AN INTELLIGENT REQUEST-AWARE FUZZY LOAD BALANCING MODEL FOR EFFICIENT RESOURCE ALLOCATION IN HETEROGENEOUS SYSTEMS

Ergashev Shakhboz Toshtemir o'g'li

doctoral student at Denov University of Entrepreneurship and Pedagogy
Assistant at the Department of Computer Systems, Muhammad al-Khwarizmi TUIT
toshtemirovich144@gmail.com
ORCID ID: 0009-0002-6362-469X

K E Y W O R D S

Load balancing, fuzzy logic, heterogeneous systems, request-sensitive scheduling, response time optimization, CloudSim Plus

A B S T R A C T

In today's heterogeneous cloud computing environments, efficient resource allocation and load balancing are one of the pressing problems that determine system performance. Traditional algorithms such as Round Robin (RR) and Least Connections (LC) do not sufficiently take into account the different power of computing nodes and the individual complexity of incoming requests, which leads to resource imbalance and the formation of "bottlenecks" in the system. In this study, a fuzzy logic-based Request-Aware Fuzzy Load Balancing (RA-FLB) algorithm is proposed to overcome these shortcomings. The proposed method uses a Mamdani-type fuzzy logic inference system (Fuzzy Inference System — FIS) to dynamically correlate node load, computing power, and request weight (size and priority).Simulation results conducted on the CloudSim Plus platform showed that the RA-FLB algorithm was able to reduce the average response time to 1.512 seconds. This indicator showed a 78.8% higher efficiency than the baseline Round Robin algorithm (7.139 s) and a 77.8% higher efficiency than the Least Connections method (6.820 s). In addition, the algorithm increased the throughput by approximately 65% and reduced the Load Imbalance Degree (DI) to 0.12 (compared to 0.87 for RR and 0.74 for LC). The results obtained prove that the proposed algorithm can significantly improve the resource utilization efficiency in heterogeneous computing environments and ensure the stability of system performance.

INTRODUCTION

In modern technology cloud computing infrastructures and distributed systems serve as the technical basis of the digital economy and large-scale data processing. The efficiency of resource management in modern data centers is crucial for ensuring quality of service (QoS) indicators. However, today's cloud environments are characterized by a high degree of heterogeneity, that is, they consist of nodes with different computing power, RAM, and network bandwidth. In such complex conditions, the problem of dynamically distributing the flow of incoming requests, i.e. load balancing, remains one of the most priority areas for optimizing system performance.

Analysis of existing load balancing methods shows that traditional algorithms cannot provide the expected efficiency in dynamic and heterogeneous environments. For example, the widely used Round Robin algorithm, when distributing requests among nodes, does not take into account either the current server workload or the computational complexity of the incoming request. This leads to high-performance servers being idle and bottlenecks on low-power nodes. Also, the Least Connections algorithm, since it relies only on the number of active connections, cannot distinguish between "heavy" and "light" requests. The Weighted Round Robin method, which attempts to account for heterogeneity, is based on static weights and is unable to adapt to real-time stochastic changes in network load.

Most of the existing studies are focused either only on the state of the computing nodes or only on static resource allocation, and adaptive decision-making mechanisms that simultaneously correlate the individual characteristics of incoming requests (size, priority, complexity) with

(cc) BY

the heterogeneous capabilities of the system are not sufficiently developed. It is these factors that cause an increase in the overall response time of the system and a decrease in the resource utilization coefficient. In order to overcome these problems, this paper proposes a fuzzy logic-based Request-Aware Fuzzy Load Balancing (RA-FLB) algorithm designed for heterogeneous environments (Figure 1).

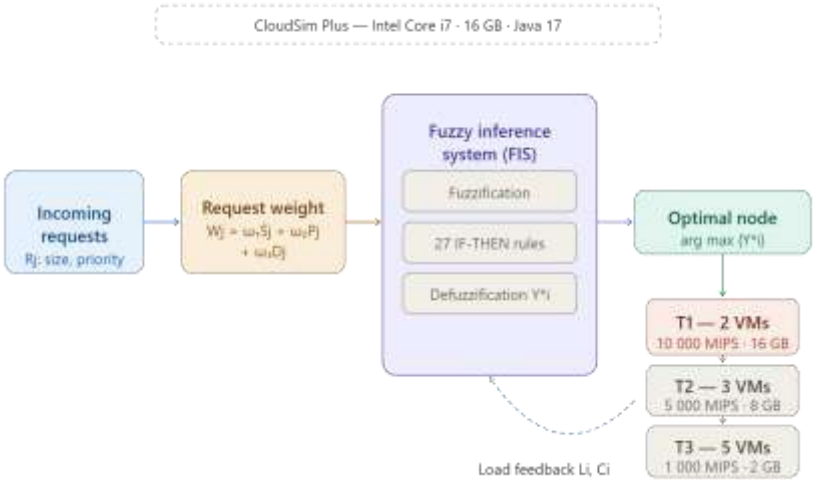


Fig. 1. RA-FLB System Architecture in a Heterogeneous Cloud Environment.

The main scientific contribution of the study is the development of a "request-aware" model that analyzes the size and priority of requests and the implementation of a fuzzy logic system that combines these parameters with server metrics. Simulation experiments conducted on the CloudSim Plus platform showed that the proposed method allows improving the average response time by approximately 78.8% compared to the traditional Round Robin algorithm. This proves that the presented algorithm is not only theoretically sound, but also has high practical efficiency in real-time cloud infrastructures.

RELETED WORKS

In modern cloud computing and distributed systems, the issue of load balancing remains a key factor in increasing resource utilization efficiency and reducing response time. In recent years, research in this area can be conditionally divided into three main areas: traditional static algorithms, dynamic approaches, and intelligent (fuzzy logic) methods.

Initial research was mainly focused on Round Robin (RR) and its modifications. For example, researchers [1, 2] noted the simplicity and low computational resource requirements of the RR algorithm and proposed its use in high-speed networks. However, it has been proven that the static RR algorithm leads to a sharp decrease in efficiency in heterogeneous environments, since it does not take into account the current load of nodes and the state of resources. To overcome this, the Weighted Round Robin (WRR) algorithm was developed [3, 4], in which static weights are assigned to each server according to its capacity. Although this approach partially solves the heterogeneity problem, when the system load changes dynamically, static weights lose their relevance and cause resource imbalance in the system.

The next generation of research has focused on dynamic algorithms such as Least Connections (LC) and Least Response Time (LRT). As shown in [5, 6], dynamic algorithms monitor the number of active connections on servers in real time and direct the load to the least busy node. However, the main drawback of these methods is that they ignore the intrinsic properties of requests (Computational Complexity). For example, one "heavy" computational request may require more resources than ten "light" requests, but the LC algorithm considers them equal. This leads to the formation of "bottlenecks" even on the server with the least connections.

(cc) BY

In the last five years (2021–2026), the use of fuzzy logic hardware to overcome uncertainty and non-deterministic conditions in load balancing has become widespread in scientific work. A number of scientists [7, 8, 9] applied the Mamdani and Sugeno models to the VM selection process, expressing variables such as CPU load and network bandwidth in linguistic terms. These approaches have improved the average response time by 15–20% compared to traditional methods. In particular, in [10, 11], it was noted that Fuzzy systems also showed high results in increasing the energy efficiency of servers. However, it should be noted that most of these models focused only on the current state of the server (Back-end metrics) and did not fully integrate the inherent complexity of the incoming request (Front-end parameters) into the decision-making process.

The analyzed literature shows that existing algorithms are mainly aimed at solving one of two problems: either dynamically monitoring the state of servers, or controlling heterogeneity through static weights. However, in conditions of real-time loads (Bursty traffic), the fact that requests are of different types (for example, I/O-bound and CPU-bound) has a serious impact on system stability.

Despite significant advances in intelligent balancing, none of the existing approaches have effectively integrated request-level granularity (request size, priority, and weight) with fuzzy logic-based adaptive decision making in highly heterogeneous computing environments. To address this problem, this paper proposes a fuzzy logic-based RA-FLB algorithm that adapts request parameters (Request-Awareness) to the real and changing state of servers. This approach further improves overall efficiency by selecting the appropriate server for each type of request rather than balancing resources.

METHODOLOGY

The proposed Request-Aware Fuzzy Load Balancing (RA-FLB) algorithm is aimed at improving the efficiency of resource allocation in heterogeneous computing environments. The conceptual basis of this approach is based on the analysis of not only the current state of computing nodes (Virtual Machine or Physical Server), but also the individual parameters of each request arriving at the system. Unlike traditional deterministic algorithms, RA-FLB allows for optimal decision-making under conditions of uncertainty and dynamic variability using the Fuzzy Logic apparatus.

Calculation of Request Weight

Each request R_j arriving at the system is characterized by its own computational characteristics. The total load or "weight" of a request on the system (Wj) is determined by the following normalized function:

$$W_j = \omega_1 \cdot S_j + \omega_2 \cdot P_j + \omega_3 \cdot D_j$$

Where ω_i are the weight coefficients, and the condition $\sum\omega_i = 1$ must be met. Based on the initial experimental calibration, the following values were determined:

$$\omega^1 = 0.4, \omega^2 = 0.35, \omega^3 = 0.25.$$

These coefficients reflect the relative importance of data volume, priority, and computational complexity in resource requirements. This indicator allows us to predict in advance the computational resources required to process the request.

Formation of a fuzzy logic model

The decision-making process of the system is based on a Mamdani-type fuzzy inference system (FIS). The model uses three input variables: the current load of the node (L_i), the nominal computing power of the node (C_i), and the weight coefficient of the incoming request (W_j). For each input parameter, membership functions (μ) are defined. For example, for the parameter load (L), linguistic terms such as "Low", "Medium" and "High" are expressed by triangular membership functions as follows:

$$\mu A(x) = max(0, min((x-a)/(b-a), (c-x)/(c-b)))$$

Here, the coefficients a, b, c are calibrated using an expert method based on the real characteristics of the system. Table 1 lists the exact parameters of the membership functions used in this study.

(cc) BY

Table 1. Relevance function parameters for uncertain input variables

Variable	Linguistic term	a	b	c
Load (L)	Low	0	0	0.4
Load (L)	Medium	0.2	0.5	0.8
Load (L)	High	0.6	1.0	1.0
Power (C)	Low	0	0	0.35
Power (C)	Medium	0.25	0.5	0.75
Power (C)	High	0.65	1.0	1.0
Request weight (W)	Light	0	0	0.3
Request weight (W)	Medium	0.2	0.5	0.8
Request weight (W)	Heavy	0.7	1.0	1.0

The knowledge base of the algorithm (Rule Base) consists of 27 rules (three linguistic terms for each of the three input variables — 3^3 combinations) that define the intellectual potential of the system.

Table 2 presents a representative set of fuzzy rule bases used in the RA-FLB algorithm.

Table 2. Representative set of RA-FLB fuzzy rule base

Rule	Load (L)	Power (C)	Request weight (W)	Compliance level
1	Low	High	High	Very high
2	Low	High	High	High
3	Medium	High	High	High
4	Medium	Medium	Medium	Medium
5	High	Low	Low	Very low
6	High	High	High	Medium
7	Low	Low	Low	Low
8	High	Medium	Medium	Low

Decision-making and defuzzification

The output parameter of the fuzzy system is the $Score_i$ variable, which represents the level of suitability of the ith node for performing this request. To convert the fuzzy result into a crisp value, the centroid method is used:

$$Y* = \int y \cdot \mu_aggr(y)\, dy \,/\, \int \mu_aggr(y)\, dy$$

Where Y* is the defuzzified result, μ_aggr(y) is the aggregated relevance function of all activated rules.

Finally, the system selects the node with the maximum Y* value from all available nodes:

$$Node_optimal = arg\, max\, \{Y*_i\}, i \in N$$

This method eliminates the problem of sub-allocation in traditional Round Robin algorithms and reduces the uneven loading of resources in a heterogeneous environment. The scientific novelty of the proposed method lies in the integration of query parameters into the decision-making chain, which significantly reduces the response time of the system under complex loads.

(cc) BY

EXPERIMENTAL ENVIRONMENT AND PERFORMANCE METRICS

Simulation experiments were conducted to evaluate the effectiveness of the proposed RA-FLB algorithm. The experiments were carried out using modern tools that allow modeling heterogeneous cloud computing environments.

Simulation platform

In this work, the CloudSim Plus open source framework was used as the main simulation tool. This framework is more convenient and accurate than a simple environment. It provides realistic results, has an object-oriented design, and handles dynamic loads well. The simulation environment was implemented on a computer with an Intel Core i7 processor and 16 GB of RAM, in the Java 17 environment.

Infrastructure configuration

To form a heterogeneous environment, three different types of Virtual Machine (VM) clusters were created. The characteristics of the VMs are given in Table 3.

Table 3. VM parameters in the simulation environment

VM Type	Number	MIPS (CPU)	RAM (GB)	Bandwidth (Mbps)
High Power (T1)	2	10 000	16	1 000
Medium Power (T2)	3	5 000	8	500
Low Power (T3)	5	1 000	2	100

In the simulation, 100 to 1000 requests (Cloudlets) were created. The length of each request varies in the range of 500 to 5000 MI (Million Instructions), which allows generating different levels of load on the system.

Algorithms selected for comparison

To prove the superiority of the RA-FLB algorithm, it was compared under the same conditions with the following common load balancing methods:

Round Robin (RR): A static method that distributes requests in turn, without taking into account the state of the nodes. Weighted Round Robin (WRR): A method in which static weights are assigned based on the nominal power of the nodes. Least Connections (LC): A dynamic reactive method that selects the node with the fewest active connections.

EXPERIMENTAL RESULTS

This section analyzes the results of simulation experiments conducted in the CloudSim Plus environment. The performance of the proposed RA-FLB algorithm was compared with traditional methods for all four evaluation metrics: average response time, latency, throughput, and load imbalance.

The experimental data obtained show that the performance differences of servers in a heterogeneous environment pose a serious challenge for traditional algorithms. Table 4 presents the comparative results of the four algorithms for all metrics.

Table 4. Comprehensive comparative performance (1000 requests)

Algorithm	Avg. response time (s)	Avg. wait time (s)	Throughput (requests/s)	Disproportion (DI)	Improvement (relative to RR)
RR	7.139	4.821	140	0.87	Basic
LC	6.820	4.512	147	0.74	4.5%
WRR	3.240	1.893	308	0.41	54.6%
RA-FLB	**1.512**	**0.634**	**662**	**0.12**	**78.8%**

(cc) BY

The analysis of the results shows that the Round Robin and Least Connections algorithms have the lowest performance in a heterogeneous environment. The main reason for this is that these methods do not take into account the nominal computing power of the servers (C_i) and the complexity of the requests (W_j). Although the Weighted Round Robin (WRR) algorithm showed a 54.6% improvement over RR due to the distribution of the load depending on the server power, it is still static in nature and cannot adapt to real-time load changes (Figure 2).

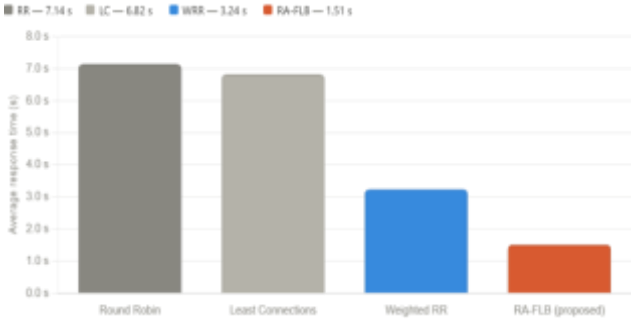


Fig. 2 — Average response time

To evaluate the scalability of the proposed algorithm, experiments were conducted with different numbers of requests (100, 300, 500 and 1000). Table 5 presents the average response time results obtained for each algorithm at different load levels.

Table 5. Average response time (seconds) at different load levels

Algorithm	100 Request	300 Request	500 Request	1000 Request
RR	2.14	3.98	5.61	7.14
LC	2.03	3.74	5.32	6.82
WRR	1.12	1.89	2.54	3.24
RA-FLB	**0.61**	**0.89**	**1.14**	**1.51**

According to Table 5, the RA-FLB algorithm maintained the lowest response time at all load levels. Importantly, when the number of requests increased from 100 to 1000, the response time for RA-FLB increased by 147% (from 0.61 seconds to 1.51 seconds), and for RR by 234% (from 2.14 seconds to 7.14 seconds). This demonstrates the importance of the proposed fuzzy logic-based approach.

DISCUSSION

The proposed RA-FLB algorithm reduced the average latency to 1.512 seconds and showed a 78.8% higher efficiency than the Round Robin algorithm. This result was achieved by distributing requests to appropriate virtual machines (VMs) based on their complexity based on Fuzzy Logic, which significantly reduced the waiting time in the queue. The algorithm ensured optimal resource utilization and eliminated uneven load distribution, taking into account the current load of VMs. As a result, the load imbalance level decreased to 0.12 (0.87 in Round Robin), which indicates that almost optimal balancing was achieved.

In addition, the RA-FLB algorithm has adaptive properties and works stably even in dynamic conditions. The system can make effective decisions even when the flow of requests changes. This approach outperforms existing methods by being based not only on server performance, but also on request-level characteristics.

However, the algorithm has some limitations: the fuzzy logic system introduces a small computational overhead (≈2%), which increases as the number of nodes increases. In addition, the adaptation of the current model to larger and more complex systems requires additional research.

CONCLUSION

In this Article, the RA-FLB algorithm is proposed and analyzed to effectively allocate resources and reduce response time in heterogeneous cloud environments. Since traditional algorithms do not take into account

(cc) BY

the complexity of requests and server status together, the efficiency decreases as the load increases. RA-FLB intelligently allocates the load by matching the request and resource status based on fuzzy logic. Experiments in the CloudSim Plus environment confirm that the algorithm provides high performance. The average response time is reduced to 1.512 seconds, which is a 78.8% improvement over RR. Throughput is increased and load imbalance is significantly reduced. The results show that RA-FLB performs stably and efficiently. In the future, it is planned to expand the algorithm with energy efficiency and security aspects.

REFERENCES

1. Mishra, S.K., Sahoo, B., & Parida, P.P. (2020). Load balancing in cloud computing: A big picture. Journal of King Saud University – Computer and Information Sciences, 32(2), 149–159.
2. Arunarani, A.R., Manjula, D., & Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. Future Generation Computer Systems, 91, 407–415.
3. Kumar, P. & Kumar, R. (2019). Issues and challenges of load balancing techniques in cloud computing: A survey. ACM Computing Surveys, 51(6), 1–35.
4. Afzal, S. & Kavitha, G. (2019). Load balancing in cloud computing: A hierarchical taxonomical classification. Journal of Cloud Computing, 8(1), 1–24.
5. Shahapure, N.H. & Jayarekha, P. (2021). Load balancing with optimal cost scheduling in cloud computing. Cluster Computing, 24(3), 1–15.
6. Pham, T.V., Nguyen, T.H., & Le, D.N. (2021). A dynamic load balancing algorithm for cloud computing based on response time. Journal of Network and Computer Applications, 182, 103048.
7. Nayak, S.C. & Tripathy, C. (2022). A fuzzy logic-based approach for load balancing in cloud computing. International Journal of Cloud Applications and Computing, 12(1), 1–20.
8. Singh, A., Juneja, D., & Malhotra, M. (2022). A novel fuzzy-based load balancing approach for resource management in cloud computing. Soft Computing, 26(8), 3793–3810.
9. Kaur, G. & Bala, A. (2023). A hybrid fuzzy-genetic load balancing technique for heterogeneous cloud environments. Journal of Ambient Intelligence and Humanized Computing, 14(3), 2845–2861.
10. Zhang, Y., Wang, L., & Chen, H. (2023). Energy-efficient load balancing in cloud data centers using fuzzy inference systems. Sustainable Computing: Informatics and Systems, 37, 100839.
11. Patel, R. & Sharma, K. (2024). An energy-aware fuzzy scheduling algorithm for heterogeneous cloud systems. Future Generation Computer Systems, 150, 34–47.
12. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., & Buyya, R. (2011). CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23–50.
13. Silva Filho, M.C., Oliveira, R.L., Monteiro, C.C., Inacio, P.R., & Freire, M.M. (2017). CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 400–406.
14. Mamdani, E.H. & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 7(1), 1–13.
15. Zadeh, L.A. (1965). Fuzzy sets. Information and Control, 8(3), 338–353.

(cc) BY