

# Deduplication Schemes for Cloud Backup

Vikram Vivek Badge

J. S. P. M, Tathawade, Pune, Savitribai Phule Pune University, India

**Abstract:** *It is observed that increasing use of cloud storage environment for data backup is increasing the challenge of performing Deduplication due to combination of resource intensive nature of the deduplication and limitation of the system resources. In this paper, a review of different Deduplication schemes such as ALG, SAM and AA-Dedupe is done. By this an attempt has been made to find good balance between cloud storage capacity saving and deduplication time reduction. This paper studies the deduplication techniques that follow fingerprinting based approach.*

**Keywords:** Deduplication, backup window, Local source deduplication, Global source deduplication, Indexing.

## 1 Introduction

The Cloud computing technology consumes significant IT resources in order to provide the customers with different types of services and backup facility. Thus different challenges arise in cloud backup services. Some challenges are discussed as follows [2]. One of the main challenges is large backup window, due to the low network bandwidth between user and service provider constraining the data transmission. The backup window is represented by the time spent on sending specific dataset to backup destination. For example, it would take more than 14 days to backup 1TB data to Amazon S3 [15] with the assumed network bandwidth of 800KB/s [16]. The rapid increase in the amount of backed-up data stored at service providers' site also demands for the vast storage space and very high data management cost.

Data deduplication describes a class of approaches that reduce the storage capacity needed to store data or the amount of data that has to be transferred over a network. The process of data deduplication is an effective data compression approach that exploits data redundancy by partitioning large data objects into smaller parts called chunks. The chunks are represented by their fingerprints, replace the duplicate chunks with their fingerprints after chunk fingerprint index lookup and only transfers or store the unique chunks for the purpose of communication or storage efficiency. [1]

To implement the data deduplication there are two approaches [8]. The first approach is fingerprinting based and the second approach is delta-based data deduplication. In fingerprinting based approach all the chunks are fingerprinted using a cryptographic fingerprinting method and a chunk index is maintained containing the fingerprint of all already stored chunks. If the fingerprint of a chunk can be found in the chunk index, it is assumed that the data of the already stored chunk and the currently processed chunk are identical and the chunk is classified as already stored.

The Delta-based data deduplication approach performs the chunking, but it is not searching similar, but necessarily identical data blocks. When a new chunk in the delta-based data deduplication system is written, similar chunks are searched and then store only a differential encoding

between the new data and chunks that have been found to be similar

On the basis of location where redundant data is eliminated, the existing source deduplication is categorized into local source deduplication [4], [13], [14] that only detects redundancy in backup dataset from the same device at the client side and only sends the unique data chunks to the cloud storage, and global source deduplication [9], [10] that performs duplicate check in backup datasets from all clients in the cloud side before data transfer over Wide Area Network.

The data deduplication is a resource-intensive process, which involve the CPU-intensive hash calculations for fingerprinting and the I/O-intensive operations to identify and eliminate redundant data. Unfortunately, such resources are limited in a typical personal computing device. Therefore, it is desirable to achieve an optimal tradeoff between deduplication effectiveness and deduplication overhead for personal computing devices with limited system resources.

## 2 Why There is Need of Data Deduplication

Consider the effect of a weekly full file server backup with a five week retention policy, the majority of the content is protected five times over [12]. Now consider an email server that is protected with a nightly full backup – much of the content is that mail server is backed up 35 times during those five weeks.

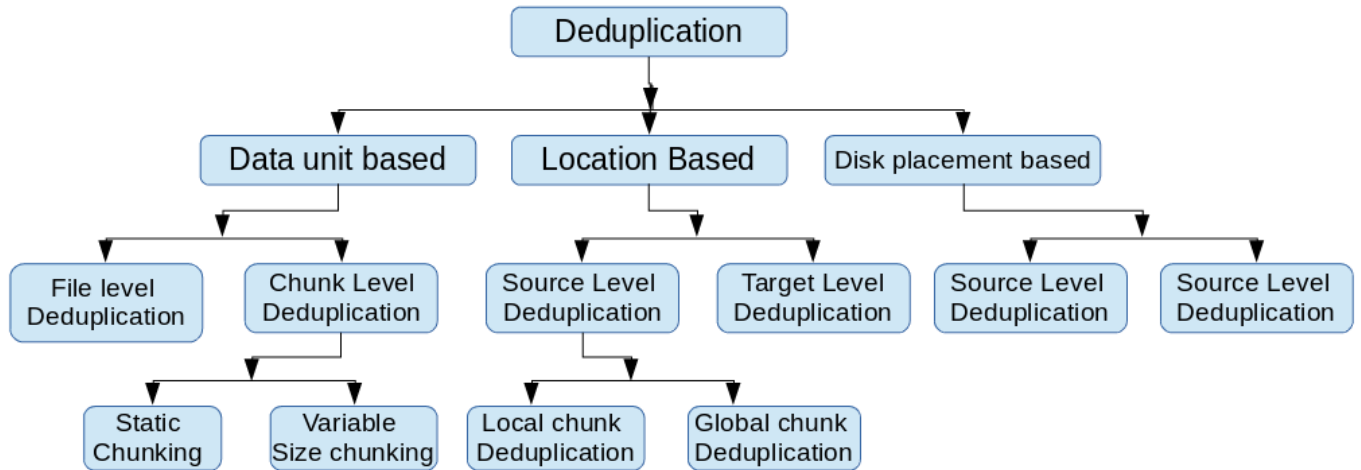
The traditional backup solutions require a rotational schedule of full and incremental backup, which move a significant amount of redundant data every week. Most organizations also create a second copy of this information to be shipped to a secondary site for disaster recovery purposes. Thus aggregating, the costs of traditional backup in terms of bandwidth, storage infrastructure, and time increases the cost of IT organizations for information management.

Backing up of redundant files and data increases the backup window size this results in over utilization of

Network resources, and require too much additional storage capacity to hold unnecessary backup data.

The organizations need solutions to manage this increasing information and data. Thus deduplication techniques can reduce your bandwidth requirements; it can improve the data transfer speed and maintain your cloud storage needs including cloud storage fees to a minimum.

### 3 Strategies in Deduplication



**Figure 1:** Strategies in Deduplication

The classification of data deduplication strategies is done in to following three types [6]. The First type of classification is based on data units. The Second classification is done on Location where deduplication can be performed. The Third classification is Disk placement as depicted in Fig. 1.

#### 3.1 Data unit based

Here Data duplication strategies are basically classified in to File level deduplication and Block (chunk) level deduplication.

In File level deduplication only one copy of the file is stored. Two files are identical if they have the same hash value.

On the other hand file is fragmented into blocks in Block level Deduplication and one copy of each block is stored. Each block may be fixed (static) or variable size chunk. In fixed size chunks, size of each block is same. In case of variable, size of each chunk is varies.

#### 3.2 Location based

Deduplication can be categorized in to two basic approaches depending on the location where redundant data is to be eliminated [6]. In the target based approach, Deduplication is performed in the Destination storage system. Here client is not aware about strategies in deduplication. The positive part of this method is storage utilization increases but bandwidth is not saved.

The elimination of duplicate data is performed closed to where data is created in source based deduplication. The

source deduplication approach is implemented at the client side.

The client software communicates with the backup server by sending hash signatures to check for the existence of files or block. The duplicate are replaced by pointers and the actual duplicate data is never sent over the network.

The Source Deduplication Method [11] is further classified based on different deduplication granularities as source local chunk level deduplication; source global chunk level deduplication [6] [14]. In the local chunk level, the redundant data chunks are removed before sending them to the remote backup destination within the same client. In the global chunk level, the duplicate chunk is removed globally across different clients.

#### 3.3 Disk Placement based deduplication

Backward reference deduplication and Forward reference deduplication are two major classifications in Disk placement. In backward reference the recent redundant data chunks are associated with pointers that point backward to the older identical data chunks. In case of forward reference deduplication the recent redundant data chunks are maintained in their entirety and all the old identical data chunks are associated with pointers that point forward to the recent data chunks.

### 4 Deduplication and Backup Approaches to Cloud

#### 4.1 CUMULUS: File system Backup to the Cloud

In Cumulus system the file system backups could be

efficiently implemented over the Internet [16]. This system was specifically designed under a thin cloud, assuming that the remote data center storing the backups does not provide any special backup services. In Cumulus aggregation of data from small files is done for storage, LFS- inspired segment cleaning is used to maintain storage efficiency.

But the approach embodied by Cumulus is for the client making a backup to do most of the work, and leave the backup itself almost entirely opaque to the server. Cumulus, as designed, does not offer coordination between multiple backup clients, and so does not offer features such as deduplication between backups from different clients.

#### 4.2 Application Driven Metadata Aware Deduplication

In this use of certain metadata information of different levels in the I/O path is used [17]. This metadata information is used to direct the file partitioning into more Meaningful data Chunks (MC) so that the inter-file level duplication is maximally reduce.

These meta-data information of archived files may include:

1. Application metadata, such as file type, file format, application software, etc.
2. Application or user tags, such as the tags used to describe the characteristics of blogs, images or multimedia,
3. File system metadata, such as directory entries, inode information of a file.

The file type and file format is used as the metadata information in order to direct the file partitioning besides the currently used cryptographic hash functions, such as MD5, SHA1 and chunking methods such as Rabin fingerprinting. Thus an attempt to improve redundancy detection is done by application-specific chunking methods that exploit the details about file formats.

#### 4.3 An Application aware framework approach for Deduplicating videos

It is a frame-work for video deduplication based on an application-level view of redundancy at the content level instead of the byte level [5]. But all these prior work only focus on the effectiveness of deduplication to remove more redundancy in the existing information. But here system overhead for high efficiency is not considered in process of deduplication in the above mentioned scheme.

#### 4.4 A Semantic aware Multi-tiered source deduplication framework

In this researchers developed a technique called SAM [2], a semantic Aware Multi tired source deduplication system. This is a combination of global file level deduplication and the local chunk level deduplication. To find the redundant data they have considered file level semantic such as file locality, file time stamp, file size and file type. Work in

synchronized way to remove the redundant data from data transmission to reduce backup times and storage cost.

According to its architecture it has three sub systems File agent, Master Server and Storage Server. The File Agent is a software program that provides a functional interface to the users. The main task of file agent is to gather datasets and send or restore them to / from storage server for backups or restores. Master Server globally manages and schedules all backups and restores jobs. Catalog database is used by master server to keep track of which file is stored on which storage servers. The Depository for storage data is the backup server. Actual data backup and restore jobs are performed by Storage Server along with File Agent under the direction of the Master Server. Storage server sends Meta data information including file Meta data and job Meta data to Master Server. Master Server stores this Meta data information in catalog database for backed-up data indexing and retrieving. Chunk Store consists of a large number of chunk containers; it is responsible for storing the backed-up data chunks from different clients.

The main drawback in the SAM is client overhead as it uses more CPU power and storage space at the client and restore performance cannot be calculated as real time data sets are not considered.

#### 4.5 Application based Source Deduplication Scheme

AA Dedupe is an application aware source deduplication scheme for the cloud backup in the personal computing environment to improve deduplication efficiency [3]. In this the computational overhead is reduced by intelligent data chunking technique and by using the hash function based on application awareness.

In AA Dedupe architecture tiny files are first filtered out by file size [7]. Then the backup data streams are broken into chunks by an intelligent chunker using an application aware strategy. After that by looking up the hash values of the data chunks in an application aware index which is stored in the local disk, data chunks from the same type of file are deduplicated in application aware deduplicator. If the match of Hash value is found the metadata for the file containing that chunk is updated to point to the location of the existing chunk. If there is no match of hash value of the chunk, it is considered as the new chunk and the new chunk is stored base on the container management in the cloud. The metadata of the associated file of the chunk is updated int to it and the new entry is added into the application aware index to index the new chunk.

AA-Dedupe are primarily attributed to its application awareness in the deduplication process. The deduplication efficiency of AA-Dedupe is 5 times that of SAM. Considering the Energy Efficiency in AA-Dedupe by using weaker hash function in deduplication it incurs only one fourth of the power consumption of Avamar and one third of SAM.

#### 4.6 ALG scheme

It's an Application-aware Local-Global source deduplication scheme in this data deduplication efficiency is improved by exploiting application awareness and also by combining local and global duplicate detection technique [1]. Here the attempt is made to maintain a good balance between saving cloud storage capacity and reducing deduplication time.

The main idea of ALG Dedupe is:

- a. Exploiting both low-overhead local resources and high-overhead cloud resources to reduce the computational overhead by using an intelligent data chunking technique and by using hash functions based on application awareness.
- b. To reduce the on-disk index lookup bottleneck splitting the full index into small independent and application-specific indices in an application-aware index structure.

It is a combination of local-global source deduplication with application awareness which improves effectiveness of deduplication with low system overhead at the client side.

ALG dedupe consist a file size filter where tiny files are first filtered out by for efficiency reasons, and backup data

Streams are broken into chunks by an intelligent chunker using an application aware chunking strategy [1]. Data chunks from the same type of files are then deduplicated in the application-aware deduplicator by generating chunk fingerprints in hash engine and performing data redundancy check in application-aware indices in both local client and remote cloud. First their fingerprints are looked up in an application-aware local index that is stored in the local disk for local redundancy check.

If a match is found, the Meta data for the file containing that chunk is updated to point to the location of the existing chunk. In case of no match, the fingerprint will be sent to the cloud for further parallel global duplication check on an application aware global index, and then if a match is found in the cloud, the corresponding file Meta data is updated for duplicate chunks, or else the chunk is considered new.

To improve the network bandwidth efficiency over WAN, at client side the fingerprints will be transferred in batch and new data chunks will be packed into large units called segments in the segment store module with tiny files before their transfers to reduce cloud computing latency.

The local duplicate detection in ALG Dedupe significantly reduces the number of global fingerprint lookup requests. Thus it can perform comparatively well as compared to SAM, and AA dedupe approach

ALG-Dedupe are shown to improve the deduplication efficiency with very low system overhead. It has reduced the backup window size and improves power-efficiency.

The cloud backup service cost is also saved with this deduplication approach.

## 5 Benefits of Deduplication

Reduction in Backup window size does not affect data transfer in case of low bandwidth. Data transfer time is clearly decreases by application-aware source deduplication so recovery time required in case of failure is less.

The existing approaches acquire heavy power consumptions due to their computational overhead during the deduplication process or high data transfer overhead due to low space saving. But the ALG scheme is more energy efficient as compared to other schemes discussed above.

The cost of using cloud services and performing cloud backup is reduced significantly due to deduplication techniques. Thus IT organizations can make proper utilization of their server space by implementing the above discussed Deduplication Schemes.

## 6 Conclusion

Deduplication technology can accelerate backup efficiency and drive down IT costs. The implementation of Deduplication technique at client side, servers and also data centers will reduce the redundant data to much extent. Also the storage space will be utilized wisely by saving the only the unique data. Thus the availability of data and proper utilization of storage space can be managed with the Deduplication schemes.

## Acknowledgments

I would like to thank my guide Dr. P. K. Deshmukh for his help and guidance throughout this project and the semester, without them this would not have been possible.

## References

- [1] Yinjin Fu, Hong Jiang, Senior Member, IEEE, Nong Xiao, Member, IEEE, Lei Tian, Fang Liu, and Lei Xu, "Application-Aware Local-Global Source Deduplication for Cloud Backup Services of Personal Storage", IEEE Transactions on Parallel and Distributed Systems, VOL. 25, NO. 5, MAY 2014
- [2] Y. Tan, H. Jiang, D. Feng, L. Tian, Z. Yan, and G. Zhou, "SAM: A Semantic-Aware Multi-Tiered Source De-Duplication Framework for Cloud Backup," in Proc. 39th ICPP, 2010, pp. 614-623. BackupPC, 2011. [Online]. Available: <http://backupper.sourceforge.net/>
- [3] Y. Fu, H. Jiang, N. Xiao, L. Tian, and F. Liu, "AA-Dedupe: An Application-Aware Source Deduplication Approach for Cloud Backup Services in the Personal Computing Environment," in Proc. 13th IEEE Int'l Conf. CLUSTER Comput., 2011, pp. 112-120.
- [4] P. Anderson and L. Zhang, "Fast and Secure Laptop Backups with Encrypted De-duplication," in Proceedings of the 24th international conference on

- Large Installation System Administration (LISA'10), 2010, pp. 29-40.
- [5] A. Katiyar and J. Weissman, "ViDeDup: An Application-Aware Framework for Video De-Duplication," in Proc. 3rd USENIX Workshop Hot-Storage File Syst., 2011, pp. 31-35.
- [6] D. Harnik, B. Pinkas, A. Shulman-Peleg, "Side channels in cloud services, the case of deduplication in cloud storage", IEEE Security Privacy, 2010, 8: 40-47.
- [7] P. Neelaveni and M. Vijayalakshmi, "A Survey on Deduplication in cloud storage", Asian Journal of Information Technology 19(6): 320-330, 2014.
- [8] D. Meister, "Advanced Data Deduplication Techniques and their Application", 2013.
- [9] D. Meister and A. Brinkmann, "Multi-level comparison of data deduplication in a backup scenario," in Proceedings of the 2nd Annual International Systems and Storage Conference (SYSTOR'09), Haifa, Israel: ACM, 2009.
- [10] P. Kulkarni, F. Douglis, J. Lavoie, and J. M. Tracey, "Redundancy elimination within large collections of files," in Proceedings of the annual conference on USENIX Annual Technical Conference (ATC'04), 2004, pp. 59-72.
- [11] Zhu, B., K. Li and H. Patterson, "Avoiding Disk bottleneck in the data domain deduplication file system. Proceedings of the 6<sup>th</sup> USENIX Conference on File and Storage Technologies, February 26-29, 2008, San Joes, CA. USA., pp: 269-282.
- [12] www.EMC.com, "An EMC Perspective on Data De-Duplication for Backup".
- [13] L. D. Bois and R. Amatruda, "Backup and Recovery: Accelerating Efficiency and Driving Down IT Costs Using Data Deduplication," EMC Corporation, Feb. 2010.
- [14] D. T. Meyer and W. J. Bolosky, "A Study of Practical Deduplication," in Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST'11), 2011, pp. 1-14.
- [15] Amazon Simple Storage Service, "http://aws.amazon.com/s3."
- [16] M. Vrable, S. Savage, and G. M. Voelker, "Cumulus: File system Backup to the Cloud," in FAST'09, Feb. 2009.
- [17] C. Liu, Y. Lu, C. Shi, G. Lu, D. Du, and D.-S. Wang, "ADMAD: Application-Driven Metadata Aware De-Deduplication Archival Storage Systems," in Proc. 5th IEEE Int'l Workshop SNAPI I/Os, 2008, pp. 29-35.

## Author Profile



**Vikram Vivek Badge** has completed his Bachelor of Engineering in Computer Science Engineering from KBP college of Engineering Satara, Shivaji University. He is pursuing ME in computer engineering from Savitribai Phule Pune University.