IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Client Server Implementation

Prerna S. Sharma
Jawaharlal Darda Institute of Engineering and Technology, Yavatmal, India

Abstract: *Recovery and Backup system in which the process involves that copying and archiving of data on different cloud server, so that this data is used to recover the unique data, afterward a loss event. Purpose of backup is to recover data after its loss and to improve data from a past time. In the existing system, data replication has been used as the ultimate solution to improve data availability and reduce access time. This systems usually need to migrate and create a large number of data replicas over time between and within data centers, incurring a large overhead in terms of network load and availability. In proposed systems, the fragments of every data file are physically distributed over multiple servers, which increase privacy of data files. We can use deduplication compressions techniques, but there is a need to improve such systems with respect to capacity of storage and to improve recovery of files. In this paper, we implement the erasure code, inline de-duplication checking and file restoring to improve the performance of backup system.*

Keywords: Recovery and Backup

I. INTRODUCTION

Cloud technology gives the different kind of benefits in our day to day use. Cloud computing is a technology also called as 'on-request computing'. We can access it from anywhere whenever we needed it. Now a days every user wants technology as soon as it needed. In cloud technology third-party data centers stores and processes users data and Cloud computing allows with their storage solutions provided to users and enterprises with various aptitudes. Cloud computing technology shares resources to conquer consistency and markets of scale on the value style costing.

Cloud computing is a very broad concept covered with huge organization and multiple shared services. Cloud computing is an archetype for enabling appropriate, on-request system access to a shared configurable computer resources such as services, storage networks, servers, applications etc. Resources are promptly examined and released with minimal management effort. Now a days cloud computing is a vastly claimed service or utility because of the benefits of high computing power, lower service cost, high performance, scalability, accessibility and availability. Every year cloud has evolution rate around 50-55% experienced by vendors. There are some difficulties in the stage of beginning which need proper consideration to create cloud services more reliable and user friendly. Information deduplication is a powerful method in the cloud reinforcement and also it is gaining applications to decrease the reinforcement window to enhance the storage system room effectiveness and system transmission capacity usage. New study uncover that balance to high information excess in VM (Virtual Machine) undertaking and High-Performance Computing (HPC) capacity frameworks. These studies utilized on the information deduplication innovation to large scale information sets accomplished a normal memory space saving of 30% with up to 90% in Virtual memory and 70% in High Performance computing capacity frameworks. For instance, minimization of the time factor for the live VM relocation in the cloud can be achieved by receiving the information deduplication.

The current information deduplication plans for essential data storage for example, inline deduplication and Offline deduplication. Offline-Dedupe and iDedup are size based such that they concentrate on size of reserve funds and choose only the substantial request to deduplicate and sidestep for small demands. The little I/O asks for record for a modest fraction of the capacity limit. It is necessary to make deduplication unprofitable and probably counterproductive since the deduplication is overhead. Previous workload studies have noticed that minor records guidelines in essential storage frameworks (over half) and are at the cause of the framework. Cache Knowledge execution bottleneck is used to mitigate fragmentation and besides because of the cradle impact on essential stockpiling workloads show evident I/O burstiness. Considering performance, the current information deduplication plans disregard to consider workload qualities in essential storage frameworks. There is chance to face most imperative issues in essential storage of performance.

DOI: 10.48175/568
ISSN 2581-9429 IJARSCT

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Erasure Coding has been widely used to provide high availability and reliability while introducing low storage overhead in storage systems. Erasure coding is a process of data protection from being corrupting and misplacing during processing in different applications. This method protects data from fragmented into segments, extended, encoded and avoid redundant data bits and stored at different locations or storage media system. The goal of erasure coding is to reconstruct corrupted data by using information that can be metadata about the data stored at in another array in the disk storage process. Erasure coding is beneficial with huge amounts of data and any applications or schemes. System should be tolerating failures, like data grids, disk array, distributed storage applications and archival storage, object storage. Currently erasure coding is used in object-based cloud storage.

II. LITERATURE REVIEW

In this paper , we propose CRANE, an Efficient Replica Migration scheme for distributed cloud Storage system Ems. CRANE complements any replica placement algorithm by efficiently managing replica creation in geo-distributed infrastructures in order to minimize the time needed to copy the data to the new replica location, avoid network congestion, and ensure the minimum desired availability for the data. Author show that CRANE provides a sub-optimal solution for the replica migration problem with lower computational complexity than its integer linear program formulation.

Chunk-level deduplication, while robust in removing duplicate chunks, introduces chunk fragmentation which decreases restore performance. Rewriting algorithms are proposed to reduce the chunk fragmentation and accelerate the restore speed. Delta compression can remove redundant data between non-duplicate but similar chunks which cannot be eliminated by chunk-level deduplication. Some applications use delta compression as a complement for chunk-level deduplication to attain extra space and bandwidth savings. However, we observe that delta compression introduces a new type of chunk fragmentation stemming from delta compressed chunks whose base chunks are fragmented. We refer to such delta compressed chunks as base-fragmented chunks. We found that this new type of chunk fragmentation has a more severely impact on the restore performance than the chunk fragmentation introduced by chunk-level deduplication and cannot be reduced by existing rewriting algorithms. In order to address the problem due to the base-fragmented chunks, author in propose SDC, a scheme that selectively performs delta compression after chunk-level deduplication.

Secure data deduplication can significantly reduce the communication and storage overheads in cloud storage services, and has potential applications in our big data-driven society. Existing data deduplication schemes are generally designed to either resist brute-force attacks or ensure the efficiency and data availability, but not both conditions. We are also not aware of any existing scheme that achieves accountability, in the sense of reducing duplicate information disclosure (e.g., to determine whether plaintexts of two encrypted messages are identical).

In this paper [3], author investigate a three-tier cross-domain architecture, and propose an efficient and privacy-preserving big data deduplication in cloud storage. EPCDD achieves both privacy-preserving and data availability, and resists brute-force attacks. In addition, we take accountability into consideration to offer better privacy assurances than existing schemes. We then demonstrate that EPCDD outperforms existing competing schemes, in terms of computation, communication and storage overheads. In addition, the time complexity of duplicate search in EPCDD is logarithmic.

Privacy has become a considerable issue when the applications of big data are dramatically growing in cloud computing. The benefits of the implementation for these emerging technologies have improved or changed service models and improve application performances in various perspectives. However, the remarkably growing volume of data sizes has also resulted in many challenges in practice. The execution time of the data encryption is one of the serious issues during the data processing and transmissions. Many current applications abandon data encryptions in order to reach an adoptive performance level companioning with privacy concerns. In this paper, author concentrate on privacy and propose a novel data encryption approach, which is called Dynamic Data Encryption Strategy
(D2ES). Our proposed approach aims to selectively encrypt data and use privacy classification methods under timing constraints. This approach is designed to maximize the privacy protection scope by using a selective encryption strategy within the required execution time requirements. The performance of D2ES has been evaluated in our experiments, which provides the proof of the privacy enhancement.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

In backup systems, the chunks of each backup are physically scattered after deduplication, which causes a challenging fragmentation problem. The fragmentation comes into sparse and out-of-order containers. The sparse container decreases restore performance and garbage collection efficiency, while the out-of-order container decreases restore performance if the restore cache is small. In order to reduce the fragmentation, Author in proposesHistoryAware Rewriting algorithm (HAR) and Cache-Aware Filter (CAF). HAR exploits historical information in backup systems to accurately identify and reduce sparse containers, and CAF exploits restore cache knowledge to identify the out-of-order containers that hurt restore performance. (13)

To protect the privacy of sensitive private data which is association with deduplication, the convergent encryption method has been proposed in reference for encryption of the data which is previously outsourced. Author makes the very first effort to properly report the problem of authorized data deduplication. Varied from old deduplication systems, the distinct privileges of users are further added in duplicate check also the backup data itself. This paper also presents some new deduplication methods which are auxiliary authorized duplicate check in hybrid cloud structure.

In to improve the effectiveness of traditional compressors in modern storage systems, author proposed Migratory Compression (MC), a coarse-grained data transformation. Relocation of similar data chunks gathers into one unit, to improve compression factors. After decompression, migrated chunks return to their previous locations. While compressing single files, MC paired with a typical compressor such as gzip or 7z provides a clear improvement. Proposed scheme improves compression effectiveness by 11 percent to 105 percent, compared to traditional compressors. Deduplication process is used for eliminating duplicates in data, thus improving the effective capacity of storage systems. Single-node raw capacity is still mostly limited to tens or a few hundreds of terabytes, forcing users to resort to complex.

In author proposed Cloud iDedup: History aware In-line Deduplication for Cloud Storage to Reduce Fragmentation by Utilizing Cache Knowledge new mechanisms called progressive sampled indexing and grouped mark and sweep, to address dedupe challenges and also to improve single-node scalability. Progressive sampled indexing removes scalability limitations by using indexing technique. Advantages of proposed scheme are, improves scalability, provide good deduplication efficiency and improvement in throughput.

In author proposed three fold approaches, first they discuss sanitization requirements in the context of de-duplicated storage, second implemented a memory efficient technique for managing data based on perfect hashing, and third they design sanitizing de-duplicated storage for EMC data domain. Proposed approach minimizes memory and I/O requirements.

Perfect hashing requires a static fingerprint space, which conflicts with proposed scheme desire to support host writes during sanitization Data deduplication has recently gine importance in most secondary storage and even in some primary storage for the storage .

A read performance of the deduplication storage has been gaining great significance.

In author introduced called Chunk Fragmentation Level (CFL) as an indicator for read performance. Proposed approach is very effective to indicate the read performance of a data stream with deduplication storage.

In paper proposed the context based rewriting algorithm which rewrites selected few duplicates. Proposed algorithm improves restore speed of the latest backups, while resulting in fragmentation increased for older backups, which are rarely used for restore. Main drawback proposed algorithm is that it reduces write performance a little bit.

In paper proposed an iDedup, inline deduplication solution for primary workloads. Proposed algorithm is based on two key insights from real world workloads that is spatial locality exists in duplicated primary data and second temporal locality exists in the access patterns of duplicated data. Proposed algorithm minimizing extra Input/Outputs and seeks.

Here in author proposed the context based algorithm for rewriting selected few duplicates. Proposed algorithm improves restore speed of the latest backups and also increased resulting in fragmentation for older backups that are rarely used for restore. Proposed algorithm has a limitation that reduces write performance a little bit.

Here in Author proposed, an iDedup i.e. inline deduplication solution for major workloads. Projected algorithm is roughly based on two keys. A perception from real dynamic world workloads i.e. spatial locality occurs in duplicated primary data and another second temporal locality occurs in the access patterns of duplicated data of backup. Proposed algorithm minimizes extra Input/Outputs and seeks.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

III. SCOPE OF PROJECT

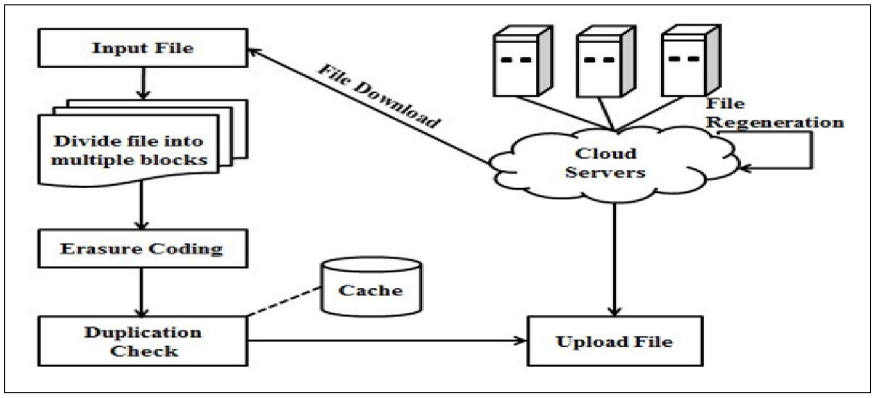


Figure 3.1 System Architecture

Module Description:

1. User Registration-login

When new user enters the system, needs to be successfully registered on cloud server for authentication. For this registration, user should provide username and password. After successful login and authentication, cloud server provides access to users to use facilities provided by server.

2. Input File Fragmentation

After successful login, user wants to store their file on cloud server in distributed manner. For this purpose, input files are subdivided into several fragments. These fragments are store on multiple servers on cloud storage, which are accomplished by single main cloud.

3. Erasure Code

Erasure code is a technique in which data get protected by scattering into fragments, extended and coded (that can be encoded or encrypted) with duplicated data pieces and further stored among the set of diverse locations of storage system. (17)

The basic target of this technique is to permit stored data that becomes corrupted because of any natural or artificial failure at some point in the storage process to be recreated by using metadata which is stored to another place in the stored array.

4. Inline De-duplication Checking

This module check the duplication file storing on server. That is when this module receives fragment to store on server, it check its hash value in cache. If hash is not available in cache, it uploads the fragment on server, otherwise discard that fragment. This module decreases the memory wastage, because only unique files are store on server and removes the duplicated files.

5. File Upload

In this module, user can upload the file on cloud. For this, file is fragmented and store on multiple servers in distributed manner. This process maintains the security and privacy of data stored on multiple servers and reduce the chances of data corruption.

6. File Download

User can also download the files distributed on multiple servers. If anyone fragment of the file is loss or corrupted, can be recovered in file restore module. This corrupted file can be restoring.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

IV. METHODOLOGY

4.1 System Overview

In backup environments field deduplication yields major advantages. Deduplication is a process in which duplicate data is automatically discarded from storage system, thereby reducing storage cost. Deduplication results inevitably in data fragmentation, because logically continuous data is spread across many disk locations. Fragmentation mainly caused by duplicates from previous backups of the same backup set, since such duplicates are frequent due to repeated full backups containing a lot of data which is not changed.

Systems with in-line de-duplicate intends to detects duplicates during writing and avoids storing them, such fragmentation causes data from the latest backup being scattered across older backups. Proposed novel method to avoid the reduction in restores performance without reducing write performance and without affecting deduplication effectiveness

4.2 Mathematical Formulation

System (S) is denoted as S= {C, B, U, D, E},
Where C- Client,
B- Data-Blocks,
U- Client (User),
D- Deduplication,
E- Erasure Coding. (19)

1) Client:

C= {c1, c2, c3,....., cn}
Here, C is the set of client and c1, c2, c3,....., cn are the number of clients.

2) Input Files:

I= {i1, i2,…, in}
Here, I is the set of all input file, which will be store on cloud server.

3) Divide Files into Blocks (Chunking):

B= {b1, b2, b3, ...,bn}
Here, B is the set of block and b1, b2, b3, ...,bn are the number of blocks.
B = $File\ \ Size/4$
Before storing the files on cloud server, Files are divided into fixed size of chunks,
F= {FC1, FC2, FC3, FC4}
Where, F is the file and, FC is the chunk of that file.

4) Apply Erasure code:

E= {E1, E2, …, En}
Here, E is the erasure code of all file blocks. Encoder takes 'B' data blocks with symbols of 'S' bits respectively. Shards symbol get added with 'B' which generates the code-word. Shards also called as parity. Encode (B)-B are shards symbols for each S bits. (20)
A Reed-Solomon decodes the encoded data
Decode (Encode (B))' that have errors in a code-word,
(Decode (Encode (B)) = Encode (B)-B
Where, (Decode (Encode (B))) = Shard,
B= Data blocks,
Encode (B) = encoded data.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

5) Hash Generation:

Before storing the file on cloud server, hash of each chunk is generated using SHA-1 Algorithm.

SH= {SH1, SH2, SH3, SH4}

Where SH be the set of Hash and {SH1, SH2, SH3, SH4} be the hash of each Chunk {FC1, FC2, FC3, FC4} respectively.

6) Inline Duplication Check:

Duplication check for each Chunk (Block):

dbHash = getting files hash from the client side database,

FileHash= Users chunked file's hash,

SH(New chunk) = h

SH'(Old chunks) = h' Compare h and h'

h != h'. (21)

7) Upload File on Cloud Server:

if h != h'

Then File is not duplicate then store on cloud server.

Otherwise reject file to store.

8) Download and Decode File:

Restoring Files from cloud server.

F' $= \sum 41\ FC$

Where, F'= Restored File,

FC= File Chunk.

4.3 History-Aware Rewriting Algorithm

Input: Values of Container_H sparse containers,

C_Historical,

Output: Values of Container_C sparse containers,

C_CurrentDataFile,

1. Create Set S: C_CurrentDataFile
2. While (backup is in progress)
3. Get chunks from cache and search hash values in the Hash_Index.
4. If (data chunks are already present) //loop 1
5. If (Container_C (values) = = C_Historical(values)) then //loop 2
6. Update chunks to a new container.
7. Else discard chunks.
8. End if //loop 2
9. Else store chunks in a new container.
10. End if //loop 1
11. Update the progress backup record

C_CurrentDataFile.

12. End while
13. Compare progress backup record with threshold
14. Eliminate all exceeding progress backup records from C_CurrentDataFile.
15. Compute the rewrite fraction for the succeeding backup file.
16. While (rewrite fraction > rewrite threshold)
17. Eliminate largest progress backup records from C_CurrentDataFile.
18. Update the computed rewrite fraction.
19. End while

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

20. Return C_CurrentDataFile.

Use Case Diagram

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors. An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

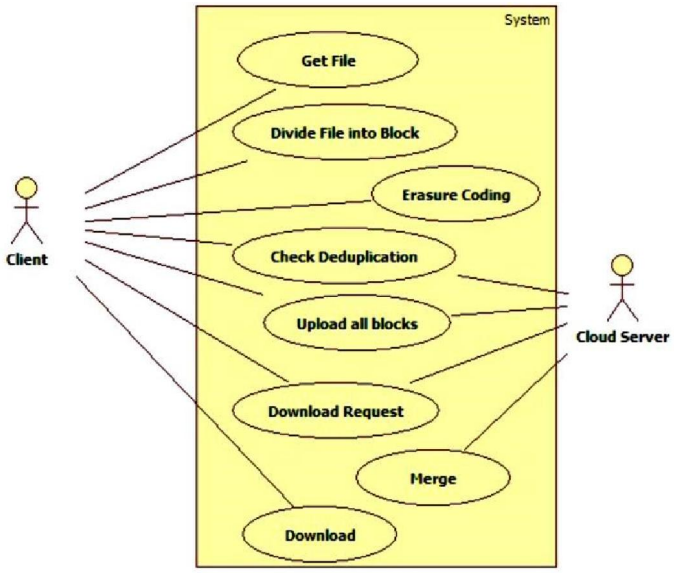
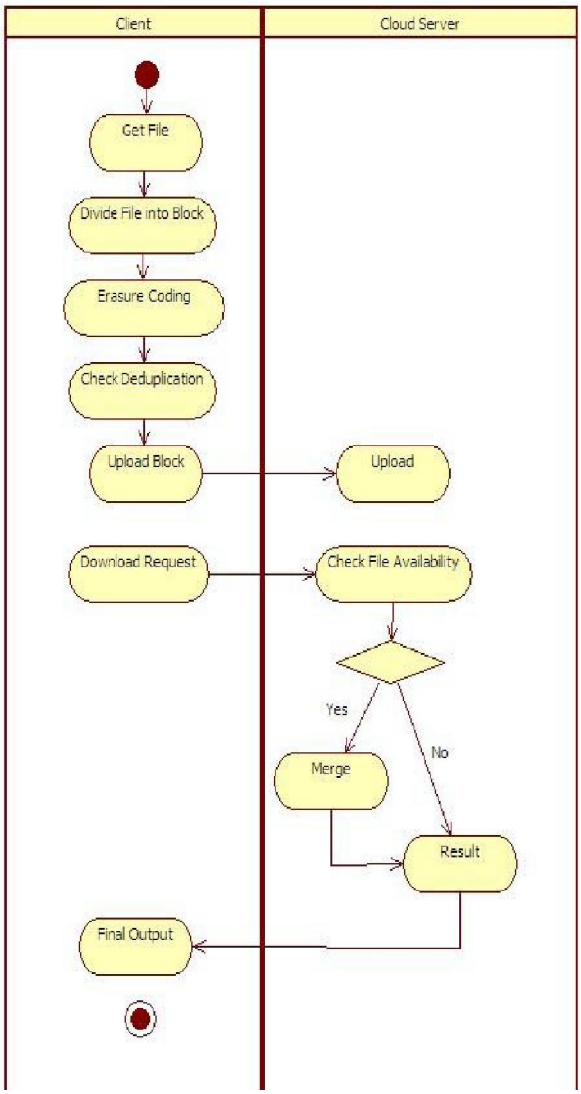


Figure 5.4 Use Case Diagram

Activity Diagram

Activity diagrams describe the work flow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024



VI. RESULT AND APPLICATIONS

The client-server model organizes network traffic using a client application and client devices. Network clients send messages to a server to make requests of it. Servers respond to clients by acting on each request and returning the results. One server supports many clients, and multiple servers can be networked together in a server pool to handle increased processing loads as the number of clients grows.

A client computer and a server computer are two separate units of hardware, each customized for a designed purpose. For example, a web client works best with a large screen display, while a web server doesn't need a display and can be located anywhere in the world. In some cases, however, a given device can function both as a client and a server for the same application. Additionally, a device that is a server for one application can simultaneously act as a client to other servers for different applications.

VII. CONCLUSION

Data backup storage is not only difficult but also challenging task in terms of storage space utilization, recovery, efficiency. With changing technology users have started to backup their data on cloud servers because of flexibility and mobility purpose. Regular data backup process creates redundant data on cloud servers. In backing up data, data blocks

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

get scattered on multiple cloud servers so it reduces chances of data loss from corruption, but at a same time it utilizes more space. For this problem our proposed system implements erasure coding for recovery and inline deduplication for cloud backup storage. Erasure coding encodes the chunked data.

History-Aware Rewriting algorithm (HAR) accurately identifies and rewrites sparse containers via exploiting historical information. We also implement an optimal restore caching scheme (OPT) and propose a hybrid rewriting algorithm as complements of HAR to reduce the negative impacts of out-of-order containers. The ability of HAR to reduce sparse containers facilitates the garbage collection. It is no longer necessary to offline merge sparse containers, which relies on chunk level reference management to identify valid chunks. We propose a ContainerMarker Algorithm (CMA) that identifies valid containers instead of valid chunks. Since the metadata overhead of CMA is bounded by the number of containers, it is more cost-effective than existing reference management approaches whose overhead is bounded by the number of chunks.Evaluation of proposed system is done by experiment of different dataset having multiple files. Result shows that this system utilizes less space thus minimize cost and recovery from data loss.

Recommendations for Future Work

This system studies reveal that information excess displays a much more elevated amount of force on the I/O way than that on plates because of moderately high temporal access region connected with little I/O requests to redundant information. Also, specifically applying information deduplication to essential storage frameworks in the Cloud will probably bring about space contention in memory and information fragmentation on disks. Based on these perceptions, propose an execution arranged I/O deduplication, called POD, as opposed to a capacity oriented I/O deduplication, exemplified by iDedup, to enhance the I/O execution of essential storage frameworks in the Cloud without giving up saving capacity.

REFERENCES AND BIBLIOGRAPHY

[1] Amina Mseddi, Mohammad A. Salahuddin, Mohamed FatenZhani, Halima Elbiaze, Roch H. Glitho, "Efficient Replica Migration Scheme for Distributed Cloud Storage Systems", IEEE 2018.

[2] Yucheng Zhang, Dan Feng, Yu Hua, Yuchong Hu, Wen Xia, Min Fu, Xiaolan Tang, Zhikun Wang, "Reducing Chunk Fragmentation for In-Line Delta Compressed and Deduplicated Backup Systems," 2017 International Conference on Networking, Architecture, and Storage (NAS), Shenzhen, 2017, pp. 1-10.

[3] X. Yang, R. Lu, K. R. Choo, F. Yin and X. Tang, "Achieving Efficient and PrivacyPreserving Cross-Domain Big Data Deduplication in Cloud," in IEEE Transactions on Big Data, 2017. doi: 10.1109/TBDATA.2017.2721444

[4] K. Gai, M. Qiu and H. Zhao, "Privacy-Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing," in IEEE Transactions on Big Data, 2017

[5] Fu, Min, et al. "Reducing Fragmentation for In-line Deduplication Backup Storage via Exploiting Backup History and Cache Knowledge." IEEE Transactions on Parallel and Distributed Systems 27.3 (2016): 855-868

[6] Li, Jin, et al. "A hybrid cloud approach for secure authorized deduplication."IEEE Transactions on Parallel and Distributed Systems 26.5 (2015): 1206-1216.

[7] X. Lin, G. Lu, F. Douglis, P. Shilane, and G.Wallace, “Migratory compression: Coarse-grained data reordering to improve compressibility,” in Proc. USENIX FAST, 2014.

[8] F. Guo and P. Efstathopoulos, “Building a highperformance deduplication system,” in Proc. USENIX ATC, 2011.

[9] F. C. Botelho, P. Shilane, N. Garg, andW. Hsu, “Memory efficient sanitization of a deduplicated storage system,” in Proc. USENIX FAST, 2013.

[10] Y. Nam, G. Lu, N. Park, W. Xiao, and D. H. Du, “Chunk fragmentation level: An effective indicator for read performance degradation in deduplication storage,” in Proc. IEEE HPCC, 2011.

[11] M. Kaczmarczyk, M. Barczynski, W. Kilian, and C. Dubnicki, “Reducing impact of data fragmentation caused by in-line deduplication,” in Proc. ACM SYSTOR, 2012. (31)

[12] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti, “iDedup: Latency-aware, inline data deduplication for primary storage,” in Proc. USENIX FAST, 2012.