DDEAS: Distributed Deduplication System with Efficient Access in Cloud Data Storage

Bhavya M, Thriveni J, Venugopal K R

Abstract*— Cloud storage service is one of the vital function of cloud computing that helps cloud users to outsource a massive volume of data without upgrading their devices. However, cloud data storage offered by Cloud Service Providers (CSPs) faces data redundancy problems. The data de-duplication technique aims to eliminate redundant data segments and keeps a single instance of the data set, even if similar data set is owned by any number of users. Since data blocks are distributed among the multiple individual servers, the user needs to download each block of the file before reconstructing the file, which reduces the system efficiency. We propose a server level data recover module in the cloud storage system to improve file access efficiency and reduce network bandwidth utilization time. In the proposed method, erasure coding is used to store blocks in distributed cloud storage and The MD5 (Message Digest 5) is used for data integrity. Executing recover algorithm helps user to directly fetch the file without downloading each block from the cloud servers. The proposed scheme improves the time efficiency of the system and quick access ability to the stored data. Thus consumes less network bandwidth and reduces user processing overhead while data file is downloading.

Keywords: Access Efficiency, Cloud Data storage, Data Deduplication, Network Bandwidth, Recovery module.

I. INTRODUCTION

Cloud computing is a computing prototype, where a massive pool of systems connected in public or private network. It provides a dynamically scalable framework for data, application, and data storage. With this approach, there is a significant reduction in computing cost and application hosting. Cloud computing has enhanced prevalent services and become a trusted service platform due to many desirable properties [1][2], such as fault tolerance, pay-per-use, elasticity, and scalability.

Cloud data storage system is one of the most extensively consumed cloud services. Cloud users have more benefits from cloud storage services, and they can store a large amount of data without enhancing their systems. The user can access the stored data in any place at any time. Cloud data storage provided by Cloud Service Providers (CSPs) still have some problems like data redundancy and data Prevention.

Data de-duplication is a technique that is used to eliminate duplicate copies of data stored in cloud storage. With data de-duplication technique, one can save a unique copy of the data file and then provide a pointer to the other user, who tries to upload a similar data file.

In the proposed work, we implemented a data de-duplication technique to determine data redundancy and eliminate redundant data, thereby minimizing the back-end storage space consumption. Here we implemented both file level and block level data de-duplication to improve system efficiency. For example, consider an organization with the virtual desktop environment with hundreds of similar workstations, all stored on an expensive storage system. We are thus managing a hundred copies of Windows 8, office 2013, ERP software, and some other tools. Each workstation consumes 25GB disk space, and for hundreds of workstations, the required storage would be 25 TB. When Security falls into two classifications: security problems challenged inevitably with cloud providers additionally security problems looked toward their customers. A few security problems are associated with cloud data service: not just all inclusive security hazards, for instance, listening in, unlawful intrusion, forswearing of administration assaults, organization assaults, yet additionally specific Distributed computing hazards, for cases, side-channel assaults, and sick usage regarding cloud service. There is numerous approaches to give data security. The most well-known and generally utilized method is "encryption." Data encryption is the process of changing plain text into cipher text using encryption algorithm. Each workstation storage is backed up to the cloud. We need 25TB of storage space to store all these data. But with deduplication technique, only single instance of data is stored in the cloud and rest of the data instances are replaced with a pointer to a stored data.

Hence the data deduplication reduces the storage overhead. Therefore data deduplication efficiently stores the data. Data Deduplication model in cloud computing is as shown in Fig 1.

Revised Manuscript Received on December 15, 2019.

Bhavya M, Department of Computer Science and Engineering UniversityVisvesvaraya College of Engineering Bengaluru - 560001, India

ThriveniJ, Department of Computer Science and Engineering University Visvesvaraya College of Engineering Bengaluru - 560001, India

Venugopal K R, Vice Chancellor Banglore University, Bengaluru - 560056, Indi

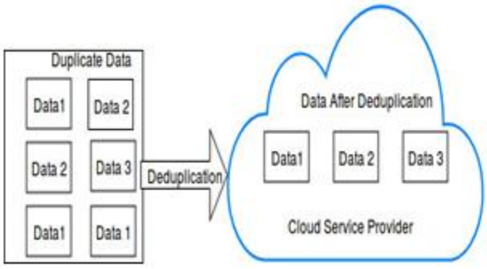

Fig. 1. Data Deduplication model in cloud computing.

Fig. 1. Shows basic deduplication model, and how data represented while uploading to cloud storage server. Data deduplication techniques help to filter the similar data contents before uploading, and the single data file uploaded into the cloud data storage system. There are different types of deduplication, such as client-side deduplication, server-side deduplication, file-level deduplication, block-level deduplication, source-deduplication, etc. In this paper, file and block-level deduplication implemented.

1. File-level deduplication: This method of deduplication compares the entire file based on a tag value of a record to avoid storing the same data.

2. Block-level deduplication: In this method, the file is divided into a fixed number of blocks, and each block is compared based on the tag value of the intersection with existing metadata, to avoid storing the same neighbourhood’s.

For data prevention, tag generation algorithm is used to generate a tag value of the file and the block. The tag value is identical for the similar data even if the file name is different. The convergent encryption technique is used to perform secure data deduplication. In traditional encryption methods, the same plaintext produces different cipher text with different keys. But by using convergent encryption, the same cipher text is generated for the same plaintext encrypted at different times. The storage efficiency of the data achieved by storing each block at individual servers. In this case, even if one or two servers go down, the remaining data can be recovered.

As data stored on multiple servers, the user must access all these servers to recover the original data. This process requires more time, and it also increases the system workload. In this paper, we proposed a module to retrieve the data on the server side. This method has the following advantages:

1. The user needs not to access the multiple servers to recover the information.
2. It increases system efficiency and decreases the workload of the system.
3. A user obtains the file/data directly
4. Reduce the critical sharing overhead.

A. Motivation

Distributed data deduplication is a technique in the cloud data storage system to eliminate the duplicate copies of data before uploading it into a cloud storage system. In the existing scheme [3], while downloading the file, users have to download each block separately and run recover algorithm, and user needs to maintain blocks, Tag values that consumes more network bandwidth and file downloading time thus increases the user processing overhead. The main issue is to minimize network bandwidth, file downloading time and to reduce user processing overhead.

B. Research Contribution

A new scheme is proposed, the Recover algorithm in the metadata server to collect blocks of data and send it to the user for a single file request. A Server Side Retrieval Algorithm is implemented and the results are compared with the existing system. The proposed method DDEAS aims at:

1. Faster data retrieval from the cloud storage system.
2. Reducing file downloading time.
3. Minimizing the network bandwidth utilization,
4. Reducing the user file downloading processing overhead.

C. Problem Definition

Given that file, F split into n blocks (b_1,b_2.....b_n) where n is the number of servers and Each block is stored on the individual server. The main objectives are to:

1. Reduce file downloading time
2. Reduce network bandwidth
3. Reduce user processing overhead
4. Improve system efficiency by fast retrieving of the data from the cloud storage

D. Organization of this paper

The paper is organized as follows. Sections II discuss the literature survey of the work, and discuss the related work that used for the implementation. Section III shows the background work, which describes the primary three modules used in the proposed system. Section IV presents the system model with description and workflow. Section V presents the algorithms of the proposed work. Section VI shows the performance evaluation of the proposed and existing system. Finally, conclusions are given in Section VII.

II. RELATED WORK

Data storage in a distributed storage system using erasure code requires less redundancy. Erasure code splits data into fragments and spread across nodes. When a node failure occurs, it has to be replaced by a another node by downloading sub-sets of data stored from the number of remaining nodes. Rebuild a failure node using the downloaded data and store it in the new node. This procedure is not efficient for node recovery. Alexandros *et al*., [4] introduced a notion of regenerating codes, it helps new node to download function of stored data from the remaining nodes. This function helps to reduce the node repair, and also authors introduced regenerating code that achieves optimal trade-off by invoking effective results at any point of time.

Frequent snapshot backup of virtual disks in a virtualized cloud computing improves hosting reliability, but it consumes enormous storage space.

To eliminate the irrelevant contents in the file, dirty bit based technique has been used, and it helps to analyze original data between versions and full deduplication with fingerprint comparison at the cost of computing resources. Zhang *et al.*, [5] have proposed a multi-level-selective deduplication scheme under a stringent resource requirement and helps to integrate inner-virtual machine and cross- virtual machine duplicate elimination. This above technique uses current data to simplify fingerprint analogy, and reduce price and offset global and local deduplication to improve equivalence. By adopting a minimal number of cloud resources, the proposed system can achieve high deduplication ratio and increase reliability.

Every day huge data is produced and restored by users over many devices in big-data; it is a complex task to coping multiform data in real time. Distributed data centers are using Hadoop Distributed File System (HDFS) to handle with these vast data. To increase data reliability HDFS storing multiple copies of similar data, which consumes extra storage space. To overcome these problems, Chang *et al.*, [6] have improved the storage utilization of data using dynamic deduplication, which uses HDFS as its file system. The proposed methods help to exploit the storage space sufficiently in the finite storage system and maintain to delete the corresponding copies of data to increase data storage space. In this work, authors try to improve storage utilization of the data center efficiently using HDFS system.

Varalaxmi and D.Venkatesh [7] have implemented data deduplication technique in Hybrid cloud data storage system. In this work, they are designed authorized duplicate check scheme and grant access to the encrypted file. Authors have implemented a prototype for authorized duplicate check scheme and have shown that their proposed system is having less overhead than the convergent encryption.

These days many industries and private sectors upload their data to the cloud storage system. Therefore data encryption is essential due to data leakage. The secure encryption scheme using data deduplication for various cost-effective storage optimizations is inefficient. Stanek *et al.*, [8] have proposed a model that provides less security to highly protected data and more security to unpopular or unprotected/ less protected data using a novel encryption scheme. In this system, encryption takes place on the client side, whereas decryption is client independent. The proposed scheme is secured using the Symmetric External Decisional Deffie-Hellman Assumption (SEDHA).

Erasure coding- Based storage is a method where data is split into *K* number of pieces and stored in *n*-nodes. When there is a node failure, it should be re-constructed by new-node, and new-node needs to download *K* encoded fragments from a subset of the surviving nodes and reconstruct the original file. This process consumes more network bandwidth and leads to network traffic. Network bandwidth is an essential parameter in the distributed storage system. Therefore Wu and Dimakis [9] have proposed new techniques that execute algebraic alignment such that the practical dimension of unwanted information is reduced.

Sun *et al.*, [10] have introduced a novel approach called DeDu for data deduplication in the cloud storage systems. DeDu is not only defined for industry or corporate data backup, it is also for general users who willing to store their data in cloud storage for further usage. They are maintaining the index table to store the hash values of the file while uploading and it's generated link files to keep huge data in a Hadoop Distributed File System. Based on the analysis, they show that fewer the data nodes then higher the writing efficiency, but lower the reading efficiency and visa versa. Authors have achieved higher transmission and improve efficiency and accuracy rate.

Harnik *et al.*, [11] have introduced a new mechanism for data deduplication and explained how deduplication could be applicable for side channel to reveal information about the data contents of other users. For this, they have proposed a simple mechanism that enables cross-use deduplication while significantly reducing the risk of data leakage. By using a randomized solution, the authors tried to provide higher privacy and reducing data leakage.

Cloud data deduplication is a technique used to discard multiple copies of a single file in a cloud storage system to improve memory storage efficiency. Authors identify two drawbacks in deduplication. First is disk bottleneck due to the large size of data index, which consumes the RAM space and leads to less deduplication throughput. The second is it is challenging to eliminate duplicate data among multiple storages. Therefore the existing deduplication techniques are unable to remove the duplicates. Wei *et al.*, [12] have proposed a new modulus called MAD2 to eliminate duplication using four methods: 1) To preserve the data locality in backups MAD2 organizes fingerprints into Hash Bucket Matrix (HBM). 2) MAD2 use Bloom Filter Array (BFA) as quick to index identification of individual data files. 3) To capture and exploit data locality, MAD2 integrated dual cache. 4) For data distribution, MAD2 used DHT-based load balance technique. By using these four techniques, MAD2 is good and better than deduplication techniques. MAD 2 achieves 100 MB/s for each deduplication throughput. Authors also minimize the RAM consumption using three possible solutions.

Data computing resources have provided on demand as utilities. Therefore utility computing is an increasingly important paradigm. The main component of utility computing is a storage system, due to the rapid growth of data, it is essential to eliminate duplicate data storage. The existing techniques are static and are not efficient for dynamic modern systems.

Leesakul *et al.*,[13] have proposed real-time adaptive deduplication systems for utility and cloud computing and helps to observe in real-time for changing the system, user, and environmental behavior. To fulfill a balance between changing in performance, storage efficiency, and fault tolerance are required. The experimental results calculate the fault tolerance and system availability by measuring the mean time of repair MTTR. In this approach, the authors prove that the proposed system is efficient and scalable.

He *et al.*, [14] have explained how data deduplication techniques help to improve system efficiency.

They have demonstrated that increasing data day by day leads to more consumption of energy, storage space, and heat emission. To overcome these problems, they have implemented different levels of deduplication techniques like file level, block level, and byte-level data deduplication that optimize the storage consumption. Here authors try to reduce energy consumption, heat, and data compression.

With the continuous increase in the size of data and number of user and data deduplication is a necessity for cloud storage providers. The implementation cost of deduplication is high in terms of privacy and security challenges. Puzio *et al.*, [15] have proposed a new technique called cloud Dedup to provide efficient and secure storage services. Cloud Dedup uses block-level deduplication and guarantees data confidentiality together. Convergent encryption has used for data encryption, and key management has handled by a new component for each block with deduplication. The proposed method shows that the system overhead is less and which will not effect on computational and storage costs.

Secure deduplication is a challenging issue in cloud storage, and convergent encryption technique used for secure deduplication. Using convergent encryption leads to manage a vast number of keys. Authors addressed the problem of achieving reliable and efficient key management in secure deduplication. Li *et al.*, [16] have introduced a baseline approach in which each user holds individual master keys for encrypting the convergent key and outsourcing them to the cloud. Dekey is presented instead of managing key by the users, and here convergent keys are securely distributed across multiple servers. In this work, authors have proved that Dekey incurs limited overhead in a real environment.

Cloud storage services provide on-demand storage resources, and customers pay for space they consumed. As the increase in data storage usage requirements, cloud storage management is essential. Deduplication technique is used to manage cloud storage efficiently. Deduplication technique helps to eliminate redundant copies of data files, but as of now data deduplication technique is static and having limited applicability in dynamic characteristics of data in cloud storage. To overcome this Leesakul *et al.*, [17] have proposed a dynamic data deduplication scheme for cloud storage service and maintaining the balance between changing fault tolerance and storage efficiently. Authors improved the performance and scalability of the cloud service provider.

In the present technology, data de-duplication is used to improve cloud storage service reliability by maintaining fault-tolerance of hardware and software of the systems. Data de-duplication having inter-file relationship dependencies, it leads to data loss and makes a negative impact on the fault tolerance of the system. For this drawback, Rozier *et al.*, [18] have designed a framework for data analysis methods and a model for data de-duplication. These techniques are useful in analyzing the reliability impact of data de-duplication. The proposed methods satisfy the reliability constraints provided by a user by determining a de-duplication strategy. Bhavya *et al.*, [19] have discussed different approaches to improve cloud data storage efficiency, including data de-duplication.

III. BACKGROUND WORK

A. *Erasure Coding*

Storage systems are of different types that are used for various sectors to store their data. But the general problem is a system failure. System Failures are in different forms, and it might be a complete system crash, content loss, etc. Preventing storage system data from failure is very much necessary. To overcome these problems, Plank J S [20] discussed erasure coding. Erasure coding includes new concepts to the storage system to manage the failure. Erasure coding commonly used in the storage system for data recovery. Consider that storage system has n disks, dividing n disks into k disks. Which holds user data so that $m=n-k$ drives, where m stored the coding information.

In erasure coding, the w-bit word stored in each disk. These w-bit words are d_0, ... ,$d_{k-1,}$ stored in data disks. c_0, ..., $c_{m-1,}$ is the coding words stored in coding disks. These coding words defined as linear combinations of the data words. Here co-efficient is also a w-bit word.

$$\begin{array}{l} C_0 = a\,(0,0)\,d_0 + \ldots\ldots\ldots\ldots + a\,(0,k-1)\,d_{k-1} \\ C_1 = a\,(1,0)\,d_0 + \ldots\ldots\ldots\ldots + a\,(1,k-1)\,d_{k-1} \\ \quad - \\ \quad - \\ \quad - \\ C_{m-1} = a\,(m-1,0)\,d_0 + \ldots\ldots\ldots\ldots + a(m-1,k-1)\,d_{k-1} \qquad (1) \end{array}$$

The encoding requires adding and multiplying words, and decoding involves a long set of linear equation with Gaussian elimination. The encoding process takes the content of k disks from them and calculates the content of the coding devices. The decoding process takes the contents of $k+m$ devices and from them reconstruct the original k disks data. Erasure coding helps to achieve fault-tolerance and improve system performance.

B. *Hash value generator*

A hash function is a function which considers any size of data as an input and produces smaller output with fixed size called digest. The hash function has two features, keyed and un-keyed. A keyed hash function used for message authentication codes and the un-keyed hash function is for modification detection codes. The MD5 (Message Digest 5) is modification detection code, is the most used hash function producing 128-bit hash code. MD5 receives variable size data as input and produces fixed length output. The processing of message blocks has four similar rounds. MD5 hash technique helps to verify data integrity.

The MD5 calculation generally used for hash value creating 128-bits of the hash value. MD5 is one of the various strategies for distinguishing, securing, and certify data. MD5 is first intended to be utilized for the cryptographic hash function. It can, in any case, be used as a checksum to assure information integrity and separated into pieces of 512-bit squares [21].

C. Tag Generation Algorithm

A Tag Generation Algorithm is used to generate the unique tag id to each file and block. In the tag generation algorithm, it takes hash values of the data as input and produces T(H) as output. For the same input hash values, it will generate the same tag id. The generated tag id has stored in metadata server with the corresponding file name, and list of block names belong to it. The same tag id has shared with the data owners and the users who own similar data. A user stores this tag and uses it for the duplicate check.

IV. SYSTEM MODEL

The DDEAS system model is shown in Fig. 2. DDEAS system is having four entities, which involved in this data deduplication system, including the S-CSP (Server side-Cloud Service Provider) and the user. File level and block levels Deduplication mechanism is deployed in DDEAS. Whenever a user wants to upload a file user should perform a file-level duplication check by uploading system generated files unique tag value. If uploading file already exist in the cloud storage, then file pointer is sent to the user to download the file, else system performs the block-level duplication check. In block level duplication check, the user divides the file into fixed number of blocks, calculate the hash value for each block and upload all te hash values to the cloud. The cloud intern check for duplicate blocks by comparing it with the stored table at metadata server. If it exist, the file pointer of the block is sent else it stores the block and its hash value onto the metadata server. Each uploading file data (i.e., a file or a block) associated with a unique tag -value for the duplicate check and is stored in the S-CSP.

The four entities in the system model are:

1. *Data owner*: who uploads the original data for the First time
2. *Metadata storage system:* helps to maintain the index table and run the retrieval algorithm
3. *The secondary storage system*: Have multiple servers in which multiple copies of data are stored
4. *Users*: requests for the file using the file tag value

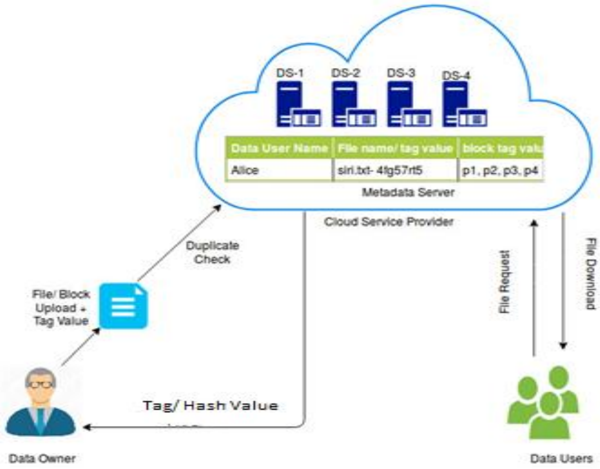

Fig. 2. DDEAS system model

Data Owner: The user who uploads the data or a file for the first time is called a data owner. Data owner plays the major roll in data deduplication. While uploading a file to the cloud storage, data owner generate hash values for the file. The generated hash value is sent to the S-CSPs. S-CSPs will check the index table for the existence of hash value. If file hash value is present in the index table, then S-CSPs will send the reference of existence. Later Data owner splits the file into n number of blocks and generates a hash value for each block. This generated hash value is sent to the S-CSPs and checks index table. If the block hash value is present, then S-CSPs send the reference of all block tags to the data owner. If no file or blocks exists in the cloud storage, then the data owner will generate a hash value for both file and blocks and send it to the cloud storage with tag value.

Metadata: A Metadata server gives information about the stored data in multiple servers. The index table is maintained in metadata server, and it contains complete details about the saved files in the server like filename, file tag value, and a block tag value. If the user tries to upload a new file, then there will be an entry in the index table. If only one or two blocks are unique and the rest of the blocks are duplicated, then the server will store the individual blocks and generate the link to duplicated blocks.

S-CSPs: The S-CSPs allows users to store and retrieve data anytime, having multiple storage servers and provides the data storage services for the users. In the deduplication system, cloud data storage stores unique data. When users try to upload the file already present in the server, then CSPs return the reference pointer and helps to reduce the storage cost at the server side.

User: If same or another user tries to upload the data, cloud system checks for the duplication and provides the tag value to the user. And reduces the file uploading time bandwidth and saves the cloud storage space.

V. ALGORITHMS

Table 1: Algorithm for File Uploading

```
Input: File name
Output: File tag value / File reference link
Step 1: User uploads file and generates hash
        value
Step 2: Send hash value to metadata server
        Check hash value in index table
Step 3: while (true)
            if( hash value present)
                return reference link
                goto Step .5
            else
                split file into n blocks
                generate hash value
                Send hash values to metadata
                server
            end if.
        end while.
Step 4: Store the hash values in index table
        Store data blocks in Distributed
        storage server
Step 5: Finish
```

Table 2: Algorithm for File Downloading

```
Input: File tag value
Output: Original file F
Step 1: User send tag value to the metadata
        server
Step 2: Metadata server checks that tag value
        is present or not in Index table
Step 3: if (tag value present)
            collect all the block tag values of the
requested file
            Request distributed servers to send
            the requested data blocks
            Run the Recover algorithm
        else
            File not present
            goto Step. 5
        end if.
Step 4: Send requested file to user
Step 5: Finish
```

The algorithms for the file uploading and downloading are shown in Table 1 and Table 2 respectively.

The proposed algorithm improved the system efficiency by reducing file downloading time and network bandwidth utilization. File-level and block-level deduplication is performed for duplicate data check before uploading. Once the duplication check is done, the single file is upload in the form of blocks into a secondary storage system, or else pointer of the duplicate file is return to the valid user. In the existing system, the user must send each block's tag value to the server while downloading the file, and once all the blocks are retrieved, the user should run the recovery algorithm to fetch the original file. This process leads to more network bandwidth utilization, high downloading time, and increase user processing time. To overcome these drawbacks, we tried to execute data recovery algorithm in the metadata server. In this technique, the metadata server receives the request from the user. The user sends the tag value of the file to the metadata server, and metadata will check the index table for the tag value. If the tag value is present in the index table, the corresponding file name and blocks name with their tag value is existing. Using that block's tag value metadata server will fetch all the data blocks from secondary storage servers and run the recover algorithm, to return the requested original file to the user.

VI. RESULTS AND PERFORMANCE EVALUATIONS

In this section, the implementation results of DDS and DDEAS are compared:

1. File Downloading Time

File downloading time is defined as the amount of time taken to download a file.

2. Bandwidth Utilization Time

It is the amount of time during which the network bandwidth is busy for transaction. As the time required for file downloading in the existing system is high, the user sends the request along with the tag value to all the servers and all the blocks are received by the user from different servers. Where as in the proposed system user sends only single tag value of the file to the metadata server. Metadata server fetch the blocks from the multiple servers for the requested tag value of the file, run the recover algorithm and return the file to the user. As the recover algorithm is run at the metadata server to generate the original file, the user receives the file in the single request. The downloading time is reduced and the duration in which the bandwidth is utilized is reduced there by reducing the bandwidth utilization time.

In file downloading, DDS follows block level file downloading techniques and DDEAS uses file level downloading technique, which helps to download the file with less time. As compared with the proposed method, the existing process consumes more time to download the file and requires more Network Bandwidth. To overcome this drawback, in the proposed system is executed at the metadata serer. In the existing method, each time the user needs to send the request to fetch the block, which consumes more time compared to DDEAS and thus the file accessing overhead is increased in Table 3.

TABLE 3. File downloading time of DDS and DDEAS.

SL No.	File Size in MB	Existing system Time in ms	Proposed system Time in ms
1	1	23.828	15.568
2	5	219.14	169.14
3	10	337.451	298.438
4	15	448.507	404.156
5	20	512.003	485.575
6	25	589.31	559.34
7	30	672.11	634.61
8	35	753.41	711.32
9	40	836.11	798.51

Table 3. Shows the time consumed to download files with various size by the existing and the proposed system. Here we consider file size in MB and downloading time in milliseconds. File with small size consumes less time in both DDS and DDEAS.

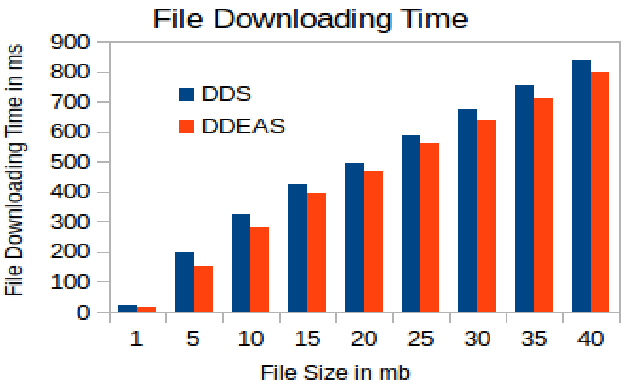

Fig. 3. Comparison of file downloading time of DDS and DDEAS

In DDS system data file recover algorithm is executed in user system to download the file, which consumes more time. As the recovery algorithm is executed in the metadata server, the time taken by the DDEAS is less as compared to the DDS and is shown in Fig. 3.

VII. CONCLUSIONS

Cloud data deduplication is important and essential for big data storage management in the cloud. In this paper, we implemented a server-side data recover algorithm in the distributed cloud data storage called DDEAS. In DDEAS, block-level and file-level deduplication techniques are implemented to upload the data in S-CSP. The DDEAS approaches a new method by executing the recover algorithm in the metadata server for downloading a complete file using a single user request. The DDEAS approach helps to reduce file downloading time, network bandwidth utilization and file accessing overhead. With DDEAS user can access a file from the S-CSP easily, which improves the cloud data storage access efficiency and make user to access file with less resources utilization.

REFERENCES

1. Q. Duan, “Cloud Service Performance Evaluation: Status, Challenges, and Opportunities – “A survey from the system modeling perspective.” Digital Communication Network., Available online 23 December 2016, ISSN 2352-8648, http://dx.doi.org/10.1016/j.dcan.2016.12.002.
2. Q. Liu, C. Tan, J. Wu, and G. J. Wang, "Towards Differential Query Services in Cost-Efficient Clouds." IEEE Transaction on Parallel Distributed System, vol. 25, no. 6, pp. 1648-1658, 2014.
3. Li, Jin, Xiaofeng Chen, Xinyi Huang, Shaohua Tang, Yang Xiang, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. "Secure distributed deduplication systems with improved reliability." IEEE Transactions on Computers, vol. 64, no.12, p. 3569-3579, 2015.
4. Dimakis, A. Godfrey, P.B., Wu, Y., Wainwright, M.J. and Ramchandran, K., “Network Coding for Distributed Storage Systems.” IEEE transactions on information theory, vol.56, no.9, pp.4539-4551, 2010.
5. Zhang, Wei, Hong Tang, Hao Jiang, Tao Yang, Xiao gang Li, and YueZeng. "Multi-Level Selective Deduplication for VM Snapshots in Cloud Storage." IEEE Fifth International Conference on Cloud Computing, pp. 550-557, 2012.
6. Chang, R.S., Liao, C.S., Fan, K.Z. and Wu, C.M., “Dynamic Deduplication Decision in a Hadoop Distributed File System. “ International Journal of Distributed Sensor Networks, vol. 10, no.4, p.630380, 2014.
7. K.Varalaxmi and D.Venkateshwarulu, “Secure Data Deduplication With Effecent Key Managament In Cloud Databases.” International Journal of Engineering & Science Research, Vol.5, Issue.7, pp.683-689, July 2015.
8. Stanek, J., Sorniotti, A., Androulaki, E. and Kencl. “A Secure Data Deduplication Scheme For Cloud Storage.” In International conference on financial cryptography and data security - Springer, Berlin, Heidelberg. pp. 99-118, March 2014.
9. Yunnan Wu and AlexandrosG .Dimakis. "Reducing Repair Traffic For Erasure Coding-Based Storage Via Interference Alignment." IEEE International Symposium on Information Theory, pp. 2276-2280, 2009.
10. Zhe SUN, Jun SHEN , and Jianming YONG. “ A Novel Approach To Data Deduplication Over The Engineering-Oriented Cloud Systems.” Integrated Computer-Aided Engineering, vol. 20, no.1, pp.45-57, 2013.
11. Danny Harnik, BennyPinkas, and Alexandra Shulman-Peleg. “Side Channels In Cloud Services: Deduplication In Cloud Storage.” IEEE Security & Privacy, vol.8, no.6, pp.40-47, 2010.
12. JianshengWei , Hong Jiang , Ke Zhou, and Dan Feng. "MAD2: A Scalable High-Throughput Exact Deduplication Approach For Network Backup Services." IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-14, 2010.
13. WarapornLeesakul, Paul Townend, Peter Garraghan, and JieXu. "Fault-Tolerant Dynamic Deduplication For Utility Computing." IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, pp. 397-404., 2014.
14. Qinlu He, Zhanhuai Li, and Xiao Zhang. "Data Deduplication Techniques." IEEE International Conference on Future Information Technology and Management Engineering, vol. 1, pp. 430-433. 2010.
15. Pasquale Puzio, RefikMolva, MelekÖnen, and Sergio Loureiro. "Cloudedup: Secure Deduplication With Encrypted Data For Cloud Storage." IEEE 5th International Conference on Cloud Computing Technology and Science, vol. 1, pp. 363-370. 2013.
16. Jin Li, Xiaofeng Chen, Ming qiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou. "Secure Deduplication With Efficient And Reliable Convergent Key Management." IEEE transactions on parallel and distributed systems, vol. 25, no. 6, pp.1615-1625, 2013.
17. Waraporn Leesakul, Paul Townend, and JieXu. "Dynamic Data Deduplication In Cloud Storage." IEEE 8th International Symposium on Service Oriented System Engineering, pp. 320-325, 2014.
18. Eric W. D. Rozier and William H. Sanders, Pin Zhou, and Nagapramod Mandagere. "Modeling The Fault Tolerance Consequences Of Deduplication." IEEE 30th International Symposium on Reliable Distributed Systems, pp. 75-84, 2011.
19. Bhavya M, Thriveni J, and Venugopal K R, “Improving Efficiency of Cloud Data Storage System: A Comprehensive Survey”, International Journal of Recent Trends in Engineering and Research, vol. 4, pp. 380-396, 2018.
20. Plank, James S. &quot, "Erasure Codes For Storage Systems: A Brief Primer". The Usenix Magazine, vol. 38, no. 44-50, pp. 44-50, 2013.
21. Yaksic, Vladimir Omar Calderón. "A Study On Hash Functions For Cryptography". Global Information Assurance Certification Paper, SANS Institute, 2003.

AUTHORS PROFILE

Mrs. Bhavya M received the Bachelor of Engineering degree in Computer Science and Engineering from The University Visvesvaraya College of Engineering Bangalore, in 2010, and the Master of Technology in Computer Science and Engineering from Cambridge Institute of Technology Bangalore, in 2014, Visvesvaraya Technological University, Belgaum, India. She is currently working toward the PhD degree from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore, Bangalore University, India. Her research interests include cloud computing, Cloud Data storage management and data security. She is a member of IEEE since 2015. She has published research papers in reputed international journals and conference. She has one years of teaching experience, One year of Industrial experience and 4 years of Research Experience.

Dr.Thriveni J has completed Bachelor of Engineering, Masters of Engineering and Doctoral Degree in Computer Science and Engineering. She has 4 years of industrial experience and 23 years of teaching experience. Currently she is Professor in the Dept. of CSE, University Visvesvaraya College of Engineering, Bangalore. She has over 90 research papers to her credit. She has produced four doctorate students and guiding 07 PhD Students. Her research interests include Networks, Data Mining and Biometrics.

Dr.K. R. Venugopal is currently the Vice Chancellor Bangalore University, Bengaluru. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bengaluru. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 64 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Micro-processor Programming, Mastering C?? and Digital Circuits and Systems etc., He has filed 101 patents. During his three decades of service at UVCE he has over 640 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining. He is a Fellow of IEEE, ACM and ISTE.