Mining and YouTube Data Analysis using Hadoop

B. Uma Maheswari, N. Mythili

Abstract: ***Analysis of structured and consistent data has seen remarkable success in past decades. Whereas, the analysis of unstructured data in the form of multimedia format remains a challenging task. YouTube is one of the most popular and used social media tool. It reveals the community feedback through comments for published videos, number of likes, dislikes, number of subscribers for a particular channel. The main objective of this work is to demonstrate by using Hadoop concepts, how data generated from YouTube can be mined and utilized to make targeted, real time and informed decisions. In our paper, we analyze the data to identify the top categories in which the most number of videos are uploaded. This YouTube data is publicly available and the YouTube data set is described below under the heading Data Set Description. The dataset will be fetched from the Google using the YouTube API (Application Programming Interface) and going to be stored in Hadoop Distributed File System (HDFS). Using MapReduce we are going to analyze the dataset to identify the video categories in which most number of videos are uploaded. The objective of this paper is to demonstrate Apache Hadoop framework concepts and how to make targeted, real-time and informed decisions using data gathered from YouTube.***

Keywords: ***Map Reduce, Mapper Algorithm, Reducer Algorithm, You Tube, data analysis***

I. INTRODUCTION

In Relational Database System to extract a large amount of data, it takes a large amount of time and complexity arises. In the existing technology, we used traditional enterprise systems to analyze, store and transport the data. Centralized server archives all data which are need to be processed. Moreover, centralized system creates too much of a bottleneck (i.e. a bottleneck is one process in a chain of processes, such that its limited capacity reduces the capacity of the whole chain) while processing multiple files simultaneously. This issue is solved by MapReduce algorithm. MapReduce divides a task into small chunks and tasks in a parallel way. The framework sorts the outputs of maps, which acts as input to the reduce tasks. The file system stores the input and output of the MapReduce process. Monitoring, scheduling and re-execution of the failed tasks are handled by the Mapreduce framework. It acts as a master-slave architecture same as in HDFS.

There are two types of nodes in MapReduce, they are Task-tracker and Job-tracker. Master node work is done by Task-tracker and slave node works is handled by Job-tracker. The task-tracker splits the whole program into a number of the individual program and assign it to the workers. The worker computes each program individually and transfers the result, assigns them to a cluster of computers. Final dataset is the collection of entire results which is stored in one place.

All financial sectors are in the challenges to extract the required data from the customer's transactional databases. Google YouTube plays a vital role to do this process. Billions of people are connected across the globe through the videos in every moment of their day today life process. This is the reason for internet traffic and creates drastic problems in scalability of YouTube videos. To overcome these issues big data is handling large amount [1] of data and the problems in the relational databases were eradicated. Big data handles billions of data and is used for data analytics and to reduce the network traffic issues.

In this modern world efficient analysis of business data and their storage in appropriate devices is tedious work. The YouTube videos and the multimedia data which are generated from it are usually unstructured. Perfect analysis of these semi-structured and unstructured multimedia data is a big challenge. Big data stores volume of data [2] with different velocities and varieties with value of data and also handles complexity of them.

II. II. RELATED WORKS

Incredible success is achieved in analyzing the structured dataset. Various companies such as Google, Amazon and Walmart working over big data set to serve for their clients for capturing the market share in the business. There is in need of predictive data analysis over the transactional databases. The performance of any organization can be improved by streaming the data and performing data analysis [7] over the data. Accurate prediction can be done over the huge amount of data in big data analytics. Some of the analytical tools such as Hadoop, Pig, Storm, Hbase plays a vital role in data research. Existence of rich variety dataset are used to handle mega byte of data in day to day businesses

Incredible success is achieved in analyzing the structured dataset. Various companies such as Google, Amazon and Walmart working over big data set to serve for their clients for capturing the market share in the business. There is in need of predictive data analysis over the transactional databases. The performance of any organization can be improved by streaming the data and performing data analysis over the data. Accurate prediction can be done over the huge amount of data in big data analytics

Billions of users were revolving over the YouTube videos. Every day one third of internet users are watching billion hours of video.

Revised Manuscript Received on January 05, 2020.

B. Uma Maheswari, Associate Professor, St. Joseph's College of Engineering, Chennai (Tamilnadu) India.
E-mail: mahespal2002@gmail.com

N. Mythili, Assistant Professor, St. Joseph's College of Engineering, Chennai (Tamilnadu) India. E-mail: mythilyna@gmail.com

Approximately 300 hours of video are uploaded every minute and views are recorded over the video files

Base statistics referred from YouTube details, there is about 500K videos are uploaded to YouTube every day. YouTube collects a wide variety of traditional data points like the number of views, likes, votes, comments, and duration. The collection of the above-listed data points constitutes a very interesting data set to analyze for obtaining implicit knowledge about users, videos, categories, and community interests. The technical functions [4] with big data processing are ingesting data into the system, persisting the data in storage, computing and visualizing data, obtaining optimized results. Data Processing [3] deals with the huge volume of data by collecting over a particular time period processed by using java and distributed processing software framework developed by Apache Hadoop and map reduce framework.

While executing the Map Reduce framework, [5] Hadoop splits the process into map and reducer task that performs operations such as issuing task, checks the completion of the task, replications of data and data transformation from one cluster to another. YouTube is used to promote various company through ads. Things that are insisted on YouTube like movies, songs, brand ads, and artists depends on the number of viewers, likes, and comments. Companies or artists can analyze their performance anywhere from anytime. The on data processing survey [6] will help in understanding various technical aspects of the Map Reduce framework and its view for handling data.

This proposed work provides an idea for people, who use YouTube for promotion and for other purpose, understand how data mining and data analytics can prove them helpful by fetching meaningful results in terms of understanding their performance and changing trends among people. There are several Big Data analytics platforms available such as HIVE, HBASE, PIG to handle such volume of data. In this paper, we have chosen the MapReduce framework for analyzing our dataset. The Operating System chosen for this experiment is Ubuntu.

In this paper, we are going to analyze a YouTube log data set and obtain the list of top 10 videos based upon the rating. Big data has lately gained more popularity and there's still much more to discover in it. In Relational Database System, to extract a large amount of data it takes a large amount of time and complexity arises that's why we use Map reduce and hive. In our paper, first, we will convert java code and Hadoop library files into a .jar file. Then the given YouTube log data set will be exported to HDFS. Thereafter executing the query, the output will be stored in HDFS. The output contains n number of video names along with the average of their rating.

III. MAP REDUCE IMPLEMENTATION OF YOU TUBE DATA

In our procedure, we use Hadoop framework and Map Reduce programming for extracting HDFS data format from YouTube API dataset, splitting various data modules using MapReduce algorithm using Java programming.

A. Fetching Youtube Data Using API

Generate API Key to Fetch YouTube Data To communicate with YouTube API an Application program interface Key is required, Google Developer allows you to create a unique key to connect to YouTube.

Step 1: Log into https://developers.Google.com/ with existing credentials.
Step 2 : To create the unique API key for retrieving data, a new project needs to be created from the Google provided developer's console.
Step3: Go to ttps://console.developers.Google.com/project
Step 4: Click create project.
Step 5: A new project needs to be created.
Step 6: To create a new API key Google provides the YouTube. Data API that is available under the developer tools.
Step 7: To utilize the YouTube Data API, it needs to be enabled under the logged in credentials. Click "Enable" under the YouTube Data API.
Step 8: Once the YouTube data API is enabled, create credentials in order to utilize the API.
Step 9: Add credentials to the project. YouTube provides three options for creating an API Key. They are API key, client ID, service account.
Step 10 : Create Client ID: to create a JSON file to fetch data, we need to select the application that will be using the data.
Step 11: Provide a name for the Client ID.
Step 12: YouTube creates the Client ID for the project to utilize and provides the API key.

B: Top 5 Video Categories Determination using Mapper

YouTube videos are collected and Top5 class holds the highest categories. This mapper class is extended with the same arguments. Then the object named "category" is declared from the defined Top5 class. During the MapReduce process the value of 'v' is always det yo '1' for all key value pairs. A static variable 'one' is declared and set it to a constant integer value '1', so that all the key-value pairs will be assigned to the value '1'

The mapper job is obtained by the Java code for determining the video categories is shown in the below Fig1

Mapper1.java

```
package youtube1;

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

public class Mapper1 extends org.apache.hadoop.mapreduce.Mapper<LongWritable, Text, Text, LongWritable>
{
        private Text word=new Text();

        @Override
        protected void map(LongWritable key, Text value,Context context)
                        throws IOException, InterruptedException {
                String line = value.toString();
        String[] words = line.split("\t");
           if((words.length)>5)
           {
                   word.set(words[3]);

           }
           context.write(new Text(word), new LongWritable(1));



        }
}
```

Fig 1: Mapper for Video Categories Determination

Top5 is a class name which has highest priority of collection video. For converting the unstructured data to structured one, the map method is running for key and their corresponding value pairs. Split the lines and store all the contents in an array, such that all columns are stored in a single row. The fourth column data which contains video category is stored. The entire process is completed and the key and values are recorded. The video category is the key value and its value is 'one' is stored. Thus the output of mapper method is obtained.

C. Running Mapreduce Videos

MapReduce has 2 phases called map and reduce. Both phases has input and output format as key-value pair. The input of map phase will be data stored in the HDFS. The output of map phase is transferred to reduce as input. Reduce phase concentrates on sorting and produces the output.

Mapper Code Algorithm:

Step 1 : Take a class by name Top5_categories, extend the Mapper default class having the arguments keyIn as LongWritable and Valuein as Text and KeyOut as Text and ValueOut as IntWritable.
Step 2: Declare a private Text variable 'category' which will store the category of videos in YouTube.
Step3:Declare a private
static IntWritable variable 'one' which will be constant for every value. MapReduce deals with Key and Value pairs. Here we can set the key as gender and value as age.
Step 4: Override the map method which will run one time for every line.
Step 5 : Store the line in a string variable 'line'.
Step 6 : Split the line by using tab "\t" delimiter and storing the values in a String Array so that all the columns in a row are stored in the string array.
Step 7 : Take a condition if we have the string array of length greater than 6 which means if the line or row has at least 7 columns then it will enter into the if condition and execute the code to eliminate the ArrayIndexOutOfBoundsException.

We are writing the key and value into the context which will be the output of the map method

D. Reducer Algorithm for Video Categories

The Reducer class is extended from the mapper class with the same arguments, the pair of key and values with their corresponding input and output. Now the reduce method will run for all the key and value pairs. A variable named 'sum' is decalred, which sum all the values of 'v' in the key and value pairs with the same key. The final key and value pair is the output for the Top5 video categories, where the key is unique and value is the sum obtained.

Reducer1.java

```
package youtube1;

import java.io.IOException;

import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;

public class Reducer1 extends org.apache.hadoop.mapreduce.Reducer<Text, LongWritable, Text, LongWritable>{

	@Override
	protected void reduce(Text word, Iterable<LongWritable> values,Context context)
			throws IOException, InterruptedException {
		long sum=0;
		for (LongWritable value : values) {
			sum+=value.get();

			}
		context.write(word, new LongWritable(sum));

	}

}
```

Fig 2: Reducer Algorithm for Youtube Data

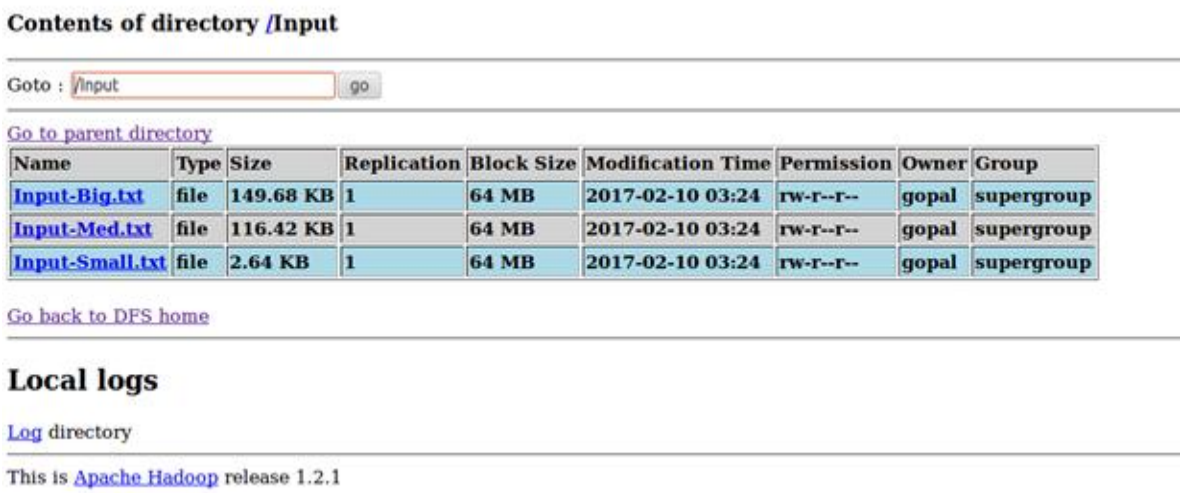
The MapOutputKeyClass and MapOutputValueClass are the configuration classes which are included in the main class. This class verifies the ouput key type and the output value type of key value pairs of the mapper and this is provided as input for reducer code. The job of the reducer is handled by the Java code to determine the video categories is shown in Fig 2. The command " Hadoop jar/YouTube.jar/youtubedata.txt /topuploader_out" access the HDFS and the result is shown in the Fig. 3.

Fig 3: Top 5 Video Up loaders in YouTube

```
jpasolutions@ubuntu:~$ start all-sh
start: Unknown job: all-sh
jpasolutions@ubuntu:~$ start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /home/jpasolutions/hadoop-2.7.1/logs/ha
doop-jpasolutions-namenode-ubuntu.out
localhost: starting datanode, logging to /home/jpasolutions/hadoop-2.7.1/logs/ha
doop-jpasolutions-datanode-ubuntu.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /home/jpasolutions/hadoop-2.7.1/
logs/hadoop-jpasolutions-secondarynamenode-ubuntu.out
starting yarn daemons
starting resourcemanager, logging to /home/jpasolutions/hadoop-2.7.1/logs/yarn-j
pasolutions-resourcemanager-ubuntu.out
localhost: starting nodemanager, logging to /home/jpasolutions/hadoop-2.7.1/logs
/yarn-jpasolutions-nodemanager-ubuntu.out
jpasolutions@ubuntu:~$ hdfs dfs -put /home/jpasolutions/BigData/Reethika/Project/youtubedata.txt /mr/youtube1/input10101
jpasolutions@ubuntu:~$
```

The reducer will start after the mapper is completed by 100%. The file system displays the number of bytes which is read from the input file on local disk and HDFS, also the number of bytes written on the output file on local disk and HDFS is shown in the Fig. 4

Fig 4: Input Files are Uploaded & Stored, Output Files are Created & Stored



Contents of directory /Input

Goto : /Input go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
Input-Big.txt	file	149.68 KB	1	64 MB	2017-02-10 03:24	rw-r--r--	gopal	supergroup
Input-Med.txt	file	116.42 KB	1	64 MB	2017-02-10 03:24	rw-r--r--	gopal	supergroup
Input-Small.txt	file	2.64 KB	1	64 MB	2017-02-10 03:24	rw-r--r--	gopal	supergroup

Go back to DFS home

Local logs

Log directory

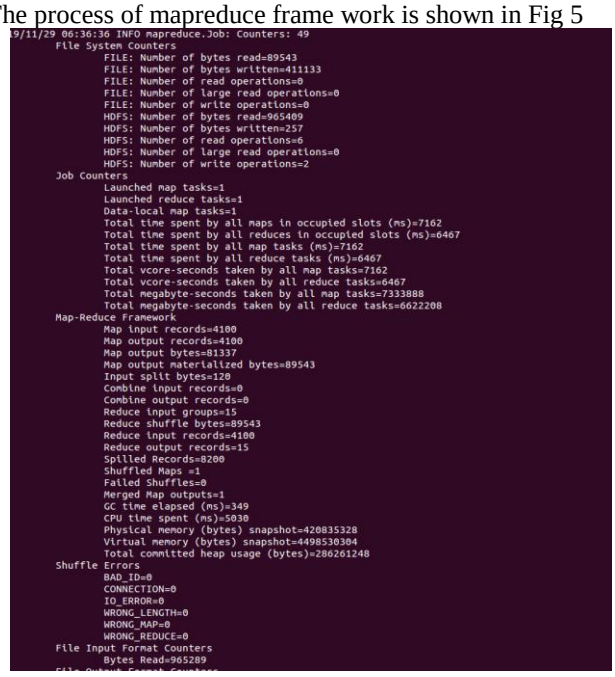
This is Apache Hadoop release 1.2.1

E. Reducer Code for YouTube Videos

1. Extend the default Reducer class with arguments KeyIn as Text and ValueIn as IntWritable which are same as the outputs of the mapper class and KeyOut as Text and arguments KeyIn as Text and ValueIn as IntWritable which are same as the outputs of the mapper class ValueOut as IntWritbale which will be final outputs of our MapReduce program.
2. Override the Reduce method which will run each time for every key.
3. Declare an integer variable sum which will store the sum of all the values for each key.
4. Each loop is taken which will run each time for the values inside the “Iterable values” which are coming from the shuffle and sort phase after the mapper phase.
5. Store and calculating the sum of the values.

Write the respected key and the obtained sum as value to the context.

The process of mapreduce frame work is shown in Fig 5



```
19/11/29 06:36:36 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=89543
                FILE: Number of bytes written=411133
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=965409
                HDFS: Number of bytes written=257
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7162
                Total time spent by all reduces in occupied slots (ms)=6467
                Total time spent by all map tasks (ms)=7162
                Total time spent by all reduce tasks (ms)=6467
                Total vcore-seconds taken by all map tasks=7162
                Total vcore-seconds taken by all reduce tasks=6467
                Total megabyte-seconds taken by all map tasks=7333888
                Total megabyte-seconds taken by all reduce tasks=6622208
        Map-Reduce Framework
                Map input records=4100
                Map output records=4100
                Map output bytes=81337
                Map output materialized bytes=89543
                Input split bytes=120
                Combine input records=0
                Combine output records=0
                Reduce input groups=15
                Reduce shuffle bytes=89543
                Reduce input records=4100
                Reduce output records=15
                Spilled Records=8200
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=349
                CPU time spent (ms)=5030
                Physical memory (bytes) snapshot=420835328
                Virtual memory (bytes) snapshot=4498530304
                Total committed heap usage (bytes)=286261248
        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=965289
        File Output Format Counters
```

Fig 5: MapReduce Framework of Youtube Data

IV. RESULTS AND DISCUSSION

Finally, after the execution of code, we get a list of the top 10 frequently viewed videos. High latency with that is unusable for real-time applications. There can be complicated to implement everything as a MapReduce program. MR is not suitable for a large number of short on-line transactions when we have OLTP needs. Implementation of interactive jobs and modals is impossible because MapReduce is only suitable for batch processing jobs. Due to more space consumption for each job, the implementation of the MapReduce jobs becomes expensive. The MapReduce does not support the interaction between the intermediate processes, which means the job is isolated and executed as multi-threads.

A. Accuracy

HDFS and Hadoop clusters are used to determine the time needed to access from different places and from this, the overall accuracy is calculated. Whenever the modules are used one by one (ie. in series), the product of accuracy of its different modules will give rise the overall accuracy of the application. For making real time decisions, We tested our application using You Tube API.

B. Time Efficiency

Time efficiency plays an important role in this application. MapReduce programming model is used in this application in achieving lower response time. This makes the Hadoop cluster to reduce the execution time. Distributed processing and reduction in access time are ensured with the help of Hadoop. These factors increase the overall time efficiency.

V. CONCLUSION

This work focuses on the area where the organizations and movie producers will rate their products and compare with their competitors. From the methodologies used in the paper, the algorithm generates reports not only for viewing public comments, likes and dislikes but also about the channels where the reviews are given and about the comparison of reviews with their competitors.

This paper is intended to analyze the YouTube Big Data and come up with significant insights which cannot be determined otherwise. The output results of YouTube data analysis show key insights that can be extrapolated to other use cases as well. This work focuses on the area where the organizations and movie producers will rate their products and compares with their competitors. From the methodologies used in the paper, the algorithm generates reports not only for viewing public comments, likes and dislikes but also about the channels where the reviews are given and about the comparison of reviews with their competitors.

One of the output results describes that for specific video categories in which most number of videos is uploaded. This concludes that video which had fallen under particular category has a direct significance to the YouTube video's ranking, according to YouTube Analytics. Hence, companies which are uploading the videos of highest category can have the high demand. For example, if the company falls under 'Comedy' or 'Education' category, a meaningful discussion in the form of comments can be triggered on YouTube. Hadoop and MapReduce are used here in analyzing the dataset in YouTube and it is justified in all aspects. A comment analysis can further be conducted to understand the attitude of people towards the specific video.

REFERENCES

1. Phaneendra SV, Reddy EM , “Big Data- Solutions for RDBMS Problems-A Survey” In 12th IEEE/IFIP Network Operations & Management Symposium, NOMS 2010
2. Rahasekar D, Dhanamani C, Sandhya SK, “A Survey on Big Data Concepts and Tools”, International Journal of Emerging Technology and Advanced Engineering, Vol. 5, No. 2, pp. 80-81, 2015.
3. Ingle A, Kante A, Samak S, Kumari A, “Sentiment Analysis of Twitter Data Using Hadoop”, International Journal of Engineering Research and General Science, Vol. 3, Issue 6, pp. 144-147, 2015
4. “Big Data Overview Big Data: Concepts, Methodologies, Tools, and Applications”, Information Resources Management Association (IRMA), IGI Global, Vol 1, 2016.
5. Mukherjee A, Datta, , Jorapur R, Singhvi, R, Haloi, S.; Akram, W, “Shared Disk Big Data Analytics with Apache Hadoop”, 19th International Conference on High Performance Computing,, pp. 1-6, Dec 2012
6. Lee KH, Lee YJ, Choi H, Chung YD, Moon B. “Parallel data processing with MapReduce: a survey” AcM sIGMoD Record, Vol. 40, No. 4, Jan 2012
7. Bifet A, “Mining Big Data In Real Time”, Informatica Vol. 37, No. 1, pp. 15–20, Dec 2012.

AUTHORS PROFILE

Dr. B. Uma Maheswari, is working as an Associate Professor in the Department of Computer Science Engineering at St. Joseph’s College of Engineering. She received M.E Degree in Computer Science and Engineering in 2004 and Ph.D Degree in 2015 from Anna University. She has 20 years of teaching experience. Her area of interest includes Software Testing, Machine Learning and Data Mining.

Dr. N. Mythili, is working as an Assistant Professor in the Department of Computer Science Engineering at St. Joseph’s College of Engineering. She received M.E Degree in Computer Science and Engineering in 2007 and Ph.D Degree in 2019 from Anna University. She has 22 years of teaching experience. Her area of interest includes Spatial Query Processing, Cloud Computing and Computer Graphics.