IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 7, July 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

BigData: A Secure Distributed Storage System For Data

Asha C R, Mr. Srinivasulu M

M.Tech, Dept. of CSE Visvesvaraya Institute of Advance Technology, Bengaluru, India.
Assistant Professor, Dept. of MCA, Visvesvaraya Institute of Advance Technology, Bengaluru, India.

***Abstract*—The cloud storage services are fast growing now days and in data storage field cloud storage service become a raising trend. We need to distribute big-file to cloud, file should have lightweight meta-data, deduplication for same file, secure distributed file, and high scalability for system. Those are the above problems trouble by cloud system while designing efficient storage for that system. In this paper proposes a technique BFC its architecture based on key value store to manage the majority BFC depository troubles. By proposing little difficult, for big files same size meta-data design, it is done, it helps quick delivered file input/output and SHA 256 hash function are used to detect duplicate files fastly when many users upload same files and secure distributed file means after file upload to BFC it split into same size chunks and encrypt the chunk of data send to ZDB stored in HDFS and user can download by decrypting file from HDFS. For the benefits of ZDB this work is applied, here zing database is in house key value store for solving big file storage problems capably; here key value store increment integer keys. For constructing scalable delivering data cloud storage the results can be used so that cloud storage holds big-file up to few terabytes.**

Keywords—Big File, Zing Database, Distributed storage System

I. INTRODUCTION

Cloud storage delivers service to many numbers of clients for each client storage capacity now a day's reaches terabytes of data. Users can upload data from many devices like mobile phone, computer. In current day's peoples use cloud storage daily for example we can also download file from cloud and share file to their friends through social networks like face book etc.,. Cloud storage is loaded by system that is heavy. So that it guarantees the client for delivering good quality of service, without slow down we need to deliver high amount of data for many number of users, in an effective storing, accessing and managing big files in the system, data deduplication if stores data from the different users it causes the wastage of storage space so we need to lessen the wastage of storage space, distribute the data securely. Those are the above problems troubles by system.

For storing data in high degree data service key value store have lot of benefits. For small and medium sized key value store have low delay response time and good scalability. In many cases for directly storing big files the present key value store are not designed.

We did few experiments in which the system does not give good performance even if we put complete file data to key value store. We need to upload a big file or download a big file it takes more time so here latency is high for big files. For fast access operations in the main memory, no extra gap present to store one more object as a worth be huge. When the data and number of users increases then to expandable the system is not at all uncomplicated. Those are the problems in order to solve those above problems using key value store when storing big files this research is implemented. To design a Big file cloud, in data management it brings a lot of benefits of key value store.

A. Objective

To make high performance cloud storage, the BFC designed an uncomplicated metadata for large file; it is store in key value in zing database.

Multiple users upload file to big file cloud and BFC checks the uploaded files are duplicate or not. If duplicate file present means, if many users upload same file having same name but contents are different so here difficult to find out duplicate files in order to find out duplicate files BFC using SHA 256 hash function to detect the duplicate files fastly.

Secure distributed file means, user upload file to big file cloud and BFC receives the file than file is divided into same size chunks and encrypts data and store that chunks to zing database and send to HDFS. And also user can download by decrypting data from HDFS.

II. BACKGROUND

For cloud systems we need to design an efficient storage engine with a few requirements we needed such as we need to deliver big file to cloud, file should have lightweight meta-data, for same file we need to deduplicate, high scalability for system, secure distributed data. While solving those above problems Key value stores showed many benefits and played a significant role.

III. LITERATURE SURVEY

F.Chang, J. Dean, S. Ghemawat et al. [4] Bigtable is managing only structured data, it is a distributed storage system. Bigtable is designed to store very large size of data as Peta bytes of data users can store. For all Google products, the big table has successfully delivers a flexible high-performance solution. Drawbacks in this work are that if the chubby lock services are present then only the big table is available. The sql is not encouraging by big table even though delivering uncomplicated data manipulation APIs. Users to get modified to suit to the special APIs, they still need time even the user experience with sql.

S. Ghemawat et al. [7] for large distributed system are not encouraged well by traditional file system. Here internal failures occurs are the system damage, out of power and even all the time catastrophe happens, those failures occurs because of cluster has limit quality and large quantity of components and also here exception are less and components failures are more. The huge file we need to store and quickly we need to handle and operation to that file here is more appending then overwriting and also for Google the self designed file system are more flexible. So for large data intensive application GFS is designed as a scalable distributed file system. Drawbacks of GFS are for backend system the GFS is originally designed, but the more usage of GFS, so GFS is needed to many infrastructures to be improved for the users.

S. Ghemawat et al. [11] Google written level db means every data stored in the key value cache very quickly and it also gives mapping to string values from string keys. The advantages in this work are for small and medium sized key value store are fast in level db. Drawbacks are now a day's data is increasing by storing huge data to key value store are slow in level db for both uploading and downloading operations.

I. Drago et al. [2] another kind of cloud type depository system was dropbox. In that clients can store files, photos, documents and videos. Chunks which have data is 4MB that is contained in depository system that is dropbox. The bulk of the file is greater than given bulk than file is divided into few chunks. Every chunks is individual one by using SHA 256 value the chunks will find out and chunks will be stored in S3. Huge file means obviously it have huge mete data that done by few chunks as well as file chunk SHA 256 list of values. The drawbacks in this work are for storing the huge files are difficult as well as to scale out the depository system also difficult.

IV. PROPOSED SOLUTION

BFC architecture based on key value store to manage the majority BFC depository troubles. By proposing little difficult, for big files same size meta-data design, it is done, it helps quick delivered file input/output.

SHA 256 hash function is used to detect duplicate files fastly when many users upload same files and ECC algorithm is used to encrypt the chunks of file and decrypting the file..

For the benefit of ZDB this work is applied, here zing database is in house key value store for constructing high distributed data cloud storage the results can be used so that cloud storage holds big-file up to several terabytes.

FIG 1 below shows the overview of BFC architecture. Multiple users upload same content of files to BFC. The BFC received the file and checks the signature and content of the file. If finds file is not duplicate then the file is divided into same size chunks and encrypts the chunks of data and sends to zing database and stored in HDFS. If duplicate file finds then send response to the user and BFC sends only the reference id of the existing file to zing database and store in HDFS. And also user can download by decrypting file which is stored in HDFS.

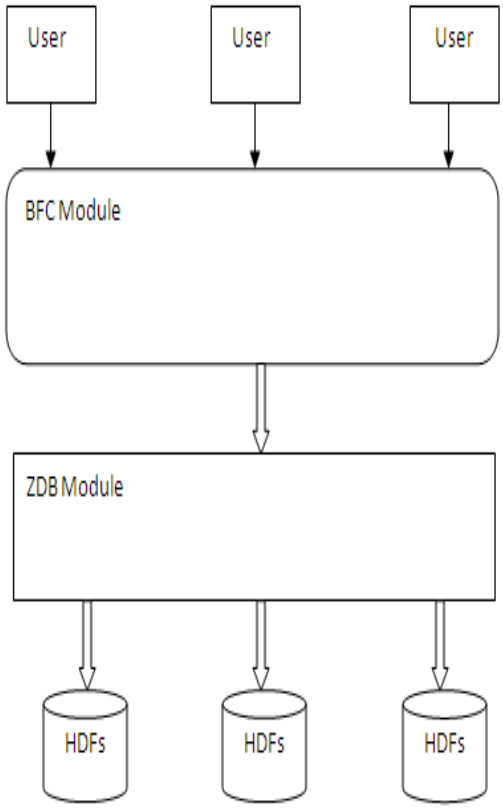

Fig. 1. BFC Architecture

A. Big file cloud module

Multiple users upload same files content to BFC. The BFC checks the signature and content of the file. If finds file is not duplicate then the file is divided into same size chunks and encrypts the chunks of data and sends to zing database. If duplicate file finds then send response to the user and BFC sends only the reference id of the existing file to zing database.

B. Zing database module

Zing database receives the encrypted chunk of data from the big cloud file and store it in key value store and send to HDFS.

C. Sig in module

Authentication for user by entering user name and password.

D. Register module

Collect all user documents like username, email id, and store that all user documents in back end.

E. Deduplication algorithm

Step 1: Begin

Step 2: Detect duplicate file using SHA 256

Step 3: If duplicate file

Then *ref FileID* of existing file send to ZDB

Store in HDFS

Step 4: Else file divide into same size chunks sends to

ZDB Store in HDFS

The deduplication algorithm working as follows. Multiple users upload big files to BFC having same filename but the contents are different in this case difficult to find out duplicate files. In order to overcome the above problem BFC use SHA 256 hash function to detect the duplicate files fastly how means for example user upload file C to BFC calculates the hash value for the content of the uploaded file C. if there is any file id related with hash value of file C then we can say that the file is duplicate send response to user and if it is duplicate then send only the reference id of existing file C to ZDB and it is store in HDFS. If no file id related with hash value of file C then we can says file is not duplicate. If file is not duplicate then file is divided into same size chunks send to ZDB and store it HDFS.

F. secure distributing file

We are using ECC algorithm to encrypt and decrypt the file. Before uploading file to BFC user has to register their details first; the details are stored in back end after registration done then user has to log in by entering user name and password after successful log in then only users can upload big file to BFC. The big file cloud divides into same size chunks and encrypts the chunks of data sends to ZDB and store in HDFS. User can also download file from HDFS by decrypting the encrypted file. In this the authorized users only upload and download files from the HDFS.

V. COMPERSION WITH OTHER CLOUD BASED STORAGE SYSTEMS

In this paper we are comparing BFC and Dropbox for metadata as well as deduplication comparison between BFC and Dropbox.

A. Comparison of metedata

Another kind of cloud type depository system was Dropbox. The chunks which have data is 4MB in the dropbox. If size of the file is greater than given size than file is divided into few chunks, every chunks are individual one by using

SHA 256 values the chunks will known and stored in S3. Huge file means obviously it have huge metadata that is done by few chunks as well as list of SHA 256 of file chunks. In dropbox for storing the big files as well to scale out the depository system also difficult. But in our research BFC for every file have same size metadata so that we can store the big files as well as to scale out the depository system easily.

B. Deduplication comparison between BFC and Dropbox

TABLE I. Deduplication Comparison

Deduplication	Dropbox	BFC
Single user	Yes	Yes
Many users	No	No

The above table 1 shows the deduplication comparison between big file cloud as well as Dropbox. Dropbox also help deduplication for single user. But big file cloud supports deduplication for many number of users. If many no of users upload same file having same name but content are different then it detect the duplicate files and save the wastage of storage space.

VI. RELATED WORKS

Google written level db means every data stored in the key value cache very quickly. For small and medium sized key value store are fast in level db. Now a day's data is increasing by storing huge data to key value store are slow in level db for both uploading and downloading operations.

In current days data is growing, for directly storing big files the present key value store are not designed. The file id as well as chunks id are in increment form in BFC so ZDB is good one to store data. Even the data is raise the ZDB have less delay for both uploading and download operations. Not only to use ZDB. We can also construct key value store in various data structures like hybrid storage device but the architecture of BFC, the chunks id of the file should be in increment form, so ZDB is good one to store data of chunk.

VII. CONCLUSION

To make high performance cloud storage, the BFC designed an uncomplicated metadata for large file; it is store in key value in zing database. Whatever files contained in the system all files have identical size metadata. Users upload file to big file cloud and BFC receives the files. The size of the file received is greater than given size than file is divided into same size chunks. Each chunks have identification in increment form until last chunk, especially when using ZDB, thus it is uncomplicated to deliver file also expanding depository of system. As multiple clients transfer identical file it causes wastage of storage space so BFC using SHA 256 hash function find out duplication files fastly and ECC algorithm are used to encrypt chunks of files and decrypting files.

ACKNOWLEDGMENT

I thanks to my guide Mr. Srinivasulu M, Assistant Professor, Dept. of MCA, VTU-CPGS, Bengaluru Region, VIAT, Muddenahalli, for his valuable guidance throughout this research work.

REFERENCES

[1] T. Nguyen and M. Nguyen. "Zing database: high-performance key-value store for large-scale storage service". *Vietnam Journal of Computer Science*, pages 1–11, 2014.

[2] I.Drago, M. Mellia, M. M Munafo, A. Sperotto, R. Sadre, and A. Pras. "Inside dropbox: understanding personal cloud storage services. *In Proceeding of the 2012 ACM conference on Internet measurement conference*, pages 481-494, ACM, 2012.

[3] L. Chappell and G. Combs. "*Wireshark network analysis: the official Wireshark certified network analyst study guide"*, Protocol Analysis Institute, Chappell University, 2010.

[4] F.Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. "Bigtable: A distributed storage system for structured data". *ACM Transactions on Computer Systems (TOCS)*, 26(2):4, 2008.

[5] Y.Gu and R. L.Grossman. "Udt: Udp-based data transfer for high-speed wide area networks". *Computer Networks*, 51(7):1777–1799, 2007.

[6] M. Placek and R. Buyya. "A taxonomy of distributed storage systems". *Reporte t´ecnico, Universidad de Melbourne, Laboratorio de sistemas distribuidos y c´omputo grid*, 2006.

[7] S. Ghemawat, H. Gobioff, and S.-T. Leung. 2003. "The Google file system". In *ACM SIGOPS Operating Systems Review*, volume 37, pages 29–43. ACM, 2003

[8] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. 2014.

[9] 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 proceeding of 7th symposium on operating systems design and implementation, OSDI '06, pages 307-320, Berkeley, CA, USA, 2006. USENIX Association.

[10] 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.

[11] S. Ghemawat and J. Dean. Leveldb is a fast key-value storage liberary written at google that provides an ordered mapping from string keys to string values. http://github.com/google/leveldb. Accessed November 2, 2014.

[12] ThanhTrung Nguyen, Tin Khac Vu, Minh Hieu Nguyen, Ha Noi, Viet Nam, "BFCSS: HighPerformance Distributed Big-File Cloud Storage Based On Key value Store", June 1-3 2015, IEEE SNPD 2015, 9781-4799-8676-7/15.

[13] T.T.Nguyen and M.H.Nguyen, "Design Sequential Chunk identity with Light weight Metadata for Big File Cloud Storage", IJCSNS International Journal of Computer Science and Network Security, VOL.15 No.9, September 2015.

[14] D.Harnik, B. Pinkas, and A. Shulman-Peleg,"Side Channels in Cloud Services: Deduplication in Cloud Storage," IEEE Security and Privacy, vol.8, no.6, pp.40-47,2010.

[15] W. K. Ng, Y. Wen, and H. Zhu, "Private Data Deduplication Protocols in Cloud Storage," Proc. 27th Ann. ACM Symposium on Applied Computing, pp. 441-446, 2012.

[16] P. ONeil, E. Cheng, D. Gawlick, and E.ONeil. The log-structured merge-tree(Ism-tree). Acta informatica, 33(4):351-385, 1996.

[17] D. Karger, A. Berkheimer, B. Bogstad, R. Dhanindina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi. Web caching with consistent hashing, computer networks, 31(11):1203-1213,1999.

[18] R. van Renesse and F. B. Schneider. Chain replication for supporting high throughput and availability. In OSDI, volume 4, pages 91-104,2014.

[19] F. PUB. Secure hash standard (shs). 2012.

[20] S. Shepler, M. Eisler, D. Robinson, B. Callaghan, R. Thurlow, D. Noveck, and C. Beame. Network file system(nfs) version 4 protocol. Network, 2003.

[21] P. Jin, P. Yang, and l. Yue. Optimizing b+-tree for hybrid storage systems. Distributed and parallel Databases, pages 1-27, 2014.

[22] D. Borthakur. HDFS architecture guide. HADOOP APACHE PROJECT http://hadoop. Apache. Org/common/docs/current/HDFS design.pdf, 2008.