ISSN: 2182-2069 / E-ISSN: 2182-2077

AI-Based Security: Effective Load Balancing and Optimum Distribution of Resources for Enhancing Financial Cloud

Dr. Amaresh Sahu[1*], R. Murugan[2], Dr. Sadaf Hashmi[3], Dr. Arun Kumar Marandi[4], Amit Kumar[5], and Sachin S. Pund[6]

[1*]HOD and Associate Professor, Master in Computer Application, Ajay Binay Institute of Technology, Odisha, India. amaresh_sahu@yahoo.com, https://orcid.org/0009-0005-9618-1380

[2]Associate Professor, Department of Computer Science and Information Technology, Jain (Deemed to be University), Bangalore, Karnataka, India. murugan@jainuniversity.ac.in, https://orcid.org/0000-0003-0903-5982

[3]Associate Professor, Department of ISME, ATLAS SkillTech University, Mumbai, Maharashtra, India. sadaf.hashmi@atlasuniversity.edu.in, https://orcid.org/0009-0008-8729-8423

[4]Associate Professor, Department of Computer Science & IT, ARKA JAIN University, Jamshedpur, Jharkhand, India. dr.arun@arkajainuniversity.ac.in, https://orcid.org/0000-0003-1507-7580

[5]Centre of Research Impact and Outcome, Chitkara University, Rajpura, Punjab, India. amit.kumar.orp@chitkara.edu.in, https://orcid.org/0009-0001-9561-2768

[6]Assistant Professor, Mechanical Engineering, Ramdeobaba University, RBU, Nagpur, Maharashtra, India. pundss@rknec.edu, https://orcid.org/0000-0002-5616-2469

Received: January 10, 2025; Revised: February 21, 2025; Accepted: April 04, 2025; Published: May 30, 2025

Abstract

Financial cloud computing is based on the virtualization technology paradigm, which has recently gained prominence in the information technology (IT) industry. The resource allocation in the existing system is not guaranteed, and convergence problems occasionally cause the process to move slowly. Consequently, there has been a notable decline in the overall efficacy of financial cloud computing. In this work, new methods are presented to improve the financial cloud. The three main stages of the proposed system are resource allocation, load balancing and cost-effective virtual machine (VM) migration. To enhance load balancing, this study employs the Enhanced Weighted Round Robin Algorithm (EWRR) technique. Load balancing is achieved by shifting workloads from overloaded to underloaded nodes. The Differential Evaluation-based Bat Algorithm (DEBA) efficiently chooses more optimal resources to accomplish the optimal resource allocation in the financial cloud. Resolute Support Vector Machine (RSVM) technique is utilized to Provide Cost-effective VM migration. The simulation's findings show that the recommended DEBA-EWRR algorithm performs better than existing techniques due to advancements in throughput, Makes pan, Fitness score and resource utilization.

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 2 (May), pp. 18-29.
DOI: *10.58346/JISIS.2025.I2.002*

*Corresponding author: HOD and Associate Professor, Master in Computer Application, Ajay Binay Institute of Technology, Odisha, India.

Keywords: Virtualization, Enhanced Weighted Round Robin Algorithm (EWRR), Resolute Support Vector Machine (RSVM), Financial Cloud computing, Differential Evaluation-based Bat Algorithm (DEBA).

1 Introduction

Public and private services can be accessed using the technology known as financial cloud computing Chitra et al., (2023). Some of the services offer scalable online storage services in place of physically stored files on users' computers or phones and easy access to data, applications and files over the internet (Akhtar et al., 2021). Essentially, load balancing involves distributing network traffic that arrives at multiple servers to prevent any one of them from becoming overloaded (Chahlaoui et al., 2019). Financial cloud computing, which offers pay-per-use services to customers anywhere at any time, has replaced traditional computer technologies due to the explosive rise of information technology (IT) (Zhang et al., 2022). This provides users with quicker responses and more dependable service by guaranteeing that web pages and applications can handle high amounts of requests and do so properly (Mittal et al., 2021). For service providers, financial cloud computing offers significant benefits in terms of ease of use, energy efficiency and low resource management costs built on virtualization technology, which has no restrictions on computer resources (Bello et al., 2021). The “Internet of Things (IoT)” has made it achievable for billions of devices, such as actuators, sensors and other items, to connect to the Internet (Iyer & Trivedi, 2023). The enormous volumes of processable data produced by these devices could strain available processing and storage capacity (Guo et al., 2021). Storage and processing limitations have vanished with the introduction of financial cloud computing. Financial cloud computing workloads and data volumes have increased in the last ten years (Lynn et al., 2020). The purpose of this research is to enhance financial cloud computing performance through resource allocation, load balancing and cost-effective VM migration using DEBA, EWRR and RSVM approaches.

The following sections comprise the study. Section 2 includes the relevant work. Section 3 contains a detailed explanation of the procedures. The results of the study are shown in Section 4. There is a conclusion in Section 5.

2 Related Works

Brahmam, (2024) suggested a load-balancing & Energy-Efficient Migration Model that uses cutting-edge security features and bioinspired algorithms to optimize VM migrations (Wu, 2024). Jyoti & Shrimali, (2020) suggested a novel strategy involving load balancing & service brokering to enable dynamic resource supply. Load balancing was used by the Local User Agents to boost throughput. “Dynamic Optimal Load-Aware Service Broker (DOLASB)” was employed in the Universal Client Agency to plan tasks, and “Multi-agent Deep Reinforcement Learning-Dynamic Resource Allocation (MADRL-DRA)” was used for task allocation. A load-balancing strategy for maximizing data center resource usage and electricity consumption was given (Kumar et al., 2021). The approach used a genetic algorithm to try and get a rough estimate of the answer. The resources of the Cloud-Fog environments can be distributed according to the needs of Internet of Things devices connected at the cloud edge using a dependable scheduling scheme, as proposed (Alqahtani et al., 2021)(William et al., 2025). The approach accounts for resource load balancing by classifying requests are urgent, critical or time-tolerant.

Jena et al., (2022) presented a novel approach to dynamic balancing of load across VM using the hybridization of an improved Q-learning algorithm, known as QMPSO, and “modified particles swarm optimization (MPSO)." Devaraj et al., (2020) presented a revolutionary load-balancing algorithm called the "Firefly Improved Multi-Objective Particle Swarm Optimization (FIMPSO)" technique. The Firefly (FF) method was utilized to narrow down the search space, and the IMPSO technique was employed to locate the increased response. Li et al., (2020) proposed the replica allocation strategy and a resource management approach for an edge computing framework (Abdullah, 2024)(Rahim, 2024;Uvarajan, 2024). The replica management method consists of consistency preservation & replica allocation algorithms.

3 Methodology

In financial cloud computing, the Enhanced Weighted Round Robin Algorithm for Load balancing. The Differential Evaluation-based Bat Algorithm for resource allocation and the Resolute Support Vector Machine for providing the cost-effective VM migration are used in this work. Figure 1 displays the methods research used in this study.

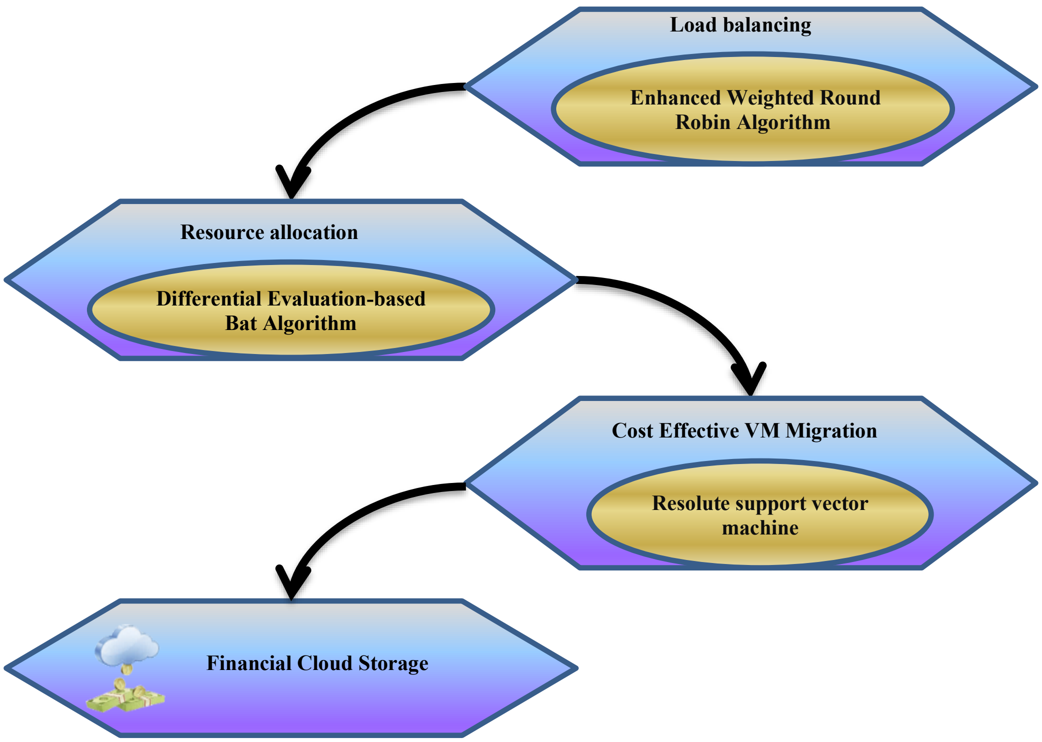

Figure 1: Flowchart of Suggested Methods

3.1 Architecture of the System

The system model used in this work has VMs, financial cloud users, CPU, memory use, tasks, and resources. A set of programs and apps is specifically designed to be executed by a financial cloud infrastructure to fulfil predetermined duties. Utilizing specific financial cloud resources is required for the implementation of these initiatives. The placement of VMs is crucial to maximizing the usage of data center resources. Every host in the data center can switch between the active and sleep states. The host does not have a sleep state assigned to it; instead, it is in the active state meaning which is reserved for functions involving execution. Based on variables like processing power, memory, & storage capacity, it is divided into numerous types of finite VMs.

3.2 Enhanced Weighted Round Robin Algorithm for Load Balancing

Providing an even distribution of the workload among the processing nodes is the load balancer's responsibility. Figure 2, shows the flow of the Enhanced Weighted Round Robin Algorithm. Load balancers are responsible for moving duties from overloaded VMs to less loaded ones, and if VMs detect any VMs to be overloaded. By managing and varying VM workloads in the network, the WRR load balance strategy, which enhances weighted robin algorithms, becomes the enhanced WRR load balance algorithm. The main problem with the WRR load balance method is always assigns the right VM to each incoming request, regardless of the user's request size. In order to determine which VM should be assigned to the next job to be completed, the suggested technique takes into account the load of VMs, processor capability and task duration. The enhanced WRR load balancing method consists of two parts: dynamic and static load balancing. While the dynamic load balancing portion uses the load on each VM and capacity information to determine the assignment of workloads to the proper VMs, the static method assigns VMs depending on their processing capability. The load balancer is in charge of determining whether VMs are underutilized, overloaded or idle. The load balancing cannot distribute tasks amongst occupied VMs. Based on a work that takes to complete, the Enhanced RR load balance algorithm determines which VMs to assign tasks or map incoming requests. The duration of an incoming task is used to compute the waiting period of the tasks, which is the first stage.

$$Waiting\ Period = \sum_{P=0}^{P} K * U \tag{1}$$

The waiting period of every VM is determined in Equation (1) by multiplying the Cloudlet length by the execution time. Equation (2) is utilized to determine the VM's waiting time upon receiving a request of length L.

$$Execution\ time = \frac{S}{VMp} \tag{2}$$

VM capacity is defined in IPS (million instructions per second), while S is evaluated in MI (million instructions). Once the load balancer has collected the task wait times from each of the virtual computers that have been built, it starts to prepare itself. The process of mapping involves selecting the task from VMs with the longest waiting periods, identifying the most suitable VM for the task from each VM list and allocating the incoming job to the selected VMs according to the waiting time. Rearranging the order is how load balancing works, and the waiting time is determined by the VM's load. Figure 2 depicts the step-by-step process for using the algorithm.

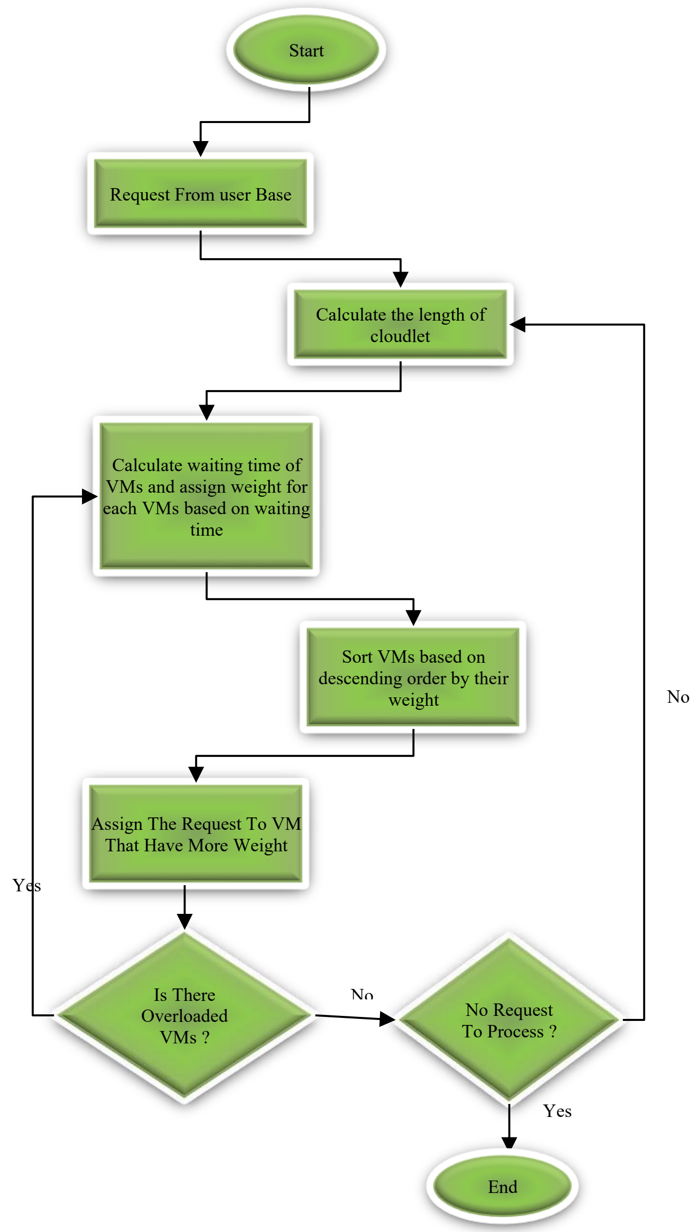

Figure 2: Flow chart for the Enhanced Weighted Round Robin Algorithm

3.3 Resource Allocation

In the financial cloud, to enhance the resource allocation process, research presented the novel method called Differential Evaluation-based Bat Algorithm.

3.3.1 Differential Evaluation (DE)

It is a population-based meta-heuristic approach. It mostly uses an evolutionary approach to enhance an ideal answer. This technique finds vast development zones quickly and requires few assumptions about fundamental optimal solutions. DE is a feature optimization technique that depends on the population and highlights individual differences. It primarily creates a community of hard issues to identify the optimal solution. Simply said, upon the initial condition, DE repeatedly performs crossover, mutation and selection procedures. For a given problem, the DE technique selects vectors of features with the highest fitness value and creates a path variable.

3.3.2 Bat Algorithm

A popular meta-heuristic based on nature that is used for global optimization issues. The echolocation patterns of miniature BATs with pulse & noise levels are the driving force behind these techniques. There are around a thousand species of BAT, with sizes varying from 1.6 mg to larger. Echolocation is usually used by BATs. They can function because they have good hearing but poor vision. They also identify insects at night using an echolocation approach. Pseudo code 1 depicts the Bat algorithm process.

Pseudo code 1. Bat Algorithm

```
Initialization: Let initial population Pxi, Pulse Prxi, Velocity
Vxi, frequency Fxi, Loudness Lxi
Begin
Set the maximum number of iterations and repeat till the max
While (Current_iteration < max)
Obtained a solution by changing the position, velocity, and frequency
If (random > Prxi)
Choose an optimal solution from all available solutions Produce a temporary solution
in front of the optimal solution End if,
If (random < Loudness Lxi) && (F (Pxi) < F (Px*)
Accept the new solution and increment Prxi and Lxi
End if,
Allocates ranking to all the Bats and obtained an optimal solution, Pxi
End while
Post-process the outcomes
End
```

3.3.3 Differential Evaluation-based Bat Algorithm

The Bat Algorithm (BA) framework is used in the "Differential Evaluation-based Bat algorithm," which provides a novel approach to resource allocation by leveraging Differential Evolution (DE) to improve performance. The goal of this hybridization is to solve problems with resource allocation optimization. Differential evolution integrates crossover and mutation processes to improve exploitation and exploration capabilities, while the bat algorithm offers a metaheuristic optimization technique based on the echolocation behavior of bats. The recommended method blends different techniques to maximize resource distribution. Following the initialization of the bat population, the algorithm carries out repetitive search operations where the bats interact with their surroundings and with one another, constantly shifting their locations in an attempt to optimize a target function that indicates resources are allocated. Such as Differential operators for crossover, mutations and selection, Differential Evaluation provides the bat search approach. The algorithm looks for the best answers by effectively allocating resources and striking a balance between exploration and exploitation through this synergistic combination.

3.4 Resolute Support Vector Machine for Cost-effective VM Migration

The following principles underlie the operation of support vector machine theory. Given n sample-label pairs ($S = (w_j, z_j), (i = 1, 2, .., n)$, a training set ($w_j \in R$) and a class label ($z_j \in \{-1, +1\}$) for the classification problem. The hyperplane is created using the formula $\omega^T w + b = 0$ (where $"b"$ is a bias factor and $"\omega"$ is the vector of hyperplane coefficients) to maximize the distance between the support vectors (the closest data points) and the hyperplane. To do this, we use the training of classifiers to find the hyperplane that can distinguish positive $(+1)$ values from negative (-1) data. One must solve an optimization issue to get the ideal separation hyperplane $\omega^T w + b = 0$.

$$\begin{gathered} min_{\omega,b} \ \frac{1}{2}\|\omega\|^2 \\ s.t.\ z_j\big(< \omega, w_j > +b\big) \geq 1 \end{gathered} \tag{3}$$

Finding a hyperplane that can accurately and fully separate data points in some applications is challenging, though. The prediction model's capacity to be generalized may be reduced by overfitting caused by a complex categorization hyperplane. The problem is rewritten as follows in order to get around this challenge.

$$\begin{gathered} min_{\omega,b,t} \ \frac{1}{2}\|\omega\|^2 + D\sum_{j=1}^{m}\varepsilon_j \\ s.t.\ z_j\big(< \omega, w_j > +b\big) \geq 1 - \varepsilon_j, j = 1,1, \dots, m. \end{gathered} \tag{4}$$

The term "slack variable” refers to ε_j and D It is a punishment coefficient.

But in real-world tasks, it happens that the data are partially separated. To solve the nonlinearity problem, research need to project the original data into a space with high dimensions using the nonlinear mapping Φ_w. In the high-dimensional space, the data may be separable linearly. The Lagrange technique can be used to solve both equation (3) and (4) in dual form. The dual form in a nonlinear instance is:

$$\begin{gathered} \min_{\alpha} \ \sum_j \alpha_j - \frac{1}{2}\sum_j\sum_i \alpha_j\,\alpha_i\,z_j\,z_i(\Phi(w_j), \Phi(w_i) >_E \\ s.t.\ 0 \leq \alpha_j \leq D, \sum_j \alpha_j\,z_j = 0 \end{gathered} \tag{5}$$

The Lagrange multiplier is represented by α_j. The ultimate decision-making function is

$$class\ (w) = sign\ (\textstyle\sum_j \alpha_j\,z_j =< \Phi(w_j), \Phi(\mathrm{w}) >_E + b \tag{6}$$

The inner result of $K(w, z) =< \Phi(w_j), \Phi(\mathrm{w}) >_E$ The definition of kernel operation, the fundamental unit of an SVM. Considering the support vector and the new data, the kernel determines an indication of similarity or difference. The dot product is the similarity metric used for a linear SVM or linear kernel because distance represents an exponential coefficient of the inputs. Other kernels, such as radial basis function and polynomial kernels, can be used to convert the space being input into higher dimensions in equation (5) and (6).

4 Result

To assess the effectiveness of the proposed methodology, use quantitative indicators such as Fitness score, Makespan and resource utilization. The suggested method's efficacy is assessed by contrasting it with the existing techniques, namely Swarm optimization (Raghav & Vyas, 2023), Ant Colony optimization (ACO) (Raghav & Vyas, 2023), and Hybrid_BirdSwarm_ACO optimization (Raghav & Vyas, 2023).

4.1 Configuration of a Virtual Machine

The configuration for the VM is as follows: There are 10 VMs in total. Each VM has an image size of 10000 MB and is equipped with 512 MB of RAM. Each VM has 250 million instructions per second. A 1000 Gbps bandwidth allotment has been made. The name of the VM is 'Xen'. Each VM has 1 Million Processing Element per Second (M_PES).

4.2 Makespan

Makespan shows the time required to finish all tasks provided to the system in a certain amount of time. The total time required to complete every task submitted to the system value that it represents. The term "system makespan" refers to the maximum duration that the host needs to operate in the data center. Figure 3 shows the makespan value of the proposed and existing approaches. The evaluation of the system's makespan (MS) is displayed in equation (7). Comparative result shows our proposed method performs better than existing methods.

$$MS = max\,\{Task\ CT\ (h\ l)\ |\ h = 1,2,..p;\ l = 1,2,..o\} \tag{7}$$

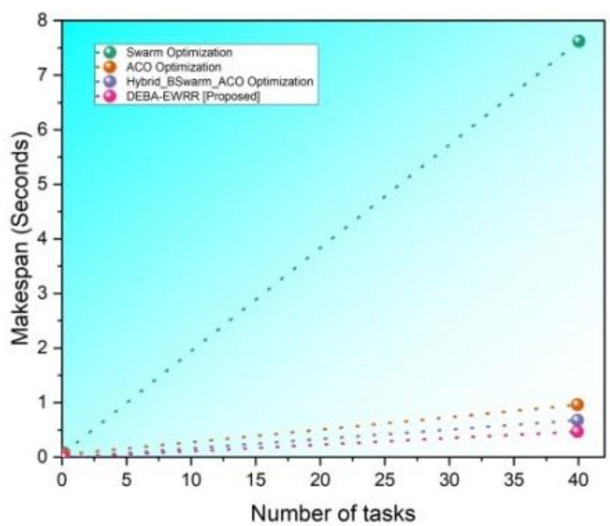

Figure 3: Result of Makespan

4.3 Fitness Score

Evaluation of a resource's suitability for managing a specific task. A few of the factors that are typically considered while computing the fitness score include the resource's capability, availability, and current load. Figure 4, shows the Fitness score for the proposed and existing approaches. A comparative analysis reveals that our suggested approach outperforms existing approaches.

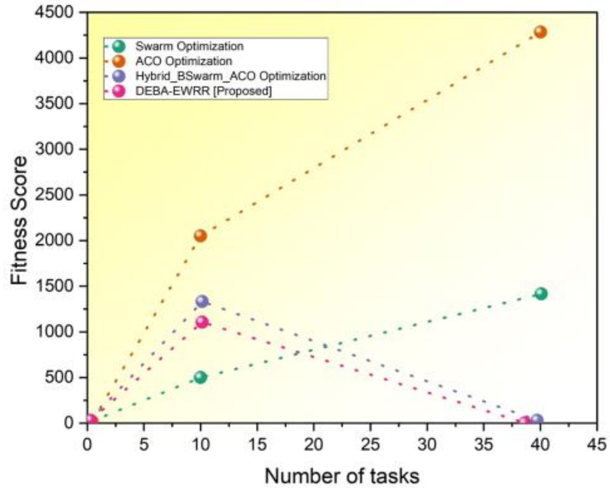

Figure 4: Result of Fitness Score

4.4 Resource Utilization

In order to attain maximum productivity, resources must be efficiently allocated and managed. This is known as resource utilization. In order to accomplish organizational objectives while reducing waste and expenses, it entails evaluating and optimizing the use of a variety of assets, including labor, funds, resources and technology. Figure 5 shows the resource utilization result of the proposed and existing approaches. Our suggested approach outperforms the existing methods, according to the comparative findings.

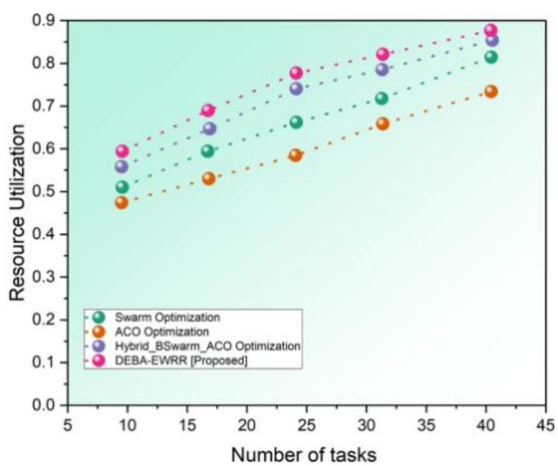

Figure 5: Result of Resource Allocation

4.5 Throughput

The rate of successful data processing or transmission in a system, connection, or equipment during a given time frame is referred to as throughput. “Bits per second (bps) or transactions per second (tps)” are common units of measurement for this quantification of data transmission or processing volume. High throughput is an indicator of effective operation. Figure 6 shows the throughput result of the proposed and existing methods. Based on a comparative analysis, our suggested approach outperforms the existing approach.

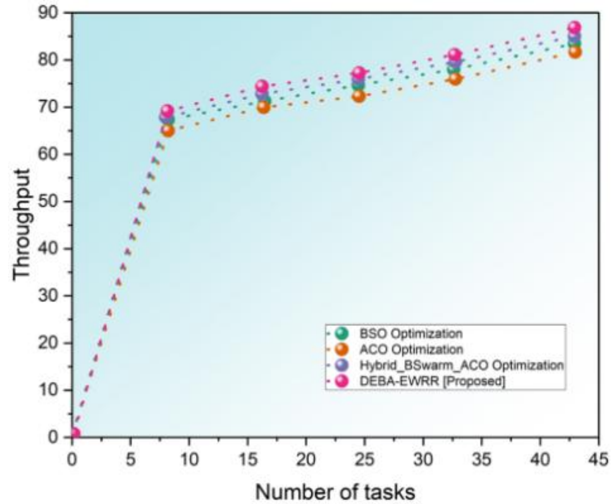

Figure 6: Result of Throughput

5 Conclusion

In financial cloud systems, performance optimization requires effective load balancing & optimal resource allocation to provide a fair distribution of computational activities, improve system stability, and minimize latency. The paper introduces new algorithms, the DEBA and the EWRR, that are aimed at improving load balancing & resource allocation in financial cloud computing. These algorithms address difficulties in reaching the best possible resource usage requirements when combined into a three-phase system. Workload balancing is made easier by the EWRR technique, and enhancing resource allocation is optimized by DEBA. The RSVM technique is applied to improve the cost-effective VM migration procedure. The DEBA-EWRR algorithm provides better performance, cost-effectiveness in financial cloud environments, according to simulation research. Larger-scale DEBA and EWRR implementation may provide scaling issues, and depending on simple assumptions, may not properly depict the dynamics of real-world financial clouds, requiring additional empirical validation. Further studies could examine flexible algorithms customized for constantly changing financial cloud settings, incorporating all-encompassing security protocols to tackle ever-changing risks.

References

[1] Abdullah, D. (2024). Leveraging FPGA-based design for high-performance embedded computing. *SCCTS Journal of Embedded Systems Design and Applications, 1*(1), 37-42. https://doi.org/10.31838/ESA/01.01.07

[2] Akhtar, N., Kerim, B., Perwej, Y., Tiwari, A., & Praveen, S. (2021). A comprehensive overview of privacy and data security for cloud storage. *International Journal of Scientific Research in Science Engineering and Technology*. https://dx.doi.org/10.32628/IJSRSET21852

[3] Alqahtani, F., Amoon, M., & Nasr, A. A. (2021). Reliable scheduling and load balancing for requests in cloud-fog computing. *Peer-to-Peer Networking and Applications, 14*(4), 1905-1916. https://doi.org/10.1007/s12083-021-01125-2.

[4] Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. *Automation in construction*, *122*, 103441. https://doi.org/10.1016/j.autcon.2020.103441

[5] Brahmam, M. G. (2024). VMMISD: an efficient load balancing model for virtual machine migrations via fused metaheuristics with iterative security measures and deep learning optimizations. *IEEE Access*, *12*, 39351-39374. https://doi.org/10.1109/ACCESS.2024.3373465

[6] Chahlaoui, F., El-Fenni, M. R., & Dahmouni, H. (2019, March). Performance analysis of load balancing mechanisms in SDN networks. In *Proceedings of the 2nd International Conference on Networking, Information Systems & Security* (pp. 1-8). https://doi.org/10.1145/3320326.3320368

[7] Chitra, S., Kamalee, S., Sasidharan, S., Sohail Ahmed, M., & Balamurugan, K. (2023). Improving Task Allotment by using Deadline Resource Provisioning and Semo Algorithm in Cloud Computing. *International Journal of Advances in Engineering and Emerging Technology, 14*(1), 64–68.

[8] Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *Journal of Parallel and Distributed Computing, 142,* 36-45. https://doi.org/10.1016/j.jpdc.2020.03.022

[9] Guo, B., Liang, G., Yu, S., Wang, Y., Zhi, C., & Bai, J. (2021). 3D printing of reduced graphene oxide aerogels for energy storage devices: a paradigm from materials and technologies to applications. *Energy Storage Materials*, *39*, 146-165. https://doi.org/10.1016/j.ensm.2021.04.021.

[10] Iyer, S., & Trivedi, N. (2023). Cloud-powered Governance: Enhancing Transparency and Decision-making through Data-driven Public Policy. In *Cloud-Driven Policy Systems* (pp. 13-18). Periodic Series in Multidisciplinary Studies.

[11] Jena, U. K., Das, P. K., & Kabat, M. R. (2022). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. *Journal of King Saud University-Computer and Information Sciences, 34*(6), 2332-2342. https://doi.org/10.1016/j.jksuci.2020.01.012

[12] Jyoti, A., & Shrimali, M. (2020). Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing. *Cluster Computing, 23*(1), 377-395. https://doi.org/10.1007/s10586-019-02928-y.

[13] Kumar, J., Singh, A. K., & Mohan, A. (2021). Resource-efficient load-balancing framework for cloud data center networks. *Etri Journal*, *43*(1), 53-63. https://doi.org/10.4218/etrij.2019-0294

[14] Li, C., Bai, J., Chen, Y., & Luo, Y. (2020). Resource and replica management strategy for optimizing financial cost and user experience in edge cloud computing system. *Information Sciences*, *516*, 33-55. https://doi.org/10.1016/j.ins.2019.12.049

[15] Lynn, T., Fox, G., Gourinovitch, A., & Rosati, P. (2020). Understanding the determinants and future challenges of cloud computing adoption for high performance computing. *Future Internet*, *12*(8), 135. https://doi.org/10.3390/fi12080135.

[16] Mittal, M., Battineni, G., Singh, D., Nagarwal, T., & Yadav, P. (2021). Web-based chatbot for frequently asked queries (FAQ) in hospitals. *Journal of Taibah University Medical Sciences*, *16*(5), 740-746. https://doi.org/10.1016/j.jtumed.2021.06.002

[17] Raghav, Y. Y., & Vyas, V. (2023). ACBSO: a hybrid solution for load balancing using ant colony and bird swarm optimization algorithms. *International Journal of Information Technology*, *15*(5), 2847-2857. https://doi.org/10.1007/s41870-023-01340-5

[18] Rahim, R. (2024). Quantum computing in communication engineering: Potential and practical implementation. *Progress in Electronics and Communication Engineering, 1*(1), 26–31. https://doi.org/10.31838/PECE/01.01.05

[19] Uvarajan, K. P. (2024). Advances in quantum computing: Implications for engineering and science. *Innovative Reviews in Engineering and Science, 1*(1), 21-24. https://doi.org/10.31838/INES/01.01.05

[20] William, A., Thomas, B., & Harrison, W. (2025). Real-time data analytics for industrial IoT systems: Edge and cloud computing integration. *Journal of Wireless Sensor Networks and IoT, 2*(2), 26-37.

[21] Wu, Z. (2024). Integrating Biotechnology Virtual Labs into Online Education Platforms: Balancing Information Security and Enhanced Learning Experiences. *Natural and Engineering Sciences, 9*(2), 110-124. https://doi.org/10.28978/nesciences.1569211

[22] Zhang, S., Pandey, A., Luo, X., Powell, M., Banerji, R., Fan, L., ... & Luzcando, E. (2022). Practical adoption of cloud computing in power systems—Drivers, challenges, guidance, and real-world use cases. *IEEE Transactions on Smart Grid*, *13*(3), 2390-2411. https://doi.org/10.1109/TSG.2022.3148978

Authors Biography

Dr. Amaresh Sahu is the Head of Department and Associate Professor in the Master of Computer Application program at Ajay Binay Institute of Technology, Odisha, India. He brings extensive experience in computer science education, with a focus on software development, programming, and system design. Dr. Sahu is dedicated to academic excellence, research, and student mentorship.

R. Murugan is an Associate Professor in the Department of Computer Science and Information Technology at Jain (Deemed-to-be University), Bangalore, Karnataka, India. His academic interests include computer networks, cybersecurity, and advanced computing technologies. He is actively involved in research, teaching, and mentoring students.

Dr. Sadaf Hashmi is an Associate Professor in the Department of ISME at ATLAS SkillTech University, Mumbai, Maharashtra, India. She specializes in management studies with a focus on strategic leadership, innovation, and entrepreneurship. Dr. Hashmi is actively engaged in academic research, teaching, and industry-oriented learning.

Dr. Arun Kumar Marandi is an Associate Professor in the Department of Computer Science & IT at ARKA JAIN University, Jamshedpur, Jharkhand, India. His areas of expertise include data science, artificial intelligence, and software engineering. He is actively involved in academic research, curriculum development, and mentoring students.

Amit Kumar is associated with the Centre of Research Impact and Outcome at Chitkara University, Rajpura, Punjab, India. His work focuses on enhancing research quality, visibility, and interdisciplinary collaboration. He actively supports institutional research initiatives and promotes impactful scholarly outputs.

Sachin S. Pund has received B.E. in Industrial Engineering from Shri Ramdeobaba College of Engineering and Management, Nagpur University, in 2000 and M. Tech degree in Industrial Engineering from Shri Ramdeobaba College of Engineering and Management, Nagpur University, 2008. He has awarded with Ph.D. degree at the Department of Mechanical Engineering, in 2024 from SSSUTMS, Sehore, Bhopal (MP), India. His research interests include Optimization Techniques, Facilities Planning, Operations Management and Project Management.