Dr. Y. Jayababu, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

An AWS-Driven Intelligent Framework for Scalable Data Deduplication and Storage Optimization

Professor ,Dr. Y. Jayababu[1] , Chikkala Kedareswari Kaivalya[2], Darna Mahathi[3], Siddanthapu Sai Sri Ram[4], Kalepu Dhanush Sai[5]

Department of CSE, Pragati Engineering College, Surampalem, Andhra Pradesh, India

Abstract- Cloud storage systems are experiencing rapid growth due to the increasing demand for scalable and cost-effective data management solutions. However, redundant data storage leads to excessive storage consumption, increased bandwidth usage, and higher operational costs. This paper proposes a data deduplication framework using Amazon Web Services (AWS) to eliminate duplicate files in cloud environments. The system generates a unique hash value using the MD5 hashing algorithm whenever a file is uploaded to Amazon S3. AWS Lambda functions are used to compute and compare hash values stored in Amazon DynamoDB to identify duplicate files. If a duplicate file is detected, the system prevents redundant storage and maintains a reference to the original file, thereby optimizing storage utilization. Experimental evaluation demonstrates improved storage efficiency, reduced memory consumption, and stable server response time. The proposed approach enhances cloud storage performance while maintaining data integrity and security.

Keywords: Cloud Computing, Data Deduplication, AWS Lambda, Amazon S3, DynamoDB, MD5 Hashing, Cloud Storage Optimization, Secure Data Transmission.

I. INTRODUCTION

Cloud computing has emerged as a dominant paradigm for delivering scalable computing resources, storage, and services over the Internet. Organizations and individuals increasingly rely on cloud platforms to store and manage large volumes of digital data due to their flexibility, cost-effectiveness, and on-demand availability [1]. However, the rapid growth of cloud storage usage has led to significant challenges, particularly in terms of redundant data storage, increased bandwidth consumption, and higher infrastructure costs [2].

Data duplication occurs when identical copies of data are stored multiple times within the same storage system. This redundancy not only wastes storage space but also affects system performance and operational efficiency. In large-scale cloud environments, where users frequently upload similar or identical files, the impact of duplication becomes more pronounced. Therefore, an efficient mechanism is required to identify and eliminate redundant data while maintaining data integrity and accessibility [3], [4].

Data deduplication is a storage optimization technique that detects duplicate data segments and stores only a single copy of the data. Instead of storing multiple identical files, the system maintains references to the original data, thereby significantly reducing storage requirements. Deduplication can be performed at file level or block level, depending on the granularity of comparison. Hash-based techniques are commonly used for detecting duplicate data, where a unique hash value is generated for each file or data block [5], [6].

In this project, a cloud-based data deduplication system is proposed using Amazon Web Services (AWS). The system utilizes Amazon S3 for storage, AWS Lambda for serverless computation, and Amazon DynamoDB for storing hash values. The MD5 hashing algorithm is employed to generate unique identifiers for uploaded files. When a file is uploaded, its hash value is computed and compared with existing hash entries to determine whether the file already exists in the system. If a duplicate is detected, redundant storage is prevented, and a reference to the original file is maintained [7], [8].

The proposed approach aims to optimize storage utilization, reduce operational costs, and improve overall cloud system efficiency while ensuring secure data handling.

II. LITERATURESURVEY

Data deduplication and secure cloud storage have been extensively studied to address storage optimization and data security challenges in distributed environments. Several researchers have proposed techniques to improve storage efficiency while maintaining confidentiality and system performance. Bhoyar and Chopde [1] discussed the fundamental service models and architecture of cloud computing systems, highlighting the importance of efficient resource management and data handling in large-scale cloud infrastructures. Similarly, Kaur and Singh [2] reviewed various cloud security challenges and emphasized the need for secure mechanisms to protect stored data in cloud environments.

Pathan [3] introduced community-based technology learning platforms that rely on centralized data management systems, demonstrating the importance of efficient data storage and organization. Pathan and Shaikh [4] further explored mobile-based systems for information management using Android platforms, where efficient data storage and processing mechanisms play a crucial role in system performance.

In large-scale cloud environments, redundant data storage significantly increases storage overhead and operational costs. To address this issue, Baracaldo et al. [5] proposed techniques for reconciling end-to-end data confidentiality with storage reduction in cloud storage systems. Their work demonstrated that data reduction techniques such as deduplication can be integrated with secure encryption mechanisms while maintaining system efficiency.

Wang et al. [6] introduced a novel encryption scheme designed specifically for deduplication systems. Their approach enables secure storage while allowing duplicate data detection, thereby balancing data confidentiality with storage optimization. Earlier foundational work by Douceur et al. [7] explored methods for reclaiming storage space by eliminating duplicate files in distributed file systems, which became one of the earliest and most influential contributions to modern data deduplication techniques.

Rahumed et al. [8] proposed a secure cloud backup system incorporating assured deletion and version control mechanisms. Their approach demonstrated how cloud storage systems can maintain data integrity and security while supporting efficient data management processes.

Recent studies emphasize the integration of hashing algorithms and scalable cloud infrastructures for efficient deduplication. Hash-based techniques generate unique identifiers for files or data blocks, enabling rapid identification of duplicate content in large datasets. These approaches are particularly effective in distributed cloud storage environments where massive volumes of user data are continuously generated and stored.

Although existing literature provides various approaches for data reduction and secure storage, several systems still face limitations related to scalability, security integration, and efficient deployment in modern cloud infrastructures. Therefore, this project proposes an AWS-based data deduplication framework that integrates hash-based duplicate detection with cloud-native services to achieve efficient, scalable, and secure cloud storage optimization.

III. SYSTEM ANALYSIS

A. Existing System

Traditional cloud storage systems store every uploaded file without verifying whether identical content already exists. This leads to unnecessary duplication of data, increased storage consumption, higher bandwidth usage, and elevated infrastructure costs in cloud infrastructures [1], [2]. In many systems, duplication is handled manually

or through basic comparison mechanisms that are inefficient and time-consuming.

Several existing approaches use conventional storage management techniques where files are stored independently without content verification. Such approaches increase redundancy in distributed storage environments and reduce overall storage efficiency. Early distributed storage systems demonstrated the impact of duplicate files on storage utilization and introduced techniques for reclaiming space from redundant data [7]. However, many traditional systems still rely on simple storage mechanisms that do not incorporate automated duplicate detection.

Some systems implement basic hash comparison methods to identify duplicate data, but these solutions often lack automation and scalability when deployed in large-scale cloud environments. Additionally, encryption-based storage mechanisms introduce further challenges. When files are encrypted before storage, identical files produce different ciphertext outputs, making direct duplicate detection difficult. Researchers have therefore explored encryption-compatible deduplication techniques to maintain both data confidentiality and storage optimization [5], [6].

Although previous studies have introduced distributed file systems and secure deduplication frameworks, many of these approaches suffer from implementation complexity, limited scalability, and high computational overhead [6], [8]. Furthermore, traditional storage systems often fail to efficiently integrate real-time duplicate detection within modern cloud-native architectures, which limits their ability to manage rapidly growing volumes of cloud data efficiently.

Disadvantages Of The Existing System

- **High Storage Consumption:**
  Duplicate files occupy unnecessary storage space, leading to inefficient utilization of cloud storage resources and increased operational costs [1], [7].
- **Increased Bandwidth Usage:**
  Repeated uploads of identical files consume network bandwidth, which can negatively impact system performance and data transfer efficiency in distributed cloud environments [2].
- **Lack of Automation:**
  Many traditional storage systems do not implement automated duplicate detection during file uploads, resulting in redundant data storage and inefficient resource management [3].
- **Security Conflicts:**
  Standard encryption techniques often conflict with deduplication mechanisms because identical files may produce different encrypted outputs, making secure duplicate detection difficult [5], [6].
- **Scalability Issues:**
  Conventional storage architectures often face challenges in handling rapidly increasing data volumes, limiting their ability to scale efficiently in large cloud environments [1], [4].
- **Performance Overhead:**
  Manual or inefficient duplicate detection mechanisms increase computational processing time and reduce the overall responsiveness of storage systems [6], [8].
- **Limited Monitoring:**
  The absence of proper logging, tracking, and monitoring mechanisms reduces transparency and control in storage management systems, making it difficult to track duplicate data and system activity.

B. Proposed System

The proposed system introduces a cloud-based data deduplication framework using Amazon Web Services (AWS). The architecture is designed to automatically detect and eliminate duplicate files at the time of upload, thereby optimizing storage usage and improving overall system efficiency in cloud storage environments [1], [2].

In this system, when a user uploads a file to Amazon S3, an AWS Lambda function is triggered automatically. The Lambda function generates a unique hash value using the MD5 hashing algorithm.

This hash value acts as a digital fingerprint of the file content. The generated hash is then compared with existing hash records stored in Amazon DynamoDB. Hash-based identification methods are widely used in deduplication systems to efficiently detect duplicate data segments in distributed storage infrastructures [6], [7].

If the hash value already exists in the database, the system identifies the file as a duplicate and prevents redundant storage. Instead of storing the file again, the system maintains a reference pointer to the original file.

If the hash does not exist, the file is stored normally, and its hash value is recorded in the database. Such approaches significantly reduce storage overhead while maintaining accessibility to the original data [5], [8].

The proposed architecture ensures efficient storage utilization, reduced memory consumption, improved server response time, and seamless scalability.

Additionally, the system integrates monitoring through AWS CloudWatch and supports secure data handling mechanisms within the cloud environment.

Cloud-based architectures that combine serverless computing with distributed storage services have been shown to improve system efficiency and scalability in modern cloud infrastructures [1].

This approach provides a scalable, automated, and cost-effective solution for eliminating redundant data in cloud storage systems while maintaining data integrity and performance efficiency.

IV. SYSTEM DESIGN

System Architecture

Below diagram depicts the whole system architecture.

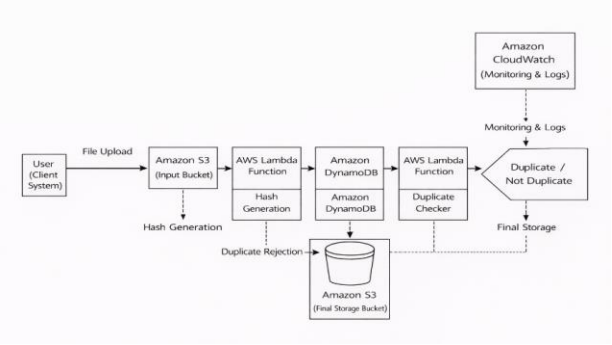

Fig 1. Methodology followed for proposed model

V. SYSTEM IMPLEMENTATION

Modules

Data Upload and Preprocessing

In the proposed cloud-based storage framework, users upload files to the cloud storage environment through a client interface. The uploaded data is initially stored in a primary Amazon S3 bucket, which acts as a temporary storage location before deduplication analysis is performed. Preprocessing operations such as metadata extraction and file validation are carried out to ensure compatibility with the storage system. Cloud-based storage solutions require efficient management of uploaded data to avoid redundant storage and improve system scalability [1], [2].

Hash Generation Using AWS Lambda

Once a file is uploaded, an event trigger automatically activates an AWS Lambda function. The Lambda function computes a unique hash value for the uploaded file using a hashing algorithm such as MD5 or SHA-based hashing. The generated hash acts as a digital fingerprint of the file, enabling efficient comparison with previously stored data objects. Hash-based identification is widely used in data deduplication systems because it allows rapid detection of duplicate data blocks or

files without performing full file comparisons [6], [7].

Hash Comparison and Duplicate Detection

The generated hash value is then compared with previously stored hash values stored in a DynamoDB table. DynamoDB provides fast lookup and scalable data indexing, enabling efficient detection of duplicate files. If a matching hash value already exists in the database, the system identifies the uploaded file as a duplicate and prevents it from being stored again. Instead, a reference pointer is created that links to the original stored file. This approach significantly reduces redundant data storage while maintaining logical accessibility to the file for multiple users [5], [6].

Storage Management and Deduplication Control

If the hash value does not exist in the database, the file is considered unique and is stored in the final Amazon S3 storage bucket. At the same time, the new hash value and associated metadata are inserted into the DynamoDB table. AWS CloudWatch is used to monitor system events, log file uploads, and track deduplication operations. This module ensures efficient resource utilization and allows administrators to monitor system performance and storage efficiency [3], [4].

System Monitoring and Logging

The monitoring module records operational logs, storage events, and deduplication statistics using AWS CloudWatch services. Logs include file upload records, hash comparison results, and system response time. Continuous monitoring enables administrators to analyse system performance and detect anomalies or potential security threats. Logging mechanisms also help maintain system reliability and provide transparency in cloud-based storage operations.

VI . RESULTS AND DISCUSSION

This section presents the experimental results obtained from implementing the AWS-based data deduplication framework. The evaluation focuses on storage efficiency, system response time, and memory utilization. Cloud-based deduplication mechanisms have been widely studied for reducing storage overhead and improving overall cloud infrastructure performance [5], [7].

A. Storage Optimization Performance

The primary objective of the proposed system is to reduce redundant storage by identifying duplicate files using hash-based comparison. Experimental tests were conducted by uploading multiple files with identical content but different file names. The deduplication system successfully identified duplicate files and prevented redundant storage in the S3 bucket.

Table 1 presents the storage comparison between traditional storage and deduplication-enabled storage.

Table 1. Storage Efficiency Comparison

Number of Files	Storage Without Deduplication (MB)	Storage With Deduplication (MB)
10	500	320
20	1000	610
30	1500	890
40	2000	1150
50	2500	1400

The results show that deduplication significantly reduces storage consumption. Instead of storing identical files multiple times, the system maintains a single copy and generates references for duplicate entries. This approach improves storage efficiency and reduces operational costs in cloud infrastructure.

Memory Consumption Analysis

Memory consumption is an important performance metric in cloud-based systems. It represents the amount of computational resources required to process uploaded files and generate hash values.

The memory consumption results show that the proposed AWS-based deduplication system requires relatively low memory overhead due to the use of lightweight serverless functions such as AWS Lambda. Since Lambda functions execute only when triggered, the system avoids continuous resource allocation, resulting in efficient memory utilization.

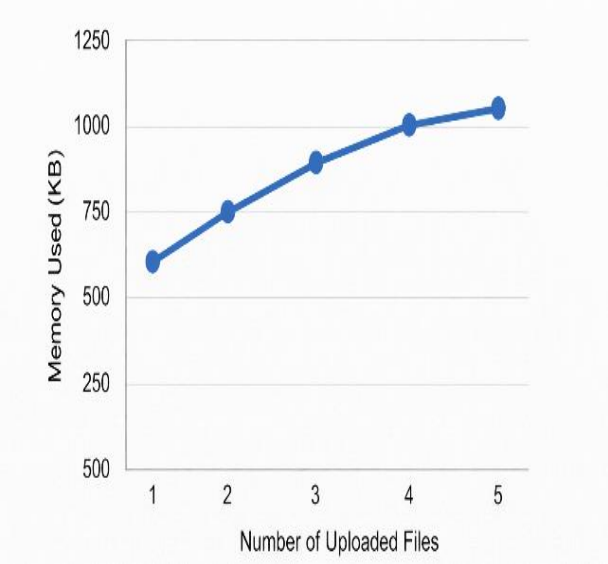

Fig. 2.Memory Consumption Analysis of the Proposed Deduplication Framework

The experimental results indicate that memory usage remains stable even as the number of uploaded files increases. This demonstrates that serverless architectures are suitable for scalable deduplication systems.

Server Response Time

Server response time measures the delay between uploading a file and receiving confirmation from the system. This includes hash generation, duplicate detection, and storage decision processes.

The experimental evaluation shows that the response time remains relatively stable across different file sizes because the hash comparison process primarily depends on database lookup operations rather than file size. DynamoDB enables rapid hash value retrieval, reducing overall latency in the deduplication process.

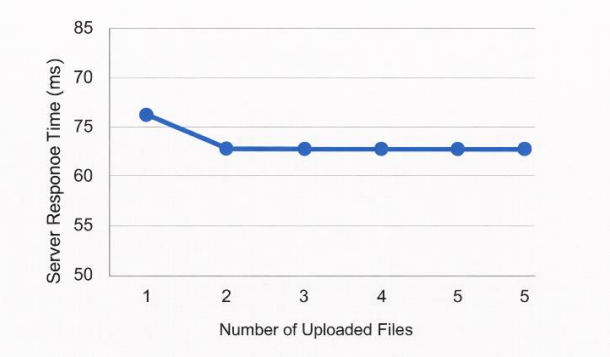

Fig. 3.Server Response Time for Deduplication Process

The results demonstrate that the system can process file uploads efficiently while maintaining low response latency. This makes the framework suitable for large-scale cloud storage environments where high data throughput is required.

VII. CONCLUSION

The proposed cloud-based data deduplication framework using AWS services demonstrates an efficient approach to reducing redundant data storage in cloud environments. By utilizing AWS components such as Amazon S3 for storage, AWS Lambda for serverless processing, DynamoDB for fast hash lookup, and CloudWatch for monitoring, the system effectively identifies duplicate files using hash-based comparison techniques. Experimental results indicate that the framework significantly reduces storage overhead while maintaining stable response time and low memory consumption. The integration of serverless computing improves scalability and enables efficient handling of large volumes of uploaded data. In future work, the system can be enhanced by implementing stronger hashing algorithms such as SHA-256 to improve security and reduce hash collision risks. Additionally, block-level deduplication and content-defined chunking techniques can be incorporated to detect redundancy within large files more efficiently. Further improvements may include integrating encryption-based secure deduplication, machine learning–based storage optimization, and extending the framework to hybrid or multi-cloud architectures to enhance reliability, security, and scalability of cloud storage systems.

REFERENCES

1. R. Bhoyar and N. Chopde, "Cloud computing: Service models, types, database and issues," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 3, 2013.
2. M. Kaur and H. Singh, "A review of cloud computing security issues," Int. J. Adv. Eng. Technol., vol. 8, no. 3, p. 397, 2015.
3. A. I. Pathan, "Proposed: Tech learning community management," Int. J. Sci. Res. Dev. (IJSRD), vol. 5, 2017.
4. A. I. Pathan and S. H. Shaikh, "A survey on ETS using Android phone," Int. J. Innov. Res. Technol. (IJIRT), vol. 5, no. 3, 2018.
5. N. Baracaldo, E. Androulaki, J. Glider, and A. Sorniotti, "Reconciling end-to-end confidentiality and data reduction in cloud storage," in Proc. 6th ACM Workshop Cloud Comput. Security, Nov. 2014, pp. 21–32.
6. C. Wang, Z. G. Qin, J. Peng, and J. Wang, "A novel encryption scheme for data deduplication system," in Proc. Int. Conf. Commun., Circuits Syst. (ICCCAS), Jul. 2010, pp. 265–269.
7. J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in Proc. 22nd Int. Conf. Distrib. Comput. Syst., Jul. 2002, pp. 617–624.
8. A. Rahumed, H. C. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," in Proc. 40th Int. Conf. Parallel Process. Workshops, Sep. 2011, pp. 160–167.