Program for user count

```
package LogFile1;

import java.io.IOException;

import java.util.StringTokenizer;


import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.LongWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.util.GenericOptionsParser;


public class UserCount {


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


        Configuration c=new Configuration();

        String[] files=new GenericOptionsParser(c,args).getRemainingArgs();

        Path input=new Path(files[0]);

        Path output=new Path(files[1]);

        Job j=new Job(c,"wordcount");

        j.setJarByClass(UserCount.class);

        j.setMapperClass(MapForUserCount.class);

        j.setReducerClass(ReduceForUserCount.class);
```

```
        j.setOutputKeyClass(Text.class);
        j.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(j, input);
        FileOutputFormat.setOutputPath(j, output);
        System.exit(j.waitForCompletion(true)?0:1);
}


public static class MapForUserCount extends Mapper<LongWritable, Text, Text, IntWritable>{

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

        public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException {
                StringTokenizer itr = new StringTokenizer(value.toString(),",");

                while(itr.hasMoreTokens()){
                        // skipping first 2 entries
                        itr.nextToken();
                        itr.nextToken();

                        // set the ip in word variable
                        word.set(itr.nextToken());

                        // tokenize via '.'
                        StringTokenizer itr2 = new StringTokenizer(word.toString(),".");

                        // validate by checking 1.0.1.0 characters as 4
```

```
                int count = 0;
                while(itr2.hasMoreTokens()){
                        count++;
                        itr2.nextToken();
                }
                if(count==4){
                        con.write(word,cntOne);
                }
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();



                }
        }
}

public static class ReduceForUserCount extends Reducer<Text, IntWritable, Text, IntWritable>{


   private IntWritable result = new IntWritable();


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


                int sum = 0;
                for(IntWritable val : values){
                        sum += val.get();
```

```
            }

            result.set(sum);

            con.write(key, result);

        }

}


}
```

Screenshots :

```
[cloudera@quickstart Assn-2]$ hadoop fs -put LogFile.csv LogFile
put: `LogFile': File exists
[cloudera@quickstart Assn-2]$ hadoop fs -ls
Found 2 items
-rw-r--r--   1 cloudera cloudera   65103303 2022-04-28 22:03 LogFile
drwxr-xr-x   - cloudera cloudera          0 2022-04-28 22:05 LogFileDir
[cloudera@quickstart Assn-2]$ hadoop jar LogFile.jar LogFile1.UserMaxFreq LogFil
e LogFileDir1
22/04/28 22:09:27 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
22/04/28 22:09:28 INFO input.FileInputFormat: Total input paths to process : 1
22/04/28 22:09:28 INFO mapreduce.JobSubmitter: number of splits:1
22/04/28 22:09:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
51200547708_0002
22/04/28 22:09:29 INFO impl.YarnClientImpl: Submitted application application_16
51200547708_0002
22/04/28 22:09:29 INFO mapreduce.Job: The url to track the job: http://quickstar
t.cloudera:8088/proxy/application_1651200547708_0002/
22/04/28 22:09:29 INFO mapreduce.Job: Running job: job_1651200547708_0002
22/04/28 22:09:38 INFO mapreduce.Job: Job job_1651200547708_0002 running in uber
 mode : false
22/04/28 22:09:38 INFO mapreduce.Job:  map 0% reduce 0%
22/04/28 22:09:50 INFO mapreduce.Job:  map 100% reduce 0%
22/04/28 22:10:00 INFO mapreduce.Job:  map 100% reduce 100%
22/04/28 22:10:01 INFO mapreduce.Job: Job job 1651200547708 0002 completed successfully
```

```
                FILE: Number of bytes read=12301975
                FILE: Number of bytes written=24824639
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=65103421
                HDFS: Number of bytes written=18
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=9492
                Total time spent by all reduces in occupied slots (ms)=7997
                Total time spent by all map tasks (ms)=9492
                Total time spent by all reduce tasks (ms)=7997
                Total vcore-seconds taken by all map tasks=9492
                Total vcore-seconds taken by all reduce tasks=7997
                Total megabyte-seconds taken by all map tasks=9719808
                Total megabyte-seconds taken by all reduce tasks=8188928
        Map-Reduce Framework
                Map input records=663257
                Map output records=663257
                Map output bytes=10975455
                Map output materialized bytes=12301975
                Input split bytes=118
                Combine input records=0
                Combine output records=0
                Reduce input groups=991
                Reduce shuffle bytes=12301975

                Bytes Written=18
[cloudera@quickstart Assn-2]$ hadoop fs -ls
Found 3 items
-rw-r--r--   1 cloudera cloudera   65103303 2022-04-28 22:03 LogFile
drwxr-xr-x   - cloudera cloudera          0 2022-04-28 22:05 LogFileDir
drwxr-xr-x   - cloudera cloudera          0 2022-04-28 22:09 LogFileDir1
[cloudera@quickstart Assn-2]$ hadoop fs -ls LogFileDir1
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2022-04-28 22:09 LogFileDir1/_SUCCESS
-rw-r--r--   1 cloudera cloudera         18 2022-04-28 22:09 LogFileDir1/part-r-00000
[cloudera@quickstart Assn-2]$ hadoop fs -cat LogFileDir1/part-r-00000
10.10.15.99     24467
[cloudera@quickstart Assn-2]$
```

```
10.10.15.49     120
10.10.15.51     90
10.10.15.53     356
10.10.15.54     38
10.10.15.57     27
10.10.15.58     631
10.10.15.6      140
10.10.15.62     2
10.10.15.66     681
10.10.15.69     4445
10.10.15.70     214
10.10.15.72     118
10.10.15.73     31
10.10.15.74     820
10.10.15.75     67
10.10.15.79     300
10.10.15.8      29
10.10.15.80     207
10.10.15.81     74
10.10.15.82     1120
10.10.15.83     378
10.10.15.84     65
10.10.15.85     921
10.10.15.86     430
10.10.15.90     287
10.10.15.92     6
10.10.15.93     8
10.10.15.94     332
10.10.15.96     315
10.10.15.98     2770
10.10.15.99     231
[cloudera@quickstart Assn-2]$
```

Program for max user frequency

```
package LogFile1;

import java.io.IOException;

import java.util.StringTokenizer;


import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;
```

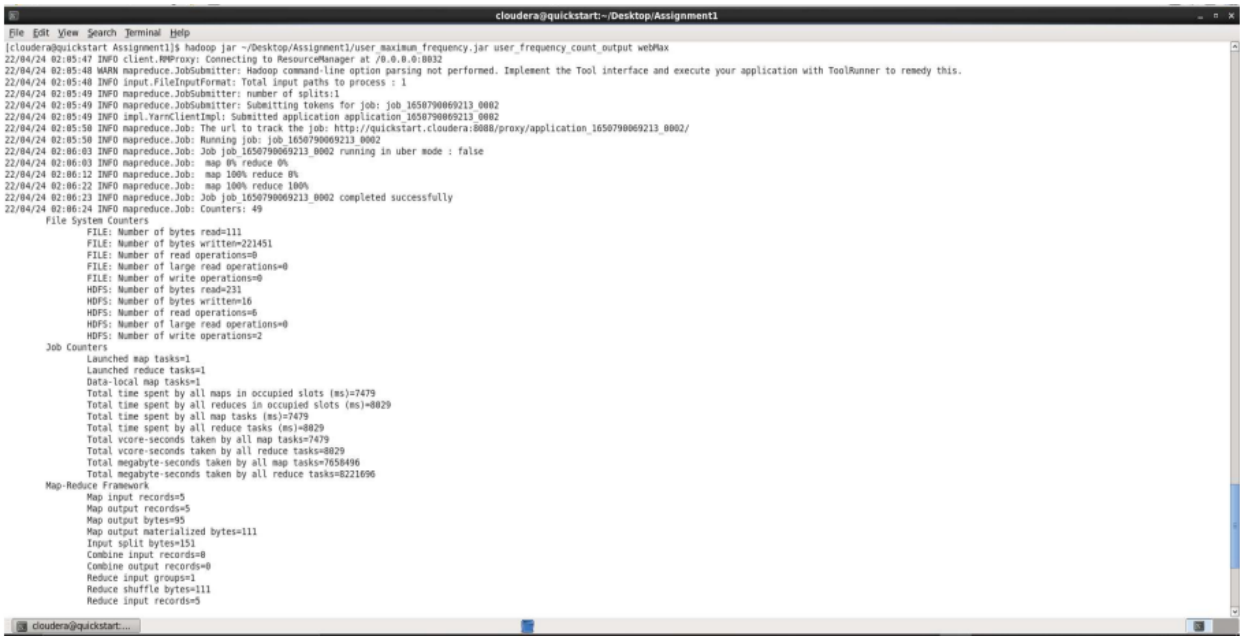
```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class UserMaxFreq {

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

        Configuration c=new Configuration();
        String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
        Path input=new Path(files[0]);
        Path output=new Path(files[1]);
        Job j=new Job(c,"wordcount");
        j.setJarByClass(UserCount.class);
        j.setMapperClass(MapForUserMaxFreq.class);
        j.setReducerClass(ReduceForUserMaxFreq.class);
        j.setOutputKeyClass(Text.class);
        j.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(j, input);
        FileOutputFormat.setOutputPath(j, output);
        System.exit(j.waitForCompletion(true)?0:1);
}
```

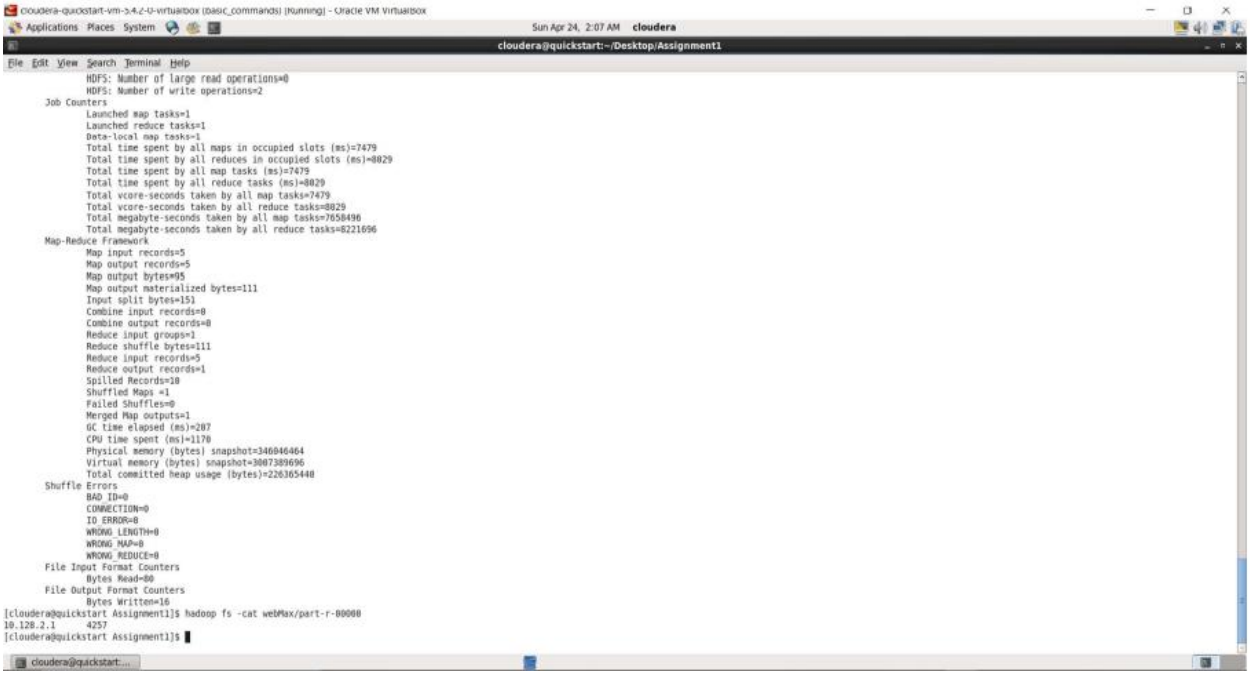
```
public static class MapForUserMaxFreq extends Mapper<LongWritable, Text, Text, IntWritable>{

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

        public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException {

                StringTokenizer itr = new StringTokenizer(value.toString(),",");

                while(itr.hasMoreTokens()){

                        // skipping first 2 entries
                        itr.nextToken();
                        itr.nextToken();

                        // set the ip in word variable
                        word.set(itr.nextToken());

                        // tokenize via '.'
                        StringTokenizer itr2 = new StringTokenizer(word.toString(),".");

                        // validate by checking 1.0.1.0 characters as 4
                int count = 0;
                while(itr2.hasMoreTokens()){
                        count++;
                        itr2.nextToken();
                }
```

```
                if(count==4){
                        con.write(word,cntOne);
                }
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();
                itr.nextToken();



                }
        }
}

public static class ReduceForUserMaxFreq extends Reducer<Text, IntWritable, Text, IntWritable>{

   private Text maxKey = new Text("");
   private int maxFreq = 0;

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

                int sum = 0;

                for(IntWritable val : values){
                        sum += val.get();
                }

            if(sum > maxFreq) {
```

```
                maxFreq = sum;
                maxKey = key;
            }
        }

    @Override
    public void cleanup(Context context) throws IOException, InterruptedException {
            // write the word with the highest frequency
       context.write(maxKey, new IntWritable(maxFreq));
    }
}

}
```

Screenshots :




cloudera-quickstart-vm-5.4.2-0-virtualbox (basic_commands) [Running] - Oracle VM VirtualBox
Applications Places System
Sun Apr 24, 2:07 AM cloudera
cloudera@quickstart:~/Desktop/Assignment1
File Edit View Search Terminal Help
HDFS: Number of large read operations=0
HDFS: Number of write operations=2
Job Counters
Launched map tasks=1
Launched reduce tasks=1
Data-local map tasks=1
Total time spent by all maps in occupied slots (ms)=7479
Total time spent by all reduces in occupied slots (ms)=8029
Total time spent by all map tasks (ms)=7479
Total time spent by all reduce tasks (ms)=8029
Total vcore-seconds taken by all map tasks=7479
Total vcore-seconds taken by all reduce tasks=8029
Total megabyte-seconds taken by all map tasks=7658496
Total megabyte-seconds taken by all reduce tasks=8221696
Map-Reduce Framework
Map input records=5
Map output records=5
Map output bytes=95
Map output materialized bytes=111
Input split bytes=151
Combine input records=0
Combine output records=0
Reduce input groups=1
Reduce shuffle bytes=111
Reduce input records=5
Reduce output records=1
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=207
CPU time spent (ms)=1170
Physical memory (bytes) snapshot=346046464
Virtual memory (bytes) snapshot=3007389696
Total committed heap usage (bytes)=226365440
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=80
File Output Format Counters
Bytes Written=16
[cloudera@quickstart Assignment1]$ hadoop fs -cat webMax/part-r-00000
10.128.2.1 4257
[cloudera@quickstart Assignment1]$
cloudera@quickstart:...