

Lab 1

Roll No.: J019 – Ayush Hendre, J031—Rohit Mittal

Aim: Word Count Using Map Reduce

Objectives:

- 1.To run Java command.
2. Copy Data file from Local to HDFS.
3. Generate a Word count query.
4. Display Word count of the file

Code & Output:

WCDriver

```
//Driver:
// Importing libraries
import
java.io.IOException;
import
org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import
org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapred.FileInputFormat;
import
org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import
org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid
            inputs"); return -1;
        }
    }
}
```



```

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new
        Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new
        Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        JobClient.runJob(conf);
        return 0;
    }

    // Main Method
    public static void main(String args[]) throws Exception
    {
        int exitCode = ToolRunner.run(new WCDriver(),
        args); System.out.println(exitCode);
    }
}

```

WCMapper

Mapper:

```

// Importing libraries
import
java.io.IOException;
import
org.apache.hadoop.io.IntWritable;
import
org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import
org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

```

```

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,

```

Text, Text, IntWritable> {

```

// Map function
public void map(LongWritable key, Text value,
                OutputCollector<Text, IntWritable> output, Reporter
                rep) throws IOException
{
    String line = value.toString();
    // Splitting the line on
    spaces for (String word :
    line.split(" "))
    {
        if (word.length() > 0)
        {
            output.collect(new Text(word), new IntWritable(1));
        }
    }
}

```

WCReducer

```

//Reducer:

// Importing libraries
import
java.io.IOException;
import
java.util.Iterator;
import
org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
i      m      p      o      r      t
org.apache.hadoop.mapred.MapReduceBase;
i      m      p      o      r      t
org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,
                                                                    IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> value,

```

```
OutputCollector<Text, IntWritable>  
    output,
```

```
Reporter rep) throws IOException
```

```

    {

        int count = 0;

        // Counting the frequency of each
        words while (value.hasNext())
        {

            IntWritable i =
            value.next(); count +=
            i.get();

        }

        output.collect(key, new IntWritable(count));

    }
}

```

```

hive> CREATE TABLE FILES1 (line STRING);
OK
Time taken: 0.099 seconds
hive> LOAD DATA INPATH 'random2.txt' OVERWRITE INTO TABLE FILES1;
Loading data to table default.files1
chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/files1/random2.txt': User does not belong to supergroup
Table default.files1 stats: [numFiles=1, numRows=0, totalSize=152, rawDataSize=0]
OK
Time taken: 0.507 seconds

```

```

cloudera@quickstart:~
File Edit View Search Terminal Help
[cloudera@quickstart workspace]$ hadoop jar WordCount.jar WCDriver random4.txt W
Output
21/02/27 10:18:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/02/27 10:18:31 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/02/27 10:18:32 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
21/02/27 10:18:32 INFO mapred.FileInputFormat: Total input paths to process : 1
21/02/27 10:18:32 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
21/02/27 10:18:32 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1281)
    at java.lang.Thread.join(Thread.java:1355)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)

```

```

cloudera@quickstart:~/workspace
File Edit View Search Terminal Help
This      2
a         2
hive      1
is        2
spark     1
tutorial  1
tutorial. 1
[cloudera@quickstart workspace]$
[cloudera@quickstart workspace]$

```

