ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 9, September 2017

Efficient Cloud Storage System with Highly Scalable and Security for Big File

H. Bindu Madhavi, R.Saravanan

M.Tech(Information technology), VIT University, Vellore, India

Assistant Professor, Dept. of IT., VIT University, Vellore, India

ABSTRACT: Cloud storage services are growing at a fast rate and are emerging in data storage field. These services are used by people for backing up data, sharing file through social networks like Facebook [3], Zing Me [2]. Users will be able to upload data from computer, mobile or tablet and also download and share them to others. Thus, system load in cloud storage becomes huge. Nowadays, Cloud storage service has become a crucial requirement for many enterprises due to its features like cost saving, performance, security, flexibility. To design an efficient storage engine for cloud based systems, it is always required to deal with requirements like big file processing, lightweight metadata, deduplication, high scalability. Here we suggest a Big file cloud architecture to handle all problems in big file cloud system. Basically, here we propose to build a scalable distributed data cloud storage that supports big file with size up to several terabytes.]. Current cloud storage services have a complex metadata system. Thereby, the space complexity of the metadata System is O(n) and it is not scalable for big file. In this research, new big file cloud storage architecture and a better solution are implemented to reduce the space complexity of metadata. So that scalability was improved in Cloud Storage.

KEYWORDS: Cloud Computing, data storage, reliability, Scalability, data security.

I. INTRODUCTION

Cloud based storage serves several users with storage capacity and each user will reach various terabytes of data. The Cloud Storage providers are in charge of keeping the information accessible and available and the physical environment to be ensured and running. Cloud storage is used by several people in many cases, like backing up data, share file to others through social networks such as, Facebook [3], zingme [2]. Uploading data from different devices like computer, mobile phone or tablet and then downloading or sharing to others. Thereby, the system load in cloud storage is huge. So, in order to assure excellent quality service for users, the system has to deal with numerous requirements like efficient storage and management of big files, data deduplication so as to reduce the wastage of storage space which is due to storing the same static data from different users. In conventional systems, there are many challenges faced by the system in order to manage large number of big files like, scale system for incredible growth of data, distributing data in large number of nodes, load balancing, fault tolerance et c. In order to solve these problems a common method used in many cloud storages is dividing big file to smaller chunks, storing chunks on distributed nodes and managing them by using a metadata system. [1], [6], [19], [21]. In one such cloud storage system Dropbox [1], metadata size will proportionately increase with the original file size, leading to difficulty when the file size is big. Thereby, the current cloud storage services have a complicated metadata system that, the file size of every file is directly proportional to the size of metadata at least. The space complexity of this metadata system is O (n) in Dropbox [1], HDFS [21].

Thereby, the space complexity of these meta-data system is not scalable for huge files. In this research, a new big-file cloud storage architecture and improved solution to reduce the space complexity of metadata is suggested. Here, in BFC we came up with a solution where the metadata size is independent of the number of chunks regardless of any file size: small or big. In case of extension of my project, file compression is used, where we first compress a huge file and then store on cloud server.

DOI: 10.15680/IJIRCCE.2017. 0509011

II. RELATED WORK

The quality of performing extraordinarily well, better, faster and more efficient than others is high performance. If we try to measure the high performance in processors we will see which processor works faster than the other. Based on which processor works faster, that one will have high performance. High performance will mainly depend on the response time and query execution time.

2.1 High Performance: Here, in this project in order to perform data deduplication we are using metadata of the files which are stored on the cloud. BFC supports lightweight metadata system wherein, the size of metadata in BFC is smaller than the existed cloud storage systems like Dropbox [6]. Thereby, it will be easier for us to check data duplication at the server side so; it will result in reducing the execution and response time of the query. The metadata in BFC consists of first chunk and last chunk that's it whereas, in existed cloud storage systems like Dropbox, the metadata consists of series of all chunks i.e., from first chunk to last chunk. This can be seen in the metadata size comparison chart between existing and proposed system in the figure 1 below. Thereby, the limitations of the previously existing cloud storage systems: Dropbox [6], Google Drive [5] like metadata complexity, data deduplication are overcome by the proposed system BFC in this project. Additionally, file compression [11] discussed in 1.6 below is also implemented as project extension which will result in reducing storage requirements. The compressed file and normal file comparison is shown below in figure 2. Also, the transmission of compressed data over a medium will be resulted in an increase in the rate of information transfer. The project is extendable to accommodate file decompression[11]. Therefore, we can conclude the proposed system as high performance.

2.2 Distributed System: A system is said to be a distributed system if it has ability to share resources among multiple systems which can also be in different locations in order to serve millions of users. The project BFC is a distributed application [18] with a client-server architecture explained in the Distributed system section 1.4.4. Here, in this project a server application (Logical layer) and a client application (Application layer) are developed, we will be able to run the server application in one system and client application in many systems which are connected through Local Area Network (LAN).

2.3 Scalability: A system is said to be scalable when it has the potential to expand in order to accommodate the growing amount of work. . The scalability of the system can be measured through an algorithm, design, and program. A system is scalable when it is efficient and practical when applied to situations like large input data set, a large number of outputs or users, in order to conclude whether a system is efficient and practical when applied to situations like large input data set, a large number of outputs or users, or a large number of participating nodes in the case of a distributed system.

In BFC, file-id and chunk-id are integer keys that can be automatically incremented. A simple hash function hash (key) = key is utilized for consistent hashing [27]. In this case, it will be easier in order to scale-out the system.

III BFC ARCHITECTURE

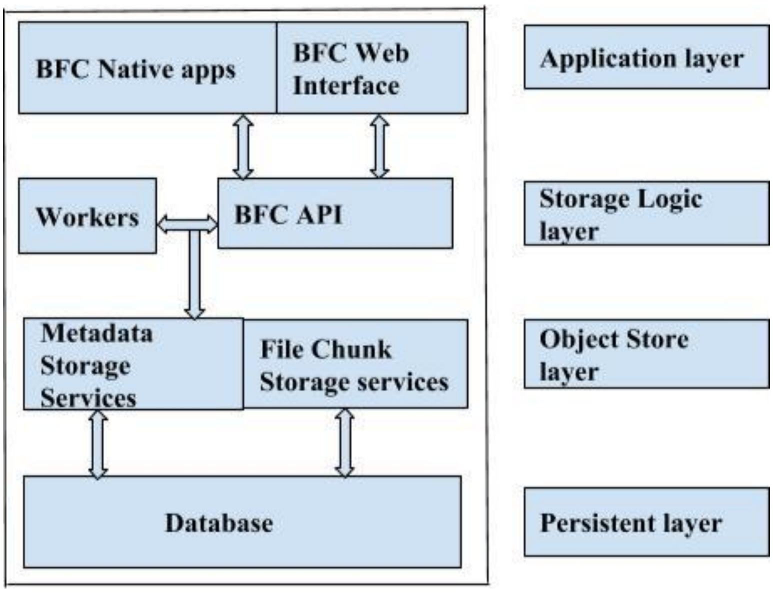
BFC Architecture [8] comprises of four layers namely: Application layer, Storage Logical layer and Object Store layer. Each layer comprises of various coordinated components. Application layer consists of indigenous software on desktop systems, mobile devices and web interface, which grants user to upload, download and share their own files. This layer utilizes API which is from Storage Logical layer and applies various algorithms for the purpose of downloading and uploading process.

Storage Logical layer contains numerous queuing services and worker services, ID Generator services and all logical API for Cloud Storage System. This layer will implement business logic part in BFC. The most essential components of this layer is upload and download service. This layer will provide a CloudApps [28] service which will serve users requests. The Storage logical layer will stores as well as recovers information from Object Store Layer.

DOI: 10.15680/IJIRCCE.2017. 0509011

Object Store Layer is the most essential layer which stores and caches objects. This layer maintains data of all objects in the system which will include client data, metadata in particular. It consists of many distributed backend services. Object Store Layer has two important services namely, FileInfoService and ChunkStoreService. Information of files is stored in FileInfoService. ChunkStoreService will store data chunks which are built by splitting the original files that client has uploaded.

In BFC system, we achieved few enhancements to make low complicated metadata. Metadata represents a file and how it is composed as a list of small chunks. The fundamental element in the BFC cloud storage system is chunk. Here, the original large file which is uploaded by the client is split into a list of chunks. This brings a great deal of advantages. Above all, it is simple to store, distribute and replicate chunks, efficient storage of small chunks, also uploading and downloading file parallel and resumble. It is hard to achieve the above benefits with a huge file in local file system.



BFC Architecture

3.1 Metadata

Metadata [23] is the data which describes about original data. It consists of a series of elements, and each element has information such as, chunk size, hash value of chunk. The length of the series will be equivalent to the number of chunks from file, making it difficult when the file size is large. BFC suggested a solution in which the size of metadata will not be dependent on the number of chunks re ga rdle ss of file size (very small file or big file). Here, the solution suggests to store the id of the first chunk, and the number of chunks which are generated by splitting original file. Since the id of chunk is progressively assigned from the first chunk, we can undoubtedly calculate the ith chunk id by the following formula: Chunkid[i] = fileInfo.startChunkID + i.Metadata is primarily represented in FileInfo structure which comprise of following

fields: *fileName* (name of the file); *filedId* (identification of file) *Ref FileID* (Id of file that have earlier existed in the system) *refFileId* is accurate if it is greater than zero. startchunkID will represent the first chunk of file, the next chunk will have id as startChunkID + 1, *numchunk* represents number of chunks, filesize is the size of file in bytes. Status shows the status of the file, it will have one in four states namely, *EUploading* file - when chunks are uploading to the server, *ECompleted* file - when all chunks are uploaded however, it is not checked as consistent.

3.2 Data Deduplication

Cloud storage faces a complicated challenge in overcoming data duplication by eliminating identical data. In BFC data deduplication is utilized. Data deduplication can work on client or server side [17]. In BFC, we have implemented on the server side. Data deduplication is one of the crucial mechanisms for minimizing identical copies of similar data. In order to enhance effective usage of storage space, indistinguishable data are found then only one copy of data is saved and is interchanged with other duplicates with reference that addresses first duplicate. In this way, we can reduce identical file being stored in cloud by different users thereby, reducing data duplication method used to detect duplicate data is SHA2 [22] hash function while uploading

3.3 Data Security In BFC

Data Security in BFC in this project is provided by using AES algorithm.

Encryption is the method of converting plaintext to cipher text through applying mathematical transformations. These transformations are known as encryption algorithms and they require an encryption key. Decryption is a reverse process of obtaining the original data from the cipher text using a decryption key.

AES [4] depends on a design principle called 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 will operate on a 4*4 matrix of bytes known as state. Most of the AES computations are performed in a special finite field. AES cipher [4] is indicated as various repetitions of transformation rounds that will transform the input data into the final output of cipher-text. Each round involves a few processing steps. Including the one that depends on encryption key. A set of reverse rounds are applied in order to transform cipher-text to the original text by using the same encryption.

IV EXISTING SYSTEM

Cloud storage is used by people for daily demands, for instance backing up data, sending file to friends through social networks. Users will also possibly upload data from different types of devices and they can download or share with others. In Cloud storage system load is usually very heavy. Thereby, in order to assure excellent quality of service for users, the system will have to deal with certain problems and requirements.

Disadvantages:

1 Efficiently storing, retrieving and managing big files in the system.

2 Data duplication in order to reduce wastage of storage space which is due to storing same static data from different users.

3 Parallel and resumable upload and download.

V. PROPOSED SYSTEM

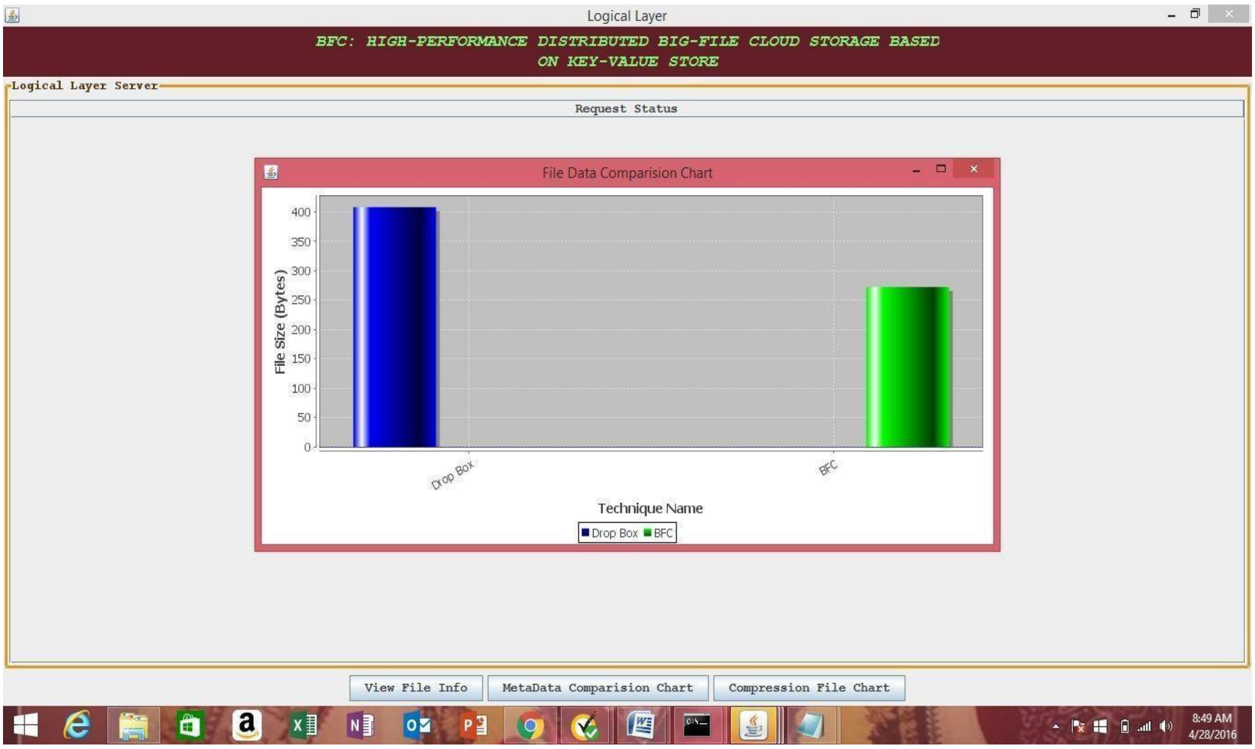
A common method which is used for solving these problems is by dividing big file to multiple smaller chunks, storing them on disks and then managing them by using a meta data system. Cloud storage providers have to face significant problems like, storing chunks and meta-data effectively and designing a light weight Meta data. After a long time, current cloud storage services have a complex Meta data system; Somewhat the size of metadata will be linear to the file size for every file. Thereby, the space complexity of this Meta data system is not scalable for big file. In this research, we implement big file cloud storage architecture and also a superior solution to reduce the space complexity of meta-data.

Advantages:

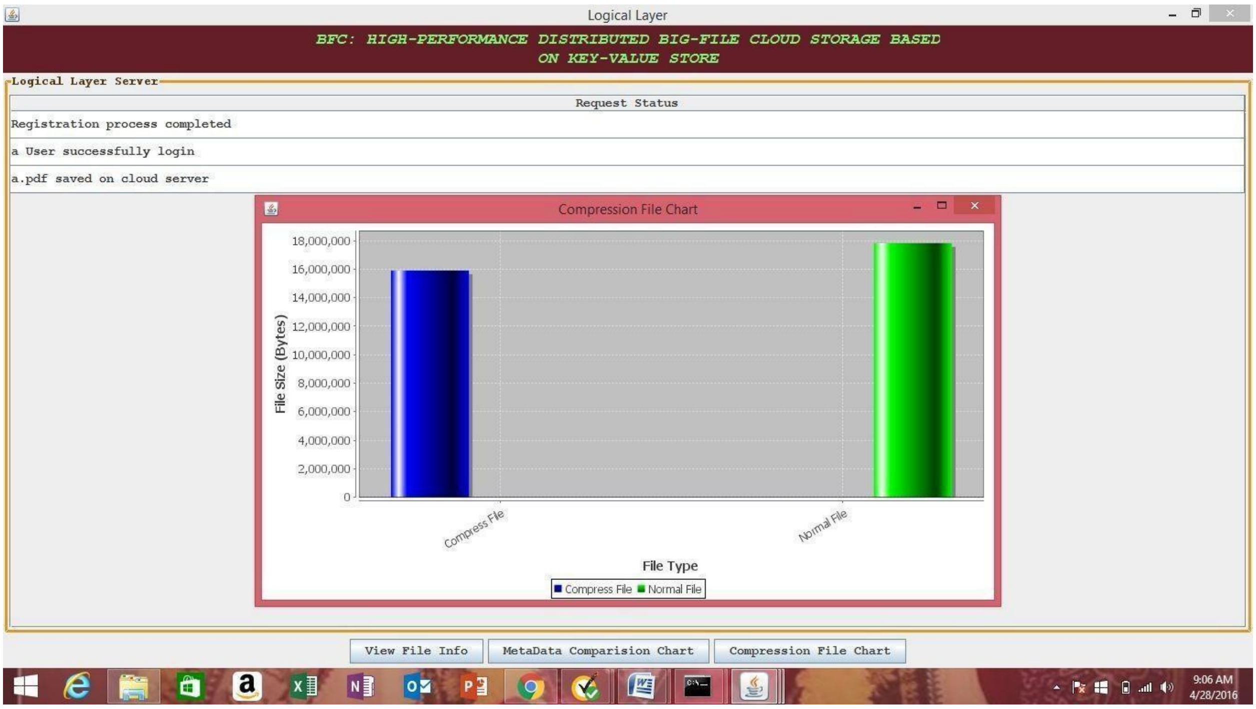
- A lightweight metadata design for big file. Every file has approximately the same size of metadata.
- A logical contiguous chunk-id of chunk collection of files that makes it manageable in order to distribute data and scale out the storage system.
- File compression which overcomes the problem of increased data storage and information transfer.

VI.SIMULATION RESULTS

Metadata and Dropbox comparison at the Logical layer home screen



When we are trying to upload large files, in order to reduce space we first compress and store in the cloud. Now let us see Compression File chart.



BFC: HIGH-PERFORMANCE DISTRIBUTED BIG-FILE CLOUD STORAGE BASED ON KEY-VALUE STORE

Logical Layer Server

Request Status
Registration process completed
a User successfully login
a.pdf saved on cloud server

VII.CONCLUSION

BFC has designed a simple metadata in order to create a high performance Cloud Storage. Every file in the system has a same size of metadata independent of file size. In BFC every big-file is split into numerous fixed size chunks. The chunks of a file have a contiguous ID range; thereby it is easy to distribute data and scale out storage system. The data de-duplication method of BFC uses SHA-2 hash function to speed up in order to detect data-de-duplication on server side. It is important to save storage space and network bandwidth when various users upload same static data. Compression is implemented as the extension of the project. The project is extendable to accommodate decompression.

REFERENCES

[1] Dropbox tech blog. https://tech.dropbox.com/. Accessed October 28, 2014.
[2] Zing me. http://me.zing.vn. Accessed October 28,2014
[3] Facebook. http://facebook.com, 2014
[4] Federal Information Processing Standards Publication 197 ADVANCED ENCRYPTION STANDARD (AES) November 26, 2001
[5] 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.
[6] Yan Kit Li, Xialofeng Chen, Patrick P.C.Lee Secured Authorized De duplication Based Hybrid Cloud Approach.
[7] Oraclehttp://www.oracle.com/technetwork/articles/java/compress-1565076.html
[8] SDLC http://www.veracode.com/security/software development lifecycle
[9] http://archive.thoughtsoncloud.com/2014/05/explained dynamic cloud kids/
[10] http://www.thoughtsoncloud.com/2014/03/ what is software as a service saas/
[11] http://archive.thoughtsoncloud.com/2014/05/building hybrid cloud 3 ways dynamic cloud Powers innovation
[12] Divyakant Agrawal, Amr El Abbadi, Sudipto Das, and Aaron J. Elmore, Database Scalability, Elasticity, and Autonomy in the Cloud.