

Trending YouTube Video Analysis

Sai Krishna Veeravelly(S544724)
Sai Krishna Vuppala(S545040)
Gayatri Devabhaktuni(S545030)
Laxman kumar Alugubelli (S545015)
Pavan Kumar Koppuravuri(S545220)

Northwest Missouri State University, Maryville MO 64468, USA

1 Introduction

This dataset includes several months of data on daily basis of the uploaded YouTube videos. This is about the data of different countries all over the world with up to 200 listed trending videos per day. The data file includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count. The data also includes a column of category id to differentiate the videos. In this Project We use variety of tools, to analyze users interactions, to select the year's top-trending videos based on number of views, shares, comments and likes from different categories like Music videos, celebrity and/or reality TV performances, and the viral videos which are among the top performers on the YouTube trending list. we can also analyze channel's performance analytics.

2 5V's respect to project

2.1 Variety:

The data used is of four different countries. The data gives the complete youtube videos of different channels from all the four countries.

2.2 Velocity:

There are about 200 records of data which is recorded everyday over the period of several months. This vast data that streams shows the power of velocity in data processing technique.

2.3 Volume:

The total volume of the data contains around 140000 records from four different countries. Huge volume of data is streamed everyday from different channels.

2.4 Value:

We can generate a statistical form of data from the big dataset and also we can bring desired output from the dataset which we have taken. The output generated gives clear analysis of the youtube videos with respect to the goal in accordings to the user.

2.5 Veracity:

The dataset is extracted from a reputed organization, it is trustworthy and we can say that this data is accurate. The accuracy in the data gives the desired output, hence veracity plays a vital role in utilizing the data.

3 Goals

We have set following goals for the dataset which we choose

1. The goal is to know the average likes a channel got in all 4 countries.(Sai krishna Vuppala)
2. The goal is to know the average number of views a channel got in all 4 countries.(Sai krishna Vuppala)
3. The goal is to determine the total number of comments received by a channel for all videos in each of the four countries.(Gayatri Devabhaktuni)
4. The goal is to know the total number of likes received by all videos in each channel for the four countries.(Gayatri Devabhaktuni)
5. The goal is to know the total dislikes received by all videos in each channel for the four countries.(Sai Krishna Veeravelly)
6. The goal is to know the total number of views for each channel got in all 4 countries.(Sai Krishna Veeravelly)
7. The goal is to know the number of videos uploaded by each channel.(Pavan Kumar Koppuravuri)
8. The goal is to know the number of likes a channel got on a particular day.(Pavan Kumar Koppuravuri)
9. The goal is to know the total views on YouTube on a particular day.(Laxman kumar Alugubelli)
10. The goal is to know the total views a channel got on a particular day.(Laxman kumar Alugubelli)

4 Tools and Technologies

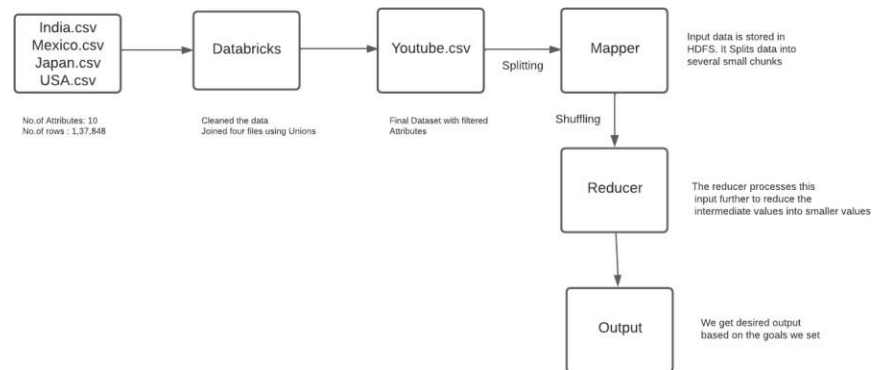
4.1 Data bricks

Data bricks is an unified and open source platform which helps to run interactive and scheduled data analysis. It is also extremely flexible and easy to get started on, making distributed analytics much easier to use. The initial dataset we took has huge amount of data from multiple countries. The data is segregated with required fields of four countries using groups/joins in the data bricks. The filtered data is used as a final dataset for processing the project.

4.2 Hadoop

In Hadoop, we used Map Reduce technique, which is a Hadoop framework for writing applications that can process vast amounts of data on large clusters. There are two primary tasks in Map Reduce: Map, reduce. In the map job, we split the input dataset into chunks. Map task processes these chunks in parallel. We used map-reduce techniques to analyze the YouTube dataset of four countries to get the different outputs based on the goals we set.

5 Block Diagram



6 Implementation

6.1 Map reduce

Steps for MapReduce Program Execution

1. Download the MapReduceDemo.zip file. To download, click on code button and select 'Download ZIP' (<https://github.com/bandiajay/MapReduce—zip>) file can be downloaded from here)
2. Extract the MapReduceDemo.zip file and copy the MapReduceDemo folder to the Eclipse-workspace (You can find Eclipse-workspace in C:\Your SID\)(if your system doesn't have Eclipse IDE follow the steps for Eclipse Installation given below)
3. In the Eclipse, File-Open Projects from File System Then, click on "Directory" in "Import Projects from File System or Archive" window, then browse the project folder from Eclipse workspace.
4. Change the program arguments by following below steps

- (a) Right click on the project select Properties
- (b) Select Run/Debug settings in the "Properties for MapReduceDemo" window, and click "New" button. Then, select Java Application, click OK. Give name as "ViewCount". Under the 'Main' tab, and in the "Main class:" search for "ViewCount-mapreducedemo", select it and click OK.
- (c) Select the ViewCount and click on 'Edit' button
- (d) Select the "Arguments" tab, and in the Program Arguments give path for input file and output file as
 - text C:-Users-YourS-ID-eclipse-workspace-MapReduceDemo-data-input-rawViews.txt C:-Users-YourS-ID-eclipse-workspace-MapReduceDemo-data-output and Click Ok.,
5. Delete the output Folder from the MapReduceDemo folder (Location: C:/Users - Your SID/eclipse-workspace/MapReduceDemo/data)
6. To check for output go to data folder inside project an output folder will be generated after successfully executing program
7. Open part-r-00000 file inside output folder in your project where the output of the program is stored. Run the ViewCount.java file to get desired output

6.2 Steps for Eclipse Installation

1. Eclipse: <https://www.eclipse.org/>
2. Download and install the latest version of Eclipse installer.
3. Get the Installer for your platform (student/faculty laptops are windows-x64).
4. Install Eclipse IDE for Enterprise Java and Web Developers from the installer menu.

6.3 Steps to add winutils.exe

1. Download winutils.exe and Hadoop.dll from <https://github.com/steveloughran/winutils/tree/master/hadoop-2.7.1/bin>
2. Create a folder hadoop in windows C in that create bin folder and place winutils.exe file and Hadoop.dll in bin folder
3. Create a new system variable give name as Hadoop-Home and give variable value as C:/hadoop and click ok
4. Click on path in system variables and click on edit option the click on new and give C:/hadoop/bin the click ok
5. Now close Exlipse IDE once and open and run ViewCount.java

6.4 DataBricks

1. Have an account in Data bricks community edition, after logging in you need to go to workspace and create a new one.
2. To run the workspace successfully we need to have cluster attached and running.
3. After attaching the cluster we can run the code in the workspace and get the desired output

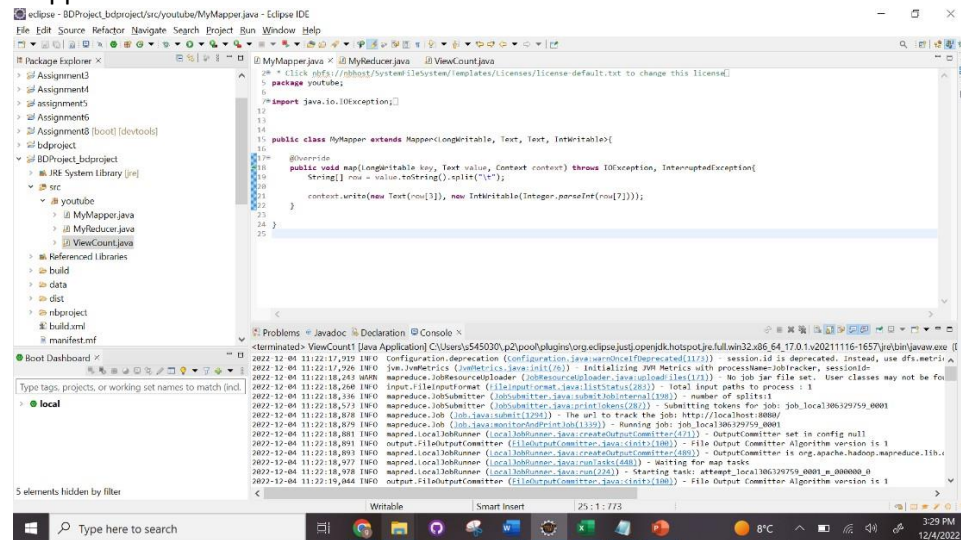
7 Results

7.1 Goal 1

The goal is to know the average number of likes a channel got in all 4 countries.

Here, we are able to get the average number of likes a channel got for the 4 countries we considered. The time taken to achieve this goal is 30 minutes. To attain this goal we combined all the data from 4 countries and cleaned it using Databricks. After cleaning we downloaded the data in .csv format and changed it to a .txt file, as it is needed for the map-reduce function in Hadoop.

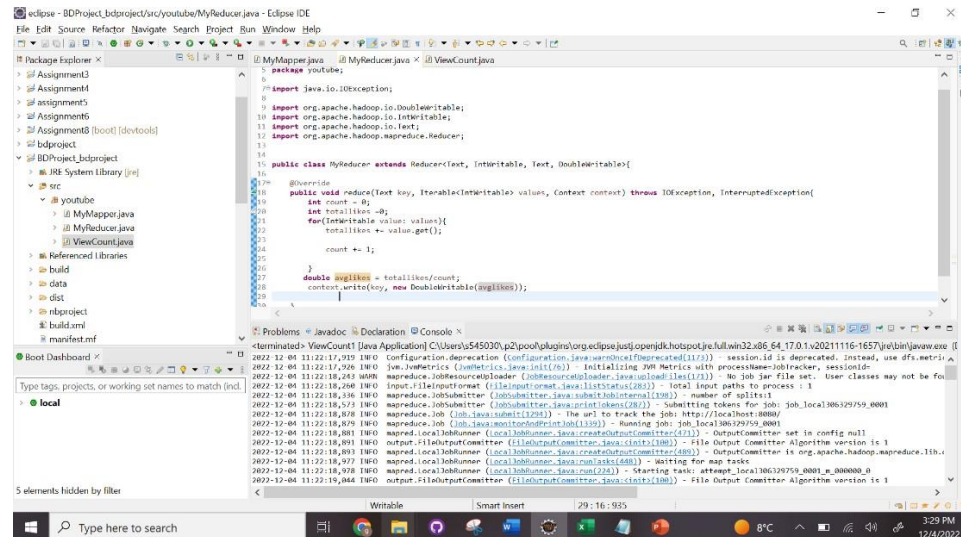
Mapper:



```
1 package youtube;
2
3 import java.io.IOException;
4
5 public class MyMapper extends Mapper<IntegerWritable, Text, Text, IntegerWritable> {
6
7     @Override
8     public void map(IntegerWritable key, Text value, Context context) throws IOException, InterruptedException {
9         String[] row = value.toString().split("\t");
10
11         context.write(new Text(row[3]), new IntegerWritable(Integer.parseInt(row[7])));
12     }
13 }
14
15
```

The console output shows the progress of the MapReduce job, including the number of splits, the number of map tasks, and the number of reduce tasks.

Reducer:



```
1 package youtube;
2
3 import java.io.IOException;
4 import org.apache.hadoop.io.DoubleWritable;
5 import org.apache.hadoop.io.IntWritable;
6 import org.apache.hadoop.mapreduce.Reducer;
7
8 public class MyReducer extends Reducer<Text, IntegerWritable, Text, DoubleWritable> {
9
10     @Override
11     public void reduce(Text key, Iterable<IntegerWritable> values, Context context) throws IOException, InterruptedException {
12         int count = 0;
13         int totalLikes = 0;
14         for (IntegerWritable value : values) {
15             totalLikes += value.get();
16             count++;
17         }
18         double avgLikes = totalLikes / count;
19         context.write(key, new DoubleWritable(avgLikes));
20     }
21 }
22
```

The console output shows the progress of the MapReduce job, including the number of splits, the number of map tasks, and the number of reduce tasks.

6

Output Screenshot:

```
channel and avg likes.txt - Notepad
File Edit Format View Help
AB Music 10.0
ABBY RABBIT 5.0
ABC Action News 493.0
ABC Malayalam 78.0
ABC News 2277.0
ABC News (Australia) 1809.0
ABC Noticias 6.0
ABC Television Network 1152.0
ABC Television Stations 2369.0
ABC la serie 1.0
ABC movies 9.0
ABC7 1032.0
ABCMART/ABC7 0.0
ABN Telugu 2200.0
ABP ANANDA 247.0
ABP Asmita 1133.0
ABP Majha 321.0
ABP NEWS 676.0
ABP NEWS HINDI 780.0
ABP Sanjha 158.0
ABS-CBN Entertainment 172.0
ABS-CBN News 495.0
ACCOUNT ANIME LOVERS 3.0
ACE SPORTS 64.0
ACE TENNIS 20.0
ACERTIJS EN 7 SEGUNDOS 35.0
ACT II 2336.0
AD-WISE MEDIA ACTION MOVIEPLEX 1460.0
ADAMI RAMDON SDLG 9.0
ADN Opini 69.0
ADRIANA LUNA VIDEOS 65.0
AE Noticias 12.0
AF FOOTBALL 113.0
AFP news agency 2.0
```

7.2 Goal 2

The goal is to know the average number of views a channel got in all 4 countries.

We got the desired output for this goal by applying map reduce techniques on channel row and views row, we have written mapper and reducer class in a way that it should display the average views for each channel in all 4 countries. The time taken to achieve this goal is 15 minutes.

Mapper

```

1 package youtube;
2
3 import java.io.IOException;
4
5 public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
6
7     @Override
8     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
9         String[] row = value.toString().split("\t");
10
11         context.write(new Text(row[1]), new IntWritable(Integer.parseInt(row[0])));
12     }
13 }
14
15
16
17
18
19
20
21
22
23
24
25

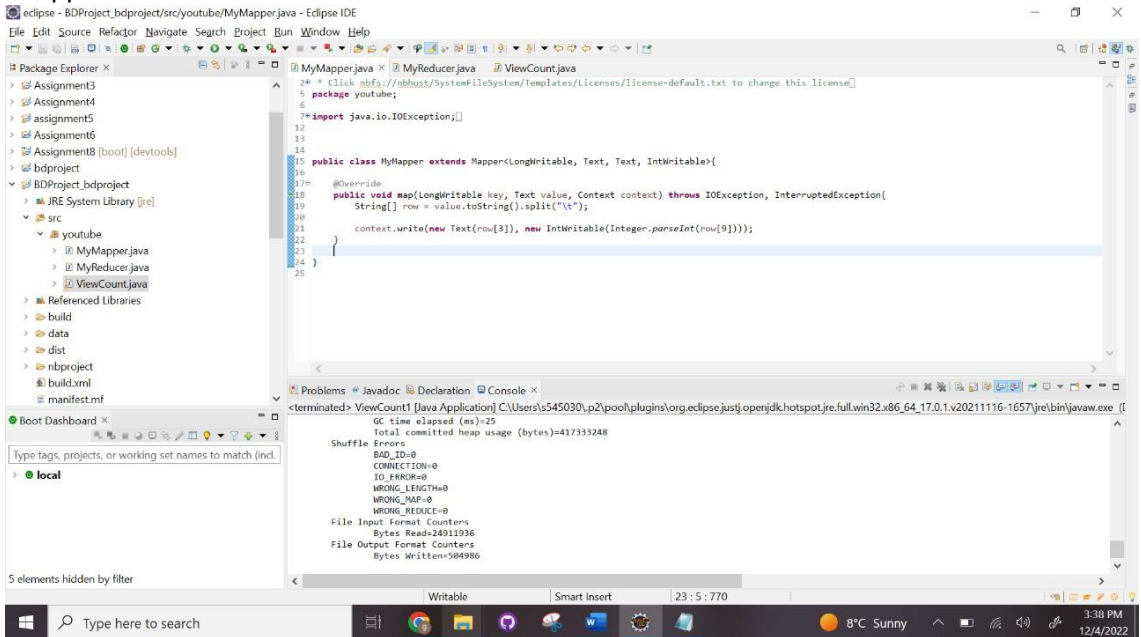
```

```

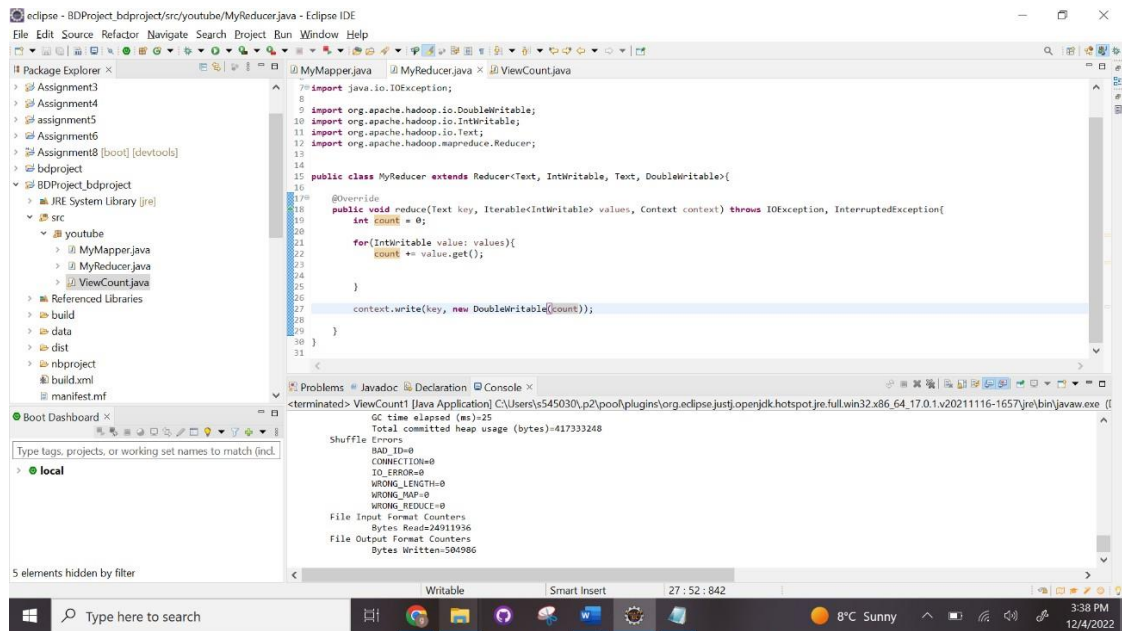
<terminated> ViewCount1 [Java Application] C:\Users\s45030\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1\v20211116-1657\jre\bin\java.exe
2022-12-04 11:22:17,912 INFO Configuration.deprecation (Configuration.java:989:Deprecated[112]) session.id is deprecated. Instead, use dfs.metrics
2022-12-04 11:22:17,920 INFO jvm.MetricInfo (JvmMetrics.java:1176) - Initializing JVM Metrics with processName=JobTracker, sessionId=
2022-12-04 11:22:18,243 WARN mapreduce.job.resourceuploader (JobResourceUploader.java:loadFiles(171)) - No job jar file set. User classes may not be found.
2022-12-04 11:22:18,260 INFO InputFileInputFormat (InputOutputFormat.java:1155:load(482)) Total input paths to process : 1
2022-12-04 11:22:18,356 INFO mapreduce.job.submitter (JobSubmitter.java:submitJobInternally(188)) - number of splits: 1
2022-12-04 11:22:18,573 INFO mapreduce.job.submitter (JobSubmitter.java:printToken(287)) - Submitting tokens for job: job_local306329759_0001
2022-12-04 11:22:18,879 INFO mapreduce.job (Job.java:submit(1324)) - The url to track the job: http://localhost:8080/
2022-12-04 11:22:18,879 INFO mapreduce.job (Job.java:monitorAndPrintJob(433)) - Running job: job_local306329759_0001
2022-12-04 11:22:18,881 INFO mapred.localjobrunner (LocalJobRunner.java:createOutputCommitter(571)) - OutputCommitter set in config: null
2022-12-04 11:22:18,891 INFO output.fileoutputcommitter (FileOutputCommitter.java:commit(180)) - File OutputCommitter Algorithm version is 1
2022-12-04 11:22:18,893 INFO mapred.localjobrunner (LocalJobRunner.java:createOutputCommitter(489)) - OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2022-12-04 11:22:18,977 INFO mapred.localjobrunner (LocalJobRunner.java:runTask(480)) - Waiting for map tasks
2022-12-04 11:22:18,978 INFO mapred.localjobrunner (LocalJobRunner.java:run(226)) - Starting task: attempt_local306329759_0001_m_000000_0
2022-12-04 11:22:19,644 INFO output.fileoutputcommitter (FileOutputCommitter.java:commit(180)) - File OutputCommitter Algorithm version is 1

```


8 Mapper



Reducer



Output:

```

channel and comment count.txt - Notepad
File Edit Format View Help
ABP Asmita 4893
ABP Majha 33658
ABP NEWS 50496
ABP NEWS HINDI 88806
ABP Sanjha 2233
ABS-CBN Entertainment 8889
ABS-CBN News 2512
ACCOUNT ANIME LOVERS 17
ACE SPORTS 1133
ACE TENNIS 315
ACERTI3OS EN 7 SEGUNDOS 232
ACT II 114
AD-WISE MEDIA ACTION MOVIEPLEX 7811
ADAMI RAMDON SOLG 76
ADM Opini0000n 1448
ADRIANA LUNA VIDEOS 398
AE Noticias 14
AF FOOTBALL 18427
AFP news agency 2
AFernandezVEVO 1080
AFmedios Noticias de Colima 13734
AGENCIA EFE 364
AGM TV NETWORK 3
AGUSTIN NIETO 95
AHATO 127
AIA awards 313
AIB Doosra 18901
AJ Official 2244
AJcompsHD 21
AK Entertainments 26905
AK news 2314
AKB48 22691
AKB48Ecuador 15
AKLO 1268

```

7.4 Goal 4

The goal is to know the total number of likes received by all videos in each channel for the four countries.

We got the desired output for this goal by applying map reduce techniques on channel row and likes row, we have written mapper and reducer classes in such a way that it should display the total number of likes received by a channel for all videos they uploaded. The time taken to achieve this goal is 15 minutes.

Mapper

```

// MyMapper.java
package youtube;

import java.io.IOException;

public class MyMapper extends Mapper{

    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
        String[] row = value.toString().split("\t");
        context.write(new Text(row[3]), new IntWritable(Integer.parseInt(row[4])));
    }
}

```

```

Shuffle: Error
BAD_T0-0
CORRECT_LENGTH-0
TO_ERROR-0
WRONG_LENGTH-0
WRONG_PATH-0
WRONG_REDUCE-0
File Input Format Counters
  Bytes Read:24311956
File Output Format Counters
  Bytes Written:504986

```

1 Reducer

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with 'BDProject' containing 'src' and 'resources'. The 'src' folder contains 'MyMapper.java', 'MyReducer.java', and 'ViewCount.java'.
- MyMapper.java:** Contains the following code:

```
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.IntWritable;
4 import org.apache.hadoop.io.Text;
5 import org.apache.hadoop.mapreduce.Mapper;
6
7 public class MyMapper extends Mapper<Text, IntWritable, Text, DoubleWritable>{
8
9     @Override
10     public void map(Text key, IntWritable value, Context context) throws IOException, InterruptedException{
11         context.write(key, new DoubleWritable(value.get()));
12     }
13 }
```
- MyReducer.java:** Contains the following code:

```
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.DoubleWritable;
4 import org.apache.hadoop.io.IntWritable;
5 import org.apache.hadoop.io.Text;
6 import org.apache.hadoop.mapreduce.Reducer;
7
8 public class MyReducer extends Reducer<Text, IntWritable, Text, DoubleWritable>{
9
10     @Override
11     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
12         int count = 0;
13         for(IntWritable value: values){
14             count += value.get();
15         }
16         context.write(key, new DoubleWritable(count));
17     }
18 }
```
- ViewCount.java:** Contains the following code:

```
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.Text;
4 import org.apache.hadoop.mapreduce.Reducer;
5
6 public class ViewCount extends Reducer<Text, DoubleWritable, Text, DoubleWritable>{
7
8     @Override
9     public void reduce(Text key, Iterable<DoubleWritable> values, Context context) throws IOException, InterruptedException{
10         double count = 0.0;
11         for(DoubleWritable value: values){
12             count += value.get();
13         }
14         context.write(key, new DoubleWritable(count));
15     }
16 }
```
- Console:** Shows the output of the ViewCount1 [Java Application] process. The output is:

```
GC time elapsed (ms)=25
Total committed heap usage (bytes)=417333248
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=24911936
File Output Format Counters
Bytes Written=504086
```

Output:

The screenshot shows a Notepad window with the following output:

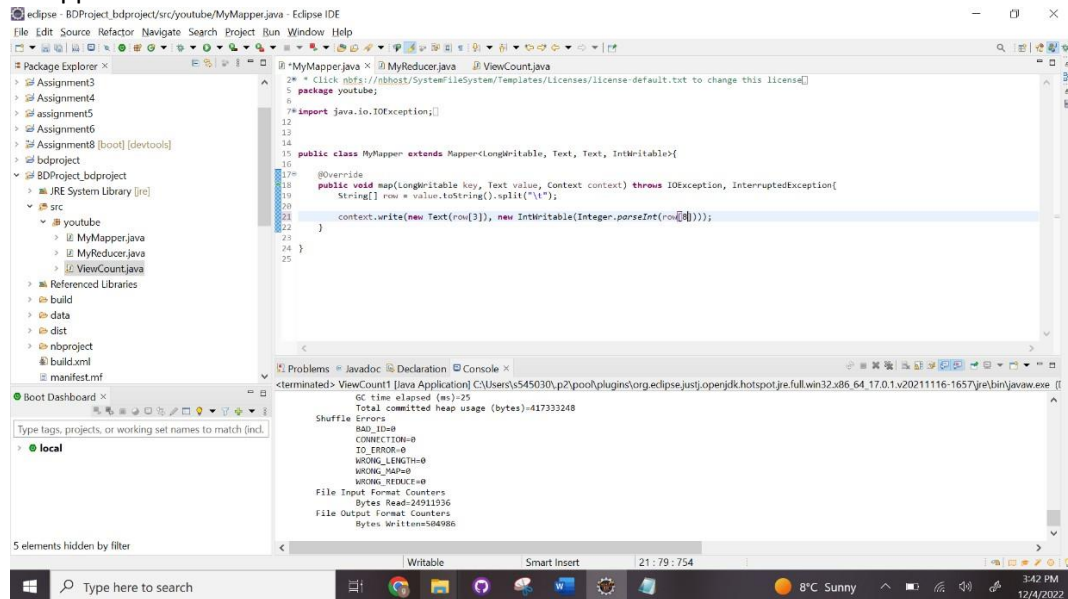
```
channel likes.txt - Notepad
File Edit Format View Help
A.I.Channel 152639
A.I.Games 1481
A0I 4209
A24 633096
AAlex Kenet 4332
AA00000 (aequal) 99
AB Music 71
ABBY RABBIT 13542
ABC Action News 82990
ABC Malayalam 3980
ABC News 464951
ABC News (Australia) 648023
ABC Noticias 49
ABC Television Network 824147
ABC Television Stations 15532
ABC 1a serie 276
ABC movies 886
ABC7 134523
ABCMART/AB000000000 0
ABN Telugu 195281
ABP ANANDA 35069
ABP Asmita 19038
ABP Mehta 202631
ABP NEWS 182263
ABP NEWS HINDI 262632
ABP Sanjha 16658
ABS-CBN Entertainment 86534
ABS-CBN News 7461
ACCOUNT ANIME LOVERS 346
ACE SPORTS 1318
ACE TENNIS 943
ACERTIJDOS EN 7 SEGUNDOS 431
ACT II 2957
AD-WISE MEDIA ACTION MOVIEPLEX 162321
ADWISERUN 316
```

7.5 Goal 5

The goal is to know the total dislikes received by all videos in each channel for the four countries.

We got the desired output for this goal by applying map reduce techniques on channel row and dislikes row, we have written mapper and reducer classes in such a way that it should display the total number of dislikes received by a channel for all videos they uploaded. The time taken to achieve this goal is 15 minutes.

Mapper:



The screenshot shows the Eclipse IDE with the 'MyMapper.java' file open. The code defines a class 'MyMapper' extending 'Mapper<LongWritable, Text, Text, IntWritable>'. It overrides the 'map' method to process input rows, splitting them into channel and video information, and writing the channel ID and video ID to the context. The console output shows the execution of 'ViewCount1' with a GC time of 25ms and a total committed heap usage of 417333248 bytes. The output also displays shuffle errors and file input/output format counters.

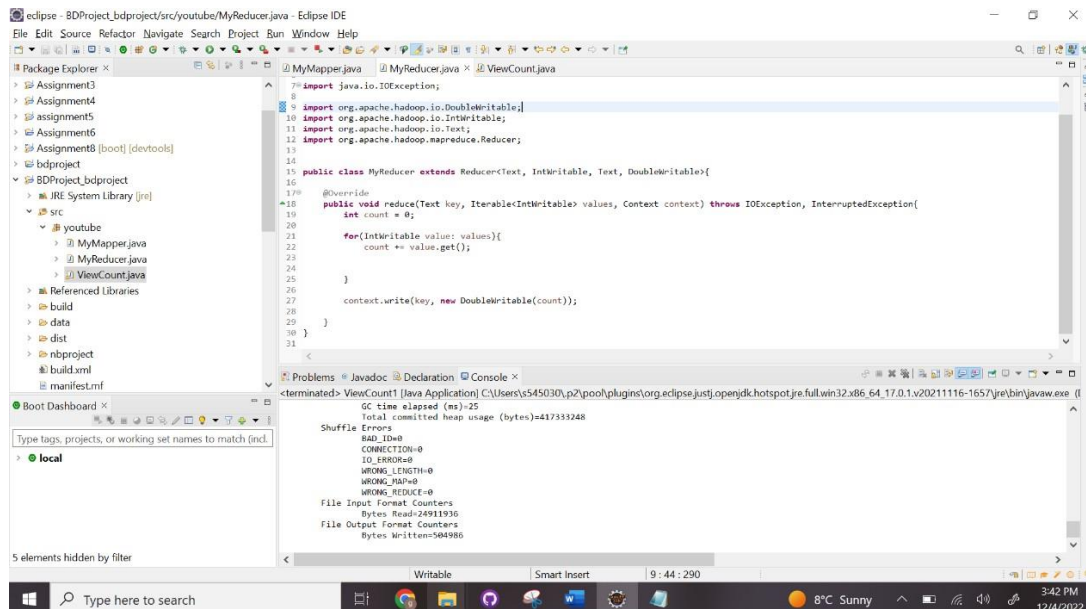
```
1 package youtube;
2
3 import java.io.IOException;
4
5 public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
6
7     @Override
8     public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
9         String[] row = value.toString().split("\t");
10
11         context.write(new Text(row[3]), new IntWritable(Integer.parseInt(row[4])));
12     }
13 }
14
15
16
17
18
19
20
21
22
23
24
25
```

terminated> ViewCount1 [Java Application] C:\Users\S45030\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\java.exe [
GC time elapsed (ms)=25
Total committed heap usage (bytes)=417333248

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=24911936
File Output Format Counters
Bytes Written=504986

Reducer:



The screenshot shows the Eclipse IDE with the 'MyReducer.java' file open. The code defines a class 'MyReducer' extending 'Reducer<Text, IntWritable, Text, DoubleWritable>'. It overrides the 'reduce' method to process input values, summing the counts for each channel ID and writing the result to the context. The console output shows the execution of 'ViewCount1' with a GC time of 25ms and a total committed heap usage of 417333248 bytes. The output also displays shuffle errors and file input/output format counters.

```
1 import java.io.IOException;
2
3 import org.apache.hadoop.io.DoubleWritable;
4 import org.apache.hadoop.io.IntWritable;
5 import org.apache.hadoop.io.Text;
6 import org.apache.hadoop.mapreduce.Reducer;
7
8 public class MyReducer extends Reducer<Text, IntWritable, Text, DoubleWritable>{
9
10     @Override
11     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
12         int count = 0;
13
14         for(IntWritable value: values){
15             count += value.get();
16         }
17
18         context.write(key, new DoubleWritable(count));
19     }
20 }
21
22
23
24
25
26
27
28
29
30
31
```

terminated> ViewCount1 [Java Application] C:\Users\S45030\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\java.exe [
GC time elapsed (ms)=25
Total committed heap usage (bytes)=417333248

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=24911936
File Output Format Counters
Bytes Written=504986

1

Output:

```
channel and total dislikes.txt - Notepad
File Edit Format View Help
Ai Sakuranbo 6
Aikatsu Friends! Channel ep1-25 20
Aikatsu Stars! English Subbed 0
Aikatsu Stars! season2 channel ep76 - ep100 14
Aimer Official YouTube Channel 221
Ainu 36
AirPano 72
Airon 7
Aishwarya Majmudar 780
Ajendra Creations 49911
Aji Gila 31
Ajuchitlan del Progreso Guerrero 148
Akane Sayuri 11
Akapellah Official 738
Akari Beauty Official 616
Akashi Tv 13651
Aki Fukazawa 10
Akira Lau 40
Akira Suzumiya 183
Akki & Ankit 2869
Akumaili 323
Al Jazeera English 2117
Al Rojo Vivo 2753
Alabama Crimson Tide on AL.com 1259
Alaila Rivera 32
Alan - Crack***via 21
Alan Quartz C. 445
Alan Rawady 5
Alan Rojas 5
Alan Salda***a oficial 26
Alan Walker 92059
Albert Ameri 1457
Albert Plash 1358
Albertiwi 879
```

7.6 Goal 6

The goal is to know the total number of views for each channel got in all 4 countries.

We got the desired output for this goal by applying map reduce techniques on channel row and views row, we have written mapper and reducer classes in such a way that it should display the total number of views received by a channel for all videos they uploaded. The time taken to achieve this goal is 15 minutes.

Mapper

```
eclipse - BDProject_bdproject/src/youtube/MyMapper.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
  BDProject_bdproject
    src
      youtube
        MyMapper.java
        MyReducer.java
        ViewCount.java
  Referenced Libraries
  build
  data
  dist
  nbproject
  build.xml
  manifest.mf
  Boot Dashboard
  local
  5 elements hidden by filter

MyMapper.java
17
18
19
20
21
22
23
24
25

public class MyMapper extends Mapper{
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
        String[] row = value.toString().split("\t");
        context.write(new Text(row[1]), new IntWritable(Integer.parseInt(row[6])));
    }
}

Problems Javadoc Declaration Console
terminated ViewCount1 [Java Application] C:\Users\s45030\p2\p2\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\javaw.exe
GC time elapsed (ms)=25
Total committed heap usage (bytes)=41733248
Shuffle Errors
BAD_ID=0
COMMITMENT=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=24911936
File Output Format Counters
Bytes Written=504980
```

Reducer:

```

7 *import java.io.IOException;
8
9 *import org.apache.hadoop.io.DoubleWritable;
10 *import org.apache.hadoop.io.IntWritable;
11 *import org.apache.hadoop.io.Text;
12 *import org.apache.hadoop.mapreduce.Reducer;
13
14
15 public class MyReducer extends Reducer<Text, IntWritable, Text, DoubleWritable>{
16
17     @Override
18     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
19         int count = 0;
20
21         for(IntWritable value: values){
22             count += value.get();
23         }
24
25         context.write(key, new DoubleWritable(count));
26     }
27 }
28
29
30
31

```

terminated> ViewCount1 [Java Application] C:\Users\s45030\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\javaw.exe (I

GC time elapsed (ms)=25
Total committed heap usage (bytes)=417333248

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_EXCEPTION=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=24911936
File Output Format Counters
Bytes Written=504986

Output:

```

channel and total views.txt - Notepad
File Edit Format View Help
!***** 11528
"1,999,999 visualizaciones" 249692
"10,031,1993 Viewss" 2547
"A to Z cricket,political & cinema" 2573673
"BTS,GOT7,EXO K-POP" 269130
"Bordados, Tejidos y Mas." 48690
"But First, Coffee" 6073847
"Cyprus Air Heating, Cooling and Fireplaces" 1259046
"Dietas, Salud y Belleza" 5953
"ED Futbol, Tennis and more" 2618052
"EL VOLCAN DEL NEPA, GUERRAS DJ" 237190
"European Space Agency, ESA" 659754
"F1, Ferrari y algo mas" 13061
"Federaci*****n Mexicana de Charrer*****a, A.C." 565702
"Ganan, Gustar, Golear" 42807
"Hades Pro, Humor a la Mexicana" 111372
"Mi Salud, Curiosidad y Mas" 45639
"One List , One Life" 241520
"P. Guillermo Serra, LC" 16382
"Pensamientos, reflexiones y dedicatorias" 139888
"Pictures, Noise and Words" 15712
"Pop, Sports & Stuff" 103942
"Private Music - Guaracha, Aleteo, Zapateo, Tribal" 42143
"Realitys, Noticias" 5655413
"Recetas Faciles, *Al estilo chef roger*" 255537
"Rock me, Joey Santiago." 7806618
"Ruedas, Potencia y M*****s" 20997
"Salud, dinero y exito" 9275
"Soy Yoh, un Chibi :D" 39502
"Tyler, The Creator" 44914541
"monday.com, formerly dapulse" 15544
"nm," 48321
"*****hiroyuki at Youtube" 207097
"***** NatureLab Co., Ltd." 99150

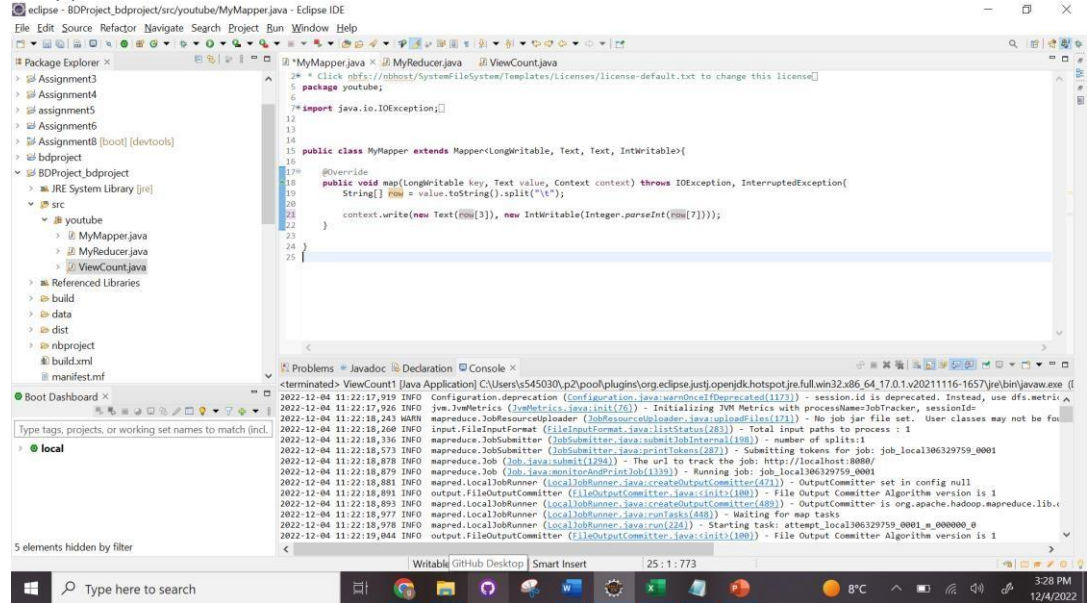
```

7.7 Goal 7

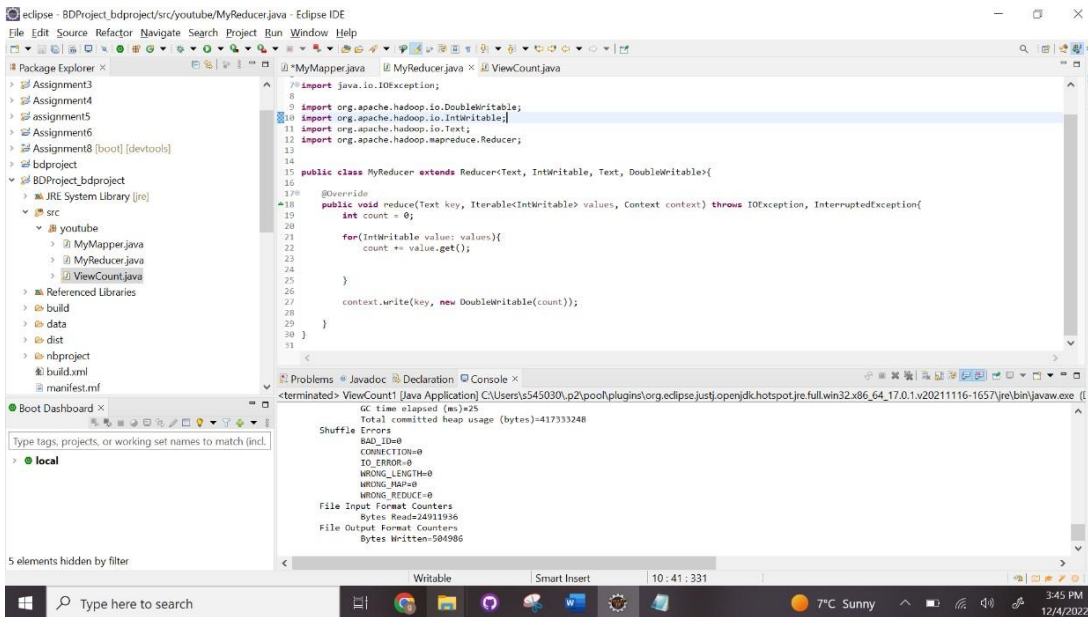
The goal is to know the number of videos uploaded by each channel.

We got the desired output for this goal by applying map reduce techniques on channel row, we have written mapper and reducer classes in such a way that it should display the total number of videos uploaded by a channel of all 4 countries. The time taken to achieve this goal is 15 minutes.

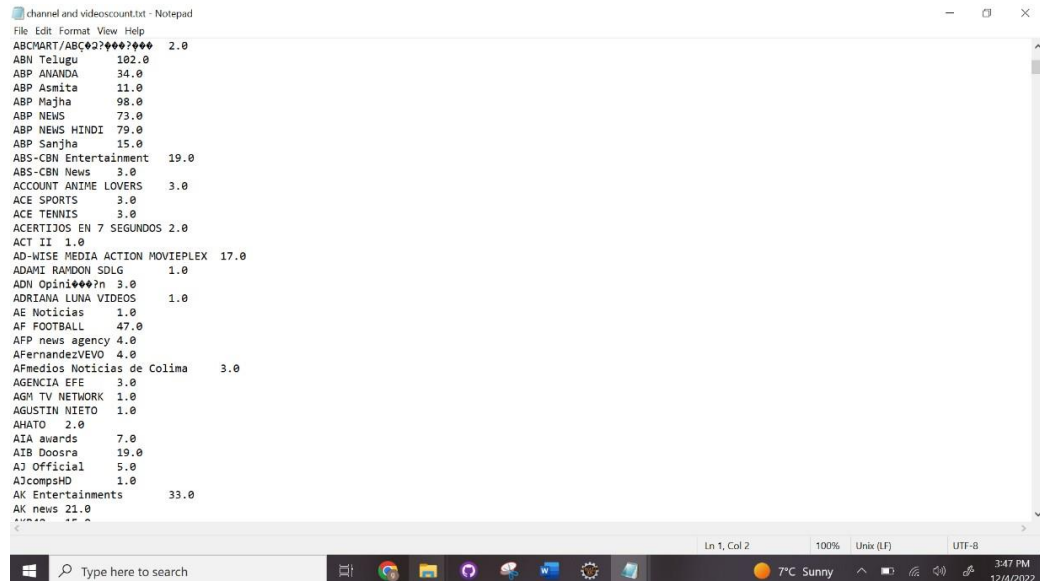
1 Mapper



Reducer:



Output:



```

channel and videocount.txt - Notepad
File Edit Format View Help
ABCMART/ABC 2.0
ABN Telugu 102.0
ABP ANANDA 34.0
ABP Asmita 11.0
ABP Majha 98.0
ABP NEWS 73.0
ABP NEWS HINDI 79.0
ABP Sanjha 15.0
ABS-CBN Entertainment 19.0
ABS-CBN News 3.0
ACCOUNT ANIME LOVERS 3.0
ACE SPORTS 3.0
ACE TENNIS 3.0
ACERTIJDOS EN 7 SEGUNDOS 2.0
ACT II 1.0
AD-WISE MEDIA ACTION MOVIEPLEX 17.0
ADAMI RANDON SDLG 1.0
ADN Opini 3.0
ADRIANA LUNA VIDEOS 1.0
AE Noticias 1.0
AF FOOTBALL 47.0
AFP news agency 4.0
AFernandezVEVO 4.0
AFmedios Noticias de Colima 3.0
AGENCIA EFE 3.0
AGM TV NETWORK 1.0
AGUSTIN NIETO 1.0
AHATO 2.0
AIA Awards 7.0
ATB Doosra 19.0
AJ Official 5.0
AJcompsHD 1.0
AK Entertainments 33.0
AK news 21.0

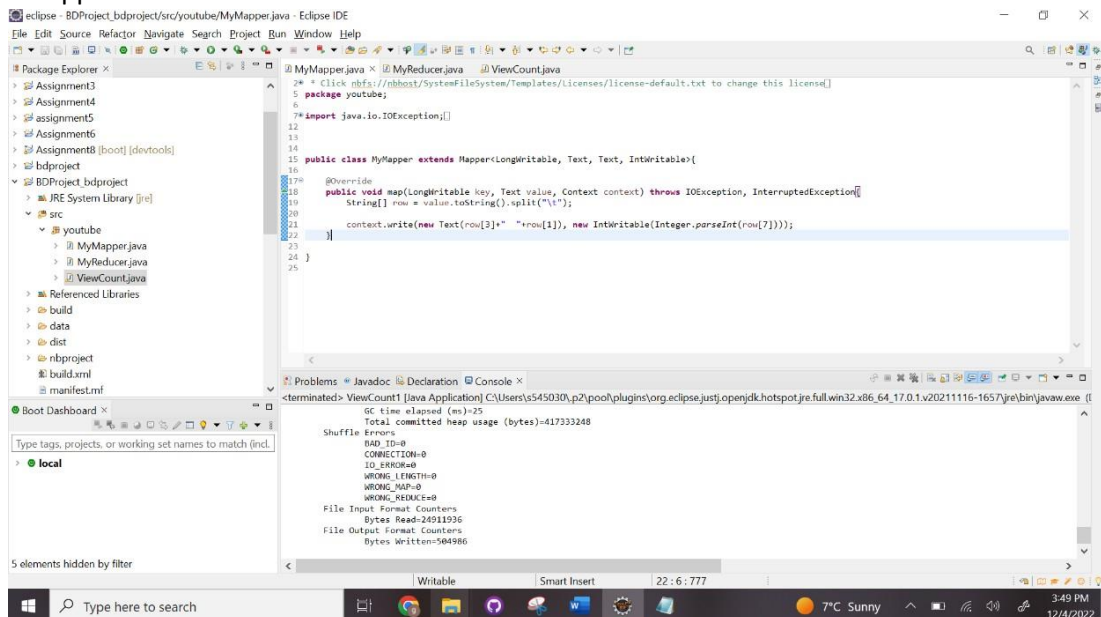
```

7.8 Goal 8

The goal is to know the number of likes a channel got on a particular day.

We got the desired output for this goal by applying map reduce techniques on likes row, channel row and dates row, we have written mapper and reducer classes in such a way that it should display the total number of likes received by a channel on a particular day for all videos they uploaded. The time taken to achieve this goal is 15 minutes.

Mapper



```

eclipse - BDProject_bdproject/src/youtube/MyMapper.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
  BDProject_bdproject
    JRE System Library [jre]
    src
      youtube
        MyMapper.java
        MyReducer.java
        ViewCount.java
    Referenced Libraries
    build
    data
    dist
    nbproject
    build.xml
    manifest.mf
  Boot Dashboard
  Type tags, projects, or working set names to match (incl. local)
  5 elements hidden by filter

MyMapper.java
package youtube;
import java.io.IOException;

public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
        String[] row = value.toString().split("\t");
        context.write(new Text(row[3]+" "+row[1]), new IntWritable(Integer.parseInt(row[7])));
    }
}

Problems Javadoc Declaration Console
terminated- ViewCount1 [Java Application] C:\Users\s45030\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\java.exe (t
GC time elapsed (ms)=25
Total committed heap usage (bytes)=417333248
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=24911936
File Output Format Counters
Bytes Written=504986

```


1

Reducer:

```

1  import java.io.IOException;
2
3  import org.apache.hadoop.io.DoubleWritable;
4  import org.apache.hadoop.io.IntWritable;
5  import org.apache.hadoop.io.Text;
6  import org.apache.hadoop.mapreduce.Reducer;
7
8  public class MyReducer extends Reducer<Text, IntWritable, Text, DoubleWritable>{
9
10     @Override
11     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
12         int count = 0;
13
14         for(IntWritable value: values){
15             count += value.get();
16         }
17
18         context.write(key, new DoubleWritable(count));
19     }
20 }

```

Problems * Javadoc Declaration Console

<terminated> ViewCount1 [Java Application] C:\Users\s454030\p2\pooh\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\javaw.exe (f

GC time elapsed (ms)=25

Shuffle Errors

Errors

BAD_ID=0

CONNECTION=0

IO_ERROR=0

WRONG_LENGTH=0

WRONG_MAP=0

WRONG_REDUCE=0

File Input Format Counters

Bytes Read=24911936

File Output Format Counters

Bytes Written=504986

Output:

```

1  channel day and likes.txt - Notepad
2
3  File Edit Format View Help
4
5  "Bordados, Tejidos y Mas." 18.18.04 18
6  "Bordados, Tejidos y Mas." 18.21.03 5
7  "Bordados, Tejidos y Mas." 18.24.04 17
8  "Bordados, Tejidos y Mas." 18.27.04 12
9  "Bordados, Tejidos y Mas." 18.30.04 8
10 "But First, Coffee" 17.08.12 966
11 "But First, Coffee" 17.09.12 1343
12 "But First, Coffee" 17.10.12 1605
13 "Cyprus Air Heating, Cooling and Fireplaces" 18.05.02 83
14 "Cyprus Air Heating, Cooling and Fireplaces" 18.06.02 402
15 "Cyprus Air Heating, Cooling and Fireplaces" 18.07.02 423
16 "Cyprus Air Heating, Cooling and Fireplaces" 18.08.02 434
17 "Dietas, Salud y Belleza" 18.30.04 11
18 "ED Futbol, Tennis and more" 18.02.04 42
19 "ED Futbol, Tennis and more" 18.04.03 19
20 "ED Futbol, Tennis and more" 18.04.04 88
21 "ED Futbol, Tennis and more" 18.14.05 217
22 "ED Futbol, Tennis and more" 18.18.03 35
23 "ED Futbol, Tennis and more" 18.18.04 173
24 "ED Futbol, Tennis and more" 18.18.05 64
25 "ED Futbol, Tennis and more" 18.19.02 66
26 "ED Futbol, Tennis and more" 18.19.03 39
27 "ED Futbol, Tennis and more" 18.19.05 66
28 "ED Futbol, Tennis and more" 18.21.04 103
29 "ED Futbol, Tennis and more" 18.21.05 92
30 "ED Futbol, Tennis and more" 18.22.04 89
31 "ED Futbol, Tennis and more" 18.24.02 49
32 "ED Futbol, Tennis and more" 18.25.02 38
33 "ED Futbol, Tennis and more" 18.26.04 218
34 "ED Futbol, Tennis and more" 18.27.04 223
35 "EL VOLCAN DEL WEPa, GUERRAS DJ" 18.20.04 173
36 "EL VOLCAN DEL WEPa, GUERRAS DJ" 18.30.03 87
37 "European Space Agency, ESA" 17.04.12 42
38 "European Space Agency, ESA" 17.05.12 64
39 "European Space Agency, ESA" 17.06.12 73

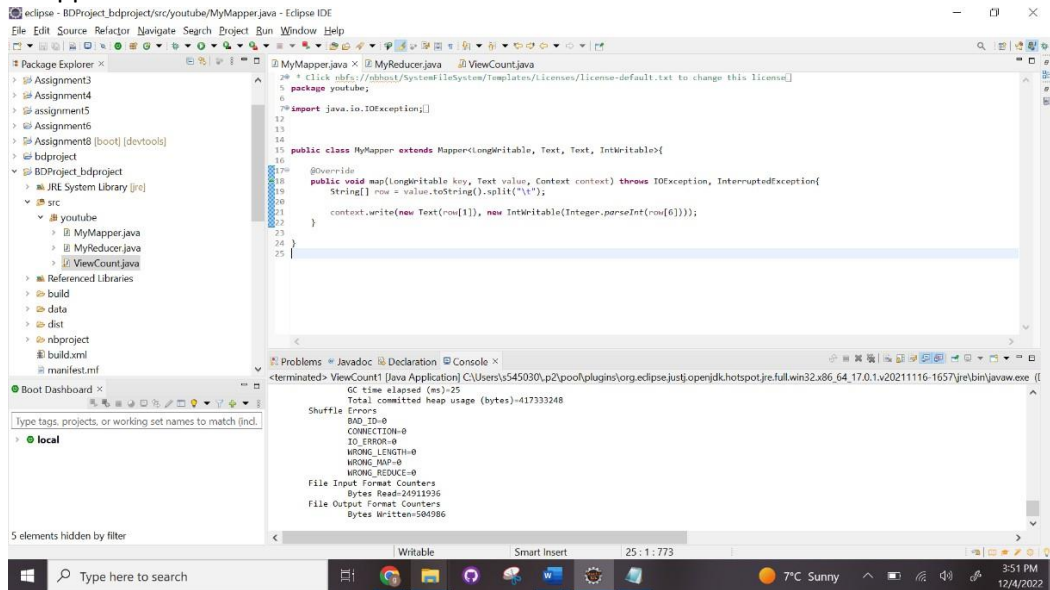
```

7.9 Goal 9

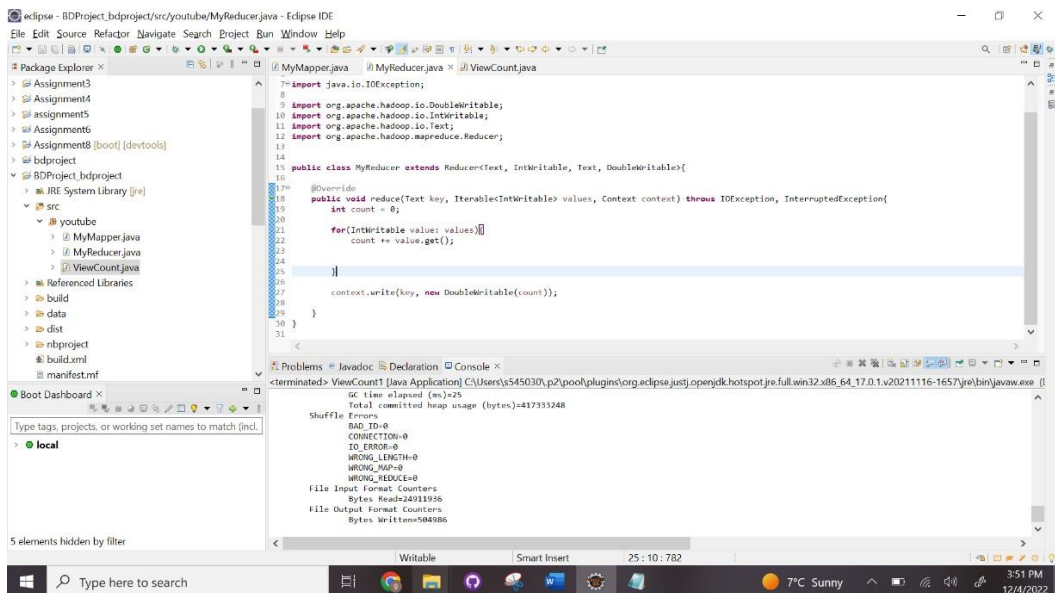
The goal is to know the total views on YouTube on a particular day.

We got the desired output for this goal by applying map reduce techniques on date row and views row, we have written mapper and reducer classes in such a way that it should display the total number of views received by all videos in the youtube on a particular day. The time taken to achieve this goal is 15 minutes.

Mapper

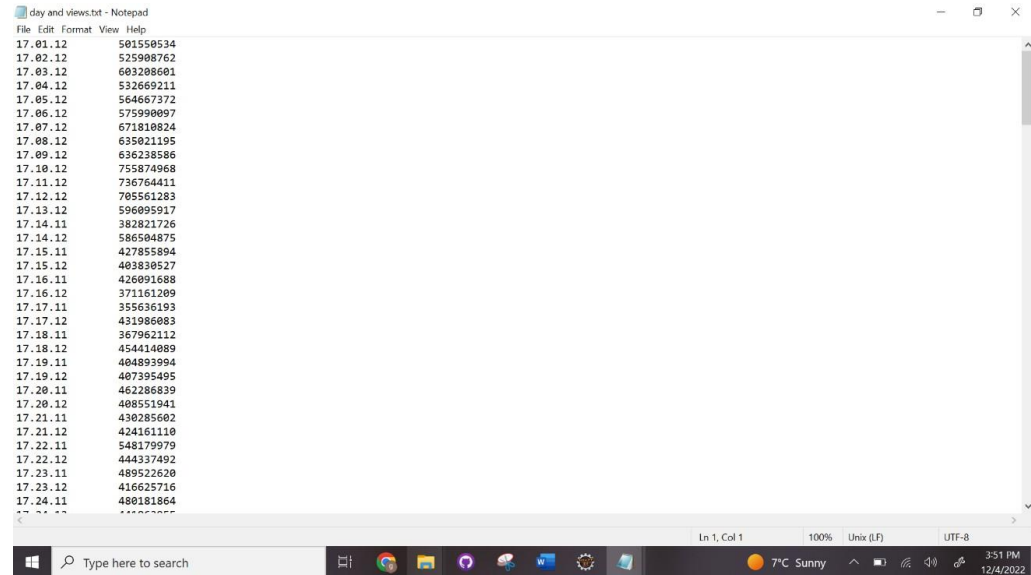


Reducer



1

Output:



day and views.txt - Notepad

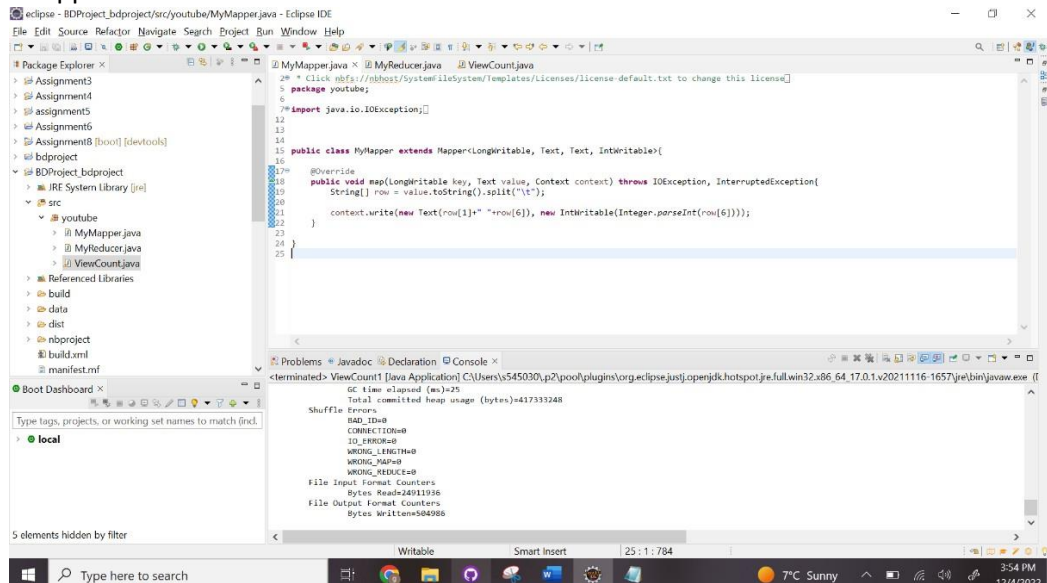
```
File Edit Format View Help
17.01.12 501550534
17.02.12 525908762
17.03.12 603208601
17.04.12 532669211
17.05.12 564667372
17.06.12 575990897
17.07.12 671810824
17.08.12 635021195
17.09.12 636238586
17.10.12 755874968
17.11.12 736764411
17.12.12 705561283
17.13.12 596095917
17.14.11 382821726
17.14.12 586504875
17.15.11 427855894
17.15.12 403830527
17.16.11 426091688
17.16.12 371161209
17.17.11 355636193
17.17.12 431906083
17.18.11 367962112
17.18.12 454414089
17.19.11 404893994
17.19.12 407395495
17.20.11 462208639
17.20.12 408551941
17.21.11 430285602
17.21.12 424161110
17.22.11 548179979
17.22.12 444337492
17.23.11 489522620
17.23.12 416625716
17.24.11 480181864
17.24.12 480181864
```

7.10 Goal 10

The goal is to know the total views a channel got on a particular day.

We got the desired output for this goal by applying map reduce techniques on channel row, date row and views row, we have written mapper and reducer classes in such a way that it should display the total number of views received by all videos in the youtube for each channel. The time taken to achieve this goal is 15 minutes.

Mapper



eclipse - BDProject_bdproject/src/youtube/MyMapper.java - Eclipse IDE

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
  Assignment3
  Assignment4
  Assignment5
  Assignment6
  Assignment8 [boot] [devtools]
  bdproject
    BDProject_bdproject
      JRE System Library [jre]
      src
        youtube
          MyMapper.java
          MyReducer.java
          ViewCount.java
      Referenced Libraries
      build
      data
      dist
      nbproject
      build.xml
      manifest.mf
Boot Dashboard
  Type tags, projects, or working set names to match (incl. local)
  5 elements hidden by filter
  Writable Smart Insert 25:1:784
  <terminated> ViewCount [Java Application] C:\Users\y545030\p2\pool\plugins\org.eclipse.justi.openjdk hotspot.jre.full.win32.x86_64.17.0.1.v20211116-1657\jre\bin\java.exe (f
  GC User elapsed (ms): 35
  Total committed heap usage (bytes): 417333248
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    ID_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_RELECT=0
  File Input Format Counters
    Bytes Read=24011936
  File Output Format Counters
    Bytes Written=504986
```

Reducer:

```

7: import java.io.IOException;
8:
9: import org.apache.hadoop.io.DoubleWritable;
10: import org.apache.hadoop.io.IntWritable;
11: import org.apache.hadoop.io.Text;
12: import org.apache.hadoop.mapreduce.Reducer;
13:
14:
15: public class MyReducer extends Reducer<Text, IntWritable, Text, DoubleWritable>{
16:
17:
18:     @Override
19:     public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
20:         int count = 0;
21:         for(IntWritable value: values){
22:             count += value.get();
23:         }
24:         context.write(key, new DoubleWritable(count));
25:     }
26: }
27:
28:
29:
30:
31:

```

Shuffle Errors
BAD_ID=0
CHECKSUM=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=24911936
File Output Format Counters
Bytes Written=504986

Output:

```

17.01.12 Billboard News 3817.0
17.01.12 BiscottiTV 1183970.0
17.01.12 Bollywood Spy 103489.0
17.01.12 BookMyTV 702133.0
17.01.12 Bordando Con Estilo 6825.0
17.01.12 Brijnaari Sumi 112456.0
17.01.12 Brothers sports 353066.0
17.01.12 Brytiago Cartel Records 336153.0
17.01.12 Business Insider 60509.0
17.01.12 Butch Hartman 145560.0
17.01.12 BuzzFeed Nifty 20879.0
17.01.12 BuzzFeedVideo 819692.0
17.01.12 CABALLERO CH 87543.0
17.01.12 CFL Insider 51053.0
17.01.12 CNCO Noticias 9765.0
17.01.12 CNN 169022.0
17.01.12 CS Dojo 169915.0
17.01.12 CYNVEVO 28983.0
17.01.12 Caja de Peliculas 40290.0
17.01.12 CalumScottVEVO 1728219.0
17.01.12 Cam'ron 913450.0
17.01.12 CanalOficial Mhovidente 22975.0
17.01.12 Carnatavv T 563636.0
17.01.12 CaseyWeistat 2710474.0
17.01.12 Caso Cerrado 129599.0
17.01.12 Casually Explained 2836959.0
17.01.12 ChiChigos 5362.0
17.01.12 Chile TV 14389.0
17.01.12 Chriselle Lim 34187.0
17.01.12 CinemaSins 1614392.0
17.01.12 Cinemakaryam 78781.0
17.01.12 Claudinea Santos 14393.0
17.01.12 Clevver News 201689.0
17.01.12 CollegeHumor 757408.0

```

8 Conclusion

This project teaches you how to leverage the power of YouTube Data and draw crucial insights into the performance of YouTube channels of four countries(India, Japan, Mexico, USA). We have achieved the goals that we set for the project using the mentioned technologies. Carrying out an in-depth analysis of the channels requires you to integrate data from a diverse set of data sources. Integrating such diverse data can be challenging and we have used map reduce techniques and data bricks to overcome this challenge and we have got the desired output from the goals we set as expected in a stipulated time

2

9 Github Repository

<https://github.com/Saiv0711/Bigdata-Project>