International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
Since 2012

http://www.ijcsjournal.com Volume 5, Issue 1, No 28, 2017 ISSN: 2348-6600
Reference ID: IJCS-294 PAGE NO: 1941-1946

SECURE AND OPTIMAL PERFORMANCE APPROACH FOR DATA DEDUPLICATION IN CLOUD

V.BHUVANESWARI
Mail id: bhuvanavettri@gmail.com , Mobile : 9943382307.

S. Nithyadhevi, M.E., Assistant Professor

Department of Computer Science and Engineering.

Sir Issac Newton College of Engineering and Technology.

Pappakoil,Nagappatinam-611102

Abstract - Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting deduplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this project makes the first attempt to formally address the problem of authorized data deduplication. Different from traditional deduplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. Several new deduplication constructions supporting authorized duplicate check in a cloud architecture is proposed. Security analysis demonstrates that is secure in terms of the definitions specified in the proposed security model. Improving the performance of primary storage systems and minimizing performance overhead of deduplication.

Keywords: I/O Deduplication, Data Redundancy, Performance, Storage Capacity

1. INTRODUCTION

Technique in Cloud backup and archiving applications to reduce backup window, improve the storage-space efficiency and network bandwidth utilization. Recent studies reveal that moderate to high data redundancy clearly exists in VM (Virtual Machine) enterprise and High-Performance Computing (HPC) storage systems. These studies have shown that by applying the data deduplication has been demonstrated to be an effective technology to large-scale data sets . HPC storage systems, can be achieved . For example, the time for the live VM migration in the Cloud can be significantly reduced by adopting the data deduplication technology. Primary and secondary deduplication systems can be further subdivided into inline and offline deduplication systems. Inline systems deduplicate requests in the write path before the data is written to disk. Since inline deduplication introduces work into the critical write path, itnoften leads to an increase in request latency. On the other hand, offline systems tend to wait for system idle time to deduplicate previously written data. Since no operations are introduced within the write path; write latency is not affected, but reads remain fragmented. The existing data deduplication schemes for primary storage,such as iDedup and Offline-Dedupe, are capacity oriented in that they focus on storage capacity savings andonly select the large requests to deduplicate a nd bypass all the small requests. The rationale is that the small I/O requests only account for a tiny fraction of storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems and are at the root of the system performance bottleneck.

Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burstiness. From a performance perspective, the existing data deduplication schemes fail to consider these workload characteristics in primary. The storage systems will missing the opportunity to address one of the most important issues in primary storage, that of performance.

2. LITERATURE SURVEY

1.Fast and secure laptop backups with encrypted de- duplication Back up algorithm (2010)

The data which is common between users to increase the speed of backup and reduce the storage requirement Supports client-end per user encryption is necessary for confidential personal data.

Disadvantages:

1. Network bandwidth can be a bottle-neck.
2. Backing up directly to a cloud can be very costly.

2.Server Deduplication with Encrypted Data for cloud storage

Deduplication unfortunately come with a high cost in terms of new security and privacy challenges.

Disadvantages:

1. Does not impact the overall storage and computational cost.
2. A system achieves confidentiality and enables block level deduplication at the same time.

3. Proofs Of Ownership in Remote storage System

Present solution based on Merkle Trees and Specific encoding we identify attacks that exploit client side deduplication attempts to identify reduplication .

Disadvantages:

1. It is impossible to verify experimentally the assumption about the input distribution.

2. Implemented a prototype of the new protocol and ran it to evaluate performance and asses the PoW scheme benefits

4. Weak leakage –resilient client side Deduplication of encrypted data in cloud storage

Propose a secure client –side deduplication scheme .Addressed an important security concern in cross-user client –side deduplication

Disadvantages:

Convergent encryption and custom encryption methods are not semantically secure.

5. Dupless: Server aided Encryption for deduplicated storage

DupLESS work transparently on top of any Storage interface. Provide strong security against External attacks. Resolve the cross user deduplication

Disadvantages

Failed to secure brute force attacks

3. PROBLEM STATEMENT:

The problem is secure deduplication. The rapid adoption of cloud services is accompanied by increasing volumes of data stored at remote cloud servers. Among these remote stored files, most of them are duplicated: according to a recent survey by EMC, 75% of recent digital data is duplicated copies.Unfortunately, this action of deduplication would lead to a number of threats potentially affecting the storage system.

4 EXISTING SYSTEM

The existing data deduplication schemes for primary storage, such as iDedup and Offline-Dedupe are capacity oriented in that they focus on storage capacity savings and only select the large requests to deduplicate and bypass all the small requests. The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved.

Proposed System

We propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic.

Advantages of Proposed System

To improving the performance of primary storage systems and minimizing performance overhead of deduplication,

5. IMPLEMENTATION

Cloud Account creation:

The Login Form module presents site visitors with a form with username and password fields. If the user enters a valid username/password combination they will be granted access to additional resources. The login is categorized into three types

1) Data Owner

2) User

*Data owner can upload the file .He/She has the privileges to delete the uploaded file.

*User can perform keyword search on those files.

Data Deduplication:

Data Deduplication involves finding and removing of duplicate data‟s without considering its fidelity. Here the goal is to store more data‟s with less bandwidth. Files are uploaded to the CSP and only the Data owners can view and download it. Deduplication is a technique where the server stores only a single copy of each file, regardless of how many clients asked to store that file, such that the disk space of cloud servers as well as network bandwidth are saved.. For example, a server telling a client that it need not send the file reveals that some other client has the exact same file, which could be sensitive information in some case

Select Dedupe:

In this module, the uploaded file is encoded using Base64 Encoding, and the file content is stored in the server. Here the duplication check is processed, if the encoded file content and the content already stored in the server matches, then the file is marked as duplicate and it tells the owner that the content already exists in the cloud server.

Encryption & Decryption:

Encryption and decryption provides data confidentiality in deduplication. A user (or data owner) derives a convergent key from the data content and encrypts the data copy with the convergent key. In addition, the user derives a tag for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag correctness property holds, i.e., if two data copies are the same, then their tags are the same

Swap:

Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message. Based on Access Monitor information, the Swap module dynamically adjusts the cache space partition between the index cache and read cache. Moreover, it swaps in/out the cached data from/to the back-end storage **FMSR – AES Algorithm**

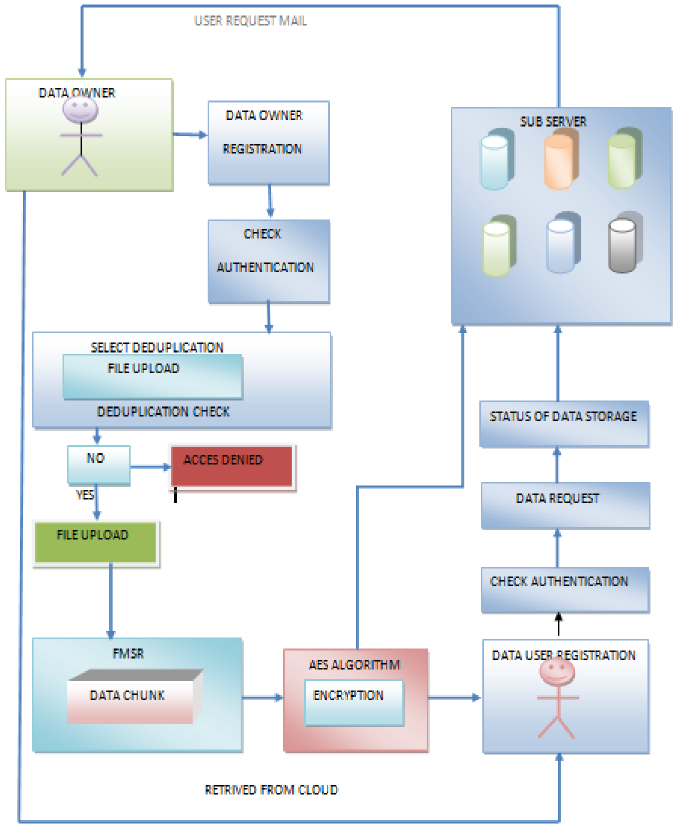
//Pseudo code for File distribution from proxy

```
i.    Get data file from user
ii.   Chunk the file into equal sizes with the
      number same as the multiple storage
iii.  Encode the data with the data from other
      chunks
iv.   Encrypt each chunk with AES-128
      algorithm with the secret key
v.    Distribute the chunks to each of store
      storage
```

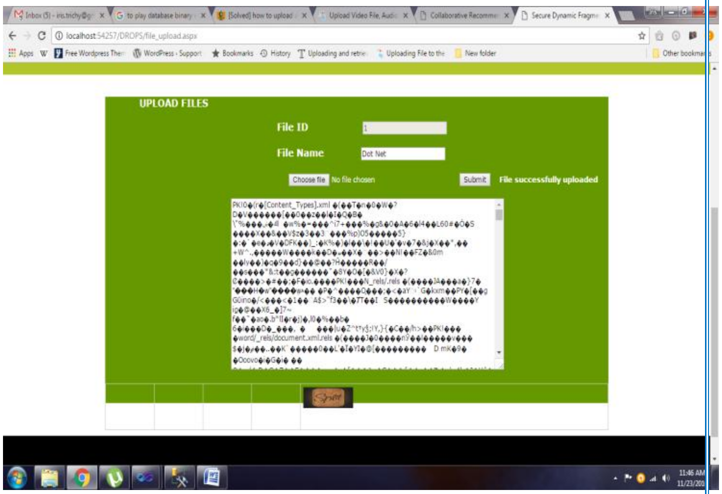
7. SYSTEM MODEL

Select-Dedupe improves the read performance indirectly by reducing the write traffic and avoiding the read amplification problem as much as possible. The significant number of reduced write requests in Select-Dedupe greatly shortens the length of the disk I/O queue and relieves its pressure, thus allowing the read requests to be serviced more quickly. We also plot the normalized storage capacity used by the different schemes. Full-Dedupesaves the largest amount of storage capacity among all the deduplication-based schemes because it deduplicates all redundant write data, which is not the case for iDedup and Select- Dedupe. Select-Dedupe achieves comparable or better capacity savings than the capacity-oriented deduplication scheme iDedup, especially for the mail trace. This is because, while iDedup only deduplicates large I/O requests, Select- deduplicates both large and small I/O requests.

When the small I/O requests become a major part of the total requests, the capacity saving is also increased accordantly.



Screenshots:



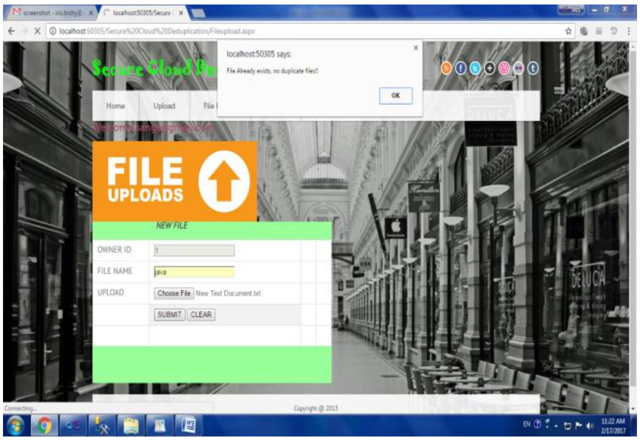
localhost:54257/DROPS/file_upload.aspx

UPLOAD FILES

File ID: 1

File Name: Dot Net

Choose file No file chosen Submit File successfully uploaded



localhost:50305/Secure%20Cloud%20Deduplication/Fileupload.aspx

localhost:50305 says:

File Already exists, no duplicate files!!

OK

NEW FILE

OWNER ID: 1

FILE NAME: java

UPLOAD: Choose File New Text Document.txt

SUBMIT CLEAR

8. CONCLUSION AND FUTURE WORK

The notion of authorized data de-duplication was proposed to protect the data security by including differential privileges of users in the duplication check. Also presented several new de-duplication constructions supporting authorized duplicate check in cloud

All Rights Reserved ©2017 International Journal of Computer Science (IJCS Journal) and
Published by SK Research Group of Companies (SKRGC) - Scholarly Peer Reviewed Research Journals

architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that the proposed scheme is secure in terms of insider and outsider attacks specified in the proposed security model.

It is strategic to develop an automatic update mechanism that can identify and update the required fragments only. The aforesaid future work will save the time and resources utilized in downloading, updating, and uploading the file again. Moreover, the implications of TCP in cast over the proposed methodology need to be studied that is relevant to distributed data storage and access.

9. REFERENCES:

1. Choi J, D. Lee, and Sam H. Noh. Caching less for better performance: Balancing cache size and update cost of flash memory cache in hybrid storage systems. In FAST'12, Feb. 2012.

2. Clements A.T, I. Ahmad, M. Vilayannur, and J. Li. Decentralized Deduplication in SAN Cluster File Systems. In USENIX ATC'09, Jun. 2009.

3. Jiang H, S. Wu, Y. Fu, and L. Tian. Read Performance Optimization for Deduplication-based Storage Systems in the Cloud. ACM Transactions on Storage, 10(2):1–22, 2014.

4. Huge Patterson R, G. Gibson, E. Ginting, D. Stodolsky, and J. Zelenka. Informed prefetching and caching. In SOSP'95, Dec. 1995.

5. Kaiser J, A. Brinkmann, T. Cortes, M. Kuhn, and J. Kunkel. A Study on Data Deduplication in HPC Storage Systems. In SC'12, Nov.2012.

6. Koller R and R. Rangaswami. I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance. In FAST'10, pages 1–14, Feb. 2010.

7. Lofstead J, M. Polte, G. Gibson, S. Klasky, K. Schwan, R. Oldfield, M. Wolf, and Q. Liu. Six Degrees of Scientific Data: Reading Patterns for Extreme Scale Science IO. In HPDC'11, Jun. 2011.

8. Savage Sand J. Wilkes. AFRAID: A Frequently Redundant Array of Independent Disks. In USENIX ATC'96, Jan. 1996.

9. Shilane P, F. Douglis, H. Shim, S. Smaldone, and G. Wallace. Nitro: A Capacity-Optimized SSD Cache for Primary Storage. In USENIX'14, Jun. 2014.

10. Srinivasan K, T. Bisson, G. Goodson, and K. Voruganti. iDedup: Latency-aware, Inline Data Deduplication for Primary Storage. In FAST'12, Feb. 2012.