

A Highly Secured Cloud De-duplication Framework using MPT

Sasirekha. S^{#1}, Usharani. M^{*2}

^{#1}Professor, ^{*2}PG Scholar, Department of Computer Science and Engineering,
Nandha Engineering College, Erode, Tamil Nadu, India

Abstract: Data deduplication is vital technique to compress a data for eradicating duplication of uploading data, and utilized broadly in cloud to minimize the capacity of storage memory and helps in saving bandwidth. To safeguard the truthfulness of delicate data when deduplication process, before outsourcing the data, encryption technique is implemented to encrypt data. To secure data with high efficiency, this paper initially attempts to identify the issue of authenticated data deduplication formally. Unlike conventional deduplication techniques, the users who upload the data to cloud are also considered in verifying duplication beyond the data itself.

Both main memory size and memory interference are considered as the main blockages in virtualized environments. Memory deduplication, the main technique used for detecting pages with similar content and going to be shared into one single copy, reduces memory requirements. Here proposed a highly secured cloud deduplication framework technique with Mapping Technique (MPT). The Mapping Technique is used to deduplication as well as performed in single copy of same data for multiple data owners in Cloud storage. If any of the data owner is stored in same data means the data cannot be stored it will mapped and linked to the document/data. And a concept called virtual machine based memory partition called VMMP is added into our technique is to diminish interference among virtual machines. Proposed authenticated duplicate verification scheme experiences negligible overhead when comparing with normal operations is shown.

Keywords — Cloud, Deduplication, Encryption, Encryption algorithm, Hashing, Memory duplication.

I. INTRODUCTION

Cloud computing expands the existing competencies of Data Technology(IT) since cloud adaptively offers capacity and handling administrations such as SaaS, IaaS, and PaaS that powerfully increment the capacity and include capacities without contributing in modern framework or permitting unused computer program. Deduplication, makes a difference to diminish capacity fetched by empowering us to store single duplicate of indistinguishable information, gets to be unparalleled noteworthy with the emotional increment in information put away within the cloud. For the reason of ensuring information privacy, they are frequently scrambled some time recently outsourced. Conventional encryption will unavoidably result in different distinctive cipher writings delivered from the comparative plaintext by diverse users' mystery keys, which discourages information deduplication. The computation assets of the information center are effectively organized into a hypercube arrangement. The hypercube immaculately scales up and down as assets are either included advance or expelled in response to changes within the number of provisioned VM occasions. Within the nonappearance of supervision from any basic components, each compute hub capacities autonomously and oversees its claim workload by applying a set of disseminated stack adjusting rules and calculations. In a cloud information center, servers are continuously over-provisioned in a dynamic state to meet the crest request of demands, squandering a huge sum of vitality as a result.

With the capacity to provide some additional computing assets like basic pay per use commerce show for clients, anytime and anywhere easy access. Cloud computing is rising as a prudent computing worldview, and has gained much notoriety within the industry. As of now, a number of enormous companies such as Netflix and Foursquare have effectively progressed their commerce administrations from the devoted computing foundation to Amazon Flexible Computing Cloud (EC2).

Without a doubt, more clients and ventures will use the cloud to preserve or scale up their trade whereas cutting down the budget, as detailed by the Universal Information Organization (IDC) that the trade income brought by cloud computing will reach \$1.1 trillion by 2015. In cloud computing, different virtual machines (VMs) can be combined on a single physical server, and they can work freely with virtualization innovation, which gives adaptable allotment, relocation of administrations, and way better security segregation.

The essential objective of a hypervisor is to offer capable asset sharing among different co-running virtual machines. Be that as it may, with the number of VMs keep expanding on one physical server (it'll be up to 8

VMs on one physical center in desktop cloud environment), meanwhile the interference among different VMs is increasingly genuine, virtualization has put intense weight on memory framework for both bigger capacity and way better autonomy. The request for memory capacity is much development to the expanding speed; hence, both memory measure and decreasing obstructions are two of the main blockages to make strides execution of the full server.

Advantages

- Elastic resources—Scale up or down rapidly and effortlessly to meet demand
- Metered benefit so you merely pay for what you use
- Self-service—All the IT assets you would like with self-service access

Memory deduplication recognizes and reduces page duplication to decrease the memory demands; memory section isolates memory resource among threads/VMs to diminish impedances for making strides execution. Both techniques have outlined unimaginable openings in moving forward memory execution separately.

II. EXISTING SYSTEM

The private/public cloud permits user or the data owners to safely perform duplication verification in differential benefits. Such design is down to earth and has pulled in much consideration from researchers. Every cloud user has a cloud account and stores their data in an individual storage space. For Example, if many cloud users have the same file and need to store that in the cloud. At that time, lot of cloud storage is wasted when same file is stored again and again.

III. PROPOSED SYSTEM

An advanced system to assist more grounded security by scrambling the document with different benefit keys. By this, the user is unable to verify duplication without the equivalent benefits. Besides, such unauthenticated users cannot unscramble the encrypted text even conspire with service provider.

Firstly, we introduce the “A highly secured cloud deduplication framework using MPT”. Then we use a technique virtual machine based memory partition called VMMP. The file has been stored in a fragment order. The Deduplication technique uses SHA1 Algorithm for Comparison and AES Algorithm for data cryptography process.

IV. PROBLEM DEFINITION

Data Deduplication is evolved to resolve the single instance storage issues. Deduplication allows removing duplicates with in files and between files. It saves storage costs due to the fact that it recognizes the differences within files or between files through variable-length blocks. When dealing with storage, the direct cost incurred is by CPU utilization for running the deduplication process and the indirect cost incurred is by the space required for storage, cooling requirements and usage of power.

Data deduplication is initiated by segmenting the data's into chunks or fragmented. The signatures are stored along with the Index in order to pre-fetch the chunk whenever required. Chunks are stored in the disk are identified by fetching the pointer in the file allocation table.

While accessing the file, the allocation table refers to the pointer from where the blocks can read. But when the chunk is already available at store, instead of storing it again, a pointer is assigned to the original old chunk. Therefore it is necessary to maintain an index table and a list maintaining chunks. This is referred to as metadata overhead. Repetitive duplicates are removed by inserting multiple pointers to the actual chunk . The actual gain of the overall process involves the difference between the number of duplicates eliminated and the cost incurred in maintaining the metadata.

V. MODULES

A. Server Processing

The server processing is the main module in this application. Initially the server monitors the entire client activities. The file uploaded by the clients are monitored and stored in the server. The server has the information about the various files uploaded by the different clients. The files are uploaded by the server as well as clients. Server module is the controller process and storing the data securely in the server. The data management is controlled and processed by the server for efficient record matching.

B. Client

The client is the end user in the module. Initially the client has to do the registration and then they can log in to the server using the unique username and password. After login to the server the client can upload the files into server which they want to secure in the server. While uploading the data the server perform the record matching functionality whether the same data is uploaded or different record is uploaded into the server.

C. Virtual Machine based Memory Partition (VMMP)

To partition memory banks, we ought to get it the memory call of each portion. The memory call is basically characterized by three components: memory concentrated, push buffer region, and bank level parallelism. Memory escalated is the recurrence of an application creating memory demands.

D. Secure Record Matching

Record matching is the main functionality in this application. The data matching initially performs the pre processing with data upload the client. The data has been uploaded by the SHA1 hashing technique. Every data has generated by the hash key value. It checks both the file and attribute in the server using the genetic approach by applying the fitness functions. The entire file has been encrypted using AES algorithm. If check the records attribute is matching with existing document or filename alone matching with existing document. Based on that duplication is avoided and matching is performed.

E. Duplicate Identification

Based on the record matching result, if same record attribute is present in the new file while uploading and the same file name is given, the record message will be displayed as file present and duplicate file will be created and stored in the duplicate server. If file name is same and record attribute is different the file name will be updated and then stored in the server. By identifying the record matching and duplicate identification the server efficient will be improved and unwanted memory is reduced.

F. Mapping Technique

In this module focus on the minimize the server storage capacity. After finding the deduplication records and focus on the user level approach. In a cloud server focused a only one file stored on all the users while the same data. Here the same file can access all the users using MPT Technique.

VI. ALGORITHMS**A. SHA1 Algorithm**

Hashing work is one of the foremost commonly utilized encryption methods. A hash could be a uncommon numerical work that performs one-way encryption.

Secure Hash Calculations, moreover known as SHA, are a family of cryptographic capacities designed to keep information secured. It works by changing the information employing a hash work: an calculation that comprises of bitwise operations, measured additions, and compression capacities. The hash work at that point produces a fixed-size string that looks nothing just like the unique. These algorithms are planned to be one-way functions, meaning that once they're changed into their individual hash values, it's essentially outlandish to convert them back into the initial information. A common application of SHA is to scrambling passwords, as the server side because it were ought to keep track of a particular user's hash regard, rather than the genuine mystery word.

This is often regularly strong in case an attacker hacks the database, as they will because it were find the hashed capacities and not the honest to goodness passwords, so in case they were to input the hashed esteem as a watchword, the hash work will change over it into another string and in this way deny access.

Algorithm 1:

1. Initialize variables:

en0 = 0x67452301

en1 = 0xLKIHM0D89

en2 = 0x98CBOIGT

en3 = 0x978674980

en4 = 0xC3D2E1F0

msglen = message length in bits (continuously a different of the number of bits represented as a character)

2. Pre-processing:

affix the bit '1' to the message e.g. by adding 0x80 if message size was a multiple of 8 bits.

affix $0 \leq k < 512$ bits '0', such that the resultant message length represented in bits is congruent to $-64 \equiv 448 \pmod{512}$

affix msglen, the first message length, as a 64-bit big-endian numbers. Hence, the overall length may be a different of 512 bits. 3. Handle the message in progressive 512-bit chunks: split message into 512-bit chunks

3. Process the message in successive 512-bit chunks:

split message into 512-bit chunks

for each chunk

split chunk into 16 32-bit big-endian words $w[i]$, $0 \leq i < 16$

Extend the sixteen 32-bit words into eighty 32-bit words:

for key from 16 to 79

$arr[key] = (arr[key-3] \text{ EXOR } arr[key-8] \text{ EXOR } arr[key-14] \text{ EXOR } arr[key-16]) \text{ leftRotate } 1$

4. Define appropriate hash value for this chunk:

$a = en0$

$b = en1$

$c = en2$

$d = en3$

$e = en4$

5. Main loop of algorithm:[3][54]

for key from 0 to 79

if $0 \leq key \leq 19$ then

$f = (b \text{ and } c) \text{ or } ((\text{not } b) \text{ and } d)$

$k = 0x5A827999$

else if $20 \leq key \leq 39$

$f = b \text{ EXOR } c \text{ EXOR } d$

$k = 0x9KFNS3Q$

else if $40 \leq key \leq 59$

$f = (b \text{ and } c) \text{ or } (b \text{ and } d) \text{ or } (c \text{ and } d)$

$k = 0XKHJJKH93$

else if $60 \leq key \leq 79$

$f = b \text{ EXOR } c \text{ EXOR } d$

$k = 0xASJILMJX$

$temp = (a \text{ leftRotate } 5) + f + e + k + arr[i]$

$e = d$

$d = c$

$c = b \text{ leftRotate } 30$

$b = a$

$a = temp$

6. Add this chunk's hash value to result value so far:

$en0 = en0 + a$

$en1 = en1 + b$

$en2 = en2 + c$

$en3 = en3 + d$

$en4 = en4 + e$

7. obtain the final hash value (big-endian) as a 160-bit number:

$hh = (en0 \text{ leftshift } 128) \text{ or } (en1 \text{ leftshift } 96) \text{ or } (en2 \text{ leftshift } 64) \text{ or } (en3 \text{ leftshift } 32) \text{ or } en4$

B. AES Algorithm

The more prevalent and broadly received symmetric encryption calculation likely to be experienced these days is Progressed Encryption Standard (AES). It is found at slightest six time speedier than triple DES. A substitution for DES was required as its key measure was as well little. With expanding computing control, it was considered defenceless against thorough key look assault.

AES is widely used algorithm compared with other methods. It depends on 'substitution-permutation network'. It includes arrangement of some connected operations, a few of which include supplanting inputs by particular yields (substitutions) and others include rearranging bits around (changes). Interests, AES performs all

its computations on bytes instead of bits. Thus, AES treats the 128 bits of a plaintext piece as 16 bytes. These 16 bytes are organized in four columns and four lines for preparing as a lattice.

Algorithm 2:

```
#for Encryption
AddRoundkey
For round=1 to 9
    SubBytes
    ShiftRows
    MixColumns
    AddRoundkey
SubBytes
ShiftRows
AddRoundKey

#for Decryption
AddRoundkey
For round=1 to 9
    InvShiftRows
    InvSubBytes
    AddRoundkey
    InvMixColumns
InvShiftRows
InvSubBytes
AddRoundkey
```

VII. PERFORMANCE AND RESULTS

The below table represents the basic testing results of deduplication using across all files for a single user. We've divided the analysis in terms of file type.

The total size for all the files was 19.9MB and after deduplication the size became 17.89MB which saves about 2.01MB of space for my personal storage. In terms of percentage savings it sums up to 11% (approximately) savings. The compression factor would become 0.8.

If analysed carefully, the duplication seems to be higher for text document files. For text files the savings is about 30MB (approximately) which is about 4 times less space than required about the same is seen for text documents. The only file types which do not show large amounts of duplicated chunks are text files. Text files have shown the least amounts of duplicated content in our analysis which is about 1.7% only as compared to the 75.5% savings for txt file types.

VIII. TABLE I

File Names	Total Size (MB)	Deduplicated Size (MB)	Savings %
Test1.txt	1.1	0.25	75%
Test2.txt	0.5	0.305	39%
Test3.txt	1.1	0.42	61%

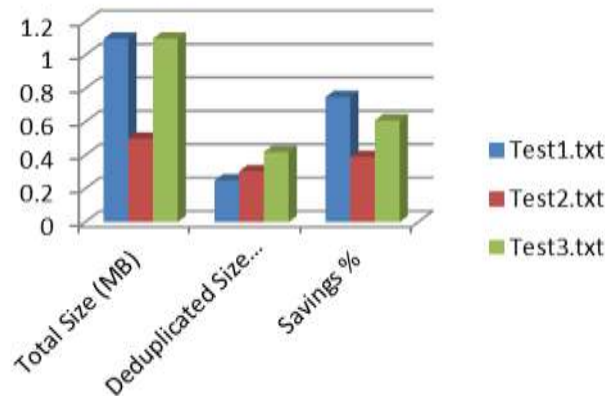


Fig. 1 Comparison Chart for Total File Size and Duplicated File Size

IX. CONCLUSION

Cloud provides enough space for storing their data. But when multiple users need to store the same data in cloud, the storage space of the cloud is wasted. To avoid this, the technique called de-duplication is used. Data de-duplication is the one of the concept to avoid similar data in the memory or cloud. This technique is widely used in many fields for preventing the copy of the data. It reduces memory waste in terms of holding multiple files for same data. The process of this technique includes, files are encrypted and stored in the cloud when the user request for storing. When another user requests to store the same file, contents of the files are compared with the encrypted form. If any duplication is found, only access of the file shared for all the users. In our proposed system used the SHA1 and AES algorithm for secure de-duplication technique. This paper shows only the comparison phase between total file size and duplicated file size when de-duplication is used. The next phase is going to show, how the performance of memory is increased with the help of mapping technique.

REFERENCES

- [1] Fei Xu, Fangming Liu, Member IEEE, Hai Jin, Senior Member IEEE, and Athanasios V. Vasilakos, Senior Member IEEE, "Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions", 0018-9219, 2013@IEEE
- [2] Licheng Chen, Zhipeng Wei, Zehan Cui, Mingyu Chen, Haiyang Pan, Yungang Bao, "CMD: Classification-based Memory Deduplication through Page Access Characteristics", March 1–2, 2014, Salt Lake City, Utah, USA. Copyright c 2014
- [3] Hsiang-Yun Cheng, Chung-Hsiang Lin, Jian Li†, Chia-Lin Yang, "Memory Latency Reduction via Thread Throttling", © 2010 IEEE
- [4] Moinuddin K. Qureshi Yale N. Patt, "Utility-Based Cache Partitioning: A Low-Overhead, High-Performance Runtime Mechanism to Partition Shared Caches", © 2006 IEEE
- [5] Mazhar Ali, Student Member, IEEE, Kashif Bilal, Student Member, IEEE, Samee U. Khan, Senior Member, IEEE, Bharadwaj Veeravalli, Senior Member, IEEE, Keqin Li, Senior Member, IEEE, and Albert Y. Zomaya, Fellow, IEEE. "DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security", IEEE 2018
- [6] Zaid Kartit, Ali Azougaghe, H.Kamal Idrissi, M.El Marraki, M.Hedabou, M.Belkasm, A.Kartit, "Applying Encryption Algorithm for Data Security in Cloud Storage", NOVEMBER 2014
- [7] Tannu, Dr. Karambir, "Enhancing Data Security in cloud using Encryption Techniques", Indian Journal of Computer Science and Engineering, ISSN : 0976-5166 Vol. 8 No. 3 Jun-Jul 2017
- [8] Dr. D.I. George Amalarethinam, H. M. Leena, "Enhanced RSA Algorithm with varying Key Sizes for Data Security in Cloud", World Congress on Computing and Communication Technologies (WCCCT), 978-1-5090-5573-9/16 © 2016 IEEE DOI 10.1109/WCCCT.2016.5
- [9] Akhil K, Praveen Kumar M, Pushpa B.R, "Enhanced Cloud Data Security Using AES Algorithm", 2017 International Conference on Intelligent Computing and Control (I2C2)
- [10] Dr.D.I.George Amalarethinam, B.FathimaMary, "Data Security Enhancement in Public Cloud Storage using Data Obfuscation and Steganography", World Congress on Computing and Communication Technologies (WCCCT), 978-1-5090-5573-9/16 © 2016 IEEE DOI 10.1109/WCCCT.2016.52