Analysis of Youtube DataSets using Hadoop MapReduce

[1]RASHMI B, [2] POOJA G, [3] USHA G R

[1,2]B.E. Student, [3]Assistant Professor
Department of Information Science
SDM Institute of Technology, Ujire-574 240, INDIA.

***Abstract*—With rapid innovations and surge of internet companies like Google, Yahoo, Amazon, eBay and a rapidly growing internet savvy population, today's advanced systems and enterprises are generating data in a very huge volume with great velocity and in a multi-structured formats including videos, images, sensor data, weblogs etc. from different sources. YouTube is one of the most popular and engaging social media tool and an amazing platform that reveals the community feedback through comments for published videos, number of likes, dislikes, number of subscribers for a particular channel. YouTube collects a wide variety of traditional data points including View Counts, Likes, Votes, and Comments. The main objective of this project 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. The data is being collected from a particular URL. This data is then stored in HDFS (Hadoop Distributed File System) in a certain format. This data is further analyzed to obtain the final output, to describe that which is the most videos uploaded, view counts, those video's ranking according to YouTube Analytics.**

***Keywords*—Hadoop, MapReduce, HDFS (Hadoop Distributed File System.**

I. INTRODUCTION

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Hadoop is the tool used to manage and process the big data contents, which is the biggest challenge in the recent years. Some notable big users such as Yahoo, Facebook and Amazon use Hadoop. As data management and analysis are facing new challenges in the age of big data, the rationality and timeliness of the data processing methods are becoming the research hotspot of big data statistical analysis, and big data association analysis greatly increases the profits of enterprises.

The two core concepts of the Hadoop are MapReduce and Hadoop distributed file system (HDFS). HDFS is the storage mechanism and Map Reduce is a distributed processing framework.

II. HADOOP DISTRIBUTED FILE SYSTEM

Hadoop two fundamental subprojects are the HDFS and the MapReduce. The distributed file system named by Hadoop Distributed File System (HDFS) is a designed to run on commodity hardware [3]. The block size of HDFS is much larger than that of normal file system i.e. 64MB by default. HDFS protects data by replicating data blocks into multiple nodes, with a default replication factor of 3. One major usage of HDFS is which has very good durability.

HDFS can be presented as the master and slave architecture as shown in Fig. 1

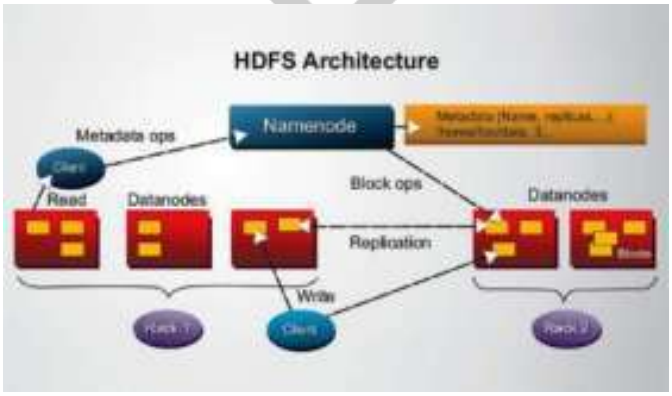


Fig.1 Hadoop Architecture

There is a master "NameNode" to keep track of overall file directory structure and the placement of chunks. This NameNode is the central control point and may re-distributed replicas as needed. DataNode reports all its chunks to the NameNode at bootup. Each chunk has a version number which will be increased for all update. Therefore, the NameNode know if any of the chunks of a

DataNode is stale (e.g. when the DataNode crash for some period of time). Those stale chunks will be garbage collected at a later time[3].

To read a file, It will calculate the chunk index based on the offset of the file pointer and make a request to the NameNode. The NameNode will reply which DataNodes has a copy of that chunk. From this points, the client contacts the DataNode directly without going through the NameNode.

To write a file, client API will first contact the NameNode who will designate one of the replica as the primary (by granting it a lease). The response of the NameNode contains who is the primary and who are the secondary replicas. Then the client push its changes to all DataNodes in any order, but this change is stored in a buffer of each DataNode. After changes are buffered at all DataNodes, the client send a "commit" request to the primary, which determines an order to update and then push this order to all other secondaries. After allsecondaries complete the commit, the primary will response to the client about the success. All changes of chunk distribution and metadata changes will be written to an operation log file at the NameNode. This log file maintain an order list of operation which is important for the NameNode to recover its view after a crash. The NameNode also maintain its persistent state by regularly check-pointing to a file. In case of the NameNode crash, a new NameNode will take over after restoring the state from the last checkpoint file and replay the operation log.

III. MAPREDUCE

Hadoop MapReduce is the most popular open source implementation of the MapReduce framework proposed by Google. MapReduce is a software framework for distributed processing of large datasets on compute clusters of commodity hardware.

MapReduce job mainly consists of three user-defined functions: *map*, *merge* and *reduce*. The input of a Hadoop MapReduce job is a set of key-value pairs (*k*, *v*) and the map function is called for each of these pairs. The map function produces zero or more intermediate key-value pairs (*k_*, *v_*). Then, the Hadoop MapReduce framework groups these intermediate key-value pairs by intermediate key *k_* and calls the reduce function for each group.

It is organized as a "map" function which transform a piece of data into some number of key/value pairs. Each of these elements will then be sorted by their key and reach to the same node[4], where a "reduce" function is use to merge the values (of the same key) into a single result are as shown in Fig.2 Parallel Algorithm for MapReduce.

Map stage: The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system *HDFS[3]*. The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

Reduce stage: This stage is the combination of the *Shuffle*stage and the *Reduce*stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

Example: The number of map tasks in a program is handled by the total number of blocks of input files. If we have a block size of 128MB and we expect 10TB of input data, we will have 82,000maps. Ultimately the number of maps is determined by the InputFormat.

The job execution starts when the client program submit to the JobTracker a job configuration, which specifies the map, combine and reduce function, as well as the input and output path of data[6].

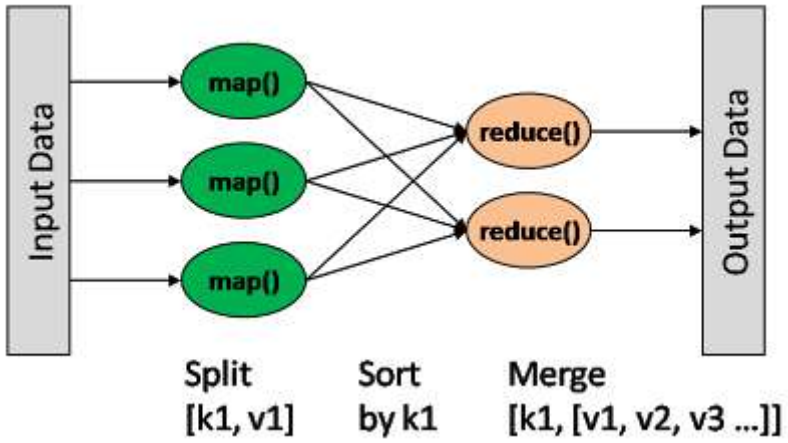


Fig.2 Parallel Algorithm for MapReduce

The JobTracker will first determine the number of splits (each split is configurable, ~16-64MB) from the input path, and select some TaskTracker based on their network proximity to the data sources, then the JobTracker send the task requests to those selected TaskTrackers as shown in Fig.3

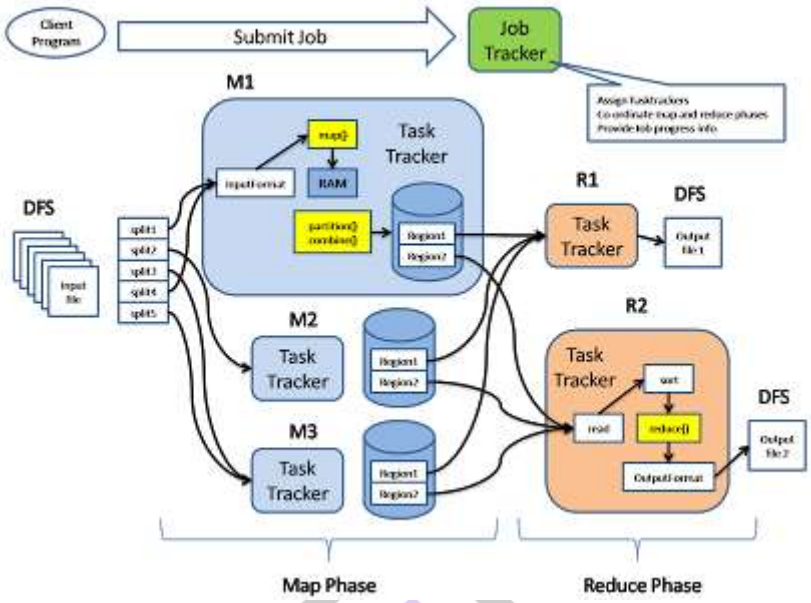


Fig.3 Hadoop MapReduce Architecture

Each TaskTracker will start the map phase processing by extracting the input data from the splits. For each record parsed by the "InputFormat", it invoke the user provided "map" function, which emits a number of key/value pair in the memory buffer.

A periodic wakeup process will sort the memory buffer into different reducer node by invoke the "combine" function. The key/value pairs are sorted into one of the R local files (suppose there are R reducer nodes). When the map task completes (all splits are done), the TaskTracker will notify the JobTracker. When all the TaskTrackers are done, the JobTracker will notify the selected TaskTrackers for the reduce phase.Each TaskTracker will read the region files remotely. It sorts the key/value pairs and for each key, it invokes the "reduce" function, which collects the key/aggregatedValue into the output file (one per reducer node).

Map/Reduce framework is resilient to crash of any components. The JobTracker keep tracks of the progress of each phases and periodically ping the TaskTracker for their health status. When any of the map phase TaskTracker crashes, the JobTracker will reassign the map task to a different TaskTracker node, which will rerun all the assigned splits. If the reduce phase TaskTracker crashes, the JobTracker will rerun the reduce at a different TaskTracker. After both phase completes, the JobTracker will unblock the client program.

IV. IMPLEMENTATION

The implementation of any system can be best done by its system design. The complete module of our analysis is shown below:

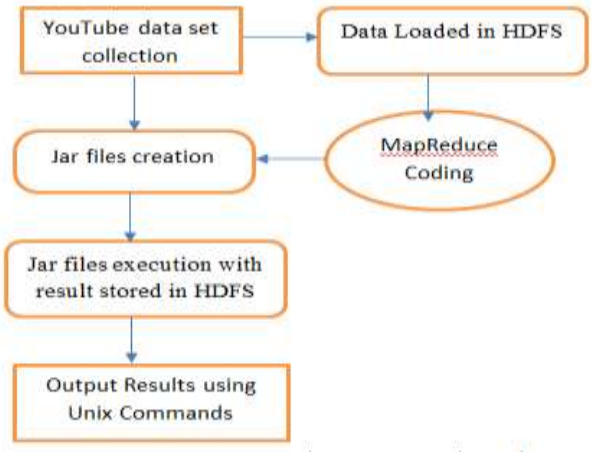


Figure 4. Data Flow Diagram

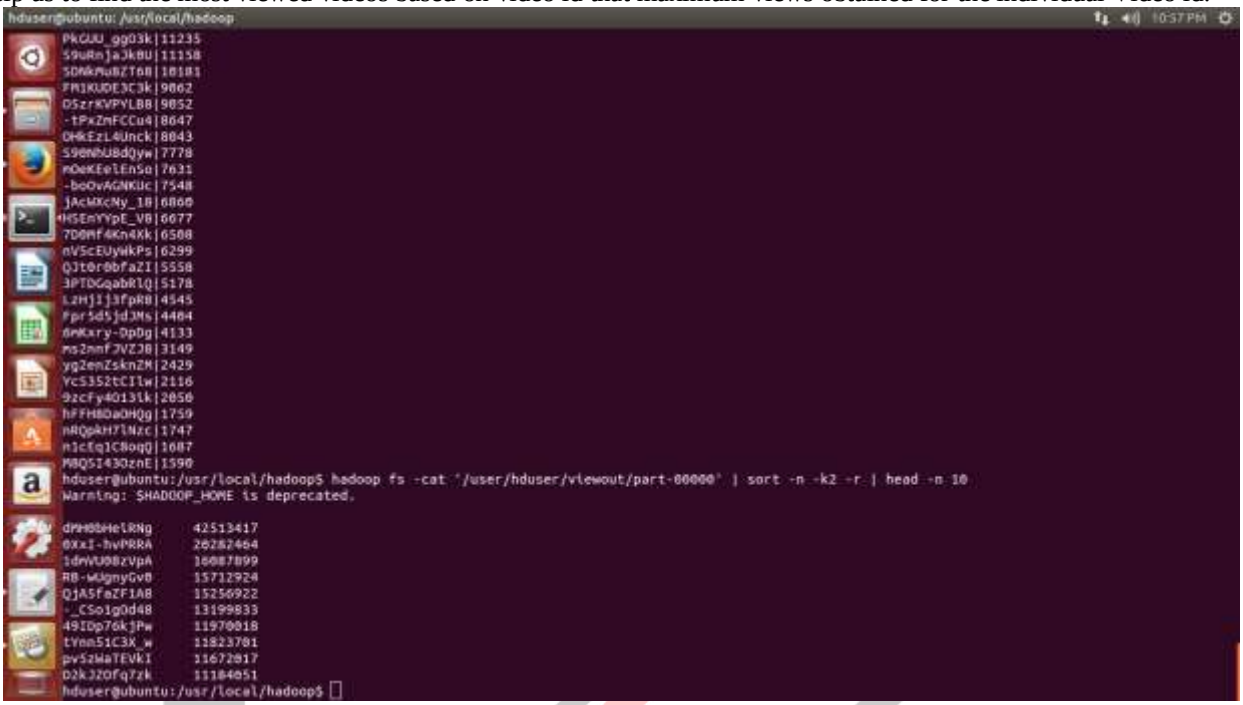
- In this project we collect YouTube sample Data Sets for data analysis using the URL https://drive.google.com/file/d/0ByJLBTmJojjzaXlsVWl0WEhPZDQ/view
- This data is then stored in HDFS (Hadoop Distributed File System) in a certain format.
- The data is analyzed by using MapReduce programming. Then the Jar files is to be created.
- The created Jar files are then executed and its result is stored in HDFS.
- We run the files using Unix Commands on Big Data through MapReduce to extract the meaningful output which can be used for analysis.

The analysis of YouTube Datasets using Hadoop MapReduce are as follows:

1. This Project will help to fetch the Top 10 Most Viewed Videos
2. To identify the Top 10 Length Videos.

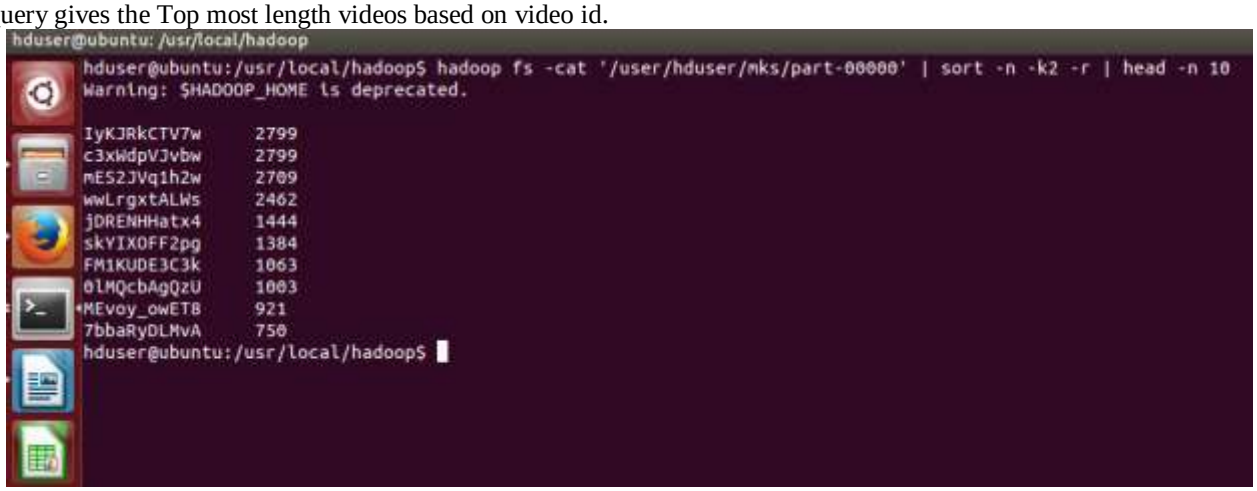
- **This Project will help to fetch the Top 10 Most Viewed Videos**

This help us to find the most viewed videos based on video id that maximum views obtained for the individual Video id.



- **To Identify the Top 10 Length Videos**

This Query gives the Top most length videos based on video id.



V. CONCLUSIONS

Hadoop and Map Reduce are used in our big data analytics and see what is the advantage of this rather using Hive and Pig. By considering different attributes analysis of youtube data is done based on name, age and on their nature. Further the youtube data has been set up for the Distributed analysis of large amount of the metadata, based on Hadoop framework. Considering the different

factors such as size of input data, Block size of the HDFS and number of nodes the performance and the efficiency of the Hadoop is analysed

REFERENCES

[1] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[2]**Wikipedia.org. 2016**. Big Data. *https://en.wikipedia.org/wiki/Big_data.* [Online] February 2016. https://en.wikipedia.org/wiki/Big_data.

[3]https:/www.dzone.com

[4] Apache Software Foundation. Official apache Hadoop website, http://hadoop.apache.org/, Aug, 2012.

[5] Jeffrey Dean and Sanjay Ghemawat, "MapReduce:Simplied Data Processing on Large Clusters" in Google, *Inc*.

[6] The Hadoop Architecture and Design, Available: http://hadoop.apache.org/common/docs/r0.16.4/hdfs_design. html,Aug, 2012.

[7] Gurav, Y.B., Jayakar, K.P., "Efficient Way for Handling Small Files using Extended HDFS", International Journal of Computer Science and Mobile Computing, Vol.3, pp.785-789, June 2014.