

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической  
кибернетики и компьютерных наук

**РАБОТА С БАЗАМИ ДАННЫХ ПРИ ПОМОЩИ РАСПРЕДЕЛЕННОЙ  
СИСТЕМЫ NADOOR**

**АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ**

студента 4 курса 411 группы  
направления 02.03.02 — Фундаментальная информатика и информационные  
технологии  
факультета КНиИТ  
Балана Виктора Петровича

Научный руководитель

доцент, к. ф.-м. н.

\_\_\_\_\_

И. А. Батраева

Заведующий кафедрой

к. ф.-м. н.

\_\_\_\_\_

С. В. Миронов

Саратов 2016

## ВВЕДЕНИЕ

В настоящее время обработка больших данных стала одной из основных проблем для информационных систем, сети интернет, хранилищ данных любого типа. Одним из основных способов хранения и обработки данных становятся облачные сервисы, которые базируются на распределенных системах. Одной из технологий распределенной обработки данных является технология Hadoop, которая и стала предметом рассмотрения в настоящей работе. Так как обработка данных осуществляется на основе какого-либо хранилища, то для больших объемов данных были разработаны различные распределенные базы данных.

Поэтому в настоящей работе были поставлены задачи исследовать технологию Hadoop MapReduce, с точки зрения использования данной системы для работы с нереляционными распределенными базами данных HBase и MongoDB. Необходимо выяснить, на что способна эта технология, каковы ее преимущества и как ее использовать. В работе ставятся задачи:

- исследовать возможности нереляционной распределенной базы данных HBase;
- исследовать возможности нереляционной распределенной базы данных MongoDB;
- реализовать автоматическое заполнение баз данными и провести подсчет уникальных значений в ключевых строках таблиц для HBase и коллекций для MongoDB;
- сравнить скорость работы с базами данных при обработке входных данных на стадии Map и получения результата на шаге Reduce.

Настоящая работа представлена введением, шестью разделами, заключением, списком использованных источников и приложением (листингом программы).

Раздел 1. В данном разделе представлено описание распределенной системы Hadoop и история происхождения данной системы.

Раздел 2. В данном разделе представлено рассмотрение модели программирования MapReduce, которая ориентированна на обработку данных.

Раздел 3. В данном разделе представлено описание распределенной файловой системы HDFS, являющаяся основной файловой системой Hadoop MapReduce.

Раздел 4. В данном разделе представлено сравнение реляционных систем

управления базами данных с MapReduce.

Раздел 5. В данном разделе представлено описание нереляционных распределенных баз данных HBase и MongoDB, а также их архитектуры.

Раздел 6. В данном разделе представлены результаты серии экспериментов обработки большого объема данных, хранящихся в нереляционных распределенных базах данных HBase и MongoDB при помощи распределенной системы Hadoop MapReduce. [1–9]

## 1 Основное содержание работы

Введение работы содержит в себе общие положения работы.

Глава 1. «Hadoop» содержит описание системы Hadoop и историю возникновения этой системы.

Hadoop изобрел Дуг Каттинг — создатель Apache Lucene, широко используемой библиотеки текстового поиска. Hadoop происходит от Apache Nutch — системы веб-поиска с открытым кодом, которая сама по себе являлась частью проекта Lucene. Состоит из набора утилит, библиотек и фреймворк для разработки и выполнения распределенных программ, работающих на кластерах из сотен и тысяч узлов.

В январе 2008 года проект Hadoop стал одним из ведущих проектов Apache, что доказало его успех и наличие разнообразного, активного сообщества. К этому времени технология Hadoop использовалась не только в Yahoo!, но и во многих других компаниях — например, в Facebook и New York Times.

В апреле 2008 года технология Hadoop побила мировой рекорд по скорости сортировки терабайта данных. С того времени технология Hadoop стремительно входит в повседневную деятельность крупных компаний. Отрасль признала ее роль как платформы хранения и анализа данных общего назначения.

Глава 2. «MapReduce» содержит в себе подробное описание модели программирования MapReduce, которая ориентирована на обработку данных.

Модель MapReduce проста, но не настолько, чтобы в ее контексте нельзя было реализовать полезные программы. Hadoop позволяет запускать программы MapReduce, написанные на разных языках. Но самое важное заключается в том, что программы MapReduce параллельны по своей природе, а следовательно, крупномасштабный анализ данных становится доступным для всех, у кого в распоряжении имеется достаточно компьютеров. Достоинства MapReduce в полной мере проявляются в работе с большими наборами данных.

Работа MapReduce основана на разбиении обработки данных на две фазы: фазу отображения (map) и фазу свертки (reduce). Каждая фаза использует в качестве входных и выходных данных пары «ключ-значение», типы которых выбираются программистом. Программист также задает две функции: функцию отображения и функцию свертки.

Глава 3. «Файловая система HDFS» содержит в себе подробное описание

распределенной файловой системы HDFS.

Hadoop поставляется с распределенной файловой системой, которая называется HDFS (Hadoop Distributed Filesystem). HDFS — основная файловая система Hadoop.

Файловая система HDFS спроектирована для хранения очень больших файлов с потоковой схемой доступа к данным в кластерах обычных машин. Рассмотрим строение файловой системы более подробно.

1. Очень большие файлы. Под «очень большими» в этом контексте подразумеваются файлы, размер которых составляет сотни мегабайт, гигабайт и терабайт. Сейчас существуют кластеры Hadoop, в которых хранятся петабайты данных.
2. Поточковый доступ к данным. В основу HDFS заложена концепция однократной записи и многократного чтения как самая эффективная схема обработки данных. Набор данных обычно генерируется или копируется из источника, после чего с ним выполняются различные аналитические операции. В каждой операции задействуется большая часть набора данных (или весь набор), поэтому время чтения всего набора данных важнее задержки чтения первой записи.
3. Обычное оборудование. Hadoop не требует дорогостоящего оборудования высокой надежности. Система спроектирована для работы на стандартном оборудовании с достаточно высокой вероятностью отказа отдельных узлов в кластере (по крайней мере, для больших кластеров). Технология HDFS позволяет в случае отказа системы продолжать работу без сколько-нибудь заметного прерывания.

Глава 4. «Сравнение реляционных систем управления базами данных с MapReduce» содержит в себе рассуждения, в которых говорится какую систему нужно применять в конкретных заданиях.

Во многих отношениях MapReduce может рассматриваться как дополнение к технологии реляционных систем управления базами данных (РСУБД) (различия между двумя системами представлены на рисунке 1). MapReduce хорошо подходит для задач, которые требуют пакетного анализа всего набора данных. РСУБД хорошо подходят для точечных запросов или обновлений, при которых база данных была предварительно проиндексирована для обеспечения быстрой выборки и обновления при относительно небольших объемах

данных. MapReduce подходит для приложений, в которых данные записываются один раз, а читаются многократно, тогда как реляционные базы данных хорошо подходят для часто обновляемых наборов данных.

<b>Параметр</b>	<b>Традиционная РСУБД</b>	<b>MapReduce</b>
Размер данных	Гигабайты	Петабайты
Доступ	Интерактивный и пакетный	Пакетный
Обновления	Множественное чтение и запись	Однократная запись, многократное чтение
Структура	Статическая схема	Динамическая схема
Целостность	Высокая	Низкая
Масштабирование	Нелинейное	Линейное

Рисунок 1 – Сравнение РСУБД с MapReduce

Глава 5. «Базы данных HBase и MongoDB» содержит в себе подробное описание нереляционных распределенных баз данных HBase и MongoDB, а так же их архитектуры.

HBase — распределенная столбцовоориентированная база данных, построенная на основе HDFS. Это приложение Hadoop, используемое в ситуациях, когда вам необходимо организовать произвольный доступ для чтения или записи к очень большим наборам данных в реальном времени.

Приложения хранят данные в таблицах, состоящих из строк и столбцов. Для ячеек таблицы действует контроль версии. По умолчанию в качестве версии используется временная метка, автоматически назначаемая HBase на момент вставки. Содержимое ячейки представляет собой неинтерпретируемый массив байтов.

Ключи строк таблицы тоже являются байтовыми массивами. Строки таблицы сортируются по ключу строк. Сортировка осуществляется в порядке следования байтов. Все обращения к таблице выполняются по первичному ключу.

MongoDB — документоориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++. Для начала рассмотрим основные концепции:

1. MongoDB — концептуально то же самое, что обычная, привычная нам база данных. Внутри MongoDB может быть ноль или более баз данных, каждая из которых является контейнером для прочих сущностей.

2. База данных может иметь ноль или более «коллекций». Коллекция настолько похожа на традиционную «таблицу», что можно смело считать их одним и тем же.
3. Коллекции состоят из нуля или более «документов». Опять же, документ можно рассматривать как «строку».
4. Документ состоит из одного или более «полей», которые, как можно догадаться, подобны «колонкам».
5. «Индексы» в MongoDB почти идентичны таковым в реляционных базах данных.
6. «Курсоры» отличаются от предыдущих пяти концепций, но они очень важны и заслуживают отдельного обсуждения. Важно понимать, что когда у MongoDB запрашиваются какие-либо данные, то она возвращает курсор, с которыми можно делать все что угодно — подсчитывать, пропускать определенное число предшествующих записей, при этом не загружая сами данные.

Глава 6. «Реализация баз данных HBase и MongoDB и их анализ» содержит в себе реализацию модели Hadoop MapReduce и результаты экспериментов, которые были проведены над базами данных HBase и MongoDB.

Подробно рассмотрим реализацию работы модели Hadoop MapReduce. Сперва данные из базы передаются в Map, после стадии Map происходит сокращение объема данных, за счет того, что в переменной *text*, хранится уникальное значение (название города), а в переменной *one*, которая имеет тип *IntWritable(1)* хранятся значения равные единице, это указывает на то, сколько раз встречается название данного города. Например, если в переменной *text* записан город «Saratov», а в переменной *one* единица встречается двадцать раз, значит город «Saratov» в базе присутствует двадцать раз. Далее данные передаются на стадию Reduce, где непосредственно происходит выполнение операции *Count*, то есть будет произведен подсчет количества одинаковых городов из Table1 и Table2, результат будет выглядеть следующим образом (см. рисунок 2).

Для сравнения скорости работы модели Hadoop MapReduce с исследованными нами базами HBase и MongoDB, был проведен эксперимент. Были созданы базы данных, в которых было записано абсолютно идентичные данные как для HBase, так и для MongoDB. В первом тесте производилась запись

```
Терминал
viktor@Infinity: ~
hbase(main):022:0> scan 'Resultat'
ROW COLUMN+CELL
Adler column=data:count, timestamp=1464278150401, value=1
Anapa column=data:count, timestamp=1464278150401, value=1
Dallas column=data:count, timestamp=1464278150401, value=1
Dortmund column=data:count, timestamp=1464278150401, value=2
Hengels column=data:count, timestamp=1464278150401, value=1
Kazan column=data:count, timestamp=1464278150401, value=1
Kishinev column=data:count, timestamp=1464278150401, value=1
Krasnodar column=data:count, timestamp=1464278150401, value=2
Liverpool column=data:count, timestamp=1464278150401, value=2
London column=data:count, timestamp=1464278150401, value=2
Manchester column=data:count, timestamp=1464278150401, value=1
Monaco column=data:count, timestamp=1464278150401, value=1
Moscow column=data:count, timestamp=1464278150401, value=2
New-York column=data:count, timestamp=1464278150401, value=1
Paris column=data:count, timestamp=1464278150401, value=1
Rome column=data:count, timestamp=1464278150401, value=2
Rostov column=data:count, timestamp=1464278150401, value=1
Samara column=data:count, timestamp=1464278150401, value=2
Saratov column=data:count, timestamp=1464278150401, value=2
Sochi column=data:count, timestamp=1464278150401, value=1
Sverdlovsk column=data:count, timestamp=1464278150401, value=1
Ufa column=data:count, timestamp=1464278150401, value=1
22 row(s) in 0.1100 seconds
```

Рисунок 2 – Время обработки данных

данных в таблицы и коллекции в количестве 1500 строк, после чего данные отправлялись на стадию MapReduce. По итогам выполнения работы программы, результат был таков, что при использовании базы данных HBase программа была выполнена за 18,804 секунды, а при использовании MongoDB программа была выполнена за 10,524 секунды. Во втором тесте в таблицы и коллекции баз данных было записано 750000 строк. По итогам выполнения программы, работа с базой данных HBase была выполнена за 26,033 секунды, а работа с MongoDB была выполнена за 34,457 секунды, в данном тесте база данных HBase стала работать быстрее MongoDB на 25%. Третий тест показал, что при количестве записей 1500000 строк в таблицы и коллекции, работа программы с MongoDB была выполнена за 51,694 секунды, это показывает то, что MongoDB существенно начинает уступать в скорости HBase, которая была выполнена за 37,293 секунды. Результаты проведенных исследований представлены в таблице 1, а также по результатам проведенных исследований был построен график отображающий за какое время каждая из баз обрабатывает определенный объем данных (см. рисунок 3).



Таблица 1 – Сравнительная характеристика работы баз данных

Количество записей в таблице	Время подсчета в секундах
<b>HBase</b>	
1500 строк	17,804
750000 строк	26,033
1500000 строк	37,923
<b>MongoDB</b>	
1500 строк	10,524
750000 строк	34,457
1500000 строк	51,694

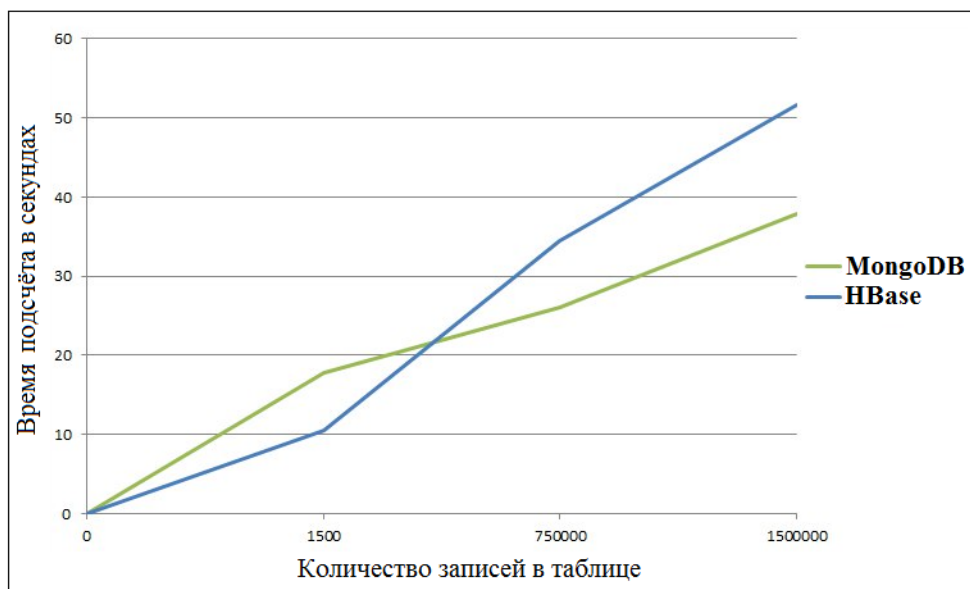


Рисунок 3 – Время обработки данных

## ЗАКЛЮЧЕНИЕ

В настоящей работе было произведено ознакомление с основами Hadoop. Было описано каким образом работает модель распределенных вычислений MapReduce, и почему нужно использовать именно данную технологию, а не традиционные реляционные системы управления базами данных. Подробно было рассмотрено строение файловой системы HDFS и каким образом обеспечивается целостность данных. По результатам проведенных тестов сравнения скорости работы модели MapReduce с нереляционными распределенными базами данных HBase и MongoDB, работа с HBase оставила больше положительных результатов. Во-первых из-за того, что она более легка в использовании, во-вторых работает более быстро чем MongoDB.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Уайт, Т. Hadoop: Подробное руководство. / Т. Уайт. — Москва, 2013. — С. 77–123.
- 2 Seguin, K. The Little MongoDB Book. / К. Seguin. — 2014. — URL: <http://jsman.ru/mongo-book/Glava-6-MapReduce.html> (дата обращения 25.05.2016) Загл. с экр. Яз. рус.
- 3 Henschen, D. MapReduce Tutorial. / D. Henschen. — 2016. — URL: <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html> (дата обращения 18.05.2016) Загл. с экр. Яз. рус.
- 4 Харрис, Д. Обработка больших данных. / Д. Харрис. — 2015. — URL: <http://alfalavista.ru/index.php/2013-06-18-22-25-47/336-hadoop-3> (дата обращения 14.06.2016) Загл. с экр. Яз. рус.
- 5 Гао, Д. Руководство по HBase. / Д. Гао. — 2016. — URL: <https://azure.microsoft.com/ru-ru/documentation/articles/hdinsight-hbase-tutorial-get-started> (дата обращения 11.06.2016) Загл. с экр. Яз. рус.
- 6 Моррисон, А. Hadoop. Модуль 2. Распределенная Файловая Система Hadoop (HDFS). / А. Моррисон. — 2015. — URL: <http://ruhadoop.blogspot.ru/2012/01/hdfs.html> (дата обращения 10.06.2016) Загл. с экр. Яз. рус.
- 7 Рысцов, Д. В чем польза ZooKeeper. / Д. Рысцов. — 2013. — URL: <https://habrahabr.ru/company/yandex/blog/234335/> (дата обращения 13.06.2016) Загл. с экр. Яз. рус.
- 8 Петухов, Д. Hadoop MapReduce. Основные концепции [электронный ресурс]. / Д. Петухов. — 2013. — URL: <http://www.codeinstinct.pro/2012/08/mapreduce-design.html> (дата обращения 26.05.2016) Загл. с экр. Яз. рус.
- 9 Джонс, М. Т. Распределенная обработка данных с помощью Hadoop: Часть 1. Начало работы [электронный ресурс]. / М. Т. Джонс. — 2011. — URL: <https://www.ibm.com/developerworks/ru/library/l-hadoop-1> (дата обращения 19.05.2016) Загл. с экр. Яз. рус.