

EXP 2: Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

AIM:

To run a basic Word Count MapReduce program.

Procedure:

Ensure that Hadoop is properly installed and configured on WSL. If it's not installed, follow the [Hadoop installation guide for WSL](#).

2. Write the Word Count Program

Create a Java file for the Word Count program. Here's a simple example:

WordCount.java

java

Copy code

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
```

```
public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
```

```
    String[] words = value.toString().split("\\s+");
```

```
    for (String wordStr : words) {
```

```
        word.set(wordStr);
```

```
        context.write(word, one);
```

```
    }
```

```
}
```

```
}
```

```
public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
```

```
    private IntWritable result = new IntWritable();
```

```
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
```

```
    int sum = 0;
```

```
    for (IntWritable val : values) {
```

```
        sum += val.get();
```

```
    }
```

```
    result.set(sum);
```

```
    context.write(key, result);
```

```
}
```

```
}
```

```
public static void main(String[] args) throws Exception {
```

```
    Configuration conf = new Configuration();
```

```
    Job job = Job.getInstance(conf, "word count");
```

```
    job.setJarByClass(WordCount.class);
```

```
    job.setMapperClass(TokenizerMapper.class);
```

```
    job.setCombinerClass(IntSumReducer.class);
```

```
    job.setReducerClass(IntSumReducer.class);
```

```
    job.setOutputKeyClass(Text.class);
```

210701238

```
        job.setOutputValueClass(IntWritable.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));

        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true) ? 0 : 1);

    }

}
```

3. Compile the Word Count Program

Compile the WordCount.java program using Hadoop's classpath:

bash

Copy code

```
javac -classpath $(hadoop classpath) -d WordCount_classes WordCount.java
```

Create a JAR file from the compiled classes:

bash

Copy code

```
jar cf wordcount.jar -C WordCount_classes/ .
```

4. Run Hadoop Word Count

Upload your input file to HDFS:

bash

Copy code

```
hdfs dfs -mkdir /input
```

```
hdfs dfs -put /path/to/your/inputfile.txt /input
```

Run the Hadoop job:

bash

Copy code

```
hadoop jar wordcount.jar WordCount /input /output
```

5. Check the Output

After the job completes, check the output:

bash

Copy code

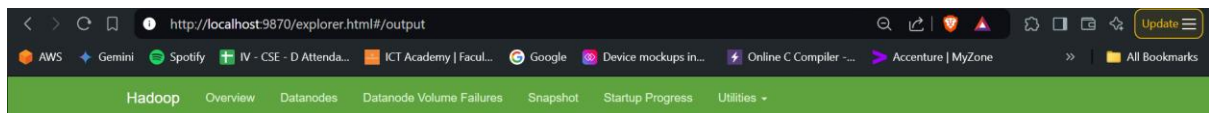
```
hdfs dfs -cat /output/part-r-00000
```

210701238

```
senthil@senthil-2463:~$ vi WordCount.java
senthil@senthil-2463:~$ javac -classpath $(hadoop classpath) -d WordCount_classes WordCount.java
javac: directory not found: WordCount_classes
Usage: javac <options> <source files>
use -help for a list of possible options
senthil@senthil-2463:~$ hadoop classpath
/usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/hadoop/share/hadoop/common/*:/usr/local/hadoop/share/hadoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoop/mapreduce/lib/*:/usr/local/hadoop/share/hadoop/mapreduce/*:/usr/local/hadoop/share/hadoop/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/*
senthil@senthil-2463:~$ mkdir -p WordCount_classes
senthil@senthil-2463:~$ javac -classpath /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/*:/usr/local/hadoop/share/hadoop/hdfs/lib/*:/usr/local/hadoop/share/hadoop/hdfs/*:/usr/local/hadoop/share/hadoop/mapreduce/lib/*:/usr/local/hadoop/share/hadoop/mapreduce/*:/usr/local/hadoop/share/hadoop/yarn/lib/*:/usr/local/hadoop/share/hadoop/yarn/* -d WordCount_classes WordCount.java
senthil@senthil-2463:~$ jar cf wordcount.jar -C WordCount_classes/ .
senthil@senthil-2463:~$ hadoop jar wordcount.jar WordCount /input /output

senthil@senthil-2463:~$ hdfs dfs -mkdir /input
2024-09-11 16:18:12,801 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
senthil@senthil-2463:~$ hdfs dfs -ls /
2024-09-11 16:18:36,787 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 3 items
drwxr-xr-x - senthil supergroup          0 2024-09-11 16:18 /input
drwxr-xr-x - senthil supergroup          0 2024-09-11 16:07 /temp
drwx----- - senthil supergroup          0 2024-09-11 16:17 /tmp
senthil@senthil-2463:~$ hdfs dfs -put /home/senthil/wct.txt /input
2024-09-11 16:20:46,560 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2024-09-11 16:20:47,713 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
senthil@senthil-2463:~$ hdfs dfs -ls /input
2024-09-11 16:21:02,197 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r--  2 senthil supergroup          28 2024-09-11 16:20 /input/wct.txt

senthil@senthil-2463:~$ hdfs dfs -ls /output
2024-09-11 16:23:03,224 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--  2 senthil supergroup          0 2024-09-11 16:21 /output/_SUCCESS
-rw-r--r--  2 senthil supergroup        33 2024-09-11 16:21 /output/part-r-00000
senthil@senthil-2463:~$ hdfs dfs -cat /output/part-r-00000
2024-09-11 16:23:11,532 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2024-09-11 16:23:12,190 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
Senthil 1
all 1
hi 3
is 1
this 1
senthil@senthil-2463:~$
```



<https://chatgpt.com/c/66e11ec5-56d8-800d-bd81-02c05c2e836e>