Quality of Service Enhancement by Using an Integer Bloom Filter Based Data Deduplication Mechanism in the Cloud Storage Environment

Kuo-Qin Yan, Yung-Hsiang Su, Hsin-Met Chuan, Shu-Ching Wang, Bowei Chen

Quality of Service Enhancement by Using an Integer Bloom Filter based Data Deduplication Mechanism in the Cloud Storage Environment

Kuo-Qin Yan[1], Yung-Hsiang Su[1], Hsin-Met Chuan[2], Shu-Ching Wang[1*], Bo-Wei Chen[1]

[1] Chaoyang University of Technology, Taiwan, R.O.C.
{kqyan, s10033905, scwang*, s10114603}@cyut.edu.tw
[2] Hsing-Kuo University, Taiwan, R.O.C.
hn88780752@yahoo.com.tw
*: Corresponding author

Abstract. Network bandwidth and hardware technology are developing rapidly, resulting in the vigorous development of the Internet. A concept, cloud computing, uses low-power hosts to achieve high quality service. According to the characteristics of cloud computing, the cloud service providers can support the service applications on the Internet. There are a lot of applications and data centers in the cloud-computing environment, then the loading of storage node is heavier than before. However, data deduplication techniques can greatly reduce the amount of data. Therefore, an integer Bloom Filter based Lightweight Deduplication Mechanism (LDM) under cloud storage is proposed. The proposed LDM can reduce the extra cost that traditional data deduplication technique needed.

Result

As network bandwidth and quality outstrip computer performance, various communication and computing technologies previously regarded as being of different domains can now be integrated, such as telecommunication, multimedia, information technology, and construction simulation. Thus, applications associated with network integration have gradually attracted considerable attention. Similarly, cloud computing facilitated through distributed applications over networks has also gained increased recognition. In a cloud-computing environment, users have access to faster operational capability on the Internet [2], and the computer systems must have high stability to keep pace with this level of activity.

In the data deduplication strategy, the block-level strategy is used. The block-level strategy has higher reduction ration than file-level but must consume more computing resource that is not suitable in cloud computing [3]. In this study, an integer Bloom Filter [4] based Lightweight Deduplication Mechanism (LDM) under cloud storage is presented to solve this problem. In the proposed LDM system architecture, several storage nodes are grouped into a cluster, each cluster has a

storage node acts cluster are called Namenode, the other storage node are called Datanode. Namenode is responsible for perform LDM and placement data to Datanode, Namenode also provides the storage capacity. Each Namenode has two components, LDM and Metadata Table. The LDM is used to perform deduplication strategy, and Metadata Table is responsible for the metadata of stored data. The proposed system architecture is shown as Fig. 1. The main job of Transfer Agent System (TAS) is used to upload data when the requests of user are received. The data is partitioned into n chunks and translated into unique identifier by SHA-1. Then, the unique identifier of data chunks is delivered to cluster by TAS for comparison data.

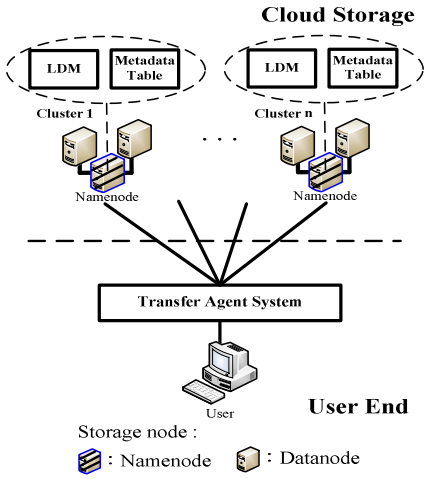

Fig. 1 The LDM System Architecture

In the proposed system architecture, each cluster executes LDM. The steps are described in below:

1. Set the number of hash function (k). For example, Cloud Service Provider (CSP) chooses seven hash functions.
2. Calculate the quantity of data that each cluster can store (n_{max}). For example, if the capacity of cluster is 40 GB and data chunk capacity is 64 MB, then this cluster can store 640 data (40 GB/64 MB).
3. Calculate the length of Bloom Filter (m) by using:
   $k=(m/n)\ln 2=m/n*0.69314$ (1)
   Through formula (1), Bloom Filter length (m) is 6463.
4. Calculate the probability of "Positive False" by using:
   $f'=(1-(1/m)^{kn})^k$ (2)
   Through formula (2), the probability of "Positive False" is 0.0078 that means the average 1000 times query will occur 8 times "Positive False".
5. CSP chooses an accept probability of "Positive False", and then the initialization of Bloom Filter is completed. If CSP cannot accept an accept probability of "Positive False", then return to Step 1. In addition, a bigger k is chosen that can get smaller probability of "Positive False".

A linked structure is used by LDM to store the metadata. When a data is added, to calculate the unique identifier by k hash functions and mapping to Bloom Filter firstly, then the hash value is computed to decide the starting position of link. The calculator formula is shown in (3), where $val(h_i)$ is the i_{th} hash value, m is Bloom Filter length, % symbol is mod function.

$$\text{Link(position)} = (\sum_{i-1}^{k} val(h_i)) \% m \quad (3)$$

By formula (3), a position in Bloom Filter can be obtained, and then link to the metadata of data chunk. For example, if there are 4 hash functions, $val(h_1) = 6069$, $val(h_2) = 36$, $val(h_3) = 5$, $val(h_4) = 642$, and m =6463, by using formula (3), the link start position is 289 [(6069+36+5+642) % 6463 = 289], therefore, this data will be linked to intBF index number 289. When the same result is gotten by different data through formula (3), then a links is generated by the last linked list. The progression of LDM is shown as Fig. 2.

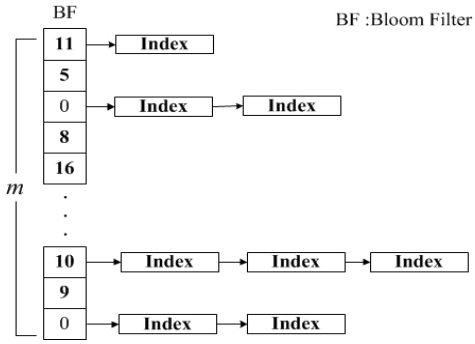

Fig. 2 The progression of LDM

In the data deduplication, the process of comparison is same as the process of data added. In the comparison process, the unique identifier is calculated by k hash functions firstly. Then, LDM judges the status of data existence by using intBF. If this data is not exist, do not perform the compare process, else if intBF judge this data is existence, to find the link position through formula (3), and search all link nodes to find the same data. However, the average comparison times in data duplication of our proposed LDM is compared with iBF (indexed Bloom Filter) proposed by Antichi et al. [1]. The capacity of storage node was generated by NS2, include current consume capacity and maximum capacity. The unique identifier was generated by md5-database [5], md5-database can transform data in to a unique identifier by MD5, SHA1, SHA256, and SHA512. However, SHA-1 was used in this experimental. There are 250 original data, include 150 movie data (2441 data chunk after partition), and 100 backup data (2518 data chunk after partition). Original data are partitioned by 64 MB, and there are 4959 data chunks totally. C++ is used to write the proposed mechanism. Finally, MS Excel 2010 is used to record experimental result. In addition, when a new data is uploaded, the unique identifier of data will be compared with storage system. In this experiment, data deduplication is performed in storage system which stored different number of data (25%, 50%, 75%, 100% of total data).

In this experiment, there are 20 data be uploaded in four kinds of situation and perform the data deduplication comparison processes. Fig. 3 is the results of average comparison times. X-axis is the different number of data; Y-axis is the average comparison times.

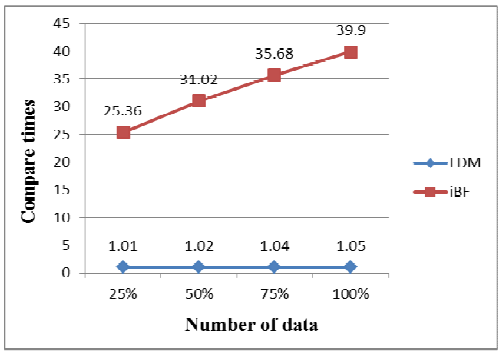

Fig. 3. The average comparison times in Data Duplication

According to the results shown in Fig. 3, whether in more or less number of data, the comparison times in LDM were keep in a constant time nearly. From the experiment results, LDM can use less computing resource to complete data deduplication process that is more suitable in cloud storage. In this study, LDM through rapid judge the status of data existence and compare the same data position can reduce the computing resource of data deduplication compare process, and then LDM more suitable in cloud computing.

Acknowledgments. This work was supported in part by the Ministry of Science and Technology MOST 102-2221-E-324-008 and MOST 103-2221-E-324-025.

References

1 Antichi, G., Pietro, A.D., Ficara, D., Giordano, S., Russo, F., Vitucci, F.: Achieving Perfect Hashing through an Improved Construction of Bloom Filters. In: the IEEE International Conference on Communications, pp. 1-5, May (2010).

2 Aymerich, F.M., Fenu, G., Surcis, S.: An approach to a cloud computing network. In: the 1st International Conference on Applications of Digital Information and Web Technologies, pp. 113-118, August (2008).

3 Tsuchiya, Y., Watanabe, T.: DBLK: Deduplication for Primary Block Storage. In: the of IEEE 27th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-5, (2011).

4 Wang, S.C., Wang, S.S., Yan, K.Q.,Chen, B.W.: HDDS: Hybrid data DeDuplication Strategy over Cloud Storage. In: the *International Conference on Innovation and Management*, pp. 103, July (2012).

5 Unique Identifier Database, http://md5-database.org/sha1/