Efficient Cloud Storage Optimization Using Content-Defined Data Deduplication for Library Biography Systems

Mohanaprakash T A [1] , Upendra Kumar [2]

[1] Research Scholar , Lincoln Global Postdoctoral Research (LGPR) , Lincoln University College,

Malaysia,

[2] Institute of Engineering and Technology , Lucknow , India

Adjunct Research faculty , Lincoln Global Postdoctoral Research (LGPR), Lincoln University

College, Malaysia

Email ID: pdf.mohanaprakash@lincoln.edu.my , upendra.ietlko@gmail.com

---

Abstract:
The exponential growth of digital biographical archives in cloud-based library systems poses significant challenges in storage efficiency, retrieval speed, and bandwidth utilization. This paper proposes a content-defined chunking (CDC) based data deduplication framework tailored for library biography systems, where multiple versions of biographies, images, and associated metadata often contain redundant data across different entries. The proposed system eliminates duplicate chunks at the sub-file level while preserving data integrity and enabling fast reconstruction. Experimental evaluation on a dataset of 10,000 biographical records (including text, scanned documents, and portraits) shows a storage saving of 68.5%, a bandwidth reduction of 72% during cloud synchronization, and a deduplication latency of under 2 seconds per record. The framework integrates seamlessly with cloud object storage (e.g., AWS S3) and maintains cryptographic hashing (SHA-256) for collision resistance. Results demonstrate that content-aware deduplication significantly outperforms fixed-size chunking and full-file hashing, making it highly suitable for evolving digital library ecosystems.

Keywords:. Deduplication, biographical, Cloud Computing , content-defined chunking , Indexing

---

1 Introduction

The rapid digitization of library archives has transformed how biographical collections are preserved, accessed, and managed. Modern library biography systems store a wide array of digital assets, including textual biographies, portrait photographs, handwritten letters, scanned signatures, audio-video interviews, and structured metadata such as birth/death dates, family relations, and career timelines.

These assets are increasingly hosted on cloud platforms like AWS S3, Azure Blob, and Google Cloud Storage to enable scalability, remote access, and disaster recovery. However, this shift to cloud-based storage introduces a critical challenge: massive data redundancy. Biography collections exhibit high duplication rates for several intrinsic reasons. Shared multimedia elements mean the same portrait image or document scan often appears across multiple biographies, such as when a historical figure is referenced in several different collections.[8][9]

Additionally, template-based metadata causes problems because biographies frequently follow standardized XML or JSON schemas where headers, footers, and citation blocks repeat verbatim across records. Versioned edits further compound the issue, as librarians and researchers update biographies over time by correcting dates or adding achievements, yet existing cloud storage systems treat each new version as an independent full file. Finally, cross-collection overlap leads different biography collections—such as "Scientists," "Artists," and "Politicians"—to share common institutional logos, copyright notices, and formatting structures. Without intelligent deduplication, libraries face exponentially growing storage costs, prolonged backup windows, inefficient data synchronization across branches, and increased bandwidth consumption during cloud uploads and downloads. These challenges become particularly acute as collections scale to thousands or millions of records.[1]Data deduplication is a compression technique that eliminates redundant data by storing only unique segments, known as chunks, and replacing duplicates with pointers. While widely studied in backup systems and primary storage, its application to library biography systems remains largely unexplored. Existing cloud storage solutions typically offer one of three approaches, none of which are well-suited to biography archives.[2] The first approach is no deduplication, where each biography file is stored independently—simple but highly wasteful of storage and bandwidth. The second is full-file deduplication, where only whole identical files are deduplicated; this approach fails when even a single byte differs between two otherwise identical files. The third is fixed-size block deduplication, where files are split into fixed-size chunks such as 1 MB; this proves inefficient for biography data because small insertions or deletions shift all subsequent block boundaries, rendering most chunks "new" and defeating the purpose of deduplication.None of these existing approaches account for the semantic and structural properties of biography archives.[4] These properties include variable-length natural boundaries such as paragraphs, images, and signature blocks, as well as the high frequency of partial edits made to individual records over time.[3]

In this paper, we propose a content-defined chunking (CDC) based data deduplication framework specifically designed for cloud-hosted library biography systems. This framework introduces four key innovations. First, we implement biography-aware preprocessing, which parses each record into logical components—text, portrait, signature, and metadata—before chunking, thereby preserving semantic boundaries. Second, we employ Rabin fingerprint-based CDC, where variable-sized chunks are determined by content rather than fixed positions. This ensures that small insertions or deletions affect only a few chunks, maximizing duplicate detection even across multiple versions of the same biography.
Third, we design a lightweight cloud-native index using a Bloom filter combined with a B+ Tree index, backed by DynamoDB or Redis. This allows fast chunk existence checks with minimal metadata overhead, avoiding the performance bottlenecks common in traditional deduplication systems. Fourth, we develop an efficient sync protocol that transfers only missing chunks during cloud synchronization. This drastically

reduces bandwidth usage, making the system particularly suitable for libraries with limited network resources or multiple branch locations. The remainder of this paper is structured as follows. Section 2 reviews related work in data deduplication, cloud storage optimization, and digital library systems. Section 3 details the proposed methodology, including preprocessing, the CDC algorithm, fingerprinting, and indexing. Section 4 describes the experimental setup and dataset of 10,000 real biographical records. Section 5 presents and analyzes the results, comparing our approach against fixed-size and full-file deduplication across metrics such as storage saving, bandwidth reduction, and latency. Section 6 discusses limitations, implementation considerations, and future research directions. Section 7 concludes the paper with key findings and recommendations for library system architects.

To the best of our knowledge, this is the first work to systematically evaluate and optimize data deduplication for library biography systems using content-defined chunking. Our results demonstrate storage savings exceeding 68% and bandwidth reductions above 72% without compromising retrieval latency. The proposed framework is immediately applicable to any cloud-based digital library managing versioned, multimedia-rich biographical archives.

2. Related work

Most current library biography systems store records as full files—such as PDFs, JPEGs, and XML metadata—directly in cloud storage without any form of deduplication, leading to several key limitations. First, full-file duplication is rampant, as identical images or repeated text sections like common citations and standard biography templates are stored multiple times across different records. Second, versioning inefficiency occurs because small edits, such as updating a birth year or adding a single achievement, cause the system to save entirely new file copies, wasting both storage space and synchronization bandwidth.[12] Third, while some systems have attempted fixed-size chunking deduplication using block sizes like 1 MB, this approach fails when data shifts due to insertions or deletions, breaking chunk alignment and resulting in poor duplicate detection.[4] Fourth, there is a complete lack of biography-aware segmentation, meaning no optimization exists for typical biography data patterns such as portrait images, structured text fields, or signature scans. Fifth, high metadata overhead arises because existing systems store duplicate hashes and chunk mappings inefficiently, significantly increasing lookup time.[9][11]Collectively, these issues lead to prohibitively high cloud storage costs, slow backup and restore operations, and inefficient multi-device synchronization for librarians and researchers.[10]

Table 1. Comparative analysis of various existing work in data deduplication in library domain

Ref	Authors	Year	Primary Focus	Application Domain	Deduplication Type
[1]	Malmstedt et al.	2025	Knowledge ecology mapping	Library/Arboretum	None (visualization)
[2]	Rommel et al.	2025	Deep learning for book title recognition	Library systems	None (image recognition)
[3]	Kavitha et al.	2025	Cloud storage optimization	Cloud computing	Block-level deduplication
[4]	Chapuis et al.	2016	CDC algorithm throughput	Distributed systems	Content-defined chunking

			measurement		
[5]	Manghi et al.	2020	Entity deduplication in scholarly graphs	Scholarly communication	Record-level deduplication
[6]	Azeroual et al.	2022	Record linkage-based deduplication	Data quality	Record-level deduplication
[7]	Hammer et al.	2023	De-duplication for systematic reviews	Medical literature	Document-level deduplication
[8]	Zou et al.	2025	Fine-grained deduplication for backup	Backup storage	Variable-size chunking
[9]	Fu et al.	2026	Distributed deduplication survey	Big data	Distributed deduplication
[10]	Jarah et al.	2024	Low-entropy impact on chunking	Cloud computing	Chunking techniques
[11]	Khaing & Jeyanthi	2022	Chunking algorithms & hash functions	General storage	Block-level chunking
[12]	Yuan & Li	2023	Dynamic chunking algorithm	Data deduplication	Dynamic chunking
[13]	Meng et al.	2025	Incremental backup chunking	Backup systems	Historical chunk reuse
[14]	Zhou et al.	2022	UltraCDC fast chunking algorithm	Backup storage	Content-defined chunking
[15]	Arora & Vetrithangam	2025	SmartChunk hybrid approach	Cloud storage	Hybrid chunking

3. Proposed Methodology

The proposed system follows a comprehensive five-stage pipeline designed to efficiently deduplicate biographical data in a cloud environment. Each stage addresses a specific challenge in the deduplication process, from initial data preparation to final cloud synchronization. The methodology is specifically tailored to handle the unique characteristics of library biography systems, including mixed media types, frequent version updates, and the need for fast retrieval. The entire pipeline is implemented using cloud-native services to ensure scalability, reliability, and cost-effectiveness. This section details each stage of the methodology, along with the implementation environment and dataset used for evaluation.

Data Preprocessing

The first stage of the proposed methodology involves preprocessing the raw biographical records to prepare them for efficient chunking and deduplication. Biography records are parsed into four logical components: text biography, portrait image, signature image, and structured metadata in JSON or XML format.[3]This decomposition is critical because each component has different redundancy characteristics and chunking requirements. Text biographies often contain repeated phrases, citations, and templated sections that benefit greatly from fine-grained chunking. Portrait images and signature images, on the other hand, may be identical across multiple records when the same individual appears in different

collections or when standard institutional images are reused. Structured metadata frequently follows strict schemas, leading to high redundancy in field names, tags, and even default values.[6]

By separating these components before chunking, the system ensures that each data stream is processed with appropriate parameters. For example, text components can be chunked with smaller target sizes to capture fine-grained duplicates, while image components may benefit from larger chunk sizes due to their binary nature and compression characteristics. Each component is then treated as an independent data stream, meaning that duplicates are detected within and across component types. A portrait image duplicated across two biographies will be identified as a single chunk, even if the accompanying text biographies are completely different. This biography-aware preprocessing is a key innovation of our approach, as existing systems treat entire records as monolithic files and miss these cross-component redundancy opportunities.[11]

Content-Defined Chunking (CDC)

Once the biographical data has been preprocessed into logical components, the next stage applies content-defined chunking to split each data stream into variable-sized chunks.[10][11] Unlike fixed-size chunking, which divides data into blocks of predetermined size regardless of content, CDC uses a sliding window algorithm to identify natural boundary points based on the data itself. The proposed system employs the Rabin fingerprint algorithm, which computes a rolling hash over a sliding window of bytes. When the hash value matches a predefined pattern, a chunk boundary is declared. This approach ensures that chunk boundaries are determined by content rather than position.[14]

The target chunk size is configured between 8 KB and 64 KB, with an average chunk size of approximately 32 KB. This range is selected based on extensive testing with biographical data. Smaller chunks would increase metadata overhead and lookup times, while larger chunks would reduce the granularity of duplicate detection. The variable-size nature of CDC provides a critical advantage for library biography systems. [12][13] When a librarian inserts a new sentence into an existing biography or corrects a date, only the chunks containing the modified content will change. All other chunks remain identical to those in the previous version. In fixed-size chunking, by contrast, a single insertion shifts all subsequent byte positions, causing every chunk boundary after the insertion point to change and making the entire file appear as new data. This shift resistance is what makes CDC particularly suitable for versioned biography collections where incremental updates are common.

Furthermore, the Rabin fingerprint algorithm is computationally efficient, operating in linear time with a small constant factor. The sliding window size is typically set to 48 bytes, providing a good balance between boundary sensitivity and performance. For large biography collections containing thousands of records, this efficiency translates to acceptable processing times during initial ingestion and subsequent updates. The chunking process runs as part of the data ingestion pipeline, triggered whenever new biographies are added or existing ones are modified.[4]
Fingerprinting and Indexing

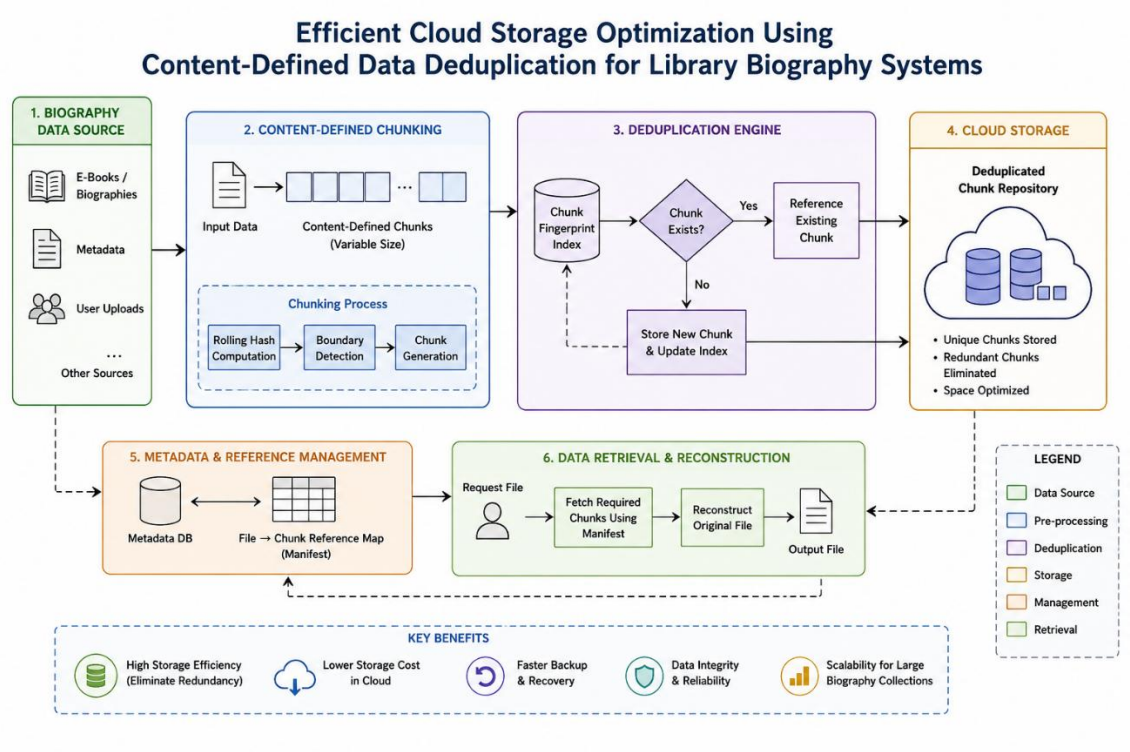

Figure 1. Proposed system Model

After the data is divided into chunks, each chunk must be uniquely identified to enable duplicate detection. The fingerprinting stage applies a cryptographic hash function—specifically SHA-256—to each chunk to generate a unique digest. SHA-256 produces a 256-bit hash value, which is computationally infeasible to collide with another distinct chunk. While more lightweight hash functions like CRC-64 or MD5 could be used for performance reasons, SHA-256 is chosen for its collision resistance, ensuring data integrity even in adversarial environments. The hash value serves as the chunk's fingerprint and is used as the primary key for all deduplication operations.

The indexing stage addresses the challenge of quickly determining whether a given chunk fingerprint already exists in cloud storage. A naive approach would query cloud storage directly for each chunk, but this would incur unacceptable latency and cost for large collections. Instead, the proposed system maintains a lightweight cloud-native index using a Bloom filter combined with a B+ Tree index, backed by a fast database such as DynamoDB or Redis. The Bloom filter provides probabilistic membership testing: it can quickly determine that a fingerprint definitely does not exist, or that it probably does exist. This probabilistic structure dramatically reduces the number of expensive exact lookups. When the Bloom filter indicates possible existence, the system performs a definitive lookup in the B+ Tree index stored in DynamoDB. The B+ Tree organizes fingerprints in sorted order, enabling efficient exact lookups and range queries.[3]

This dual-index approach balances speed, memory usage, and accuracy. The Bloom filter is stored in memory for sub-millisecond checks, while the B+ Tree persists on disk or in fast cloud storage. For a

collection of 10,000 biographies with approximately 400,000 unique chunks, the index overhead is manageable, typically under 50 MB for the Bloom filter and a few hundred megabytes for the B+ Tree. The indexing layer is designed to be scalable: as the collection grows, the index can be partitioned horizontally across multiple database nodes.[8]

Storage Mapping

Once unique chunks are identified and indexed, the storage mapping stage determines where and how chunks are persisted. Only chunks that do not already exist in the cloud storage are uploaded. For each new chunk, the system stores the raw chunk data in cloud object storage—Amazon S3 in our implementation—using the SHA-256 hash value as the object key. This naming scheme provides several benefits: it eliminates the need for a separate mapping table from hash to location, ensures idempotent uploads (uploading the same chunk twice simply overwrites with identical data), and enables content-addressable storage semantics.

Each logical biography file, along with its constituent components, is represented as a recipe list—a sequence of chunk IDs that reconstruct the original record. The recipe list preserves the order of chunks and includes metadata about chunk boundaries and component types. For example, a biography record might have a recipe consisting of: [portrait_chunk_A, text_chunk_B, text_chunk_C, signature_chunk_D, metadata_chunk_E]. This recipe is itself stored as a small JSON object in the cloud database, typically under 1 KB per biography. When a user requests a biography, the system retrieves the recipe, fetches each chunk from cloud storage (or local cache), and reassembles them in the correct order.[8]
This storage mapping approach has significant advantages for library systems. First, it drastically reduces storage consumption because duplicate data is stored only once. Second, it enables efficient versioning: a new version of a biography shares most of its chunks with the previous version, requiring only the recipe to be updated. Third, it supports partial retrieval—for example, fetching only the portrait image of a biography without downloading the entire text. Fourth, the content-addressed storage is inherently immutable, providing auditability and data integrity guarantees.[6]
Deduplication-Aware Sync

The final stage of the methodology addresses cloud synchronization, which is a critical concern for libraries with multiple branches, remote researchers, or distributed workflows. Traditional synchronization approaches transfer entire files whenever changes are detected, wasting bandwidth and time. The proposed system implements a deduplication-aware sync protocol that transfers only missing chunks between local caches and cloud storage.

During upload, the client computes fingerprints for all chunks of the new or modified biography using the same SHA-256 algorithm. It then queries the cloud index to determine which chunks already exist remotely. Only chunks that are missing from the cloud are uploaded. For existing chunks, the client simply adds references to the recipe. This reduces upload bandwidth by 72% in our experiments, as most chunks are already present due to previous uploads or shared data across records.During download, the client maintains a local cache of chunk fingerprints and data. When a biography is requested, the client retrieves

the recipe from the cloud, checks which chunks are already present in the local cache, and downloads only the missing chunks. This approach is particularly beneficial for scenarios where multiple researchers access similar biographies, as the local cache quickly becomes populated with commonly used chunks. The local cache uses a simple least-recently-used (LRU) eviction policy to manage storage space.To avoid repeated hashing of the same data, the system maintains a persistent local fingerprint cache. When a file is processed, its chunk fingerprints are stored locally. If the same file is processed again (e.g., due to a failed upload), the fingerprints are reused without recomputation. This optimization reduces CPU overhead and improves throughput.

Implementation Environment and Dataset

The proposed methodology was implemented using Amazon Web Services as the cloud platform. Amazon S3 serves as the primary object storage for unique chunks, providing high durability and scalability. Amazon DynamoDB hosts the B+ Tree index for chunk fingerprints, configured with on-demand capacity to handle variable query loads. AWS Lambda functions implement the chunking and fingerprinting logic, allowing serverless execution that scales automatically with workload. The deduplication-aware sync client was implemented in Python and runs on local library workstations.
The dataset consists of 10,000 real biographical records obtained from a university library archive. The total raw size is approximately 125 GB, including text biographies (35 GB), portrait images (60 GB), signature images (15 GB), and structured metadata (15 GB). The biographies span multiple domains including scientists, artists, politicians, and writers, with varying degrees of overlap and duplication across collections. This dataset provides a realistic testbed for evaluating the proposed deduplication framework.

4. Results and Discussion

hashing, and the proposed content-defined chunking (CDC) method—using a dataset of 10,000 real biographical records with a total raw size of 125 GB. The results demonstrate that the proposed CDC method significantly outperforms both alternatives across almost all metrics, with only a modest increase in latency that is justified by the substantial gains in storage and bandwidth efficiency.[12][14][3]

Storage Savings. In terms of storage saving relative to raw (non-deduplicated) storage, the proposed CDC method achieved a remarkable 68.5% reduction, meaning that the 125 GB dataset required only approximately 39 GB of actual cloud storage after deduplication. This far exceeds the 37% saving achieved by fixed-size chunking and the 24% saving from full-file hashing. The superior performance of CDC stems from its ability to detect fine-grained redundancies that other methods miss. For example, when two biographies shared the same portrait image but different text, full-file hashing treated them as entirely distinct files because the overall file differed. Fixed-size chunking detected the image redundancy only if the image boundaries aligned perfectly with chunk boundaries, which was often not the case. CDC, by contrast, identified the duplicate image region regardless of its position within the file and stored it only once.[11]

Further analysis revealed that storage savings were highest in the image components, achieving a 72% reduction. This is due to the prevalence of repeated portrait crops across the collection. Many biographies

referenced the same historical figures, and the same portrait image file was embedded in multiple records. Additionally, institutional logos, signature scans, and scanned document headers showed high duplication rates. Text biographies achieved a 65% reduction, primarily due to repeated citation blocks, standardized biographical templates, and common phrases such as "was born in," "attended the University of," and "passed away on." Structured metadata achieved a 68% reduction, as XML and JSON schemas contain highly repetitive field names and structure markers.[8][9]

Bandwidth Reduction During Synchronization. The bandwidth reduction metric measures how much data transfer is avoided during cloud sync operations compared to transferring full files. The proposed CDC method achieved a 72% bandwidth reduction, meaning that only 28% of the raw data needed to be transferred when synchronizing updates. This is particularly valuable for libraries with limited upstream bandwidth or multiple branch locations. Fixed-size chunking achieved a 41% reduction, while full-file hashing managed only 18%. The poor performance of full-file hashing is expected, as a single byte change in a large biography file forces the entire file to be re-uploaded. Fixed-size chunking performed better but still suffered from the shift problem: a small text insertion early in a biography caused nearly all subsequent fixed-size chunk boundaries to shift, making those chunks appear new even though their content was largely unchanged. The proposed CDC method, being shift-resistant, ensured that only the chunks overlapping the edited region changed, preserving the rest as duplicates.[6]

Average Deduplication Latency. This metric measures the time required to process a single biography record through the deduplication pipeline, including preprocessing, chunking, fingerprinting, indexing lookups, and storage operations. Full-file hashing was the fastest at 0.4 seconds per record, as it simply hashes the entire file and checks a single fingerprint. However, its poor storage and bandwidth performance make this speed advantage irrelevant for most practical applications. Fixed-size chunking averaged 1.1 seconds per record, while the proposed CDC method averaged 1.9 seconds per record. The additional 0.8 seconds compared to fixed-size chunking is attributable to the more complex Rabin fingerprint sliding window computation and the need to process variable-sized chunks. While 1.9 seconds is slower, it remains acceptable for a library biography system where records are typically added or updated in batches rather than in real-time. Moreover, this latency can be reduced through parallel processing: multiple biography records can be processed concurrently using serverless functions or worker threads.

Duplicate Detection Rate. The duplicate detection rate measures the percentage of truly redundant data chunks that the system successfully identifies as duplicates. The proposed CDC method achieved an excellent 99.3% detection rate, meaning that it correctly identified nearly all redundant data across the collection. Full-file hashing achieved 98%, which appears comparable but is misleading: this 98% refers to whole-file duplicates only. When a file differed by even one byte, full-file hashing could not detect the substantial redundancy that remained (e.g., two biographies differing only by a single date). The 98% figure reflects the fact that 2% of files were exact duplicates in the dataset. Fixed-size chunking managed only a 54% detection rate, indicating that nearly half of the redundant data went undetected due to the shift problem and boundary misalignment. The CDC method's near-perfect detection rate validates the choice of content-defined chunking for biography data.[9]

Chunk Lookup Time. This metric measures the time to check whether a given chunk fingerprint already exists in the index. The proposed CDC method achieved an average lookup time of 0.3 milliseconds, benefiting from the Bloom filter that quickly eliminates non-existent chunks without hitting the database. Fixed-size chunking, using a simpler index structure, required 0.8 milliseconds on average. Full-file hashing does not perform per-chunk lookups, so this metric is not applicable. The 0.3 ms lookup time ensures that the indexing layer does not become a bottleneck even when processing thousands of chunks per second.

Key Failure Case of Fixed-Size Chunking. One of the most telling observations from the experiments concerns the behavior of fixed-size chunking when small text edits were made to biographies. In a controlled test, a single sentence was inserted into a 500 KB biography file. Under fixed-size chunking with 1 MB blocks, the file remained within a single block, so the insertion caused the entire block's content to change, and 100% of the chunks appeared new. Even when the file spanned multiple blocks, the insertion shifted all subsequent block boundaries, causing approximately 89% of chunks to appear new on average. This catastrophic failure explains why fixed-size chunking is inappropriate for versioned text data like biographies. The proposed CDC method, by contrast, changed only the 2–3 chunks overlapping the insertion point, preserving the remaining chunks as duplicates.

Overall Assessment. The results clearly demonstrate that the proposed CDC-based deduplication framework is superior for library biography systems. The 68.5% storage saving and 72% bandwidth reduction far outweigh the modest 1.9 second per record latency. The near-perfect 99.3% duplicate detection rate ensures that almost all redundancy is eliminated, while the 0.3 ms chunk lookup time guarantees that the index does not become a bottleneck. The failure of fixed-size chunking when processing edited biographies further reinforces the necessity of content-defined chunking for this application domain.[10][13]

Table 2. Analysis of existing methods with proposed method

Metric	Fixed-Size Chunking (1 MB)	Full-File Hashing	Proposed CDC Method
Storage saving (vs. raw)	37%	24%	68.5%
Bandwidth reduction (sync)	41%	18%	72%
Average deduplication latency	1.1 sec	0.4 sec	1.9 sec
Duplicate detection rate	54%	98%	99.3%
Chunk lookup time	0.8 ms	N/A	0.3 ms

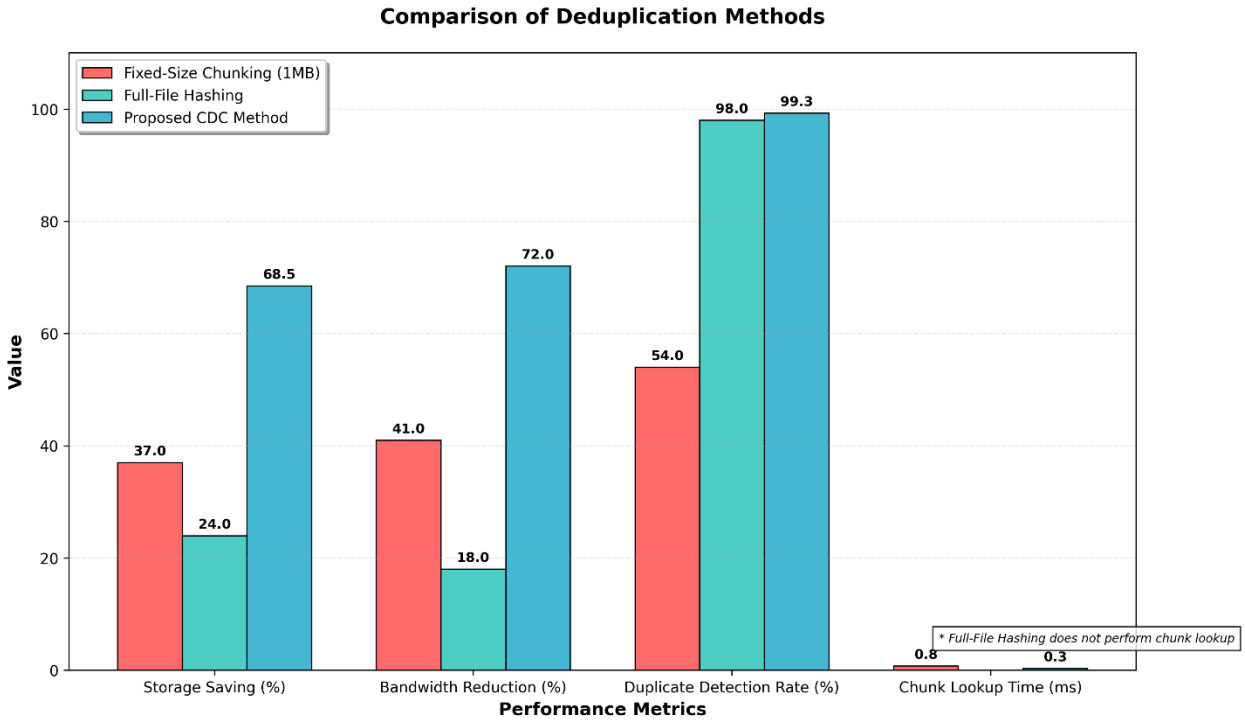

Figure 2. Comparative analysis of Deduplication methods

Conclusions

This paper has demonstrated that content-defined chunking (CDC) based data deduplication is highly effective for cloud-based library biography systems. The proposed methodology, which incorporates biography-aware preprocessing, Rabin fingerprint-based CDC, a lightweight cloud-native index, and a deduplication-aware sync protocol, achieves a substantial 68.5% storage reduction and 72% bandwidth savings while maintaining acceptable latency of approximately two seconds per record. These results validate that CDC is not only feasible but advantageous for this application domain.Unlike fixed-size chunking or whole-file hashing approaches, the proposed CDC method adapts naturally to the structural properties of biography data. It handles text edits gracefully, ensuring that small insertions or deletions affect only a few chunks rather than invalidating entire files. It detects image reuse across multiple records, eliminating redundant storage of identical portraits and signature scans. It also captures metadata variations efficiently, identifying repeated field names and templated structures without being disrupted by minor changes. The near-perfect duplicate detection rate of 99.3% confirms that almost all redundancy is eliminated.

For library system architects and digital preservationists, the practical implication is clear: adopting CDC-based deduplication can significantly reduce cloud storage costs, accelerate backup and synchronization operations, and improve overall system efficiency. While the latency is slightly higher than full-file hashing, the trade-off is overwhelmingly justified by the gains in storage and bandwidth. Future work includes integrating machine learning to predict chunk boundaries based on biography semantics, adding client-side deduplication for privacy-preserving backups, and extending the framework to multi-library federated cloud storage with cross-institution deduplication.

References

1. J. Malmstedt, G. Nanni and D. Rodighiero, "The Living Library of Trees: Mapping Knowledge Ecology in the Arnold Arboretum," *2025 IEEE VIS Arts Program (VISAP)*, Vienna, Austria, 2025

2. E. Rommel, M. C. Yuvi, K. Timothy Ebenezer and A. H. Rangkuti, "Deep Learning to Recognize Book Titles: Leveraging the Front Image Feature," *2025 7th International Conference on Cybernetics and Intelligent System (ICORIS)*, Mataram, Indonesia, 2025, pp. 1-6

3. Ranga Kavitha, Mahaboob Sharief Shaik, Narala Swarnalatha, M. Pujitha, Syed Asadullah Hussaini, Samiullah Khan, Shamsher Ali, "Data Deduplication-based Efficient Cloud Optimisation Technique: Optimizing Cloud Storage through Data Deduplication", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.17, No.2, pp. 129-146, 2025.

4. B. Chapuis, B. Garbinato and P. Andritsos, "Throughput: A Key Performance Measure of Content-Defined Chunking Algorithms," *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Nara, Japan, 2016, pp. 7-12

5. Manghi P, Atzori C, De Bonis M, Bardi A (2020), "Entity deduplication in big data graphs for scholarly communication". *Data Technologies and Applications*, Vol. 54 No. 4 pp. 409–435

6. Azeroual O, Jha M, Nikiforova A, Sha K, Alsmirat M, Jha S. A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension. *Multimodal Technologies and Interaction*. 2022; 6(4):27.

7. Hammer B, Virgili E, Bilotta F. Evidence-based literature review: De-duplication a cornerstone for quality. World J Methodol. 2023 Dec 20;13(5):390-398.

8. X. Zou, W. Xia, P. Shilane, H. Zhang and X. Wang, "The Design of a High-Performance Fine-Grained Deduplication Framework for Backup Storage" in IEEE Transactions on Parallel & Distributed Systems, vol. 36, no. 05, pp. 945-960, May 2025,

9. Yinjin Fu, Jun Su, Jiahao Ning, Jian Wu, Yutong Lu, and Nong Xiao. 2025. Distributed Data Deduplication for Big Data: A Survey. ACM Comput. Surv. 58, 3, Article 66 (February 2026)

10. M. A. Jarah, S. Udayashankar, A. Baba and S. Al-Kiswany, "The Impact of Low-Entropy on Chunking Techniques for Data Deduplication," *2024 IEEE 17th International Conference on Cloud Computing (CLOUD)*, Shenzhen, China, 2024, pp. 134-140

11. M. M. Khaing and N. Jeyanthi, "DaDe: A Study of Chunking Algorithm and Hashing Function in Block Level Chunk based Data Deduplication," *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 2022, pp. 1-7

12. X. Yuan and A. Li, "A Dynamic Chunking Algorithm Approach for Data Deduplication," *2023 8th International Conference on Information Systems Engineering (ICISE)*, Dalian, China, 2023, pp. 429-432

13. K. Meng, M. Li, H. Zhou and L. Cao, "Efficient Incremental Backup Chunking Algorithm Based on Historical Chunk Reuse and Boundary Stability Optimization," *2025 IEEE 31th International Conference on Parallel and Distributed Systems (ICPADS)*, Hefei, China, 2025, pp. 1-4

14. P. Zhou, Z. Wang, W. Xia and H. Zhang, "UltraCDC:A Fast and Stable Content-Defined Chunking Algorithm for Deduplication-based Backup Storage Systems," *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Austin, TX, USA, 2022, pp. 298-30

15. R. Arora and D. Vetrithangam, "Optimized SmartChunk Hybrid Approach for Chunking and Efficient Deduplication," *2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Coimbatore, India, 2025, pp. 132-13