Data Deduplication in Cloud Computing Systems

Yingdan Shang[1,2]
[1]96616 unit
[2] National Lab for Parallel and Distributed Processing
National University of Defense Technology
Beijing, China
altsuzy-1101@163.com

Huiba Li[2]
[2]National Lab for Parallel and Distributed Processing
National University of Defense Technology
Changsha, China
lihuiba@gmail.com

***Abstract*—Cloud computing is a paradigm shift in the Internet technology. Data deduplication can save storage space and reduce the amount of bandwidth of data transfer. There always exists a trade-off between deduplcation efficiency and system performance since data deduplication also brings high system overhead. We analysed several latest studies on adopting data deduplcation technique to cloud system and pointed out the shortcomings of these existing work. From the result, we proposed several challenges and discuss the corresponding possible solution. We expect that our suggestions would achieve high deduplication efficiency and maintain a reasonable storage throughput.**

Keywords-data deduplication; cloud system; distributed processing

I. INTRODUCTION

Cloud Computing is an important transition in the Internet technology, it provides users with low cost computing and storage resources via internet in a pay-as-you-go manner, and also offers scalability and flexibility in terms of capacity and performance. Enterprise data volumes are exploding as organizations collect and store increasing amounts of information both for their own use and government regulations. However, much of the data in storage is duplicated data. Data De-duplication is the research hotspot in storage domain in recent years which can increase storage efficiency, save data management cost and reduce the amount of bandwidth required during data transmission, nowadays this technique has already been widely used in data back-up system.

In cloud system there are mass of virtual machine images and duplicated user data, data de-duplication technology can be used to eliminate the redundant data, reduces storage needs and increase the dispatch speed of virtual machine image.

A challenging issue of adopting deduplication to cloud system is to balance the trade-off between storage efficiency and performance. Deduplcation engine should be constructed to save the space while improve or do not harm the storage throughput. As a result of enormous size of cloud storage system, the deduplication will bring large amount of hash computation and hash index search which can easily become the bottleneck of the cloud storage system.

II. RALATED WORK

A. Data-deuplication efficiency

The most recent research work of Chulmin Kim[1] roughly analyse the benefit and overhead when we adopt data deduplication technique to the cloud storage system, and give some suggestion on how cloud storage architecture should be organized in a more smart way.

Since the overall storage of cloud system are used both for VM image store and user data store. As research shows, the deduplication efficiency of VM image repository is much higher than that of user data storage. In fact, the VM storage can get an inspiring deduplication ratio as high as 80% or even more [2].

Dutch T.Meyer [3] collected file system content data from 857 desktop computers at Microsoft over a span of 4 weeks and made data dedupcation in different ways, the results shown that file-level deduplcation can achieve three quarters of the deduplcation ratio of the block-level deduplication ratio. Figure 1 shows that as the size of the deduplcation domains increased, the deduplication becomes more effective. The most aggressive block-level deduplication can achieve a space saving as high as 70%.

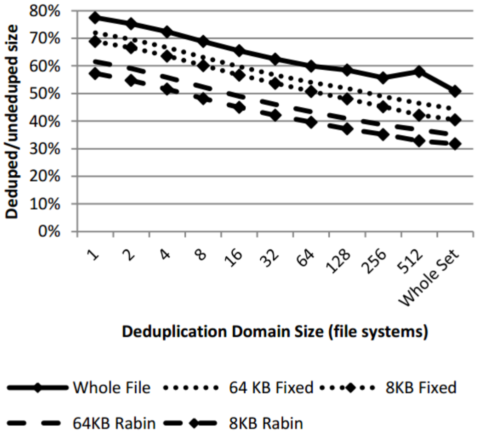

Figure 1. Deduplication efficiency on VM storage [3]

Although the deduplication efficiency of VM storage is quite high, the major part of entire cloud storage is user data repository, so block-level deduplication should be employed to ensure high deduplication efficiency while it also brings more performance overhead.

B. *Deduplication overhead*

Deduplication overhead mainly consisted of hash computation and hash index match. A number of works have been done to improve or do not harm the storage throughput in deduplication system. Hash computation overhead can be easily solved by parallelism technique. The most common forms of data deduplication implementation works by comparing data blocks to detect duplicates, and replace those duplicated data by a pointer to the existed data blocks. We use cryptographic hash functions such as SHA, MD5 to identify data blocks. To facilitate fast data block match, a single index containing all existed block hash values should be maintained in RAM, as the data grows and the index overflows the amount of available RAM, almost every index access will require a random disk access. This disk bottleneck [4] will severely reduce the storage throughput. Existed approaches exploit spatial locality, chunk locality, file similarity and so on and apply the method including index prefetching, chunk sampling, two-tier chunk index architecture to solve the disk bottleneck. As well, some experiment suggest to use flash-based solid state drives(SSD) to relieve the main memory requirement and realize a high deduplication throughput.

III. Preliminary study on deduplication in cloud system

A. *Live deduplication in Open-Source cloud*

Chun-Ho Ng's work LiveDFS [5] focus on the deduplication of VM image in the Open-Source cloud. LiveDFS can achieve at least 40% of storage saving for VM images storage while maintaining the reasonable performance in importing and retrieving VM images. To enable deduplication for VM images, LiveDFS mainly addressed three issues:

1. The performance of VM operations such as VM startup should be maintained.

2. General file system operations such as data modification and deletion should be considered while current data deduplication technique generally considers data to be immutable.

3. The deduplication solution should be compatible with the standard commodity hardware configurations.

The main design features of LiveDFS includes spatial locality, prefetching of metadata and journaling. LiveDFS is implemented as a Linux kernel space driver module which can be loaded to the Linux kernel without the need of re-compiling the kernel source code.

LiveDFS is deployed as a storage layer for VM images storage with deduplication, the week point is its limited scalability. It is designed as a branch of Ext3 file system and only employs deduplication in the same partition. LiveDFS designs an in-memory index structure called fingerprint filter to accelerate search process, furthermore, LiveDFS takes advantage of spatial locality by storing fingerprints with respect to the disk layout of the file system and prefetching the fingerprint store to tackle the disk bottleneck problem.

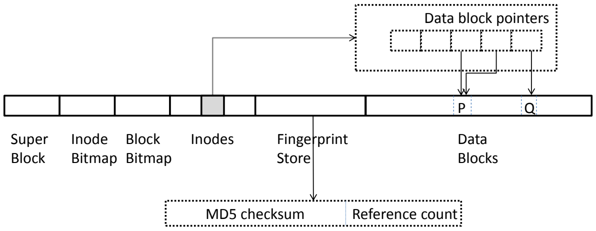

Figure 2. File system layout of LiveDFS

Figure 2 shows the deployment of a fingerprint store in a block group. A reference count is maintained for each data block, when the reference count is decremented to zero, LiveDFS deallocates the block.

Chun-Ho Ng made several experiments to evaluate the performance of LiveDFS including I/O throughput, storage efficiency, time for inserting VM images, time for VM startup. Compared LiveDFS with Ext3FS which does not support deduplication, LiveDFS can save more than 40% space while getting a reasonable I/O throughput.

B. *Extreme Binning and Distributed deduplication backup system*

Extreme Binning [6] is a scalable, parallel deduplcation solution for chunk-based file backup. It splits the chunk index into two tiers by exploiting file similarity. Extreme Binning uses sliding window technique chunk algorithm to chunk data stream into variable length chunk, then it chooses the representative chunk ID (hash value of chunk) for every file using Broder's theorem. Previous work has shown that Broder's theorem can be used to identify similar files with high accuracy. If two files are highly similar they share many chunks then their representative chunk ID is the same with high probability. Those representative chunk ID build up a primary index in memory and are used to identify the similar files. When a new file arrives, Extreme Binning only computes the representative chunk ID of the file and find it in main memory to determine the corresponding deduplication. Since the deduplication process is restricted only in the similar files, Extreme Binning do allow duplicates, it represents a trade-off between performance and duduplication effiency. Experiments has shown that Extreme Binning can get a perfect deduplication ratio while using less RAM, fewer disk accesses so as to maintain a high deduplication throughput.

In corresponding with the special two-tier chunk index structure, the distributed process in Extreme Binning simply use chunk ID mod K backup nodes to determine which backup node the chunk should go. So the architecture of a distributed deduplcation backup system using Extreme Binning is quite simple as the figure below shows.

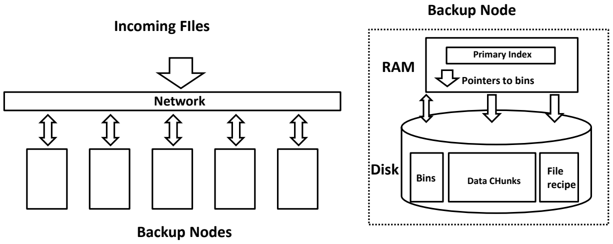

Figure 3. Extreme Binning Architecture [6]

C. SAM Framework for cloud backup

SAM [7] is a semantic-aware multi-tiered source deduplication framework for cloud backup system. Its motivation is based on two observations on the redundancy of backup data. One is that most duplicated data across clients is dominated largely by duplicate files instead of duplicate chunks, thus SAM proposed to combine global file-level deduplication with local chunk-level deduplication method instead of global chunk-level deduplication across clients. The second observation is many file-level semantic attributes can be used to improve deduplication throughput.

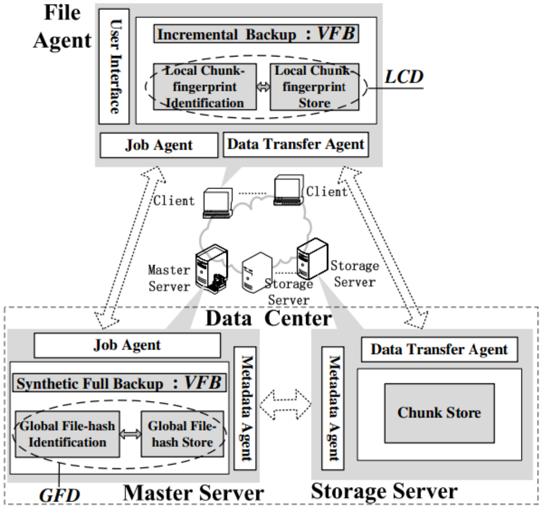

Figure 4. SAM system architecture [7]

The workflow of SAM consists of three phases, First, Virtual Full Backup removes unchanged files according to file time stamp. Second, Global File-level Deduplication (GFD) removes duplicated files cross clients in Master Server, during this process, SAM exploits file locality and file size to narrow the search space of redundant data, GFD excluded small files to both reduce disk accesses significantly and reduce files hashes need to be transferred from File Client to Master Server. Since preliminary studies show that 63.8% of the identical files are smaller than 8KB accounting for 0.53% of redundant data. LCD exploits file similarity using methods adopted from Extreme Binning.

Experts build a SAM simulator and use real-word datasets to evaluate the performance of SAM framework. Experiments are mainly consisted of deduplication efficiency, deduplication overhead, backup window, the benefits of exploiting file semantics. The results show that SAM get a high deduplication ratio which is as better as global chunk-based deduplication, and very low overhead than that of global chunk-based deduplication.

IV. DISCUSSION

Although LiveDFS made some progress in employing deduplication to cloud platform, but since cloud platform is typically a distributed system while LiveDFS implemented deduplication on a single storage partition, there are still lots of things to do in utilizing deduplication to cloud platform. Methods adopted by LiveDFS in solving disk bottleneck may not be suitable to a distributed system. Another limitation of LiveDFS is that it only targets for deduplication in VM images storage while the major part of cloud storage system is user data storage.

To resolve the ubiquitous disk bottleneck problem, Research [1] suggested to import hardware components like general purpose computing on graphics processing units (GPGPU) and flash-based solid state drives (SSD) to cloud platform. GPGPU can make hash computation parallel and free the system compute resources from hash computation bottleneck. SSD has better latency than that of a hard disk and relieve disk bottleneck at a certain extent.

To distribute and parallelize deduplication for cloud system, we can see from Extreme Binning that the distributed system architecture is strongly associated with its deduplication manner. A smart cloud architecture is quite neccessary.

SAM is a deduplication framework for cloud backup, there is lots of experience we can learn from. SAM uses more CPU power and storage space at the client site to reduce the heavy burden at the server side.

In cloud platform, we can only do file-level deduplication for VM images repository in order to get a high system throughput, as for cloud backup, a chunk-level deduplication is more appropriate due to the critical factor of storage efficiency. The file semantic can be used to intensely speed up the deduplication process by making use of file locality, file similarity, file size, etc.

V. CONCLUSION

By analysing several latest study on adopting data deduplcation technique to cloud system, we point out the shortcoming of these existing work and propose several possible solution. Since most previous deduplication work are concentrated on the centralized backup system, we hope our work would light the way to realize deduplication for cloud computing environment.

ACKNOWLEDGMENT

This work is supported by National Basic Research Program of China 2011CB302600 and National High Technology Research and Development Program of China 2011AA01A202. First and foremost, I would like to show my deepest gratitude to my supervisor, Dr. Xicheng Lu, a respectable, responsible and resourceful scholar, who has

provided me with valuable guidance in every stage of the writing of this thesis. Without his enlightening instruction, impressive kindness and patience, I could not have completed my thesis. I shall extend my thanks to Mr. Li for all his encouragement and support.

REFERENCES

[1] Kim C, Park K W, Park K S, et al. Rethinking deduplication in cloud: From data profiling to blueprint[C]//Networked Computing and Advanced Information Management (NCM), 2011 7th International Conference on. IEEE, 2011: 101-104.

[2] Jin K, Miller E L. The effectiveness of deduplication on virtual machine disk images[C]//Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference. ACM, 2009:7.

[3] Meyer D T, Bolosky W J. A study of practical deduplication[J]. ACM Transactions on Storage (TOS), 2012, 7(4): 14.

[4] Zhu B, Li K, Patterson H. Avoiding the disk bottleneck in the data domain deduplication file system[C]//Proceedings of the 6th USENIX Conference on File and Storage Technologies. 2008, 18.

[5] Ng C H, Ma M, Wong T Y, et al. Live deduplication storage of virtual machine images in an open-source cloud[C]//Proceedings of the 12th International Middleware Conference. International Federation for Information Processing, 2011: 80-99.

[6] Bhagwat D, Eshghi K, Long D D E, et al. Extreme binning: Scalable, parallel deduplication for chunk-based file backup[C]//Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on. IEEE, 2009: 1-9.

[7] Tan Y, Jiang H, Feng D, et al. Sam: A semantic-aware multi-tiered source de-duplication framework for cloud backup[C]//Parallel Processing (ICPP), 2010 39th International Conference on. IEEE, 2010: 614-623.

[8] Zeng W, Zhao Y, Ou K, et al. Research on cloud storage architecture and key technologies[C]//Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human. ACM, 2009: 1044-1048.

[9] Hocker C. Cloud De-Duplication Cost Model[D]. Ohio State University, 2012.

[10] Lokeshwari Y V, Prabavathy B, Babu C. Optimized Cloud Storage with High Throughput Deduplication Approach[C]//International Conference on Emerging Technology Trends (ICETT). 2011, 2: 1.