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 1, August 2024

A Review on Comparative Study on Two Algorithms of Data De-duplication

Priyanka S T[1], Pradeep Nayak[2], Punnyashree K N[3], Ranjitha M[4], Ravi Kumar[5]
Department of Information science and Engineering[1-5]
Alva's Institute of Engineering and Technology, Mijar, Karnataka, India
priyankatotager@gmail.com , pradeep@aiet.org.in,
punnyashreekn@gmail.com, metiranjitha@gmail.com, grk462433@gmail.com

Abstract: *The process of identifying and eliminating duplicate data copies in order to improvespeed and free up storage space is known as deduplication. In deduplication, duplicates are replaced by references to the one instance of each unique piece of information that was originally stored. This is especially helpful for backup systems, which frequently store numerous copies of the same data. It saves expenses, minimizes the quantity of storage required, and enhances data management. Conversely, deduplication improves performanceby reducing footprint sizes and the amount of data processed or sent [1][2].*

Keywords: Deduplication, Data Reduction, Storage Optimization, and Backup Systems

I. INTRODUCTION

Deduplication means the removal of replicated data, such that there remains only one single instance of each unique piece of information. This is realized by identification of duplicate files or data blocks and replacing all copies with references to the original. Put differently, deduplication aids in the simplification of storage systems, especially when large volumes of files need to be transmitted or preserved [3]. This results in overall performance gains and a large reduction in storage needs, which speeds up and lowers the cost of operations. Data is divided into smaller parts (often chunks or segments) by deduplication, and these parts are uniquely identified by use of a cryptographic techniqueknown as hashing. The system compares these identifiers to those that are currently being tracked as new data is ingested.The fresh chunk is referred instead of being kept as a duplicate if a match is discovered. This procedure can be effectively controlled with finer granularity and implemented at many levels, such as file or block [4][5]. Reduced storage expenses, quicker data transfers during backups and restorations, and enhanced system performance are all results of successful deduplication[6]. Because of this, deduplication is an essential technique for contemporary cloud computing settings,assisting businesses in managing their data growth in an inexpensive and sustainable manner [7][8].

II. NEED FOR DEDUPLICATION

1. Storage Efficiency: By eliminating redundant data, deduplication dramatically lowers the amount of storage space required. This maximizes the use of storage resources whilesaving money [4].
2. Improved Performance: When handling less redundant data, systems operate more efficiently. Shorter processing times and less data transfer are the results, which are especially advantageous for backup and recovery operations [5].
3. Cost Reduction: Hardware, maintenance, and energy costs can be minimized by reducing storage capacity, which is particularly important for companies that handlelarge amounts of data [4].
4. Faster Backups and Restorations: Deduplication accelerates these procedures and boosts the effectiveness of data management overall by reducing the quantity of data that needsto be backed up or restored [6].
5. Network Optimization: By lowering the amount of data sent over networks, deduplication increases bandwidth utilization and accelerates data transfers [7].
6. Data Administration and Integrity: By centralizing distinct data instances, deduplication facilitates data management and helps preserve data integrity [8].

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 1, August 2024

III. ALGORITHMS EXPLANATION

Source-based deduplication algorithms:

Before data is transferred to the cloud, source-based deduplication algorithms look for duplicates in the original data source. This finds and eliminates duplicates locally, cutting downon the quantity of data sent across the network [3]. In situations where there is a lot of data coming from several sources, this approach works well [4].

Algorithm:

```
def source_deduplication(data_stream, local_index, global_index):chunk_size = 4096
unique_chunks = []metadata = []
for chunk in chunk_data(data_stream, chunk_size):chunk_hash = hash_function(chunk)
if chunk_hash in local_index or global_index_exists(chunk_hash):metadata.append(get_chunk_reference(chunk_hash))
else:
local_index[chunk_hash] = chunk upload_chunk_to_cloud(chunk) global_index_add(chunk_hash)
metadata.append(get_chunk_reference(chunk_hash))
return metadata
def global_index_exists(chunk_hash):# Implement cloud index check pass
def  upload_chunk_to_cloud(chunk): # Implement chunk upload to cloudpass
def global_index_add(chunk_hash):# Implement global index updatepass
def get_chunk_reference(chunk_hash):# Implement  reference retrieval Pass
```

Working:

1. Data Chunking: Data will be divided into chunks, usually through a fixed-size chunking algorithm.
2. Hashing: Compute for each piece a hash value using cryptographic hash functions like SHA-256.
3. Index Check: The client maintains a local index of chunk hashes
   - For each chunk, the client checks if the hash exists in the local index.
   - If the hash is not found locally, the client checks a global index stored in the cloud.
4. Transmission: If the chunk is unique (the hash is not found in both the local and global indexes), the chunk is transmitted to the cloud, and both the local and global indexes are updated.
   - If the chunk is a duplicate (the hash is found in either index), only a reference to the existingchunk is sent.
5. Metadata Management: Metadata is maintained to know which chunks belong to which files, so thatoriginal data can be restored easily.

2. Target-based deduplication algorithms:

At the backend of the storage system, such cloud storage, target-based deduplication occurs. First, the data is moved to the cloud, where storage servers handle the deduplication process. This method efficiently handles deduplication on cloud infrastructure while reducing client- side processing [7][8].

Algorithm:

```
def target_deduplication(data_stream):chunk_size = 4096
hash_table = {} unique_chunks = []metadata = []
for chunk in chunk_data(data_stream, chunk_size):chunk_hash = hash_function(chunk)
if chunk_hash not in hash_table: hash_table[chunk_hash] = len(unique_chunks)unique_chunks.append(chunk)
metadata.append(hash_table[chunk_hash])store_unique_chunks(unique_chunks) store_metadata(metadata)
def store_unique_chunks(unique_chunks):# Implement unique chunk storage
pass
def store_metadata(metadata):
# Implement metadata storagePass
```

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 1, August 2024

Working:

1. Chunking: After reaching the storage backend, the data is further granularized into pieces.
2. Hashing: Using cryptographic hash functions, a hash value will be computed for each of thesechunks.
3. Index Check: The storage backend maintains an index of chunk hashes.
   - For each chunk, the storage backend checks if the hash exists in the index.
4. Storage: In case of a unique chunk, the hash is not found in the index; the same is stored andthe index updated.
   - In case of a duplicate chunk, the hash is found in the index; only a reference to theexisting chunk is stored.
5. Metadata Updating: The metadata is maintained to keep track of which chunks make up a file so that the original data may later be reconstituted.

IV. CONCLUSION

In today's data management, deduplication represents one of the most important techniques. In particular, it gets huge attention in cloud storage and backup systems. Deduplication is a process efficient in storage use, cost reduction, and system performance enhancement by finding and eliminating redundant copies of data. This procedure reduces the amount of physical storage needed while simultaneously speeding up backup and data transfer, which results in significant cost savings and increased productivity [1][2][4].

The exact limits and requirements of the system determine which deduplication approach to use: source-based or target-based. While target-based deduplication concentrates on maximizing storage efficiency and administration at the cloud storage level, source-based deduplication is beneficial for lowering network load by removing duplicates prior to data transmission [3][7].

All things considered, effective deduplication technique application promotes sustainable data expansion and management, which makes it a crucial part of modern IT infrastructure. Enterprises will need deduplication technology to deliver on a truly effective and cost-efficient data storage solution while data volumes continue to grow [5][6][8].

REFERENCES

[1]. P. K. Kundu and V. K. Gupta, "Data Deduplication for Backup and Archive," IEEE Transactions on Computers, 2010.

[2]. L. K. Shun and D. J. McKay, "A Survey of Data Deduplication Techniques," ACM Computing Surveys, 2011.

[3]. S. P. Yao, J. H. Smith, "Evaluating the Impact of Data Deduplication on Storage Systems," ACMTransactions on Storage, 2015.

[4]. "Cost-Efficient Data Management Using Deduplication Techniques" by L. Zhang et al., IEEETransactions on Parallel and Distributed Systems, 2016.

[5]. "A Comprehensive Study of Deduplication Algorithms" by C. Zhang, M. Chen, and R. Hu, IEEETransactions on Knowledge and Data Engineering, 2018.

[6]. "Data Deduplication: A Survey and Research Directions" by H. Li and D. Xu, ACM Computing Surveys, 2014.

[7]. "Source-Based and Target-Based Deduplication: A Comparative Study" by M. D. McArthur et al., IEEE Transactions on Cloud Computing, 2017.

[8]. "Efficient Deduplication Algorithms for Cloud Storage" by X. Wang, H. Zhao, and W. Li, IEETransactions on Network and Service Management, 2019