Assignment 2: Word Count

* Aim: Design a distributed application using MapReduce which process a text file. List out the count of each word occurring in the file.

* Theory:

Q1) Explain the need & concept of MapReduce?

Ans: - MapReduce is a software framework which is used to write applications to process huge amounts of data simultaneously on a large number of nodes in reliable way.

MapReduce is required when we need to process huge amounts of data that cannot be handled with regular frameworks.

Map Reduce primarily refers to two tasks:

- i) Map: Here one set of data is converted into another set of data in which the individual elements are broken down into tuples is into key/value pairs.
- 2) Reduce: This task takes the output of the map task as its input and combines the data tuples into a smaller tuples. It is

FOR EDUCATIONAL USE

	always performed after the map task.	
_	Advantages of MapReduce-Francwork:	
2000	1) Easy to scale 2) Handles task scheduling, monitoring, failure, & execu	ti- a
sit a r	3 Fault tolerant of and	MON &
	4) Simple & easy to understand 5) Has support for unstructed data	•
	14 001	
	map() Apall to Apones & bosn out in 192	
10 pl	reduce()	- anA
	3 (map()) & & & & & & & & & & & & & & & & & &	0
alub de et	reduce()	
	map()	
	lop reduce primary refers to two casis	
Q2)	Explain the following are a made to be an order gets	
)	Job Tracker	
Ans:	- Job Trocker is the master for job management, scheduli execution in the Hadoop framework.	ng &
<u>Sundaram</u>	FOR EDUCATIONAL USE	

2000年1月日4日日日日

81 23	Initially the user copies files into HDFS with the -put or the - copy FromLocal commands. The job is submitted via the job tracker. It runs on the same node which runs other jobes on data nodes.
	The job is initialized in the job queue & the job tracker creates maps & reduces. The map & reduce tasks will depend on the input programs that user provides.
Joseph	i) Resource management i) Aesource Availability iii) Monitoring iv) Scheduling
2)	Task Tracker
Ans	- The task tracker accepts tasks assigned by job tracker on the master node while itself running on slave nodes.
	It divides the JVM (Java Virtual Machine) processes & threads to run these tasks. The task tracker reports the progress of these tasks & health status.
undaram	FOR EDUCATIONAL USE

undaram

11-jac	Hadoop maintains 3 lists for task trackers:
	= 3Janu mings Jacolmond ygas =
	i) Blacklist: used to blacklist a task tracker if performance is not optimal or unstable.
note	not optimal or unstable.
	The state of the s
	ii) Grey list: a list of potentially problematic nodes
in col	and restant de out a comment of the last tender of the last tender one
dugar	iii) Excluded list: list of excluded task trackers.
	pregrams unit user provides
*	Conclusion: Union the Madredum Framework application to
1	Conclusion: Using the MapReduce Framework, application to design a distributed app to process a text file & list out word count has been successfully designed.
	count has been successfully designed and a second
	parliater out somewall (if
	phoning/A Curl
	portubación de la contraction
	national alease (2)
201 1	Ans - The look uncless accepts to be a signal by job tractor. Ans - The look uncless accepts to be a signal and allow tractors.
	COCOT STATE OF PLANT DESCRIPTION OF PLANT OF PARTY OF PAR
d also	all d'esecució (compala lorany and) may sel caballo of 4
<u>अल्ब</u> न्त	run bluer losks the took trocker reports the pregress of
	Ex see Land Stockus - autout all adeat
Sundaram	FOR EDUCATIONAL USE

Mapper class: WordMapper.java

```
package Words;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import java.io.IOException;
import java.util.StringTokenizer;
public class WordMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
    public void map(LongWritable key, Text value, Context con) throws
IOException, InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while(tokenizer.hasMoreTokens())
            con.write(new Text(tokenizer.nextToken()), new IntWritable(1));
    }
}
```

Reducer Class: WordReducer.java

```
package Words;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
import java.io.IOException;
public class WordReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{
    public void reduce(Text word, Iterable<IntWritable> values, Context con)
throws IOException, InterruptedException {
    int sum = 0;
    for(IntWritable value : values) {
        sum += value.get();
    }
    con.write(word, new IntWritable(sum));
}
```

Driver Class: WordDriver.java

```
package Words;
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.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordDriver {
    public static void main(String[] args) throws Exception {
        Job job = new Job(new Configuration(), "WordCount");
        job.setJarByClass(Words.WordDriver.class);
        job.setMapperClass(Words.WordMapper.class);
        job.setReducerClass(Words.WordReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[1]));
        FileOutputFormat.setOutputPath(job, new Path(args[2]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output Screenshot

```
Activities

    Terminal ▼

                                            May 18 17:13
                                                                    Q
                                     hduser@yatish-VirtualBox: ~
        Ħ
               File Input Format Counters
                       Bytes Read=565
               File Output Format Counters
                       Bytes Written=128
       hduser@yatish-VirtualBox:~$ hdfs dfs -cat /PVG/Output5
       21/05/18 17:12:50 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
       ary for your platform... using builtin-java classes where applicable
       ^Chduser@yatish-VirtualBox:~$ hdfs dfs -cat /PVG/Output5/*
       21/05/18 17:12:56 WARN util.NativeCodeLoader: Unable to load native-hadoop libr
       ary for your platform... using builtin-java classes where applicable
       am
       content.
                        б
       count
       example 6
       file
       given
               б
       going
       here
       is
       random
               б
               12
       so
       that
       the
               б
       type
               б
      word
       words
       hduser@yatish-VirtualBox:~$
```