De-duplication Approaches in Cloud Computing Environment: A Survey

Fatemeh Shieh
Department of Computer Engineering, Mahallat Branch,
Islamic Azad University,
Mahallat, Iran

Mostafa Ghobaei Arani
Department of Computer Engineering, Parand Branch,
Islamic Azad University,
Tehran, Iran

Mahboubeh Shamsi
Department of Computer Engineering, Qom Branch,
University Of Technology
Qom, Iran

ABSTRACT
Nowadays increasing the data storage capacity is one of the important challenges, due to the more demands for using cloud services. There have been offered several approaches to identify and remove duplicated data in virtual machines prior to sending their data to a shared storage resource. Therefore method of storage information should be efficient also the method of finding data should be intelligent as much as possible. However, there is no approach among various storing data approaches, to be absolutely expected to have the best performance in the use of bandwidth for storage. One of the useful strategies to have fast and efficient data storage is de-duplication. In this paper, we will address various de-duplication approaches and consider advantages and disadvantages of them.

Keywords
Cloud computing, Virtual machine, Data storage system, De-duplication.

1. INTRODUCTION
Changing the business requirements and outburst of digital data has been launched huge demands for high volume and efficient data storage. Due to the limited financial resources and high electronic data storage expenses, users prefer to store their data in the cloud environments [1]. Cloud computing allows its users to transfer their data and applications on the web so that they will be able to operate those programs without any special necessary physical infrastructures [2, 3, 4]. There are limited storage and networking resources in the cloud system. The entire of cloud services which have been offered so far, allows users to stop problems by using the two important aspects of dependability and elasticity.

The other important aspect is using virtualization technology by cloud services [5, 6, 7].Virtual machines allow to increase services by transferring applications into clouds. Virtual machines are one of the key aspects to access elasticity. Both cloud services and online support services have huge amount of data which will be needed to store them continuously so that it is expected to have many duplicated data among them. Therefore removing duplicated data has an important role in cloud structures [8]. There are many techniques to remove extra stored data. Researchers' community has been interested in deleting duplicated data. The paper has been organized as follow:

Second section considers the concept of De-duplication. In third section we address various approaches for De-duplication. The fourth part is a comparison among various approaches and at last the fifth section is deducted to conclusion and future works.

2. DATA DE-DUPLICATION
Since the use of storage capacity has become an important issue in cloud storage, deleting replicated data can save storage capacity in cloud also helps to use it more efficiently [9]. De-duplication or deleting replicated data is a technique to simplify and improve management of data storage. De-duplication is a method for decreasing required storage capacity which causes to store just unique data. It is clear that De-duplication helps to decrease size of data center [2, 4, 5, 10].

De-duplication algorithm recognizes replicated data in the storage servers using hash codes. Hashing decreases the amount of complexity on two recorded data because the size of hash is so smaller than the data. Firstly server calculates signature of hash for each of entry then it searches for the signature among existing hash indexes on the system. Whenever server finds one entry for signature on the hash index, then server makes a reference for that which points to the block place on the disk. Otherwise server stores the record on the disk and adds a new entry of hash signature on the hash index [6, 8, 10]. De-duplication process decreases storage expense because smaller disks and so less disk purchase will be required. Less data means smaller support and more quick retrieval.

3. Various De-duplication Approaches
There have been offered various researches regarding finding and removing replicated data. In this section we introduce them summarizing their pros and cons.

3.1 Delta-Encoding Approach
Fred Douglis et al. [9, 11] proposed Delta encoding approach in 2003 to decrease redundancy among similar files. Delta has ever been used to calculate amount of change in special thing both in science and mathematics [12]. Delta encoding approach is able to remove redundant pieces from files and web pages. The approach accomplishes compression at the same time so it is called Delta compression. In Delta encoding approach, files will be compared by the fingerprints related to its file. This approach will save storage space substantially but it has relatively high cost [13].

3.2 Block Level De-duplication Approach

Wayne Sawdon et al. [14] suggested an approach in 2003 called Block Level De-duplication. This strategy divides file to several block with fixed or various sizes so it calculates the amount of hash to consider replicated blocks. Then it will be determined whether the calculated amount equals to the previously saved data blocks or not. (There will be created a digital signature and a unique code for each of data blocks using hash algorithm). If the amount of hash differs, the block will be saved on disk and its code will be saved on hash index, otherwise it will be referred to the real place of data block. It should be mentioned that the space required for the reference field is so smaller than the storing block frequently. So that it saves lots of capacity on disk. The approach is scalable with high efficiency. Also it has high throughput also it is a low cost operation [15, 16, 17].

3.3 REBL[1] Approach

J.M.Tracey and et al. [8, 18] suggested REBL method in 2004, which was in fact a combination of compression and elimination at the block level. It uses compression to remove repeated blocks and Delta encoding, as scalable and efficient approach, to eliminate repeated data. This approach is able to detect and eliminate replicated data. It has done by SHA digest and fingerprints of each block comparing each other [19,20]. This approach is able to decrease bandwidth used for storage data using elimination of unnecessary data also it is scalable approach. Its efficiency is high but its throughput is relatively low with high cost.

3.4 Fixed Size Block Approach

J.Lavoieet al. [18], offered a method called "fixed size block" which can find duplicates in the block. Using this method any update which changes part of file, cancels SHA-1 digest for the other blocks. As a result, the reference counter of the block is reduced and the SHA-1 digest is calculated for the new block. Using this approach increases the amount of storage space [21].

3.5 RSYNC[2]Approach

RSYNC method was offered in 2004[22] by Policroniades et al. to reduce the use of bandwidth also to update two files (were used on separate computers) with the same content. By this solution, receiver separates files inside blocks and calculates hash functions for each block. The sender receives hash blocks and compares them with hashed file blocks. So RSYNC sends data only to those blocks whose receiver has lost data. Also it sends data to the other files or blocks whose receivers are present there.

3.6 File Level De-duplication Approach

Kai Li et al. [23] invented an approach according to de-duplication at file level in 2008. De-duplication at file level can recognize repeated files easily using calculation a checksum for each of existed data in file and comparing to other checksums in other files. It is a fast and simple approach, but it has a low De-duplication rate so that its efficiency is relatively low too. Also it is a scalable method with high throughput. Method is able to save bandwidth for storage in addition to low cost [9, 15, 24].

3.7 Simple De-duplication Approach

"Simple De-duplication approach" [25], was promoted by Mr. João Tiago et al. in 2009. The approach detected and destroyed duplicated data on those servers with multiple virtual machines were running. Virtual machines store their pictures on a shared storage. This process reduced the amount of used memory of CPU and RAM at some extent. It was a scalable approach with relatively high efficiency also it can save required bandwidth for storage data. In addition it has high throughput with low cost. However this approach has a weak and low De-duplication rate whenever shared blocks need to be updated.

3.8 Super-fingerprint Approach

George Bebis et al. invented "Super-fingerprint method" to detect similar data in 2009 [26]. A Super-fingerprint is a group of fingerprints belonging to different parts of a file. As a Super-fingerprint is taken from several files, so that files with one or more similar super fingerprints will be similar.

3.9 Chunk Level De-duplication Approach

Kave Eshghi et al. [27] proposed an approach called de-duplication at chunk level in 2009. In the approach we use fingerprint of chunk which has been encoding by hash algorithm like MD5 and SHA-1 to compare chunks quickly. In this approach, however the chunk size is smaller, repeated items will be recognized more. But the number of comparisons will be increased to find replicated chunks. Increase in number of comparisons causes to increase cost [20]. It is a scalable approach with high efficiency. Also it can save bandwidth used for storage but its throughput is relatively low [28].

3.10 Server Side Based De-duplication Approach

Benny Pinkas et al. [29] proposed an approach in 2010 which remove replications from server side. In this method service provider does not have any De-duplication performance. It is a scalable approach but the amount of transformed data will not be decreased so there is no improvement in used bandwidth. It has relatively high cost. However it has high efficiency (see Figure 1).

[1] Redundancy Elimination at the Block level

[2] Remote synchronize approach

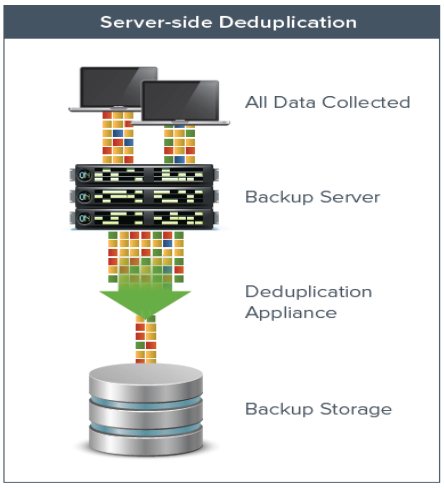

Figure 1: Server Side Based De-duplication [30]

3.11 Client Side Based De-duplication Approach

Stringham et al. [31] invented an approach in 2010 called de-duplication at receiver side which operates on data at the service receiver side exactly before sending them to server. It is a scalable method and improves bandwidth use for storage (Figure 2). However it imposes workload of extra calculations on the receivers who needs support. So its throughput is relatively low. On the other side, it has high efficiency and low cost [32].

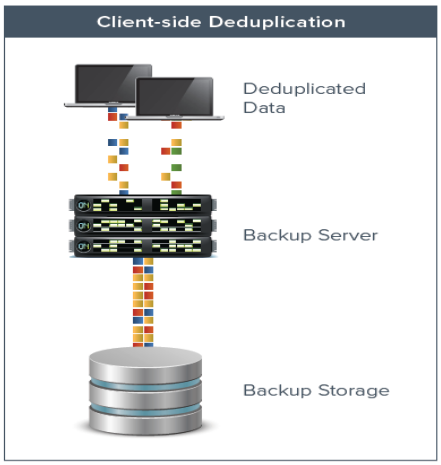

Figure 2: Client Side Based De-duplication [30]

3.12 FILE-CAC and BLK-FSC Approaches

F.Chen et al. [33, 34] invented two methods including "File-level Content-Aware Chunking" (FILE-CAC) and "Block-level Fixed-Size Chunking" (BLK-FSC) in 2011. Both approaches are scalable with high efficiency rate. Also their throughputs are high and they use small amount of bandwidth for storage. These are low cost operations. Researches on these methods showed that it was difficult to achieve high De-duplication rate in these two methods. Because the methods were not able to detect duplicated chunks at the time of inserting or removing a Chunk among others, hence there would be a relatively low de-duplication rate.

3.13 Whole File Hashing Approach

Meyer et al. offered an approach in 2012 which was related to the total hash in the content of file [35]. In this technique, all the files drive to a hash function. Hash function always encoding hash such as MD5 or SHA-1 encoded hash will be used to find duplicate files. Researchers evaluated the effectiveness of this strategy in the sections of similar data. If two files have the same hash, they will be duplicate content. The method replaced at the bottom of ranking compared to fixed block size and Chunking approaches. As in this method updating files, by calculating the SHA-1 digest for large amounts of data, causes to reduce throughput thus it is not able to save a considerable amount of space for storage. On the other hand it is a fast method with small amount of calculation so it has high efficiency and low cost[12,20].

3.14 BLK-CAC[3] Approach

Jin-Yong Ha et al. in 2013 [33] suggested a chunking scheme called Block Level Content Aware of Chunking "(BLK - CAC)" to increase the rate of De-duplication in the solid state drives (SSD). In this scheme, each block is divided into several chunks according to its content. They reviewed the results of related simulation, and concluded that rate of de-duplication in method of BLK-CAC is higher than the other similar methods. Also the method BLK-CAC, can more effectively serve large size files. It is a scalable approach with very high efficiency. Also it has very high throughput and it uses very small amount of bandwidth for storage. In addition its implementation has little cost.

4. COMPARING VARIOUSDE-DUPLICATION APPROACHES

We are going to compare various de-duplication approaches in Table 1 regarding to their scalability, efficiency, throughput, the amount of bandwidth used and cost. Hence, first, it is necessary to describe each of these parameters briefly. The scalability of a method is the ability to respond to the increasing number of requests entered for efficient data storage; also efficiency of a method means high-speed operation and low calculation needed. High throughput refers to storage more de-duplicated data per unit of time. The bandwidth used for data storage is the rate of bits used for data compare to total available bits in the storage space.

[3]Block Level Content Aware Chunking

Table 1.Comparing various de-duplication approaches

Approach	Scalability	Throughput	Efficiency	Amount of Used Bandwidth	Cost
REBL [8,18]	✓	Medium	High	Medium	High
Delta Encoding [9,11]	×	Low	Medium	Low	High
Compression(TGZ)[12,13]	×	Medium	Low	High	Very high
LBFS [36]	×	Medium	Medium	Low	High
De-duplication at Chunk level [27]	✓	Low	High	Low	High
De-duplication at file level[23]	✓	High	Low	Low	Low
Rabin Fingerprint [26,37]	✓	Medium	Low	High	Very high
Total Hash [35]	✓	Low	High	Medium	Low
De-duplication at Server Side [29]	✓	Medium	High	High	High
De-duplication at Service Receiver Side [31,32]	✓	Medium	High	Low	Relatively low
De-duplication Aware of Block Content [14,15]	✓	High	High	Very low	Low
FILE – CAC [33,34]	✓	High	High	Very low	Low
BLK – FSC [33,34]	✓	High	High	Very low	Low
Simple De-duplication[25]	✓	Very high	Very high	Very low	Low
BLK – CAC [33]	✓	Very high	Very high	Extremely low	Low

5. CONCLUSION

It is clear that De-duplication can help to decrease required space for storage. Hence it causes to decrease cost of storage as we need low-capacity disks to store data. So less data means less support and faster data retrieval. In the first section of this paper, described concept of De-duplication. In second part, addressed the various available De-duplication and their pros and cons and at last, compared the entire of described approaches in a table framework according to parameters like scalability, efficiency, throughput, amount of used bandwidth and cost. This paper can be a very helpful guidance for anew researcher who is going to study more on De-duplication.

6. REFERENCES

[1] ChengzhangPeng, ZejunJiangb. "*Building a Cloud Storage Service System*". scienceDirect, pp.691-696, 2011.

[2] M. Armbrust, A. Fox, R. Gri_th, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia , " *Above the clouds: A berkeley view of cloud computing*", Technical report, University of California at Berkeley, pp.2 , 2009.

[3] Monireh Fallah, Mostafa Ghobaei Arani and Mehrdad Maeen. "NASLA: Novel Auto Scaling Approach based on Learning Automata for Web Application in Cloud Computing Environment."*International Journal of Computer Applications* 113(2):18-23, March 2015.

[4] Monireh Fallah, Mostafa Ghobaei Arani"ASTAW: Auto-Scaling Threshold-based Approach for Web Application in Cloud Computing Environment." International Journal of u- and e- Service, Science and Technology (IJUNESST),Vol.8, No.3, pp.221-230, 2015.

[5] J. E. Smith and R. Nair, "*The architecture of virtual machines*", Computer, 38(5): 2005, pp.32-38, 2005.

[6] Afife Fereydooni, Mostafa Ghobaei Arani and Mahboubeh Shamsi, "EDLT: An Extended DLT to Enhance Load Balancing in Cloud Computing." *International Journal of Computer Applications* 108(7):6-11, December 2014.

[7] Behnaz Seyed Taheri, Mostafa Ghobaei Arani and Mehrdad Maeen, "ACCFLA: Access Control in Cloud Federation using Learning Automata. "International Journal of Computer Applications 107(6):30-40, December 2014.

[8] T. E.Denehy and W. W. Hsu, "*Duplicate management for reference data*", Technical report, IBM Research, 2003.

[9] Mishra, Deepak, and Sanjeev Sharma. "Comprehensive study of data de-duplication." *International Conference on Cloud, Big Data and Trust , Nov 13-15, RGPV* , 2013.

[10] Lin, Iuon-Chang, and Po-ChingChien. "Data De-duplication Scheme for Cloud Storage." *International Journal of Computer and Control (IJ3C), Vol1* 2 (2012).

[11] Kaurav, Neha. "*An Investigation on Data De-duplication Methods And it's Recent Advancements*." (2014).

[12] Deepu S R, BhaskarR, Shylaja B S, "*PERFORMANCE COMPARISON OF DE-DUPLICATION TECHNIQUES FOR STORAGE IN CLOUD COMPUTING ENVIRONMENT* ", Asian Journal of Computer Science And Information Technology 4:5 (2014), pp.42–46, 2014.

[13] InduArora, Dr. Anu Gupta. "*Opportunities, Concerns and Challenges in the Adoption of Cloud Storage*" ,*(IJCSIT)* International Journal of Computer Science and Information Technologies, vol 3(3), pp.4543-4548, 2012.

[14] Shilane, Philip, Grant Wallace, Mark Huang, and Windsor Hsu. "*Delta compressed and de-duplicated storage using stream-informed locality*." In Proceedings of the 4th USENIX conference on Hot Topics in Storage and File Systems, pp. 10-10.USENIX Association, 2012.

[15] Curran, Robert, Wayne Sawdon, and Frank Schmuck. "*Efficient method for copying and creating block-level incremental backups of large files and sparse files.*" U.S. Patent Application 10/602,159, filed June 24, 2003.

[16] He, Qinlu, Zhanhuai Li, and Xiao Zhang. "*Data de-duplication techniques*." In Future Information Technology and Management Engineering (FITME), 2010 International Conference on, vol. 1, pp.430-433. IEEE, 2010.

[17] Meyer, Dutch T., and William J. Bolosky, "*A study of practical de-duplication*." ACM Transactions on Storage (TOS) 7, no. 4 (2012): 14.

[18] Philip Shilane, Grant Wallace, Mark Huang, Windsor Hsu,"*Delta Compressed and De-duplicated Storage Using Stream-Informed Locality*" , Journal of Backup Recovery Systems Division EMC Corporation , pp.3 , 2012.

[19] M.Szeredi, "*File system in user space*." , 5th October, 2014.

[20] Purushottam Kulkarni, Fred Douglis, Jason LaVoie , John M. Tracey ,"*Redundancy Elimination Within Large Collections of Files*" , In Proceedings of the 2004 USENIX Annual Technical Conference, Boston, MA, pp.7-10 , June 2004.

[21] Kulkarni, Purushottam, Fred Douglis, Jason D. LaVoie, and John M. Tracey. "*Redundancy Elimination Within Large Collections of Files*." In USENIX Annual Technical Conference, General Track, pp. 59-72. 2004.

[22] P. Kulkarni, F. Douglis, J. LaVoie, and J. M. Tracey ,"*Redundancy elimination within large collections of files*", In ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference, USENIX Association , pp.5-5 , 2004.

[23] Policroniades, Calicrates, and Ian Pratt. "*Alternatives for Detecting Redundancy in Storage Systems Data*." In *USENIX* Annual Technical Conference, General Track, pp. 73-86. 2004.

[24] Zhu, Benjamin, Kai Li, and R. Hugo Patterson. "*Avoiding the Disk Bottleneck in the Data Domain De-duplication File System.*", In*Fast*, vol. 8, pp. 1-14. 2008.

[25] Deepu S. R., "*Performance Comparison of De-duplication techniques for storage in Cloud computing Environment*." Asian Journal of Computer Science & Information Technology 4, no. 5 (2014).

[26] Joao Tiago , Medeiros Paulo , Escola De Engenharia, Mestrado Em Engenharia ,"*Efficient Storage Of Data In Cloud Computing*" , Journal Of ACM Computing Surveys(CSUR) , pp.3-7 , July 2009 .

[27] Uz, Tamer, George Bebis, Ali Erol, and Salil Prabhakar. "*Minutiae-based template synthesis and matching for fingerprint authentication*." Computer Vision and Image Understanding 113, no. 9 (2009): 979-992 , 2009.

[28] Bhagwat, Deepavali, KaveEshghi, Darrell DE Long, and Mark Lillibridge. "*Extreme binning: Scalable, parallel de-duplication for chunk-based file backup*." In Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on, pp. 1-9. IEEE, 2009.

[29] Lillibridge, Mark, KaveEshghi, and DeepavaliBhagwat."*Improving restore speed for backup systems that use inline chunk-based de-duplication*.", In*FAST*, pp. 183-198. , 2013.

[30] K. Jin and E. Miller, "*The effectiveness of de-duplication on virtual machine disk images*", The Israeli Experimental Systems Conference, In Proc. SYSTOR,pp.6 , 2009.

[31] Harnik, Danny, Benny Pinkas, and Alexandra Shulman-Peleg. "*Side channels in cloud services, the case of de-duplication in cloud storage*." IEEE Security & Privacy 8, no. 6 (2010): 40-47, 2010.

[32] Stringham, Russell R. "*Client side data de-duplication*." U.S. Patent 7,814,149, issued October 12, 2010.

[33] Xu, Jia, Ee-Chien Chang, and JianyingZhou."*Weak leakage-resilient client-side de-duplication of encrypted data in cloud storage*". In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, pp. 195-206.ACM, 2013.

[34] Jin-Yong Ha, Young-Sik Lee, and Jin-Soo Kim, "*De-duplication with Block-Level Content-Aware Chunking for Solid State Drives (SSDs)*", IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing ,pp.2 , 2013.

[35] A. Gupta, R. Pisolkar, B. Urgaonkar, and A. Sivasubramaniam, "*Leveraging value locality in optimizing NAND flash-based ssds*", in Proc. USENIX Conference on File and Storage Technologies, 2011, pp. 7–7 , 2011.

[36] Meyer, Dutch T., and William J. Bolosky. "*A study of practical de-duplication.*" ACM Transactions on Storage(TOS) 7, no. 4 (2012): 14.

[37] A. Muthitacharoen, B. Chen, and D. Mazières , "*A low bandwidth network file system*", In Proceedings of the 18thACM Symposium on Operating Systems Principles (SOSP'01), pp. 174–187 , Oct. 2001.

[38] P. Kulkarni, F. Douglis, J. LaVoie, and J. M. Tracey," *Redundancy elimination within large collections of files*", In ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference, USENIX Association, pp.5-5 , 2004.