Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 15, Issue. 1, January 2026, pg.45 – 55

Throughput Evaluation of Load Balancing Algorithms (Round Robin, Number of Connection, Server Load) using Identical Servers and Multiple-Application Environments

Mustafa ElGili Mustafa

Applied College- Huraymila, Computer Programs, Imam Muhammad Ibn Saud University, Saudi Arabia
meelmogadem@imamu.edu.sa

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i01.004

Abstract: This paper compares various load balancing algorithms, including round robin, least connections, through simulations implemented with OPNET® Modeller, we analyse the throughput and severs load variation of these algorithms when connected to 4, 8, and 12 servers serving 10000 clients under a 10,000-client mixed workload consisting of HTTP, database, and mail applications. The results illustrate that Round Robin consistently achieves the highest throughput and exhibits stable performance across all cluster sizes, Number of Connections performs poorly in small clusters but demonstrates substantial improvement as server count increases. Variance analysis further confirms stability of Round Robin and Number of Connections in large clusters, while Server Load remains the least stable algorithm.
Keywords: Load Balance, Round Robin, Least Connection, number of connection, identical servers

I. INTRODUCTION

The increasing internet usage has made information availability crucial [1]. The internet is a major hub of heavy traffic, with numerous applications attracting millions of users. Many application providers use multiple servers as a computing unit to support specific applications, as a single server struggles to handle numerous client accesses. It is crucial to maintain a balance in the network's load [2]. Distributed computing and load-balancing technologies are commonly utilized to manage this workload [3]. A suitable load-balancing algorithm can effectively tackle these issues in a system cluster [4].

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

II. THE ARCHITECTURE OF SERVER CLUSTER

A common issue when numerous visitors access distributed system simultaneously is that the server struggles to handle the requests, leading to service failure [1]. A cluster-based distributed system consists of N server machines connected via a high-speed network, each acting as a node with its own disk and operating system. It uses a single interface for external users, requiring a mechanism to manage requests [3]. Load-balancing is a method that distributes workloads across computing resources to improve performance by distributing them evenly across resources [1]. An incoming request can be directed to the first available cluster resource or randomly selected from all available resources to fulfil the request [5].

III. LOAD BALANCING

Load balancing is a technique that distributes a larger processing load across smaller nodes to improve system performance [6]. Load balancing is a technique that distributes workload across computing resources, ensuring no idle resources are used while others are utilized [7]. This technique efficiently distributes computing resources and workloads across multiple systems, networks, and servers, balancing network traffic by evenly distributing loads among them [8]. Load balancing is a computer networking technique that distributes connection loads across multiple computers or clusters to optimize resource utilization, throughput, response time, and avoid overload [9]. An optimal load balancing algorithm should prevent overloading or underloading of any particular node [6]. Load balancing between links significantly impacts network performance in terms of delay, packet loss, and throughput [2]. Load balancing is a distributed system technique that distributes workload among nodes, improving job response time and resource utilization, and preventing overloading or inactivity [10]. Load balancing maximizes network efficiency by increasing throughput and performance, minimizing resource usage, preventing job starvation, and utilizing redundancy to enhance computing resource reliability [11] [10] [12] [13].

A. Load Balancing Aims: [10] [14] [15]

- Enhance service availability and user satisfaction.
- Maximize resource utilization.
- Minimize waiting and task processing time.
- Increase performance.
- Maintain system stability.
- Build system fault tolerance.
- accommodate future modifications.
- Provide on-demand resource availability.
- Ensure efficient resource utilization under high and low load conditions.
- Improve user satisfaction.
- Reduce resource costs.
- Improve overall system performance.
- Substantially enhance performance.

IV. LOAD BALANCING ALGORITHMS

Load-balancing algorithms in web switches enhance cluster performance by selecting the most suitable server for new requests [3]. Load balancing algorithms are typically categorized into two groups based on the current state of the system; static and dynamic [16] [10].

A. Dynamic load balancing algorithms:

Dynamic load balancing algorithms are flexible, reliable, and capable of handling more user requests than static ones, making them ideal for dynamic environments [10]. Dynamic processing involves monitoring node load, exchange of load and state information, and redistribution of workloads, transferring overloaded nodes to underloaded ones [17].

B. Static load balancing algorithms

Static load balancing algorithms are simple to design and efficient, but only practical in cluster scenarios where most hosts have identical processing capabilities [10]. These strategies are straightforward to implement and execute, but they are primarily suitable for small-scale systems [18]. The technique, though simple and easy to implement, often fails to detect attached servers, resulting in an uneven distribution of resources [17].

V. POPULAR LOAD BALANCING ALGORITHMS

Below, we review some popular load balancing algorithms:

A. *Round Robin*

This static load balancing algorithm is the simplest implementation, utilizing a circular allocation of jobs to processors [19]. The Round Robin (RR) algorithm distributes requests to nodes in a circular manner, assigning each process a defined time slot without priority, making it a common and easy implementation method [12] [20]. Round-robin is a scheduling algorithm that distributes load evenly across backend servers in a group, utilizing time-sharing as its main concept [9]. The RR scheduling algorithm divides time into multiple slices, allocating resources to nodes within the cloud network. The time quantum, which can be longer or short, affects the scheduling process [21]. The system randomly selects the first node and assigns jobs to all other nodes in a round-robin manner, ensuring no starvation [22]. The Round Robin scheme is a simple and starvation-free algorithm that uses a circular queuing process for implementation [23]. The Round Robin algorithm is highly effective in a uniform network environment with identical server configurations [24]. Round Robin algorithm treats all nodes equally, regardless of the number of connections [3]. offers faster process response times, preventing starvation as processes don't need to wait long for their tasks to be executed [19]. Because process's execution time is not determined in advance, leading to some nodes becoming overloaded and others remaining underloaded due to uneven workload distribution. operates solely on time slicing, allocating load to different nodes based on time without considering the need for resources [19][16].

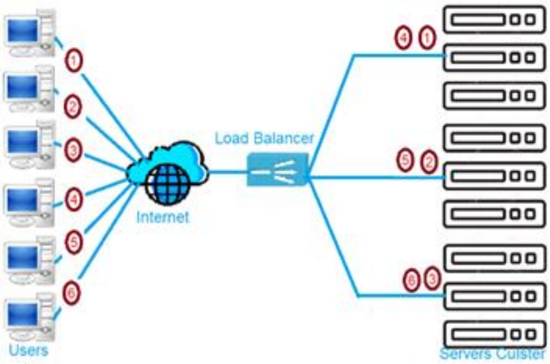

Fig 1. Round Robin Algorithm

B. *Number of Connections*

Load balancing algorithms use dynamic scheduling algorithms' least connection mechanism, counting server connections dynamically to estimate load, with load balancers recording these connections. The number of connections on a server increases with the dispatch of a new connection, and decreases with the completion or timeout of a connection [25]. The least-connection algorithm is a scheduling technique that distributes more requests to real servers by connecting active servers with fewer connections [9]. The system is designed to handle high traffic load and varying connection durations, requiring an even distribution of load based on active connections [26]. The Least Connection method considers the existing server load, while Round Robin does not [23]. This method selects servers with the least active connections to handle new requests, typically resulting in good performance. For example, if server S1 has 10 connections and server S2 has 12, the next request will be sent to server S1 because it has fewer connections. However, if server S1 has a connection capacity of 50, while server S2 can handle 150 connections, S1 is more likely to fail due to being overloaded [3].

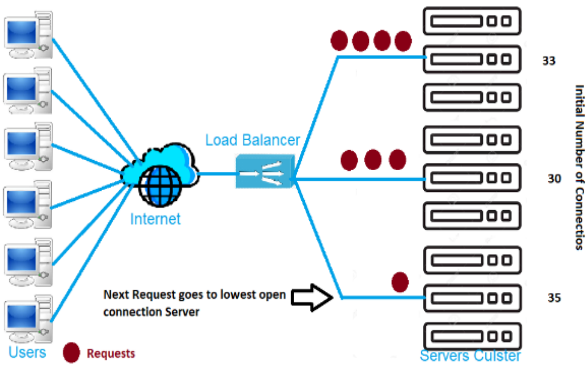

Fig 2. Number of Connection Algorithm

C. *Least Loaded Algorithm*

The dispatcher assigns the next request to the server with the lowest workload, which is the sum of all pending requests on the server. The baseline algorithm, despite its difficulty in practice due to the lack of service time information for client requests, serves as a benchmark for comparing other schemes' performance [3].

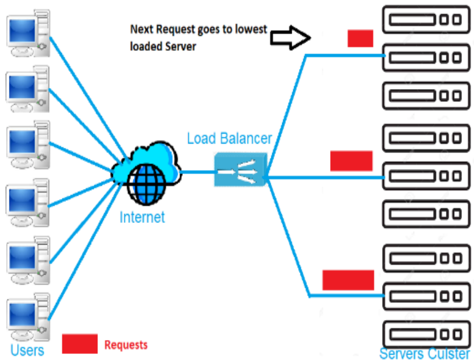

Fig 3. Server Load Algorithm

VI. RELATED WORKS

A. *Load Balancing Analysis Using Round-Robin and Least Connection Algorithms for Server Service Response Time*

The research paper investigates the use of the Round-Robin and Least Connection algorithms for load balancing to enhance server performance by efficiently distributing the connection load. The study involved creating web and load balancing servers, gradually increasing the number of connections, and testing various scenarios to evaluate system performance, particularly improvements in response time. Load balancing is discussed as a model in computer networking to improve throughput, response time, and prevent overload. This study compares algorithms aimed at reducing response times and provides insights into improving system performance through load balancing strategies. Overall, this research offers valuable information on enhancing server performance through effective load balancing techniques.

B. *Load Balancing Algorithms: Round-Robin (RR), Least Connection, and Least Loaded Efficiency [3]*

This paper discussed Load Balancing Algorithms Round-Robin (RR), Least-Connection, and Least Loaded. Evaluating the distribution traffic efficiency, between 8 HTTP Servers and the Load Balancer CPU utilization for each algorithm. Opnet software has been used to simulate the network. The paper conclude that the Server Load algorithm has more CPU utilization comparing with Round Robin algorithm and Number of Connection algorithm and when Server Load algorithm has been used there will be big variance in the load between the 8 HTTP. Number of Connection algorithm distribute traffic in a fair way rather than others algorithms.

C. *7.3 Rahmika et al. [27] LC + Multi-Agent System for Web Traffic*

Rahmika et al. develop a hybrid mechanism that augments the Least Connections algorithm with a Multi-Agent System (LC-MAS) to improve web-server performance. Experiments show that LC-MAS achieves higher throughput, fewer errors, and better resource usage compared to standard LC, particularly under high HTTP traffic. Though insightful for LC improvements, the work focuses exclusively on HTTP traffic and does not study multi-application mixes or Least-Load policies, which your research incorporates.

D. *Fawwazi et al. [28]– Weighted Least Connections vs Round Robin.*

Fawwazi et al. compare Weighted Least Connections (WLC) to Round Robin in cloud environments, analyzing response time, throughput, and CPU utilization. Their findings show that WLC significantly outperforms Round Robin in heterogeneous server settings by adjusting request distribution based on server capacity. The work is relevant to studies of Least Connections-type algorithms but differs from yours by focusing on weighted approaches and heterogeneous clusters rather than Least Load and classical LC on identical servers.

E. *Shahid et al. [29] Performance Evaluation with Service Broker Policies.*

Shahid et al. present a comprehensive analysis of several algorithms—PSO, Round Robin, ESCE, and Throttled—under multiple service-broker policies using CloudAnalyst. The study measures optimized response

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

time, VM processing time, and operational costs across diverse workloads. The authors highlight limitations in prior work, such as ignoring the relationship between task type and VM characteristics. Their cloud-simulation orientation differs from your work, which measures throughput directly on identical physical servers across multi-application scenarios.

F. Li et al. [30] – Unified Testing Platform for Cloud Load Balancing

Li et al. introduce a unified benchmarking platform for evaluating static and dynamic load-balancing algorithms under uniform, tide, and spike workload patterns. Their analysis highlights that dynamic algorithms often outperform static methods, especially under bursty workloads. While the study includes Round Robin, it focuses on heterogeneous server environments and advanced dynamic strategies rather than Least Load or basic LC on identical servers with multi-application workloads—the core focus of your paper.

VII. IMPLEMENTATION

Network is simulated using OPNET® Modeler. OPNET® which it is considerable, powerful simulation software tool with a lot of different abilities. It makes it possible to simulate entire heterogeneous networks with that use different protocols [3]. The first level simulated network designed with (4,8 and 12) http servers provide http service to 2000,5000 and 10000 clients, in second level simulated network designed with (4,8 and 12) (http + DB + Mail) servers provide (http + DB + Mail) service to (2000,5000 and 10000 clients) and there is load balancer device between the servers and clients as shown in the Figures (4,5 and 6) configured with Round-Robin algorithm, Least Connection algorithm and Least load algorithm.

VIII. SCENARIOS

The paper proposed three main Scenarios, the first scenario as shown in figure (4) there are 4 Identical servers with 10000 clients, with multiple applications (HTTP, DB, Mail).The second scenario as shown in figure (5) there are 8 Identical servers with 10000 clients, with multiple applications (HTTP, DB, Mail). The third scenario as shown in figure (6) there are 12 servers with 10000 clients, with multiple applications (HTTP, DB, Mail).

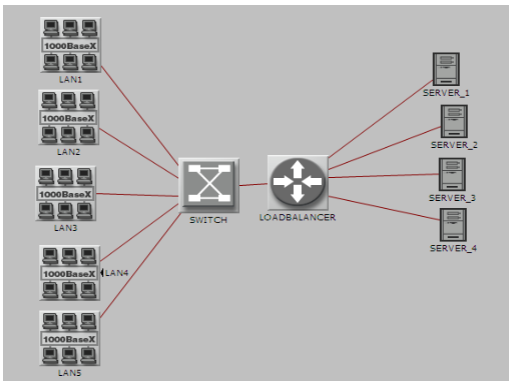

Fig 4. 4SERVERS - 10000 CLIENTS

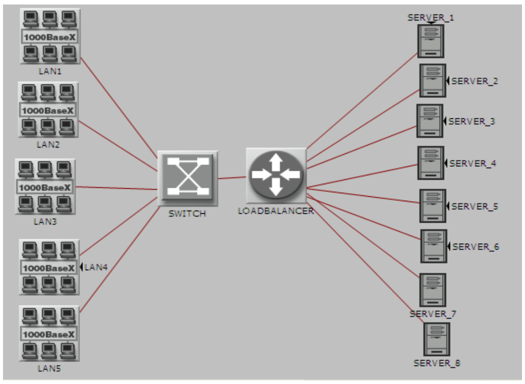

Fig 5. 8SERVERS - 10000 CLIENTS

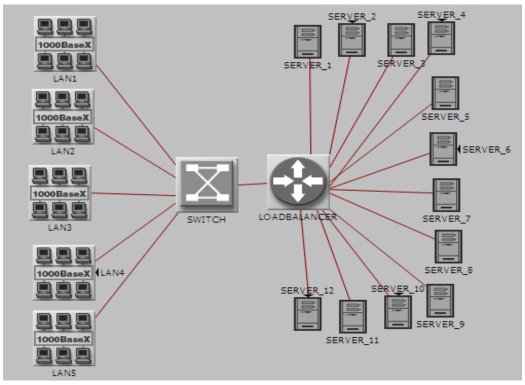

Fig 6. 12 SERVERS - 10000 CLIENTS

IX. RESULTS AND DISCUSSION

A. Four Identical Servers

The results of using 4 identical servers handling 10000 clients for multiple applications (http+ DB+ FTP), as shown in figure (7) and table-1, the Round Robin algorithm has the highest throughput (8401.28974 packet/second), Server Load algorithm throughput (8386.018098 packet/second), and the less throughput is Number of Connection algorithm with (7038.705308 packet/second) , and the variation between servers in throughput as show in figure (8), the Number of Connection algorithm is the less one, because it is variation (18.853865 packets), which is mean it treats all servers equally regardless of their capabilities, instead of the Server Load algorithm which is has biggest variation(1668.911839 packets) , that means this algorithm treats the server according to their capabilities.

TABLE 1. TOTAL THROUGHPUT (PACKETS/SEC) -4 SERVERS – 10000 CLIENTS MULTIPLE APPLICATIONS (HTTP-DB-MAIL)

	SERVER LOAD	ROUND ROBIN	NUMBER OF CONNECTION
TOTAL	8386.018098	8401.28974	7038.705308
MIN	1656.6812	1671.209387	1748.485277
MAX	3325.593039	2524.477241	1767.339142
VAR	1668.911839	853.267854	18.853865

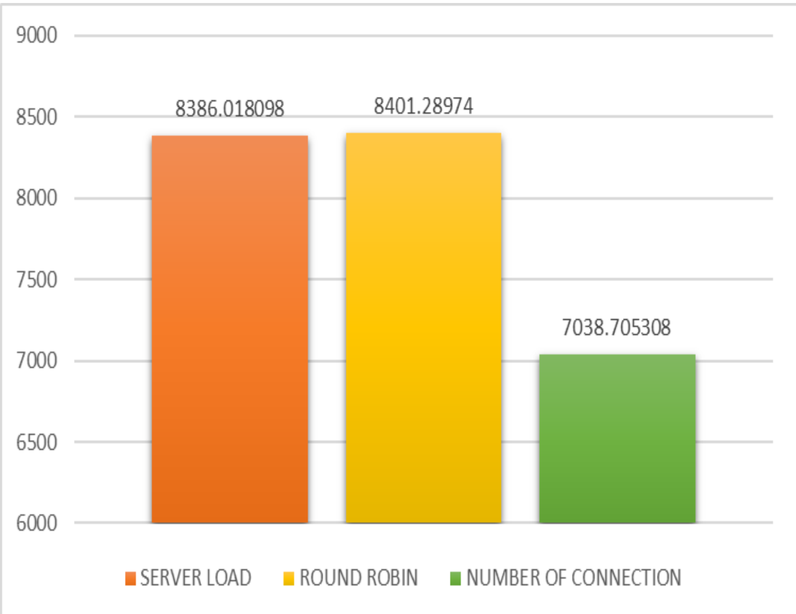

Fig 7. Total Throughput (packets/sec) -4 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

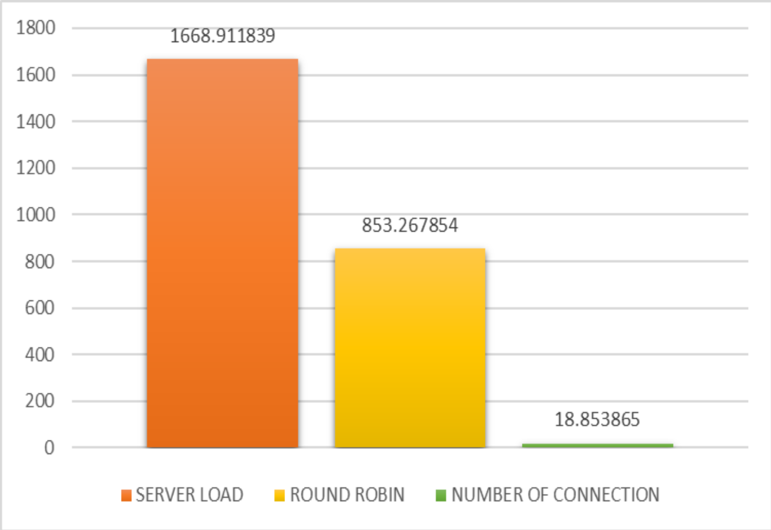

Fig 8. Servers throughput variation (packets) -4 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

B. Eight servers

The results of using 8 identical servers handling 10000 clients for multiple applications (http+ DB+ FTP), as shown in figure (9) and table-2, the Round Robin algorithm has the highest throughput (8430.384219 packet/second), Number of Connection algorithm throughput (8423.672505 packet/second), and the less throughput is Server Load algorithm with (8412.070288 packet/second) , and the variation between servers in throughput as show in figure (10), the number of connection algorithm is the less one, because it is variation (83.759916 packets), which is mean it treats all servers equally regardless of their capabilities, instead of the Server Load algorithm which is has biggest variation(974.9216759 packets), that means this algorithm treats the server according to their capabilities.

TABLE 2. TOTAL THROUGHPUT (PACKETS/SEC) -8 SERVERS – 10000 CLIENTS MULTIPLE APPLICATIONS (HTTP-DB-MAIL)

	SERVER LOAD	ROUND ROBIN	NUMBER OF CONNECTIONS
TOTAL	8412.070288	8430.384219	8423.672505
MIN	901.3058211	920.9972571	1022.915228
MAX	1876.227497	1154.107028	1106.675144
VAR	974.9216759	233.1097709	83.759916

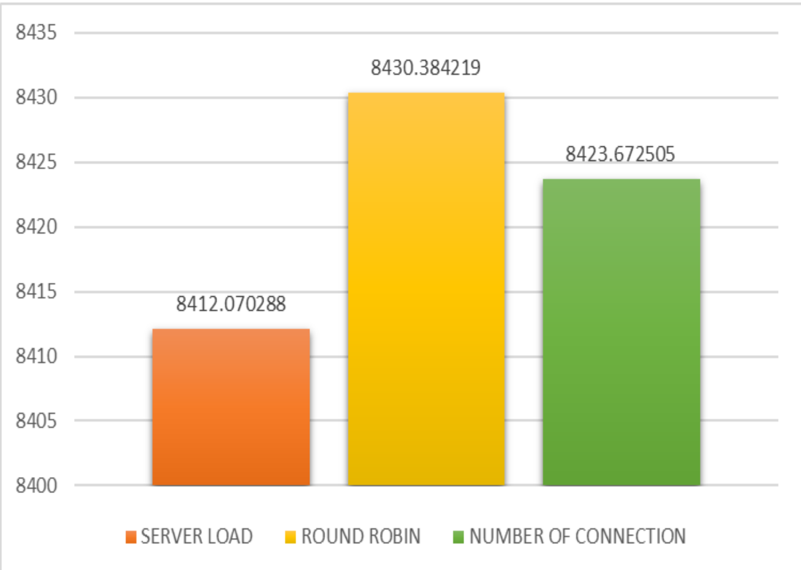

Fig 9. Total Throughput (packets/sec) -8 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

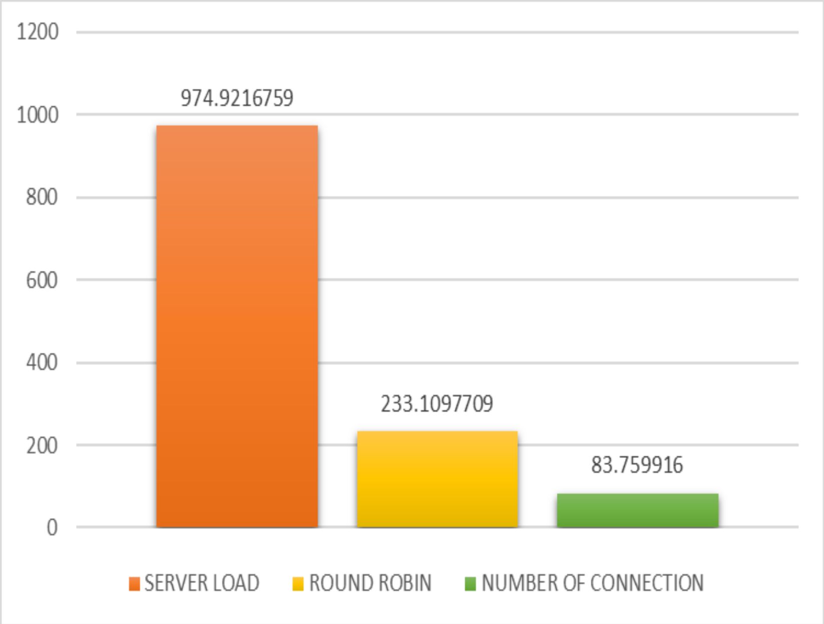

Fig 10. Servers throughput variation (packets) -8 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

C. *Twelve Servers*

The results of using 12 identical servers handling 10000 clients for multiple applications (http+ DB+ FTP), as shown in Figure 11 and Table 3, the Round Robin algorithm and Number of Connection have the high throughput (8437.347007 packet/second), , and the less throughput is Server Load algorithm with (7880.825542 packet/second) , and the variation between servers in throughput as show in figure (12), the Round Robin and Number of Connection algorithm are less, (109.883313 packets), the Server Load algorithm has high variation when it compares with Round Robin and Number of Connection, because Server Load algorithm need more process to check the status of the server to know it load, because of that there is some delay which affect the throughput of this algorithm comparing to Round Robin and Number of Connection algorithms.

TABLE 3. TOTAL THROUGHPUT (PACKETS/SEC) -12 SERVERS – 10000 CLIENTS MULTIPLE APPLICATIONS (HTTP-DB-MAIL)

	SERVER LOAD	ROUND ROBIN	NUMBER OF GONNECTIONS
TOTAL	7880.825542	8437.347007	8437.347007
MIN	621.2090073	636.5547167	636.5547167
MAX	798.1365945	746.4380297	746.4380297
VAR	176.9275872	109.883313	109.883313

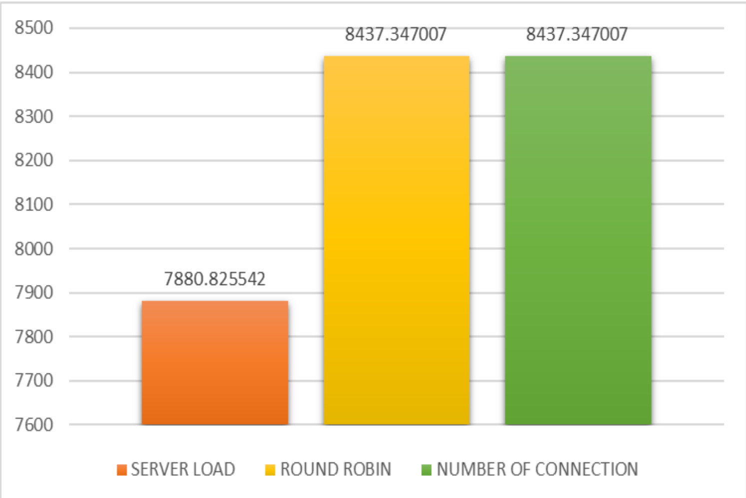

Fig 11. Total Throughput (packets/sec) -8 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

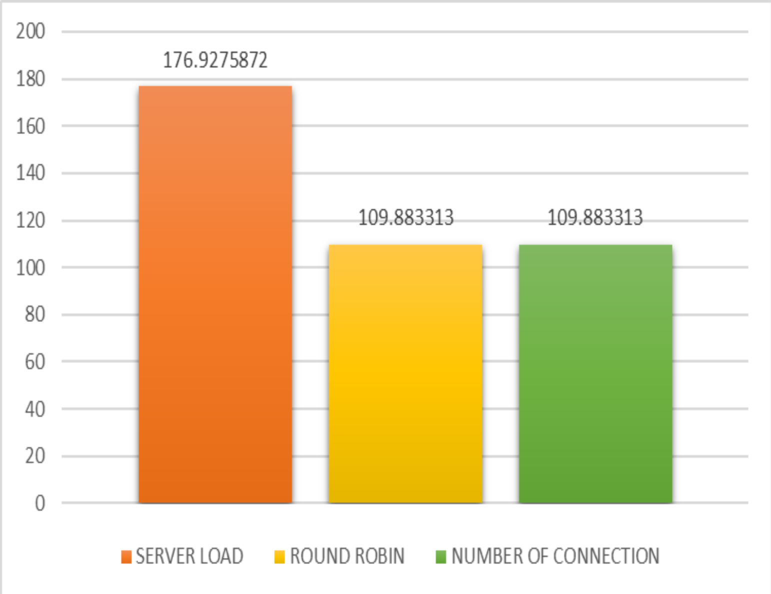

Fig 12. servers throughput variation (packets) -12 SERVERS – 10000 clients, MULTIPLE APPLICATIONS (http-DB-MAIL)

X. CONCLUSION

The summary Table 4 indicates that Round Robin provides highest throughput regardless the number of servers, that means it is the suitable algorithm with multiple application networks. Number of Connection has poor performance in small number of servers but it has the same performance with Round Robin when the number of servers increased to 12 servers. The Server Load algorithm perform well with 4 servers and 8 servers but at 12 servers it has poor performance. In general Round Robin has best balance for multiple application networks, stability, scalability, and throughput efficiency.

Table 5 shows that; the Number of Connection algorithm is best choice for big server cluster because it increases the throughput by about 16% when going from 4 to 12 servers. Round Robin is stable, but It does not improve throughput much, it is performance same regardless of the number of the servers. When it comes to Server Load Algorithm Table 5 shows that it is not work well when there are multiple application because it is throughput dropped by 6.4% when the number of servers reach 12.

TABLE 4. TOTAL THROUGHPUT (PACKETS/SEC) –SERVER LOAD ALGORITHM, ROUND ROBIN ALGORITHM AND NUMBER OF CONNECTION ALGORITHM- MULTIPLE APPLICATIONS (HTTP-DB-MAIL)

Number of servers	SERVER LOAD	ROUND ROBIN	NUMBER OF CONNECTION
4_ Servers	8386.018098	8401.28974	7038.705308
8_Servers	8412.070288	8430.384219	8423.672505
12_ Servers	7880.825542	8437.347007	8437.347007

TABLE 5. TOTAL THROUGHPUT INCREMENT PERCENTAGE –4 TO 8 AND FROM 8 TO 12

	SERVER LOAD	ROUND ROBIN	NUMBER OF CONNECTION
Increment percentage 4 to 8 servers	0.309700101%	0.345114502%	16.44137039%
Increment percentage 4 to 12 servers	-6.410401469%	0.42735313%	16.57679478%

REFERENCES

[1]. Bagus Wibowo , Abdul Haq , Mochamad Taufiqurrochman Abdul , *A Comparative study of load balancing methods in cloud-based village information systems*, 2023 IEEE 7th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE).

[2]. Jin Guo, Mangui Liang, Zhiyu Liu, *A Load Balancing Algorithm Based on End-to-end Rerouting, 2021 IEEE International Conference on Electronic Technology*. Communication and Information (ICETCT)

[3]. Mustafa ElGili Mustafa, *LOAD BALANCING ALGORITHMS ROUND-ROBIN (RR), LEASTCONNECTION, AND LEAST LOADED EFFICIENCY*, GESJ: Computer Science and Telecommunications 2017|No.1(51)

[4]. Chenhao Gao, Hengyang Wu , *An Improved Dynamic Smooth Weighted Round-robin Load balancing Algorithm*, Journal of Physics: Conference Series, doi:10.1088/1742-6596/2404/1/012047

[5]. Valery P. Mochalov , Natal'ya Yu. Bratchenko , Gennady I. Linets , A Dynamic Load Balancing Method for Data Centers with Fractal Properties of Network Traffic, 2022 International Russian Automation Conference (RusAutoCon).

[6]. SHANTI SWAROOP MOHARANA , RAJADEEPAN D. RAMESH2 & DIGAMBER POWAR, ANALYSIS OF LOAD BALANCERS IN CLOUD COMPUTING, International Journal of Computer Science and Engineering (IJCSE) ISSN 2278-9960 Vol. 2, Issue 2, May 2013, 101-108

[7]. Sally F. Issawi , Alaa Al Halees , Mohammed Radi, *Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workload An Efficient Adaptive Load Balancing Algorithm for Cloud Computing Under Bursty Workloads, Engineering*, Technology & Applied Science Research Vol. 5, No. 3, 2015, 795-800

[8]. Shalu Rani, Dharminder Kumar, Sakshi Dhingra, *A review on dynamic load balancing algorithms*, 978-1-6654-6200-6/22/$31.00 ©2022 IEEE.

[9]. Trinugi Wira Harjanti , Hari Setiyani , Joko Trianto, *Load Balancing Analysis Using Round-Robin and LeastConnection Algorithms for Server Service Response Time*, Applied Technology and Computing Science Journal, Vol. V, No. II, December 2022, 40-49.

[10]. Saumendu Roy , Dr. Md. Alam Hossain, Sujit Kumar Sen , Nazmul Hossain & Md. Rashid Al AsifGlobal , *Measuring the Performance on Load Balancing Algorithms*, Journal of Computer Science and Technology Volume XIX Issue I Version I

[11]. Hadeeel Alazzam, Abdulsalam Alsmady, Wail Mardini, and Amira Enizat, *Load Balancing in Cloud Computing Using Water Flow-Like Algorithm*, DATA'19, December 2–5, 2019, Dubai, United Arab Emirates

[12]. Ajay Katangur , Somasheker Akkaladevi , Sadiskumar Vivekanandhan , *Priority Weighted Round Robin Algorithm for Load Balancing in the Cloud*, 2022 IEEE 7th International Conference on Smart Cloud (SmartCloud).

[13]. Vidya S.Handur , Prakash R.Marakumbi , *Response time analysis of dynamic load balancing algorithms in Cloud Computing,* 978-1-7281-6823-4/20/$31.00 c 2020 IEEE.

[14]. Madhavi Avhankar, Bhakti Shinde, *Comparitive analysis of dynamic load balancing algorithm in cloud computing, International Journal of Advanced Science and Research*, Volume 3; Special Issue 1; March 2018; Page No. 59-62

[15]. Chukwuneke, Chiamaka Ijeoma, Prof. Inyiama, Hyacinth C, Amaefule, Samuel, Onyesolu, Moses Okechukwu, Asogwa, Doris Chinedu, International Journal of Trend in Research and Development, Volume 6(6), ISSN: 2394-9333, Nov – Dec 2019

[16]. Abhay Kumar Agarwal, Atul Raj, *A New Static Load Balancing Algorithm in Cloud Computing*, International Journal of Computer Applications (0975 – 8887) Volume 132 – No.2, December2015

[17]. PAWAN KUMAR and RAKESH KUMAR, NITTTR, Chandigarh, *Issues and Challenges of Load Balancing Techniques in Cloud Computing*: A Survey , ACM Computing Surveys, Vol. 51, No. 6, Article 120. Publication date: February 2019.

[18]. K.Balaji , P.Sai Kiran , M.Sunil Kumar, *Load balancing in Cloud Computing: Issues and Challenges*, Turkish Journal of Computer and Mathematics Education, Vol.12 No.2 (2021), 3077 – 3084

[19]. Tasneem, R., Jabbar, M.A. (2022). *An Insight into Load Balancing in Cloud Computing*. In: Qian, Z., Jabbar, M., Li, X. (eds) Proceeding of 2021 International Conference on Wireless Communications, Networking and Applications. WCNA 2021. Lecture Notes in Electrical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-19-2456-9_113

[20]. HARIZI Bouabdellah, Dr. Ishak MEDDAH, *Load balancing in cloud computing*, MASTER THESIS, Higher School of Computer Science of Sidi Bel Abbès (ESI-SBA) 2020/2021

[21]. M. T. Priya, P. Bala Srinivas and, Sanjeev Kumar, *Challenges and Issues of Load Balancing Algorithms in Cloud System*, Advanced Production and Industrial Engineering R.M. Singari and P.K. Kankar (Eds.) © 2022 The authors and IOS Press

[22]. Nadeem Shah, Mohammed Farik, *Static Load Balancing Algorithms In Cloud Computing: Challenges & Solutions*, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 4, ISSUE 10, OCTOBER 2015

[23]. Hristoski , T. Dimovski , Z. Kotevski , R. Markoska and N. Rendevski, *Performance Evaluation of Clustered Web Servers Using the Round Robin Scheme*, International Conference on Applied Internet and Information Technologies, 2018 , DOI:10.20544/AIIT2018.P09

[24]. S WP, P D. DServ- LB: *Dynamic server load balancing algorithm*. Int J Commun Syst. 2018;e3840. https://doi.org/10.1002/dac.3840.

[25]. Amandeep Kaur Sidhu , Supriya Kinger, *Analysis of Load Balancing Techniques in Cloud Computing*, International Journal of Computers & Technology, Volume 4 No. 2, March-April, 2013, ISSN 2277-3061

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden

[26]. Dmytro BILUSYAK, Oleh DOKUKA, *Research of traffic balancing algorithms and their implementation into RSocket*, UKRAINIAN CATHOLIC UNIVERSITY , Lviv 2023

[27]. Afiyah Rifkha Rahmika, Zulkifli Tahir, Ady Wahyudi Paundu, and Zahir Zainuddin, *Web Server Load Balancing Mechanism with Least Connection Algorithm and Multi-Agent System*, CommIT Journal 17(2), 245–258, 2023

[28]. Mohammad Mika Fawwazi, Eka Putra, Nadya Andhika Putri, *Evaluation and Comparison of Load Balancing Algorithm Performance in the Implementation of Weighted Least Connections and Round Robin in Cloud Computing Environment*, Journal of Computer Science, Information Technology and Telecommunication Engineering (JCoSITTE) Vol. 6, No. 1, March 2025, pp. 741~747

[29]. MuhammadAsimShahid, MuhammadMansoorAlam, and MazlihamMohdSu'ud, *Performance Evaluation of Load-Balancing Algorithms with Different Service Broker Policies for Cloud Computing*, Appl. Sci. 2023, 13, 1586. https://doi.org/10.3390/app13031586.

[30]. Bingyan Li, Weiliang Liu, Salim Nader, Jiaqi Song, Chengyu Zhang, Michal Aibin, *Cloud Load Balancing Algorithms Performance Evaluation Using a Unified Testing Platform*, Conference: International Conference on Computing, Networking and Communications (ICNC 2024)

© 2026, IJCSMC All Rights Reserved, ZAIN Publications, Fridhemsgatan 62, 112 46, Stockholm, Sweden