IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Intelligent Backup System

Shrinivas Dube[1], Rashmi Mandave[2], Vaibhav Chhajed[3], Pratik Bobde[4], Mrs. Shikha Pachouly[5]

Department of Computer Engineering, AISSMS COE, Pune, India[1, 2, 3, 4, 5]

Abstract: Nowadays, computing devices that rely on a cloud storage environment for data backup, an imminent challenge facing waste of upload bandwidth, storage space due to large amount of redundant data. In this paper, we present Intelligent Backup System, an intelligent data backup scheme that improves data deduplication efficiency by exploiting application awareness, and further combines client and cloud duplicate detection to strike a good balance between cloud storage capacity saving and deduplication time reduction.

Keywords: Data storage, cloud backup, application awareness, effective deduplication.

I. INTRODUCTION

The ever-growing volume and value of digital information have raised a critical and increasing requirement for data protection in the personal computing environment. 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.

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 (i.e., generally a cryptographic hash of the chunk data), replaces the duplicate chunks with their fingerprints after chunk fingerprint index lookup, and only transfers or stores the unique chunks for the purpose of communication or storage efficiency. 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. Unfortunately, such resources are limited in a typical personal computing device. Therefore, it is desirable to achieve a trade-off (i.e., deduplication efficiency) between deduplication effectiveness (i.e., duplicate elimination ratio) and system overhead for personal computing devices with limited system resources. The existing source deduplication strategies can be divided into two categories: local source deduplication 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, that performs duplicate check in backup datasets from all clients in the cloud side before data transfer over WAN. The former only eliminates intra-client redundancy with low duplicate elimination ratio by low-latency client-side duplicate data check, while the latter can suppress both intra-client and inter-client redundancy with high deduplication effectiveness by performing high-latency duplication detection on the cloud side. Local-global source deduplication scheme that eliminates intra-client redundancy at client before suppression inter-client redundancy in the cloud, can potentially improve deduplication efficiency in cloud backup services to save as much cloud storage space as the global method but at as low latency as the local mechanism.

In this paper, we propose IBS, a deduplication scheme that not only exploits application awareness, but also combines local and global duplication detection, to achieve high deduplication efficiency by reducing the deduplication latency to as low as the application-aware local deduplication while saving as much cloud storage cost as the application-aware global deduplication. Our application-aware deduplication design is motivated by the systematic deduplication analysis on personal storage. We observe that there is a significant difference among different types of applications in the personal computing environment in terms of data redundancy, sensitivity to different chunking methods, and independence in the deduplication process. Thus, the basic idea of IBS is to effectively exploit this application difference and awareness by treating different types of applications independently and adaptively during the local and global duplicate check processes to significantly improve the deduplication efficiency and reduce the system overhead.

We design a deduplication scheme that employs an intelligent data chunking method and an adaptive use of hash functions to minimize computational overhead and maximize deduplication effectiveness by exploiting application awareness. We combine local deduplication and global deduplication to balance the effectiveness and latency of deduplication. We also propose a data aggregation strategy at the client side to improve data transfer efficiency by grouping many small data packets into a single larger one for cloud storage[9].

II. DATA ANALYSIS

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

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Observation
A disproportionally large percentage of storage space is occupied by a very small number of large files with very low chunk-level redundancy after file-level deduplication.

Implication
File-level deduplication using weak hash functions for these large files is sufficient to avoid hash collisions for small datasets in the personal computing environment. To reveal the relationship between file count and storage capacity under various file size, we collect statistics on the distribution of file count and storage space occupied by files of different sizes and shows the results in Fig. 1. We observe that about 55 percent of all files are smaller than 10 KB, accounting for only 2.1 percent of the total storage capacity, and only 1.2 percent files are larger than 1MB but occupy 71 percent of the storage capacity. This suggests that tiny files can be ignored during the deduplication process as so to improve the deduplication efficiency, since it is the large files in the tiny minority that dominate in determining the deduplication efficiency. In Fig. 1. The left primary axis states Storage Capacity (GB) and secondary axis states File Count (Million).

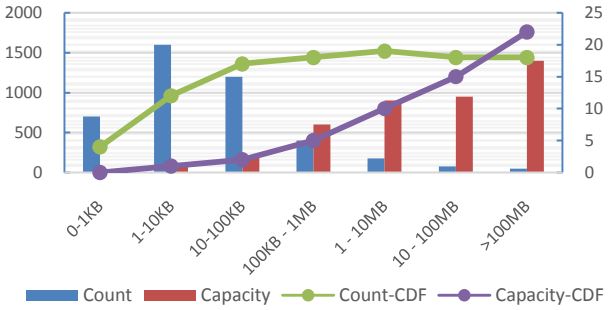

Fig. 1. File Size

III.DESIGN AND IMPLEMENTATION

IBS, motivated in part by our observations made in Section 2, is designed to meet the requirement of deduplication efficiency with high deduplication effectiveness and low system overhead. The main idea of IBS is exploiting both low-overhead local resources and high-overhead cloud resources to reduce the computational overhead by employing an intelligent data chunking scheme and an adaptive use of hash functions based on application awareness, and it combines local-global source deduplication with application awareness to improve deduplication effectiveness with low system overhead on the client side.

A. Architectural Overview
An architectural overview of IBS is illustrated in Fig. 2, where 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 application aware 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

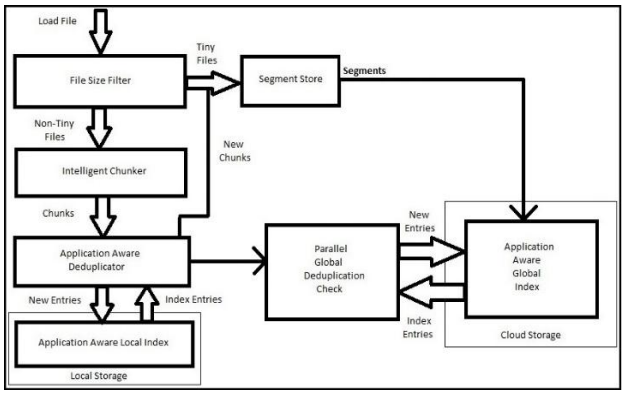

Fig. 2. Architectural overview of IBS design.

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. On the client side, 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 and improve network bandwidth efficiency over WAN. On the cloud datacentre side, segments and its corresponding chunk fingerprints are stored in cloud storage. We will now describe the deduplication process in more detail in the rest of this section.

B. File Size Filter
Most of the files in the PC dataset are tiny files that less than 10 KB in file size, accounting for a negligibly small percentage of the storage capacity. As shown in our statistical analysis in Section 2, about 55 percent of all files are tiny files, accounting for only 2.1 percent of the total storage capacity of the dataset. To reduce the metadata overhead, IBS filters out these tiny files in the file size filter before the deduplication process, and groups data from many tiny files together into larger units of about 1 MB each in the segment store to increase the data transfer efficiency over WAN.

C. Intelligent Data Chunking
The deduplication efficiency of data chunking scheme among different applications differs. Whether SC can outperform CDC in deduplication efficiency, we divide files into two main categories: static files and dynamic files. The dynamic files are always editable, while the static files are un-editable in common. To strike a better trade-off between duplicate elimination ratio and deduplication overhead, we deduplicate static files into fix-sized chunks by SC with ideal chunk size, and break dynamic files into variable-sized chunks with optimal

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

average chunk size using CDC based on the Rabin fingerprinting to identify chunk boundaries.

D. 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. IBS strikes a good balance between alleviating computation overhead on the client side and avoiding hash collision to keep data integrity. In both local and global detection scenarios, a SHA-1 value of chunk serves as chunk fingerprint of SC in static files and a MD5 value is used as chunk fingerprint of dynamic files since chunk length is another dimension for duplicate detection in CDC-based deduplication. To achieve high deduplication efficiency, the application aware deduplicator first detects duplicate data in the application-aware local index corresponding to the local dataset with 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. Only the unique data chunks after global duplicate detection are stored in the cloud storage.

E. Index Structure

IBS requires two application-aware chunk indices: a local index on the client side and a global index on the cloud side. Comparing with traditional deduplication mechanisms, IBS can achieve high deduplication throughput by looking up chunk fingerprints concurrently in small indices classified by applications rather than a single full, unclassified index for both local and global scenarios. Furthermore, a periodical data synchronization scheme is also proposed in IBS to back up the application-aware local index and file metadata in the cloud storage to protect the data integrity of the PC backup datasets.

F. Segment Management

Aggregation of data produces larger files for the cloud storage, which can be beneficial in avoiding high overhead of lower layer network protocols due to small transfer sizes, and in reducing the cost of the cloud storage. Amazon S3, for example, has both a per-request and a per-byte cost when storing a file, which encourages the use of files greater than 100 KB. IBS will often group deduplicated data from many smaller files and chunks into larger units called segments before these data are transferred over WAN.

IV. EVALUATIONS

We have built a prototype of IBS in approximately 5000 lines of C++ code. We have evaluated the advantages of our design over the state-of-the-art source deduplication based cloud backup services in terms of deduplication effectiveness by feeding the real-world datasets in a personal computing device. The following evaluation subsections will show the results, beginning with a description of the experiment platform with PC backup datasets we use as inputs.

A. Experimental Platforms

Our experiments were performed on a HP Pavilion Notebook client with 2.40 GHz Intel Core i3 processor, 4 GB RAM, and one 500 GB SATA disk, and the Amazon Web Service for cloud storage. The Notebook is connected to the Internet by campus wired network connectivity with 2 Mbps ~ 5 Mbps data transfer speed. To support our application-aware global index structure, we create different domains for each file-type by horizontal partitioning of chunk fingerprints to improve overall throughput with parallel fingerprint lookup. First, all the datasets are deduplicated by our IBS scheme, but we only store the global chunk index without unique data chunk to save cloud storage cost and protect data privacy. Then, we use new backup datasets as workloads to continue our evaluations with a total of 300MB logical data consisting of about 950 files.

We compare IBS against a number of state-of- the-art schemes, including Jungle Disk , a file incremental cloud backup scheme[6], BackupPC, a local file-level source deduplication based cloud backup[4], Cumulus, a local chunk-level source deduplication method[5].

B. Effectiveness

Our experimental results in Fig. 3 present the cumulative cloud storage capacity (MBs) required of the providers at each backup session for individual user with the six cloud backup schemes. Different from source deduplication schemes, Jungle Disk fails to achieve high cloud storage saving due to the fact that its incremental backup scheme cannot eliminate file copies written in different places.

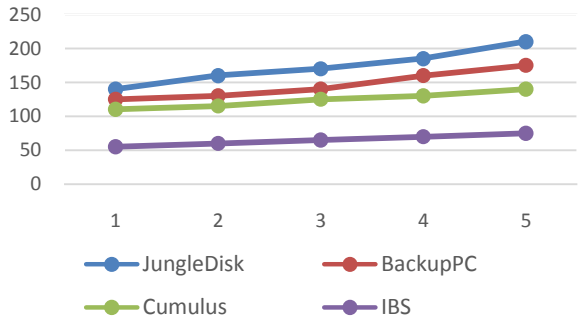

Fig. 3. Backup Sessions

In the source deduplication schemes, the coarse-grained method BackupPC cannot find more redundancy than other fine-grained mechanisms. The fine-grained Cumulus only performs local duplicate check, and limits the search for unmodified data to the chunks in the previous versions of the file, so it achieves lower space saving than the local deduplication. Due to the application-aware design and global duplicate detection, IBS can outperform Jungle Disk by 50 percent and save 35 percent space for Cumulus.

C. System Overhead

Considering the limited system resources in PC clients, we estimate the system overhead in terms of CPU processing speed and RAM usage for source deduplication based cloud backup services in personal devices. In IBS design,

we adaptively select chunking method and hash function for different application data to achieve high deduplication efficiency with low system overhead. Since the computational overhead of deduplication is dominated by CDC based chunking and hash fingerprinting, we test the average throughput of performing both chunking and fingerprinting in client for the five source deduplication based cloud backup services in backup sessions.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose IBS, a deduplication scheme for cloud backup in the personal computing environment to improve deduplication efficiency. An intelligent deduplication strategy in IBS is designed to exploit file semantics to minimize computational overhead and maximize deduplication effectiveness using application awareness. It combines local deduplication and global deduplication to balance the effectiveness and latency of deduplication.

As a direction of future work, we plan to further optimize our scheme for other resource-constrained mobile devices like smartphone or tablet.

ACKNOWLEDGMENT

Apart from our own, the success of this report depends largely on the encouragement and guidelines of many others. We are especially grateful to our guide **Prof S.J. Pachouly and Prof D. P. Gaikwad,** Head of Computer Engineering Department, AISSMSCOE who has provided guidance, expertise and encouragement. We are thankful to the staff of Computer Engineering Department for their cooperation and support .We would like to put forward our heartfelt acknowledgement to all our classmates, friends and all those who have directly or indirectly provided their over whelming support during this project work and the development of this report.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, „„A View of Cloud Computing,"" Commun. ACM, vol. 53, no. 4, pp. 49-58.

[2] H. Biggar, „„Experiencing Data De-Duplication: Improving Efficiency and Reducing Capacity Requirements,"" Enterprise Strategy Grp., Milford, MA, USA, White Paper.

[3] 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, pp. 29-35. .

[4] BackupPC, [Online]. Available: http://backuppc.sourceforge.net/

[5] A. Muthitacharoen, B. Chen, and D. Mazie `res, „„A LowBandwidth Network File System,"" in Proc. 18th ACM SOSP, pp. 174-187.

[6] Jungle Disk. [Online]. Available: http://www.jungledisk.com/

[7] S. Kannan, A. Gavrilovska, and K. Schwan, „„Cloud4HomeV Enhancing Data Services with Home Clouds,"" in Proc. 31st ICDCS, pp. 539-548.

[8] D. Meister and A. Brinkmann, ‘‘Multi-Level Comparison of Data Deduplication in a Backup Scenario,’’ in Proc. 2nd Annu. Int’l SYSTOR, pp. 1-8.

[9] Maximizing Data Efficiency: Benefits of Global DeduplicationNEC, Irving, TX, USA, NEC White Paper.