ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

Secure-Cloud Storage with File Compression and De-Duplication

[1] Shaik Naval Bhanu, [2] Dr. A. Subramanyam, [3] M. Rudra Kumar
[1]M. Tech Student, [2] Professor and Head of Department [3]Associate Professor
[1][2][3]Department of CSE, Annamacharya Institute of Technology and Sciences, Village New Boyanapalli, Mandal Rajampet, District Kadapa, Andhra Pradesh, India.

Abstract: **Cloud storage services are emerging at a fast rate and rising in data storage field. These services are utilized by people for uploading and backing up data, sharing file through social networks like Cloud, Email, Google Drives Users are able to upload data from pc, mobile or tablet and also download and share them to others. Thus, system load in cloud storage becomes large. Nowadays, Cloud storage service has become an important requirement for several enterprises because of its features like cost saving, performance, security, flexibility. To design an efficient storage engine for cloud based storage systems, it is always required to deal with requirements like huge file processing, light-weight metadata, de-duplication, and high scalability. Here we propose big file cloud architecture to handle all issues in big file cloud system. Basically, here we propose to make a scalable distributed data cloud storage that supports huge file with size up to several terabytes to gigabytes. In cloud storage, system load is usually heavy. Data de-duplication with file compression to reduce the storage space caused by storing same static data from different users. In order to solve the above issues, a common method used in Cloud storage, is by dividing big file into small blocks, storing them on disks and then dealing them using a metadata system. Current cloud storage services have a complex metadata system.**

Index Terms: **-Secure-Cloud Storage, Chunk Storage Mechanism, Meta Data Storage and Data Security.**

I. INTRODUCTION

Nowadays, Cloud-based storage services serves lots of users with storage capacity for every user will reach to several gigabytes to terabytes of data to ensure a good quality of services for users, the system has to face several difficult issues and necessities as such as Serving intensity data service for a large number of users without bottle-neck, Storing, Retrieving and managing big-files within the system efficiently, Parallel and resumable uploading and downloading, data de-duplication to cut back the waste of storage space caused by storing identical static data from many different users. There are several challenges in traditional file systems for service builder once managing a huge number of big file: how to scale system for the improbable growth of data; how to distribute data during a large number of nodes; how to replicate data for load-balancing how to cache frequently accessed data for fast I/O, etc. a common method for finding these issues that is used in several Distributed File Systems and Cloud Storages is splitting huge file to multiple smaller chunks, storing them on distributed nodes and then managing them employing a meta-data system. Storing blocks and meta-data efficiently and designing a lightweight meta-data are significant issues that cloud storage suppliers need to face. This research has done by proposing new scalable distributed big-file cloud storage system and a better solution to reduce the storage space .Key-Value stores have several advantages for storing data in data-intensity services. This analysis is implemented to solve those issues once storing big-values or big file using key-value stores.

II. RELATED WORK

Maintaining the big set of data in correct manner becomes challenging data confidentiality is also necessary when the files are distributed wide and is to be maintained with privacy preserving requirements. Transmission the data safely and accurately becomes difficult. Files sent to the server are divided into multiple varieties of chunks and stored it on a distributed system or on disks and are managed by data. But designing a light-weight data and storing these chunks and data with high performance were a number of the issues to be faced. The issue in storage space of data resulted in O (n) and was infeasible. a solution was found to use the advantages of key worth store wherever the entire document is place into it and the data is keep within the type of key and value combine as related to the info security, the files are

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

encrypted and keep within the enclosure rather than keeping it directly.

Big File Cloud is introduced to manage the storage space. For every divided file, every chunk is give fixed size data; results in quick access distributed file IO. SHA algorithm is applied and a key is generated to check for the duplication of the content of the file meanwhile the comparison is formed between the previous key and therefore the new key generated for succeeding file and verify for the duplication of the file content. This paper proposes the benefits of ZDB within the type of key worth combine .An idea of digital signature is introduced to check whether the user is authenticated while uploading the data to maintain the info privacy. A big file table is maintained to stay track of the file details. For a big file a light weighted data is planned, fixed size of data is given for each file. The space complexity problem of data is solved that resulted in O (1), whereas the data size of a Drop box, HDFS has O(n) because the quality, the initial file size is zero(0) to n. To support sequential read and write operation ZDB is used, a little memory-index overhead the larger go in the shape of key value pair is not sustained by utilizing its benefits. Data redundancy is reduced so the storage space is not wasted. Digital signatures are used to increase the authenticity.

III. FRAME WORK

The proposed distributed big file cloud storage system has four distinct layers as such as- Application Layer, Storage Logic Layer, Object Store Layer, Persistent Layer. Application Layer: It consists of software on computers, mobile devices. This allows user to upload, download and share their files.

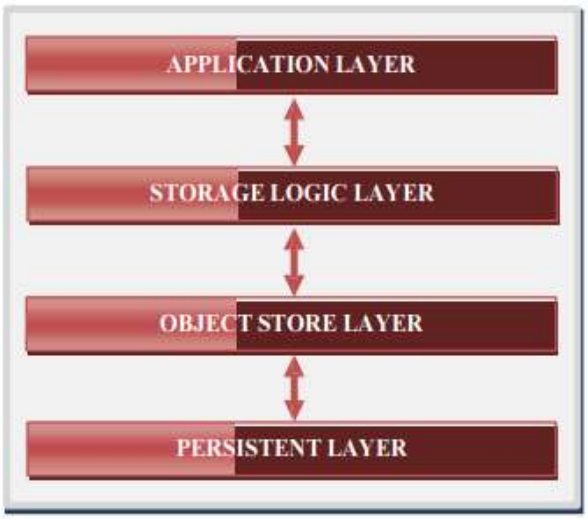


Figure 1: System architecture

Storage Logical Layer: It consisted of many user services, ID-Generator services and all logical API for Cloud Storage System. This layer implements business logic part. Object Store Layer: It contains several distributed backend services. Two necessary services of object Store Layer are File data Service and Chunk Store Service. File data Service stores data of files. Chunk Store Service stores data chunks that are created by splitting from the original files that user uploaded. Object Store Layer is the most important layer that has responsibility for storing and caching objects. This layer manages data of all objects, within the system including user data, file data, and especially meta- data. Persistent layer is used to the data.

A. Chunks Storage Mechanism:

Once users transfer a file, if the file size is larger than the organized size, it will be split into a collection of chunks. All chunks that are generated from a file except the last chunk have the same size. After that, the ID generator will generate id for the file and also the initial chunk with auto-increment mechanism. Next chunk within the chunks set are going to be assigned an ID step by step increase until the final chunk. A File data object is made with data like file-id, size of file, id of initial chunk.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

B. Meta Data Storage:

Metadata is the data that describes concerning original data. It consists of a series of elements, and each part has info similar to, chunk size, hash value of chunk. The length of the series are similar to the number of chunks from file, creating it difficult when the file size is massive.BFC suggested an answer in which the scale of metadata will not be dependent on the quantity of chunks regardless of file size(very small file or big file). Here, the solution suggests storing the id of the primary chunk, and also the number of chunks which are generated by splitting original file. have one in four states specifically, E-Uploading file - once chunks are uploading to the server, E-Completed file - once all chunks are uploaded but, it is not checked as consistent ,E-Corrupted file - once all chunks are uploaded to server but, when checking it is not consistent. E-Good Completed - once all the chunks are uploaded to server and obtained sensible result when consistent is checking. The below model is used in decoding into programming code. Thereby, the data of a file (size of File-info object) are almost an equivalent for all files within the system freelance of size of the file large or little. By utilizing this approach, we have created a light-weight metadata whereas designing a big file storage system.

C. Data Security:

Data Security in BFC during this project is provided by using AES algorithm. Encryption is that the method of changing plaintext to cipher-text through applying mathematical transformations. These transformations are called encoding algorithms and they need an encryption key. Decryption could be a reverse method of getting the initial data from the cipher-text using a decryption key.

AES depends on a design principle known as Substitution permutation network. It is fast in software as well as hardware. It has a fixed block size of around 128 bits and a key size of 128, 192, or 256 bits. AES can care for a 4*4 matrix of bytes called state. Most of the

AES computations are performed in a very special finite field. Every round involves a number of process steps as well as the one that depends on encoding key. Collection of reverse rounds is applied in order to transform cipher-text to the original text by using an equivalent encoding.

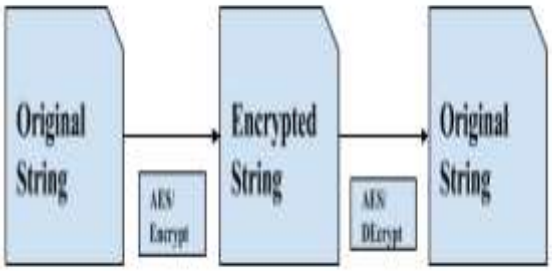


Figure 2: Encryption and Decryption

Data confidentiality is one of strict needs of cloud storage system. In order to produce security service such as confidentiality within the cloud services will use SHA algorithm. In this proposed system, the SHA is used for secure transmission of information.

D. Comparison:

In Cloud storage, various files may contain redundant data. This may lead to huge quantity of data being transmitted between user and the server. One such solution which can overcome this drawback is to use further storage devices in order to enhance the present communication. However, this may result in increased operation costs and time for the organization.

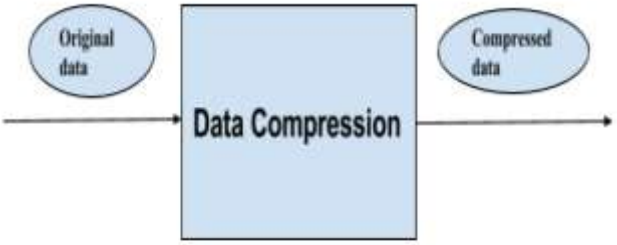


Figure 3: File Compression

One best resolution in order to overcome the problem of increased data storage and information transfer is by using efficient code. By implementing information compression, there is reduction in storage in the cloud thereby, enhancing the utilization of storage space. We will also say that the speed of data transfer will increase as compressed data is transferred between user and server systems.

ISSN (Online) 2394-2320

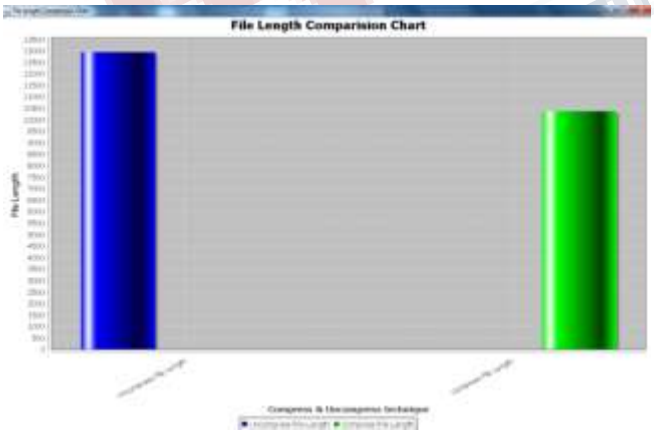
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 8, August 2016

IV. EXPERIMENTAL RESULTS

In our experiments, any number of users can registered and login into the system. Who are authorized users they can upload the files into the cloud. Those uploaded files are stored in chunk format in cloud. If any duplicate files are available in cloud, then that file cannot uploaded in the cloud and to that particular file reference will be assigned to the user. But that file can be downloaded by data owner as well as data users.

In the below image we can observe that complete status of uploaded files.

In the below image we can observe that difference between the length of both compressed and uncompressed files.



When we are trying to upload large files, in order to reduce space we first compress and store in the cloud.
We can observe that uncompressed file length is higher than compressed file length. The difference will be shown in the sense of file length. So we can consider that the advantage of file compression.
Through our implementation we can store the big file in chunks format and detect the duplicate files as well as we can increase the storage space of cloud with file compression.

V. CONCLUSION

Big File Cloud Storage, a simple meta-data to make a high performance Cloud Storage supported MYSQL key value store. Each and every file in the system contains a same size of meta-data regardless of file-size. Every big-file hold on in Big File Cloud Storage system is split into multiple fixed-size chunks may accept the last chunk of file. The chunks of a file have a contiguous ID, thus it is easy to distribute data and scale-out storage system, especially once using MYSQL. This research also brings the benefits of key worth store into big-file data store that is not default supported for big-value. The data de-duplication method of Big File Cloud Storage system uses SHA and AES perform and a key store to fast detect data-duplication on server-side with file compression. It is useful to save storage space and network bandwidth when several users upload an equivalent static data.

REFERENCES

[1] Thanh Trung Nguyen · Minh Hieu Nguyen, "Zing Database: High-Performance Key value Store For Large-Scale Storage Service", 17 August 2014, Springer - Vietnam J Comput Science (2015), DOI 10.1007/s40595-014-0027-4.

[2] Mihir Bellare, Sriram Keelveedhi, Thomas Ristenpart, "DupLESS: Server-Aided Encryption for Deduplicated Storage", 2013, USENIX Security Symposium.

[3] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", Google, Inc.

[4] P. FIPS. 197: the official aes standard. "Figure2: Working scheme with four LFSRs and their IV generation LFSR1" LFSR, 2, 2001.

[5] 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.

[6] 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.

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

[8] Martin Placek, Rajkumar Buyya, "A Taxonomy of Distributed Storage Systems".

[9] 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.

[10] P. ONeil, E. Cheng, D. Gawlick, and E. ONeil. The log-structured merge-tree (lsm-tree). Acta Informatica, 33(4):351–385, 1996.

[11] M. Placek and R. Buyya. A taxonomy of distributed storage systems. Reporte tecnico, Universidad de Melbourne, Laboratorio de sistemas distribuidosy computo grid, 2006.

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

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

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

[15] M. Szeredi et al. Fuse: Filesystem in userspace. Accessed on, 2010.

[16] 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.