# Improving the Efficiency of Deduplication Process by Dedicated Hash Table for each Digital Data Type in Cloud Storage System

**G. Sujatha**
Sathyabama Institute of Science and Technology, Chennai, India.
E-mail: sujathag@srmist.edu.in

**Dr. Jeberson Retna Raj**
Sathyabama Institute of Science and Technology, Chennai, India.
E-mail: jebersonretnarajr@gmail.com

*Received November 12, 2020; Accepted December 10, 2020*
*ISSN: 1735-188X*
*DOI: 10.14704/WEB/V18SI01/WEB18060*

## Abstract

Data storage is one of the significant cloud services available to the cloud users. Since the magnitude of information outsourced grows extremely high, there is a need of implementing data deduplication technique in the cloud storage space for efficient utilization. The cloud storage space supports all kind of digital data like text, audio, video and image. In the hash-based deduplication system, cryptographic hash value should be calculated for all data irrespective of its type and stored in the memory for future reference. Using these hash value only, duplicate copies can be identified. The problem in this existing scenario is size of the hash table. To find a duplicate copy, all the hash values should be checked in the worst case irrespective of its data type. At the same time, all kind of digital data does not suit with same structure of hash table. In this study we proposed an approach to have multiple hash tables for different digital data. By having dedicated hash table for each digital data type will improve the searching time of duplicate data.

## Keywords

Deduplication, Cryptographic Hash, Searching Time, Digital Data, Cloud Storage.

## Introduction

Cloud storage is a collection of logical pools in which digital data can be stored. The actual physical space extents multiple servers possibly in multiple locations that is typically owned by a hosting company or cloud storage providers and they are the one responsible for making data available all the time for the authenticated users. The cloud storage is a computing model that stores digital data which is managed by cloud storage

provider who operates data storage as a service. It provides on-demand service kamala Kannan (2012), Rajan (2012). So that organization or any individual users can free from owning and managing their own data storage infrastructure. It will reduce the cost and time. There are different types of cloud models are available. They are Private, Public and Hybrid cloud models. By which the individual users or any organization can even share their storage space with others. When many users are sharing the storage space then existence of duplicate data is unavoidable. This duplicate data can be from different users or even from the same user. This will leads to wastage of storage space. To avoid such duplicate copies get stored in the storage space, "Data Deduplication", Tang (2016) can be implemented which will prevent the duplicate data copies.

The cloud storage space can be used to store any type of digital data. The data which is stored in a digital format like image, audio, video or document is called digital data. There are different types of digital data existing. They are unstructured data, semi-structured data and structured data. Unstructured data may be bitmap images like Image, video, and audio files and textual objects like word document, email or excel. Semi-structured data may be Email, XML or TCP/IP packets. And structured data may be databases like Access, OLTP systems and even SQL. In these type of digital data, structured data are very easy to store, retrieving, indexing, updating, deleting and searching. They are easily scalable and highly secure. Semi-structured data is almost same as structured data except that is one processing step away from structured data and there is a possibility of storing in structured format.

The problem is with handling unstructured data. It has the following challenges 1. Storage space, since it requires lot of space 2. Scalability 3. Security Rao (2016), 4. Retrieving information Wu (2018), Pritha (2015) 5. Update and 6.Indexing and searching Paulo (2014). To reduce the storage space requirement, we can employ compression. For security we can encrypt the data. The main problem is with indexing and searching. Uniquely identifying the unstructured data is a very big challenge. We can use the hash value of the data for indexing. Using that we can identify the unstructured data uniquely. Hash value is a unique value that can be generated for each data. When considering storage space, this unstructured data will occupy much space when compared to other types of data. When duplicate copy exists then that will leads to poor utilization of storage space.

For implementing deduplication, each data should be identified uniquely. Then only we can conclude whether the duplicate copy is been tried to store in the storage space or not. For unique identification as well as for correctness of data, we can generate hash value for

each data. Whenever we want to access the data, we need to calculate the hash value and that should be compared with the stored hash of that data to check its correctness. At the same time, to search the existence of the digital data without accessing the actual data is possible by searching its unique hash value. If that hash value exists, this means that particular data also exists in the storage space. The hash value will be calculated for all digital data that are stored in the storage space and these hash values will be stored in a hash table for future reference. Whenever we want to compare the stored hash value with the input hash value, and it can be done by traversing the table linearly to search for the hash value. In the era of information the next challenge is scalability. When the amount of such unstructured data increased, the size of the hash table should also be increased proportionately. This in turn increases the searching time of the hash. In this study we are proposing a Multiple Hash table model to reduce the searching time.

## Related Work

The proof of ownership Maruti (2015) for data being uploaded in cloud storage is also an important issue. The duplication check is processed with the help of this ownership detail. The content level duplication is identified with the help of hash value of file content. The confidentiality is added by using convergent key concept. The data is encrypted with the help of convergent key received from the user while uploading the file. The user should add a tag along with the data and this tag will be helpful to detect duplicate copies.

In finding duplicate copies, the information retrieval plays a vital role. The chunk-lookup disk bottleneck problem is a major problem in the process of information retrieval. The Linear hashing with key groups (LHs), Zhang (2102) overcome the problem of memory wastage due to maintain index for all the data chunks. Instead of maintaining indexes, in this method they used linear hashing for bin address computation. This address will be used for finding duplicate copies.

The deduplication process is considered to be costly because of its high computation and storage space requirement. This issue is overcome by weak-hash-based deduplication Ni F (2019). The authors proposed Write-once data journaling technique using the weak-hash based deduplication to avoid duplicate copies getting stored in data journaling. This will reduce the usual overheads in deduplication process.

## Proposed Methodology

The duplicate copies can be identified with the help of hash value Du (2020). Hash value is a unique value created for digital data. Using this hash value, a digital data can be

uniquely identified Vijayakumar (2020) in turn helpful in finding duplicate copy. In general, whenever a digital data is getting stored in the cloud storage space, its hash value will be created and it needs to be checked against all the existing data for its similarity. Instead of comparing the data directly with all other data, their hash values will be compared for finding the similarity. For this process, the entries in the hash table are searched sequentially. When the size of the hash table increased, proportionately the searching time will be increased. The hash table will contain the hash entries for all types of digital data.

In the proposed methodology, to decrease the time complexity of searching a hash value in the hash table, we recommend a multiple hash table, each for each type of digital data. If each type of digital data has its dedicated hash table, then the time required to search a hash value will be decreased.

## Proposed Architecture

In the proposed architecture, we suggested to have multiple hash tables for various kinds of digital data.

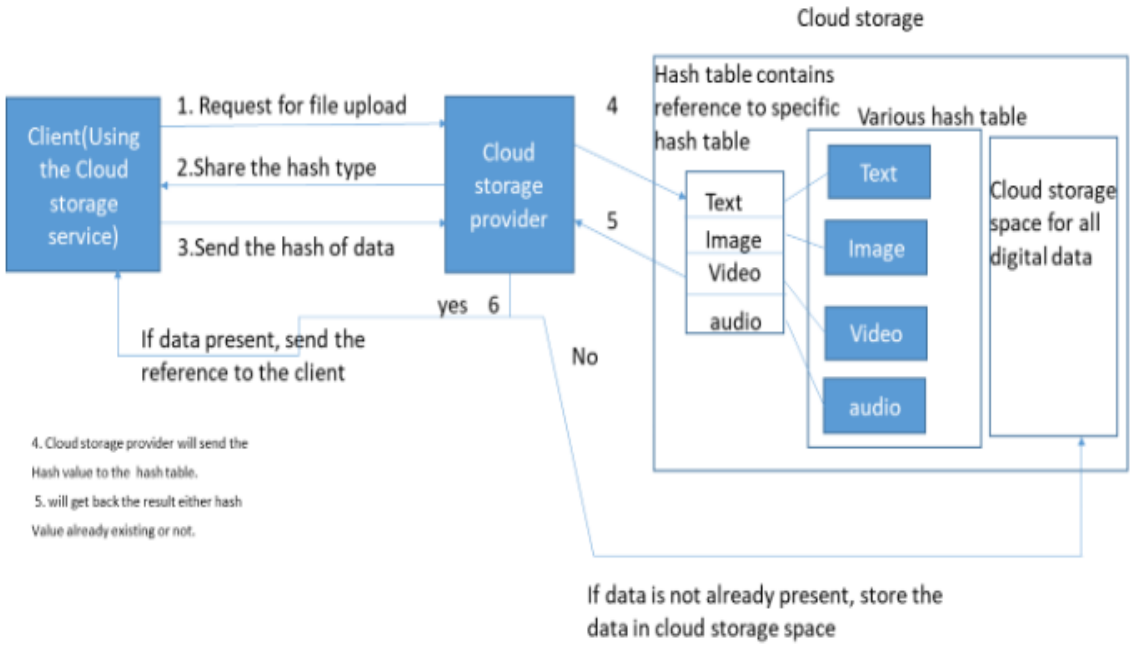


**Figure 1 Architecture of dedicated hashtable**

When the cloud storage user sends a request to the cloud storage provider to store its digital data in the cloud storage space, the user has to negotiate with the cloud storage provider regarding the hash algorithm to be used to compute the hash value of the digital

data. When the user calculated the hash value, the user has to send the hash value along with the type of the digital data to the CSP. The CSP in turn get the reference of the corresponding hash table and search for its match in the respective hash table. If it finds the hash value, CSP can conclude that the particular digital data is already present and send that reference link to the storage client. Otherwise it will store the data in the cloud storage space and send the reference link to the client.

### A. Flow Diagram

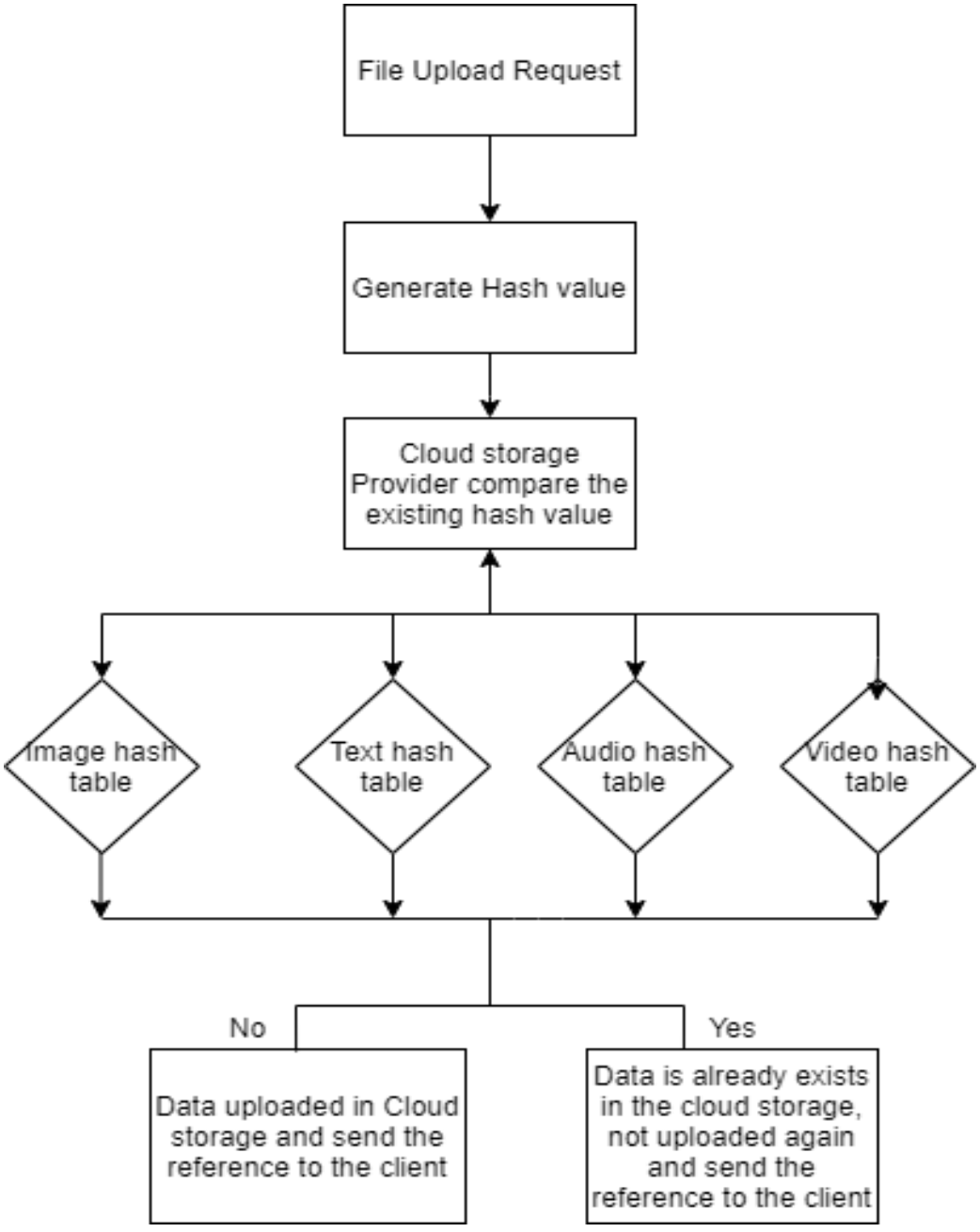


**Figure 2 Flow diagram of dedicated hash table process**

## B. Steps Involved in Deduplication Process Using Multiple Hash Table

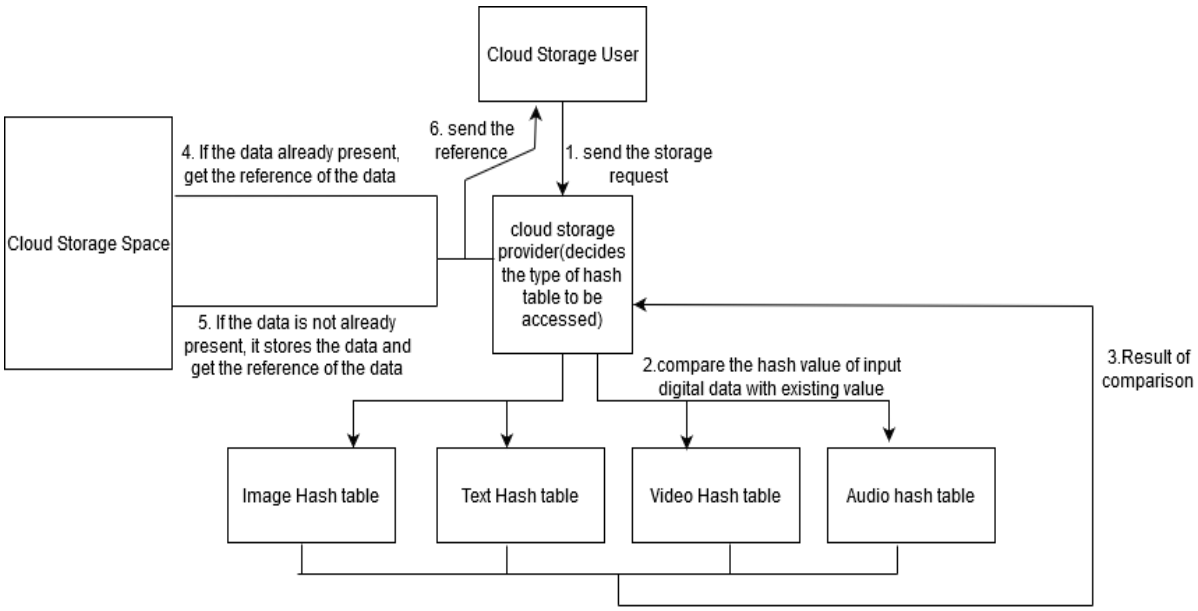


**Figure 3 Steps Involved in Deduplication Process**

## C. Steps

1. The storage client sends the request to the Cloud storage provider by sending its hash value.
2. The Cloud storage provider will check its existence in the hash table.
3. Depends upon the result of comparison, Cloud storage providers will move on to the next step
4. If the hash value is already present, it means the digital data is also already stored in the storage space. Then the Cloud storage provider will get the reference link of the existing data.
5. If there is no match for the hash value, then Cloud storage provider will conclude that the data is not already present in the storage space. It stores the data in storage space.
6. And finally send the reference link to the client.

## D. Reducing Time Complexity

The time complexity in searching hash value in the hash table plays a vital role in the process of Data deduplication. The searching time is proportional to the size of the hash table, since the search process is in linear fashion Goyal(2018), Morrice(2018). When we have a single hash table for all the type of digital data, then definitely the size of the hash table will be huge. When we have dedicated hash table for each type of digital data, then the corresponding hash table size will be reduced, at the same time the searching space is also getting reduced. This in turn reduces the searching time.

## Results and Discussions

For the experimental purpose, we derived four datasets named Dataset1, Dataset2, Dataset3 and Dataset4 by combining existing data sets specified in the table. We made four data sets having mixed data types like audio, video, text, and image as given in the table. We used these datasets from various sources like

1. Video data set source: https://www.di.ens.fr/~miech/datasetviz/
2. Audio data set source: https://lionbridge.ai/datasets/voice-and-sound-data-for-machine-learning/
3. Image data set source: https://www.imageannotation.ai/blog/top-10-image-datasets-for-machine-learning
4. Text data set source:
   https://www.kaggle.com/crawford/20-newsgroups
   https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports

**Table 1 Existing Datasets**

| Sl No | Name of the dataset | Type of the data | Type of dataset | No. of instances |
|---|---|---|---|---|
| 1 | HMDB51 | Video | Action Recognition | 6766 |
| 2 | UCF101 | Video | Sports | 13320 |
| 3 | Sports-1M | Video | Sports | 1100000 |
| 4 | Charades | Video | Human Activities | 9848 |
| 5 | ActivityNet | Video | Human Activities | 28000 |
| 6 | Kinetics | Video | Action Recognition | 500000 |
| 7 | Google Audio set | Audio | Human audio from youtube videos | 2,084,320 |
| 8 | Urban sound dataset | Audio | Urban sounds | 8732 |
| 9 | MNIST | Image | Handwritten Images | 70000 |
| 10 | MS COCO | Image | Labelled images | 3,00,000 |
| 11 | 20 Newsgroup | Text | News group | 20,000 |
| 12 | Legal case Reports dataset | Text | Case details | 4000 |

Table 1 represents the name of the datasets, type of those datasets and total number of instances in each data sets which are used to derive the required data sets for our experiment.

**Table 2 Derived Datasets**

| Sl No | Name of the data set | Data sets combined to create new data set | Total No. of instances |
|---|---|---|---|
| 1 | Dataset1 | Sports-1M, MNIST, Google audio set, and legal case reports | 3,258,320 |
| 2 | Dataset2 | HMDB51, UCF101, Charades, Google audio set, MS COCO, and 20 Newsgroup | 2,434,254 |
| 3 | Dataset3 | ActivityNet, Kinetics, Urban sound dataset, MS COCO, and 20 Newsgroup | 856,732 |
| 4 | Dataset4 | Kinetics, Google Audio set, MNIST, 20 Newsgroup, and Legal case Reports dataset | 2,678,320 |

Table 2 represents the details of derived data sets. It specifies the details of existing data sets which are combined to derive these required datasets and representing the total number of instances in each derived datasets.

### A. Time Complexity of Searching Hash Value

The total number of instances specified in the Table2 represents the total number of digital data present in the cloud storage space. We considered four datasets for our experiment. Each dataset contains digital data of different data types like audio, video, image and text. For our experimental purpose, purposefully we combined existing data sets of various data type and created new datasets.

The hash value generated for each digital data while uploading in the cloud storage space will be stored in hash table. If we have single hash table, all the hash values will be stored in the same single hash table. In our implementation, we are considering single hash value for a video instead of creating hash values for each and every frame. If we have multiple hash tables, in the sense dedicated hash table for each type of digital data, then the hash values will be stored separately in the corresponding hash table. Due to this dedicated hash table, the searching space will be decreased drastically and which in turn improves the searching time of hash. The time complexity of searching hash in both the single table and dedicated hash table is O(n), where ‘n’ refers the number of hash value stored in the hash table. In case of single hash table, the total number of instances includes all type of digital data. But dedicated hash tables contains hash values of only those digital data. So ultimately the searching time in single hash table would be very high when compared to the searching time in dedicated hash tables.

**Table 3 Total Number of Hash Entries in Case of Single Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total no. of hash entries in hash table(single) |
|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 3,258,320 |
| 2 | Dataset2 | 2,434,254 | 2,434,254 |
| 3 | Dataset3 | 856,732 | 856,732 |
| 4 | Dataset4 | 2,678,320 | 2,678,320 |

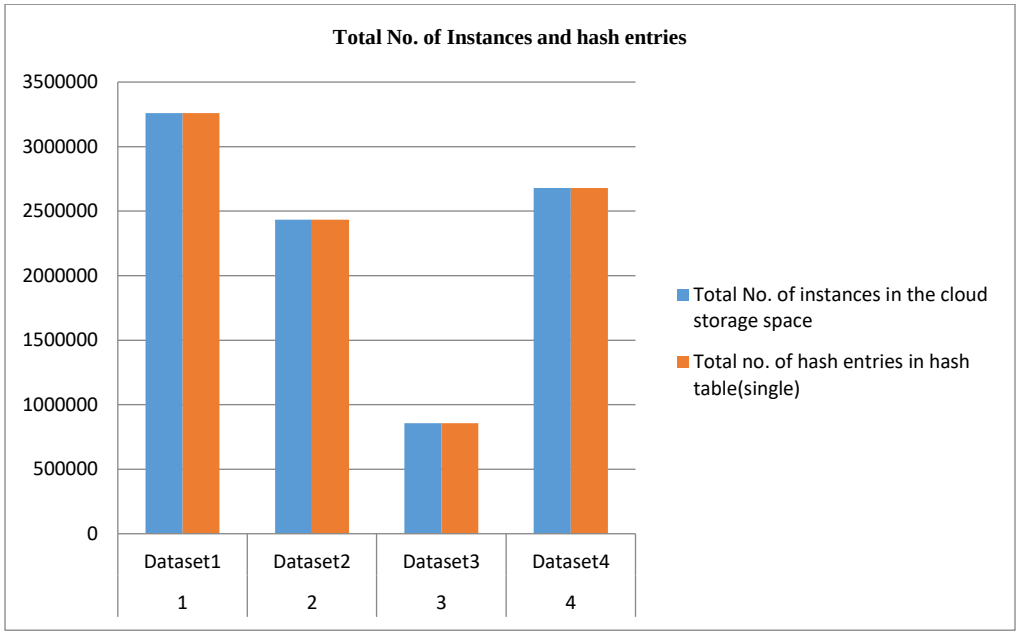


**Figure 4 Total number of instances and hash entries in single hash table**

The total number of hash entries in single hash table will be same as the total number of instances which includes all types of digital data.

## B. Dedicated Hash Table

**Table 4 Total Number of Hash Entries in each Dedicated Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total no. of hash entries in text hash table | Total no. of hash entries in image hash table | Total no. of hash entries in audio hash table | Total no. of hash entries in video hash table |
|---|---|---|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 4000 | 70000 | 2,084,320 | 1100000 |
| 2 | Dataset2 | 2,434,254 | 20,000 | 3,00,000 | 2,084,320 | 29,934 |
| 3 | Dataset3 | 856,732 | 20,000 | 3,00,000 | 8732 | 528000 |
| 4 | Dataset4 | 2,678,320 | 24,000 | 70000 | 2,084,320 | 500000 |

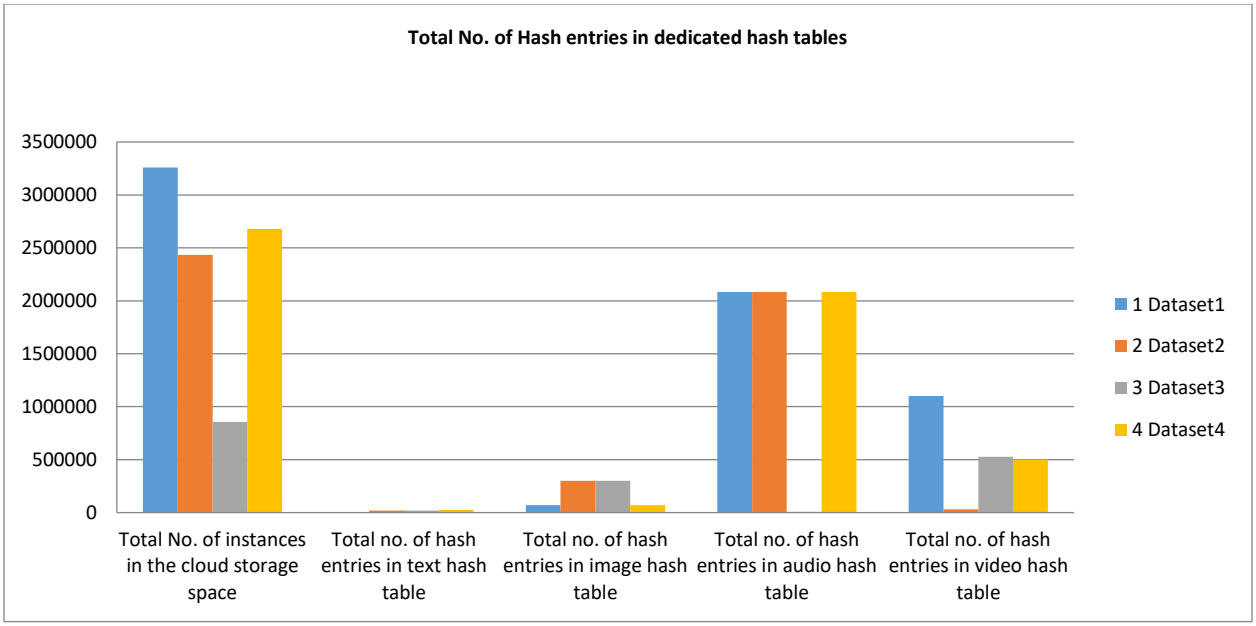


**Figure 5 Total number of hash entries in dedicated hash tables**

The total number of hash entries in each dedicated hash table will be the number of corresponding digital data instances in the dataset.

If we have four new digital data to be uploaded in the cloud storage space, then during file upload the cloud storage provider has to check whether these digital data are already existing in the cloud storage space or not. If they are already present, then CSP should not upload them again in cloud storage space otherwise it has to upload. Out of these four digital data, one is audio, one is video, one is image, and one is text file respectively. The total number of comparison required for checking the duplication in case of single hash table and multiple hash tables for the above mentioned case are derived in the following tables.

## C. Time Complexities with Dedicated Hash Tables

### a. Image Data

**Table 5 Searching Time in Image Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total number of comparison in single hash table | Total number of comparison in dedicated image hash table |
|---|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 3,258,320 | 70000 |
| 2 | Dataset2 | 2,434,254 | 2,434,254 | 3,00,000 |
| 3 | Dataset3 | 856,732 | 856,732 | 3,00,000 |
| 4 | Dataset4 | 2,678,320 | 2,678,320 | 70000 |

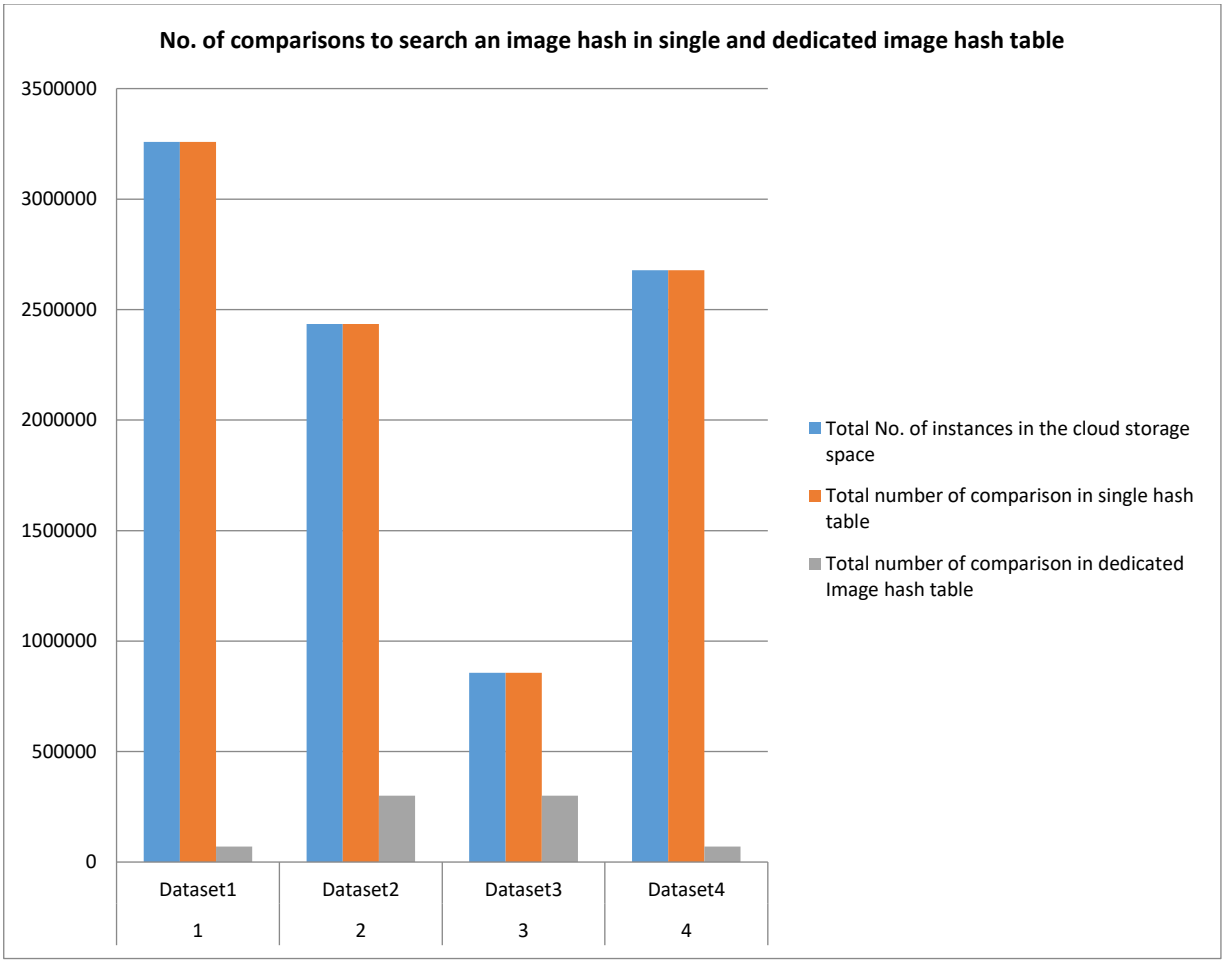


**Figure 6 No. of comparisons required in image hash table**

## b. Video Data

**Table 6 Searching Time in Video Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total number of comparison in single hash table | Total number of comparison in dedicated video hash table |
|---|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 3,258,320 | 1100000 |
| 2 | Dataset2 | 2,434,254 | 2,434,254 | 29,934 |
| 3 | Dataset3 | 856,732 | 856,732 | 528000 |
| 4 | Dataset4 | 2,678,320 | 2,678,320 | 500000 |

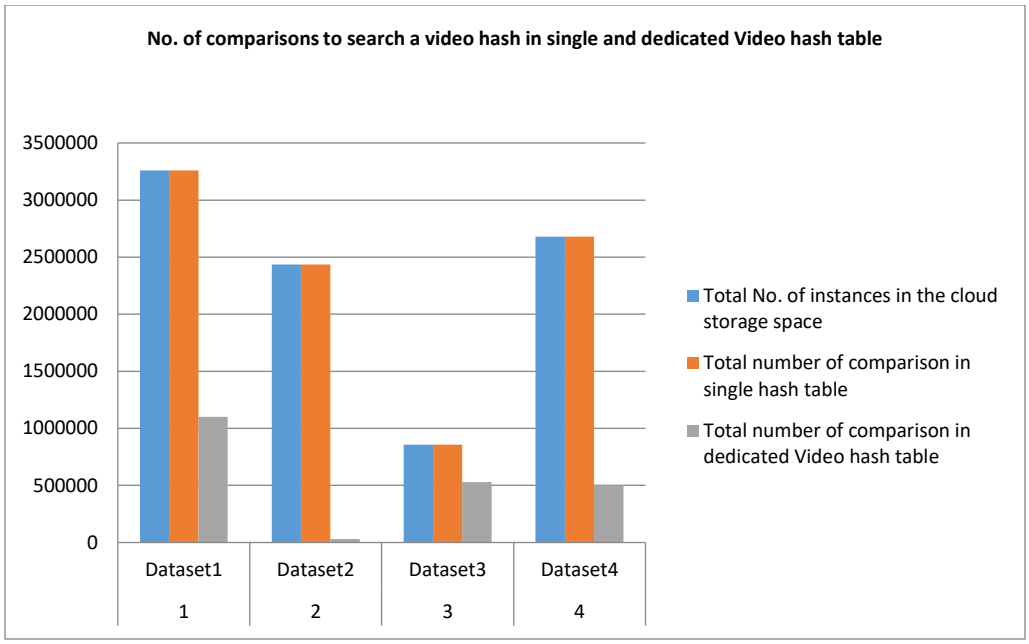


**Figure 7 No. of comparisons required in Video hash table**

## c. Audio Data

**Table 7 Searching Time in Audio Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total number of comparison in single hash table | Total number of comparison in dedicated audio hash table |
|---|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 3,258,320 | 2,084,320 |
| 2 | Dataset2 | 2,434,254 | 2,434,254 | 2,084,320 |
| 3 | Dataset3 | 856,732 | 856,732 | 8732 |
| 4 | Dataset4 | 2,678,320 | 2,678,320 | 2,084,320 |

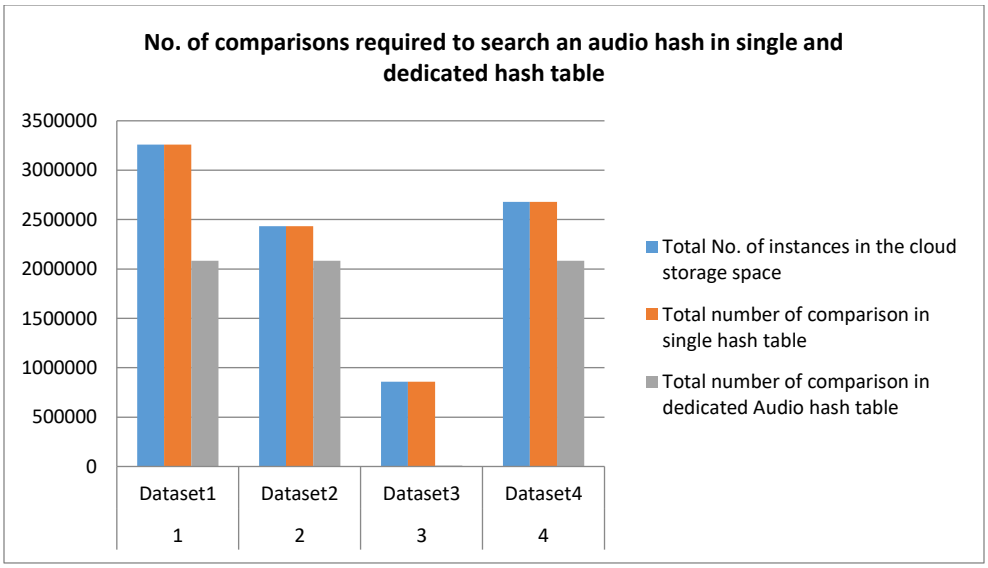


**Figure 8 No. of comparisons required in Audio hash table**

### d. Text Data

**Table 8 Searching Time in Text Hash Table**

| Sl No | Data set | Total No. of instances in the cloud storage space | Total number of comparison in single hash table | Total number of comparison in dedicated text hash table |
|---|---|---|---|---|
| 1 | Dataset1 | 3,258,320 | 3,258,320 | 4000 |
| 2 | Dataset2 | 2,434,254 | 2,434,254 | 20,000 |
| 3 | Dataset3 | 856,732 | 856,732 | 20,000 |
| 4 | Dataset4 | 2,678,320 | 2,678,320 | 24,000 |

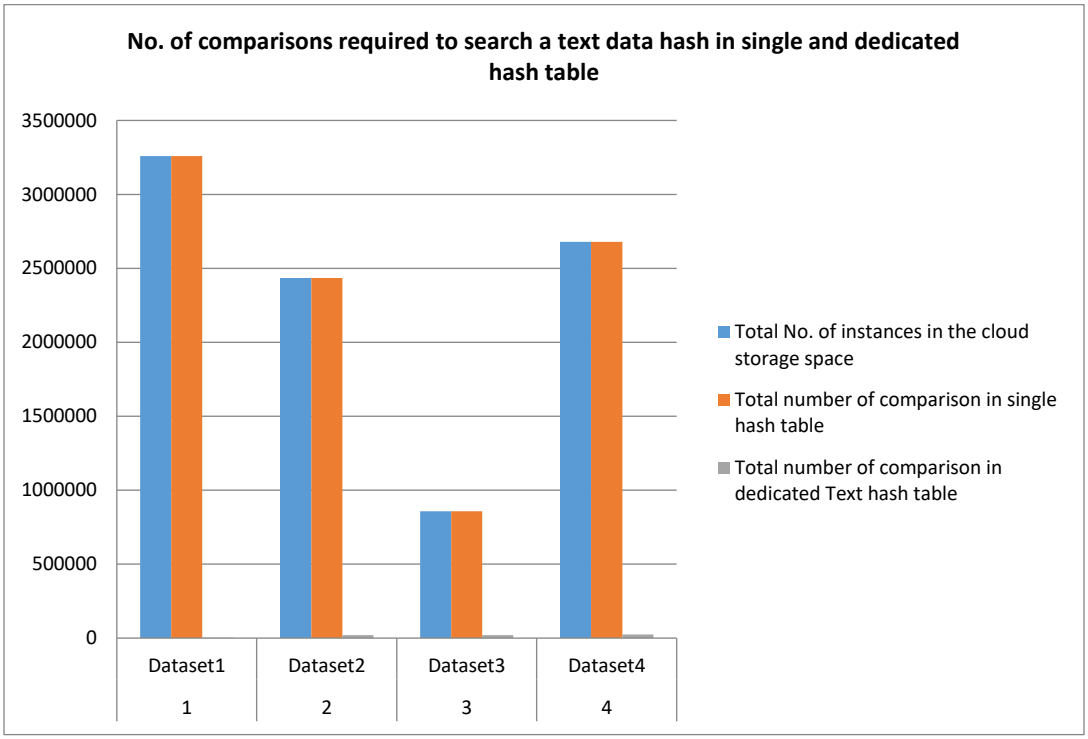


**Figure 9 No. of comparisons required in text hash table**

From the above results, it is clearly understand that when we have dedicated hash table for various digital data type, then the time complexity for finding duplicate copies of digital data is extremely minimized. This will improve the efficiency of deduplication process in the cloud storage system.

## Conclusion

The cloud storage service is utilized by various kinds of users ranging from large organization to individual users. Deduplication is a procedure used to avoid duplicate copies getting stored in the storage space. In hash-based deduplication, deduplicate copies are identified with the help of hash values of digital data. But the time complexity of

finding duplicate copies added overhead to the users when there is a single hash table for all kind of digital data. In this research work, we suggested dedicated hash tables for each digital data type. The experimental results proved that the time requirement is heavily reduced because of these dedicated hash tables. The time complexity can be further reduced by having suitable hash table structure for each digital data type. Our future research work is to identify the suitable hash table structure for various digital data.

## References

Kamala Kannan, T., Sharmila, K., Shanthi, M.C., & Devi, M.R. Study on Cloud Storage and its Issues in Cloud Computing. *International Journal of Management, Technology and Engineering*, *9*(1), 976-981.

Rajan, A.P. (2012). Evolution of cloud storage as cloud computing infrastructure service. *IOSR Journal of Computer Engineering (IOSRJCE)*, *1*(1), 38-44.

Tang, Y., Yin, J., & Wu, Z. (2016). Try Managing Your Deduplication Fine-Grained-ly: A Multi-tiered and Dynamic SLA-Driven Deduplication Framework for Primary Storage. *In IEEE 9th International Conference on Cloud Computing (CLOUD*, 859-862.

Rao, B.T. (2016). A study on data storage security issues in cloud computing. *Procedia Computer Science*, *92*, 128-135.

Wu, P.J., & Lin, K.C. (2018). Unstructured big data analytics for retrieving e-commerce logistics knowledge. *Telematics and Informatics*, *35*(1), 237-244.

Pritha, N.L., Velmurugan, N., Winster, S.G., & Vijayaraj, A. (2015). Deduplication based storage and retsrieval of data from cloud environment. *In International Conference on Innovation Information in Computing Technologies*, 1-6.

Paulo, J., & Pereira, J. (2014). A survey and classification of storage deduplication systems. *ACM Computing Surveys (CSUR)*, *47*(1), 1-30.

Maruti, M.V., & Nighot, M.K. (2015). Authorized data Deduplication using hybrid cloud technique. *In International Conference on Energy Systems and Applications*, 695-699.

Zhang, Z., Jiang, Z., Liu, Z., & Peng, C. (2012). LHS: A novel method of information retrieval avoiding an index using linear hashing with key groups in deduplication. *In International Conference on Machine Learning and Cybernetics*, *4*, 1312-1318.

Ni, F., Wu, X., Li, W., Wang, L., & Jiang, S. (2019). WOJ: Enabling Write-Once Full-data Journaling in SSDs by Using Weak-Hashing-based Deduplication. *ACM SIGMETRICS Performance Evaluation Review*, *46*(3), 39-40.

Du, L., Ho, A.T., & Cong, R. (2020). Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, *81*, 115713.

Ponnusamy, V., Kottursamy, K., Karthick, T., Mukeshkrishnan, M.B., Malathi, D., & Ahanger, T. A. (2020). Primary user emulation attack mitigation using neural network. *Computers & Electrical Engineering*, *88*, 106849.

Goyal, L., Sharma, K., Rai, T., Gupta, B., & Arora, N. Left-Right Mid Way Linear Searching Algorithm. *International Journal of Computer Applications*, *975*, 8887.

Parmar, V.P., & Kumbharana, C.K. (2015). Comparing linear search and binary search algorithms to search an element from a linear list implemented through static array, dynamic array and linked list. *International Journal of Computer Applications*, *121*(3), 13-17.