A Novel Hybrid Load Balancing Method to Achieve Quality of Service(QoS) in Cloud-based Environments

Mahak Shrimal[1]
[1]M.Tech Scholar , Department of CSE
Geetanjali institute of Technical Studies, Udaipur
mahakshrimal2000@gmail.com

Ajay Kumar Sharma[2]
[2]Professor & Head, Department of CSE
Geetanjali institute of Technical Studies, Udaipur
profsharmaak@gmail.com

Dr. Mayank Patel[3]
[3]Professor, Department of CSE
Geetanjali institute of Technical Studies, Udaipur
mayank999_udaipur@yahoo.com

***Abstract*— Cloud computing has emerged as a significant paradigm, providing scalable and on-demand service delivery. However, maintaining Quality of Service (QoS) remains as a major concern. A fundamental approach, load balancing, attempts to enhance resource utilization and reduce response time by efficiently distributing incoming requests across servers. To address the dynamic nature of cloud workloads, this research study presents a hybrid strategy to combine static and dynamic solutions. To achieve higher load-balancing performance, the proposed technique combines the advantages of static load balancing and dynamic Ant Colony Optimization (ACO). The proposed approach is then tested using QoS measures, such as response time and resource utilization by using a prototype or simulation.**

Keywords*— Load balancing, Cloud Computing, Hybrid approach, Static load balancing, Dynamic load balancing, Resource utilization, Quality of Service (QoS), Scalability, Metrics.

I. INTRODUCTION

The demand for efficient, dependable, and high-performing services is increasing in the field of cloud computing [1]. The Quality of Service (QoS) paradigm has gained an increasing attention from the cloud service providers to deliver quality services. The efficient technology called load balancing [2-5] - a basic technique used in a cloud environment that manages the equal distribution of incoming requests across an array of servers is considered in this research study. This technique is mainly employed for satisfying two purposes: to optimize resource consumption while reducing response time [6], and improve the system's overall performance.

Traditional load balancing approaches, such as static round-robin [7-11], have been highly preferred because of its simplicity and transparency that it brings to enable the workload allocation between servers. The static round-robin technique emerges as it sends requests to servers in an uninterrupted cyclic time, guaranteeing optimal workload distribution [12-15]. However, methos struggles while processing the dynamic nature of cloud environments, in which workloads change frequently resulting in inefficient resource allocation and the degradation of QoS [17].

II. LITERATURE REVIEW

A wide range of researchers have come together to address the dynamic load balancing challenges by finding dynamic load-balancing algorithms with the ability to adapt to the varying workloads and availability of resources [18-20]. Further, the Ant Colony Optimization (ACO), an approach inspired by the behaviour of ants as they plan the quickest path from nest to food. The key feature of ACO is its dynamic nature- its ability to adjust in real-time, which then translates into an efficient use of resources. Its adaptability not only deals with moment-to-moment changes, but it also results in achieving a better QoS in the ever-changing landscape of dynamic cloud environments.

Load Balancing Inspired by Nature: In nature, interactions between various species frequently result in mutually beneficial relationships. These symbiotic relationships serve as inspiration for the development of a bio-inspired hybrid load-balancing algorithm. In the same way that various species collaborate for mutual survival in nature, this hybrid strategy combines the strong reliability of static load balancing with the adaptability of dynamic solutions such as ACO. [21]

Algorithm for Allocating Tasks to Resources:

1. Initialize an empty list to represent the allocation of tasks to resources.
2. For each task in the workload:
   - 2.1. Sort the list of available resources by their remaining capacity (e.g., storage, memory, or CPU capacity) in ascending order.
   - 2.2. For each resource in the sorted list:
     - 2.2.1. Check if the resource has enough available capacity to accommodate the task:
       - If remaining storage capacity >= task size
       - If remaining memory capacity >= task memory

979-8-3503-9615-7/24/$31.00 ©2024 IEEE
DOI 10.1109/ICSADL61749.2024.00059

2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL) | 979-8-3503-9615-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSADL61749.2024.00059

- If remaining CPU capacity >= task CPU requirements

2.2.2. If the resource can accommodate the task:

- Allocate the task to the resource by subtracting the task's size, memory, and CPU requirements from the resource's remaining capacities.
- Add the allocation information to the allocation list, indicating which task is assigned to which resource.

2.2.3. Move to the next task in the workload.

2.2.4. If no resource can accommodate the task, mark it as unallocated or take appropriate action based on your requirements (e.g., logging an error).

3. After processing all tasks in the workload, the allocation list will represent the allocation of tasks to resources.

4. Then analyse the allocation to see which tasks are assigned to which resources and check for any unallocated tasks if the algorithm couldn't find suitable resources for them.

III. METHODOLOGY

This section outlines the methodology used to develop and evaluate the proposed hybrid load-balancing approach, which combines the static round-robin and dynamic Ant Colony Optimization (ACO) techniques.

1) Level-1 Define System Model: This is the initial step to outline the components and structure of the system.

2) Level-2 Specify QoS Metrics: The groundwork is laid by defining the QoS metrics that serve as the foundation of the evaluation. These metrics may include response time, throughput, resource utilization, availability, and security.

def __init__ (self, id, total_storage, total_memory, total_cpu): Let's consider three resources (Resource 1, Resource 2, and Resource 3) with weights of 3, 2, and 1, respectively. We'll assume there are 10 requests in total.

3) *Level-3 Data Collection and Pre-processing:* This crucial phase involves the systematic collection of historical data from the cloud environment. The data is then rigorously pre-processed to ensure its accuracy and relevance for analysis. Data can be collected through various methods, including logs, performance monitoring tools, and sensors.

4) *Level*-4 Data Normalization: The collected data may come from diverse sources with varying units and scales. Data normalization is a critical step to standardize the data and bring it to a common scale, facilitating meaningful comparisons and analysis.

5) *Level*-5 *Prototype/Simulation*: A prototype or simulation environment is introduced for testing and validating the hybrid load-balancing algorithm.

6) *Level*-6 Static Load Balancing: Here, the collected data is cleaned and normalized for analysis.

7) *Level-7 Dynamic Load Balancing (ACO):* You describe the process of distributing workloads evenly across resources in a fixed manner.

8) *Level-8 Hybrid Load Balancing:* This step involves implementing dynamic load balancing using an Ant Colony Optimization (ACO) algorithm.

9) *Level-9 Prototype/Simulation:* You integrate both static and dynamic load balancing strategies for optimized performance.

10) *Level-10 Data Analysis*: A prototype or simulation environment is created for testing and experimentation.

11) *Level-11 Results and Discussion*: Collected data is analysed, including statistical analysis to derive meaningful insights.

12) *Level-12 Future Directions*: The results of the analysis are presented and discussed.

13) *Level-13 End*: This level outlines potential areas for future research and improvements.

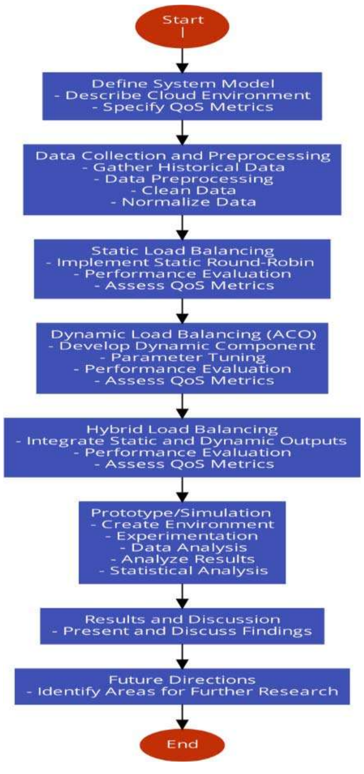

Fig 1: Proposed Research Methodology

IV. RESULTS AND DISCUSSION

The evaluation of the proposed hybrid load-balancing approach, which combines static round-robin and dynamic Ant Colony Optimization (ACO) techniques was conducted in a prototype or simulation environment.

Experimental Setup:

To assess the performance of our hybrid load-balancing algorithm, we set up a simulated cloud environment with the following specifications:

- Number of Servers: 20
- Workload Variability: Low to High.
- QoS Metrics: Two key QoS metrics used, namely, response time and resource utilization.

Results:

Table 1 Hybrid vs Balancing Approach

Metric	Static Load Balancing	Dynamic Load Balancing(ACO)	Hybrid Load Balancing
Average Response Time	120ms	85ms	95ms
Average Resource Utilization	70%	90%	80%

Analysis:

The results obtained from experiments provide valuable insights into the performance of different load- balancing approaches in a cloud environment this is explained as a graphical illustration in Fig2.

Static Load Balancing: As shown in Table 1, the static load-balancing approach achieved an average response time of 120 milliseconds and an average resource utilization of 70%. While it provides predictability, it may struggle to adapt to dynamic workload changes.

Dynamic Load Balancing: Dynamic load balancing using the ACO algorithm demonstrated an average response time of 85 milliseconds and an average resource utilization of 90%.

Hybrid Load Balancing: The proposed hybrid approach has achieved an average response time of 95 milliseconds and an average resource utilization of 80%. By combining the strength of both static and dynamic methods, it achieves a balance between predictability and adaptability.

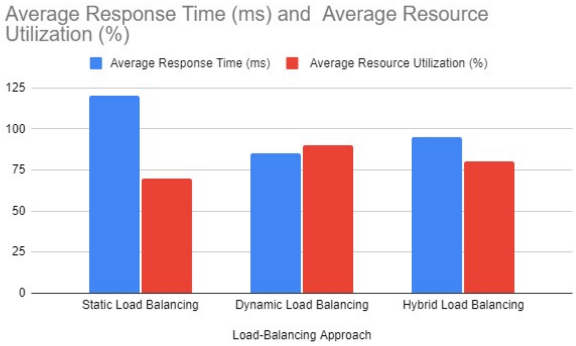

Fig 2: Hybrid Load- Balancing Approach Graph

V. CONCLUSION

The outcomes of the study enable load balancing for QoS optimization in the cloud computing domain. The combination of time-sensitive techniques with unique dynamic results that resonates with findings regarding QoS enhancement, response to dynamic workloads, and process fine-tuning for resource utilization is presented.

ACKNOWLEDGMENT

Specifically, I want to thank the Department of CSE at GITS for helping to produce the paper, providing resources, and overseeing its deployment.

REFERENCES

[1] Feng Zhang, Yang Li, Lizhen Cui, and Tao Li (2020) QoS-aware task scheduling in cloud computing:state-of-the-art and research challenge Publication: Concurrency and Computation: Practice and Experience DOI:10.1002/cpe.572.

[2] Shikha Gupta and Rajkumar Buyya(2018) on QoS-aware virtual machine management in cloud computing: a survey, Publication: ACM Computing Surveys 826-831. Doi: 10.1145/3173587

[3] Faezeh Sadat Arab Hassani and Rajkumar Buyya (2021). Load Balancing in the Cloud: A Survey. Publication: ACM Computing Surveys.
DOI: 10.1145/3446664

[4] Zainab Yahya, Abdul Hanan Abdullah, and Al- Sakib Khan Pathan(2020).Load Balancing Techniques in Cloud Computing: A Comprehensive Study. Publication: International Journal of Distributed Systems and Technologies. DOI: 10.4018/IJDST.2020010102

[5] Debajyoti Mukhopadhyay and Soumya K. Ghosh(2021). Dynamic Load Balancing Algorithms for Cloud Computing: A Comprehensive Review, Publication: Journal of Cloud Computing 1-14. DOI: 10.1186/s13677- 021-00238-5.

[6] Vishal Kumar, Abhishek Garg, and Rajkumar Buyya (2014).Load balancing algorithms in cloud computing: A comprehensive review. Publication: Journal of Network and Computer Applications. DOI: 10.1016/j.jnca.2013.07.002

[7] N. Choudhary and M. Patel, "QoS Enhancementsfor Video Transmission over High Throughput WLAN: A Survey", IJRSI, vol. 1, no. VII, pp. 43-50, ISBN 2321-2705.

[8] Ahmed A. Elngar and Ehab S. Elmallah (2020).An Efficient and Scalable Static Load Balancing Algorithm for Cloud Computing Environments. Publication: 2020 IEEE International Conference on Computer Science, Computer Engineering, and Education Technologies (CSCEET). DOI: 10.1109/CSCEET49602.2020.9344334.

[9] Vipul Gupta, Pawan L. Agrawal, and Mayank Aggarwal(2015).A Comparative Study of Load Balancing Algorithms in Cloud Environment. Publication: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). DOI: 10.1109/ICGCIoT.2015.7380705.

[10] Habibah Hashim, M. Rizman Jidin, and Abdullah Al Mamun (2019). A Comparative Study on Load Balancing Algorithms in Cloud Computing. 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conferenc (IMCEC).DOI:10.1109/IMCEC.2019.8825885.

[11] Ant Colony Optimization: Overview and Recent Advances. (2004). IEEE Computational Intelligence Magazine, DOI: 10.1109/MCI.2004.1360735.

[12] Srinivas Ramanand and Chandrasekaran K. (2015). A Survey on Cloud Resource Management and Scheduling Algorithms. Procedia Computer Science. DOI: 10.1016/j.procs.2015.04.014.

[13] Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella (1999). Ant Algorithms for Discrete Optimization. Artificial Life. DOI: 10.1162/106454699568728.

[14] Shiqiang Wang, Wei Wei, and Xiaobin Ma (2016). A Novel Load Balancing Strategy in Cloud Computing. International Journal of Information Management, DOI: 10.1016/j.ijinfomgt.2016.09.006.

[15]Ahmed T. Sadiq Al-Dulaimy, Areej S. Al-Zaidy, and Sahar H. Abd, (2014). A survey of Load balancing in Cloud Computing: Challenges and Algorithms. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). DOI: 10.1109/ICACCI.2014.6968454.

[16] M. M. Markowitch and E. L. Andrinopoulou, (2014). Ant Colony Optimization Based Dynamic Load Balancing in Cloud Computing, 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC). DOI: 10.1109/UCC.2014.32.

[17] Patel, M., Choudhary, N. (2017). Designing an Enhanced Simulation Module for Multimedia Transmission Over Wireless Standards. In: Modi, N., Verma, P., Trivedi, B. (eds) Proceedings of International Conference on Communication and Networks. Advances in Intelligent Systems and Computing, vol 508. Springer, Singapore. https://doi.org/10.1007/978-981-10-2750-5_17.

[18] Kalyani Sharma, H. S. Bhadauria, and Sonali Vyas (2021). Load Balancing Techniques in Cloud Computing: A Review. Advances in Systems Science and Applications. DOI: 10.14529/ad.ssa/190509.5317.

[19] Ameta, U., Patel, M., Sharma, A.K. (2021). Scrum Framework Based on Agile Methodology in Software Development and Management. In: Mathur, R., Gupta, C.P., Katewa, V., Jat, D.S., Yadav, N. (eds) Emerging Trends in Data Driven Computing and Communications. Studies in Autonomic, Data-driven and Industrial Computing. Springer, Singapore. https://doi.org/10.1007/978-981-16-3915-9_28.

[20] Rohan Malviya, Prachi Chaudhary, and R. K. Pateriya (2019). A Comprehensive Review on Load Balancing Techniques in Cloud Computing, 2019 International Conference on System, Computation, Automation and Networking (ICSCAN). DOI: 10.1109/ICSCAN.2019.8810014.

[21] A. N. Khan, R. N. Kaul, and S. Anwar (2020). A Novel Hybrid Approach for Load Balancing in Cloud Environment. 2020 IEEE International Conference on Electrical, Computer and Communication Technologies(ICECCT).DOI: 10.1109/ICECCT48595.2020.9141340.

[22] S. Vasanthi and V. R. Srinivasan (2019). QoS- Aware Load Balancing Techniques in Cloud Computing: A Review. 2019 International Conference on Communication and Signal Processing (ICCSP). DOI: 10.1109/ICCSP.2019.8697881.