International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

Image Storage Optimization using Deduplication

Rajalakshmi K. R.[1], Dr. Saritha Chakrasali[2]

[1]Department of ISE, BNMIT, Bangalore, India
[2]Department of ISE, BNMIT, Bangalore, India

Abstract: ***One of the challenges of the today's industry is the storage of similar data, repeatedly and thereby consumption of disk space for it, whether local on the physical standalone server or on cloud. An increase in volumes of same data across several user profiles causes heavy increase in amount of data stored and causes more disk space usage. Storage of same redundant data can be avoided as much as possible. Major Contribution to the reduction of the storage space is elimination of data duplication and usage of hash coding techniques to conserve space. Images play an important role in today's use of data. An attempt is made in this paper to develop an application to reduce data duplication using hashing technique on a Hadoop Framework.***

Keywords: Optimization**,** Hadoop, De-Duplication, Images, Storage

1. Introduction

The biggest challenge for many applications whether it is social networking, hospital, and data analytics is the storage space management. A day's activity of any user involves uploading the images and other information. Chances of same image being recirculated across several profiles of users are another possibility and challenge for the owners of these websites. A huge growth of data stored has led to high demand in various methodologies to be incorporated to save disk space & its management strategies. An application has been developed in an attempt to conserve space while storing images using simple hashing technique and mapping tables on Hadoop framework [1]. Cygwin provides an environment for Hadoop in Windows platform.

2. Literature Review

Dimitrios Markonis et, al [2] discussed that the growth of the amount of medical image data produced on a daily basis in modern hospitals forces the adaptation of traditional medical image analysis and indexing approaches towards scalable solutions. The number of images and their dimensionality increased dramatically during the past 20 years. Enterprises today acquire vast volumes of data from different sources and leverage this information by means of data analysis to support effective decision-making and provide new functionality and services. The key requirement of data analytics is scalability, simply due to the immense volume of data that need to be extracted, processed, and analyzed in a timely fashion. The most popular framework for contemporary large-scale data analytics is MapReduce [3], mainly due to its salient features that include scalability, fault-tolerance, ease of programming, and flexibility.

Zhou Lei et, al [4] proposed an effective image file storage technique using data deduplication with a modified fixed block scheme. In this scheme, when the user requests to store an image file, it calculates the fingerprint for image file. If the fingerprint already exists in the library, a pointer to the existing fingerprint is used to store the image.

Data Deduplication is a process of eliminating the redundant data without using any compression techniques. In this paper, data deduplication process is achieved using hashing technique by implementing Message Digest5 algorithm. This is a pilot project which has used Hadoop for storage as Hadoop is scalable and reliable framework. Though all features of Hadoop framework [5] are not used, the proposed system provides the scope to extend this project to a distributed environment.

3. System Architecture

System architecture of proposed system is as shown in Figure 1. In the proposed system, an efficient hashing technique is implemented. A user uploads a given image file using user interface. During upload phase, file will be divided into chunks and for each chunk; a hash code will be generated. For all the unique hash codes, unique id will be assigned and a <key, value> pair will be created where key is the unique hash code id, and value is the respective hash code (which resembles the chunk) that are maintained in the hash table. The unique chunks will be stored in the Hadoop HDFS. If the file chunks already exist in the Hadoop HDFS, then it will be mapped to the existing chunk and duplicate chunk will not be uploaded to Hadoop HDFS [5]. File chunk details will be maintained as the logical block addressing of keys.

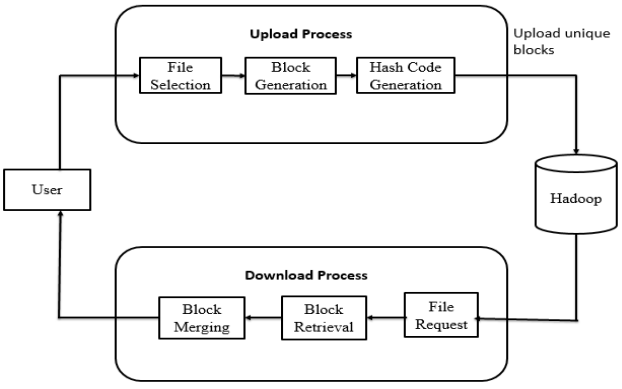

Figure 1: System Architecture

The overall upload process of image into Hadoop HDFS involves three primary steps, Chunk Splitting technique, Hashing Technique and Deduplication checking technique.

3.1 Image Upload Process

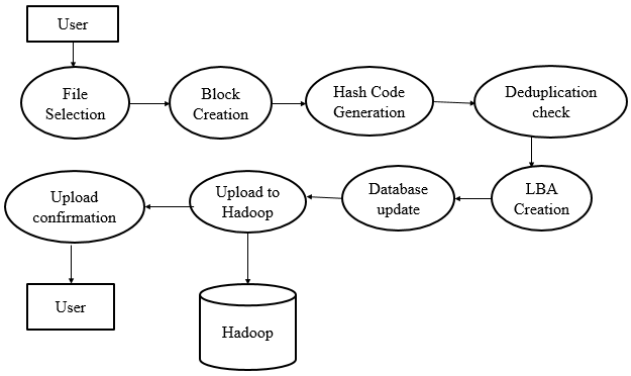

Figure 2: Image Upload Process

Figure 2 shows the Data Flow Diagram of Image upload process for the proposed work. User makes the choice of image that needs to be uploaded. Based on row and column size defined, image is slit into blocks. For each block, a hash will be generated using MD5.The blocks are checked for duplicates and only unique blocks are stored in Hadoop.

3.2 Image Download Process

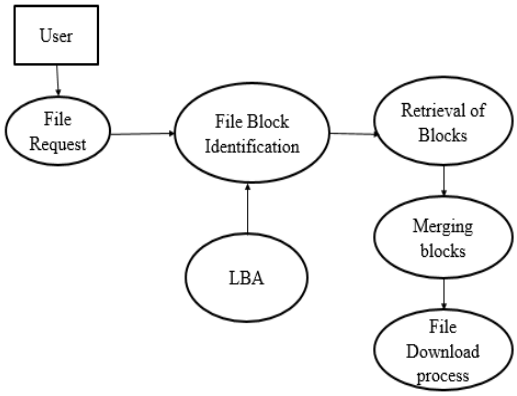

Figure 3: Image Download Process

Figure 3 shows the Data Flow Diagram of Image download process for the proposed work. User request the image to be downloaded from application. Requested image is looked through logical block addressing and each of the blocks associated with that particular image are individually retrieved. Image chunk details will be maintained as the Logical Block Addressing. While downloading, the file is downloaded based on the logical block addressing and the key value will be retrieved, where value is the hash code present in the hash table. Based on the hash codes, the image chunks will be downloaded to the server and it will be merged and the original file will be sent to the client system without data loss. The retrieved files are merged together according to their keys and is offered to the user as single downloaded file.

4. Evaluation

The proposed system was applied on burst images where the user will upload similar images. Multiple blocks were generated for images and the block level evaluation metrics has been generated as shown in Figure 4.

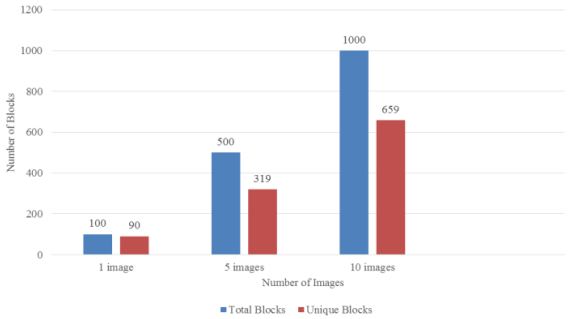

Figure 4: Block Level Evaluation Metrics

The proposed system was initially tested for 10 burst images of size 1MB each and the percentage of disk space saved is calculated for 1image, 5 images and 10 images as shown in Table 1.

Table 1: Percentage of disk space saved for images

Images	Image Size	% Disk Space Saved
1 Image	1 MB	20
5 Images	5 MB	32
10 Images	10 MB	43

5. Conclusion

With implementation of hashing technique and logical block addressing, there has been significant reduction in disk storage space. The proposed system implemented above considers only limited size images. This is can further be enhanced to incorporate huge sized files greater than 10 Mb, usually seen in much usage in Medical Fields. Analysis and research can be envisaged on these huge sized files to see the impact on the performances. This work can be extended to a distributed network where Hadoop MapReduce [8] will further help in efficient storage.

References

[1] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler "The Hadoop Distributed File System" In Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies,2010.
[2] Dimitrios Markonis, "Using MapReduce for Large-scale Medical Image," In Proceedings of the IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology,2012
[3] Doulkeridis, c., & Norvag, K. (2014). A survey of large-scale analytical query processing in MapReduce. The VLDB Journal, 23(3), 355-380.
[4] Zhou Lei, "An Improved Image File Storage Method Using Data Deduplication," In Proceedings of the IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 2014.
[5] Tom White, Hadoop: The Definitive Guide. Fourth Edition, O'REILLY, April 2015
[6] Eclipse IDE Tutorial [Online], Available: http://www.vogella.com
[7] Apache Hadoop [Online], http://hadoop.apache.org.
[8] Jeffrey Dean and Sanjay Ghemawat "MapReduce: Simplified data processing on large clusters" in OSDI conference, 2004