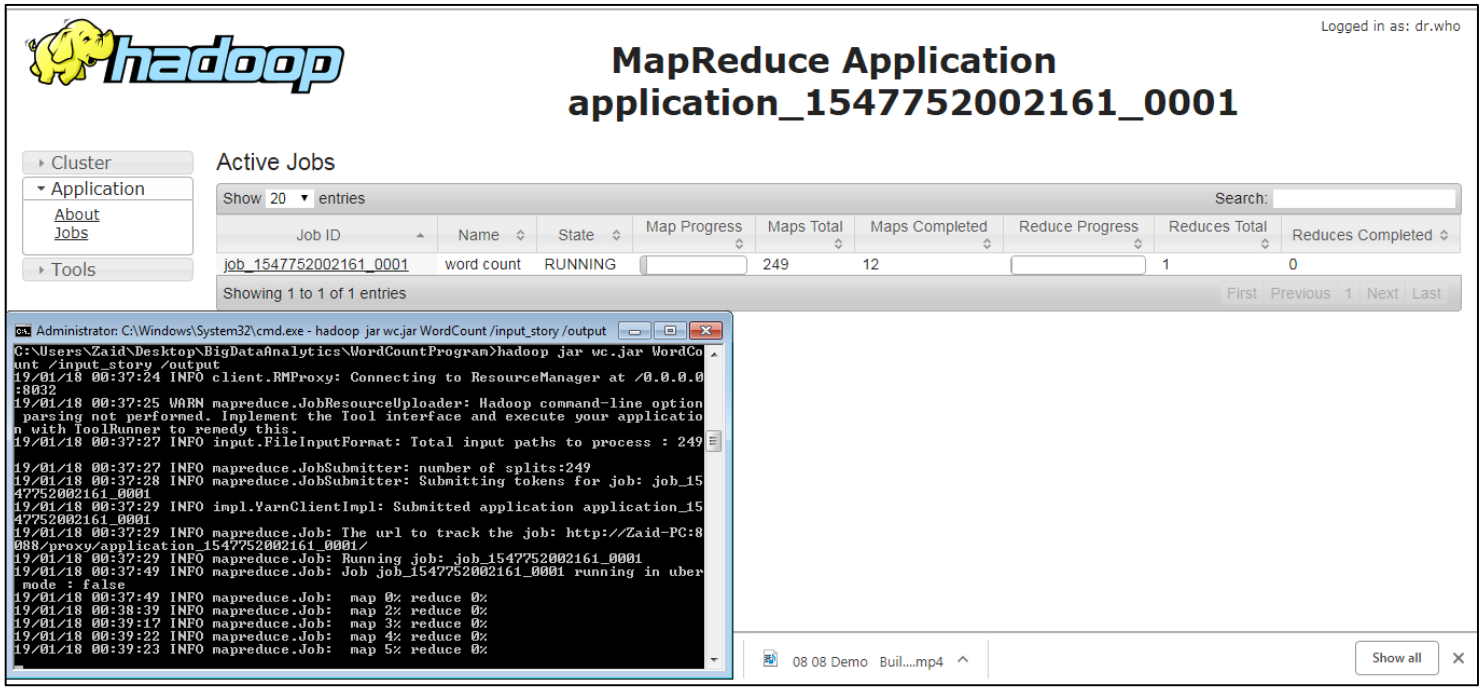
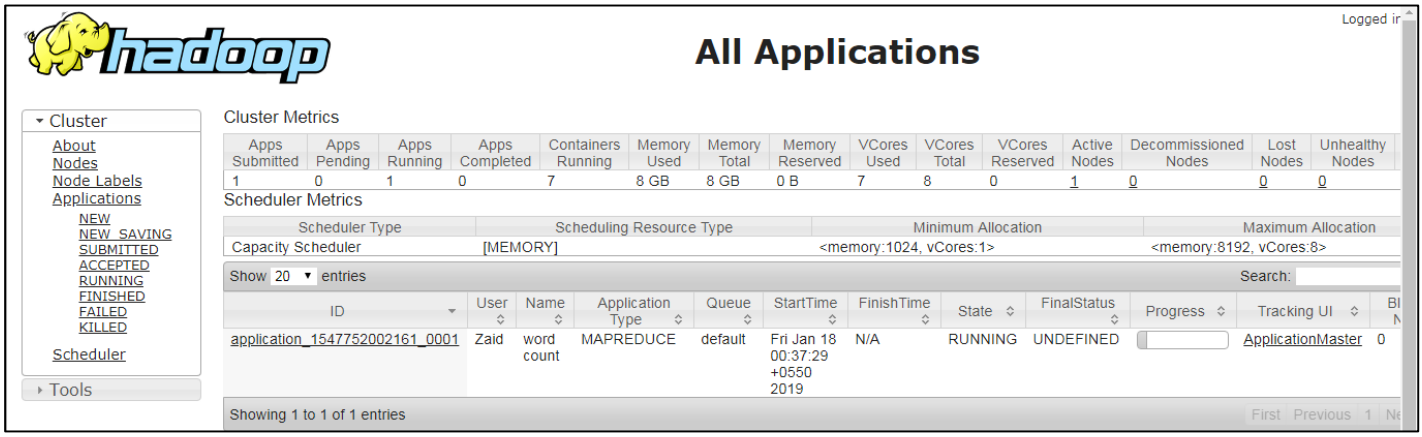
Program for Word Count

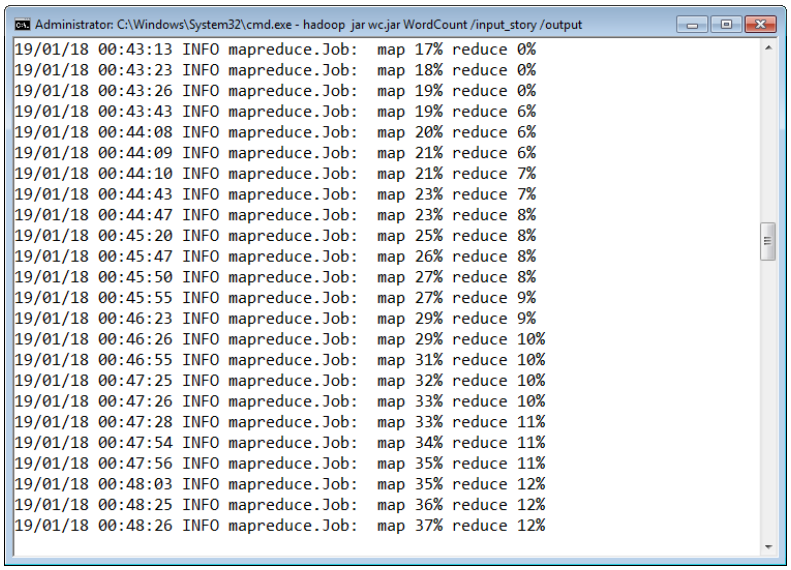
```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.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;

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 {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
      word.set(itr.nextToken());
      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);
  }
 }
```

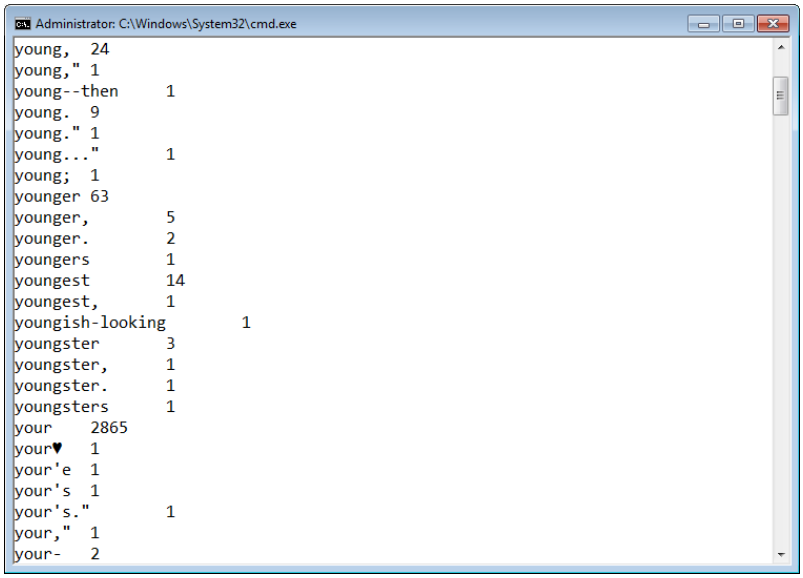
```
 public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "word count");
  job.setJar("wc.jar");
  job.setJarByClass(WordCount.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  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);
 }
}
```

OUTPUT




Logged in
All Applications
Cluster
About
Nodes
Node Labels
Applications
NEW
NEW_SAVING
SUBMITTED
ACCEPTED
RUNNING
FINISHED
FAILED
KILLED
Scheduler
Tools
Cluster Metrics
Apps Submitted 1
Apps Pending 0
Apps Running 1
Apps Completed 0
Containers Running 7
Memory Used 8 GB
Memory Total 8 GB
Memory Reserved 0 B
VCores Used 7
VCores Total 8
VCores Reserved 0
Active Nodes 1
Decommissioned Nodes 0
Lost Nodes 0
Unhealthy Nodes 0
Scheduler Metrics
Scheduler Type: Capacity Scheduler
Scheduling Resource Type: [MEMORY]
Minimum Allocation: <memory:1024, vCores:1>
Maximum Allocation: <memory:8192, vCores:8>
Show 20 entries
Search:
ID: application_1547752002161_0001
User: Zaid
Name: word count
Application Type: MAPREDUCE
Queue: default
StartTime: Fri Jan 18 00:37:29 +0550 2019
FinishTime: N/A
State: RUNNING
FinalStatus: UNDEFINED
Progress
Tracking UI: ApplicationMaster
0
Showing 1 to 1 of 1 entries
First Previous 1


Administrator: C:\Windows\System32\cmd.exe - hadoop jar wc.jar WordCount /input_story /output
19/01/18 00:43:13 INFO mapreduce.Job: map 17% reduce 0%
19/01/18 00:43:23 INFO mapreduce.Job: map 18% reduce 0%
19/01/18 00:43:26 INFO mapreduce.Job: map 19% reduce 0%
19/01/18 00:43:43 INFO mapreduce.Job: map 19% reduce 6%
19/01/18 00:44:08 INFO mapreduce.Job: map 20% reduce 6%
19/01/18 00:44:09 INFO mapreduce.Job: map 21% reduce 6%
19/01/18 00:44:10 INFO mapreduce.Job: map 21% reduce 7%
19/01/18 00:44:43 INFO mapreduce.Job: map 23% reduce 7%
19/01/18 00:44:47 INFO mapreduce.Job: map 23% reduce 8%
19/01/18 00:45:20 INFO mapreduce.Job: map 25% reduce 8%
19/01/18 00:45:47 INFO mapreduce.Job: map 26% reduce 8%
19/01/18 00:45:50 INFO mapreduce.Job: map 27% reduce 8%
19/01/18 00:45:55 INFO mapreduce.Job: map 27% reduce 9%
19/01/18 00:46:23 INFO mapreduce.Job: map 29% reduce 9%
19/01/18 00:46:26 INFO mapreduce.Job: map 29% reduce 10%
19/01/18 00:46:55 INFO mapreduce.Job: map 31% reduce 10%
19/01/18 00:47:25 INFO mapreduce.Job: map 32% reduce 10%
19/01/18 00:47:26 INFO mapreduce.Job: map 33% reduce 10%
19/01/18 00:47:28 INFO mapreduce.Job: map 33% reduce 11%
19/01/18 00:47:54 INFO mapreduce.Job: map 34% reduce 11%
19/01/18 00:47:56 INFO mapreduce.Job: map 35% reduce 11%
19/01/18 00:48:03 INFO mapreduce.Job: map 35% reduce 12%
19/01/18 00:48:25 INFO mapreduce.Job: map 36% reduce 12%
19/01/18 00:48:26 INFO mapreduce.Job: map 37% reduce 12%


Administrator: C:\Windows\System32\cmd.exe
young, 24
young," 1
young--then 1
young. 9
young." 1
young..." 1
young; 1
younger 63
younger, 5
younger. 2
youngers 1
youngest 14
youngest, 1
youngish-looking 1
youngster 3
youngster, 1
youngster. 1
youngsters 1
your 2865
your♥ 1
your'e 1
your's 1
your's." 1
your," 1
your- 2

File System Counters
- FILE: Number of bytes read=4948059
- FILE: Number of bytes written=40763189
- FILE: Number of read operations=0
- FILE: Number of large read operations=0
- FILE: Number of write operations=0
- HDFS: Number of bytes read=8347191
- HDFS: Number of bytes written=1162025
- HDFS: Number of read operations=750
- HDFS: Number of large read operations=0
- HDFS: Number of write operations=2

Job Counters
- Killed map tasks=2
- Launched map tasks=250
- Launched reduce tasks=1
- Data-local map tasks=250
- Total time spent by all maps in occupied slots (ms)=7175456
- Total time spent by all reduces in occupied slots (ms)=1183371
- Total time spent by all map tasks (ms)=7175456
- Total time spent by all reduce tasks (ms)=1183371

Map-Reduce Framework
- Map input records=160578
- Map output records=1395206
- Map output bytes=13319863
- Map output materialized bytes=4949547
- Input split bytes=27420
- Combine input records=1395206
- Combine output records=368133
- Reduce input groups=105959
- Reduce shuffle bytes=4949547
- Reduce input records=368133
- Reduce output records=105959
- Spilled Records=736266
- Shuffled Maps =249
- Failed Shuffles=0
- Merged Map outputs=249
- GC time elapsed (ms)=16977
- CPU time spent (ms)=144310
- Physical memory (bytes) snapshot=60321419264
- Virtual memory (bytes) snapshot=64892907520
- Total committed heap usage (bytes)=45005668352

Shuffle Errors
- BAD_ID=0
- CONNECTION=0
- IO_ERROR=0
- WRONG_LENGTH=0
- WRONG_MAP=0
- WRONG_REDUCE=0

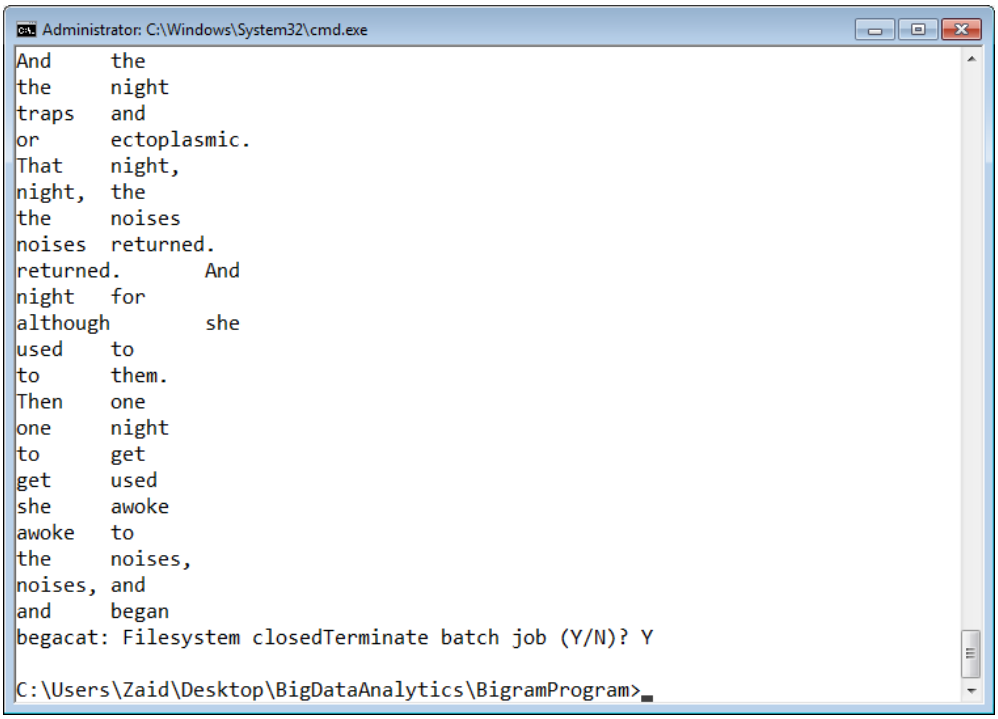
File Input Format Counters
- Bytes Read=8319771

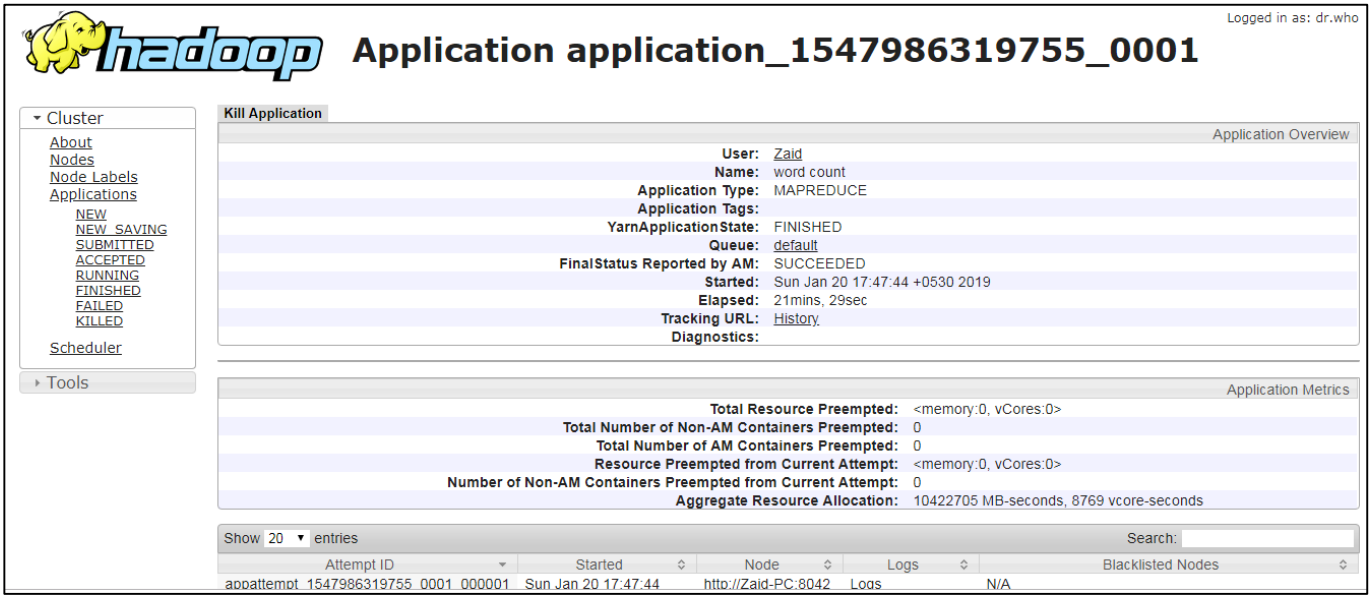
File Output Format Counters
- Bytes Written=1162025

Program for Bigram/Multigram

```
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.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;

public class Multigram {

  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 {
      StringTokenizer itr = new StringTokenizer(value.toString());
      String curr = new String();
      String ngram = new String();
      int ng = 2;
      int n = ng-1;
      String prev_Tokens[] = new String[n];
      Boolean notNull = true;

      while (itr.hasMoreTokens()) {
        curr = itr.nextToken();
        for(int i = 0;i < n;++i){
          if( prev_Tokens[i] == null && prev_Tokens[i] == ""){
            notNull = false;
            break;
          }
        }
        if( notNull){
          for(int i = 0; i < n; i++){
            ngram += " "+prev_Tokens[i];
          }
        }
        ngram += " "+curr;
        word.set(ngram);
        context.write(word, one);
        ngram = "";
        for(int i = 0; i < n-1; ++i){
          prev_Tokens[i] = prev_Tokens[i+1];
```

```
      }
      prev_Tokens[n-1] = curr;
      notNull = true;
    }
   }
  }

  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();
    }
    if(sum > 1){
     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(Ngram.class);
   job.setMapperClass(TokenizerMapper.class);
   job.setCombinerClass(IntSumReducer.class);
   job.setReducerClass(IntSumReducer.class);
   job.setOutputKeyClass(Text.class);
   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);
  }
 }
```





Commands for Hadoop

start-all.cmd

hadoop fs -mkdir /input_dir

hadoop fs -put C:/input_file.txt /input_dir

```
hadoop fs -ls /input_dir/*

hadoop dfs -cat /input_dir/input_file.txt

hadoop dfsadmin -safemode leave

hadoop fs -rm -r /input_dir/input_file.txt

hadoop fs -rm -r /input_dir

hadoop com.sun.tools.javac.Main WordCount.java

jar cf wc.jar WordCount*.class

hadoop com.sun.tools.javac.Main WordCount.java

jar cf wc.jar WordCount*.class

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