Improving Security and Storage availability in Deduplication Storage System

Miss.JayashriPatil, Dr.SunitaBarve, Mrs. MayuraKulkarni
DepartmentofComputerengineeringMITAcademyofEngineering,SavitribaiPhulePuneUniversity

Abstract

Data Deduplication has become increasingly important in a storage system. Deduplication is one such storage optimization technique that avoids storing duplicate copies of data and only one occurrence of the data is stored on storage media. It detects and eliminates redundant data. Data Deduplication storage system saves the storage space and storage cost is reduced. The proposed system manages to detect duplicate data and stores unique data over the storage nodes. The deduplication algorithm detects maximum duplicate data which is the main challenge. To distribute fragments to multiple nodes fragment placement algorithm uses T-coloring. Security is increased in deduplication storage. The proposed system increases the storage efficiency and achieves the high level of security for data.

Keywords

Deduplication, Similarity, Locality, Storage System.

I. INTRODUCTION

Deduplication is method of removing duplicates copies of data and duplicate copies are replace with pointers, which points to the identical copy which is stored in storage as a single instance of data. Data set or stream is examined at sub-file level and only identical data is stored or saved. The work flow of data deduplication consists of Input file, Hash Computation, Computing hash with hash index table, whether match found or notify esset pointer to existing data location and if no save data to memory and its hash to hash index. The duplicate data segments in Deduplication technology are detected with the help of finger print. Hash algorithm SHA512 isused to assign hash value to the data chunk, hash value which is used to identify identical data segment.

Deduplication is classified as file level deduplication and block level deduplication. In the File level single file is considered as chunk, a small difference in file make that file unique due to which deduplication ratio decreases. In block level, file is divided into segments and segments are considered as duplicate data. Smaller granularity eliminates more duplicate data. Data deduplication is also classified as source and target based or online and offline based deduplication. The difference between source based and target is that deduplication is processed before or after writing to the disk. In source based, duplicated data is eliminated before writing to the storage media which consumesfewer resources and duplicate is not transferred over the network that saves the network bandwidth. In source-based deduplication client do the deduplication process only identical data is the backup, it saves bandwidth as well as storage space, but there is the extra computational load on the backup client. In target-based deduplication, the repeated data is removed after writing to storage devices, space is consumed until repeated data is eliminated.

T-coloring is used to distribute data blocks to several nodes. To prevent data from unauthorized access, fragments are stored in such a way that attacker fails in guessing the location of fragments. Each node consists of the single fragment of the particular file. Attacker surface area is increased by storing the chunks at a distance from one another. If a file is stored on the single node, there are chances of data loss and load on single node increases. This provides the high level of security.

The sequence of data stream appear is same order for each backup with high probability. In some backupstreamsthelocalitybetweenthefirst,second,andnextbackupshavea very high probability that chunksare in the same order and normally chunks lookups are one by one. However, this approach shows lows peed on backup stream with weak locality.

Section I contains the Introduction, Section II contains related work of deduplication. Section III contains Methodology, architecture and some algorithm. Section IV contains conclusion.

II. RELATED WORK

Wen Xia, Hong Jiang, Dan Feng, and Lei Tian, (2015) considered two different approaches, resemblance detection and duplicate detection. In resemblance detection, similar data objects are detected at a byte level.Where as in a case of duplicate detection, duplicate data is detected at chunk level. Duplicate detection uses Secure-fingerprint based deduplication method and resemblance detection considers super-feature based delta compression method. To detect similar data chunks Deduplication-Aware Resemblance and Elimination (DARE) efficiently exploit existing duplicate-adjacency information, this achieves highest throughput data reduction. Dup Adj considers two fragments similar if in duplicate system their neighboring chunks are the duplicate. For removing redundancy among similar data chunks delta compression gained attention. If chunk 1 is similar to chunk 2, delta compression calculates and stores the mapping and difference between this two chunks.

Xia Wen, Hong Jiang, Dan Feng, and Yu Hua (2015), Proposed Similarity and locality based approach (SiLo). This combined approach reduce RAM usage, keep duplication accuracy, and it also increases throughput.Silo approach can effectively improve the disk bottleneck with an adequate overhead of CPU, memory, and storage when performing fingerprint lookup, thus improving the throughput of data deduplication. Locality based algorithm is used to distribute data blocks to multiple storage nodes, due to which load is balance among storage nodes.

T. Yang, H. Jiang, D. Feng, Z. Niu, K. Zhou, and Y. Wan, (2010) proposed DEBAR, a scalable and high-performance deduplication storage architecture for Backup and archiving, Several backup clients are considered by DEBAR. DDFS uses the bloom filter to lower disk index access, Limited memory space is required in this approach compared to DDFS. No of a backup server is used in parallel for high throughput. Two schemes are used in TDFS for data deduplication. Deduplication performance in increased with the drop in scalability. Data fragments collected in dedupe first and new data chunk are detected in dedupe second.

Fu M. et al., (2016),Fragmentation is categories into two types sparse container and out-of-order container. At the time of restore, in sparse more chunks are not at all accessed, in out of order thefragments are accessed again and again. Both affect the restore performance. Toincrease restore performance sparse containers is decreased. To improve the performance History-Aware Rewriting algorithm is used but deduplication ratio is slightly decreased. To determine outoforder container that disturbs restore performance.Cache-Aware Filter exploit restore cache knowledge.

J.Liu,Y.Chai,C.Yan and X.Wang(2016),propose a new Delayed Container Organization, to increase the restore performance in data deduplication system. The construction of containers is delayed after assigning data chunk in non volatile memory.DCO have higher restore speed, Better optimization based on a large amount of information, space saving is medium. DCO has three advantages Higher UDRs Containers are produced, More data is duplicated, Restore is speedup.

Bo Mao, Hong Jiang Suzhen Wu, Lei Tian (2014) have proposed Performance Oriented I/O Deduplication approach. If data deduplication is directly applied on primary storage then it will cause two problems, fragmentation of data on disks and space contention in memory. Selective dedupe and iCache two approaches are considered in Performance oriented deduplication(POD).POD support features like capacity saving, performance enhancement, small writes elimination, large writes elimination and cache partitioning strategy. POD achieves comparable or better capacity saving than idedupe.

C. Li, S. Wang, Xiaochunyun, X. Zhou and G. Wu (2014), have proposed MMD. Multiple disks are used to boast the reading performance, each disk is used independently as the logical device. Due to fragmentation in data Deduplication system, reading performance is decreased. MMD storage approach is used which increases read performance and it is different from RAID. An algorithm is used to assign the container to disk.MMD performance is higher compared to RAID0.

MazharAli,KashifBilal,SameeU.Khan,BharadwajVeeravalli,KeqinLi,AlbertY.Zomaya,(2015), proposed Division and Replication of data for Optimal Performance Security (DROP). In this methodology file is divided into fragments, fragments are stored at centrality distance from each other and finally fragments are replicated. Fragments are replicated is such way that it reduces the access the time. In this it focuses on performance and security. For reconstruction of file, itprovidesimproveretrievaltimeforaccessingparticularfile fragment. To increase the surface area of attacker, maximum nodes are used to store data on storage nodes. Nodes consist of single fragment of particular file.

III.METHODOLOGY

Architectural Design gives the overall view of system components and there interface with each other as shown in Fig.1. We proposed distributed Deduplication system with higher security in which fragments are distributed across multiple storage nodes. There are two issues to bead dressed Firstly, how does the system identify the data duplications. Secondly, how does the system manage and store data for security. Two kinds of entities are involved in Deduplication system, including, user and storage service provider. User entity that wants to out source data storage to the storage service provider and access the data later. Storage service provider is an entity that provides the out sourcing data storage service for the users. The user data is distributed across multiple nodes. In Deduplication mechanism block level deduplication is consider.

A. SystemArchitecture

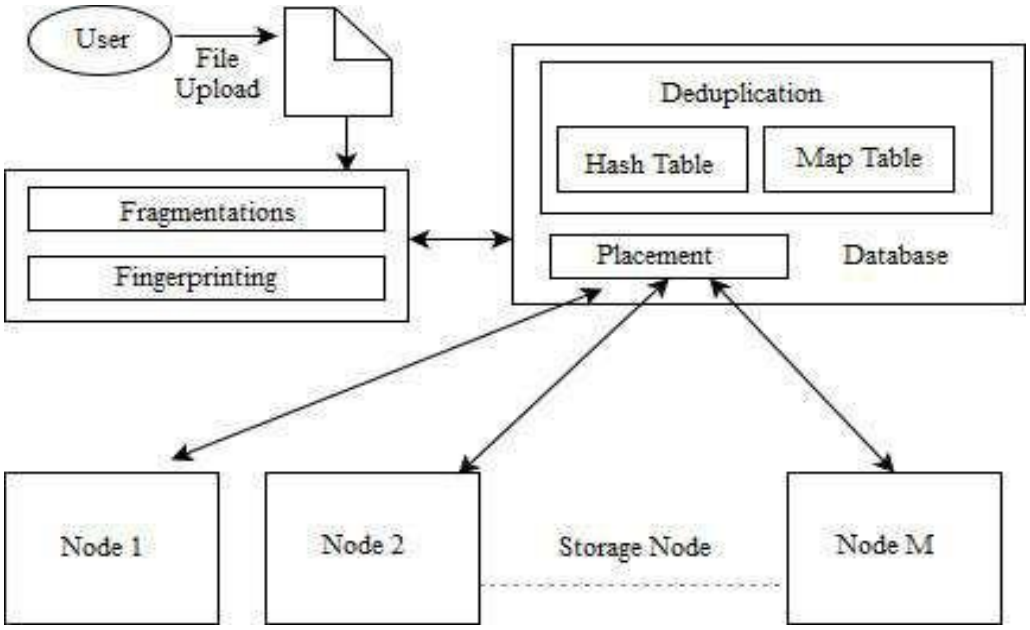

Fig1:Architecture Diagram

1) ***Datadeduplication***: Data Deduplication include four steps- data fragmentation, hash computing, index querying and index updating. Data Deduplication module divides incoming data into fragments.Input file is fragmented into fixed sized blocks. Each fragment is assigned a hash value Hash Table is used for deduplication. It contains records of the index value of each fragment. The fingerprint is queried in the Hash table. It is used for filtering the duplicate chunk.Fingerprint index is checked to identify whether the corresponding fragment is redundant or not. A new fragment is identified and the corresponding fingerprint is written to a hash table.

2) Storing fragments on different nodes: For Security, T-coloring is used. Nodes are separated by T coloring. The intention is to find a node for placing fragment. The technique uses fragment placement algorithm. The aim of this algorithm is to place fragments of file on different storage node so that attacker fails in guessing the location of fragment. M no of nodes we have to consider. Each node consist of only one fragment of the particular file.

3) Database: The database consists of various tables such as User registration table, User File, Hash Table, Hash 2-MB, Node, Tpa and admin table. User registration table- Consist of all registered user details such as Full name, email, contact no, username, password. User File table-Consist of file details information such as File name, date upload, deduplication in percentage, file size in bytes, status. Hash table- Hash table is used to store the fingerprint of all fragments. It contains the list of all fragments of the file along with its hash value, file name, duplicate, reference file name, part name. Node table- It consists of all nodes. It shows where actually the on which node.

4) ***Algorithm For deduplication***

```
Input: Input data file
Output: Unique Fragments of file
BEGIN
Divide file into no of fragments
Assign has h value to each fragment of file
Receive a fragment and lookup its finger print in the finger print index.
If the fragment is duplicate, then Eliminate the fragment
Else
Update the finger print index in database. End if
END
```

5) ***Algorithm For Fragment Placement***

```
Input: Fragments of file
Output: Fragments stored on nodes.
BEGIN
Let Mbe then o of nodes
Nbe then o of fragments of the file For each fragment of file
DO
Check the available nodes
Calculate centrality
End DO
If col=open_color and size of node is greater than fragment size.
then
ASSIGN fragment to the node.
Return all the nodes at distance T from fragment and Store in temporary setT'.
```

```
ENDIFEND
FOREND
```

B. Mathematical model

The mathematical model of the project can be represented by using at uple S={In,P,O}

Input-Data Files

Process-(X,F,M)

Where X is the set of fragments

X={X1,X2,X3……}

F is set

Of function F={f1,f2,f3}f1=Hash function

f2=Perform Deduplication check.f3=Store fragment to the particular node.

M is the no of nodes

M={M1,M2,….,Mn}where the fragment is stored.

Output-Unique fragments stored on no

IV.CONCLUSIONS

To eliminate duplicate data and increase the security level in deduplication storage system, we proposed duplicate detection and fragment placement algorithm. The deduplication algorithm increases duplication percentage by eliminating more redundant data and saves more storage space. Duplicate data is detected by considering fixed fragment size. Security level is increased by increasing the attacker surface. T-coloring mechanism is used for fragment placement. This prevents unauthorized access to data from other users which increase the security.

REFERENCES

[1] Xia Wen, Hong Jiang, Dan Feng, and Yu Hua (2015), “Similarity and locality based indexing for high performance data deduplication", IEEE Transactions on Computers 64(4): 1162-1176.

[2] Ali Mazhar, Kashif Bilal, Samee Khan, Bharadwaj Veeravalli, Keqin Li, and Albert Zomaya (2015),“DROPS: Division and Replication of Data in the Cloud for Optimal Performance and Security", IEEE Transactions on Cloud computing: 1-14.

[3] Banu A. and Chandrasekar C. (2012), “A survey on methods”, International Journal of Computer Trends and Technology 3(3): 364-368.

[4] Dutch T and William J. Bolosky (2012), “A study of practical deduplication", ACM Transactions on Storage 7(4): 1-14.

[5] Vickerman, R and Abirami S (2014), “A study on various data de-duplication systems",International Journal of Computer Applications 94(4).

[6] Xia Wen, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou (2016), “A comprehensive study of the past, present, and future of data deduplication", Proceedings of the IEEE 104(9): 1681-1710.

[7] Mao Bo, Hong Jiang, Suzhen Wu, and lei Tian (2014), “POD: Performance oriented I/O deduplication for primary storage systems in the cloud" In Parallel and Distributed Processing Symposium : 767-776.

[8] Xia Wen, Hong Jiang, Dan Feng, and lei Tian (2016), “DARE: A deduplication-aware resemblance detection and elimination scheme for data reduction with low overheads", IEEE Transactions on Computers 65(6): 1692-1705.

[9] Yang Tianming, Hong Jiang, Dan Feng, Zhongying Niu, Ke Zhou, and Yaping Wan (2010), “DEBAR: A scalable high performance deduplication storage system for backup and archiving",IEEE International Symposium on Parallel Distributed Processing : 1-12.

[10] Li Chao, Shupeng Wang, Xiaoyang Zhou, and Guangjun Wu (2014), “MMD: An Approach to Improve Reading Performance in Deduplication Systems", International Conference on Networking, Architecture, and Storage : 93-97.

[11] M.Fu(2016), “ReducingFragmentation forIn-lineDeduplicationBackupStorage viaExploitingBackupHistoryand Cache Knowledge,IEEETransactionsonParallelandDistributedSystems” , 27(3):855-868.

[12] Liu Jian, Yunpeng Chai, Chang Yan, and Xin Wang (2016), “A Delayed Container Organization Approach to Improve Restore Speed for Deduplication Systems", IEEE Transactions on Parallel and Distributed Systems 27(9): 2477-2491.