

Министерство образования и науки Российской Федерации

Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Ю.С. Белов, С.С. Гришунов

**ОСНОВЫ HADOOP. УСТАНОВКА HADOOP. ОСНОВНЫЕ
КОМАНДЫ ФАЙЛОВОЙ СИСТЕМЫ HDFS**

Методические указания по выполнению лабораторной работы
по курсу «Технологии обработки больших данных»

Калуга - 2018

УДК 004.62
ББК 32.972.5
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий и прикладной математики».


Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий и прикладной математики» (ФН1-КФ) протокол № 6 от «12» января 2018 г.

Зав. кафедрой ФН1-КФ  д.ф.-м.н., профессор Б.М. Логинов

- Методической комиссией факультета ФНК протокол № 1 от «30» 01 2018 г.


Председатель методической
комиссии факультета ФНК

 к.х.н., доцент К.Л. Анфилов

- Методической комиссией

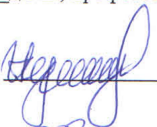
КФ МГТУ им.Н.Э. Баумана протокол № 1 от «06» 02 2018 г.

Председатель методической комиссии
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:



к.т.н., зав. кафедрой ЭИУ2-КФ

 И.В. Чухраев

Авторы

к.ф.-м.н., доцент кафедры ФН1-КФ

ассистент кафедры ФН1-КФ

 Ю.С. Белов
 С.С. Гришунов

Аннотация

Методические указания по выполнению лабораторной работы по курсу «Технологии обработки больших данных» содержат краткое описание Hadoop Distributed File System, порядок установки и конфигурирования платформы Hadoop, а также примеры команд для работы с файловой системой HDFS.

Предназначены для студентов 4-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2018 г.

© Ю.С. Белов, С.С. Гришунов, 2018 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
УСТАНОВКА НАDOOP.....	10
РАБОТА С HDFS.....	17
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	19
ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ.....	19
ВАРИАНТЫ ЗАДАНИЙ.....	19
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	21
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ	21
ОСНОВНАЯ ЛИТЕРАТУРА.....	22
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	22

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Технологии обработки больших данных» на кафедре «Программное обеспечение ЭВМ, информационные технологии и прикладная математика» факультета фундаментальных наук Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 4-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание Hadoop Distributed File System, порядок установки и конфигурирования платформы Hadoop, а также примеры команд для работы с файловой системой HDFS и задание на выполнение лабораторной работы.

Методические указания составлены для ознакомления студентов с платформой hadoop и овладения навыками работы с файловой системой HDFS. Для выполнения лабораторной работы студенту необходимы минимальные знания по программированию на высокоуровневом языке программирования (Java, Python или др.).

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является формирование практических навыков по установке и настройке кластера Hadoop и работе с файловой системой HDFS.

Основными задачами выполнения лабораторной работы являются:

1. Изучить основы Hadoop.
2. Научиться устанавливать и конфигурировать Hadoop
3. Изучить основные команды для работы с файловой системой HDFS.
4. Получить навыки написания программ для работы с HDFS

Результатами работы являются:

- Настроенный Hadoop-кластер
- Программа, использующая HDFS API, для решения задачи согласно варианту задания
- Подготовленный отчет

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Большие данные (big data) — серия подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети.

Одной из основных технологий распределенной обработки больших данных является Hadoop. Hadoop разработан на Java в рамках вычислительной парадигмы MapReduce, согласно которой приложение разделяется на большое количество одинаковых элементарных заданий, выполнимых на узлах кластера и естественным образом сводимых в конечный результат.

Настроить работу Hadoop можно как в режиме моносервера (подходит только для обучения, т.к. такой режим не позволяет ускорить обработку данных, распределяя задачи между серверами и по сути не отличается от работы обычного сервера), так и в режиме кластера, объединяющего группу серверов, что позволяет использовать множество узлов как единое хранилище данных, создавать реплики данных на случай отказа и распределять задачи по обработке данных между серверами.

Hadoop состоит из 4 основных модулей:

1. **Hadoop Common** — связующее программное обеспечение, набор инфраструктурных программных библиотек и утилит, используемых для других модулей и родственных проектов.
2. **Hadoop Distributed File System ([HDFS](#))** — распределённая файловая система, позволяющая хранить информацию практически неограниченного объёма.
3. **Hadoop YARN** — фреймворк для управления ресурсами кластера и менеджмента задач.
4. **Hadoop MapReduce** — платформа программирования и выполнения распределённых MapReduce-вычислений.

Также существует большое количество проектов, непосредственно связанных с Hadoop, но не входящих в Hadoop core, например:

1. **Hive** – инструмент для SQL-like запросов над большими данными (превращает SQL-запросы в серию MapReduce-задач);
2. **Pig** – язык программирования для анализа данных на высоком уровне. Одна строка кода на этом языке может превратиться в последовательность MapReduce-задач;
3. **Hbase** – колоночная база данных, реализующая парадигму BigTable;
4. **Cassandra** – высокопроизводительная распределенная key-value база данных;
5. **ZooKeeper** – сервис для распределённого хранения конфигурации и синхронизации изменений этой конфигурации;
6. **Mahout** – библиотека и движок машинного обучения на больших данных.

Hadoop Distributed File System

HDFS — распределенная файловая система, используемая в проекте Hadoop. HDFS-кластер в первую очередь состоит из NameNode-сервера и DataNode-серверов, которые хранят непосредственно данные. NameNode-сервер управляет пространством имен файловой системы и доступом клиентов к данным. Чтобы разгрузить NameNode-сервер, передача данных осуществляется только между клиентом и DataNode-сервером.

Secondary NameNode

Основной NameNode-сервер фиксирует все транзакции, связанные с изменением метаданных файловой системы, в log-файле, называемом EditLog. При запуске основного NameNode-сервера, он считывает образ HDFS (расположенный в файле FsImage) и применяет к нему все изменения, накопленные в EditLog. Затем записывается новый образ уже с примененными изменениями, и система начинает работу уже с чистым log-файлом. Следует заметить, что данную работу NameNode-сервер выполняет единожды при его первом запуске. В последующем, подобные операции возлагаются на вторичный

NameNode-сервер. FsImage и EditLog в конечном итоге хранятся на основном сервере.

Механизм репликации

При обнаружении NameNode-сервером отказа одного из DataNode-серверов (отсутствие heartbeat-сообщений от одного), запускается механизм репликации данных:

- выбор новых DataNode-серверов для новых реплик
- балансировка размещения данных по DataNode-серверам

Аналогичные действия производятся в случае повреждении реплик или в случае увеличения количества реплик присущих каждому блоку.

Стратегия размещение реплик

Данные хранятся в виде последовательности блоков фиксированного размера. Копии блоков (реплики) хранятся на нескольких серверах, по умолчанию — трех. Их размещение происходит следующим образом:

- первая реплика размещается на локальном узле
- вторая реплика на другом узле в этой же стойке
- третья реплика на произвольно узле другой стойки
- остальные реплики размещаются произвольным способом

При чтении данных клиент выбирает ближайшую к нему DataNode-сервер с репликой.

Целостность данных

Ослабленная модель целостности данных, реализованная в файловой системе, не гарантирует идентичность реплик. Поэтому HDFS перекладывает проверку целостности данных на клиентов. При создании файла клиент рассчитывает контрольные суммы каждые 512 байт, которые в последующем сохраняются на DataNode-сервере. При считывании файла, клиент обращается к данным и контрольным суммам. И, в случае их несоответствия происходит обращение к другой реплике.

Запись данных

При записи данных в [HDFS](#) используется подход, позволяющий достигнуть высокой пропускной способности. Приложение ведет запись в потоковом режиме, при этом HDFS-клиент кэширует записываемые данные во временном локальном файле. Когда в файле накапливаются данные на один HDFS-блок, клиент обращается к NameNode-серверу, который регистрирует новый файл, выделяет блок и возвращает клиенту список datanode-серверов для хранения реплик блока. Клиент начинает передачу данных блока из временного файла первому DataNode-серверу из списка. DataNode-сервер сохраняет данные на диске и пересылает следующему DataNode-серверу в списке. Таким образом, данные передаются в конвейерном режиме и реплицируются на требуемом количестве серверов. По окончании записи, клиент уведомляет NameNode-сервер, который фиксирует транзакцию создания файла, после чего он становится доступным в системе.

Удаление данных

В силу обеспечения сохранности данных (на случай отката операции), удаление в файловой системе происходит по определенной методике. Вначале файл перемещается в специально отведенную для этого /trash директорию, а уже после истечения определенного времени, происходит его физическое удаление:

- удаление файла из пространства имен HDFS
- освобождение связанных с данными блоков

УСТАНОВКА HADOOP

Необходимое ПО

Для установки Hadoop необходима операционная система Linux. Далее будет рассмотрена установка и настройка Hadoop для Ubuntu (возможно использование виртуальной машины).

Для работы Hadoop требуется следующее программное обеспечение: Java, ssh, rsync.

Сначала необходимо установить ssh и rsync, для этого нужно выполнить в терминале следующие команды:

```
sudo apt-get install ssh  
sudo apt-get install rsync
```

Для работы Hadoop можно использовать Java версии 6 и выше. Установить можно как версию от Oracle, так и OpenJDK, для этого нужно выполнить следующую команду

```
sudo apt-get install default-jdk
```

Создание учетной записи для Hadoop

Для запуска Hadoop будет использоваться отдельная учетная запись Linux. Этот шаг не является обязательным, но рекомендуется. Команды для создания группы hadoop и добавления в нее пользователя hduser, также предоставления новому пользователю права sudo:

```
sudo addgroup hadoop  
sudo adduser --ingroup hadoop hduser  
sudo usermod -aG sudo hduser
```

Предполагается, что все дальнейшие действия будут выполняться от созданного таким образом пользователя.

Настройка SSH

Hadoop требует доступ SSH для управления узлами. Необходимо настроить SSH доступ к каждому из узлов кластера для пользователя hduser, команды для генерации нового ssh ключа и добавления созданного ключа в список авторизованных:

```
ssh-keygen -t rsa -P ""  
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Для проверки подключения к localhost нужно выполнить команду:

```
hduser@ubuntu:~$ ssh localhost
```

```
The authenticity of host 'localhost (::1)' can't be established.
```

```
RSA key fingerprint is d7:87:25:47:ae:02:00:eb:1d:75:4f:bb:44:f9:36:26.
```

```
Are you sure you want to continue connecting (yes/no)? yes
```

```
Warning: Permanently added 'localhost' (RSA) to the list of known hosts
```

Распаковка Hadoop

Скачать файлы Hadoop можно с сайта www.apache.org/dyn/closer.cgi/hadoop/common/, также можно выполнить команду:

```
sudo wget http://apache-mirror.rbc.ru/pub/apache/hadoop/common/  
hadoop-2.8.1/hadoop-2.8.1.tar.gz
```

После скачивания архива, его необходимо распаковать и переместить файлы в каталог /usr/local/Hadoop, для этого нужно выполнить команды:

```
sudo mv hadoop-2.8.1.tar.gz /usr/local/  
cd /usr/local/  
sudo tar xzf hadoop-2.8.1.tar.gz  
sudo mv hadoop-2.8.1 hadoop
```

Также необходимо дать пользователю `hduser` права создателя на директорию:

```
chown -R hduser:hadoop Hadoop
```

Если узел будет являться `DataNode` или `NameNode`, то необходимо создать каталог, в котором будут храниться файлы HDFS, выполнив команды:

```
sudo mkdir -p var/app/hadoop/tmp
sudo chown hduser:hadoop var/app/hadoop/tmp
sudo chmod 750 /app/hadoop/tmp
```

Настройка переменных окружения

В файл `$HOME/.bashrc` добавим следующие переменные окружения:

```
#Hadoop variables
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
```

Настройка Hadoop

Финальным шагом является конфигурирования работы кластера, для этого необходимо задать в файлах конфигурации значения соответствующих параметров.

В файле `$HADOOP_INSTALL/etc/hadoop/hadoop-env.sh` необходимо задать переменную `JAVA_HOME`:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

Основные настройки Hadoop выполняются в файле `$HADOOP_INSTALL/etc/hadoop/core-site.xml`, в котором указывается имя файловой системы (в одном кластере может физически быть несколько файловых систем, однако настроить взаимодействие между ними стандартными средствами не представляется возможным), а также порт, по которому можно к ней обратиться:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
<property>
  <name>hadoop.tmp.dir</name>
  <value>var/app/hadoop/tmp</value>
</property>
</configuration>
```

Настройки HDFS для каждого узла хранятся в файле `$HADOOP_INSTALL/etc/hadoop/hdfs-site.xml`:

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop/tmp/hdfs/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop/tmp/hdfs/datanode</value>
</property>
</configuration>
```

Параметр `dfs.replication` задает количество [реплик](#), которые будут храниться на файловой системе. Также в этом файле прописываются все узлы файловой системы, присутствующие на данной машине (все [dataNode](#) и [nameNode](#)).

Настройки MapReduce прописываются в файле `$HADOOP_INSTALL/etc/hadoop/mapred-site.xml`:

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Настройка фреймворка управления ресурсами кластера YARN производится в файле `$HADOOP_INSTALL/etc/hadoop/yarn-site.xml`:

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce_shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>localhost:8025</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>localhost:8030</value>
  </property>
```

```
<property>
  <name>yarn.resourcemanager.address</name>
  <value>localhost:8050</value>
</property>
</configuration>
```

Отключение IPv6

Для стабильной работы Hadoop, необходимо отключить IPv6 в файле `$HADOOP_INSTALL/etc/hadoop/hadoop-env.sh`:

```
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

На каждом узле также хранится файл `slaves`, в котором перечислены все дочерние узлы данного master-узла. Т.е. для NameNode-узла необходимо перечислить в этом файле сетевые имена всех узлов кластера, а на остальных серверах этот файл должен оставаться пустым.

Форматирование HDFS

После завершения конфигурирования, необходимо отформатировать файловую систему HDFS. Для этого на NameNode необходимо выполнить команду:

```
$HADOOP_INSTALL/bin/hadoop namenode -format
```

Для запуска Hadoop необходимо запустить следующие службы на master-узле (на всех дочерних узлах необходимые демоны запускаются автоматически, используя сконфигурированные подключения по ssh):

```
$HADOOP_INSTALL/sbin/start-dfs.sh
$HADOOP_INSTALL/sbin/start-yarn.sh
```

Если Hadoop был запущен правильно будут запущены следующие java процессы:

\$HADOOP_INSTALL/jps
4868 SecondaryNameNode
5243 NodeManager
5035 ResourceManager
4409 NameNode
4622 DataNode
5517 Jps

Web-интерфейс

Hadoop также имеет несколько web-интерфейсов для получения информации о работе системы. По умолчанию:

1. <http://localhost:50070/> – NameNode
2. <http://localhost:50030/> – JobTracker
3. <http://localhost:50060/> – TaskTracker

РАБОТА С HDFS

Для работы с hdfs в терминале существует набор команд, схожих с командами работы с файловой системой в unix-системах (просмотр файлов, создание, удаление, просмотр и т.д.), для их вызова необходимо указать исполняемый файл `hadoop` и в виде ключа передать команду, например, для создания каталога необходимо выполнить команду:

```
hadoop fs -mkdir /usr/hduser
```

Полный перечень доступных команд можно просмотреть в прилагаемой к дистрибутиву `hadoop` документации.

Следует выделить следующие команды, не имеющие аналогов среди стандартных команд работы с файловой системой:

`-copyFromLocal <localsrc> ... <dst>]` – позволяет скопировать файл из локальной файловой системы в hdfs

`-copyToLocal [<src> ... <localdst>]` – обратная команда, позволяющая скопировать файл из hdfs в локальную файловую систему.

Также, как упоминалось выше, возможно управление файлами через [web-интерфейс](#).

HDFS API

Помимо управления файлами через консольный режим доступен Java hdfs api, содержащий набор функций для управления файлами в java программе. Все необходимые библиотеки входят в дистрибутив `hadoop`, но также могут быть скачены, используя сборщик проектов `maven`, путем указания следующих зависимостей в pom-файле:

```
<dependency>  
  <groupId>org.apache.hadoop</groupId>  
  <artifactId>hadoop-hdfs</artifactId>  
  <version>2.9.0</version>  
</dependency>
```

После подключения всех необходимых библиотек необходимо сконфигурировать файловую систему:

```
Configuration conf = new Configuration();  
conf.set("fs.default.name", hdfsPath);
```

После этого с hdfs можно работать как и с обычной файловой системой, например, для создания каталога необходимо использовать команду:

```
client.mkdir(name, conf);
```

Помимо java hdfs api, распространяемого вместе с дистрибутивом hadoop, существует множество библиотек для работы с hdfs, используя различные языки программирования, такие как python, scala и др.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Для всех вариантов настроить кластер Hadoop, состоящий из двух серверов, изучить команды HDFS для работы с файлами и выполнить следующие задания:

1. Проверить существует ли директория /user/hduser в HDFS, если нет, то создать. Создать директорию /user/hduser/Hadoop
2. Создать файл в директории /user/hduser/hadoop, название файла – ваше имя и группа. После создания файла, все, что вы вводите в консоль должно сохраниться в файле. Ввести несколько строк и сохранить.
3. Убедиться в существовании файла через web-интерфейс.
4. Перенести файл в локальную файловую систему.
5. Создать новый текстовый файл в локальной файловой системе. Перенести файл в HDFS. Убедиться в существовании файла через web-интерфейс.
6. Просмотреть права доступа на файл. Изменить права доступа к файлу, чтобы только владелец и члены группы имели полный контроль над файлом.
7. Написать программу на каком-либо языке высокого уровня для решения задачи, указанной в варианте.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

Программа может быть реализована на любом языке высокого уровня, для которого существует поддержка работы с HDFS (Java, Python, Scala или др.). Имена файлов должны передаваться приложению в качестве ключей при вызове в терминале.

ВАРИАНТЫ ЗАДАНИЙ

1. Напишите программу, которая будет выводить на экран список подкаталогов и файлов для заданного каталога в HDFS.
2. Напишите программу, которая будет выводить на экран содержимое файла в HDFS.

3. Напишите программу, которая будет копировать файл из локальной файловой системы в систему HDFS.
4. Напишите программу, которая будет рекурсивно выводить на экран список подкаталогов и файлов для заданного каталога в HDFS.
5. Напишите программу, которая будет выводить на экран список всех подкаталогов и файлов в заданной директории HDFS, которые были изменены в промежуток между start_ts и end_ts (передаются через параметры командной строки).
6. Напишите программу, которая будет копировать все содержимое каталога из файловой системы HDFS в локальную.
7. Напишите программу, которая будет принимать 2 входных аргумента – путь в локальной файловой системе и путь в HDFS. Программа должна проверить существование файлов в обеих файловых системах. Если в одной из них файл не существует, то программа должна скопировать его из второй файловой системы. Если файлы существуют в обеих файловых системах, то сохранить в обеих системах только файл, который был изменен позже.
8. Напишите программу, которая будет для заданного каталога в HDFS добавлять в содержимое всех файлов из этого каталога текущую дату.
9. Напишите программу, которая будет сравнивать содержимое двух текстовых файлов в HDFS.
10. Напишите программу, которая рекурсивно удаляет все файлы из заданного каталога HDFS.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Раскройте область применения платформы Hadoop.
2. Опишите назначение основных модулей Hadoop.
3. Опишите назначение механизма репликации.
4. Раскройте значение терминов NameNode и DataNode.
5. Изложите роль SecondaryNameNode.
6. Опишите процесс записи файла в HDFS.
7. Перечислите основные этапы установки Hadoop-кластера.
8. Перечислите ПО, необходимое для установки Hadoop.
9. Опишите назначение основных файлов конфигурации Hadoop.
10. Дайте определения master-узла и slave-узла.
11. Приведите команду для копирования файла из локальной файловой системы в HDFS.
12. Изложите основные методы управления файловой системой HDFS.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 3 занятия (6 академических часов: 5 часов на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), этапы выполнения работы (со скриншотами), результаты выполнения работы. выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Федин Ф.О. Анализ данных. Часть 1. Подготовка данных к анализу [Электронный ресурс] : учебное пособие / Ф.О. Федин, Ф.Ф. Федин. — Электрон. текстовые данные. — М. : Московский городской педагогический университет, 2012. — 204 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/26444.html>
2. Федин Ф.О. Анализ данных. Часть 2. Инструменты Data Mining [Электронный ресурс] : учебное пособие / Ф.О. Федин, Ф.Ф. Федин. — Электрон. текстовые данные. — М. : Московский городской педагогический университет, 2012. — 308 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/26445.html>
3. Чубукова, И.А. Data Mining [Электронный ресурс] : учеб. пособие — Электрон. дан. — Москва : , 2016. — 470 с. — Режим доступа: <https://e.lanbook.com/book/100582>. — Загл. с экрана.
4. Воронова Л.И. Big Data. Методы и средства анализа [Электронный ресурс] : учебное пособие / Л.И. Воронова, В.И. Воронов. — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 33 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/61463.html>
5. Юре, Л. Анализ больших наборов данных [Электронный ресурс] / Л. Юре, Р. Ананд, Д.У. Джефффри. — Электрон. дан. — Москва : ДМК Пресс, 2016. — 498 с. — Режим доступа: <https://e.lanbook.com/book/93571>. — Загл. с экрана.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

6. Волкова Т.В. Разработка систем распределенной обработки данных [Электронный ресурс] : учебно-методическое пособие / Т.В. Волкова, Л.Ф. Насейкина. — Электрон. текстовые данные. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2012. — 330 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/30127.html>
7. Кухаренко Б.Г. Интеллектуальные системы и технологии [Электронный ресурс] : учебное пособие / Б.Г. Кухаренко. —

Электрон. текстовые данные. — М. : Московская государственная академия водного транспорта, 2015. — 116 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47933.html>

8. Воронова Л.И. Интеллектуальные базы данных [Электронный ресурс] : учебное пособие / Л.И. Воронова. — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2013. — 35 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/63324.html>
9. Николаев Е.И. Базы данных в высокопроизводительных информационных системах [Электронный ресурс] : учебное пособие / Е.И. Николаев. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 163 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/69375.html>

Электронные ресурсы:

10. <http://hadoop.apache.org/> (англ.)