# Task: Exploration of Multi Node Hadoop Cluster and DFS

Sections:

* Running a sample Map Reduce job – wordcount (Page 3)
* Checking Fault Tolerance (Page 5)
* Programming Map Reduce job (Page 6)
  1. Testing on local Linux Machine
  2. Execution on Hadoop cluster

# Running a sample Map Reduce job – wordcount

1. Create a directory for keeping files, the files must be of HDFS file system.

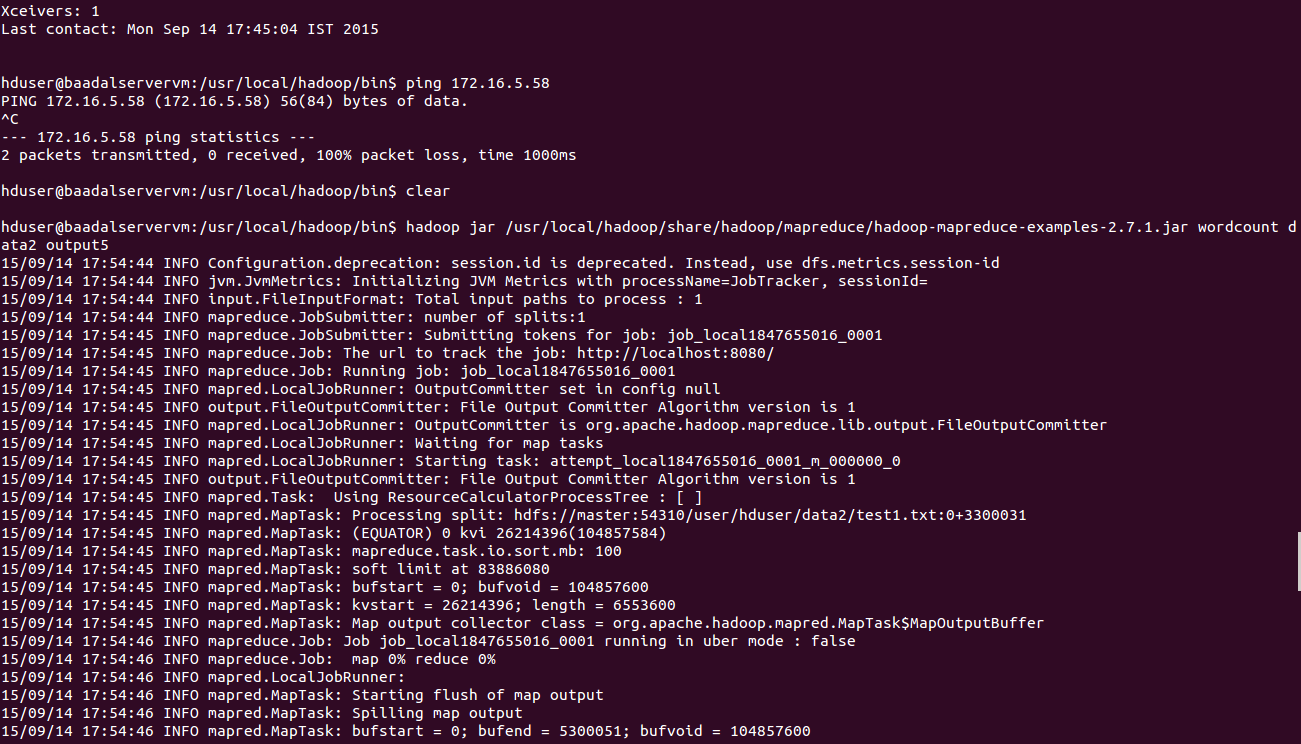
**$ ./hdfs dfs -mkdir /user/hduser**

1. Creating a sample .txt file and copying it to user/hduser. We created a 3MB file (consisting of ebook content).

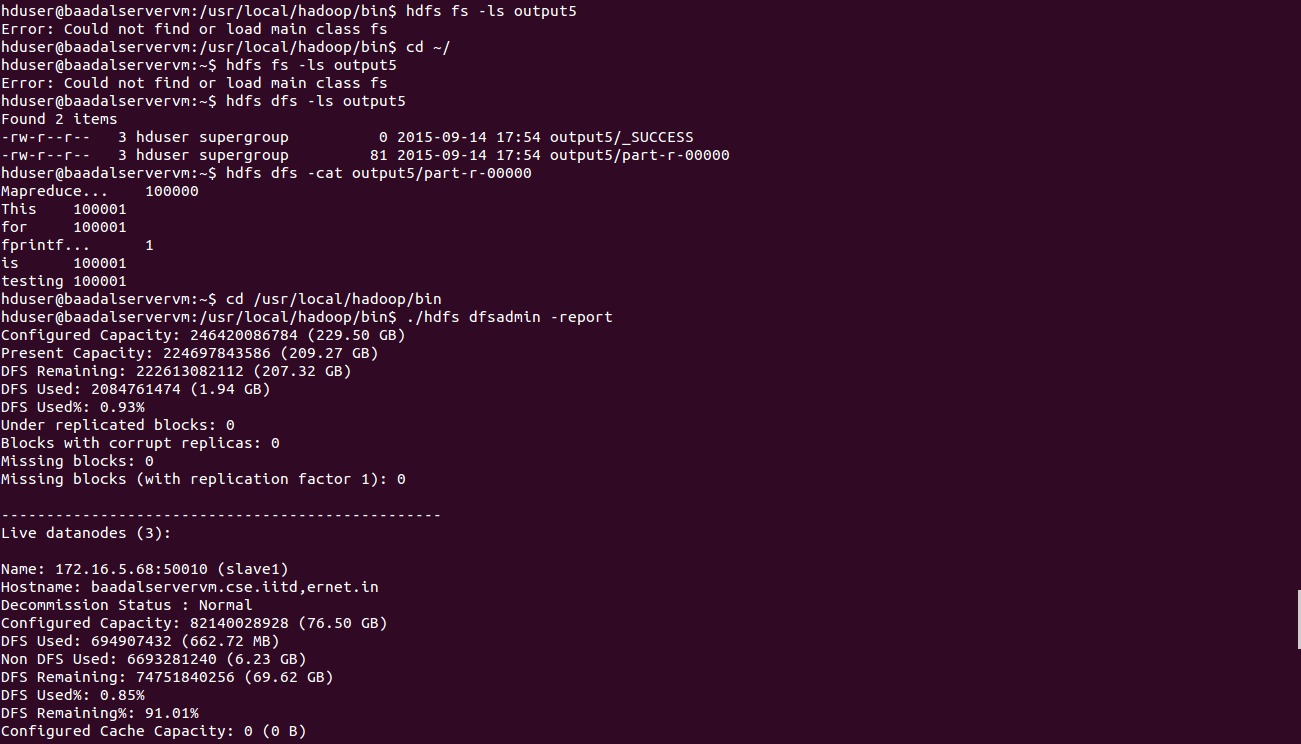
**$ ./hdfs dfs –copyFromLocal filename /user/hduser**

1. Now run the hadoop command to perform the wordcount job on the given files present in the directory created above, using the java jar example file present in the map reduce location.

**$ /usr/local/hadoop$ bin/hadoop jar hadoop-mapreduce-example-2.7.1.jar wordcount data2 output5**



The output can be seen.



# Checking Fault Tolerance:

After setting up the multi-node cluster, we created a 12MB file (consisting of ebook content). The replication factor maintained is 3. After starting the wordcount job, we shut down one of the slave nodes. We found that the mapper was repeatedly run and finally, the missing blocks reported. We checked the detailed report and found that some blocks were under replicated. We re-formatted the whole system and executed the whole procedure again successfully. The output was successfully obtained as below.

# Programming Map Reduce job:

1. Generate a file with rollno, course\_code and grade using c++ Code and write the output in rollFile.

***#include<iostream>***

***#include<cstdlib>***

***#include<fstream>***

***using namespace std;***

***int main(){***

***ofstream fout("rollFile");***

***for(int i=1;i<10000;i++)***

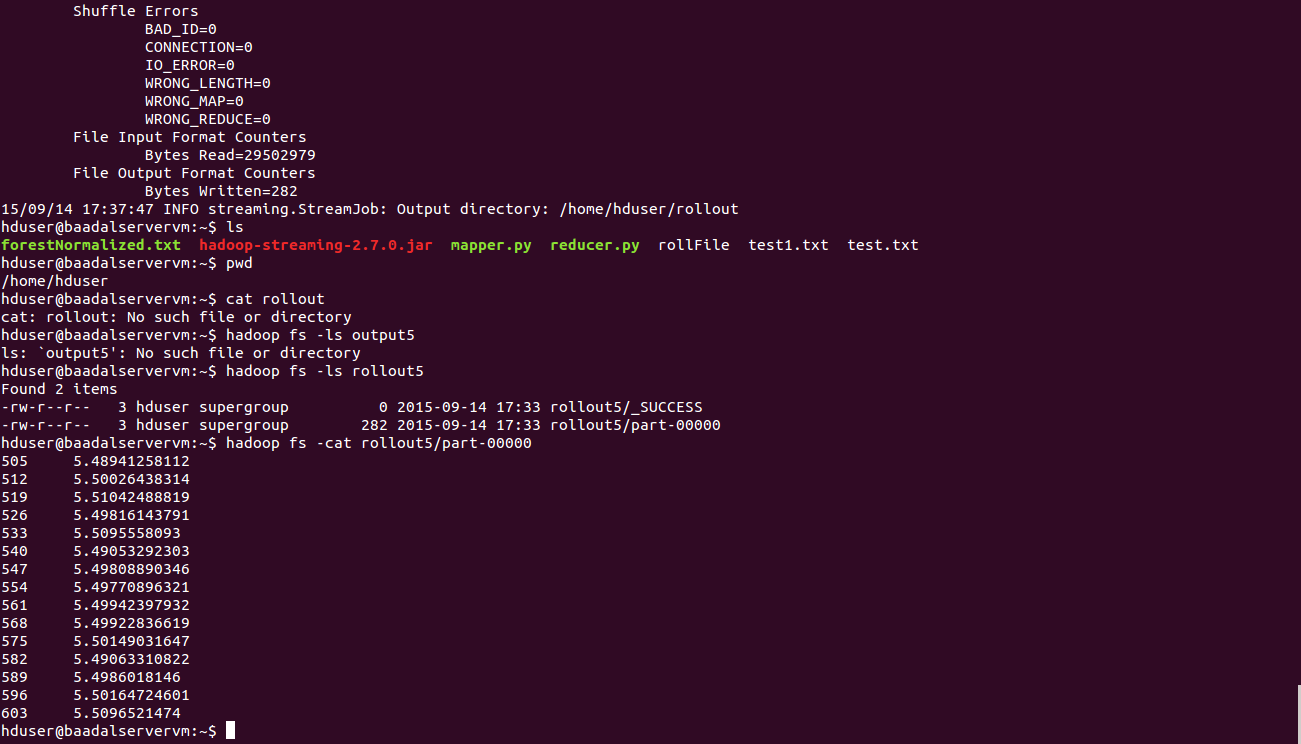
***fout<<rand()%100 + 12<<"\t"<< (rand()%15)\*7 + 505 << "\t"<<rand()%10 + 1<<"\n";***

***}***

1. Upload rollfile on HDFS file System.

$ **hadoop dfs -copyfromLocal rollFile /usrroot**

As rollfile can be seen after doing ls



1. *Mapper Code-* mapper.py

import sys

for line in sys.stdin:

data = line.strip().split("\t")

if len(data) == 3:

rollno,coursecode,grade = data

print "{0}\t{1}".format(coursecode, grade)

1. Reducer Code - reducer.py

|  |
| --- |
| **import sys gradesTotal = 0 oldKey = None**  **noOfGrades=0**    **for line in sys.stdin:**  **data\_mapped = line.strip().split("\t") if len(data\_mapped) != 2:**  **# Something has gone wrong. Skip this line.** |
| **continue**    **thisKey, thisGrade = data\_mapped**    **if oldKey and oldKey != thisKey:**  **print oldKey, "\t", gradesTotal/noOfGrades oldKey = thisKey; gradesTotal = 0 noOfGrades = 0**    **oldKey = thisKey**  **gradesTotal += float(thisGrade) noOfGrades = noOfGrades + 1 if oldKey != None:**  **print oldKey, "\t", gradesTotal/noOfGrades** |

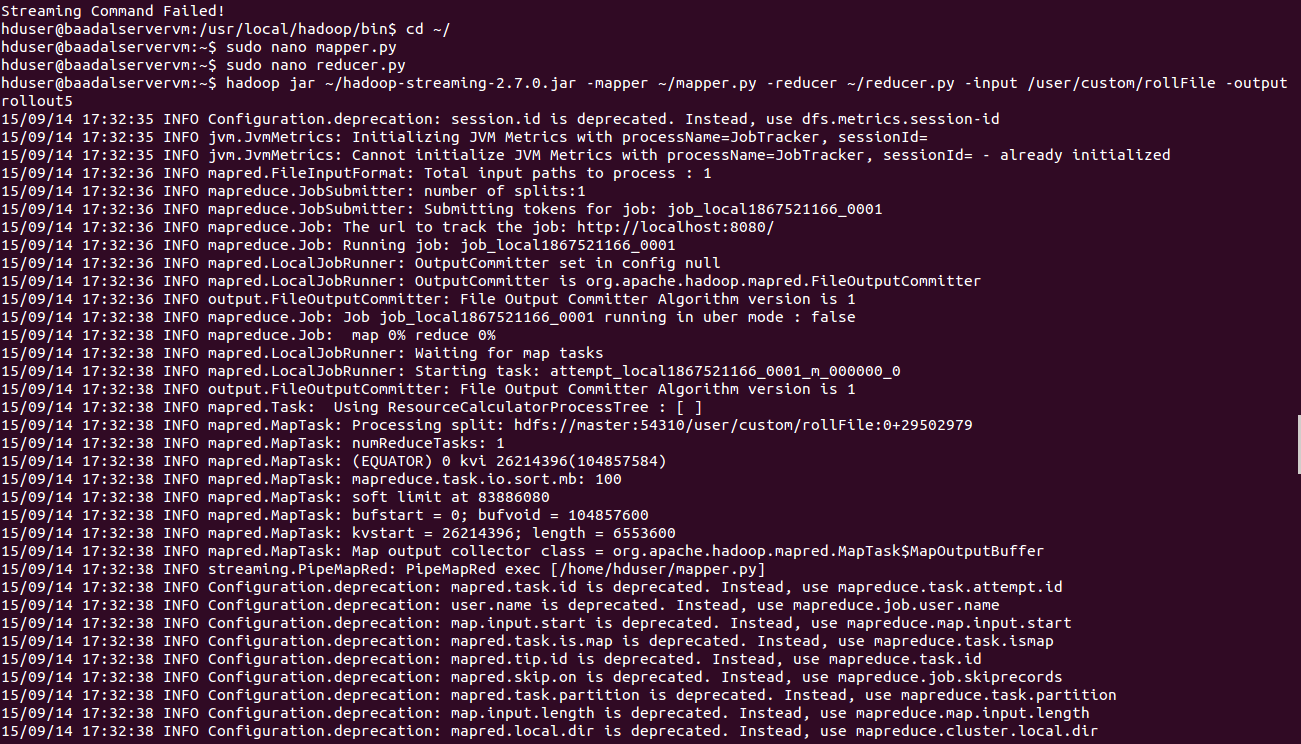
1. Testing mapper and reducer on local linux machine.

$ **cat rollFile | mapper.py | sort | reducer.py**

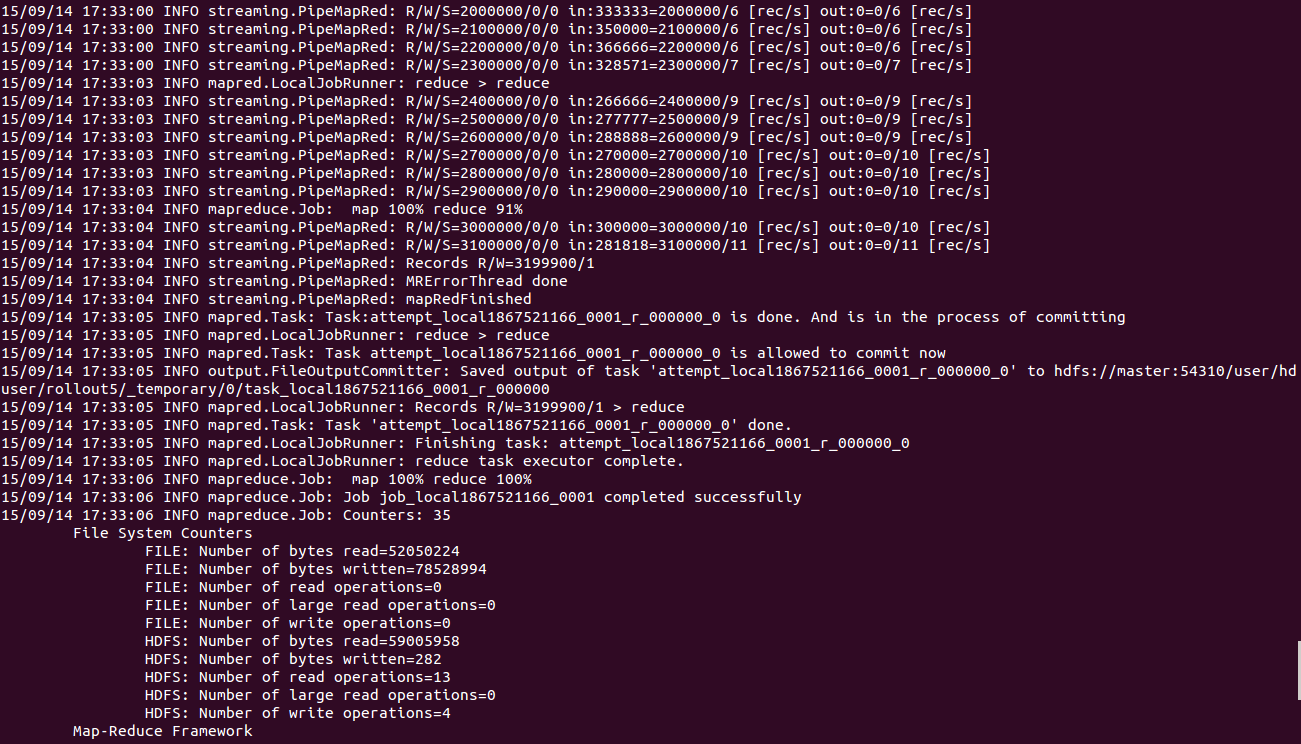
1. Running Mapper and Reducer on Hadoop Cluster

We used hadoop streaming utility to run python code on our hadoop cluster to run map reduce job.

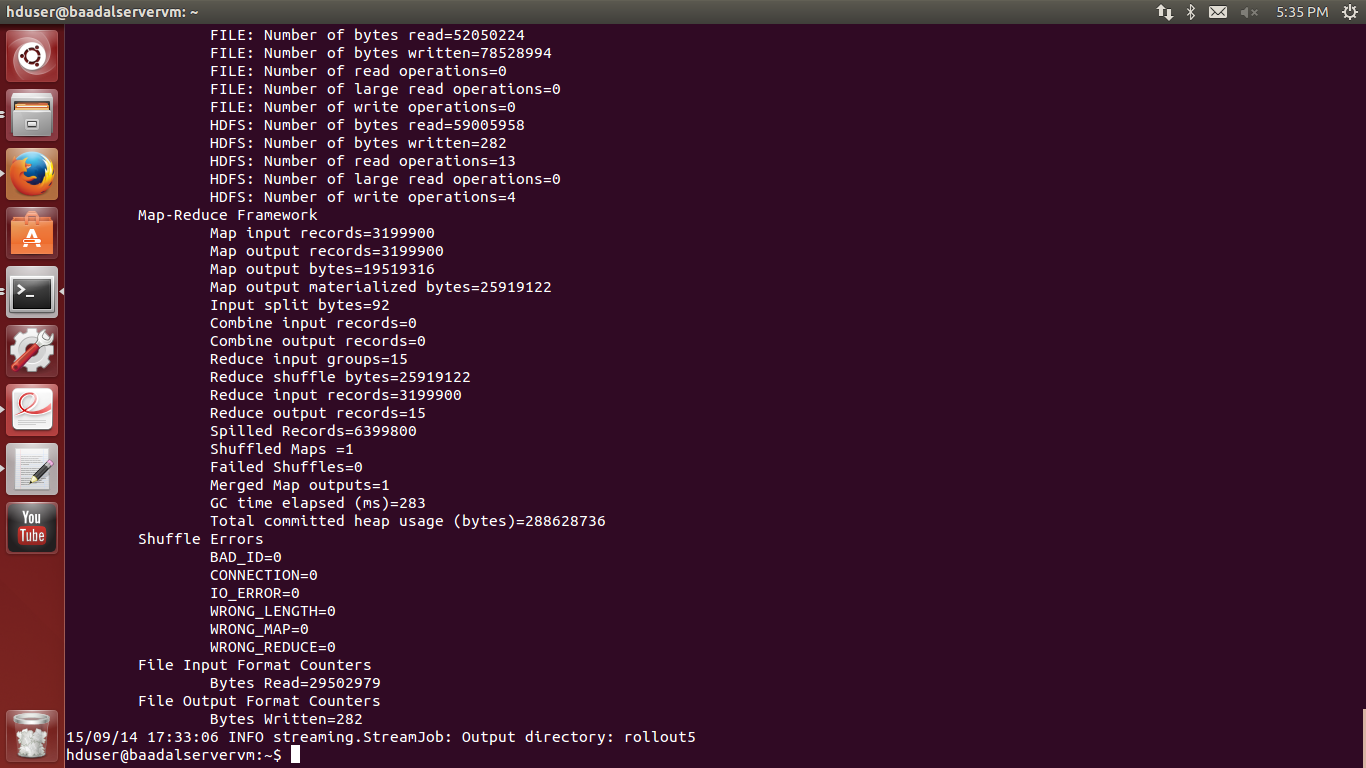
**$ hadoop jar hadoop-streaming-2.7.0.jar -mapper mapper.py -reducer reducer.py -input /usrroot -output rollout5**

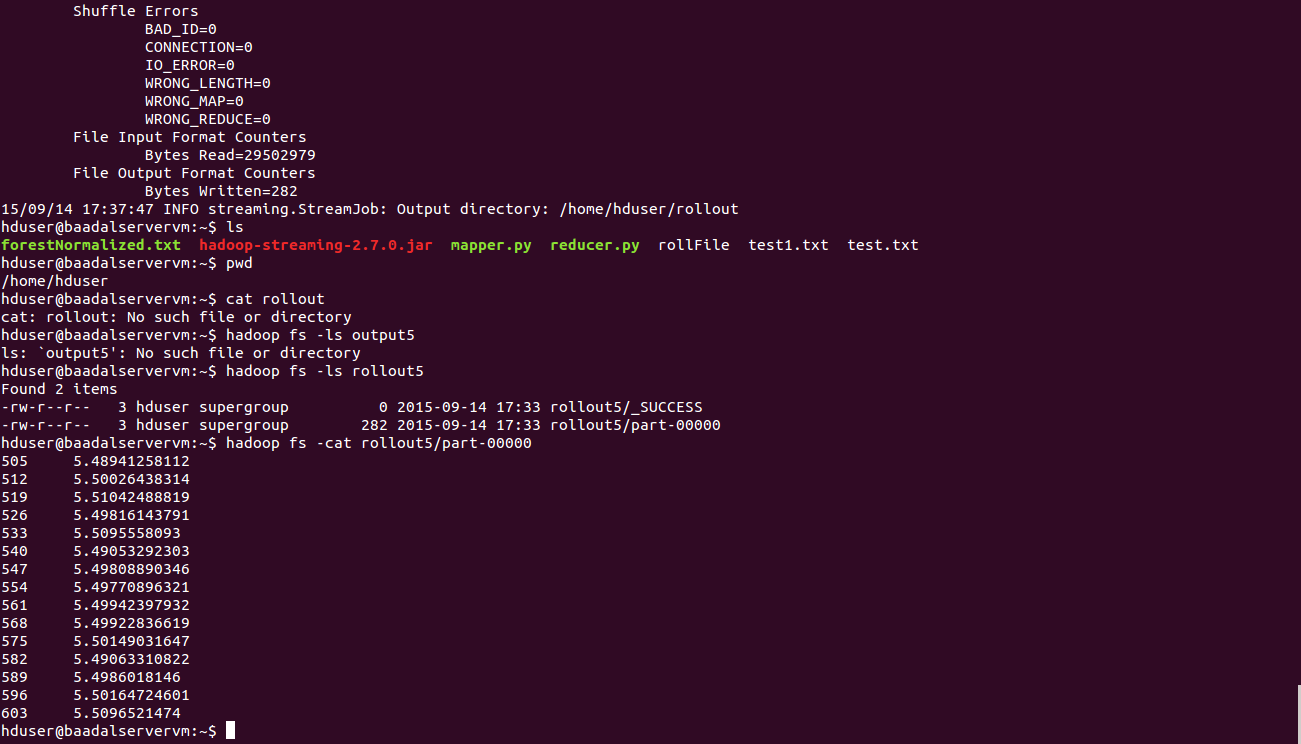


We have shown the mapRedFinished as seen in the screenshot.



We have shown File output Format counters



We have shown the output: average grade of the course

Main References:

* + <http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/>
  + <https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-on-ubuntu-13-10>
  + <http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/>