**ASSIGNMENT 5**

**MAPREDUCE WORD COUNT**

**Harshil Patel – 40163431**

REQUIREMENTS :

Any Linux distro

JDK 8

Apache Hadoop 3.3.0 distribution

(Downlaod and extract zip to any folder)

Save the following variables for future use

Find java path by using “which javac” command then using “readlink -f <output of which javac command>”

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP\_CLASSPATH=$JAVA\_HOME/lib/tools.jar

Steps:

sudo -apt update

sudo apt install openjdk-8-jdk -y (If Java is not installed)

Install openssh to establish connection to localhost

sudo apt install openssh-server openssh-client -y

Generate an SSH key pair and define the location is is to be stored in:

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

Use the cat command to store the public key as authorized\_keys in the ssh directory:

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

Set the permissions for your user with the chmod command:

chmod 0600 ~/.ssh/authorized\_keys

Verify everything by ssh to localhost :

ssh localhost

**SINGLE NODE HADOOP DEPLOYMENT (Pseudo Distributed Mode)**

Edit following config files:

1. bashrc

sudo nano .bashrc

Copy the following data to the end of the file

export HADOOP\_HOME=/home/patel/Downloads/hadoop-3.3.0

export HADOOP\_INSTALL=$HADOOP\_HOME

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

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

export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin

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

Apply the changes to the current running environment by using the following command:

source ~/.bashrc

2. hadoop-evn.sh

sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

Add the following values in the above file :

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HDFS\_NAMENODE\_USER="patel"

export HDFS\_DATANODE\_USER="patel"

export HDFS\_SECONDARYNAMENODE\_USER="patel"

export YARN\_RESOURCEMANAGER\_USER="patel"

export YARN\_NODEMANAGER\_USER="patel"

3. core-site.xml

Create tmpdata directory in Downloads folder

sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/home/patel/Downloads/tmpdata</value>

</property>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

4. hdfs-site.xml

Create the directories dfsdata, namenode and datanode inside Downloads folder first.

sudo nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

<configuration>

<property>

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

<value>/home/patel/Downloads/dfsdata/namenode</value>

</property>

<property>

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

<value>/home/patel/Downloads/dfsdata/datanode</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

</configuration>

5. mapred-site.xml

sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

6. yarn-site.xml

sudo nano $HADOOP\_HOME/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.hostname</name>

<value>localhost</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME,HADOOP\_COMMON\_HOME,HADOOP\_HDFS\_HOME,HADOOP\_CONF\_DIR,CLASSPATH\_PERPEND\_DISTCACHE,HADOOP\_YARN\_HOME,HADOOP\_MAPRED\_HOME</value>

</property>

</configuration>

**Format the HDFS NAMENODE**

hdfs namenode -format

**START HADOOP CLUSTER**

Navigate to hadoop-3.3.0/sbin directory and run the following command:

./start-dfs.sh

If you get the following error, run the next command. Otherwise skip this step

“localhost: rcmd: socket: Permission denied”

echo 'echo "ssh" > /etc/pdsh/rcmd\_default' | sudo -s

./start-dfs.sh

Start yarn manager :

./start-yarn.sh

Type this simple command to check if all the daemons are active and running as Java processes:

jps

You will see the following processes running:

4865 SecondaryNameNode

4657 DataNode

5218 NodeManager

5364 Jps

4518 NameNode

5085 ResourceManager

ACCESS HADOOP UI FROM BROWSER through following url

<http://localhost:9870>

The default port 9864 is used to access individual DataNodes directly from your browser:

<http://localhost:9864>

The YARN Resource Manager is accessible on port 8088:

<http://localhost:8088>

**RUNNING THE MAPREDUCE PROGRAM**

Program and required files are stored inside following directory

/home/patel/Downloads/data/

File list :

WordCount.java

test1.txt (test file)

test2.txt (test file)

stopwords.txt

Navigate to Hadoop-3.3.0 directory

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP\_CLASSPATH=$JAVA\_HOME/lib/tools.jar

1. Compile the WordCount.java file

bin/hadoop com.sun.tools.javac.Main /home/patel/Downloads/data/WordCount.java

2. Create the required Directories inside the Hadoop File System:

hadoop fs -mkdir /WordCount

hadoop fs -mkdir /WordCount/Input

3. Upload the Input files in HDFS

hadoop fs -put '/home/patel/Downloads/data/test1.txt' /WordCount/Input/

hadoop fs -put '/home/patel/Downloads/data/test2.txt' /WordCount/Input/

hadoop fs -put '/home/patel/Downloads/data/stopwords.txt' /WordCount/

4. Executing the JAR

Create JAR file by the following command, the class files for JAR are located in data directory where we compiled out WordCount.java

jar -cvf wordcount.jar -C /home/patel/Downloads/data/ .

Finally, run the Hadoop application:

Format : hadoop jar <jar\_name> <Class Name containing psv main function> <Input file location> <Output file location> <Extra arguments>

hadoop jar wordcount.jar WordCount /WordCount/Input /WordCount/Output /WordCount/stopwords.txt

To check output in terminal

hadoop dfs -cat /WordCount/Output/\*

This will display the word count after removing stop words from the input file

Extra

hadoop fs -rm -r /WordCount/Output

To delete existing directories in hadoop file system

hdfs dfs -rm -r <directory>