Aum Sri Sai Ram

MDSC-304(P)
Hadoop Programming - Practicals
Lab Report -Word Count

Date 06 Sept 2021

Tasks Accomplished:

- Implemented Word count using Apache Hadoop, Map-Reduce.

Procedure:

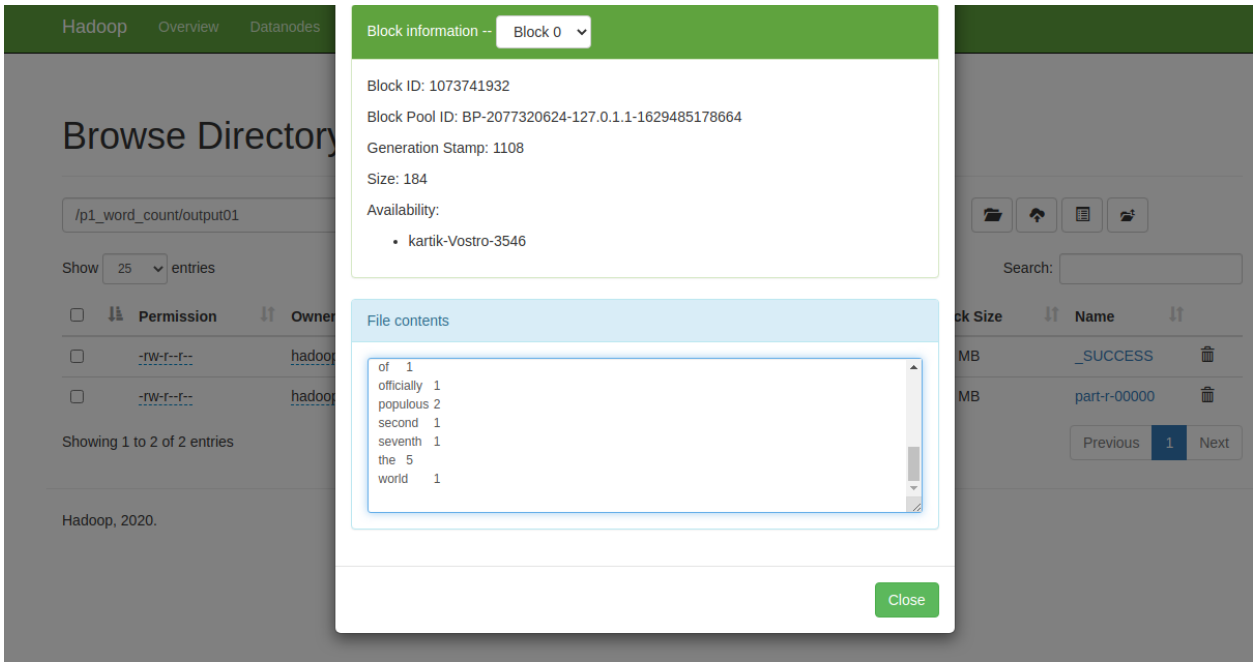
- Started the Hadoop Cluster
-
```
hadoop@kartik-Vostro-3546:~/programs_hadoop/nnum_of_lines$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [kartik-Vostro-3546]
```
-
```
hadoop@kartik-Vostro-3546:~/programs_hadoop/nnum_of_lines$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```
- Keep the text file in hdfs , on which the word count is to be performed .
-
```
hdfs dfs -put '/home/hadoop/India.txt' /p1_word_count/input
```
- Wrote the map class and reducer class in Java File for solving word count problem using hadoop
- Exported the Hadoop Classpath
-
```
hadoop@kartik-Vostro-3546:~/word_count$ export HADOOP_CLASSPATH=$(hadoop classpath)
hadoop@kartik-Vostro-3546:~/word_count$ echo $HADOOP_CLASSPATH
/home/hadoop/hadoop/etc/hadoop:/home/hadoop/hadoop/share/hadoop/common/lib/*:/home/had
/hadoop/hdfs:/home/hadoop/hadoop/share/hadoop/hdfs/lib/*:/home/hadoop/hadoop/share/had
```
- Run the file with javac and kept the classes file in a specified folder
-
```
hadoop@kartik-Vostro-3546:~/word_count$ javac -classpath ${HADOOP_CLASSPATH} -d '/hom
e/hadoop/word_count/j_files' '/home/hadoop/word_count/WordCount.java'
```
- Make a jar file with those classes.

```
hadoop@kartik-Vostro-3546:~/word_count$ jar -cvf WordCount.jar -C j_files/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1752) (out= 764)(deflated 56%)
adding: WordCount$IntSumReducer.class(in = 1755) (out= 749)(deflated 57%)
adding: WordCount.class(in = 1511) (out= 825)(deflated 45%)
```

- Run the hdfs jar command to execute the program and also passed the source and destination folder

```
hadoop@kartik-Vostro-3546:~/word_count$ hadoop jar '/home/hadoop/word_count/WordCount
.jar' WordCount /p1_word_count/input /p1_word_count/output01
```

```
                Peak Reduce Virtual memory (bytes)=2745217024
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=197
        File Output Format Counters
                Bytes Written=184
```



Input File:

```
India , officially the Republic of India  is a country in South Asia. It is the second most
populous country the seventh largest country by land area , and the most populous democracy in
the world
```

Output File:

```
Asia.   1
India   2
It      1
Republic        1
South   1
a       1
and     1
area    1
by      1
country         3
democracy       1
in      2
is      2
land    1
largest         1
most    2
of      1
officially      1
populous        2
second  1
seventh         1
the     5
world   1
```

Problems occurred:

- ➔ Making the jar file
- ➔ Compiling the java program
- ➔ Running the hdfs with the jar file
- ➔ [All problems were solved by writing the apt. Syntax for each thing]

Learnings:

From this lab session learned about

- ★ Writing the Map function and the reduce Function in Java.
- ★ Compiling the java program and using compiled classes to make a jar file.

★ Viewing the hdfs files in the localhost
★ Using hadoop framework to do word count.

*** Reg.No: 20231 ***