Journal of Scientific & Industrial Research
Vol. 80, March 2021, pp. 260-269

Weight Based Deduplication for Minimizing Data Replication in Public Cloud Storage

E Pugazhendi, M R Sumalatha* and Lakshmi Harika P*

Department of Information Technology, Anna University, MIT Campus, Chennai, 600 044, India

Received 04 October 2020; revised 18 December 2020; accepted 05 January 2021

The approach to optimize the data replication in public cloud storage when targeting the multiple instances is one of the challenging issues to process the text data. The amount of digital data has been increasing exponentially. There is a need to reduce the amount of storage space by storing the data efficiently. In cloud storage environment, the data replication provides high availability with fault tolerance system. An effective approach of deduplication system using weight based method is proposed at the target level in order to reduce the unwanted storage spaces in cloud. Storage space can be efficiently utilized by removing the unpopular files from the secondary servers. Target level consumes less processing power than source level deduplication. Multiple input text documents are stored into dropbox cloud. The top text features are detected using the Term Frequency (TF) and Named Entity Recognition (NER) and they are stored in text database. After storing the top features in database, fresh text documents are collected to find the popular and unpopular files in order to optimize the existing text corpus of cloud storage. Top Text features of the freshly collected text documents are detected using TF and NER and these unique features after the removing the duplicate features cleaning are compared with the existing features stored in the database. On the comparison, relevant text documents are listed. After listing the text documents, document frequency, document weight and threshold factor are detected. Depending on average threshold value, the popular and unpopular files are detected. The popular files are retained in all the storage nodes to achieve the full availability of data and unpopular files are removed from all the secondary servers except primary server. Before deduplication, the storage space occupied in the dropbox cloud is 8.09 MB. After deduplication, the unpopular files are removed from secondary storage nodes and the storage space in the dropbox cloud is optimized to 4.82MB. Finally, data replications are minimized and 45.60% of the cloud storage space is efficiently saved by applying the weight based deduplication system.

Keywords: Cloud Computing, Document Frequency, Document Retrieval, Document Weight, Dropbox Cloud

Introduction

Cloud service providers provide services to both individuals and company. Many organizations approach cloud service providers for using cloud as a platform and also for storage. Cloud service providers offer three types of services to end users. The services include Infrastructure as a service, Platform as a service and Software as a service. Infrastructure as a service enables user to save hardware memory by working in a virtual server. Platform as a service provides the cloud platform to accomplish a task. Software as a service delivers software applications over the internet.[1] Cloud services are deployed in four ways. They are private cloud, public cloud, community cloud and hybrid cloud. Public cloud will be owned and maintained by the service provider. In public cloud, the hardware and software infrastructures are wholly managed by the service providers. Private cloud will be provided to an individual or organization by the service providers on pay per usage basis. The individual or organization will not have the supervision of the service providers. Hybrid cloud is a combination of both public and private cloud. Community cloud allows accessing systems and services by group of organizations. It is a tedious task for the service providers to manage and provide services to millions of users. In order to provide efficient service, the service providers must ensure faster access, security and reliability. Data replication provides full availability by storing duplicate files to all the servers of cloud. Replication can happen in the host, hypervisor, storage array or network. Host based replication uses server to copy data from one site to another. Hypervisor based replication replicates the entire virtual machine from one server to another. Array based replication allows

*Author for Correspondence:
E-mail: sumalatha.ramachandran@gmail, lakshmiharika@mitindia.edu

copying data between arrays. But it is possible to use array based replication only in homogeneous storage environments.[2] Network based replication could be used in heterogeneous storage environment. It works in all arrays and supports any host platform. Data replication can be synchronous or asynchronous depending on when it takes place. Synchronous data replication takes place at real time. It involves high end transactional process as the data once lost could not be recovered. It also involves latency and is a costly process. It is supported by array based and network based replication. Asynchronous data replication is designed to work over distances and requires less bandwidth. Data could be recovered if lost. As there is a delay in copying data from one end to another, the copies on the two sides may not be identical. It is supported by array based, network based and host based replication. The availability of text documents are shown in Fig. 1.

Data replication is implemented in cloud environment to duplicate the files that are accessed often to increase availability. Deduplication is a process of removing duplicate or repeated files. Deduplication can be deployed either at client side or server side. In client side deduplication, each file is checked for duplication before adding the file to storage. In server side deduplication, all the files are initially loaded to the storage. Then the duplicate files are removed and the storage will contain only the unique files. Server side deduplication is advantageous over client side deduplication because it involves minimum overhead than client side deduplication.[3] Deduplication is implemented at block level and file level. In block level deduplication, each file is split into multiple chunks and then processed. Each chunk of the file occupies a separate storage area. In file level deduplication, each file is processed as a single entity. Each file occupies only a single block of memory. Thus file level deduplication efficiently manages storage space than block level deduplication. Though compression also helps in efficiently managing memory, data deduplication and data compression differ significantly. Data deduplication removes the repeated data whereas data compression reduces the size of the data.[4,5] Data deduplication is very much used by cloud service providers. They could efficiently manage memory and also increase the number of users due to enormous storage. The drawbacks of the existing system are removed using server side and file level deduplication. This research is to efficiently manage the data storage in public cloud (Dropbox) for faster access and also to increase the number of users. This can be achieved by categorizing the files into two divisions as popular and unpopular files. A file will be considered popular if it is requested by users very often. Unpopular files are those that are requested very rarely. The service will be efficient if the requests of the end users are quenched quickly. For faster service, popular files are stored in multiple servers to increase the availability. The popular files are made to be available in multiple locations because of its high frequency of request. If the unpopular files are also made to be available in multiple servers, this would unnecessarily block the storage space because they are very rarely requested by the users. Hence the unpopular files have to be removed from all the secondary servers. They would be made available only in the primary server. This can be achieved using data replication and deduplication.

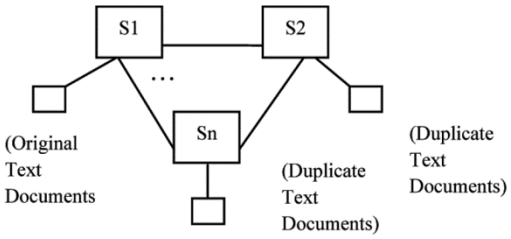

Fig. 1 — Data Replication

Materials and Methods

Client Side Deduplication

Deduplication is a process of removing repeated or duplicate files. The popular and the unpopular files were segregated using a threshold data deduplication technique. Source level deduplication was implemented in this technique in which each file was checked for duplication before adding it to storage. It increases the overhead at client side. Besides, Jan Stanek, *et al.* also focused on security because it was performed only on a single node.[6] Threshold convergent cryptosystem algorithm was used to segregate the popular and unpopular files. It ensured security by allowing users to encrypt messages. Further Identity Provider was used to check authorization and Index Repository Service provides indexation for unpopular files. Besides, some user invoked algorithms were also implemented in this paper. As a result, it not only enhances the security of popular files but also ensures security for unpopular files. Hence it operated on a single system, the whole

system collapses upon the failure of a single node. Thus the system failed to achieve fault tolerance. Zheng Yan, *et al*. proposed a methodology to manage storage efficiently and also provide security in a heterogeneous environment.[7] The storage or memory area will be provided by multiple cloud service providers. The cloud environment is maintained by the data owner or trusted third party. Client side deduplication was used for efficient storage management. In client side deduplication, each file was checked for duplication before adding the file to storage. Security was achieved by using Attribute Based Encryption (ABE) algorithm. Attribute Based Encryption was implemented in two stages. They were key policy and cipher text policy. In key policy, the key required for encryption and decryption were generated. In cipher text policy, the original data was encrypted using the key value generated and stored in cloud. As the data were encrypted, the access control was limited to authorized user who would be provided with the key for decryption. This protected the data from unauthorized user. Besides, storage space was also managed efficiently using deduplication technique. As the deduplication was implemented in client side, every document has to be checked for duplication before adding it to the storage. This increases the overhead involved in the client side.

Replication Management

Data Replication is a process of duplicating files to increase availability. It is widely used by cloud service providers. They provide faster access to cloud users by replicating the files in multiple locations. Boru, *et al.* presented models for energy consumption and bandwidth demand of database access in cloud computing datacenter.[8] Energy consumption depends upon the capacity of the datacenter and the frequency of access by the cloud users. The bandwidth depends upon the communication channel between the service provider and the user and the delays associated with the communication channel. Data replication technique could optimize the bandwidth and the energy consumption between the datacenters and also within the datacenter. Two methods used for achieving the above objectives include Dynamic Power Management (DPM) and Dynamic Voltage and Frequency Scaling (DVFS). DPM method puts the idle components into sleep mode. DVFS used the formula P=V2f where P referred to the power consumption, V referred to the voltage supplied and f referred to the operating frequency. Voltage and frequency are directly proportional to power consumption. Voltage and frequency influences the power consumption to a greater level. DVFS was limited because it reduced the power consumption only to CPU whereas the other components keep consuming the power at their peak rates. The models used for achieving energy consumption and bandwidth include Energy Consumption of Computing Servers, Energy Consumption of Network Switches, Bandwidth Model and Database Access and Energy Consumption. Green Cloud Simulator was used for verifying the results obtained from implementing the above models. The explosive growth of data increases the burden of storage systems. At the same time, the redundant data result in a waste of storage space and an increase in cost. Data deduplication is urgently required to alleviate these problems. Similarity detection was very critical to the performance of data deduplication. This technique reduced memory consumption and improved the data deduplication ratio by sampling. Qi *et al*. presented the Enhanced Position-Aware Sampling algorithm (EPAS), proposing minimum value sampling algorithm based on Content-Defined Chunking (CDC), which introduced the sampling algorithm into the data chunking process, continuously calculating the hash value of the sliding window and then selecting the minimum value as a sample.[9] This experiment showed that data deduplication ratio of the proposed algorithm is higher than EPAS algorithm at the same sampling ratio.

Chunk Level Deduplication

Bo Mao, *et al*. described the capacity oriented deduplication method. This deduplication technique reduced the I/O performance.[10] In this paper, Performance Oriented Deduplication (POD) was used to improve the performance of primary storage system in the cloud and to minimize the performance overhead of deduplication. The techniques used to achieve the above objectives include Select Dedup and Icache. The access monitor module under Icache monitors the arriving requests and dynamically resizes blocks in storage space. The Request Deduplicator module used under Select Dedup splits the write requests into chunks and identifies the duplicates in order to decide whether deduplication should be implemented or not. If any duplicate was encountered, its count would be incremented. Else, it would be added as a new entry in hash table. Thus the

removal of duplicates helps to manage the storage efficiently and also minimizes the overhead to a greater extent. But computing hash value for each file increases computational and storage space overhead. To overcome this, an efficient hash function was required which would be costlier. As POD works only for small requests, using a costly hash function will not be worthy.

Compression and Weight Estimation

Compression is a technique used to reduce the size of the data without making any modifications to the original data. As the world is growing more computerized, the data are mostly stored in digital form. Ensuring security to the sensitive data during online transformation is important. In this paper, Novel Data Storage Solution for Cloud by Bhagya *et al*. a compression technique has been devised to manage storage efficiently and securely.[11] The model used in this paper consists of three entities namely user, processing unit and storage system. The user is the client who gets access to upload or download a file from the cloud. Processing unit takes care of all system activities like file optimization, dictionary matching, data encryption or decryption, de-optimization and de-tagging. Storage system is the cloud that provides services to the user. An inbuilt dictionary was used which consists of a set of words. The performance of the system is directly proportional to the strength of the dictionary. Initially, the user stored the file in the storage system through intermediate process. The steps involved in intermediate process include file optimization, dictionary matching and encryption. In file optimization the special characters were removed and the file was split into chunks. In dictionary matching, each chunk will be compared with the index in dictionary. If the chunk was not present in the dictionary, then it would be added as a new entry in the dictionary. Encryption was achieved using convergent key. The user would have the access to download the file from the storage system. The encrypted file could be decrypted using AES algorithm. Hence the system reduces the size of the file and also ensures security. Kurniawati *et al*. presented a methodology to rank documents based on user query. This paper used TF-IDF-ICF algorithm to perform this task.[12] Initially TF-IDF algorithm was used for assigning weight to the term based on its frequency. Term frequency counts the occurrence of the term depending on its appearance in a specific document. Inverse document frequency weighs the term depending on its appearance in multiple documents. The combination of TF-IDF helps in assigning weights to terms. The rarely occurring terms would be assigned with less weight and the terms that occur often would be assigned with more weight. However, TF-IDF algorithm does not calculate the weight of the terms in a particular class. Inverse Class Frequency (ICF) was used for this purpose. Inverse Class Frequency gives significance only to existing terms rather than the number of words in a document. Thus it produces less accuracy when documents were ranked using TF-IDF-ICF algorithm. To overcome this, ICSdF was used. Inverse Class Density Frequency Space algorithm gives significance to every term in a document which was a member of the class. Thus the ranking of documents using TF-IDF-ICSdF algorithm produces 93 percent accuracy in retrieving documents according to user's query.

Secure Deduplication

The single level security based convergent encryption technique has been addressed to encrypt the data before outsourcing using deduplication technique. However, the issue of keyword search over encrypted data in deduplication storage system has to be addressed for efficient data utilization. The paper fully focused security aspects of deduplication but not the consider the issues of data replication in multiple storage instances in cloud.[13]A scheme based on attribute-based encryption (ABE) was proposed to deduplicate encrypted data stored in the cloud while also supporting secure data access control which is limited to single instance storage of cloud. However, the paper achieved basic security but the minimizing data replications were not addressed when targeting multiple instances in cloud.[14]A novel selective encryption and component-oriented deduplication (SEACOD) application was developed which achieves both fast and effective data encryption and reduction for mobile cloud computing services. This system deduplicated redundant objects in files, emails, as well as images exploiting object-level components based on their structures. However, this methodology is based on single instance storage which fails to consider the problems of data replication when targeting multiple storage instances in cloud.[15] A combined solution for keyword search, phrase search and auditing for encrypted cloud data storage has been provided. Secure based deduplication system using encryption scheme was

addressed over the data which stored on remote cloud servers. However, optimizing the cloud data storage using deduplication system was not considered when targeting the multiple instances of cloud.[16] The problem of integrity auditing and secure deduplication on cloud data was studied. Also two security systems are addressed such as SecCloud and SecCloud+. SecCloud introduces an auditing entity with a maintenance of a Map Reduce cloud, which helps clients generate data tags before uploading as well as audit the integrity of data having been stored in cloud. SecCloud + is designed motivated by the fact that customers always want to encrypt their data before uploading, and enables integrity auditing and secure deduplication on encrypted data. However, the single level security with encryption was achieved but deduplication has not implemented to optimize the unwanted data which are consumed by the data replication when targeting multiple instances in cloud.[17] Different types of file level deduplication schemes for storage devices were designed and implemented. This file level deduplication system is based on the different duplication checking rules such as file name, file size and full / partial content hash value. But the main issues of data replication such as unwanted storage spaces and cost, when targeting multiple instances of cloud are not discussed.[18] The data storage management is the main challenging issues in the cloud storage environment when handing duplicate copies in multiple instances. To reduce the storing space, cost and save the bandwidth, share the data with others in cloud storage using public-key cryptosystems which produce constant-size cipher texts such that the active delegation of decoding rights for any set of cipher texts but it leads to many privacy problems. To avoid this problem, hybrid cloud was considered which is the combination of the private and public cloud. Here the confidential data are stored on private cloud and other data are stored in the public cloud. The convergent encryption technique has been addressed to protect the confidentiality of sensitive data before outsourcing. Additionally, in order to provide high security, the authorized data deduplication was addressed in this hybrid cloud. However, this method focused the security aspects of deduplication but data optimization of cloud storage using deduplication techniques are not addressed.[19]

Weight Based Deduplication

This system aims at developing a cloud storage system in order to make use of the available cloud storage efficiently by minimizing the data replication using weight based deduplication technique. In the proposed work, deduplication is deployed at target level as the source level deduplication increases the system overhead and takes longer processing time. The detailed architecture of the proposed system is shown in Fig. 2.

Initially, multiple input text documents are collected and stored to the dropbox storage nodes. The text features of the input text documents are collected using term frequency and Named Entity Recognition. Top text features detected from Term Frequency (TF) and Named Entity Recognition (NER) is stored in the text database for future retrieval. The same procedure is repeated for fresh text files. The unique features of fresh text files are collected and compared with text features of database and the corresponding text documents are listed. The average weight of all the listed documents provides the threshold factor which could be used to segregate the popular and unpopular files. The text documents with weight greater than threshold value are considered as popular files and only those text documents are stored in all the storage nodes. The rest of the text documents are considered as the unpopular files and stored only in the primary storage node. By removing the unpopular files from secondary storage nodes, storage space could be utilized efficiently.

Corpus Storage

In this system, multiple text documents are taken as input and stored to the cloud server (Dropbox) for efficient storage and retrieval. API v2 was used for accessing Dropbox account and storing text corpus. Here, politics text documents are used. Initially, four storage nodes such as S_1, S_2, and $S_3$are created in dropbox cloud. S_1 is the primary storage node and the remaining nodes are considered as secondary storage nodes.

Text Features Collection

Text features are detected using Term Frequency (TF) and Named Entity Recognition (NER). Term Frequency is to provide the frequency of each word in a document. Term frequency is found for all the terms excluding stop words. The estimated TF is stored in a database file for future reference. Named Entity Recognition extracts features like person name, location name, organization name and date in a text document. Open NLP tool is used to identify text features in the collected text documents. This tool also provides various classes and methods for generating a

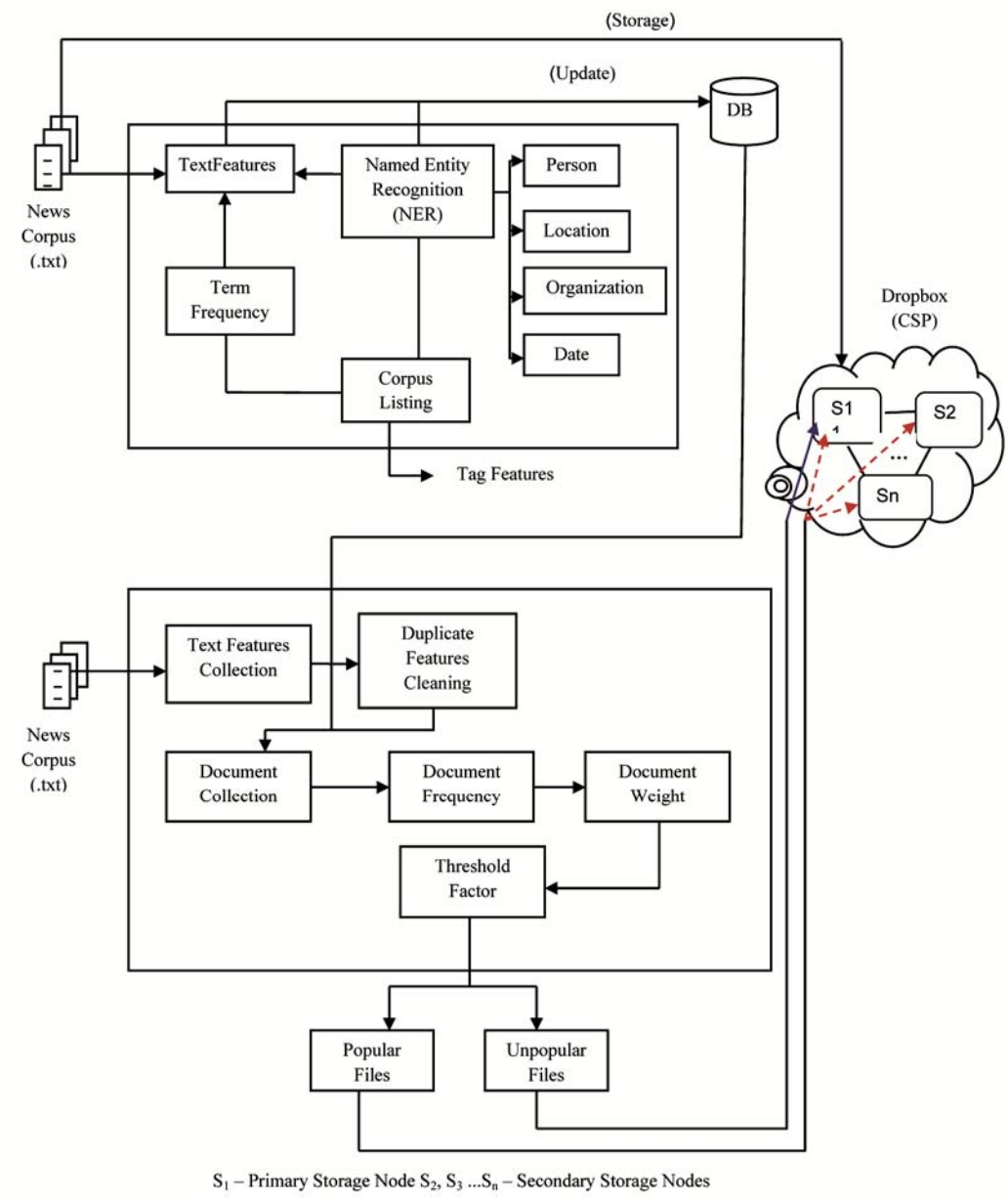

Fig. 2 — System Architecture for Weight Based Deduplication

custom binary model. The detected text features are stored in a database for future retrieval of text documents. The output of the Term Frequency and Named Entity Recognition are written to a file and the contents of the file are copied to the database. Database is created using php MyAdmin and SQL queries are used to transform the contents to the database.

Document Collection

Document collection is the first level retrieval task in weight based deduplication. Text features are detected again for a fresh corpus using TF and NER. Duplicate text features are removed using text feature cleaning process. Finally, relevant text documents are collected by comparing the text features stored in database with the unique features obtained from the fresh corpus.

Weight Based Deduplication

Weight based deduplication is mainly used to minimize data replications in public cloud storage by the detections of popular and unpopular files. The popular and unpopular files are the resultant

document collection of threshold based document weight. The techniques of this deduplication include document weight calculation, threshold factor calculation and detection of popular and unpopular files. Document weight is mainly used to segregate files as popular and unpopular. Document weight is estimated from document frequency. Document frequency is the count of each document in the retrieved results of document collection. The average weight of the text documents those are listed in the document collection are considered as threshold factor. The text document with weight is greater than the threshold value will be considered as the popular file and stored in all storage nodes S_1, S_2 …S_n in order to provide full availability of data. Other files are unpopular files which are stored only in primary storage node S_1 and deleted from all other nodes. Hence unwanted storage space of cloud which are consumed by data replication is reduced. The complete algorithm of weight based deduplication is given below.

Algorithm: Weight based deduplication
Input: Set of input files (.txt format)
Output: (i) Detection of popular and unpopular files
(ii)Minimize data replications in the cloud
Parameters:
Let S←S_1,S_2,…S_n be the cloud servers with S_1 as primary.
Let X is the number of input text files in corpus.
Let Df be the number of fresh input files for testing
Let P be the list of popular files.
Let features be a list of features containing TF and NER results.
Let updated features be a list of features containing TF and NER results of fresh corpus.
Consider the size of input corpus is 1000 files.
Let person, location, organization and date be the set of person names, location names, organization names and date used in the file X_i.
1. for each i←1 to X
Find term frequency features and Extract NER features like person, location, organization, date from X_i
Store top features in text DB.
end for
2. for each i←1 to Df
find top features for Df_i using TF and NER
end for
3. Remove the duplicate features from obtained results on step 2.
4. Compare the updated features on step 4 with the features stored in Database.
5. Using queries, text documents are listed based on step 4.
Let document weight (W) for all files be initialized to 1.
Let D_{cc} be the count of retrieved documents from step 5.
6. for each file accessed
Update DF_i count based on documents listed.
end for
7. Change the order of files based on the document frequency count of files.
8. for i←1 to D_{cc}
Find Document Weight W using the formula
$W= DF_i / X$
end for
9. Find threshold value t using the formula
$t= \sum W / X$
10. for each i←1 to X
if $W_i > t$, then X_i is the popular file.
Unpopular files are X-P
end for
Files in unpopular are removed from secondary servers.

Performance Evaluation

Execution Levels

The weight based deduplication technique is implemented using dropbox cloud service provider. S_1, S_2,…S_n are the storage nodes used where S_1 is the primary node and the remaining are secondary nodes. The corpus contains a total of 1000 input text files. All the three nodes contain these 1000 files in order to achieve full data availability. So there are 3 copies of each file which increases the storage space. The main aim is to optimize the cloud data storage of existing corpus by detecting popular and unpopular files using document weight deduplication by testing the fresh corpus. At the end, cloud data storage is optimized by removing unpopular files which are tested by four run levels. The system with windows 10 operating system is used for implementation. Code is executed in Eclipse platform using java programming language. Microsoft Visual Studio 2010 is used for user interface design. The performance of the system is tested at four run levels. Initially, all the input text files are stored to the dropbox cloud in all the four run levels. In run level R1, the number of tested files is 25. After applying deduplication and estimating the threshold factor, the unpopular files are removed from all the secondary storage nodes. The storage space saved in the first run level is 58%. Similarly, the

system is tested with 50, 75 and 100 files. The storage space saved in the run level R2, R3 and R4 are 43.5%, 40% and 41% respectively. The Table 1 shows the execution at various run levels.

Estimation of Threshold Factor

Using weight based deduplication technique, the weight of all text documents which are stored to the dropbox cloud are calculated. The average weights of all the documents yield the threshold factor. As the number of fresh files used increases, the threshold factor also increases. They are directly proportional. The Fig. 3 shows the gradual increase in threshold factor as the number of fresh files used increases.

Detection of Popular and Unpopular Files

Threshold factor helps to segregate the popular and unpopular files. The files with weight greater than threshold factor will be segregated as popular files and those with weight lesser than or equal to threshold factor will be segregated as unpopular files. The fig. 4 shows the number of popular and unpopular files at each run levels.

Results of Weight based Deduplication

Initially, when the multiple input text files stored in all the storage nodes, the storage space occupied was 8.09 MB. The Fig. 5 shows the storage space occupied at various run levels after removing the unpopular files from the secondary storage nodes.

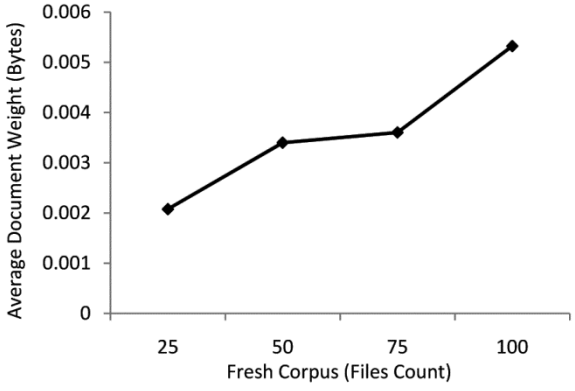

Fig. 3 — Threshold Factor

Performance Comparison

The Fig. 6 shows the comparison of deduplication results of three other existing models with proposed weight based deduplication. The File Level

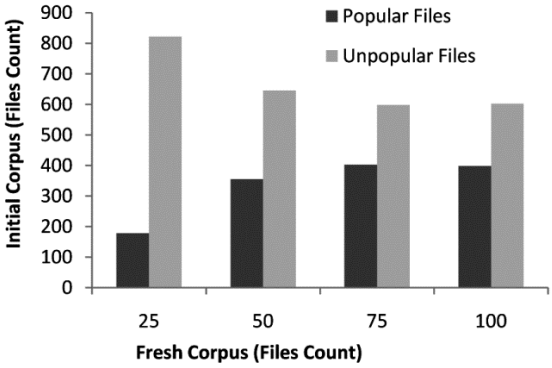

Fig. 4 — Popular and Unpopular Files Detection

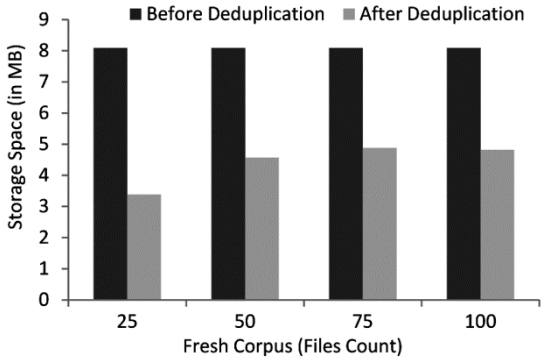

Fig. 5 — Weight Based Deduplication Results

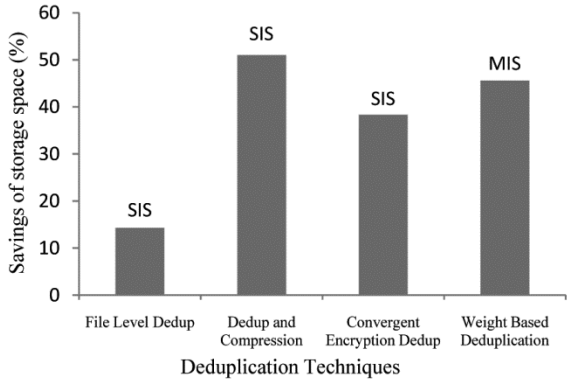

Fig. 6 — Storage Savings Comparison

Table 1 — Execution Levels

Run Levels	Number of fresh files(.txt)	Threshold Factor	Number of popular files	Number of unpopular files	Before deduplication (Size in MB)	After deduplication (Size in MB)	Savings (Size in MB)
R1	25	0.002078	178	822	8.09	3.38	4.71(58%)
R2	50	0.003400	355	645	8.09	4.57	3.52 (43.5%)
R3	75	0.003603	402	598	8.09	4.88	3.21 (40%)
R4	100	0.005323	398	602	8.09	4.82	3.27(41%)

Table 2 — Storage Utilization

Run Levels	No. of input files	No. of tested files	Before Deduplication (Storage Space in MB)	After Deduplication (Storage Space in MB)
R1	1000	25	8.09	3.38
R2	1000	50	8.09	4.57
R3	1000	75	8.09	4.88
R4	1000	100	8.09	4.82

Dedupsystem saves only 14.3% which is much lower than that of Weight based deduplication which saves 45.6% of storage space, whereas Dedup and Compression saves 51%, which is slightly higher than that of Weight based deduplication. Convergent Encryption Dedup saves 38.3% which is nearer to Weight based deduplication. But, the results of other three systems are based only on Single Instance Storage which is not in the case of Weight based deduplication which uses multiple storage nodes with S_1 as primary node and $S_2,S_3,\ldots S_n$ as secondary nodes. Due to this reason, Weight based deduplication provides better data optimization performance and achieves data availability but the other existing techniques does not achieve the data availability.

Weight Based Deduplication Efficiency

The table 2 shows the storage space utilization in cloud before and after deduplication at various run levels.

Conclusions

The proposed research work weight based deduplication has implemented and optimized the cloud data storage of existing text corpus with multiple instances storage. This technique is mainly used for detecting the popular and unpopular files from existing text corpus by running the fresh testing files. Initially multiple text corpus is considered and given to the system and stored to the cloud using the cloud service provider dropbox. In the input corpus, the top text features of input corpus are detected using the NLP techniques such as TF and NER and then these features are stored to the text database for future retrieval. The performance of the weight based deduplication is tested with four run levels. In the testing corpus, again top text features of fresh text corpus are detected using TF and NER. Top features are collected after the duplicate features cleaning. Document collection is retrieved which are the result files of each top text feature which is identified from the database. Document frequency is calculated from these retrieved documents of document collection. Document weight has been assigned and calculated for all the files based on the document frequency counts. Finally, popular and unpopular files are detected after calculating the average threshold value from the document weight. Popular files are most frequently accessed files which are retained in all the servers in order to maintain the full availability of data and unpopular files are unwanted files which are removed from all the servers except the primary server. The corpus size for multiple instances consumed 8.09 MB in the dropbox cloud before the deduplication. After running four run levels using this weight based deduplication technique, the existing text corpus is reduced to 4.82 MB and 45.60% storage spaces are effectively saved by minimizing the data replications.

In future, this work may be extended to consider the other types of deduplication such as image deduplication, video deduplication, etc. Also multilevel data security enhancements in the cloud may be extended.

Acknowledgments

Thanks much to the Department of Information Technology, MIT Campus, Anna University Chennai for giving the technical support.

References

1 Goncalves da Silva G H, Holanda M & Araujo A, Data Replication Policy in Cloud Computing Environment, *IEEE 11th Iberian Conference on Information Systems and Technologies* (2016) 01-06.

2 Alghamdi M, Tang B & Chen Y, Profit based File Replication in Data Intensive Cloud Datacenters, *IEEE International Conference on Communications ICC)* (2017) 01-07.

3 Xu X &Tu Q, Data deduplication mechanism for cloud storage Systems, *IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (2015) 286-294.

4 Upadhyay A,Balihalli P R, Ivaturi S & Rao S, Deduplication and Compression Techniques in Cloud Design, *IEEE International Systems Conference SysCon*(2012)01-06.

5 Deepu S R, Bhaskar R &Shylaja B S, Performance Comparison of Deduplication Techniques for Storage in Cloud Computing Environment, Asian Journal of Computer Science and Information Technology (AJCSIT)**05** (2014) 42-46.

6 Stanek J &Kencl L, Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage, *IEEE Transactions on Dependable and Secure Computing*, **04** (2017) 694–707.

7 Yan Z, Zhang L, Ding W & Zheng Q, Heterogeneous Data Storage Management with Deduplication in Computing, *IEEE Transactions on Big Data***03** (2017) 393–407.

8 Boru D, Kliazovich D, Granelli F, Bouvery P & Zomaya A Y, Models for Efficient Data Replication in Cloud Computing Datacenters, *IEEE International Conference on Communications ICC* (2015) 6056-6061.

9 Qi H, Han Y, Di X &Sun F, Minimum value sampling algorithm based on CDC, *IEEE International Conference on Computer Science and Network Technology (ICCSNT)* (2016) 350-354.

10 Mao B, Jiang H, Wu S & Tian L, Leveraging Data Deduplication to improve the Performance of Primary Storage Systems in the Cloud, *IEEE Transactions on Computers***06** (2016) 1775–1788.

11 Bhagya S.L & Gopal R K, A Novel Data Storage Solution for Cloud, *IEEE International Conference on Networks & Advances in Computational Technologies NetACT* (2017) 192-195.

12 Kurniawati&Syauqi A, Term Weighting Based Class Indexes Using Space Density for Al-Qur'an Relevant Meaning Ranking, *International Conference on Advanced Computer Science and Information Systems ICACSIS* (2016) 460-463.

13 Li J, Chen X, Xhafa F &Barolli L, Secure Deduplication Storage Systems with Keyword Search, *IEEE 28th International Conference on Advanced Information Networking and Applications* (2014) 971-977.

14 Yan Z, Wang M, Li Y &Vasilakos A V, Encrypted Data Management with Deduplication in Cloud Computing, *IEEE Cloud Computing***02**(2016) 28–35.

15 Song S, Choi B & Kim D, SEACOD-Selective Encryption and Component-Oriented Deduplication for Mobile Cloud Data Computing, *International Conference on Computing, Networking and Communications (ICNC)* (2016) 01-05.

16 Poon H T & Miri A, A Combined Solution for Conjunctive Keyword Search, Phrase Search and Auditing for Encrypted Cloud Storage, *Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart World Congress* (2016) 936-941.

17 Li J, Li J, Xie D &Cai Z, Secure Auditing and Deduplicating Data in Cloud, IEEE Transactions on Computers 08 (2016) 2386–2396.

18 Wu Y, Yu M, Leu J S, Lee E &Song T, Design and implementation of various file Deduplication schemes on storage devices, *11thIEEE International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE)* (2015) 80-84.

19 Jayapandian N Rahman A M J M Z &Nandhini I, A novel approach for handling sensitive data with Deduplication method in hybrid cloud, *IEEE Online International Conference on Green Engineering and Technologies (IC-GET)* (2015) 01-05.