Storage and Access in Product Review System using Hadoop

[1]Praveen Kumar K R, [2]R. Aparna

P G Student, S.I.T Tumkur , Associate Professor, S.I.T Tumkur
Email: [1]praveenkr337@gmail.com, [2]raparna@sit.ac.in

Abstract— With rapid development of Product Review System(PRS) web application, amount of images, audio, video and text based user profiles, product details, ratings, likes/dislikes, comments etc, being uploaded to the internet is rapidly increasing and difficult to store and process the structured and unstructured data. How to store, manage and process huge amount of big data effectively and provide an excellent experience for users has been an important problem. The goal of this work is to build and analyze a scalable and dynamic big data processing system, including storage, execution engine and query language. This paper presents a Hadoop-based structured and unstructured data storage and access architecture in PRS web application. The core idea of this architecture is to transcode video files using Xuggler and FFMPEG and store in HDFS, location of such data and product based structured data is stored in HBase. To access mulitimedia files in HDFS, we use WebHDFS it supports complete file system interface for HDFS and give full bandwidth of hadoop cluster for streaming data. The experimental results show that the approach can achieve a better performance.

Index Terms— Big Data, Hadoop, HDFS, HBase, Multimedia.

I. INTRODUCTION

Big Data contains exponential growth and availability of data, both structured and unstructured. It collects large data sets whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract the value.

Now a days, with the development of social network and video sites, the amount of images, audio and videos being uploaded to the internet is rapidly increasing, with Facebook users uploading over 300 Million new photos every day, 100 hours of video are uploaded to YouTube every minute. However, applications that make use of this data are severely lacking. How to store, manage, process, access and classify the mass network images, audios and videos effectively and provide an excellent experience for users has been an important problem.

A common notion about the application which consumes or analyzes Big Data is that they require a massively scalable and parallel infrastructure. The hopeful vision is that organizations will be able to take data from any source, harness relevant data and analyze it to find answers that enable in 1) cost reductions, 2) time reductions, 3) new product development and optimized offerings, and 4) smarter business decision making.

Apache Hadoop[1] is an open source implementation of Google's Map-Reduce model, which consists of (Hadoop Distributed File System) HDFS, Map Reduce, HBase, Phoenix and Zoo Keeper and other projects. It became extremely popular over the years for building Big Data analytics platform. Hadoop allows organizations to load, store and query massive scale of data sets on a large grid of inexpensive servers, as well as execute advanced analytics in parallel. It is developed for reliable, scalable, distributed computing and storage. HDFS, which is inspired by Google file system, is a representative for Internet service file systems running on clusters and is widely adopted to support lots of Internet applications as file systems. HBase[2] is a distributed, fault-tolerant, highly scalable, no-SQL database, built on top of HDFS to create a massively scalable and highly performance platform for dealing with heterogeneous data including non-textual data types (blob, clob etc.). Here, in this paper, we describe performance evaluation of a hybrid architecture. The HBase contains the information regarding the data storage location, whereas the actual data in the form of image files, audio and video files (non-textual data) are stored in HDFS. While evaluating performance, we perform random reads from HBase to retrieve location information and then extract the actual data from the HDFS. There is no support for SQL query language in HBase. However, there is a Phoenix SQL skin over HBase delivered as a client-embedded JDBC driver targeting low latency queries over HBase data. Phoenix takes SQL query, compiles it into a series of HBase scans, and orchestrates the running of those scans to produce regular JDBC result sets.

The aim of our research every day, new product is introduced to make everyone's life easy and luxurious. Most of the consumers purchase products online and the manufacturers usually provide the details about the products in their websites. Product reviews provides

thousands of objective third-party products to help our consumers make informed purchasing decisions. In PRS, web application contains both structured and unstructured data. Structured data contains user reviews in the form of likes, dislikes, ratings, comments, user profiles generated and stored in HBase database. Structured data has the advantage of being easily entered, stored, queried and analyzed. The Product owner uploads product details and multimedia files include images, audios, videos and other types of resources, and the format of video files transcoded using Xuggler[6] and FFmpeg[7] due to browser issues and stored in HDFS. To perform fast and inexpensive big data analytics, we use a processing system represented by a stack of frameworks for data storage (distributed file system, such as HDFS), data processing(execution engine, such as MapReduce [3]),data manipulation (query language over HBase, such as Phoenix SQL [4]), and data streaming(Hadoop Streaming, such as WebHDFS[5]) deployed over a large distributed system (such as hadoop cluster). In the context of the data explosion phenomenon, existing performance models for MapReduce are applicable for specific production workloads to process tens of terabytes of data. Therefore, the goal of my research is to optimize the MapReduce processing system for processing terabytes of data.

This project aims to display product advertisement in the form of video, audio and image and review responses for the same, which inturn help both users and manufacturers in web application. Product reviews take advantage of the latest web technologies, social media standards and the power of the product integrated community platform to deliver a highly flexible reviews system.

The rest of paper is organized as follows section I describes introduction, section II describes related work, section III describes proposed work and section IV describes results.

II. RELATED WORK

Most of the PRS websites like Consumer Search, Mouthshut, Epinions, Cnet, Choice etc, can only view ratings, comments about product and consumers cannot post comments, likes and dislikes and the reviews are moderated. Consumers find it difficult to review and buy the product.

In PRS we are storing and accessing large amount of structured and unstructured data. To process big data with high performance, there has been a lot of research activity in the areas of Hadoop/HDFS and Hadoop/HBase.

Hyeokju Lee[8], proposed multimedia data conversion using JAI library and the data stored in HDFS. JAI library handles only image data transcoding, but its difficult to transcode video and audio files. One of the closest methods of our solution is proposed by Hua Luan[9], It proposes a parallel technique for improving 3-D models of storage and accessing architecture. But researcher has implemented image file storage and accessing, where as video files and other structured data process to difficult in 3d models. Chen Zhang[10], proposed multi-row distributed transactions with global SI scans over HBase. This paper similar to our approach and suggested for future work of more complex database queries. Moreover large data access in HBase and its difficult to query in NoSQL language. Further optimize to overcome the drawback of existing schemes;

III. PROPOSED WORK

In our proposed work, Fig. 1 represents storage and access architecture, it consists of 3 modules in PRS web application. Users are generating an enormous amount of data in website, posting information through status updates, likes, dislikes, star ratings, comments, user profiles, product details etc. Product vendors upload unstructured data which contains multimedia datasets like audio, video and images they are up to a few terabytes in size. Large scale processing of such data needs a distributed framework such as Hadoop Map Reduce, where computational resources could easily be stored and accessed. With the knowledge that storage and access architecture would be used for process Big Data, we designed and overcome existing problems with the following goals.

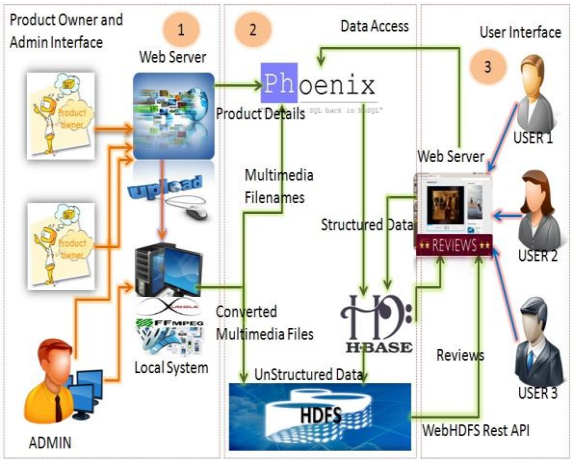

Fig. 1. Storage and Access Architecture

A. Product Owner and Admin Interface

The Admin and Product Owner interface is a unified interface providing functionality and interactive interface which can be used to upload product details and multimedia files, download files from the Internet and browse files. Product details can be store directly into HBase database with help of Phoenix SQL which gives fast performance where as multimedia files store in local system for transcoding video files due to browser issues. Many of the video formats are not supported in web browser because target device may not support the format or has limited storage capacity, reduced file size, low

quality of video etc. For such issues, we will use Xuggler API and FFmpeg to convert audio and video files in different formats, it gives full functionality of transcoding video conversion.

Video Transcoding using Xuggler and FFmpeg

Multimedia files [8] transcoding can be done with advanced technology. Xuggler uses the very powerful FFmpeg media handling libraries under the hood, essentially 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. It converts image, audio and video files in different formats with high performance and more features in server side transcoding process in PRS.

B. Storage and Access Business Process layer

For performance evaluation, we are focusing on measuring the total time to transcode a data set of multimedia files. We have proposed the solution that, video transcoding becomes smart and speeds-up due to the efficiency of Xuggler library and dynamically stored in HDFS where as multimedia file names store in HBase. In order to process such large-scale [9] video, audio and image datasets we are using the Hadoop MapReduce framework. We will dive further into the distributed video transcoder part of the framework that ingests the video into Hadoop, decodes the bit stream chunks in parallel and produces a sequence file (which is much more amenable for video analytics in Hadoop). Hadoop framework stores large files in a HDFS as small chunks of certain block size (typically 64MB) across a cluster of commodity machines. Image, audio and video datasets can store in different path locations in HDFS, it helps to easily retrieve the data in random access.

Given this framework, when the large input file to be processed and split into 64MB chunks. However, when the input file is video file (bitstream) and split into many chunks, each Mapper process needs to interpret the bitstream chunk appropriately to provide access to the individual decoded video frames for subsequent analysis. Map/Reduce framework use hierarchically parallelize operations on a Big Data set. The Map/Reduce framework will take the responsibility of multiple operations, including deploy the Map and Reduce methods to multiple nodes, aggregate the output from the map methods, passing the same to the reduce method, taking care of fault tolerance in case if a node goes down etc. In addition, our system improves the distributed processing capabilities and simplifies system design and implementation by incorporating data replication, fault tolerance, load balancing, file splitting and merging policies provided by Hadoop.

Et al. [10] the database queries are more complex, to overcome the more complexity, we use Phoenix SQL as fast performance due to use of server side coprocessor for aggregation, query parallelization for storing and accessing HBase database. Phoenix uses a Skip Scan for intra-row scanning which allows for significant performance improvement over Multi Gets and Range Scan when rows are retrieved based on a given set of keys. The number of rows to be scanned with a large range for data items read frequently.

C. User Interface

Whenever user requests the web page it retrieves the structured data from HBase using Phoenix SQL queries and display the details into web browser. Authenticated users can post the reviews in the form of ratings, like/dislikes, and comments for individual products; this data can store and parallel process in database. From these reviews we calculate most popular products, featured products, recent products, average ratings, number of like and dislike counts, analytics on comments and parallel we display product details in web browser.

In our proposed work more extension of user profiles, multimedia file names, product details, admin and manufacture profiles, ratings, comments, likes and dislikes values in our PRS web application of large scale data stored in HBase[10]. In this paper, we took challenge to improve HBase data query performance by improving the communication time. We also describe performance evaluation of hybrid architecture. The HBase contains the information regarding the data storage location, whereas the actual data in the form of image files (non-textual data) are stored in HDFS. For accessing data stored in HBase and HDFS. We use Phoenix SQL API for fast performance data retrieve in HBase and multimedia files can retrieve from HDFS by appending file names in WebHDFS REST API.

PRS Using Hadoop for Video Streaming:

Hadoop streaming is an utility that comes with Apache Hadoop and the utility allows to create and run map/reduce jobs. Many video formats are found on the web including Windows Media (.wmv), RealMedia (.rm), QuickTime (.mov), MPEG, Adobe Flash (.flv), etc. In order to display a video, we used jPlayer, which can be incorporated in the Web browser. jPlayer allows rapidly weave cross platform audio and video into web pages. Streaming occurs when the video file is split into fragments which are sent from the Web server to the player, giving the illusion of a continuous stream. From the user point of view, it looks as if a window is swept over the video content, saving the need of a full initial download of the whole file. Obviously the Fig. 2 illustrates that streaming is a more involved method because it requires strong coordination between the components involved in the process, namely the player, the Web server, and the file system from which the video is retrieved. WebHDFS provides read and write access, it

supports complete HDFS File system. WebHDFS uses the full bandwidth of the hadoop cluster for streaming data and gives better performance to display video in web browser.

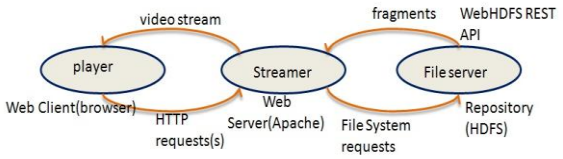

Fig. 2. Video Streaming using WebHDFS

Finally, To overcome this drawback of other review systems our new PRS plays an important role in collecting and displaying consumer assessments of products, services or experiences. Customer post reviews likes, dislikes, ratings, comments etc. Display product details and advertisement in the form of audio, video and graphics, these reviews helps consumer's to make quick purchasing decisions and manufactures to make good decisions like whether to add extra features or not, in which area to concentrate more for marketing the product.

IV. EXPERIMENT RESULTS

The experiment platform deployed on a Hadoop cluster having one master and eight slaves which serve as one NameNode and eight DataNodes. Each Hadoop cluster node is running on the UNIX OS (Debian 3.2.46 x86 64). The hardware platform for the master was Intel-Itanium (II) processor with core -i5 CPU and 5GB of RAM with 64GB registered ECC DDR memory and 3 TB SATA-2. All slaves were dual core machines with 3GB of RAM and 64-bit architecture with 320GB SATA-2. The web application is deployed on Tomcat web engine running on a PC machine. All machines under Hadoop cluster were running under 64-bit Unix operating system connected by the same switch. To benchmark the performance, the latest stable releases of Hadoop-1.20.1, HBase-0.94.0, Phoenix-2.1.2, Xuggler-5.4, FFMPEG 1.0.6, Java 1.6.0.37, and Apache Tomcat 7.0.39 were chosen. Daemons running on master node include NameNode, Secondary Name Node, Job Tracker, HMaster and HQuorum Peer (ZooKeeper). Daemons running on slave machine include Data Node, Task Tracker, and HRegion Server. In order to verify the performance for our transcoding functions on video transcoding process.

Table 1. Video data sets for performance evaluation

Parameter	Original video	Transcoded video file	Transcoded video file	Transcoded video file	Transcoded video file	Transcoded video file
Codec	MPEG-4	Xvid	X-flv	MPEG	X-flac+ogg	VP8
Container	MP4	AVI	FLV	MPG	OGV	WEBM
Size	100MB	60MB	70MB	75MB	40MB	60MB
Duration	3 min 46s	3 min 46s	3 min 46s	3 min 46s	3 min 46s	3 min 46s
Resolution	1280 x 720	840 x 360	840 x 360	840 x 360	840 x 360	840 x 360

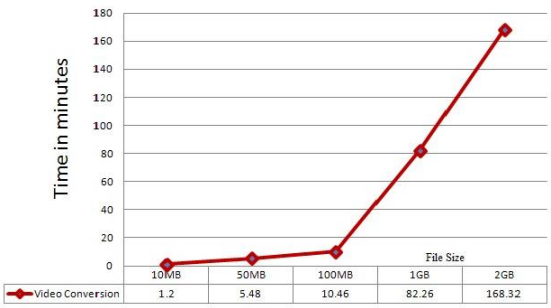

Fig. 3. Converted videos using Xuggler API and FFMPEG with time duration.

Table I demonstrates that the time consumptions fluctuate a little when file size increases and meanwhile the durations are fixed. The time consumption of video transcoding depends principally on the duration of video files rather than their sizes. It suggests that duration-based splitting mechanism would be more controllable than the size-based method. Graph in Fig. 3, shows that performance of faster conversion and higher quality of each video file is converted with full control over the dimension, frame rate, bit rate etc. with different formats in specific time.

Performance evaluation of Phoenix SQL

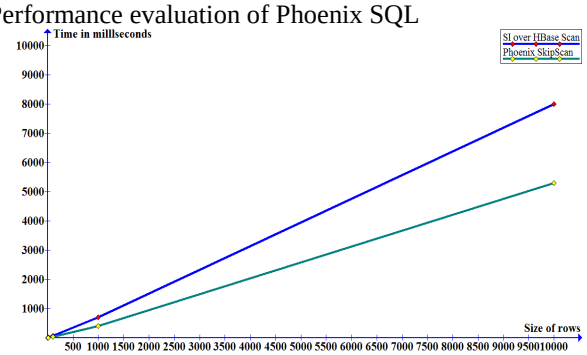

Fig. 4. Comparison of Phoenix SQL scan and SI over HBase scan time for scanning a row range in an HBase table.

Fig. 4 shows the comparison of Phoenix SQL and Snapshot Isolation (SI) over HBase. The time spent in scanning an number of rows in a single table and the table metadata is stored in an HBase table are versioned, such that snapshot queries over less prior versions, in case of Phoenix performs faster to scan the number of rows in HBase table than the SI over HBase scan.

V. CONCLUSION AND FUTURE WORK

The advent of internet and growing number of people uploading multimedia files and text based data in the PRS web application, there is an increase in the amount of data has put heavier burden on the internet infrastructure to process structured and unstructured data. In this paper, we have designed a storage and access architecture in PRS using hadoop technology to process big data and also implemented to transcode the video files using Xuggler API and stored in HDFS. We use WebHDFS to access complete file system of HDFS, it gives full bandwidth of hadoop cluster for streaming data. The experiments demonstrate that it can effectively reduce the cost and improve the efficiency of large amount of data.

As for future work, we will develop more mass multimedia files, big data analytics, and evaluate the performance of HBase and HDFS to come up with a highly optimized and scalable design.

REFERENCES

[1] Apache Hadoop, http://hadoop.apache.org/

[2] Apache HBase, http://hbase.apache.org

[3] MapReduce, http://en.wikipedia.org/wiki/Mapreduce.

[4] Apache Phoenix, http://phoenix.apache.org/.

[5] WebHDFS, http://hadoop.apache.org/docs/r1.0.4/webhdfs.html

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

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

[8] Hyeokju Lee, Myoungjin Kim, Joon Her, Hanku Lee, "Design and Implementation of Map Reduce-based Image Conversion Module in Cloud Computing Environment" Proc. of Int. Conf. on Advances in Computing, Control, and Telecommunication Technologies 2011.

[9] Hua Luan, Mingquan Zhou "Parallel Techniques for Improving Three-dimensional Models Storing and Accessing Performance" 2013 Ninth International Conference on Natural Computation (ICNC).

[10] Chen Zhang, "Supporting Multi-row Distributed Transactions with Global Snapshot Isolation Using Bare-bones HBase" 11th IEEE/ACM International Conference on Grid Computing-2010

[11] Mehul Nalin Vora "Hadoop-HBase for Large-Scale Data" 2011 International Conference on Computer Science and Network Technology.

[12] B. Dong, J. Qiu, Q. Zheng, X. Zhong, J. Li and Y. Li. A novel approach to improving the efficiency of storing and accessing small files on hadoop: a case study by powerpoint files. 2010. IEEE International Conference on Services Computing, SCC, IEEE (2010), pp. 65–72.

[13] Jian Huang, Xiangyong Ouyang "High-Performance Design of HBase with RDMA over InfiniBand" 2012 IEEE 26th International Parallel and Distributed Processing Symposium.

[14] Shanthi.B.R, Prakash Narayanan.C, "Dynamic Resource Allocation And Distributed Video Transcoding Using Hadoop Cloud Computing" Vol.2, Special Issue 1, March 2014, IJIRCCE.

[15] Haoyu Xu, Liangyou Wang,Huang Xie, "Design and Experiment Analysis of a Hadoop-Based Video Transcoding System for Next-Generation Wireless Sensor Networks" International Journal of Distributed Sensor Networks Volume 2014, Article ID 151564.

[16] Dipali Patil, Snehal Patil "Excerption of User Profile from Web Log Data using Hadoop Framework" Volume 3, Issue 4, April 2013, IJARCSSE

◈◈◈