IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 55-58

Survey on Load Balancing Algorithms

Darryl G.S. Pereira[#1], Louella Mesquita Colaco[#2], Ravina Rodrigues[#3]

[#]*Department of Information Technology,Padre Conceicao College of Engineering, Goa University, Goa,India*
darrylgreg.2@gmail.com
lmesquita@rediffmail.com
ravinarodrigues@gmail.com

Abstract: *Load Balancing is a common feature used in distributed systems to distribute workload on multiple servers. A wide range of load balancing algorithms have been used to implement this concept. There exists algorithms like Round Robin, Min-Min etc which have been tried and tested. This paper aims to summarize the pros and cons of these commonly used algorithms.*
Keywords: *Load Balancing, Cloud Computing, Distributed Systems, Genetic Algorithm, Comparison*

I. Introduction

Speeds on servers when accessed by a huge amount of users,tend to drastically decrease as they cannot handle the load alone. As a result,the users in turn face slow download and uploads speeds while accessing a particular server. The solution to this problem is to use efficient load balancing algorithms which are able to take care of deadlocks and starvation while allowing flexibility in terms of handling the load.

Load Balancing is defined as the process of distributing workload on multiple servers in a network. The main objectives are increase in performance, stability and to provide a back-up plan when the system fails[14].

Load balancing is an important issue in a network of workstations. An important aspect of load balancing is deciding on the appropriate algorithm to use. Various algorithms have been implemented and tested. Each one has its own specific motivations and design issues. Generalizing these algorithms does not always prove to be efficient for all loads. Therefore it is always best to strategically choose an algorithm based on the amount of load on the nodes, how much load each server can take, network design distribution and various other factors of the network.

The performance of load balancing can be evaluated using the following metrics [14,15]:

- *Makespan: D*efined as the total amount of time taken for executing all the tasks.
- *Throughput:* Deals with the maximum number of tasks being completed within a minimum completion time.
- *Response Time:* The minimum time taken for the system to respond to the load balancing algorithm.
- *Performance:* Refers to the effectiveness of the system after performing a load balancing algorithm.
- *Scalability:* Performing load balancing with a minimum number of servers.

Load balancing algorithms can be categorized as Static load balancing and Dynamic load balancing.

A. Static load balancing:

These algorithms allocate tasks to workstations prior based on the load at the time the nodes are allocated to some tasks based on average load. Advantage of these algorithm is that there is no need to constantly monitor the workstation.These algorithms work well only when there is not much variation in the load.

B. Dynamic load balancing:

These algorithms are dynamic in nature, i.e, they make changes in the load among the workstations at runtime. They are able to do this by using current or recent load information in making decisions .Advantage of these algorithms is that it can handle variations in the load, i.e. once load imbalance reaches a predefined level,redistribution of tasks takes place.These algorithms also monitor the load on all processors at all times [13].

Some of the commonly used load balancing algorithms are:

1. *Round Robin load balancing algorithm*:

Round Robin load balancing algorithm is a static load balancing algorithm which first sends a client request across a pool of servers.Further, it forwards a client request to each server in turn. When it reaches the last node it looks back and repeats the process again. This algorithm is one of the most standard and easy to use. However

it is not always accurate and efficient in the distribution of traffic because it usually assumes that all the servers are the same.

2. Min-Min load balancing algorithm :

This algorithm is a static load balancing algorithm whichtakes the minimum tasks fist and allocates them to the required resources and then proceeds to allocate the larger tasks. This reduces the chances of starvation of the smaller tasks and increases the time at which all the tasks are completed.The disadvantage of this algorithm is that if there are too many small tasks then there is a possibility that the larger tasks might take a long time to be allocated to a resource.

3. Load balancing min-min(LBMM):

In this algorithm, first the min-min algorithm is performed and then the LBMM selects high load resources and relocates low load resources. This solves the problem of makespan and utilization of resources. LBMM is a static load balancing algorithm.

4. Genetic load balancing algorithm:

Genetic algorithm is an optimization algorithm which makes use of a search based on natural selection and genetics.This procedure is carried out with randomly generated chromosomes with which survival of fittest test is carried out to achieve new solutions. Genetic algorithm is a dynamic load balancing algorithm

In this work, we compare some of the existing load balancing algorithms in literature. Section II provides a detailed survey of the various algorithms, Section III provides a table comparing the algorithms reviewed. And Section IV concludes the review.

II. Literature Review

Mahitaet al.[1], proposed that in the field of cloud computing,resource management remains a big concern. Inefficient resource management can lead to bigger issues like deadlock and hence result in a disaster. In cloud computing disaster management is one of the features that are usually not taken care off ,though there are very few cases of such problems occurring ,there is always a possibility of it occurring and hence for such occurrences certain protocols have to be put in place .In a recent research ,it stated that there is low business performance caused due to the lack of disaster management .this paper deals with an efficient load balancing technique to control disaster by using a throttled algorithm for load balancing where a comparison of system performance is carried out with and without the load balancer.Throttled algorithm is implemented using a cloud analyst tool. The simulation resulted in better resource sharing with minimum overall response time and better throughput.

Kaur et al.[2], implemented Round Robin load balancing strategy in Software Defined Networks. The implementation was carried out using softwares like VMPlayer, Mininet, POX Controller, and OpenLoad. The request to the server was redirected by the load balancer based on Round Robin load balancing strategy as well as random load balancing strategy. The results were compared based on total transactions per second and average response time of the web server. Round Robin strategy was found to be better in terms of the transactions per second and average response time. Round Robin strategy also distributed the load evenly.

Kabir et al,[3], proposed a simplified genetic algorithm for carrying out load balancing in cloud computing. First a population is initialized followed by calculation of a main value for each population. If iterations have reached the maximum threshold then delete chromosomes having maximum fitness value and choose chromosomes having the lowest fitness value. Perform crossover and mutation to obtain a new offspring in the new population. The process ends by testing for the end condition. This algorithm was found to perform better load balancing in cloud computing.

Suntharam[4], proposed a Max-Min load balancing algorithm for a private cloud environment. The algorithm consists of five modules namely Storage Node Sub Module,Path Transmission Sub Module,Virtual Machine Sub Module,Job Provisioning Sub Module and Load Balancing module. In max-min algorithm, the completion time of each machine is calculated and the machine having the largest completion time is chosen.Tasks are assigned to machines and reassigned to machines having completion time less than the mean value. Max-Min algorithm helps in faster storage of jobs in the nodes and reduces deadlocks in the private cloud network.

Rajput et al. [5], proposed a genetic based Improved Load Balanced Min-Min (ILBMM) algorithm.It begins by selecting million instructions (MI) of task and million instruction per second (MIPS) of virtual machine (VM). Crossover and mutation is performed followed by evaluation of the fitness of the new offspring. If the fitness is found to be greater than the maximum value(initial task value) then begin again by selecting MI of task and MIPS of VM else the task is assigned to the VM. The process terminates when all the tasks are assigned to the VMs. A simulation was conducted on Cloudsim between four cloudlets(tasks) and two VMs. This genetic based ILBMM proved to minimize the makespan.

Jiadong et al.[6], presented a strategy of load balancing for cloud environment where workload time

series are predicted based on the values of the latest workload sequences. Virtual machines are migrated if the workload is greater than the upper threshold or lower than the lower threshold. The virtual machine having a large workload and has not been migrated recently, is chosen. Finally, the destination host is chosen to be the machine which can handle the workload of the virtual machine. The result of this strategy showed a solution to the frequent migration problem.

Coronado et al. [7],proposed a Software Defined Network–based load–balancing algorithm called Wi-Balance which enabled efficient traffic load in WiFi networks and it aimed to reduce the collisions on the network. 5G–EmPOWER platform was used to deploy Wi-Balance.The Wi-Balance consists of three layers Clients, Radio Access Network and Controlled Network. It demonstrated an effective mechanism for balancing traffic load.

Table I: Comparison Between Different Load Balancing Algorithm

Algorithm	Dynamic	Response Time	Throughput	Scalability	Resource Utilization	Fault Tolerance
Max-Min Load Balancing Algorithm[4]	NO	YES	YES	NO	YES	NO
Min-Min Load Balancing Algorithm[12]	NO	YES	YES	NO	YES	NO
Improved Load Balancing Min Min[5]	NO	YES	YES	NO	YES	NO
Round Robin Load Balancing Algorithm[2]	NO	YES	YES	NO	YES	NO
Genetic Algorithm[3]	YES	YES	YES	YES	YES	YES
Swarm Optimization Based Load Balancing Algorithm[11]	NO	NO	YES	NO	YES	NO
Throttled Load Balancing Algorithm[1]	YES	YES	NO	YES	YES	YES
Bandwidth Based LaodBalancing[8]	NO	YES	NO	NO	YES	NO
Wi-Balance[7]	YES	YES	NO	NO	YES	NO

Hamed et al. [8], presented a Bandwidth based load balancing (BWBLB) approach using SDN. An OpenFlow Controller called “Ryu” was used. Open vSwitch is an OpenFlow switch used in this implementation. Round Robin and Connectios-based were the two load balancing algorithms implemented. The server was selected based on the least bandwidth consumption. A summarized result of the bandwidth consumption was obtained by using a software called “ bwm-ng”. Service’s average response time was calculated using the OpenLoad software tool. Raspberry Pi OpenFlow-enabled switch and Mininet were used as hosts. BWBLB approach was compared with Round Robin and connectionsbased load balancing algorithm using minimet and Raspberry Pi. BWBLB was found to perform better as compared to the other load balancing approaches.

Zakia et al. [9], proposed a load balancing algorithm for a Data Center Network (DCN) and aimed to improve traffic management in a network.Dijkstra's algorithm was used to find multiple paths. Path having the least cost and load was selected for traffic flow.Representable State Transfer Application Program Interface was used to implement the proposed approach. Traffic load is calculated before and after load balancing. Delay and packet loss is measured. This approach is found to have lower delays and a reduction in the round trips.

Prakash et al. [10], presented a dynamic server load balancing algorithm which had an architecture consisting of a controller, an OpenFlow switch, multiple servers and a client. The network was divided into numerous VLANs. Server load was computed by adding up the CPU usage rate,memory usage and connection count. Minimet was used as a topology and an OpenFlow controller called Floodlight was used. The proposed approach was compared using RoundRobin and Random algorithms. After evaluation it was found that the CPU usage of Round –Robin and Random algorithms.After evaluation it was found that the CPU usageof Round-Robin and Random algorithms were unequal. The dynamic server-based load balancing algorithm proved to be better in terms of CPU usage among servers.

Govindarajan et al.[11],proposed to introduce a particle swarm optimization based load balancing technique which evenly guided and distributed cloud consumer applications in an optimal balanced method. This study investigated the software defined networks cloud infrastructure which is said to be dynamic in nature,thus providing network paths on demand by deriving these decisions based on the current network load in the system. This system was simulated and tested on real world application traces. The advantages of using a particle swarm optimization based load balancing mechanism proved to minimize the average application response time, provided maximized throughput, and more efficient resource utilization. This type of method was solely proposed for a networking cloud infrastructure in software defined networks for more efficient task scheduling and optimal load balancing.

Sharma et al. [12], presented a comparative analysis between max-min and min-min algorithms based on makespan. In min-min algorithm, the completion time of eachtask is calculated and the task with the least completion timeis assigned to the resource. This is done till all the tasks are assigned to the resources. Whereas,in max-min, the task with the maximum completion time is selected and assigned to the resource which gives minimum completion time. The simulation results showed min-min algorithm to have a higher makespan than max-min algorithm when the numberof large sized tasks were more than small sized tasks.

III. Result

In the previous section different load balancing algorithms proposed in literature have been discussed. Table I gives a comparison between the different load balancing algorithms based on different parameters such dynamicity, response time, throughput, scalability, resource utilization, and fault tolerance.

IV. Conclusion

In this work, the various algorithms are surveyed and compared based on different parameters. From the above survey we conclude that a Genetic algorithm based on a load balancing Min-Min is efficient in terms of handling a huge variety of tasks and their servers along with a load balancing algorithm which can also prioritize certain tasks when needed as per productivity needs.

References

[1]. Mahitha.O,Suma. V,"Deadlock Avoidance through Efficient Load Balancing to Control Disaster in Cloud Environment",4th ICCCNT 2013

[2]. SukhveerKaur, Japinder Singh, Krishan Kumar, Navtej Singh Ghumman ,"Round-Robin Based Load Balancing in Software Defined Networking",2nd InternationalConference on Computing for SustainableGlobal Development(INDIACom),2015

[3]. Md. ShahjahanKabir, Kh. MohaimenulKabir , Dr. RabiulIslam,"Process of Load Balancing in Cloud Computing Using Genetic Algorithm", Electrical & Computer Engineering: An International Journal (ECIJ) Volume 4, Number 2, June 2015

[4]. S M S Suntharam," Load Balancing By Max-Min Algorithm in Private Cloud Environment", International Journal of Science and Research (IJSR),2015.

[5]. Shaym Singh Rajput and Virendra Singh Kushwah, "A Genetic based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing",8th International Conference on Computational Intelligence and Communication Networks, 2016

[6]. Zhang Jiadong, Liu Qiongxin, Chen Jiayu," An Advanced Load Balancing Strategy For Cloud Environment", 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies

[7]. Estefan´ıa Coronado, Jose´ Villalo´n, Antonio Garrido,"Wi–Balance: SDN–based load–balancing in Enterprise WLANs",978-1-5090-6008-5/17/ 2017 IEEE

[8]. Mohamed I. Hamed, Basem M. ElHalawany, Mostafa M.Fouda, Adly S. Tag Eldien,"A New Approach for Server-based Load Balancing Using Software-Defined Networking", The 8th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS 2017)

[9]. UmmeZakia and Hanene Ben Yedder,"Dynamic Load Balancing in SDN-Based Data Center Networks",978-1-5386-3371-7/17/$31.00 ©2017 IEEE

[10]. S.WilsonPrakash, Dr.P. Deepalakshmi ,"Server-based Dynamic Load Balancing",978-1-5090-6590-5/17/$31.00 ©2017 IEEE

[11]. KannanGovindarajan,Vivekanandan Suresh Kumar,"An Intelligent Load Balancer for Software Defined Networking (SDN) based Cloud Infrastructure",978-1-5090-3239-6/17/$31.00©2017IEEE

[12]. Neha Sharma, Dr. Sanjay Tyagi, Swati Atri," A Comparative Analysis of Min-Min and Max-Min Algorithms based on the Makespan Parameter", International Journal of Advanced Research in Computer Science,2017

[13]. Sandeep Singh Waraich ,"Classification of dynamic load balancing strategies in a Network of Workstations", Fifth International Conference on Information Technology: New Generations

[14]. Nitin Kumar Mishra, Nishchol Mishra," Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing-A Systematic Review", International Journal of Computer Applications (0975 – 8887) Volume 131 – No.18, December 2015

[15]. Jaimeel M Shah, DrSharnilPandya, Dr Narayan Joshi, DrKetanKotecha, DrD.B.Choksi," Load Balancing in cloud computing: Methodological Survey on different types of algorithm", International Conference on Trends in Electronics and Informatics ICEI 2017