IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 84-86
www.iosrjournals.org

Robust Inline Data Reduction Technique in Multi-tenant Storage

Mr. Ankur Sinha[a], Mr. Anand Khandare[b]
[a]*PG Student, Thakur College of Engineering and Technology, Mumbai-400 101, India*
[b]*Assistant Professor, Thakur College of Engineering and Technology, Mumbai-400 101, India*

Abstract: *Data deduplication has gained increasing popularity as a space-reduction approach in backup storage systems. One of the main challenges for centralized data deduplication is the scalability of fingerprint-index search. In existing system, deduplication mainly focuses on backup system. In this paper, we propose a system that effectively exploits similarity and locality of data blocks to achieve high duplicate elimination and well balanced load at low RAM overhead in a runtime storage environment. The main idea is to employ an algorithm by considering random data chunks which are similar with adjacent data chunk, and eliminate duplicate chunks from the storage system, and then further enhance the resemblance detection efficiency. This would further help us to enhance the data delivery capabilities in cloud storage system with reduced latency and high throughput.*
Keywords: *Data Chunks, Data Deduplication, Cloud Storage.*

I. Introduction

Cloud computing provides a low-cost, scalable, location-independent infrastructure for data management and storage. The increasing data volume has led more and more people to pay attention to use the capacity of cloud storage than before. Thus, how to utilize the cloud storage capacity well, has becomes an important issue nowadays.

Cloud computing provides many virtualized resources to users, as services across the entire Internet, while hiding platform and implementation details. GMAIL is one of the best examples of cloud storage which is used by most of us regularly. Cloud service providers provide highly available cloud database storage and slightly parallel computing resources at a comparatively low cost. One difficult challenge of cloud storage services is the management of the ever-increasing amount of data. To control data management in cloud computing, deduplication has been a well-known technique and has attracted more and mostly technology used in today's world. Data deduplication is removal of redundant data. Thus, reducing the amount of data in actual physical storage reduces a lot of storage requirements costs, infrastructure management costs and power consumptions as well.

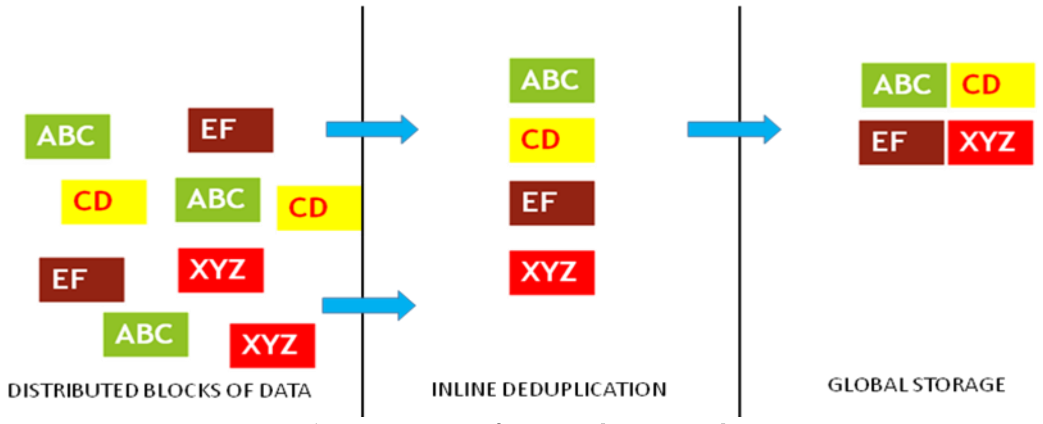

Fig 1. Overview of Data Reduction Technique

II. Related work

Data deduplication is an efficient data reduction approach that not only reduces storage space by eliminating duplicate data but also minimizes the transmission of redundant data in low bandwidth network environments. In general, a chunk-level data deduplication scheme splits data blocks of a data stream (e.g., backup files, databases, and virtual machine images) into multiple data chunks that are each uniquely identified and duplicate-detected by a secure SHA-1 or MD5 hash signature (also called a fingerprint). Storage systems then remove duplicates of data chunks and store only one copy of them to achieve the goal of space savings.

One of the well-known static chunking schemes is Venti [1]. Venti is a storage system using static chunking, where 160-bit SHA1 hash key is used as the address of the data. This enforces a write-once policy since no other data block can be found with the same address. The addresses of multiple writes of the same data are identical, hence duplicate data is easily identified and the data block is physically stored only once.

Another known content-defined chunking system is LBFS [2] which is a network file system designed for low bandwidth networks. LBFS exploits similarities between files or versions of the same file to save bandwidth. It avoids sending data over the network when the same data can already be found in the server's file system or the client's cache.

Delta encoding [3] stores data in the form of differences between sequential data. Lots of backup system adopts this scheme in order to give their users previous versions of the same file from previous backups. This reduces associated costs in the amount of data that has to be stored as differing versions.

DEDE [4, 5] is a decentralized deduplication system designed for SAN clustered file systems that supports a virtualized environment via a shared storage system. Each host maintains a write-log that contains the hashes of the blocks it has written. Periodically, each host queries and updates a shared index for the hashes in its own write-log to identify and reclaim storage for duplicate blocks.

In [6], they propose a data deduplication system using file modification pattern. This approach can detect how file is modified and what types of deduplication is best for data deduplication.

M. Bellare [7] designed a system known as DupLESS that combines a CE-type scheme with the ability to obtain message-derived keys with the use of a key server (KS) shared amongst a group of clients. These mechanisms ensure that DupLESS provides strong security against external attacks and that the security of DupLESS gracefully degrades in the face of comprised systems.

Aim of M. Bellare [8] is to formalize a new cryptographic primitive, Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure de-duplication, a goal currently targeted by numerous cloud-storage providers.

J. Xu [9] proposed growing need for secure cloud storage services and the attractive properties of the convergent cryptography lead us to combine them, thus, defining an innovative solution to the data outsourcing security and efficiency issues. The solution is also shown to be resistant to unauthorized access to data and to any data disclosure during sharing process, providing two levels of access control verification.

Issue with the Existing System

1. The existing system is limited to file level deduplication in real time scenarios, thus, leading to limitation in application level framework.
2. We propose a system to achieve data deduplication on block-level storage to achieve high performance in cloud storage with reduced overhead in real time system.
3. The proposed system will deliver efficient data access with reduced cost in terms of bandwidth usage in distributed environment.

III. Methodology

The proposed system focuses on use of block level deduplication technique to reduce the redundant chunks of data. The algorithm will read the input stream block by block to identify unique chunks of data blocks. An initial database will be constructed containing information of unique data chunks along with their frequency and physical address. The metadata from database will be used further to exploit the data distribution details with respect to their storage location, for further optimization. Each client in the environment will also run the same algorithm, thus, reducing the task at primary storage nodes. The metadata information from will be shared among clients to determine the density of data chunks with high frequency of hits to optimize the storage location. For reducing the index-lookup searching, we will use BST to reduce the time complexity of algorithm to enhance the performance.

Objectives:

1. To achieve deduplication at low RAM overhead for index-lookup with agent.
2. To achieve near-exact efficiency of duplicate elimination.
3. To reduce memory contention between bursty read and writes traffic.
4. To increase the throughput of storage system using density based information.
5. To obtain load balance among various storage nodes.

Block level deduplication is always efficient than file-level deduplication because in file level one has to dump whole file in data storage if its version changed where as in block level you have to dump the changed block which takes comparatively less space in data storage.

```
1. Start
2. Declare Variable
3. Initialize variable
4. Read 1024 bytes from file in tone iteration
5. Read from file until reach EOF
5.1 Generate Hash Value from strBuff[BLOCKSIZE]
5.2 if (FirstBlock)
      Consider node as root element
      Inc BlockCtr
    else
      search the generated Hash in BST
      if (Find Hash == True)
        Compute the Node
        Add the Node to a linked List
        Change the EndLink of SLL
      else
        Add the node in BST
        Inc The BLockCounter
6. End
```

IV. Expected Outcomes

The existing system is limited to IOPs handling on single storage node, hence performance enhancement is limited to known scenarios. All the data block handling operations are held on single node. This might result into bottleneck during peak hours, when there are bursty read and write operation. The proposed system along with distributed dedup agents, will allow to load balance the data flow across the storage nodes in cloud environment. Thus, reducing the CPU utilization to some extent on primary storage node. The system will also provide fault tolerance to the SAN environment to some extent in case of node failure. With distributed metadata across the client nodes, the data request can be completed from client nodes itself, for the data blocks with high frequency.

V. Conclusion

Data deduplication is a scalable and efficient data reduction technique for large-scale storage systems, which addresses the challenges imposed by the explosive growth in demand for data storage capacity. Data deduplication can reduce the number of disks used in the operation to reduce disk energy consumption costs. This paper provides a cloud based model for deduplication of large data. Apart from this, the proposed model also aims to provide security to the user data that has been stored in the Cloud Storage.

Reference

[1] S. Quinlan and S. Dorward, "Venti: a new approach to archival storage", Proceedings of the FAST 2002 Conference on File and Storage Technologies, (2002).

[2] A. Muthitacharoen, B. Chen and D. Mazieres, "A low-bandwidth network file system", ACM SIGOPS Operating Systems Review, vol. 35, no. 5, (2001), pp. 174-187.

[3] M. Ajtai, R. Burns, R. Fagin, D. D. E. Long and L. Stockmeyer, "Compactly encoding unstructured inputs with differential compression", Journal of the Association for Computing Machinery, vol. 49, no. 3, (2002), pp. 318-367.

[4] F. Douglis and A. Iyengar, "Application-specific delta-encoding via resemblance detection", Proceedings of the USENIX Annual Technical Conference, (2003), pp. 1-23.

[5] P. Kulkarni, F. Douglis, J. LaVoie and J. Tracey, "Redundancy elimination within large collections of files", Proceedings of the annual conference on USENIX Annual Technical Conference, USENIX Association, (2004).

[6] H. M. Jung, S. Y. Park, J. G. Lee and Y. W. Ko, "Efficient Data Deduplication System Considering File Modification Pattern", International Journal of Security and Its Applications, vol. 6, no. 2, (2012).

[7] M. Bellare, S. Keelveedhi, and T. Ristenpart, 2013, Dupless: Server aided encryption for deduplicated storage.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart,2013, Message-locked encryption and secure deduplication.

[9] J. Xu, E.-C. Chang, and J. Zhou,2013, Weak leakage-resilient clientside deduplication of encrypted data in cloud storage.

[10] http://blog.calsoftinc.com/2016/03/deduplication-algorithm_3.html.