RESEARCH ARTICLE OPEN ACCESS

Similarity and Location Aware Scalable Deduplication System for Virtual Machine Storage Systems

A.Stenila[1], M. Kavitha[2], S.Alonshia[3]
[1,2,3](Computer Application, Annai Vailankkani Arts and Science College, Thanjavur)

Abstract:

Cloud computing is widely considered as potentially the next dominant technology in IT industry. It offers basic system maintenance and scalable source management with Virtual Machines (VMs). As a essential technology of cloud computing, VM has been a searing research issue in recent years. The high overhead of virtualization has been well address by hardware expansion in CPU industry, and by software realization improvement in hypervisors themselves. However, the high order on VM image storage remains a difficult problem. Existing systems have made efforts to decrease VM image storage consumption by means of deduplication inside a storage area network system. Nevertheless, storage area network cannot assure the increasing demand of large-scale VM hosting for cloud computing because of its cost limitation. In this project, we propose SILO, improved deduplication file system that has been particularly designed for major VM deployment. Its design provide fast VM deployment with similarity and locality based fingerprint index for data transfer and low storage consumption by means of deduplication on VM images. And implement heart beat protocol in Meta Data Server (MDS) to recover the data from data server. It also provides a comprehensive set of storage features including backup server for VM images, on-demand attractive through a network, and caching through local disks by copy-on-read techniques. Experiments show that SILO features execute well and introduce minor performance overhead.

Keywords **— Deduplication, Storage area network, Load Balancing, Hash table, Disk copies**

I.INTRODUCTION

In this paper with the potentially unlimited storage space offered by cloud providers, users tend to use a large amount space as they can and vendors continually look for techniques aimed to reduce redundant data and exploit space savings. A technique which has been widely adopted is cross-user deduplication. The simple idea behind deduplication is to accumulate duplicate data only once. Therefore, if a user wants to upload a file which is already store, the cloud provider will insert the user to the owner list of that file. Deduplication has proved deduplication eliminates unneeded data segments from the backup and reduces the size of toalize high space and cost savings and many cloud storage providers are currently adopt it. the backup data. This is particularly useful in Cloud Storage Where data is transferred to the storage target over WAN. Deduplication with Cloud Storage not only reduces the storage space requirements, but also reduces the data that is transferred over the network resulting in earlier and capable of data protection operations. In Direct Deduplication to cloud, the cloud storage is defined as the storage target in the Media Agent. Deduplication is enabling either at the client or at the Media Agent. The deduplicated data is transferred to Cloud Storage library. The deduplication database resides on the Media Agent (or on a designated volume that is attached to the Media Agent). Deduplication can be enabled for derivative copies during Storage Policy Copy creation. In this setup, data on the prime copy is not deduplicated. The non-deduplicated data can be deduplicated by enabling deduplication on the secondary copy. The deduplication database resides on the Media Agent, and the deduplicated data is stored in the library. The deduplication process as shown in fig 1.

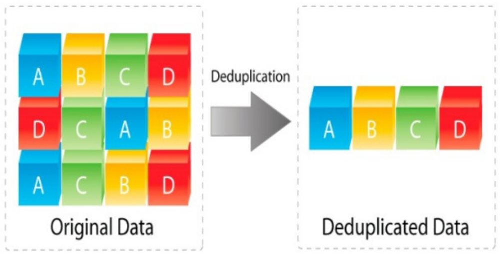


Fig 1: Deduplication process

II.RELATED WORK

D. Meyer et.al…, [7] studied file system data, metadata, and layout on nearly one thousand Windows file systems in a commercial environment. This new dataset contains metadata records of interest to file system designers; data content findings that will help create space efficiency techniques and data layout information useful in the evaluation and optimization of storage systems.

B. Debnath, et.al…, [5] design a flash-assisted in order deduplication system using ChunkStash, a chunk metadata store on flash. ChunkStash use one blaze read per chunk lookup and works in concert with RAM pre-fetching strategies. It organizes chunk metadata in a log-structure on flash to use fast sequential writes. It uses an in memory hash table to index them, with hash collisions resolved by a variant of cuckoo hashing.

W. Dong, et.al…, [6] presents super-chunk routing as an important technique for building deduplication clusters to achieve scalable throughput and capacity while maximizing effective deduplication and have investigated properties of both stateless and stateful versions of super-chunk routing.

E. Kruus et.al.., [4] proposed bimodal algorithms that vary the expected chunk-size dynamically and able to perform content-defined chunking in a scalable manner, involving a constant number of chunk existence queries per unit of input. Significantly, these algorithms require no special-purpose metadata to be stored.

G. Wallace et.al…, [3] confirm and things to see the different requirements between backup and primary storage. Whereas main storage capacities have grown quickly, write throughput requirements have not desired to scale as quickly because only a small percentage of the storage capacity is written every week and most of the bytes are longer lived.

III. FILE SYSTEM BASED DEDUPLICATION

According to the data granularity, deduplication strategies can be categorized into two main categories: file-level deduplication and block-level deduplication, which is nowadays the most common strategy. In block-based deduplication, the block size can either be fixed or variable. Another categorization criteria is the location at which deduplication is performed: if data are deduplicated at the client, then it is called source-based deduplication, otherwise target-based. In source-based deduplication, the client first hashes each data segment he wishes to upload and sends these results to the storage provider to check whether such data are already stored: thus only ”undeduplicated” data segments will be actually uploaded by the user. While deduplication at the client side can achieve bandwidth savings, it unfortunately can make the system vulnerable to side-channel attacks whereby attackers can immediately discover whether a certain data is stored or not. On the other hand, by deduplicating data at the storage provider, the system is protected against side-channel attacks but such solution does not decrease the communication overhead. The file system based deduplication can be described in fig 2.

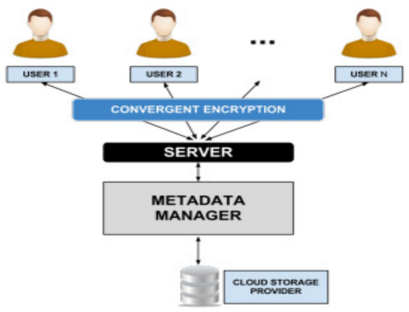


Fig 2: File system deduplication framework

IV. SILO BASED DEDUPLICATION SYSTEM

In deduplication framework, propose system implement block level deduplication system and named as similarity and locality based deduplication (SILO) framework that is a scalable and short overhead near-exact deduplication system, to defeat the aforementioned shortcomings of existing schemes. The main idea of SILO is to consider both similarity and locality in the backup stream concurrently. Specifically, expose and utilize more similarity through grouping strongly correlated small files into a division and segmenting large files, and leverage locality in the backup stream by grouping closest segments into blocks to confine similar and duplicate data missed by the probabilistic similarity detection. By keeping the parallel index and preserving spatial locality of help streams in RAM (i.e., hash table and locality cache), SILO is able to remove huge amounts of redundant data, dramatically reduce the numbers of accesses to on-disk index, and substantially increase the RAM utilization. This approach divides a large file into many little segments to better expose similarity among large files while increasing the efficiency of the deduplication pipeline. This architecture consists of four key functional components such as File Deamon (FD), Deduplication Server (DS), Storage Server (SS), and Backup Server (BS), which are distributed in the datacenters to serve the backup requests. BS and DS reside in the metadata server (MDS) while FD is installed on each client machine that requires backup/restore services and shown in fig 3.

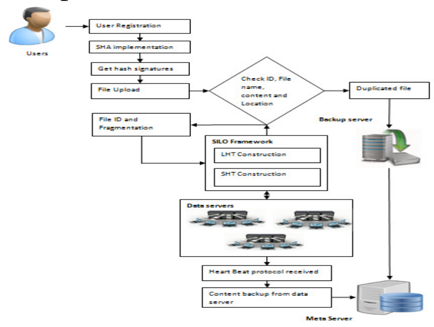
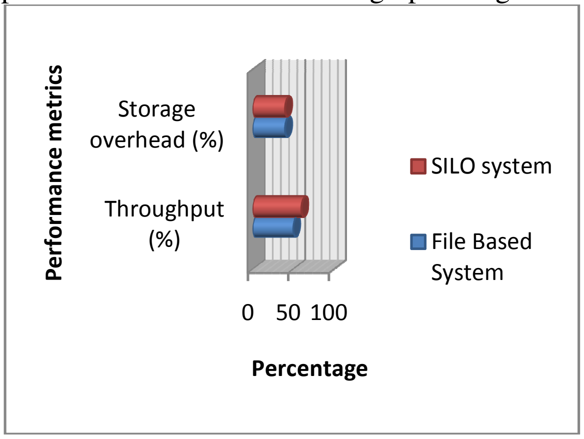


Fig 3: SILO Framework

V. EXPERIMENTAL SETTINGS

The proposed experiment can be measure the performance of the system using throughput and storage overhead metrics. Compare to existing approach, proposed SILO framework provide high level throughput and decrease the storage overhead problems and shown in below graph in fig 4.



VI. CONCLUSION

In cloud many data are stored again and again by user. So the user need more spaces store another data. That will reduce the memory space of the cloud for the users. To overcome this problem uses the deduplication concept. Data deduplication is a method for sinking the amount of storage space an organization wants to save its data. In many associations, the storage systems surround duplicate copies of many sections of data. For instance, the similar file might be keep in several dissimilar places by dissimilar users, two or extra files that aren't the same may still include much of the similar data. Deduplication remove these extra copies by saving just one copy of the data and replace the other copies with pointers that lead reverse to the unique copy. So we proposed Block-level deduplication frees up more spaces and exacting category recognized as variable block or variable length deduplication has become very popular. In cloud using the SHT and LHT tables the user easily searches the data and retrieves the searched data from the cloud. And implemented heart beat protocol to recover the data from corrupted cloud server. Experimental metrics are proved that our proposed approach provide improved results in deduplication process.

REFERENCES:

1. S. Quinlan and S. Dorward, “Venti: a new approach to archival storage,” in Proceedings of

the first Usenix Conference on File and Storage Technologies, 2002.

2. P. Kulkarni, F. Douglis, J. LaVoie, and J. Tracey, "Redundancy elimination within large collections of files," in Proceedings of the USENIX Annual Technical Conference, 2004, pp. 59–72.

3. G.Wallace, F. Douglis, H. Qian, P. Shilane, S. Smaldone, M. Chamness, and W. Hsu, "Characteristics of backup workloads in production systems," in Proceedings of the Tenth USENIX Conference on File and Storage Technologies, 2012.

4. E. Kruus, C. Ungureanu, and C. Dubnicki, "Bimodal content defined chunking for backup streams," in Proceedings of the 8^{th} USENIX conference on File and storage technologies. USENIX Association, 2010.

5. B. Debnath, S. Sengupta, and J. Li, "Chunkstash: speeding up inline storage deduplication using flash memory," in Proceedings of the 2010 USENIX conference on USENIX annual technical conference. USENIX Association, 2010.

6. W. Dong, F. Douglis, K. Li, H. Patterson, S. Reddy, and P. Shilane, "Tradeoffs in scalable data routing for deduplication clusters," in Proceedings of the 9th USENIX conference on File and storage technologies. USENIX Association, 2011.

7. D. Meyer and W. Bolosky, "A study of practical deduplication," in Proceedings of the 9th USENIX Conference on File and Storage Technologies, 2011.

8. A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," in Proceedings of the eighteenth ACM symposium on Operating systems principles. ACM, 2001, pp. 174–187.

9. M. Lillibridge, K. Eshghi, D. Bhagwat, V. Deolalikar, G. Trezise, and P. Camble, "Sparse indexing: large scale, inline deduplication using sampling and locality," in Proccedings of the 7th conference on File and storage technologies, 2009, pp. 111–123.

10. D. Bhagwat, K. Eshghi, D. Long, and M. Lillibridge, "Extreme binning: Scalable, parallel deduplication for chunk-based file backup," in IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. IEEE, 2009, pp. 1–9.