

# Введение в Data Science

## Занятие 12. MapReduce

Николай Анохин    Михаил Фирулик

25 мая 2014 г.

ТЕХНОСФЕРА @mail.ru

## Вопрос 1 (1)

Кого на Боярском языке обозначают следующими понятиями?

*Всеведъ-Воевода, Князь Явственность, Сотникъ Вестимо,  
Догада-Богатырь*

Подсказка: lurkmore

Примечание

*один человек не может победить больше 2 раз подряд и 4 раз в общей сложности*

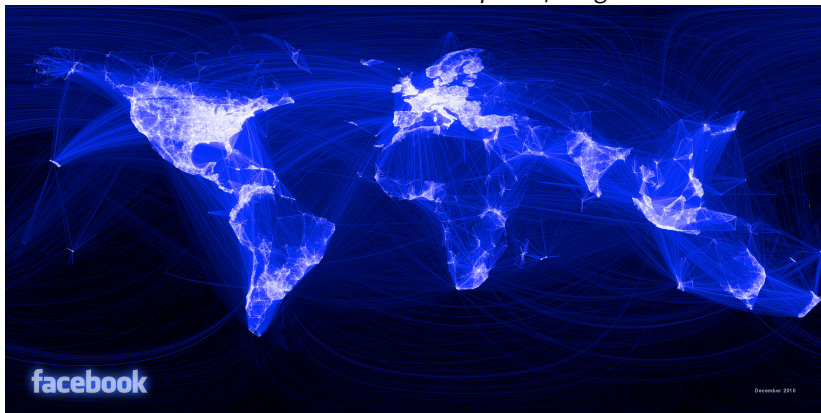
Стек технологий Big Data

(H)DFS

MapReduce

# Стек технологий Big Data

*Сотни Гигабайт – нижняя граница Big Data*



- ▶ Расчет PageRank всех страниц в интернете
- ▶ Поиск по друзьям в социальной сети

# Суперкомпьютер VS кластер

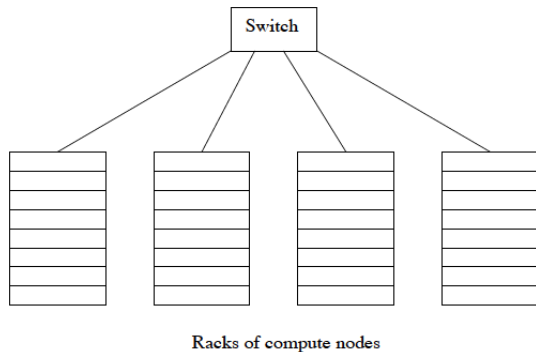
## Проблемы

- ▶ данные не помещаются на HDD одной машины
- ▶ чтение со скоростью порядка сотен MB/s

## Решения

- ▶ Суперкомпьютер  
*много процессоров, специальное железо*
- ▶ Кластер  
*много “обычных” машин, соединенных сетью*

# Архитектура кластера



- ▶ Dual-processor x86, 2-4 GB, Linux machines
- ▶ 1Gb/s Network switches
- ▶ Inexpensive IDE disks

# Многое может пойти не так...

**DFS** Хранить несколько реплик данных

**MR** Вычисления нужно разбивать на части



# Реализации DFS

## Примеры

- ▶ The Google File System (GFS)
- ▶ Hadoop Distributed File System (HDFS)
- ▶ CloudStore DFS by Kosmix

## Свойства

- ▶ Файлы могут быть огромного размера
- ▶ Данные не меняются, только добавляются
- ▶ Файлы хранятся кусками (chunks)

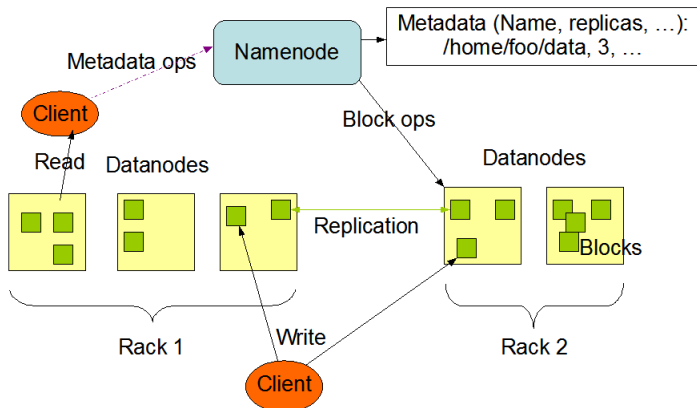


# HDFS

HDFS – не файловая система общего назначения

- ▶ Создана для хранения огромных массивов данных (Петабайты)
- ▶ Предоставляет *надежный* доступ к данным
- ▶ Поддерживает горизонтальное масштабирование
- ▶ Хорошо интегрирована с Hadoop MapReduce

# Архитектура HDFS



- ▶ файлы хранятся блоками на Data Node по умолчанию 64M
- ▶ метаданные хранятся в RAM на Name Node  
имя, права, расположение блоков на Data Node

# Доступ к файлам в HDFS

Команды HDFS

[http://hadoop.apache.org/docs/r0.18.3/hdfs\\_shell.html](http://hadoop.apache.org/docs/r0.18.3/hdfs_shell.html)

```
$ cat sample.txt
A 12
B 12
A 14
A 22
C 12
$ hadoop fs -put sample.txt /user/anokhin
$ hadoop fs -ls /user/anokhin
```

Команды: cat, cp, mv, rm, ls, put, get, ... (см документацию)

# Подключение к кластеру Hadoop

Для Windows

*скачиваем putty, подключаемся к `sfera-ds.openrise.org`*

Для других ОС

*`ssh username@sfera-ds.openrise.org`*

Пользователи

- |                    |                |            |
|--------------------|----------------|------------|
| ▶ alibekov         | ▶ kulikov      | ▶ nikolaev |
| ▶ blagoveschenskiy | ▶ kulpinov     | ▶ novikov  |
| ▶ filipenko        | ▶ ludovichenko | ▶ ovlasuk  |
| ▶ kemaev           | ▶ medvedev     | ▶ shvets   |
| ▶ koltsov          | ▶ melnikov     | ▶ sovetov  |
| ▶ kondratiev       | ▶ mozharova    | ▶ taraban  |

## Данные

Данные об активности пользователей в интернете за апрель 2014  
находятся в директории HDFS /data/logs

```
$ hadoop fs -cat /data/logs/20140425/part-00008 | head -5
```

N	Название	Описание	Пример
1	<i>user_id</i>	ID пользователя	100034b5
2	<i>timestamp</i>	Unix time (сек)	1398409877
3	<i>gender</i>	0 – муж. 1 – жен.	1
4	<i>age</i>	кол-во полных лет	26
5	<i>os</i>	операционная система	win/win-xp
6	<i>browser</i>	браузер и версия	chrome/chrome-34
7	<i>resolution</i>	разрешение экрана	4
8	<i>touch</i>	наличие тачскрин	1
9	<i>hit_url</i>	URL посещенной страницы	https://e.mail.ru/
10	<i>referrer_url</i>	URL-referrer	http://mail.ru
11	<i>load_start</i>	время начала загрузки	1398065613566
12	<i>load_end</i>	время окончания загрузки	1398065613590

## Вопрос 2 (1)

Найти *user\_id* последней записи в файле

```
/data/logs/20140421/part-00008
```

### Вопрос 3 (1)

Найти *referrer\_url* четвертой записи в файле

```
/data/logs/20140421/part-00000
```

Подсказка: head

## Вопрос 4 (1)

Посчитать количество записей в файле

```
/data/logs/20140421/part-00000
```

Подсказка: `wc -l`



## Вопрос 5 (2)

Посчитать количество записей, сделанных мужчинами и женщинами в файле

```
/data/logs/20140421/part-00000
```

Подсказка: sort, uniq, cut

## Вопрос 6 (2)

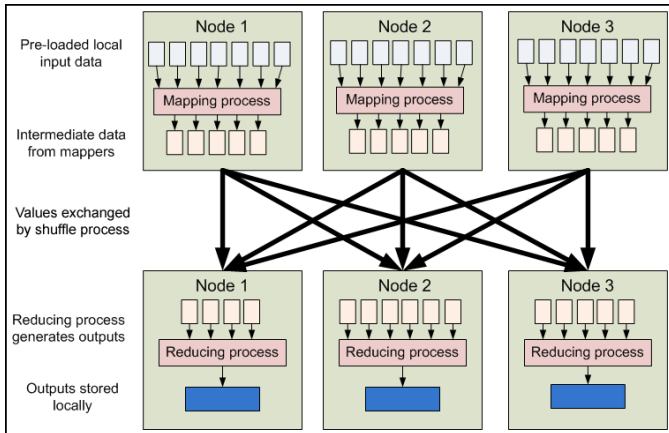
Вывести список `user; d[tab]age` из 10 самых старых пользователей в файле

```
/data/logs/20140421/part-00000
```

Подсказка: `sort`, `cut`, `head`

# Идея MapReduce

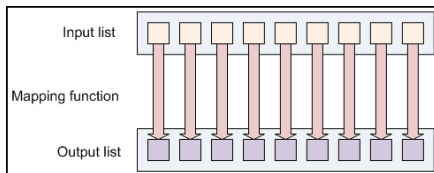
Цель – обработка больших объемов данных параллельно на нескольких машинах



# Map

Из исходного файла последовательно считываются пары ключ-значение и подаются в функцию **map**

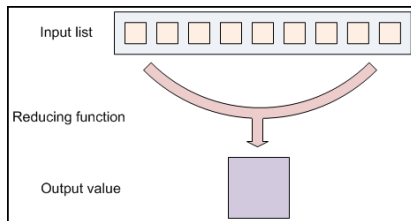
Сигнатура:  $(k_1, v_1) \rightarrow \text{list}(k_2, v_2)$



# Reduce

Все значения, принадлежащие одному ключу, обрабатываются функцией **reduce**

Сигнатура:  $(k_2, \text{list}(v_2)) \rightarrow \text{list}(k_2, v_2)$



# Умножение матрицы на вектор

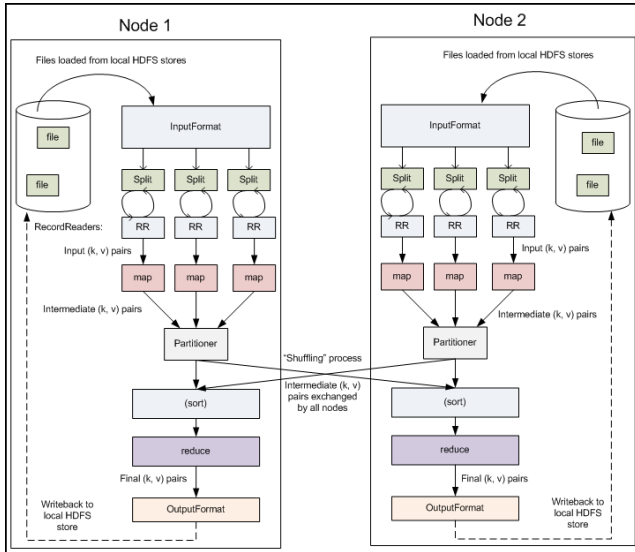
## Пример

Дана матрица  $M$  размера  $n \times n$  с элементами  $m_{ij}$  и вектор  $v$  с элементами  $v_j$ .

Рассмотрим случаи

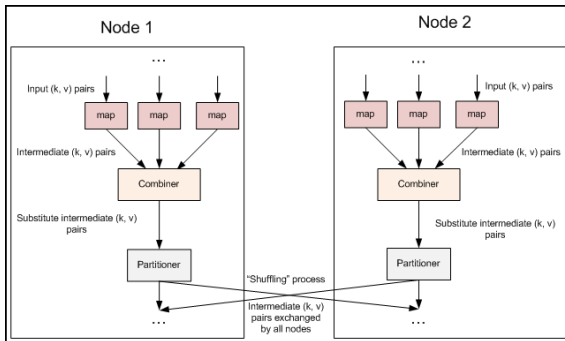
1.  $v$  помещается в память одной машины
2.  $v$  не помещается в память одной машины

# MapReduce на Hadoop



# Combine

Пусть **reduce** – коммутативен и ассоциативен





Когда что-то все-же пошло те так

- П1 Недоступна машина, выполняющая Reduce  
*другие машины выполняют заново **не законченные** ей задачи*
- П2 Недоступна машина, выполняющая Map  
*другие машины выполняют заново **все** ее задачи*

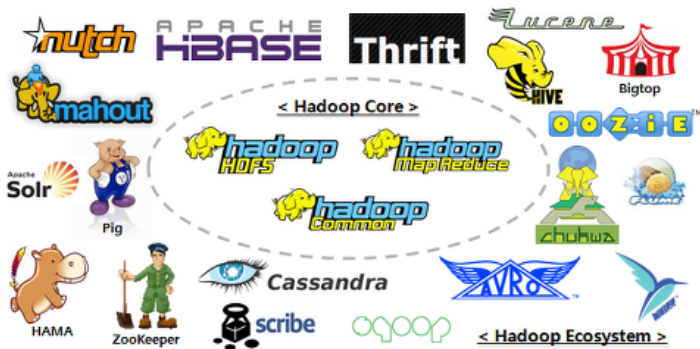
Вывод: минимизировать коммуникацию между машинами

# Операторы реляционной алгебры

- S Selection с условием  $C$ :  $\sigma_C(R)$
- P Projection на подмножество  $S$ :  $\pi_S(R)$
- J Natural Join:  $R \bowtie S$
- U Union (intersection, difference):  $R \cup S$
- G Grouping атрибутами  $X$ :  $\gamma_X(R)$

# Hadoop экосистема

- ▶ Hadoop (MapReduce)
- ▶ Hive (SQL-like via MR)
- ▶ Pig (язык запросов для Hadoop)
- ▶ Mahout (ML)
- ▶ Spark (SQL + ML + Graphs + Streaming)
- ▶ Tez (ациклические workflow)



### Вопрос 8 (3)

Реализовать Hadoop job для вычисления количества посещений на каждом из доменов. Распечатать список из 20 самых посещаемых доменов 15 апреля (вместе с количеством посещений).

### Вопрос 9 (3)

Построить график распределения количества посещений доменов в натуральной шкале и в логарифмической по обоим осям.