

Хранение и Обработка Больших Объёмов Данных

Антон Горохов

старший разработчик, Яндекс

anton.gorokhov@gmail.com

Big Data

Антон Горохов

старший разработчик, Яндекс

anton.gorokhov@gmail.com

План лекции

- **I. Введение**
- II. Пример: статистика сайта
- III. MapReduce:
 - 1) идеи
 - 2) история и выводы
- IV. Hadoop
- V. Заключение. Как будет устроен курс

Цели курса

- Общая культура в области Больших Данных
- Taste of Big Data – практика
- Инструмент для ~~решения различных задач~~ творчества с Данными

“Organizations will become more like jazz bands and less like orchestras”

Luke Loneragan, "5 Big Questions about Big Data"

Big Data: 3V

- **V**olume – объем
 - >> объема 1 HDD
- **V**elocity – скорость
 - > 10000 событий в секунду
- **V**ariety – разнообразие
 - текст, разные форматы
 - изображения (графика + EXIF)
 - видео, звук

Области применения

- Научные расчеты (ядерная физика, биоинформатика, ...)
- Финансы (оценка кредитоспособности, скоринг)
- Интернет (поиск, реклама, статистика,...)
- ...

Конкретнее: работа

IPONWEB

WHO WE ARE

U-PLATFORM

BIDSWITCH

NEWS & INS

Jobs

Analyst (Big Data)

Engineering | Moscow, Russia

Tasks:

- The main tasks are the development / adaptation algorithms of mathematical modeling of various dependencies, available in large s experimental verification. We take Methods from the scientific literature, developing or finalizing them. In addition, there are a nur development for analyzing the result of our methods.

Why our tasks are especially interesting:

- You will learn how to practically use innovative ideas in such important areas as:
 1. Big Data;
 2. Programmatic marketing;
- You will get an opportunity to understand the details of the studied approaches rather than 'stupid' ready to use implementation;
- Our data volumes are measured in billions of events per day, it is one of the highest number among Russian companies of the similar

Конкретнее: работа



“... interview Software Development Engineers and Software Development Managers who would be interested in relocating to Seattle or London to work for Amazon Instant Video.

We are looking for experienced candidates such as yourself, but are open to a variety of work backgrounds/experiences. Video domain experience is not required for these positions, as we have opportunities for you to **work in areas such as machine learning, big data, distributed/scalable systems**, server side applications, mobile, customer intelligence and many more. “

Пример: трафик в городе

- Данные о загруженности магистралей, прогноз погоды - **V**olume
- ... датчики и камеры на дорогах – **V**ariety
- ... в реальном времени - **V**elocity

Пример: статистика сайта

- Сколько посетителей, их характеристики
 - география
 - источники посещений (закладки, поиск, реклама, соц.сети, ...)
 - сколько страниц посмотрели
- Новые / постоянные посетители
 - как часто возвращаются
- Достижение целей
 - покупки
 - просмотр > N страниц
- Технические характеристики (для дизайна и юзабилити)
- Интересы посетителей
 - общая аудитория с другими сайтами
- Мониторинг сайта

План лекции

- I. Введение
- **II. Пример: статистика сайта**
- III. MapReduce:
 - 1) идеи
 - 2) история и выводы
- IV. Hadoop
- V. Заключение. Как будет устроен курс

Пример: статистика сайта

- Сколько посетителей, их характеристики
 - география
 - источники посещений (закладки, поиск, реклама, соц.сети, ...)
 - сколько страниц посмотрели
- Новые / постоянные посетители
 - как часто возвращаются
- Достижение целей
 - покупки
 - просмотр > N страниц
- Технические характеристики (для дизайна и юзабилити)
- Интересы посетителей
 - общая аудитория с другими сайтами
- Мониторинг сайта

Исходные данные – логи

```
89.169.243.120 - - [01/Apr/2012:00:00:02 +0400] "GET /13385393/
HTTP/1.1" 200 26404 "http://www.ya.ru/" "Mozilla/5.0 (iPad; U;
CPU OS 4_3_3 like Mac OS X; ru-ru) AppleWebKit/533.17.9 (KHTML,
like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5"
"uid=000000014ED4E0AD34C5064F00E74901" "-" 1333224002.813
92.194.73.237 - - [01/Apr/2012:00:00:02 +0400] "GET /13389254/
HTTP/1.1" 200 25610 "http://www.lenta.ru/" "Mozilla/5.0
(Windows NT 5.1; rv:11.0) Gecko/20100101 Firefox/11.0"
"uid=00000001D4E779AAA4CCC66FC01D27601" "-" 1333224002.827
46.221.141.0 - - [01/Apr/2012:00:00:02 +0400] "GET /13389756/
HTTP/1.1" 200 26394 "http://www.rambler.ru/" "Opera/9.80
(Windows NT 5.1; U; Edition Yx; ru) Presto/2.10.229
Version/11.61" "uid=000000014EFA41333071114204017201" "-"
1333224002.895
```

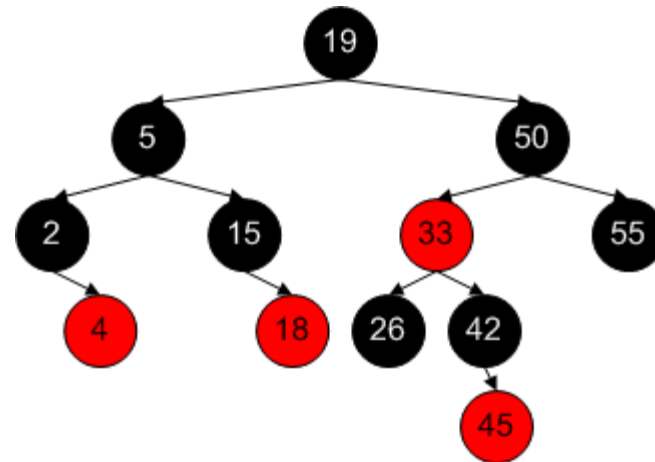
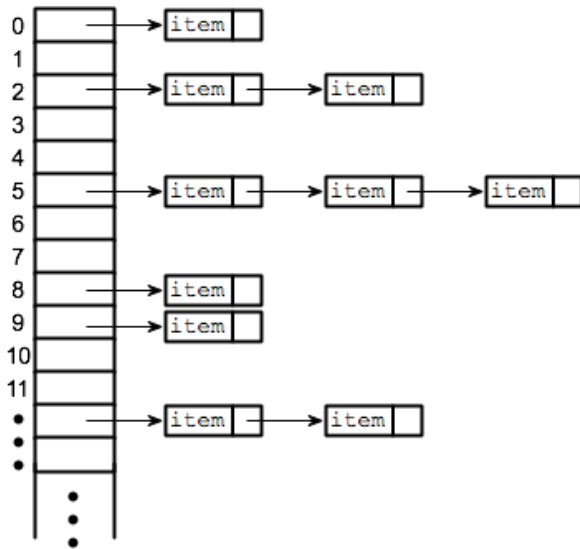
IP **время** **URL**
referer **браузер (User-Agent)**
cookie – идентификатор пользователя

Задача: сколько уникальных?

```
0000002A4F776242285A553B01E8FC01
000000014ED4E0AD34C5064F00E74901
0000001D4E779AAA4CCC66FC01D27601
0000002A4F776242285A553B01E8FF01
0000002A4F776242285A553B01E8FD01
000000014F5A413832E111430401FA01
0000002A4F70C38B0B8F3B2300842701
05FE817949848B9300011D02F8CDF001
0000002A4F44135A990142CF007FE701
06199C9C4F043E2D000032F181E57A01
000000014D605672AED8116B03857501
00000BB94EDBCD4241CD2D1A0889D801
0000002A4F7762432A2555410201D301
0000002A4F7762432AB6553E01F6A101
0000002A4F77624310AD553802030601
000000014F75575E27308DC005855E01
05F656904F0578B60000080AD99ADC01
0000002A4F7762431890553701F95601
0000002A4F776243285A553B01E90101
05F9DAFE4ED2238B000179629AA20C01
060C116D4F6F995700002468D58B9001
0000002A4F77624316AB553F01F32501
000000014F0165D51EFB23200F37CA01
000000014F7761DABC730A5508D2A901
000000014F4E1AF776B57E5901B3E101
0000002A4F7762432C85554301FC8D01
```

Решение 1: память

- C++ (`std::map`, `std::unordered_map`)
- python (`dict`)
- Java (`Map`)



–Минус: память может закончиться

Решение 2: диск

- Активность посетителей сайта:

```
$ cat access.log | get_uid.sh | sort | uniq -c
34  0000002A4F776242285A553B01E8FC01
12  000000014ED4E0AD34C5064F00E74901
3   0000001D4E779AAA4CCC66FC01D27601
...
```

- Уникальные посетители:

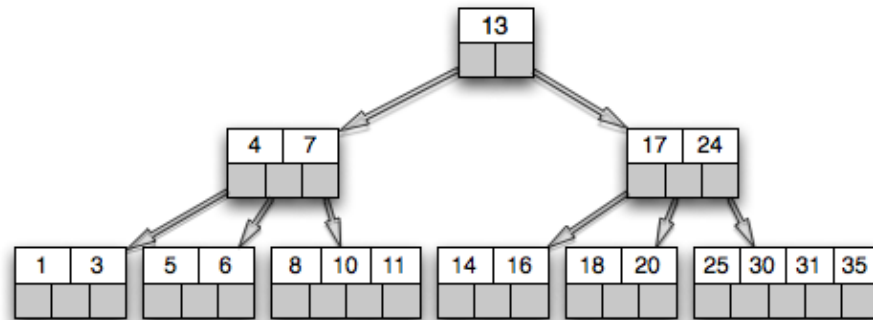
```
$ cat access.log | get_uid.sh | sort | uniq | wc -l
```

Минусы:

- в один поток, т.е. медленно
- диски тоже ограничены

Решение 3: DB на диске

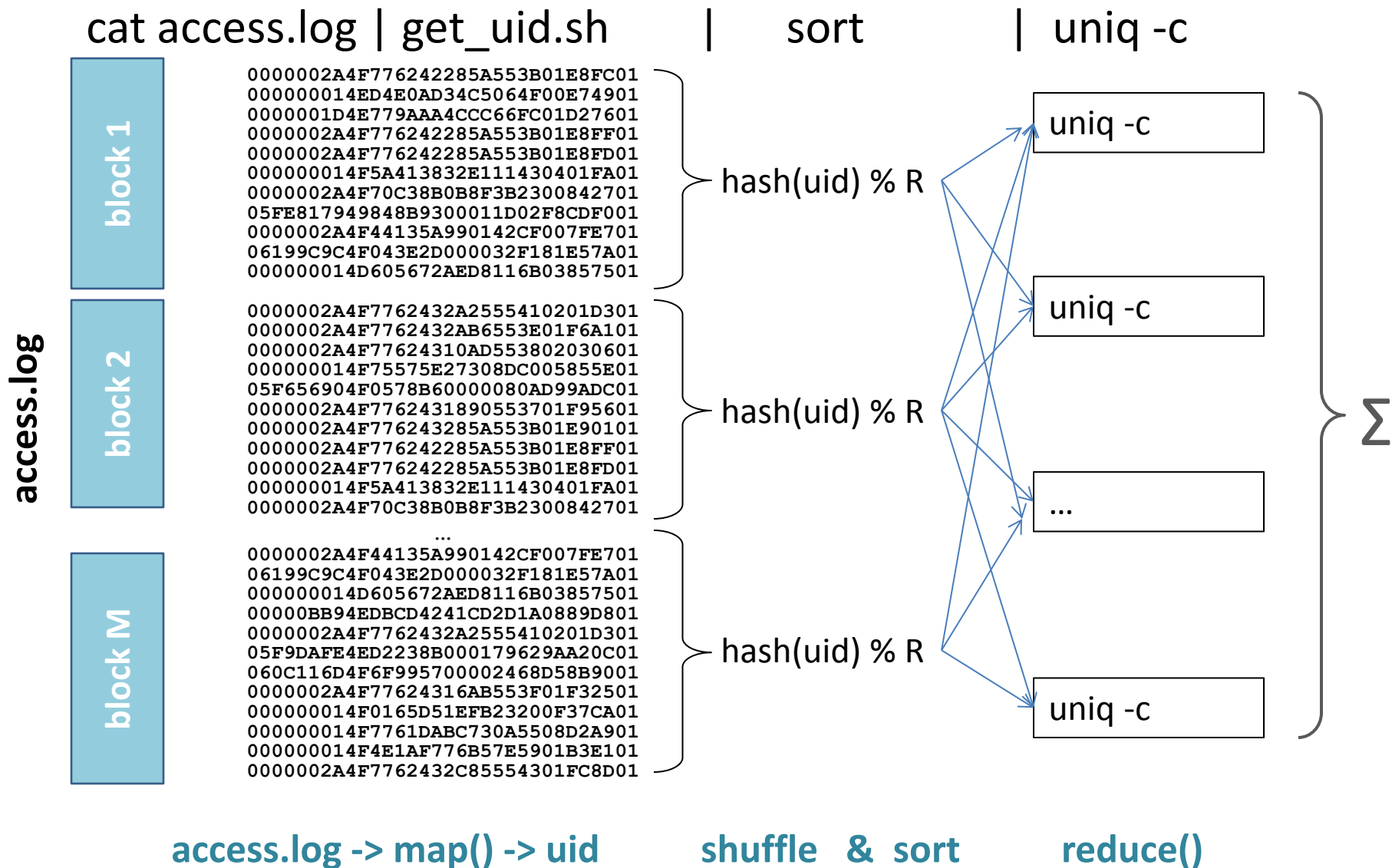
- PostgreSQL
- MySQL
- Oracle



Минусы:

- у B-Tree overhead для произвольного доступа (он нам не нужен)
- множественные вставки – медленны

Решение 4: кластер



```
$ cat access.log | get_uid | sort | uniq -c
```

- `cat access.log: (key, value), value=line`
- `get_uid: (-, line) → (uid, 1)`
- `sort`
- `uniq -c: (uid, [1]) → (uid, count)`

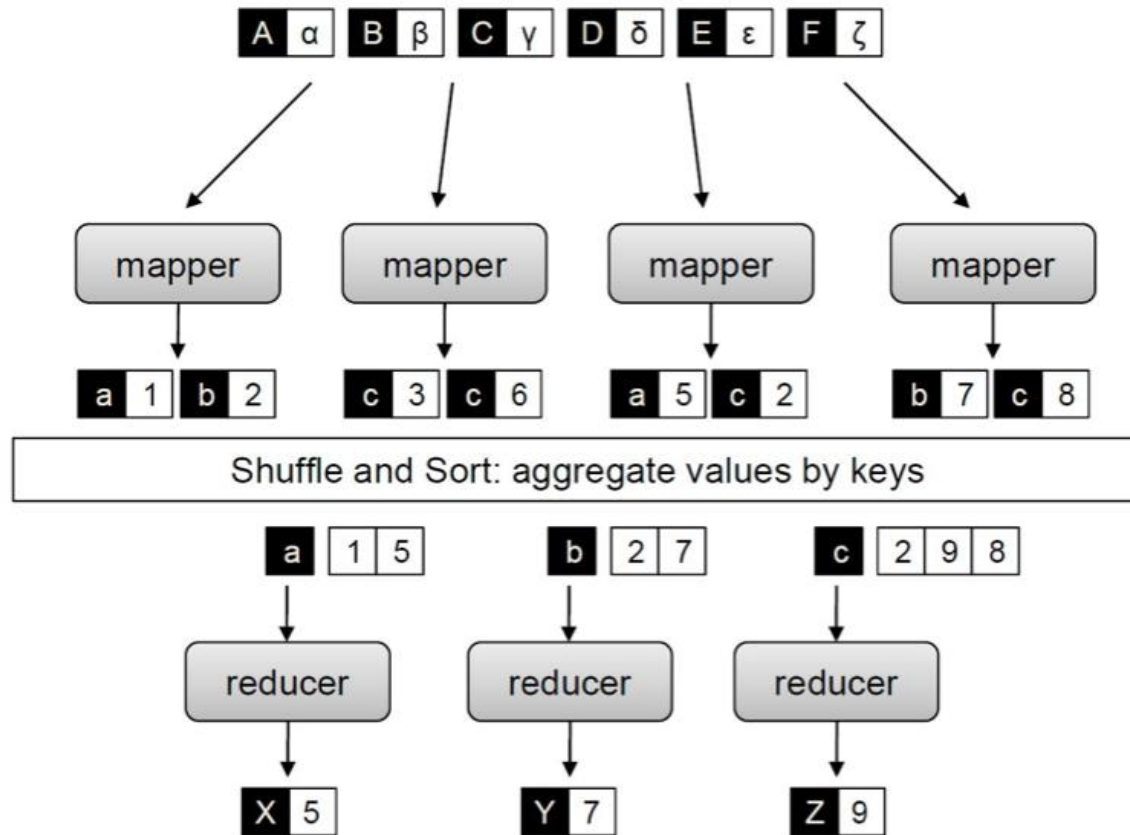
Этапы решения

- `cat access.log: (key, value), value=line`
- *чтение $(k1, v1)$*
- `get_uid: (-, line) → (uid, 1)`
- *map: $(k1, v1) → [(k2, v2)]$*
- `sort`
- *сортировка и группировка по ключу $k2$*
- `uniq -c: (uid, [1]) → (uid, count)`
- *reduce: $(k2, [v2]) → (k3, v3)$*

Идея MapReduce

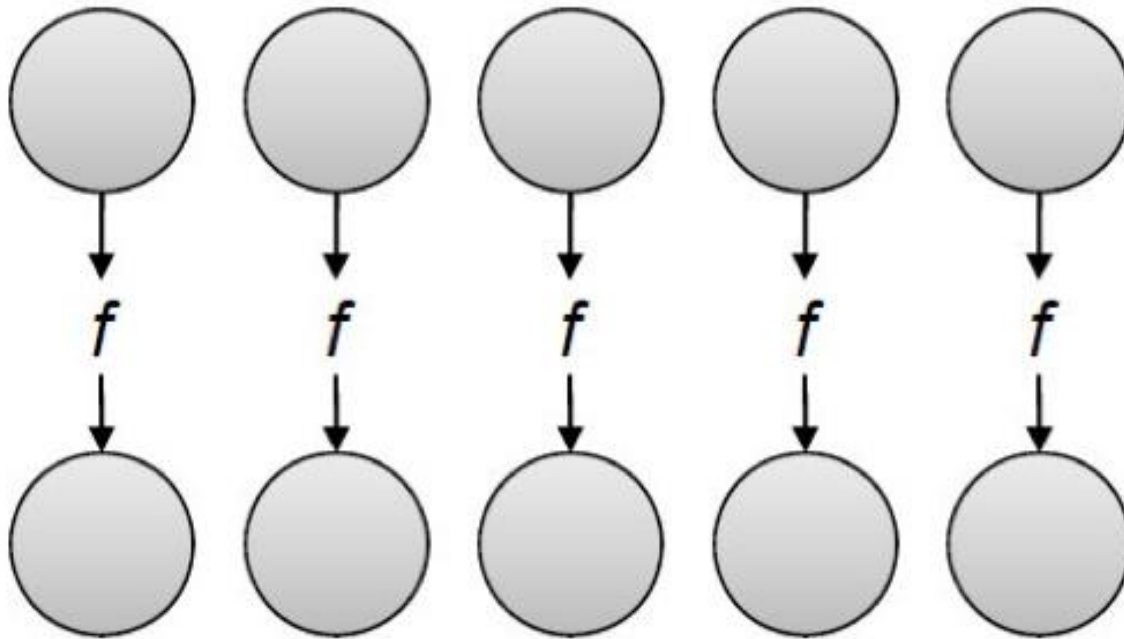
- *чтение $(k1, v1)$*
- *map: $(k1, v1) \rightarrow [(k2, v2)]$*
- *сортировка и группировка по ключу $k2$*
- *reduce: $(k2, [v2]) \rightarrow (k3, v3)$*

Идея MapReduce



Функциональное программирование

Map = apply-to-all



python

```
In: map(lambda x: x*x, range(1,5))
```

```
Out: [1, 4, 9, 16]
```

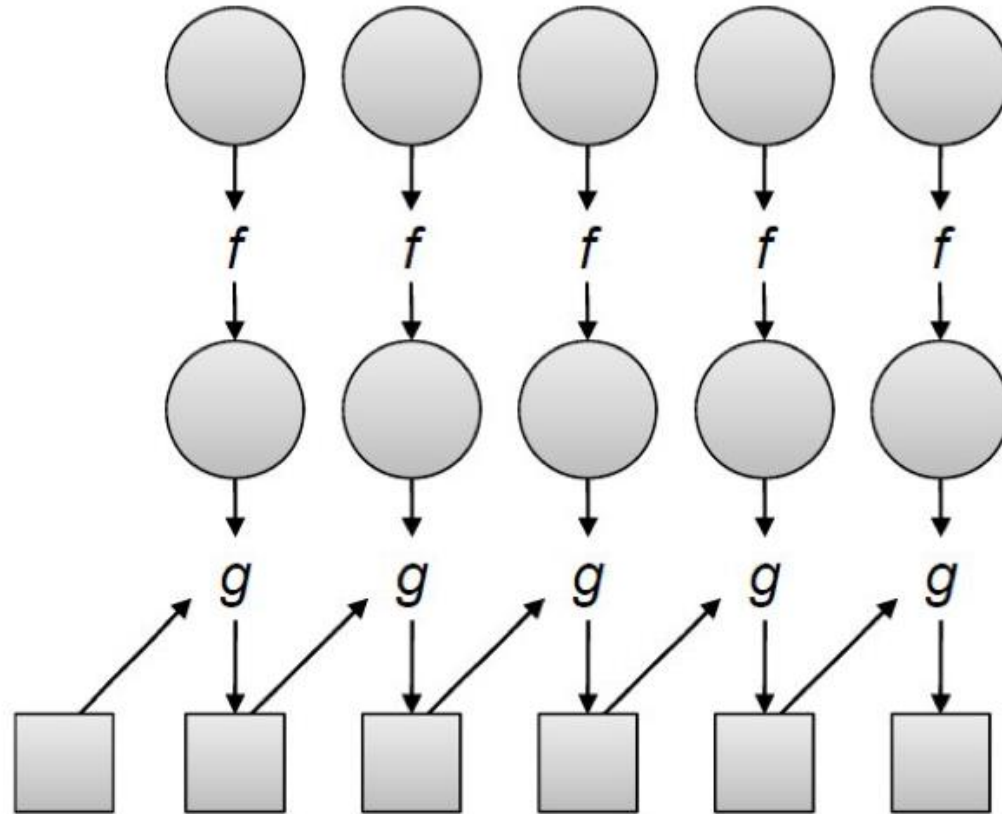
Функциональное программирование

Reduce =
aggregate

python

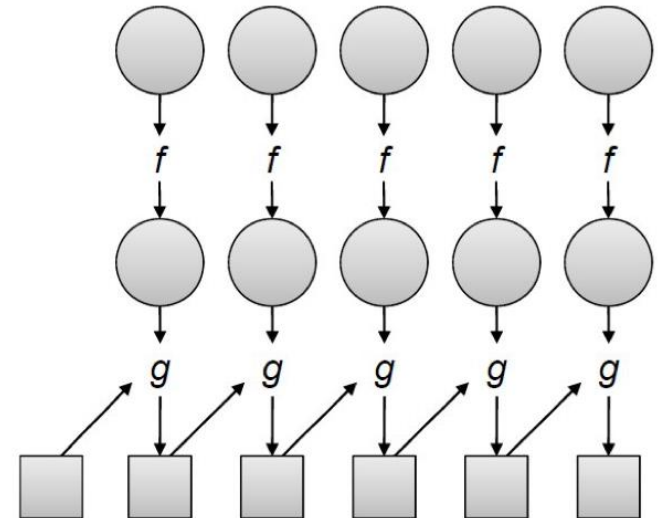
```
In: sum(map(lambda x: x*x, range(1,5)))
```

```
out: 30
```



Почему это работает

- Не изменяем данные, а создаем новые
- Считаем там, где лежат данные
 - пересылаем перед reduce
 - проще распараллелить: (почти) не нужна синхронизация
- Последовательное чтение и запись на диск
- Линейное ускорение при увеличении кластера



Google, 2004

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.

Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the

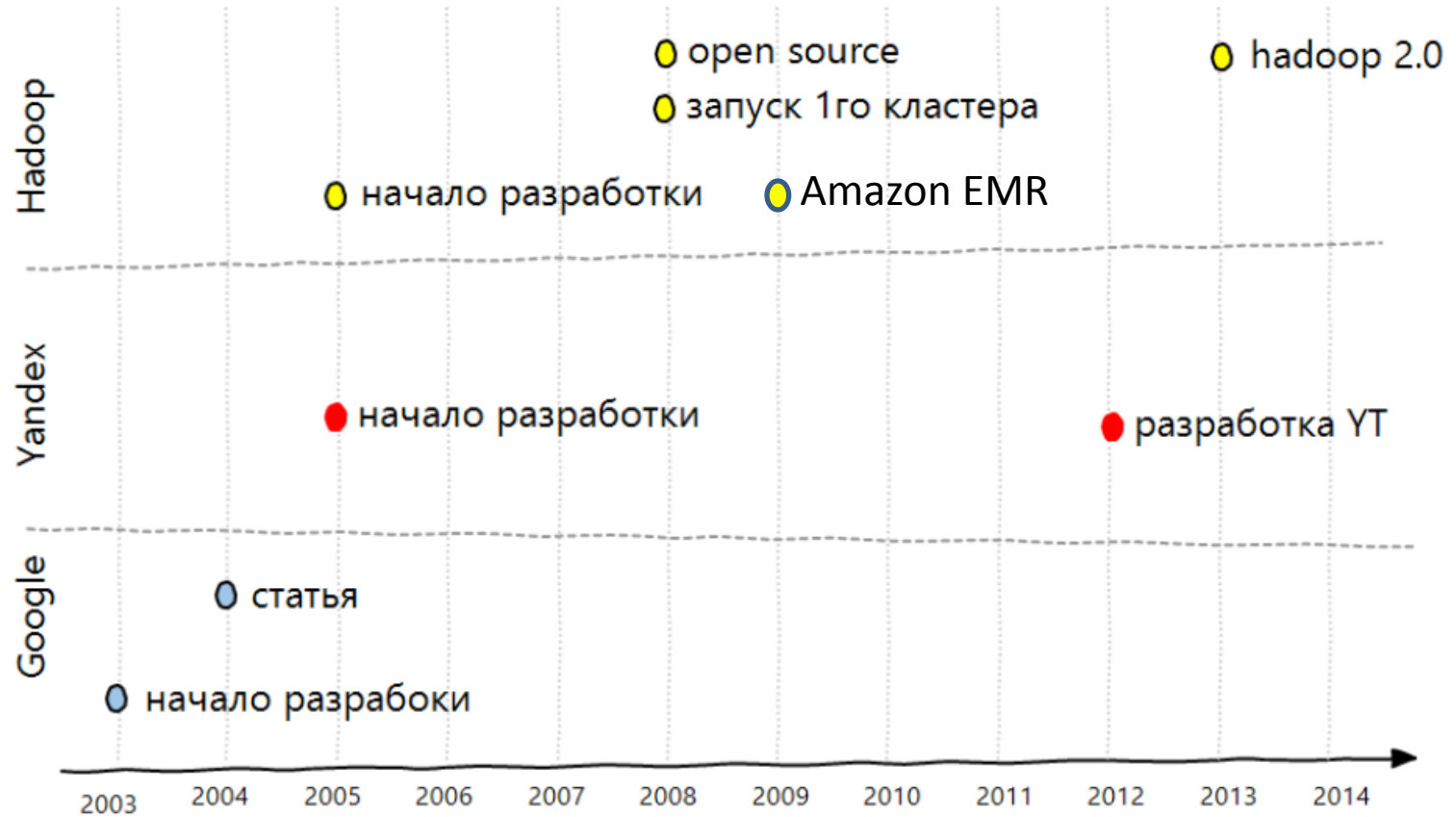
given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

As a reaction to this complexity, we designed a new abstraction that allows us to express the simple computations we were trying to perform but hides the messy details of parallelization, fault-tolerance, data distribution

Проблемы

- Только пакетная (batch) обработка
 - не подходит для realtime задач
- Нужно быть готовым к отказам серверов:
следить и перезапускать неуспешные задачи
- Распределять ресурсы между задачами
 - т.е. надежная реализация среды вычислений
- Удобные средства разработки
 - API для решения разных задач

Реализации



Сравнение

MPI

- Общее хранилище на Storage Area Network (SAN)
- Вычисления CPU-intensive
- Потоки сильно связаны, точки синхронизации
- Об отказах заботится разработчик

MapReduce

- Данные «размазаны» по всем серверам
- Disk-intensive, считаем там, где данные
- Слабая связь между потоками вычислений
- Об отказах заботится среда выполнения

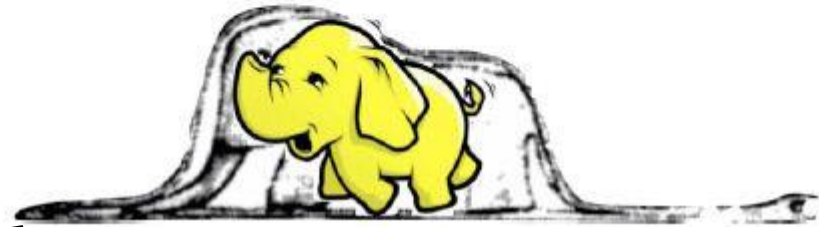
План лекции

- I. Введение
- II. Пример: статистика сайта
- III. MapReduce:
 - 1) идеи
 - 2) история и выводы
- **IV. Hadoop**
- V. Заключение. Как будет устроен курс

Hadoop

- Реализация MapReduce (Java)
- Open Source <http://hadoop.apache.org>
- Состав:
 - общие компоненты
 - HDFS
 - Hadoop MapReduce и YARN
- Дополнительно
 - Hive – запросы на SQL-подобном языке
 - Spark – вычисления в памяти
 - HBase – колонко-ориентированная БД
 - Mahout – алгоритмы машинного обучения
 - ...

Apache Hadoop is an
open-source system
to reliably store and process
gobs of information
across many commodity computers.



Работа с HDFS

```
$ hadoop fs [-ls] [-put] [-get] [-cat] [-help] ...
```

```
$ hadoop fs -ls /data/user_events
```

```
Found 1 items
```

```
-rw-r--r--    3 hdfs supergroup 2644746725 2014-12-01 18:48  
  /data/user_events/events_dt-20141014
```

```
$ hadoop fs -cat /data/user_events/events_dt-20141014
```

```
4759261629905287639      1413013675      http://worldoftanks.ru/ diff_time:9745  
2884716973831715191      1412319859      http://www.motorpage.ru/ diff_time:100  
-4146066766843543561      1412588829      http://afisha.mail.ru/ diff_time:25  
1428167192746476774      1413015139      http://www.championat.com/ diff_time:24  
...
```

В колонках: user_id, время, страница, время на странице

Программирование для Hadoop

- Hadoop Java API
 - реализовать классы Mapper, Reducer
 - класс для запуска задачи с main()
- Streaming
 - любой язык программирования; взаимодействие через stdin/stdout
 - Python: mrjob, dumbo, pydoop, hadoopу
<http://blog.cloudera.com/blog/2013/01/a-guide-to-python-frameworks-for-hadoop/>
- Pipes
 - C++, сокеты

Hadoop Java API: Mapper

Mapper<k1, v1, k2, v2>

```
public static class UserCountMapper
    extends Mapper<LongWritable, Text, Text, IntWritable>
{
    private final static IntWritable one = new IntWritable(1);
    private Text uid = new Text();

    @Override
        map(k1, v1, Context)
    public void map(LongWritable offset, Text line, Context
context) throws IOException, InterruptedException
    {
        парсинг строки
        String [] fields = line.toString().split("\\t");
        uid.set(fields[0]);

        context.write(uid, one);    write(k2, v2)
    }
}
```

Hadoop Java API: Reducer

Reducer<k2, v2, k3, v3>

```
public static class UserCountReducer extends Reducer<Text,
    IntWritable, Text, IntWritable>
{
    @Override reduce(k2, [v2], Context)
    protected void reduce(Text uid, Iterable<IntWritable> values,
        Context context) throws IOException, InterruptedException
    {
        int sum = 0;
        for (IntWritable value: values) { вычисления
            sum += value.get();
        }
        context.write(uid, new IntWritable(sum)); write(k3, v3)
    }
}
```

Hadoop Java API: main()

```
public class UserCount extends Configured implements Tool {  
    public static void main(String[] args) throws Exception {  
        ToolRunner.run(new UserCount(), args);  
    }  
    @Override public int run(String[] args) throws Exception {  
        Configuration conf = this.getConf();  
        Job job = new Job(conf);  
        job.setJarByClass(UserCount.class);  
        job.setMapperClass(UserCountMapper.class);  
        job.setReducerClass(UserCountReducer.class);  
        job.setInputFormatClass(TextInputFormat.class);  
        job.setOutputFormatClass(TextOutputFormat.class);  
        job.setMapOutputKeyClass(Text.class);  
        job.setMapOutputValueClass(IntWritable.class);  
        job.setNumReduceTasks(8);  
        FileInputFormat.addInputPath(job, new Path(args[0]));  
        FileOutputFormat.setOutputPath(job, new Path(args[1]));  
        return job.waitForCompletion(true) ? 0 : -1;  
    }  
}
```

разбор аргументов

создание задачи

классы Mapper и Reducer

форматы in и out

типы k3, v3

число редьюсеров

пути in и out

ждём завершения

Hadoop Java API: пуск

- Собираем jar:

```
$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d build src/UserCount.java
$ jar cf jar/UserCount.jar -C build .
```

- и запускаем Hadoop задачу

```
$ hadoop jar jar/UserCount.jar ru.mipt.UserCount
/data/user_events /user/gorokhov/usercount
```

```
15/02/11 03:53:51 WARN mapred.JobClient: Use
GenericOptionsParser for parsing the arguments.
Applications should implement Tool for the same.
```

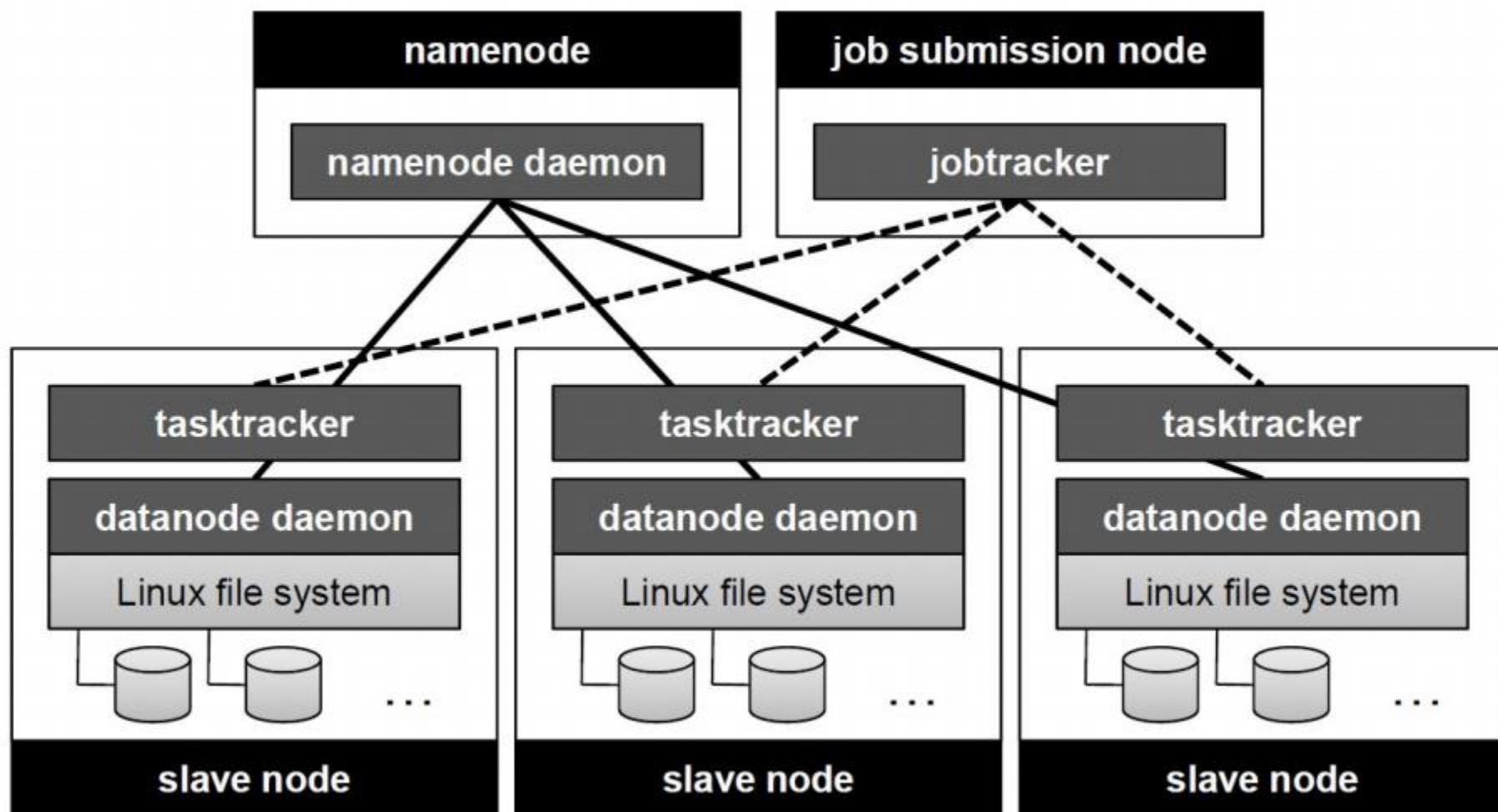
```
15/02/11 03:53:51 INFO input.FileInputFormat: Total input
paths to process : 1
```

```
15/02/11 03:53:52 INFO mapred.JobClient: Running job:
job_201411132102_2645
```

```
15/02/11 03:53:53 INFO mapred.JobClient: map 0% reduce 0%
```

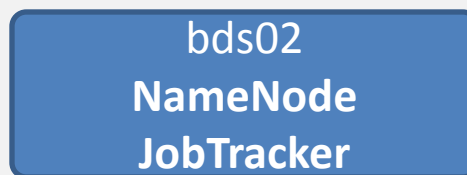
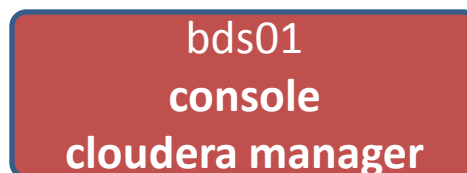
```
15/02/11 03:54:16 INFO mapred.JobClient: map 8% reduce 0%
```

Кластер Hadoop



Наш кластер

- 9 рабочих нод
- 510 Gb HDFS



- сервер для управления (bds01)
- центральный сервер кластера (bds02)
- 1 физическая машина
- Cloudera Distribution including Hadoop 5.2

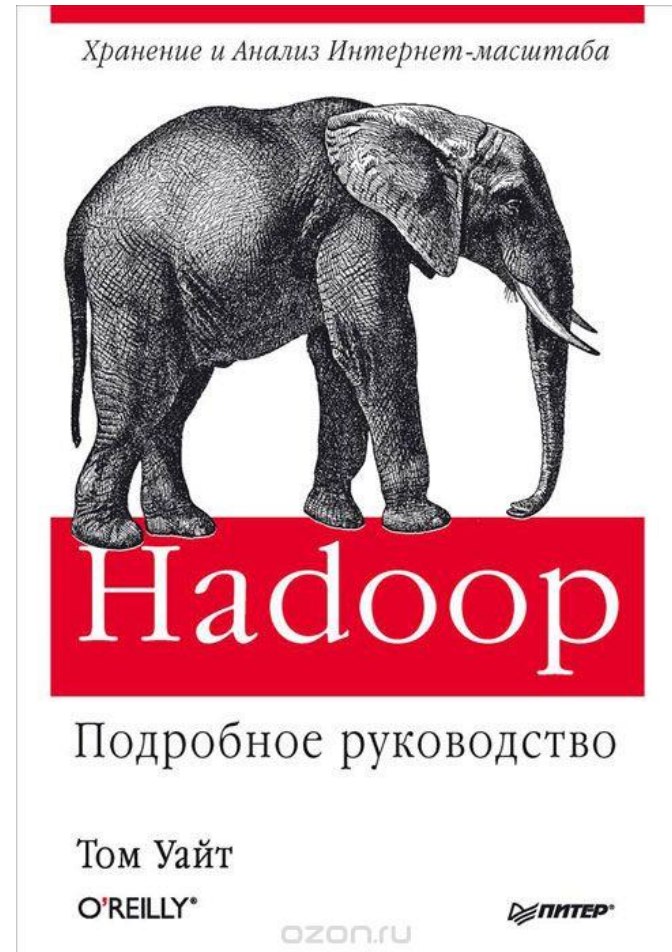
Дистрибутивы Hadoop

- Открытый
 - Apache <http://hadoop.apache.org>
- Коммерческие
 - Cloudera: CDH – Cloudera's Distribution inc. Hadoop
 - Hortonworks: HDP - Hortonworks Data Platform
 - MapR: M3, M5, M7
- Сервисы
 - Amazon Elastic MapReduce (EMR)
 - Microsoft Azure HDInsight
- Appliance (аппаратно-программные комплексы)
 - Oracle Exalogic
 - Teradata Appliance for Hadoop
 - IBM PureData for Analytics



Литература

- Tom White. Hadoop: The Definitive Guide, 3rd edition



План лекции

- I. Введение
- II. Пример: статистика сайта
- III. MapReduce:
 - 1) идеи
 - 2) история и выводы
- IV. Hadoop
- **V. Заключение. Как будет устроен курс**

План курса

Задания по темам

- статистика
- графы
- информационный поиск (IR)

NoSQL

Redis

Apache
Cassandra

Aero
spike

SQL-like

Hive

Column oriented DB

HBase

Resident distr. dataset (RDD)

Spark

MapReduce

YARN

HDFS

План курса

- HDFS – распределенная файловая система
- MapReduce, YARN – фреймворки для вычислений
- Hive – SQL-like язык для запуска MR задач
- HBase – база данных на HDFS
- Spark – вычисления в памяти на кластере
- NoSQL базы – для хранения и быстрой выдачи данных

Ближайшие планы


- Лекции
- Семинары
- Домашние задания
 - Новое задание – каждые 2 недели (7 – 8 заданий)

MapReduce


HDFS

Домашние задания


<https://github.com/agorokhov/mapreduce>
tasks.md


 GitHub, Inc. [US] <https://github.com/agorokhov/mapreduce/blob/master/tasks.md>

GitHub [Explore](#) [Features](#) [Enterprise](#) [Blog](#) [Sign up](#)

 **agorokhov / mapreduce** Watch 1 Star 0

branch: master **mapreduce / tasks.md** ⋮

 **agorokhov** on Dec 15, 2014 Update tasks.md
1 contributor

119 lines (80 sloc) | 11.515 kb Raw Blame History 

Часть 1 (Hadoop Java API)

1

Данные: статьи википедии (id -> text) `/data/wiki/en_articles_part`, `/data/wiki/en_articles`

Посчитать число вхождений слов (wordcount), начинающихся на ту же букву, что и ваше имя. Плюс за очистку

Состав докладчиков



Тинькофф
Банк



Нигма.РФ



Вопросы?

- I. Введение
- II. Пример: статистика сайта
- III. MapReduce:
 - 1) идеи
 - 2) история и выводы
- IV. Hadoop
- V. Заключение. Как будет устроен курс