ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2016

Similarity and Locality Based Indexing Approach for High Performance Data Deduplication

Vishal Pachangane[1], Prof.Priti Gode[2]

M. E Student, Dept of Computer Engineering, Alard College of Engineering and Management, SavitribaiPhule Pune University, Pune, India [1]

Dept of Computer Engineering, Alard College of Engineering and Management, SavitribaiPhule Pune University, Pune, India [2]

ABSTRACT: Data deduplication is a method of reducing storage needs by eliminating redundant data. Only one unique instance of the data is actually retained on storage media, such as disk or tape. Redundant data is replaced with a pointer to the unique data copy. Data deduplication has gained increasing attention and popularity as a space-efficient approach in backup storage systems. Data deduplication not only reduces the storage space requirements by eliminating redundant data but also minimizes the network transmission of duplicate data in the network storage systems. It splits files into multiple chunks that are each uniquely identified by a hash signature (e.g., MD5, SHA-1, and SHA-256), also called a Fingerprint. It removes duplicate chunks by checking their fingerprints, which avoids byte-by-byte comparisons. One of the main challenges for centralized data deduplication is the scalability of fingerprint-index search. In our project, we present SiLo, a similarity-locality approach that exploits both similarity and locality in backup streams to achieve higher deduplication throughput, well balanced load, and near-complete duplicate elimination at an extremely lower RAM overhead than existing state-of the-art approaches. The main idea behind SiLo is to expose and exploit more similarity by grouping strongly correlated small files into a segment and segmenting large files, and to leverage the locality in the data stream by grouping contiguous segments into blocks to capture similar and duplicate data missed by the probabilistic similarity detection. SiLo also employs a locality based stateless routing algorithm to parallelize and distribute data blocks to multiple backup nodes.

I. INTRODUCTION

Cloud computing is an internet based computing which enables sharing of services. Cloud allows users to useapplications without installation any application and access their personal files and application at any computer with internet or intranet access.In computing, data deduplication is a data compression technique in which redundant or repeated copies of data are removed from a system. It is implemented in data backup and network data mechanisms and enables the storage of one unique instance of data within a database or information system (IS). Data deduplication is also known as intelligent compression, single instance storage and commonality factoring or data reduction. Data deduplication not only reduces the storage space requirements by eliminating redundant data but also minimizes the network transmission of duplicate data in the network storage systems. It splits files into multiple chunks that are each uniquely identified by a hash signature (e.g., MD5, SHA-1, and SHA-256), also called a fingerprint. It removes duplicate chunks by checking their fingerprints, which avoids byte-by-byte comparisons. Despite recent progress in data deduplication studies many challenges remain, particularly in the petabyte-scale deduplication based backup storage systems that are generally centralized. One of the main challenges for centralized data deduplication is thescalability of fingerprint-index search. The fingerprinting indexing has become the main performance bottleneck of large-scale data deduplication systems. In order to address this performance bottleneck, many approaches have been proposed. There are two primary approaches to scaling data deduplication: locality based acceleration of deduplication, and similarity based deduplication. In locality-based approach, chunk lookups are one by one but some backup streams have high locality i.e. between first, second and next backups have a very high probability that chunks are in the same

DOI: 10.15680/IJIRCCE.2016. 0410177

order. However this approach shows low speed on backup stream with weak locality. In similarity-based approach, instead of lookups per chunks or per local chunks (locality) the lookups are per files. Although is much faster than locality approach it can sacrifice the duplication accuracy. We propose SiLo, a similarity-locality approach that exploits both similarity and locality in backup streams to achieve higher deduplication throughput, well balanced load, and near-complete duplicate elimination at an extremely lower RAM overhead than existing state-of the-art approaches. The main aim ofSiLo is to expose and uses more similarity by grouping strongly correlated small files into a segmentand segmenting large files, and to leverage the locality in the data stream by grouping contiguous segments into blocks to capture similar and duplicate data missed by the probabilistic similarity detection. SiLo also uses a locality based stateless routing algorithm to parallelize and distribute data blocks to multiple backup nodes.

II. PROBLEM STATEMENT

One of the main issues is the scalability of fingerprint-index based search schemes. The fingerprinting indexing has become the main performance bottleneck of large-scale data deduplication systems. In locality-based approach, chunk lookups are one by one but some backup streams have high locality. However this approach shows low speed on backup stream with weak locality. In similarity-based approach, instead of lookups per chunks or per local chunks (locality) the lookups are per files. Although is much faster than locality approach it can sacrifice the duplication accuracy.

III. EXISTING SYSTEM

The existing system contains thefingerprint-index based search schemes .This fingerprinting indexing has become the main performance bottleneck of large-scale data deduplication systems. In locality-based approach, chunk lookups are one by one but some backup streams have high locality. However this approach shows low speed on backup stream with weak locality. In similarity-based approach, instead of lookups per chunks or per local chunks (locality) the lookups are per files. Although is much faster than locality approach it can sacrifice the duplication accuracy.

Disadvantage of Existing System:-

- Low data deduplication.
- Low speed on backup stream with weak locality.
- Sacrifice the duplication accuracy.

IV. PROPOSED SYSTEM

The proposed system, design SiLo scheme that contain both similarity and locality based indexing for high performance data deduplication. The main idea behind SiLo is to expose and exploit more similarity by grouping strongly correlated small files into a segment and segmenting large files, and to leverage the locality in the data stream by grouping contiguous segments into blocks to capture similar and duplicate data missed by the probabilistic similarity detection. SiLo also employs a locality based stateless routing algorithm to parallelize and distribute data blocks to multiple backup nodes.

Advantage of Proposed System:-

- High data deduplication.
- High speed on backup stream.
- Achieve duplication accuracy.
- Maintain load balance among backup nodes.

V. LITERATURE SURVEY

W. Xia, H. Jiang, D. Feng, L. Tian, M. Fu, and Z. Wang (2012) have proposed PDedupe: Exploiting Parallelism in Data Deduplication System [1]. The AuthorproposePDedupe, a fast and scalable deduplication system. The main aim of P-Dedupe is to fully compose pipelined and parallel computations of data deduplication by effectively exploiting the idle resources of modern computer systems with multi-core and many-core processor architectures.

P. Bhatotia, R. Rodrigues, and A. Verma, (2012) have represented Shredder: GPU Accelerated Incremental Storage and Computation [2]. Represent the design, implementation as well as evaluation of Shredder, a high performance content-based chunking framework for supporting incremental storage and computation systems. Shredder exploits the imperially parallel processing power of GPUs to overcome the CPU bottlenecks of content-based chunking in a cost-effective manner as well as unlike previous uses of GPUs, which have focused on applications where computation costs are dominant.

G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu (2012) has shown Characteristics of Backup Workloads in Production Systems(3). The author present a comprehensive characterization of backup workloads by analyzing statistics and content metadata collected from a large setof EMC Data Domain backup systems in production use.

K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti(2012) have developed iDedup: Latency-Aware, Inline Data Deduplication for Primary Storage[4The author proposed an inline deduplication solution, iDedup, for primary workloads, while minimizing extra IOs and seeks. The algorithm is based on two key insights from real world workloads: i) spatial locality exists in duplicated primary data - Using this, we selectively deduplicate only sequences of disk blocks which reduces fragmentation and amortizes the seeks caused by deduplication. ii) Temporal locality exists in the accesspatterns of duplicated data- It allows us to replace the expensive and on-disk, deduplication metadata with a smaller, in-memory cache. These two techniques enable us to tradeoff capacity savings for performance.

D. Meyer and W. Bolosky (2011) have represented A Study of Practical Deduplication [5]. The author analyzed the data to determine the relative efficacy of data deduplication, by considering whole-file versus block-level elimination of redundancy. Here found that whole-file deduplication achieves about three quarters of the space savings of the most aggressiveblock-level deduplication for storage of live file systems and 87% of the savings for backup images. He also studied file fragmentation finding that it is not prevalent and updated prior file system metadata studies and finding that the distribution of file sizes continues to skew toward very large unstructured files.

F. Guo and P. Efstathopoulos(2011) has performed Building a High-Performance Deduplication System[6].The author present high-performance deduplication

Prototype and designed from the ground up to optimize overall single-node performance, by making the best possible use of a node's resource as well as achieve three important goals: i) *scale* to large capacity ii) provide good deduplication efficiency iii) near-raw-disk throughput.

W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane(2011) have discovered Tradeoffs in Scalable Data Routing for Deduplication Clusters[7].The author represent a cluster-based deduplication system which can deduplicate with high throughput and support deduplication ratios comparable to that of a single system as well as maintain a low variation in the storage utilization of individual nodes.

J. Wei, H. Jiang, K. Zhou, and D. Feng (2010) have designed MAD2: A Scalable High- Throughput Exact Deduplication Approach for Network Backup Services [8]. The deduplication technology has been widely applied in disk-based secondary storage systems. There are two technical challenges i) duplicate-lookup disk bottleneck which determine if an incoming data object is a duplicate and the index can become too large for RAM to hold in its entirety. Ii)storage node island effect which eliminate duplicates among multiple servers.

B. Debnath, S. Sengupta, and J. Li(2010) has represented Chunkstash: Speeding Up Inline Storage Deduplication Using Flash Memory[9].The author present the method of identifying duplicate data by using disk-based indexes on chunk hashes which can create throughput bottlenecks due to disk I/Os involved in index lookups. To reduce the penalty of index lookup misses in RAM by orders of magnitude by serving such lookups from a flash-based index and increasing inline deduplication throughput. Flash memory is responsible to reduce the huge gap between RAM and hard disk in terms of both cost and access times and is a suitable choice for this application. The author designs a flash-assisted inlinededuplication system using ChunkStashwhere a chunk metadata store on flash.

Y. Tan, H. Jiang, D. Feng, L. Tian, Z. Yan, and G. Zhou(2010) have proposed SAM: A Semantic-Aware Multi-Tiered Source De-Duplication Framework for Cloud Backup[10]. The author proposed SAM, a Semantic-Aware Multitiered source de-duplication framework which first responsible to combines the global file-level de-duplication and local chunk-level deduplication and then uses file semantics in each stage in the framework, to obtain an optimal tradeoff between the deduplication efficiency and de-duplication overhead and finally achieve a shorter backup window than existing work.

VI. SYSTEM ARCHITECTURE

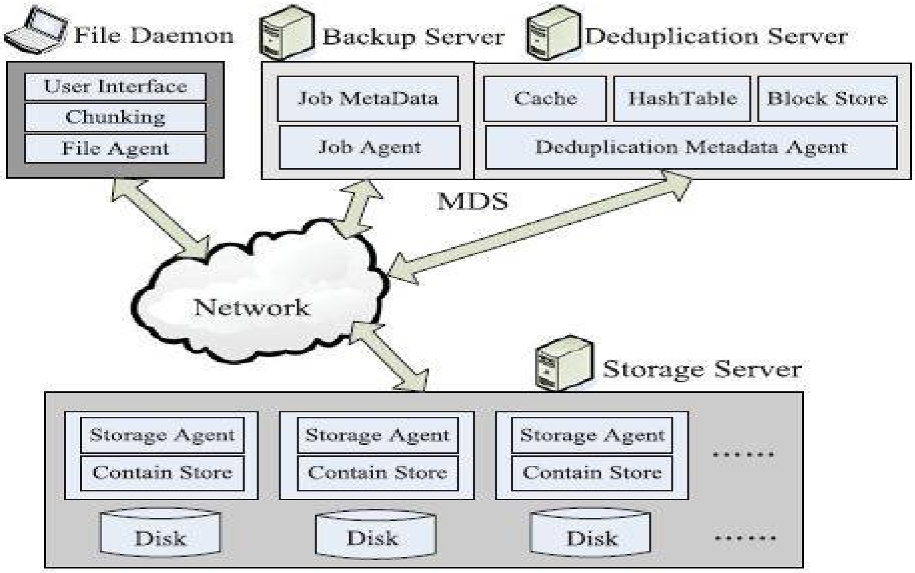

Fig.System Architecture

VII. CONCLUSION

To maintain the scalability of data deduplication and meet increasing size of data storage scale in mass storage system, we design SiLo architecture, a similarity-locality based deduplication system that exploits both similarity and locality in backup streams to achieve higher deduplication throughput, well balanced load, and near-complete duplicate elimination at an extremely lower RAM overhead than existing system. The proposed system shows that theSiLo similarity algorithm reduces the RAM usage, its locality algorithm helps eliminate most of the duplicate data that is missed by the probabilistic similarity detection, and its load distribution algorithm obtains a well-balanced load.

REFERENCES

[1] W. Xia, H. Jiang, D. Feng, L. Tian, M. Fu, and Z. Wang, “PDedupe: Exploiting Parallelism in Data Deduplication System,” Proc. IEEE Seventh Int’l Conf. Networking,Architecture and Storage (NAS), pp. 338-347, 2012.

[2] P. Bhatotia, R. Rodrigues, and A. Verma, “Shredder: GPUAccelerated Incremental Storage and Computation,” Proc. 10th USENIX Conf. File and Storage Technologies, 2012.

[3] G. Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, “Characteristics of Backup Workloads in Production Systems,” Proc. 10th USENIX Conf. File and Storage Technologies, 2012.

[4] K. Srinivasan, T. Bisson, G. Goodson, and K. Voruganti, “iDedup: Latency-Aware, Inline Data Deduplication for Primary Storage,” Proc. 10th USENIX Conf. File and Storage Technologies, 2012.

[5] D. Meyer and W. Bolosky, “A Study of Practical Deduplication,” Proc. Ninth USENIX Conf. File and Storage Technologies, 2011.

[6] F. Guo and P. Efstathopoulos, “Building a High-Performance Deduplication System,” Proc. 2011 Conf. USENIX Ann. Technical Conf., 2011.

[7] W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, “Tradeoffs in Scalable Data Routing for Deduplication Clusters,” Proc. Ninth USENIX Conf. File and Storage Technologies, 2011.

[8] J. Wei, H. Jiang, K. Zhou, and D. Feng, “MAD2: A Scalable High- Throughput Exact Deduplication Approach for Network Backup Services,” Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST), pp. 1-14, 2010.

[9] B. Debnath, S. Sengupta, and J. Li, “Chunkstash: Speeding Up Inline Storage Deduplication Using Flash Memory,” Proc. 2010 USENIX Conf. USENIX Ann. Technical Conf., 2010.

[10] 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,” Proc. IEEE 39th Int’l Conf. Parallel Processing, pp. 614-623, 2010.