Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 1, 2021, Pages. 01 - 08
Received 15 December 2020; Accepted 05 January 2021.

Framework for Reducing Response Time using Semi-Distributed Load Balancing

Atul Garg, Pinaki Ghosh, Kamali Gupta, Devendra Prasad
Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab
Corresponding Author: pinaki.ghosh@chitkara.edu.in

Abstract: The advancements taking place in present technological era has led to increased traffic onto the web services. This escalation has posed challenges before service providers in satiating scalability, reliability and availability needs of consumers while meeting the bandwidth constraints due to increased congestion. The lack of provisioning of such metrics may lead to distrust in consumer and can be a hindrance in user adaptability to web based solutions. This has arisen a need to distribute the load evenly among the machines without compromising the performance of existing network configurations. This research paper thus, presents a brief discussion on various performance metrics of a load balancing system and proposes a framework for reducing response time and improving throughput.

Keywords: *Load balancing; throughput; response time; Assessment of Load; Estimation of Load; Unexpected Load*

I. INTRODUCTION

Growing innovations have totally changed the aspects of web services. Today web services fulfill numerous needs with a single figure touch. The tremendous growth in usage of Internet and related services the demand for high scalability, availability and reliability has also raised. These requirements have increased the storage and web traffic problem. Further, service delay is decreased whereas response time and network congestion is increased [1]. To overcome these problems multiple servers are used to balance the load of overloaded single server.

Social media, banking system, online booking system, Government agencies, and Internet Service providers etc. replicate their information on multiple servers for better capacity, reliability and service. The technology that distributes the traffic to these multiple servers is known as load balancing. The load balancer is better to distribute the load among the available hosts control the overall system. On the other way the client always remains unaware about all this process. [2].

Load balancing is method of gathering information about all nodes on the systems and sharing the workload globally for upgrading the system performance. Proper resource utilization, high throughput and low response time are some benefits provided by load balancing. In distributed systems solely relies on precise learning of state information of nodes in system. This information is used to redirect the client request to an appropriate node for speedy processing. The main aim is to avoid the condition of load imbalance in system [3]. Figure 1 depicts the use of load balancer in the distribution system.

The organization of paper inculcates literature review in Section2. Section 3 presents issues and challenges. Section 4 discusses about performance metric. The proposed framework is presented in Section 5. Section 6 gives conclusion.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 1, 2021, Pages. 01 - 08
Received 15 December 2020; Accepted 05 January 2021.

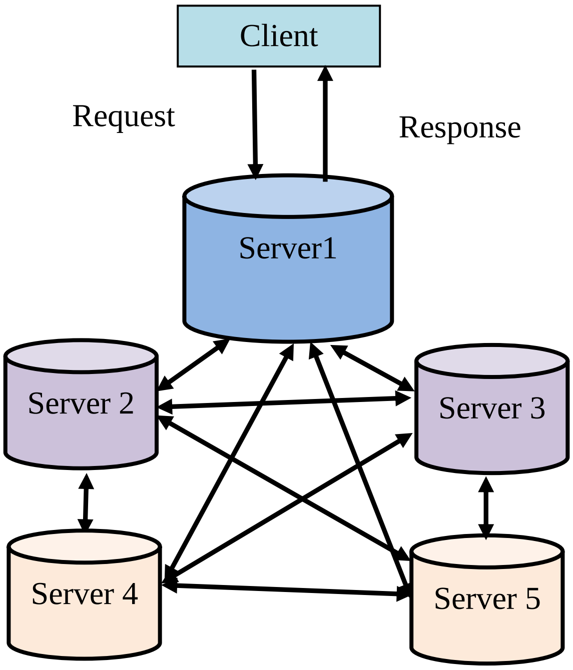

Figure 1: Load Balancer

II. LITERATURE REVIEW

Many researchers are working in the related work to improve the quality and efficiency of web. Research work of some renowned researchers is discussed in this section. In [4] researchers presented a diffusive load balancing policies for dynamic applications. Various local dynamic load balancing policies are implemented using diffusion. Goal is to compare the performance of the dynamic applications. The work concluded that the proposed diffusion policy is beneficial only in case of less loaded dynamic applications. Static applications make their decision on non-obsolete information. Further, by the researchers [4] with context of diffusive policies, no policy is suitable for all kinds of parallel applications.

Authors in [5] proposed an algorithm decentralized in nature for P2P (Peer-To-Peer) systems. P2P systems are considered to be decentralized in nature and having dynamic applications. The algorithm used the concept of Anthill and is named- Messor. Messor Ants move around the system for collecting information of under-loaded and overloaded nodes. The algorithm basically presented the adaptive nature of species for organizing process on the system comparing the messor algorithm with the others is kept as future work.

In [6] the author proposed the load balancing algorithm of dynamic nature for D-GIS systems. The system setup has number of GIS servers, these servers execute the client's request and provide the necessary information to the processors. The request sent by client is firstly intercepted by the designed algorithm and it directs the request to the desired server. The solution provides high scalability but still need to be implemented for business purposes.

Researchers in [7] presented a meta-heuristic technique for load balancing in distributed systems using Ant Colony Optimization. The proposed algorithm used different color Ants. Ants of same color only follow the trailing path created earlier by same color Ant. No two different colored Ants can use the same path to reach the destination. The scheme provides the different routes to single server. Also, the distribution of Ants on a whole system improves the average response time of system. The comparison of proposed algorithm and work-stealing approach is also compared in their research.

The author in [8] proposed a load balancing algorithm by deploying the fuzzy logic concepts. The fuzzy parameter provides the features of accurate load information, effective decision making for distributing load and maintaining system stability. The system consist of N nodes having single processor each connected by local area network. All the tasks arrive at process queue and are served on FCFS basis. Each node has a unique identification number. Communication cost is same for all nodes. It is assumed that the system is in steady state at the initial level. Their approach of load balancing technique is implemented in centralized form and time is in distributed form. Experimental study of proposed algorithm using simulation is kept as future work.

The next section discusses Issues and challenges.

III. ISSUES & CHALLENGES

Distributed system is a collection of multiple computers connected with each other via some Network. Clients send their request to host computer for response. The multiple or continuous request to the single server could be the cause of overloading for the single server while others server are either idle or just daintily loaded. Load balancing improves performance by distributing load among all the servers. The fundamental objective of load balancing is to optimize the average response time and provide better service to the users. Load balancing is crucial for managing various functions proficiently in distributed atmosphere. 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 [9 & 10]. 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.

Load balancing can be defined as the process of finding an appropriate server to execute the client request. Researchers [11, 12 & 13] are using various techniques to modify the algorithms time-to-time to better suit the demands of web. The concepts of grid computing [14 & 15], cloud computing [16, 17 & 18], swarm intelligence [19] and fuzzy [20] are widely used by researchers for the purpose of creation of effective load balancing solution. The load balancing algorithm can improve a distributed system's performance by judiciously redistributing the workload among nodes. The performance metrics discussed are needed for the development of effective load balancing algorithms [21-25]. The study of work carried out by different researchers show that there are still several key challenges need to be addressed for competent working of web services. Some of the challenges are as follows:

- **Job Selection:** To decide whether the job is eligible for the transfer from current node to remote note or not.
- **Unexpected Load:** Dynamic nature of requests sent by users cause sudden increase of load on web servers. This condition leads to fall out of one or more required performance parameters, resulting in a lower quality of service. Maintaining the predetermined limits of various parameters is a challenging task for load balancing algorithms.
- **Performance Degradation:** The system need to be efficient enough to handle the increasing load.
- **Load Estimation:** Collecting information of load on system as a whole and load on each individual node also.
- **Availability:** The expansion in web service applications out turn the number of users accessing the Internet. This has increased the liability on web servers to provide continuous access of services to users. To ensure, that user get 24/7 access to server at a reasonable cost is emerging as a major issue of concern for researchers developing load balancing algorithms.
- **Load Level Comparison:** To compare the load on individual nodes for deciding which one is capable of handling the new requests.
- **Amount of Information Exchanged among Nodes:** To provide necessary information needed for making load balancing and distributing decisions between the nodes.
- **Assessment of Load:** As large numbers of applications are executed at a single point of time, collecting the information about load estimation on a system as a whole and on individual nodes is a tough task to be achieved.
- **Estimation of Load:** To avoid the load imbalance condition, comparison is to be performed between load handling capacities of individual nodes. This analysis helps in formation of constructive decision

making measures. The load balancing decision should be effective in manner to avoid any overloaded or under loaded condition on a node.

- **Transmit of Information between Nodes:** To get optimized results of load balancing, communication between nodes is must. The communication network of system provides necessary information needed for making load balancing and distributing decisions. It should have capability of providing correct and updated state information about nodes. So, designing of a load balancing algorithm which transmits information at reasonable cost is a challenging task.
- **Job Preference:** To abolish the overloaded condition from a node, the decision for selecting whether the job is eligible for the transfer from current node to remote node or not is an difficult task to be achieved.
- **Performance Indicators:** To evaluate the proficiency of a system various indices are defined. Throughput, response time, CPU utilization, resource utilization, transfer delay etc are some measures which are studied on timely basis to check the efficiency of system. To design the method for measuring the productivity of these indicators is a major issue in designing of load balancing algorithm.

The performance metrics for a good load balancer system is discussed in next section.

IV. PERFORMANCE METRICS

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. The metrics are further divided into two parts such as quantitative and qualitative. Further, some of the metrics are independent whereas many of them are dependent on values and behavior of other metrics. Various parameters that act as performance matrices are shown in Figure 2[10]:

- **Overhead**
  It refers to the amount of extra resources needed. It is the combination of excess computational bandwidth, execution time, system memory required to process a particular request. Lesser the overhead in implementation of any algorithm greater will be the efficiency of the algorithm.
- **Throughput**
  In general it is the rate at which something can be processed. It is defined as the number of tasks executed successfully within a specific time. It should be high for the better performance of an algorithm.
- **Utilization of resources**
  This parameter is discussed about the resource utilization of a particular server. With the help of automatic load balancing the resources can be used effectively. A distributed system has number of processors each having different resource needs. The load balancing algorithm should be capable of transferring the resources from one node to another. This reallocation of resources is performed from overloaded to under-loaded nodes. The utilization of resources should be optimized in order to get good performance.

Received 15 December 2020; Accepted 05 January 2021.

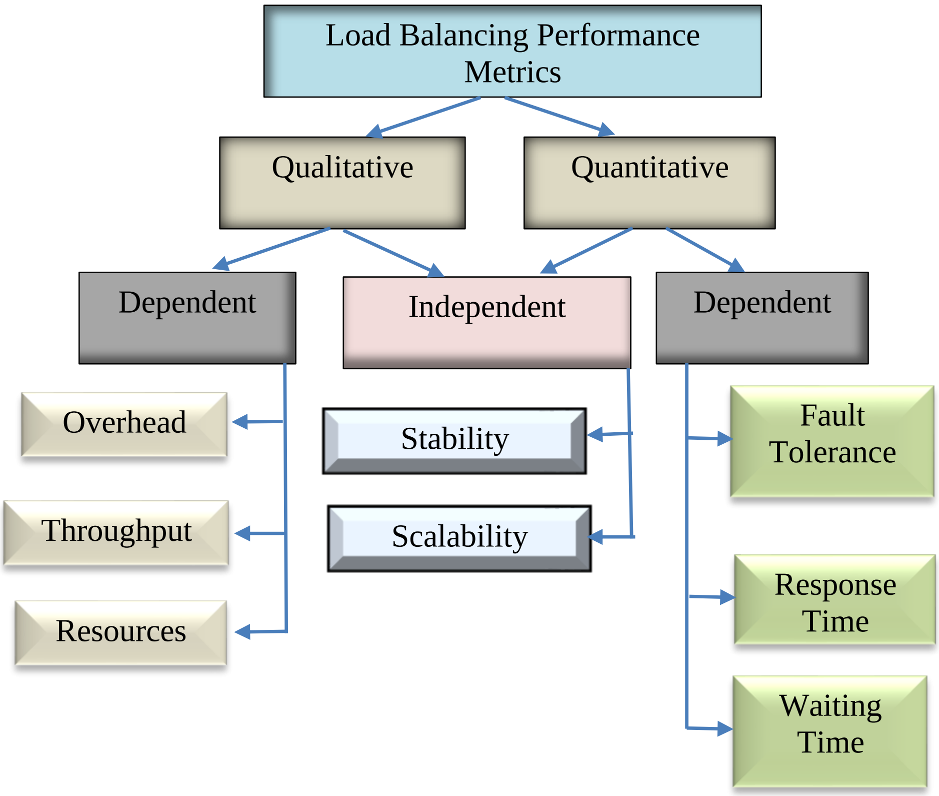

Figure 2: Load Balancing Performance Metrics

- **Fault Tolerance**
  This parameter indicates that tolerance rate of during the processing time of any request from clients. This parameter makes its processing even during the unexpected arrival of any kind of error.
- **Response Time**
  Response time is the important parameter in the performance of any request sent by the clients. This parameter can be achieved with proper distribution of load of main servers. Lesser the response time better will be the performance of the system.
- **Adaptability**
  The parameter defines the capability of a load balancing algorithm to perform with pre-defined measures in case of change in nature of incoming requests.
- **Waiting Time**
  It is defined as the time period of the newly arrived request in the ready queue of the system. Request having lesser waiting time shows that system is running efficiently without any fault in its processing. It also yields in better load balancing decisions.
- **Stability**
  This parameter measures the delay in exchange of state information among nodes. Greater the delay, lesser will be the transfer rate. This will result in poor performance of system.
- **Scalability**
  It is defined as the capability of the system to perform productively in case of system expansion. It is the ability of an algorithm to give optimized result in case of sudden increase in load on the system.

V. PROPOSED FRAMEWORK

Rapid growth in use of technology, computer, web and smart phones have increased the load on servers. During the COVID-19, the use of web users is also increased drastically. In its results the servers became more overloaded and the performance is decreasing continuously. In literature distributed system is proposed to reduce this problem. With this method the load is further distributed among sub servers. But,

Received 15 December 2020; Accepted 05 January 2021.

the main problem with this system is to optimize the performance of these distributed servers. In this work a framework is proposed in which Ant Colony Optimization (ACO) is used with the sub semi distributed servers. The sub semi distributed servers are attached with the main distributed servers on the basis of load in the region.

The objective of this proposed framework is to apply ACO to optimize a Semi-Distributed server network in Distributed Systems. It is anticipated that by including a separate responsive entity, response time will be reduced and throughput will be improved. The semi servers are used with the main servers to distribute the load of the main server. The Semi-distributed servers are optimized with the different states of ants. The types of Ants are namely active, passive and responsive. All the ants are created at the server end. Once a request is received from the client, the active ant (AA) and passive ant (PA) becomes active to find the appropriate server and to keep the current status of other servers respectively. AA allocate the load to least loaded sub-server which can process the query efficiently and in a faster way. The status of other servers is shared among the distributed servers to allocate the load on them for future allocations.

Once the appropriate server is found responsive ant (RA) is activated and respond to the client and PA will updated information of semi-servers to the main servers. The proposed framework is presented in Figure 3.

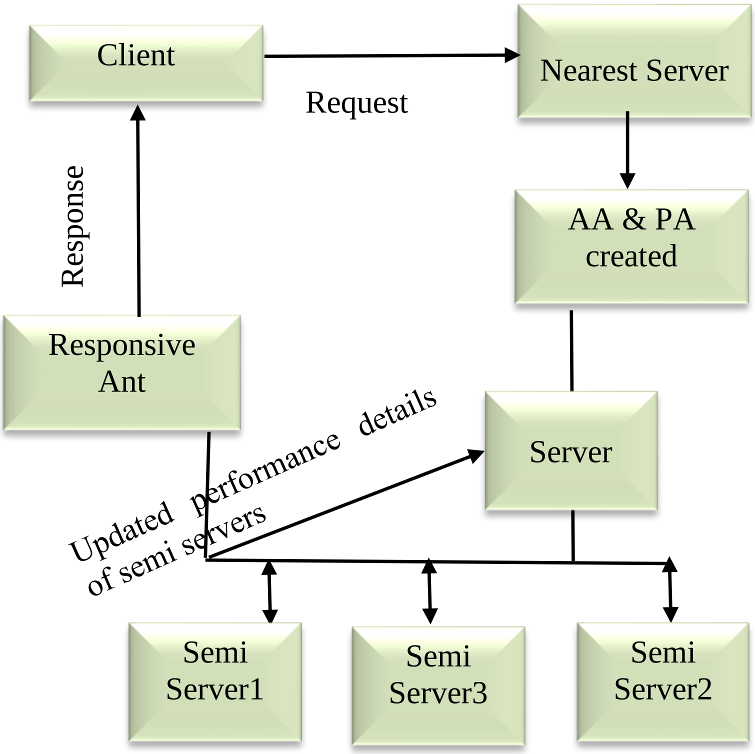

Figure 3: Proposed framework for load distribution and Optimization

The algorithm for the proposed framework is presented below:

```
Step 1: Request from client to Server
Step 2: Active ant (AA) and Responsive Ant
        (RA) created
Step 3: AA optimize the performance of server
Step 4: If sub-server appropriate
            Then go to next step
        Else
            go to previous step 4
Step 5: Create Responsive Ant and go to
```

```
        step 7
    Step 6: AA will be destroyed.
    Step 7: RA will respond to client and PA will update information (about load and performance) of sub servers to main server.
    Step 9: Exit
```

The next section presents conclusion.

VI. CONCLUSION

The increasing traffic over internet has originated the problems of network congestion which further cause's latency in service delivery and hence has become a major source of user distrust in web based services. Envisioned with the aim to explore solutions in catering to the improved performance of internet services, the study comprehends important performance metrics that should be considered while deploying any load balancing technique. A brief discussion on understanding the concept of load balancing system has also been presented using a load balancing system for a distributed network. A framework is proposed for reducing response time and improving throughput.

REFERENCES

[1] Payal Beniwal & Atul Garg, "A comparative study of static and dynamic Load Balancing Algorithms," in International Journal of Advance Research in Computer Science and Management Studies, Vol. 2, Issue 12, pp. 386-392, December 2014.

[2] Aweya, M. Ouellette, D. Y. Montuno, B. Doray and K. Felske, "An Adaptive Load Balancing Scheme for Web Servers", International Journal of Network Management Vol. 12 pp 3-39, 2012.

[3] Bryhni, E. Klovning and O. Kure, "A Comparison of Load Balancing Techniques for Scalable Web Servers", IEEE Network Vol. 14 No.4 pp 58-64, July/August 2000.

[4] A. Corradi, L. Leonardi and F. Zambonelli, "Diffusive load balancing policies for dynamic applications", IEEE Concurrency, Vol.7 No.1 pp 22–31, Jan.-March 1999.

[5] Dimple Juneja & Atul Garg, "Collective Intelligence based Framework for Load Balancing of Web Servers", in International Journal of Advancements in Technology, Vol. 3(1), pp. 64 to 70, January 2012

[6] Aissatou Diasse, "Dynamic-Distributed Load Balancing for Highly-Performance and Responsiveness Distributed-GIS (D-GIS)", Journal of Geographic Information System, Vol.3 pp 128-139, April 2011.

[7] Al-Dahoud Ali and Mohamed A. Belal, "Load Balancing of Distributed Systems based on Multiple Ant Colonies Optimization", American Journal of Applied Science, Vol.7 No.3, 2010.

[8] Ali M.Alakeel, "A Fuzzy Dynamic Load Balancing Algorithm for Homogeneous Distributed Systems", World Academy of Science, Engineering and Technology, Vol.6, 2012.

[9] Ali M.Alakeel, "A Guide to Dynamic Load balancing in Distributed systems", International Journal of Computer Science and Network Security, Vol.10 No.2, 2012.

[10] Aweya, M. Ouellette, D. Y. Montuno, B. Doray and K. Felske, "An Adaptive Load Balancing Scheme for Web Servers", International Journal of Network Management Vol.12 pp 3-39, 2012.

[11] Edmundo Madeira, and Luiz E. Buzato, "Improving the QoS of Web Services via Client-Based Load Distribution", XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (Aceito para apresentação). May, 2011.

[12] Hao Liu, Shijun Liu, Xiangxu Meng, Chengwei Yang, Yong Zhang, "LBVS:A Load Balancing Strategy for Virtual Storage", IEEE International Conference on Service Sciences, 2010.

[13] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" Symposium on NCCA, IEEE, 2012.

[14] M. Zaki, W. Li and S. Parthasarathy, "Customized dynamic load balancing for a network of workstations", Journal of Parallel and Distributed Computing: Special Issue on Performance Evaluation, Scheduling, and Fault Tolerance, June 1997.

[15] Mohsen and Hossein Delda, "Balancing Load in a Computational Grid Applying Adaptive, Intelligent Colonies of Ants", Informatica 32, pp327–335, 2008.

[16] 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.

[17] 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.

[18] Jitender Grover and Shivangi Katiyar, "Agent Based Dynamic Load Balancing in Cloud Computing", Human Computer Interactions International Conference, pp 1-6, 23-24 Aug. 2013.

[19] Kwang, M.S., Sun and H.W., "Ant colony optimization for routing and load-balancing: survey and new directions", IEEE Trans. Syst. Man Cybern. Vol.33 No.5 pp 560–572, 2003.

[20] Anand, L., Ghose, D. and Mani, V. "ELISA: an estimated load information scheduling algorithm for distributed computing systems", International Journal of Computers and Math With Applications, Vol. 37, No. 8, pp.57-85, 1999.

[21] 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.

[22] Branko and Mario, "Analysis of Issues with load Balancing Algorithms in Hosted (Cloud) Environments", MIPRO Proceedings of 34th International Convention IEEE, pp 416-420, 23-27 May 2011.

[23] Meenakshi Gupta and Atul Garg, "Optimizing and Load Balancing for Flash Crowd to Improve Quality of Service in Content Delivery Network", published in International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol-8 Issue-11, pp 2568-2574, September 2019.

[24] Hioual Ouided, Laskri Mhamed Tayeb, Hemam Sofiane Mounine, Hioual Ouassila and Maifi Lyes, "Towards An Implementation of A Modified Static Load Balancing Algorithm to Minimize Execution Time", Recent Patents on Computer Science, Vol-12 Issue-1, pp 69-74, 2019.

[25] Talaat, F.M., Saraya, M.S., Saleh, A.I. et al., "A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment", Journal of Ambient Intelligence and Humanized Computing11, pp 4951–4966, 2020.