www.ijiarec.com

ISSN: 2348-2079

MAY-2015

International Journal of Intellectual Advancements and Research in Engineering Computations

SECURED DEDUPLICATION MECHANISM UNDER CLOUD BACKUP SERVICES

[1]V. Gandhimathi, [2]M. Shanthamani.

ABSTRACT

Cloud backup service provides offsite storage for the users with disaster recovery support. De duplication methods are used to control high data redundancy in backup dataset .Data de duplication is a data compression approach applied in communication or storage environment. Limited resource level and I/O overhead are considered in the de duplication process. Data redundancy is controlled using Application aware Local-Global source Deduplication (ALG-Dedupe) scheme. 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. The deduplication scheme is enhanced with security features as Security ensured Application aware Local-Global source Deduplication (SALG-Dedupe) scheme. 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.

INTRODUCTION

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25 GB. Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. The amount of digital information created in 2007 was 281 exabytes; by 2011, it is expected to be 10 times larger [10]. 35% of this data originates in enterprises and consists of unstructured content, such as office documents, web pages, digital images, audio and video files and electronic mail. Enterprises retain such data for corporate governance, regulatory compliance, litigation support and data management. To mitigate storage costs associated with backing up such huge volumes of data, data deduplication is used. Data deduplication identifies and eliminates duplicate data. Storage space requirements can be reduced by a factor of 10 to 20 or more when backup data is de duplicated.

Chunk-based deduplication, a popular deduplication technique, first divides input data streams into fixed or variable-length chunks. Typical chunk sizes are 4 to 8 kB. A cryptographic hash or chunk ID of each chunk is used to determine if that chunk has been backed up before. Chunks with the same chunk ID are assumed identical. New chunks are stored and references are updated for duplicate chunks. Chunk based deduplication is very effective for backup workloads, which tend to be files that evolve slowly, mainly through small changes, additions and deletions.

Inline deduplication is deduplication where the data is deduplicated before it is written to disk as opposed to post process deduplication where backup data is first written to a temporary staging area and then deduplicated offline. One advantage of inline deduplication is that extra disk space is not required to hold and protect data yet to be backed up. Data Domain, Hewlett Packard and Diligent Technologies

Author for Correspondence:

[1]Final year ME CSE, Velalar College Of Engineering And Technology, Thindal, Erode, Tamilnadu, India.

[2]Asst.Prof-Sr.Gr, Velalar College Of Engineering And Technology, Thindal, Erode, Tamilnadu, India.

are a few of the companies offering inline, chunk-based deduplication products.

RELATED WORK

There largely exist three approaches for reducing the size of information: delta encoding, duplication elimination and compression. Each of these techniques is used independently or in a combined manner to improve the space efficiency and network bandwidth utilization. Delta encoding stores only the differences between sequential data. It is a common and efficient method to reduce data redundancy when changes are small. It is used in many applications including source control and backup. Kalkarni et al. proposed redundancy elimination at block level (REBL), which is a combination of block suppression, delta encoding and compression.

Backup applications also exploit information redundancy to reduce the amount of information to be backed up [6], [7]. In a distributed file system, the deduplication technique has been used to reduce the network traffic involved in synchronizing file system contents between a client and a server. In a SAN file system, when different files share an identical piece of information, each file harbors the pointer to the shared data chunk instead of maintaining redundant information. Detecting information redundancy is widely used when locating the document source for a multisource download application. The web environment is an important area for duplication detection and elimination. These applications compute the fingerprints of the contents in the proxy server and eliminate the retrieval of the same data.

Won et al. found that chunking is one of the major overheads for the deduplication process [13], [6]. There are a two basic approaches in partitioning a file: variable-size chunking and fixed-size chunking. A number of preceding works have adopted fixed-size chunking for backup applications and large-scale file systems. The variable-size chunking algorithm is widely used in various application domains of duplication elimination such as backups, file systems and data transfers [7]. Policroniades et al. examined the effectiveness of variable-size chunking and fixed-size chunking using website data, different data profiles in academic data, source codes, compressed data and packed files. A few works proposed to apply variable size chunking and fixed size chunking based upon the characteristics of the file. Liu et al. proposed ADMAD scheme [8] which applies different file chunking methods based upon the metadata of individual files. Context-Aware chunking proposed in our work shares the basic idea with Liu's work. We only use file extension rather than all file metadata to reduce the overhead of accessing file metadata.

Meister et al. exploited the file characteristics, e.g., compressed file, email archive, etc., in chunking a file [9] and analyzed the deduplication efficiency under various chunking schemes. While they show that delta encoding yield the best deduplication ratio in desktop applications, e.g., MS Office, Zip, compared to variable size and fixed size chunking, they did not address the issue of excessive fingerprint generation in delta encoding. Mandagere et al. examine the effect of different chunking methods and different chunk size settings over deduplication performance metrics: fold factor, reconstruction overhead and CPU utilization [2]. They did not examine effect of chunk size over entire backup speed, on which we perform comprehensive study. Similar with Meister's work [9], the reduction on redundancy detection rate is marginal, from 34.9 to 33.2 percent when chunk size increase from 8 to 16 KB. Instead of deduplicating files at chunk granularity, Bolosky et al. proposed a method to detect duplicate data using file granularity. Deep Store is designed to adaptively accommodate different types of chunking mechanisms.

There are a number of aspects to expedite the fingerprint Looku. The first issue is to introduce main memory filter. To avoid disk I/O when looking up the index, a main memory filter called the Bloom filter has been introduced in various applications including backups [7], distributed file systems and web proxies. The important question yet to be answered is the relationship between the false positive rate and the overall lookup performance. Mitzenmacher made the interesting observation that minimizing the false positive rate of the Bloom filter did not necessarily yield the optimal performance of the web proxy lookup. Rather, sacrificing the false positive rate and making the bit vector of the Bloom filter more compressive actually improves the

fingerprint lookup overhead. The second aspect is to reduce the number of fingerprints used in comparison. Lillibriged et al. proposed to use sampling to reduce the number of fingerprints. It can reduce the memory requirement and the degradation in Deduplication ratio is reasonable [5]. This approach does not work when there is a significant change in chunk size, i.e., removal or insertion of data. To reduce the number of comparison, Aronovich et al. proposed to maintain summary information in larger unit, e.g., 10 MB and de duplicate the data based upon its similarity with the existing data [11].

They maintain tree of fingerprints where a parent node's fingerprint is the hash value of the fingerprints of the child nodes. Bhagawat et al. exploit the file similarity instead of locality in deduplication [12]. Their work manifests itself when backing up a set of small files arriving from different hosts. Third approach is to reduce the disk overhead in fingerprint lookup. The key ingredient is to enforce access locality in storing fingerprints at the storage. Zhu et al. [7] proposed a technique called SISL, where they simply append the incoming fingerprints at the end of existing table. Spyglass, a file metadata search system, proposed hierarchical partitioning of name space organization for performance and scalability [3]. Spyglass exploits namespace locality to improve performance since the files that satisfy a query are often clustered in only a portion of the namespace.

Since a number of files share same piece of information, loss of a chunk may result in loss of multiple files. A number of works addressed the reliability issue in deduplication. To enhance reliability of deduplicated data, Liu et al. proposed to form a set of variable size chunks into fixed size objects and to append ECC [4]. Bhagawat et al. proposed to apply different levels of replication for each chunk. They proposed to determine replication level based upon the amount of information loss when the respective chunk becomes unavailable. Recently, Efstathopoulos et al. dealt with the issue of Chunk garbage collection. They proposed a mechanism called grouped mark-and Sweep [14].

SOURCE DEDUPLICATION SCHEMES AND ITS ISSUES

The 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. 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 [15]. 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.

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 de duplicator de duplicate 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. The following issues are identified from the current source deduplication methods.

- 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.

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

An architectural overview of ALG-Dedupe is illustrated in Fig. 4.1, 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 de duplicator 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 computing latency and improve network bandwidth efficiency over WAN.

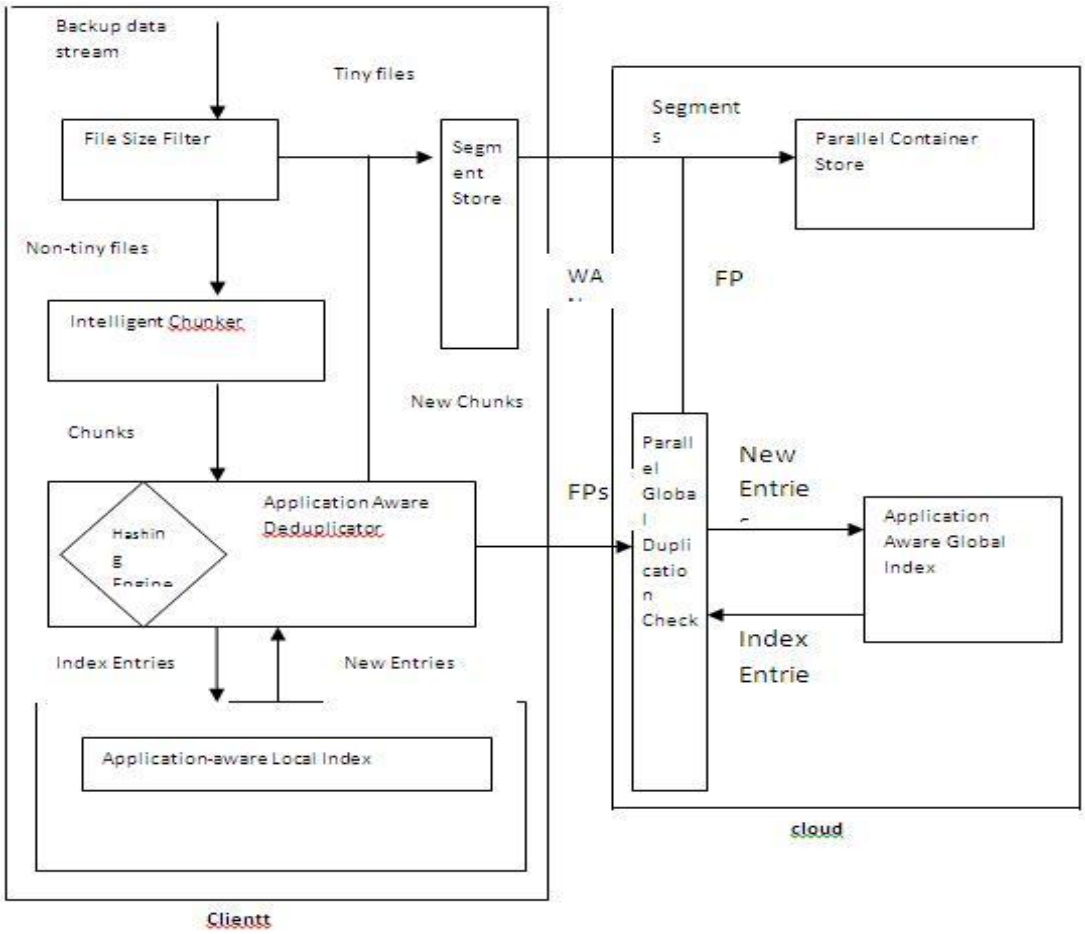

Fig. 4.1. Architectural of the ALG-Dedupe Scheme

SMARTPHONES

A smartphone is a mobile phone with an operating system. Smartphones typically include the features of a phone with those of another popular consumer device, such as a personal digital assistant, a media player, a digital camera and/or a GPS navigation unit. Later smartphones include all of those plus the features of a touchscreen computer, including web browsing, Wi-Fi, 3rd-party apps, motion sensor, mobile payment and 3G.Devices that combined telephony and computing were first conceptualized by Theodore G. Paraskevakos in 1971 and patented in 1973 and were offered for sale beginning in 1993. He was the first to introduce the concepts of intelligence, data processing and visual display screens into telephones which gave rise to the "Smartphone." In 1971, Paraskevakos, working with Boeing in Huntsville, Alabama, demonstrated a transmitter and receiver that provided additional ways to communicate with remote equipment, however it did not yet have general purposePDA applications in a wireless device typical of smartphones. They were installed at Peoples' Telephone Company in Leesburg, Alabama and were demonstrated to several telephone companies. The original and historic working models are still in the possession of Paraskevakos.

The first mobile phone to incorporate PDA features was an IBM prototype developed in 1992 and demonstrated that year at the COMDEX computer industry trade show. A refined version of the product was marketed to consumers in 1994 by BellSouth under the name Simon Personal Communicator. The Simon was the first cellular device that can be properly referred to as a "smartphone", although it wasn't called a smartphone in 1994. In addition to its ability to make and receive cellular phone calls, Simon was also able to send and receive faxes and e-mails and

included several other apps like address book, calendar, appointment scheduler, calculator, world time clock and note pad through its touch screen display. Simon is the first smartphone to be incorporated with the features of a PDA.

CLOUD BACKUP SERVICES FOR SMART DEVICES

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, 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.

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.

CHUNKING PROCESS

File size filter is used to separate the tiny files. Intelligent chunker is used to breakup 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 un editable 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 broken into variable-sized chunks by Content Defined Chunking (CDC).

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. De duplicate 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.

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.

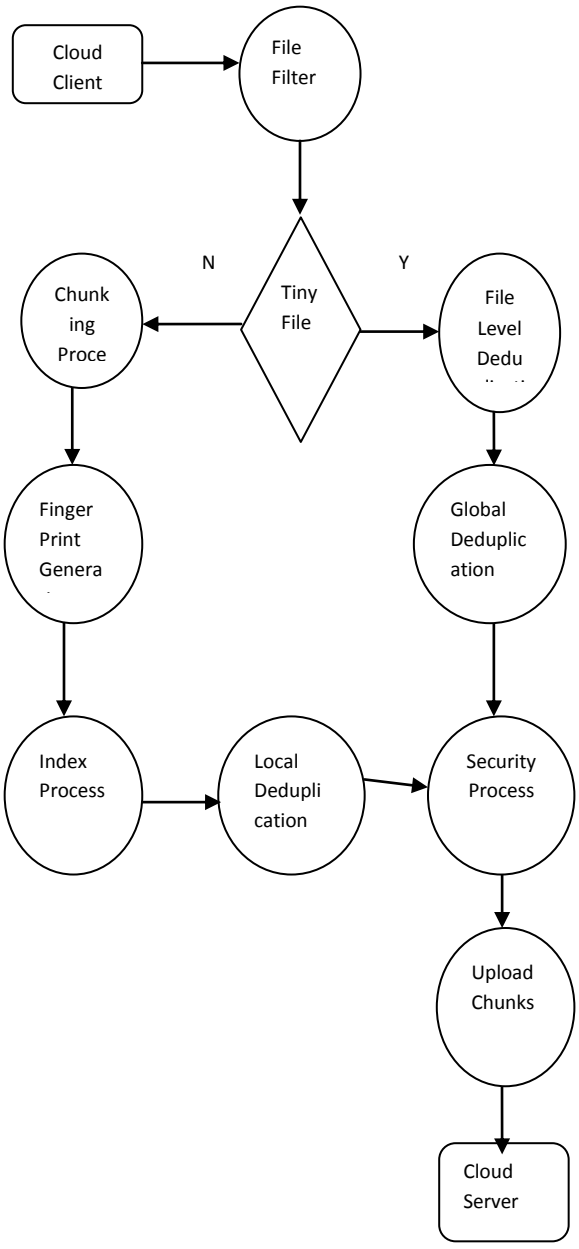

Fig. No: 6.1. Cloud Backup Services For Smart Devices

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.

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.

CONCLUSION

Cloud data centers are used to backup the personal data values. 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]. J. Gantz, C. Chute and A. Toncheva, The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth through 2011.

[2]. N. Mandagere and S. Uttamchandani, "Demystifying Data Deduplication," Proc. ACM/IFIP/USENIX Middleware '08 Conf. Companion, Dec. 2008.

[3]. A. Leung, M. Shao and E. Miller, "Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems," Proc. Six USENIX Conf. File and Storage Technologies, 2009.

[4]. C. Liu, Y. Gu, L. Sun, B. Yan and D. Wang, "R-ADMAD: High Reliability Provision for Large-Scale De-Duplication Archival Storage Systems," Proc. 23rdInt'l Conf., 2009.

[5]. Lillibridge and P. Camble, "Sparse Indexing: Large Scale, Inline Deduplication Using Sampling and Locality," Proc.

Seventh USENIX Conf. File and Storage Technologies, 2009.

[6]. Y. Won, J. Ban and J. Lee, "Efficient Index Lookup for De-Duplication Backup System," Proc. IEEE Int'l Symp. Modeling, Analysis and Simulation of Computers and Telecomm. Systems, Sept. 2008.

[7]. B. Zhu and H. Patterson, "Avoiding the Disk Bottleneck in the Data Domain Deduplication File System," Sixth USENIX Conf. 2008.

[8]. Liu and D. Wang, "ADMAD: Application-Driven Metadata Aware De-Duplication Archival Storage System," Proc. Fifth IEEE Int'l Workshop Storage Network Architecture and Parallel I/Os, 2008.

[9]. D. Meister and A. Brinkmann, "Multi-Level Comparison of Data Deduplication in a Backup Scenario," Proc. SYSTOR '09: The Israeli Experimental Systems Conf., May 2009.

[10]. J. F. Gantz, C. Chute and A. Toncheva, "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," IDC, An IDC White Paper - sponsored by EMC, March 2008.

[11]. L. Aronovich, Hirsch and S. Klein, "The Design of a Similarity Based Deduplication System," Proc. SYSTOR '09: The Israeli Experimental Systems Conf., May 2009.