ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

Effective Techniques for Distributed Storage with Data Security

Vaishnavi N[1], G GSivasankri[2]

P.G. Student, Department of Computer Engineering, AMC Engineering College, Bengaluru, India[1]

Professor, Department of Computer Science and Engineering, AMC Engineering College, Bengaluru, India[2]

ABSTRACT:Cloud storage provides an economical and prominent solution to share the resources among cloud users. However, the storage capacity in the cloud is increasing and has become an emerging trend in the field of data storage. As the users load the data into the cloud there may be the chances of causing the major problem like redundancy of files leading to wastage of storage space. To solve this problem Key-value store played a crucial role and exhibited positive results. Data in the cloud need to be provided with security since the information of the user is confidential. Here we propose the concept of Big File Cloud with its evaluations and architectures for the huge storage of data stuffed by the users .The researches also applied the advantages of Zing Data Base (ZDB) which literally solved the problems in BFC. The outcomes can be utilized for building adaptable circulated information distributed storage that backing huge record with size up to terabytes of data. In addition here we analyze and show the security of the schemes.

KEYWORDS:Key-Value store, Big File Cloud, ZDB, Digital Signatures

I. INTRODUCTION

Storage pools are the sufficient space assembled from diverse physical resources in a distributed environment. In that cloud storage provides the space to keep the data for abundant users. For example, capacity for each user provided by the cloud infrastructure could be in gigabytes and terabytes. The popularity of the applications has become enormous since cloud is used by the users for their everyday life right from uploads, downloads of data to exchange of information in any social networking sites[1], Zing me[2]. So, the data dumped in a massive repository will be bulky in cloud, increasing the holding ability of the system. Problems, and difficulties are faced by the system to provide good quality of services to the people. Hence these considerations apply to the term

Big Data that is a massive buzzword used today to change the world and applied in every applications like healthcare and finance trading etc. Henceforth, with this huge and complex foundation it has become easy to store a bulk number of data for the users. Specifically, big data servers managed by the providers causes uncertainty to the users. These set of data is required to be secured and efficient taking into an account of data privacy and system performance.

The essential technique to tackle the issue is to cipher the information and after that transfer the same. Miserably, providing an effective and secured data-sharing scheme is quite challenging. On uploading files to bigdata by several number of users sometimes the documents might be similar and could cause to data redundancy which is again a problem of waste in the storage space. There are several other problems while designing an efficient storage engine such as big file processing, de-duplication, distributed and high scalability.

II. RELATED WORK

Maintaining the large set of data in proper manner becomes challenging. Data confidentiality is also necessary when the files are distributed widely and is to be maintained with privacy preserving requirements. Transmission the data safely and accurately becomes challenging. Files sent to the server are divided into multiple number of chunks and stored it on a distributed system or on disks and are managed by metadata [3],[4],[5],[6]. However designing a

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

lightweight metadata and storing these chunks and metadata with high performance were some of the problems to be faced. The difficulty in storage space of metadata resulted in O(n) and was infeasible. A solution was found to apply the advantages of key value store where the whole document is put into it and the data is stored in the form of key and value pair as related to the data security, the files are encrypted and stored in the storage area instead of keeping it directly.

Big File Cloud is introduced to manage the storage space. For each divided file, each chunk is provided with fixed size metadata ,results in fast access distributed file IO. SHA256 algorithm is applied and a key is generated to check for the duplication of the content of the file meanwhile the comparison is made between the previous key and the new key generated for the next file and verify for the duplication of the file content. This paper proposes the advantages of ZDB in the form of key value pair .An idea of digital signature is introduced to check whether the user is authenticated while uploading the data to maintain the data privacy. A big file table is maintained to keep track of the file details.

For a big file a light weighted metadata is proposed, fixed size of metadata is given for every file.The space complexity problem of metadata is solved which resulted in O (1), while the metadata size of a Drop box, HDFS hasO(n) as the complexity, the original file size is defined as n.To support sequential read and write operation ZDB is used, a small memory-index overhead The larger file in the form of key value pair is not sustained by utilizing its benifits. Data redundancy is reduced so that the storage space is not wasted. Digital signatures are used to increase the authenticity.

III. BIG FILE TABLE AND DIGITAL SIGNATURE

Usually when a file is uploaded a big file table is maintained which consists the following attributes.

- Chunk_id[i] = fileInform.startChunk_id + i
  Meta-information is primarily portrayed in FileInform structure comprise of taking after fields:
- File_Name - the name of document;
- file_id: - one of a kind distinguishing proof of record in the entire framework ;
- SHA: - hash esteem by utilizing SHA calculation of document information;
- reference_file:- id of past existed file in Framework and have the same sha256 –

we regard these documents as one, reference_file is legitimate in the event that it is more prominent than zero;

- start_Chunkid : - the ID of the primary lump of record, the following piece will have id as start_Chunkid +1 et cetera;
- num_Chunk:- the quantity of lumps of the record;
- file_Size :- size of record in bytes;
- file_status:- the status of record, it has one in four qualities to be specific

Status of the file could be:
Uploading File - when lump are transferring to server;
Completed File - when all blocks are transferred to server ,and not checked as predictable;
Corrupted File - when all piece are transferred to server and not steady subsequent to checking;
Good Completed - the block of file is transferred to the server and steady checking finished with result. By utilizing this arrangement, a lightweight meta-information is characterized in distributed storage.

A text document of various size is applied with SHA256 to generate a key value of 256bytes and stored as a file id. In case there is a second chance where a file with same text document with different file name is to be uploaded then same 256 bytes of key will be generated which will be similar to the previous key. This file is not uploaded to the HDFS directory in order to avoid the duplicity .instead a reference id is copied from the id of the matched document. Group Signatures are usedto achieve secured data sharing. It enables the users to use the cloud anonymously. This technique also allows the clients to transfer the data in a secured manner. A pseudorandom numbers are generated by taking three variables and a signature is created for the authenticity. Before the data is sent to the cloud server the admin verifies

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

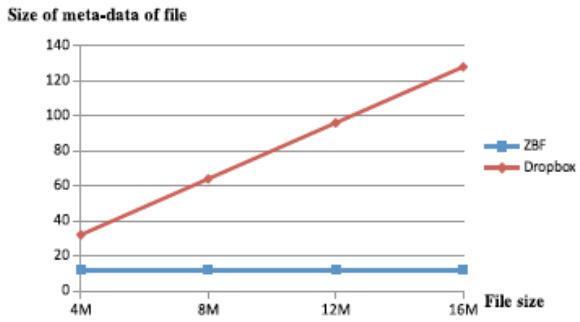
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

the signatures and then the process is continued.For this there are two methodologies followed Signature Generation and Signature Verification. After the data is uploaded to the HDFS directory it is stored in an encrypted format for further security purpose.

IV. EXPERIMENTAL RESULTS



(a)

Fig (a) shows the meta data comparison of BFC and Drop box. A list of chunks is maintained for the metadata files in Dropbox. Hence, both the size of file is linear. Big metadata is created to big file due to more number of chunks and a list of SHA256 values from them which is a lengthy process In BFC the fixed length of metadata is given which reduces thevolume of information for trading metadata in the middle of customers and servers.

De-duplication	Dropbox	One Drive	Google Drive	BFC
Single User	yes	no	no	yes
Multi-user	no	no	no	yes

(b)

Table (b) shows Dropbox supports the De-duplication as per user accounts. Global de-duplication mechanism is supported by BFC and saves the network traffic and internal storage space when many users store the same file content. Google Drive and One Drive do not support de-duplication.

```
Execution time was 1750 ms.

---------------DONE--------------

---------------Verify protocol--------------

c'= 484054655432247290380518667834027973513156171998061303243482455734136749942 70
c = 484054655432247290380518667834027973513156171998061303243482455734136749942 70

Group signature checked!
```

(c)

Figure (c) shows the generation of the certificates at user and client side obtained by the computations of many pseudo random generation numbers and using SHA256. If the certificates are matched at both sides then the file is uploaded to the server.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Special Issue 10, May 2016

V. CONCLUSION

In this paper designing Big File Cloud obtains high performance for distributed storage. Whenever the user uploads a file to the cloud sometimes the same redundant files may be stored which leads to the waste in the storage capacity. To solve the above method the paper uses some of the algorithms like SHA256 which generates a key and is stored in a big file table after uploading a file. the data duplication is checked on the server side. If the other file with same file content is uploaded it is matched with the hash key and make it as the reference Id. The files are divided into chunks and provided with IDs. The advantages of key value store into large files of data is utilized. Sequential read and write operation is completed by ZDB. Confidentiality of the data is maintained by securing the file information by providing an authentication of the user using Digital Signatures. The data could be shared easily privacy protected without revealing any identity privacy. The security is ensured and certifies the efficiency as well.

REFERENCES

[1] Facebook.http://facebook.com, 2014
[2] Zing me. http://me.zing.vn. Accessed October 28, 2014.
[3] Dropbox tech blog. https://tech.dropbox.com/. Accessed October 28, 2014
[4] D. Borthakur."*Hdfs architecture guide.*"HADOOP APACHE PROJECT http://hadoop.apache. org/common/docs/current/hdfs design. pdf, 2008.
[5] I. Drago, M. Mellia, M. M Munafo, A. Sperotto, R. Sadre, and A. Pras."*Inside dropbox: understanding personal cloud storage services"*. InProceedings of the 2012 ACM conference on Internet measurement conference, pages 481–494. ACM, 2012
[6] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. "*Ceph: A scalable, high performance distributed file system*". In Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06, pages 307–320, Berkeley, CA, USA, 2006. USENIX Association.
[7] I. Drago, E. Bocchi, M. Mellia, H. Slatman, and A. Pras. "*Benchmarking personal cloud storage"*. In Proceedings of the 2013 conference on Internet measurement conference, pages 205–212. ACM, 2013.
[8] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. "*Bigtable: Adistributed storage system for structured data*". ACM Transactions on Computer Systems (TOCS), 26(2):4, 2008.
[9] P. FIPS. 197: the official aes standard. "*Figure2: Working scheme with four LFSRs and their IV generation LFSR1"* LFSR, 2, 2001.
[10] S. Ghemawat and J. Dean."*Leveldb is a fast key-value storage library written at google that provides an ordered mapping from string keys to string values."* https://github.com/google/leveldb. Accessed November 2, 2014.
[11] S. Ghemawat, H. Gobioff, and S.-T. Leung. *"The google file system"*. In ACM SIGOPS Operating Systems Review, volume 37, pages 29–43. ACM, 2003.
[12] Y. Gu and R. L. Grossman. "*Udt: Udp-based data transfer for high-speed wide area networks.*" Computer Networks, 51(7):1777–1799, 2007.