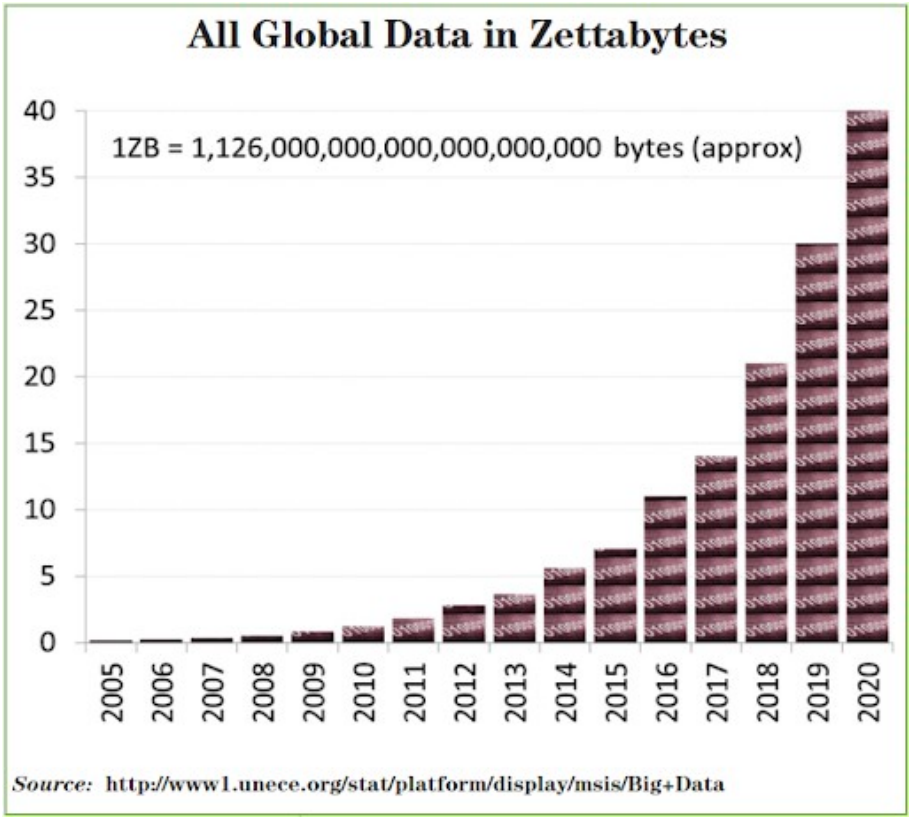
MACHINE LEARNING AND AI

ЛЕКЦИЯ 11

BIG DATA


All Global Data in Zettabytes
1ZB = 1,126,000,000,000,000,000,000 bytes (approx)
40
35
30
25
20
15
10
5
0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
Source: http://www1.unece.org/stat/platform/display/msis/Big+Data

МНОГО ДАННЫХ – ЭТО СКОЛЬКО?

640K ought to be enough
for anyone.

Bill Gates, 1981

BIG DATA – VVV

VOLUME

Объём данных

VELOCITY

Скорость прироста и генерации данных

VARIETY

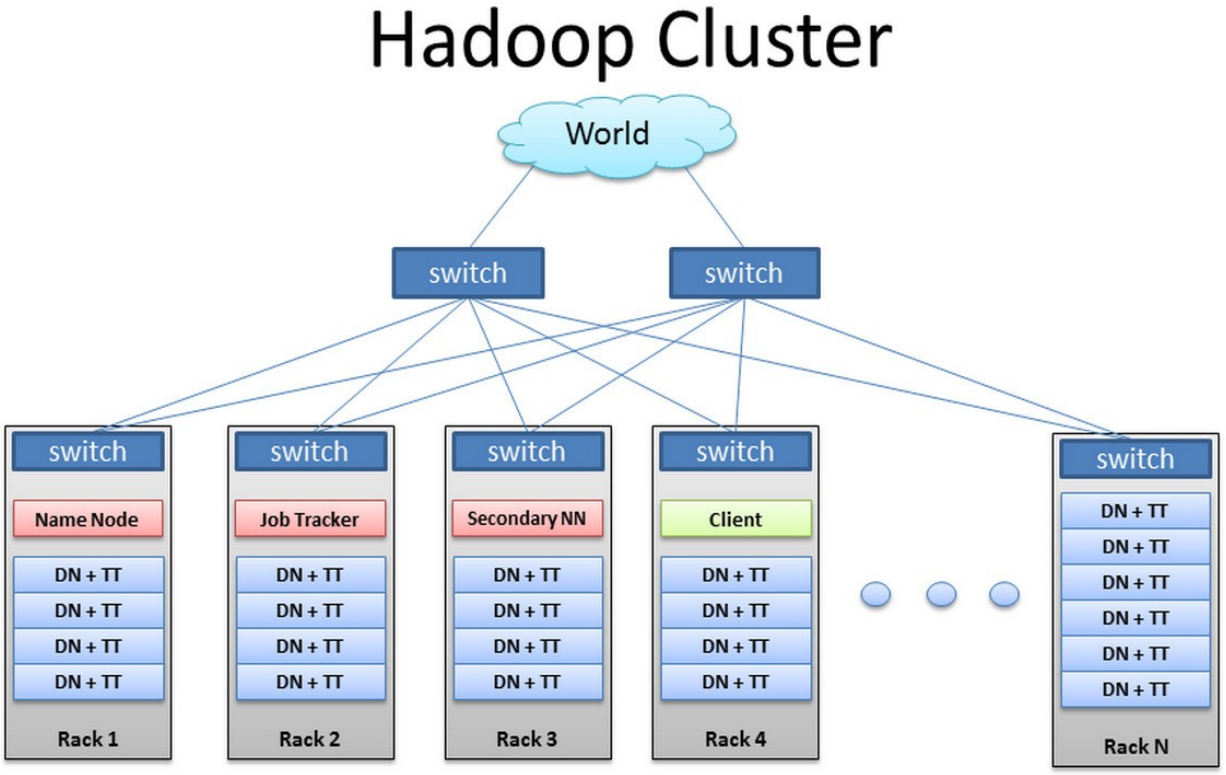
Разнообразие различных источников данных

HADOOP – СИСТЕМА ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Особенности:

- Горизонтальная масштабируемость кластера
- HDFS – распределённая файловая система
- Парадигма Hadoop MapReduce
- Наличие разнообразной экосистемы для разных типов задач
- YARN – подсистема эффективного управления ресурсами

АРХИТЕКТУРА КЛАСТЕРА HADOOP



HDFS (Hadoop Distributed File System)

HDFS (Hadoop Distributed File System) — файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера. Все блоки в HDFS (кроме последнего блока файла) имеют одинаковый размер, и каждый блок может быть размещён на нескольких узлах, размер блока и коэффициент репликации (количество узлов, на которых должен быть размещён каждый блок) определяются в настройках на уровне файла. Благодаря репликации обеспечивается устойчивость распределённой системы к отказам отдельных узлов.

СТРУКТУРА HDFS

Name Node: HDFS состоит только из одного **Name Node**, который называется **Master Node**. **Master Node** может отслеживать файлы, управлять файловой системой и иметь метаданные всех хранимых в нем данных. В частности, узел имени содержит сведения о количестве блоков, расположении узла данных, в котором хранятся данные, где хранятся репликации, и другие подробности.

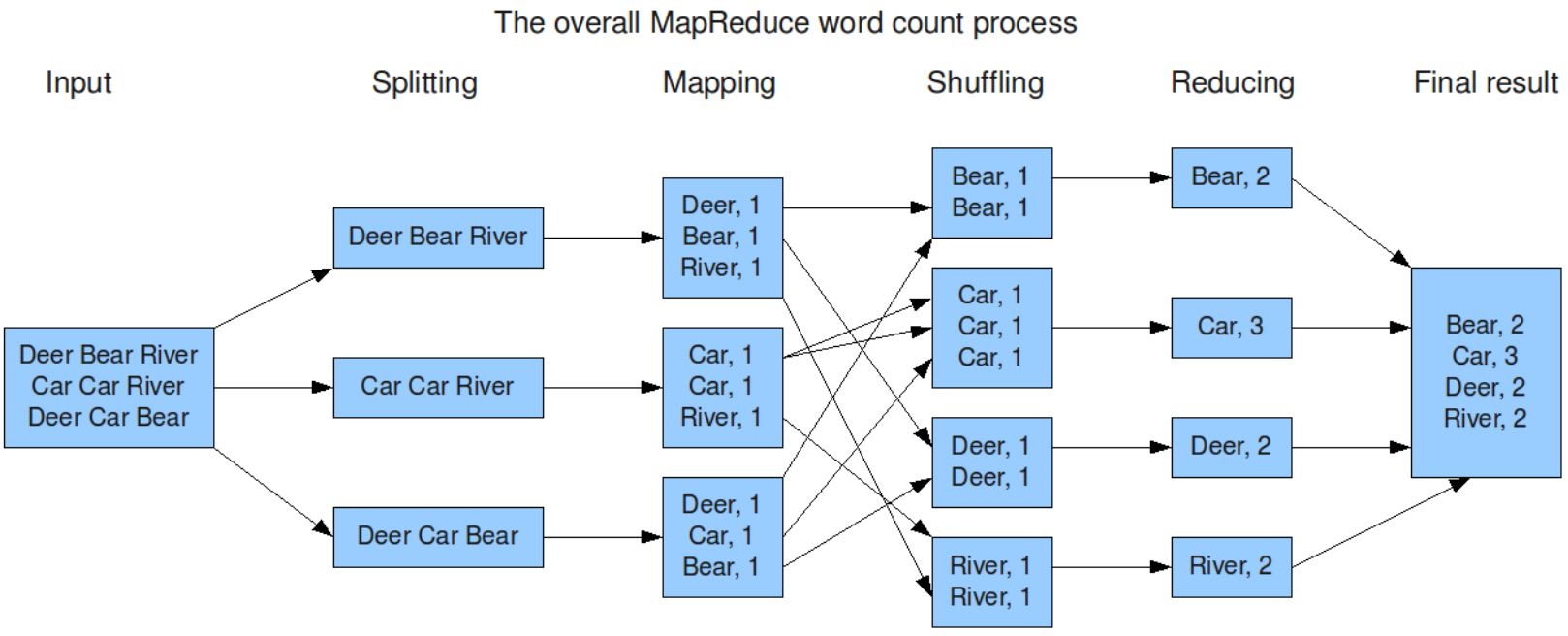
Data Node: хранит данные в виде блоков. Каждый узел данных отправляет сообщение **Heartbeat** узлу **Name Node** каждые 3 секунды и сообщает, что он активен. Таким образом, когда **Name Node** не получает **Heartbeat** от узла данных в течение 2 минут, он считает этот **Data Node** мертвым и запускает процесс репликации блока на каком-то другом **Data Node**.

Secondary Name Node: нужен для хранения метаданных файловой системы, которые находятся в **Name Node**. Это резервный узел для **Name Node.**

Job Tracker: средство отслеживания задач получает от клиента запросы на выполнение **Map Reduce**. **Job Tracker** обращается к **Name Node**, чтобы узнать о местонахождении данных, которые будут использоваться при обработке. **Name Node** отвечает метаданными необходимых данных обработки.

Task Tracker: это подчиненный узел для **Job Tracker**, и он берет задачу из **Job Tracker**. Он также получает код от **Job Tracker**. **Task Tracker** исполняет код применительно к конкретному файлу. Процесс исполнения этого кода к файлу называется **Mapper**.

MAP REDUCE


The overall MapReduce word count process
Input
Splitting
Mapping
Shuffling
Reducing
Final result
Deer Bear River
Car Car River
Deer Car Bear
Deer Bear River
Car Car River
Deer Car Bear
Deer, 1
Bear, 1
River, 1
Car, 1
Car, 1
River, 1
Deer, 1
Car, 1
Bear, 1
Bear, 1
Bear, 1
Car, 1
Car, 1
Car, 1
Deer, 1
Deer, 1
River, 1
River, 1
Bear, 2
Car, 3
Deer, 2
River, 2
Bear, 2
Car, 3
Deer, 2
River, 2

HADOOP STREAMING

Hadoop streaming

это утилита, входящая в состав дистрибутива Hadoop. Утилита позволяет создавать и запускать задачи Map/Reduce с любым исполняемым файлом или скриптом в качестве маппера и/или редьюсера.

ПРИМЕР РЕШЕНИЯ ЗАДАЧИ WORD COUNT

MAPPER

```
import sys

for line in sys.stdin:
    words = line.split()
    for word in words:
        print(word, 1)
```

REDUCER

```
import sys

current_word = None
current_value = None
for line in sys.stdin:
    word, value = line.split('\t')
    if word != current_word:
        if current_word:
            print(current_word,current_value)

        current_word = word
        current_value = 1
    else:
        current_value += int(value)

if current_word:
    print(current_word,current_value)
```

ДРУГАЯ ЗАДАЧА

Есть выборка транзакций

Нужно определить количество транзакций за каждый день

user_ID	transaction_date	amount
1	2020-04-01	110
2	2020-04-01	120
1	2020-04-01	90
2	2020-04-02	30
3	2020-04-02	25
4	2020-04-02	400
1	2020-04-02	152
2	2020-04-03	293
3	2020-04-03	91

ЭКОСИСТЕМА HADOOP

Состав экосистемы

Pig (свинья)

Hive (улей)

Impala (импала)

Spark (искра)

Cassandra (Кассандра)

Oozie

Sqoop

Kafka

Zookeeper

PIG

Apache Pig - это платформа высокого уровня для создания программ, работающих на Apache Hadoop. Язык этой платформы называется **Pig Latin**. **Pig** может выполнять задачи Hadoop в MapReduce или Apache Spark.

Pig Latin абстрагирует программирование от идиомы Java MapReduce до нотации, которая делает программирование MapReduce высоким уровнем, аналогичным уровню SQL для систем управления реляционными базами данных.

Pig Latin может быть расширен с помощью пользовательских функций (UDF), которые пользователь может писать на Java, Python, JavaScript, Ruby или Groovy, а затем вызывать непосредственно из языка.

HIVE

Apache Hive — система управления базами данных на основе платформы Hadoop. Позволяет выполнять запросы, агрегировать и анализировать данные, хранящиеся в Hadoop.

- Работа с данными используя SQL-подобный язык запросов;
- Поддержка различных форматов хранения данных;
- Работа напрямую с HDFS и Apache HBase;
- Выполнение запросов через Apache Tez, Apache Spark или MapReduce

IMPALA

Impala предлагается аналитикам и специалистам по обработке данных для анализа данных, хранящихся в Hadoop, с помощью SQL или инструментов бизнес-аналитики. В результате крупномасштабная обработка данных (через MapReduce) и интерактивные запросы могут выполняться в одной и той же системе с использованием одних и тех же данных и метаданных, что устраняет необходимость в переносе наборов данных в специализированные системы и/или собственные форматы просто для выполнения анализа.

Возможности включают:

Читает форматы файлов Hadoop, включая текст, LZO, SequenceFile, Avro, RCFile, Parquet и ORC

Использует метаданные и синтаксис SQL из Apache Hive.

SPARK

Apache Spark — фреймворк с открытым исходным кодом для реализации распределённой обработки неструктурированных и слабоструктурированных данных, входящий в экосистему проектов Hadoop. В отличие от классического обработчика из ядра Hadoop, реализующего двухуровневую концепцию MapReduce с хранением промежуточных данных на накопителях, Spark работает в парадигме ленивых вычислений (англ. in-memory computing) — обрабатывает данные 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.

Проект предоставляет программные интерфейсы для языков Java, Scala, Python, R.

CASSANDRA

Apache Cassandra — распределённая система управления базами данных, относящаяся к классу NoSQL-систем и рассчитанная на создание высокомасштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша.

Требование к реляционным БД – ACID

Atomicity – атомарность гарантирует, что никакая транзакция не будет зафиксирована в системе частично.

Consistency – транзакция, достигающая своего нормального завершения и тем самым фиксирующая свои результаты, сохраняет согласованность базы данных.

Isolation – во время выполнения транзакции параллельные транзакции не должны оказывать влияния на её результат.

Durability – независимо от проблем на нижних уровнях (к примеру, обесточивание системы или сбои в оборудовании) изменения, сделанные успешно завершённой транзакцией, должны остаться сохранёнными после возвращения системы в работу.

KAFKA

Apache Kafka — распределённый программный брокер сообщений, проект с открытым исходным кодом, разрабатываемый в рамках фонда Apache. Написан на языках программирования Java и Scala.

Спроектирован как распределённая, горизонтально масштабируемая система, обеспечивающая наращивание пропускной способности как при росте числа и нагрузки со стороны источников, так и количества систем-подписчиков. Подписчики могут быть объединены в группы. Поддерживается возможность временного хранения данных для последующей пакетной обработки. Одной из особенностей реализации инструмента является применение техники, сходной с журналами транзакций, используемыми в системах управления базами данных.