Доклад

на тему: Big Date (Большие данные)

Выполнил: студент I курса магистратуры
направления «Прикладная информатика»
Нестерович А.А.

Проверил:
ст. преподаватель Глазов А.Б.

Большие данные — совокупность подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети, альтернативных традиционным системам управления базами данных и решениям класса **Business Intelligence** .

- NoSQL

  NoSQL в информатике — термин, обозначающий ряд подходов, направленных на реализацию хранилищ баз данных, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.
- MapReduse

  MapReduce — модель распределённых вычислений, представленная компанией Google, используемая для параллельных вычислений над очень большими наборами данных в компьютерных кластерах.
- Hadoop

  Hadoop — проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов.

Введение термина «большие данные» относят к Клиффорду Линчу, редактору журнала Nature, 3 сентября 2008 года

В 2011 году Gartner (исследовательская и консалтинговая компания, специализирующаяся на рынках информационных технологий.) отмечает большие данные как тренд номер два в информационно-технологической инфраструктуре (после виртуализации).

Существуют разные определения больших данных, но большинство из них базируется на концепции «**трех V**» больших данных:

- **Объем** (Volume)
- **Разнообразие** (Variety)
- **Скорость** (Velocity)

В большинстве случаев работа с большими данными подразумевает стандартный рабочий процесс: от сбора необработанных данных и до получения пригодной для использования информации.

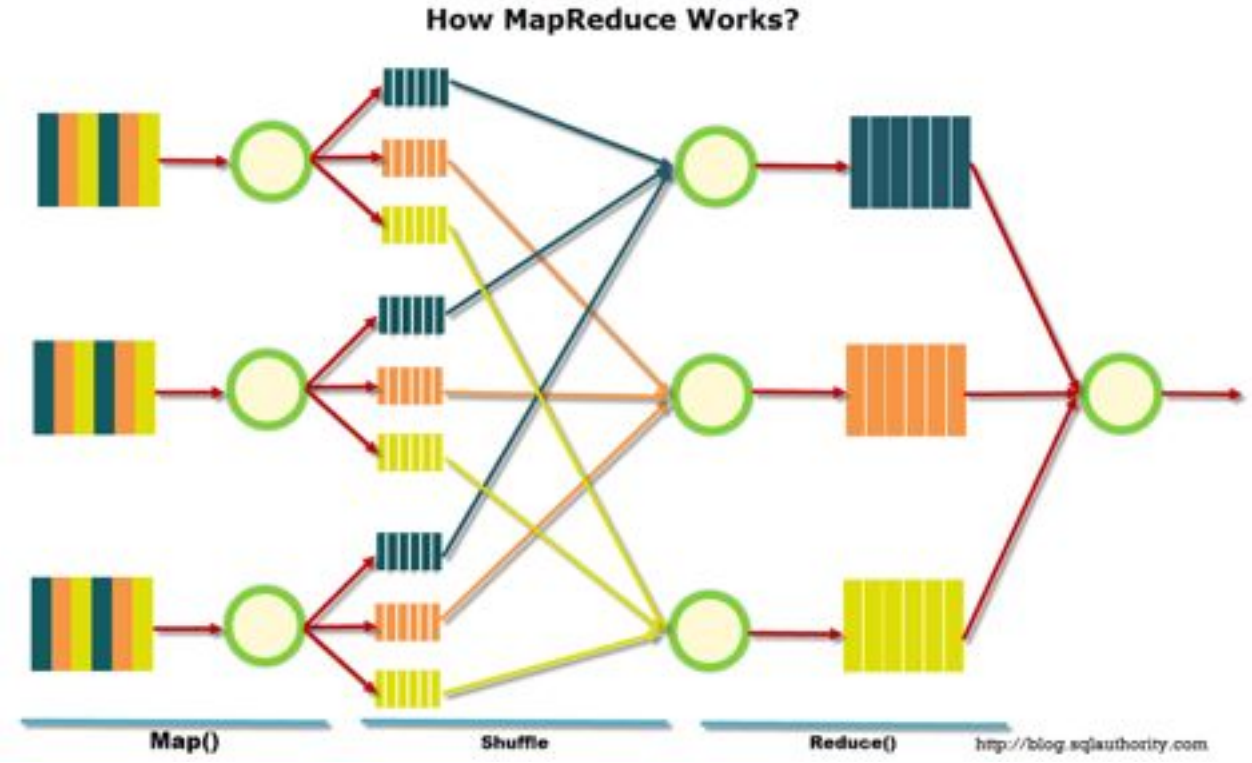
- Сбор. Сбор необработанных данных
  - Хранение. Любая платформа для работы с большими данными должна включать надежный, безопасный и масштабируемый репозиторий для хранения данных как до обработки, так и после таковой.
- Обработка и анализ достигается за счет сортировки, агрегации, объединения или применения специальных расширенных функций и алгоритмов
  - Визуализация и использование. Основная цель работы с большими данными – это получение на их основании ценных аналитических выводов для практического применения.

Принципы работы с большими данными

- 1. Горизонтальная масштабируемость
- 2.Отказоустойчивость
- 3.Локальность данных

Все современные средства работы с большими данными так или иначе следуют этим трём принципам.

MapReduce предполагает, что данные организованы в виде некоторых записей. Обработка данных происходит в 3 стадии:



1. Стадия Map.
2. Стадия Shuffle.
3. Стадия Reduce.

Примеры задач, эффективно решаемых при помощи MapReduce

Word Count

Имеется большой корпус документов. Задача – для каждого слова, хотя бы один раз встречающегося в корпусе, посчитать суммарное количество раз, которое оно встретилось в корпусе.

Решение:

Функция map превращает входной документ в набор пар (слово, 1);

shuffle прозрачно для нас превращает это в пары (слово, [1,1,1,1,1,1]);

reduce суммирует эти единички, возвращая финальный ответ для слова.

```
def map(doc):
    for word in doc:
        yield word, 1
```

```
def reduce(word, values):
    yield word, sum(values)
```

Обработка логов рекламной системы

Второй пример взят из реальной практики Data-Centric Alliance.

Задача: имеется csv-лог рекламной системы вида:
<user_id>,<country>,<city>,<campaign_id>,<creative_id>,<payment></p>

11111,RU,Moscow,2,4,0.3

22222,RU,Voronezh,2,3,0.2

13413,UA,Kiev,4,11,0.7

Необходимо рассчитать среднюю стоимость показа рекламы по городам России.

Решение:

```
def map(record):
    user_id, country, city,
campaign_id, creative_id,
payment = record.split(",")
payment=float(payment)
if country == "RU": yield city,
payment
```

```
def reduce(city, payments):
yield city,
sum(payments)/len(payments)
```