

Secure and Privacy-Preserving Distributed File Systems on Load Rebalancing in Cloud Computing

A Sumanth¹, Bhaludra Raveendranadh Singh², Moligi Sangeetha³

¹A Sumanth , pursuing M.Tech (CSE) from Visvesvaraya College of Engineering and Technology(V CET), M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy (D)-501510, India

²Bhaludra Raveendranadh Singh is working as Principal from Visvesvaraya College of Engineering and Technology (VCET), M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy (D)-501510, India

³Moligi Sangeetha is working as Associate Professor & HOD (CSE) from Visvesvaraya College of Engineering and Technology (VCET), M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy (D)-501510, India

Abstract—Distributed file systems in cloud computing because Google File System GFS and Hadoop Distributed file systems HDFS scheduled central servers in the direction of manage and load balancing in the metadata. Enabling technology for large-scale computation for big data Originated by Google. Open source implementation by Yahoo and Facebook. Distributed file systems - DFS are keys structure block for cloud computing application based on the shrink programming. In this file system swelling at the same time storage functions and grant computing, a file submerge into a number of parts billed in specifically nodes therefore, particulars Reduce functions can be execute in parallel to all the systems. In this cloud computing environment, crash standard and the nodes may be replaced, updated and added in the system. Files are also dynamically created, append and delete. These consequences in load imbalance in a distributed file system, person's files parts are not dispersed in the same way as much as probable between the nodes mounting distributed file systems in manufacture systems stretched are based on the central node for the part transport. This depended is it appears that inaccurate in a large scale and failure chances locations since there will be alive central load is put beneath a important level workload that is simple level by means of system size and may consequently become the management prevention and the single point of failure. A fully complete load rebalancing algorithm is exhibited to cope with the load rebalance trouble. In proposed system sending the files separated different parts of files using load rebalancing and Data Encryption Stranded - DES algorithm behind stored into clouds.

Keywords— load balancing , Distributed file systems , clouds

1 INTRODUCTION

Cloud computing are persuasive technology customers can dynamically assign their resources on the aspiration to without complicated management and deployment of possessions. Dissimilar parts of technologies are used in clouds such as Map lessen programming example, DFS - Distributed File Systems, effectuation. In a large cloud we can add there are thus ands of records with. The main aim is to assign files to these files without formation weighty load to any of the files, for that files are partition into dissimilar parts. one more intention is to lessen the network inequality and arrangement traffic since of the imbalance of loads. The scalability possessions we can add, update, delete new files so that it chains strangeness of the system. Development the aptitude of nodes we use Distributed file System DFS in Cloud

computing application. In these file systems the main movement of files is to dish up multiply and storage space functions. If we desire to store a file into the system initially we will split the file into dissimilar parts and store it in different files. We introduced latest load rebalancing algorithm, information encryption standard algorithm to keep away from these entire drawback.

Distributed file system is conventional model of file system that is used in the form of break apart for cloud computing. Cloud computing function is based on the MapReduce indoctrination used in distributed file system. MapReduce is the master and slave architecture in Hadoop. Slave act like Data node and Master act like Name node. As the augment in storage space and system, load balancing is the main issue in the large scale distributed systems. Load should be balance over multiple nodes to get better system appearance, resource utilization, response stability and time. Load balancing is divided into two type dynamic and static. In stationary load balancing algorithm, it does not consider the preceding behaviour of a node while dispense the load. But in this case of active load balancing algorithm, it make sure the preceding behaviour of node while allocate the load. In cloud, if number of cargo space nodes, number of files and charge to that file amplify then the inners node develop into blockage. The load rebalancing task is used to eradicate the load on inner node, in the load balancing algorithm, storage nodes are structured over set of connections based on the DHT - distributed hash table, each file chunk having rapid key lookup in DHTs, in that unique identifier is allocate to each file chunk. DHTs enable nodes to self identify and repair while it always offers lookup occupationally in node.

Data Encrypt Stranded - DES As talk about before there are two main categories of cryptography in use symmetric or asymmetric and furtive key cryptography or public key cryptography. Symmetric key cryptography is the older type but asymmetric cryptography. Clientele sending files divided into dissimilar parts of files using load rebalancing and DES algorithm after stored into cloud servers.

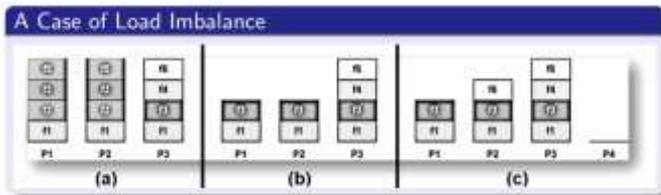


Fig. 1. An example illustrates the load rebalancing problem, where (a) an initial distribution of chunks of five files f_1, f_2, f_3, f_4 and f_5 in three nodes P_1, P_2 , and P_3 , (b) files f_2 and f_5 are deleted, (c) f_4 is appended, and node P_4 joins. The nodes in (x), (y), and (z) are in a load-imbalanced state.

Mostly, storage space nodes by involve the storage nodes imbalance in to we supporter off-load the load rebalancing task to storage nodes their loads impetuously this removals the confidence on middle nodes. Network based on distributed hash tables in the store nodes are prearranged as a judgment a chunk files can frequently suggests to rapid key search for in DHTs, given by a single identifier is allocated to each file chunk. DHTs allowed nodes to repair and self-organist while constantly at time serving search for functionality in node vitality, simplifying the system conditions and management.

As flowers in spirit, our works are three categories:

1. By implement able to DHT distributed hash table, we resent a load rebalancing algorithm for distributing file break apart minimizing and as consistently as probable the movement cost very much like probable. Completely our proposed algorithm handles in a distributed method in which nodes to their load balancing works separately without global knowledge or management about the system.
2. Load-balancing algorithms depending on DHT distributed hash table s have been extensively studied if most at present the solutions are made from bearing in mind both movement node and cost may bring in heterogeneity and significant maintenance network transfer to the distributed hash table DHTs. In a different way, our proposal not only bring usefulness of novel network preferred local in the transport of file break apart to lesser the movement cost but also develop have the capacity nodes to organization presentation the generally enhanced . In addition, launch to the DHT distributed hash table as much as probable our algorithm lowers algorithmic transparency.
3. Our suggestion is opinion by computer imitation reproduction. The imitation results propose if each node finger our load re-balancing algorithm separately without attain global knowledge, it is equal with the centralized approach in extraordinarily and Hadoop HDFS as of Handles the contest distributed algorithm in conditions of movement cost, algorithmic transparency load, and imbalance factor. As well our load-balancing algorithm shows a rapid combination rate. There logical models to improvement the efficacy and efficiency of our devise. In addition, we have executed our load-balancing algorithm in examined and HDFS its show in a groups environment.

II. CLOUD PARTITIONING TECHNIQUE TO LOAD BALANCING

Cloud partitions have some issue regarding resource organization and load complementary. Cloud environment is separated into number of parts by making use of cloud group technique which helps for the procedure of load balancing. The cloud consists of number of nodes and it partition into n group based on cloud cluster technique. Within this light or heavily weighed down. Second part is load balance manager that take the exploit of load redeployment if node turn into light or heavy. Load balancing executive balance the load statically at that time new node joins and dynamically when offered node become light or heavily weighed down it consists of main organizer which maintains all information concerning all load balancer in cluster and directory table.

Initial steps is to choose the accurate groups and it follows algorithm as

1. In cloud computing environment, the nodes associated to inner controller are initialized as 0 in catalog table.
2. When controller takes delivery of new request, it reservation the load balancer of each groups for job allowance.
3. Then controller pass catalog table to find next available node contain less weight. If establish then carry on the dispensation otherwise catalog table reinitialized to 0 and in an increment manner then again controller bypass table to find next available node.
4. Following completing the development the load balancer updates the position in allocation table.

Cloud partitioning method concededly 2 steps:-

- ✓ Visit each node randomly to match it with fellow citizen node. If it contain same individuality and shares alike data with negligible cost then two nodes are collective into new node by means of share same details. Do again until there is no fellow citizen node having similar individuality. Consequently update the cost between fellow citizen two users and current neighbors.
- ✓ Following joining two nodes into new node contain similar personality appointment node send the in sequence to new node in its place of sending it twice.
- ✓ It provides the high performance, constancy, minimum reaction time and optimal reserve operation.

LOAD BALANCING IN STRUCTURED PEER TO PEER SYSTEM

Peer to Peer system having solution for locating and sharing resources over internet. In proposed there are two key mechanisms. First is histogram administrator that maintains histogram that reproduces worldwide view of distribution of load. Histogram stores arithmetical information about

standard load of no partly cover groups of nodes. It is used to check whether node is usually loaded, heavily or Light loaded. Second component is load balance manager that take the exploit of load redeployment if node develop into heavy or light. Load balancing manager equilibrium the load statically at what time novel node join and energetically when existing node develop into heavily or light loaded. The cost of construct histogram and preserve it may be luxurious in dynamic system. To reduce the luxurious cost two practices are used. Build and maintaining histogram is luxurious if node joins and leaves system regularly. Every new node in peer to peer system find its fellow citizen node and these neighbor nodes need to share its information with new node to setup connection. .

It contains three different modules.

- 1) File Allocation module
- 2) DHT Division module
- 3) Load Rebalancing module
- 4) Advantage of Node heterogeneity module

1) FILE ALLOCATION MODULE

File allocation large file is partition into number of break apart C1, C2, C3.....Cn is Chunkserver and it allocates to sub servers Chunkserver. Here the files can be appended, added or deleted dynamically to sub server. It will help to avoid the data loss. Fig. [1] Shows that, given large file is dived into number of parts and that parts are distributed over different Chunkserver.

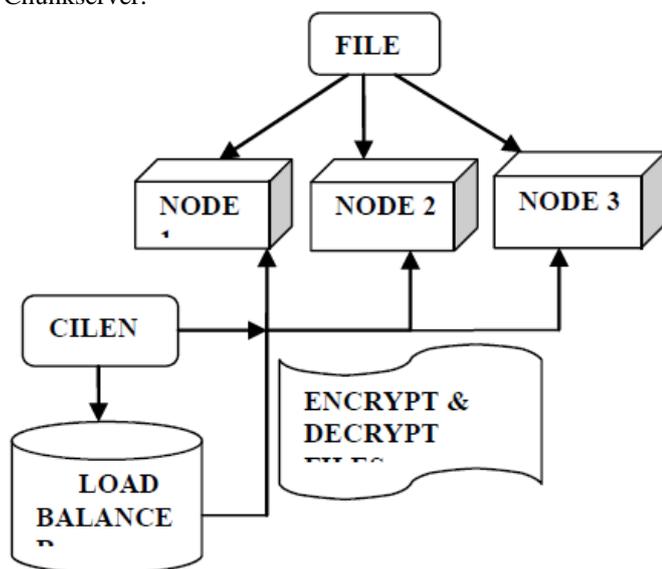


Fig.2 : File allocation with load balanced

2) DHT DEVISING MODULE

The storage nodes are prearranged over network based on the DHT distributed hash table; each file break apart having speedy key search for in DHTs, in that unique identifier is assign to each file break apart. Distributed hash

table DHT agreement that if any node vegetation then allocated break apart are migrate to its successor; if node joins then it allocate the break apart which are stored in successor. distributed hash table DHT network is transparent.

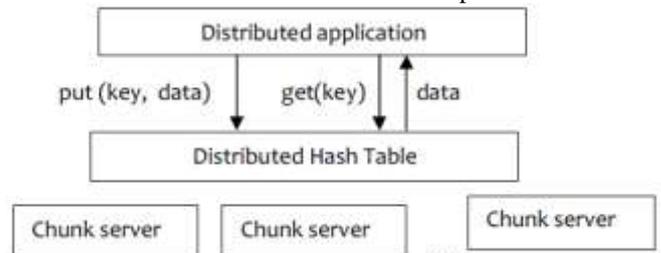


Fig 2: Distributed Hash Table

3) LOAD REBALANCING MODULE

Now node equalization modulus operandi is used to trim down latency, overload and resource utilization. The seriously loaded node relocates requested chunks to flippantly loaded node and it discuss through novel network connections.

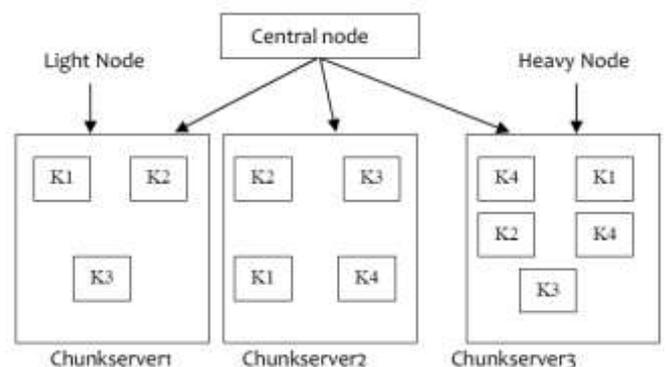


Fig 3: Heavy node and light node

Regard as example, the aptitude of each break apart server is 296MB i.e. 3 break apart each break apart is of 88 MB then according to above Fig. 3 and load rebalancing algorithm chunk server1 is glow node comprise load 272MB no of c break apart and Chunk server 3 is the heavy node comprise load 450MB. Using the load equalization technique, heavy node moving its load to light node i.e. Chunk server 3 move its some load to Chunk server 1.



Fig4 (a): Initial load of chunks

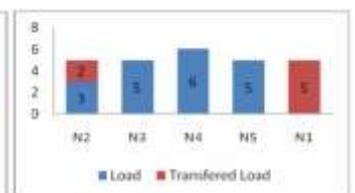


Fig4 (b): Rebalancing the load

LOAD BALANCING ALGORITHM:

1. Initialize server and its sub-servers
2. Establish connection between sub-server and servers using the IP or Port number.

3. Upload File to server that should be shared.
4. Split the file into multiple chunks
5. Calculate the each sub server memory
6. Divide the total chunks value by total number of sub-servers
7. Upload each chunk into sub servers based on its memory capacity
8. If Capacity is less then transfer the excess chunks into next sub-servers
9. Each chunk will be appended with a index value.
10. When the client ask for a file, which will be conventional from dissimilar sub-servers based on the index value.
11. Client collects all the chunks then the file will be decrypted, after that so will be view by user.

ADVANTAGE OF NODE HETEROGENEITY MODULE

Nodes on which file is dispersed are heterogeneous in nature. According to nodes aptitude there is one blockage resource. believe, capacity of nodes (C1 , C2, C3,....., Cn). Each node consist estimated number of file section. The load on those node requirements to be balanced as follows:

$$A_i = \gamma C_i$$

where γ is the weight per unit competence of node. And

$$\gamma = m / \sum_{k=1}^n C_k$$

where m is the quantity of file break apart stored on system.

DATA ENCRYPTION STANDARDS (DES):

The main standard for DES is encrypting data was a symmetric algorithm recognized as the Data Encryption Standard (DES). However, these have now been substitute by a new standard well-known as the Advanced Encryption Standard (AES) which we determination glance at later. DES is a 64 bit chunk ciphers which resources that it encrypts data 64 bits at a time. This is dissimilarity to a watercourse cipher in which only one bit at a time is encrypted.

Once a plain - text message is received to be encrypted, it is set into 64 bit blocks necessary for input. If the numeral of bits in the message is not consistently isolatable by 64, then the last block will be pad. Multiple variations and replacements are included all through in order to augment the complexity of the stage a cryptanalysis on the cipher. On the other hand, it is generally accepted that the original and final permutations offer modest or no payment to the protection of DES and in fact some software completions omit them even though severely.

CONCLUSION

A cloud application is based on the MapReduce programming used in distributed file system. Load unbalancing Problem mostly occur in dynamic, distributed file system and large scale. Load be supposed to be balance over multiple nodes to

develop system performance, resource used, response stability and time. The load rebalancing algorithm the load of users is balanced as well as the group cost also concentrated. The load balancing assignment performs separately without global knowledge of system. This load balancing algorithm has high-speed concurrency rate. Load balancing decline the movement and load cost while it uses corporeal network district and node heterogeneity. A fully wealthy load rebalancing algorithm is demonstrated to cope with the load imbalance problem. Transfer the file data separated different parts of files using load rebalancing and data encryption stranded DES algorithm after store up into clouds.

References:-

- [1]. Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh-Yi Chung, HaiyingShen, Member, IEEE, and Yu-Chang Chao, proposed a "Load Rebalancing for Distributed File Systems in Clouds" IEEE transaction on parallel and distributed systems, vol. 24, no. 5, May 2013
- [2]. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Proc. Sixth Symp. Operating System Designed Implementation (OSDI '04), pp. 137-150, Dec. 2004.
- [3]. S. Mohammad, S. Breß, and E. Schallehn, "Cloud Data Management: A Short Overview and Comparison of Current Approaches," Proc. 24th GI-Workshop Foundations of Databases (Grundlagen von Datenbanken), 2012.
- [4]. Apache Hadoop, <http://hadoop.apache.org/>, 2013
- [5]. P.Jamuna and R.Anand Kumar "Optimized Cloud Computing Technique To Simplify Load Balancing"International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, 11,November 2013.
- [6]. Kokilavani .K, Department Of Pervasive Computing Technology, Kings College Of Engineering, Punalkulam, Tamil nadu "Enhance load balancing algorithm for distributed file system in cloud" International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 6, December 2013
- [7]. Hadoop Distributed File System, <http://hadoop.apache.org/hdfs/>, 2012

AUTHOR PROFILE



A. Sumanth is currently pursuing M.Tech in the Department of Computer Science & Engineering, Visvesvaraya College of Engineering and Technology, M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy(D), India.



Sri Dr. Bhaludra Raveendranadh Singh working as Associate Professor & Principal in Visvesvaraya College of Engineering and Technology. He obtained M.Tech, Ph.D(CSE)., is a young, decent, dynamic Renowned Educationist and Eminent Academician, has overall 20 years of teaching experience in different capacities. He is a life member of CSI, ISTE and also a member of IEEE (USA).



Ms's. Sangeetha M working as Assoc. Professor & HOD (CSE). She has completed bachelor of technology from Swamy Ramananda Theertha Institute of Science & Technology and Post-graduation from Jawaharlal Nehru Technological University, Kakinada campus and is having 12 years of teaching experience and she has published so many papers in national and international journals.