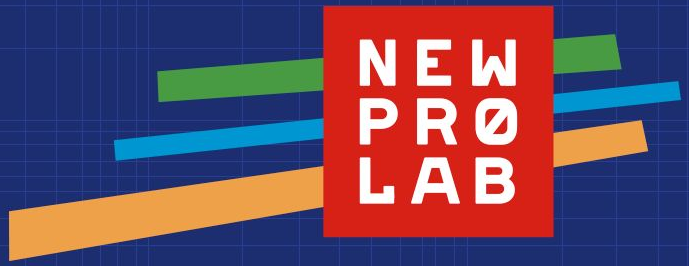


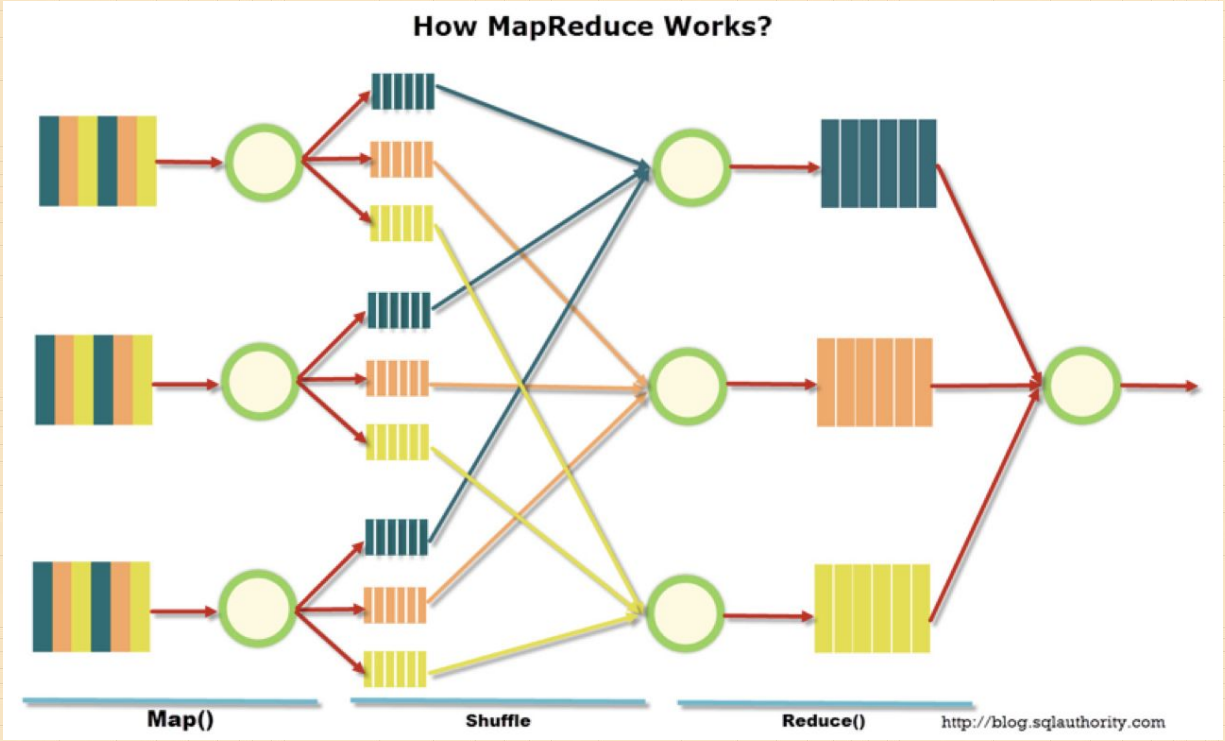
MapReduce: наше всё

Николай Марков, Aligned Research Group

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!

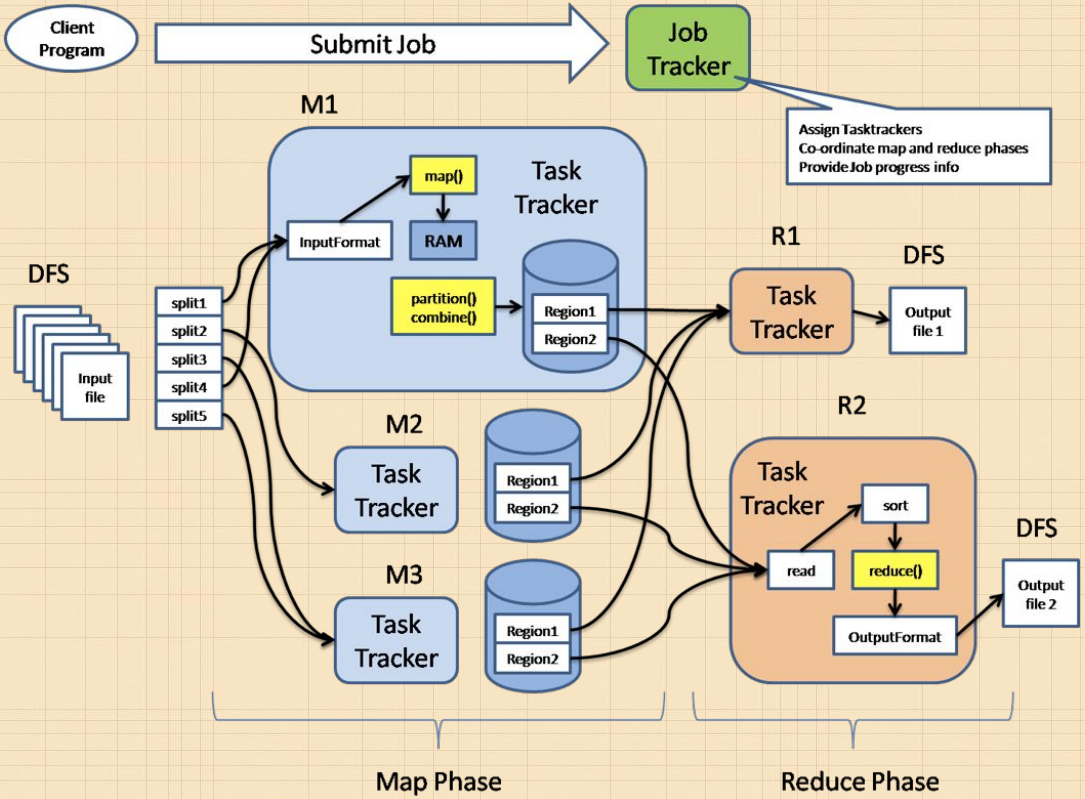
Методы **map()** и **reduce()** есть во многих языках программирования, включая Python, но они по умолчанию обычно не являются параллельными.

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

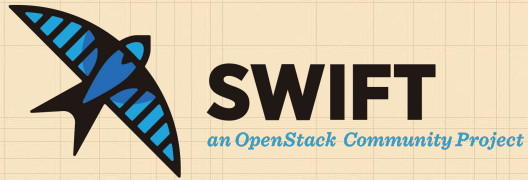
Что в Python, что в Hadoop, MapReduce не модифицирует входные данные и не приводит ни к каким побочным эффектам, кроме генерации результата.

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.



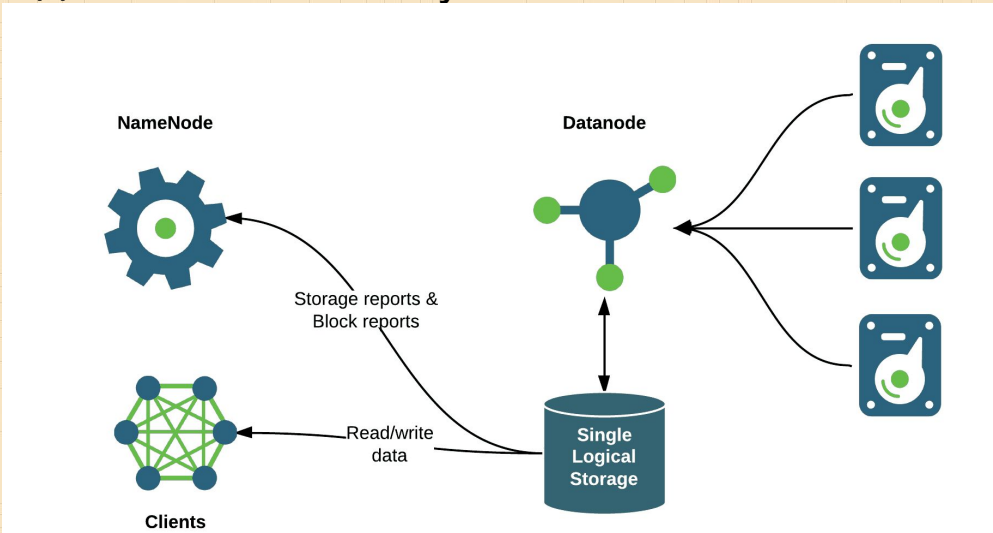


Figure 1: A DataNode presented itself as a single logical storage

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’а:

~$ 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:

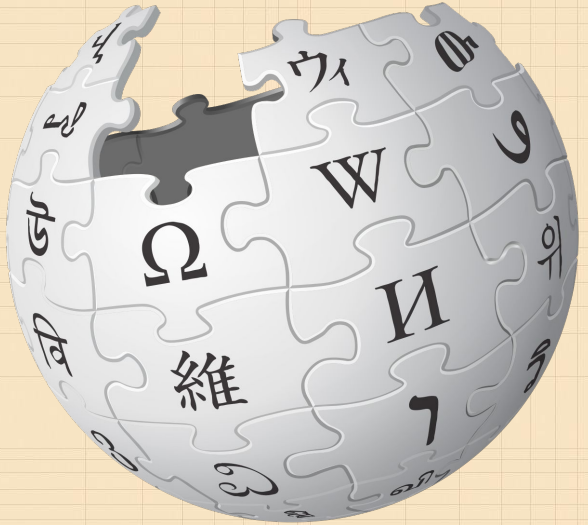
~$ yarn logs -applicationId %id_приложения% | less

Либо через прокси: https://github.com/newprolab/content_bigdata9/blob/master/extra/proxy.md

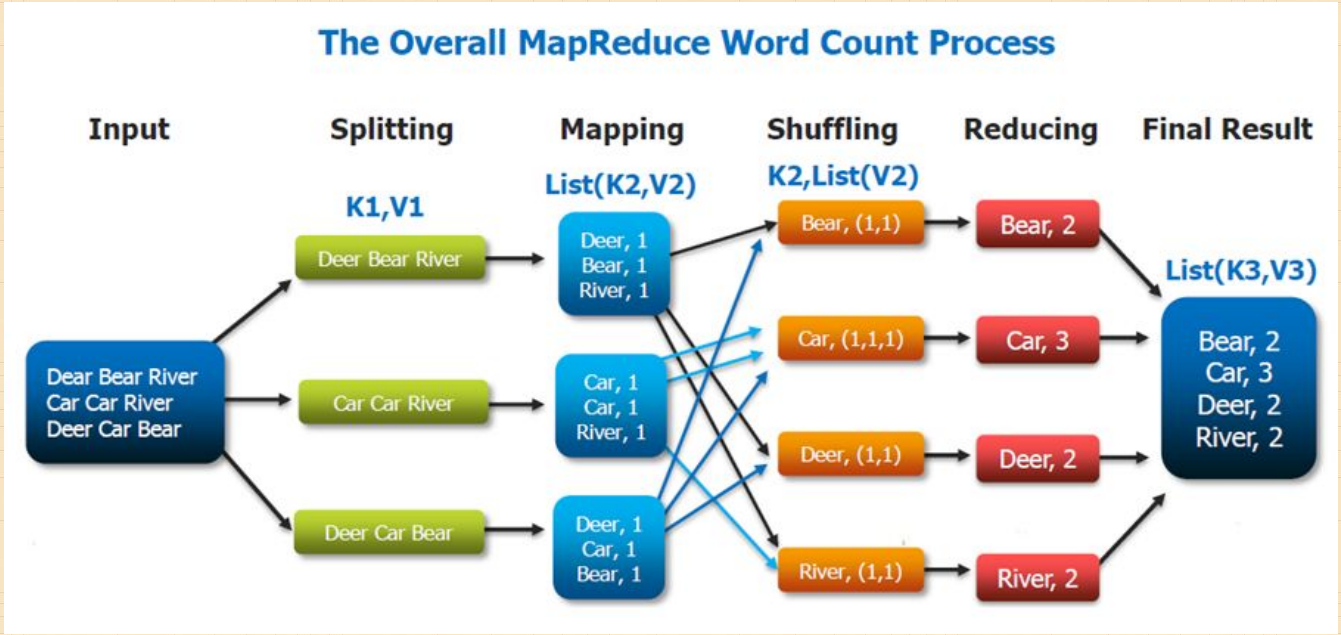
Как посчитать частоты слов на Википедии?

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

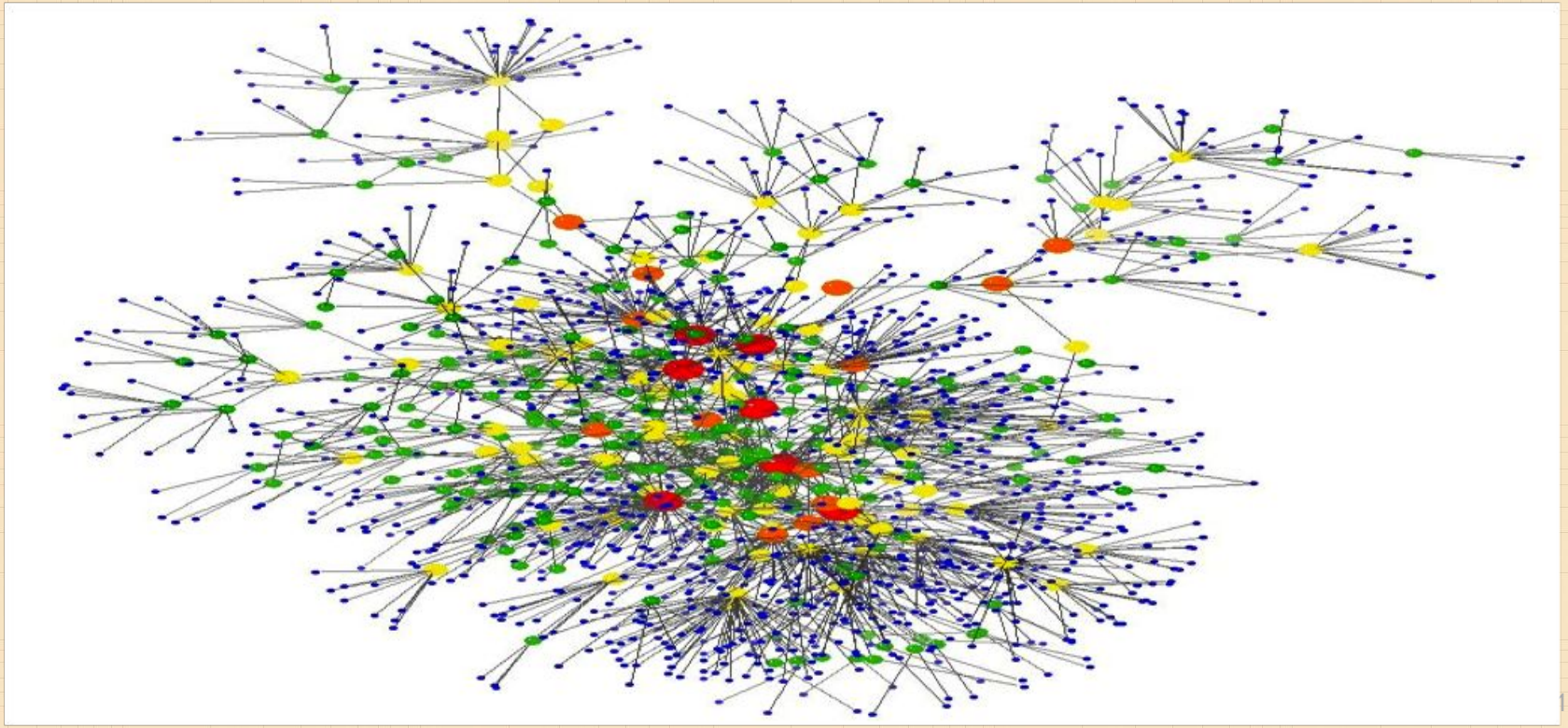
А как вывести самые часто встречающиеся слова?



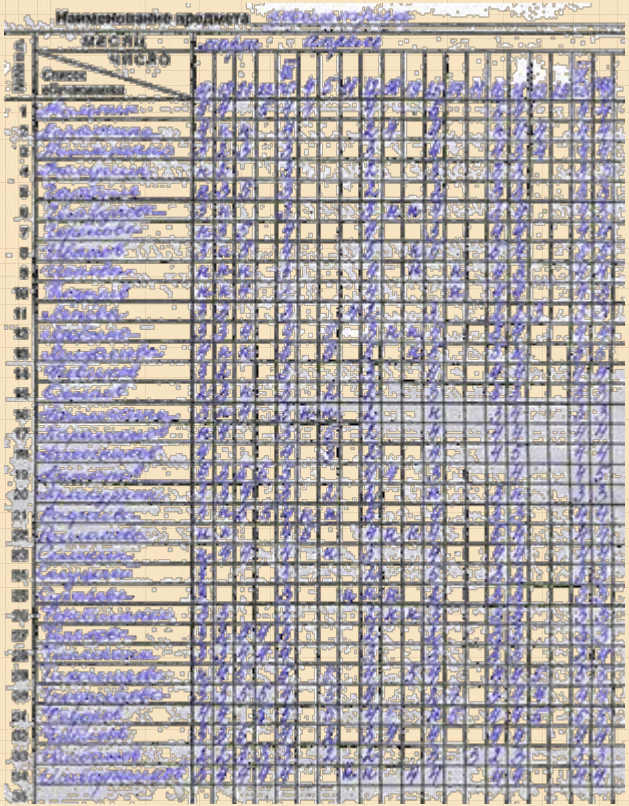
Давайте доделаем WordCount



Как поменять исходящие ссылки на входящие?



Как посчитать отличников?



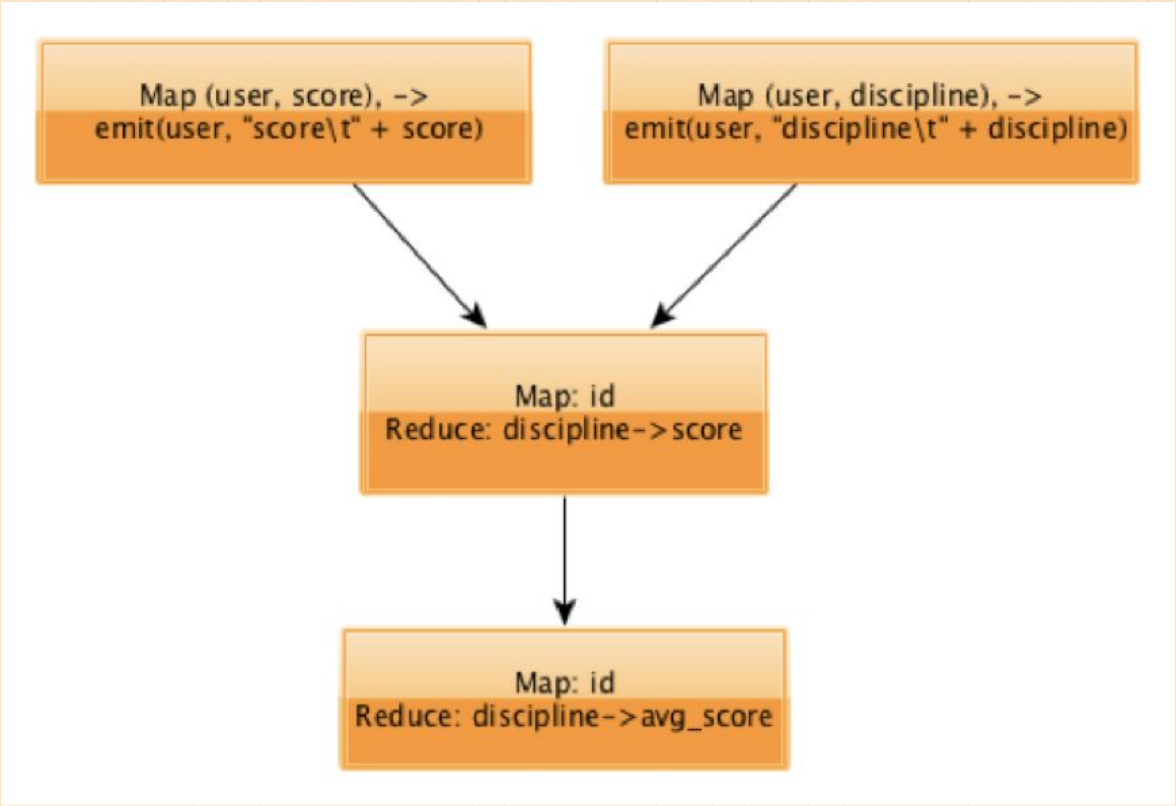
Как построить гистограмму оценок?

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)