IJCSE International Journal of Computer Sciences and Engineering Open Access

Survey Paper Volume-3, Issue-1 E-ISSN: 2347-2693

A Study of Load Balancing Techniques in Cloud

Martina Poulose[1*] and M. Azath[2]

[1]PG Scholar, Department of computer Science,Met's School of engineering, Kerala, India
[2]Head of Department, Department of computer Science, Met's School of engineering, Kerala, India

www.ijcaonline.org

Received: Dec /26/2014 Revised: Jan/8/2015 Accepted: Jan/20/2015 Published: Jan/31/2015

Abstract— Cloud computing is a promising computing paradigm. Load balancing and rebalancing in cloud are important and challenging research area. Distributed file systems is the main building block in cloud computing. The large files will be divided into number of chunks and distributed into different systems. These chunks were allocated to each node to perform map reduce functions parallel over the nodes. Cloud is a dynamic environment, updating, replacing and adding of new nodes to the environment is a normal concern. This will impact the anatomy of the system and the chunk distribution will become uneven among the nodes. To overcome this, reallocate the chunks uniformly in the nodes. Load balancing and re-balancing helps to achieve high user satisfaction and well resource utilization. Emerging distributed systems are strongly depends on a central node for chunk reallocation. In a giant cloud central load balancer is put under significant workload and may lead to a performance bottleneck and single point of failure. This survey aims to study the different algorithms and issues of load balancing in cloud computing.

Keywords—Allocation;Chunks;Cloud Computing;Distributed File System;Load Balancing

I. Introduction

Cloud computing is the "wave of future". It is a buzzword that means it is different for different people. Cloud computing is an on demand service in which resources, information, software and other devices are provided according to the clients requirement at specific time. The information stored on servers, but can be accessed and changed by all cloud clients [13]. In fact, Cloud computing connects so many nodes together for allocating resources dynamically. The entities in large clouds may randomly fail or join while maintaining system reliability.

Distributed File System is a key component of large scale cloud computing platforms. It includes many systems that allow multiple users to access files distributed on multiple machines via computer network. Cloud computing supports a distributed architecture and its operations are performed in a distributed manner. Load balancing and re-balancing is the main key factor in a large distributed environment for the proper distribution of load among the nodes. In distributed file system, a large file is divided into number of chunks and allocates each chunk to separate nodes to perform Map-Reduce function parallel over each node [1]. The main aim of load balancing is to allocate files to these nodes without applying heavy load to any of the nodes. Another objective is to reduce the network traffic and inconsistencies due to the imbalance of loads. Effective load balancing will reduce the network bandwidth utilization in distributed file systems. The load of each node should be balanced to improve system performance, resource utilization, response time and stability.

The existing cloud systems heavily depend on a central node to balance the loads in their nodes. If the number of subscribing client increases, linearly, the central node becomes a performance bottle neck. They cannot accommodate a large number of file accesses due to the large number of process for Map-Reduce function in central node. It intensifies the load imbalance problem. A load balancing algorithm tries to balance the total system load by transferring the workload from heavily loaded nodes to lightly loaded nodes to ensure good overall performance relative to some specific metric of system performance. In this study there were several existing load balancing techniques are discussed.

II. CHALLENGES FOR LOAD BALANCING

Load balancing and rebalancing is a complicated task in cloud computing. The main challenges [2] in load balancing are summarized below.

A. Automated Provisioning Of Service

The key feature of cloud computing is flexibility. The advantage of automated provisioning is to provide the flexibility in the cloud**.** That means resources can be allocated or released automatically.

B. Node's Spatial Distribution

Several existing algorithms are efficient only for closely located nodes where negligible communication delays are present. It is a basic issue for implementation of load balancing algorithm that work for spatially distributed nodes. To give maximum service quality, the speed of the network links, the distance between the task nodes and client node, and also the distances between the nodes must be considered. A good load balancing algorithm efficiently and effectively tolerates large delays.

C. Storage And Replica Management

Replica management is important and algorithm that supports full replication does not give efficient storage utilization. All replication nodes contain the same data and it causes higher costs for more storage. The algorithm which has constant number of replica provides better performance. But replication increases the complexity of the load balancing algorithm when file availability needed due to node failure.

D. Centralized Approach

Centralized approach simplifies the design and work effectively and efficiently. The disadvantage of these algorithms is the single point of failure. They have one central node for controlling all activities. The central node becomes the performance bottleneck when the number of file access increases. If the central node fails, then the whole system would fail. The distributed load balancing algorithm avoids the single point of failure. This approach gives better performance and reduces the network traffic.

E. Complexity

Load balancing algorithms with less complexity are better for good performance. There will be negative effect for more complex algorithms. So simple load balancing algorithms are more preferred.

F. Migration Time

It is the time taken by the process to migrate from one node to another. If migration time increases, performance of the algorithm decreases and vice versa.

III. REVIEW OF LOAD BALANCING ALGORITHMS

A. Round Robin Algorithm

This is a famous load balancing algorithm based on the random sampling method. This algorithm randomly selects the load when imbalance occurs. An unbalanced system contain both overload and under-load nodes. It requests for the node with least connections [4]. This algorithm is not suitable for all kinds of system.

B. Throttled Load Balancing Algorithm

It selects proper virtual machine for allocating a particular job. The job manager contain list of all virtual machines [4]. It assigns particular job to a particular machine using index list. If the job is suited for a particular machine than current machine, then that job assigns to that machine. It put the job in queue if no machines are available.

C. Biased Random Sampling

In this method servers are treated as nodes. Virtual graph is constructed by load on the nodes with degree directed to respective resources to the server [5]. When the server starts executing the job it reduces the in-degree. That means there is a reduction in availability of free resources. Also the server completes the job; the in-degree gets incremented. It indicates the increase in availability of resources. When the execution starts at any node and the random neighboring node will be select for next job to be executed. This load balancing helps to achieve the efficiency and suited for many cloud networks.

D. Equally Spread Execution Algorithm

It spreads the loads into different virtual machines. The loads are distributed randomly in virtual machines based on the priority [4]. The file size and load capacity of each virtual machine would be checked before performing the distribution. This method uses spread spectrum technique and also maintains a queue. It achieves high throughput and take less time.

E. Active Clustering

This is a modified version of random sampling. This algorithm works on the principle of grouping similar nodes together and start working on this group nodes .This method [12] uses the resources efficiently. The match-maker concept was introduced in this algorithm. The process gets initiates and searches for the next matching node when an execution starts in the network. It satisfies the criteria that the node is different from the former one. If the match-maker is found the process gets initiated, gets over the match-maker and gets detached from the network. It is an iterative process for balancing the nodes.

F. A Two Level Task Scheduling Algorithm

This method obtains dynamic requirements of users and high resource utilization. It maps the tasks to virtual machines and then virtual machines to host resources [6]. Thus the load balancing is achieved. It improves the task response time and resource utilization.

G. Min-Min Algorithm

It contains a set of tasks and calculates the minimum execution time for all tasks [10].The minimum time is selected and according to the minimum time, the task is scheduled on corresponding machine. It updates the execution time of all other tasks by adding the execution time of the assigned task. The assigned task is removed from the list of the tasks. The same procedure will repeat until all the tasks are assigned on the resources. The drawback of this method is starvation.

H. Max-Min Algorithm

This method is little different from the min-min algorithm. The minimum execution time is calculated and maximum value is selected [11].The task with maximum time is scheduled for the corresponding machine. The execution time of all other tasks will get updated in machines. And apply the remaining procedure of min-min algorithm

I. Token Routing Algorithm

The routing algorithm needs information about workload distribution. This drawback can be removed with the help of heuristic approach of token based load balancing. This algorithm [7] provides the fast and efficient routing decision. To make their decision about the token where to pass, they actually build their own knowledge base. It is derived from previously received tokens.

J. Honey Bee Foraging Algorithm

This algorithm is based on the behavior of honey bees in finding their food [3].It is a nature inspired algorithm. It groups virtual servers. It also maintains a queue for each virtual server. When a request is served by the server, it calculates the profit and compares it with the colony profit. If high profit exists then the server stand on the current virtual server otherwise server returns to the forage behavior.

K. Ant Colony Optimization Algorithm

This algorithm based on ant's pheromone to Collect and update information about the nodes and selecting a particular node to assign the work. Ant starts its movement when the request is initiated [9].There are two types of movements. In Forward movement ant continuously moving forward direction and check each node is overloaded or under loaded. In Backward Movement ant moves in backward direction. If ant finds the target node for overloaded node, it will commit suicide. And thus load balancing is done. This is a nature inspired algorithm.

TABLE I
Comparison of Load Balancing Techniques in Cloud

LOAD BALANCING ALGORITHMS	**MERITS**	**DEMERITS**
Round Robin Algorithm	High throughput, Less Overhead	Less Fault tolerance, Increased network traffic
Biased Random Sampling	High Resource Utilization, Increased Performance	Overhead is High, More migration time
Equally Spread execution Algorithm	Improved Performance	High computational overhead
Token Routing Algorithm	Minimum System cost, No Communication Overhead	Less Scalable
Min-Min Algorithm	Less Overhead, High Resource Utilization	More Migration time, Less Scalability
Max-Min Algorithm	High Throughput, Minimum Response Time	Less Scalability, Increased Migration time
Active Clustering	Less Overhead, Minimum Migration time	Less Throughput, Less Efficient
Ant Colony Optimization	Performance increased, High Resource utilization and Fault tolerance	Complex Network
Throttled Load Balancing Algorithm	High Load Movement Factor	High Communication Cost, High Network Delay
Two Level Task Scheduling Algorithm	Maximize Response Utilization, Increased Performance	Increased Response time
Honey Bee Foraging Algorithm	Maximize throughput, Low overheads	Low Priority Load Continuously on the Queue

IV. CONCLUSION

This paper reviews various existing load balancing methods for distributed file systems in cloud. Each method has its own merits and demerits. Several limitations are faced by most of the previously described algorithms. By achieving effective load balancing, system attains high resource utilization and user satisfaction. So there is a need of effective load balancing algorithm which ensures shorter access time and more accuracy. According to this study, a distributed load balancing and re-balancing algorithm needed, which provides high performance and less failure risk. It will also be focus on reduced data transfer and the network bandwidth usage.

REFERENCES

[1] Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh Yi Chung, HaiyingShen, Member, IEEE, and Yu-Chang Chao, “Load Rebalancing for Distributed File Systems in Clouds”, IEEE Trans. Parallel and Distributed Systems, vol. 24, no.5, May. 2013.

[2] Randles, M., D. Lamb and A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.

[3] Dhinesh Babu L.D, P. VenkataKrishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Applied Soft Computing 13 (2013) 2292–2303.

[4] Nitika, Shaveta and Gaurav Raj, “Comparative analysis of Load Balancing Algorithms in Cloud Computing”, International Journal of Advanced Research in Computer Engineering and Technology, vol.1, Issue 3, pp. 120-124, May 2012.

[5] Liu H., Liu S., Meng X., Yang C. and Zhang Y.,International Conference on Service Sciences (ICSS),257-262, 2010.

[6] Sadhasiva,DR.S.Jayarani,”Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment.” International Conference on Advances in Recent Technologies in Communication and Computing, vol. 148, pp. 884–886 (2009)

[7] Zenon Chaczko, Venkatesh Mahadevan, ShahrzadAslanzadeh,ChristopherMcdermid, “Availabity and Load Balancing in Cloud Computing”International Conference on Computer and Software Modeling, IPCSIT, volume14, IACSIT Press, Singapore 2011

[8] Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast" in proc. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.

[9] Buyya R, R. Ranjan, RN. Calheiros, “InterCloud: Utilityoriented federation of cloud computing environments for scaling of application services”, International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Busan, South Korea, 2010.

[10] Stanojevic R. and Shorten R., IEEE ICC, 1-6, 2009.

[11]Vouk, "Cloud Computing- Issues, Research and Implementations," Information Technology Interfaces, pp. 31-40, June 2008.

[12] O. Abu- Rahmeh, P. Johnson and A. Taleb-Bendiab,“A Dynamic Biased Random Sampling Scheme orScalable and Reliable Grid Networks”, INFOCOMP Journal of Computer Science, ISSN 1807-4545,VOL.7, December2008

[13] http://computer.howstuffworks.com/cloud-computing/cloud-computing.htm

AUTHORS PROFILE

Martina Poulose has completed B Tech in CS from Sahrdaya College of Engineering and Technology, Thrissur, Kerala, in 2008. Presently pursuing M.Tech in CSE from Met’s School of engineering, Thrissur, Kerala.

Dr. M. Azath is Head of Department of Computer Science and engineering, Met’s School Of Engineering, Mala. He has received Ph.D. in Computer Science and Engineering from Anna University in 2011. He is a member in Editorial board of various international and national journals and also a member of the Computer society of India, Salem. His research interests include Networking, Wireless networks, Mobile Computing and Network Security.