International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-03, Issue-08, Nov 2017

Load Balancing & De-duplication In Cloud Computing

[1]Monali Jadhav, [2]Pooja Ranalkar, [3]Pooja Gavade, [4]Meghali Diwate, [5]Prof. Pankaj Badgujar
[1,2,3,4]UG Student, [5]Professor, Dept. of Comp .Engg. JES'ITMR, Nashik, Maharashtra, India.

Abstract- Cloud computing enables on-demand network access to a shared pool of configurable computing properties such as servers, storage also applications. These shared properties can be quickly provisioned to the consumers on the basis of paying only for whatever they use. Cloud storage discusses to the delivery of storage resources to the consumers over the Internet. Private cloud storage is restricted to a particular organization and data security risks are less compared to the public cloud storage. Hence, private cloud storage is built by exploiting the product machines within the organization also the main data is stored in it. When the use of such private cloud storage increases, there will be an increase in the storage demand. It leads to the expansion of the cloud storage with additional storage nodes. During such development, storage nodes in the cloud storage need to be balanced in terms of load. In order to maintain the load across some storage nodes, the data need to be migrated across the storage nodes. This data migration consumes additional network bandwidth. The key idea overdue this paper is to develop a dynamic load balancing algorithm created on de-duplication to balance the load across the storage nodes during the expansion of private cloud storage.

Keywords: Cloud Computing, De-duplication, Load Balancing, Secure Hash Key, Advanced Encryption Standards.

I. INTRODUCTION

Nowadays, cloud computing is very important in the Information Technology. Cloud Computing enables access to a shared pool of configurable computing resources like servers, storage and applications, etc. The storage services provided to users are though internet. Load balancing and de-duplication is being an important task for doing operations in cloud. As cloud computing has been growing and many clients all over the world are demanding more services and better results, so load balancing is necessary. Load balancing assure efficient resource utilization to customers on their demand and build up the overall performance of cloud. Every increasing volume of back up data in cloud storage may be a vital challenge.de-duplication for eliminating the duplicate data. Many algorithms have been developed for allocating client's requests to available remote nodes. The key idea behind this paper is to develop a dynamic load balancing algorithm based on de-duplication to balance the load across the storage nodes during the expansion of private cloud storage.

II. GOALS AND OBJECTIVES

1. The system is to removing a load on cloud base servers and avoiding a data Duplications using the some methodologies and algorithm.
2. This system is basically performed on Hash Code detection technique which is avoiding the multiple storage of the files on the Cloud Server.
3. For the load balancing techniques we are split the file into three chunks and stored into the three different locations and the access is only for the valid persons or an authorized person only who has login credentials with the valid user key (private key) which is given by the admin.
4.

III. MOTIVATION

Main motivation of the system is to remove a load on cloud base servers and avoiding data Duplications using the some methodologies and algorithm. This system is basically performed on Hash Code a detection technique which is used for avoiding multiple storage of the files on the Cloud Server.

For the load balancing techniques system split the file into three chunks and stored into the three different locations and the access is only for the valid person's or authorized persons only who has login credentials with the valid user key which is given by the admin.

This System has a functionality to ask information for the customer to the login and send the username, password and private key to the user with the help of the admin. Those have a login credentials as well as private key for the login who can easily perform upload, delete, and download operations.

Using the Advanced Encryption standards (AES) and Secure Hash Code (SHA) algorithm the data security and load balancing will be manage. The Hash Code is create code according to the file data and stored into database if the code

 DOI : 10.18231/2454-9150.2017.0047

is same then Duplicate file message will be arrive otherwise the code is unique then file split into three different chunk and stored it into three Different location.

If the user try to Delete or Download the file without Private Key and its login credential it gets fails. The Login credential gets match then the all of three chunks gets merged into a single file and Delete/Download Operations performed this makes the faster and more secure.

IV. EXISTING SYSTEM

Following load balancing techniques are currently prevalent in clouds –

Vector Dot- A. Singh et al. proposed a novel load balancing algorithm called Vector Dot. It handles the hierarchical complexity of the data-center and multidimensionality of source loads across servers, network switches, and storage in an agile data center that has integrated server also storage virtualization technologies.

Compare and Balance- This algorithm declares that the migration of VMs is always from high-cost physical hosts to low-cost host but assumes that each physical host has enough memory which is a weak assumption.

CLBVM- A. Bhadani et al. Suggested a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load consistently in a extend virtual machine/cloud computing environment.

LBVS- H. Liu et al. Suggested a load balancing virtual storage strategy (LBVS) that offers a large scale net data storage model and Storage as a Service model built on Cloud Storage.

Task Scheduling based on LB- Y. Fang et al. discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization.

Active Clustering- M. Randles et al. investigated a self-aggregation load balancing technique that is a self-aggregation algorithm to improve job tasks by relating similar services using local re-wiring.

V. PROPOSED SYSTEM

We are developing the system to remove a load on cloud base servers and avoiding data Duplications using the some methodologies and algorithm. This system is basically performed on Hash Code detection techniques which are used for avoiding multiple storage of the file on the Cloud Server. For the load balancing techniques system split the file into three chunks and stored into the three different locations and the access is only for the valid persons or authorized persons only who have login credentials with the valid user key which is given by the admin.

1. Load Balancing

It is a process of reallocating the total load to the single nodes of the collective system to make resource utilization effective and to recover the reply period of the job, instantaneously eliminating a situation in which some of the nodes are over loaded while some others are below loaded. A load balancing algorithm which is dynamic in nature does not consider the earlier state or performance of the system, that is, it depends on the existing performance of the system.

The main belongings to consider while developing such algorithm are : estimation of load, comparison of load, steadiness of different system, presentation of system, communication between the nodes, nature of work to be moved, choosing of nodes and many additional ones. This load considered can be in terms of CPU load, quantity of memory used, interruption or Network load.

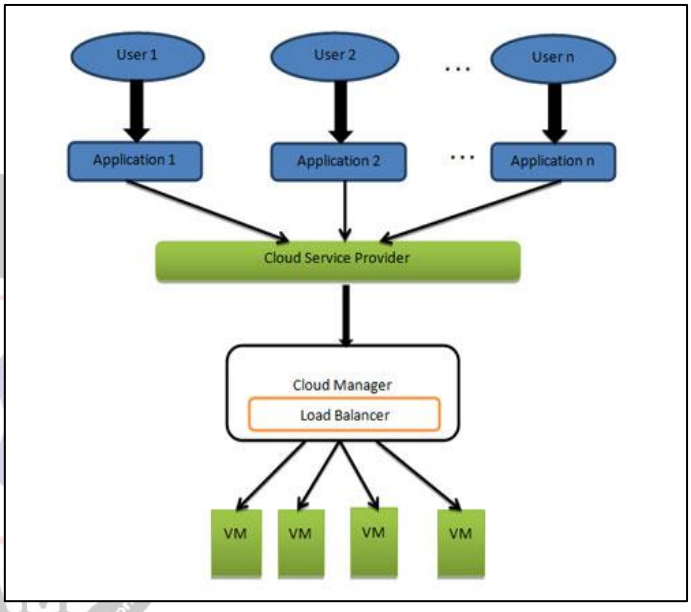

Fig.1 Load Balancing

2. De-duplication

Data de-duplication-often called intelligent compression or single-instance storage- is a procedure that reduces redundant reproductions of data and reduces storage overhead. Data de-duplication methods guarantee that only one unique instance of data is reserved on storage media, such as disk, flash or tape. Dismissed data chunks are changed with a indicator to the unique data copy. In that way, data de-duplication closely arrange in a line with incremental backup, which copies only the data that has renewed since the previous backup.

For example, a typical email system might contain 100 instances of the same 1 megabyte file attachment. If the email platform is backed up or archived, all 100 instances are saved, requiring 100 MB of storage space. With data de-duplication, only one illustration of the attachment is kept; each following illustration is referenced back to the one saved copy. In this example, a 100 MB storage request drops to 1 MB.

 DOI : 10.18231/2454-9150.2017.0047

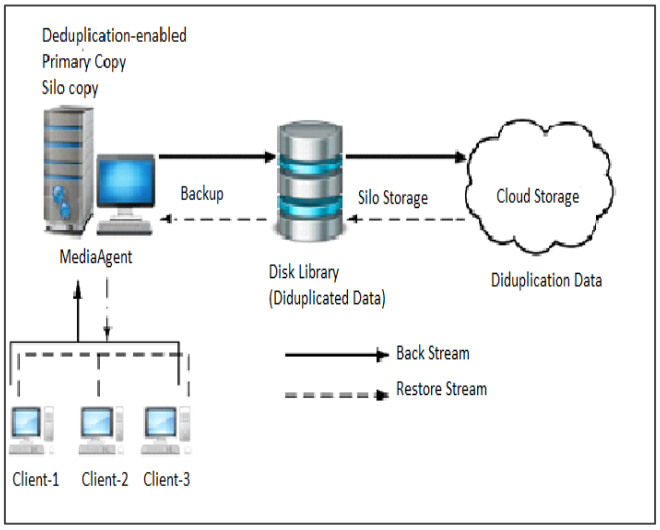

Fig.2 De-duplication

VI. BLOCK DIAGRAM

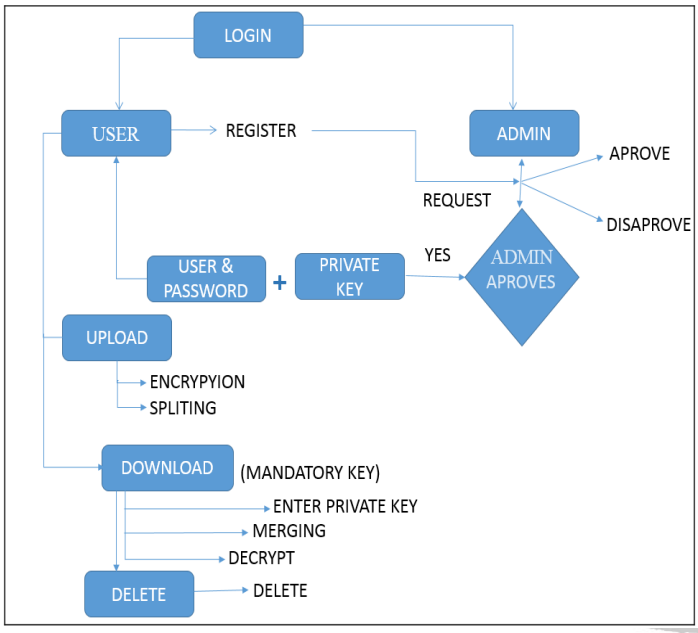

Fig. 3 Block Diagram

Block Diagram Description:

1. User can apply for log in credentials.
2. Admin:

a) Admin will approve or disapprove the User request for log in credentials.

b) If admin approves the request then following details will send to the user
User name
Password
Secret Pin to download and delete file.

c) If admin disapproves the request then user record will be deleted and disapprove mail will send to the user.

3. User can upload the file.
4. User can delete his own file by using secret pin.
5. On Upload following operation will happen to achieve DE-duplication:
Hash Code generation on the basis of content of the file.
If same hash code exists in database table then pointer will set to theexisting file.
If hash code is unique then file will be split into three equal chunks.
Then every chunk will be uploaded into three different locations.

6. On Delete following operation will happen:
User provides secret pin on rise of delete request.
If file has any pointer then only database entry will be deleted.
If there is no pointer to the file then its a unique file and databaseentry and file chunks will be deleted.

7. On Download following operation will happen:
User provides secret pin on rise of download request.
If secret pin matched then only file chunks will be merged.
Then system will decrypt the file and will be downloaded to the client side.

VII. ALGORITHMS

A. *Advanced Encryption Standard (AES)*

AES algorithm is used to encrypt the data. AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192-and 256-bits, respectively. (Rijndael was designed to handle additional block sizes and key lengths, but the functionality was not adopted in AES.) Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and the receiver must know and use the same secret key.

All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192- or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12rounds for192-bit keys, and 14 rounds for 256-bit keys { around consists of several processing steps that include substitution, transposition and mixing of the input plain text and transform it into the final output of cipher text.

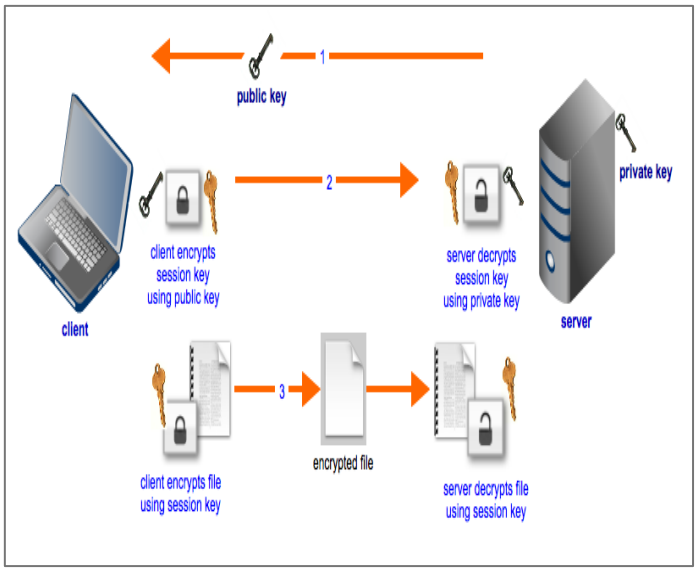

Fig.4 Working of AES

 DOI : 10.18231/2454-9150.2017.0047

AES algorithm has few steps are as follows:

Step 1: The original data is ciphered using Rijindaels key schedule.

Step 2: Round keys are derived from the cipher key known as Key Expansion. AES requires a separate 128-bit round key block for each round plus one more.

Step 3: Initial Round
Step 3.1: AddRoundKey- each byte of the state is combined with a block of the round key using bitwise XOR.

Step 4: Each round consist of several processing steps.
Steps 4.1: SubBytes- a non-linear substitution step where each byte is replaced with another according to a lookup table.
Step 4.2: ShiftRows- a transposition step where each row of the state is shifted cyclically a certain number of steps.
Step 4.3: MixColumns- a mixing operation which operates on the columns of the state, combining the four bytes in each column.
Step 4.4: AddRoundKey

Step 5: Final round will have only SubBytes, ShiftRows and AddRoundKey. MixColumns performed in this round.

B. Secure Hash Algorithm (SHA)

SHA algorithm generates a hash code on the basis of file content. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their collision resistance: nobody should be able to and two different input values that result in the same hash output.

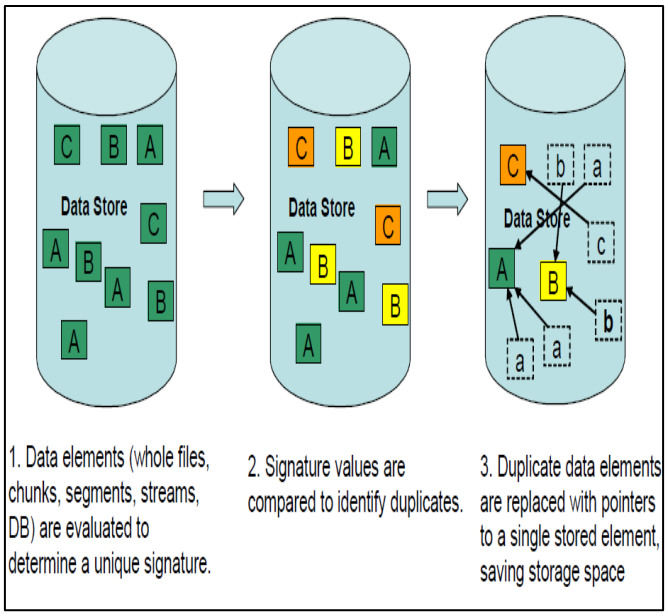

Fig.5 Basic Concept of De-duplication using SHA

SHA algorithm has few steps are as follows:

Step 1: Append Padding Bits- Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits fewer than an even multiple of 512.

Step 2: Append Length- 64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

Step 3: Prepare Processing Functions-
SHA1 requires 80 processing functions defined as:
f(t;B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
f(t;B,C,D) = B XOR C XOR D (20 <= t <= 39)
f(t;B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <=59)
f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)

Step 4: Prepare Processing Constants-
SHA1 requires 80 processing constant words defined as:
K(t) = 0x5A827999 (0 <= t <= 19)
K(t) = 0x6ED9EBA1 (20 <= t <= 39)
K(t) = 0x8F1BBCDC (40 <= t <= 59)
K(t) = 0xCA62C1D6 (60 <= t <= 79)

Step 5: Initialize Buffers-
SHA1 requires 160 bits or 5 buffers of words (32 bits):
H0 = 0x67452301
H1 = 0xEFCDAB89
H2 = 0x98BADCFE
H3 = 0x10325476
H4 = 0xC3D2E1F0

Step 6: Processing Message in 512-bit blocks (L blocks in total message)-
This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.

Input and predefined functions:
M[1, 2, ..., L]: Blocks of the padded and appended message f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): 80 Processing Functions K(0), K(1), ..., K(79): 80 Processing Constant Words
H0, H1, H2, H3, H4, H5: 5 Word buffers with initial values

Step 7: Pseudo Code-
For loop on k = 1 to L
(W(0),W(1),...,W(15)) = M[k] */* Divide M[k] into 16 words */*
For t = 16 to 79 do:

 DOI : 10.18231/2454-9150.2017.0047

W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1

A = H0, B = H1, C = H2, D = H3, E = H4

For t = 0 to 79 do:

TEMP = A<<<5 + f(t;B,C,D) + E + W(t) + K(t) E = D, D = C,

C = B<<<30, B = A, A = TEMP

End of for loop

H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E

End of for loop

Output:

H0, H1, H2, H3, H4, H5: Word buffers with final message digest.

VIII. CONCLUSION

These systems propose the architecture of de-duplication system for cloud storage environment and give the process of avoiding de-duplication in each stage. In Client, system employ the file-level and chunk-level de-duplication to avoid duplication. The algorithm also supports mutual inclusion and exclusion. Load sharing algorithm which is having policy to partitions the system into various domains and also having concept of cache manager and information dissemination for the various cloudlets.

REFERENCES

[1] J. Wu, L. Ping, X. Ge, Y. Wang, and J. Fu, Cloud storage as the infrastructure of cloud computing, in Proc. 2010 Int. Conf. Intell. Comput. Cognitive Inform. (ICICCI), KualaLumpur,2010,pp.380- 383.

[2] J. Gantz and D. Reinsel, The digital universe decade-Are you ready, IDC WhitePaper,http://www.emc.com/ collateral/analyst-reports/idc-digitaluniverse-are-you-ready.pdf,2010.

[3] P. Xie,Survey on de-duplication techniques for storage systems,Comput. Sci., vol. 41, no. 1, pp. 22-30, Jan. 2014.

[4] R. Hu, Y. Li, and Y. Zhang, Adaptive Resource Management in PaaS Platform Using Feedback Control LRU Algorithm, International Conference on Cloud and Service Computing, 2011.

[5] C. S. Pawar, and R. B. Wagh, Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue, 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), 2013.

[6] Buyya.R.et al.,Market-Oriented Cloud Computing: Vision, Hype and Reality for Delivering it Services as Computing Utilities,.c 2008.

[7] Ghalem Belalem, Said Limam, Fault Tolerant Architecture to Cloud Computing using Adaptive Checkpoint,International Journal of Cloud Applications and Computing, 1(4), pp 60-69, 2011.

[8] Prof. Vinayak D. Shinde, "Fear of Data Privacy and Security in Cloud Computing Technology", International Journal for Research in Engineering Application & Management (IJREAM), Volume 01, Issue 09, 2015, Pages 05-13.

[8] Malte Schwarzkopf, Derek G. Murray, Steven Hand, The Seven Deadly Sins of Cloud Computing, Research University of Cambridge Computer Laboratory.

[9] Sandeep sharma, Sarabjit singh, and Meenakshi Sharma, Performance Analysis of Load Balancing Algorithms,World Academy of Science, Engineering and Technology, 2008.

[10] R.Buyya et al, Cloud Computing Principles and Paradigms,Published by John Wiley Sons, Inc., Hoboken, New Jersey.

 DOI : 10.18231/2454-9150.2017.0047