Проектирование баз данных, ч.2

ФИО преподавателя: Смирнов Михаил Вячеславович

e-mail: smirnovmgupi@gmail.com

Лекция 11

Средства извлечения и обработки Больших данных (BigData)

5V’s of Big Data. Volume (объем)

- Для каждой организации или компании существует предел объема данных (**Volume**) которые компания или организация способна обрабатывать одновременно для целей аналитики.
- Как правило этот объем ограничен объемами оперативной памяти серверов корпоративных приложений и баз данных и необходимостью партиционирования (Partitioning) хранимых данных.

- #терабайты #таблицы #файлы #распределение

5V’s of Big Data. Velocity (скорость, быстрота)

- Для каждой организации или компании существуют физические ограничения на количество транзакций/объем данных (**Velocity**), которая корпоративная система может обработать или передать за единицу времени вследствие ограничений (**scale in)** архитектуры этой системы.

- #пакеты #процессы #потоки #реальное/оценочное время

5V’s of Big Data. Variety (разнообразие)

- Традиционные корпоративные системы (построенные на реляционных моделях) могут использовать эффективно только структурированные источники поступления информации, не принимая во внимание разновариантные и неструктурированные источники данных (**Variety**), или имея серьезные ограничения по работе с такими источниками.

- #структурированность #неструктурированность #вероятностность #связанность #динамика

5V’s of Big Data. Veracity (точность)

- Большое количество данных и разнообразие источников требует качества и аккуратности при обработке и анализе данных (твиты, хэштэги, аббревиатуры, сокращения и т.д.). Ошибки, надежность и точность контента ставят под сомнение достоверность (**Veracity**) самих данных так и принятых решений на основе этих данных. Количество не переходит в качество.

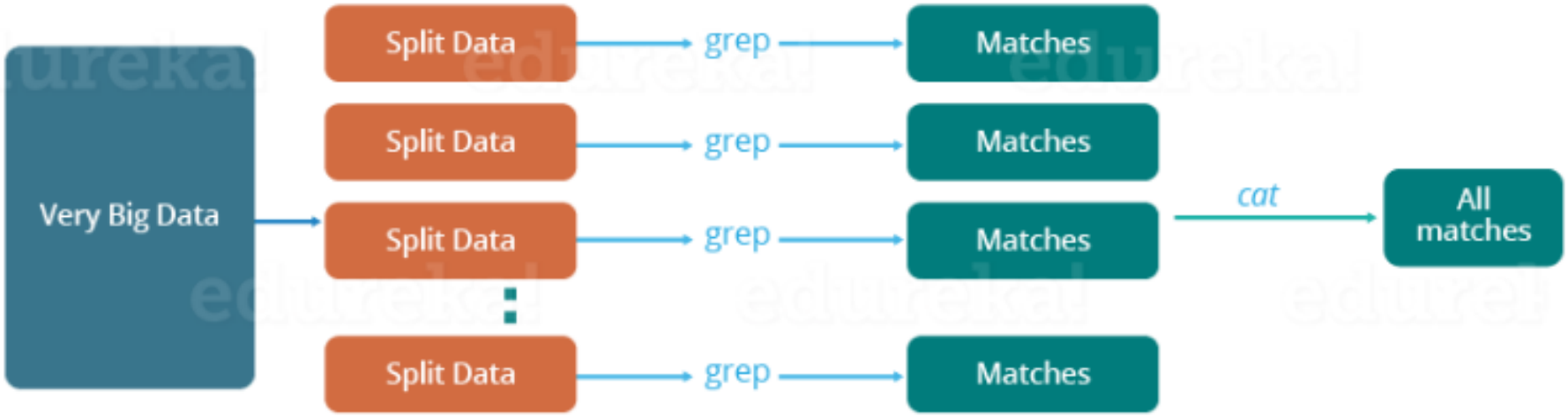
- #достоверность #аутентичность #публичность #доступность

5V’s of Big Data. Value (ценность)

- Сбор и анализ больших данных должен предоставлять определенную ценность (**Value**) для бизнеса. Ценность данных неразрывна связанна со стоимость владения и ценностью для бизнеса.

- #корреляция #гипотезы #статистика

Традиционный способ обработки данных*

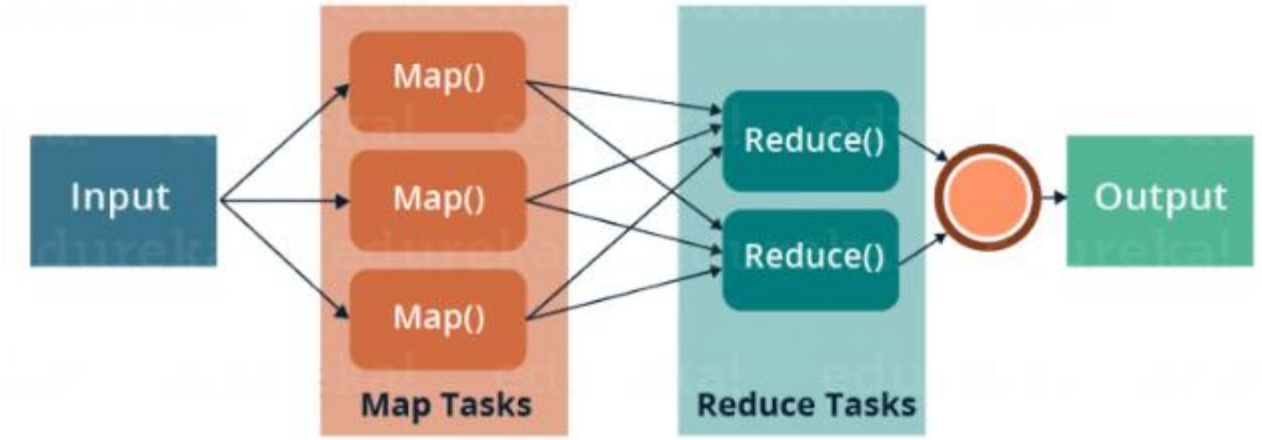


Проблемы традиционной обработки большого объема данных:

- проблема надежности кластера
- вопрос равномерного разделения данных
- проблема надежности “разделки” (split)
- проблема агрегации результата

*Ссылка на статью по теме (англ): https://thirdeyedata.io/hadoop-mapreduce/

Обработка данных через фреймворк MapReduce



Суть метода:

- две разделенные задачи – map, reduce
- map – разложение входных массивов на пары ключ-значение в промежуточный массив
- передача массива на reducer
- агрегация промежуточных пар ключ-значение в результатный массив

Классы фреймворка MapReduce

- Mapper class. Задача – разделить входящий поток данных на пары ключ-значение, заданные исследователем.

  Функции:

  - Input Split – задача в программе

  - RecordReader – разбиение массива на пары ключ-значение

- Reducer class.

  Функция: Формирование выходного массива в формате HDFS (Hadoop Distributed File System).

- Driver class.

  Функция – выполнение указанных выше классов в Hadoop.

Код mapper

```
public static class Map extends Mapper<LongWritable,Text,Text,IntWritable>
<< определяем входные ключи, строки текста, выходные ключи, выходные целочисленные значения
{public void map(LongWritable key, Text value, Context context) throws IOException,InterruptedException {
<< размечаем пары ключ-значение, пишем результат в Context
String line = value.toString();
StringTokenizer tokenizer = new StringTokenizer(line);
<< конвертируем результат в строку, разбиваем строку по пробелам на слова
while (tokenizer.hasMoreTokens()) {
value.set(tokenizer.nextToken());
context.write(value, new IntWritable(1));}
<< для каждого значения в результате берем слово и в качестве ключа присваиваем ему единицу
```

Код reducer

```
public static class Reduce extends Reducer<Text,IntWritable,Text,IntWritable> {
```

<< определяем входные ключи, единички от mapper, выходные ключи, выходные суммы

```
public void reduce(Text key, Iterable<IntWritable> values,Context context)
throws IOException,InterruptedException {
```

<< В контексте, для каждого уникального слова формируется коллекция всех единиц для этого слова, iterable – работа с массивом, без загрузки в память целиком, важно!!

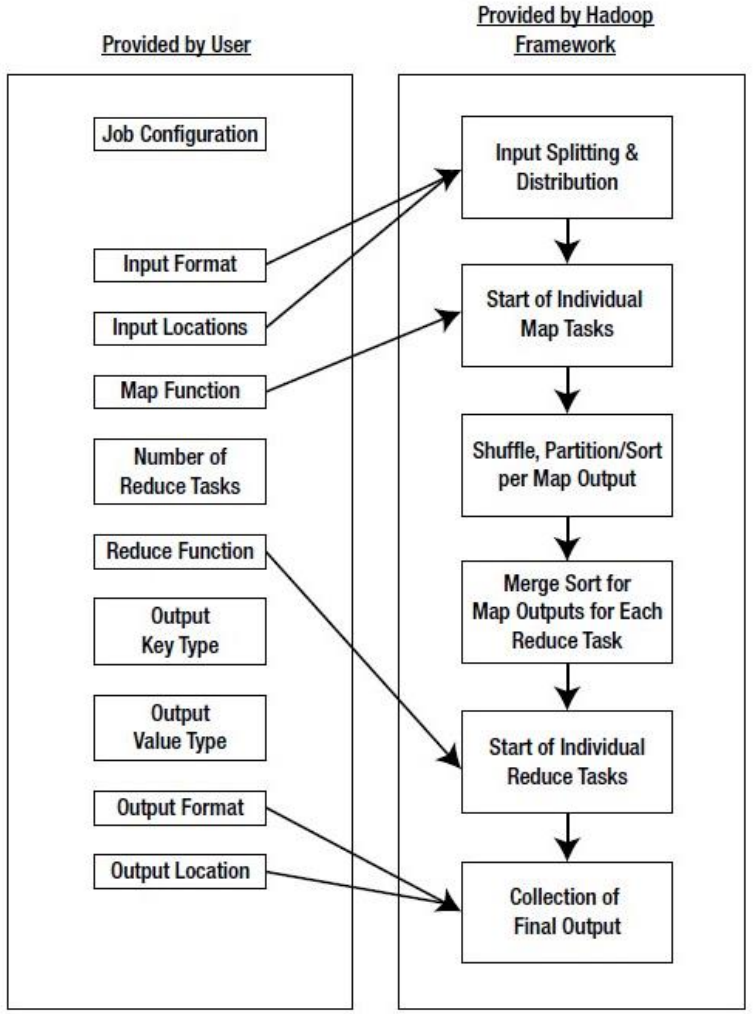
```
int sum=0;
for(IntWritable x: values)
{sum+=x.get();}
context.write(key, new IntWritable(sum)); }}
```

<< Сбрасываем счетчик, для каждой единицы в коллекции (уникальное слово) суммируем, преобразовывая в int и пишем результат в контекст.

Код driver

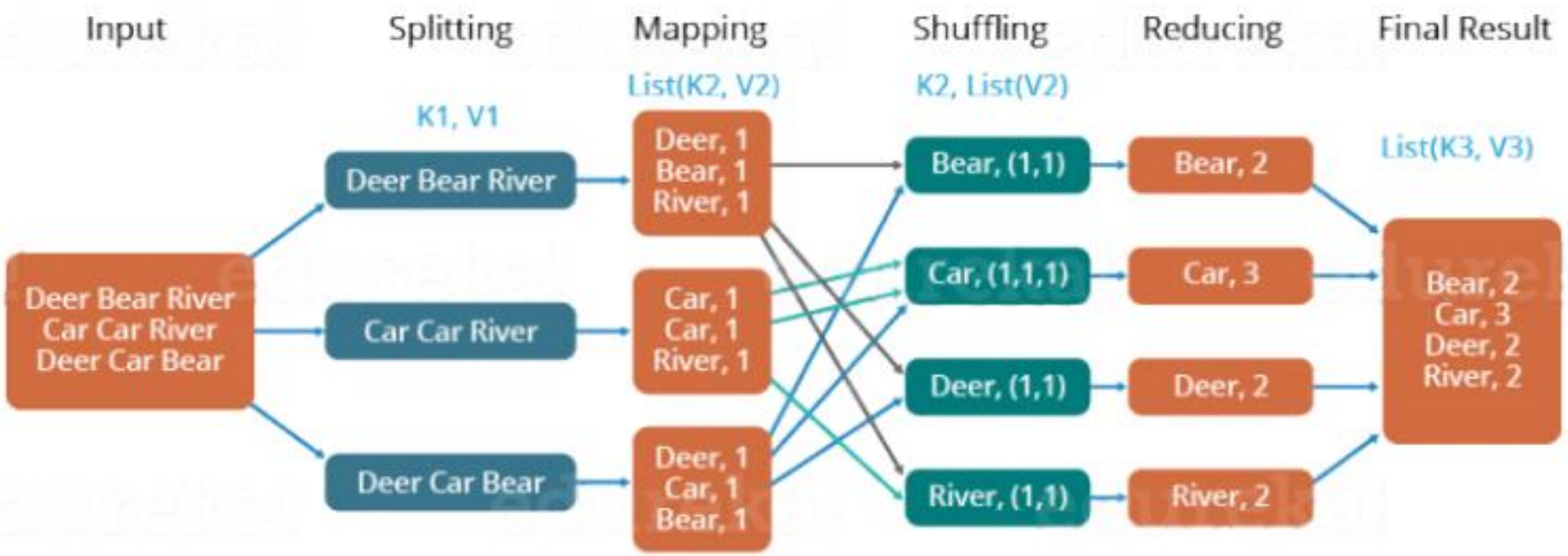
```
Configuration conf= new Configuration();
Job job = new Job(conf,"My Word Count Program");
<< сохраняем настройки Hadoop, создаем новую задачу hadoop
job.setJarByClass(WordCount.class);
job.setMapperClass(Map.class);
job.setReducerClass(Reduce.class);
<< создаем классы для поиска файла с данными, классы mapper и reducer
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
<< определяем типы выходных значений, формат входных и выходных данных
Path outputPath = new Path(args[1]);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
<< определяем входную и выходную директорию для файлов
```

Архитектура MapReduce

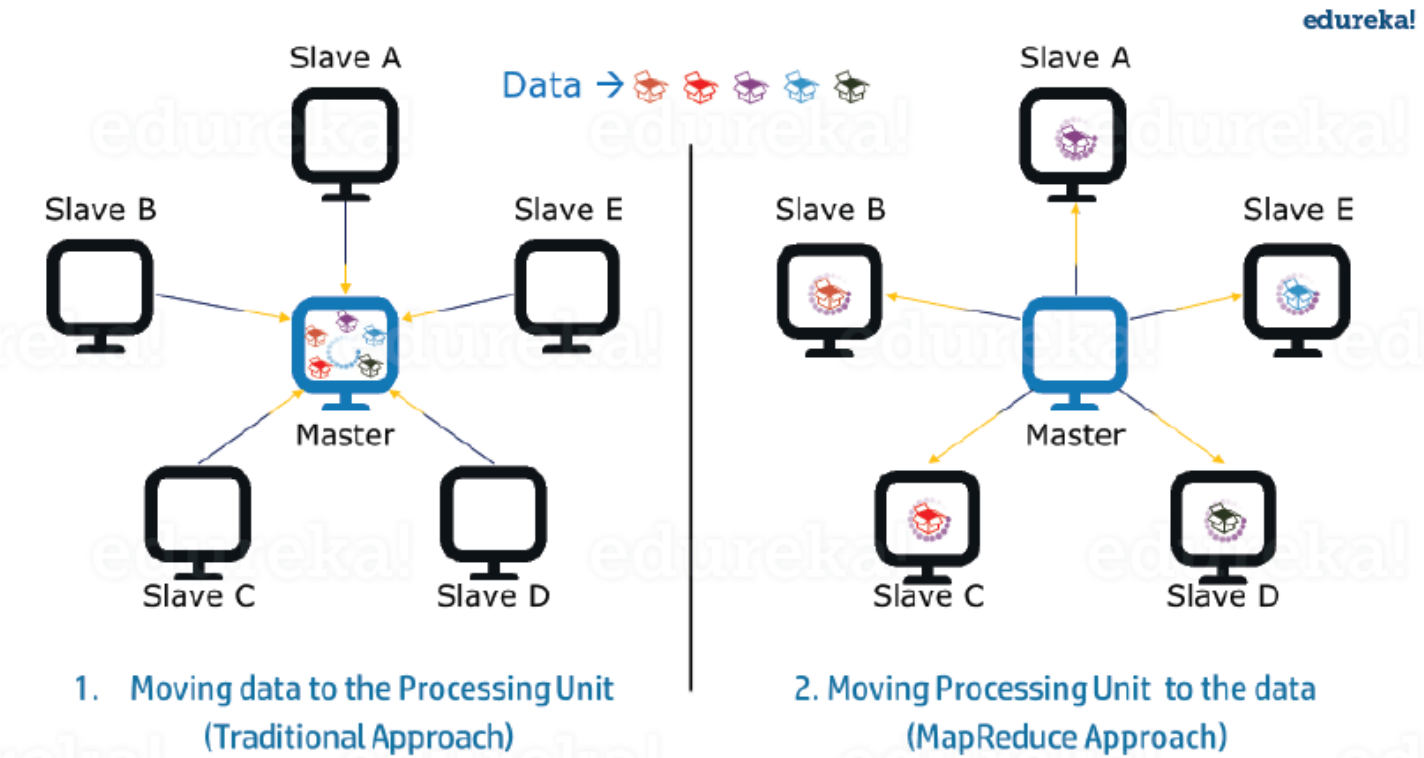


Пример реализации классов фреймворка mapreduce

Задача. Разбор текстового файла с содержимым **Dear, Bear, River, Car, Car, River, Deer, Car and Bear.** Визуализация задачи:



Параллельная обработка в MR, локальность данных



Простейший MapReduce в Python и PHP*

```python
words = ["foo", "bar", "baz"]
def map1(word):
  return [word, 1]

arr = ["foo", [1,1]]
def reduce1(arr):
  return [ arr[0], sum(arr[1]) ]
```

```php
$words = array("foo", "bar", "baz")
function map1($word) {
  return array($word, 1);
}

arr = array("foo", array(1,1))
function reduce1(arr) {
  return array( $arr[0], array_sum($arr[1]) );
}
```

*Ссылка на статью MapReduce без зауми на русском: https://habr.com/ru/post/103467/

Спасибо за внимание!