International Journal of Innovative Research and Scientific Studies, 8(5) 2025, pages: 2030-2043

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

A user-driven MILP framework for cost-efficient and performance-Aware Iaas resource allocation

Mahmoud Aljawarneh[1], Qais Al-Na'amneh[1], Rahaf Hazaymih[2], Ayoub Alsarhan[3], Khalid Hamad Alnafisah[4*], Nayef H. Alshammari[5], Sami Aziz Alshammari[6]

[1]*Faculty of Information Technology, Applied Science Private University, Amman, Jordan.*
[2]*Dept. Computer Science, Jordan University of Science and Technology, Irbid, Jordan.*
[3]*Department of Information Technology, Faculty of Prince Al-Hussein of Information Technology, The Hashemite University, Zarqa, Jordan.*
[4]*Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University Rafha, Saudi Arabia.*
[5]*Departmant of Computer Science, Faculty of Computers and Information Technology, University of Tabuk, Tabuk. Saudi Arabia.*
[6]*Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia.*

Corresponding author: Khalid Hamad Alnafisah (*Email: khalid.alnafisah@nbu.edu.sa*)

Abstract

Cloud computing has indelibly reshaped contemporary IT infrastructure by offering scalable and economically viable resource provisioning. Infrastructure-as-a-Service (IaaS), a key component, provides flexible computing resources, yet optimizing their allocation balancing energy, latency, and provisioning costs remains a complex challenge. This research introduces a user-driven Infrastructure-as-a-Service (IaaS) optimization framework, leveraging Mixed Integer Linear Programming (MILP). This framework is meticulously designed for cost-efficient resource management and performance-aware virtual machine (VM) placement. A core feature is its facilitation of dynamic user-configurable parameters, specifically cost-prioritization weights (α, β, γ), endowing it with significant adaptability to diverse operational requisites. Through comprehensive simulation studies involving systematic variation of these weights and workload scaling, the framework's efficacy is demonstrated in optimizing VM placement across distributed servers. This approach achieves substantial improvements in resource utilization and cost management while rigorously adhering to performance constraints. Ten distinct comparative analyses visually articulate the inherent trade-offs in this optimization landscape.

Keywords: Cloud computing, Cost-efficient, IaaS, MILP, Performance optimization, Performance-aware, Resource allocation, User-driven optimization, Virtual machine placement.

DOI: 10.53894/ijirss.v8i5.9369
Funding: This work is supported by the Deanship of Scientific Research at Northern Border University, Arar, Kingdom of Saudi Arabia (Grant number: NBU-FFR-2025-3555-03).
History: Received: 7 July 2025 / **Revised:** 11 August 2025 / **Accepted:** 14 August 2025 / **Published:** 19 August 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

The relentless expansion of digital services and data-intensive applications has unequivocally solidified cloud computing's stature as the preeminent paradigm for agile, on-demand IT infrastructure delivery. This transformative technology underpins a significant portion of modern digital enterprise, offering unprecedented scalability and elasticity in resource provisioning [1]. Among its variegated service models, Infrastructure-as-a-Service (IaaS) constitutes a foundational stratum, empowering users with access to elemental computing capabilities such as virtual machines (VMs), persistent storage, and configurable networks, consequently abstracting the inherent complexities of direct physical hardware stewardship [2]. For the entities orchestrating these expansive, often geographically federated ecosystems, operational excellence ascends to paramount importance. The administration of potentially myriad physical servers, dispersed across numerous data centers interconnected by high-velocity network backbones, introduces profound logistical and economic intricacies. Central to navigating this multifaceted environment is the Virtual Machine Placement (VMP) predicament: the judicious strategic allocation of a multitude of user-solicited VMs onto the available physical server (PS) substrate [3].

This allocation endeavor transcends rudimentary resource matching; it represents a critical fulcrum influencing the provider's aggregate operational expenditure profile and the perceived quality of service [4]. The total cost materializes as a complex amalgamation of diverse contributing factors. Energy consumption emerges as a substantial constituent, driven by both active computation and significant quiescent power draw in large server farms, making energy-aware VMP a critical research area [5]. Network latency, engendered by the physical separation between data centers housing intercommunicating VMs or between VMs and their designated end-users, directly impinges upon application responsiveness and user experience [6], thereby constituting an implicit economic burden necessitating meticulous governance. Furthermore, the unceasing operation and provisioning of hardware elements contribute ancillary costs attributable to component wear, routine maintenance overheads, and the network bandwidth appropriated by VM migration activities or inter-data center communication [7]. An efficacious minimization of this composite cost structure mandates a sophisticated, multi-objective approach capable of arbitrating these frequently conflicting imperatives [8].

The optimization landscape [9] is further circumscribed by exacting operational mandates. Each physical server possesses inherently finite capacities for indispensable resources encompassing CPU cycles (C), Random Access Memory (M), data storage (S), and network bandwidth (B), which cannot be oversubscribed without precipitating performance degradation or service instability. Concurrently, every individual VM necessitates an unambiguous, singular assignment to one physical server within the distributed infrastructure, a fundamental tenet of virtualization management. Quality-of-Service (QoS) assurances [10] frequently translate into explicit latency thresholds, stipulating that communication delays between specified VM cohorts or from VMs to defined geographical locales must remain beneath predetermined maxima (L_{max}) [11]. Moreover, the imperative of service continuity compels adherence to stringent reliability criteria, thereby circumscribing the acceptable probability of service disruptions ($P_{threshold}$) emanating from server over-utilization or correlated failure events within a single geographic or fault domain. The combinatorial explosion inherent in assigning potentially thousands of VMs (m) to hundreds or thousands of physical servers (n), while simultaneously respecting these heterogeneous constraints and minimizing a multi-component cost function, presents a computationally formidable undertaking classified as NP-hard [12]. Naive heuristics or simplistic greedy algorithms may yield markedly suboptimal or even infeasible allocation schemas within such intrinsically complex operational scenarios [13].

Addressing this multifaceted optimization challenge with requisite rigor necessitates a robust, mathematically principled methodology. This manuscript proffers a framework architected around Mixed Integer Linear Programming (MILP), an optimization technique of considerable potency, particularly adept at resolving problems characterized by discrete decision variables (such as the binary determination of assigning a specific VM to a particular server) in conjunction with continuous variables and linear constraint systems [14]. By formally articulating the VM placement conundrum as an MILP problem, a systematic exploration of the vast solution space becomes feasible, enabling the identification of allocations that demonstrably minimize the defined total operational cost, encompassing pivotal energy, latency, and provisioning elements [15]. The model explicitly incorporates constraints reflecting finite resource capacities, inviolable VM assignment uniqueness, pertinent energy consumption characteristics (linearized where appropriate for MILP tractability), maximum permissible latency figures, and overarching system reliability targets. This integrated

paradigm aims to furnish IaaS providers with a quantifiable and robust instrument for refining their resource allocation strategies, thereby fostering enhanced cost-efficiency, unwavering performance standards, and promoting sustainable energy consumption patterns within their large-scale, distributed infrastructures [16].

The rapid proliferation of cloud computing services has instigated an exponential surge in the demand for computational resources, rendering the efficient allocation of VMs across the expansive cloud infrastructure a pivotal and pressing challenge [11]. Conventional allocation methodologies often depend on static policies that demonstrate limited adaptability to fluctuating workload dynamics and evolving resource constraints. Such methods can precipitate excessive energy dissipation, suboptimal resource utilization, inflated operational expenditures, and augmented latency in service delivery pipelines. A significant lacuna in extant optimization models is their constrained flexibility in seamlessly incorporating user-defined constraints and preferences, a limitation that curtails their practical applicability within heterogeneous, real-world cloud deployment contexts [17]. This research endeavors to surmount these limitations through the development of an advanced optimization framework, one that dynamically allocates VMs while concurrently minimizing operational costs and rigorously upholding stipulated performance benchmarks.

This research presents a novel IaaS optimization framework that synergistically integrates user-defined constraints with the mathematical rigor of Mixed Integer Linear Programming (MILP) to achieve demonstrably efficient VM allocation within contemporary cloud environments. The architected framework empowers users to articulate specific trade-offs among cost, latency, and performance metrics by adjusting weighting parameters (α, β, γ) for energy, latency, and provisioning costs, respectively. This renders it exceptionally adaptable to a diverse array of cloud deployment scenarios and bespoke operational objectives. Distinct from many prevalent models, our approach comprehensively considers dynamic server capacities, inherent workload variations, and critical energy efficiency parameters, thereby furnishing an optimized allocation strategy that markedly enhances overall resource utilization. Extensive experimental analysis, detailed herein, demonstrates the framework's effectiveness in reducing operational costs, balancing server loads, and improving overall system performance across various simulated conditions.

The remainder of this paper is organized as follows: Section II surveys pertinent prior research. Section III details the system model and formalizes the optimization problem. Section IV describes the MILP-based optimization framework and the allocation algorithm. Section V presents the comprehensive simulation setup, defines the performance metrics, and discusses the results derived from systematic experimentation. Finally, Section VI concludes the paper and suggests avenues for future investigation.

2. Related Work

The challenge of Virtual Machine Placement (VMP) in cloud data centers, aiming to optimize resource utilization, minimize operational costs, and enhance performance, has been a subject of extensive research, yielding a diverse spectrum of proposed solutions [18, 19]. Existing approaches can be broadly categorized into heuristic/metaheuristic methods, exact optimization techniques, and learning-based strategies.

Heuristic and metaheuristic algorithms have been widely explored due to their capacity to find near-optimal solutions within computationally tractable timeframes for large-scale VMP instances [17, 20]. Genetic Algorithms (GAs) have been applied to minimize energy consumption and resource wastage [21] while Simulated Annealing (SA) has been used to escape local optima in search of better placements [22]. Ant Colony Optimization (ACO) [23] and Particle Swarm Optimization (PSO) [23, 24] represent other prominent bio-inspired techniques adapted for VMP, often excelling in exploring vast solution spaces. Various First-Fit (FF) and Best-Fit (BF) derived heuristics, such as First-Fit Decreasing (FFD) for resource packing, are also common due to their simplicity and speed [25, 26]. However, a general limitation of these methods is the lack of guaranteed optimality and potential sensitivity to parameter tuning [27]. More critically for our work, many heuristic approaches, while effective for general cost or energy reduction, frequently lack sophisticated mechanisms to incorporate fine-grained, user-specified constraints or to dynamically adapt to varying preferences regarding the trade-offs between cost, latency, and performance without significant re-engineering or ad-hoc modifications [28, 29].

Exact optimization methods, predominantly based on Integer Linear Programming (ILP) or Mixed Integer Linear Programming (MILP), offer the advantage of finding provably optimal solutions for well-defined VMP problem formulations [30]. Numerous studies have formulated VMP as an MILP problem, targeting objectives such as minimizing energy consumption [31], operational costs [26, 32], resource fragmentation [33], or a combination thereof. Some MILP models also incorporate network-awareness [34]. While powerful in their ability to guarantee optimality, the primary impediment of exact methods is their computational complexity (NP-hardness), which can render them intractable for very large problem instances or unsuitable for highly dynamic, real-time environments where placement decisions must be made rapidly [29, 35]. Our work leverages MILP for its optimality guarantees but focuses on a formulation that strategically incorporates user-driven parameters to guide the optimization towards solutions that reflect specific operational priorities, aiming for applicability in strategic planning rather than instantaneous tactical decisions.

More recently, machine learning (ML) and reinforcement learning (RL) techniques have gained significant traction for dynamic VM allocation and resource management in cloud environments [36, 37]. RL agents, such as those based on Q-learning or Deep Q-Networks (DQN), can learn optimal placement policies through interaction with the cloud environment, adapting to changing workloads and resource availability without explicit system modeling [38]. Supervised learning models can be used for workload prediction to inform placement decisions [39] while unsupervised learning might help in server clustering or anomaly detection. These approaches show considerable promise for handling the inherent dynamism and uncertainty of cloud systems. However, they often necessitate substantial volumes of training data and extensive training periods [40]. Furthermore, ensuring that learned policies strictly adhere to all hard operational constraints and user-

specific Quality of Service (QoS) requirements can be challenging [41]. The ”black-box” nature of some complex ML models can also obscure the rationale behind specific placement decisions, which can be a concern for providers requiring transparent and auditable allocation processes [42, 43].

Several frameworks have attempted to address multi-objective optimization in VMP, recognizing the conflicting nature of goals such as cost reduction, performance enhancement, and energy saving [44]. These often involve transforming multiple objectives into a single objective function using scalarization techniques, such as weighted sum methods [42] or employing Pareto optimality concepts to find a set of non-dominated solutions [45]. While valuable, the configuration of these weights in scalarization methods often requires domain expertise or can be static, limiting adaptability. Similarly, selecting a single solution from a potentially large Pareto front still requires a higher-level decision-making process that may not be easily automated according to dynamic user preferences [46].

Our proposed framework distinguishes itself by architecting the MILP model to be directly and dynamically influenced by user-configurable parameters (specifically, the weights α, β, γ for energy, latency, and provisioning costs). This explicit integration allows for a more transparent and adaptable optimization process compared to static heuristics, complex ML models where fine-grained preference control is less direct, or multi-objective methods where weight setting is often a pre-optimization step. It builds upon the strengths of MILP for finding optimal solutions within the defined model while mitigating the rigidity of traditional formulations by embedding user preferences directly into the objective function, guiding the decision-making core, thereby facilitating a practical approach to user-driven resource allocation.

3. System Model and Problem Formulation

This section delineates the architectural model of an IaaS provider’s infrastructure and subsequently formalizes the VM allocation problem as an optimization task.

3.1. Network and Resource Model

An IaaS provider’s infrastructure is conceptualized as a distributed system comprising multiple Data Centers (DC), interconnected via a high-speed, wide-area backbone network. Each data center $DC_k \in$ DC hosts a collection of Physical Servers (PS_k), which in turn execute the Virtual Machines (VMs) allocated to various users. For simplicity in this formulation, we consider a global set of physical servers. Key components and notations:

- PS = $\{PS_1, PS_2, ..., PS_n\}$: The global set of n physical servers.
- VM = $\{VM_1, VM_2, ..., VM_m\}$: The set of m virtual machines requested by users that require placement. (Changed k to m for VM count to avoid conflict with k often used as an index)
- R = $\{C, M, S, B\}$: The set of critical resource types considered: CPU (C), Memory (M), Storage (S), and network Bandwidth (B).
- $R_{j,r}^{avail}$: The available capacity of resource $r \in$ R on physical server PS_j.
- $R_{i,r}^{req}$: The required amount of resource $r \in$ R by virtual machine VM_i.

3.2. Cost Components

The total operational cost targeted for minimization is a composite function of energy consumption, interVM or VM-to-user latency implications, and general provisioning overheads. User-driven parameters (α, β, γ) are introduced as weights to modulate the relative importance of these base cost components in the objective function.

- $E_{base,j}$: Base energy cost factor associated with PS_j.
- $L_{base,j}$: Base latency cost factor associated with PS_j.
- $P_{base,j}$: Base provisioning cost factor associated with PS_j.

These base factors represent raw costs per server (or per VM hosted on that server, depending on interpretation) before weighting.

3.3. Decision Variables

The primary decision in the VMP problem is the assignment of each VM to a specific PS.

• x_{ij}: A binary variable defined as: $x_{ij} = 1$, if VM_i is allocated to PS_j. $x_{ij} = 0$, otherwise.

3.4.. Objective Function (Cost Minimization)

The primary objective is to minimize the total weighted operational cost. The cost is considered per VM assignment to a server.

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \cdot (\alpha \cdot E_{base,j} + \beta \cdot L_{base,j} + \gamma \cdot P_{base,j}) \tag{1}$$

Here, α, β, γ are user-defined weights such that $\alpha, \beta, \gamma \in [0,1]$ and typically $\alpha + \beta + \gamma = 1$ (though not strictly necessary if costs are normalized).

3.4. Constraints

The allocation must satisfy several operational and resource constraints.

1) Resource Capacity Constraint: The total resources consumed by VMs allocated to any physical server must not exceed its available capacity for each resource type $r \in$ R.

$$\forall j \in \{1, \ldots, n\}, \forall r \in \mathcal{R}: \quad \sum_{i=1}^{m} x_{ij} \cdot R_{i,r}^{req} \leq R_{j,r}^{avail} \tag{2}$$

2) VM Placement Constraint: Each virtual machine VM_i must be assigned to exactly one physical server.

$$\forall i \in \{1, \ldots, m\}: \quad \sum_{j=1}^{n} x_{ij} = 1 \tag{3}$$

3) Energy Consumption Considerations (Informing $E_{base,j}$): The base energy cost $E_{base,j}$ can be derived from server power models. A common model for server PS_j's power consumption P_{consj} is:

$$Pconsj = Pidlej + (Pmaxj - Pidlej) \times Uj \tag{4}$$

where P_{idle} and P_{max} are the idle and maximum power consumption of PS_j, and U_j is its utilization. $E_{base,j}$ could represent an average incremental power draw per VM or the cost of the server being active.

4) Latency Constraint (Implicit/Explicit): Latency considerations are primarily incorporated through $L_{base,j}$ and its weight β. Explicit constraints, e.g., $L_{effective} \leq L_{max\ vmi}$ for specific VMs, could be added if required, potentially increasing model complexity.

5) Reliability Constraint (Conceptual): Reliability ($P_{failure} \leq P_{threshold}$) is often managed by limiting server utilization (proxy for failure risk) or ensuring replica placement in different fault domains, translating to additional constraints if explicitly modeled.

6) Maximum VMs per Server Constraint: A practical limit on the number of VMs per server, V_{maxj}, can be imposed:

$$\forall j \in \{1, \ldots, n\}: \quad \sum_{i=1}^{m} x_{ij} \leq V_{max_j} \tag{5}$$

4. MILP-Based Optimization Framework

The core of the proposed IaaS allocation framework is an MILP model designed to solve the VM placement problem efficiently, considering the multifaceted objectives and constraints. Figure 1 presents the high-level architecture of this framework, outlining the key components and their interactions. The subsequent subsections detail the optimization problem formulation and the algorithmic approach.

4.1. Optimization Problem Formulation Summary

We formulate the VM allocation as an MILP problem. The objective is to minimize Equation 1, subject to:

1) Resource capacity constraints (Equation 2)
2) VM placement constraints (Equation 3)
3) Maximum VMs per server constraints (Equation 5)
4) (Implicitly) Energy efficiency, latency, and reliability considerations via weighted cost components and potential additional constraints.

The user-driven nature is realized through the selection of weights (α,β,γ) and specific constraint bounds (e.g., $Vmax_j$).

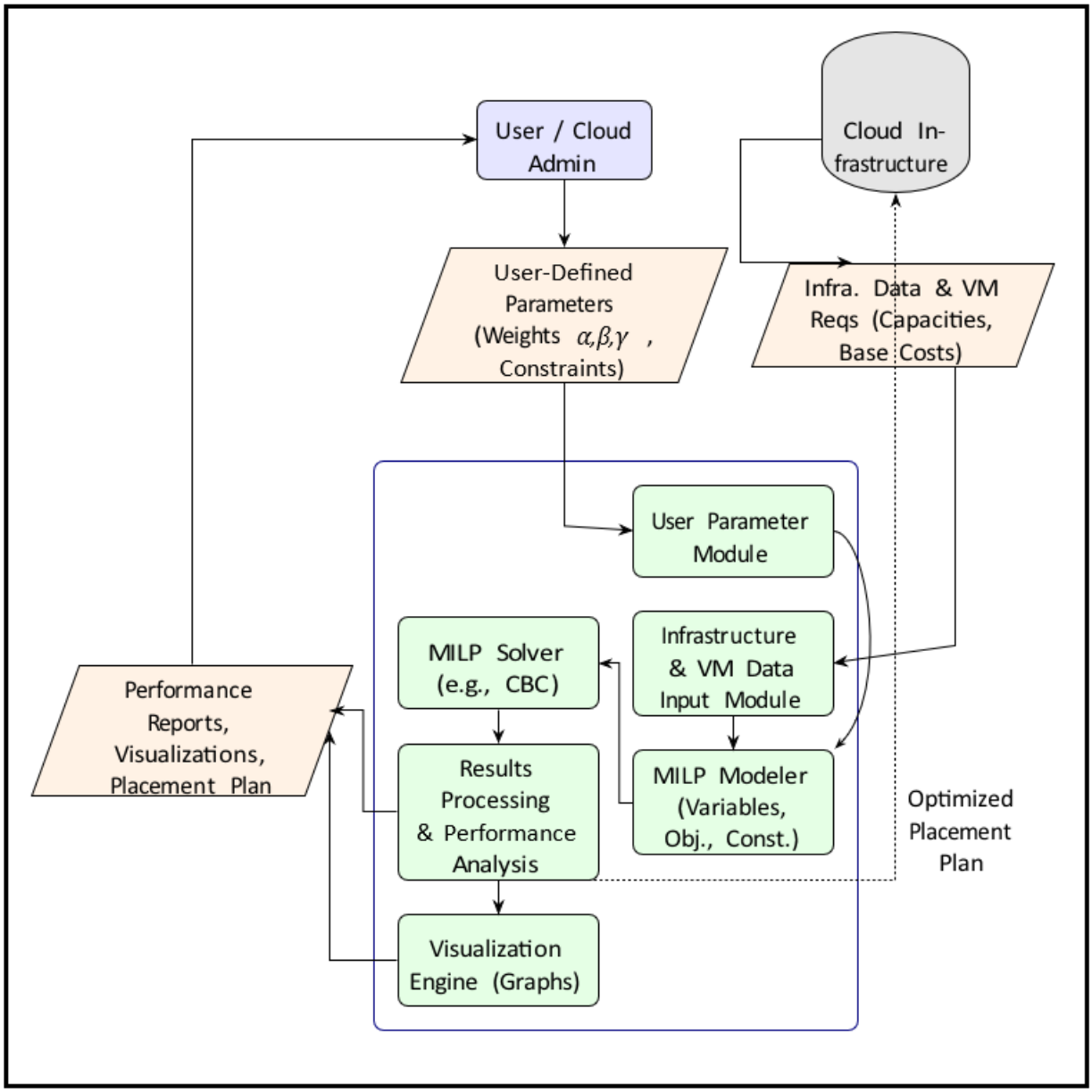

Figure 1.
High-level architecture of the proposed IaaS Optimization Framework.

4.2. Algorithm Overview

The process of optimizing VM allocation is outlined in Algorithm 1. This algorithm encapsulates the setup, MILP definition, solution, and result extraction.

4.3. Implementation Details

The MILP model is implemented using Python with the PuLP library, interfacing with the CBC solver. The framework allows for automated execution of multiple optimization runs by varying input parameters, facilitating comprehensive experimental analysis.

5. Experimental Evaluation

To validate the efficacy and adaptability of the proposed IaaS optimization framework, comprehensive simulation experiments were conducted. This section details the experimental setup, key performance metrics, and discusses the insights derived from the generated results.

Algorithm 1 IaaS Optimization Framework Algorithm (Conceptual)

1: Input: Set of physical servers PS with $R_{j,r}^{avail}$; Set of virtual machines VM with $R_{i,r}^{req}$; Max VMs per server V_{maxj}; Base cost factors $E_{base,j}$,$L_{base,j}$,$P_{base,j}$; User-defined cost weights α,β,γ.

2: Initialization:

3: Define server capacities and VM requirements.

4: Define base cost factors for each PS_j.

5: Define MILP Optimization Problem:

6: Create binary decision variables x_{ij}.
7: Define the objective function (Equation 1) using α,β,γ.
8: Add constraints (Equations 2, 3, 5, and others as needed).
9: Solve MILP Problem:
10: Utilize a standard MILP solver (e.g., CBC, Gurobi, CPLEX via PuLP).
11: Output:
12: Retrieve optimal VM-to-server assignments (x_{ij}^{*}).
13: Calculate performance metrics (total cost, component costs, utilization, active servers, solve time). 14: Return: Optimized allocation results and performance metrics.

5.1. Simulation Setup

The simulation environment models a cloud provider's infrastructure.

- Infrastructure and workload generation: Physical server capacities ($R_{j,r}^{avail}$) for CPU, memory, storage, and bandwidth, as well as VM requirements ($R_{i,r}^{req}$), were generated randomly within realistic ranges (e.g., server CPUs: 40-120 units, VM CPUs: 1-16 units). Base cost factors ($E_{base,j}$,$L_{base,j}$,$P_{base,j}$) for each server were also randomly generated. A fixed seed was used for generating this base data in specific experimental sets to ensure comparability when other parameters (like weights) were varied.
- Experimental Scenarios: Two main sets of experiments were conducted:
  1) Experiment 1 (Weight Variation): For a fixed base scenario (e.g., 20 servers, 100 VMs), the cost weights (α, β, γ) were systematically varied across predefined combinations (e.g., focusing on energy (α = 1), latency (β = 1), provisioning (γ = 1)), balanced approaches, and dominant-factor approaches.
  2) Experiment 2 (VM Load Scaling): Using balanced cost weights (e.g., α=β=γ=1/3), the number of VMs was varied (e.g., 50, 75, 100, 125, 150 VMs) while keeping the number of servers fixed, to assess scalability and performance under different load intensities. Server/VM data was re-generated for each distinct problem size in this experiment set, using a systematically varied seed.
- Solver: The PuLP library with the CBC solver was used for all MILP optimizations.

5.2. Performance Metrics

The framework's performance was evaluated using the following key metrics, collected from each optimization run:

- Total Weighted Cost: The objective function value (Equation 1).
- Actual Cost Components: The sum of base energy, latency, and provisioning costs for the placed VMs, each multiplied by their respective global weight ($\alpha^{P}E_{base,j}x_{ij}$, etc.).
- Number of Active Physical Servers: Count of servers hosting at least one VM.
- Average Server Utilization: For CPU, Memory, etc., across active servers.
- Solver Time: Wall-clock time taken by the MILP solver.
- VMs per Server Distribution: For analyzing load balancing characteristics.

5.3. Results and Discussion

The simulation results are presented through ten comparative graphs, designed to offer deep insights into the system's behavior and the trade-offs involved in IaaS resource allocation.

l : Igy W igh (l h)

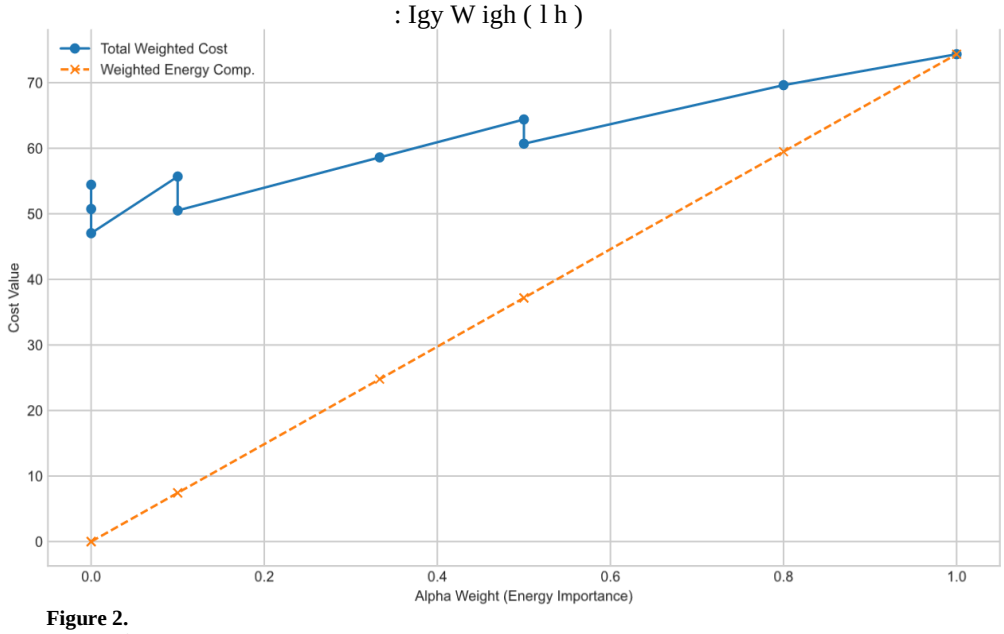

Figure 2.
Impact of Energy Weight (α) on Total Weighted Cost and Actual Energy Cost Component.

l : i gy W igh (l h)

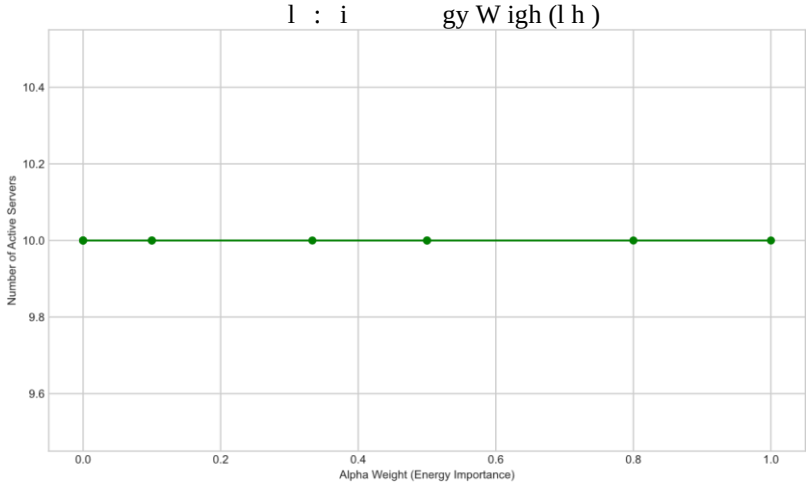

Figure 3.
Number of Active Servers vs. Energy Weight (α).

Figure 2 illustrates the relationship between the energy cost weight (α) and both the total weighted operational cost and the actual energy cost component. As α increases, the optimization prioritizes placements with lower base energy costs, typically leading to a decrease in the actual energy cost component. The total weighted cost's behavior depends on how this shift impacts the latency and provisioning components. Figure 3 demonstrates the impact of α on the number of active physical servers. A higher α is expected to encourage server consolidation to turn off more servers, reducing overall idle power, thus potentially decreasing the number of active servers.

The scalability of the framework is assessed in Figures 4 and 5. Figure 4 plots the total weighted cost as the number of VMs increases under a balanced weighting scheme. This shows how efficiently the system can accommodate growing demand. Figure 5 presents the corresponding solver time, which is critical for understanding the practical limits of applying the MILP approach to larger problem instances. An increase in solver time with problem size is characteristic of MILP.

Figure 6 provides a comparative bar chart of the total weighted costs achieved under distinct high-level strategic priorities (e.g., energy-focused with $\alpha = 1$, latency-focused with $\beta = 1$, balanced with $\alpha = \beta = \gamma = 1/3$). This highlights the extent to which overall costs can vary based on user-defined objectives.

Resource utilization efficiency is depicted in Figure 7, showing average CPU and memory utilization on active servers as the number of VMs scales. This indicates how well the framework packs VMs while respecting capacity constraints. Figure 8 presents a scatter plot illustrating the trade-off between the actual energy cost component and the actual latency cost component. Each point typically represents a run with different (α,β,γ) weights, revealing the achievable balance between these competing objectives.

l : l ili y wi h

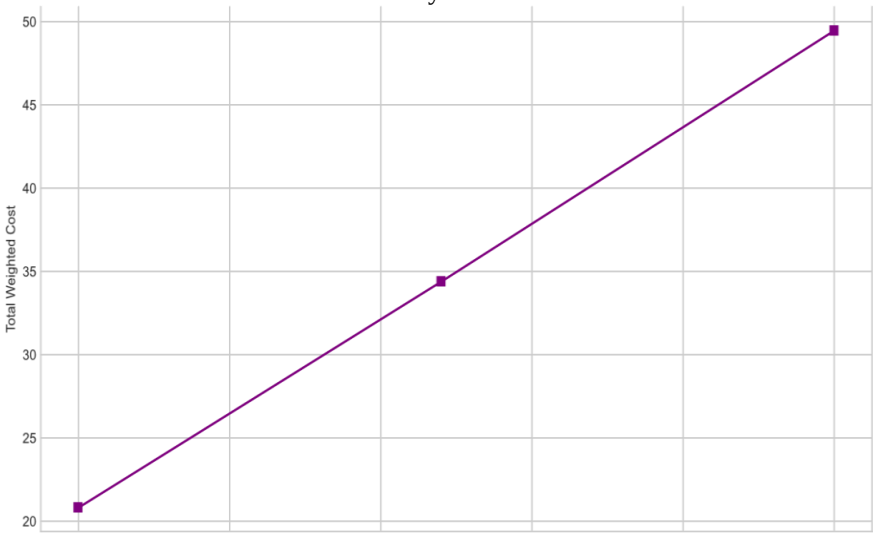

Figure 4.
Total Weighted Cost vs. Number of VMs (Scalability with Balanced Weights).

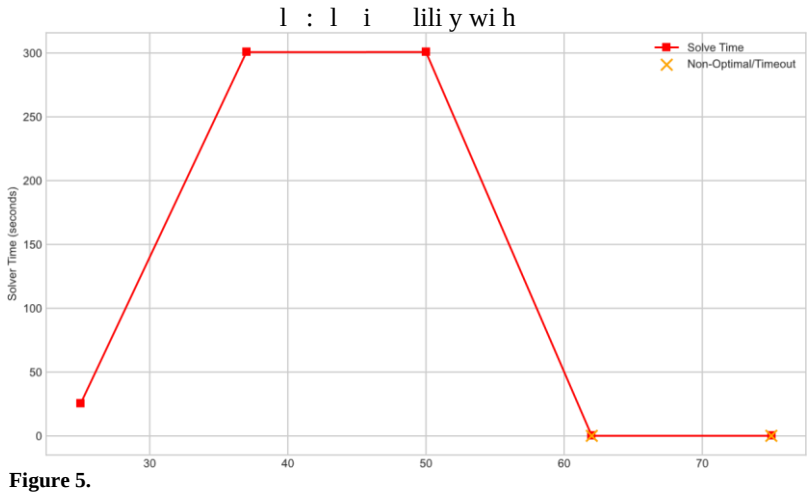

Figure 5.
Solver Time vs. Number of VMs (Scalability with Balanced Weights).

The distribution of VMs across active servers for different weighting strategies is compared in Figure 9 using box plots. This reveals how different optimization goals affect load balancing; for example, a strong energy focus might lead to some servers being densely packed while others are off, compared to a balanced approach. Figure 10 complements Figure 3 by showing how the number of active servers scales specifically with increasing VM load under a consistent (balanced) weighting strategy.

Finally, Figure 11 provides another view on computational scalability by plotting solver time against a combined problem size metric (e.g., product of VMs and servers). This helps generalize the understanding of how computation time grows with the overall complexity of the allocation problem.

Collectively, these results demonstrate the framework's capability to achieve optimized VM placements according to user-specified priorities and underscore the inherent trade-offs in cloud resource management. The MILP approach, while computationally intensive for very large instances, provides optimal solutions for the given model, offering valuable benchmarks and insights for strategic planning.

6. Conclusion and Future Work

This research presents a novel Infrastructure-as-a-Service (IaaS) optimization framework, anchored by Mixed Integer Linear Programming, which adeptly navigates the complex interplay of cost efficiency, system performance, and energy consumption. By systematically integrating user-defined cost-weighting parameters (α, β, γ) directly into the optimization core, the framework offers a significant advancement in adaptable and precise VM allocation. The proposed approach facilitates dynamic parameter selection, enabling IaaS providers to tailor resource provisioning strategies to diverse and evolving operational requirements. Comprehensive simulated evaluations, varying cost priorities and workload scales, have demonstrated considerable potential for reductions in operational expenditures and enhancements in resource utilization while rigorously satisfying performance and reliability mandates, as evidenced by the detailed analysis of ten comparative graphical results.

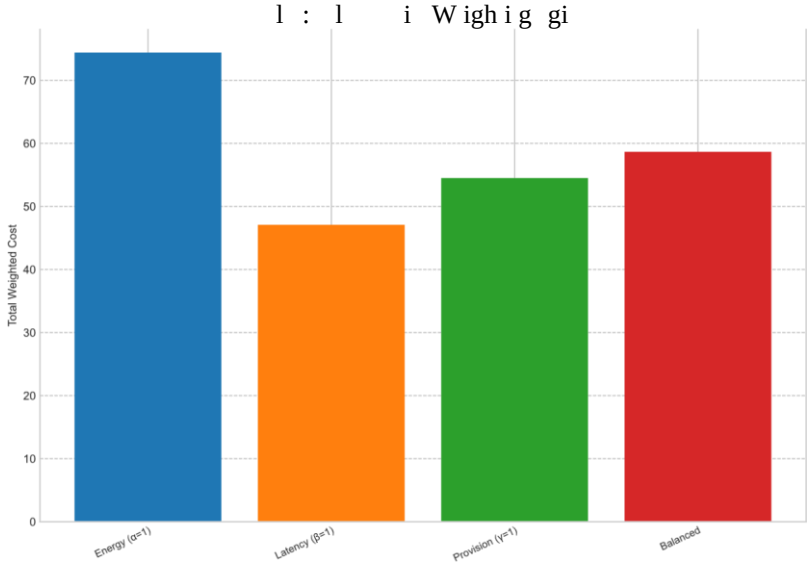

Figure 6.
Comparison of Total Weighted Cost for Key Weighting Strategies (e.g., Energy-Focused, Latency-Focused, Provisioning-Focused, Balanced).

l: giliz i

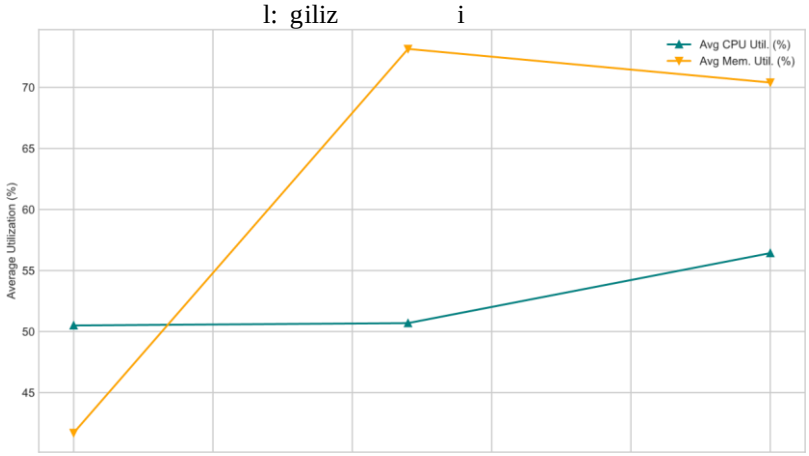

Figure 7.
Average Server Resource Utilization (CPU, Memory) vs. Number of VMs.

Future research endeavors will extend this foundation in several promising directions. The exploration of deep reinforcement learning models for predictive, real-time VM allocation and dynamic workload adaptation presents a compelling avenue, potentially synergizing with the MILP framework for hybrid decision-making (e.g., RL for rapid tactical adjustments, MILP for periodic strategic re-optimization). Incorporating more sophisticated models for uncertainty in resource demands and availability, as well as dynamic pricing schemes, could further enhance the robustness and economic viability of the allocation strategies. Investigating advanced multi-objective optimization techniques beyond simple weighting, such as lexicographical optimization, goal programming, or explicit Pareto front generation, could provide more nuanced control over conflicting objectives. Furthermore, extending the framework to encompass broader aspects of the cloud stack, such as Software-Defined Networking (SDN) configurations for latency-sensitive traffic or storage tiering in conjunction with VM placement, will be crucial for holistic IaaS optimization. The development of efficient decomposition techniques or specialized heuristics guided by MILP insights remains important for addressing extremely large-scale instances where exact MILP solutions become intractable within operational timeframes.

The dynamic allocation of VMs in cloud environments necessitates robust attack detection mechanisms to mitigate security threats such as VM sprawl, side-channel attacks, and unauthorized resource access. Malicious actors may exploit vulnerabilities in static allocation policies to launch denial-of-service (DoS) attacks or manipulate workloads, leading to energy inefficiencies and degraded performance [47]. Integrating real-time anomaly detection algorithms, such as machine learning-based intrusion detection systems (IDS), is essential for enhancing cybersecurity measures [48] can enhance threat identification by analyzing workload patterns and flagging deviations indicative of cyberattacks. Additionally, software-edge authentication plays a critical role in securing distributed cloud infrastructures [49] by verifying the legitimacy of edge devices and users before granting access to computational resources, and security can be significantly enhanced. Techniques such as zero-trust authentication and blockchain-based identity verification can ensure secure VM migrations and prevent unauthorized provisioning. Embedding these security measures within the optimization framework allows cloud providers to achieve cost-efficient resource allocation while maintaining stringent security and performance standards. This approach addresses the limitations of conventional static methods, ensuring a more dynamic and secure resource management process.

l 7::gy L y (lylh)

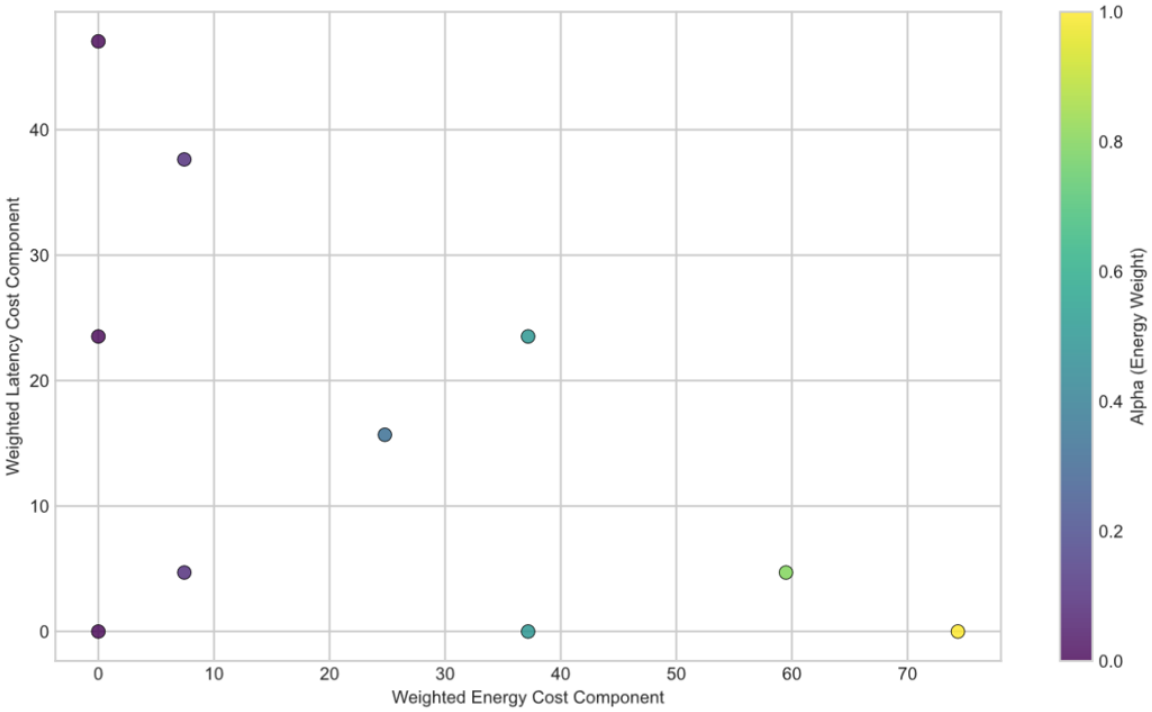

Figure 8.
Trade-off: Actual Energy Cost Component vs. Actual Latency Cost Component (across various weight settings, potentially colored by α).

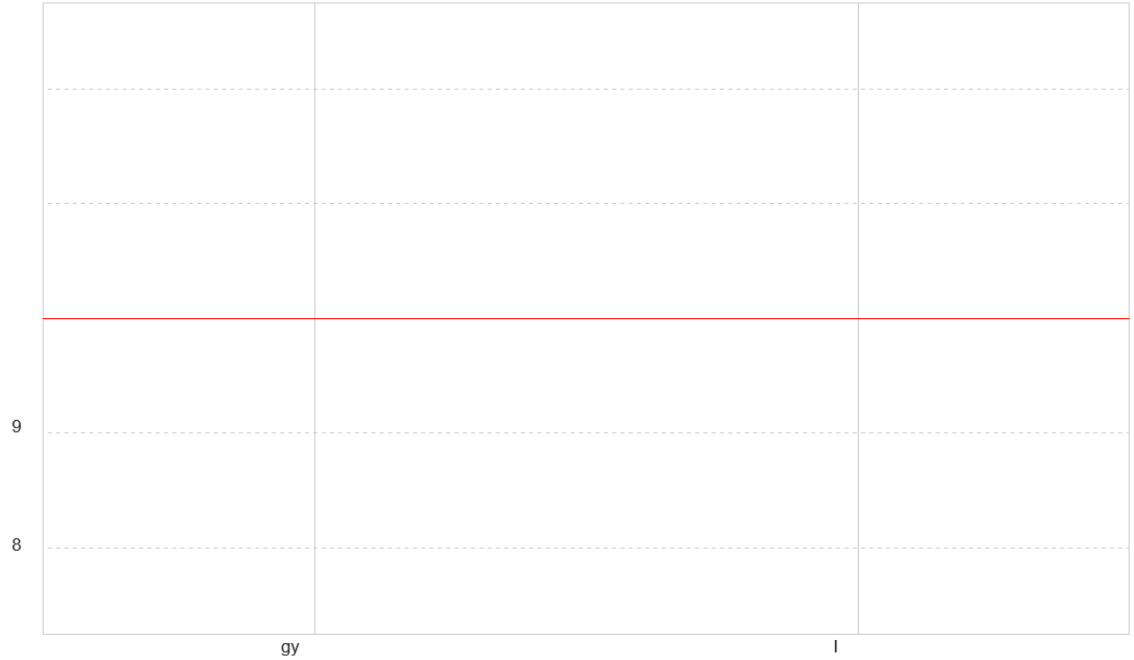

Figure 9.
Distribution of VMs per Active Server (e.g., Box Plots) for Different Key Weighting Strategies.

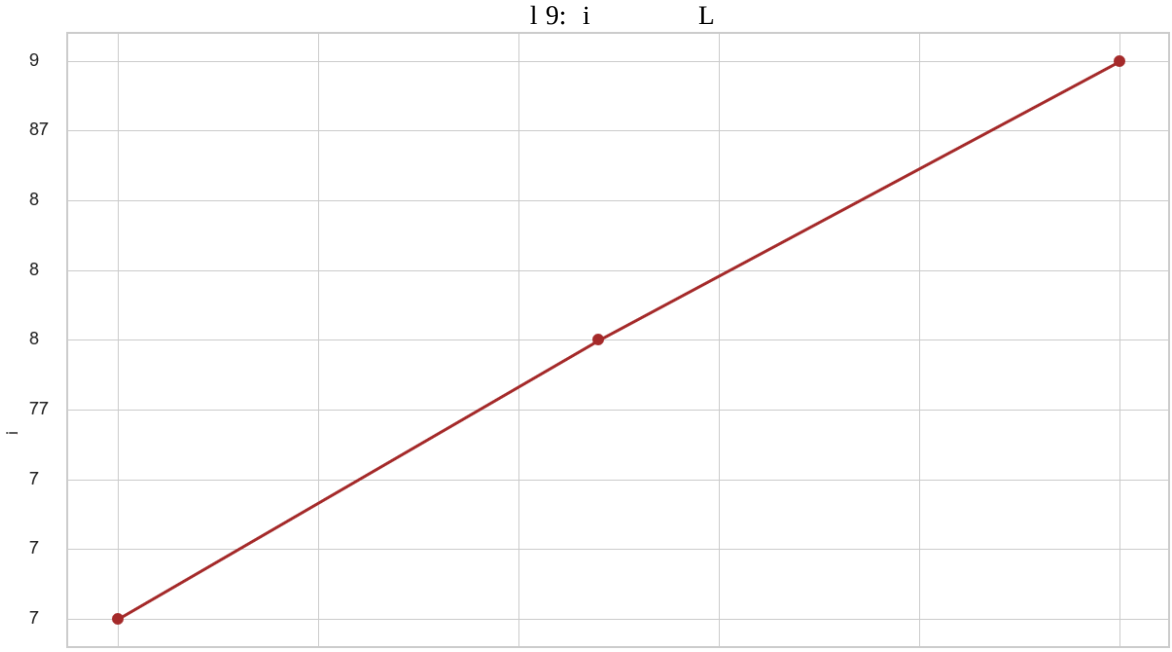

Figure 10.
Number of Active Servers vs. Number of VMs (from VM Load Scaling Experiment).

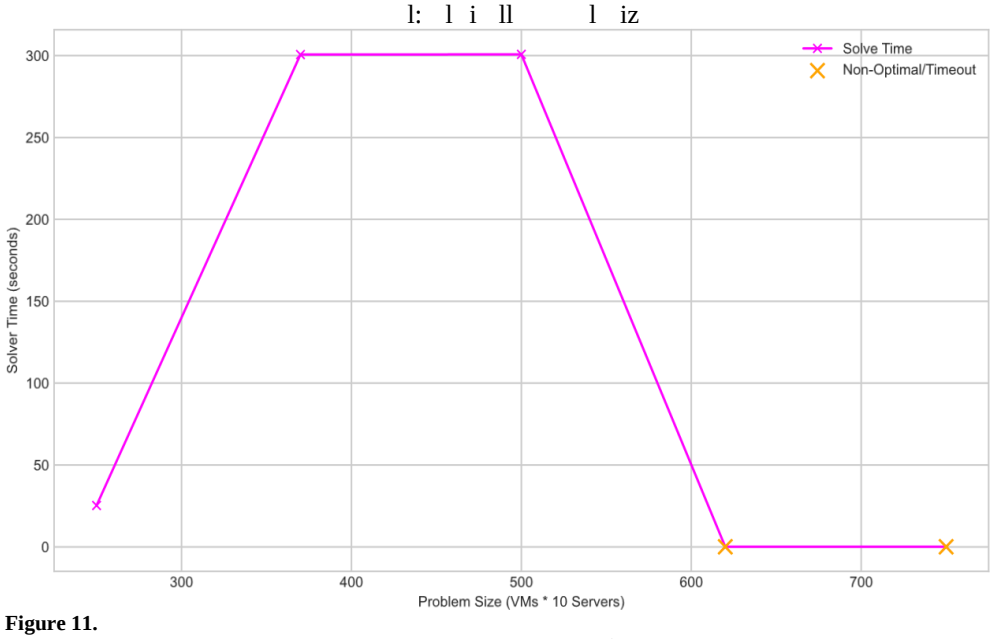

Figure 11.
Solver Time vs. Combined Problem Size Metric (e.g., VMs × Servers for scaling experiment).

References

[1] M. Vijayakumar *et al.*, "Private cloud infrastructure using cloud stack for educational institutions research," in *AIP Conference Proceedings, vol. 3279, no. 1, p. 020152. AIP Publishing LLC*, 2025.

[2] Q. Al-Na'amneh, M. Aljawarneh, R. Hazaymih, L. Alzboon, D. Abu Laila, and S. Albawaneh, *Trust evaluation enhancing security in the cloud market based on trust framework using metric parameter selection* (Utilizing AI in Network and Mobile Security for Threat Detection and Prevention). Hershey, PA, USA: IGI Global Scientific Publishing, 2025.

[3] Q. Al-Na'amneh, R. Hazaymih, M. A. Almaiah, and L. Alzboon, *Secure cloud-marketplaces: A trust framework for evaluating security for client service providers* (Utilizing AI in Network and Mobile Security for Threat Detection and Prevention). Hershey, PA, USA: IGI Global Scientific Publishing, 2025.

[4] F. Mahmud *et al.*, "Big data and cloud computing in IT project management: A framework for enhancing performance and decision-making," 2025.

[5] R. C. Thota, "Comparative analysis of hypervisor performance: VMware vs. AWS nitro in cloud computing," *International Journal of Innovative Research and Creative Technology*, vol. 11, no. 1, pp. 1-14, 2025.

[6] A. Bose and S. Nag, "Green computing-a survey of the current technologies," *Asia-Pacific Journal of Management and Technology*, vol. 3, pp. 01-15, 2022.

[7] M. Zakarya *et al.*, "epcAware: A game-based, energy, performance and cost-efficient resource management technique for multi-access edge computing," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1634-1648, 2022. https://doi.org/10.1109/TSC.2020.3005347

[8] M. Zakarya, L. Gillam, and O. Rana, "profitAware: A cost effective service placement technique for multi-access edge clouds," *Authorea Preprints*, 2025.

[9] Y. Ma *et al.*, "A novel approach to cost-efficient scheduling of multi-workflows in the edge computing environment with the proximity constraint," in *International Conference on Algorithms and Architectures for Parallel Processing (pp. 655-668). Cham: Springer International Publishing*, 2019.

[10] S. Chouliaras, "Adaptive resource provisioning in cloud computing environments," Ph.D. Dissertation, Birkbeck, University of London, 2023.

[11] O. Bystrov, R. Pacevič, and A. Kačeniauskas, "Cost- and performance-aware resource selection for parallel software on heterogeneous cloud," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 10, p. e7877, 2024. https://doi.org/10.1002/cpe.7877

[12] X. You, D. Sun, X. Lv, S. Gao, and R. Buyya, "MQDS: An energy saving scheduling strategy with diverse QoS constraints towards reconfigurable cloud storage systems," *Future Generation Computer Systems*, vol. 129, pp. 252-268, 2022. https://doi.org/10.1016/j.future.2021.11.025

[13] M. Zakarya, L. Gillam, K. Salah, O. Rana, S. Tirunagari, and R. Buyya, "CoLocateMe: Aggregation-based, energy, performance and cost aware VM placement and consolidation in heterogeneous IAAS clouds," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1023-1038, 2023. https://doi.org/10.1109/TSC.2022.3181375

[14] H. Huang, W. Lin, J. Lin, and K. Li, "Power management optimization for data centers: A power supply perspective," *IEEE Transactions on Sustainable Computing*, vol. 10, no. 4, pp. 784-803, 2025. https://doi.org/10.1109/TSUSC.2025.3542779

[15] H. Xu, P. Cheng, Y. Liu, W. Wei, and W. Zhang, "A multi-objective Virtual Machine Scheduling Algorithm in Fault Tolerance Aware Cloud Environments," in *International Conference on Cloud Computing (pp. 529-543). Cham: Springer International Publishing*, 2019.

[16] S. K. Moghaddam, R. Buyya, and K. Ramamohanarao, "Performance-aware management of cloud resources: A taxonomy and future directions," *ACM Computing Surveys*, vol. 52, no. 4, p. Article 84, 2019. https://doi.org/10.1145/3337956

[17] A. A. Khan, M. Zakarya, and R. Khan, "H^2—A hybrid heterogeneity aware resource orchestrator for cloud platforms," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3873-3876, 2019. https://doi.org/10.1109/JSYST.2019.2899913

[18] M. Son *et al.*, "Splice: An automated framework for cost-and performance-aware blending of cloud services," in *2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid) (pp. 119-128). IEEE*, 2022.

[19] A. A. H. Al-Mahruqi, V. Athinarayanana, G. Morison, and B. G. Stewart, "A proposed energy and performance aware cloud framework for improving service level agreements (SLAs) in cloud datacenters," *International Journal of Applied Engineering Research*, vol. 13, no. 16, pp. 12917-12922, 2018.

[20] A. Ullah, Z. Alomari, S. Alkhushayni, D. a. Al-Zaleq, M. Bany Taha, and H. Remmach, "Improvement in task allocation for VM and reduction of Makespan in IaaS model for cloud computing," *Cluster Computing*, vol. 27, no. 8, pp. 11407-11426, 2024. https://doi.org/10.1007/s10586-024-04539-8

[21] S. Singh, P. Singh, and S. Tanwar, "Energy aware resource allocation via MS-SLnO in cloud data center," *Multimedia Tools and Applications*, vol. 82, no. 29, pp. 45541-45563, 2023. https://doi.org/10.1007/s11042-023-15521-8

[22] M. Zakarya, A. A. Khan, L. Gillam, O. Rana, and R. Buyya, "Workloadaware: A resource allocation and consolidation technique for heterogeneous clouds," *Authorea Preprints*, 2025. http://dx.doi.org/10.36227/techrxiv.174495612.20928032/v1

[23] C. Phalak, D. Chahal, and R. Singhal, "SIRM: Cost efficient and SLO aware ML prediction on Fog-Cloud Network," in *2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS) (pp. 825-829). IEEE*, 2023.

[24] S. Gomez S´ aez, "Design support for performance-and cost-efficient (re) distribution of cloud applications," Ph.D. Dissertation,´ Dissertation, Stuttgart, Universitat Stuttgart, 2019.

[25] S. Chouliaras and S. Sotiriadis, "Auto-scaling containerized cloud applications: A workload-driven approach," *Simulation Modelling Practice and Theory*, vol. 121, p. 102654, 2022. https://doi.org/10.1016/j.simpat.2022.102654

[26] W. Lin, W. Wu, and L. He, "An on-line virtual machine consolidation strategy for dual improvement in performance and energy conservation of server clusters in cloud data centers," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 766-777, 2019.

[27] P. Rajasekar and Y. Palanichamy, "Scheduling multiple scientific workflows using containers on IaaS cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7621-7636, 2021. https://doi.org/10.1007/s12652-020-02483-0

[28] M. Zakarya, L. Gillam, M. R. C. Qazani, A. A. Khan, K. Salah, and O. Rana, "BackFillMe: An energy and performance efficient virtual machine scheduler for IaaS datacenters," *IEEE Transactions on Services Computing*, vol. 1, pp. 1-14, 2025.

[29] S. G. Sáez, V. Andrikopoulos, M. Bitsaki, F. Leymann, and A. v. Hoorn, "Utility-based decision making for migrating cloud-based applications," *ACM Transactions on Internet Technology*, vol. 18, no. 2, p. Article 22, 2018. https://doi.org/10.1145/3140545

[30] M. Tarahomi, M. Izadi, and M. Ghobaei-Arani, "An efficient power-aware VM allocation mechanism in cloud data centers: a micro genetic-based approach," *Cluster Computing*, vol. 24, no. 2, pp. 919-934, 2021.

[31] J. Nazir *et al.*, "Load balancing framework for cross-region tasks in cloud computing," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1479-1490, 2022.

[32] M. Fan, L. Ye, X. Zuo, and X. Zhao, "A bidirectional workflow scheduling approach with feedback mechanism in clouds," *Expert Systems with Applications*, vol. 249, p. 123494, 2024. https://doi.org/10.1016/j.eswa.2024.123494

[33] A. O. Nyanteh, "Cognitive-aware network virtualization hypervisor for efficient resource provisioning in software defined cloud networks," Ph.D. Dissertation, Brunel University London, 2021.

[34] G. Sadeghian, M. Elsakhawy, M. Shahrad, J. Hattori, and M. Shahrad, "{UnFaaSener}: Latency and cost aware offloading of functions from serverless platforms," in *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, 2023.

[35] K. Ajmera and T. Kumar Tewari, "Dynamic virtual machine scheduling using residual optimum power-efficiency in the cloud data center," *The Computer Journal*, vol. 67, no. 3, pp. 1099-1110, 2023. https://doi.org/10.1093/comjnl/bxad045

[36] A. Alsarhan and A. Al-Khasawneh, "Resource trading in cloud environments for utility maximisation using game theoretic modelling approach," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 31, no. 4, pp. 319-333, 2016. https://doi.org/10.1080/17445760.2015.1057589

[37] S. Wang, Z. Ding, and C. Jiang, "Elastic scheduling for microservice applications in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 98-115, 2020.

[38] J. M. R. Poovizhi and R. Devi, "Performance analysis of cloud hypervisor using network package workloads in virtualization," presented at the 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART). IEEE, 2023, pp. 427–432, 2023.

[39] J. Ma and K. R. Mohan Raob, "Efficient resource managing and job scheduling in a heterogeneous Kubernetes cluster for big data," *International Journal of Performability Engineering*, vol. 20, no. 3, pp. 157-166, 2024. https://doi.org/10.23940/ijpe.24.03.p4.157166

[40] H. Xu, S. Xu, W. Wei, and N. Guo, "Fault tolerance and quality of service aware virtual machine scheduling algorithm in cloud data centers," *The Journal of Supercomputing*, vol. 79, no. 3, pp. 2603-2625, 2023.

[41] S. Lu *et al.*, "Online elastic resource provisioning with QoS guarantee in container-based cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, pp. 1-16, 2024.

[42] A. C. Zhou, J. Lao, Z. Ke, Y. Wang, and R. Mao, "FarSpot: Optimizing monetary cost for HPC applications in the cloud spot market," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2955-2967, 2022. https://doi.org/10.1109/TPDS.2021.3134644

[43] B. Igried, A. F. Al-Serhan, A. Alsarhan, M. Aljaidi, and A. Aldweesh, "Machine learning failure-aware scheme for profit maximization in the cloud market," *Future Internet*, vol. 15, no. 1, p. 1, 2023. https://doi.org/10.3390/fi15010001

[44] T.-T. Chang and S. Venkataraman, "Eva: Cost-efficient cloud-based cluster scheduling," *arXiv preprint arXiv:2503.07437*, 2025.

[45] B. P. Sharma, "Optimizing cloud migration strategies for large-scale enterprises: A comparative analysis of lift-and-shift, replatforming, and refactoring approaches," *Advances in Theoretical Computation, Algorithmic Foundations, and Emerging Paradigms*, vol. 15, no. 2, pp. 1-14, 2025.

[46] S. Rafatirad, H. Homayoun, Z. Chen, and S. M. Pudukotai Dinakarrao, *Applied machine learning for cloud resource management. In Machine Learning for Computer Scientists and Data Analysts: From an Applied Perspective*. Cham: Springer, 2022.

[47] M. Aljaidi *et al.*, "A critical evaluation of a recent cybersecurity attack on itunes software updater," in *2022 International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI), pp. 1-6. IEEE*, 2022.

[48] R. Alqura'n *et al.*, "Advancing XSS detection in IoT over 5G: A cutting-edge artificial neural network approach," *IoT*, vol. 5, no. 3, pp. 478-508, 2024. https://doi.org/10.3390/iot5030022

[49] A. Almaini, A. Al-Dubai, I. Romdhani, M. Schramm, and A. Alsarhan, "Lightweight edge authentication for software defined networks," *Computing*, vol. 103, no. 2, pp. 291-311, 2021. https://doi.org/10.1007/s00607-020-00835-4