International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 12 – SEPTEMBER 2017.

An Efficient Hash Based Data De-Duplication Approach in Large Scale Distributed Storage Servers

S.U.Muthunagai [#1] and R. Anitha [*2]

[#] *Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, India*

[*] *Department of Information Technology, Sri Venkateswara College of Engineering, India*

***Abstract*— Distributed Storage is a service where data is remotely maintained, managed and backed up. Whenever data is broadcast to the network for communication, it is routed to the server for storage. But accumulating the existing data repeatedly will cause the server to overload and leads to increase in the need of the data storage servers and further increases latency during data retrieval. Hence in order to overcome this issue, whenever the data is uploaded to the server it will be compared with the existing data in the server and if the data exists previously, an additive implication is exploited to locate and retrieve the respective data. The additive implication resolves the issue of redundancy and helps in efficient access of data. This paper proposes an Efficient Hash-Based Data De-Duplication approach in Storage System which avoids the replication of data in the server and also provides an effective way of accessing the data with reduced latency thereby guaranteeing the content delivery efficiently at the other end. The proposed Hash-Based Data De-Duplication technique spawns the unique identification which encounters the existence of data and the further analysis of data profound reduces the collision effect.**

Index Terms*— *Data De-Duplication, Hash-based data De-Duplication, Large scale distributed storage.

I. INTRODUCTION

In computing, a data server or file server is a computer attached to a network meant for providing a location for shared disk access, i.e. shared storage of computer files such as documents, sound files, photographs, movies, images, databases, etc., that can be accessed via workstations that are attached to the same computer network. The large scale distributed data servers are designed primarily to enable data storage and data retrieval and at the same time the computation task is also conceded out by the workstations. Due to increase in redundant data exponentially, issues related to the storage space, effective retrieval and network complexity are augmented. Data de-duplication is one of the important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth [4]. In the existing approach data de-duplication technique is carried out during the storage process in the server, after undergoing many process required before storage which is time consuming. The above mentioned issue has been overcome in the proposed approach where the data de-duplication is carried during the upload process in the storage server. Thus, the proposed efficient hash based data de-duplication approach, reduces the space in storage system by eliminating the redundant data in storage system.

The remaining section of the paper is organized as follows: Section 2 summarizes the related work and problem statement. Section 3 describes the proposed system model. Section 4 illustrates the Hash based secure data de-duplication approach and issues of the proposed model. The performance evaluation of proposed model is shown in Section 5 and finally draws conclusions at Section 6.

II. RELATED WORK

The proposed hash-based data de-duplication approach has been developed with analysis of storage issues such as availability of redundant data in storage servers, increased latency during data retrieval etc., which has been addressed in [1], [3]. while a complete survey of data de-duplication is provided with encryption and decryption. The de-duplication is done with integration of files into useful resources that are can be accessed via centralized management and virtualization [2] help to set the goal of proposed work. The idea of Authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and transfer has been mentioned in [4] which gives an idea about duplication check in storage server. Data consistency in performance oriented De-duplication [8] furnishes an idea about the referenced data be reliably stored on disks and the key data structures not to be lost in case of a power failure. However, not a large amount of work has yet been done to address hash based data de-duplication and its associated issues.

III. SYSTEM MODEL

Fig. 1 shows the architecture diagram of the proposed model. The operations are performed in order to avoid redundant data in the storage system.

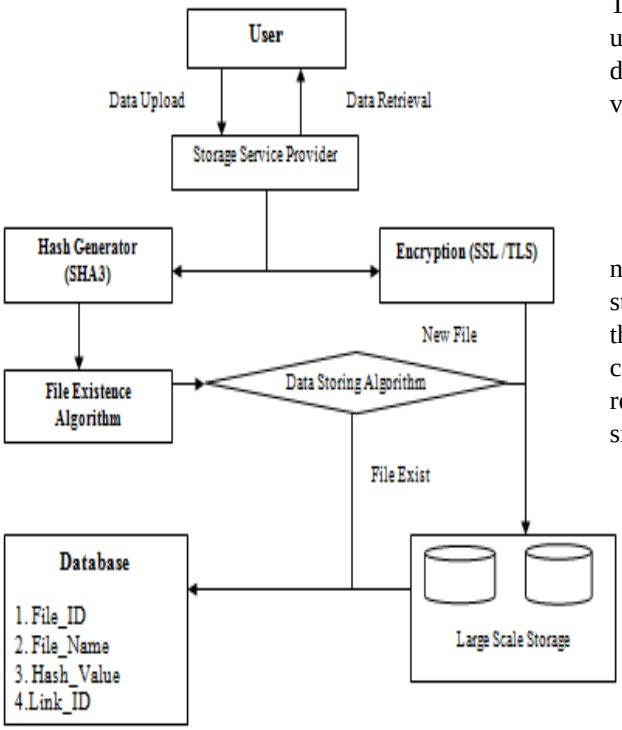

Fig. 1. Hash Based Data De-duplication Model

In this model, when a user uploads the file the corresponding hash value is generated using SHA3 and the file is stored in the large scale distributed storage servers. If the other user attempts to upload the same file, the hash value generated by the file remains the same. In this case, link is created for file located in the storage system through which 'n' users can access the same file and thus shrinks the space in a database and increases the efficiency by means of quick processing.

IV. HASH BASED DATA DE-DUPLICATION

The proposed efficient hash based data de-duplication approach in large scale storage servers is designed to run on commodity hardware. The proposed model decreases the space consumption and increases the efficiency of the storage system. This system performs two different tasks: massive data storage and fast processing. The succeed modules of the proposed model performs the crucial part of the storing the data in the storage server. Similar content of the data or images in pixels are identified at the initial stage and the link has been generated else different images or data gets uploaded on the already existed name of the file then the conflict occurs between the two data.

A. User Specific File Management and Hash Generator

In this module, the various user management tasks are performed. This includes the user login, data accessing, data deleting etc., A user can register themselves and login using their credentials. Once the user logs in he/she can use their account to manage the files. The files are specific to each user. Similar files are shared but a hashing algorithm is used to compare and differentiate files of different user. Separate file comparison algorithms are developed for text and image files. The Hash based file existence algorithm ensures that the uploaded file is existed in the database. The comparison is done with the new generated hash value and existed hash value hoarded in the log entries.

B. File Existence Algorithm

This algorithm ensures that the uploaded file is existing or not, by means of comparing the generated hash value with the stored hash value in the storage server. If hash value collides, the individual files are compared by using separate file comparison algorithms. After the contents are compared the result is passed out to data storing algorithm whether the file is similar or not.

```
FileExist (inputFile , inputHash)
FileIDs[ ]=hashTable(inputHash);
for FileID in FileIDs[ ] do
if( compare (inputFile , File. FileID )
return FileID ;   //returns Existing file_id
return DOESN'T_EXIST; //returns  NULL if doesn't exist
END
```

C. Data Storing Algorithm

The data storing algorithm decides the storing part of the proposed work. This ensures whether the uploaded file has to be saved in the repository or it is sufficient to link the Id for the file. Hence the two users will share the same file through link bent. When one user edits the already stored shared file then a new file is created for that user.

```
DataStoring (inputFile , inputHash)
FileID = FileExist (inputFile , inputHash);
if (FileID == NULL)
inputFile.Link_ID=NULL;
LOAD(inputFile);      //Stores File in Repository
UPDATE DB;     //Updates DB
else
inputFile.Link_ID = FileID ;
UPDATE DB;     //Updates DB alone
 END
```

V. IMPLEMENTATION RESULTS

The experiments were carried out in large scale distributed system. In our experiment, hundreds of file has been uploaded in the storage system. Once the file is uploaded either text file or image file the hash value is generated for each file so that the comparison between new file and existing file is made which is shown in Fig. 2 and Fig. 3.

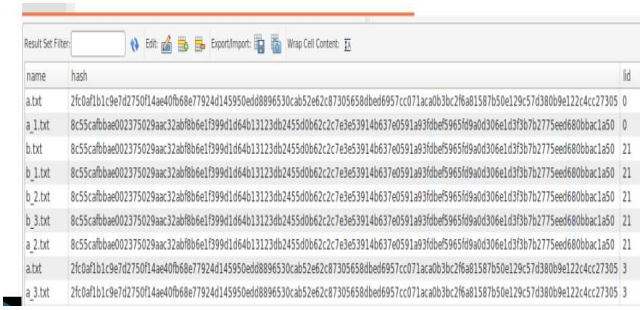

Fig. 2. Generation of Hash Value for Text File

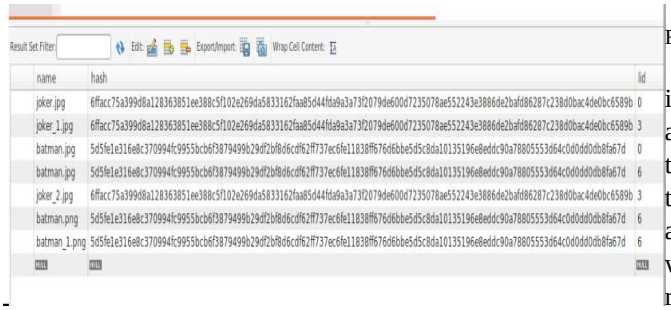

Fig. 3. Generation of Hash Value for Image File

The duplicate file of a content will not get uploaded to the storage server, instead, it will be presented with the uploaded name and a link will be set to the original file which is shown in Fig. 4.

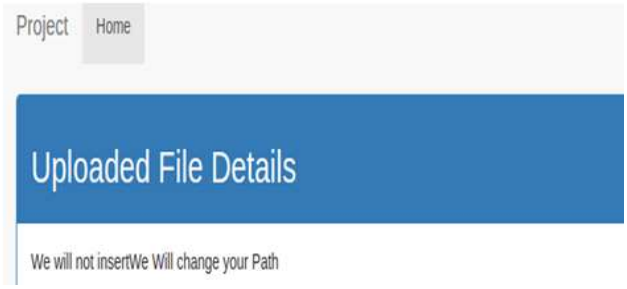

Fig. 4. Comparison of File with Hash value

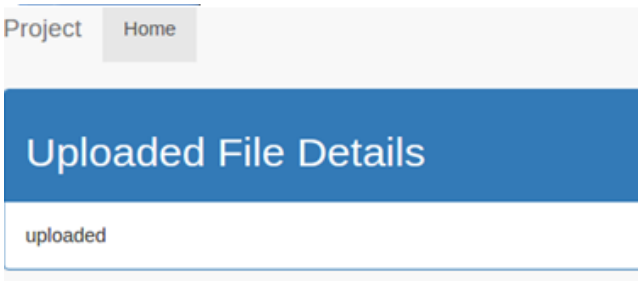

Fig. 5. Result of Data Storing Algorithm.

In comparison, if the uploaded data is found to be new then the file is saved as a unique file by means of data storing algorithm as shown in Fig. 5. The File_ID and Hash value are generated for the new file which is uploaded.

The proposed hash-based de-duplication paves the way for eliminating redundant data from storage with the help of unique hash value generated for each data during upload of data into the storage system.

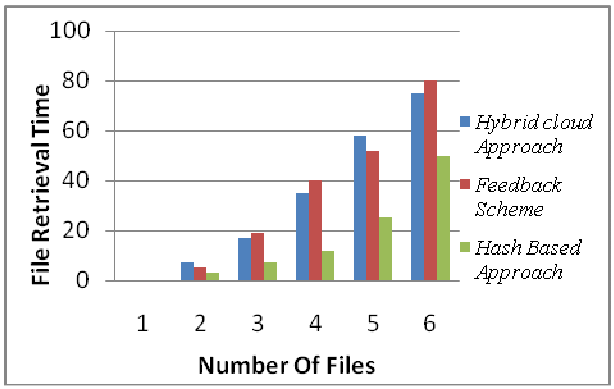

Fig. 6. Evaluation of Efficiency

The Efficiency of hash based data de-duplication algorithm is compared with Hybrid approach and feedback scheme approach for efficiency evaluation. The above Fig. 6 shows that the latency time of file retrieval from storage increases if the number of files increases in the storage system. since there approach is dealt with cryptographic and feedback schemes. whereas in the hash based approach the time obtained to retrieve the file decreases since it creates link to access the file by many user simultaneously.

The proposed hash based data de-duplication uses the technique prior to the data residing in the storage system henceforth it takes minimal time to retrieve the data from storage system.

VI. CONCLUSION

In this paper, the approach of hash-based data de-duplication is constructed to discharge the redundant data in the storage system. This proposed work presents the data storage with the efficient way of handling and maintaining the data with no placing of the duplicate content in the database server. Hence, formerly the file content will get stored in the database regardless how many times when a user endeavors to upload the same content with diverse name. Thus, accumulating the content of the file at once in the storage results in an effective way of using the storage system. Henceforth, the data content of the file will also be hashed and so the hashed value has been used for substantiating the existence of the file. The future work will be implemented towards the different formats of data like audio, video file etc., thus in turn reflects with more efficient way of usage of storage space in the servers for accumulating the huge volume of data.

REFERENCES

[1] Junbeom Hur, Dongyoung Koo, Youngjoo Shin, and Kyungtae Kang "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage." *IEEE Transaction On Knowledge And Data Engineering*, Vol. 28, No. 11, pp. 3113-3125, 2016.

[2] Shengmei Luo, Guangyan Zhang, Chengwen Wu, Samee U. Khan and Keqin Li "Distributed Deduplication for Big Data Storage in the Cloud." *IEEE Transaction On Cloud Computing*, Vol. 61, No. 11, pp. 1-13, 2015.

[3] Victor Chang, Muthu Ramachandran "Towards achieving Data Security with the Cloud Computing Adoption Framework. " *IEEE Transactions on Services Computing* , Vol. 9, No 1, pp. 138-151, 2016.

[4] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P.C. Lee, and Wenjing Lou "A Hybrid Cloud Approach for Secure Authorized Deduplication"

IEEE Transaction On Parallel And Distributed Systems, Vol. 26, No. 5,pp. 1206-1216, 2015.

[5] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng "Deduplication on Encrypted Big Data in Cloud". *IEEE Transaction On Big Data*, Vol. 2, No. 2,pp. 138-150, 2016.

[6] Tin-Yu Wu, Jeng-Shyang Pan, and Chia-Fan Lin "Improving Accessing Efficiency of Cloud Storage Using De-Duplication and Feedback Schemes" *IEEE systems journal*, Vol. 8, No. 1,pp. 208-218, 2014.

[7] T. Baker, B. Al-Dawsari, H.Tawfik, D.Reid, Y.Ngoko "GreeDi: An energy efficient routing algorithm for big data on cloud", *Journal of Adhoc Networks*, Vol. 35, pp. 83-96, 2015.

[8] Bo Mao, Hong Jiang, Suzhen Wu, Lei Tian, "Leveraging Data Deduplication to improve the performance of Primary Storage System in the cloud", *IEEE Transaction on Computers*, Vol. 65, No.6, pp.1775-1788, 2016.

[9] Maomeng Su, Lei Zhang, Yongwei Wu, Kang Chen, Keqin Li, "Systematic Data Placement Optimization in Multi-Cloud Storage for Complex Requirements", *IEEE Transactions on computers*, Vol.65, No.6, pp. 1964-1977, 2016.

[10] Chien-An Chen, Myoungyu Won, Radu Stoleru, Geoffrey G. Xie, "Energy-Efficient Fault-Tolerant Data Storage and Processing in Mobile Cloud", *IEEE Transactions on cloud computing*, Vol. 3, No. 1,pp.28-41,2015.

[11] Zhen Huang, et al, Jinbang Chen, Yisong Lin, Pengfei You, Yuxing Peng, "Minimizing data redundancy for high reliable cloud storage", *Journal of computer Networks*, Vol. 81, pp. 164-177, 2015.

[12] Waraporn Leesakul, Paul Townend, Peter Garraghan, Jie Xu, "Fault-Tolerant Dynamic Deduplication for Utility Computing", *IEEE International Conference on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 397-404, 2014.

[13] Leesakul, Waraporn, Paul Townend, and Jie Xu. "Dynamic Data Deduplication in Cloud Storage." *IEEE International Conference on Service Oriented System Engineering*, pp. 1-9, 2014.

[14] Thanasis G. Papaioannou, Nicolas Bonwin, Karl Aberer, "Scalia: An Adaptive Scheme for Efficient Multi-cloud Storage", IEEE proceeding High performance Computing, Networking , Storage and Analysis, 2015.