International Journal of Computer Engineering and Information Technology

VOL. 8, NO. 5, May 2016, 85–89
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

A New Technique for Optimal Data Manipulation through WANs

Abdelrahman M. Ahmed[1] and Hazem M. El-bakry[2]

[1, 2] Faculty of Computer Science & Information system, Mansoura University, EGYPT

ABSTRACT

One of the most familiar challenges of the cloud computing is the availability of the service or connectivity issues like internet speed or poor infrastructure, so in this paper we will present a new technique in data transformation through Wide Area Network (WAN) to accelerate the traffic between two or more data centers to get the high level of maturity of our cloud services, so the proposed algorithm will inspect the original data in high level archichture and running specific reference indexing technique to convert the original data to meta data after that the algorithm with run compression technique to compress the metadata itself this is from the source datacenter, on another side we the compressed metadata will transferred to destination datacenter then the algorithm will decompressed the file to metadata then convert back the metadata file by using the referencing table to original file, so this algorithm will assist us to transfer a lot of data between datacenters through poor WAN links rapidly.

Keywords: *Datacenters, Reference-table, Metadata, DBMS and SHA-1.*

1. INTRODUCTION

In the third world countries, we suffering about the poor infrastructure in these countries so the challenge here is "How the corporations enable their business in this countries?" now all of the multinational corporations working with cloud computing technologies "Public cloud computing, Private cloud computing and Hybrid cloud computing" now we need to centralize the services and the all accessed it from different locations around the world so we need to enhance the availability these services to be accessed from anywhere.

With the continuous increasing in the size of data now the transferring through WAN will be difficult and the digital data size in continuous increasing, so to avoid this challenge the only way increase the WANs speed witch is very expensive so now there was an technique called with WAN optimization or WAN acceleration and a lot of technological companies released this products like "Riverbed, CISCO,….etc" these products used the data de-duplication technique, in this technique the system just send the unique chunks of the data or the bytes that duplicated form the sources and store a pointer for each one and duplicate on the another side, almost of this systems are working with the data type "the compression ratio is based on data type itself" also these systems are very expensive.

The proposed algorithm will work with another approach "not data de-duplication" the proposed system will inspect the original file on block level "Byte", lookup for this bytes form the reference database to get the record number for each blow, collect the metadata "Pointers or record numbers" onto data file, then compressing this file by compression technique and send it to destination site.

When the compressed file received by the destination site the system received the file then decompressed it into metadata "Pointers or record numbers" after that the system divided the metadata onto set of database queries to get the bytes of the original data.

2. RESEARCH METHODOLOGY

As search in online resources that related to data deduplication, to know how it actually work, to know the methodology itself, so it's very difficult because the all of corporations that working in this area not announcing for own systems, so we going to the scientific journals and papers for the data storage techniques and cloud computing to get the high level of maturity to proposed a new approach to facilitate the working with this cloud challenge "Availability in third world countries – internet speed and availability" therefore am used the below online journals for searching:

- Academic Search Premier;
- Business Source Premier;
- Emerald Full text;
- Ingenta Journals;

- Science Direct; and
- IEEE Transaction.

3. RESEARCH CHALLENGES

Increasing developing data causes many challenges to the existing storage systems and data deduplication. Some challenges that faces data deduplication are [4]:

Cost-effective Remote Replication

Using standard technology, remote data replicate ion is unrealistic in many situations due to data volumes over constrained bandwidth h. With data reduction, unique deduplicated data is transferred achieving result a 90-95% reduction in transferred data, thus making replication over a WAN realistic.

Unified Data Protection

With Unified Data Protection (UDP), backup administrators can achieve high level of optimization of storage requirements and bandwidth, also accelerate protection and recovery across multiple sites. In addition, the solution enables in-place data re-hydration, for fast granular restore, including from tape.

Complete backups faster.

Because of less data transmitted and stored, backups are faster. This is important for situations where the total data amount threatens to take so long to backup that one backup isn't finished before the next one is due to start.

Improve client performance.

The agent has a minimal impact on client performance because the majority of processing is done in the Point Server Recovery. For virtualized environments, agentless VMware and Hyper-V backup reduces the risk of bottlenecks and performance issues at the hardware level. In other words, it ensures that backups don't stall virtualized servers.

Simplify infrastructure of backup.

By routing backups through the Recovery Point Server, it's a lot easier to transfer backup data to different places, for example Azure or Amazon cloud Web Services or to local tape or another offsite private cloud.

Improve availability and resilience.

Because the deduplication data is stored in the Recovery Point Server, it is a lot easier to protect the backup infrastructure. For example, all the information in the store can be replicated to a cloud, another server in the same data center or offsite.

Reduce required bandwidth for backups.

The Recovery Point Server pulls new or changed data only from a client with high levels of granularity – reduced to 4kb chunks only. This makes backups extremely efficient.

4. RELATED WORK

Deduplication technique based on chunking has been used in a large scoop today specially for the secondary storage. The using mechanism of this technique is depending on dividing a certain file to identical chunks then removes any duplicate chunks using a hash digests for all chunks (an example of these hash digests is SHA-1 or MD5) [5]. Many researchers have discussed depduplication techniques and method of implement by using either fix-size chunking [5, 6] or bimodal chunking [2] or content defined chunking [7]. Briefly it has become a fact that chunking techniques are considered great and widely used to develop duplication removal when small-sized files updates stored.

Novel studies [8] show that the Content-Defined Chunking (CDC) techniques can actually remove duplicate data till 10 % more that the Fixed-Sized Chunking (FSC) technique in the dataset of the primary storage.

The large scoop of using deduplication techniques is now a very important issue in the storage system that faces a big problem of the increasing size of data the fingerprint-lookup disk in the same time. A previous study discussed the DDFS which depend on exploiting the locality of inherent backup streams to limit the possible accesses to "on-disk index" which also enable to get rid of disk bottleneck of inline deduplication [9]. Another discussed deduplication solution is "sparse Indexing" that uses sampling approaches to minimize fingerprint index in the memory as it requires apparently half of the required RAM used by DDFS. Data sampling and chunk locality of backup streams is the method that sparse indexing uses to duplicate removal.

Chunkstash is another example of the techniques that use the chunk fingerprint storage on an SSD not on an HDD to speed up the index-lookup stage [10]. It exploits stream locality and runs cuckoo hash in the fingerprint index in RAM [9].

Today, primary storage based on data deduplication has a great attention because of its ability to reduce the space needed to store data which increase the empty space in the storage device. There are two new models of SSD deduplication, these models are the CA-SSD and CAFTL. These two models can improve the writing method by removing duplicate writes and any other duplicate data.

Virtual machines also can be involved in data deduplication techniques which is implemented by a solution called VMFLOCK . Now there are also many open source applications for data deduplication such as opendedup and ZFS dedupe.

5. DESIGN AND IMPLEMENTATION

We implemented source-based deduplication network file system using Byte-index based chunking approach and file transferring. In source-based approach, data deduplication process is performed in the source side and the source sends only non-duplicated files or blocks to the destination. The source performs file data deduplication process by checking file hash key at a reference database by a DBMS (Database Management System).The source starts block-level deduplication, divides a file into several blocks and calculates hashes of the each block. The list of hash keys is stored in a reference database, DBMS checks duplicated blocks by comparing the hash key with hash keys stored in the database to generate a metadata file. Finally, the source sends the metadata to the destination which uses the same techniques to generate the original file. All these transactions are shown in figure 1.

If file information has already saved to the DBMS, server accumulates them and produces Index-table of file. After that server sends produced Index-table to the client. Using this Index-table, client retrieves the duplicated chunks with high probability in very fast time. The client calculates the SHA-1 hash values from this result, and then the client also sends back them to the server. The server confirms whether the chunk is actually duplicated or not by comparing these hash values to the corresponding hash values from server DBMS. Finally the proposed system knows the non-duplicated region and duplicated region of a file and then accumulates all this information and transfers only non-duplicated region information to the client.

We can see how that can be done in the following algorithm;

```
Start
      Read the file directly from the disk;
        Looking up for each block in the cash database then
        in the Reference database;
           Extract Metadata;
           Text file "Metadata". Save ();
           Text file. Compress ();
           Text file. Transfer ();
     If (Text file arrives to destination);
   {
     Text file. Decompress ();
     Retrieve the metadata from the cash database or the
     Reference database;
     Generate the original file;
   }
End
```

The following steps illustrate the algorithm and show how it works:

5.1 Store file in binary code

The first step starts when the sender gets the file, it copies the file and store a copy in the hard disc that is formatted by the NTFS (New Technology File System).NTFS stores data as a binary code in one or more block depending on the file size, each block size is 1k bits and stores only one binary code for only one file, this helps to achieve the maximum use of the block space. After storing the file in the binary form, an application is run to read it directly from the hard disc in its binary form.

Look up for metadata in the reference database(searching and retrieving from Reference)

When the application reads the binary code of the file, we come to the second step which is Looking up for metadata in a reference database, this process is performed by the DBMS which searches in the reference that consists of multiple databases, each database consists of huge number of records with all probabilities of files to be sent between the source and the destination, each record is a number refer to blocks that previously used to store the file in the binary form. Once all the numbers of the blocks identified, the DBMS generates a text file with all these numbers, this text file is called metadata. To accelerate the process of metadata extraction, DBMS creates a cash database to store every generated metadata and every binary code for every sent file. When the source sends a file, DBMS searches first in the cash database for the metadata. If the metadata exists in the cash database it will be used for next step and if it didn't exist DBMS will search in the reference database for it. Figure 2 shows how a table in the reference database looks like.

5.2 Compress the text file

Then in the third, when the DBMS extracts the text file, it is compressed using a lossless compression technique such as Huffman coding technique to reduce its size. This step helps in time saving as the more of file size decreased the more of transferring time saved.

Transfer the file to the destination

The fourth step is sending this compressed text file from the source to the destination through the network which can be via cloud computing or regular LAN, WAN or any other type of networks.

5.3 Decompress the text file

When the file arrives to the destination we come to the fifth step as the receiver decompresses the file using the same technique used by the sender to compress.

Look up for the binary code in the reference database (searching and retrieving from Reference)

After decompressing the text file and in the sixth step, the receiver Looks up for the binary code in the reference database depending on the metadata from the text file. This process is also performed by a DBMS in the same way the source uses. Once the DBMS generates the file, it stores the received metadata and the binary code in a cash database to accelerate the process of file generation as each time the destination receives a file, DBMS will search first in the cash database for the binary code. If the metadata exists in the cash database it will be used for next step and if it didn't exist DBMS will search in the reference database for it.

5.4 Save the binary code to the hard disc

In the seventh step the receiver saves the binary code to the hard disc also using NTFS file system with 1k bits size for every block. After saving the binary code to the disc, the system will be able to generate the original file back. After generating the binary back to the original file an acknowledgment can be sent from the destination to the source to ensure that the file arrived correctly. A good example of a type of acknowledgment that can be used is SHA-1 algorithm which is considered one of the best techniques used in cryptography and data transferring to verify and acknowledge the arrival of data correctly.

By converting the file back to its original form, the transferring operation is completed after seven several processes as shown in figure 3.

6. CONCLUSION

Our work in this paper presents a structured model for a new technique for file transferring between datacenters. The proposed solution provides great results to improve the performance of file transferring and decrease needed time for that.

The key point of our system is to chunking the file then find out identical data in fast time, looking up for its probabilities and eliminate identical chunks. We use file-index chunking as a chunking technique in our system. Experiment result shows the proposed system can achieves very high file transferring capability. In this field of searching there are many issues still open, as in our future work we planning to build massive deduplication system and file transferring system with huge number of files. Such huge systems will definitely need more elaborated model to handle identical files information.

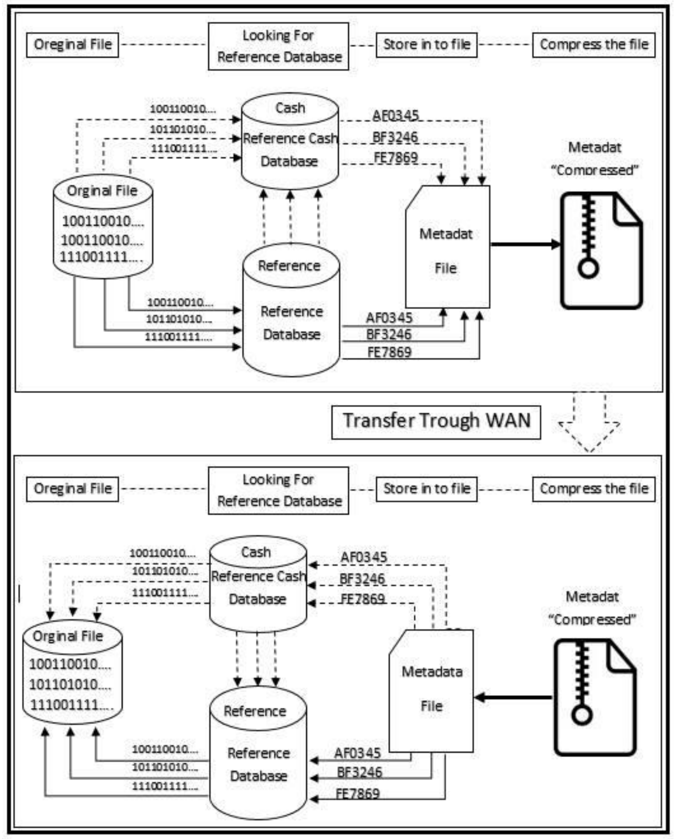

Fig. 1. System Architecture for the stages.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200

Fig. 2. blocks numbers stored in the reference

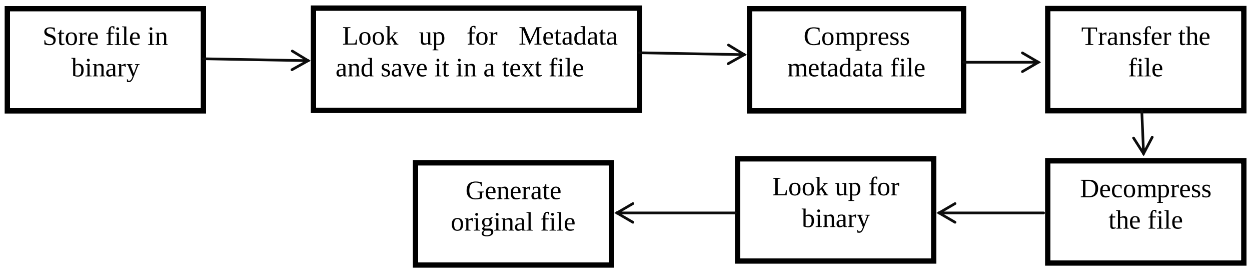

Fig. 3. The seven steps of the algorithm

REFERENCES

[1] Guanlin Lu, Yu Jin, and David H.C. Du, Department of Computer Science and Engineering University of Minnesota, Twin-Cities Minneapolis, Minnesota, USA, pp.1- 10, Frequency Based Chunking for Data De Duplication, 2010.

[2] Erik Kruus, Cristian Ungureanu, Cezary Dubnickim, NEC Laboratories America, pp.1-14, Bimodal Content Defined Chunking for Backup Streams,2009

[3] Ider Lkhagvasuren, Jung Min So, Jeong Gun Lee, Chuck Yoo and Young Woong Ko, International Journal of Security and Its Applications, Vol.7, No.5, pp.415 – 424, Byte-index Chunking Algorithm for Data Deduplication System, 2013.

[4] Christophe Bertrand, Arcserve CO. , pp.1-5, Modern Data Deduplication with Arcserve UDP,2015.

[5] Dutch T. Meyer, William J. Bolosky, Microsoft Research and the University of British Columbia,pp.1 – 13, A Study of Practical Deduplication, 2012.

[6] Keren Jin, Ethan L. Miller, Storage Systems Research Center, University of California, Santa Cruz, pp. 1-12, The Effectiveness of Deduplication on Virtual Machine Disk Images.

[7] Chuanshuai Yu, Chengwei Zhang, Yiping Mao, Fulu Li, Huawei Technologies Co., Ltd, pp.1-12, 978-1-4673-7619-8/15/$31.00 ©2015 IEEE Leap-based Content Defined Chunking --- Theory and Implementation,2015

[8] Vasily Tarasov, Deepak Jain, Geoff Kuenning, Sonam Mandal, Karthikeyani Palanisami, Philip Shilane, Sagar Trehan, Erez Zadok, Appears in the Proceedings of the 2014 Ottawa Linux Symposium (OLS'14), pp.1-12, Dmdedup: Device Mapper Target for Data Deduplication, 2014.

[9] Jin-Yong Ha, Young-Sik Lee, and Jin-Soo Kim, 2013 IEEE International Conference on High Performance Computing and Communications, pp. 1982-1990, Deduplication with Block-Level Content-Aware Chunking for Solid State Drives (SSDs), 2013.

[10] Samer Al-Kiswany, Dinesh Subhraveti, Prasenjit Sarkar, Matei Ripeanu, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.380.8529&rep=rep1&type=pdf , VMFlock: Virtual Machine Co-Migration for the Cloud,2012.

[11] Xu Xiaowei, Bachelor's Thesis in Information Technology, pp 1-60, Data Deduplication in Windows Server 2012 and Ubuntu Linux,2013