International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-5, Issue-4 ,2018 Regular Edition E-ISSN: 2349-7084

BIG DATA ANALYSIS ON YOUTUBE USING HADOOP and MAPREDUCE

Soma Hota*

Amity School of Engineering and Technology - Computer Science Engineering, Amity *University, Mumbai - Pune*

Available online at: http://www.ijcert.org

Received: 18/April /2018, Revised: 19/April /2018, Accepted: 24/April /2018, Published: 27/April/2018

Abstract: We live in a digitalized world today. An enormous amount of data is generated from every digital service we use. This enormous amount of generated data is called Big Data. According to Wikipedia, Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them .Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating and information privacy. Google's video streaming services, YouTube, is one of the best examples of services which produces a huge quantity of data in a very short period. Data mining of such an enormous quantity of data is performed using Hadoop and MapReduce to measure performance. Hadoop is a system which provides a reliable shared storage of such huge datasets on the cloud and also provides an analysis system. The storage is provided by HDFS (Hadoop Distributed File System) and analysis by MapReduce. MapReduce is a programming model and an associated implementation for processing large data sets. This paper presents the algorithmic work on big data problem and its optimal solution using Hadoop cluster and HDFS for YouTube dataset storage and using parallel processing to process large data sets using Map Reduce programming framework. In this paper, we solve two problem statements using the YouTube dataset – top 5 video categories (genres) with the maximum number of videos uploaded and top 5 video uploaders on YouTube. A particularly distinguishing feature of this paper is its focus on analytics performed in unstructured data, which constitute 95% of big data.

Key words: Big Data definition, Data mining, YouTube data analysis, Hadoop, HDFS, MapReduce, unstructured dataset analysis.

1. Introduction

Analysis of structured dataset has demonstrated tremendous success. In a recent whitepaper from Filene Research Institute, author Philipp Kallerhoff states: ―Companies as varied as Amazon, Google, Walmart, and Wells Fargo are turning to ―big data for member insights that will help them serve clients and capture market share [6]. He added, a prerequisite for developing these (predictive) and other models is a well-maintained database with as much transactional detail as possible [6]. Financial companies and the finance departments of companies are already facing challenges in extracting the required information from the huge transactional data from the customers. However, the nature of such data is structural and easily manageable. Google's YouTube allows billions of people to connect, inform, and inspire others across the globe using originally created videos on a daily, every minute, basis. Thus, unsurprisingly, YouTube has a great impact on Internet traffic nowadays, yet itself is suffering from a severe problem of scalability. Storage, processing and efficient analysis of such enormous data over a short period of time is a very demanding task. The data generated from billions of YouTube videos is primarily unstructured. Quick, efficient and accurate analysis of this unstructured or semi-structured data remains a challenging task. According to statistics published by Google, YouTube has over a billion users — almost one-third of all people on the Internet — and each day those users watch a billion hours of video, generating billions of views [4]. YouTube has approximately 300h of video uploaded every minute and billions of views generated every day [3].

Table 1. YouTube statistics

YouTube Company Statistics	**Data**
Total number of YouTube users	1,325,000,000
Hours of video uploaded every minute	300 hours
Number of videos viewed everyday	4,950,000,000
Total number of hours of video watched every month	3.25 billion hours
Number of videos that have generated over 1 billion views	10,113
Average time spent on YouTube per mobile session	40 minutes

The above image, Figure1 [3], provides us with important statistics and helps us infer that approximately 300K 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. Movie production houses release their movie promos and songs on YouTube. Company brands release their ads on YouTube for promotion. Budding artists present and promote their art on YouTube for publicity. These are just a few examples. The success rate of movies, songs, brand ads, and artists largely depends on the number of viewers, likes, and comments. Companies or artists can not only analyze their own performance but also analyze their competitors'. The main objective of this project is to help organizations or people in general, who use YouTube for marketing/promotion, 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. The procedure is very simple and broken down into 6 steps.The below Flow Diagram (Figure 2) helps illustrate the steps very effectively.

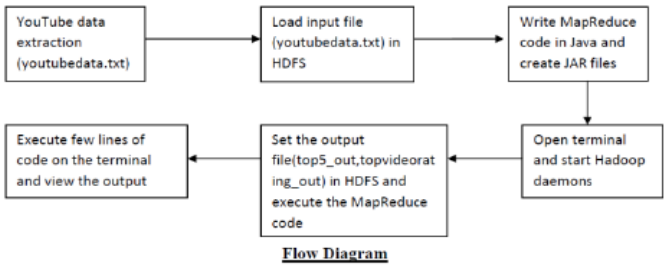

Figure 2-overview of methodology

A. Apache Hadoop Platform

Considering the magnitude of data produced by YouTube over a very short period of time, Hadoop is definitely the most preferred framework for data analysis. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high- availability,

the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.[9]

The project includes these modules:

- Hadoop Common: The common utilities that support the other Hadoop modules.
- Hadoop Distributed File System (HDFS™): A distributed file system that provides high-throughput access to application data.
- Hadoop YARN(Yet Another Resource Negotiator): A framework for job scheduling and cluster resource management.
- Hadoop MapReduce: A YARN-based system for parallel processing of large datasets. [9]

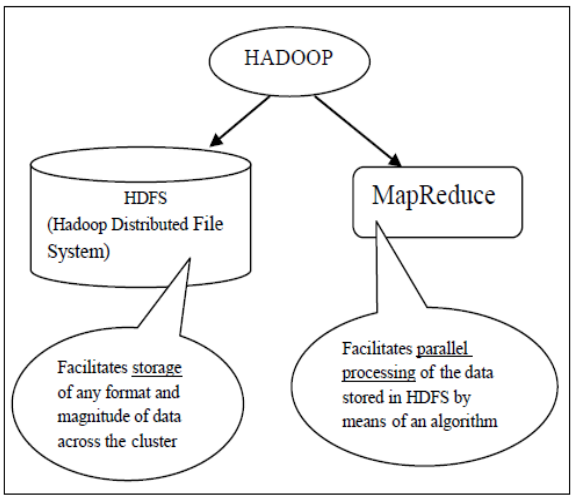

Figure 3-Apache Hadoop Ecosystem

From Figure 3, we understand that Hadoop stores any type of data across the cluster. A cluster is a group of interconnected systems which produce data, collectively called as nodes, which work together.

Hadoop Distributed File System (HDFS):

HDFS has two main classes:
1. Name Node: Contains metadata about the data stored
2. Data Node: Where actual data is stored
3. Secondary Name Node: Contains copy of NameNode DF – Data File this is best illustrated in Figure 4.

Each data block in the DataNode is replicated by a factor of 3(default value) .i.e. there are 3 copies of each data block in the data node. This replication mechanism is provided to ensure that there is no loss of data in case any of the data nodes fail. The replication factor can be decided by the organization using Hadoop system as per their requirements for storing and processing their data.

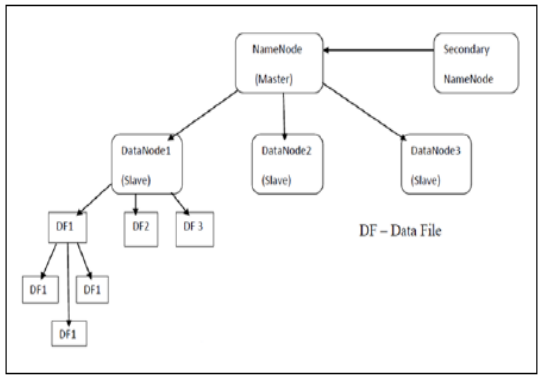

Figure 4 – Hadoop Cluster

B. Determining Which System is Best Suited for This Project

While Hadoop provides the ability to store data on HDFS, there are many programming frameworks available that allow us to perform distributed and parallel processing and analyzing on large datasets in a distributed environment. The most popular ones are MapReduce, Pig, and Hive. Figure 5 [10] helped us analyze which application is better suited for this project.

Featured	MapReduce	Pig	Hive
Language	Algorithm of Map and Reduce Functions (Can be implemented in C, Python, Java)	PigLatin (Scripting Language)	SQL-like
Schemas/Types	No	Yes (implicit)	Yes(explicit)
Partitions	No	No	Yes
Server	No	No	Optional (Thrift)
Lines of code	More lines of code	Fewer (Around 10 lines of PIG = 200 lines of Java)	Fewer than MapReduce and Pig due to SQL Like nature
Development Time	More development effort	Rapid development	Rapid development
Abstraction	Lower level of abstraction (Rigid Procedural Structure)	Higher level of abstraction (Scripts)	Higher level of abstraction (SQL like)
Joins	Hard to achieve join functionality	Joins can be easily written	Easy for joins
Structured vs Semi-Structured Vs Unstructured data	Can handle all these kind of data types	Works on all these kind of data types	Deal mostly with structured and semi-structured data
Complex business logic	More control for writing complex business logic	Less control for writing complex business logic	Less control for writing complex business logic
Performance	Fully tuned MapReduce program would be faster than Pig/Hive	Slower than fully tuned MapReduce program, but faster than badly written MapReduce code	Slower than fully tuned MapReduce program, but faster than bad written MapReduce code

Figure 5 – Comparison between MapReduce, Pig and Hive

Since the YouTube dataset used is unstructured, Hive cannot be used since Hive deals mostly with structured and semi-structured data(Figure 5). Also, Pig requires the knowledge of a new scripting language which is different from MySQL, whereas, MapReduce can be implemented

using any of the three most well-known and dominating languages -C, Python or Java. Also, a well-developed MapReduce algorithm has a higher efficiency than Pig. Thus, MapReduce is the best choice for this project.

C. Mapreduce

The MapReduce programming algorithm is clearly summarized in the following quote [11]: ―The computation takes a set of input key/value pairs, and produces a set of output key/value pairs. The user of the MapReduce library expresses the computation as two functions: map and reduce. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the reduce function. The reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. Typically just zero or one output value is produced per reduce invocation. The intermediate values are supplied to the user's reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

Phases in MapReduce : The main concept behind MapReduce job is splitting a large data set into independent smaller data sets, mapping those smaller data sets to form a collection of <key,value> pairs and reducing overall pairs having the same key for parallel processing. A key-value pair (KVP) is a set of two inter-connected data items: a key is a unique identifier for a particular data item in the dataset, and the value is either the count of the data that is identified or the position value of that data. Because this parallel processing mechanism follows the Divide and Process rule, it significantly improves the speed and reliability of the cluster, returning solutions more quickly and with greater reliability.

Every MapReduce job consists of the following two main parts:
I. The Mapper
II. The Reducer

I. Mapper Phase

The first phase of a MapReduce program is called mapping. A mapping algorithm is designed. The main objective of the mapping algorithm is to accept the large input dataset and divide it into smaller parts (sub-dataset). These sub data sets are distributed to different nodes by the Job Tracker. The nodes perform parallel processing (map task) on these sub-datasets and convert them into <Key,Value> pairs as output. The value of 'Value' in each KVP is always set to 1. Each KVP output is then fed as input to the reducer phase.

II. Reducer Phase

The reducing phase aggregates values of KVP together. A reducer function receives the KVP input and iterates over each KVP. It then combines the KVP containing the same Key and increments the 'Value' by 1. It then combines these values together, returning a single output value which is the aggregate of same keys in the input dataset.

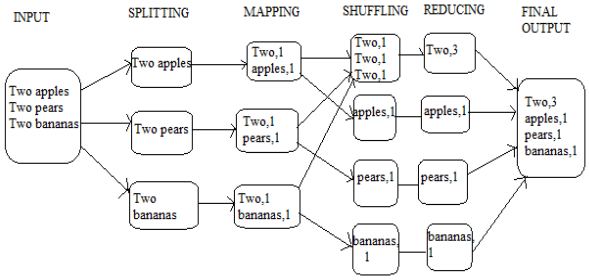

Figure 6 – Word Count Example illustrating MapReduce concept

The following diagram, Figure 7, gives an overview summary of the MapReduce concept.

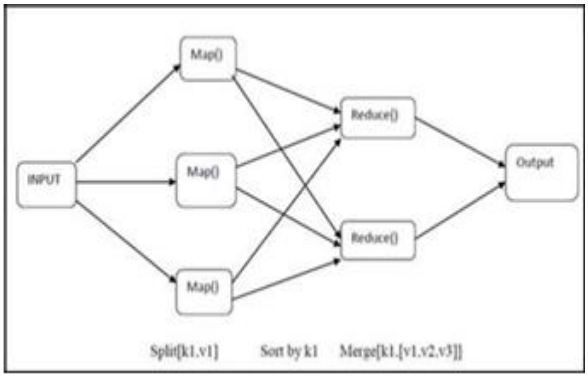

Figure 7 – MapReduce Overview concept

2. Methodology

A. Dataset

The dataset used for analysis in this project is named 'youtubedata.txt'. It is publicly available on the following link,

https://drive.google.com/file/d/0ByJLBTmJojjzR2x0MzVpc2Z6enM/view

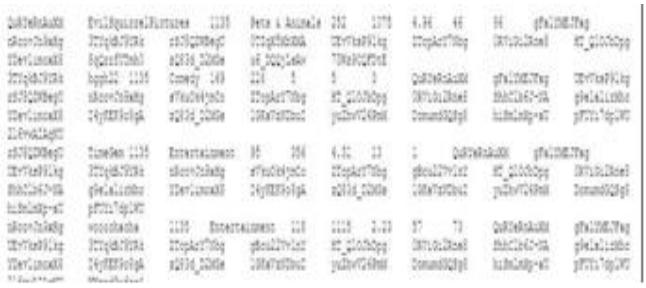

Figure 8-screenshot of the youtubedata.txt dataset

This unstructured dataset consists of data from approximately 3000 videos and contains 10 columns in total.

Dataset Description

Column 1: Video id of 11 characters. Channel ID is an 11-character string key that is used to uniquely identify YouTube video. This ID is case sensitive. The individual characters come from a set of 64 possibilities (A-Za-z0-9_-)

Column 2: name of video uploaded (username). YouTube usernames are limited to 20 characters, are case insensitive and must be alphanumeric characters. Minimum length of a channel name is 6 characters

Column 3: Interval between the day of establishment of you tube and the date of uploading of the video.

Column 4: Category of the video. A video category identifies a category that has been or could be associated with uploaded videos. YouTube provides 16 video categories such as Games, Movies, Music, and Comedy etc.

Column 5: Length of the video. The duration is given in the form of minutes.

Column 6: Number of views for the video. View count is unsigned long integer. View count is the number of times the video has been viewed. Maximum number of views possible is 9,223,372,036,854,775,808 (as defined by YouTube)

Column 7: Rating on the video.

Column 8: Number of ratings given for the video.

Column 9: Number of comments on the videos. Comment count is unsigned long integer. Comment count is the number of comments for the video. There is no limit on the maximum number of comments. YouTube restricts comment length to 500 characters

Column 10: Related video ids with the uploaded video.

B. Installing Hadoop Cluster (Setting up a Single Node Cluster) [15]

This experiment is based on Ubuntu. I have set up a single node cluster. A single node cluster means only one Data Node is running and all the Name Node, Data Node, Resource Manager and Node Manager are set on a single machine. This project uses Hadoop-2.7.3 version.

C. MapReduce

Problem Statement 1: To determine top 5 video categories on YouTube

Mapper Algorithm:

We take a class by name Top5_categories. We then extend the Mapper class which has arguments .We then declare an object 'category 'which stores all the categories of YouTube. As explained before, in the <k,v> pairs in MapReduce, the value of 'v' is always set to 1 for every key-value pair. In the next step, we declare a static variable 'one 'and set it to the constant integer value 1 so that every 'value 'in every <k,v> pair automatically gets assigned to value 1.

We override the Map method which will run for all <k,v> pairs.

We then declare a variable 'line' which will store all the lines in the input youtubedata.txt dataset.

We then split the lines and store them in an array so that all the columns in a row are stored in this array. We do this to make the unstructured dataset structured.

We then store the 4th column which contains the video category.

Finally, we write the key and value, where the key is 'category 'and value is 'one'. This will be the output of the map method.

Reducer Algorithm:

We first extend the Reducer class which has the same arguments as the Mapper class .i.e. <k(input),v(input)> and <k(output),v(output)>.

Again, same as the Mapper code, we override the Reduce method which will run for all <k,v> pairs.

We then declare a variable sum which will sum all the values of the 'v'in the <k,v> pairs containing the same 'k'(key) value.

Finally, it writes the final <k,v> pairs as the output where the value of 'k' is unique and 'v' is the value of sum obtained in the previous step.

The two configuration classes (MapOutputKeyClass and MapOutputValueClass) are included in the main class to clarify the Output key type and the output value type of the <k,v> pairs of the Mapper which will be the inputs of the Reducer code.

Execute

We first create a jar file YouTube.jar. We then execute the following commands on the Ubuntu terminal. We first start Hadoop's HDFS by starting all the daemons and checking their assigned ports as shown in Figure 9.

Figure 9 – Starting Hadoop daemons

Next we come out of the hadoop-2.7.3 folder and access hdfs using the command below:

Command: hadoop jar Desktop/YouTube.jar /youtubedata.txt /top5_out

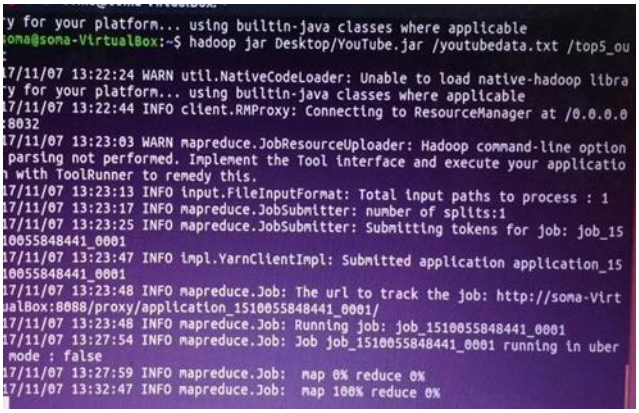

Figure 10 Problem Statement 1 executions

Here 'hadoop' specifies we are running a Hadoop command so it activates HDFS and jar specifies which type of application we are running and YouTube.jar is the jar file which we have created consisting of the above source code. The path of the Input file in our case is root directory of hdfs denoted by /youtubedata.txt and the output file location to store the output has been given as top5_out. This command immediately starts the MapReduce to analyze the youtubedata.txt dataset. Mapper code gets executed first and once it is 100% completed the reducer gets executed (as shown in Figure 10.)

The produced output is then sorted using the command hadoop fs -cat /top5_out/part-r-00000 | sort –n –k2 –r | head – n5

The final output is then displayed as shown in Figure11.

Figure 11 Problem Statement 1 Output

Problem Statement 2: To find the top 5 video uploaders on youtube

The mapper and reducer algorithm for this problem statement is very similar to that of Problem statement1.

Mapper Algorithm:

In this mapper code,the <k,v> pairs are associated as: key=uploader , and value=views where uploader is the username of the uploader and views is the number of views for the video.

These <k,v> pairs will be passed to the shuffle and sort phase and is then sent to the reducer phase where the total count(sum) of the values is performed.

We take a class by name TopUploader

We then extend the Mapper class which has the same arguments as the Mapper class in Problem Statement 1.i.e. <k(input),v(input)> and <k(output),v(output)>. We then declare an object 'uploader' which will store the username of the uploader.

Next we declare a variable 'views' which will store the video views. Then we override the map method so that it runs once for every line.

Next we declare a variable 'record' which stores the lines.

We then split the line and store them in an array. All the columns in a row are stored in this array.

We then store the up loaders' username.

Finally, we write the key and value, where key is 'uploader' and value is 'views'. This will be the output of the map method.

Reducer Algorithm:

We first extend the Reducer class which has the same arguments as the Mapper class .i.e. <k(input),v(input)> and <k(output),v(output)>.

Again,same as the Mapper code,we override the Reduce method which will run for all <k,v> pairs.

We then declare a variable 'totalviews' which will check all the values of the 'v' in the <k,v> pairs containing the same 'k'(key) value.

Finally, it writes the final <k,v> pairs as the output where the value of 'k' is unique and 'v' is the highest value obtained in the previous step.

The two configuration classes (MapOutputKeyClass and MapOutputValueClass) are included in the main class to clarify the Output key type and the output value type of the <k,v> pairs of the Mapper which will be the inputs of the Reducer code.

Execute

We access hdfs using the command below:

Command: hadoop jar Desktop/YouTube.jar /youtubedata.txt /topuploader_out

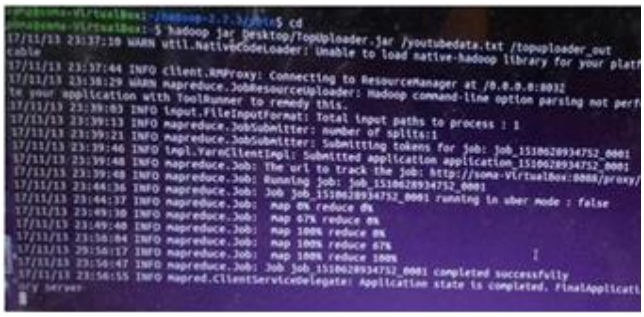

Figure 12 Execute Problem Statement 2

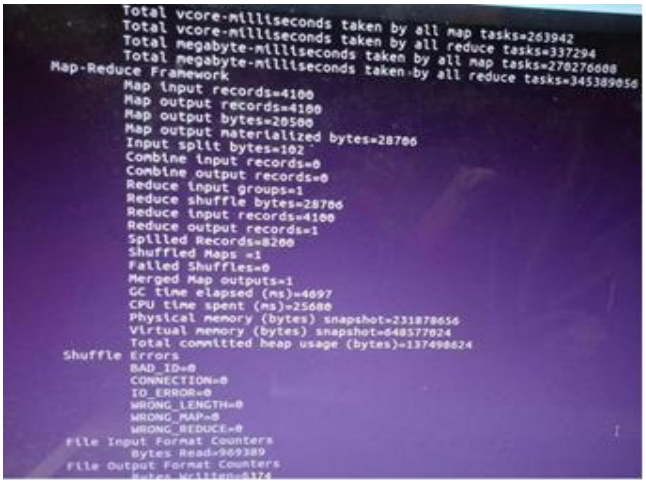

Figure 13 – MapReduce Framework

As seen above in Figure12, the mapping takes place first and the reducer starts only after map is 100% completed. After map and reduce are both 100% completed, the file system displays the number of bytes read from the input file on local disk and on HDFS and the number of bytes written on the output file on local disk and on HDFS,as seen in Figure13 and Figure14.

Figure 14-HDFS Directory where input file and output files are stored

Figure14 shows where input file, youtubedata.txt, is uploaded and stored and output files, top5_out (Problem Statement 1) and topuploader_out(Problem Statement 2) are created and stored.

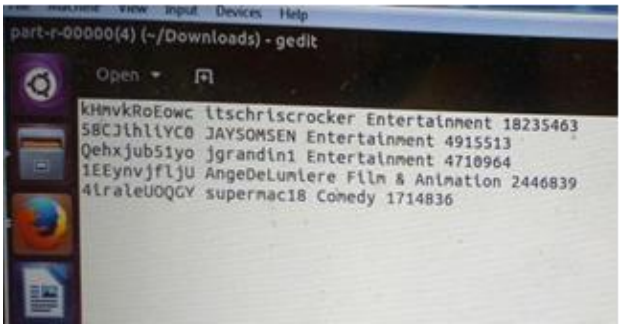

Figure 15 – Problem Statement 2 Output after downloading the output file part-r-00000 contained inside topuploader_out file in HDFS

3. Conclusion

Big data is not just an emerging trend but also a necessity. There has been a lot of investment in Big Data by various companies in last few years including Google, Amazon, Microsoft, and Facebook to increase their efficiency, strategy and competency. The measure of how successful a product will be majorly depends on public reaction. Decades ago companies would only use television as a medium to promote their products. Similarly, the movie makers would promote their films and songs through television media only. However, given our smarter and digitalized era today, companies use YouTube for marketing and promoting their products and brand by uploading their product advertisement video to YouTube and movie makers promotes their movies by uploading songs and movie trailers to YouTube. The measure of how well the product and movie is received by the public are determined by the number of views, likes (ratings) and comments on the video. This project intends to hit on those key areas which companies and organizations use or can use to measure their product's/movie's success against their competitors.

As seen from the methodology, the basic algorithm retrieves reports to better understanding and viewing statistics and trends for users' channel depending on the number of views and likes not only on their respective videos but also check if their competitors' are at the top. Another output result gives us insights on what categories of videos interest the public more. This can be done by analyzing the top video categories. This also helps budding YouTubers who upload YouTube videos to earn money. They can analyze the most popular video categories and upload videos accordingly to gain more views, more subscription and thus more money and popularity. As we have already seen above in depth, MapReduce is a very simple programming tool which makes use of basic programming languages like C, Python, and Java. These are languages which every programmer will be adept at, thus, it eliminates the need to hunt for a programmer specializing in a special language. Thus, my project on analyzing the huge YouTube dataset using Hadoop and MapReduce is well justified and successful.

4. Future Scope

This project can be further advanced by designing a MapReduce algorithm to perform sentiment analysis on YouTube video comments. Also, an algorithm on comment analysis can help analyze and identify the numbers of trolls harassing authentic users and spam users. Another useful advancement that can be done is to be able to present the analysis output graphically in the form of a histogram or a pie chart for easier and more efficient representation of the statistics. Almost all social media platforms allow paid advertising now. Designing a MapReduce algorithm for analysing how many users click on the ads or how many users enable or disable Adblock for their website can give a very interesting and important insight to the company. These algorithms can also be implemented on any social networking site promoting ads.

Acknowledgement

I would like to thank Prof Rajesh Bhise, an assistant professor in the Computer Science Department of Amity University Mumbai, for being my mentor in this project. He has supported my project idea, reviewed my paper and assisted me to make some necessary changes in my paper.

References

1. Webster, John. "MapReduce: Simplified Data Processing on Large Clusters", "Search Storage", 2004. Retrieved on 25 March 2013. https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

2. Bibliography: Big Data Analytics: Methods and Applications by Saumyadipta Pyne, B.L.S. Prakasa Rao, S.B. Rao

3. YOUTUBE COMPANY STATISTICS. https://www.statisticbrain.com/youtube-statistics/

4. Youtube.com @2017. YouTube for media. https://www.youtube.com/yt/about/press/

5. Big data;Wikipedia https://en.wikipedia.org/wiki/Big_data

6. Kallerhoff,Phillip. ―Big Data and Credit Unions: Machine Learning in Member Transactions https://filene.org/assets/pdfreports/301_Kallerhoff_Machine_Learning.pdf

7. Marr,Barnard.―Why only one of the 5 Vs of big data really matters http://www.ibmbigdatahub.com/blog/why-only-one-5-vs- big-data-really-matters

8. Resources Management Association (IRMA). 2016. Information. "Chapter 1 - Big Data Overview". Big Data: Concepts, Methodologies, Tools, and Applications, Volume I. IGI Global. http://common.books24x7.com/toc.aspx?bookid=114046

9. Apache Hadoop
10. http://hadoop.apache.org/
11. How To Analyze Big Data With Hadoop Technologies ; 3pillarglobal.com. 2017 https://www.3pillarglobal.com/insights/analyze-big-data-hadoop-technologies Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, in:
12. OSDI'04, 6th Symposium on Operating Systems
13. Design and Implementation, Sponsored by USENIX, in cooperation with ACM SIGOPS, 2004, pp. 137– 150
14. Big Data Tutorial1:MapReduce https://wikis.nyu.edu/display/NYUHPC/Big+Data+Tutorial+1%3A+MapReduce
15. MacLean,Diana.‖A Very Brief Introduction to MapReduce http://hci.stanford.edu/courses/cs448g/a2/files/map_reduce_tutorial.pdf
16. Edureka.'Install Hadoop:Setting up a single node cluster'. https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster

