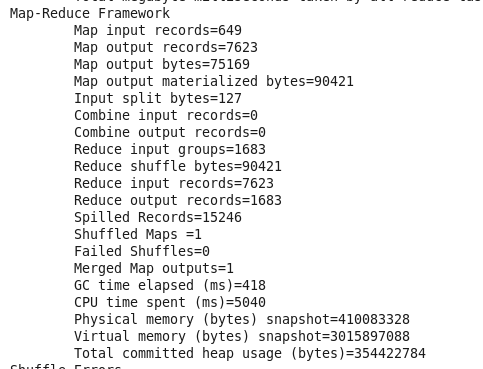
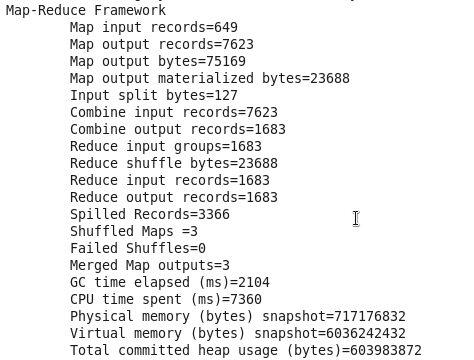
Without Combiner

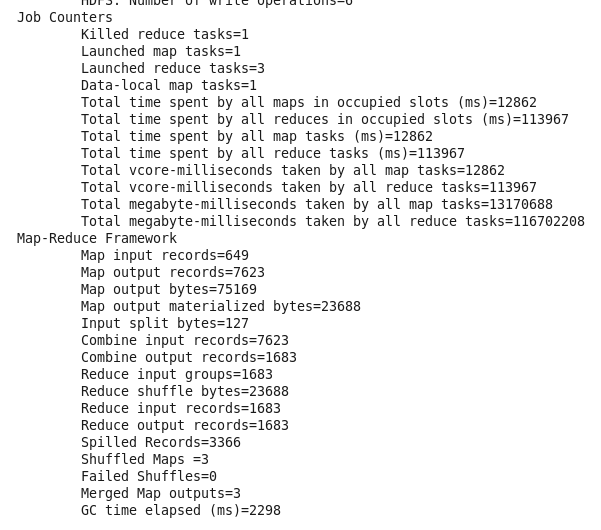


With combiner



.

Using Partitioner



5] Implementing Counter

1) Download an attachment in **STAGING\_AREA** location

2) Extract it

3) Pull it to **LABS\_AREA**/counter

4) Study the code

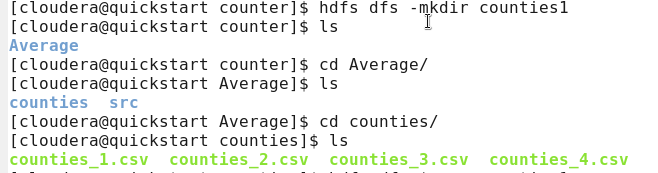
5) Create **counties** folder in your home on HDFS

6) Put all **counties\_\*.csv** files from local linux file system to **counties** folder on HDFS

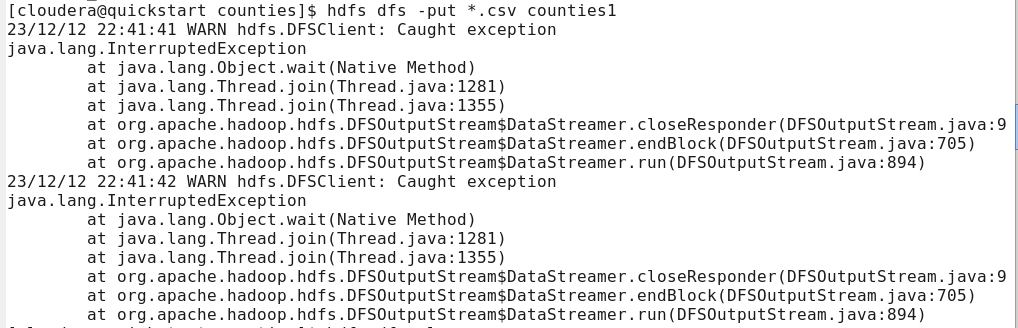
7) Run the application with command as **yarn jar average.jar <Main Class>**

8) See the output of the job on terminal window and confirm whether counters you created are working or not

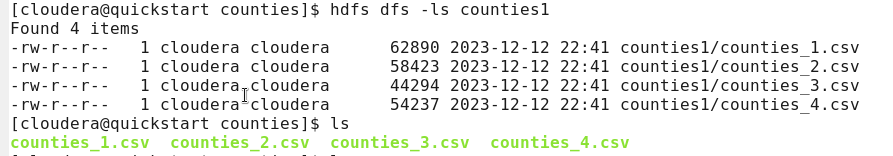
Code/Dataset:- data/Average.rar

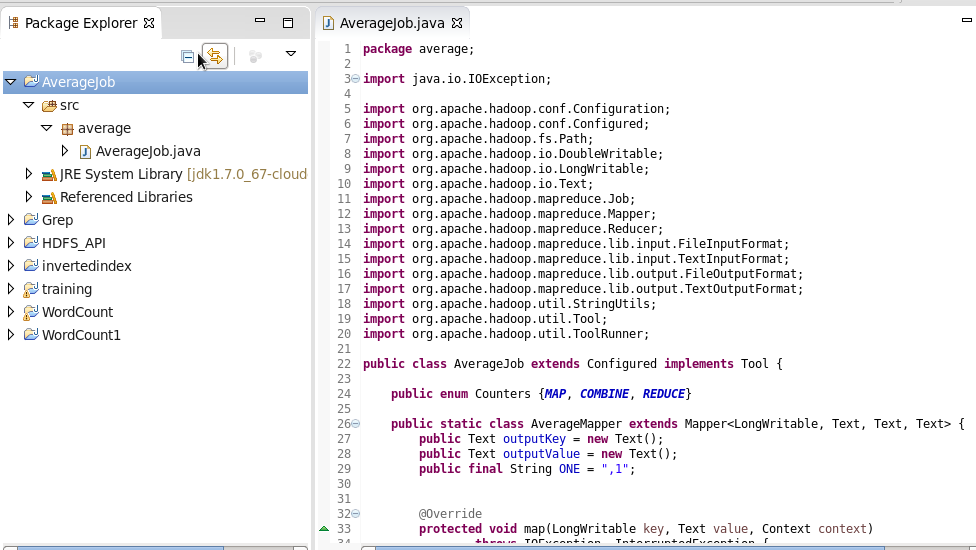


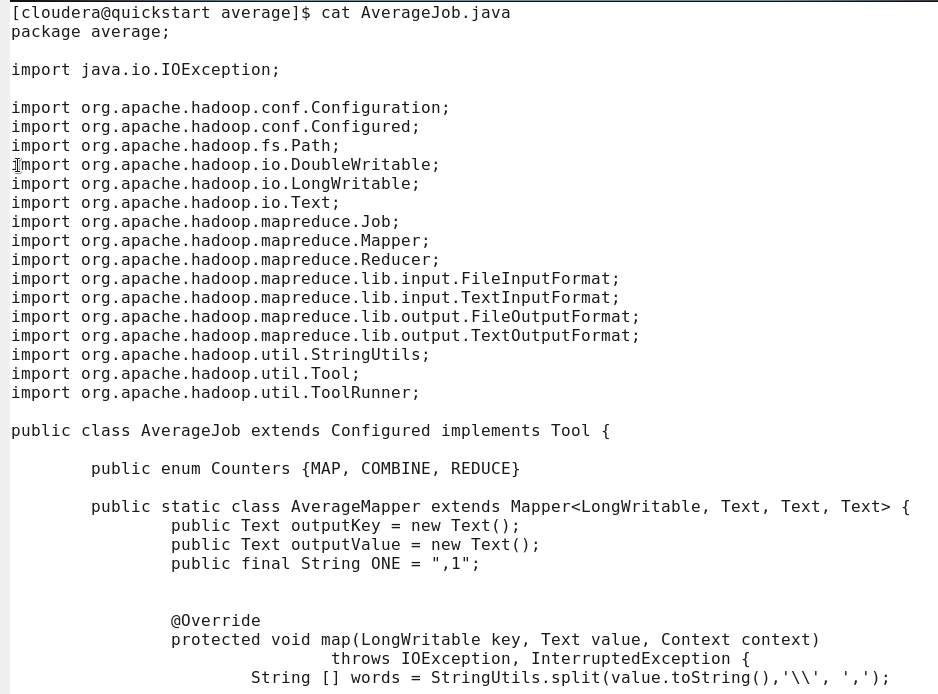
…



…

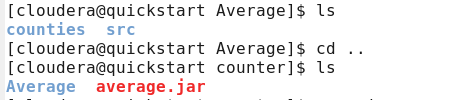




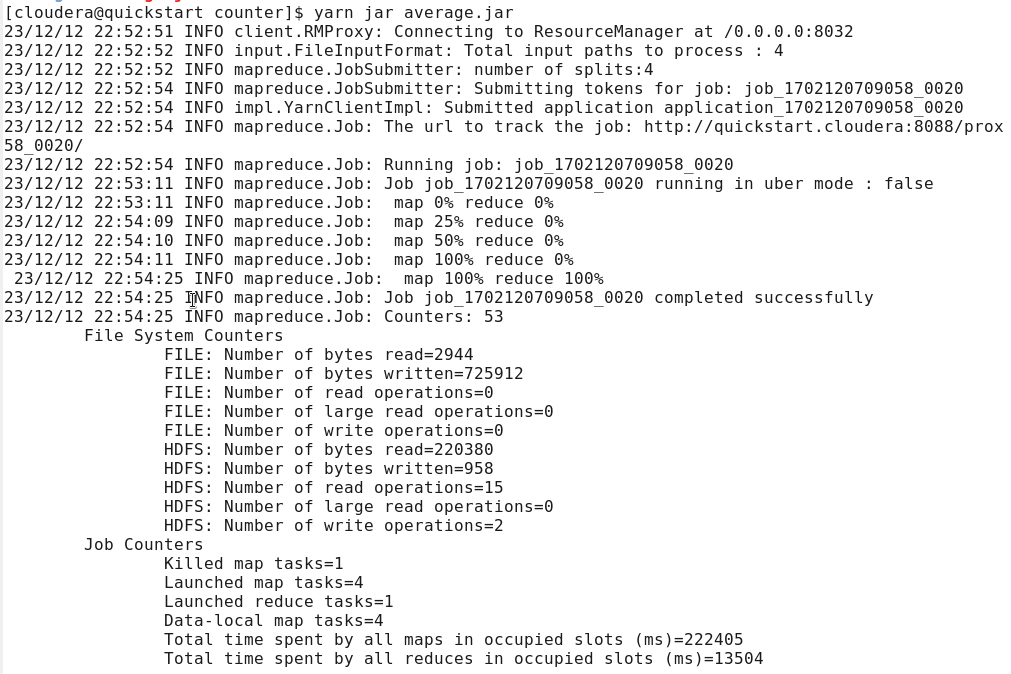


Make the normal .jar file

Run it using : yarn jar average.jar average.AverageJob



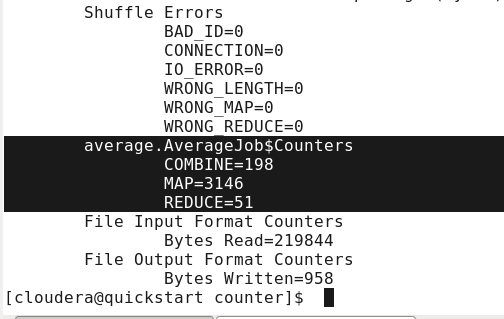
…



…



…



7] Compression

1) Download the rar file in **STAGING\_AREA**

2) Extract it

3) Pull the extracted folder in **LABS\_AREA**

4) Create a **logfiles** folder on HDFS and put the log files from project folder into it

5) Create an eclipse project **compression**

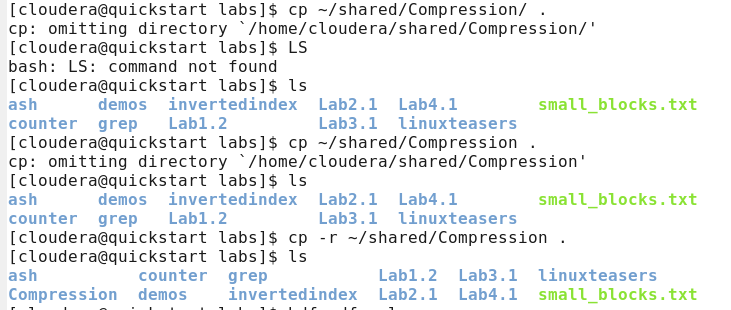
6) Import the java files from project folder in **STAGING\_AREA**   in to this project

7) Remove the compilation errors by adding hadoop client side libraries

8) Create a jar **compression.jar**

9) Have a first Run using **yarn jar compression.jar INFO** command

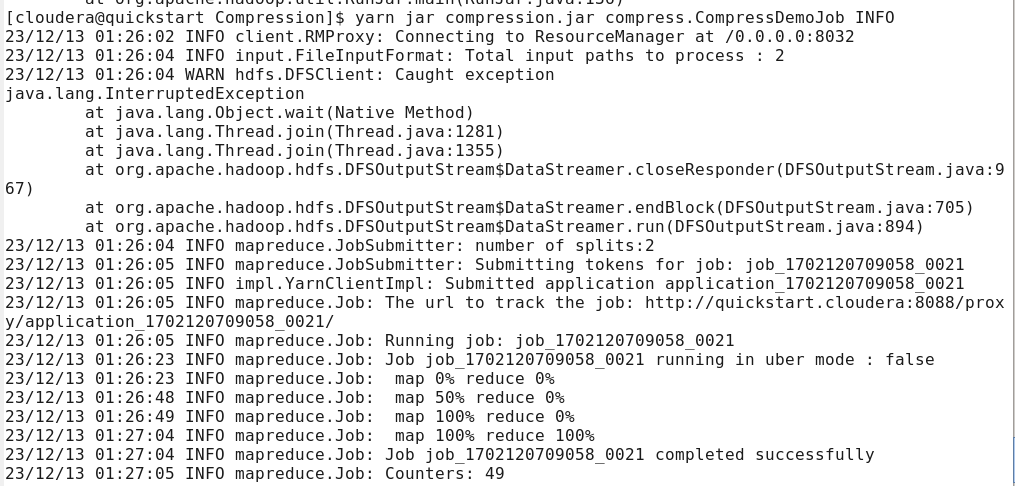
10) Wait for the in



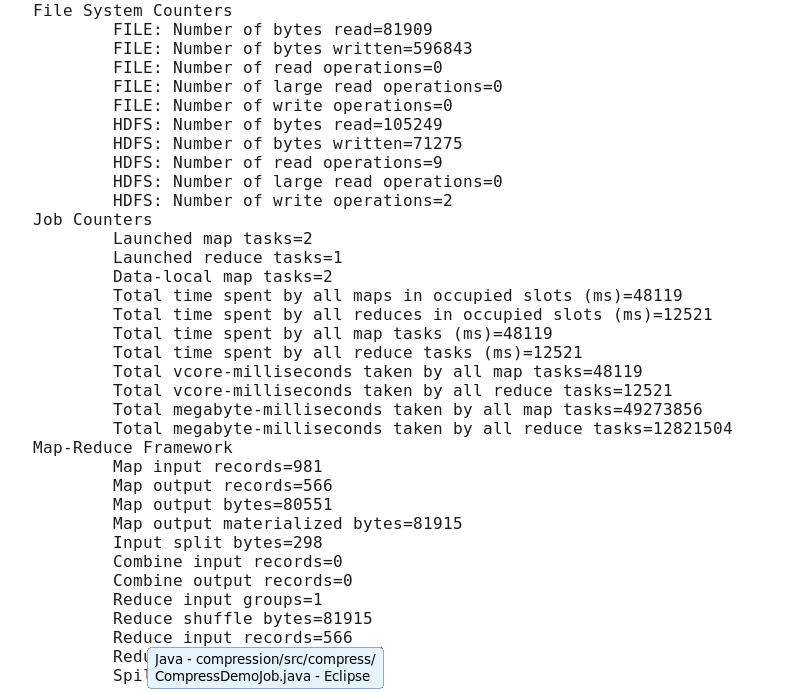
…



…



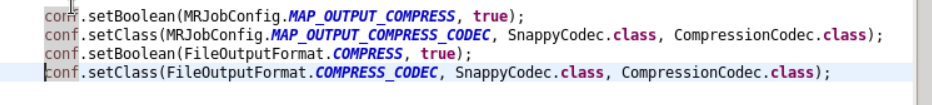
…



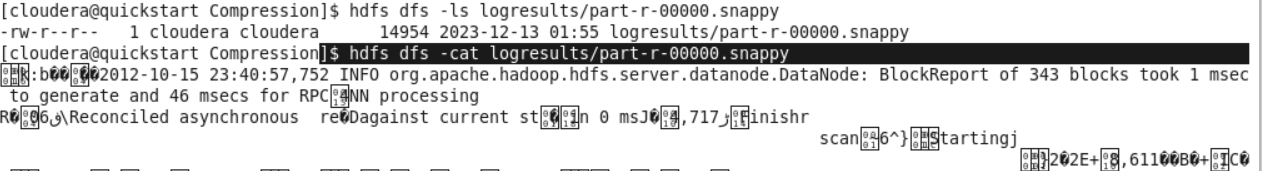




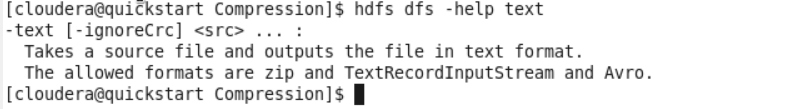
Now uncomment these lines from code



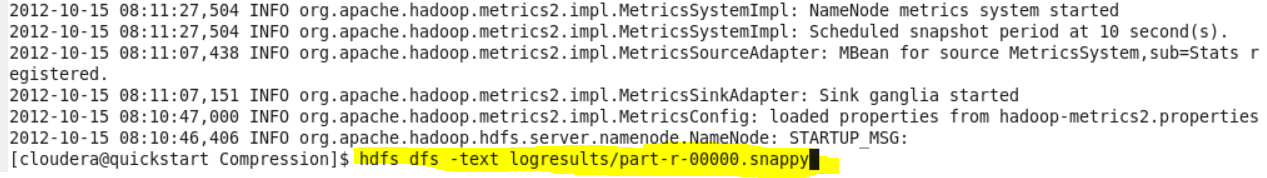
…

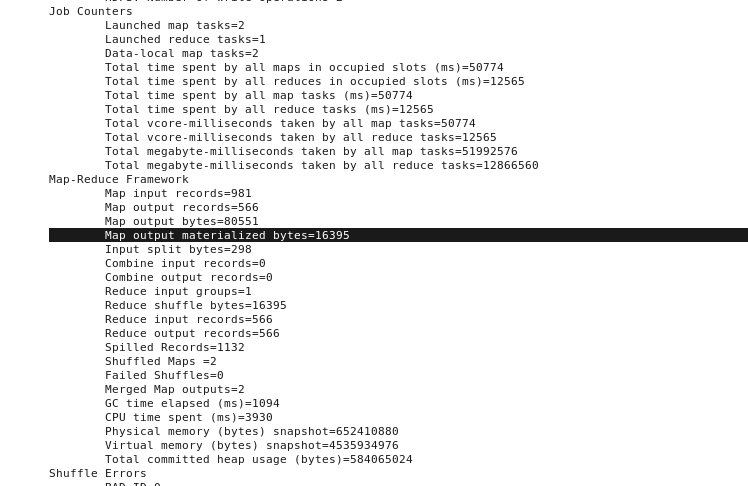


…



…





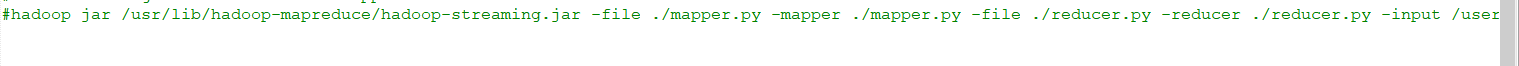
Compare same with the without compression output

LAB-> MRStreaming

Python

Put MRStreaming folder in LABS\_AREA in our staging area, it contains mapper.py and reducer.py

Run the following command as it is



Then

hdfs dfs –ls

dhfs dfs –ls dft-output

dhfs dfs –cat dft-output/……00000