TASHKENT UNIVERSITY OF
INFORMATION TECHNOLOGIES
NAMED AFTER MUHAMMAD AL-KHWARIZMI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI

BULLETIN OF TUIT:

MANAGEMENT AND COMMUNICATION TECHNOLOGIES

ISSN:2181-1083 DOI:
Rakhimov Mekhriddin, Ergashev Shakhboz, Karimberdiyev Jakhongir № 3s(56)2024

Analysis of load balancing algorithm in information systems

Rakhimov Mekhriddin,,
Computer engineering,
Tashkent university of information technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan
Rakhimov@gmail.com

Ergashev Shakhboz,
Doctoral student of the Denov Institute of Entrepreneurship and Pedagogy
toshtemirovich144@gmail.com

Karimberdiyev Jakhongir,
Computer engineering,
Tashkent university of information technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan
karimberdiyevjkb@gmail.com

***Abstract*— In this article will show comparative analysis of common load balancing algorithms. The primary purpose of** *load balancing analysis* **is to ensure that the distribution of network traffic or computing workload across multiple servers or resources is** *efficient, fair, and resilient*. **Load balancing is crucial in maintaining high availability, performance, and reliability in systems such as web applications, cloud services, and data centers. Here's a detailed look at the key purposes of analyzing load balancing strategies. Analyzing load balancing algorithms involves evaluating their performance based on various metrics and considerations. There are several established load balancing algorithms, each with unique characteristics and suitable use cases. In this article analaysed some of the most commonly used ones.**

***Key words**: Load Balancing, Delay, Data Plane, P4, Round Robin, IP hash.*

1. INTRODUCTION

Currently, the use of distributed systems for computing is becoming an increasingly common model for building a computing system[1] . Such systems are allowed to distribute the load between computing processors working with a large amount of information. However, when such systems operate, the problem of load distribution between computing units arises. To solve this problem, a load balancing task is posed. With a certain time static load on computing units, there are many balancing algorithms. However, the time is often uncertain, so most balancing methods are not effective enough. Thus, it is necessary to conduct an analysis of dynamic balancing methods to find the most effective[2].

2. RELATED WORKS

Vishwas Kumar, Sandeep Jangir, Deven GPatanvariya presented a new strategy for a load balancing algorithm on a software defined network(SDN). Such problems of the popular load balancers have been solved by using SDN load balancer.The authors executed a load balancer using a round-robin approach and match edit with the existing basic random methodology. For a particular load balancing algo rithm, the round-robin strategy goes well as compared to the random approach.The constructive fact about the round-robin approach is that it allocates the load consistently while random is not. The authors also implemented the Dijkstra algorithm and the round-robin algorithm on fat-tree topology for traffic load balancing[8].

In this case, the Dijkstra algorithm gives better through put as compared to the round-robin algorithm.

Irengbam Tilokchan Singh, Thounaojam Rupachandra Singh, Tejmani Sinamwe paper's primary focus is to benchmark the Round Robin algorithm in Server Load Balancing (SLB) using Software Defined Networking (SDN). Experimental results show that the Round Robin algorithm gives 100% availability of the servers. Load balancing is done among the live servers from the server pool. In their experiment, the Round Robin load-balancing algorithm is implemented using a POX controller and an OpenFlow switch. Also they will introduce a new load balancing technique to maintain a consistent response time in the future[12].

Yue Zhou, Chunyou Liu showed that traditional traffic load balancing algorithms possess their own advantages and drawbacks, unable to provide an optimal solution for various complex network traffic scenarios. In their context of SDN technology, traffic load balancing algorithms can be innovatively designed. The W-RRLC load balancing algorithm proposed in this paper, using the server's comprehensive weight as a key factor, combines the Least Connections algorithm and Round Robin algorithm. Through simulation experiments, it has been verified that this algorithm exhibits advantages over Round Robin and Least Connections algorithms, especially in high-load scenarios, where the load balancing distribution is more even, effectively harnessing the performance of each server[9].

Ajay Katangur, Somasheker Akkaladevi, Sadiskumar Vivekanandhan presented that their work provides a new scheduling algorithm, the PWRR algorithm, to efficiently handle the incoming tasks of different priorities. In general, it performs better than the RR and WRR algorithms in terms of QoS parameters: execution cost, total waiting time, total turnaround time, average time taken to complete a single task, and total time taken to complete all tasks. It also performs equivalent to the DWRR algorithm, which is widely used by cloud providers for task scheduling, in terms of the above QoS parameters. In terms of handling the tasks with priority, the PWRR algorithm shows significant improvement when compared to all the three (RR, WRR, andDWRR) algorithms. From the results, it can be concluded that the PWRR algorithm performs equivalent to the DWRR algorithm in all the QoS factors, but it shows significant improvement over the DWRR algorithms when handling high priority tasks. Hence, the PWRR algorithm is more suitable than the DWRR algorithm for task scheduling, which has the ability to handle tasks with and with out priority[6].

3. METHODS

Load Balancing is the practice of distributing network traffic or computational workloads between two or more computers. Most modern applications have to process millions of request from it's users simultaneously. A load balancer acts as a traffic controller which sits in front of your servers as a facilitator ensuring that all servers are used equally. A load balancer might be a physical device, a software process or a virtualized instance running on a specialized hardware. When a request arrives the load balancer from a user, the load balancer sends the request to a given server. The load balancer repeats that process for all incoming request **(**Figure 1).

Consider a restaurant scenario where a restaurant has fewer customers that the number of waiters. I customers are allowed to select the waiters they want then some waiters would be idle while others would be overloaded with jobs. Let's say we employ a manager (load balancer) who is responsible for assigning customers to waiters. The manager now ensures no waiter would be idle and also make sure no waiter would be overloaded[3, 4, 7, 10].

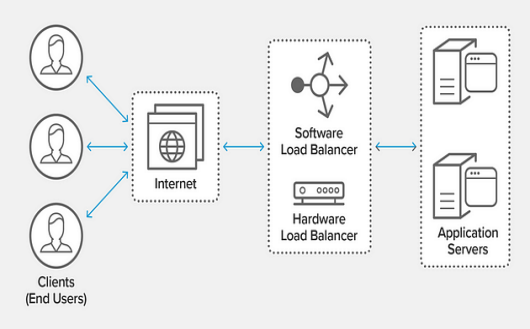

Figure 1. What is load balancing

Load balancers also detect the health of backend resources and do not send traffic to servers that are not able to fulfill request[5,8].Load balancers helps to improve the overall performance and reliability of cloud-based applications by ensuring that resources are used efficiently and that there is no single point of failure. Here's a table summarizing the key metrics for analysis of various load balancing algorithms, based on Efficiency, Scalability, Response Time, Fairness, Overhead, and Robustness[1, 6, 9, 11].

If there is an increase in network traffic, a load balancer can bring up additional servers online to keep up with the demand during the spike. Also, if there is a fall in network traffic, the load balancer can reduce the amount of servers online.

4.1. THE OBTAINED RESEARCH RESULTS

Round Robin is a simple method that ensures each server gets requests in order. However, it doesn't consider how busy each server is, which can lead to inefficiencies.

Least Connections balances based on how many connections each server currently has, which is useful for longer sessions. The downside is it requires monitoring, adding some complexity.

Least Response Time directs requests to the fastest server, improving user experience. It requires constant performance checks, which can be resource-intensive.

Weighted Round Robin allows different load distribution based on server capabilities, providing better performance. However, it's more complex to set up and maintain.

IP Hash consistently routes requests from the same IP to the same server, improving user experience for stateful sessions. But, it may lead to uneven distribution if many clients have similar IPs.

Dynamic Load Balancing adapts to changes in workload and server status in real-time, but it can be complex and involve significant monitoring overhead.

Content-Based Routing optimizes performance by routing requests based on the type of content, like static or dynamic. This adds complexity and requires deeper request inspection[12, 13, 14, 15].

4.2. RESULTS

Random Selection is easy to set up with minimal configuration but can be inefficient, potentially overloading servers (Table 1).

Table 1. Comparative Analysis of Common Load Balancing Algorithms

Algorithm	Efficiency	Scalability	ResponsT	Fairness	Overhead
Round Robin	Medium	High	Medium	High	Low
Least Connection	High	High	Medium	High	Medium
Least Response Time	High	Medium	Low	Medium	High
Weighted Round Robin	High	High	Medium	Medium	Medium
IP Hash	Medium	Medium	Medium	Low	Low
Dynamic Load Balanc	High	High	Low	High	High

4.3. DISCUSSION

The discussion about the analysis of load balancing algorithms in information systems focuses on evaluating and improving methods used to distribute workloads across multiple computing resources, such as servers, databases, or network nodes, to achieve optimal performance, reliability, and scalability. Here's a updown of key aspects that such discussions often cover

The analysis aims to identify the most effective algorithm for specific scenarios while considering the trade-offs in performance, complexity, and resource requirements.

5. CONCLUSION

The choice of load balancing algorithm should align with specific application needs, infrastructure, and performance requirements. A comprehensive evaluation of these metrics can guide the selection process, ensuring optimal resource utilization and user experience. In many cases, hybrid approaches that combine elements from multiple algorithms can yield the best results. The best algorithm is the one that aligns with your traffic patterns**,** server configurations**,** and performance requirements. For example, Round Robin may work well for small, static websites with identical servers, while Least Connections or Dynamic Load Balancing would

be a better choice for complex, high-traffic websites with varying server capacities and needs.

checks and session persistence to optimize performance, availability, and scalability.

Each load balancing algorithm has its advantages and trade-offs, making them suited for different types of

complexity and performance. For simple environments, Round Robin or Random Selection may suffice, but for more complex, high-traffic systems, algorithms like Least

It's often beneficial to combine algorithms based on specific use cases or implement additional features like health applications and server infrastructures. The choice of algorithm should depend on factors such as the nature of the application (e.g., whether session persistence is important), server capabilities, and the desired balance between

Connections, Weighted Round Robin, or Dynamic Load Balancing might offer better performance and scalability.

REFERENCES

[1] T. Bourke, "Server load balancing" in , OReilly Media, 2021.

[2] M. M. Maxmudovich, R. M. Fazliddinovich, B. U. Abdumurodovich and S. K. E. o'gli, "Parallel algorithms for acoustic processing of speech signals," 2016 IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, China, 2016, pp. 421-425.

[3] Zubair Khan, Ravendra Singh, Jahangir Alam and Shailesh Saxena, "Classification of Load Balancing Conditions for parallel and distributed systems", *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 5, September 2011.

[4] Kh. Khujamatov, D. Khasanov, E. Reypnazarov and N. Akhmedov, "Networking and Computing in Internet of Things and Cyber-Physical Systems", *The 14th IEEE International Conference Application of Information and Communication Technologies*, 07–09 October 2020.

[5] T. Bourke, "Server load balancing" in , OReilly Media, 2001.

[6] A. Rao et al., "Load balancing in structured p2p systems", *International Workshop on Peer-to-Peer Systems*, 2003.

[7] H. Shida et al., "Results of EMC round robin test on emission and immunity tests (3) Conducted immunity round robin test," 2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC), Sapporo, Japan, 2019, pp. 69-72

[8] L. Zhu, J. Cui and G. Xiong, "Improved dynamic load balancing algorithm based on Least-Connection Scheduling," 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2018, pp. 1858-1862

[9] M. R. Baihaqi, R. M. Negara and R. Tulloh, "Analysis of Load Balancing Performance using Round Robin and IP Hash Algorithm on P4," 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2022, pp. 93-98

[10] S. Salifu, N. Turlington and M. Galloway, "Performance Profiling of Load Balancing Algorithms in a Cloud Architecture," 2021 IEEE Cloud Summit (Cloud Summit), Hempstead, NY, USA, 2021, pp. 77-82

[11] Y. Hu and R. Blake. An Improved Diffusion Algorithm for Dynamic Load Balancing, Parallel Computing, 1999, 25: pp. 417-444.

[12] G. Cybenko. Dynamic Load Balancing for Distributed Memory Multiprocessors, Journal of Parallel and Distributed Computing, 1989, 7(2): pp. 279-301.

[13] Tang, Z., Birdwell, J. D. and Chiasson, J., Resource-Constrained Load Balancing Controller for a Parallel Database, IEEE Transactions on Control Systems Technology, 2008, 16(4): pp. 834-840

[14] A. N. Nazarov, *Choosing Hardware-and-Software Facilities for Designing the Microprocessor Switching Module of the Multistage ATM Switch Using by the Real-Time Criterion*, [online] Available: https://doi.org/10.1023/A:1023217725866.

[15] Nevin Vunka Jungum, Nawaz Mohamudally and Nimal Nissanke, "A Dynamic Load Balancing Algorithm for Distributing Mobile Codes in Multi-Applications and Multi-Hosts Environment", *Zenodo*, 2020, [online] Available: https://doi.org/10.5281/zenodo.3991567.

[16] Nidhi Kataria Chawla, "A New Fuzzy Algorithm for Dynamic Load Balancing In Distributed Environment", *Asian Journal of Technology & Management Research [ISSN: 2249-0892]*, vol. 04, no. 01, Jan - Jun 2014.