МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Развёртывание системы распределённых вычислений Hadoop**

Отчет по лабораторной работе №4  
по дисциплине

«Большие данные»

Выполнил студент гр. ИТб-4302-02-00 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Шумилов М.Р./

(Подпись)

Руководитель ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Родионов К.В./

(Подпись)

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Цель: научиться развёртывать и настраивать распределённую систему «Hadoop» для обработки больших данных.

В ходе работы необходимо выполнить следующее задание:

* настройка брандмауэра для корректной работы «Hadoop»;
* установка необходимого ПО на сервера кластера;
* создание пользователя и генерация «SSH» ключей;
* настройка файлов конфигурации «HDFS», «YARN», «MapReduce»;
* форматирование файловой системы и запуск сервисов;
* тестирование работы кластера с помощью задачи «Word Count».

Для выполнения лабораторной работы установим «Hadoop» на Linux Ubuntu версии «20.04». Используем 3 сервера:

* hadoop1, 192.168.1.15 (мастер);
* hadoop2, 192.168.1.20 (слейв);
* hadoop3, 192.168.1.25 (слейв).

Для корректной работы кластера нам нужно открыть следующие порты:

* 9870 – веб-интерфейс для управления;
* 8020 – RPC адрес для клиентских подключений;
* 9866 – «DataNode» (передача данных);
* 9864 – «DataNode» (http-сервис);
* 9867 – «DataNode» (IPC-сервис).

Для этого необходимо выполнить следующий код:

|  |
| --- |
| iptables -I INPUT -p tcp --dport 9870 -j ACCEPT  iptables -I INPUT -p tcp --dport 8020 -j ACCEPT  iptables -I INPUT -p tcp --match multiport --dports 9866,9864,9867 -j ACCEPT |

Для сохранения правил используем утилиту «netfilter-persistent», выполнив следующий код:

|  |
| --- |
| apt install iptables-persistent  netfilter-persistent save |

«Hadoop» разработан на языке программирования «Java», поэтому устанавливаем «OpenJDK 8», для этого необходимо выполнить следующий код:

|  |
| --- |
| apt install default-jdk |

Необходимо скачать бинарный файл с сайта разработчика и разместить на сервере, создать файлы с переменными окружения и настроить автозапуск с помощью «systemd». Данные действия выполняем на всех серверах. Также необходимо обеспечить возможность подключения по «SSH» ко всем серверам кластера.

Используя ссылку, загружаем на серверы архив, выполнив следующий код:

|  |
| --- |
| wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1.tar.gz |

Установим и настроим среду окружения. Для этого создадим каталог и распакуем содержимое загруженного ранее архива в него, выполнив следующий код:

mkdir /usr/local/hadoop

tar -zxf hadoop-\*.tar.gz -C /usr/local/hadoop --strip-components 1

Создаем пользователя «hadoop» и задаём ему пароль. Задаем в качестве владельца каталога ранее созданного пользователя, для этого необходимо выполнить следующий код:

useradd hadoop -m

passwd hadoop

chown -R hadoop:hadoop /usr/local/Hadoop

Создаем файл с профилем «/etc/profile.d/hadoop.sh» и записываем в него следующий код:

export HADOOP\_HOME=/usr/local/hadoop

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_OPTS="$HADOOP\_OPTS -Djava.library.path=$HADOOP\_HOME/lib/native"

export YARN\_HOME=$HADOOP\_HOME

export PATH="$PATH:${HADOOP\_HOME}/bin:${HADOOP\_HOME}/sbin"

Для работы «hadoop» нужен сертификат, так как внутренние обращения выполняются с помощью запросов «SSH». Необходимо сгенерировать его на одном из серверов и скопировать на остальные. Для этого заходим под пользователем «hadoop», создаём ключи и копируем публичный ключ на локальный компьютер, выполнив следующий код:

$ su - hadoop

$ ssh-keygen

$ ssh-copy-id localhost

Отредактируем некоторые конфигурационные файлы (на всех узлах кластера), для этого открываем файл для общих настроек «/usr/local/hadoop/etc/hadoop/core-site.xml», приведя его к виду:

...

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://hadoop1:9000</value>

</property>

</configuration>

Где «fs.default.name» указывает на узел и порт обращения к внутренней файловой системе. В данном примере на мастер-сервер «localhost» порту 9000. Данная настройка должна быть такой на всех нодах.

Редактируем файл «/usr/local/hadoop/etc/hadoop/hdfs-site.xml» с настройками файловой системы «HDFS», приведя его к виду:

...

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<name>dfs.name.dir</name>

<value>file:///hadoop/hdfs/namenode</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>file:///hadoop/hdfs/datanode</value>

</property>

</configuration>

«HDFS» является распределенной файловой системой, разработанной для хранения больших файлов, которые блочно распределяются между узлами вычислительного кластера. Блоки в «HDFS» имеют одинаковый размер (за исключением последнего блока файла), и их репликация обеспечивает устойчивость системы к отказам узлов.

Открываем для редактирования файл «/usr/local/hadoop/etc/hadoop/mapred-site.xml» для настройки «MapReduce» и задаём следующие параметры:

...

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

Основная идея «Hadoop MapReduce» заключается в том, чтобы разбивать задачу на два этапа: «Map» (отображение) и «Reduce» (сведение). В процессе «Map» данные разбиваются на небольшие куски, которые обрабатываются параллельно на различных узлах кластера. Затем результаты этапа «Map» объединяются и сортируются по ключу, и передаются на этап «Reduce». На этом этапе данные снова разбиваются и обрабатываются параллельно, а затем объединяются в конечный результат.

Открываем файл «/usr/local/hadoop/etc/hadoop/yarn-site.xml» для настройки YARN, приведя его к виду:

...

<configuration>

<!-- Site specific YARN configuration properties -->

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

«ResourceManager» в «YARN» (Yet Another Resource Negotiator) - центральный компонент, управляющий ресурсами в кластере. Его основные функции включают распределение ресурсов, планирование выполнения задач, мониторинг состояния нод, отслеживание приложений, и распределение ресурсов. «YARN» поддерживает параллельное выполнение различных задач в рамках кластера.

Открываем файл с узлами кластера «/usr/local/hadoop/etc/hadoop/workers» и перечислим все slave-узлы:

haddop2

haddop3

На мастер-сервере заходим под пользователем «hadoop», создаем файловую систему и запускаем кластеры, выполнив следующий код:

su - hadoop

$ /usr/local/hadoop/bin/hdfs namenode -format

$ /usr/local/hadoop/sbin/start-dfs.sh

$ /usr/local/hadoop/sbin/start-yarn.sh

После запуска кластеров проверяем работу, открывая «http://<IP-адрес мастер-сервера>:9870» сайт в браузере (рисунок 1).

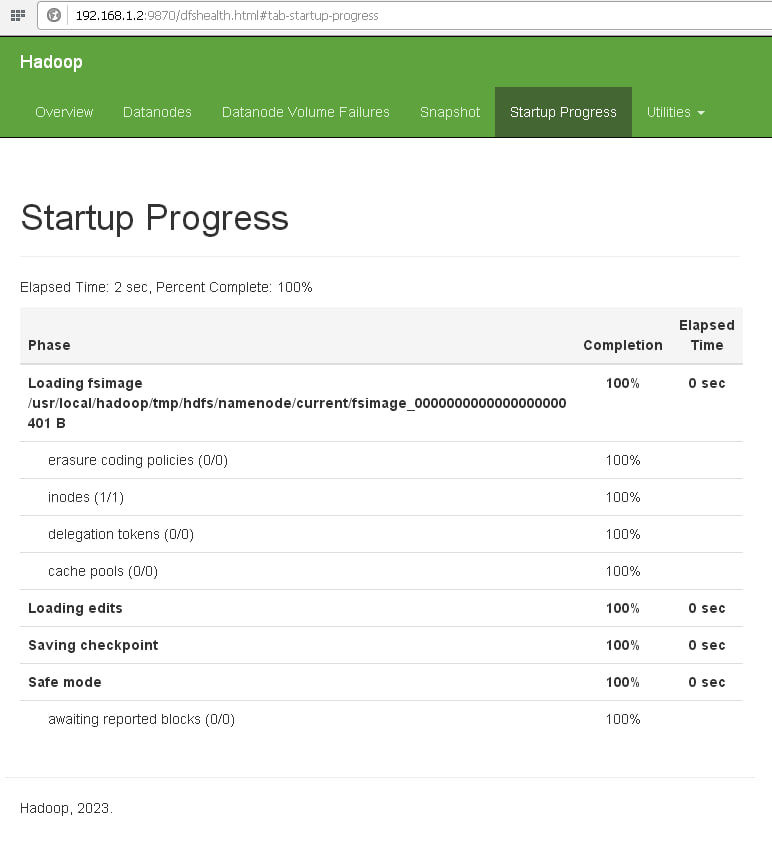


Рисунок 1 – Результат запуска кластеров

Выше представлен результат запуска кластеров Hadoop – веб-интерфейс для мониторинга и управления кластером. В нём видим информацию о состоянии нод кластера, использовании ресурсов и т.д.

Для запуска задачи «Word Count» необходимо загрузить в «HDFS» текстовый файл, для примера возьмем книгу «Гари Поттер: Философский камень», выполнив следующий код:

cd /home/hadoop1

mkdir books

cd books

wget <https://www.rulit.me/download-books-167949.html?t=txt>

cd /usr/local/Hadoop

bin/hdfs dfs -mkdir /in

bin/hdfs dfs -copyFromLocal /home/hduser/books/\* /in

bin/hdfs dfs -ls /in

Запустим «Word Count», выполнив следующий код:

/usr/local/hadoop/bin/hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount /in /out

В результате создался текстовый файл «part-r-00000» содержимое которого представлено на рисунке 2.

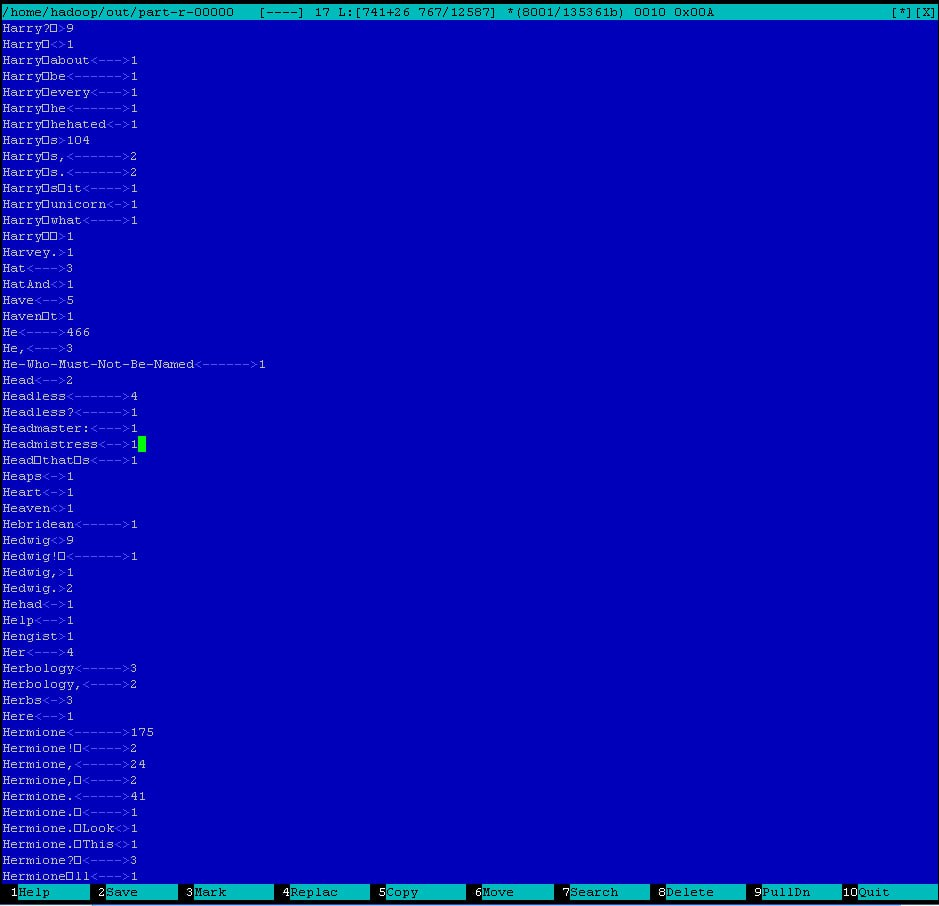


Рисунок 2 – Результат выполнения задачи «Word Count»

"Word Count" – подсчет количества вхождений слов в тексте. В нём видим список слов и количество их вхождений в анализируемом тексте

В ходе выполнения лабораторной работы научились развёртывать распределённую систему обработки больших данных «Hadoop» на нескольких серверах. Были установлены и настроены компоненты «HDFS», «YARN», «MapReduce». Тестирование кластера выполнялось с помощью запуска типовой задачи «Word Count».