Proceedings of
INTERNATIONAL CONFERENCE ON INNOVATIVE TRENDS IN ENGINEERING RESEARCH (ICITER-2016)
International Journal of Innovations in Engineering, Research and Technology,IJIERT-ICITER-16,ISSN:2394-3696
26th June,2016

DEDUPLICATION WITH MINIMUM WORKLOAD ON CLOUD SERVER APPROACH

[PAPER ID: ICITER-D193]

MOHITTAWARE
Sinhgad Academy Of Engineering,Pune, India
Email: dr.mohittaware9028@gmail.com

PROF.S.B. RATHOD
Sinhgad Academy Of Engineering, Pune, India
Email: sbrathod.sae@sinhgad.edu

ABSTRACT:

Cloud computing has a wide range of user to access its features, services and resources over the internet. The centralize cloud management was develop to satisfy efficiency. In cloud services resource control, resource management and workload on the server are some challenging issues. Load balancing helps to minimize the workload on the server by evenly distributing cloud resources on each server in cloud. Data de-duplication involves the process of detecting repetitive data. Elimination of duplicated data leads to reduction in data and network bandwidth. The proposed method focuses on the storage efficiency, provide security and load balancing. To carry out this process perform de-duplication process, chunking of data, hash function, load balancing.

INTRODUCTION:

Now a day the use of CC is increasing day by day as accessing multiple services over cloud is easy. Each user stores or accesses his/her data on cloud causing more requirement for data volumes and corresponding data storage, so need to pay attention to economize the capacity of the cloud storage. Cloud storage capacity gaining more attention. As a CC is more popular that provides more storage with storage as a service to all users over the cloud. More concentration need to be given in order to provide better services to the end user and also fulfill their requirements. The use of load balancing strategy to increase the performance and to minimize the overhead from the cloud. Data de-duplication is a technology that can reduce cloud storage capacity requirement. In simple term data de-duplication is a method that stores unique contents. The de-duplication technique does not allow storing duplicated data on the cloud storage. End users does not like to wait for long period of time to do complete the task, so that a mechanism need to employed that minimizes the cloud overhead and to provide better utilization to increase the speed and the performance to complete the task within least period of time. Load balancing strategy better utilizes resources and fairly allocates resources from pool of resources leading more resource utilization to finish task associated with servers over cloud.

DE-DUPLICATION:

Data De-duplication is a technique that can minimize the storage need by removing therepeated data. It can store the unique contentof data on the server, duplicate data don't be allowed on the server. For Example, as shown in Fig 1, a storage server stores 400 instance of data, each instance has attach 1GB Data. If storage server backup, all 400 instance are store, need 400GB size on the server. On those 400 instances only 100 instances are unique. By using de-duplication, we need to store only 100 instances rather than 400 instances, so that when backup or stored data on the server we actually store unique content of data rather than duplicated data so that, actually store 100GB rather than 400GB data. By using deduplication, we can save the 300GB storage space. De-duplication identifies repeated data in the storage servers using hash algorithm.

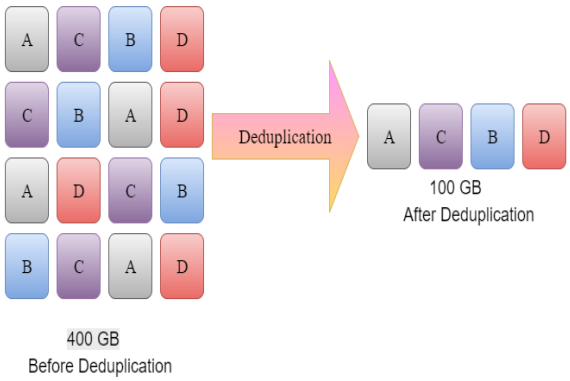

Fig. No.1.De-duplication

LOAD BALANCING:

Load Balancing is the challenging task in CC. The primary focus of load balancing is that; minimize the load from heavily loaded server so that the load on the single server can reduce. In CC many time it happens that, one server is heavily loaded while others are idle. By using load balancing we can overcome this issue. Load balancing technique can fairly distribute each tasksto the server. A CLBDM algorithm [5] is use to balance the load on the servers. In CLBDM, it can calculate the connection time between the end user and the server. If the connection time is above the set threshold, then at that time the connection between client and server is closed and then round robin algorithm take place for further connection to complete the task.

LITERATURE REVIEW:

FILE LEVEL DE-DUPLICATION:

Kai Li et al. [3] suggested file level de-duplication, In the file level de-duplication approach is performed over a single file. Compared this file with existing stored file on the cloud storage. The duplicate file can be identified on the basis of hash value, If the hash value is matched of input file with the existing files in the cloud storage then it store the reference to the existing file and discard the input file.

Disadvantage:

1) Low de-duplication ratio.
2) Low throughput.

BLOCK LEVEL DE-DUPLICATION:

Wayne Sawdon et al. [9] suggested block level de-duplication, in block level de-duplication the input data stream can be broken into block by either fix size block or either variable six blocks. Then the hash value will calculate using hash algorithm and then using this hash value to identify the duplicated data. If the hash value is not same then store this block on the cloud storage and store the hash value in the hash table.

Block level de-duplication approach divided into two types.

1) Fixed size: Divide the data into fixed size blocks.

Disadvantage:

a. Fails to find the repeated data as a small change in the block result.

2) Variable size: Divide the data into variable size blocks.

Disadvantage:

a. Requires a more time compared to fixed size.

ROUND ROBIN:

Soto mayor et al. [10], Round Robin is widely used and most popular algorithm in load balancing. In round robin algorithm, it can be proceed the request to the least number of connection server. The disadvantage of round robin is that at some point of instance one of the server is heavily loaded while other are idle.

OPPORTUNISTIC LOAD BALANCING:

Wang et al. [6] proposed OLB, It does not consider the current workload on the server. The main aim of this algorithm is too busy each server, so that it can assign the task randomly to the server. The OLB work in slower manner and the provide poor result. The current execution time of the server does not calculate.

CHUNK LEVEL DE-DUPLICATION:

Kave Eshghi et al. [4] proposed chunk level deduplication. In the chunk level de-duplication calculate the hash value of each chunks and compared this hash value with existing chunks. In this approach de-duplication ratio is high but chunks size is small, but it take more time to compare each chunk for duplicate data.

DISADVANTAGE:

1) Take more time to find duplicate data.

PROPOSED SYSTEM:

In the proposed system the main aim is to store the unique data on the cloud server with minimum workload on the server. We can done this by using the hashing algorithm. Calculate the hash value of the data that has very small chance of collision. To minimize the workload on the server we can use CLBDM algorithm [5] for rebalancing the load on the cloud server.

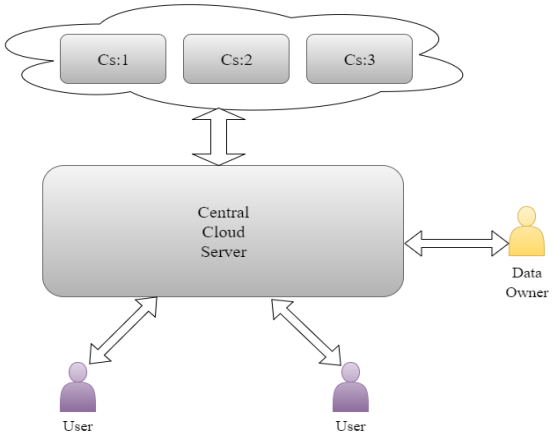

Fig. No.2. System Architecture

In the proposed system the de-duplication can be done at the client side, hence extra round trip will minimize and speed up the system. When the data owner upload the data, the data owner first calculate the hash value of the data and check that data exist on the cloud or not. If data does not exist in cloud then divide the data into fixed size of chunks and encrypt each chunks before uploading on the cloud server.

Fig2. Show this scenario data owner upload the data on the cloud. When the user want to access the data, user sent the request to the central cloud, only the authorize user can access the data.

UPLOADING ALGORITHM:

1. Select file to upload.
2. Calculate hash value of the file.
3. At client side check for de-duplication.
4. If file is exist then assign pointer to the existed file.
5. If not then divide the file into chunks.
6. Encrypt every chunk.
7. Now upload the encrypted chunks on the cloud server in distributed manner by selecting minimum workload server first.

DOWNLOADING ALGORITHM:

1. Request from user to access the file.
2. Allow the request if authorize user.
3. Send the decryption key to the user.
4. User download the file and decrypt it.

RESULT:

We perform the time required to upload file on three pc configured with intel core i5 processor with 2.66 GHz and 6 GB Ram with 100 mbps speed internet, the result may slightly vary with the configuration. The experimental result show that the proposed system can upload 80 MB file within 4-6 seconds.

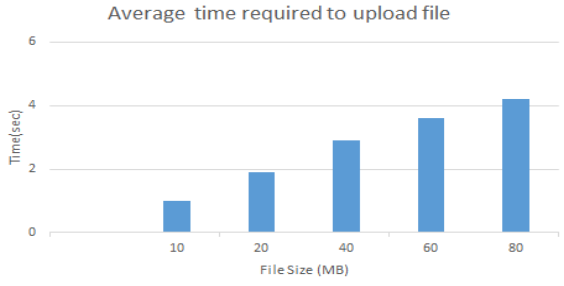

Fig. No.3. Average time required to upload file on the cloud server.

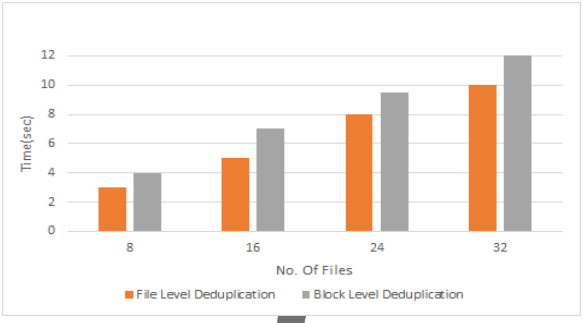

Fig.No.4.Comparison between FLD and BLD.

We also compare the FLD with BLD, The experimental result show that the FLD is slightly faster than the BLD.

CONCLUSION AND FUTURE SCOPE OF WORK:

Used to protect the data security by providing authentication to users. We also use de-duplication scheme that can store unique data to increase the storage efficiency, Less data means less support and fast data access so performance also increased. Here we proposed the upload and download algorithm. In order to manage load between the servers, central load balancing decision model is used. By using load balancing strategy, we can overcome the heavily loaded server issue.

We add a cache memory on the client side to speed the process of deduplication. We can also create a shadow copy of central server, so that it is easy to recover the system from failure. Provide a web interface so that user can modify the file on the cloud without downloading it.

REFERENCES:

i. Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou, "*A Hybrid Cloud Approach for Secure Authorized De-duplication*", in vol: pp no-99, IEEE, 2014.

ii. Jin-Yong Ha, Young-Sik Lee and Jin-Soo Kim,"*Deduplication with Block-Level Content-Aware Chunking for Solid State Drives (SSDs)*", IEEE International Conference on High Performance Computing andCommunications IEEE International Conference on Embedded and Ubiquitous Computing ,pp.2 , 2013.

iii. Zhu, Benjamin, Kai Li and R. Hugo Patterson. "*Avoiding the Disk Bottleneck in the Data Domain Deduplication File System.*",InFast, vol. 8, pp. 1-14, 2008.

iv. Lillibridge, Mark, KaveEshghi and DeepavaliBhagwat."*Improving restore speed for backup systems that use chunk-based deduplication*.",InFAST, pp. 183-198. 2013.

v. Radojevic, B. and M. Zagar, "*Analysis of issues with load balancing algorithms in hosted (cloud) environments.*" In proc.34th International Convention on MIPRO, IEEE, 2011.

vi. Wang, S-C., K-Q. Yan, W-P. Liao, and S-S. Wang, "*Towards a load balancing in a three-level CC network*," in proc. 3rd International Conference on Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp:108-113, July 2010.

vii. Lee, R. and B. Jeng, "*Load-balancing tactics in cloud*," in proc. International Conference on cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp:447-454, October 2011.

viii. Daemen, Joan; Rijmen, Vincent "*AES Proposal: Rijndael*" (PDF) National Institute of Standards and Technology. p. 1. Retrieved 21 February 2013.

ix. 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.

x. Sotomayor, B., RS. Montero, IM. Llorente, and I. Foster, "*Virtual infrastructure management in private and hybrid clouds*," in IEEE Internet Computing, Vol. 13, No. 5, pp: 1422, 2009.