Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Big Video Data Analytics using Hadoop

Vatsal Shah
Department of Information Technology,
Thakur College of Engineering & Technology,
Mumbai, India

Abstract*—There is a plethora of data being collected in the world today. Although analysis of structured data involving textual and semi-textual elements has seen tremendous success in the past few years, analysis of large scale unstructured data in the form of video format still remains a challenging area of research. With the amount of video data being collected today, via surveillance cameras, digital devices and other recording devices present everywhere, this large scale video data requires in depth analysis, and may provide vital insights. However, analysis of large scale unstructured video data is faced with a number of issues.This paper attempts to discuss the current issues in large scale video analytics, and proposes solutions to the same using Apache's Hadoop platform.

***Keywords*— large scale video analytics, big data, Hadoop, analytics, unstructured data, data science**

I. INTRODUCTION

90% of all the data in the world has been created in the past two years [1], but 80% of that data is in unstructured format, in the form of videos and images. Currently most of the big data analysis revolves around structured and semi structured data, for example analysis of twitter feeds and LinkedIn data is fairly predominant at the moment. However, withvideo recording devices in the form of smartphones and surveillance cameras generating petabytes of data every day and online video archives like YouTube having over 300 hours of video data uploaded every minute, analysis of this large scale video data has become very important. Analysis of this video data may provide certain valuable insights which may prove useful for domains such as security, retail, traffic and even aiding in enabling content based video retrieval. The primary issue in performing video analytics is dealing with such massive quantities of unstructured data. Unstructured data is data that does not have a predefined data model or is not organized in a predefined manner [2]. It does not follow any conventional schema.

Apache Hadoop is a open source project that allows reliable, faster and distributed processing of large scale data. The major advantage of using a platform like Hadoop is its cost effectiveness. It enables distributed parallel processing of large scale data using inexpensive servers and eliminates the need for sophisticated and expensive proprietary hardware. The Apache Hadoop platform has proven successful in the past in the storage and processing of large scale unstructured data with technologies such as HDFS, MapReduce and HIVE.

The advantages of using Hadoop for storage and processing of large scale video data are:

1. Hadoop provides reliable computing across distributed systems, much required for big video data.
2. Hadoop provides scalability, which is an essential characteristic, considering the ever increasing size of the data being produced and stored today.
3. Hadoop is a universal platform, familiar with most users, eliminating the need for working with numerous proprietary software.
4. Hadoop is an open source project, which leverages current hardware in a way that minimizes cost while maximizing resource utilization.
5. Hadoop provides flexibility i.e. it enables storage and processing of all forms of data (structured and unstructured), which makes it suitable for efficient storage and processing of the big unstructured video data.

II. NEED FOR LARGE SCALE VIDEO ANALYTICS

All existing solutions to analyse video streams are based on real time event detection using computer vision algorithms, which help in detection of abnormal events, which in turn trigger alarms for a variety of user defined events involving vehicles, people and other objects [3].The core of such a system is the increasingly intelligent and robust video analytics that are capable of analysing the videos from low-level image appearance and feature extraction to midlevel object / event detection to high-level reasoning and scene understanding [4].But in order to analyse historic data, which may allow us to gather deeper insights into data, we need a sophisticated framework like Hadoop to solve our problems. A platform like Hadoop would allow engineers to run video analytics algorithms multiple times on the same data to elicit useful information. Traditionally, security officers would have to browse through hours of video footage to find a specific event, but by developing video analytics solutions using the Apache Hadoop platform, we may detect the event in a matter of minutes. Also, since most engineers would be familiar with a general unified platform such as Hadoop, it would save them the effort of coding for different proprietary platforms.

III. CURRENT ISSUES IN BIG VIDEO DATA ANALYTICS

Performing large scale video analytics using unstructured video datasets is a tough task posed with numerous challenges as stated below:

A. Making the compressed video data Hadoop friendly

Processing specialized file formats like video files using Hadoop is not easy. It requires the user to write a custom InputFormat and RecordReader that understands how to convert the video file into splits and then reads the splits back to values. Moreover, most video files are compressed using standard compression formats (MPEG-4, H.264) which makes the above task more complicated, since the MPEG compression format is not suitable for distributed systems like Hadoop. Therefore an effective mechanism to transcode the large scale video data into a Hadoop friendly format needs to be developed.

B. Leveraging the Hadoop framework in commonly used video analytics algorithms

The objective of most video analytics algorithms is to gain structured insights using unstructured data [5].Currently, most engineers would have to ensure scalability and parallelism of the video analytics algorithms in order to efficiently process the data. Although this problem could be solved by taking advantage of Hadoop's MapReduce architecture, incorporating the algorithm within it so that it can improve efficiency and speed is a challenge. Also it requires identification of specific components of the algorithm that can be implemented using MapReduce tasks.

C. Use of an advanced query language to ask complex questions and gain valuable insights

Using Hadoop is not easy for end users, especially for those users who are not familiar with MapReduce. End users have to write map-reduce programs for simple tasks. Hadoop lacks the expressiveness of popular query languages like the structured query language (SQL) and as a result users spend a lot of time in performing simple analysis [6]. Therefore in order to glean deeper insights from the data stored, it is of utmost importance to improve the querying capabilities of Hadoop by use of an advanced query language.An advanced query language will enable users to ask complicated questions, and gain valuable information.

IV. SOLUTIONS PROPOSED

A. Conversion of the compressed video data to a sequence file of image frames

As discussed above, video compression formats like MPEG are difficult to handle in a distributed framework like Hadoop. This problem can be avoided by developing a video transcoding system that can convert large scale video data into image sequences both quickly and efficiently. [7] proposes quick and efficient algorithms for generating spatially reduced images for a MPEG-2 video. However since the framework in use is Hadoop, the above task can be achieved using custom MapReduce jobs. Fig.1. describes the general flow for the conversion of the compressed video data into a sequence file of image frames using MapReduce tasks.

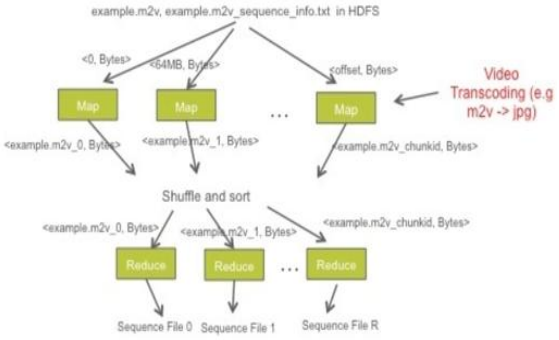

Fig. 1.Conversion of the compressed video data to a sequence file of image frames using Hadoop MapReduce jobs [10]

MapReduce (implemented on Hadoop) is a framework for parallel distributed processing large volumes of data. In programming using MapReduce, it is possible to perform parallel distributed processing by writing programs involving the following three steps: Map, Shuffle, and Reduce [8].

A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks [9].

The Map function traditionally requires a (Key,Value) pair as an input and it produces a corresponding (Key',Value') pair as an output.

Map(k1,v1) → list(k2,v2)

Reduce(k2, list (v2)) → list(v3)

In order to perform distributed parallel processing of the video data, the data is split into several parts and key-value pairs need to be generated for the same. However, generating key value pairs for unstructured data is not easy.

The Hadoop Distributed File System (HDFS) stores large data in the form of blocks (generally of size 64MB) distributed across a cluster of nodes. When the input file is a video file (a bitstream compressed using MPEG) it is split into many

blocks, and each Mapper process needs to interpret eachbitstream block separately in order tocreate individual blocks of decoded video frames, which can be used for further analysis.
In order to overcome this, the following methods may be adopted:

1. Multiple MapReduce tasks may be defined to perform the above task in the following manner:
   i. MapReduce Job1: This MapReduce job would extract the sequence information of the first block of the stored video data compressed in MPEG format and output the result in a text file that can be stored in HDFS.
   ii. MapReduce Job2: This MapReduce job would decode the individual blocks using the sequencing information generated by MapReduce Job1 and produce a sequence file of images decoded from the blocks of video data. This sequence file of images could now be processed using Hadoop and considered Hadoop friendly.

2. Another possible solution to the above issue could be using Xuggler[11], which is an open source project that uses the FFmpeg [12] media handling libraries,hence playing the role of a java wrapper around them. It is the easy way to uncompress, modify, and re-compress any media file (or stream) from Java.

The above processes would help in converting the compressed video files into a sequence of image sequences thus making the data Hadoop friendly and hence solving the issue discussed above.

B. Storage and parallel processing of the transcoded image files using HIPI (Hadoop Image Processing Interface)

Once the encoded video frames have been decoded to a sequence of images, the use of HIPI is proposed. This would solve the problem of leveraging the Hadoop framework in use of video analytics algorithms (problem discussed above).
HIPI is a Hadoop based image processing interface for performing image based Map-Reduce tasks[13]. Computer vision algorithms that use the current Hadoop MapReduce framework require a very high degree of complexity. The HIPI interface provides a flexible platform for all existing video analytics or computer vision algorithms, while accommodating them within Hadoop's MapReduce framework. It provides both flexibility and reliability, allowing engineers to successfully run their preferred algorithms on the image sequences generated in preceding steps, hence solving the issue.

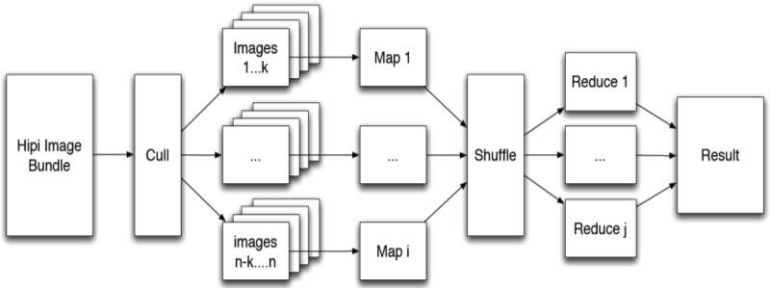

Fig.2. Organization of a HIPI/MapReduce program [13]

HIPI provides a simple and clean interface for fast, distributed image processing on the MapReduce platform.
This allows the programmers to focus on building the analytic algorithm more, and allowing the platform to handle the rest. Prominent tasks achieved through computer vision algorithms like object detection, face recognition, anomaly detection and feature extraction can be achieved easily using the MapReduce framework and it requires little effort to develop architectures for parallel processing. The results of the image processing can be output back to the HDFS.

C. Use of HIVE (an open source datawarehousing solution) to run advanced queries on Hadoop in order to answer complex questions

HIVE is a popular datawarehousing software that is built over the Hadoop framework, which allows users to effectively query large scale data stored on distributed platforms such as Hadoop. HIVE supports a simple query language called QL (HIVE-QL) which is a SQL like query language that allows users familiar with the syntax and structure of SQL to efficiently query datasets stored using the Apache Hadoop framework in a similar manner.

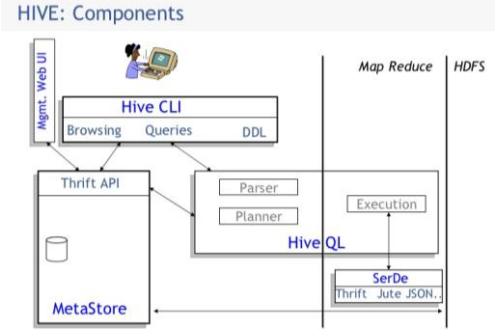

Fig.3.Components of HIVE

Traditional SQL features like from clause subqueries, various types of joins – inner, left outer, right outer and outer joins, cartesian products, group bys and aggregations, union all, create table as select and many useful functions on primitive and complex types make the language very SQL like. In fact for many of the constructs mentioned before it is exactly like SQL. This enables anyone familiar with SQL to start a hive command line interface and begin querying the system right away [6].

Fig.3. shows the most important components of the HIVE platform. HIVE-QL also allows complex analysis performed using MapReduce programs to be expressed in the form HIVE-QL queries. This facilitates users familiar with the MapReduce framework to run their programs easily.

A complex query language like HIVE-QL, therefore solves the problem of performing complex analytical operations on Hadoop. Such a query language would allow users to gain valuable insights from the large scale video data. Users may simply write "SQL like" queries in order to extract valuable information. Existing solutions are focused on performing real time analytics on using video recognition and other methodologies. In case any historic information needs to be extracted from such data, users would need to manually browse through hours and days of video footage in order to find a specific event.

With HIVE-QL the users can easily query the database in order to find specific events in history or even perform sophisticated analysis on the video data. For example, in the case of the theft of a car from a parking lot, security personnel would generally have to look through hours of video footage in order to identify the stolen car and possibly identify the criminal. However, with the help of HIVE-QL, the users might just need to run a simple query, and the results would be displayed in a short interval of time. They may even find answers to complex questions like the speed, color and time of movement of the car in a matter of seconds.

V. CONCLUSIONS AND FUTURE SCOPE

There is a need for a reliable and efficient mechanism for storage and processing of large scale video data today. With the amount of video data being collected every day, analysis of this data might provide valuable insights, which may aid organizations and governments in many ways. An efficient video processing analyzing system will not only help in analyzing existing video archives, but also move towards scaling existing video processing algorithms to frameworks such as Hadoop. Thispaper has discussed the need for performing efficient big video data analytics in today's world. Moreover, the paper has also discussed the issues and challenges involved in performing analysis of the large scale video data using Hadoop, and finally proposed solutions to the same.

REFERENCES

[1] SINTEF. "Big Data, for better or worse: 90% of world's data generated over last two years." ScienceDaily. ScienceDaily, 22 May 2013. <www.sciencedaily.com/releases/2013/05/130522085217.htm>.

[2] https://en.wikipedia.org/wiki/Unstructured_data

[3] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust RealTime Unusual Event Detection Using Multiple Fixed-Location Monitors," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555-560, Mar. 2008.

[4] L.-Q.Xu "Issues in video analytics and surveillance systems: Research/prototyping vs. applications/user requirements", Proc. IEEE Conf. Adv. Video Signal Based Surveill. (AVSS), pp.10 -14 Sep. 2007

[5] http://blog.pivotal.io/data-science-pivotal/features/large-scale-video-analytics-on-hadoop

[6] A. Thusoo, J. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Anthony, H. Liu, and R. Murthy. Hive - a petabyte scale data warehouse using hadoop. In ICDE, pages 996–1005, 2010.

[7] Song, Junehwa, and Boon-Lock Yeo. "Fast extraction of spatially reduced image sequences from MPEG-2 compressed video." Circuits and Systems for Video Technology, IEEE Transactions on 9.7 (1999): 1100-1114.

[8] Yamamoto, Muneto, and Kunihiko Kaneko. "Parallel image database processing with MapReduce and performance evaluation in pseudo distributed mode." International Journal of Electronic Commerce Studies 3.2 (2013): 211-228.

[9] http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[10] http://blog.pivotal.io/data-science-pivotal/products/using-hadoop-mapreduce-for-distributed-video-transcoding

[11] Xuggler, http://www.xuggle.com/xuggler/.

[12] FFmpeg, https://www.ffmpeg.org/.

[13] Sweeney, Chris, et al. "HIPI: a Hadoop image processing interface for image-based mapreduce tasks." Chris. University of Virginia (2011).

