальном времени.

Интеграция с бизнес-аналитикой

Используйте предпочтаемые инструменты визуализации и отчетности.