ISSN No. 0976-5697

Volume 5, No. 5, May-June 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

AES Encryption based Load Rebalancing Algorithm

Nithya Kuriakose
Department of Computer Science and Engineering
Nehru College of Engineering and Research Center,
Pampady, Thrissur, Kerala.

Shinu Acca Mani
Department of Computer Science and Engineering
Nehru College of Engineering and Research Center,
Pampady, Thrissur, Kerala.

Abstract: Cloud Computing is an emerging technology, it is based on demand service in which shared resources, information, software and other devices are provided according to the clients to the requirements at specific time with the availability of internet. Load balancing is one of the challenging issue in cloud computing. The load rebalancing includes passing on load to different sub servers or peers. Thus none of the peers get lightly or heavily loaded. File can be uploading to the server, can download and can distribute to another servers if the load is high with the help of encryption. The remaining space can be viewed.

Keywords: *Cloud* computing, Load rebalancing, Distributed File system, Encryption

I. INTRODUCTION

Cloud Computing is an emerging technology, it is based on demand service in which shared resources, information, software and other devices are provided according to the clients to the requirements at specific time with the availability of internet. Load balancing is one of the challenging issue in cloud computing. An efficient load balancing makes cloud computing more efficient and improves user satisfaction. The goal of load rebalancing includes performing replica management, to improve performance, to maintain the system stability and to increase the flexibility. When the number of storage nodes, the number of files and the number of access to the files increases linearly the central nodes become a performance bottleneck, as they are unable to accommodate a large number of files accesses due to the clients and MapReduce applications. Thus depending on the central nodes to tackle the load imbalance problem exacerbates their heavy loads. Even with the latest development in the distributed file systems, the central nodes may still be overloaded. The main objective is to allocate the chunks of file uniformly among the nodes [7] [8] [9].

The organization of the paper is as follows. After the introduction given in Chapter I, Chapter II gives a brief description about Load rebalancing with existing methods. Chapters III explain AES encryption based proactive LRA. In Chapter IV and Chapter V gives Implementation with experimental setup and Conclusion is given respectively.

II. LOAD REBALANCING

Load rebalancing is a new approach and it is the process of assigning the complete load to the individual nodes of the system in the direction of making resource utilization effective by providing a high throughput and less response time in other words it is the process of shifting the load among the nodes for providing high performance. The object of load rebalancing is to increase client satisfaction and maximize resource utilization and substantially increase the performance of the cloud system. The purpose of load rebalancing is to make every node in the file system perform the same amount of work which helps in minimizing the response time and increasing the throughput. Distributed hash tables are key building block for variety of distributed applications. it uses the hashing approach. One of the solutions is the use of virtual peers that is for each peers, assigning number of virtual peers. in this case large size request may not be processed because of the tightly bounded expected value. Substitute solution is that of power of two choice paradigms. In this paradigm use standard hashing scenarios using bins to reduce or balance the load. Less shared routing information stored at each peer [2]. The use of range partitioning can make partitioning a dynamic relation transversely a large number of nodes. Range portioning is frequently popular in large scale parallel in addition to peer-to-peer databases. Load balancing is necessary in such scenarios to eliminate skew. The data movement cost per tuple insert or delete is constant, and was shown to be close to 1 in experiments. Advantages are decentralized system, automatically performs all operations, avoid data skew. One of the disadvantages is that it take too much of time to complete the task [3]. The design and evaluation of pastry which includes application level routing and object location in a potentially very large overlay network of nodes connected via the internet. In application level routing, different applications will have different requirements according to that routing is performed. For example video conferencing requires high requirements, if any one use this path the requirement will decreases and hence leads to complete no sharing of path. In the case of low requirement application such as email and text messages gives a busy path. According to the requirement application the routing is performed. Advantages are Decentralized System, Automatically performs all operations and one of the disadvantages is that Every time lookup operation is needed [4]. Efficient load balancing protocols for distributed data storage in peer-to-peer systems are simple, and easy to implement, so an obvious next research step should be a practical evaluation of these schemes. In addition, several concrete open problems follow

from our work. First, it might be possible to further improve the consistent hashing scheme. It uses the hashing approach. One of the solutions is the use of virtual peers. Second, our range search data structure does not easily generalize to more than one order. It provides efficient load balancing but hard to achieve [5]. MapReduce is the programming model and associated implementation for processing and generating large data sets. Users specify a map function that processes a key-value pair to generate a set of intermediate key-value pairs and reduce function that merges all intermediate value associated with the same intermediate key [6]. Advantages are highly scalable, can compute large data set, but it is expensive, more time to compute the reducing functions. The different qualitative metrics or parameters that are considered important for load balancing in cloud computing are throughput, associated overhead, fault tolerant, migration time, response time, resource utilization, scalability, and performance. The major concerns of cloud computing that is load balancing. The goal of load balancing is to increase client satisfaction and maximize resource utilization and substantially increase the performance of the cloud system [10]. Because of the load imbalance problem and security issues in cloud environment a new encryption based proactive load rebalancing algorithm is proposed.

III. AES ENCRYPTION BASED PROACTIVE LRA

Encryption based proactive load rebalancing algorithm (LRA) includes the following functions. They are client login, file uploading, file splitting, file distribution, file downloading and user logs. It automatically activate its performance hence it is proactive.

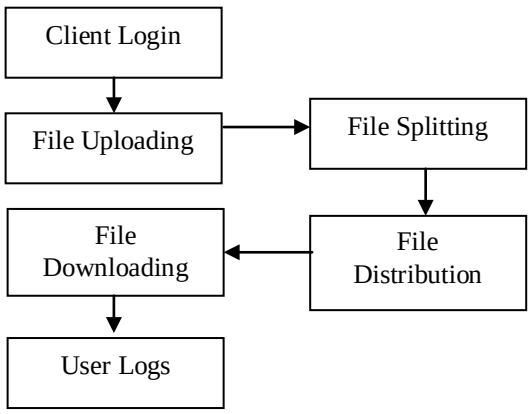

Fig 1: Functions in Load Rebalancing

A. Client Login: A data which is stored in the cloud server can be accessed or retrieved by the client if he/she registers their detail which is stored in the database. After storing the data or uploading the data, it will be in the encrypted form. This encrypted data can be decrypted with a key value only by an authorized user.

B. File Uploading: The data can be uploaded in the client server, while uploading a file, file will be partitioned in to number of chunks allocated in distinct node so that can perform MapReduce task over all nodes. That is the number of file chunks the node contain is directly proportional to the load of a node. Because of its automatic nature file in the cloud can be automatically crated appended and deleted and in the file system, nodes may be upgraded, added and replaced. This leads to load imbalance among the nodes.

C. File Splitting: It is used to split the uploading files and temporarily save it before uploading to the peers**.** After uploading the file in the cloud server, it will calculate the memory capacity of each server which can be calculated by total file size divide by the number of sub servers a main server possesses. Then the data will be giving to the distributed hash table with an index value. The distributed hash table offers lookup functionality and also guarantees that, move to the next successor when a node leaves and immediately allocate chunk when a node joins.

D. File Distribution: After file uploading and file splitting, it is used for the file distribution to the peers. It will send the set of packets to the corresponding peers**.** The data will be distributed uniformly over the nodes. In this time a copy of the request is saved in the sub server this is to improve the file availability. It is termed as managing replica. Based on the client request server will provide the required application and keep a backup copy then finally given back to the client.

E. File Downloading: If the user is an authorized user, then the user can download the file. Downloading data will be in different sub servers are sorted and merged. While downloading the following details will be displayed that is from where the data is downloading, the name of the file, server name, total number of packets, how many packets are received and also the distribution type. The distribution type can be parallel, chunk and periodic. Here in this proposal, parallel distribution type is used.

F. Encryption: To achieve data security data can be translated in to secret code. Security is a main issue in the cloud environment that is data in the cloud server may not be secure. With the help of cryptography concepts of encryption and decryption process can provide confidentiality. A secret key is used for reading an encrypted file and the same key will used to decrypt the file. Unencrypted data is called plain text and encrypted data is called cipher text. Advanced Encryption Standard (AES) Encryption is used. Authorized users can have a login id and password, they can access the data. Other users can only get the cipher text. AES is comes under symmetric cryptography, uses the same key to encrypt and decrypt its data. AES has fixed block size of 128 bits and a key size of 128,192 or 256 bits.

G. User Logs: The activity of each user can be monitored and also prevent unauthorized access to the system and the data. User logs gives tremendous insights in to user name, date of login and file viewed.

The Load Rebalancing Algorithm will perform the following process. First of all it will initialize server and its sub servers, establish connection between sub server and servers using the IP or port number, upload file to server that should be shared, split the file into multiple chunks,

calculate the each sub server memory by divide the total chunks value by total number of sub servers, upload each chunk into sub servers based on its memory capacity, if capacity is less then transfer the excess chunks into next sub servers and if not each chunk will be appended with an index value, when the client request for a file, that will be received from different sub servers based on the index value, then client collects all the chunks then the file will be decrypted, that will be viewed by client.

IV. IMPLEMENTATION AND ANALYSIS

The system consists of one main server, four sub servers, centralized system and clients. Clients can upload and download the files from the main server through the sub server. The system is a centralized so that all operations are performed by the central system. After uploading data in the server, it will be in encrypted form then split to n chunks. Server will calculate the memory capacity of each sub server and given to it uniformly. If chunks greater than sub server capacity, then transfer chunks to next node with an index value and if not transfer chunks with an index value. Finally an authorized user, who having the key can download the data. That is can decrypt the data. Other users only get cipher text. The load rebalancing algorithm will perform in windows os and the languages used are java, swing. The distribution type can be parallel, chunk and periodic. Here in this proposal, parallel distribution type is used. So that MapReduce task can be performed parallel over the nodes . Encryption based proactive LRA provides the best resource utilization, reduce the traffic in the network, extends the overall system performance, avoid the data loss by keeping the backup copy, provide high security with the help of cryptographic concepts and analyze the user logs details.

AES has three approved key length: 128 bits, 192 bits, and 256 bits. To try to explain the process in simple terms, an algorithm starts with a random number, in which the key and data encrypted with it are scrambled though four rounds of mathematical processes. The key that is used to encrypt the number must also be used to decrypt it. The four rounds are called SubBytes, ShiftRows, MixColumns, and AddRoundKey. During SubBytes, a lookup table is used to determine what each byte is replaced with. The ShiftRows step has a certain number of rows where each row of the state is shifted cyclically by a particular offset, while leaving the first row unchanged. Each byte of the second row is shifted to the left, by an offset of one, each byte in the third row by an offset of two, and the fourth row by an offset of three. This shifting is applied to all three key lengths, though there is a variance for the 256-bit block where the first row is unchanged, the second row offset by one, the third by three, and the fourth by four. The MixColumns step is a mixing operation using an invertible linear transformation in order to combine the four bytes in each column. The four bytes are taken as input and generated as output. In the fourth round, the AddRoundKey derives round keys from Rijndael's key schedule, and adds the round key to each byte of the state. Each round key gets added by combining each byte of the state with the corresponding byte from the round key. Lastly, these steps are repeated again for a fifth round, but do not include the MixColumns step. These algorithms essentially take basic data and change it into a code known as ciphertext.

Memory analysis helps to determine the information about overall state of the system which shows in figure 2. In our proposal four sub servers or peers are taken and the total of 100 GB is equally divided then given to the peers. So that after uploading the data we can analyze how many memory space is free on each peers. Objective of the current system is to determine if the load rebalancing can help to reduce performance problems such as low resource utilization, network traffic and the overall system performance. The chart below shows the average time to take for uploading and downloading in each peer. The uploading and downloading time in milliseconds respectively. As shown, most of the client activities take same amount of time thus each peers have uniform distribution of files. The results clearly shows that the proposed cryptogrophic concept solves the problems assosiated with load rebalancing.

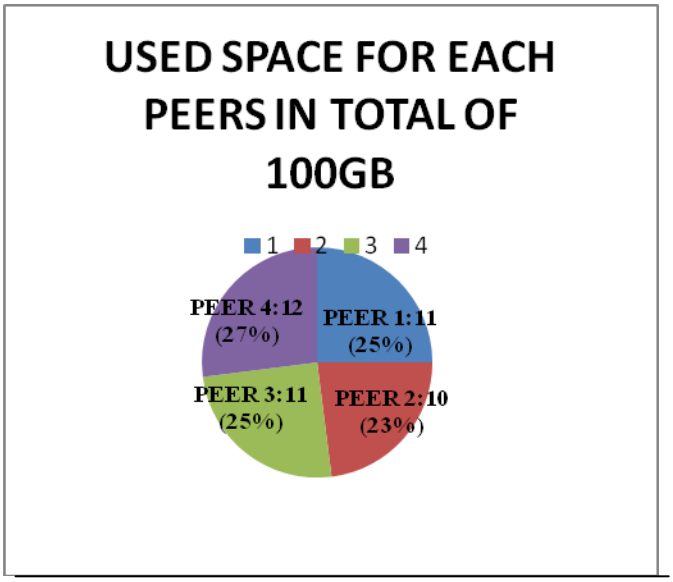

Fig.2. Used space for each peer

V. CONCLUSION

The load rebalancing is used to improve performance by distributing user request among a number of servers by reducing the lord on a single server at any time. Additionally the algorithm has the ability to monitor and compensate for system failure by keeping number of copies of data, minimizing response time, increasing throughput and if a single or multiple server breakdown it eliminate the potential downtime. AES encryption is used to achieve data security. So that only authorized users who having the key can only perform the decryption. If not other users only get the cipher text. It is fully based on the centralized system and also it is a proactive system. Thus all the operations are done automatically and insert, delete and replace are done through the central system. Here the server will calculate memory capacity then assigns the load uniformly among the nodes. Hence the load balancing problem can be avoided. In future increase efficiency and effectiveness of design are further validated by analytical models and a real

implementation with a small scale cluster environment. Highly desirable to improve the network efficiency by reducing each user's download time.

VI. REFERENCES

[1] Hung-Chang Hsiao, Hsueh-Yi Chung,Haiying Shen, and Yu-Chang Chao,"Load Rebalancing for Distributed File Systems in Clouds," *IEEE Transactions On Parallel And Distributed Systems*, Vol. 24, No. 5, May 2013.

[2] J.W. Byers, J. Considine, and M. Mitzenmacher, "Simple Load Balancing for Distributed Hash Tables," *Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '03), pp.* 80-87, Feb. 2003.

[3] P. Ganesan, M. Bawa, and H. Garcia-Molina, "Online Balancing of Range-Partitioned Data with Applications to Peer-to-Peer Systems," *Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04), pp.* 444-455, Sept. 2004.

[4] A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems," Proc. IFIP/ACM Int'l Conf. Distributed Systems Platforms Heidelberg, pp. 161-172, Nov. 2001.

[5] D. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," Proc. 16th ACM Symp. Parallel Algorithms and Architectures (SPAA '04), pp. 36-43, June 2004.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Proc. Sixth Symp. Operating System Design and Implementation (OSDI '04), pp. 137-150, Dec. 2004.

[7] T.Sakthisri and S.Pallavi, "Balancing Blocks For Distributed File System In Clouds By Using Load Rebalancing Algorithm," Proc. International Conference on Information Systems and Computing (ICISC-2013), pp. 220, Jan. 2013.

[8] Revathy R and A.Illayarajaa, "Efficient Load Re Balancing Algorithm for Distributed File Systems," Proc. International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.

[9] Yatendra Sahu and R.K. Pateriya, "Cloud Computing Overview With Load Balancing Techniques," Proc. International Journal Of Computer Applications (0975 – 8887) Volume 65– No.24, March 2013.

[10] Aarti Khetan, Vivek Bhushan and Subhash Chand Gupta, "A Novel Survey On Load Balancing In Cloud Computing," Proc. International Journal Of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 2 Issue 2, February 2013.