Cloud Backup Services using GENEIC Algorithm

N. Gomathi[1] Meenalochini M.[2] K. Saranya[3]
[1,2,3]Assistant Professor
[1,2,3]Department of Computer Science & Engineering
[1,2,3]Jaishriram Engineering College, Tirupur, Tamil Nadu, India

Abstract— Cloud computing empowers individual users by providing storage space with better security and privacy, which in addition also provides anytime and anywhere access to data. Increase in the usage of the personal computing devices resulted in the dramatic increase in the data to be stored in the cloud backup services. Thus, these personal computing devices mainly rely on the cloud backup and restore services. The duplicates in the cloud environment introduce two major challenges of bandwidth and the storage space. Data deduplication is the ideal technique for reducing the bandwidth and storage space utilization. The proposed Genetic Algorithm based deduplication approach combines the application-based and file similarity function for supporting its functionality. The application-based scheme improves data deduplication efficiency by exploiting application awareness and reduces backup window.

Key words: Deduplication, Cloud Backup Service, Genetic Algorithm

I. INTRODUCTION

Cloud is a large pool of easily usable and accessible virtualized resources. These resources can be dynamically re-configured to adjust to a variable load, allowing also for optimum resource utilization. Cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to users over a network. Cloud storage is a massive and public accessible storage available for use in the internet. This is termed as Data storage as a Service (DaaS) with respect to services of cloud. Cloud backup service is the core technology of cloud storage. Cloud backup stores data located at the client side into the cloud storage service provider through network so as to recover data in time. Cloud backup service has become cost-effective solution that is adopted by many organizations as their alternate data protection strategy. The serious challenge reported by Clements et al (2009) is large backup window that represents the time spent on sending specific dataset to backup destination, due to the low network bandwidth between user and service provider constraining the data transmission. For example, it would take more than 14 days to backup 1TB data to Amazon S3 with the assumed network bandwidth of 800KB/s. Another challenge discussed by Yinjin Fu et al (2014) stems from the vast storage space and very high data management cost required for the rapidly increasing amount of backed-up data stored at service providers' site.

Deduplication is an effective technique to optimize the utilization of storage space. Data deduplication technology identifies duplicate data, eliminate redundancy and reduce the need to transfer or store the data in the overall capacity. Data deduplication can greatly reduce the amount of data, thereby reducing energy consumption and reduce network bandwidth in cloud data centres.In the deduplication process, duplicate data is determined and only one copy of the data is stored, along with references to the unique copy of data thus redundant data is removed. The most common deduplication technique partitions data into chunks of non-overlapping data blocks. It calculates a fingerprint for each chunk using a cryptographic hash function (e.g. SHA-1) and stores the fingerprint of each chunk in a hash table (chunk index).

Each chunk stored on the storage system has a unique fingerprint in the chunk index. To determine whether a chunk is already stored on the system or not, the fingerprint of the incoming data item is first looked up in the chunk index and if there is a match, the system only stores a reference to the existing data. Otherwise the incoming chunk is considered unique and is stored on the system and its fingerprint inserted in the chunk index.

Data deduplication strategies are basically classified into two types based on data units as File-level deduplication, Block-level deduplication. Each block may be of fixed-sized (static) or variable-sized chunks. Depending on the location where redundant data is eliminated, deduplication can be categorized into two basic approaches based on Harnik et al. (2009). In target-based approach, deduplication is performed in the destination storage system. The client is not having knowledge about the deduplication strategies. This method have the advantage of increasing storage utilization, but does not save bandwidth. In Source based deduplication, elimination of duplicate data is performed close to where data is created. The Source deduplication approach works on the client machine before it is transmitted specifically, the client software communicates with the backup server (by sending hash signatures) to check for the existence of files or blocks. Duplicates are replaced by pointers and the actual duplicate data is never sent over the network.

Increase in the usage of the personal computing devices resulted in the dramatic increase in the data to be stored in the cloud backup services. For dataset with logical and physical size, source deduplication can reduce the data transfer time to that of traditional cloud backup. However, data deduplication is a resource-intensive process, which entails the CPU-intensive hash calculations for chunking and fingerprinting and the I/O intensive operations for identifying and eliminating duplicate data. Unfortunately such resources are limited in a typical personal computing device.

The main objective of proposed system is to implement an advanced and novel method of deduplication in the personal computing devices using Genetic Algorithm. Source deduplication has to be implemented such as to increase efficiency of the deduplication process by checking the data at the source level, instead of checking the data at the target level. Thus, the data has to be checked for the duplicates at the source level such that the duplicates could be avoided at before uploading the data. And the use of the bandwidth could be easily optimized.The proposed system exploits file similarity information for efficiency checking

All rights reserved by *www.ijsrd.com*

duplicated files on the server. The implementation of the Genetic Algorithm increases the deduplication efficiency, and highly optimizes the deduplication process. The application based module and file similarity module reduces the complexity in the data chunks, thus the time required for the deduplication is highly reduced. This automatically reduces the backup window size and the bandwidth required for the backup.

II. RELATED WORK

This section is divided into two parts. The first part reviews work related to Deduplication based on Genetic Programming and second part reviews deduplication methods in cloud backup services. Moises et al(2012) proposed Active Learning Genetic Programming(AGP) which is a semi-supervised GP for the data deduplication problem. AGP uses an active learning approach in which a committee of multi-attribute functions votes for classifying record pairs as duplicates or not. When the committee majority voting is not enough to predict the class of the data pairs, a user is called to solve the conflict. The method was applied to three datasets and compared with supervised GP based deduplication method. Results show that AGP guarantees the quality of the deduplication while reducing the number of labeled examples needed. The other method [6] based on GP for the data deduplication task is used to find record-level similarity functions that combine single-attribute similarity functions, aiming to improve the identification of duplicate records and, at the same time, avoiding errors.

A Semantic-Aware Multi-tiered designed by Yujuan Tan et al,(2010) is a source de-duplication framework that first combines the global file-level de-duplication and local chunk-level deduplication. They also considered file level semantic attributes like file locality, file time stamps, file size and file type which are used to find redundant data The main drawback of this hybrid approach is more storage space at the client and restore performance. Yinjin Fu et al(2014) implemented Application Aware deduplication (AA-Dedupe) techniques which handles the computational overhead by implementing an intelligent data chunking scheme and the adaptive use of hash functions based on application awareness, and to alleviate the on-disk index lookup bottleneck by separating the entire index into small independent and application specific indices in an application-aware index structure. Backup window size of AA-Dedupe is reduded 10-32%.CAusality- Based deduplication reported by Yujuan, et al., (2011) captures the causal relationship among chronological versions of datasets to remove the unmodified data from transmission during not only backup operations but also restore operations. This system reduces both the backup time and restore time with a optimal reduction.

III. MATERIALS & METHODS

Architecture Overview: The backup stream of data is sent from the user/client system to the deduplication system, when the backup is generated by a user request. Then the backup data stream is sent to the application based module. The application based module chunks the data and maintains the log in the application based index. The data from the application-based chunk are stored in the local storage. Then the stored data is checked for the similarity, through similarity function. Then the data are guided to the cloud storage through the file agent. The file agent stores the unique data to the cloud storage, and the other data to the parallel container store. The parallel container store creates the population input with the available set of documents in the parallel container store. This population is later sent to the Genetic Algorithm module.

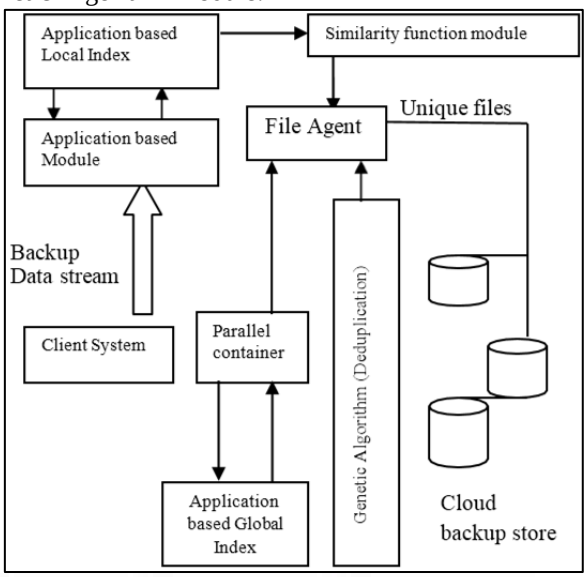

Fig. : Genetic based deduplication Architecture

The proposed system consists of following components and their functionalities are explained in the following section

1) Client system

The client system is the source of the backup data stream, where the data is backed up in a frequent time interval, or when the user is in need for necessary backup. When the backup is initiated from the client system, the stream of backup data is sent to the deduplication system.

2) Backup Data Stream

The backup data that is being generated from the client system is called as the cbackup data stream. This backup data stream is the input for the deduplication; this stream has to be further optimized in order to have and optimized cloud storage system. Application-Based Index: The application-based index is the place, where the details of the files or documents that are divided according to the applications such as text, pdf, audio, video. This maintains the log of the document, such that the details of the documents are stored and maintained.

3) Parallel Container Store

The parallel container store is the place which stores the documents and data that are chunked and stored into the cloud by the cloud agent module.

4) File Similarity Search (Filter)

File similarity search concept is widely used in data processing system area. The main idea is to extract hash keys from a file. Usually, hash function calculates hash key from a file and stores the hash key in a queue. By shifting one byte step by step, hash function repeatedly generates hash key and insert the hash key to the queue. The queue contains only several numbers of keys in ascending order or in descending

All rights reserved by *www.ijsrd.com*

order by configuration of the system. If hashing is finished, there remains several hash keys whose value is maximum or minimum. These key values are used for file similarity search. When A file and B file have duplicated hash keys, this means that the file have duplicated region of data. The Pseudo code for file similarity search is given as follows:

5) Pseudo code

```
Begin
isEqualChange←init
for i←0 to Array1.lenght.do
        isEqual←false
        for j←0 to Array2.Length do
        if Array1[i].hash = Array2[j].hash then
                isEqual← true
                Shift←Array1[i].offset-Array2[j].offset; break
If isEqual =true then
   If isEqualChange = true then
        Flip←true
   Else ifisEqualChange = false then
       flip←false;flipcnt++;
         if flipcnt==2 then cnt++;    flipcnt←0
  isEqualChange = true
  else    ifisEqualChange = true then
        flip←false;flipcnt++
          if flipcnt==2 then
           cnt++; flipcnt←0
         else if isEqualChange = false
                then
          flip←true;isEqualChange= false
          if Shift!=0 and cnt == 0 thenHeadSection()
          Else if cnt>0 then EndSection ();
                HeadSection()
          Else EndSection ()
End
```

6) File Agent

File Agent is a software program that provides a functional interface (file backup/restore) to users. It is responsible for gathering datasets and sending/restoring them to/from Storage Servers for backups/restores.

7) File Pattern Search

The main purpose of file pattern search tool is to predict the relationship between two files using file similarity information such 1) No duplicated region 2) Non-duplicated data is located in the front of a file 3)Non-duplicated data is located at the end of the file 4) Non-duplicated data is located in the middle section of a file or several section have non-duplicated data.

8) Genetic Algorithm

We present a genetic Algorithm (GA) approach to deduplication. Our approach combines several different pieces of evidence extracted from the data content to produce a deduplication function that is able to identify whether two or more entities in a repository are replicas or not. The classification algorithm with adjustable parameters W that identifies duplicate vector pairs from P.

9) Algorithm:

1) Initialize Duplicate Vector D =NULL
2) Set the parameters W of C1 according to N
3) Use Classification algorithm to get a set of duplicate vector pairs d1, f from P and N
4) P=P – d1;. While d1≠ NULL do N'=N-f; D=D+d1+f;
5) Train C2 using D and N'
6) Classify p using C2 and get a set of newly identified duplicate vector pairs d2
7) Set P=P - d2 and D=D+d2
8) 8 Adjust the parameters W of C1 according to N' and D
9) N=N'
10) Return D

First, each record's weight is set according to dissimilarity, among records. Then GA utilizes the weights set to match records from different data sources. Next, with the matched records being a positive set and the nonduplicate records in the negative set, then GA further identifies new duplicates. Finally, all the identified duplicates and nonduplicates are used to adjust the field weights set in the first step and a new iteration begins. The iteration stops when no new duplicates can be identified.

IV. Results

Our experiments were performed on a Private cloud which constructed using EUCALYPTUS. The Cloud Client with intel core 1.5Ghz processor, 8 GB RAM, and one 1TB SATA disk, and the Godaddy server for cloud storage. University employee dataset is considered for testing our system. The documents that are included of type such as, text, pdf, document, word, etc. From the point at which the backup button is selected the backup operation starts, the data that are stored in the local storage are transferred to the Application-Based module. The application based module chunks the data according to the type of data being uploaded. Such that the images of the employees are chunked separately, and the documents that are said to be the resume of the employees are chunked separately. The chunked data are then sent to the pre-filter, which is the file similarity search module. The file similarity search module search for the similarity between the files through the fitness functions. The fitness functions, search for the contents in each file and check for its similarity through its contents. A file is considered to be similar, when the content of a file matches with the content of the other file.

V. Discussion

Our experimental results present both the de-duplication efficiency (DE) and Backup Window Size (BWS) for individual users respectively. The following section compares proposed method with Semantic Aware, Application aware and causality based deduplication system. Fig 2 plots backup windows as a function of network bandwidth from an experiment where we select three backups and simulate a network environment with different bandwidths: 100KB/s, 800KB/s, 1MB/s, 2MB/s. The backup windows of Backup that has no duplicate files and no duplicate chunks already backed up by the same client. Backup window size is also reduced by 8% while it is compared with SAM, AADedup and CAB.

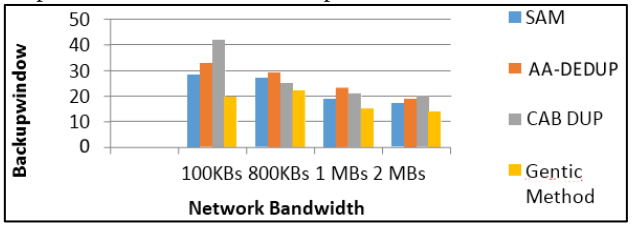

Fig. 2: Backup Window

All rights reserved by *www.ijsrd.com*

The following Fig compares the cumulative de-duplication efficiency of the three de-duplication methods. We define deduplication efficiency as the ratio between the amount of the redundant data actually removed and the total amount of the redundant data in each de-duplication method.The results shows that proposed Genetic method removes about 90.69% of redundant data as it is compared with SAM and AADedup for the backup sessions.

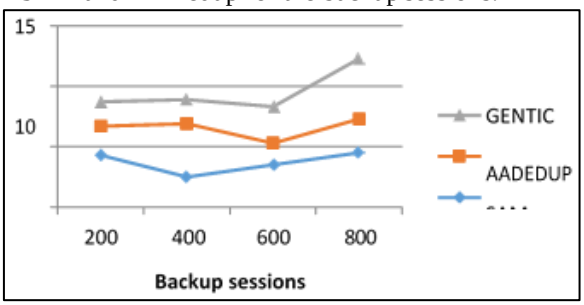

Fig. 3: Deduplication Efficiency

VI. Conclusion

The proposed system enhanced storage by utilizing file modification pattern to generate the active pairs of input for optimized deduplication algorithm. The active pairs in GA of input generated are sent as population and the high optimized process of deduplication could done with large of results in a minimal amount of time. Thus, increasing the deduplication efficiency of the personal computing system, and reducing the time taken for each and every backup operation. In the future, the same deduplication methodology can be implemented in multiple client in multicloud environments using a more improved and optimized machine learning technology such as active learning techniques.

References

[1] Akshara k., Soorya P.(2012) "Replica free repository using genetic programming with decision tree" in International Journal of Advanced Engineering Applications, Vol.1, Iss.2, pp.62-66
[2] D. Harnik, B. Pinkas, and A. Shulman-Peleg, 2010.“Side channels in cloud services: Deduplication in cloud storage,” IEEE Security and Privacy,vol. 8, pp. 40–47.
[3] D. T. Meyer and W. J. Bolosky,2011. "A study of practical deduplication," in FAST'11: Proceedings of the 9th Conference on File and Storage Technologies.
[4] M. G. de Carvalho, M. A. Goncalves, A. H. F. Laender, and A. S. da Silva,2006. “Learning to deduplicate,” in Proc. of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 41–50.
[5] Moises G. de Carvalho, Alberto H.F. Laender, Marcos Andre Goncalves, and Altigran S. da Silva 2012.” A Genetic Programming Approach to Record Deduplication” IEEE transactions on knowledge and data engineering, vol. 24, no. 3.
[6] P. Mell, and T. Grance. 2009.The NIST Definition of Cloud Computing, The National Institute of Standards and Technology (NIST). United States Department of Commerce Version 15.
[7] Shu-Ching Wang, Kuo-Qin Yan, Shun-Sheng Wang, Bo-Wei Chen 2013. “LDMCS: a Lightweight Deduplication Mechanism under Cloud Storage”, Business and Information, E32-E40.
[8] T. Clements, I. Ahmad, M. Vilayannur, and J. Li, 2009 “Decentralized deduplication in SAN cluster file systems,” in USENIX’09.
[9] T. Yujuan, et al., 2011."CABdedupe: A Causality-Based Deduplication Performance Booster for Cloud Backup Services,"IPDPS IEEE Interntional, pp. 1266-1277.
[10] V. Michael, S. Savage, and G. M. Voelker,2009. “Cumulus: file system Backup to the Cloud,” in Proceedings of the Conference on File and Storage Technologies (FATS '09), San Francisco, Calif, USA.
[11] Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu:2014. “Application-Aware Local-Global Source Deduplication for Cloud BackupServices of Personal Storage”IEEETransactions on Parallel and Distributed Systems, Vol. 25, No. 5.
[12] Yujuan Tan · Hong Jiang · Edwin Hsing-Mean Sha ·Zhichao Yan · Dan Feng 2013.“SAFE: A Source Deduplication Framework for Efficient Cloud Backup Services”, Journal of Sign Process Syst (2013) 72:209–228 Springer Science, Business Media New York
[13] Yujuan Tan, Hong Jiang, Dan Feng Lei Tian, Zhichao Yan,Guohui Zhou 2010.“SAM: A Semantic-Aware Multi-Tiered Source De-duplication Framework for Cloud Backup”, 39th IEEE International Conference on Parallel Processing.
[14] Zhe SUN, Jun SHEN, Jianming YONG, 2011.“DeDu: Building a Deduplication Storage System over Cloud Computing”, 15th IEEE International Conference on Computer Supported Cooperative Work in Design.

All rights reserved by *www.ijsrd.com*