International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

Efficient Storage of Similar Images in Big Data

Ramanujan A[1], Subramanya Swamy V [2]

[1]*Department of ISE,BMS College Of Engineering, Bangalore ,Karnataka ,India*
[2]*Department of ISE,BMS College Of Engineering, Bangalore ,Karnataka ,India*

Abstract –*There is a huge increase in data in terms of volume and variety which given rise to "Big Data" concept. This paper deals with storing and retrieving of similar image with few changes using Hadoop Framework. We use map reduce programming model and Hadoop HDFS to store similar Images. We divide the Image into chunks and then we generate key- value pair for each chunk and then compare it with chunks present in HDFS. If the chunk exists then it's skipped else the chunk is added to HDFS thereby reducing the space occupied by the Images. This paper evaluates the design and the prototype of the proposed system and it's performance. This is found useful in social networking sites[1] and in applications where huge number of Images are stored and retrieved.*

1. INTRODUCTION

Data has become an integral part in today's modern world. Data is collected knowingly or unknowingly for example let us consider Amazon shopping website which would collect your data of searching for various products and then display it as advertisement while browsing which is an example of data being collected unknowingly and Google Maps would collect data of location which is collected knowingly. The data collected from single person would amount to megabytes and for millions of users it would amount to peta bytes of data .The number of netizens are growing rapidly everyday and the volume of data grows drastically from time to time and also the type of data collected would vary .So we need new methodologies to process the data to get the required information which would meet the requirements such as processing huge amount of data and different types of data. Big Data is the term which we use to specify the above type of data .These data which is not easy to categorize and process using the relational data bases where queries are written to extract the required information can be solved using new methodologies and technologies . Hadoop is a framework for distributed computing which consists of two main components .

1.HDFS(Hadoop Distributed File System)

HDFS is a file system which stores all the information similar to the various file systems that exist in operating systems. It helps in the distributed processing.

2.MapReduce

It is a programming model where the data is divided into the chunks and key value pairs are generated for these chunks and later the key value pairs are processed to get the required information.

2. RELATED WORK

[2]The Big Data community focuses on Map Reduce and has been one of the key approaches for compromising on computing resources required by huge data sets. Map Reduce allows for high scalability and provides execution over massively parallel and distributed nodes. This paper identifies Map Reduce issues and challenges in handling Big Data and identifies opportunities for further research. The challenges are categorized based on data task types:

- Storage of data
- Analytics corresponding to big data
- online processing of data
- Privacy& security.

The paper tries to solve the above challenges with the use of map reduce.

[5] Map Reduce is a framework for distributed computing of batch jobs. Simplification of fault tolerance is done by materializing the entire output of each map and reduce task before it can be processed further .This paper gives a better Map Reduce architecture that allows data to be pipelined between the operators. This feature would improve Map Reduce model by decreasing the latency of the operations performed. Hadoop Online Prototype (HOP) supports for unceasing queries, enabling Map Reduce programs for applications such as event monitoring and stream processing. The costs of the network resources i.e. the network energy costs are reduced to a great extent.

[8] Various Organizations focus on data analytics to improve their organizational processes, they help the organization in making very crucial decisions and provides for new policies to be formed for better functioning of an organization. Scalability and latency of data are the two crucial parts in data analytics. The Map reduce framework provides for the above requirements for processing the data. There are disadvantages pertaining to Map Reduce with respect to the analytical tasks. The paper emphasizes on parallel query processing of queries using the map reduce framework and finds the glitches in the MapReduce framework and provides the appropriate solutions for solving the problem .Taxonomy is used for classifying based on the problems faced.

[9] This paper tells about the enhanced version of Phoenix, a Map Reduce framework for shared-memory.

Phoenix has a lot of disadvantages

•Uniform consequent storage of key value pairs.

•Combiner implementation is not up to the mark.

•poor task overhead fails to support a wide range of applications
An enhanced version of phoenix i.e. Phoenix++ provides for an efficient pipelining that would support various applications on the run. Compared to Phoenix, Phoenix++ has a better performance.

[10] Concurrency can be achieved by using Dynamic runtimes in parallel programming.
Implementing a dynamic runtime system in parallel programming system is very hard. Optimizing Phoenix, for shared-memory multi-cores and multiprocessors, is performed using a multi-layered approach that includes optimizing the algorithm used, improving the implementation, and OS communication leading to huge speed improvement.
The paper focuses on the following topics
•Identification of the obstacles that would limit the scalability of parallel runtimes on shared-memory systems.
•OS scalability on the large scale systems.

3. PROSPOSED SYSTEM

3.1 Block Diagram

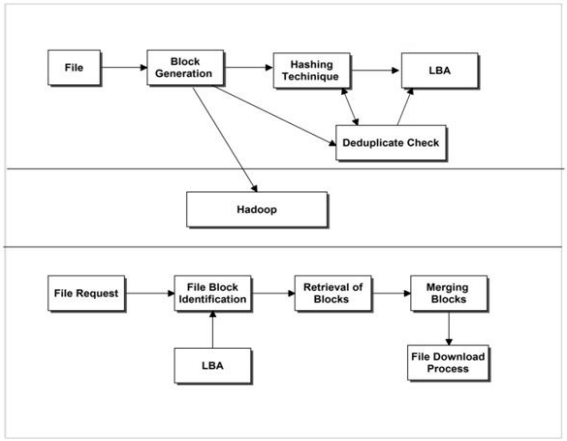

Fig -1:Block Diagram

When the user upload the image the following operation takes place:

- The Image is broken down into small blocks.
- Hash Code is generated for each block using MD5 Algorithm.
- Then duplication check is performed using multilevel indexing comparison of hash code.
- LBA (Logical Block Address) is generated for the Image.
- This information is updated in the Database.
- Then unique blocks are uploaded into Hadoop HDFS.
- A confirmation message is prompted indicating Image is uploaded.

When the user selects the image for downloading the following operation takes place:

- The Image details are searched in the Database.
- The LBA (Logical Block Address) of the selected Image is retrieved from the Database.
- Then the blocks are retrieved from Hadoop HDFS.
- The retrieved blocks are merged to get back the original Image.
- A message box is prompted as to where the Image which is downloaded needs to be stored.
- A download confirmation message is given to user in the webpage.

3.2 Software Requirements

- Operating system: Windows 7
- Coding Language: Java (Jdk 1.7)
- Web Technology: Servlet,JSP,HTML,CSS,JS
- Web Server: TomCAT 6.0
- IDE: Eclipse Indigo
- Database: My-SQL 5.0 32-bit
- UGI for DB: SQLyog
- JDBC Connection: Type 4 - Native Drive
- Tools: Cygwin
- Big data Software: Hadoop

3.3 Implementation

•Efficient map reducing technique is implemented, while uploading the file to the Hadoop HDFS, file will be divided in to the chunks for each chunks hash code will be generated, for all the unique hash codes, unique id will be assigned and create <key, value> pair whereas key is the unique hash code id, and value is the respective hash code.
•The unique chunks will be stored in the Hadoop HDFS. If the file chunks are already exists in the Hadoop HDFS, the chunk would be skipped else would be added to HDFS.
•File chunk details will be maintained as the Logical Block Addressing (set of keys) which is used to retrieve the Images.
•While downloading the file based on the Logical Block Addressing get the keys and with the keys value will be retrieved, where value is the hash code present in the hash table.
•Based on the hash codes chunks will be downloaded to the server and will be merged and the original file will be sent to the client system without data lose.

Input Splits:
The Image is split into blocks of size based on multiple length and width of the image.

Mappers (Algorithm):
For each Block the Algorithm generates the key value pairs i.e. the key would be the id i.e. LBA and value would the hash code.

Reduce Phase:

In the Reduce Phase the hash values are compared with the ones which are in the there in the Hadoop (HDFS).

Each phase is shown in Fig-2.

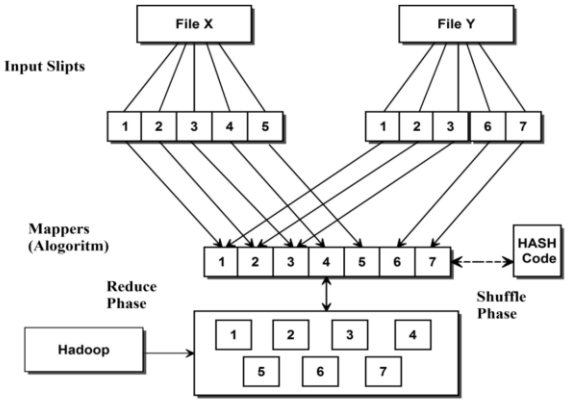

Fig -2:MapReduce Implementation

3.3.1 Division of Images to chunks

- Consider the image which has to be divided .Values of rows and columns have to be determined, for ex, if the value of row and col is 5, then the image will be divided into 25 blocks.
- Then the height and width of the chunks will be determined based on its co-ordinates.
- Finally all the chunks will be stored in an array.
- On displaying the array the whole image which is divided into chunks will be displayed.

3.3.2 Md5 to generate hash code

- Here the data is nothing but a sequence of bytes. Additional bytes are added at the end of the sequence so that after padding, the whole length must be a multiple of 64.The content of padding (which would be usually a lot of zeros and encoding of the input data length) would be mentioned by specification.
- Here the data, after padded will be divided into blocks of 64 bytes. Every block will be processed and each block takes a 16 bytes input from the previous block and gives a new 16 bytes output value which will be the input for the next block.
- Since there won't be a previous block for the first block, there will be a fixed value to start the process and MD5 gives the detail of this fixed value.
- So after processing the last block, a 16 bytes output will be generated and this will be the complete output of the MD5.

4. CONCLUSION AND FUTURE WORK

In the existing system if same image is uploaded twice ,if size of an Image uploaded first be 150Kb and if the same image is uploaded again then total storage space would be150+150=300k.

Just imagine in Huge companies like Facebook, Google, Yahoo, and Instagram would have billions of users getting Gigabytes of data generated for every hour from users if the same image is uploaded then it would take double the space. The Same Image would be uploaded by many people in Facebook during birthdays of famous people. During these times the scalability of the servers needs to be increased to store large data.

Our Proposed system would reduce the storage space by storing only one instance of the image. To be much clearer the person's eyes and nose wouldn't change in different image so those block needs to stored only once.

We can show analysis of this using the following graph

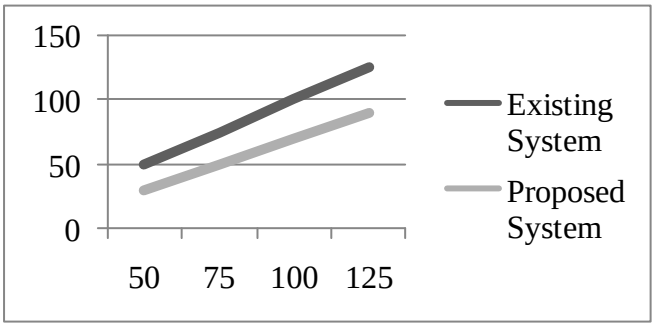

Chart -1: Graphical Analysis

The x-axis indicates the number of users and y-axis indicates the size of the image in Mb and the dark shaded line indicates the existing system and the lightly shaded line indicates the proposed system. We can clearly notice that the storage space is used efficiently in the proposed system.

Future Enhancements

•Increasing the scalability and storage of the nodes in HDFS.
•Extending the same technique to audio formats.
•Extending the same technique to video formats.

REFERENCES

[1] Phoenix: A MapReduce Implementation With NewEnhancementsAmer AI-Badarneh, Hassan NajadatMajd AI-Soud, RashaMosaidISBN: 978-1-4673-8913-6

[2] Grolinger, K., Hayes, M., Higashino, W. A., L'Heureux, A., Allison, D.S., &Capretz, M. A. (2014). Challenges for Map Reduce in Big Data

[3] Ohlhorst, F. J. (2012). Big data analytics: turning big data into big money. lohn Wiley & Sons.

[4] Dean, 1.,&Ghemawat, S. (2008). Map Reduce: simplified data processing on large clusters. Communications of the ACM, 51 (I), 107-113.

[5] Condie, T., Conway, N., Alvaro, P., Hellerstein, 1. M., Elmeleegy, K., &Sears, R. (2010, April). Map Reduce Online. In NSDI (Vol. 10, No. 4, p.20).

[6] Gu, Y., & Grossman, R. L. (2009). Sector and Sphere: the design andimplementation of a high-performance data cloud. Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences,367(l897), 2429-2445

[7] Yun hongGu, Robert L. Grossman, Alex Szalay and AniThakar, Distributing the Sloan Digital Sky Survey Using UDT and Sector, Proc. of e-Science 2006.

[8] Doulkeridis C &NovagK(2014).A survey of large-scale analytical query processing in Map Reduce. The VLDB Journal,23(3),355-380.

[9] JustinTalbot,RichardM.Yoo,andChristos Kozyrakis:Phoenix++:Modular Map Reduce for Shared Memory Systems.

[10] YooR.M,Romano,A&Kozyrakis C (2009 October).Phoenix rebirth: Scalable Map Reduce on a large shared-memory system. In Workload Characterization 2009.IISWC 2009 .IEEE International Symposium on(pp.198-207),IEEE