iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Duplicate Finder: Application Aware Data Protection in the Personnel Computing Environment Using Cloud Backup Service

Miss. S.Sathya Priya[1] M.C.A, Mrs. A.Kavitha.M.phil [2]
[1]*PG Scholar,* [2]*Assistant Professor*
Department Of Computer Science Engineering, Kalaignar Karunanidhi Institute of Technology

Abstract- In widespread cloud environment, cloud data storage services are tremendously growing due to large amount of personal computation data. Data deduplication, an effective data compression approach that exploits data redundancy, partitions large data objects into smaller parts, called chunks, represents these chunks by their fingerprints. A cloud storage environment for data backup in personal computing devices facing various challenges in source deduplication based on the cloud backup services with low deduplication efficiency. Challenges facing in the process of deduplication for cloud backup service are-1)Low deduplication efficiency due to exclusive access to large amount of data and limited system resources of PC based .2)Low data transfer efficiency due to transferring deduplicate data from source to backup server. In this paper, we propose a Duplicate finder, an application aware deduplication detection technique modelled in terms of schematic representation to exploit the duplicate in the data storage through aggregation detector in order to reduces the latency and storage cost . Experimental results proves that proposed technique outperforms state of approaches both in efficiency which duplicate elimination ratio and communication bandwidth management which aids for less data transfer
Keywords: Cloud computing, Deduplication, cloud backup, application awareness

I. 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. 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. Cloud backup service has become a cost-effective choice for data protection of personal computing devices ,since the centralized cloud management has created an efficiency and cost inflection point, and offers simple offsite storage for disaster recovery, which is always a critical concern for data backup. And the efficiency of IT resources in the cloud can be further improved due to the high data redundancy in backup dataset. Deduplication is classified into source(local) and target (Global) deduplication, Source deduplication that eliminates redundant data at the client site is obviously preferred to target deduplication due to the former's ability to significantly reduce the amount of data transferred over wide area network (WAN) with low Communication bandwidth [1]. 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 However, data deduplication is a resource-intensive process, which entails the CPU-intensive hash calculations for chunking and fingerprinting and the I/O-intensive operations for identifying and eliminating duplicate data. But all these prior work only focus on the effectiveness of deduplication to remove more redundancy without consider the system overheads for high efficiency in deduplication process. The remainder of this paper is organized as follows: We analyse the related works in Section 2 and proposed system modelling in Section 3. We describe duplicate finder by comparing it with the existing state-of-the-art schemes in Section 4. We conclude with remarks on future work in Section 5.

II. RELATED WORKS

©IJRASET 2015: All Rights are Reserved

A. Application-Driven Metadata Aware De-Deduplication Archival Storage Systems

ADMAD [2] improves redundancy detection by application- specific chunking methods that exploit the knowledge about concrete file formats. application-driven metadata aware de-duplication archival storage system, which makes use of certain meta-data information of different levels in the I/O path to direct the file partitioning into more meaningful data chunks (MC) to maximally reduce the inter-file level duplications. However, the chunks may be with different lengths and variable sizes, storing them into storage devices may result in a lot of fragments and involve a high percentage of random disk accesses, which is very inefficient.

B. An Application-Aware Framework For Video De-Duplication

ViDeDup [3] is a frame-work for video deduplication based on an application-level view of redundancy at the content level rather than at the byte level. Intuitively, introducing application-level intelligence in redundancy detection can yield improved data compression. We propose ViDeDup (Video De-Duplication), a novel framework for video de-duplication based on an application-level view of redundancy. The framework goes beyond duplicate data detection to similarity-detection, thereby providing application-level knobs for defining acceptable level of noise during replica detection. Our results show that by trading CPU for storage, a 45% reduction in storage space could be achieved, in comparison to 8% yielded by system level de-duplication for a dataset collected from video sharing sites on the Web.

C. ALG-Dedupe –Application Aware Deduplication Scheme

ALG-Dedupe, an Application-aware Local-Global source deduplication scheme that improves data deduplication efficiency by exploiting application awareness, and further combines local and global duplicate detection to strike a good balance between cloud storage capacity saving and deduplication time reduction.

III. PROPOSED SYSTEM – DUPLICATE FINDER

In this section, we will analyse and model a data redundancy elimination mechanism, space utilization efficiency of popular data chunking methods and computational overhead of typical hash functions change in different applications of personal computing.

A. Modelling Data Compression Approach

Classifying the large amount of data objects into smaller parts called chunks represents these chunks by their fingerprints. Chunk fingerprint Index lookup is constructed in order to avoid the duplicate chuck. Transfers or stores the unique chunks for the purpose of communication or storage efficiency. However, data deduplication is a resource-intensive process, which entails the CPU-intensive hash calculations for chunking and fingerprinting and the I/O-intensive operations for identifying and eliminating duplicate data. tiny files are first filtered out by file size filter for efficiency reasons, and backup data streams are broken into chunks by an intelligent chunker using an applicationaware chunking strategy. 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. Their fingerprints are first 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 metadata for the file containing that chunk is updated to point to the location of the existing chunk. When there is 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 metadata is updated for duplicate chunks, or else the chunk is new.

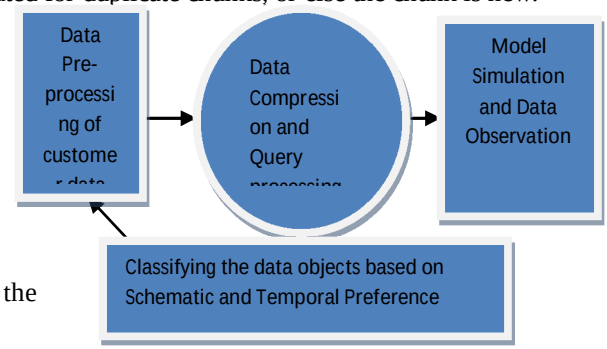


Figure 3.1 Architecture diagram of the Proposed System

©IJRASET 2015: All Rights are Reserved

B. Application-Aware Deduplicator

After data chunking in intelligent chunker module, data chunks will be deduplicated in the application-aware deduplicator by generating chunk fingerprints in the hash engine and detecting duplicate chunks in both the local client and remote cloud.

We can define a metric for deduplication efficiency to balance the cloud storage cost saving and backup window shrinking in source deduplication based cloud backup services. It is well understood that the deduplication efficiency is proportional to deduplication effectiveness that is always defined by duplicate elimination ratio and inversely proportional to the average backup window size per chunk BWS with average chunk size C. Based on this understanding and to better quantify and compare deduplication efficiency of a wide variety of deduplication techniques, we propose a new metric, called ‘‘bytes saved per second,’’ To achieve high deduplication efficiency, the application naware deduplicator first detects duplicate data in the application-aware local index corresponding to the local datasetwith low deduplication latency in the PC client, and then compares local deduplicated data chunks with all data stored in the cloud by looking up fingerprints in the application-aware global index on the cloud side for high data reduction ratio.

C. Application-Aware Index Structure

It consists of an in- RAM application index and small hash-table based on-disk indices classified by application type. According to the accompanied file type information, the incoming chunk is directed to the chunk index with the same file type. Each entry of the index stores a mapping from the fingerprint (fp) of a chunk or with its length (len) to its container ID (cid).As chunk locality exists in backup data streams , a small index cache is allocated in RAM to speedup index lookup by reducing disk I/O operations. The index cache is a key-value structure, and it is constructed by a doubly linked list indexed by a hash table. When the cache is full, fingerprints of those containers that are ineffective in accelerating chunk fingerprint lookup are replaced to make room for future prefetching and caching.

IV. EXPERIMENTAL RESULTS

In this section, we have modelled the data compression technique with creation of different domains for each file-type by horizontal partitioning of chunk fingerprints to improve overall throughput with parallel fingerprint lookup.

We compare Duplicate finder with ALG-Dedupe and against a number of state of- the-art schemes. By leveraging application awareness to achieve global chunk-level deduplication effectiveness, it also outperforms ALG –dedupe that combines local chunk-level and global file-level source deduplication. The high effectiveness of data deduplication of the fine-grained or global deduplication schemes comes at a significant overhead that throttles the system throughput. The backup window represents the time spent on sending a backup dataset to cloud storage, which mainly depends on the volume of the transferred dataset and available network bandwidth. Though the cloud latency affects global deduplication, our local duplicate detection can significantly reduces the number of global fingerprint lookup requests. And data transfer time is sharply decreased by our application-aware source deduplication even though the upload bandwidth is low in WAN. Application-aware deduplication has a potential to improve the efficiency of deduplication by eliminating redundancy in each application independently and in parallel. To price cloud-based backup services attractively requires minimizing the capital costs of data enter storage and the operational bandwidth costs of shipping the data back and forth.

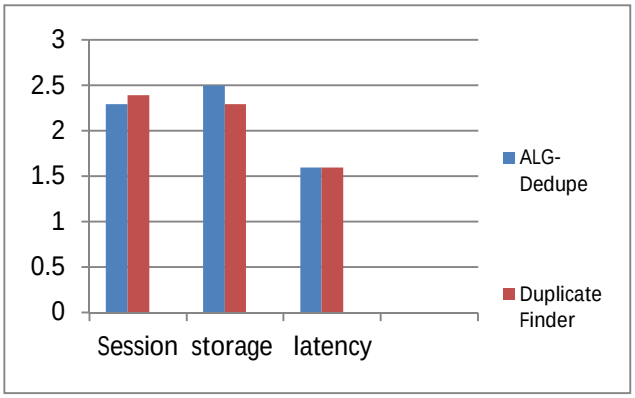


Figure 4.1 Experiment Results of the Methodology

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue IV, April 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Figure 4.2 explains the performance of the methodology with respect to the Session and Storage and latency. The Schematic and Semantic representation of the data chuck has yielded better results.

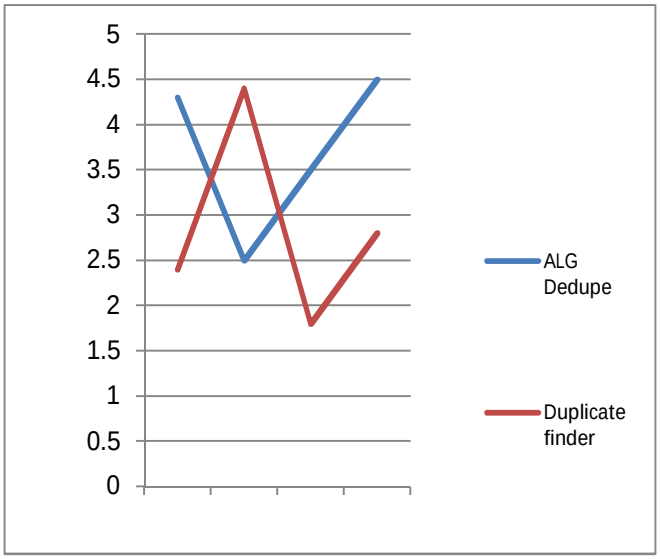


Figure 4.2 Experiment results of the Duplicate elimination ratio

The figure 4.2 depicts the performance of the data compression efficiency in terms of duplicate elimination ratio for methodology schemes.

Duplicate Elimination ratio

The formula to estimate the DER for the cloud backup window

DER = 1/ BWS *C

Where

BWS = Backup window size per chunk

C = chuck size

To discover high chunk-level redundancy, we need to choose chunking method and chunk size to strike a good balance between the capability of redundancy discovery and the deduplication overhead. To balance cloud storage cost saving and backup window shrinking in these two schemes, we choose local-global source deduplication, which reduce the backup window size by exploiting local resources to reduce deduplication latency and save cloud storage cost by leveraging cloud resources to improve deduplication effectiveness.

V. CONCLUSION

We have proposed and implemented a Duplicate finder, an application aware deduplication scheme for cloud backup in the personal computing environment to improve deduplication efficiency. An intelligent deduplication strategy is designed to exploit file semantics to minimize computational overhead and maximize deduplication effectiveness using application awareness. It groups the deduplication techniques like source deduplication and global deduplication to balance the effectiveness and latency of deduplication. The proposed application-aware index structure can significantly relieve the disk index lookup bottleneck by dividing a central index into many independent small indices to optimize lookup performance.

VI. FUTURE WORK

As a future work, we plan to further optimize our scheme for other resource-constrained methodology by using supervised learning models and investigate the secure deduplication issue in cloud backup services of the personal computing environment.

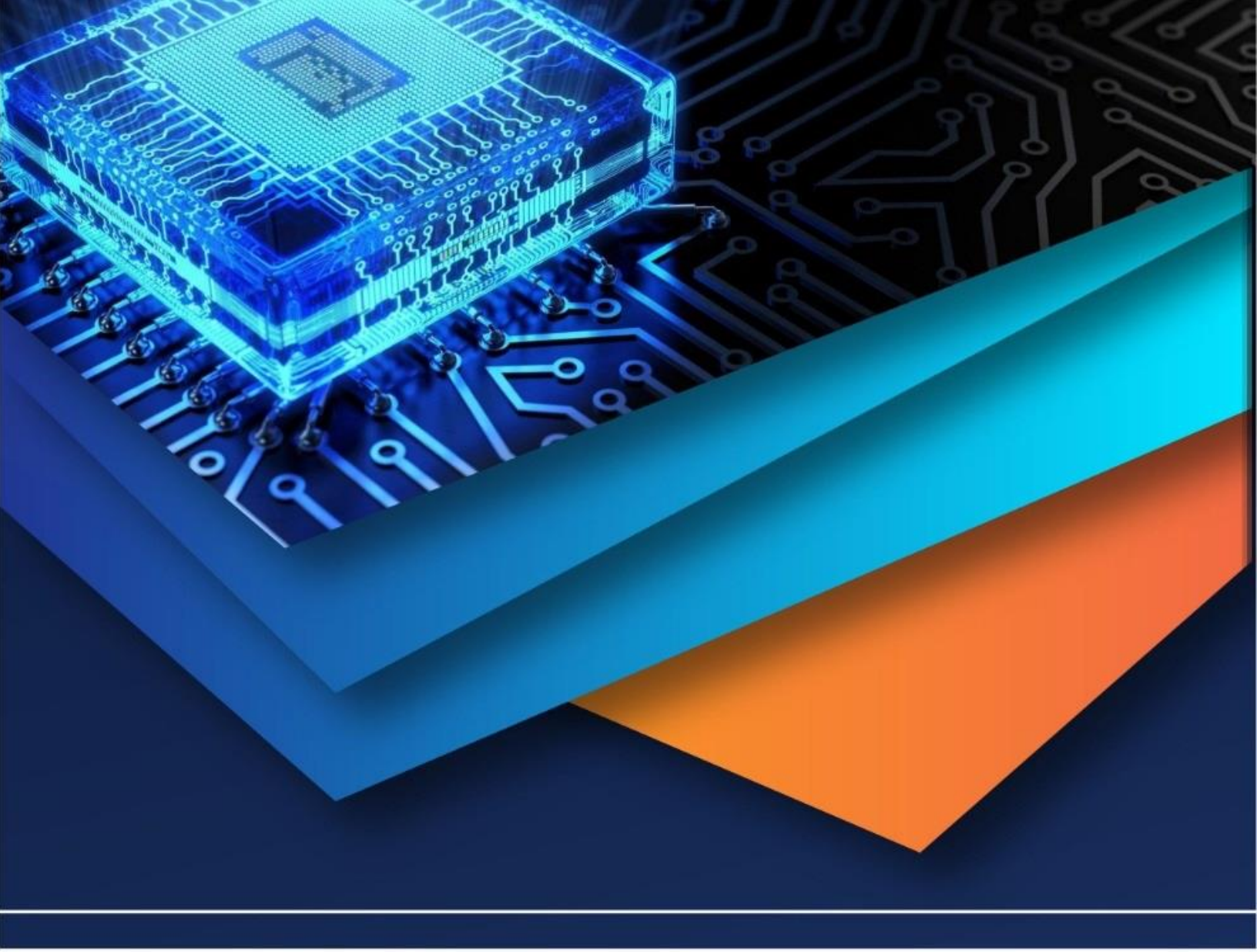
REFERENCES

[1] P. Shilane, M. Huang, G. Wallace, and W. Hsu, ‘‘WAN Optimized Replication of Backup Datasets Using Stream- Informed Delta Compression,’’ in Proc. 10th

©IJRASET 2015: All Rights are Reserved

USENIX Conf. FAST, 2012, pp. 49-64.
[2] C. Liu, Y. Lu, C. Shi, G. Lu, D. Du, and D.-S. Wang, ‘‘ADMAD: Application-DrivenMetadata Aware De-Deduplication Archival Storage Systems,’’ in Proc. 5th IEEE Int’l Workshop SNAPI I/Os, 2008, pp. 29-35.
[3] 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.
[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 DeDuplication,’’ in Proc. 3rd USENIX Workshop HotStorage 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

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)