Available at: www.researchpublications.org

A Review:Rebalancing The Load For Distributed File System In Clouds

Ms.Komal Pralhad Sonawane [#1], Prof.Ankush Narkhede [*2]

[#] Computer Engineering Department, Sant Gadge Baba Amravati University
Padmashri Dr. V.B.Kolte college of Engineering and Polytechnic, Malkapur, District Buldhana, (M.S.) 443101, India.
[1]komal.sonawanep@gmail.com

[*]Padmashri Dr. V.B.Kolte college of Engineering and Polytechnic, Malkapur, District Buldhana, (M.S.) 443101, India.
[2]ankushnarkhede1989@gmail.com

ABSTRACT

For cloud computing applications the Distributed file system is used as a key building block which is simply a classical model. In such file system a file is partitioned into a number of chunks allocated in distinct nodes .Each chunk allocates to separate node to perform MapReduce function parallel over each node. In cloud, the central node (master in MapReduce) becomes bottleneck if number of storage nodes, number of files and assesses to that file increases. In this survey paper to overcome the above load imbalance problem the fully distributed load rebalancing algorithm is used to exclude the load on central node and also the movement cost is reduced. In this paper the load misbalancing problem is overcome.

Keywords— MapReduce, Load balancing, Distributed file systems, clouds.

I.INTRODUCTION

Cloud computing is a recently evolved computing terminology or metaphor based on utility and consumption of computing assets. Cloud computing includes deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. It is model in which the resources can be authorized on per use basis thus reducing the cost and complexity of service providers. Instead of keeping data centres running, cloud computing assures to cut operational and capital costs and more importantly let IT departments focus on crucial & important projects. It is much simpler than internet. It is a form that allows user to access applications that actually settled at location other than other Internet-connected devices or user's own computer or. There are several advantages of this construct. Other company hosts user application. This indicates that they manage software updates handle cost of servers and depending on the contract user pays less i.e. for the service only. Authenticity Availability, Confidentiality, Integrity and Privacy are crucial a cause of anxiety or worry for both customer and Cloud providers as wellDistributed file system which is classical model of file system that is utilized in the form of chunks for cloud computing. MapReduce programming used in distributed file system on which the Cloud computing application is based on. In Hadoop MapReduce is the master-slave architecture. Name node act as Master and Data node act like Slave Master takes immensely colossal quandary divides it into sub quandary and assigns it to worker node i.e. to multiple slaves to solve quandary individually. In distributed file system, a sizably voluminous file is divided into number of chunks and allocates each chunk to disunite node to perform MapReduce function parallel over each node. For example in word count application it identifies the occurrences of each distinct word astronomically immense file. In this application a sizably voluminous file is divided into fine-tuned-size chunks (components) and assigns each chunk to different cloud storage nod. Then each storage node calculates the existence of each distinct word by scanning and parsing its own chunk. Then result is given to master to calculate the final result. In this way in distributed file system, the load of each node is directly proportional to number of file chunks that node consists.

For efficient operations in distributed environments Load balancing is essential. It means distributing the quantity of work to do between different servers for the sake of getting more work done in the same amount of time and serve customers faster. In this case, look attentively at a large-scale distributed file system. The system contains N chunk servers in a cloud where a certain number of files are stored. Each file is break into various parts or chunks of fixed size (for example 64 megabytes). The load of each chunk server is proportional to the number of chunks hosted by the server. In a load-balanced cloud, the resources can be well used while maximizing the performance of MapReduce-based applications.

Cloud load balancing reduces costs associated with document management systems and maximizes availability of resources. Cloud load balancing is the process of distributing workloads across multiple computing resources. It is a type of load balancing and not to be discombobulated with Domain Name System (DNS) load balancing. On other hand when DNS load balancing uses software or hardware to perform the function, cloud load balancing uses accommodations offered by different computer network companies.

Cloud computing brings advantages in "cost, flexibility and availability of accommodation users." The requirement upends technical issues in Accommodation Oriented Architectures and Internet of Accommodations (IoS)-style applications, like high scalability and

accessibility. As a major trouble in these issues, load balancing sanctions cloud computing to "scale up to incrementing demands" by efficiently allocating dynamic local workload evenly across all nodes.

II.LITERATURE SURVEY

A.MapReduce: Simplified Data Processing On Large Clusters [8]

MapReduce is the programming model in implementation which is used for generating or causing sizably and processing large amount of datasets. It is used at Google for many different causes or say occasions. Here map and reduce functions are used. Map function generate set of median key pairs and reduce function merges all median key values associated with same median key. The map and reduce function sanctions to perform parallelize operation facilely and re-execute the mechanism for fault toleration. At the execution-time, system deals with particular information of partitioning the input data, schedule the program execution over the number of available machines, handling failures and managing intercommunication between machines.

In distributed file system nodes concurrently perform computing and storage operations. The relatively huge file in partitioned into number of chunks and allocate it to distinct nodes to perform MapReduce task parallel over nodes. Typically, MapReduce task processes on different terabytes of data on thousands of machines. This model is is over simple to use; it hides the details of optimization, fault-tolerance, parallelization, and load balancing. MapReduce is utilized for Google's engenderment Web search accommodation, machine learning, data mining, etc. Utilizing this programming model, redundant execution used to reduce the impact of slow machines, handle machine failure as well as data loss.

B.Algorithm: Load Balancing for DHT based structured Peer to Peer System [2]

Peer to peer system have an emerging application in distributed environment. As compared to client-serverarchitecture, peer to peer system ameliorated resource utilization by making utilization of unutilized resources over network. Peer to peer system uses Distributed Hash Table (DHTs) as an allocation mechanism. It perform join, leave and update operations. Here load balancing algorithm utilizes the concept of virtual server to ad interim storage of data. Utilizing the heterogeneous indexing, peers balanced their loads proportional to their capacities. In this, decentralized load balance algorithm construct network to manipulate ecumenical information and organized in tree shape fashion. Each peer can independently compute probability distribution capacities of participating peers and reallocate their load in parallel.

C.Optimized Cloud splitting/partitioning Technique to Simplify Load Balancing [7]

Cloud computing has some difficulty regarding resource management and load balancing. In this paper, Cloud Environment is partitioned into number of components by making utilization of cloud cluster technique which utilizes for the process of load balancing. The cloud related to number of nodes and it partitioned into n cluster predicated on cloud cluster technique. In this, it consists of main controller which maintains all information regarding all load balancer in cluster and index table. Very initial step is to select the correct cluster and it follows algorithm as

1) In the environment of cloud, the nodes connected to centralized controller which is initialized as 0 in index table.
2) When controller receives incipient request, it queries the load balancer of each cluster for job allocation.
3) Then controller pass index table to find next available node having less weight. If found then sustain or continue the Processing otherwise index table again initialized to 0 and in an increment manner then again controller passes table to find next available node.
4) After fulfilling or say completing the process the load balancer update the status in allocation table.

Cloud splitting/partitioning method consists of following 2 steps:-

1) To match it with neighbor node visits each node randomly. If it having same characteristics and allocates and distributes familiar data with minimal cost then two nodes are merged into incipient node by sharing same details. Reiterate until there is no neighbor node having comparable characteristics. Eventually update the cost between neighbor two nodes and current neighbor node.
2) After combining two nodes into new node having similar characteristics visited node send the information to new node instead of sending it twice. It gives the stability, minimum replication, high performance, time and optimal resource utilization

D.Histogram-Based Global Load Balancing in Structured Peer to Peer System [9]

Peer to peer system having solution for sharing and locating resources over internet. In this paper there are two key components. First is histogram manager that maintain histogram that reflect ecumenical view of distribution of load.

Histogram stores statistical information about average load of no overlapping groups of nodes. It is utilized to check if node is mundanely loaded, light or heavily loaded. Second component is load balance manager that take the Action of load redistribution if node becomes light or cumbersomely hefty. Load balancing manager balance the load statically when incipient node joins and dynamically when subsisting node become light or heavily loaded. The cost of constructing histogram and maintaining it may be expensive in dynamic system. To minimize the maintaining cost two techniques are used. Constructing and maintaining histogram is expensive if node join and leave system

frequently. Every new node in peer to peer system find its neighbor node and these neighbor nodes need to share its information with new node to setup connection. Now the cost of histogram is totally based on histogram update message caused by changing the load of nodes in the system .To reduce the cost approximate value of histogram is taken.

III. EXISTING SYSTEM:

Existing system as fig shows is depending on centralized node. Due to central node failure performance bottleneck is occurred. And because of load imbalance low productivity is achieved.

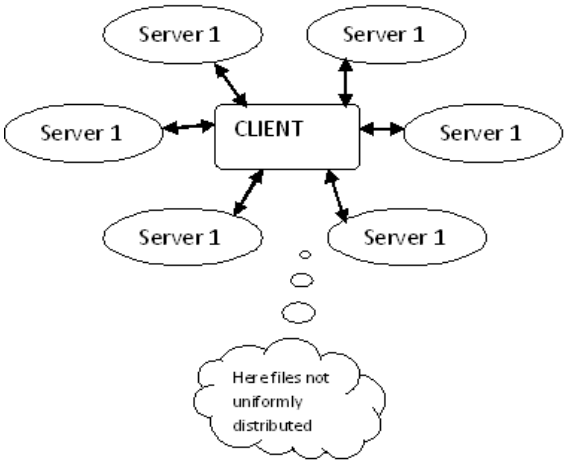


Fig. Existing System Diagram

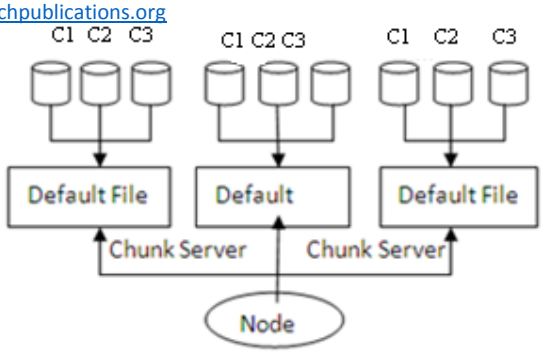
Some limitations of existing system are high network traffic; movement cost is high, algorithm overhead, load imbalance, relying on Central Node, security difficulties.

VI.PRAPOSED SYSTEM **[1][[3][4]**

Proposed system contains following different modules.

1) File Allocation
2) DHT Division
3) Load Rebalancing
4) Advantage of Node Heterogeneity

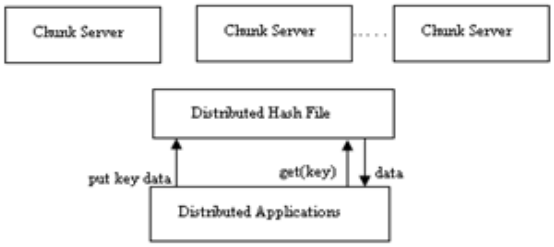
1) File Allocation: In this, large file is partitioned into number of chunks (C1, C2, and C3…..Cn) and it allocates to sub servers (Chunk server). Here the files can be added, deleted or appended dynamically to sub server. It will help to avoid the dataloss. Fig. [1] Shows that, given large file is dived into number of parts and that part are distributed over differentChunk server.



Fig[1] File Allocation

2)DHT Devising: The storage nodes are structured over network based on the distributed hash table (DHT); each file chunk having rapid key lookup in DHTs, in that unique identifier is assign to each file chunk. DHT guarantees that if any node leaves then allocated chunks are migrated to its successor; it allocates the chunks which are stored in successor only if node joins. DHT network is clear and it specifies the location of chunks that can be integrated with existing distributed file system where master node manages namespace of file system and mapping of chunks to storage nodes.

Fig.[2] shows the structure of Distributed Hash Table which is visible to all nodes.

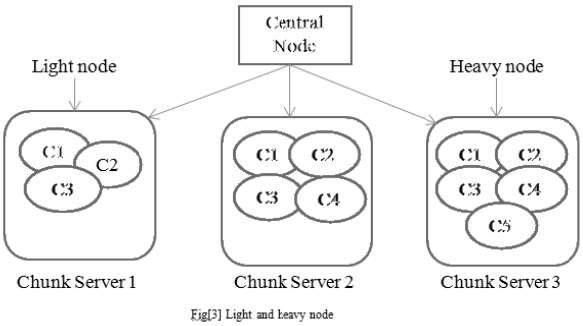


Fig[2] Distributed Hash Table

3) Load Rebalancing: First calculate whether loads are heavy or light in each sub server. And a node is heavy if, Number of chunk > (1-ΔU) A node is Light if, Number of chunk < (1-ΔL) A**.**

Where ΔL and ΔU are parameters of the system. All heavy nodes shared its load with light node. If node 'i' is the lightlyloaded then it contact to its successor 'i+1'and migrate its load to its successor and then join instantly to heavy node.

Here node equalization technique is used to reduce latency, overload and resource usage. The highly loaded node transfer requested chunks to lightly loaded node and it traverse through physical network link.Consider example, the capacity of each Chunk server is 256MB i.e. 3 chunks (each chunk is of 64 MB) then according to above Fig. 3 and load rebalancing algorithm chunkserver1 is light node having load 192MB (no of chunk <(1-ΔL) A) and Chunk server 3 is the heavy node having load 320MB (no of chunk > (1-ΔU)A).



Fig[3] Light and heavy node

Using the load equalization technique, heavy node transfers its load light node i.e. Fig [3] Chunk server 3 transfers its some load to Chunk server 1.

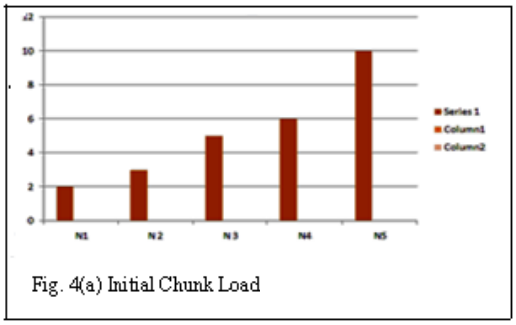


Fig. 4(a) Initial Chunk Load

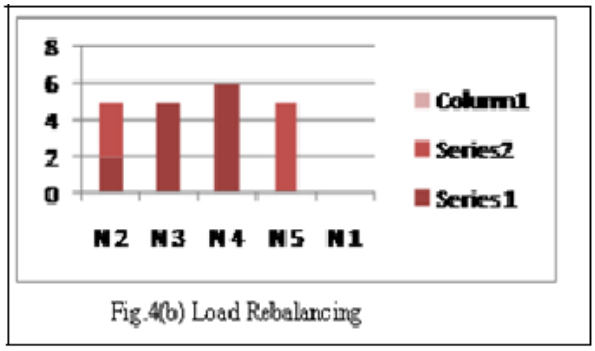


Fig.4(b) Load Rebalancing

Consider example, in Fig. 4(a) capacity of each node is 5 chunks (A) but here N1 is the light node having 2 hunks and N5 is heavy node made up of 10 chunks. Then N1 identifies that N5 is the heavy node then it contact to its successor i.e. to N2 and transfer its own load to N2. And request to heavy node for some load. Min (Lj-A),A) that much load is transfer from heavy node to light node. According to formulae 5 chunks are transfer from N5 to N1 (Fig 4(b)).

4) Advantage of Node Heterogeneity:Nodes on which file is distributed are heterogeneous in nature. According to nodes capacity there is one bottleneck resource. Consider capacity of nodes (Cp1, Cp2, Cp3,...., Cpn). Each node consist approximate number offile chunks. The load on that node needs to be balanced as follows: $A_i = \gamma C_{pi}$where γ is the load per unit capacity of node. And$\gamma = m/\Sigma_{k=1} C_{pk}$ where m is the number of file chunks stored on system.

V. CONCLUSION

Cloud computing application is based on the MapReduce programming which is used in DFS. Load unbalancing Problem mostly occurs in dynamic, large-scale and distributed file system. Load should be balance over multiple nodes to upgrade system performance, resource utilization, replication time and stability. In the load rebalancing algorithm the load of node is balanced as well as the kineticism cost additionally reduced. The load balancing task performs independently without ecumenical erudition of system. This load balancing algorithm has fast concurrency rate. Load balancing reduce the load and movement cost while it uses physical network locality and node heterogeneity.

REFERENCES

[1] Hung-Chang Hsiao, Yi Chung, Haiying Shen, Yu-ChangChao,"Load Rebalancing for DistributedFile Systems in Clouds" IEEE transaction onparallel and distributed systems, vol. 24, no. 5,May 2013

[2] ChahitaTanak, Rajesh Bharati "Load Balancing Algorithm DHT Based Structured Peer to Peer System"International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 1, January 2013)

[3] Suriya Mary,Metal, International Journal of Computer Science and Mobile Computing, Vol.2 Issue. 12, December- 2013,

[4] Kokilavani .K, Department Of Pervasive ComputingTechnology, Kings College Of Engineering, Punalkulam, Tamilnadu "Enhanced load balancing algorithm for distributed filesystem in cloud" International Journal of Engineering and Innovative Technology Volume 3, December 2013

[5] Apache Hadoop, http://hadoop.apache.org/, 2012.

[6] Hadoop Distributed File System "Rebalancing Blocks,"http://developer.yahoo.com/hadoop/tutorial/module2.html#rebalancing,2012

[7] P.Jamuna, R.Anand Kumar "Optimized Cloud Computing Technique to Simplify LoadBalancing "International Journal of AdvancedResearch in Computer Science and Software Engineering, Volume 3,November 2013.

[8] 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.

[9] QuangHieu Vu, Member, IEEE, Beng Chin Ooi, MartinRinard and Kian-Lee Tan "Histogram-BasedGlobal Load Balancing in Structured Peer-to-Peer Systems" IEEE transaction on knowledge and data engineering,vol. 21, no. 4, April2009