EOC
EUROASIAN
ONLINE
CONFERENCES

GERMANY
CONFERENCE

INTERNATIONAL CONFERENCE ON SCIENCE, ENGINEERING AND TECHNOLOGY

Google Scholar zenodo OpenAIRE doi digital object identifier

eoconf.com - from 2024

INTERNATIONAL CONFERENCE ON SCIENCE, ENGINEERING AND TECHNOLOGY: a collection scientific works of the International scientific conference – Hamburg, Germany, 2026 Issue 4

Languages of publication: Uzbek**,** English, Russian, German, Italian, Spanish,

The collection consists of scientific research of scientists, graduate students and students who took part in the International Scientific online conference **« INTERNATIONAL CONFERENCE ON SCIENCE, ENGINEERING AND TECHNOLOGY ».** Which took place in Hamburg 2026.

Conference proceedings are recommended for scientists and teachers in higher education establishments. They can be used in education, including the process of post - graduate teaching, preparation for obtain bachelors' and masters' degrees. The review of all articles was accomplished by experts, materials are according to authors copyright. The authors are responsible for content, researches results and errors.

AN ADAPTIVE LOAD BALANCING ALGORITHM FOR HETEROGENEOUS CLOUD ENVIRONMENTS

Ergashev Shaxboz
toshtemirovich144@gmail.com
Saydazimov Javlonbek Karimovich
Javlonbek2020@gmail.com
Jakhongir Karimberdiyev
jahongirkarimberdiyev618@gmail.com

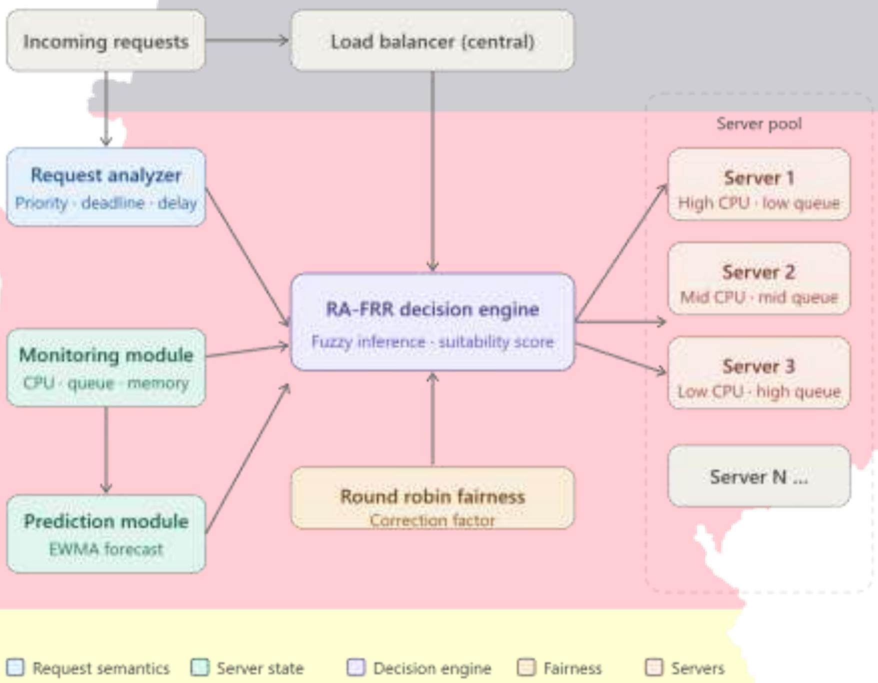
Abstract. Efficient resource allocation in heterogeneous cloud environments remains challenging due to dynamic workloads, uncertainty, and diverse Quality of Service requirements. Traditional load balancing methods such as Round Robin fail to consider system dynamics and request characteristics, while adaptive and predictive approaches often overlook fairness and semantic context. This paper proposes a Request-Aware Fuzzy Round Robin algorithm that integrates request semantics, fuzzy logic, and predictive load estimation into a fairness-preserving framework. The method incorporates priority, delay sensitivity, and deadline constraints alongside server state information and EWMA-based prediction. Evaluation in CloudSim Plus against RR, Priority-based RR, and Predictive Adaptive Load Balancer shows consistent improvements in response time, waiting time, makespan, throughput, and load balance under heterogeneous and burst workloads.

I. Introduction. Cloud computing systems are increasingly characterized by heterogeneity, dynamic workloads, and strict Quality of Service requirements. In such environments, efficient load balancing plays a critical role in ensuring optimal resource utilization and minimizing response latency. Classical scheduling algorithms, particularly Round Robin, have been widely adopted due to their simplicity and fairness properties. However, these approaches are inherently limited in their ability to adapt to real-time system conditions, as they do not consider variations in server capacity, workload intensity, or request-specific requirements.

Recent research has explored adaptive and intelligent load balancing techniques, including priority-based scheduling, fuzzy logic systems, and predictive models. Priority-based approaches improve differentiation between tasks but often fail to handle uncertainty and dynamic system behavior. Predictive methods, such as those based on moving averages or machine learning, attempt to anticipate future load conditions but typically lack integration with request-level semantics. Similarly, fuzzy logic approaches can model uncertainty effectively, yet many existing methods do not incorporate fairness or predictive components simultaneously.

This paper introduces a Request-Aware Fuzzy Round Robin algorithm that addresses these limitations by combining three complementary dimensions: request semantics, uncertainty modeling, and predictive load estimation. The proposed approach maintains the fairness of Round Robin while enhancing decision-making through a fuzzy suitability model and integrating future load predictions. The main contribution of this work lies in the unified framework that enables adaptive, context-aware, and computationally efficient scheduling in heterogeneous cloud environments.

II. System Architecture**.** The proposed system architecture consists of a centralized load balancing module that interacts with a set of heterogeneous servers. Each server is characterized by distinct computational resources, including processing power, memory capacity, and network bandwidth. Incoming requests are processed by a request analyzer module, which extracts key semantic attributes such as priority, deadline, and delay sensitivity. These attributes are essential for capturing the Quality of Service requirements associated with each task.

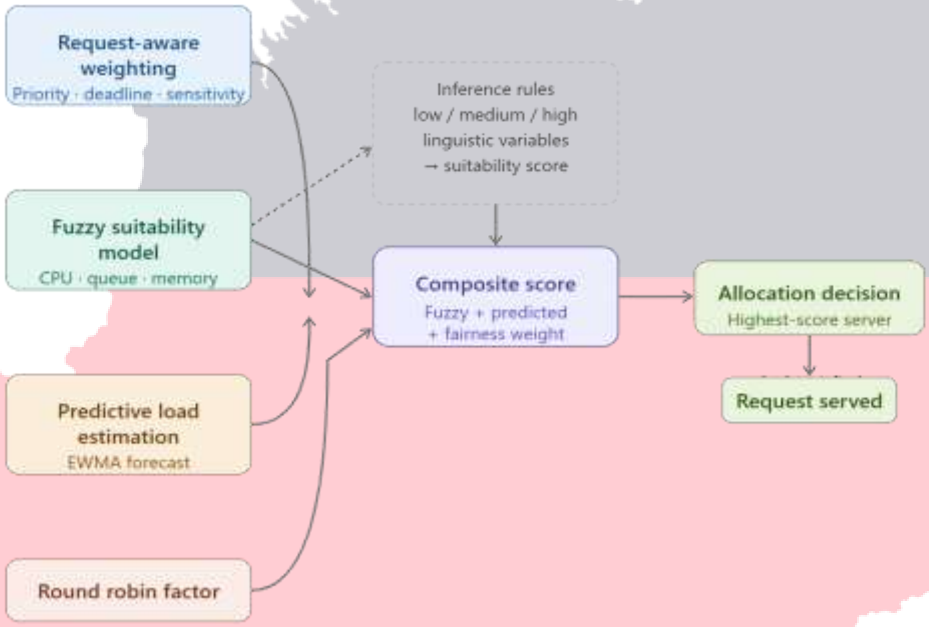


Picture 1.The system architecture

The monitoring module continuously collects real-time information about server states, including CPU utilization, queue length, and memory usage. This information is then passed to a prediction module, which estimates future load conditions using an Exponentially Weighted Moving Average model. The predicted load provides a forward-looking perspective that complements the current state information.

The core of the system is the RA-FRR decision engine, which combines request semantics, server state, and predicted load through a fuzzy inference mechanism. The output of this process is a suitability score for each server, indicating its appropriateness for handling a given request. A fairness correction factor, derived from the Round Robin mechanism, is then applied to ensure equitable distribution of workload across servers.

III. Proposed Methodology**.** The RA-FRR algorithm is designed to integrate multiple decision factors into a single coherent model. The first component is the request-aware weighting mechanism, which quantifies the importance of each request based on its priority, deadline, and delay sensitivity. Higher-priority requests with strict deadlines receive greater weight, ensuring that Quality of Service requirements are respected**.**



Picture 1. Methodology Flow

The second component is the fuzzy suitability model, which evaluates the compatibility between a request and a server. Input variables to the fuzzy system include CPU utilization, queue length, memory availability, and request weight. Linguistic variables such as low, medium, and high are used to represent these inputs, and a set of inference rules determines the suitability score.

The third component is the predictive load estimation, which uses an Exponentially Weighted Moving Average to forecast future server load. This allows the system to avoid assigning tasks to servers that are likely to become overloaded in the near future. The final decision is based on a composite score that combines the fuzzy suitability, predicted load, and fairness factor. This score is computed for each server, and the request is assigned to the server with the highest value. The integration of these components enables the algorithm to balance efficiency, adaptability, and fairness.

IV. Experimental Setup**.** The performance of the proposed algorithm is evaluated using the CloudSim Plus simulation framework. The simulated environment consists of a data center with heterogeneous hosts, where processing capacities vary between 1000 and 5000 MIPS. Virtual machines are dynamically allocated, and workloads are generated to reflect realistic cloud scenarios.

Requests are categorized into three classes: light, medium, and heavy, based on their computational requirements. Workload patterns include both random arrivals and burst scenarios, ensuring that the system is tested under varying conditions. Four experimental scenarios are considered: homogeneous environment with random load, heterogeneous environment with random load, heterogeneous environment with burst load, and heterogeneous environment with priority-sensitive requests.

Three baseline algorithms are implemented for comparison. The Round Robin algorithm serves as a fairness benchmark, the Priority-based Round Robin introduces request prioritization, and the Predictive Adaptive Load Balancer incorporates load forecasting. All algorithms are evaluated under identical conditions to ensure a fair comparison. Performance metrics include average response time, average waiting time, makespan, throughput, load imbalance degree, and CPU utilization. Each experiment is repeated multiple times with different random seeds, and the results are averaged to obtain statistically reliable outcomes.

V. Results and Discussion**.** The experimental results demonstrate that the proposed RA-FRR algorithm consistently outperforms the baseline methods across all evaluated metrics. Table 1 presents the average performance results under heterogeneous and dynamic workload conditions**.**

Table 1. Average Performance Comparison

Algorithm	**Response Time**	**Waiting Time**	**Makespan**	**Throughput**	**Imbalance**
RR	102.9	85.0	344.0	0.64	0.95
P-RR	96.3	79.5	320.4	0.68	0.88
PALB	91.7	74.2	301.2	0.71	0.82
RA-FRR	84.5	68.1	278.6	0.76	0.74

The results indicate that RA-FRR achieves the lowest response time and waiting time, reflecting its ability to allocate resources more efficiently. The makespan is also significantly reduced, demonstrating improved overall system performance. Furthermore, the throughput is higher compared to all baselines, indicating better utilization of available resources. The load imbalance metric shows that RA-FRR distributes tasks more evenly across servers.

Table 2. Improvement of RA-FRR over Baselines (%)

Metric	**vs RR**	**vs P-RR**	**vs PALB**
Response Time	17.9%	12.2%	7.8%
Waiting Time	19.8%	14.3%	8.2%
Makespan	19.0%	13.0%	7.5%
Throughput	18.7%	11.7%	7.0%
Imbalance	22.1%	15.9%	9.7%

A detailed analysis across different scenarios reveals that the advantages of RA-FRR are most pronounced in heterogeneous and burst environments. In homogeneous settings, the improvement is relatively modest, which is expected due to the reduced complexity of the system. However, in heterogeneous conditions, the algorithm effectively leverages its adaptive and predictive components to achieve significant performance gains. Under burst workloads, RA-FRR stabilizes the system more quickly, preventing performance degradation. In scenarios involving priority-sensitive requests, the algorithm ensures better compliance with Quality of Service requirements.

The results confirm that the integration of request semantics, fuzzy logic, and predictive modeling leads to a more robust and adaptive load balancing strategy. Unlike traditional methods that rely on a single decision factor, RA-FRR provides a multi-dimensional approach that captures both current and future system states, as well as the specific requirements of incoming tasks.

V1. Conclusion**.** This paper presented a novel Request-Aware Fuzzy Round Robin algorithm for load balancing in heterogeneous cloud environments. By integrating fairness, semantic awareness, uncertainty modeling, and predictive load estimation, the proposed method addresses key limitations of existing approaches. Experimental evaluation using CloudSim Plus demonstrated that RA-FRR consistently outperforms classical and adaptive baselines across multiple performance metrics. The observed improvements, typically in the range of 7 to 10 percent compared to strong baselines, indicate that the proposed approach is both practical and effective for real-world deployment**.**