International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 131-135
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

AN EFFICIENT SCHEDULING AND LOAD REBALANCING STRATEGY IN DFS WITH EXCELLENT SECURITY MECHANISM

Miss. Priyanka A. Dhande
Department Of Computer Engineering, Aissms
College Of Engineering, Keneddy Road, Pune
411001

Prof. A. J. Kadam
Department Of Computer Engineering, Aissms
College Of Engineering, Keneddy Road, Pune
411001

***Abstract*— disbursed file system could be a preliminary section building block within the cloud computing. A widespread quantity of statistics is categories over the amount of bite servers and allocates each chunk to separate distinctive node to hold out Map-Reduce characteristic parallel over each precise node that's the primary functionality in distributed system. If immense wide range of documents are present, and assesses to it large vary of document is will increase then the principal node turns into impediment in cloud computing. the burden balancing task is employed to limit the burden on basic centralized node with the assist of load rebalancing set of rules the load of nodes is balanced additionally to the movement value is additionally reduced. It leads to load asymmetry in disbursed system. To beat the burden inequality problem, distributed load rebalancing algorithmic has been enforced that depends upon on:**

- **Performance of device.**
- **Evaluation of load in among precise nodes.**
- **Stability of various structures.**
- **Interaction among the nodes.**
- **Nature of load to be moved.**

Protection mechanism has been applied on documents. At an equivalent time as importing or downloading files are holds on sub servers and retrieved from sub servers. Earlier than importing or downloading that record are saved in encrypted format and retrieved from the decrypted format.

Keywords— Cloud computing, Load re-balancing, Distributed file system, MD5 with DES.

I. INTRODUCTION

On-demand get admissions to from shared pool configurable computing assets are often equipped through the cloud computing version. software , application, platform associated offerings ar well-found via the cloud computing which are often accessible while not requiring any ability more or less the technology, and accessible with abundant less management efforts. Cloud is often personal or public and it imply less understanding to use as a result of its user friendly system. The aim of cloud computing is to supply smooth, climbable get right of entry to computing assets and it offerings. Allotted report systems in producing structures powerfully depend on a significant node for bite reallocation. this dependence is just inadequate in an exhaustive, failure-prone surroundings as a result of the very fact the central load balancer is place beneath full-size workload that's linearly scaled with the device length, and should consequently turn out to be the performance bottleneck and also the point of failure. The aim to cut back movement cost as a results of rebalancing the many nodes as lots as potential to maximise the network bandwidth to be had to everyday packages. We have a tendency to examine to place good safety provided for cloud computing and compare machine performance. In cloud master server controls amount of slaves, documents it might transfer, eliminate, and append dynamically. As an example uploading or downloading of report is completed that's kept in slaves and retrieved from slaves itself. Before uploading and downloading that files are saved in encrypted layout and retrieve during a decrypted format. For encryption and decryption schemes we are the usage of md5 set of rules.
If the large file is shipped over the system which is completely depending on crucial node then there's lot of load can be available over the central node. Right here the load rebalancing algorithm is used to cope with imbalance of load [1].

II. EXISTING SYSTEM

Traditional model is simply now not suitable to technique massive extent of records and that database isn't accommodated through upscale database server. Centralized

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 131-135
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

machine creates many bottlenecks while processing more than one file simultaneously.
Boundaries within the existing device encompass load imbalance factor, protection problems, absolutely dependence on principal node, and the movement cost of load elements.
Traditional model is not acceptable to methodology massive volume of data at intervals that, information is not accommodated by traditional information servers. Centralized system creates several bottlenecks whereas methodology multiple files at the same time shown at intervals the prevailing System style Diagram. Limitations among the prevailing system embody load imbalance issue, security issues, altogether dependence on central node, and therefore the movement cost of load factors.
State-of-the-art distributed file systems (e.g., GFS and HDFS) in clouds hold central nodes to manage knowledge of filing systems and to counterbalance the commutuality of storage nodes based mostly completely on it data. The centralized views clarify the planning and implementation of a distributed classification system.
However, gift days ignorance concludes that once the choice of storage nodes, the variability of files and therefore the variability of accesses to files increase unceasingly, the centralized nodes (e.g., the master in GFS) become a performance bottleneck, as they are unable to accommodate an oversized choice of file accesses as a result of sizable amount of purchasers and existing operations.[1][2]
The most existing solutions are designed whereas not considering movement cost and node.
Conclusion from the literature is that current system is having a number of the constraints that want to be addressed.

III. OBJECTIVES

General objectives of this system are

- Basically depiction on data processing to handle the huge records and to boom the system performance
- To modify the load imbalance aspect, allocate chew of documents unambiguously most of the distinctive nodes such no node manages an excessive wide selection of chunks
- The security provided in cloud is incredibly confined. So, for reinforcing the security md5 algorithms are enclosed to the device.

IV. SYSTEM MODEL

Here, tremendous amount of info document is divided into wide selection of chunks and it distributes to nodes like (node 1, node 2) the files are often dropped at the sub server, deleted or appended dynamically over sub server. It will assist to attenuate the shortage of statistics. Fig.1 shows that, given massive record is dived into amount of components which part are distributed over distinctive bite servers shown by using node one, node 2.

Suppose person needs to add his record the job manager can distribute the burden of major server into sub servers. The decision node within which the job manager will work is used to control all the meta-facts records and calculate load of each node.
However, it actions from the centralized to distributed system. The equally load rebalancing set of rules is also distributed over each node and also the md5 with DES set of rules are applied over node subsequently the facts node provides all of the community services. Customers are capable of access statistics from sub server that's records node notably do to permit to save lots of statistics or retrieval of records whereas computation is performed that will minimizes the workload of the principle server.

- Suppose, the system consisting Module 1, Module 2, Module 3, and Module 4 which consist for analysis and reading the document.

Where,
Module 1 shows creation of virtual machines.
Module 2 shows creation of the data chunk module.
Module3 shows particular weight of the node calculation.
Module 4 shows rebalancing module.

- For creation of virtual machines:

Let Module1 consisting set of parameters for virtual machine creation and Module2 for creation of data chunk.
Module1& Module2= {VM Size of virtual machine, data in chunks}
Where,
Chunk Size of data=Exact size of file
Data Node consist data which can be used to store.
Weight calculation of data node can be performed.
Lets module 3 calculate the weight of data node.
Module3= all data nodes with chunks}

- Automatic Rebalancing can be active through

If weight of the node is less than the average weight => proceed
Otherwise check next node
If node having Excessive load
While (check all data nodes weight)
Otherwise create new node.
Module4 is used to generate maps and store and retrieve data from data nodes, analysis graph.
Security part has been done using MD5_DES algorithm because it is simple to implement and it is used to provide hashing with message digest of the message of arbitory length and data encryption standard.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 131-135
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

A. System Architecture:

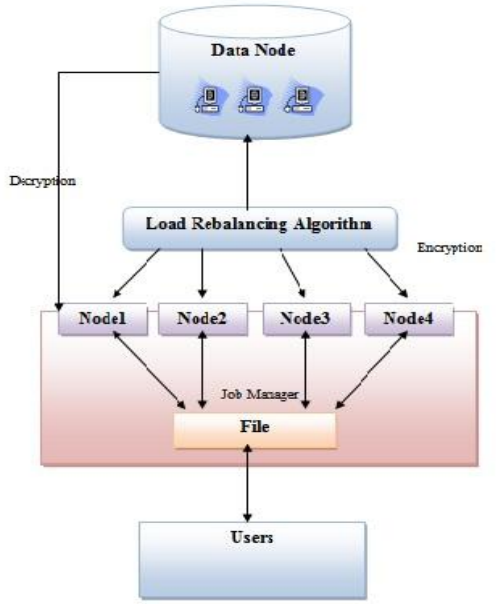

Fig 1: System Architecture

B. Procedure:

- To begin with the facts document need to be uploading from client over the server.
- It goes towards activity manager or job manager; with the assist of weight calculation set of rules activity manager calculate the each facts node load, before dedicate on information.
- Advent of the report chunks for every server primarily based whole on server load are often acting.
- Statistics cryptography should be done through server with MD5-DES cryptography.
- Assign each chunk to specific slaves.
- Load rebalancing venture are often finished.
- Consumer collects all of the chunks then the record can be decrypted.
- Then a good way to be regarded by means of consumer.

C. Necessity of Load Rebalancing:

- Do away with the dependence on relevant nodes.
- Operates in a distributed manner wherein nodes perform their load-balancing duties independently without international knowledge of the system.

***A. Algorithm**:* Equally Load Re-Balancer Algorithm
Input: every node load base on present day hitting
Output: allotted data to every node.
Step 1: initialize all information nodes or slaves which may be connected to master node as n.
Step 2: for every (i…n)
Take every i^{th} node server load.
A[i] =&weight; i^{th} node load degree or hit load.
End for.
Step 3: get total length of A.
Create the statistics asked chunks.
K=A. length ();
Step 4: generate k mappers for distribute records.
Step 5: assign every block to every mapper.
Step 6: request to server for handle an info.
Step 7: finish procedure.

***B. Algorithm**: Load Balancer Algorithm.*
Input: files with data.
Output: File procedure with server load balancing.
Step 1: Initialize servers and its sub-servers.
Step 2: Establish connection between sub-server and servers using the IP address or Port number.
Step 3: Upload File to server that should be shared.
Step 4: Data encryption should be performed by server with MD5 Encryption.
Step 5: Disperse the file into multiple chunks.
Step 6: Calculation of each sub server memory
Step 7: Divide the total chunks value by total number of sub-servers
Step 8: Based on chunk memory capacity Upload each chunk into sub servers
Step 9: If Capacity is less then transfer the excess chunks into next sub-servers
Step 10: Each chunk will be appended with an index value.
Step 11: When the client request for a file, that will be received from different sub-servers based on the index value.
Step 12: Client collects all the chunks then the file will be decrypted, then that will be viewed by client.

V. IMPLIMENTATION

During this work the proposed solution are investigating and presenting the new framework for addressing the problem of finding relevant result. The aim of this work is to improve the performance of algorithm. The results demonstrated in this work are showing the current state of work done over practical implementation of this algorithm.

A. Modules

This proposed architecture consists of four modules.

- First module is job scheduler. The client submits their requests on server in the cloud environment. When an available resource task is assigned to a cloud, availability of resources can be checked by job scheduler. First file uploading and user authentication.
- Second server load degree calculation and chunk creation.
- Third module provide the re-balancing the data between servers base on current loads.
- Final describe the performance analysis of system.

International Journal of Engineering Applied Sciences and Technology, 2016
Vol. 1, Issue 9, ISSN No. 2455-2143, Pages 131-135
Published Online July – August 2016 in IJEAST (http://www.ijeast.com)

B. Module Description

1) Authentication and File upload:

This module describe the user login registration likewise as file uploading half, once transfer the file key generation and encoding has done.

2) Server Load Calculation:

This module every server calculation done base on memory likewise processor.

3) Load Rebalancing:

This module describes the load distribution and chunks in numerous servers.

4) Analysis Module:

This module reflects the projected system accuracy comparison with different systems. Totally different graphs can show the planned system accuracy, and eventually we tend to conclude our system is healthier than all the present systems.

VI. Result And Disscussion

After implementing some a part of system we tend to got system performance on satisfactory level. Fig.2 shows the primary algorithmic program performance for user plain conversion in addition decoding. For the implementation we tend to use virtual machines as information node and distribute the info into totally different servers. Within the check part system initial save the information into every data node parallel with cryptography and collect as same decoding. Following graph shows system performance of each ways.

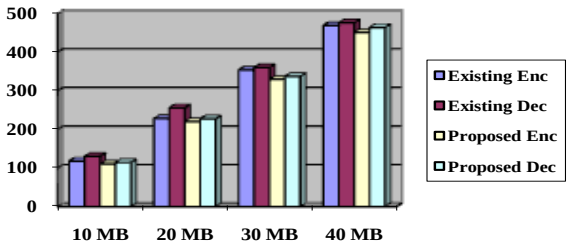

Fig.2 Execution Performance with Data Nodes

The above graph compares with existing results taken from ECC encryption and used MD5. Both algorithms bar shows the how much time or execution complexity is required for both encryption as well as decryption.

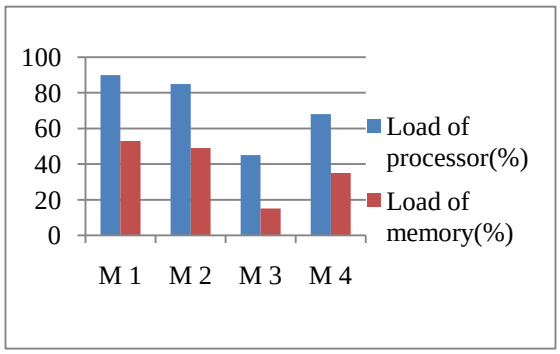

Fig.3 Calculation of Load for each VM

Graph shows the each VM load in percentage with memory as well processor. Second algorithm calculate each virtual machine load dynamically after create data chunks. X graph show the each VM id and Y shows the load of percentage.

VII. Conclusion

Load rebalancing drawback principally happen in distributed filing system. Load ought to be balance over numerous hubs to boost framework execution, latent period and MD5 security mechanism is additionally applied over the rebalancing task. Load rebalancing for distributed file systems is specially used for: big-scale, dynamic and intensive clouds of knowledge that reallocates the file chunks in an exceedingly uniform manner. None of the node will manage excessive quantity of load.

VIII. References

[1]. Hung-Chang Hsiao and H.C et al, “Load Rebalancing for Distributed File Systems in Clouds”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 5, MAY 2013.

[2] L.Ismail et al, “FSBD: A Framework for Scheduling of Big Data Mining in Cloud Computing”. 2014 IEEE International Congress on Big Data.

[3] 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.

[4] 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.

[5] 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.

[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] S.Indira and P.Jyothi, "Load Rebalancing In Large-Scale Distributed File System," Proc. International Journal Of Reviews On
Recent Electronics and Computer Science (IJRRECS), Volume-1, Issue-6, Issn 2321-5461, October 2013.
[13] Ch. Mounika, L. RamaDevi and P.Nikhila, "Simple Load Rebalancing For Distributed Hash Tables In Cloud," Proc. IOSR Journal
of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p-ISSN: 2278-8727 Volume 13, Issue 2, pp 60-65, Jul. - Aug. 2013.