An Approach for Load balancing in Decentralized Cloud Computing

[1]Prof. Chandu Dajiba Vaidya, [2]Prof. Ravi Asati, [3]Prof. Ashish Golghate

Assistant Professor Department of CSE, RGCER Nagpur Maharashtra India

Abstract- In this era of developing technologies, one of the most promising technology is cloud computing that has been functioning since years and used by individuals to large enterprises to provide different kind of services to the world. A cloud can provide computing power, storage system to virtual machines for different services as required. Since it is a large scale technology, it needs proper way of handling data that can be sensitive for an individual or a large firm also by considering security issues. Hence, we propose an approach to distribute data between different nodes by enhancing security using various mechanisms.

Keywords: Cloud computing, Cloud storage, cloud services encryption/Decryption, load balancing.

I. INTRODUCTION

Today the present world mostly depends on exchange of information i.e. Communication of data from one person to other person. Cloud computing plays an important role in this system. It is the on- demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform.

Cloud computing platform is a set of scalable large- scale data server clusters. The cloud storage is widely applied service which can provide users with stable, massive data storage space. Research shows, the architecture of current cloud computing system is central structured one; all the data nodes must be indexed by a master server which will bottleneck the system.

The proposed work is of decentralized cloud computing architecture, wherein Statistical approach for [11,12] distribution is to be found. The system based on the new architecture has better scalability and fault tolerance and proposed system designed a cloud based environment where request and response is taking place between client and middleware directly.

II. BACKGROUND

In earlier times, cloud computing used to follow a traditional client/server architecture, but due to this tradition, many attackers found a flaw in this system, researchers kept enhancing different methods to operate cloud and provide better services with security. Soon, many new proposals were presented to develop this technology to another level with an approach for building a perfect cloud architecture. A cloud based on distributed systems was introduced to improve the utilization of resources and to serve a better compute power. In terms of data storage, the basic distributed systems were prone to different data leakage issues leading to loss of information and secrecy of data. Different algorithms were proposed to tackle this kind of scenarios like- Encryption/Decryption algorithms like RSA, SHA, SSL, AES, etc. But, there was more than Encryption/Decryption that motivated researchers to develop a better system.

III. LITERATURE REVIEW

Our main motivation for this survey paper started from enhancing the data security in a simple cloud mechanism, for this approach, the concept of data leakage detection [1] was proposed. In this approach, the methodology of fake objects and watermarking were developed in order to prevent stealing of data by an unauthorized user. If an unauthorized user tries to access the data, a fake object will be passed on to the unauthorized user so that the real data won't be exposed. The authorization is managed by a unique authorization key which is encrypted under secure hash algorithm (SHA). It was meant to ensure the user gets the file without any third party application intercepting the data.

Another approach is similar to [1], the file which would be uploaded and downloaded through a cloud environment will be handled by a trusted party to ensure the data integrity [2] and exposing risks of cloud services on behalf of the cloud client to verify data integrity. This approach used RSA encryption technique to calculate hash value with distributed verification of erasure coded data. But using RSA lead to high resource utilization and hence makes the execution longer and very inefficient to implement. This is dealt with the methodology proposed in [3].

When it comes to security in cloud, high resource utilization occurs and large computing power is required to maintain functionality. This excessive resource utilization leads to lesser throughput which affects the QoS (Quality of Service) of cloud computing. To deal with this, the concept of load balancing [3] can be used to minimize the resource utilization and to prevent system overload. In addition to this approach DES (Data encryption standard) cipher text policy algorithm is used to encrypt the data. Hence it serves in better QoS and manage the organization of data. We can see a more advanced technique in [4].

Steganography [4] is a technology that is used for the communication of messages secretly. These secret messages are deployed through a trusted carrier [2]. Different kinds of carriers can be images, audio, video, and so on. The size limitation of sending files can be overcome by taking video files as carriers in file transfer. According to this paper, internet services such as Skype, BitTorrent, GoogleSuggest, and WLANs are targets of information hiding. Plotters use the carriers and also they consists of protocols that are responsible for handling the route of that carrier in the network. The research paper [4] proposes to use steganography to encrypt the files and integrate it into a video file format when uploaded to a cloud storage and decrypt it back to original form when the user needs the file back the users storage drive. Various algorithms are to be implemented to achieve this method of cryptography and hiding the data in video file format encryption.

Every time when a file is downloaded or uploaded, it requires a key to encrypt or decrypt a file by solving a hash function based on the key provided. [5] This paper provides a deep knowledge, and puts a mechanism to ensure the data availability when there occurs problem of data integrity and incomplete data. The research in this paper focuses on the confidentiality of data, the loss of data and data recovery. This paper proposes a data secure storage methodology based on Tornado codes (DSBT) by combining the technique of symmetric encryption and erasure codes.

Adding to the approach of [2] there is a data integrity auditing mechanism by using a homomorphic token and distributed erasure-coded data proposed by [6]. This approach deals with file distribution, challenge token precomputation, correctness verification and localization and file retrieval and recovery. This approach consumes less computing power to ensure the integrity and guarantee correctness with a fast error recovery mechanism. [6] Serves and extension to what we understood from [3] with extra security and data integrity measures. We have studied various encryption algorithms, but of them all, we have to choose the most feasible that is, the algorithm which is secure and uses the least computing power to maintain efficiency in the process. Therefore the approach by [7] helps in deciding the right algorithm that is the AES (Advanced Encryption Standard). This paper tells us about how AES can be used to secure the data present on databases, cloud, etc. and other methodologies used in AES encryption like symmetric and asymmetric keys, hash functions, etc. this approach focuses on various transformation techniques like- byte substitution, permutation, mixing, round keys and decryption.

If a cloud system consisting of a common server crashes, its services may come down for some period of time and lead to high losses in data and businesses, the [8] proposes an approach to use a decentralized mechanism to overcome the failures occurring in a system crash. Indexing through a master server becomes a bottleneck in operations. The methodology used in this approach is establishment of a peer to peer (P2P) connection between user and chunk server to create a high throughput tunnel through which files can be uploaded and downloaded. The user makes a request and request goes to the gateway. The gateway constructs a search request and sends the request to the chunk server P2P network. The P2P search request locates the nearest chunk server based on the memory usage and the request is processed by that chunk server. The client will upload, download and deploy the information from

the nearest server (here nearest server means which server is containing the greater value of memory usage, that server will select as nearest server).

While using a decentralized approach in cloud, it is necessary to use a favorable load distribution algorithm to manage/organize the files present on cloud. So the [9] is a survey paper on various techniques through which load distribution can be achieved in distributed cloud storage and has an analysis of these various algorithms based on- throughput, overhead, fault tolerance, response time, resource utilization, scalability, performance. The load balancing techniques are based on metrics like CPU utilization, memory utilization, storage utilization to serve best cloud services.

In [9] we came across some load balancing algorithms, in which we decided to choose an appropriate algorithm which is called as Ant Colony Optimization Algorithm [10]. According to analysis provided in [9] we concluded that this algorithm is most feasible. Individual ants are behaviorally much unsophisticated insects. They have a very limited memory and exhibit individual behavior that appears to have a large random component. Acting as a collective however, ants manage to perform a variety of complicated tasks with great reliability and consistency. The complex social behaviors of ants have been much studied by science, and computer scientists are now finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems. The attempt to develop algorithms inspired by one aspect of ant behavior, the ability to find what computer scientists would call shortest paths, has become the field of ant colony optimization (ACO), the most successful and widely recognized algorithmic technique based on ant behavior.

IV. PROBLEM STATEMENT

In Centralized Cloud Computing, usually the problem of data leakage is proven and along with that the security of data is also a major concern whereas in Decentralized Cloud Computing these problems can be overcome. In Decentralized Cloud Computing, load distribution will be done, to achieve high performance and load management of the servers with the help of statistical comparison of different resource parameters.

V. PROPOSED WORK

Propose work has comprise of some mechanism are as follows.

A. Modules

1) Authentication

- User details will be retrieved via the front end of the web application.
- Details will be checked from the present database.
- Once the details are matched, the user will be authorized to access the data.

2) Uploading

- The user can choose files from his local storage to upload it on the cloud.
- The chosen files will be uploaded via the simple upload script (*Figure 1).*

3) Encryption and Splitting

- Once the file is in uploading phase, it would be encrypted via SHA encryption algorithm.
- After the file is successfully encrypted, the file will be split into chunks so that it can be deployed.
- Once the file is in uploading phase, it would be encrypted via SHA encryption algorithm.

- After the file is successfully encrypted, the file will be split into chunks so that it can be deployed.

4) Decider API

- The Decider API broadcasts request to resource nodes and gains statistics from various nodes.
- The best node is chosen and chunks of the files are deployed to that particular node.
- The other mirror chunks of the file are uploaded to other competitive nodes.
- This increases machine utilization and probability of retrieving data chunks.
- It provides API endpoints for seamless and secured data transfer.

5) Download/Retrieve

- Each file a user upload has a unique identifier termed as file_id.
- When the user wants to download the previously uploaded file, the client request the decider API to retrieve chunks of the file from the particular resource node using the file_id.
- When the chunks are received, they are decrypted and merged at the user side and hence the complete file is downloaded.

B. Workflow of Uploading a file

1) Find resource node

- When the file is being uploaded, it is necessary to know which nodes will be getting which chunks of file in order to backtrack the location.

2) Broadcast to all resource nodes

- In order to find present nodes, the request packets will be broadcasted to all the resource nodes.

3) Response to broadcast with statistics

- When the request packet is received, the resource node respond with the statistics of itself, these statistics include- memory usage, CPU usage, disk usage and latency.

4) Return the best resource node

- When the decider API receives the statistics from all the resource nodes, it calculates and analyzes the best suitable node for uploading a file chunk

5) Direct data transfer between selected nodes.

- When the best node is selected, a peer to peer connection is established between the user node and the resource node to transfer the data.

6) Mirror data transfer between competitive nodes

- If only the best node (prime node) is selected for data transfer, the utilization of other resource nodes decreases, in order to overcome that, the replica of chunks are created and uploaded to other resource nodes so, when the prime node is not in operation, the file will still be downloaded via the mirror node.

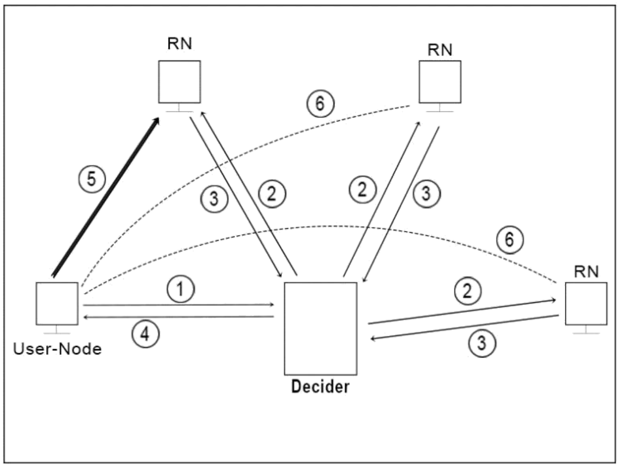

Fig.1: Flow Diagram for Upload

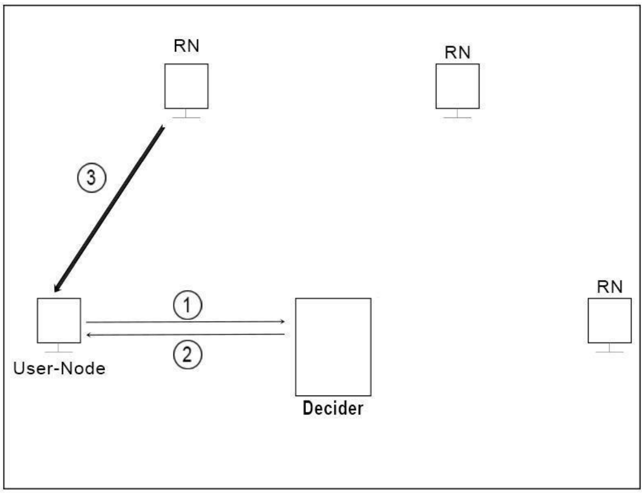

Fig.2: Flow Diagram for Download

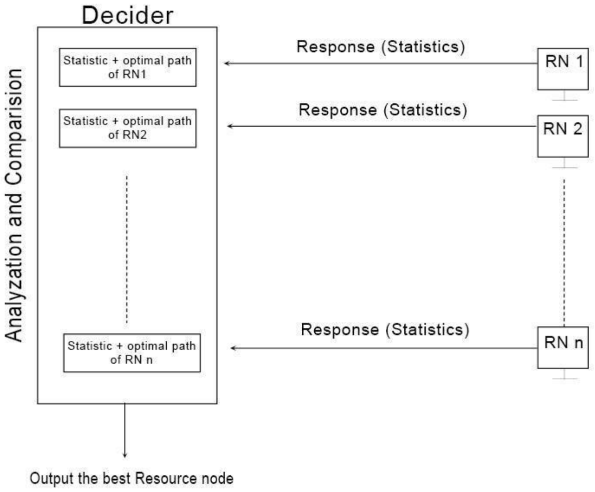
Decider
Analyzation and Comparision
Statistic + optimal path of RN1
Statistic + optimal path of RN2
Statistic + optimal path of RN n
Response (Statistics)
Response (Statistics)
Response (Statistics)
RN 1
RN 2
RN n
Output the best Resource node

Fig.3: Decider API

VI. CONCLUSION

To develop a decentralized cloud platform through which the user may upload download files on the cloud without worrying about his data or to be lost or leaked to any other third party sources throughout the network. Achieve load distribution among various servers by statistically comparing the different resource parameters and finding an appropriate load distribution method, which will give best results. Also sender side and receiver side initiative will be preferable.

VII. REFERENCES

[1]. “Data Leakage Detection and Security in Cloud Computing”, 2016, Chandu Vaidya,

[2]. “Data Security in Cloud Computing”, 2015, Chandu Vaidya, Prashant Khobragade.

[3]. “Security in Cloud Computing”, 2015, Tejashri Khandve, Megha Talekar, Sheetal Dhiwar, Madhuri Patil.

[4]. A Novel Approach for Hiding Data in Videos Using Network Steganography Methods”, 2015, Amritha Sekhar, Manoj Kumar G., M. Abdul Rahiman.

[5]. ”Research on data security technology based on cloud storage”, 2017, Rongzhi Wang.

[6]. ”Towards secure and dependable services in cloud computing.” 2015, Cong Wang, Qian Wang.

[7]. ”Data storage security in computing using AES”, 2017, Tamilselvi.S.

[8]. ”A Secure Decentralized Cloud Computing Environment over Peer to Peer”, 2013, Tanupriya Choudhury, Vasudha Vashisht, Himanshu Srivastava.

[9]. ”A survey of various load balancing algorithms in cloud computing”, 2014, Dharmesh Kashyap, Jaydeep Viradiya.

[10].”Ant Colony Optimization: A solution of Load Balancing in Cloud”, 2012, Ratan Mishra and Anant Jaiswal.

[11].Chandu Vaidya, Dr. Manoj Chandak “Efficient Parallel Process Migration Algorithm Using Statistical Approach” Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on, IEEE

[12].Chandu Vaidya, International Conference on Advanced Material Technologies (ICAMT)- 2016”An Approach for Processor Utilization in Master Slave Environment”