A Critical Analysis of Dynamic Resource Allocation and Load Balancing in Multi-Cloud Environments

H. Rehan Vibashana Perera
School of Computer Science, University of Sunderland

Abstract
This research paper addresses the pressing challenge of dynamic resource allocation and load balancing in multi-cloud environments. With the growing adoption of cloud computing, organizations increasingly rely on multiple cloud providers to meet their diverse needs. Effective resource allocation and load balancing are critical to optimizing performance and cost efficiency. This paper reviews and evaluates current research, presents a methodology for enhancing dynamic resource allocation, conducts a real world validity analysis of existing research, and formulates well-reasoned conclusions. It emphasizes the significance of customized multi-cloud solutions in the ever-evolving realm of cloud computing.

Keywords*— *Multi-Cloud, Dynamic Resource Allocation, Load Balancing, Critical Analysis

1. Introduction

The advent of cloud computing has revolutionized the IT landscape, offering scalability, flexibility, and cost effectiveness. Multi-cloud environments, characterized by using multiple cloud providers, have become a common strategy for organizations to diversify their infrastructure, minimize vendor lock-in, and ensure redundancy.

research carried out by Saxena, Gupta, and Singh, (2021) assessed the CHARM model for multi-cloud data hosting. This model aims to balance cost-effectiveness and high availability. It achieves this by using a "Proxy" system to manage data distribution across the multi-cloud infrastructure and combining replication and erasure coding for cost reduction and data assurance.

Alam, Fadlullah, and Choudhury, (2021) state “The core idea behind optimal resource allocation is to approach it in a conversational tone. It involves finding the best way to manage resources by considering various factors, including cost efficiency, resource utilization, and the maximization of quality standards.”

In the study by Rodigari et al., (2021), state “the focus lies on assessing the performance of CPU, memory usage, and latency in the context of HTTP requests when Istio is implemented in a multi-cloud environment.” Their research introduces a multi-cloud framework along with a testing workflow aimed at evaluating the data plane's performance under heavy loads and understanding how enabling zero trust affects the control plane.

Initial test findings indicate that Istio contributes to a reduction in latency variability when responding to sequential HTTP requests. Additionally, the study highlights that the overall CPU and memory utilization can fluctuate based on the specific service mesh configuration and the cloud environment in use.

However, managing resources effectively and achieving proper load balancing in such diverse cloud environments presents a set of substantial challenges.

In a recent study by Saif et al., (2023) they pointed out that traditional load balancing methods often suffer from excessive communication overhead and don't adequately address the complexities of multi-cloud setups.

Moreover, web-based applications often encounter sudden surges in traffic, known as "flash crowds," which can lead to resource shortages and performance bottlenecks.

Adewojo and Bass, (2022) found that adopting decentralized systems with dynamic geographical load balancing, coupled with a load distribution algorithm based on various server metrics, resulted in significant performance enhancements for three-tier web applications in multi-cloud environments.

Throughout the course of this paper, we will delve into the dynamic allocation of resources and the implementation of effective load-balancing strategies in the context of multi-cloud environments, with the overarching goal of optimizing resource utilization and elevating system performance.

2. Current Work for effective multi-cloud environment

Following section is related to dynamic resource allocation and load balancing in multi-cloud environments reveals several key trends and contributions:

I. Dynamic Allocation

Much of the existing research has historically focused on static resource allocation strategies, which provision resources based on anticipated workloads. However, the dynamic nature of multi-cloud environments necessitates the adoption of real-time allocation mechanisms.

Alam, Fadlullah, and Choudhury, (2021) put forth a model wherein they propose an evaluation framework for trust in cloud computing, grounded in four crucial Key quality attributes, including data integrity, availability, reliability, and efficiency, are essential factors. Also, primary objectives in this model (fig1) are to streamline resource allocation, taking trust into account while simultaneously minimizing communication delays.

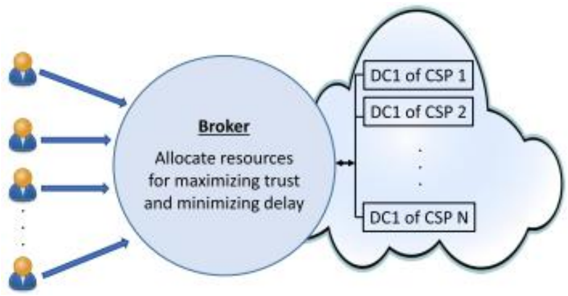

Figure 1 Resource Allocation Model (Alam, Fadlullah, and Choudhury, 2021)

To tackle this challenge, they employ a genetic algorithm (fig2) as a heuristic approach, and they substantiate the effectiveness of their method through a series of experiments.

Input:
- *Resource capacities for all servers of all CSPs*
- *Availability, reliability, data integrity, and time efficiency of the servers of all CSPs*
- *Matrix for communication traffic of all VMs*

Output:
- *The best VM allocation set with maximum fitness value CR_b*

```
for all incoming requests rq ∈ RQ
    for all existing CSPs n ∈ N
        for all available servers s ∈ S_n do
            Estimate trust
            Estimate delay
        end for
    end for
end for
CR_b ← Initialize this set empty
Randomly generate a set of C chromosomes
Evaluate fitness (trust and delay) for the above C chromosomes
generation = 0
while generation <= MAX_generation
    while new chromosomes set ≠ C
        Select two chromosomes applying roulette wheel operator
        Based on the crossover probability, crossover occurs
        According to the mutation probability, mutation occurs
    end while
Estimate fitness of the new set of chromosomes
Select the fittest chromosomes CR_n using selection operation
    if f(CR_n) > f(CR_b)
        CR_b = CR_n
    end if
generation = generation + 1
end while
return the best allocation set (CR_b)
```

Figure 2 Algorithm1 (Alam, Fadlullah, and Choudhury, 2021)

They conduct experiments to test evaluating the effectiveness of their proposed method in enhancing quality attribute performance in a cloud context, they employ Cloud Sim for data collection and MATLAB for optimizing resource allocation (fig3). Then the trust evaluation model is implemented using the collected data, and the trust value is The optimizer and Genetic Algorithm (GA) are seamlessly integrated into the optimization model, executed within MATLAB environments for analysis.

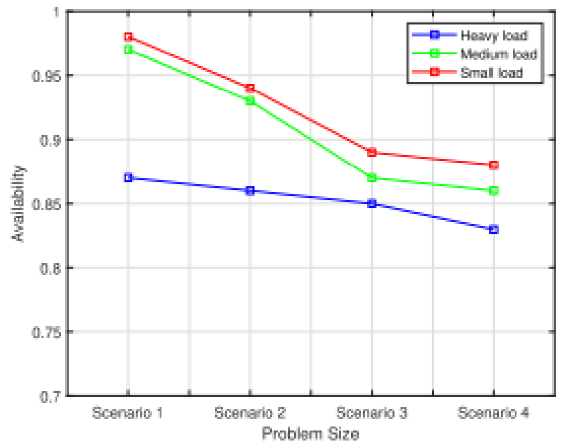

Figure 3 Resources under Different loads (Alam, Fadlullah, and Choudhury, 2021)

Then the experiments encompass various scenarios to assess the approach's practicality. They conduct a performance comparison between their heuristics and the optimal solution, demonstrating that the Genetic algorithm achieves a remarkable 90% similarity to the exact solution.

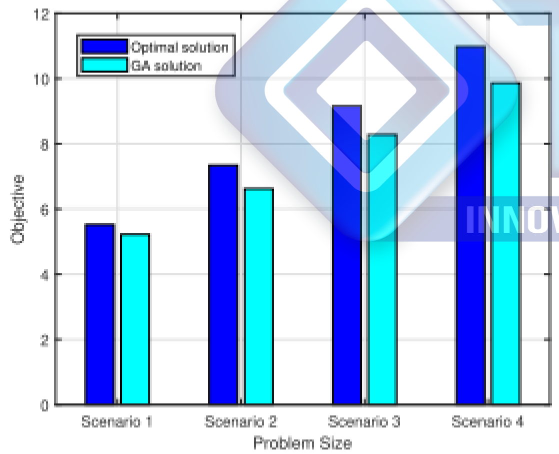

Figure 4 Optimal and Genetic Algorithm Comparison (Alam, Fadlullah, and Choudhury, 2021)

The execution time of the Genetic Algorithm increases linearly with an increase in the number of the CSPs and servers, ensuring the validity of the proposed trust model in a practical cloud environment.

Alyas et al. (2023) presents a framework aimed at Improving the Quality of Service (QoS) and allocating resources effectively in multi-cloud environments (see fig5). The framework focuses on three key parameters: data accessibility, optimization, and collaboration, which are derived from existing literature and various cloud models. These parameters guide dynamic resource allocation, improving QoS within decentralized multi-cloud platforms.

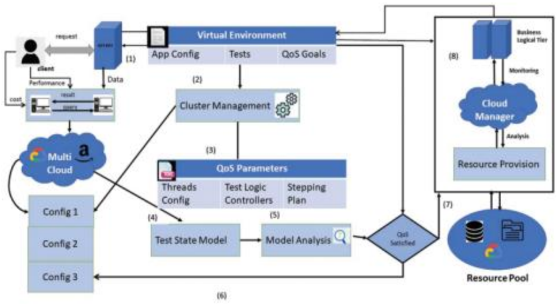

Figure 5 proposed framework (Alyas et al. 2023)

The framework employs an optimization technique based on these parameters, which are further subdivided for resource allocation and long-term service quality assessment. While the framework's effectiveness is confirmed through simulation experiments using the CloudSim simulator.

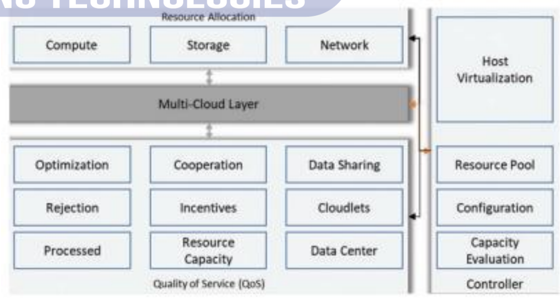

Figure 6 multi-cloud resource allocation Parameters (Alyas et al. 2023)

The framework also addresses resource allocation in decentralized multi-cloud environments. It distinguishes between scenarios with a central entity and decentralized multi-cloud configurations, with a particular focus on the latter's complexity. In decentralized multi-cloud settings, cooperation among resource providers is pivotal for achieving optimal quality of service (fig6).

However, this cooperation is challenged by the heterogeneous nature of multi-cloud environments, where providers may have varying incentives and goals. Balancing optimization and cooperation efficiency is crucial for achieving performance and service goals.

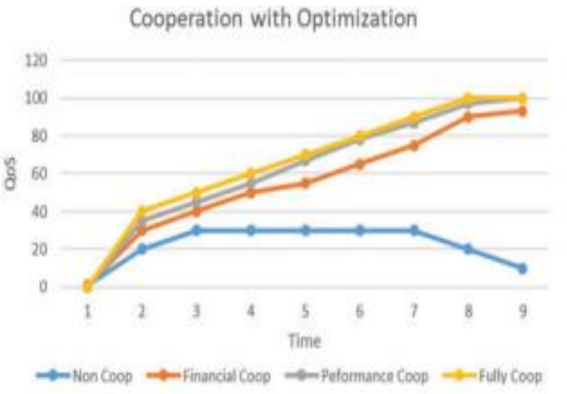

Figure 7 Impact Analysis Optimization (Alyas et al. 2023)

Using the CloudSim simulator, the study validates the proposed resource allocation framework in decentralized multi-cloud settings (fig7). The simulation considers key factors: a decentralized multi-cloud, a broker, and a load manager. Initially, without the framework, resource capacity declines during peak hours.

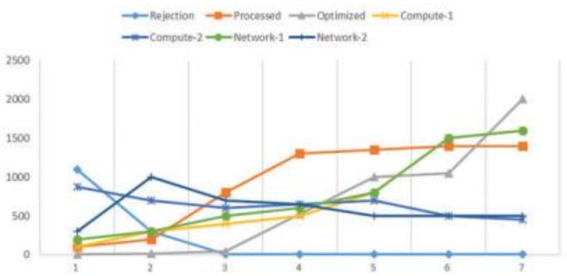

Figure 8 performance improvement (Alyas et al. 2023)

Subsequently, with the framework, there is a significant resource boost (fig7,8), particularly in computers, storage, and networks. This framework optimizes resource allocation, enhances service quality, and demonstrates stability in the simulation, suggesting real-world potential.

Selvapandian and Santhosh, (2022) introduce a novel framework that focuses on optimizing resource allocation in multi-cloud environments (fig9). The research presents a hybrid optimization algorithm called the Bat Algorithm Particle Swarm Optimization (BAPSO) for this purpose.

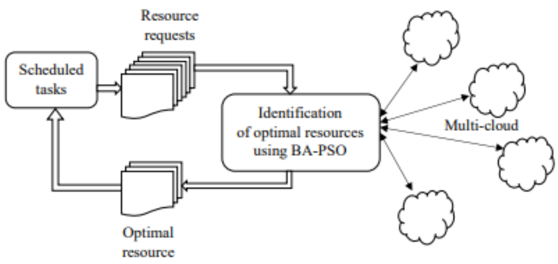

Figure 9 Introduced Resource Allocation Model (Selvapandian and Santhosh, 2022)

This algorithm combines the strengths of bat optimization and particle swarm optimization to address challenges in multi-cloud resource allocation. The bat optimization algorithm excels in finding global optimal solutions, while the particle swarm optimization algorithm is known for its quick convergence characteristics. By leveraging these attributes, the hybrid approach aims to provide an efficient resource allocation model for multi-cloud environments (fig10).

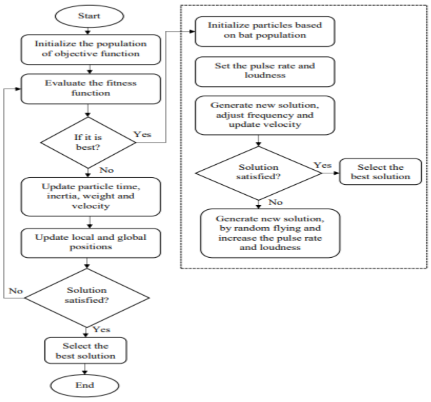

Figure 10 Hybrid Optimization Model (Selvapandian and Santhosh, 2022)

Extensive simulation experiments validated their model's effectiveness. The BAPSO-based approach outperformed conventional methods, excelling in resource allocation efficiency, energy conservation, reduced SLA violations, and faster allocation times (fig11).

Parameter	Value
Total number of clouds	16
Total number of hosts	40
VM speed	200–2000 MIPS
Memory	500–1 TB
Input task length	3000–5000
Hypervisor	Xen
Parameter	**Value**
Bat size	15
Acceleration constants	1.04
Maximum number of iterations	150
Frequency, loudness, pulse rate (minimum)	0
Frequency, loudness, pulse rate (maximum)	3, 1, 2
Loudness constant	0.96
Pulse rate constant	0.9

Figure 11 Testing Parameters (Selvapandian and Santhosh, 2022)

This study emphasizes the inherent complexity of resource allocation in cloud computing, particularly in multi-cloud environments. It underscores the importance of considering resource availability and capability prior to allocating resources to requested tasks.

Algorithm	Energy consumption [kWh]	Execution time [sec]
Genetic algorithm [23]	380	110
ACO [23]	460	135
GA-RF [23]	310	90
Proposed BAPSO	200	47.22

Figure 12 Performance Comparison (Selvapandian and Santhosh, 2022)

Additionally, this study shows the dynamic update of membership functions, a critical aspect ensuring QoS requirements are met, and resource allocation performance is optimized (fig12). Ultimately, *Selvapandian and Santhosh, (2022)* present a pioneering approach that significantly improves resource allocation within multi-objective cloud environments.

In *Chen, (2023)* study, an innovative approach to multi-objective optimization task scheduling within a multi-cloud environment was introduced, emphasizing the application of dynamic programming (fig13). To support this research, a Java-based task-scheduling simulation tool was meticulously crafted, serving as a platform to evaluate the proposed algorithm's efficacy through comprehensive simulation experiments.

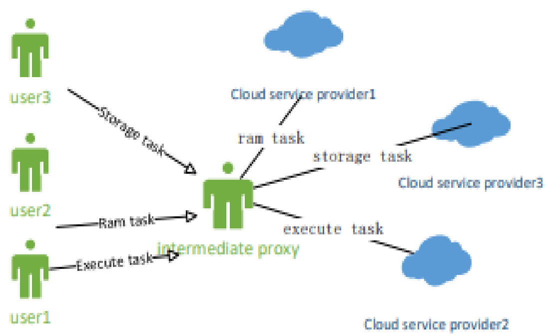

Figure 13 Scheduling process (Chen, 2023)

The experiments were intricately designed to address the challenge of efficiently managing multi-user tasks originating from lower-tier mobile devices, which were then dispatched to the nearest cloud server for processing. This complex allocation process was facilitated by a tailored multi-objective dispatching model and a well-conceived kinetic planning algorithm (fig14), specifically tailored to the intricacies of multi-cloud environments.

Algorithm： A dynamic programming driven multi-objective optimised task dispatching algorithm

Input: n-Tasks to be processed, a[]-The set of time required by server a to process the task,b[]-The set of time required by server b to process the task,sum-Initialize array

Output: result = get_result(a, b, n)

Procedure:

```
sum= a[0];
for(int k = 1;k < n;k++) sum += a[k];
f[k][x] = f[k-1][x]+b[k];
f[k][x] = min(f[k-1][x-a[k]],f[k-1][x]+b[k]);
if(k == n-1) val = max(x, f[k][x]);
if(val < result)result = val  return result;
result = get_result(a, b, n);
```

Figure 14 Description of the algorithm (Chen, 2023)

Notably, the research delved into the intricate realm of resource allocation within cloud environments, culminating in the development of a multi-goal task dispatching model fine-tuned to cloud-specific nuances (fig14). Additionally, they ventured into the formalization of the optimization problem for multi-objective tasks, resulting in a solution algorithm grounded in dynamic programming principles.

Crucially, the study rigorously examined the proposed method's validity through meticulous simulation experiments, affirming its effectiveness within the controlled simulated environment (fig15,16).

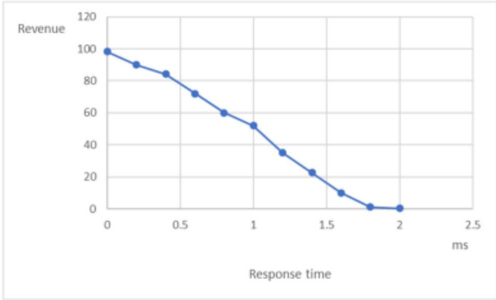

Figure 15 increase of response time (Chen, 2023)

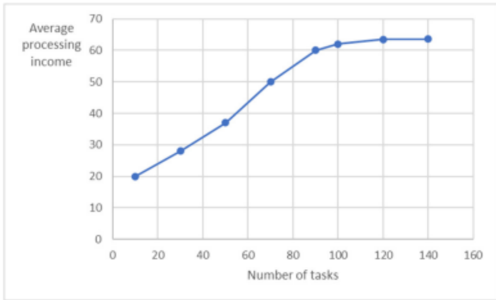

Figure 16 Cost effectivity (Chen, 2023)

However, it's worth noting a limitation: the study lacks conclusive evidence regarding real-world applicability. Though the research provides a solid foundation, practical validation in multi-cloud scenarios is crucial.

Those recaches highlights the crucial importance of trust assessment in multi-cloud settings, emphasizing the need to consider various quality attributes when allocating resources to cloud infrastructures.

II. Load Balancing Techniques

Various load-balancing techniques and strategies aim to distribute workloads evenly across cloud providers, thereby preventing overutilization or underutilization of resources.

Saif et al. (2023) presents a groundbreaking solution, the CSO-ILB (Chicken Swarm Optimization for Load Balancing), designed to revolutionize load distribution within containerized multi-cloud environments (fig17).

This innovative approach leverages a chicken swarm optimization algorithm, drawing inspiration from the efficient foraging behavior of chickens. It excels in selecting the optimal under-loaded container, effectively distributing workloads across available containers, and addressing load balancing challenges.

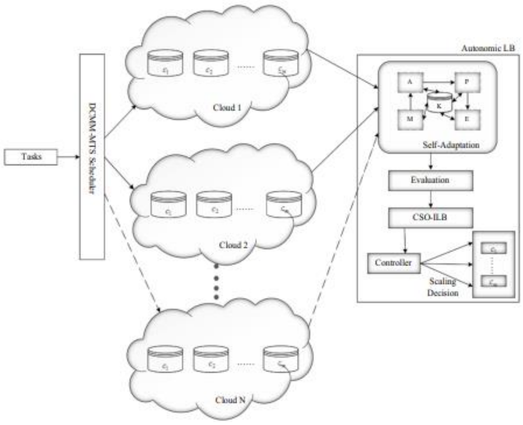

Figure 17 proposed solution (Saif et al. 2023)

However, CSO-ILB's contributions extend beyond load balancing. It showcases impressive horizontal and vertical scalability, achieved through adaptive scaling options.

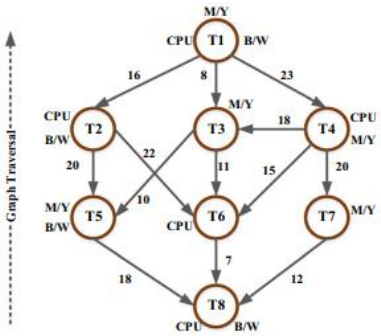

Figure 18 Make-Span scheduling. (Saif et al. 2023)

A notable feature is its autonomic multi-loop self-adaptation system, which dynamically responds to changes within the multi-cloud ecosystem, thereby enhancing the efficiency of scaling decisions (fig18).

To assess its effectiveness, the research subjected CSO-ILB to rigorous simulations within a containerized multi-cloud environment, employing the ContainerCloudsim toolkit. These simulations entailed a comprehensive comparison with existing algorithms, showcasing CSO-ILB's superiority across a diverse spectrum of metrics (fig19).

Input: C_U, C_O
Output: Pareto Solution S indicating the optimally chosen containers for task migration

Initialize all the parameters R_n, H_n, C_n, M_n and B
Initialize the chickens in the swarm randomly as $C_{U_i}(i=1,2,\ldots,y)$
Initialize the total count of iterations as Max_{itr}
While $T_r < Max_{itr}$ do
If $(T_r \% B = 0)$ then
Establish the hierarchical order through ranking of chickens
Partition the swarm group and identify the mother-child relationship
End if
For $(i=1)$ do
If $(i == rooster)$ do
Perform local update of the rooster's location using (25)
End if
If $(i == hen)$ do
Perform local update of the hen's location using (27)
End if
If $(i == chick)$ do
Perform local update of the chick's location using (30)
End if
Estimate the fitness of the obtained solution using (25)
If the solution outperforms the older one $\rightarrow$ update location
End for
Label the best solution as pareto optimal solution S
End while
Return S

Figure 19 CSO-ILB Algorithm (Saif et al. 2023)

These metrics encompassed CPU utilization, make-span, response time, execution cost, reliability, energy utilization, idle time, and task migrations.

Beyond load balancing, CSO-ILB aligns seamlessly with the objectives of both cloud users and providers. It represents a significant leap forward in addressing the intricate challenges of load balancing within modern multi-cloud environments.

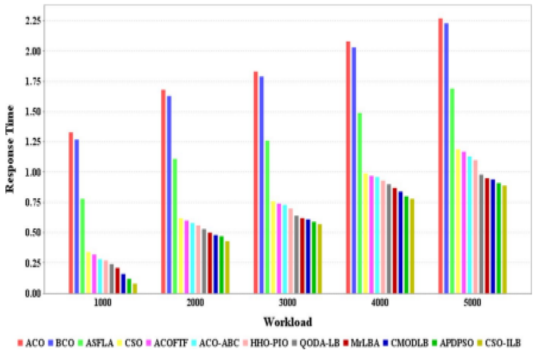

Figure 20 Response Time Improvement (Saif et al. 2023)

In doing so, it contributes substantially to the ever-evolving landscape of cloud computing, promising enhanced efficiency and performance in this dynamic domain (fig20).

Zhang et al. (2021) present a comprehensive exploration of cloud computing, with a specific focus on Continuous Writing Applications (CWAs) that generate substantial real-time data. Recognizing the growing demand for efficient data transportation, storage, and analysis, the study introduces an innovative multi-cloud load-balancing architecture designed to address user resource requirements while optimizing costs.

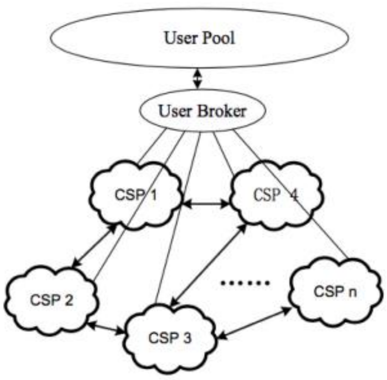

Figure 21CSP user and broker (Zhang et al. 2021)

The investigation centers on Infrastructure-as-a-Service (IaaS) Cloud Service Providers (CSPs) and utilizes a mathematical model to encapsulate crucial factors, including user resource needs, CSP utility costs, and inter-cloud communications (fig21). Furthermore, the paper delves into the collaborative potential of multi-cloud setups to enhance data backup and fortify system reliability.

Step 1: ***Initialization***
Construct the set of load paths, P_0, using *SaLPF*.
For $k = 1$ to m, where m = $|\mathcal{C}_0|$.
Set $B_k = 0, S_k = 0, C_k = 0$.
End For.
Calculate the FDL of each path in P_0 by using (10).
Construct $\mathcal{U}$, sorting the users according to SA, SB, or SC
Step 2: ***Main Loop***
For each priority j.
For $i = 1$ to n, where $n = |\mathcal{U}|$.
Sort the path in P_0 according to the FDL from lowest to highest.
For $l = 1$ to $|P_0|$.
Set CSP_{i-edge} and CSP_{i-back} along path P_l;
If $r_i^j + \mathcal{R}_{i-edge} \preceq \bar{\mathcal{R}}_{i-edge}$ and $r_{i-back}^j + \mathcal{R}_{i-back} \preceq \bar{\mathcal{R}}_{i-back}$;
Assign u_i to path P_l;
Set $\mathcal{R}_{i-edge} = \mathcal{R}_{i-edge} + r_i^j$ and $\mathcal{R}_{i-back} = \mathcal{R}_{i-back} + r_{i-back}^j$;
Calculate the FDL of all paths in P_0 that involve CSP_{i-edge} or CSP_{i-back} by using (10).
Break Loop of l;
Else
Continue;
End For
If $l > P_0$
Set a warning "Out of Recourses, User u_i cannot be scheduled!";
End If
End For
Step 3: ***End Algorithm***
Output a list of users that cannot be scheduled.

Table 3
Algorithm 2: Sub-algorithm of Load Path Finding (*SaLPF*).

Set $P_0 = \emptyset$
For $k = 1$ to n
For $l = 1$ to $|\mathcal{N}_i|$, where $\mathcal{N}_i$ is the set of neighboring CSPs of CSP_i.
Construct the load path (CSP_i, CSP_l).
Include (CSP_i, CSP_l) into P_0.
End For
End For
Return P_0.

Figure 22 OSMC algorithm (Zhang et al. 2021)

In this study, they introduce the novel cooperative multi-cloud load-balancing algorithm, aptly named Optimal user Scheduling for Multi-Cloud (OSMC) (fig22). Engineered to fulfill all user resource requirements while minimizing costs per user, OSMC leverages the mathematical model that encompasses key elements such as user demands, CSP expenses, and inter-cloud communication (fig22). Comparative analysis, pitting OSMC against the round-robin algorithm selected for its simplicity and efficiency in job scheduling, underscores OSMC's superior performance in terms of cost efficiency.

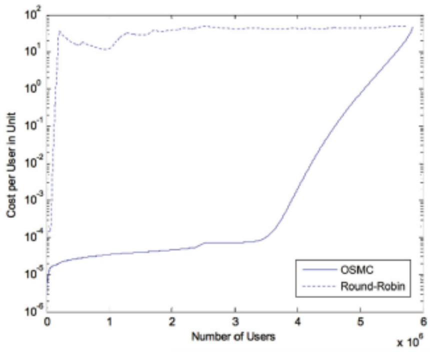

Figure 23 Low consumption (Zhang et al. 2021)

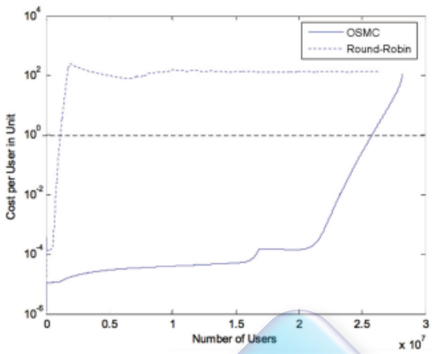

Figure 24 standard comparison (Zhang et al. 2021)

Through meticulous simulations of OSMC and the round-robin algorithm, the research reveals OSMC's distinct advantages in cost-effectiveness, system scalability, and efficient utilization of CSP resources within a 10 CSP Multi-Cloud environment. Additional experiments further substantiate OSMC's scalability across various multi-cloud sizes, affirming its practicality and ease of implementation. Lastly, this study underscores OSMC's scalability, flexibility, and efficiency across diverse practical scenarios (fig23,24), signifying its potential as a valuable resource provisioning strategy for multi-cloud load balancing.

Phalak, (2023) research tackles the dynamic management of machine learning inference workloads across diverse cloud providers. The study introduces the Multi-cloud System for Inference Request Management (mSIRM), designed for the edge-fog-cloud continuum (fig25). mSIRM leverages multiple cloud vendors and fog computing resources to minimize Service Level Objective (SLO) violations.

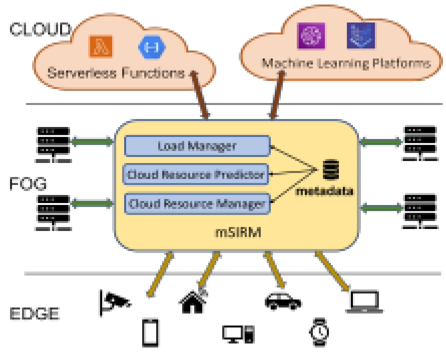

Figure 25 MSIRM Architecture (Phalak, 2023)

The research meticulously evaluates and compares various edge-cloud frameworks (fig26), with a specific focus on machine learning and serverless platforms from major cloud providers. It also explores related research and algorithms for task scheduling and resource allocation within this paradigm.

Instance	α	β	γ
Fog	-0.0000100931	0.1157314712	-4.338963016
SageMaker	-0.0000725881	0.1904910006	12.7724733681

Figure 26 Constants of edge-fog-application (Phalak, 2023)

The proposed methodology prioritizes SLO-aware and cost-effective execution of inference requests through Machine Learning as a Service (MLaaS), Function as a Service (FaaS) platforms, and real-time instance state data. mSIRM stands out by employing a regression model-based approach for workload balancing across Fog, MLaaS (fig27), and serverless platforms, optimizing cost efficiency while reducing SLO violations, distinguishing it from prior studies using Reinforcement Learning (RL) algorithms.

Simulation results illustrate mSIRM's effectiveness in distributing AI workloads between Fog and cloud services, reducing SLO violations significantly.

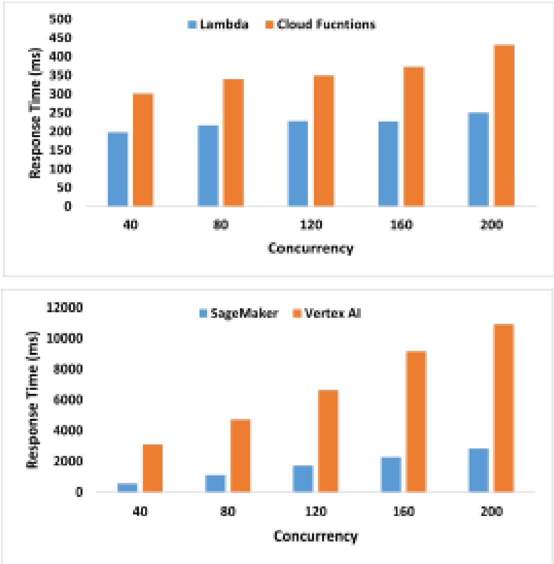

Figure 27 Load Split 50% Between Each Faas & Mlaas (Phalak, 2023)

AWS SageMaker outperforms GCP Vertex AI, and AWS Lambda exhibits lower latency than GCP Cloud Functions for warm starts, albeit with slightly higher costs. In summary, Phalak's study demonstrates mSIRM's practicality in optimizing AI workload distribution, effectively managing costs while mitigating SLO violations in the edge-fog-cloud landscape.

Selvakumar, (2020) in his study introduces a novel load balancing solution tailored for multi-cloud setups, harnessing an optimization algorithm rooted in binary ant colony optimization. This approach meticulously distributes workloads among virtual machines, contingent on traffic patterns and a preliminary evaluation of each machine's capabilities. The integration of online services simplifies resource access.

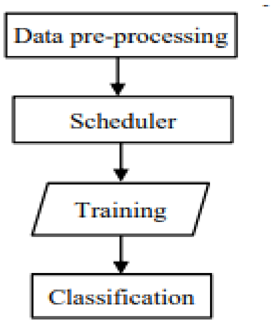

Figure 28 NNS Method (Selvakumar, 2020)

Furthermore, they employ an advanced load-balancing approach that integrates improved neural network scheduling and workflows, thereby enhancing the utilization of underused virtual machines. This redesigned resource and load-balancing framework contributes to a boost in overall efficiency.

Various load balancing algorithms, like GA and Cloud Balancing Mechanism, strive to evenly distribute workloads in cloud infrastructures while minimizing mission scope. They employ an optimization algorithm designed for binary ant colonies, known for cost-effectiveness. Simulations and analyses consistently affirm the system's efficiency, especially in scenarios with more virtual machines, as indicated by average makespan, time study, and cost efficiency metrics (fig29,30).

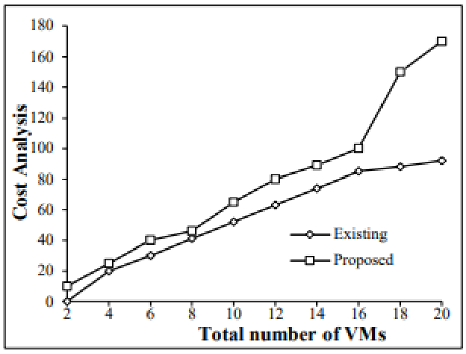

Figure 29 Make span average (Selvakumar, 2020)

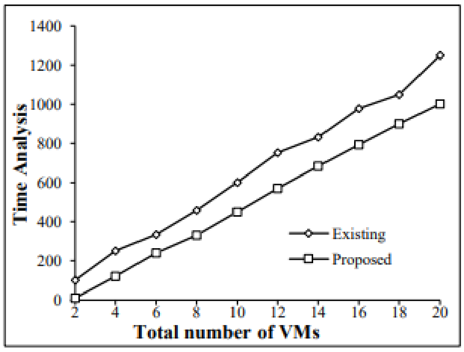

Figure 30 cost Average (Selvakumar, 2020)

Perform a cost-benefit analysis to assess the economic implications of resource allocation and load balancing strategies. Consider cloud provider pricing models and evaluate the trade-offs between performance and cost.

3. Comparison of Current Work and Real-World Validity

In this section, we'll assess how prior work validates and applies dynamic resource allocation and load balancing research in real-world multi-cloud scenarios.

Trust-Based Allocation (Alam, Fadlullah, and Choudhury, 2021): This technique introduces trust evaluation into resource allocation, considering quality attributes like availability and reliability. It utilizes a genetic algorithm for optimization, demonstrating effectiveness in simulations. Trust-Based Allocation excels in ensuring resource allocation aligns with trust levels but may require additional validation in practical scenarios.

Multi-Objective Framework (Alyas et al., 2023): Alyas et al.'s framework prioritizes QoS and resource allocation, especially in decentralized multi-cloud setups. It offers scalability and collaboration benefits. However, its real-world feasibility and adaptability to diverse use cases require further exploration.

Hybrid Optimization (Selvapandian and Santhosh, 2022): The BAPSO algorithm combines the strengths of bat optimization and particle swarm optimization for resource allocation. It outperforms traditional methods in terms of efficiency and energy consumption. Its adaptability and cost-effectiveness make it a strong contender.

Multi-Objective Task Scheduling (Chen, 2023): Chen's approach using dynamic programming address's multi-objective task scheduling efficiently. However, its real-world applicability remains a point of consideration.

Load Balancing Techniques (Saif et al., 2023; Zhang et al., 2021; Phalak, 2023; Selvakumar, 2020): In the domain of load balancing, various techniques like CSO-ILB, OSMC, and BAPSO offer different advantages, including scalability, cost-efficiency, and real-time adaptability.

4. Conclusion

In conclusion, dynamic resource allocation and load balancing are paramount in multi-cloud environments, where organizations strive to optimize resource utilization and enhance performance while managing costs. This research paper has reviewed and evaluated current research on this topic, emphasizing its critical relevance in the realm of cloud computing.

The analysis of real-world validity demonstrates the potential benefits and challenges associated with existing tools and strategies. While they offer valuable capabilities, integration complexities and practical considerations must be addressed. As organizations continue to embrace multi-cloud strategies, the development and refinement of dynamic resource allocation and load balancing solutions remain essential.

This paper underscores the importance of tailored multi-cloud solutions and offers insights into how organizations can harness the benefits of dynamic resource allocation and load balancing while efficiently managing their multi-cloud environments. The future of cloud computing relies on the ability to adapt to evolving workloads, optimize resource usage, and ensure cost-effective, high-performance operations.

REFERENCES

Adewojo, A.A. and Bass, J.M. (2022) 'Multi-cloud Load Distribution for Three-tier Applications', *International Conference on Cloud Computing and Services Science, CLOSER - Proceedings*, pp. 296–304. Available at: https://doi.org/10.5220/0011092100003200.

Alam, A.B.M.B., Fadlullah, Z.M.D. and Choudhury, S. (2021) 'A Resource Allocation Model Based on Trust Evaluation in Multi-Cloud Environments', *IEEE Access*, 9, pp. 105577–105587. Available at: https://doi.org/10.1109/ACCESS.2021.3100316.

Alyas, T. *et al.* (2023) 'Optimizing Resource Allocation Framework for Multi-Cloud Environment', *Computers, Materials and Continua*, 75(2), pp. 4119–4136. Available at: https://doi.org/10.32604/cmc.2023.033916.

Chen, X. (2023) 'Multi-objective optimization task scheduling method based on dynamic programming for multi-cloud environment', *2023 4th International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, pp. 278–283. Available at: https://doi.org/10.1109/ISPDS58840.2023.10235565.

Phalak, C. (2023) 'mSIRM : Cost-Efficient and SLO-aware ML Load Balancing on Fog and Multi-Cloud Network', pp. 19–26. Available at: https://doi.org/10.1145/3589013.3596676.

Rodigari, S. *et al.* (2021) 'Performance Analysis of Zero-Trust multi-cloud', *IEEE International Conference on Cloud Computing, CLOUD*, 2021-Septe, pp. 730–732. Available at: https://doi.org/10.1109/CLOUD53861.2021.00097.

Saif, M.A.N. *et al.* (2023) *CSO-ILB: chicken swarm optimized inter-cloud load balancer for elastic containerized multi-cloud environment*, *Journal of Supercomputing*. Springer US. Available at: https://doi.org/10.1007/s11227-022-04688-w.

Saxena, D., Gupta, R. and Singh, A.K. (2021) 'A Survey and Comparative Study on Multi-Cloud Architectures: Emerging Issues And Challenges For Cloud Federation'. Available at: http://arxiv.org/abs/2108.12831.

Selvakumar, S. (2020) 'Enhanced Neural Network Scheduling For Load Balancing In Multi-Cloud', *Journal On Data Science And Machine Learning*, Volume: 01(June), pp. 77–80.

Selvapandian, D. and Santhosh, R. (2022) 'A Hybrid Optimized Resource Allocation Model for Multi-Cloud Environment Using Bat and Particle Swarm Optimization Algorithms', *Computer Assisted Methods in Engineering and Science*, 29(1–2), pp. 87–103. Available at: https://doi.org/10.24423/cames.405.

Zhang, B. *et al.* (2021) 'A novel cooperative resource provisioning strategy for Multi-Cloud load balancing', *Journal of Parallel and Distributed Computing*, 152, pp. 98–107. Available at: https://doi.org/10.1016/j.jpdc.2021.02.003.