ISSN:0975-9646

An Application Awareness Local Source and Global Source De-Duplication with Security in resource constraint based Cloud backup services

S.Meghana
Assistant Professor, Dept. of IT, Vignana Bharathi Institute Of Technology,Hyderabad.

Abstract: Cloud computing technology shares the resources through the internet. Processors, computing architecture and storage pools are shared as software and infrastructure service. Remote data centers provide data and software's using the network bandwidth. Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are the popular cloud service providers. Cloud backup service provides offsite storage for the users with disaster recovery support. Deduplication methods are used to control high data redundancy in backup dataset .Data deduplication is a data compression approach applied in communication or storage environment. Limited resource level and I/O overhead are considered in the deduplication process. Source deduplication strategies can be divided into two categories. They are local source deduplication (LSD) and global source deduplication (GSD). LSD 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. GSD performs duplicate check in backup datasets from all clients in the cloud side before data transfer over WAN. Chunking and finger printing operations are carried out to identify and eliminate duplicate data. Chunking task partitions large data objects into smaller parts. Finger print is a cryptographic hash of the chunk data. Chunk fingerprint index lookup is used to replace the duplicate chunks. Application aware Local-Global source Deduplication (ALG-Dedupe) scheme is used to control redundancy in cloud backups. File size filter is used to separate the small size files. Application aware chunking strategy is used in Intelligent Chunker to break the backup data streams. Application aware deduplicator deduplicates the data chunks from the same type of files. Hash engine is used to generate chunk finger prints. Data redundancy check is carried out in application-aware indices in both local client and remote cloud. File metadata is updated with redundant chunk location details. Segments and corresponding finger prints are stored in the cloud data center using self describing data structure. Security ensured Application aware Local-Global source Deduplication (SALG-Dedupe) scheme is proposed to perform deduplication with security. Encrypted cloud storage model is used to secure personal data values. Deduplication scheme is adapted to control data redundancy under Smart Phone environment. File level deduplication scheme is designed for global level deduplication process.

I. INTRODUCTION

A powerful underlying and enabling concept is computing through service-oriented architectures (SOA) – delivery of an integrated and orchestrated suite of functions to an end-user through composition of both loosely and tightly coupled functions, or services – often network based. Related concepts are component-based system engineering, orchestration of different services through workflows and virtualization. The key to a SOA framework that supports workflows is componentization of its services, an ability to support a range of couplings among workflow building blocks, fault-tolerance in its data- and process-aware service-based delivery and an ability to audit processes, data and results, i.e., collect and use provenance information.

Virtualization is another very useful concept. It allows abstraction and isolation of lower level functionalities and underlying hardware. This enables portability of higher level functions and sharing and/or aggregation of the physical resources. The virtualization concept has been around in some form since 1960s. Since then, the concept has matured considerably and it has been applied to all aspects of computing – memory, storage, processors, software, networks, as well as services that IT offers. It is the combination of the growing needs and the recent advances in the IT architectures and solutions that is now bringing the virtualization to the true commodity level. Virtualization, through its economy of scale and its ability to offer very advanced and complex IT services at a reasonable cost, is poised to become, along with wireless and highly distributed and pervasive computing devices, such as sensors and personal cell-based access devices, the driving technology behind the next wave in IT growth.

Not surprisingly, there are dozens of virtualization products and a number of small and large companies that make them. Some examples in the operating systems and software applications space are VMware1, Xen – an open source Linux-based product developed by XenSource and Microsoft virtualization products, to mention a few. Major IT players have also shown are newed interest in the technology. Classical storage players such as EMC, NetApp, IBM and Hitachi have not been standing still either. In addition, the network virtualization market is teeming with activity.

II. RELATED WORK

Recently, data de-duplication has been emerged as an alternative lossless data compression scheme that has been employed in various backup and archival systems for storage optimization [7]. In cloud backup environment, it is desirable to have the redundant data removed at the source site before reaching the backup destination to significantly reduce network bandwidth consumption, which is different from the target de-duplication schemes widely employed in such systems as DDFS [7], Falconstor [2], Exgrid [3], Sepaton DeltaStor [5], etc. compares the source de-duplication methods used in five well-known developed backup systems, along with our proposed SAM scheme.

There are two main types of source de-duplication schemes, the source global chunk-level deduplication which removes all the redundant data among different clients globally and thus incurring heavy system overhead and the source local chunk-level de-duplication which incurs very little system overhead by only removing redundant data locally within the individual client. SAM, distinct from all of them, combines the source global file level de-duplication scheme and the source local chunk level de-duplication scheme to effectively tradeoff between the de-duplication efficiency and overhead. Although this hybrid method of combining file-level and chunk-level deduplication schemes has been used by other vendors the context of file system design and replica synchronization design, SAM, to the best of our knowledge, is the first of such hybrid methods designed for cloud backup environment.

In addition to SAM's source hybrid method, other target de-duplication methods also attempt to achieve an optimal tradeoff between the de-duplication efficiency and overhead at the chunk level [9], [8], [12]. DDFS is the first system to exploit the data streams' locality to reduce the de-duplication overhead for nightly backups in datacenters. Considering low-locality backup workloads, Extreme Binning [10] further exploits file similarity to achieve a deduplication throughput of one disk access per file, while ignoring some redundant data with a certain probability. In SAM, while the implementations of GFD and LCD are learned from the approaches used in DDFS and Extreme Binning respectively, they are noticeably different from the latter in the following aspects. GFD is implemented at the file level that takes advantage of file locality, as opposed to the chunk-level implementation of DDFS. Likewise, LCD is implemented in the client site locally, instead of Extreme Binning's implementation at the remote backup destination, to reduce the overhead of data transmission. More importantly, SAM relies significantly on the exploitation of file semantics, such as file locality, file timestamps, file size and file type, to effectively reduce the de-duplication overhead in each of its three stages. The file semantics have been widely used in the design and optimization of file systems, such as perfecting and caching [13]. Extracting file semantics, motivated by our experimental observations, is proven very useful to improve SAM's de-duplication performance.

III. DEDUPLICATION FOR CLOUD BACKUP SERVICES

Nowadays, 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 [1], 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 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 [11]. For dataset with logical size L and physical size P, source deduplication can reduce the data transfer time to P/L that of traditional cloud backup. 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. Unfortunately, such resources are limited in a typical personal computing device. Therefore, it is desirable to achieve a tradeoff between deduplication effectiveness and system overhead for personal computing devices with limited system resources. Code sign for storage and application is possible to optimize deduplication based storage system when the lower-level storage layer has extensive knowledge about the data structures and their access characteristics in the higher-level application layer. ADMAD improves redundancy detection by application-specific chunking methods that exploit the knowledge about concrete file formats. ViDeDup [4] 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. 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 existing source deduplication strategies can be divided into two categories: local source deduplication [6] 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. Inspired by Cloud4Home that enhances data services by combining limited local resources with low latency and powerful Internet resources with high latency, 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 ALG-Dedupe, an Application aware Local-Global source 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 ALG-Dedupe 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 have made several contributions in the paper. We propose a new metric, ‘‘bytes saved per second,’’ to measure the efficiency of different deduplication schemes on the same platform. We design an application-aware 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. To relieve the disk index lookup bottleneck, we provide application-aware index structure to suppress redundancy independently and in parallel by dividing a central index into many independent small indices to optimize lookup performance. 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. Our prototype implementation and real dataset driven evaluations show that our ALG-Dedupe outperforms the existing state-of-the-art source deduplication schemes in terms of backup window, energy efficiency, and cost saving for its high deduplication efficiency and low system overhead.

IV. PROBLEM STATEMENT

Application aware Local-Global source Deduplication (ALG-Dedupe) scheme is used to control redundancy in cloud backups. File size filter is used to separate the small size files. Application aware chunking strategy is used in Intelligent Chunker to break the backup data streams. Application aware deduplicator deduplicates the data chunks from the same type of files. Hash engine is used to generate chunk finger prints. Data redundancy check is carried out in application-aware indices in both local client and remote cloud. File metadata is updated with redundant chunk location details. Segments and corresponding finger prints are stored in the cloud data center using self-describing data structure (container). The following problems are identified from the existing system.

- Resource constrained mobile devices are not supported
- Data security is not considered
- Deduplication is not applied for small size files
- Backup window size selection is not optimized

V. APPLICATION-AWARE LOCAL-GLOBAL DEDUPLICATION (ALG-DEDUPE) SCHEME

An architectural overview of ALG-Dedupe is illustrated in Fig. 5.1, 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 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 omputing latency and improve network bandwidth efficiency over WAN. On the cloud datacenter side, segments and its corresponding chunk fingerprints are stored in a self describing data structure Vcontainer Vin cloud storage, supported by the parallel container store.

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. ALG-Dedupe will often group deduplicated data from many smaller files and chunks into larger units called segments before these data are transferred over WAN.

After a segment is sent to the cloud, it will be routed to a storage node in the cloud with its corresponding fingerprints, and be packed into container, a data stream based structure, to keep spatial locality for deduplicated data [14]. A container includes a large number of chunks and their metadata, and it has a size of several MB. An open chunk container is maintained for each incoming backup data stream in storage nodes, appending each new chunk or tiny file to the open container corresponding to the stream it is part of. When a container fills up with a predefined fixed size, a new one is opened up. If a container is not full but needs to be written to disk, it is padded out to its full size. This process uses chunk locality to group chunks likely to be retrieved together so that the data restoration performance will be reasonably good. Supporting deletion of files requires an additional process in the background. The similar scheme is also adopted in the state-of-the-art schemes such as DDFS and Sparse Indexing to improve manageability and performance.

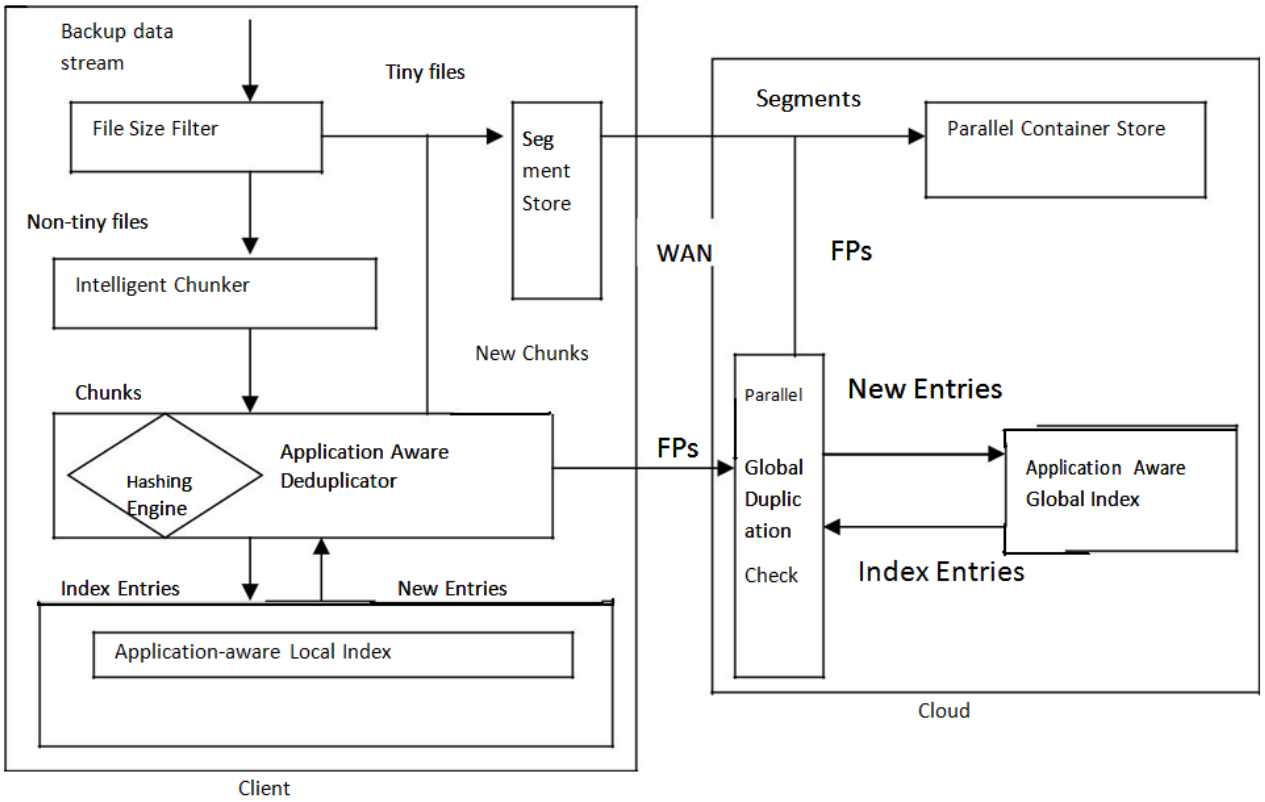

Figure 1: Architectural of the ALG-Dedupe Scheme

VI. SECURED DATA REDUNDANCY CONTROL SCHEME

The deduplication system is adapted for the Computer and Smart phone clients. The system provides security for the backup data values. Small size files are also included in the deduplication process. The system is divided into six major modules. They are Cloud Backup Server, Chunking Process, and Block level Deduplication, File level Deduplication, Security Process and Deduplication in Smart Phones. The cloud backup server module is designed to maintain the backup data for the clients. Chunking process module is designed to split the file into blocks. Block signature generation and deduplication operations are carried out in block level deduplication module. File level deduplication module is designed to perform deduplication in file level. Data security module is designed to protect the backup data values. Deduplication process is performed in the mobile phones in Deduplication under Smart phones module.

A. Cloud Backup Server

The cloud backup server module is designed to maintain the backup data for the clients. Chunking process module is designed to split the file into blocks. Block signature generation and deduplication operations are carried out in block level deduplication module. File level deduplication module is designed to perform deduplication in file level. Data security module is designed to protect the backup data values. Deduplication process is performed in the mobile phones in Deduplication under Smart phones module.

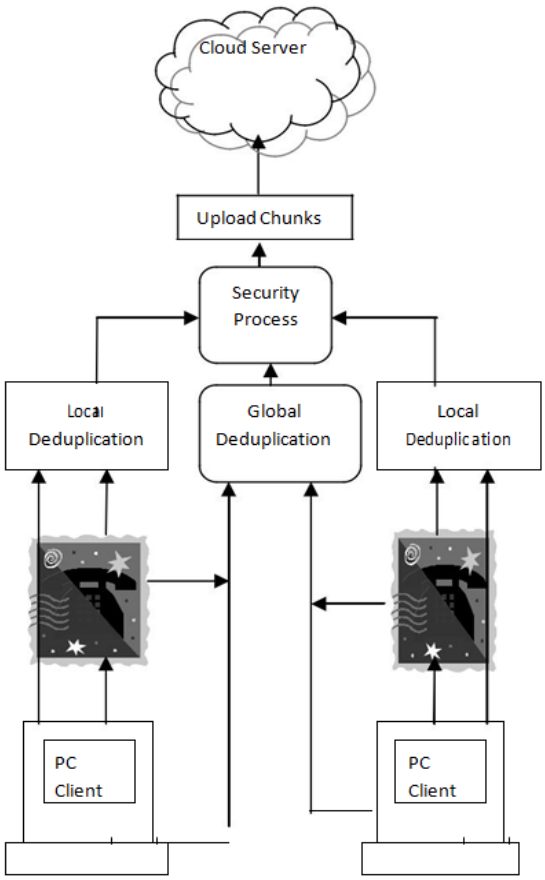

Figure 2: Secured Data Redundancy Control Scheme

B. Chunking Process

File size filter is used to separate the tiny files. Intelligent chunker is used to break up the large size files into chunks. Backup files are divided into three categories. They are compressed files, static uncompressed files and dynamic uncompressed files. Static files are uneditable and dynamic files are editable. Compressed files are chunked with Whole File Chunking (WFC) mechanism. Static uncompressed files are partitioned into fix-sized chunks by Static Chunking (SC). Dynamic uncompressed files are braked into variable-sized chunks by Content Defined Chunking (CDC).

C. Block Level Deduplication

Chunks finger prints are generated in the hash engine. Rabin hash functions with 12 bytes are used as chunk fingerprint for local duplicate data detection for compressed files. Message Digest MD5 algorithm is used for global deduplication process in compressed files. Secure Hash Algorithm (SHA1) is used for deduplication in uncompressed static files. Chunks finger prints are generated in the hash engine. Rabin hash functions with 12 bytes are used as chunk fingerprint for local duplicate data detection for compressed files. Message Digest MD5 algorithm is used for global deduplication process in compressed files. Secure Hash Algorithm (SHA1) is used for deduplication in uncompressed static files. Dynamic uncompressed files are hashed using Message Digest (MD5) algorithm. Deduplicate detection is carried out in the local client and remote cloud. Fingerprints are indexed in local and global level. Deduplication is performed by verifying the finger print index values.

D. File Level Deduplication

Tiny files are maintained under segment store environment. File level deduplication is performed on files with the size less than 10 KB. File level fingerprints are generated using Rabin hash Function. Deduplication is performed with file level fingerprint index verification mechanism.

E. Security Process

The backup data values are maintained in encrypted form. Modified Advanced Encryption Standard (MAES) algorithm is used in the encryption/decryption process. Encryption process is performed after the deduplication process. Local and global keys are used for the data security process. Deduplication in Smart Phones.

F. Deduplication in Smart Phones

The deduplication process is tuned for smart phone environment. Smart phones are used as client for cloud backup services. File level and block level deduplication tasks are supported by the system. Data security is also provided for the smart phone environment.

VII. CONCLUSION

Cloud backup services re used to maintain the personal data in cloud data centers. Source deduplication methods are applied to limit the storage and communication requirements. Application aware Local-Global source Deduplication (ALG-Dedupe) mechanism performs redundancy filtering in same client and all client environments. The Security ensured Application aware Local-Global source Deduplication (SALG-Dedupe) scheme is designed with security and mobile device support features. Deduplication and power efficiency is improved in the computer and smart devices environment. The system reduces the cost for cloud backup services. Data access rate is increased by the system. The system achieves intra-client and inter-client redundancy with high deduplication effectiveness.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, 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, Apr. 2010.

[2] Falconstor, http://www.falconstor.com.

[3] Exgrid, http://www.exagrid.com.

[4] 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.

[5] Sepaton DeltaStor, http://www.sepaton.com.

[6] BackupPC, 2011. [Online]. Available: http://backuppc.sourceforge. net/

[7] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the Data Domain deduplication file system," in *FAST'08*, Feb. 2008.

[8] S. Rhea, R. Cox, and A. Pesterev, "Fast, inexpensive content addressed storage in Foundation," in *USENIX'08*, Jun. 2008.

[9] M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar and P. Campbell, "Sparse Indexing: Large scale, inline deduplication using sampling and locality," in *FAST'09*, Feb. 2009.

[10] D. Bhagwat, K. Eshghi, D. D. Long, and M. Lillibridge, "Extreme Binning: Scalable, Parallel Deduplication for Chunk based File Backup," HP Laboratories, Tech. Rep. HPL-2009-10R2, Sep. 2009.

[11] P. Shilane, M. Huang, G. Wallace and W. Hsu, ''WAN Optimized Replication of Backup Datasets Using Stream- Informed Delta Compression,'' in Proc. 10th USENIX Conf. FAST, 2012, pp. 49-64.

[12] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li, "Decentralized deduplication in SAN cluster file systems," in *USENIX'09*, Jan. 2009.

[13] P. Xia, D. Feng, H. Jiang, L. Tian, and F. Wang, "FARMER: A Novel Approach to File Access Correlation Mining and Evaluation Reference Model for Optimizing Peta-Scale File System Performance," in *HPDC08*, Jun. 2008.