INFO 7250 – Engineering Big-Data Systems

Report On

BIG DATA ANALYSIS ON YOUTUBE VIDEO

Name: Kimaya Khilare

NUID: 002958773

SUMMARY OF THE ANALYSIS:

Dataset Description

The most popular videos on YouTube, the well-known video-sharing website, are kept on a list. YouTube employs a combination of metrics, including analyzing users' interactions, to select the year's top-trending videos (number of views, shares, comments, and likes).

The dataset is collected from Simon Fraser University, British Columbia and the dataset has been extracted from YouTube API which contains all the meta-data.

I will be performing analysis using Hadoop (Map Reduce), Pig, and Hive.

Dataset Link: - https://netsg.cs.sfu.ca/youtubedata/

1)video ID:	a unique 11-digit string
2)uploader:	a string of the video uploader's username
3) age:	an integer number of days between the date when the video was uploaded and Feb.15, 2007 (YouTube's establishment)
4)category:	a string of the video category chosen by the uploader
5)length:	integer number of the video length
6)views:	integer number of the views
7)rate:	float number of the video rate
8)ratings:	integer number of the ratings
9)comments:	integer number of the comments
10)related IDs:	up to 20 strings of the related video Id

List of Analysis Performed:

Map Reduce:

1) Calculating the Minimum comment count of each category
2) View Summarization
3) Top 15 YouTube Average Video Rating

Pig:

1) Calculating the Least 5 Categories of YouTube video
2) Top 15 Viewed by category
3) Bad 5 Rated YouTube Video

Hive:

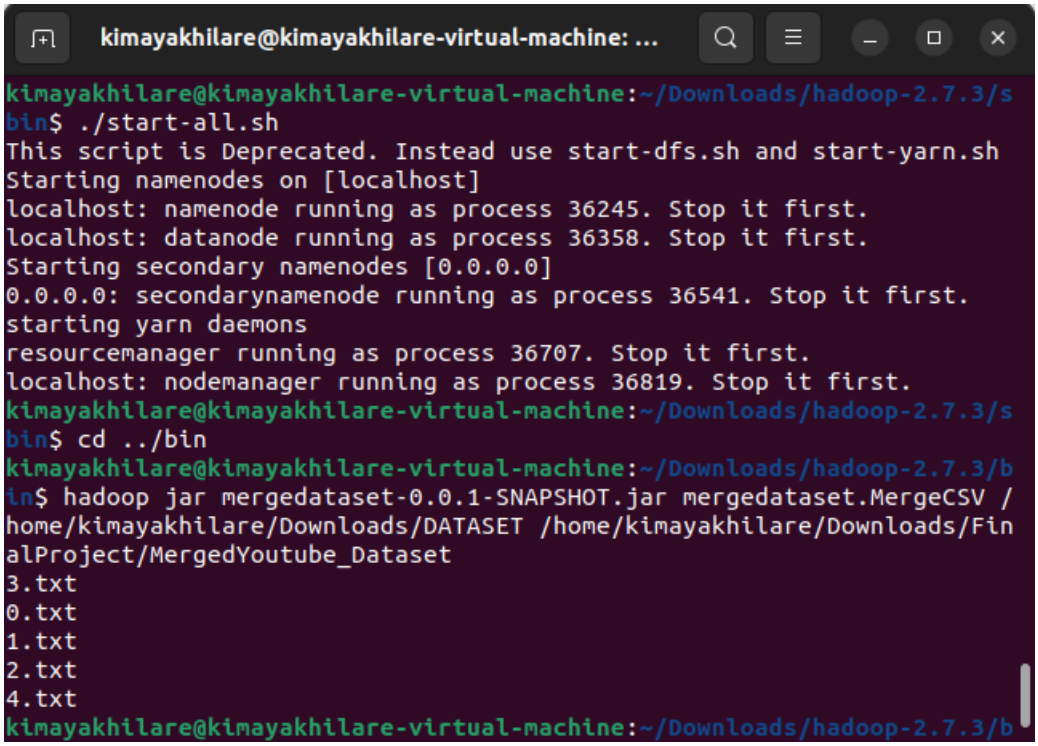
1) Calculating the Top 10 Categories of YouTube videos
2) Calculate the Top 15 lengthy Video
3) Calculate the Top 10 channels with the maximum number of comments

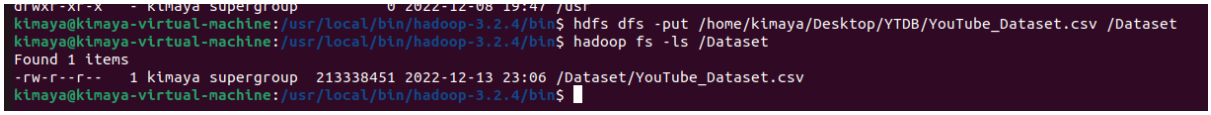
Merging the Dataset

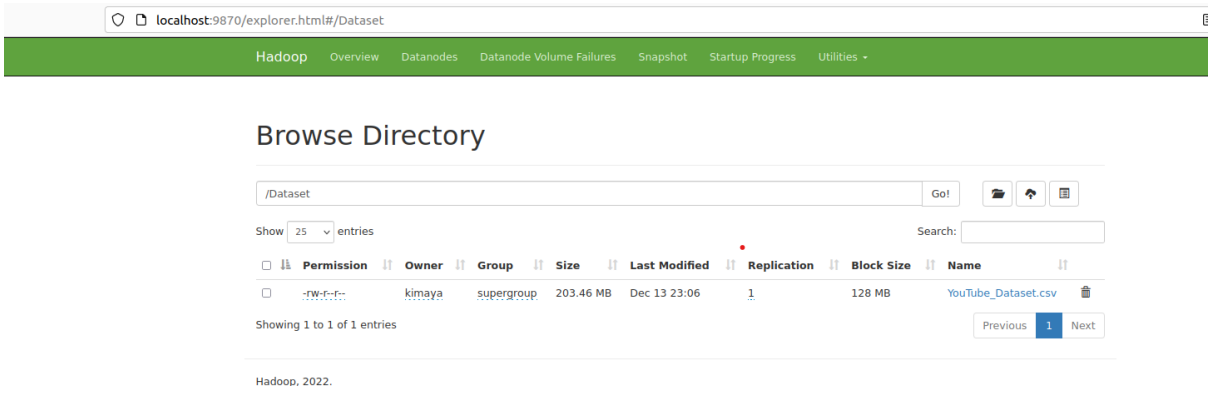
Top_VideoLength_Driver.java × Top_VideoLength_Mapper.java × Top_VideoLength_Reducer.java × MergeCSV.java ×

Source History

```
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;

public class MergeCSV {

        public static void main(String[] args) throws IOException {

                Configuration conf = new Configuration();
                conf.addResource(new Path( pathString: "/home/kimayakhilare/Downloads/hadoop-2.7.3/etc/hadoop/core-site.xml"));
                conf.addResource(new Path( pathString: "/home/kimayakhilare/Downloads/hadoop-2.7.3/etc/hadoop/hdfs-site.xml"));
                FileSystem hdfs = FileSystem.get(conf);
                FileSystem local = FileSystem.getLocal(conf);
                Path inputDir = new Path(args[0]);
                Path hdfsFile = new Path(args[1]);

                try {
                        FileStatus[] inputFiles = local.listStatus( path: inputDir);
                        FSDataOutputStream out = hdfs.create( f: hdfsFile);
                        for(int i = 0; i < inputFiles.length; i++) {
                                System.out.println( x: inputFiles[i].getPath().getName());
                                FSDataInputStream in = local.open( f: inputFiles[i].getPath());
                                byte buffer[] = new byte[256];

                                int bytesRead = 0;
                                while((bytesRead = in.read( b: buffer)) > 0) {
                                        out.write( b: buffer,  off: 0,  len: bytesRead);
                                }
                                in.close();

                        }
                out.close();
                }
                catch (IOException e) {
                        e.printStackTrace();
                }
        }
```


kimayakhilare@kimayakhilare-virtual-machine: ...
kimayakhilare@kimayakhilare-virtual-machine:~/Downloads/hadoop-2.7.3/sbin$ ./start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: namenode running as process 36245. Stop it first.
localhost: datanode running as process 36358. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 36541. Stop it first.
starting yarn daemons
resourcemanager running as process 36707. Stop it first.
localhost: nodemanager running as process 36819. Stop it first.
kimayakhilare@kimayakhilare-virtual-machine:~/Downloads/hadoop-2.7.3/sbin$ cd ../bin
kimayakhilare@kimayakhilare-virtual-machine:~/Downloads/hadoop-2.7.3/bin$ hadoop jar mergedataset-0.0.1-SNAPSHOT.jar mergedataset.MergeCSV /home/kimayakhilare/Downloads/DATASET /home/kimayakhilare/Downloads/FinalProject/MergedYoutube_Dataset
3.txt
0.txt
1.txt
2.txt
4.txt
kimayakhilare@kimayakhilare-virtual-machine:~/Downloads/hadoop-2.7.3/b


kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hdfs dfs -put /home/kimaya/Desktop/YTDB/YouTube_Dataset.csv /Dataset
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop fs -ls /Dataset
Found 1 items
-rw-r--r--   1 kimaya supergroup 213338451 2022-12-13 23:06 /Dataset/YouTube_Dataset.csv
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$

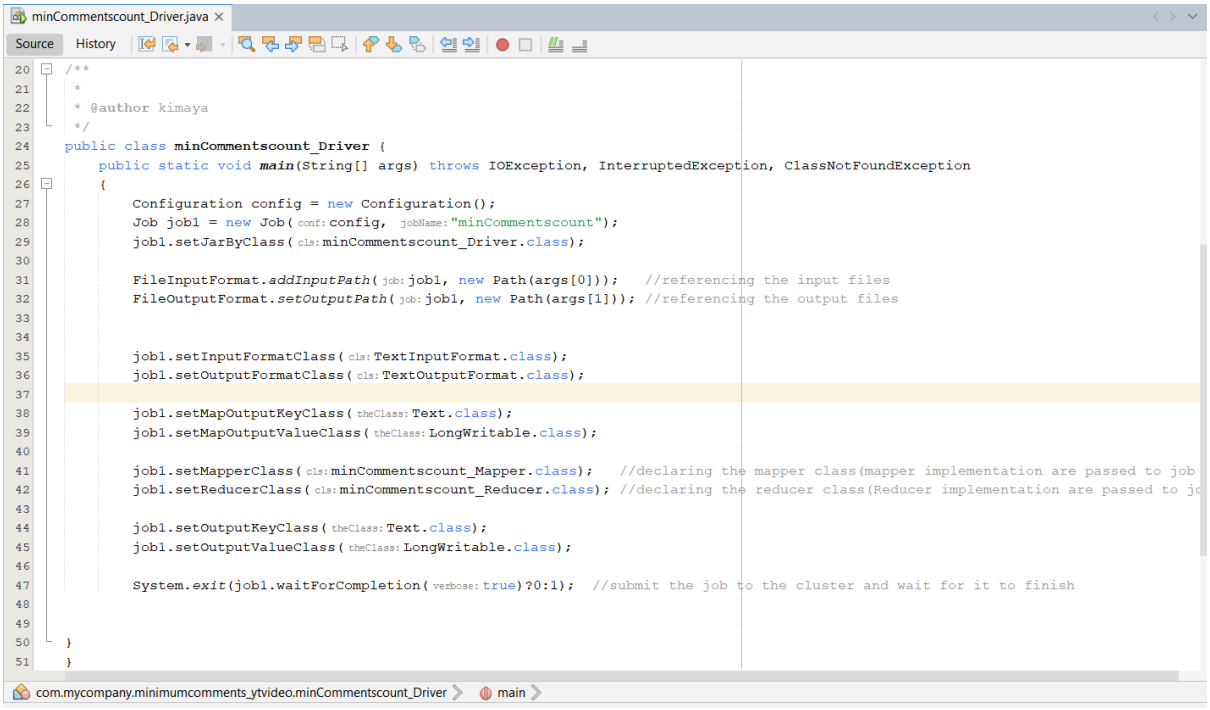

localhost:9870/explorer.html#/Dataset
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directory
/Dataset
Go!
Show 25 entries
Search:
Permission
Owner
Group
Size
Last Modified
Replication
Block Size
Name
-rw-r--r--
kimaya
supergroup
203.46 MB
Dec 13 23:06
1
128 MB
YouTube_Dataset.csv
Showing 1 to 1 of 1 entries
Previous
1
Next
Hadoop, 2022.

Map Reduce

1) Minimum Comment Count by each category of YouTube Video

Here, I am analyzing each category of YouTube videos by calculating the minimum comments of each category.

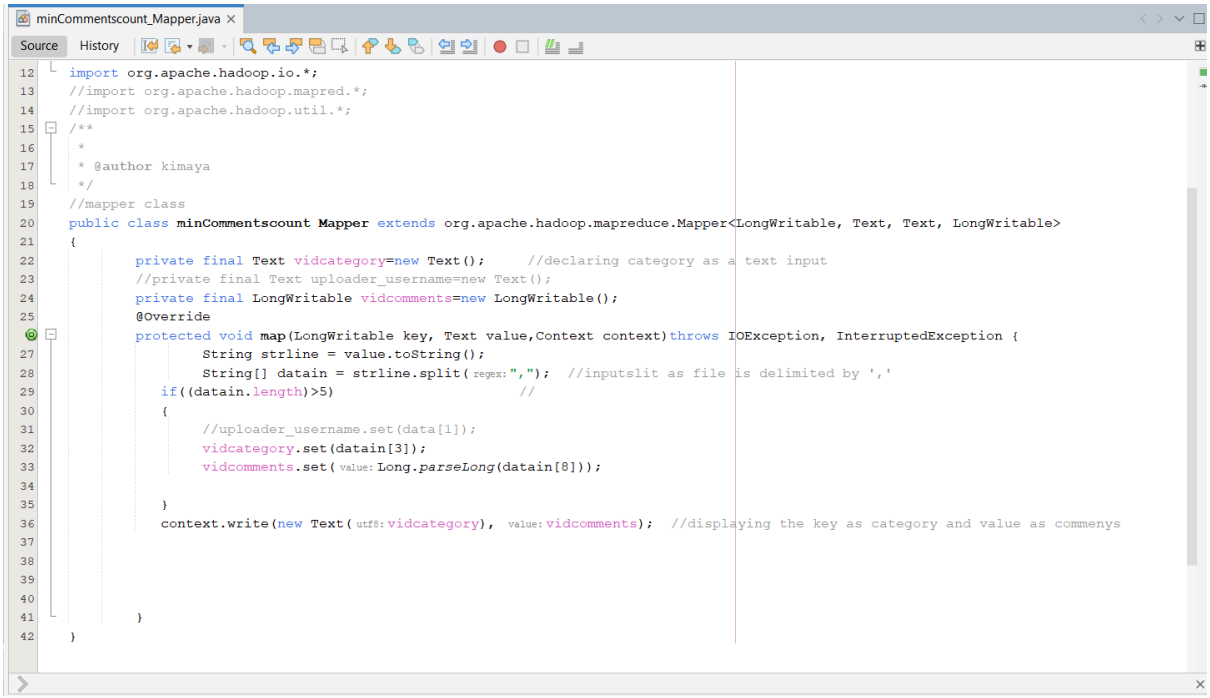
Driver class



```
/**
 *
 * @author kimaya
 */
public class minCommentscount_Driver {
    public static void main(String[] args) throws IOException, InterruptedException, ClassNotFoundException
    {
        Configuration config = new Configuration();
        Job job1 = new Job( conf: config,  jobName: "minCommentscount");
        job1.setJarByClass( cls: minCommentscount_Driver.class);

        FileInputFormat.addInputPath( job: job1, new Path(args[0]));   //referencing the input files
        FileOutputFormat.setOutputPath( job: job1, new Path(args[1])); //referencing the output files


        job1.setInputFormatClass( cls: TextInputFormat.class);
        job1.setOutputFormatClass( cls: TextOutputFormat.class);

        job1.setMapOutputKeyClass( theClass: Text.class);
        job1.setMapOutputValueClass( theClass: LongWritable.class);

        job1.setMapperClass( cls: minCommentscount_Mapper.class);   //declaring the mapper class(mapper implementation are passed to job
        job1.setReducerClass( cls: minCommentscount_Reducer.class); //declaring the reducer class(Reducer implementation are passed to jo

        job1.setOutputKeyClass( theClass: Text.class);
        job1.setOutputValueClass( theClass: LongWritable.class);

        System.exit(job1.waitForCompletion( verbose: true)?0:1);  //submit the job to the cluster and wait for it to finish


    }
    }
```

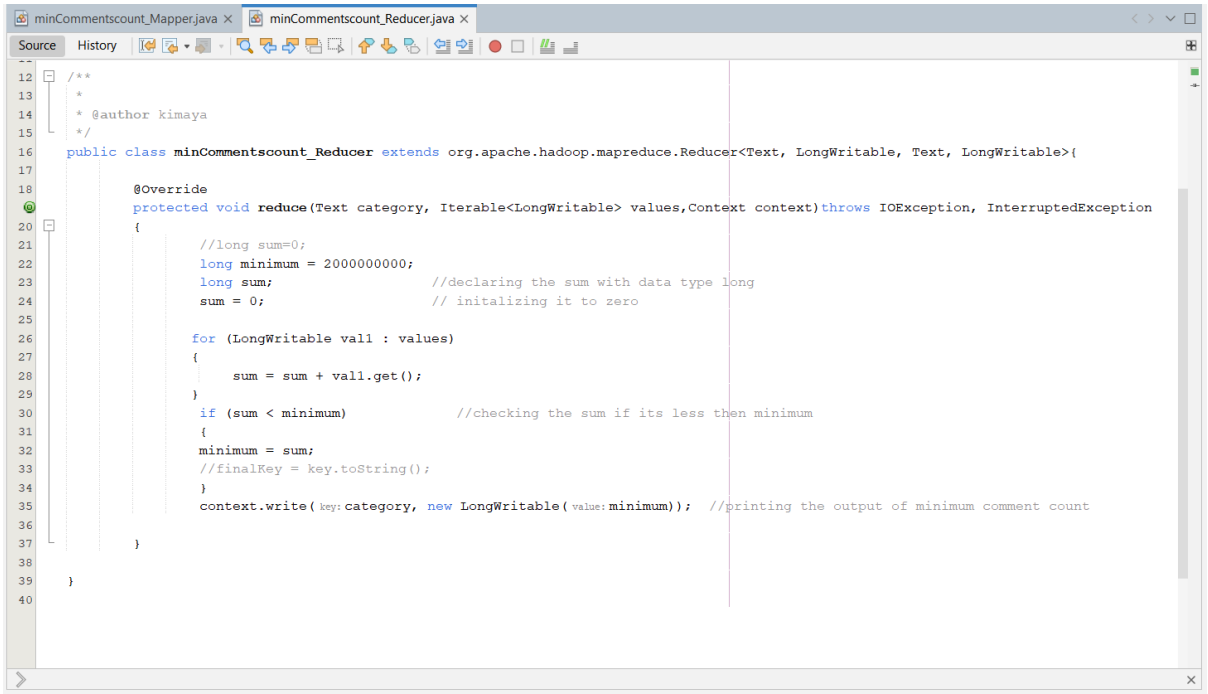
com.mycompany.minimumcomments_ytvideo.minCommentscount_Driver > main

Mapper class



```
import org.apache.hadoop.io.*;
//import org.apache.hadoop.mapred.*;
//import org.apache.hadoop.util.*;
/**
 *
 * @author kimaya
 */
//mapper class
public class minCommentscount Mapper extends org.apache.hadoop.mapreduce.Mapper<LongWritable, Text, Text, LongWritable>
{
        private final Text vidcategory=new Text();     //declaring category as a text input
        //private final Text uploader_username=new Text();
        private final LongWritable vidcomments=new LongWritable();
        @Override
        protected void map(LongWritable key, Text value,Context context)throws IOException, InterruptedException {
                String strline = value.toString();
                String[] datain = strline.split( regex: ",");  //inputslit as file is delimited by ','
           if((datain.length)>5)                        //
           {
                //uploader_username.set(data[1]);
                vidcategory.set(datain[3]);
                vidcomments.set( value: Long.parseLong(datain[8]));

           }
           context.write(new Text( utf8: vidcategory),  value: vidcomments);  //displaying the key as category and value as commenys



        }
}
```

Reducer class




```java
/**
 *
 * @author kimaya
 */
public class minCommentscount_Reducer extends org.apache.hadoop.mapreduce.Reducer<Text, LongWritable, Text, LongWritable>{

        @Override
        protected void reduce(Text category, Iterable<LongWritable> values,Context context)throws IOException, InterruptedException
        {
                //long sum=0;
                long minimum = 2000000000;
                long sum;                   //declaring the sum with data type long
                sum = 0;                    // initalizing it to zero

               for (LongWritable val1 : values)
               {
                    sum = sum + val1.get();
               }
                if (sum < minimum)              //checking the sum if its less then minimum
                {
                minimum = sum;
                //finalKey = key.toString();
                }
                context.write( key: category, new LongWritable( value: minimum));  //printing the output of minimum comment count

        }

}
```

Executing the hdfs comment with jar file

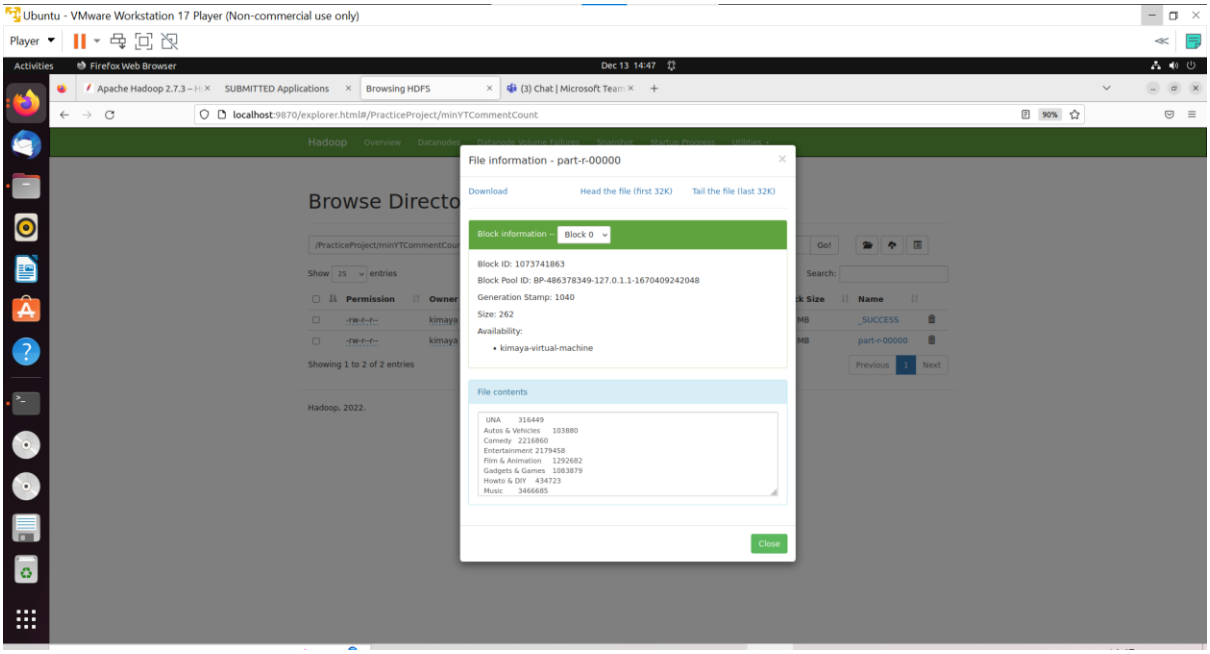
```
kimaya@kimaya-virtual-machine: /usr/local/bin/hadoop-3.2.1/bin$ hadoop jar MinimumComments_YtVideo-1.0-SNAPSHOT.jar com.mycompany.minimumcomments_ytvideo.minCommentscount_Driver /Dataset/YouTube_Dataset.csv /FinalProject/minimumYTCommentCount
```

```
    File Output Format Counters
            Bytes Written=262
2-12-13 14:46:45,931 INFO mapred.LocalJobRunner: Finishing task: attempt_local1108297537_0001_r_000000_0
2-12-13 14:46:45,931 INFO mapred.LocalJobRunner: reduce task executor complete.
2-12-13 14:46:46,457 INFO mapreduce.Job:  map 100% reduce 100%
2-12-13 14:46:46,457 INFO mapreduce.Job: Job job_local1108297537_0001 completed successfully
2-12-13 14:46:46,476 INFO mapreduce.Job: Counters: 36
    File System Counters
            FILE: Number of bytes read=31835118
            FILE: Number of bytes written=59396195
            FILE: Number of read operations=0
            FILE: Number of large read operations=0
            FILE: Number of write operations=0
            HDFS: Number of bytes read=560906918
            HDFS: Number of bytes written=262
            HDFS: Number of read operations=24
            HDFS: Number of large read operations=0
            HDFS: Number of write operations=5
            HDFS: Number of bytes read erasure-coded=0
    Map-Reduce Framework
            Map input records=749361
            Map output records=749361
            Map output bytes=14409802
            Map output materialized bytes=15908536
            Input split bytes=224
            Combine input records=0
            Combine output records=0
            Reduce input groups=13
            Reduce shuffle bytes=15908536
            Reduce input records=749361
            Reduce output records=13
            Spilled Records=1498722
            Shuffled Maps =2
            Failed Shuffles=0
            Merged Map outputs=2
            GC time elapsed (ms)=1171
            Total committed heap usage (bytes)=2580545536
    Shuffle Errors
            BAD_ID=0
            CONNECTION=0
            IO_ERROR=0
```

Output displaying the minimum comment count with each category

```
            Bytes Written=262
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop fs -cat /PracticeProject/minYTCommentCount/part-r-00000 | head
 UNA    316449
Autos & Vehicles        103880
Comedy  2216860
Entertainment   2179458
Film & Animation        1292682
Gadgets & Games 1083879
Howto & DIY     434723
Music   3466685
News & Politics 1117201
People & Blogs  1003874
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$
```

Output in Hdfs:


Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Firefox Web Browser
Dec 13 14:47
Apache Hadoop 2.7.3 – H
SUBMITTED Applications
Browsing HDFS
(3) Chat | Microsoft Team
localhost:9870/explorer.html#/PracticeProject/minYTCommentCount
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities
Browse Directo
/PracticeProject/minYTCommentCou
Go!
Show 25 entries
Search:
Permission
Owner
Name
-rw-r--r--
kimaya
_SUCCESS
part-r-00000
Previous
1
Next
Showing 1 to 2 of 2 entries
Hadoop, 2022.
File information - part-r-00000
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information --
Block 0
Block ID: 1073741863
Block Pool ID: BP-486378349-127.0.1.1-1670409242048
Generation Stamp: 1040
Size: 262
Availability:
kimaya-virtual-machine
File contents
UNA 316449
Autos & Vehicles 103880
Comedy 2216860
Entertainment 2179458
Film & Animation 1292682
Gadgets & Games 1083879
Howto & DIY 434723
Music 3466685
Close

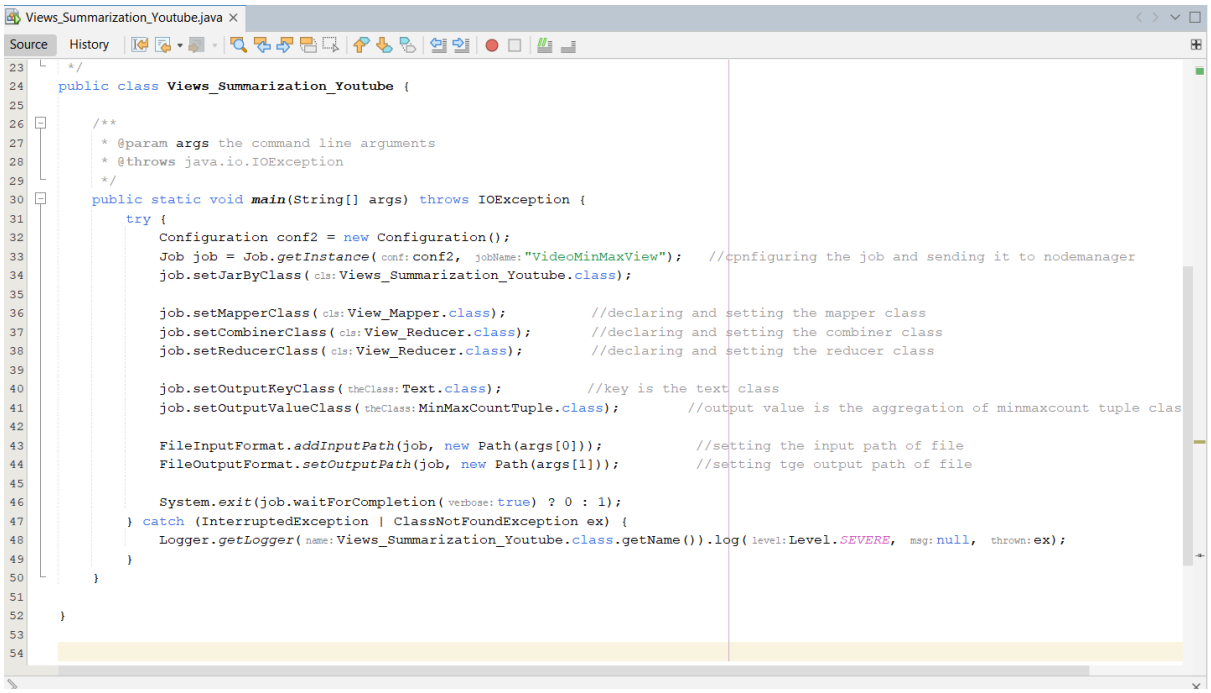
2) Summarization of YouTube Video Views

(It provides the summarization of views, rates, and comments)

This analysis gives the summarization of Views with rate and comments

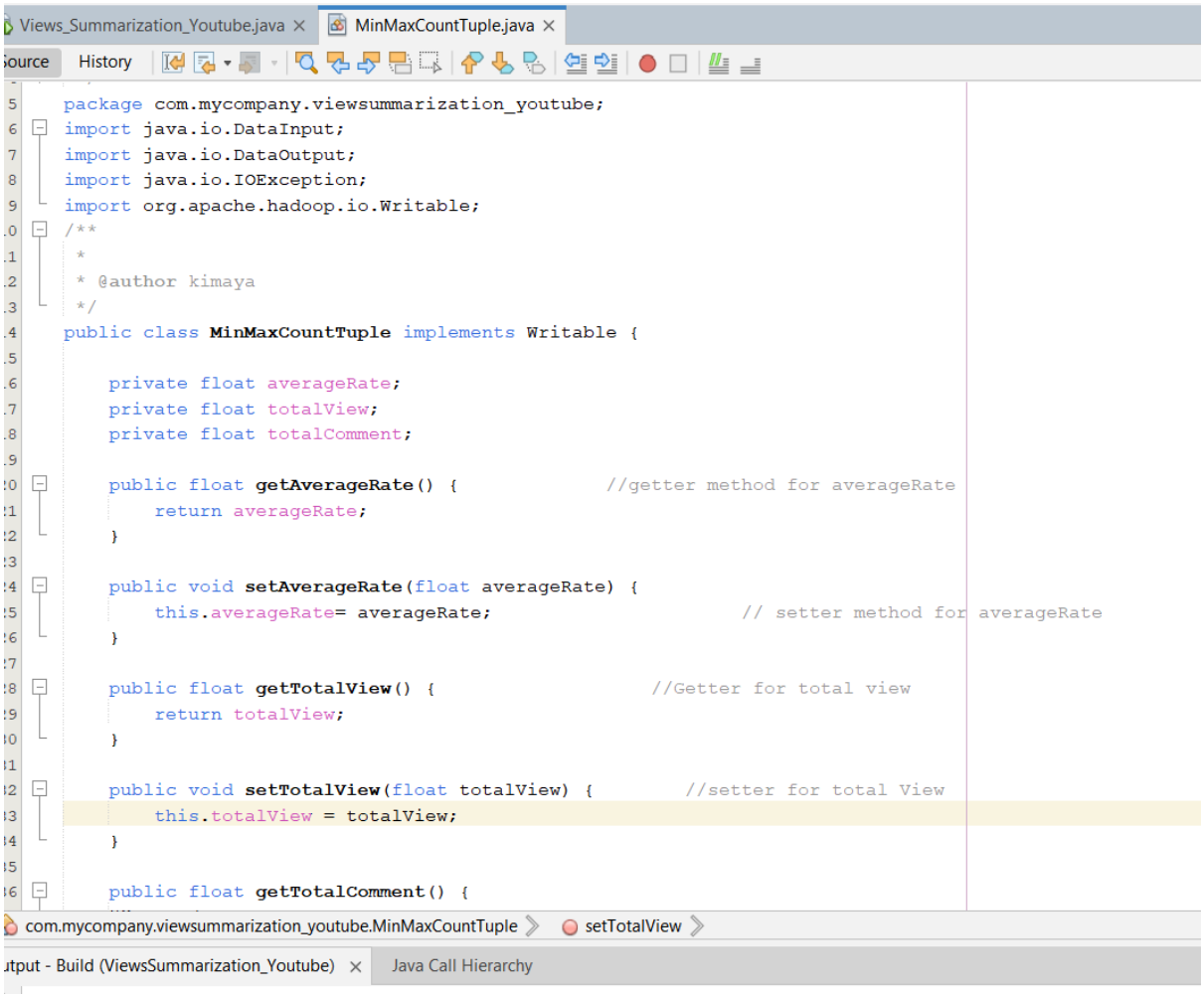
Here I also used the Custom Writable class to calculate the minimum and maximum count of views.

Driver class

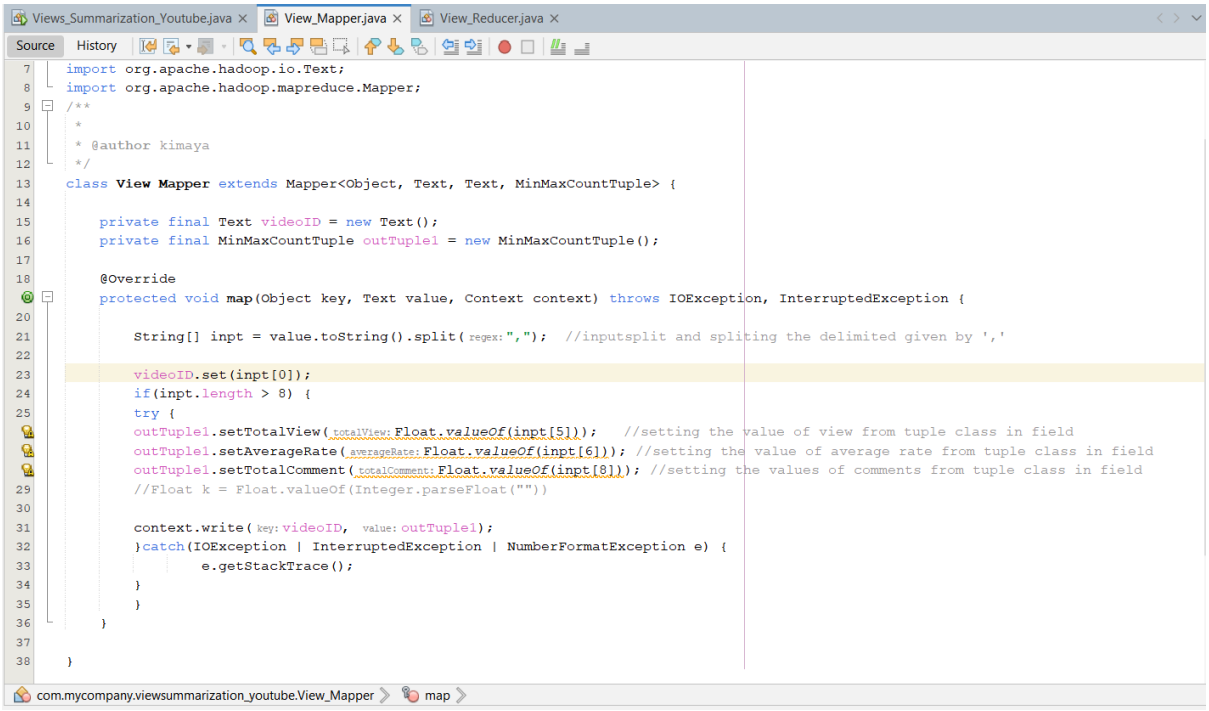


```
Views_Summarization_Youtube.java
Source  History
 */
public class Views_Summarization_Youtube {

    /**
     * @param args the command line arguments
     * @throws java.io.IOException
     */
    public static void main(String[] args) throws IOException {
        try {
            Configuration conf2 = new Configuration();
            Job job = Job.getInstance( conf: conf2,  jobName: "VideoMinMaxView");   //cpnfiguring the job and sending it to nodemanager
            job.setJarByClass( cls: Views_Summarization_Youtube.class);

            job.setMapperClass( cls: View_Mapper.class);          //declaring and setting the mapper class
            job.setCombinerClass( cls: View_Reducer.class);       //declaring and setting the combiner class
            job.setReducerClass( cls: View_Reducer.class);        //declaring and setting the reducer class

            job.setOutputKeyClass( theClass: Text.class);         //key is the text class
            job.setOutputValueClass( theClass: MinMaxCountTuple.class);        //output value is the aggregation of minmaxcount tuple clas

            FileInputFormat.addInputPath(job, new Path(args[0]));           //setting the input path of file
            FileOutputFormat.setOutputPath(job, new Path(args[1]));         //setting tge output path of file

            System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
        } catch (InterruptedException | ClassNotFoundException ex) {
            Logger.getLogger( name: Views_Summarization_Youtube.class.getName()).log( level: Level.SEVERE,  msg: null,  thrown: ex);
        }
    }

}
```

Tuple class


Views_Summarization_Youtube.java
MinMaxCountTuple.java
Source
History
package com.mycompany.viewsummarization_youtube;
import java.io.DataInput;
import java.io.DataOutput;
import java.io.IOException;
import org.apache.hadoop.io.Writable;
/**
*
* @author kimaya
*/
public class MinMaxCountTuple implements Writable {
private float averageRate;
private float totalView;
private float totalComment;
public float getAverageRate() { //getter method for averageRate
return averageRate;
}
public void setAverageRate(float averageRate) {
this.averageRate= averageRate; // setter method for averageRate
}
public float getTotalView() { //Getter for total view
return totalView;
}
public void setTotalView(float totalView) { //setter for total View
this.totalView = totalView;
}
public float getTotalComment() {
com.mycompany.viewsummarization_youtube.MinMaxCountTuple
setTotalView
Output - Build (ViewsSummarization_Youtube)
Java Call Hierarchy

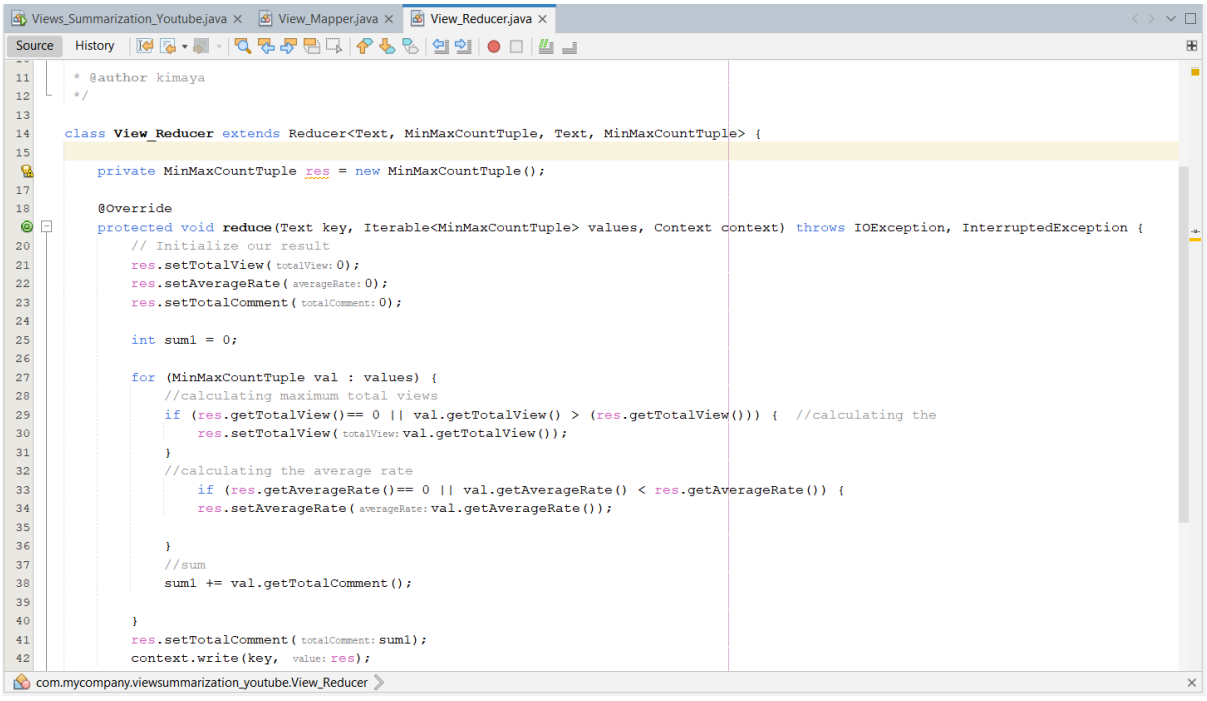
Mapper class



```
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
/**
 *
 * @author kimaya
 */
class View Mapper extends Mapper<Object, Text, Text, MinMaxCountTuple> {

    private final Text videoID = new Text();
    private final MinMaxCountTuple outTuple1 = new MinMaxCountTuple();

    @Override
    protected void map(Object key, Text value, Context context) throws IOException, InterruptedException {

        String[] inpt = value.toString().split( regex: ",");  //inputsplit and spliting the delimited given by ','

        videoID.set(inpt[0]);
        if(inpt.length > 8) {
        try {
        outTuple1.setTotalView( totalView: Float.valueOf(inpt[5]));   //setting the value of view from tuple class in field
        outTuple1.setAverageRate( averageRate: Float.valueOf(inpt[6])); //setting the value of average rate from tuple class in field
        outTuple1.setTotalComment( totalComment: Float.valueOf(inpt[8])); //setting the values of comments from tuple class in field
        //Float k = Float.valueOf(Integer.parseFloat(""))

        context.write( key: videoID,  value: outTuple1);
        }catch(IOException | InterruptedException | NumberFormatException e) {
                e.getStackTrace();
        }
        }
    }

}
```

com.mycompany.viewsummarization_youtube.View_Mapper > map

Reducer class



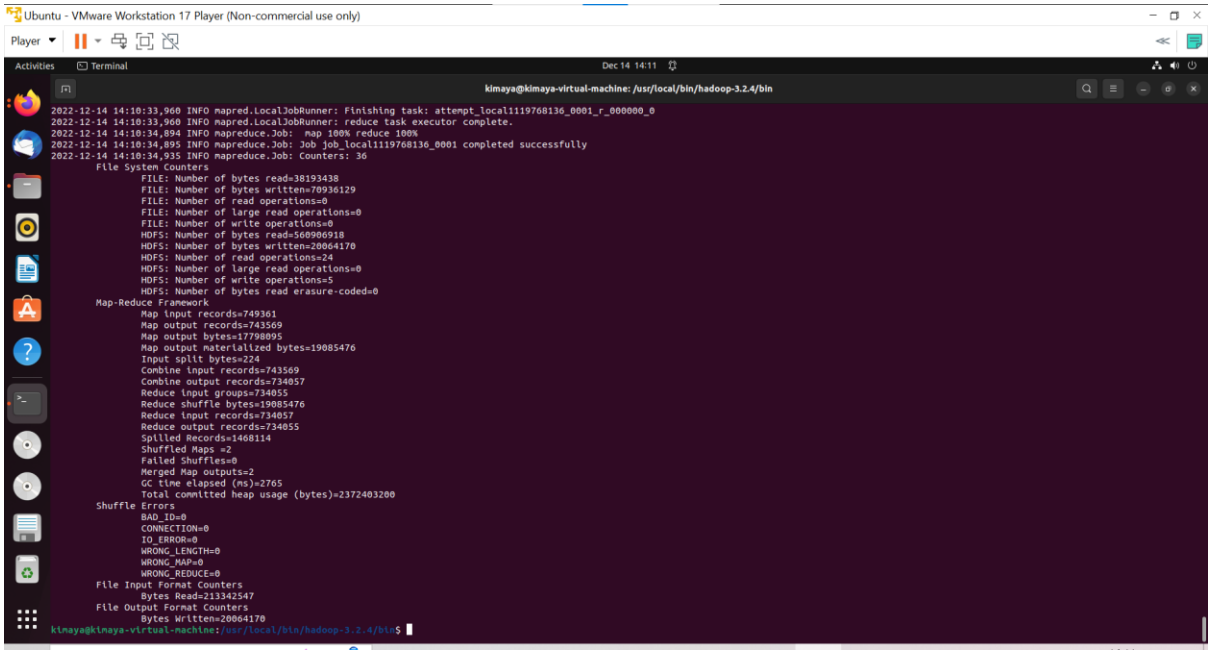
```
 * @author kimaya
 */

class View_Reducer extends Reducer<Text, MinMaxCountTuple, Text, MinMaxCountTuple> {

    private MinMaxCountTuple res = new MinMaxCountTuple();

    @Override
    protected void reduce(Text key, Iterable<MinMaxCountTuple> values, Context context) throws IOException, InterruptedException {
        // Initialize our result
        res.setTotalView( totalView: 0);
        res.setAverageRate( averageRate: 0);
        res.setTotalComment( totalComment: 0);

        int sum1 = 0;

        for (MinMaxCountTuple val : values) {
            //calculating maximum total views
            if (res.getTotalView()== 0 || val.getTotalView() > (res.getTotalView())) {  //calculating the
                res.setTotalView( totalView: val.getTotalView());
            }
            //calculating the average rate
                if (res.getAverageRate()== 0 || val.getAverageRate() < res.getAverageRate()) {
                res.setAverageRate( averageRate: val.getAverageRate());

            }
            //sum
            sum1 += val.getTotalComment();

        }
        res.setTotalComment( totalComment: sum1);
        context.write(key,  value: res);
```

com.mycompany.viewsummarization_youtube.View_Reducer

Hadoop command including the jar file to

```
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop jar ViewsSummarization_Youtube-1.0-SNAPSHOT.jar com.mycompany.viewsummarization_youtube.Views_Summarization_Youtube /Dataset/YouTube_
Dataset.csv /FinalProject/ViewSummarizationYT
```

Executing the file: -



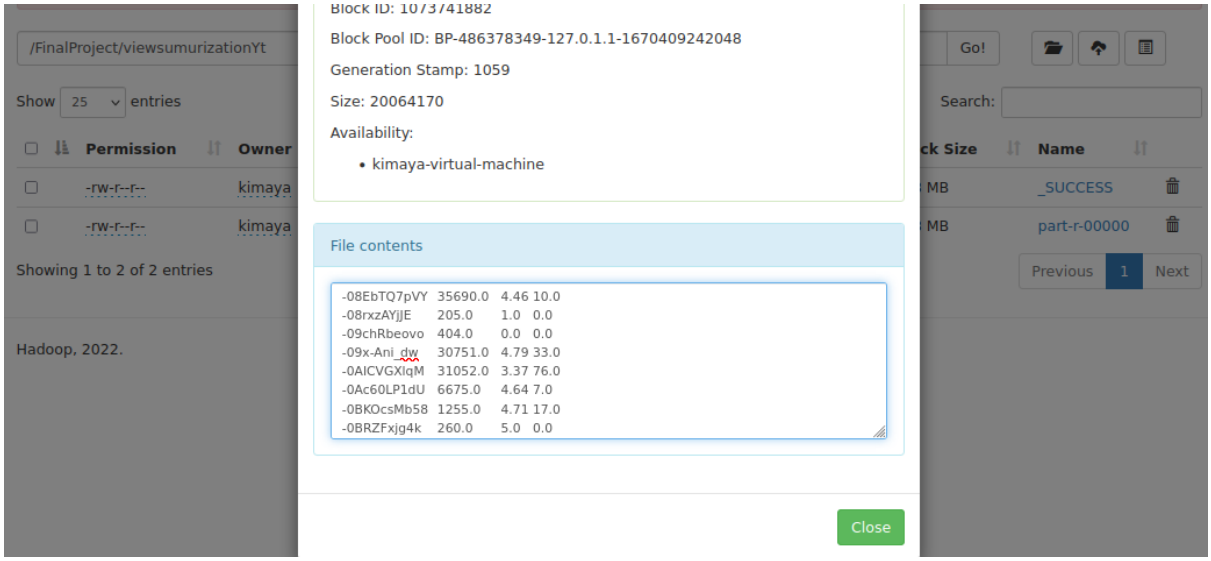
Hadoop fs -cat /finalproject/viewsummarization/part-r-00000 | head

```
--09WIapUUc     154.0   1.0     1.0
--0R69A3CVU     202.0   0.0     0.0
--0VhTcNyzs     2357.0  4.55    6.0
--0eZHhaV08     892.0   1.0     3.0
--0ts5liqos     1923.0  4.38    5.0
--1K0JeTg2I     2010.0  4.44    9.0
--1Li0TO0rU     585.0   0.0     0.0
--1SP0EiUZY     665.0   4.29    2.0
cat: Unable to write to output stream.
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$
```

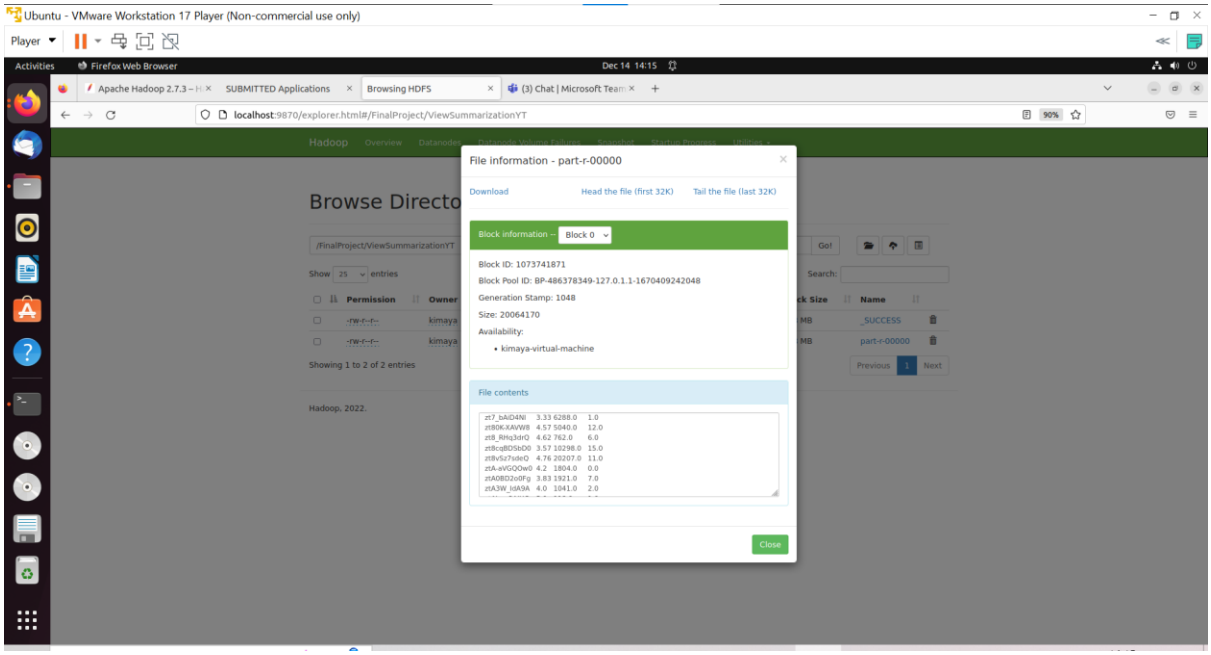
Tail

```
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop fs -cat /FinalProject/viewsumurizationYt/part-r-00000 | tail
zzvDSwOA96s     9155.0  4.95    6.0
zzvLbBkB3Dw     1965.0  5.0     6.0
zzvcX0GO_oU     1044.0  1.76    18.0
zzw0Ocjgl8o     2042.0  4.74    4.0
zzwIk2bKUC8     42.0    0.0     0.0
zzwttYFZAiY     30006.0 4.53    16.0
zzwzh4lwWJI     390.0   2.0     1.0
zzyqUJvwXpQ     17770.0 4.82    58.0
zzzVgNWq0zI     9137.0  4.89    43.0
zzzeAlILG_s     26.0    0.0     0.0
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$
```

Output in hdfs localhost


/FinalProject/viewsumurizationYt
Go!
Show 25 entries
Search:
Permission
Owner
ck Size
Name
-rw-r--r--
kimaya
MB
_SUCCESS
-rw-r--r--
kimaya
MB
part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2022.
Block ID: 1073741882
Block Pool ID: BP-486378349-127.0.1.1-1670409242048
Generation Stamp: 1059
Size: 20064170
Availability:
• kimaya-virtual-machine
File contents
-08EbTQ7pVY 35690.0 4.46 10.0
-08rxzAYjJE 205.0 1.0 0.0
-09chRbeovo 404.0 0.0 0.0
-09x-Ani_dw 30751.0 4.79 33.0
-0AlCVGXlqM 31052.0 3.37 76.0
-0Ac60LP1dU 6675.0 4.64 7.0
-0BKOcsMb58 1255.0 4.71 17.0
-0BRZFxjg4k 260.0 5.0 0.0
Close

Tail


Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Firefox Web Browser
Dec 14 14:15
Apache Hadoop 2.7.3 – H
SUBMITTED Applications
Browsing HDFS
(3) Chat | Microsoft Team
localhost:9870/explorer.html#/FinalProject/ViewSummarizationYT
Hadoop
Overview
Browse Directo
/FinalProject/ViewSummarizationYT
Go!
Show 25 entries
Search:
Permission
Owner
ck Size
Name
-rw-r--r--
kimaya
MB
_SUCCESS
-rw-r--r--
kimaya
MB
part-r-00000
Showing 1 to 2 of 2 entries
Previous
1
Next
Hadoop, 2022.
File information - part-r-00000
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information --
Block 0
Block ID: 1073741871
Block Pool ID: BP-486378349-127.0.1.1-1670409242048
Generation Stamp: 1048
Size: 20064170
Availability:
• kimaya-virtual-machine
File contents
Close

3)YouTube Average Rating by category

This analysis gives the rating by mentioning the Video Id

Driver class

AvgYT_Driver.java

Source | History

```
10  import org.apache.hadoop.io.Text;
11  import org.apache.hadoop.mapreduce.Job;
    import org.apache.hadoop.mapreduce.Mapper;
    import org.apache.hadoop.mapreduce.Reducer;
14  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
15  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
16  /**
17   *
18   * @author kimaya
19   */
20  public class AvgYT_Driver
21  {
22      @SuppressWarnings("deprecation")
23   public static void main(String[] args) throws Exception {
24          Configuration conf = new Configuration();
25
26          Job job = Job.getInstance(conf,  jobName: "Top50");
27          job.setJarByClass( cls: AvgYT_Driver.class);   //declared the driverclass
28          job.setMapperClass( cls: AvgYT_Mapper.class);     //Mapper class declared
29          job.setReducerClass( cls: AvgYT_Reducer.class);   //Reducer class declared
30          job.setOutputKeyClass( theClass: Text.class);
31          job.setOutputValueClass( theClass: FloatWritable.class);
32
33          //Configuring the input/output path from filesystem into job
34          FileInputFormat.addInputPath(job, new Path(args[0]));      //adding the input file
35          FileOutputFormat.setOutputPath(job, new Path(args[1]));    // the output file
36
37
38          System.exit(job.waitForCompletion( verbose: true) ? 0 : 1);
39      }
40  }
```

Mapper class

AvgYT_Driver.java × AvgYT_Mapper.java ×

Source History

```
import org.apache.hadoop.mapreduce.Mapper;
/**
 *
 * @author kimaya
 */

public class AvgYT Mapper extends Mapper<Object, Text, Text, FloatWritable>
{

        private FloatWritable video_rating = new FloatWritable( value: 1);
        private Text video_id = new Text();

        @Override
        public void map(Object key, Text value, Mapper.Context context
        ) throws IOException, InterruptedException {

            String[] fields = value.toString().split( regex: ",");      //splitting the line by ","
            video_id = new Text(fields[0]);
            try {
            if(fields.length > 6)                                       //checking if the row has more than 6 columns
            {                                                           //Mapper class is emitting videoID and rating
            //if (!fields[6].isEmpty()) {
                video_rating = new FloatWritable( value: Float.parseFloat(fields[7]));
            }

            context.write( key: video_id,  value: video_rating);
            }catch (IOException | InterruptedException | NumberFormatException e) {
                //if its greater than 6 then it will throw the exception

}
        }
    }
```

Reducer class

AvgYT_Reducer.java ×

Source History

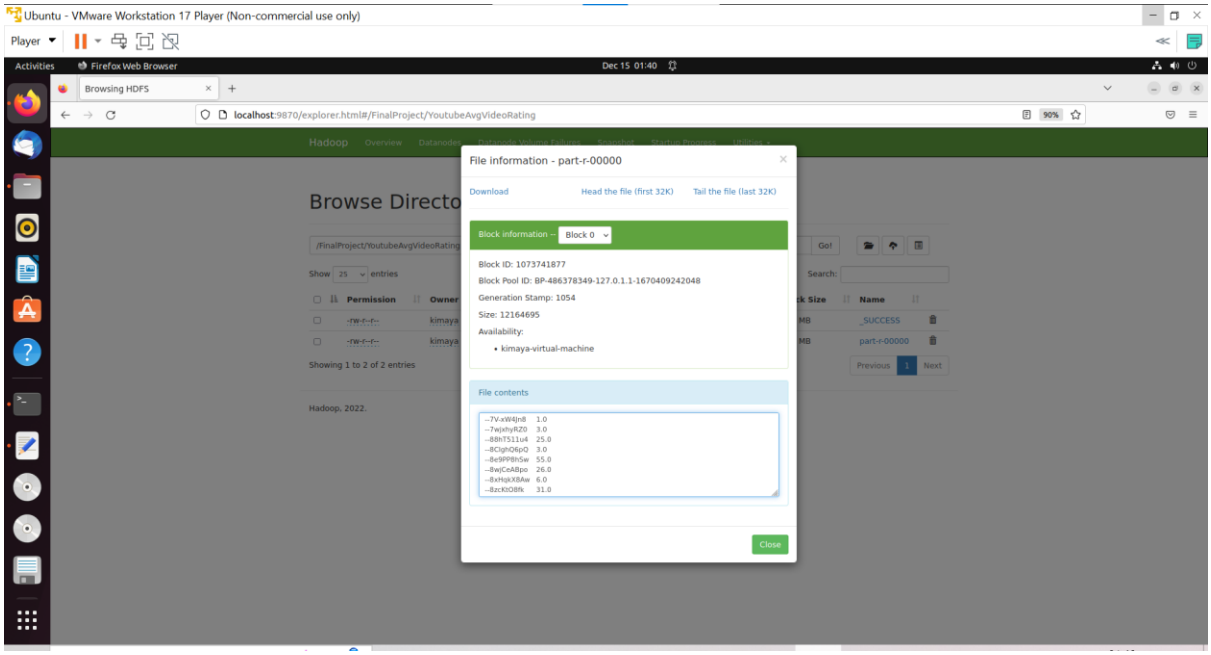
```
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
/**
 *
 * @author kimaya
 */
public class AvgYT_Reducer extends Reducer <Text, FloatWritable,Text, FloatWritable>
{
    private FloatWritable res = new FloatWritable();

    @Override
    public void reduce(Text key, Iterable<FloatWritable> values, Context context) throws IOException, InterruptedException
    {
        int cnt = 0;   //initializing count to 0
        float sum = 0;
        float average = 0;
        // calculates the number of occurance in single word

        for (FloatWritable val : values)
        {
            sum += val.get();
            ++cnt;
        }
        average = sum / cnt;
        res.set( value: average);
        context.write(key,  value: res);
        //context.write(key,new FloatWritable(sum));

    }
}
```

Hadoop command and execute the map reduce

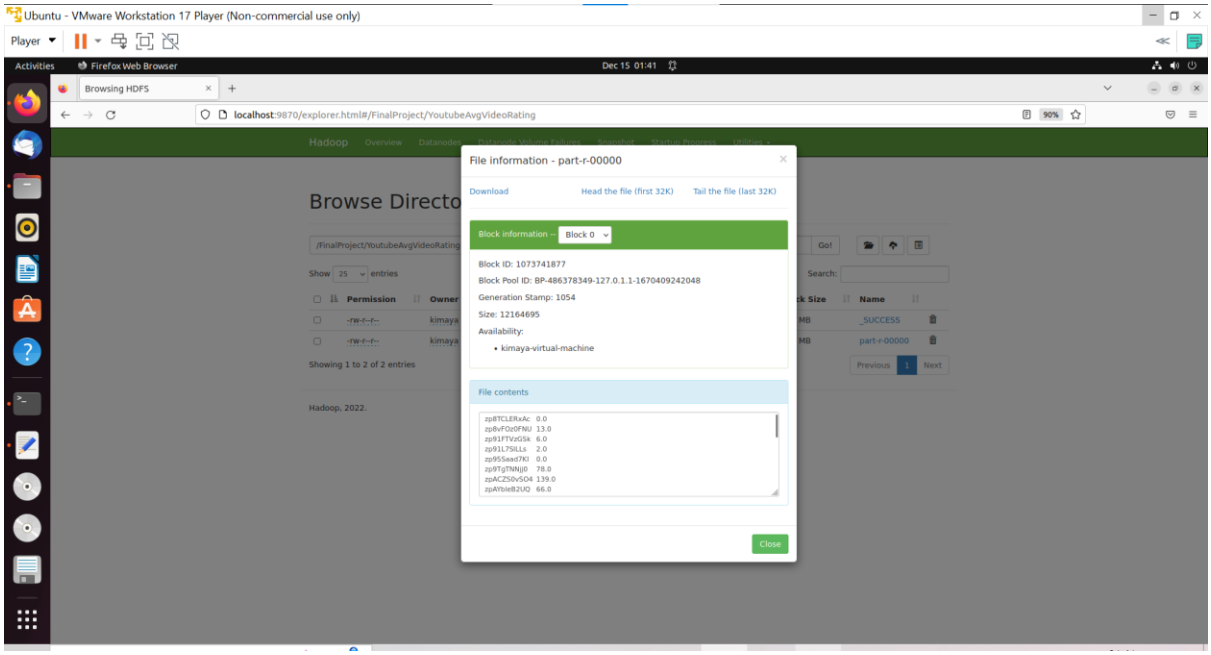
```
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop jar YoutubeVideoRating-1.0-SNAPSHOT.jar com.mycompany.averageyoutubevideorating.AvgYT_Driver /Dataset/YouTube_Dataset.csv /FinalProj
t/YoutubeAvgVideoRating
2022-12-15 01:36:01,499 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
```

```
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$ hadoop fs -cat /FinalProject/YoutubeAvgVideoRating/part-r-00000 |sort -n -k2 -r | head -n15
QjA5faZF1A8     120506.0
dMH0bHeiRNg     87520.0
0XxI-hvPRRA     80710.0
R0049_tDAU8     70972.0
nojWJ6-XmeQ     62265.0
JahdnOQ9XCA     59008.0
VcQIwbvGRKU     46472.0
sdUUx5FdySs     42417.0
pv5zWaTEVkI     42386.0
Ggg0bsUfLBk     42168.0
D2kJZOfq7zk     42162.0
vr3x_RRJdd4     39079.0
aRNzWyD7C9o     36815.0
CQO3K8BcyGM     36460.0
p78oOk7dylQ     36271.0
kimaya@kimaya-virtual-machine:/usr/local/bin/hadoop-3.2.4/bin$
```

Head



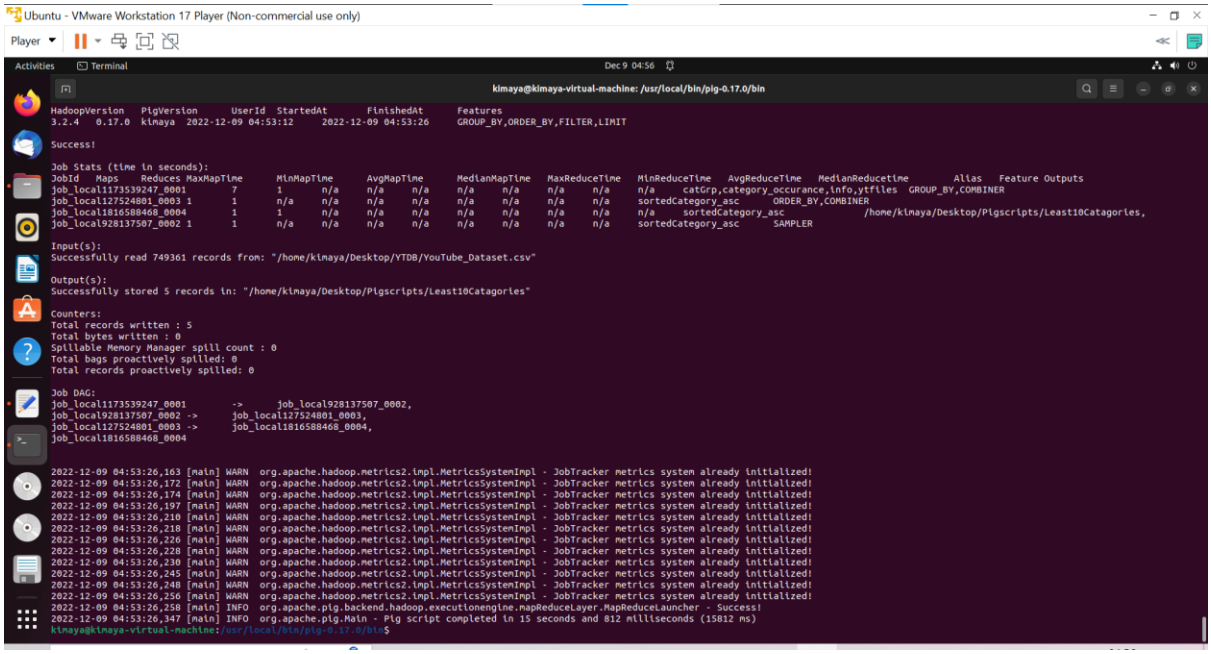
Tail



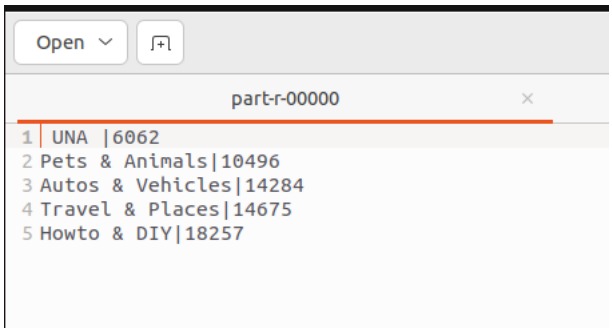
Analysis Using Pig

1) Least5Categories

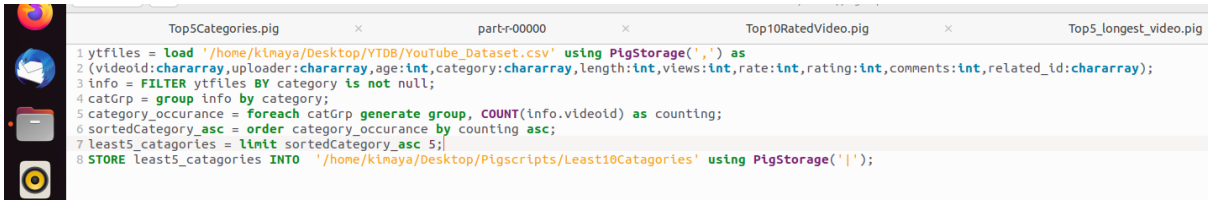
```
ytfiles = load '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' using PigStorage(',') as
(videoid:chararray,uploader:chararray,age:int,category:chararray,length:int,views:int,rate:in
t,rating:int,comments:int,related_id:chararray);
info = FILTER ytfiles BY category is not null;
catGrp = group info by category;
category_occurance = foreach catGrp generate group, COUNT(info.videoid) as counting;
sortedCategory_asc = order category_occurance by counting asc;
least5_catagories = limit sortedCategory_asc 5;
STORE least5_catagories INTO  '/home/kimaya/Desktop/Pigscripts_final/Least5Catagories'
using PigStorage('|');
```



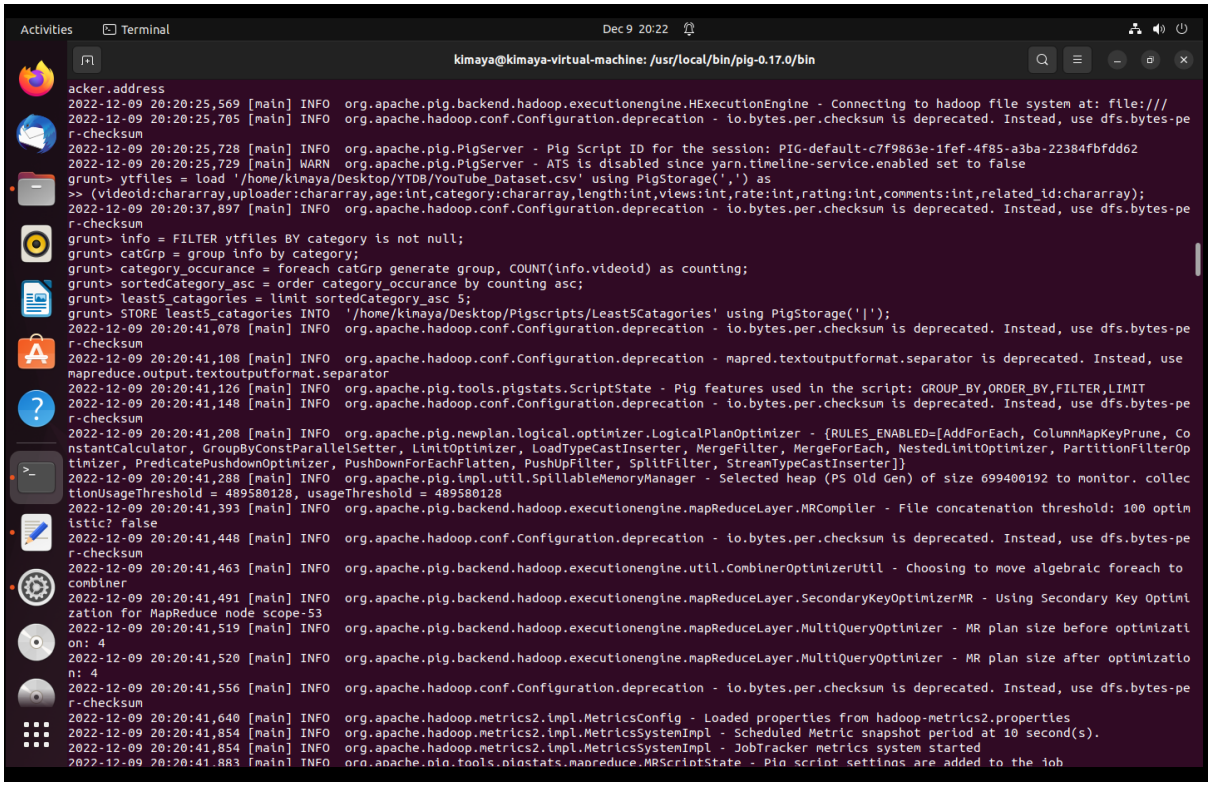
o/p



Pig script



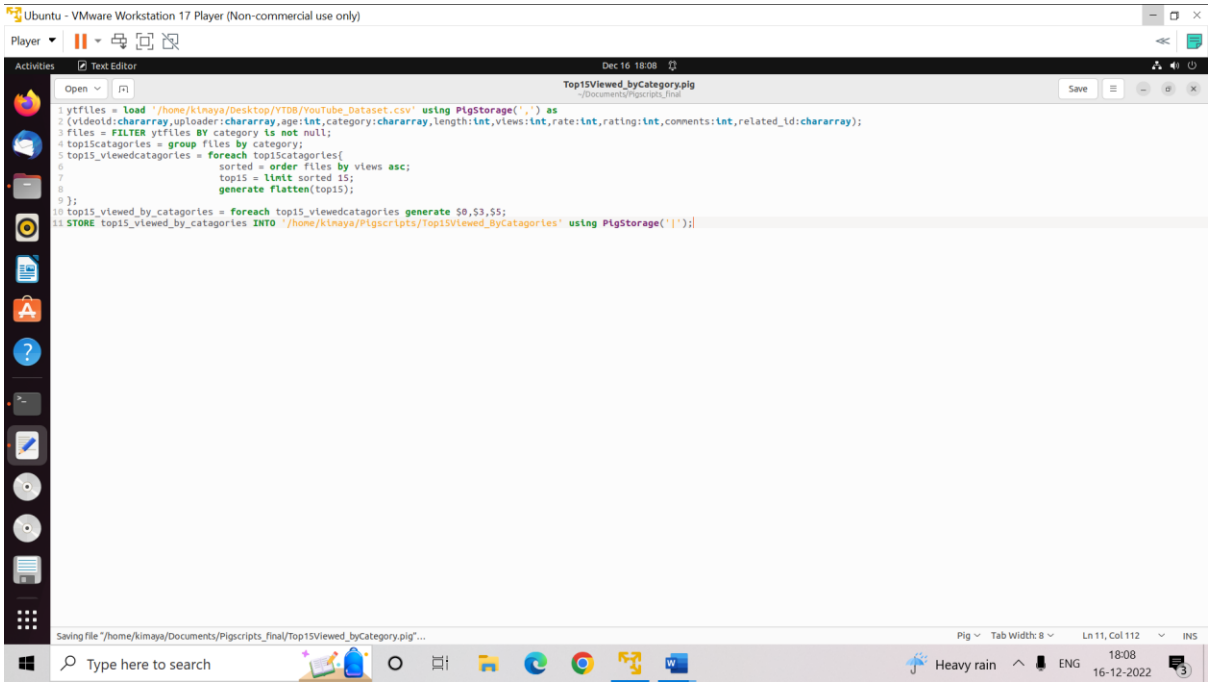
By Grunt shell

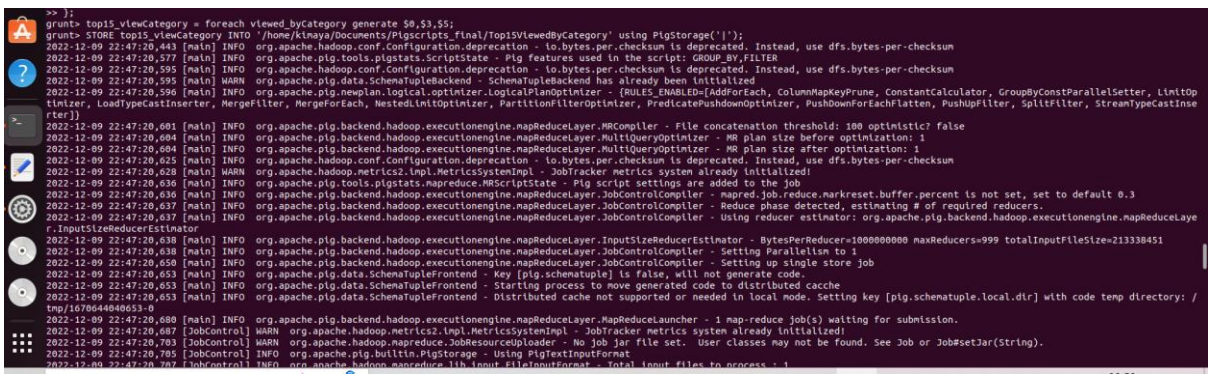


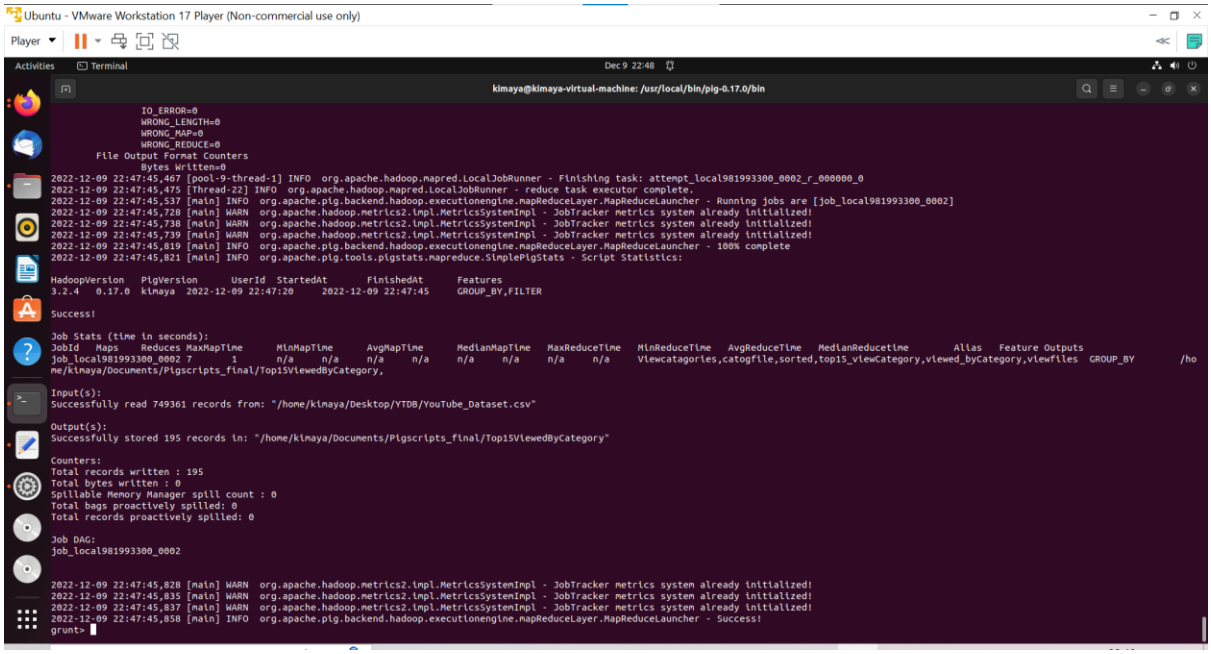

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
3.2.4   0.17.0  kimaya  2022-12-09 20:20:41     2022-12-09 20:21:16     GROUP_BY,ORDER_BY,FILTER,LIMIT

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime      Alias   Feature Outputs
job_local1726312092_0001        7       1       n/a     n/a     n/a     n/a     n/a     n/a     catGrp,category_occurance,info,ytfiles  GROUP_BY,COMBINER
job_local1869038395_0002        1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     sortedCategory_asc      SAMPLER
job_local623033142_0003 1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     sortedCategory_asc      ORDER_BY,COMBINER
job_local978720023_0004 1       1       n/a     n/a     n/a     n/a     n/a     n/a     n/a     sortedCategory_asc              /home/kimaya/Desktop/Pigscripts/Least5Catagories,

Input(s):
Successfully read 749361 records from: "/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv"

Output(s):
Successfully stored 5 records in: "/home/kimaya/Desktop/Pigscripts/Least5Catagories"

Counters:
Total records written : 5
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
```

2) Top15 Viewed by category

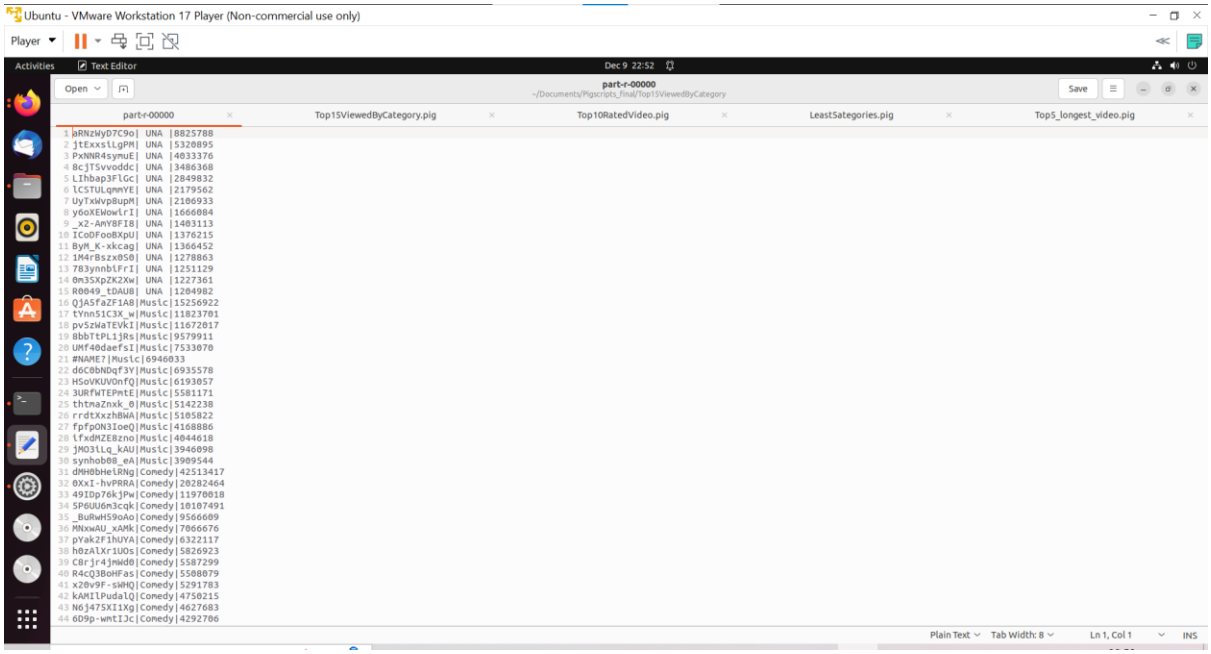
```
ytfiles = load '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' using PigStorage(',') as
(videoid:chararray,uploader:chararray,age:int,category:chararray,length:int,views:int,rate:in
t,rating:int,comments:int,related_id:chararray);
files = FILTER ytfiles BY category is not null;
top15catagories = group files by category;
top15_viewedcatagories = foreach top15catagories{
                    sorted = order files by views asc;
                    top15 = limit sorted 15;
                    generate flatten(top15);
};
top15_viewed_by_catagories = foreach top15_viewedcatagories generate $0,$3,$5;
STORE top15_viewed_by_catagories INTO
'/home/kimaya/Pigscripts/Top15Viewed_ByCatagories' using PigStorage('|');
```





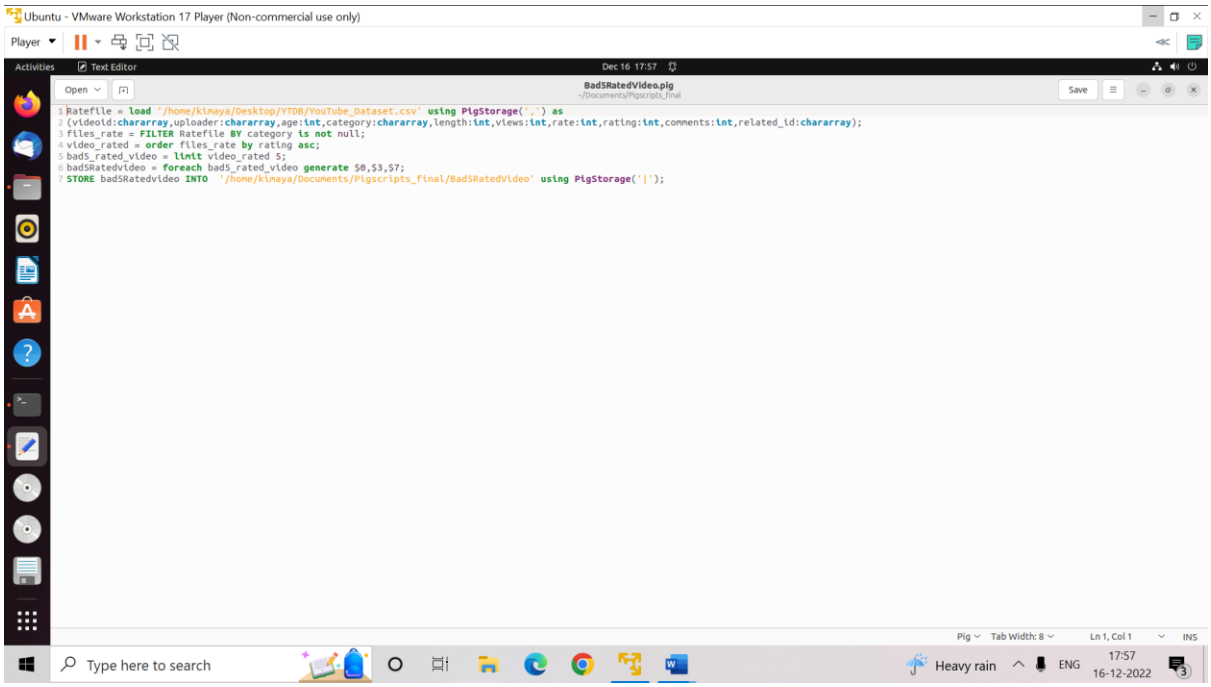

Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Dec 9 22:48
kimaya@kimaya-virtual-machine: /usr/local/bin/pig-0.17.0/bin
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Output Format Counters
Bytes Written=0
2022-12-09 22:47:45,467 [pool-9-thread-1] INFO org.apache.hadoop.mapred.LocalJobRunner - Finishing task: attempt_local981993300_0002_r_000000_0
2022-12-09 22:47:45,475 [Thread-22] INFO org.apache.hadoop.mapred.LocalJobRunner - reduce task executor complete.
2022-12-09 22:47:45,537 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Running jobs are [job_local981993300_0002]
2022-12-09 22:47:45,728 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,738 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,739 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,819 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - 100% complete
2022-12-09 22:47:45,821 [main] INFO org.apache.pig.tools.pigstats.mapreduce.SimplePigStats - Script Statistics:
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
3.2.4 0.17.0 kimaya 2022-12-09 22:47:20 2022-12-09 22:47:45 GROUP_BY,FILTER
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs
job_local981993300_0002 7 1 n/a n/a n/a n/a n/a n/a n/a n/a Viewcatagories,catogfile,sorted,top15_viewCategory,viewed_byCategory,viewfiles GROUP_BY /home/kimaya/Documents/Pigscripts_final/Top15ViewedByCategory,
Input(s):
Successfully read 749361 records from: "/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv"
Output(s):
Successfully stored 195 records in: "/home/kimaya/Documents/Pigscripts_final/Top15ViewedByCategory"
Counters:
Total records written : 195
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_local981993300_0002
2022-12-09 22:47:45,828 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,835 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,837 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 22:47:45,858 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>

Output

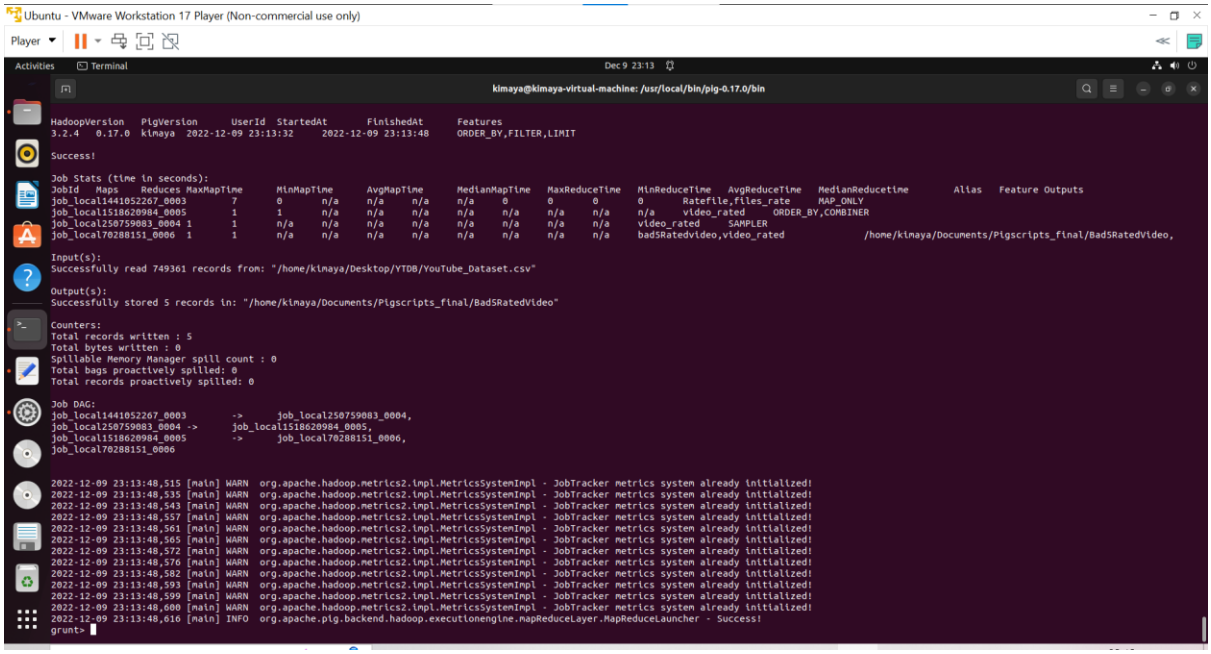

Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Text Editor
Dec 9 22:52
Open
part-r-00000
~/Documents/Pigscripts_final/Top15ViewedByCategory
Save
part-r-00000
Top15ViewedByCategory.pig
Top10RatedVideo.pig
LeastSategories.pig
Top5_longest_video.pig
1 aRNzWyD7C9o| UNA |8825788
2 jtExxslLgPM| UNA |5320895
3 PxNNR4symuE| UNA |4033376
4 8cjTSvvoddc| UNA |3486368
5 LIhbap3FlGc| UNA |2849832
6 lCSTULqmmYE| UNA |2179562
7 UyTxWvpBupM| UNA |2106933
8 yooXEWowlrI| UNA |1666084
9 _x2-AmY8FI8| UNA |1403113
10 ICoDFooBXpU| UNA |1376215
11 ByM_K-xkcag| UNA |1366452
12 1M4rBszx0S0| UNA |1278863
13 7B3ynnbiFrI| UNA |1251129
14 0m3SXpZK2Xw| UNA |1227361
15 R0049_tDAU8| UNA |1204982
16 QjA5faZF1A8|Music|15256922
17 tYnn51C3X_w|Music|11823701
18 pvSzWaTEVkI|Music|11672817
19 8bbTtPL1jRs|Music|9579911
20 UMf40daefsI|Music|7533070
21 #NAME?|Music|6946033
22 d6C0bNDqf3Y|Music|6935578
23 HSoVKUVOnfQ|Music|6193057
24 3URfWTEPmtE|Music|5581171
25 thtmaZnxk_0|Music|5142238
26 rrdtXxzhBWA|Music|5105822
27 fpfpON3IoeQ|Music|4168886
28 lfxdMZE8zno|Music|4044618
29 jMO3iLq_kAU|Music|3946098
30 synhob08_eA|Music|3909544
31 dMH0bHelRNg|Comedy|42513417
32 0XxI-hvPRRA|Comedy|20282464
33 49IDp76kjPw|Comedy|11970018
34 SP6UU6m3cqk|Comedy|10107491
35 _BuRwH59oAo|Comedy|9566609
36 MNxwAU_xAMk|Comedy|7066676
37 pYak2F1hUYA|Comedy|6322117
38 h0zAlXr1UOs|Comedy|5826923
39 C8rJr4jmwd0|Comedy|5587299
40 R4cQ3BoHFas|Comedy|5508079
41 x20v9F-sWHQ|Comedy|5291783
42 kAMIlPudalQ|Comedy|4750215
43 N6j475XI1XQ|Comedy|4627683
44 6D9p-wmIIJc|Comedy|4292706
Plain Text
Tab Width: 8
Ln 1, Col 1
INS

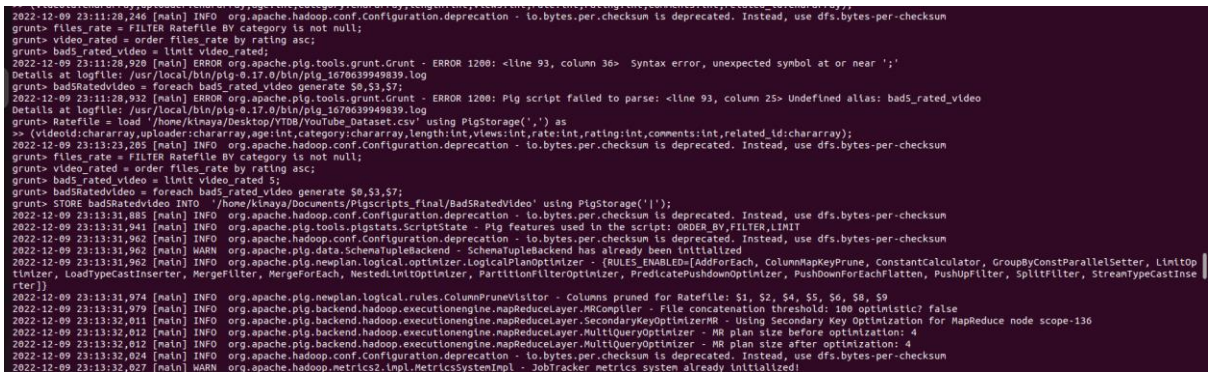
3) 5 Bad Rated YouTube Videos

```
Ratefile = load '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' using PigStorage(',') as

(videoid:chararray,uploader:chararray,age:int,category:chararray,length:int,views:int,rate:int,rating:int,comments:int,related_id:chararray);

files_rate = FILTER Ratefile BY category is not null;

video_rated = order files_rate by rating asc;

bad5_rated_video = limit video_rated 5;

bad5Ratedvideo = foreach bad5_rated_video generate $0,$3,$7;

STORE bad5Ratedvideo INTO  '/home/kimaya/Documents/Pigscripts_final/Bad5RatedVideo' using PigStorage('|');
```

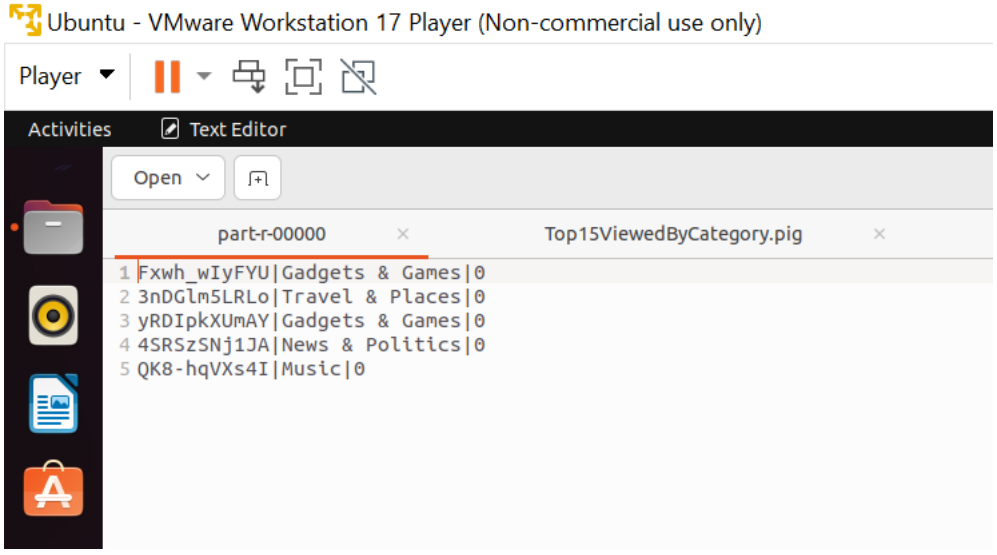


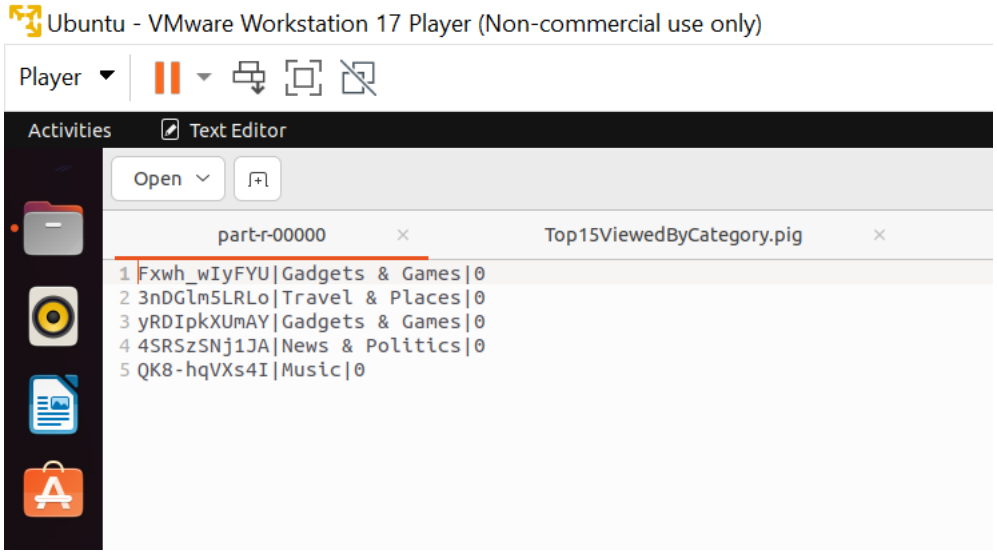

```
grunt> files_rate = FILTER Ratefile BY category is not null;
grunt> video_rated = order files_rate by rating asc;
grunt> bad5_rated_video = limit video_rated;
2022-12-09 23:11:28,920 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1200: <line 93, column 36>  Syntax error, unexpected symbol at or near ';'
Details at logfile: /usr/local/bin/pig-0.17.0/bin/pig_1670639949839.log
grunt> bad5Ratedvideo = foreach bad5_rated_video generate $0,$3,$7;
2022-12-09 23:11:28,932 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1200: Pig script failed to parse: <line 93, column 25> Undefined alias: bad5_rated_video
Details at logfile: /usr/local/bin/pig-0.17.0/bin/pig_1670639949839.log
grunt> Ratefile = load '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' using PigStorage(',') as
>> (videoid:chararray,uploader:chararray,age:int,category:chararray,length:int,views:int,rate:int,rating:int,comments:int,related_id:chararray);
2022-12-09 23:13:23,205 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> files_rate = FILTER Ratefile BY category is not null;
grunt> video_rated = order files_rate by rating asc;
grunt> bad5_rated_video = limit video_rated 5;
grunt> bad5Ratedvideo = foreach bad5_rated_video generate $0,$3,$7;
grunt> STORE bad5Ratedvideo INTO  '/home/kimaya/Documents/Pigscripts_final/Bad5RatedVideo' using PigStorage('|');
```


Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Terminal
Dec 9 23:13
kimaya@kimaya-virtual-machine: /usr/local/bin/pig-0.17.0/bin
HadoopVersion PigVersion UserId StartedAt FinishedAt Features
3.2.4 0.17.0 kimaya 2022-12-09 23:13:32 2022-12-09 23:13:48 ORDER_BY,FILTER,LIMIT
Success!
Job Stats (time in seconds):
JobId Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs
job_local1441052267_0003 7 0 n/a n/a n/a n/a 0 0 0 0 Ratefile,files_rate MAP_ONLY
job_local1518620984_0005 1 1 n/a n/a n/a n/a n/a n/a n/a n/a video_rated ORDER_BY,COMBINER
job_local250759083_0004 1 1 n/a n/a n/a n/a n/a n/a n/a n/a video_rated SAMPLER
job_local70288151_0006 1 1 n/a n/a n/a n/a n/a n/a n/a n/a badSRatedvideo,video_rated /home/kimaya/Documents/Pigscripts_final/BadSRatedVideo,
Input(s):
Successfully read 749361 records from: "/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv"
Output(s):
Successfully stored 5 records in: "/home/kimaya/Documents/Pigscripts_final/BadSRatedVideo"
Counters:
Total records written : 5
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0
Job DAG:
job_local1441052267_0003 -> job_local250759083_0004,
job_local250759083_0004 -> job_local1518620984_0005,
job_local1518620984_0005 -> job_local70288151_0006,
job_local70288151_0006
2022-12-09 23:13:48,515 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,535 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,543 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,557 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,561 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,565 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,572 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,576 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,582 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,593 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,599 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,600 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!
2022-12-09 23:13:48,616 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>


2022-12-09 23:11:28,246 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> files_rate = FILTER Ratefile BY category is not null;
grunt> video_rated = order files_rate by rating asc;
grunt> bad5_rated_video = limit video_rated;
2022-12-09 23:11:28,920 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1200: <line 93, column 36> Syntax error, unexpected symbol at or near ';'
Details at logfile: /usr/local/bin/pig-0.17.0/bin/pig_1670639949839.log
grunt> badSRatedvideo = foreach bad5_rated_video generate $0,$3,$7;
2022-12-09 23:11:28,932 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1200: Pig script failed to parse: <line 93, column 25> Undefined alias: bad5_rated_video
Details at logfile: /usr/local/bin/pig-0.17.0/bin/pig_1670639949839.log
grunt> Ratefile = load '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' using PigStorage(',') as
>> (videoid:chararray,uploader:chararray,age:int,category:chararray,length:int,views:int,rate:int,rating:int,comments:int,related_id:chararray);
2022-12-09 23:13:23,285 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> files_rate = FILTER Ratefile BY category is not null;
grunt> video_rated = order files_rate by rating asc;
grunt> bad5_rated_video = limit video_rated 5;
grunt> bad5Ratedvideo = foreach bad5_rated_video generate $0,$3,$7;
grunt> STORE bad5Ratedvideo INTO '/home/kimaya/Documents/Pigscripts_final/Bad5RatedVideo' using PigStorage('|');
2022-12-09 23:13:31,885 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-12-09 23:13:31,941 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: ORDER_BY,FILTER,LIMIT
2022-12-09 23:13:31,962 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-12-09 23:13:31,962 [main] WARN org.apache.pig.data.SchemaTupleBackend - SchemaTupleBackend has already been inttalized
2022-12-09 23:13:31,962 [main] INFO org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, ConstantCalculator, GroupByConstParallelSetter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NestedLimitOptimizer, PartitionFilterOptimizer, PredicatePushdownOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter]}
2022-12-09 23:13:31,974 [main] INFO org.apache.pig.newplan.logical.rules.ColumnPruneVisitor - Columns pruned for Ratefile: $1, $2, $4, $5, $6, $8, $9
2022-12-09 23:13:31,979 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2022-12-09 23:13:32,011 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.SecondaryKeyOptimizerMR - Using Secondary Key Optimization for MapReduce node scope-136
2022-12-09 23:13:32,012 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 4
2022-12-09 23:13:32,012 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 4
2022-12-09 23:13:32,024 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2022-12-09 23:13:32,027 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl - JobTracker metrics system already initialized!

Output


Ubuntu - VMware Workstation 17 Player (Non-commercial use only)
Player
Activities
Text Editor
Open
part-r-00000
Top15ViewedByCategory.pig
1 Fxwh_wIyFYU|Gadgets & Games|0
2 3nDGlm5LRLo|Travel & Places|0
3 yRDIpkXUmAY|Gadgets & Games|0
4 4SRSzSNj1JA|News & Politics|0
5 QK8-hqVXs4I|Music|0



Hive

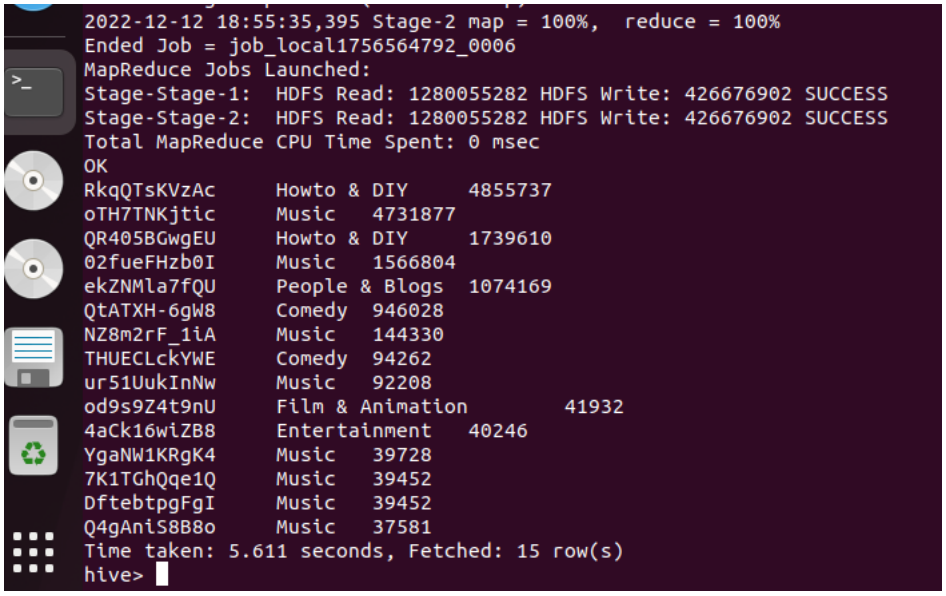
1) Calculating Top 10 Category of YouTube videos

hive> LOAD DATA LOCAL INPATH '/home/kimaya/Desktop/YTDB/YouTube_Dataset.csv' OVERWRITE INTO TABLE ytvideo_info_table;

hive> select vidcategory, count(*) as count_videos from ytvideo_info_table group by vidcategory sort by count_videos desc limit 10;

2) Calculate TOP 15 lengthy Video

Query :- select idvideo,vidcategory,vidlength from YTVideo_info_table SORT BY vidlength DESC LIMIT 15;



3) Calculate top 10 channels with maximum number of comments
Hive> select idvideo,uploader_name,total_comments FROM ytvideo_info_table ORDER BY total_comments DESC LIMIT 10;