International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

Database Management and Storage Optimization Using Data De-Duplication and Fog Computing

Ashutosh Avadhani[1], Amruta Chaudhari[2], Aniket Powar[3], Ishwar Borse[4], Mrs. M.D. Sale[5]

[1,2,3,4] B.E. (Computer Engineering), Sinhgad College of Engineering, Pune, Maharashtra, India
[5]Assistant Professor, Dept. of Computer Engineering, Sinhgad College of Engineering, Pune, Mahrashtra, India

---

Abstract – *Data generation has shown a huge rise in the last few years, thanks to boom of internet services. Multiple file formats have been created to store and share the data. Storage and sharing of data has also changed over the years. Earlier there was traditional model, then distributed databases, then came cloud services. These models were effective, but in the future, they will need to go beyond and therefore De-duplication and Fog computing are introduced. Thus, our efforts to contribute, we have created college level fog system. This system will help teachers and students to upload and download files with each other seamlessly and without the help of internet. This system is designed to be low cost and expandable. The aim of this project is to build an android application which will store only unique contents and de-duplicated files from the server which can be accessed from anywhere in the campus.*

Key Words: Data Deduplication, Fog Computing, Raspberry Pi server, SEED Block Algorithm, Block Level Chunking.

1. INTRODUCTION

Today in the digital age, data is considered as the most important entity and therefore steps are being taken to store all kinds of data efficiently. The need of data storage has reached to such level that traditional methods are soon going to be obsolete and even according to tech experts it's being difficult for the cloud architecture to sustain for long. To tackle these problem Fog computing was introduced and new deduplication methods were found.

Fog computing, also known as fog networking or fogging, is a decentralized computing infrastructure in which data, compute, storage and applications are distributed in the most logical, efficient place between the data source and the cloud. It essentially extends cloud computing and services to the edge of the network, bringing the advantages and power of the cloud closer to where data is created and acted upon.

Data Deduplication, often called intelligent compression or single-instance storage, is a process that eliminates redundant copies of data and reduces storage overhead. Data Deduplication techniques ensure that only one unique instance of data is retained on storage media, such as disk, flash or tape. Taking this into account and learning few basics of Raspberry pi we have designed an efficient, portable and cheap alternative to the traditional bulky server machine to design our very own server for use at college level which can be operated in intranet as well as internet medium.

For our proposed system, we have taken the case study of our college and developed an android application for teachers and students to share college data with each other without having to physically transfer files through traditional methods such as pen-drives or placing the files in centralized FTP server.

System consists of Raspberry Pi as our backend server which will store files, Android App as our front end and an HTTP website for admin purposes.

2. LITERATURE REVIEW

[1] The idea proposed in paper [1] was to set up a private cloud server using raspberry pi and to be used for storage application. Raspberry Pi is a cheaper microprocessor in which cloud computing infrastructure can be obtained. This paper gives us a detailed procedure of how to set up a cloud server for different types of cloud related services.

[2] The idea proposed in paper [2] was related to the security of the data stored on the server and multiple methods retrieving the data were proposed. This paper compares different search methods of the encrypted data such as, Single Keyword Search, Multi-keyword Search, Fuzzy Keyword Search, Conjunctive Keyword Search, Similarity Search and Synonym Search. Also, this paper investigates the various aspects of data sharing on basis of user revocation, competency, encryption techniques, identity privacy and key distribution. Plutus, Sirius, Secure scalable data access scheme, improved proxy encryption and Multi-Owner Data Sharing are briefed based on the above mentioned significant parameters.

[3] The idea proposed in paper [3] was to design a system which is beneficial to client as well as storage provider. This paper proposes an algorithm called Temporal Data Deduplication Algorithm to get better performance of data storage in cloud.

[4] The idea proposed in paper [4] was to give different chunking methods in data deduplication. The paper provided with different chunking types and methods which were helpful to finalize the chunking method.

[5] The idea proposed in paper [5] was to tackle the issue of security in data deduplication present in the cloud storage. This paper proposes a method PerfectDedup, a novel method for secure data deduplication, which takes into account the popularity of the data segments and leverages the properties of perfect hashing in order to assure us deduplication as well as data security as well. The method also sees to it that client IDE data overhead is minimal while all the work happens at the server side.

[6] The proposed method in paper [6] was a deduplication protocol designed for private files. Intuitively, a private data deduplication protocol allows a client who holds a private data proves to a server who holds a summary string of the data that he/she is the owner of that data without revealing further information to the server. The security of private data deduplication protocols is formalized in the simulation-based framework in the context of two-party computations. A construction of private deduplication protocols based on the standard cryptographic assumptions is then presented and analyzed.

[7] This paper [7] proposes a deduplication system called Smart Deduplication for Mobiles (SDM) for mobile devices. SDM chooses the best deduplication method without specific configurations for any file type. SDM achieves higher deduplication accuracy and faster deduplication speed over existing systems for mobile devices with negligible additional time in most scenarios. The additional time required by our system is negligible when compared to the power saved from uploading fewer data to a cloud storage.

[8] In this paper [8], an Android based home automation system that allows multiple users to control the appliances by an Android application or through a web site is presented. The system has three hardware components: a local device to transfer signals to home appliances, a web server to store customer records and support services to the other components, and a mobile smart device running Android application. Distributed cloud platforms and services of Google are used to support messaging between the components. The prototype implementation of the proposed system is evaluated based on the criteria considered after the requirement analysis for an adequate home automation system.

3. SYSTEM OVERVIEW

Our system is an android application which has been designed for college purposes. Therefore, the App GUI has been designed to be efficient and easy to access with minimal complexity as possible. The system is consists of three users i.e. Teacher, Student, Admin. Teacher and Student will have the functionality of Login, Registration, Upload/Download files. The Admin will have the ability to delete and view the Deduplication Statistics.

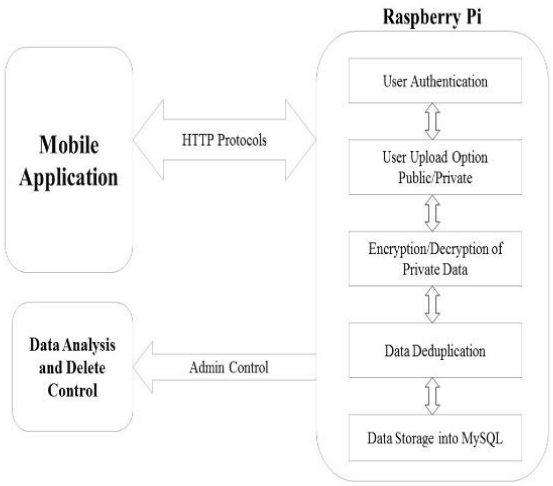

Fig. 1: System Flow

4. SYSTEM ARCHITECTURE

The proposed system consists of three main modules:

1. Fog Computing.
2. Data Deduplication.
3. Encryption/Decryption.

4.1 Fog Computing

Fog computing is basically using advantages of cloud near the edge of the network. The advantage of using Fog is that it helps in reduced communication time for the user and reduces the dependencies on the cloud. The architecture of any Fog network is same as that of the cloud only the difference is that it doesn't require any internet services specifically. Our system can work on both intranet as well as Internet. Fog network has been established using a Raspberry Pi as our server and using Socket Programming between android and server to communicate with each other. HTTP protocols have been used to maintain these sockets. Parallel programming has been done to allow multiple user to communicate with the server simultaneously.

4.2 Data Deduplication

In simplified terms, data deduplication compares chunk of data and removes objects (copies) that already exist in the data set. The deduplication process removes blocks that are not unique. Simply put, the process consists of four steps:

- Divide the input data into blocks or "chunks."
- Calculate a hash value for each block of data.
- Use these values to determine if another block of the same data has already been stored.
- Replace the duplicate data with a reference to the object already stored in the database.

Once the data is chunked, an index can be created from the results, and the duplicates can be found and eliminated. Only a single instance of every chunk is stored. Every block or chunk of data stored on the disk is mathematically hashed and the hashes are indexed. Every new block of data being stored is also hashed, and the hashes are compared in the index. If the new data hash matches a hash for a block already stored, the new data does not get stored, thus eliminating duplicates. For the proposed system Block Level Chunking algorithm is used. The proposed steps for Deduplication to happen in our system are as follows: -

- While uploading, the file gets divided into blocks or "chunks."
- Calculate a hash value for each chunk using MD5 algorithm.
- Then check whether the uploaded chunk is present in MySQL database.
- If present, then create a reference id for the already present chunk and store only the reference id along with the file ID.
- If the chunk is not present, then enter the chunk into the table along with file ID and reference ID of the file.

While downloading any file from the server, the system works in the reverse way. Reference Id is the id which is assigned to every chunk and this Id points to the metadata of the file.

4.3 Encryption/Decryption

For encryption and decryption purposes, we have used SEED Block Algorithm. Seed algorithm uses 128 bit block and 128 bit key technology for encrypting the data. The seed is a key-value pair where key is a random number generated uniquely for every user and it is XORed with the hash value of every chunk.

The steps of Encryption are as follows:

- User will log-in into private upload/download mode.
- Once the file gets uploaded and distributed into chunks, then the seed number and the hash value of every chunk will be XORed with each other and an encrypted chunk of data will be created.
- While downloading, only that user will be able to view that file and the same process will be carried out in reverse.

5. MATHEMATICAL MODEL

Let 'S' be the "College Level Fog System"
S= S1, S2, S3,. . ., Sn
Set S is divided into 7 modules
S1= Request Handler (RH)
S2= Request Validator (RV)
S4= Configuration Manager (CM)
S5= Response Generator (RG)
S6= Database Manager (DM)
_ Identify the inputs as I. Inputs = X1, X2, X3, ...Xn
X1= Files to upload
_ Identify the output as O.
Outputs = Y1, Y2, Y3, ... Yn
Y1= File storage

6. METHODOLOGY

6.1 Login and Registration

For log-in and registration, socket programming has been used for the communication between user and the server. The database contains a user details table which will store all the information of the user.

Fig. 2: Registration Page

6.2 Public and Private Upload/Download

Users will have the option of uploading data in public or else private storage space. Public upload basically means that all the users will be able to view the files and will be able to download it. There will not be any encryption on files that are being uploaded to a public domain. Private upload is user specific upload and only that user can upload these files and will be able to view them. All private uploads are encrypted and then stored in the database. They cannot be accessed by any other user and will not be listed in the download option.

Fig. 3: Private-Public Log-in

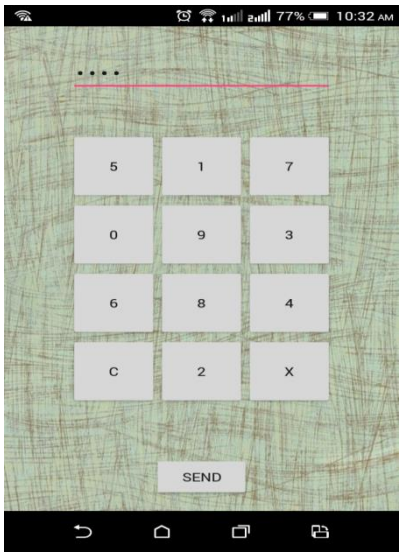

Fig. 4: Private User Authentication

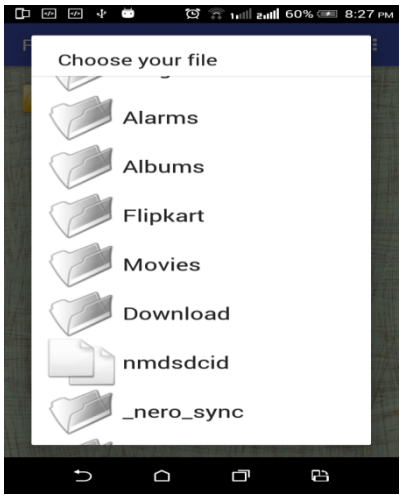

Fig. 5: Upload file

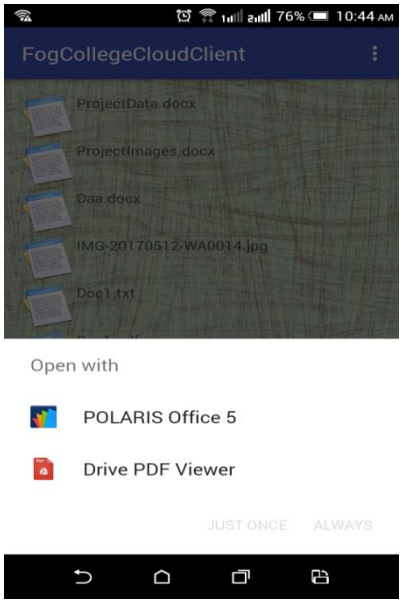

Fig. 6: Download File

5.3 Database Manager

System database contains four tables to store all the system information. They are as follows:

- File Table.
- FileDetails Table.
- FilePart Table.
- UserDetails Table.

5.3.1 File Table

This table will contain the name of the file, size of the file, unique file ID and user ID who has uploaded the file. This table helps in identifying who has uploaded the file and while downloading the name of the file helps in identifying the file chunks and size of the file.

5.3.2 FileDetails Table

This table contains the chunk ID of the file and duplicated chunk ID. This table helps to determine the number of deduplicated chunks and unique chunks.

5.3.3 FilePart Table

This table stores the hash values of all the chunks uploaded to the file system. These values are unique and all are pointed by the reference ID present in the FileDetails table.

5.3.4 Userdetails Table

This table stores all the user data and information. This table also contains the User ID which is unique to all the users.

5.4 Admin Control

This feature has been added for deletion of any file and for viewing the deduplicated statistics of the system.

6. CONSTRAINTS

The phone should be an android device. Since we are using Raspberry Pi, the bandwidth of the device is less. Therefore it results in slightly less speed while uploading the data. The deduplication rate falls when less number of files are uploaded or if files of totally different formats are uploaded.

7. RESULTS

The following results were obtained.

1) The speed of uploading file to Raspberry Pi is slow but can be increased by taking the chunk size larger.
2) The rate of deduplication increases when the chunk size is smaller.
3) Encrypted files cannot be read by any other user, therefore the privacy of the user is maintained.
4) Downloading of file is faster.
5) The rate of deduplication increases by adding multiple files.
6) Deduplication doesn't depend upon the extension of the file but the content of the file, therefore the code can find duplicated chunks between different files as well as same file.

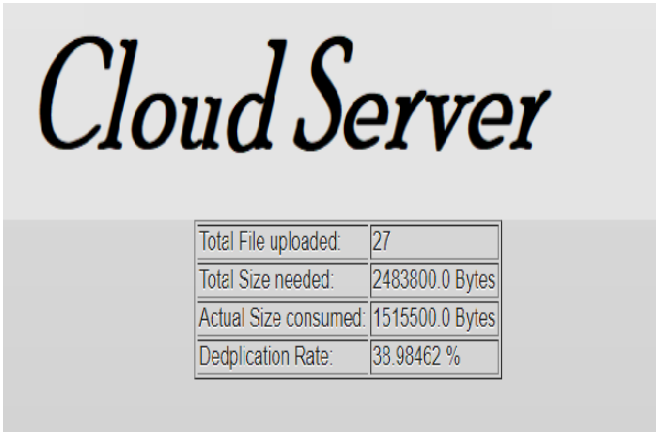

Fig. 7: Deduplication Statistics

8. CONCLUSION

In the proposed system, the notion of authorized data management system at the college level was proposed by including different privileges for users. Hybrid cloud architecture was used combining fog computing and data de-duplication in the system. An additional module was introduced for individual security purpose. Proposed system is efficient and secure. While upload speed is less than the other client server applications, more efforts have been taken on memory management which ensures better rate of de-duplication. It uses intranet networking for a private college. The designed system is an application consisting of private and public log-in.

REFERENCES

1. Princy, S. Emima, and K. Gerard Joe Nigel. "Implementation of cloud server for real time data storage using Raspberry Pi" Green Engineering and Technologies (IC-GET), 2015 Online International Conference on. IEEE, 2015.

2. Raghavendra, S., et al. "Survey on Data Storage and Retrieval Techniques over Encrypted Cloud Data". International Journal of Computer Science and Information Security 14.9 (2016): 718.

3. Muthurajkumar, S., M. Vijayalakshmi, and A. Kannan. "An effective data storage model for cloud databases using temporal data de-duplication approach". Advanced Computing (ICoAC), 2016 Eighth International Conference on. IEEE, 2017.

4. A.Venish and K. Siva Sankar. "Study of Chunking in Data Deduplication". Proceedings of the International Conference on Soft Computing Systems, Advances in Intelligent Systems and Computing 398:2016.

5. Pasquale Puzio, Re_k Molva, Melek Onen, Sergio Loureiro SecludIT, Sophia Antipolis, Sophia Antipolis. "PerfectDedup: Secure Data Deduplication."

6. Wee Keong Ng, Yonggang Wen, Huafei Zhu. "Private Data Deduplication Protocols in Cloud Storage."

7. Ryan N. S. Widodoa, Hyotaek Limb,*, Mohammed Atiquzzaman. "SDM: Smart Deduplication for Mobile Cloud Storage."

8. Alper Gurek, Caner Gur, Cagri Gurakin,Mustafa Akdeniz, Senem Kumova Metin, Ilker Korkmaz. "An Android Based Home Automation System."