Indian Institute Of Technology, Delhi

# COL733: Cloud Computing Technology Fundamentals

*Instructor: S. C. Gupta*

---

**Assignment 4: Map Reduce**

---

***Report***

October 25, 2019


*Submitted To:*
S. C. Gupta
Professor
Computer Science Department

*Submitted By: (Group 4)*

| | |
|---|---|
| Shantanu Verma | 2016CS10373 |
| Pradyumna Meena | 2016CS10375 |
| Manav Rao | 2016CS10523 |
| Shubham | 2016CS10371 |

# Index


| S.No. | Topic | Page Number |
|---|---|---|
| 1. | Installation of Map Reduce framework | 2 |
| | 1.1. Preview | |
| | 1.2. Commands | |
| 2. | Word count for large text collection | 3 |
| | 2.1. Word count of 100 mb file | |
| | 2.2. Word count of 26 mb file | |
| 3. | Average grades for class records | 5 |

# 1. Installation of Map Reduce framework

## 1.1. Preview

Map Reduce framework was installed when we install Hadoop on the virtual machines. To test the framework and working we calculated the value of pi using quasi-Monte Carlo method

## 1.2. Commands

```
$ su - hadoop #(command to log in as hadoop user)
$ yarn jar/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.3.jar
π 16 1000 #(running map reduce 16 maps with 1000 samples per map)
```

**Example:** Calculation of π value and testing of test already available jar of MapReduce/

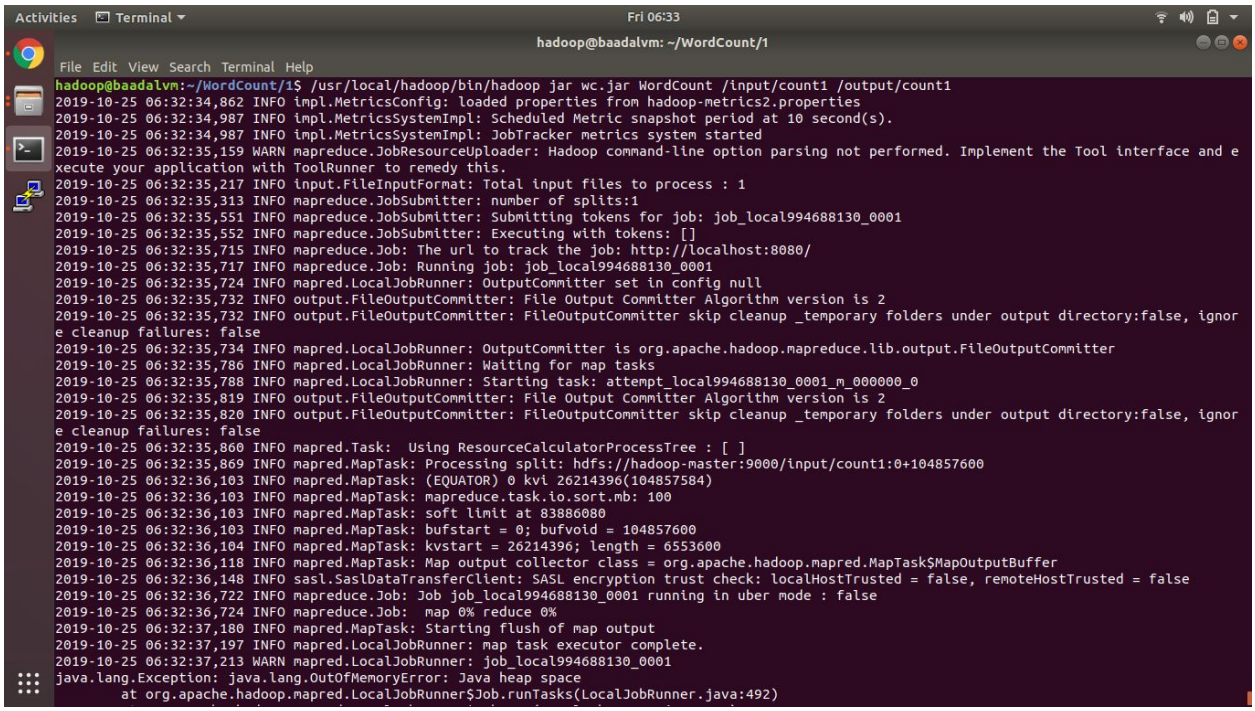
```
2019-10-24 23:07:14,719 INFO mapreduce.Job: Job job_local170161807_0001 completed successfully
2019-10-24 23:07:14,771 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=5676115
                FILE: Number of bytes written=14011834
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=17936
                HDFS: Number of bytes written=32311
                HDFS: Number of read operations=529
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=309
        Map-Reduce Framework
                Map input records=16
                Map output records=32
                Map output bytes=288
                Map output materialized bytes=448
                Input split bytes=2422
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=448
                Reduce input records=32
                Reduce output records=0
                Spilled Records=64
                Shuffled Maps =16
                Failed Shuffles=0
                Merged Map outputs=16
                GC time elapsed (ms)=55
                Total committed heap usage (bytes)=4018143232
        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=1888
        File Output Format Counters
                Bytes Written=97
Job Finished in 2.989 seconds
Estimated value of Pi is 3.14250000000000000000
```

# 2. Word Count for large text collections

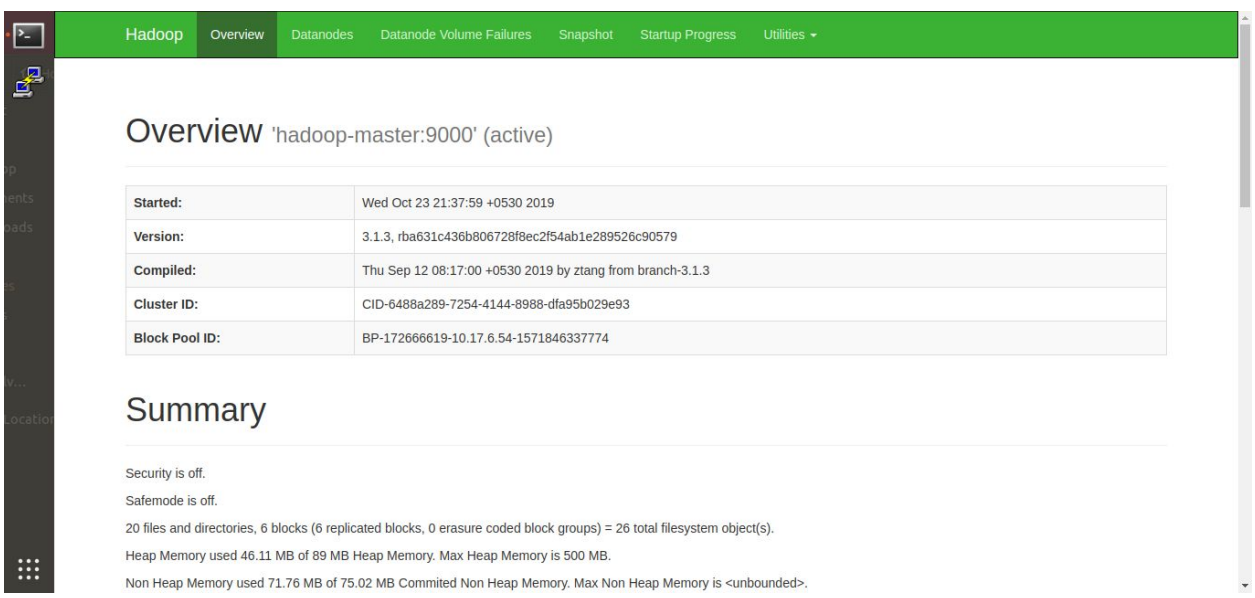
Dataset was generated by cleaning a collection of movie reviews provided by various users consisting of over a hundred thousand data points using python scripts. However the complete dataset resulted in heap space error. Hence dataset was reduced to a quarter of its original size.

## 2.1. Word count for 100 mb file

**Screenshot: 100 mb file used memory heap error**



```
hadoop@baadalvm:~/WordCount/1$ /usr/local/hadoop/bin/hadoop jar wc.jar WordCount /input/count1 /output/count1
2019-10-25 06:32:34,862 INFO impl.MetricsConfig: loaded properties from hadoop-metrics2.properties
2019-10-25 06:32:34,987 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2019-10-25 06:32:34,987 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2019-10-25 06:32:35,159 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and e
xecute your application with ToolRunner to remedy this.
2019-10-25 06:32:35,217 INFO input.FileInputFormat: Total input files to process : 1
2019-10-25 06:32:35,313 INFO mapreduce.JobSubmitter: number of splits:1
2019-10-25 06:32:35,551 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local994688130_0001
2019-10-25 06:32:35,552 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-10-25 06:32:35,715 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2019-10-25 06:32:35,717 INFO mapreduce.Job: Running job: job_local994688130_0001
2019-10-25 06:32:35,724 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2019-10-25 06:32:35,732 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-10-25 06:32:35,732 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignor
e cleanup failures: false
2019-10-25 06:32:35,734 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2019-10-25 06:32:35,786 INFO mapred.LocalJobRunner: Waiting for map tasks
2019-10-25 06:32:35,788 INFO mapred.LocalJobRunner: Starting task: attempt_local994688130_0001_m_000000_0
2019-10-25 06:32:35,819 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2019-10-25 06:32:35,820 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:false, ignor
e cleanup failures: false
2019-10-25 06:32:35,860 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2019-10-25 06:32:35,869 INFO mapred.MapTask: Processing split: hdfs://hadoop-master:9000/input/count1:0+104857600
2019-10-25 06:32:36,103 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2019-10-25 06:32:36,103 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
2019-10-25 06:32:36,103 INFO mapred.MapTask: soft limit at 83886080
2019-10-25 06:32:36,103 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2019-10-25 06:32:36,104 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2019-10-25 06:32:36,118 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
2019-10-25 06:32:36,148 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2019-10-25 06:32:36,722 INFO mapreduce.Job: Job job_local994688130_0001 running in uber mode : false
2019-10-25 06:32:36,724 INFO mapreduce.Job:  map 0% reduce 0%
2019-10-25 06:32:37,180 INFO mapred.MapTask: Starting flush of map output
2019-10-25 06:32:37,197 INFO mapred.LocalJobRunner: map task executor complete.
2019-10-25 06:32:37,213 WARN mapred.LocalJobRunner: job_local994688130_0001
java.lang.Exception: java.lang.OutOfMemoryError: Java heap space
        at org.apache.hadoop.mapred.LocalJobRunner$Job.runTasks(LocalJobRunner.java:492)
```

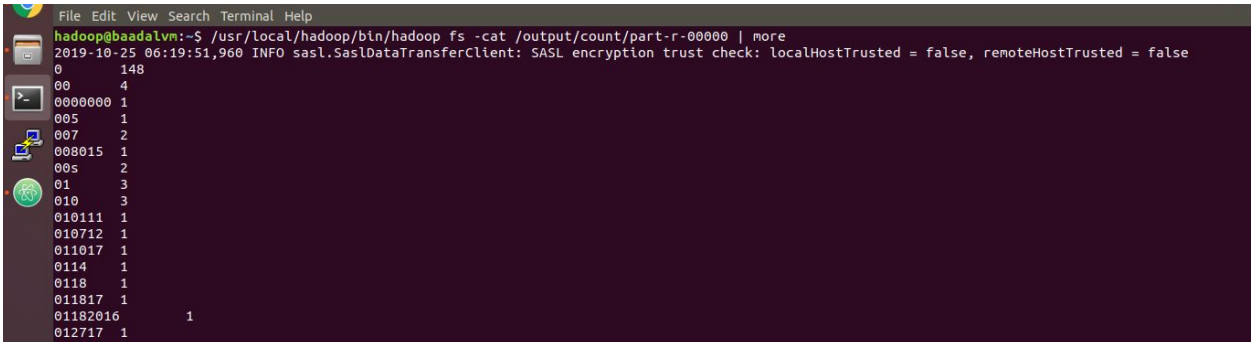


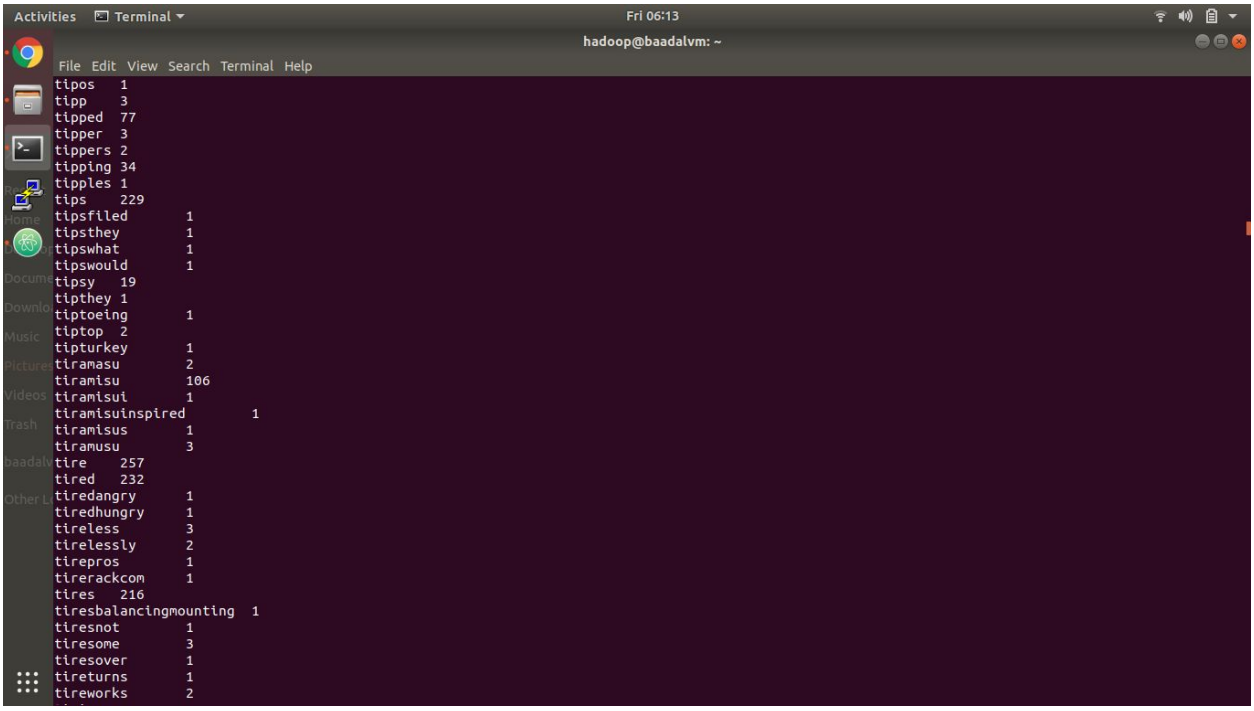

Code was written in WordCount.java and converted into a jar file and following is the list of commands required to run the code.

```
$ su -h hadoop
$ HADOOP_HOME/bin/hadoop fs -copyFromLocal /input/count/data.txt /input/count
(copying file to hadoop filesystem)
$ HADOOP_HOME/bin/hadoop com.sun.tools.javac.Main WordCount.java (compiling the
java code file)
$ jar cf wc.jar WordCount*.class (making the jar file to be run on hadoop file system)
$ HADOOP_HOME/bin/hadoop jar wc.jar WordCount /input/count /output/count (generates
the count file in the /output/count/part-r-00000 file)
$ HADOOP_HOME/bin/hadoop fs -cat /output/count/part-r-00000 (retrieves the data from the
generated file to the terminal)
```

Note:- Replacing WordCount with WordCount1 gives additional features of exempting specified strings(like ‘“‘ or ‘@’ or ‘.’ which are mentioned in different text file) from being counted.

## 2.2. Word Count of 26 mb file



# 3. Average Grades for class records

Dataset was generated containing 10,000 students and 5 courses with grades ranging from 5 to 10. Dataset is stored in grades.text file. Following is the list of commands required to run the code (on the namenode only)

$ su -h hadoop
$ HADOOP_HOME/bin/hadoop fs -copyFromLocal /input/avg/grades.txt /input/avg (copying file to hadoop filesystem)
$ HADOOP_HOME/bin/hadoop com.sun.tools.javac.Main Average.java (compiling **the** java code file)
$ jar cf wc.jar Average*.class (making **the** jar file to be run **on hadoop file system**)
$ HADOOP_HOME/bin/hadoop jar wc.jar Average /input/avg /output/avg (generates **the** count file **in the** /output/avg/part-r-00000 file)
$ HADOOP_HOME/bin/hadoop fs -cat /output/avg/part-r-00000 (retrieves **the** data from **the** generated file to **the** terminal)

For grades in range uniformly distributed in range [5,10]

```
hadoop@baadalvm:~$ /usr/local/hadoop/bin/hadoop fs -cat /output/avg/part-r-000002019-10-25 06:12:20,989 INFO sasl.SaslDataTransferClient: SASL e
ncryption trust check: localHostTrusted = false, remoteHostTrusted = false
COL106  Total: 12381.0 :: Average: 7.471937
COL202  Total: 12569.0 :: Average: 7.5308566
COL333  Total: 12662.0 :: Average: 7.527943
COL351  Total: 12510.0 :: Average: 7.4642005
COL352  Total: 12577.0 :: Average: 7.464095
COL733  Total: 12199.0 :: Average: 7.4794602
hadoop@baadalvm:~$
```

For grades in range uniformly distributed in range [1,10]

```
hadoop@baadalvm:~/WordCount/1$ /usr/local/hadoop/bin/hadoop fs -cat /output/avg1/part-r-00000
2019-10-25 06:37:36,386 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
COL106  Total: 8775.0 :: Average: 5.4
COL202  Total: 9446.0 :: Average: 5.656287
COL333  Total: 9373.0 :: Average: 5.5991635
COL351  Total: 9359.0 :: Average: 5.450786
COL352  Total: 8950.0 :: Average: 5.3052754
COL733  Total: 8994.0 :: Average: 5.5279655
hadoop@baadalvm:~/WordCount/1$
```

# 4. References


4.1. https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

4.2. https://medium.com/@diogo.fg.pinheiro/how-to-setup-hadoop-3-1-1-multi-node-cluster-on-ubuntu-18-04-2234986e2089