IJRSET AUGUST 2021 Volume 8, Issue 8 Pages: 22-29

ISSN 2394-739X

International Journal for Research in Science Engineering and Technology

ANALYSIS OF DIFFERENT DATA DEDUPLICATION ALGORITHM

[1] Dr. A. Nithya Rani,
[1] Associate Professor,
[1] Department of Computer Science,
[1] CMS College of Science and Commerce,
[1] Coimbatore.

ABSTRACT - Deduplication has turned into a generally sent technology in cloud data focuses to further develop IT resources effectiveness. Nonetheless, customary strategies face an incredible test in big data deduplication to strike a reasonable tradeoff between the various objectives of versatile deduplication throughput and a high copy end proportion. An application-mindful adaptable inline circulated deduplication framework in a cloud climate, to address this difficulty by taking advantage of application mindfulness, data likeness and region to streamline disseminated deduplication with between hub two-layered data steering and intra-hub application-mindful deduplication. Here various data deduplication algorithms are broke down and contrasted and their performance and discover which algorithm creates elite and low dormancy. Testing results show that the technique can beat a couple of best in class strategies for computational performance and security levels.

Keywords: [data, deduplication, cloud storage, SCDS, LSRA, CSP.]

1. INTRODUCTION

Lately, an ever increasing number of people and organizations decide to store data in cloud workers rather than local storage frameworks, which not just mitigates the expense pressure brought about by local data storage and maintenance yet additionally wipes out the requirement for complex organizations and backing of programming and equipment frameworks. In the cloud storage framework, the client can utilize and handle the data on the cloud worker whenever, however the client fails to keep a grip on the data put away on the remote node. The data transferred by clients on the cloud includes private information. At the point when client uploads data on the cloud it loses its power over the data which chances private data spillage by Cloud Service Provider (inquisitive). To further develop storage utilization CSP perform data deduplication, the adversary might utilize relative assaults to take client's private information. With the expansion of data transferred on cloud decrease of management, consumption is exceptionally urgent alongside further developing storage utilization. In this way, further developing storage proficiency and decreasing management use is a quick issue to figure out.

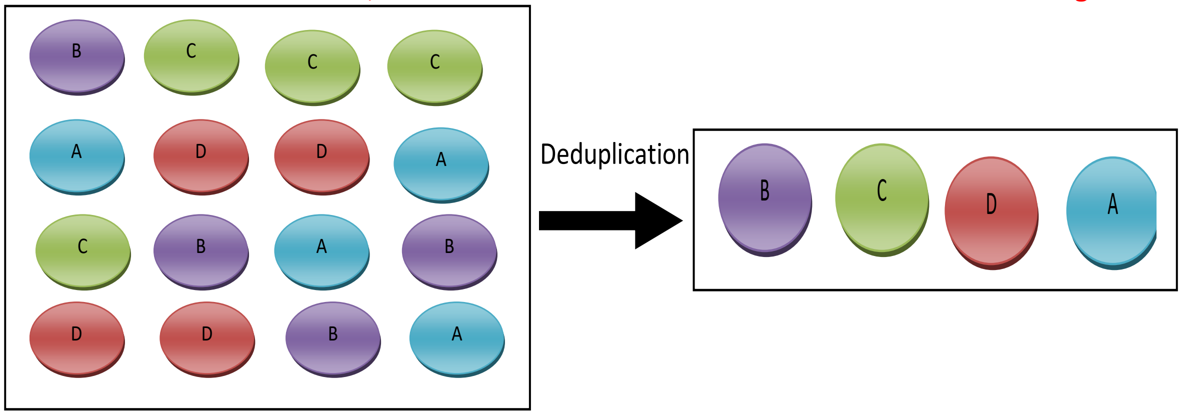

Figure 1. Data deduplication process

- The main objectives of this scheme are as follows:
- This framework forestalls transmitting of privacy data by performing approved data deduplication.
- The framework utilizes a joined encryption algorithm to secure data classification, uses the job re-encryption algorithm for approved access as it were.
- Management focus is acquainted with manage client asks for and produce the job re-encryption key which permits a specific client with that key to access a specific file.

Categorization of data deduplication approaches

Primary storage technique: In this methodology, data deduplication is done at primary memory or fundamental memory under the influence focal handling framework. The performance of this methodology might be diminished when a lot of data is stacked into the primary memory [22]. It is a decent methodology for private data and effectively performs primary exercises. Model: primary responsibilities and mail servers.

Secondary Storage technique: The deduplication cycle is occurred at a reinforcement worker or auxiliary memory. It stores the significant primary data in helper memory to stay away from the deficiency of data because of catastrophic events. It isn't constrained by the CPU. As data are put away in a remote gadget, this methodology performs quicker data deduplication. This takes out totally duplicated data and gives preferable performance over the primary data deduplication [2]. This strategy dumps and loads old data proficiently. Model: files and file backups.

Source deduplication: Data Redundancy is done at the customer or source terminal before data are sending to the reinforcement worker or target terminal. It requires less storage space and equipment contrasting with the objective deduplication. Hence the measure of memory space, network move rate, and season of putting away data in the cloud is at last decreased. For little datasets, this methodology is useful for deduplication.

Multinode or global deduplication: This methodology performs data deduplication in multi nodes in the disseminated storage framework. It totally eliminates all redundant data in multi-servers in a disseminated climate.

Inline deduplication approach: Redundant data is taken out at the source terminal or prior to putting away data into the plate. It doesn't need additional storage spaces or a circle. This methodology is an adaptable and proficient strategy along these lines it executes the data all at once as it were. The computation time is high.

2. LITERATURE SURVEY

1. A. Miri and F. Rashid (2019) et.al proposed Secure Textual Data Deduplication Scheme Based on Data Encoding and Compression. As the requirement for storage has filled dramatically lately, cloud storage has been giving an answer for this need by giving users extended limit and access. Giving satisfactory security and privacy, and bringing down storage costs are a portion of the key difficulties confronting this arrangement. A typical practice utilized by cloud service providers (CSPs). The security of the data against the semi-honest CSP and noxious users is guaranteed by utilizing Burrows Wheel Transform encoding plan. The encoded data is additionally packed to acquire compelling reserve funds as far as storage and diminished size of the data. Data encoding and data compression - to give proficient deduplication that is secure against foes, including conceivable semi-honest CSPs. Our proposed framework architecture is particular in its construction, as in the implementation of any of its modules should conceivably be possible by utilizing a wide scope of security algorithms intended to accomplish explicit designated objectives relying on the privacy and effectiveness requirement of the arrangement. This would yield tremendous cloud space investment funds for the CSP and thusly the expense reserve funds for the users.

2. M. Oh et al (2018) et.al proposed Design of Global Data Deduplication for a Scale-Out Distributed Storage System. Scale-out disseminated storage systems can maintain adjusted data development as far as limit and performance on an on-request premise. Be that as it may, it is a test to store and oversee enormous arrangements of substance being produced by the blast of data. One of the promising answers for alleviate big data issues is data deduplication, which eliminates redundant data across numerous nodes of the storage system. Present a worldwide deduplication plan for the current common nothing scale-out storage system, which can be joined with existing storage components like high accessibility, data recuperation while minimizing performance degradation.

3. S. Jiang, T. Jiang and L. Wang (2020) et.al proposed Secure and Efficient Cloud Data Deduplication with Ownership Management. In powerful possession management [9], to oppose a toxin assault on label consistency, the substantial data proprietor ought to download the relating ciphertext to ascertain the randomized convergent key and to really look at the tag, since the cloud worker will be unable to confirm consistency between the cipher text and the comparing plaintext for client side randomized convergent encryption (RCE). Take on the sluggish update methodology to diminish the update recurrence and computation overhead for cross-client file-level deduplication while ensuring the forward and backward secrecy of the resulting uploaded. Moreover, we utilize the client supported CE to accomplish inside-client block-level deduplication. The PoW confirmation cycle to accomplish shared PoW check while saving the bandwidth for the data deduplication.

4. D. Zheng, (2021) et.al proposed Dynamic data compression algorithm for wireless sensor networks based on grid deduplication. A unique data compression technique dependent on matrix deduplication is proposed. Framework based sensor node spatial situating and big data combination techniques are embraced to acknowledge dynamic component mining of wireless sensor network data, extricate highlight arrangement points of wireless sensor network data, recreate wireless sensor network data include space by taking on spatial matrix node recombination, fabricate a factual recognition model of dynamic element mining of wireless sensor network data by consolidating lattice region gathering compression strategy, and acknowledge implanted fluffy control and joint element disseminated versatile learning. Dynamic data compression strategy for wireless sensor networks dependent on matrix de-duplication. Consolidating with the custom data handling rationale structure plan technique, the component grouping and sensor node sending model of wireless sensor network data are acknowledged, and the spatial matrix node revamping is taken on to recreate the element space of wireless sensor network data

Analysis of different deduplication Method

Bloom filter algorithm

Bloom filter (hereinafter alluded to as Bf) which has a more productive construction in space is made out of a bunch of hash planning capacities and a word size. We frequently utilize a Bf data design to address its Eigen values. Lessening the dimensionality of file highlights is its center thought. Contrasted and general algorithms, the Bf algorithm has higher spatial effectiveness and more limited question time and enjoys critical benefits in handling big data. Numerous data set trials to show that Bf enjoys the accompanying benefits when applied to likeness detection.

1. It can be applied to remote replica system: In remote replica system, metadata overhead is usually very small.
2. It has the ability to pair quickly: Bf makes rapid matching possible with Bitwise AND operation.
3. It can scientifically determine inclusion relation: Bf is a set of complete tables.
4. It can significantly reduce metadata overhead: It can reduce metadata overhead, and can be combined with sliding windows and other technologies.

Similarity clustering-based deduplication (SCDS)

Similarity clustering based deduplication strategy (named SCDS), which means to erase more copy data without fundamentally expanding system overhead. The fundamental thought of SCDS is to limit the question scope of fingerprint list by data dividing and similarity clustering algorithms. In the data preprocessing stage, SCDS utilizes a data apportioning algorithm to group comparable data together. In the first place, the files are arranged by their own attributes, partitioned into chunks to compute their fingerprints. The fingerprint collections of the little file are blended, and the fingerprint collections of the huge file are isolated so that each fingerprint collection contains roughly similar number of fingerprints. Then, at that point, it chooses RFs for each file fingerprint set.

Algorithm SCDS

```
1.  Input: All the files and their full name, all cluster tables the cluster threshold $\delta$
2.  Classification by the full name or header information
3.  Block the files, calculate the chunk fingerprints, get the file fingerprint collection and
    upload them as the fingerprint stream to the memory deduplication
4.  While file finger print set $\neq \emptyset$ do
5.  Select RF For the Fingerprint set;
6.  Calculate the similarity with corresponding cluster center according to the corresponding
    type of table. Get the maximum similarity cluster and the similarity;
7.  If $R_i > \delta$
8.  While fingerprint detection is not finished in the set do
9.  For each fingerprint, detects its duplicate in the   corresponding cluster;
10. If fingerprint is duplicate in the set then
11. Fingerprint frequency is increased by one;
12. Remove the fingerprint from the set;
13. Else
14. Add the fingerprint to the cluster;
15. End if
16. Add RFs to the corresponding cluster center;
```

```
17. End while
18. Else
19. Add RFs to the corresponding cluster-table as new cluster center;
20. End if
21. Put the fingerprint set in the corresponding new cluster;
22. End while
```

Second, for each fingerprint set, SCDS calculates its similarity to the group in the comparing type, as displayed in Algorithm 2. Third, we set a limit. On the off chance that its similarity to all groups is beneath the limit, we save every one of the fingerprints in the bunch and store its RFs in the bunch table as another bunch community. Else, we select the most elevated similarity bunch for deduplication. In the wake of erasing the redundant fingerprints, we embed the new fingerprints into the bunch and supplement the RFs of the collection into the group place.

Algorithm clustering similarity

```
1. Input : fingerprint set s, the corresponding file type in the storage node cluster center list
   ( assuming there are l cluster centers), RFs
2. Output : Duplicated target cluster ID i, similarity $R_i$
3. Calculate the similarity between the RFs to cluster centers according to the file type, and
   get $r_1, r_2, \ldots r_l$;(see Equation(3)
4. Calculate the number of repeated fingerprints b of each cluster and finger print set s using
   cardinality estimation, denoted as $b_1, b_2, \ldots b_l$;
5. Select ID I to satisfy $R_i = \max \text{Equation}(6)$ as the target cluster;
6. Return the target cluster ID I, similarity$R_i$.
```

Local similarity routing algorithm (LSRA)

Algorithm Local similarity routing algorithm.

```
1. Input :
      a. S: The superblock to be routed.
2. Output:
      a. Id : The same data server selected.
3. Split superblock s into k segments;
4. fingerprint$f_i$ ← min hash of segment i, $(0 \le i \le k)$;
5. respective fingerprint set $s_f \leftarrow \{f_0, f_1, \ldots, f_{k-1}\}$;
6. For each data server$D_j$ Do
7. Connect data server $D_j$ through RPC
8. $c_j \leftarrow$ hits_from_similarity_index($D_j$, $S_f$ ) ;
9. end for
10. hit count set$C \leftarrow \{c_0, c_1, \ldots c_{n-1}\}$ ;
11. if $\forall c_j \epsilon C, c_j = 0$ then
12. id ← min _used_node();
13. Return id
14. Else
15. For each data server $D_j$ do
16. $u_i \leftarrow$ storage_usage($D_j$);
17. $u_j \leftarrow$ compute_value($D_j, c_j, u_j$);
18. End for
```

```
19. Value set V ← $\{u_0, u_1, \dots, u_{n-1}\}$;
20. id← m, **if** $v_m$ is max(V) ;
21. Return id;
22. End if
```

The local similarity routing algorithm is based on data locality and data similarity, we take superblock as routing granularity and rethink superblock's association to satisfy the needs of the performance of big data's storage in cloud storage. In our implementation, we do rationale segment on superblock, select comparable trademark fingerprint locally, and get the data circulation of every node to choose the best deduplication node as indicated by a stateful data routing algorithm. In the interim, to keep up with the equilibrium of system storage, we design the routing reference worth of every node as indicated by the node's present storage status, and the best routing address is dictated by the worth of this reference esteem.

3. EXPRIMENT RESULT

Comparing the algorithm Deduplication rate

File size (kb)	Bloom filter	SCDS	LSRA
8	15	21	27
16	12	18	26
32	14	24	29
64	19	26	31

Table 1 Comparison Table of Deduplication Rate

Comparing the three algorithm deduplication rate was show the above table that was different values on different file size inputs. Deduplication percentage is nothing but which algorithm identifying more duplication data on given file. LSRA was found more deduplication in all files.

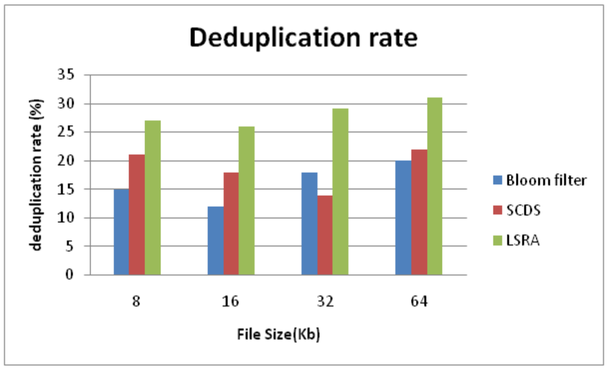

Figure 2 Comparison chart of Deduplication Rate

The above chart shows the three different algorithms of data deduplication rate in percentage. Every strategy was found deduplication separately on similar data it gives different qualities. Be that as it may, the LSRA algorithm tracks down the maximum data deduplication percentage.

Read Latency Accuracy

No. of files	Bloom filter	SCDS	LSRA
5	45.12	65.01	31.35
10	44.23	62.54	30.29
15	50.49	55.20	28.36
20	48.17	57.42	27.27

Table 2 Comparison Table of Read Latency Accuracy

Table 2 shows the comparison of read latency accuracy for three algorithms. Which algorithm was giving low latency is the best for researchers. LSRA was giving low latency in this analysis.

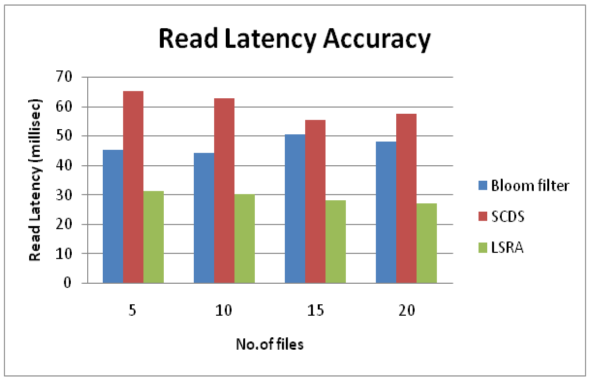

Figure 3 Comparison chart of Read Latency Accuracy

The above Chart represents the Comparison of three algorithm Read latency Accuracy. All algorithms read same file with different time latency. The lowest latency is the LSRA algorithm. Other two algorithms was the more or less same latency but LSRA was more different from that algorithms.

CONCLUSION

The deduplication interaction has been done by eliminating excess data which are gathered from different sources to the cloud storage. There might be some arrangement of data is produced by true items that are taken care of cautiously with no accident in the data. This paper investigates the different data deduplication algorithms with their presentation. Those algorithms are giving the best outcome however LSRA is giving better outcomes among them.

REFERENCES

[1]. Y. Fu, N. Xiao, H. Jiang, G. Hu and W. Chen, "Application-Aware Big Data Deduplication in Cloud Environment," in IEEE Transactions on Cloud Computing, vol. 7, no. 4, pp. 921-934, 1 Oct.-Dec. 2019, doi: 10.1109/TCC.2017.2710043.
[2]. Q. Hu, "A Data Integrity Verification Scheme of Deduplication for Cloud Ciphertexts," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 865-869, doi: 10.1109/ITAIC49862.2020.9339001.
[3]. M. B. Waghmare and S. V. Padwekar, "Survey on techniques for Authorized Deduplication of Encrypted data in Cloud," 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1-5, doi: 10.1109/ICCCI48352.2020.9104184.
[4]. A. Miri and F. Rashid, "Secure Textual Data Deduplication Scheme Based on Data Encoding and Compression," 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2019, pp. 0207-0211, doi: 10.1109/IEMCON.2019.8936222.
[5]. M. Oh et al., "Design of Global Data Deduplication for a Scale-Out Distributed Storage System," 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018, pp. 1063-1073, doi: 10.1109/ICDCS.2018.00106.
[6]. S. Jiang, T. Jiang and L. Wang, "Secure and Efficient Cloud Data Deduplication with Ownership Management," in IEEE Transactions on Services Computing, vol. 13, no. 6, pp. 1152-1165, 1 Nov.-Dec. 2020, doi: 10.1109/TSC.2017.2771280.
[7]. A. Shrivastava and A. Tiwary, "A Big Data Deduplication Using HECC Based Encryption with Modified Hash Value in Cloud," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 484-489, doi: 10.1109/ICCONS.2018.8662984.
[8]. N. Chhabra and M. Bala, "A Comparative Study of Data Deduplication Strategies," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), 2018, pp. 68-72, doi: 10.1109/ICSCCC.2018.8703363.
[9]. D. Zheng, "Dynamic data compression algorithm for wireless sensor networks based on grid deduplication," 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), 2021, pp. 178-182, doi: 10.1109/CISCE52179.2021.9445966.

[10]. S. Long, Z. Li, Z. Liu, Q. Deng, S. Oh and N. Komuro, "A similarity clustering-based deduplication strategy in cloud storage systems," 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), 2020, pp. 35-43, doi: 10.1109/ICPADS51040.2020.00015.

[11]. S. Luo, G. Zhang, C. Wu, S. U. Khan and K. Li, "Boafft: Distributed Deduplication for Big Data Storage in the Cloud," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1199-1211, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2015.2511752.