A PUBLICATION OF MOTHERHOOD UNIVERSITY, ROORKEE

(Recognized by the UGC with the right to award degrees u/s 22(1) of the UGC act 1956
andestablished under Uttarakhand Government Act No. 05 of 2015)

Motherhood International Journal of Multidisciplinary
Research & Development

A Peer Reviewed Refereed International Research Journal

Volume II, Issue I, July 2017, pp. 19-27

ONLINE ISSN-2456-2831

ONLINE ISSN-2456-2831

Motherhood International Journal of Multidisciplinary
Research & Development

Volume I Issue I August 2016

A Peer reviewed Refereed Quarterly International
Research Journal

A Publication of
Motherhood University, Roorkee
Village Karoundi, Post Bhagwanpur
Tehsil Roorkee, Dist. Haridwar
Uttarakhand

Editor-in-Chief
Dr. S. B. Sharma

Load Balancing Techniques, Challenges & Performance Metrics

Dr. Atul Garg & Shilpa Dang
MMICT&BM, Maharishi Markandeshwar University
Mullana Maharishi Markandeshwar University, Mullana
Ambala, Haryana, India

Abstract

The growth in the popularity of the web has increased. There is need for high availability computer systems which are capable of processing the client request as quickly as possible. As a result the use of distributed systems is growing rapidly, so the area of load balancing has gained higher attention and importance. Load balancing is crucial for managing various operations proficiently in distributed atmosphere. Load balancing is an activity that circulates the workload equitably over every node of the system. It accomplishes higher user fulfillment and resource usage, henceforth enhancing the general execution and adaptability of the system. Many algorithms were suggested that provide efficient mechanism to execute client request at minimal cost. The inspiration of the work is to urge the researchers in developing more efficient load balancing algorithms.

KEYWORDS: *Distributed Systems, Load Balancing, Load Index, Performance Matrices.*

Introduction

In the modern era, to manage and for security purposes it is the better way to distribute the information. Users send their request to host computer for processing. The arbitrary entry of requests can cause some computers to be intensely loaded while others are idle or just daintily loaded. Load balancing improves performance by transferring requests from heavily loaded to lightly loaded nodes. The fundamental objective of load balancing is to optimize the average response time and provide better service to the users.

Load balancing is described as a technique of dividing and circulating tasks to all nodes of the system so that more jobs can be served and the system can perform efficiently [13]. It anticipates circumstance of bottleneck in system which can happen because of load irregularity. When, any node stops working unexpectedly then to continue the service load balancing can be used by implementing failover feature. It also ensures that all resources are distributed efficiently and fairly [6]. In the load balancing process the important elements are load measurement and balancing mechanism [17]. Generally, load balancing is enforced in two ways: Static and Dynamic. The Static load balancing is applied on the fixed type of situation whereas in Dynamic load balancing can be applied according to the new requirements of the server or resources.

Load balancing is implemented in various fields like telecommunications, cloud computing, routing, mobile networks, grid computing etc. [9, 18, 22, 27]. This paper defines basic load balancing mechanism, performances metrics used in load balancing, major issues and challenges to overcome for designing an efficient algorithm.

Literature Review

The research work done by few renowned researchers is discussed in this section.

Luis & Azer [29] proposed a new technique to redirect TCP connections named as Distributed Packet Rewriting (DPR). In their research work, each system node keep the record of the nearest or related system nodes. Load information is maintained using periodic multicast amongst the cluster nodes.

Authors in [3] presented a novel method using connection mechanism. A new connection is established each time a request arrives. The number of connections increases on arrival of sudden load on the network and decreases when connections time outs.

Authors in [7] compared various evolutionary methods for query optimization on system nodes. Evolutionary methods are compared on the basis of their flexibility, transmission methods and behavior etc. Their results showed that PSO method is better as compared to other algorithms but it lack behind in processing time.

Researchers in [12] collectively discussed the “Dynamic File Migration” (DFM) method based on distributed architecture. In large file system there were various problems like dynamic file transfer and centralized algorithm system. The proposed Self Acting Load Balancing (SALB) algorithm reduces the problems to an extent. In the parallel file system the data are transferred between the memory and the storage devices continuously. For this purpose data management of a file system should be dynamic in nature. The various challenges in the parallel file system are scalability, the availability of the system, network transmission and the load migration.

Team of authors in [15] proposed an algorithm named as- honey bee load balancing algorithm. The designed algorithm is implemented in distributed cloud computing environment. For designing of algorithm the foraging behavior of bees is taken into consideration. The bee hives there are two types of bees- scout bee and forger bee. Scout bees are responsible for finding the food sources. Once they find the source, they return back to bee hive and inform others about the source by performing the dance known as tremble dance. Afterwards both scout and forger bees together go to collect the food. It is assumed that tasks arriving at VMs act as honey bee. All VMs are assembled in ascending order and execute the requests of priority basis. After receiving a new request VM update its hash table of load information and priority table. Whenever the task is transferred to VM, firstly it checks whether it will execute the task with priority or not. If the selected node executes the request is balanced.

Researchers in [10] proposed a framework to balance the dynamic load at server. In their framework they proposed two Ants one at the client end for request and other at the server end for the response. The Ant at the server end can choose the different path for its destination. The authors [39] improved the server performance in their work.

In [11] the researcher discussed VM load balancer algorithm to find the suitable VM in a short time period. Maximum length of VM is considered for the allocation of new request. A new VM is added to

Maintain the proper length of the VM. After that all the VM's load needs to be counted and least loaded VM would be selected to handle the new request.

The authors in [23] presented an Equally Spread Current Execution Algorithm. The algorithm is implemented in cloud environment. For all incoming request the cloud manger module make estimation for their execution time by measuring the size of job. It then, checks the execution capacity of node. If the job size and resources available are same, the job scheduler immediately allocates the required resource to the job and node starts its execution.

Valeria et al., [5] discussed the various approaches used for sending the client requests to distributed web servers. Different methods of load balancing like Client-Based, DNS Based, Dispatcher Based and Server Based are evaluated on the basis of compatibility with web servers and geographical scalability. It concluded that LAN distributed web servers can be considered as one the solutions for handling the rising demand of clients.

Young & Rasul [6] carried out the experimental and comparative analysis of load balancing algorithms on two platforms- hardware test bed (consisting of 16 PC's) and discreet event simulator. France football world cup (1998) traces and traces of workload generator SURGE are used as input values in simulator. Three scheduling algorithms: round robin, least connected and least loaded are compared on the basis of average response time and average web server utilization. The paper concluded that least load performs best but is difficult to implement. Least connection performs well for medium workload and is easy to implement. The performance of round robin is worse as compared to other two algorithms.

Daniel & Anthony [8] studied the problems present in protocols of resource allocation which are managed by selfish agents or organizations. These agents sometimes manipulate the resources for their own benefit. This results in poor efficiency and performance of system. A mechanism is designed to solve the problem of static load balancing in distributing systems. The optimal allocation algorithm is proposed and also a protocol is designed which implements this algorithm. Experimental results show that the derived algorithm gives satisfactory outcome. Implementing the proposed work

in real distributed systems and designing the distributed load balancing algorithm is kept as future work.

Belabbas et al., [9] presented a paper on survey of load balancing algorithms in grid computing. The paper describes the load balancing process in detail along with the challenges of grid computing. The grid computing is compared with traditional distributing systems and the conclusion is drawn that the traditional load balancing algorithms are not capable of handling load in grid computing due to its heterogeneous and dynamic nature.

Beniwal & Garg [16] proposed a semi-distributed system to balance the overloaded servers. Klaithem et al., [20] presented a survey paper on load balancing algorithms in cloud computing. Advantages and disadvantages of different algorithms are discussed. The comparison is done using the parameters- speed, network overhead, implementation complexity, fault tolerance.

Sweekiriti & Sudheer [30] proposed a novel algorithm for cloud data center. The algorithm is named as “Modified Central Load Balancer” which sent the client request to virtual machines on the basis of priority and state of machine. CPU speed and memory are used to prioritize the virtual machines. Proposed algorithm is also compared with Round Robin Algorithm, Throttled Algorithm and Equally Spread Current Execution algorithm. The results show that the proposed algorithm improves the resource utilization and speed of the system.

Load Balancing

Load balancing is a method of reassigning the entire load to the individual nodes to enhance resource utilization and response time, whereas conjointly avoiding a state where number of nodes are densely loaded and others are doing little or no work [16]. Load balancing assure that every node in system perform equal quantity of work [22]. A load balancer intercepts the web traffic sent by client by dividing the traffic into individual requests and decides which node should receive and process the requests. It maintains a log of the available nodes, ensuring that they are responding to traffic. If they are not, they are taken out of rotation. It also deploys separate units for failovers [13].

By [13] following is the process for basic load balancing transaction and the process is described in Figure 1:

- The client/user sends his/her request to the nearest server.

- Then the server connects to the next servers which become load balancer. The load balancer then decides the destination server and forwards the request to that server.
- The selected server accepts the request, process the request and send the response back.
- The load balancer did its processing and responds to the client.
- The client receive the response without any information of the server.

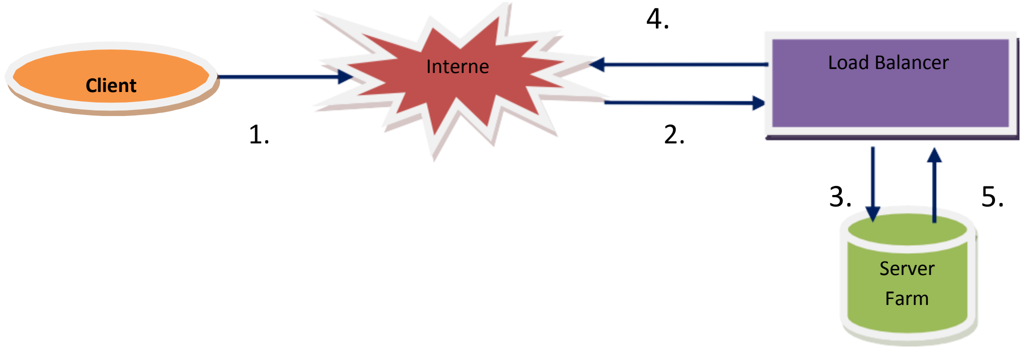

Figure 1: Load Balancing Transaction

A leading matter in modeling of load balancing algorithms is determining a suitable load index. A load Index anticipates the performance of a task. Various load index parameters that are studied and used are: length of the CPU queue, the amount of memory used, the context-switch rate and CPU utilization [1]. It is very important that mechanism used should be efficient and impose minimal overhead. Major goals of load balancing are [13]:

- To achieve overall system improvement at reasonable cost.
- To treat all jobs equally.
- To be fault tolerance so that under sudden partial failure of system, performance does not degrade.
- To have capacity to alter itself as per adjustments in system setup.
- To maintain system stability in emergency situations.

Types of Load Balancing

Load balancing is of two types – Static and Dynamic. Static load balancing is needed when there is no variation in load. In static method all the information related to the system and its related system must be available in advance. The system nodes are selected and rejected on the basis of their execution performance. The load balancer assigns the work to system nodes on the basis of their performance [5]. The load is assigned to the system nodes in advance and it is non-preemptive. Least communication among system nodes is required and this method reduces the execution time [17]. The main negative aspect of this method is that it allocates and fix the job work in advance and it does not did any change in later stages. In the case of load fluctuation or server problem it affects the overall performance of the system [27].

On the other hand, dynamic load balancing performs load distribution at run time. This method invigilates the load and performance of the node system and then it distribute the work. The main components are "Transfer Strategy, Location Strategy and Information Strategy". Transfer strategy decides which request should be migrated to other node from its current node. It decides whether a job is eligible for transfer or not. Location strategy decides which node should act as a destination node. It decides which node should start the transfer strategy. Information strategy on the other hand provides necessary information about nodes and system to the transfer and location strategy [28]. The goal of dynamic load balancing can be achieved in three forms: Centralized, Completely- Distributed and Semi-Distributed. Table 1 shows the main features and drawbacks of all three forms.

Table 1: Forms of Dynamic Load Balancing

PARAMETERS	CENTRALIZED	COMPLETELY-DISTRIBUTED	SEMI- DISTRIBUTED
WORKING	Algorithm is carried out by single central node. All other nodes are connected with this central node only.	Algorithm is carried out by all nodes whether in co-operative form or non co-operative form.	Nodes are divided in clusters. Each cluster has a central node interacting with its members and other central nodes.
ENVIRONEMENT	Best suited in environment where numbers of nodes in the system are lesser.	Best suited in environment where each node is given chance to act alone and lesser interaction with others.	Best suited in environment where there is large number of nodes in the system.
COMMUNICATION	Requires lesser Communication between nodes of the system.	Requires very much communication between nodes of the system.	Communication is more than centralized but lesser than completely distributed.
SYSTEM FAILURE	In case central nodes crashes whole system shuts down completely.	There is no central node so failure of one node does not shut down system.	Failure of one central node does not affect the whole system.
BOTTLENECK	Bottleneck arises mostly.	No such condition arises.	No such condition arises.

Performance Matrices

Performance matrices are the various features of which differentiate one load balancing algorithm form the other one. Also it checks the performance and effectiveness of the load balancing algorithm once it implemented. Various parameters that act as performance matrices are [2, 14, 18]:

- **Overhead Associated –** Decides the measure of overhead included while executing an algorithm. This ought to be minimized so that system can work proficiently.

- **Response Time** - is the measure of time taken to react by an algorithm. It ought to be minimized for stable working of system.
- **Fault Tolerance** - is the capacity of an algorithm to function consistently regardless of any unusual failure in system.
- **Throughput -** is the measure for computing the number of jobs which has been finished their execution. It ought to be high to increase the proficiency of system.
- **Scalability -** is the capability of an algorithm to perform load balancing for a system with any limited number of system nodes.
- **Migration time** – Time needed to reallocate the resources from one node to another. The goal is to keep it minimized in order to enhance the performance of the system.
- **Resource Utilization** – is the measure to analysis the usage of resources in system.

Issues and Challenges

Major issues that are considered while designing any load balancing algorithms are as follows [19, 21, 24, 25, 26]:

- **Performance Degradation:** The system need to be efficient enough to handle the increasing load.
- **Scalability:** The system should have the ability to accommodate the changes due to increase in web traffic.
- **Availability:** To ensure that clients get 24/7 access to the server at a reasonable response time.
- **Load Estimation:** Collecting information of load on system as a whole and load on each individual node also.
- **Load Level Comparison:** To compare the load on individual nodes for deciding which one is capable of handling the new requests.
- **Performance Indices:** Indices by which performance of a system is measured such as throughput, resource utilization, mean response time.
- **Stability:** Maintaining system stability in case of sudden increase in load on the system.
- **Amount of Information Exchanged among Nodes:** To provide necessary information needed for making load balancing and distributing decisions between the nodes.
- **Job Selection:** To decide whether the job is eligible for the transfer from current node to remote note or not.

Conclusion

Load balancing algorithm can improve a distributed system's performance by attentively distributing the workload among its nodes. This paper presented the review on load balancing terms and techniques required for the growth of an effective load balancing algorithm. Several key issues and policies present in existing systems are discussed. The objective is to provide a guide on the concepts of load balancing that are needed to be considered in development or study of an effective load balancing algorithm.

Refercenes

- Niranjan et al., “Load Distributing for Locally Distributed Systems”, IEEE 1992.
- Marc & Anthony, “Strategies for Dynamic Load Balancing on Highly Parallel Computers”, IEEE Trans. on Parallel and Distributed Systems, Vol.4, No.9, 1993.
- Amandeep Kaur Sidhu and Supriya Kinger, “Analysis of Load Balancing Techniques in Cloud Computing”, International Journal of Computers & Technology, Vol.4 No. 2, March-April, 2013.
- Wolfgang & Michael, “Load Balancing in Distributed Systems”, Electronics and Energetics, Vol.10, No.2, 1997.
- Valeria et al., “Dynamic Load Balancing on Web Server Systems”, IEEE Trans. on Internet Computing, Vol.3, No.3, 1999.
- Yong & Rasul, “Comparison of Load Balancing Strategies on Cluster-Based Web Servers”, Trans. on Modeling and Simulation, 2001.
- Atul Garg and Dimple Juneja, “A Comparison and analysis of various extended techniques of query optimization”, International Journal of Information and Communication Technology, Vol.3 No.3, July-2012.
- Daniel & Anthony, “Algorithmic Mechanism Design for Load balancing in Distributed Systems”, IEEE Trans. on Systems, Man and Cybernetics”, Vol.34, No.1, 2004.
- Belabbas et al., “Load Balancing in grid Computing”, AJIT, Vol.5, No.10, 2006.
- Dimple Juneja and Atul Garg, “Collective Intelligence based framework for load balancing of web servers”, IJICT Vol.3 No.1, 2012.
- Jaspreet Kaur, “Comparison of load balancing algorithms in a Cloud”, International Journal of Engineering Research and Applications , Vol. 2 No.3 pp 1169-1173, 2012.
- Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao and Li Ruan, “A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers”, Journal of Parallel and Distributed Computing, Vol.72 No.10 pp 1254–1268, 2012.
- Ali M. Alakeel, “A Guide to Dynamic Load balancing in Distributed systems”, IJCSNS, Vol.10, No.2, 2012.
- Branko & Mario, “Analysis of Issues with Load Balancing Algorithms in Hosted (Cloud) Environments”, MIPRO, Procee. of 34th Intl. Conf., IEEE, 2011.
- Dhinesh Babu L.D and P. VenkataKrishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments”, Applied Soft Computing, Vol.13 No.5 pp 2292–2303, May 2013.
- Payal and Atul Garg, “A Framework to Optimize Load Balancing to Improve the Performance of Distributed Systems”, International Journal of Computer Applications (0975-8887), Vol.122 No.15, July-2015.

- Camellia & Hamid, "A Survey for Load Balancing in Mobile WiMAX Networks", ACIJ, Vol.3, No.2, 2012.
- Nidhi & Inderveer, "Cloud Load Balancing Techniques: A Step towards Green Computing", IJCSI, Vol.9, No.1, 2012.
- Yaser at al., "Emerging Issues & Challenges in Cloud Computing- A Hybrid Approach", JSEA, Vol.5, No.11, 2012.
- Klaithem et al., "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" Symposium on NCCA, IEEE, 2012.
- P. Beaulah et al., "Comparative Study on Load Balancing Techniques in Distributed Systems", IJITKM, Vol.6, No.1, 2012.
- Dimple Juneja & Atul Garg, " Collective Intelligence based framework for load balancing of web servers", IJICT, Vol 3 No.1, 2012.
- Jayant Adhikari and Sulbha Patil, "Load Balancing the Essential Factor in Cloud Computing", International Journal of Engineering Research & Technology ISSN: 2278-0181 Vol.1 No.10, December 2012.
- Mayank & Atul, "Comparative Survey of Load Balancing Algorithms in Cloud Computing Environment" ,IJDCC, Vol.1, No.2, 2013.
- Tushar & Jignesh, "A Survey on Various Load Balancing Techniques and Challenges in Cloud Computing", IJSTR, Vol.2, No.11, 2013.
- Rajesh & Jeykrishan, "A Survey on Load Balancing in Cloud Computing Environment", IJARCEE, Vol.2, No.12, 2013.
- Bushan & Urmila, "Pros and Cons of Load Balancing Algorithms for Cloud Computing", International Conference on Information Systems and Networks, 2014.
- Ritika & Harjot, "Load Balancing Techniques using Mobile Agents", IPASJ-IIJCS, Vol.2, No.12, 2014.
- Luis Aversa & Azer Bestavros, "Load Balancing a Cluster of Web Servers Using Distributed Packet Rewriting" NSF research grants CCR-9706685.
- Sweekriti & Sudher, "Distributed and dynamic load balancing in cloud data center", IJCSMC, Vol.4 No.5, 2015.