# A Review on Load Balancing Algorithm in Cloud Computing

Komal Purba[1], Nitin Bhagat[2]
[1]*(Department of CSE, SIET Manawala, India)*
[2]*(Department of CSE, SIET Manawala, India)*

***Abstract:****Cloud computing represents different ways to design and manage remotely computing devices. Company using customized Design of cloud computing to public network allows users to establish an account. The security requirements in cloud computing environment is to find the Security threats in the structure of clouds to find the security solutions, and finding reasons so that pre security steps should be taken in concerned with security proposed model. This paper represents how to build a trusted computing environment for cloud computing system by combining the trusted computing platform into cloud computing system which is free from vulnerabilities and threats. Various techniques are discussed to design the cloud computing system model. The system is designed with trusted computing platform and trusted platform models.*

***Keywords-****Cloud computing, load balance, Algorithm, software, Attack,High Jack, Denial of Services(DOS)*

## I. INTRODUCTION

Cloud computing is e-technology, where virtual resources are provided as services over the Internet. Users need not have gain the technology knowledge or control over the cloud that support the resources [12]. Cloud Computing is high utility software having the ability to change the software industry and making thesoftware even more attractive. Cloud computing has strong impact on software industry and their business. The elasticity of resources is unprecedented in the history of information technology. The increase in webtraffic and different services are increasing day by day making load balancing a big research topic. Cloud computing is anew technology which uses implicit machine instead of physical machine to host, store and network the differentcomponents. Cloud computing are highly automated to manage servers launching and shutting, load balancing, failure detection and handling, etc [11]. Load balancing is a technique used to distribute the load uniformly among all the network nodes. With maximum throughput and response time, and avoiding the overload among the network nodes, optimum utilization of resources is done by static or dynamic load balancing [13]. The load balancing needs to be done properly because failure in any one of the node can leadto unavailability of data.

Load balancing is a generic term used for distributing a larger processing load to smaller processing nodes for enhancing the overall performance of system [1, 15]. An ideal load balancing algorithm should avoid overloading or under loading on specific node. But, in case of a cloud computing environment the selection of load balancing algorithm is not easy because it involves additional constraints like security, reliability, throughput etc. So, the main goal of a load balancing algorithm is to improve the response time of job by distributing the total load of system. The algorithm must also ensure that it is not overloading any specific node. Load balancers can work in two ways: cooperative and non-cooperative. In cooperative, the nodes work simultaneously in order to achieve the common goal of optimizing the overall response time. In non-cooperative mode, the tasks run independently in order to improve the response time of local tasks.

A static load balancing algorithm does not take into account the previous state or behavior of a node while distributing the load. On the other hand, a dynamic load balancing algorithm checks the previous state of a node while distributing the load. The dynamic load balancing algorithm is applied either as a distributed or non-distributed. The advantage of using dynamic load balancing is that if any node fails, it will not halt the system; it will only affect the system performance. In a dynamic load balanced system, the nodes can interact with each other generating more messages when compared to a non-distributed environment. However, selecting an appropriate server needs real time communication with the other nodes of the network and hence, generates more number of messages in the network.

Load balancer uses different technical methods for keeping track of updated information.

## II. STATIC LOAD BALANCING

1. Round-Robin Load Balancer

It is a static load balancing algorithm, which does not take into account the previous load state of a node at the time of allocating jobs. It uses the round robin scheduling algorithm for allocating jobs. It selects the nodes randomly and then allocates jobs to all nodes in a round robin manner. This algorithm will not be suitable for cloud computing because running time of any process is not known prior to execution; there is a possibility that nodes may get heavily loaded. Hence, weighted round-robin algorithm was proposed to solve this problem [3, 14]. In this algorithm, each node is assigned a specific weight. Depending on the weight the node would receive appropriate number of requests. The weight assign to the nodes is helpful to control the network traffic.

## III. MIN-MIN

It is a static load balancing algorithm. So, all the information related to the job is available in advance. Some terminology related to static load balancing:

1. Expected Time to Compute (ETC)The expected running time of the jobs on all nodes is stored in an ETC matrix. If a job is not executable on a particular node, the entry in the ETC matrix is set to minimum Execution Time algorithm. In this, each job is assigned to the node which has the smallestexecution time as mentioned in ETC table, regardless of the current load on that processor. Minimum Execution Time tries to find thebest job-processor pair, but it does not take into consideration the current load on the node. This algorithmimproves the make-span to some extent, but it causes a severe load imbalance.

2. Minimum Completion Time (MCT)MCT algorithm assigns jobs to the node based on their minimum completion time. The completion time is calculated by adding the expected execution time of a job on that node with node's ready time. First of all, minimum completion time for all jobs is calculated. The job with minimum completion time is selected. Then, the node which has the minimum completion time for all jobs is selected. Finally, the selected node and the selected job are mapped. The ready time of the node is updated. This process is repeated until all the unassigned jobs are assigned [4, 5].

## IV. MAX-MIN

Max-Min [4,5, 16] is almost same as the min-min algorithm except the following: after finding out minimum completion time of jobs, the maximum value is selected. The machine that has the minimum completion time for all the jobs is selected. Finally the selected node and the selected job are mapped. Then the ready time of the node is updated by adding the execution time of the assigned task.

## V. Load Balance Min-Min (LBMM)

LBMM [5,6] is a static load balancing algorithm. This algorithm implements load balancing among nodes by considering it as a scheduling problem. The Min –Min algorithm first finds the minimum execution time of all tasks. Then it chooses the task with the least execution time among all the tasks [17]. The algorithm proceeds by assigning the task to the resource that produces the minimum completion time. The same procedure is repeated by Min-Min until all tasks are scheduled.

The algorithm executes the Min-Min schedule and selects the node with the highest make-span value. Corresponding to that node, selects the job with minimum execution time. The completion time of the selected job is calculated for all the resources. Maximum completion time of the selected job is compared with the make-span value. If it is less, then selected job is allocated to the node, which has the maximum completion time. Else, the next maximum completion time of the job is selected and the steps are repeated. The process stops if all the nodes and all the jobs are assigned. In scenarios where the number of small tasks is more than the number of large tasks in meta-task, this algorithm achieves better performance. This algorithm does not consider low and high machine heterogeneity and task.

## VI. Load Balance Max-Min-Max

LBMMM proposed a two-phase scheduling algorithm that combines OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms [7]. OLB scheduling algorithm keeps every node in working state to achieve the goal of load balancing and LBMM scheduling algorithm is utilized to minimize the execution time of each task on the node, thereby minimizing the overall completion time. This combined approach helps in efficient utilization of resources. The algorithm performs average completion time of each task for all the nodes. Select the task with maximum average completion time. Select an unassigned node with minimum completion time that should be less than the maximum average completion time for the selected task. Then, the tasks dispatched to the selected node for computation.  If all the nodes are already assigned, re-evaluate by considering both assigned and unassigned nodes. Minimum completion time is computed as. Minimum completion time of the assigned node is the sum of the minimum completion time for all the tasks assigned to that node and minimum completion time of the current task.  Minimum completion time of the assigned node is the minimum completion time of the current task. Repeat again, until all tasks are executed. This gives better results than the above discussed algorithms.

## VII. DYNAMIC LOAD BALANCERS

1. Equally Spread Current Execution

Equally Spread current execution is a dynamic load balancing algorithm, which handles the process with priority [2]. It determines the priority by checking the size of the process. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a Virtual Machine, which is lightly loaded. The load balancer spreads the load on to different nodes, and hence, it is known as spread spectrum technique.

2. Throttled Load Balancer

Throttled load balancer is a dynamic load balancing algorithm. In this algorithm, the client first requests the load balancer to find a suitable Virtual machine to perform the required operation [2].

In Cloud computing, there may be multiple instances of virtual machine. These virtual machines can be grouped based on the type of requests they can handle. Whenever a client sends a request, the load balancer will first look for that group, which can handle this request and allocate the process to the lightly loaded instance of that group.

3. Honeybee Foraging Algorithm

Honeybee Foraging algorithm first goes outside of the honey comb and find the honey sources. After finding the source, they return to the honey comb and do a waggle dance indicating the quality and quantity of honey available [8]. Then, reapers go outside and reap the honey from those sources. After collecting, they return to beehive and does a waggle dance. This dance indicates how much food is left. M. Randles propose decentralized honeybee based algorithm for self-organization. In this case, the servers are grouped as virtual server and each virtual server have a process queue. Each server, after processing a request from its queue, calculates the profit which is analogous to the quality.

4. Biased Random Sampling

Biased Random Sampling is a dynamic load balancing algorithm uses random sampling of system domain to achieve self-organization thus, balancing the load across all nodes of system [9]. In this algorithm, a virtual graph is constructed with the connectivity of each node representing the load on server. Each node is represented as a vertex in a directed graph and each in-degree represents free resources of that node. Whenever a client sends a request to the load balancer, the load balancer allocates the job to the node which has atleast one in-degree. Once a job is allocated to the node, the in-degree of that node is decremented by one. After the job is completed, the node creates an incoming edge and increments the in-degree by one. The addition and deletion of processes is done by the process of random sampling. Each process is characterized by a parameter know as threshold value, which indicates the maximum walk length. A walk is defined as the traversal from one node to another until the destination is found. At each step on the walk, the neighbor node of current node is selected as the next node. In this algorithm, upon receiving the request by the load balancer, it would select a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Else, the walk length of the job is incremented and another neighbor node is selected randomly. The performance is degraded as the number of servers increase due to additional overhead for computing the walk length.

5. Active Clustering

Active Clustering is a clustering based algorithm which introduces the concept of clustering in cloud computing. In cloud computing there are many load balancing algorithms available. The performance of an algorithm can be enhanced by making a cluster of nodes [9, 10]. Each cluster can be assumed as a group. The principle behind active clustering is to group similar nodes together and then work on these groups. The process of creating a cluster revolves around the concept of match maker node. In this process, first node selects a neighbor node called the matchmaker node which is of a different type. This matchmaker node makes connection with its neighbor which is of same type as the initial node. Finally the matchmaker node gets detached. This process is followed iteratively. The performance of the system is enhanced with high availability of resources, thereby increasing the throughput. This increase in throughput is due to the efficient utilization of resources.

6. Join-Idle Queue

Join-Idle Queue uses distributed dispatchers by first load balancing the idle processors across dispatchers and then assigning jobs to processors to reduce average queue length at each processor [9]. The disadvantage of this algorithm is that it is not scalable. Y. Lua et al.[3] proposed this load balancing algorithm for dynamically scalable web services. It effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time. It can perform close to optimal when used for web services.

## VIII. CONCLUSIONS AND FUTURE WORK

Any individual cannot fulfill the requirements of all the resources. Remote resources are utilized to run the operation. Cloud computing technique helps the flow of data and operations among remote devices. Cloud computing provides a platform that act as a service, an application, a service, an infrastructure. Uncontrolled traffic cause load balancing. Overloading of a system may lead to poor performance which can make the technology unsuccessful. So there is always a requirement of efficient load balancing algorithm for efficient utilization of resources. Our paper focuses on the various load balancing algorithms and their applicability in cloud computing environment. We first categorized the algorithms as static and dynamic. Then we analyzed the various algorithms which can be applied in static environments. After that we described the various dynamic load balancing algorithms. For solving any particular problem some special conditions need to be applied. So we have discussed some additional algorithms which can help in solving some sub-problems in load balancing which are applicable to cloud computing. In our future work we will analyze the algorithms with numerical analysis and simulation.

## REFERENCES

[1] N. Ajith Singh, M. Hemalatha, An approach on semi distributed load balancing algorithm for cloud computing systems, International Journal of Computer Applications Vol-56 No.12 2012.

[2] Nitika, Shaveta, Gaurav Raj, International Journal of advanced research in computer engineering and technology, Vol-1 issue-3 May-2012.

[3] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, IPCSIVol-14, IACSIT Press Singapore 2011.

[4] T. Kokilavani, Dr. D. I. George Amalarethinam , Load Balanced Min-Min Algorithm for Static Meta Task Scheduling in Grid computing, International Journal of Computer Application Vol-20 No.2, 2011.

[5] Graham Ritchie, John Levine, A fast effective local search for scheduling independent jobs in heterogeneous computing environments, Center for Intelligent Systems and their applications School of Informatics University of Edinburg.

[6] Shu Ching Wang, Kuo-Qin Yan Wen-pin Liao, and Shun Sheng Wang Chaoyang University of Technology, Taiwan R.O.C.

[7] Che-Lun Hung, Hsiao-hsi Wang, and Yu- ChenHu, Efficient Load balancing Algorithm for cloud computing network.

[8] Yatendra sahu, M. K. Pateriya, Cloud Computing Overview and load balancing algorithms, Internal Journal of Computer Application Vol-65 No.24, 2013.

[9] Nayandeep Sran, Navdeep kaur , Comparative Analysis of Existing Load balancing techniques in cloud computing, International Journal of Engineering Science Invention, Vol-2 Issue-1 2013.

[10] Nidhi Jain Kansal, Inderveer Chana , Existing Load balancing Techniques in cloud computing: A systematic review, Journal of Information system and communication Vol-3 Issue-1 2012

[11] Ken Birman, Gregory Chockler, R V Renesse, Towards a cloud computing research agenda, ACM SIGACT News Distributed Computing Column. Volume 40, Issue 2, pp 68-80. June 2009.

[12] Krzysztof Ostrowski, Ken Birman, Storing and Accessing Live Mashup Contents in the Cloud, Krzysztof Ostrowski and Ken Birman. 3rd ACM SIGOPS International Workshop on Large Scale Distributed Systems and Middleware (LADIS 2009). Volume 44, Issue 2, April 2010.

[13] Ruhi Gupta, Review on Existing Load Balancing Techniques of Cloud Computing, International Journal of Advanced Research in Computer Science and Software Engg. Vol.4 Iss. 2, Feb. 2014. Pp: 168-171

[14] Tushar Desai, Jignesh Prajapati, A Survey of Various Load Balancing Techniques and Challenges in Cloud Computing, International Journal of Scientific & Technology Research. Vol.2 Iss.11. Nov.2013 PP:158-161.

[15] N.S. Raghava and Deepti Singh, Comparative study on Load balancing techniques in cloud computing, Open Journal of Mobile Computing and Cloud Computing. Vol. 1, Num. 1 . August 2014. PP 18-25

[16] O M Elzeki, M Z Reshad, Improved Max-Min Algorithm in Cloud Computing, International Journal of Computer Applications. Vol. 50, Iss. 12. July 2012. PP: 22-27.

[17] T. Kokilavani, D I George, Load Balanced Min-Min Algorithm for static meta task scheduling in Grid Computing, International Journal of Computer Applications. Vol. 20, Iss. 2. April. 2011. PP 43-49

**Komal Purba**

She obtained her B.Tech (Information Technology) from College of Engineering and Management, Kapurthala, Punjab, India, pursuing M.Tech (Computer science & engineering) from Sai Institute of Engineering and Technology, Manawala, Amritsar, Punjab, India. Her area of interest is Cloud Computing and Load balancing in cloud computing.

**Nitin bhagat is** working as an assist. professor in Department of Computer Science & Engineering, Sai Institute of Engineering and Technology, Manawala, Amritsar, Punjab, India. He obtained his B.Tech (computer science engineering) from Guru Nanak Dev University, Punjab, India, M.Tech (computer science & engineering from Guru Nanak Dev University, Punjab, India.