EXPERIMENT – 4

Wordcount Analysis using MapReduce Programming

Aim: To perform wordcount operation using text file (unstructured)

Code:

Mapper.java

```
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 wordmapper extends MapReduceBase implements
Mapper<LongWritable,Text,Text,IntWritable>

            public void map(LongWritable key, Text value,
OutputCollector<Text,IntWritable> output, reporter r) throws IOException{

   String s = value.toString();

   for(String word:s.split(" "))

         {

            if(word.length()>0){

                  output.collect(new Text(word),new IntWritable(1));

         }

      }

}
```

Reducer.java

```
import java.io.IOException;

import java.util.Iterator;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapred.MapReduceBase;
```

```
import org.apache.hadoop.mapred.OutputCollector;

import org.apache.hadoop.mapred.Reducer;

import org.apache.hadoop.mapred.Reporter;


public class wordreducer extends MapReduceBase implements
Reducer<Text,IntWritable,Text,IntWritable>

   public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text,
IntWritable> ourput, reporter r) throws IOException{

   int count = 0;

   while(values.hasNext()){

         IntWritable i = values.next();

      count+=i.get();

   }

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

}
```

Main.java

```
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 wordcount extends configured implements Tools{

   @Override

         public int run(String[] args) throws Exception{

         if(args.length<2){

            System.out.println("Please give Input Output Directory correctly");
```

```
        return -1;
        }
      JobConf conf = new JobConf(wordcount.class);
      FileInputFormat.setInputPaths(conf,new Path(args[0]);
      FileOutputFormat.setOutputPath(conf, new Path(args[1]);
      conf.setMapperClass(wordmapper.class);
      conf.setReducerClass(wordreducer.class);
      conf.setMapOutputKeyClass(Text.class);
      conf.setMapOutputValueClass(IntWritable.class);
      conf.setOutputKeyClass(Text.class);
      conf.setOutputValueClass(IntWritable.class);
      JobClient.runJob(conf);
      return 0;
    }
   public static void main(String args[]) throws Exception{
            int exitcode = ToolRunner.run(new wordcount(), args);
            system.exit(exitcode);
}
```

Text File:

```
cloudera@quickstart:~/workspace
File  Edit  View  Search  Terminal  Help
  GNU nano 2.0.9              File: WCFile.txt                        Modified

HI
HI
Bottom up
Bottom Down
Down

^G Get Help   ^O WriteOut   ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit       ^J Justify    ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Output:

```
cloudera@quickstart:~/workspace
File  Edit  View  Search  Terminal  Help
23/03/10 02:01:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
23/03/10 02:01:38 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
23/03/10 02:01:42 INFO mapred.FileInputFormat: Total input paths to process : 1
23/03/10 02:01:42 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$DataStreamer.closeResponder(DFSOutputStre
am.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.jav
a:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894
)
23/03/10 02:01:43 INFO mapreduce.JobSubmitter: number of splits:2
23/03/10 02:01:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678438009122
_0001
23/03/10 02:01:53 INFO impl.YarnClientImpl: Submitted application application_1678438009122
_0001
23/03/10 02:01:54 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:
8088/proxy/application_1678438009122_0001/
```

File Edit View Search Terminal Help

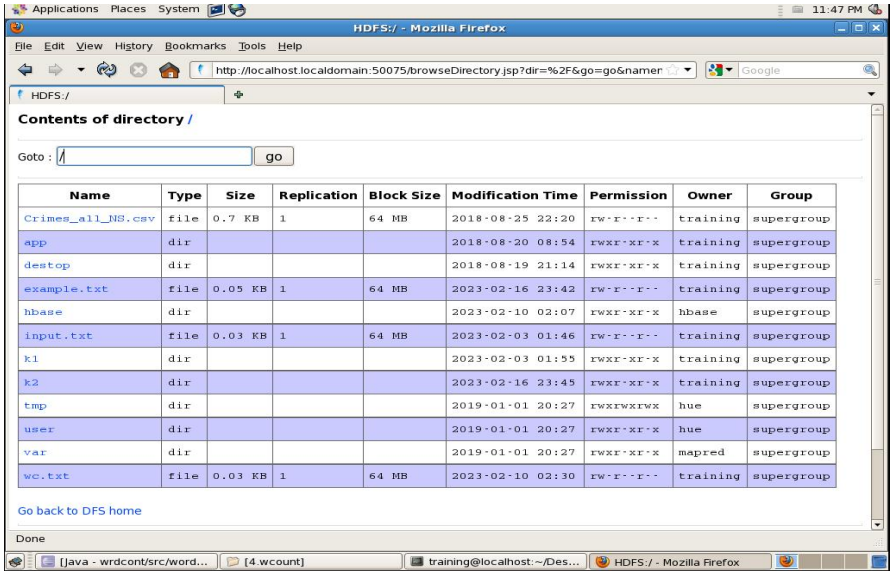
```
23/03/10 02:01:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678438009122
_0001
23/03/10 02:01:53 INFO impl.YarnClientImpl: Submitted application application_1678438009122
_0001
23/03/10 02:01:54 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:
8088/proxy/application_1678438009122_0001/
23/03/10 02:01:54 INFO mapreduce.Job: Running job: job_1678438009122_0001

23/03/10 02:04:01 INFO mapreduce.Job: Job job_1678438009122_0001 running in uber mode : fal
se
23/03/10 02:04:01 INFO mapreduce.Job:  map 0% reduce 0%

23/03/10 02:06:32 INFO mapreduce.Job:  map 100% reduce 0%


23/03/10 02:07:32 INFO mapreduce.Job:  map 100% reduce 100%
23/03/10 02:07:35 INFO mapreduce.Job: Job job_1678438009122_0001 completed successfully
23/03/10 02:07:36 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=81
                FILE: Number of bytes written=430787
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=266
                HDFS: Number of bytes written=26
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=298143
                Total time spent by all reduces in occupied slots (ms)=53993
                Total time spent by all map tasks (ms)=298143
                Total time spent by all reduce tasks (ms)=53993
                Total vcore-milliseconds taken by all map tasks=298143
                Total vcore-milliseconds taken by all reduce tasks=53993
```

cloudera@quickstart:... [How to Execute Word... [Java - WordCount/src/...

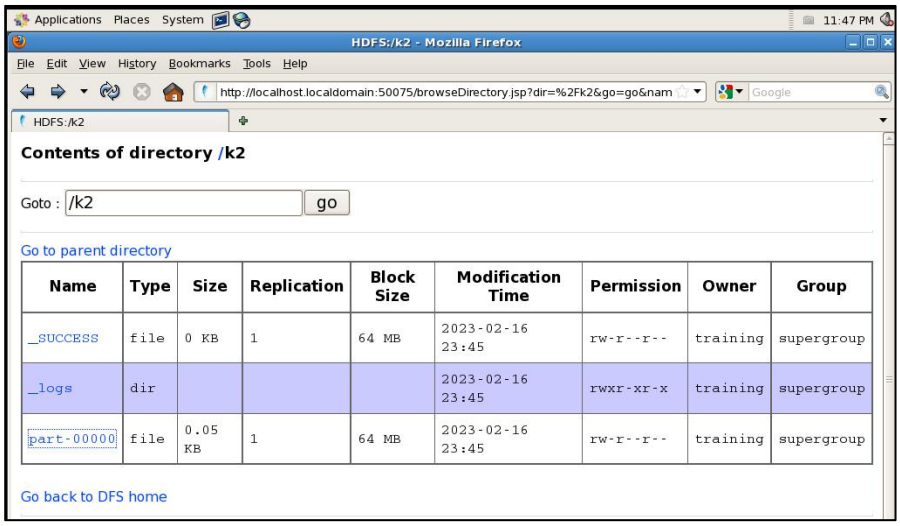
File Edit View Search Terminal Help

```
                Total megabyte-milliseconds taken by all reduce tasks=55288832
        Map-Reduce Framework
                Map input records=5
                Map output records=7
                Map output bytes=61
                Map output materialized bytes=87
                Input split bytes=216
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=87
                Reduce input records=7
                Reduce output records=4
                Spilled Records=14
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=2519
                CPU time spent (ms)=48790
                Physical memory (bytes) snapshot=729976832
                Virtual memory (bytes) snapshot=4712329216
                Total committed heap usage (bytes)=734003200
        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=50
        File Output Format Counters
                Bytes Written=26
0
[cloudera@quickstart workspace]$
[cloudera@quickstart workspace]$
[cloudera@quickstart workspace]$
[cloudera@quickstart workspace]$
[cloudera@quickstart workspace]$
```

cloudera@quickstart:... [How to Execute Word... [Java - WordCount/src/...



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
Crimes_all_NS.csv	file	0.7 KB	1	64 MB	2018-08-25 22:20	rw-r--r--	training	supergroup
app	dir				2018-08-20 08:54	rwxr-xr-x	training	supergroup
destop	dir				2018-08-19 21:14	rwxr-xr-x	training	supergroup
example.txt	file	0.05 KB	1	64 MB	2023-02-16 23:42	rw-r--r--	training	supergroup
hbase	dir				2023-02-10 02:07	rwxr-xr-x	hbase	supergroup
input.txt	file	0.03 KB	1	64 MB	2023-02-03 01:46	rw-r--r--	training	supergroup
k1	dir				2023-02-03 01:55	rwxr-xr-x	training	supergroup
k2	dir				2023-02-16 23:45	rwxr-xr-x	training	supergroup
tmp	dir				2019-01-01 20:27	rwxrwxrwx	hue	supergroup
user	dir				2019-01-01 20:27	rwxr-xr-x	hue	supergroup
var	dir				2019-01-01 20:27	rwxr-xr-x	mapred	supergroup
wc.txt	file	0.03 KB	1	64 MB	2023-02-10 02:30	rw-r--r--	training	supergroup



Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 KB	1	64 MB	2023-02-16 23:45	rw-r--r--	training	supergroup
_logs	dir				2023-02-16 23:45	rwxr-xr-x	training	supergroup
part-00000	file	0.05 KB	1	64 MB	2023-02-16 23:45	rw-r--r--	training	supergroup

```
[cloudera@quickstart workspace]$ hadoop fs -cat WCOutput/part-00000
Bottom  2
Down    2
HI      2
up      1
[cloudera@quickstart workspace]$
```