

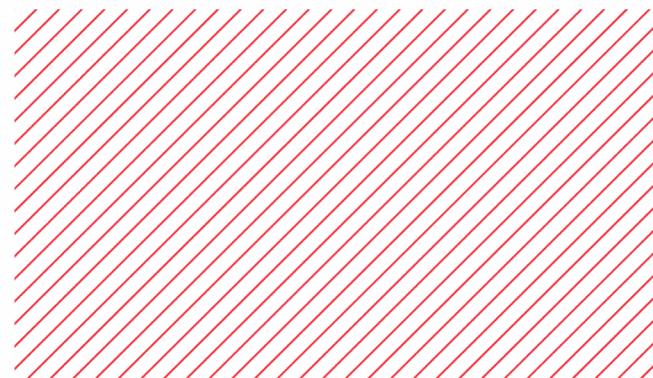
академия
больших
данных



Hadoop экосистема и MapReduce

Дмитрий Музалевский

26.09.2020

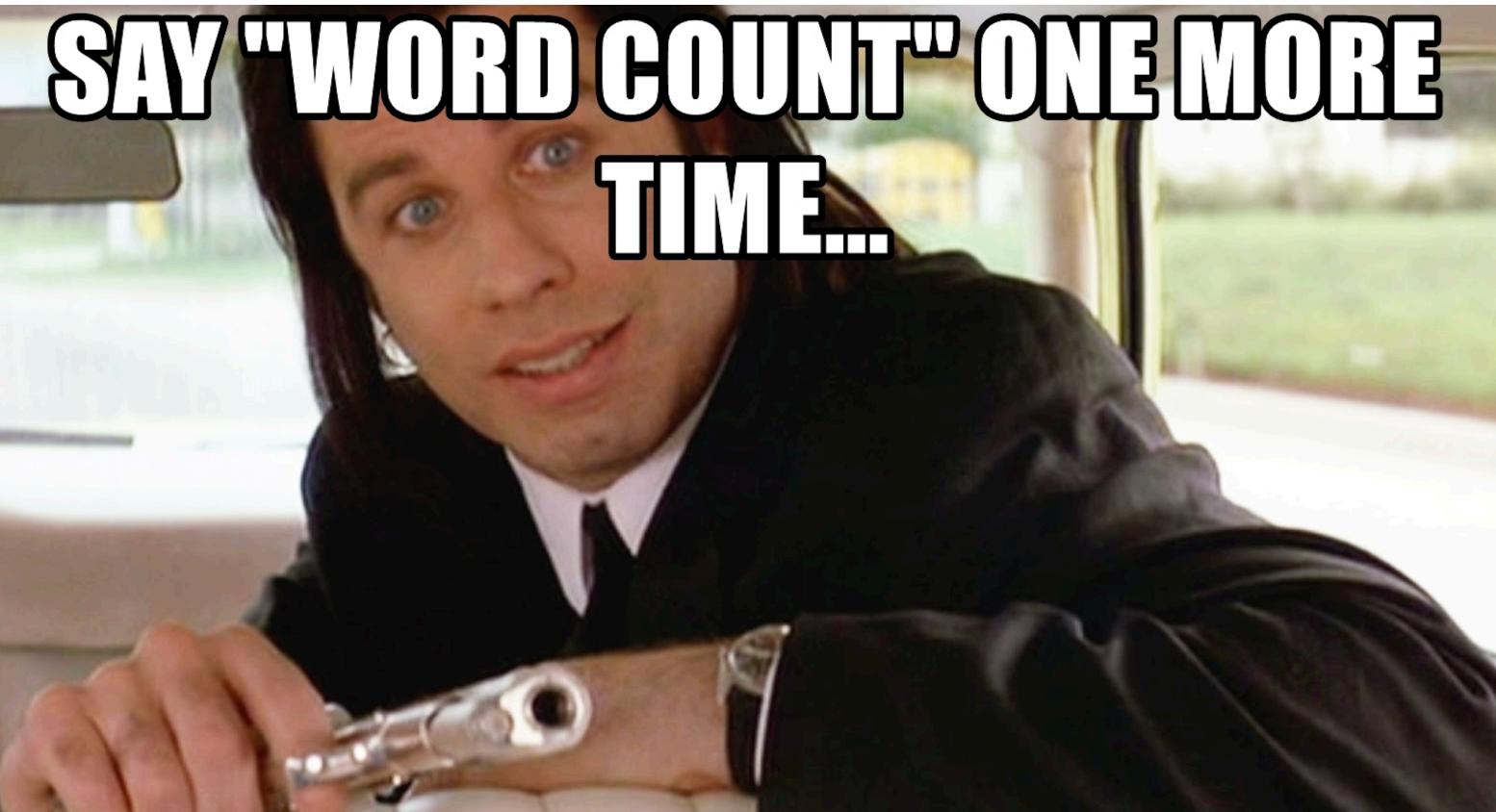




План занятия

1. WordCount
2. MapReduce в Python
3. Практика 1: MapReduce в Java
4. Практика 2: Запуск как step в EMR

Word Count





Word Count

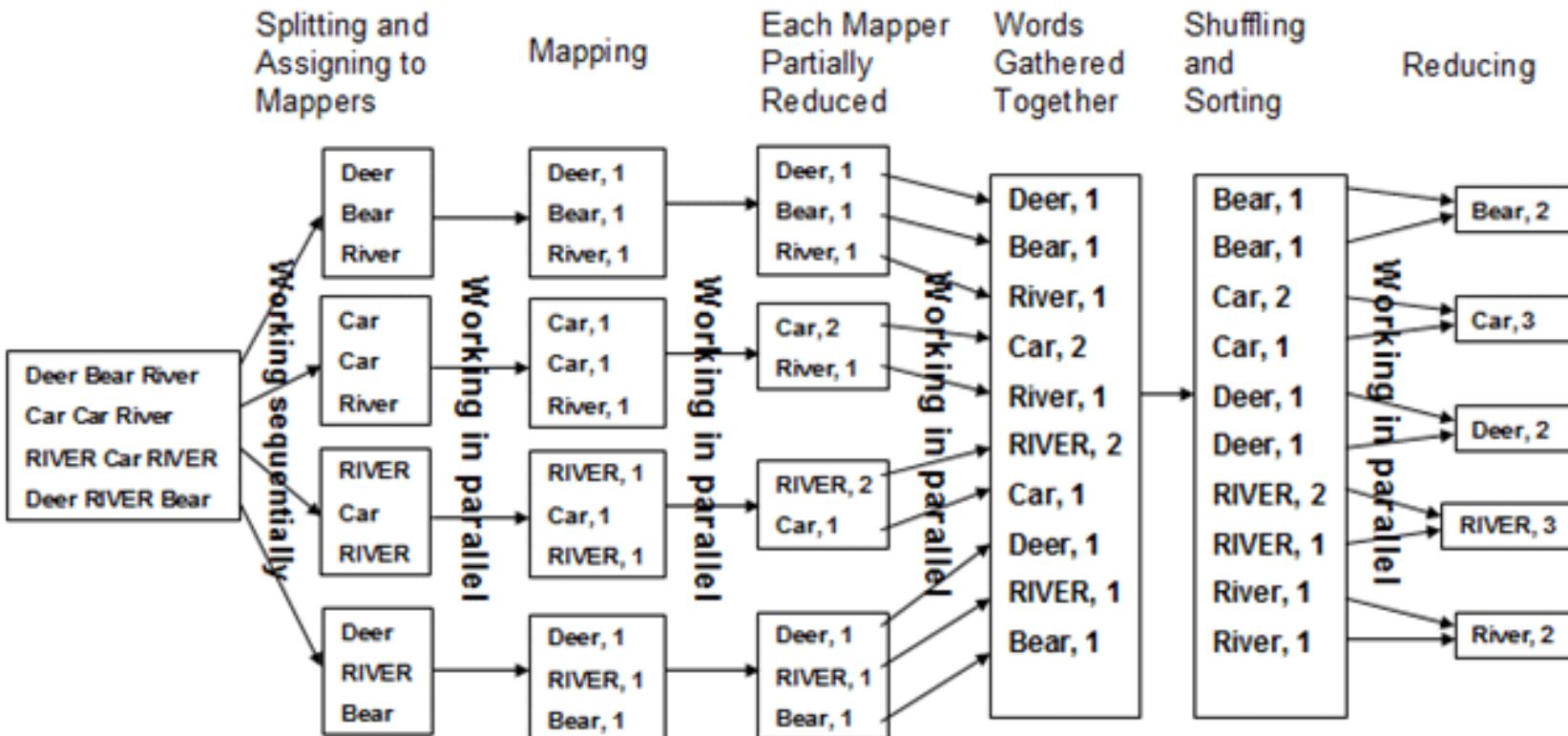
WordCount (Подсчет Слов) - это процесс подсчета того, сколько раз отдельные слова встречаются в тексте.

Input является текстовым файлом, output также является текстовым файлом, каждая строчка которого содержит слово и подсчет (count) того, насколько часто это слово встречается в тексте.

Каждый mapper берет отдельную строчку текста и разделяет ее на слова. Затем выделяет пару "ключ-значение" и 1. Затем reducer суммирует количество отдельных слов и выделяет пару "ключ-значение" и общую сумму.

Word Count

Word Count with four Mappers/Splits





MapReduce (Python): Mapper

```
"""mapper.py"""

import sys

# Стандартный input
for line in sys.stdin:
    # удаление пробелов
    line = line.strip()
    # разделение строчки в слова
    words = line.split()
    # increase counters
    for word in words:
        # пишем результаты
        # результат mapper будет данными для reducer
        # пара ключ-значение
        print '%s\t%s' % (word, 1)
```

MapReduce (Python): Reducer

```
"""reducer.py"""

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# Стандартный input
for line in sys.stdin:
    # удаление пробелов
    line = line.strip()

    # парсинг данных из mapper.py
    word, count = line.split('\t', 1)

    # перевод count (string) в int
    try:
        count = int(count)
    except ValueError:
        continue

    # hadoop сортирует результат работы map
    # по ключу (слово) перед тем как он подается на reducer
    if current_word == word:
        current_count += count
    else:
        if current_word:
            # записываем результат
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

    if current_word == word:
        print '%s\t%s' % (current_word, current_count)
```



MapReduce (Python)

```
hadoop@ubuntu:/usr/local/hadoop$ bin/hadoop jar contrib/streaming/hadoop-*streaming*.jar -mapper  
/home/hadoop/mapper.py -reducer /home/hadoop/reducer.py -input /user/hadoop/texts/henry.txt -output  
/user/hadoop/texts-output/henry.txt  
additionalConfSpec_:null  
null=@@@@userJobConfProps_.get(stream.shipped.hadoopstreaming  
packageJobJar: [/app/hadoop/tmp/hadoop-unjar54543/]  
[] /tmp/streamjob54544.jar tmpDir=null  
[...] INFO mapred.FileInputFormat: Total input paths to process : 7  
[...] INFO streaming.StreamJob: getLocalDirs(): [/app/hadoop/tmp/mapred/local]  
[...] INFO streaming.StreamJob: Running job: job_200803031615_0021  
[...] INFO streaming.StreamJob: map 0% reduce 0%  
[...] INFO streaming.StreamJob: map 43% reduce 0%  
[...] INFO streaming.StreamJob: map 86% reduce 0%  
[...] INFO streaming.StreamJob: map 100% reduce 0%  
[...] INFO streaming.StreamJob: map 100% reduce 33%  
[...] INFO streaming.StreamJob: map 100% reduce 70%  
[...] INFO streaming.StreamJob: map 100% reduce 77%  
[...] INFO streaming.StreamJob: map 100% reduce 100%  
[...] INFO streaming.StreamJob: Job complete: job_200803031615_0021  
[...] INFO streaming.StreamJob: Output: /user/hadoop/texts-output
```