Counting Words after Removing Stop Words in File using MapReduce

Data Science and Analytics [IF 7202]
Assignment-II
-Prepared by
ARULALAN V
[Reg. No. : 18194391475]
Research Scholar, Dept. of Information Technology, MIT.

1 OVERVIEW

This document fills in as an instructional exercise to setup and run a straightforward application in Hadoop MapReduce framework. A vocation in Hadoop MapReduce more often than not parts input data-set into independent hurls which are prepared by delineate. Afterward, the output frame maps are arranged and afterward input to the reduce tasks. Typically every one of the outputs are stored in file frameworks. With a specific end goal to run an application an occupation client will presents the activity which can be a JAR file or an executable to a single ace in Hadoop called Resource Manager. This ace will at that point distribute tasks, configure nodes, and monitor tasks and timetable tasks. In addition, every one of the files for correspondence in the framework should be moved to Hadoop File System (HDFS); the client needs to sustain input files into the HDFS index and the output files will likewise be spared in HDFS directories. This instructional exercise will stroll through of these principle ventures by running an application that will count the number of words in file(s) after removing stop-words. The application will run it in a Single Node setup.

2 SETUP

2.1 Prerequisites

# Linux System [here ubuntu-16.04.3 is used].
# Java-8-oracle jdk.
# Download the stable release of Hadoop from one of the Apache Download Mirrors.

2.2 Check HADOOP and Java

# To check the HADOOP installation [hduser@alan $ hadoop version]
# To check the java installation [hduser@alan $ java -version]

2.3 Run the daemons

After installation and configuration of Hadoop, The hadoop daemons could be started by typing the command [hduser@alan $ start-all.sh], this will start the following nodes viz. namenode, datanode, NodeManager, ResourceManager, Jps and secondarynamenode.

To check running nodes type [hduser@alan $ jps].

To check the web interface for NameNode, go to http://Localhost:50070/ by default.

3 EXECUTION STEPS:

3.1 Compiling WordCount.java

Create a Folder in your system where .java file and input files can be stored.

```
$mkdir '/home/hduser/Desktop/WCSW/'
```

```
$mkdir '/home/hduser/Desktop/WCSW/input_files'
```

Compile your .Java code in terminal to create a class file, before that create a folder in local repository to hold class files.

```
$mkdir '/home/hduser/Desktop/WCSW/classfiles/
$javac –classpath${HADOOP_CLASSPATH} –d 'home/hduser/Desktop/WCSW/classfiles' 'home/hduser/Desktop/WCSW/WordCount.java'
```

3.2 Creating Directories in HDFS

After the above step create a directory for the WordCount application in Hadoop File System.

```
$hadoop fs –mkdir/WordCount/
```

Inside WordCount directory, create another folder for Input

```
$hadoop fs –mkdir/WordCount/Input
```

To check go to http://Localhost:50070/ ->Browse the File System.

3.3 Uploading Input files in HDFS

After creating directories in Hadoop File System, upload your input file (Shakespeare Dataset) and you're StopWord.txt in HDFS Input directory.

```
$hadoop fs –put '/home/hduser/Desktop/WCSW/input_files/f7.txt' /WordCount/Input/
$hadoop fs –put '/home/hduser/Desktop/WCSW/StopWord.txt' /WordCount/
```

3.4 Execute the JAR

To create the JAR file use the below command

```
$ jar –cvf wordcount.jar –C classfiles/.
```

Now JAR file is created, with that now run the Hadoop application

```
$ hadoop jar wordcount.jar org.myorg.WordCount /WordCount/Input /WordCount/Output -skip /WordCount/StopWords.txt
```

To check the output in Terminal type

`$hadoop dfs –cat /WordCount/Output/*` this will display the word count after removing stop words from the input file.

4 REFERENCES

The references for running this application were found in the hadoop website. The contents which is present in the following websites were useful for this program.

1. Setting Java_Home: Link
2. Setting Hadoop_Classpath: Link
3. Files copied from local system to HDFS: Link
4. Deprecation of DFS: Link
5. HDFS commands: Link

Note: Source Code is in **src_1 Folder and Screenshots of tasks is in **sample_screenshots** folder.*