



NEW  
PRO  
LAB

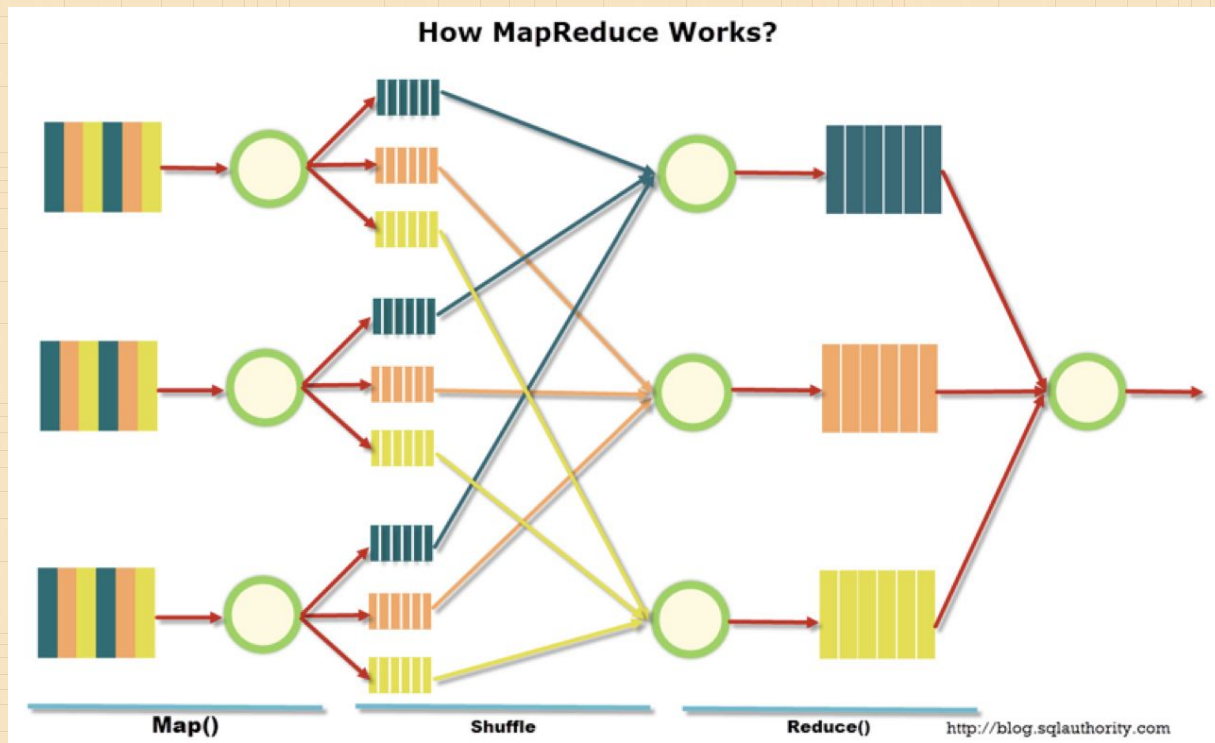
# MapReduce: наше всё

Николай Марков, Aligned Research Group

[NEWPROLAB.COM](http://NEWPROLAB.COM)

# Общая информация

Ссылка на публикацию (2004 год): <https://bit.ly/2hFE1MY>



# Термины

**Split** - Скрытая стадия - разделение входных данных на куски для параллельной обработки

**Map** - Первая ручная стадия обработки - преобразование входных данных в виде пары ключ-значение (например, фильтрация, подмена ключа и т.д.). В терминах ООП, **Mapper** - это класс, реализующий интерфейс этой стадии, например, метод **map()**

**Shuffle** - Скрытая стадия - группировка значений по ключам + сортировка

**Reduce** - Вторая ручная стадия обработки - преобразование входных данных для конкретных ключей. **Reducer** - это класс, реализующий метод **reduce()**

# В Python та же история!

Методы **map()** и **reduce()** есть во многих языках программирования, включая Python, но они по умолчанию обычно не являются параллельными.

Но в Python есть реализация и для параллельной обработки коллекции:

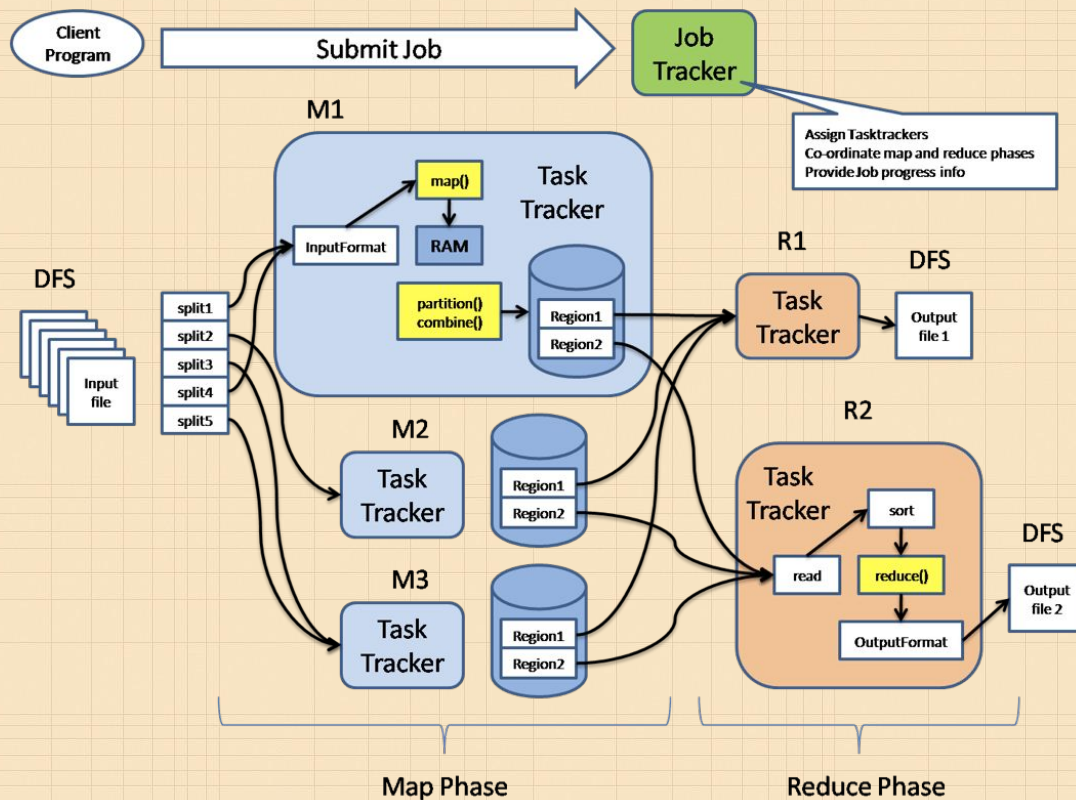
<https://bit.ly/2OhgWDp>

Плюс есть сторонние модули для желающих, но для нашего занятия сегодня мы их использовать не будем:

<https://pythonhosted.org/mrjob/guides/quickstart.html>

Что в Python, что в Hadoop, MapReduce не модифицирует входные данные и не приводит ни к каким побочным эффектам, кроме генерации результата.

# Адская схема всего



# Откуда берем данные - объектные хранилища

HDFS - это Open Source реализация GFS (Google File System). Это далеко не единственная распределенная файловая система, также есть [Ceph](#), [GlusterFS](#) и т.д. Важный термин здесь - Data Locality.



Microsoft Azure  
Blob Storage



**SWIFT**

*an OpenStack Community Project*

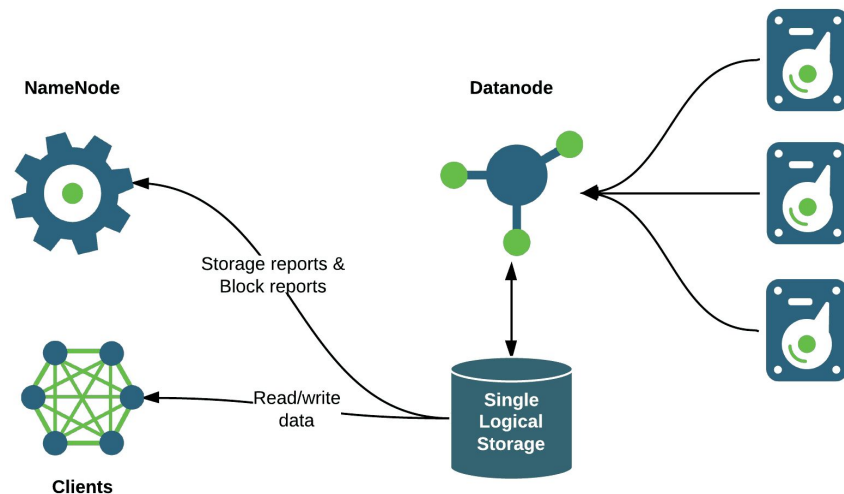


Figure 1: A DataNode presented itself as a single logical storage

# Какие команды нужны

Локально тестируем:

```
~$ cat path/to/data.txt | python3 mapper.py | sort -k1,1 | python3 reducer.py > /path/to/result.txt
```

Копирование папки с файлами на HDFS:

```
~$ hadoop fs -put input_files
```

Смотреть файлы:

```
~$ hadoop fs -ls
```

Скачать файлы:

```
~$ hadoop fs -get output_files
```



# Какие еще команды нужны

Запуск Job'a:

```
~$ yarn jar ./hadoop-streaming.jar -input inp_dir -output out_dir -file mapper.py -file reducer.py  
-mapper "python3 mapper.py" -reducer "python3 reducer.py"
```

Логи:

```
~$ yarn logs -applicationId %id_приложения% | less
```

Либо через прокси: [https://github.com/newprolab/content\\_bigdata9/blob/master/extra/proxy.md](https://github.com/newprolab/content_bigdata9/blob/master/extra/proxy.md)



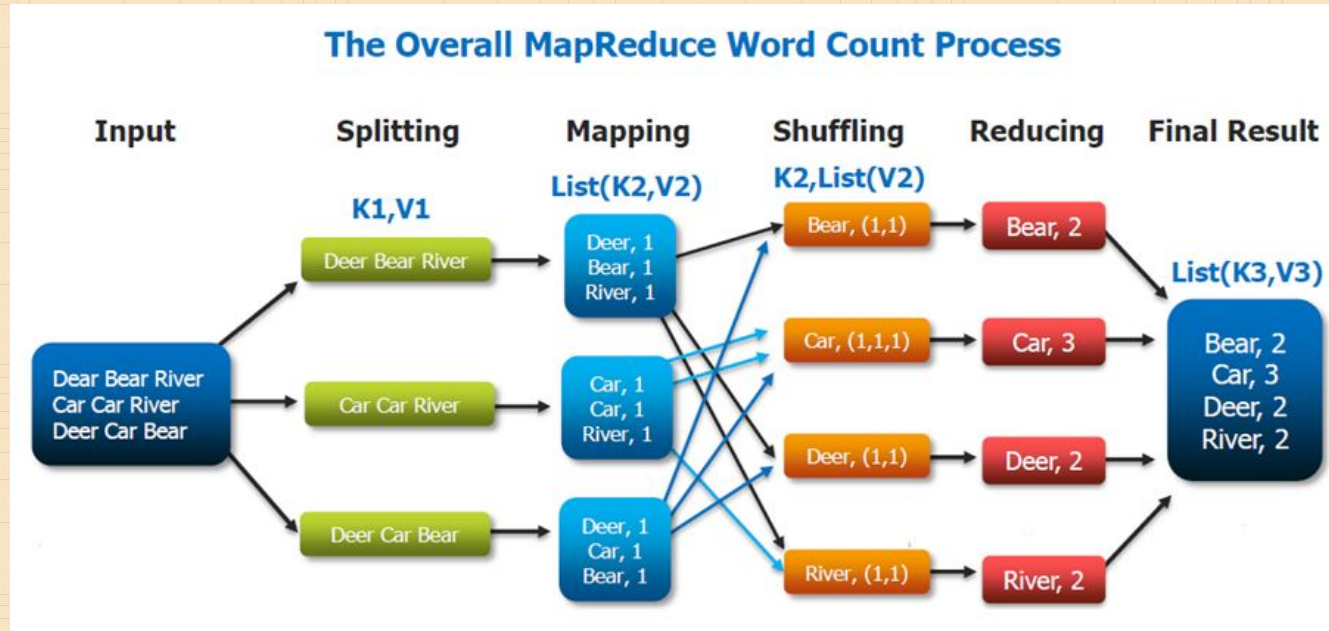
# Как посчитать частоты слов на Википедии?

В оперативку пары “слово -> 1”, скорее всего, не влезут, слов слишком много.

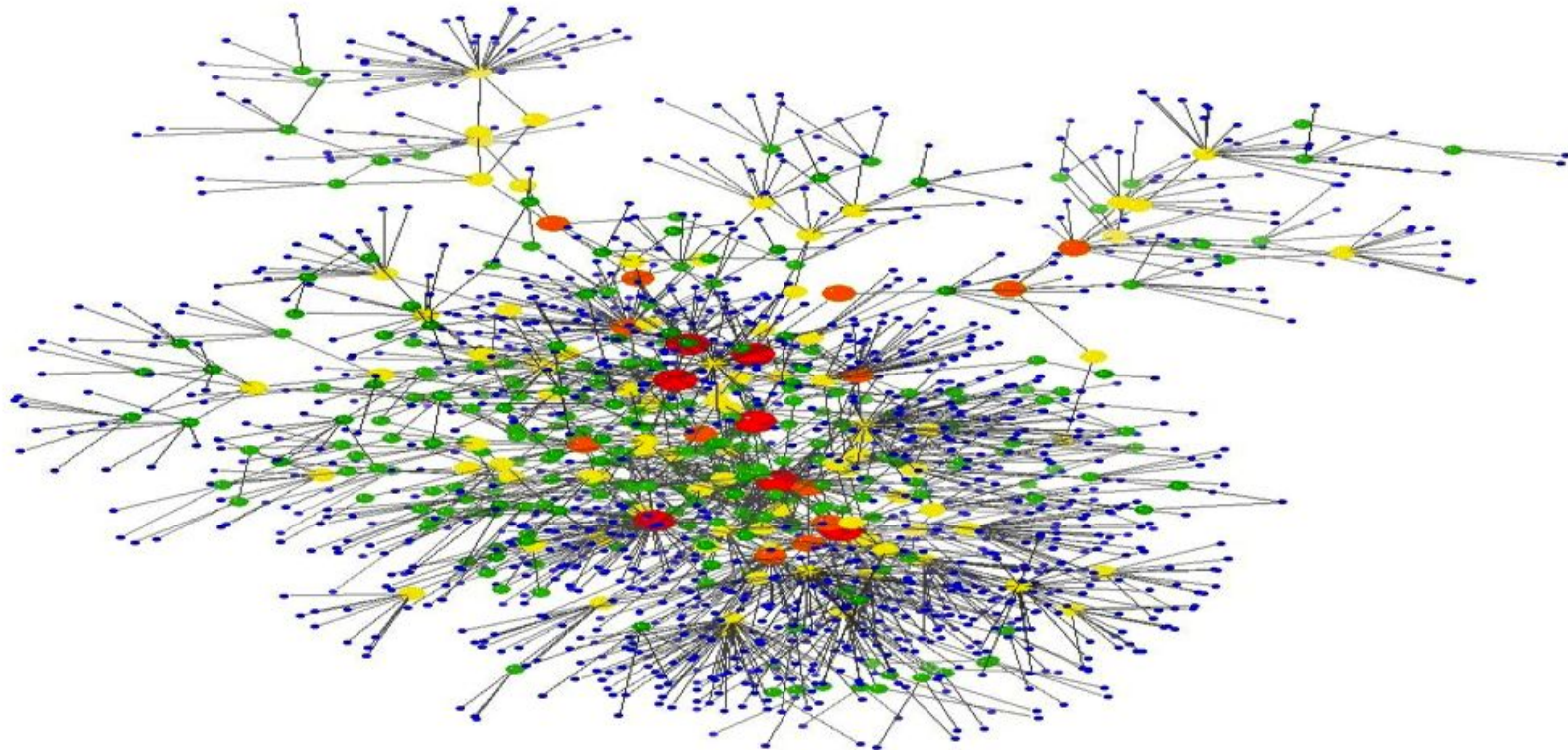
А как вывести самые часто встречающиеся слова?



# Давайте доделаем WordCount



# Как поменять исходящие ссылки на входящие?



# Как посчитать отличников?

Наименование предмета		МЕСЯЦ											
Место	Список обучающихся	ЧИСЛО											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Александров	1	2	3	4	5	6	7	8	9	10	11	12
2	Борисов	1	2	3	4	5	6	7	8	9	10	11	12
3	Васильев	1	2	3	4	5	6	7	8	9	10	11	12
4	Григорьев	1	2	3	4	5	6	7	8	9	10	11	12
5	Давыдов	1	2	3	4	5	6	7	8	9	10	11	12
6	Зинченко	1	2	3	4	5	6	7	8	9	10	11	12
7	Иванов	1	2	3	4	5	6	7	8	9	10	11	12
8	Климов	1	2	3	4	5	6	7	8	9	10	11	12
9	Колесников	1	2	3	4	5	6	7	8	9	10	11	12
10	Королев	1	2	3	4	5	6	7	8	9	10	11	12
11	Кузнецов	1	2	3	4	5	6	7	8	9	10	11	12
12	Лавров	1	2	3	4	5	6	7	8	9	10	11	12
13	Левченко	1	2	3	4	5	6	7	8	9	10	11	12
14	Михайлов	1	2	3	4	5	6	7	8	9	10	11	12
15	Новиков	1	2	3	4	5	6	7	8	9	10	11	12
16	Осипов	1	2	3	4	5	6	7	8	9	10	11	12
17	Петров	1	2	3	4	5	6	7	8	9	10	11	12
18	Романов	1	2	3	4	5	6	7	8	9	10	11	12
19	Сидоров	1	2	3	4	5	6	7	8	9	10	11	12
20	Смирнов	1	2	3	4	5	6	7	8	9	10	11	12
21	Тихонов	1	2	3	4	5	6	7	8	9	10	11	12
22	Толкачев	1	2	3	4	5	6	7	8	9	10	11	12
23	Трофимов	1	2	3	4	5	6	7	8	9	10	11	12
24	Федотов	1	2	3	4	5	6	7	8	9	10	11	12
25	Филиппов	1	2	3	4	5	6	7	8	9	10	11	12
26	Харьков	1	2	3	4	5	6	7	8	9	10	11	12
27	Хохлов	1	2	3	4	5	6	7	8	9	10	11	12
28	Цыганов	1	2	3	4	5	6	7	8	9	10	11	12
29	Чайков	1	2	3	4	5	6	7	8	9	10	11	12
30	Чирков	1	2	3	4	5	6	7	8	9	10	11	12
31	Шаров	1	2	3	4	5	6	7	8	9	10	11	12
32	Шевченко	1	2	3	4	5	6	7	8	9	10	11	12
33	Шестаков	1	2	3	4	5	6	7	8	9	10	11	12
34	Щеглов	1	2	3	4	5	6	7	8	9	10	11	12
35	Юсупов	1	2	3	4	5	6	7	8	9	10	11	12



# Как построить гистограмму оценок?

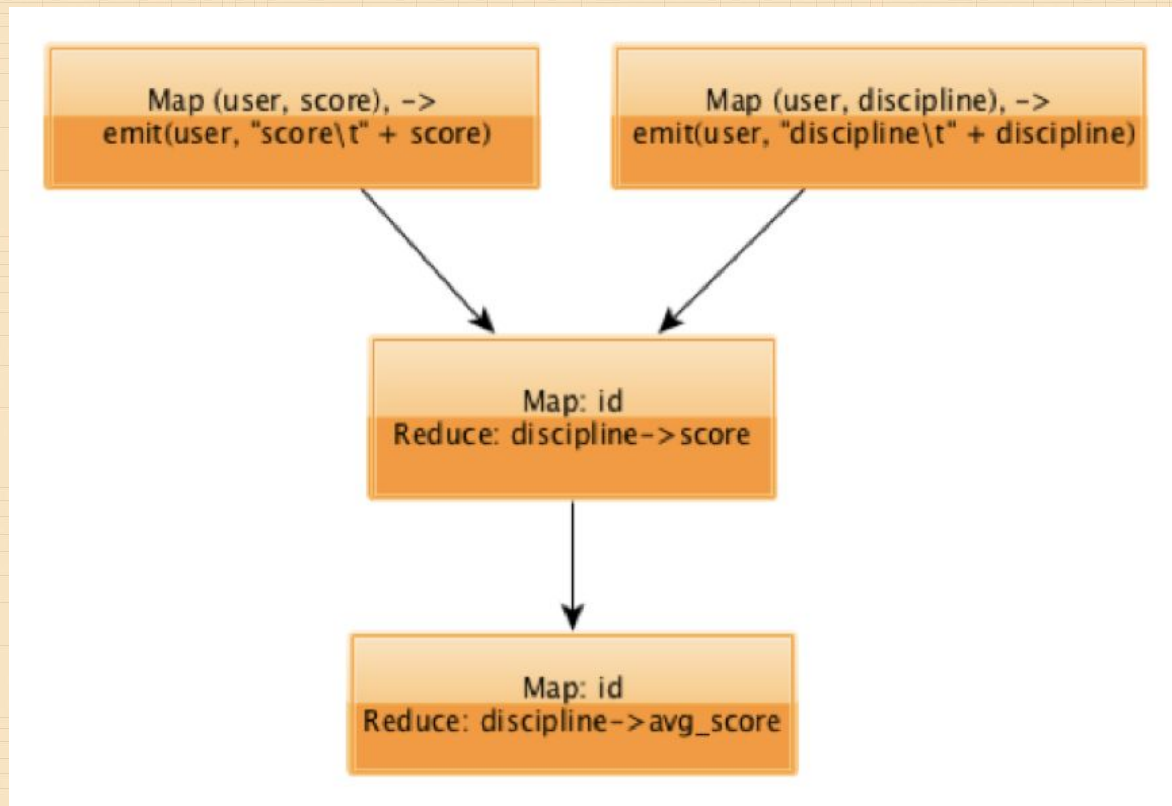
```
3.1      *
3.2      ****
3.3      *****
3.4      *****
3.5      *****
3.6      *****
3.7      *****
3.8      *****
3.9      *****
4.0      *****
4.1      *****
4.2      *****
4.3      *****
4.4      *****
4.5      *****
4.6      *****
4.7      *****
4.8      *****
```

# Средняя оценка и любимый предмет

- **Файл 1** — как в предыдущем задании
- **Файл 2** — `<user>\t<любимый предмет>`

Посчитать среднюю оценку среди любителей данного предмета

# Reduce Join - среднее по средним





# Map Join

Подсчет средней оценки по предмету без усреднения по пользователям.

(Сработает, если размер одного из файлов относительно небольшой)