Computer Science, Engineering and Technology
Vol: 2(2), June 2024
REST Publisher; ISSN No: 2583-9179 (Online)
Website: https://restpublisher.com/journals/cset/
DOI: https://doi.org/10.46632/cset/2/2/6

Improving Cloud Resource Scaling: A Comparative Analysis of Automatic Scaling Techniques Using the WASPAS Methodology

***Vamsi Krishna Kavuri**

Senior Lead Software Engineer
*Corresponding Author Email: kavurivk@gmail.com

Abstract: *Cloud computing has transformed resource management by enabling scalable, on-demand services. Effective cloud management is critical to balancing performance, cost, and resource utilization. This study examines various cloud scaling techniques, including rule-based, predictive, container-based, serverless, and hybrid scaling, and evaluates their effectiveness in optimizing cloud resources while ensuring flexibility and reliability. This research provides a comparative analysis of cloud resource scaling strategies, which helps organizations select the most appropriate approach. By evaluating different scaling techniques, this study improves understanding of efficient cloud resource allocation, reduces costs, and ensures performance optimization. It contributes to cloud computing research by addressing key challenges in resource management. The alternative: Rule-Based Auto-Scaling, Predictive Auto-Scaling with AI/ML, Container-Based Scaling, Serverless Computing, Hybrid Scaling. The evaluation criteria consist of Scalability Efficiency, Resource Utilization, Over-Provisioning Risk, Complexity of Implementation. According to the results, Serverless Computing was ranked highest, while Hybrid Scaling was ranked lowest. Based on the Weighted Aggregated Sum Product Assessment System (WASPAS), serverless computing offers the highest value for cloud management for advanced resource scalability.*

Keywords: *Cloud Computing, Resource Scaling, Auto Scaling, Serverless Computing, Predictive Scaling, Container-Based Scaling, Hybrid Cloud Management, Cloud Resource Optimization.*

1. INTRODUCTION

The environment in which contemporary cloud computing systems operate is dynamic and ever-changing, with changing performance requirements and computing conditions. These changes occur unexpectedly, unpredictably, and beyond the control of cloud providers. Virtualization, a key technology that enables multiple user applications to share the same physical machine while ensuring performance guarantees, serves as the foundation for developing effective management policies for modern cloud systems. [1] To evaluate the performance of the proposed framework, several experiments were conducted on a file upload service and compared with the results of a different algorithm based on its primary statistic of request-to-arrival ratio. Together with the published performance metrics, a thorough experimental study demonstrates how specific factors affect the new framework, providing a significant

and innovative addition to the field of cloud resource management research. [2] Scaling computing resources to achieve optimal resource efficiency and high performance is a key feature of cloud computing platforms. This scalability is especially important for web applications such as e-commerce for two main reasons. First, web applications often have an n-tier design, which includes a web tier, an application server tier, a database tier, and additional tiers such as load balancers and caching systems. By changing the number of server virtual machines, each tier can be increased or decreased instantly. Second, workloads for online applications can increase quickly and without warning. [3] Resource management in cloud computing refers to the distribution of computing, storage, networking, and, indirectly, power resources among different applications to collectively meet the performance objectives of cloud resource users, infrastructure providers (such as data centers), and applications. Prioritizing resource optimization is a top priority for providers when establishing predetermined service level agreements (SLAs) with cloud users. Virtualization technologies, which enable the dynamic sharing of resources among a large number of customers and applications, are often used to accomplish this efficiency. [4] The Automated Cloud Computing on Demand (ACCRS) framework has been introduced to dynamically scale resources based on system availability, usage levels, and service level agreements (SLAs), using resource provisioning in cloud environments. By combining proactive fault detection to prevent potential issues and reactive recovery mechanisms to address faults when they occur, the framework improves the reliability and availability of cloud computing. [5] Careful monitoring of resource and service consumption is essential to enable variable pricing, charging, and billing, a fundamental prerequisite for cloud elasticity. The key challenge is to ensure that consumers are only charged for services they use within a given period of time.. While cloud computing itself does not determine cost allocation, it serves as the foundation for infrastructure design that supports charging models for accurate metering and billing. [6] As cloud computing continues to expand and the number of providers increases, the need for cross-cloud usage will increase, making federated management a critical requirement. Insights from past systems should be used to improve federated cloud management. While current cloud environments accommodate basic cloud explosion scenarios, more standardized usage models will emerge as cloud computing advances. [7] Cloud services can be divided into three categories: Software as a Service (SaaS), which allows access to remote software services; Platform as a Service (PaaS), which provides APIs for building applications on the platform; and Infrastructure as a Service (IaaS), which includes the underlying infrastructure along with middleware. IaaS and PaaS cloud platforms are an alternative for hosting enterprise and scientific applications, in which resources are rented from large data centers rather than being directly managed by businesses. [8] When designing management systems for large-scale environments such as networks and cloud infrastructures, ensuring scalability is critical. However, it is also desirable to maintain visibility across the entire system for monitoring and performance optimization. These objectives often conflict, making it challenging to strike a balance. While centralized management systems provide a comprehensive view of the system, they struggle to scale efficiently as the number of nodes increases. In contrast, distributed approaches provide better scalability but make it more difficult to assess and improve the overall system health. [9] While most cloud data centers rely on real-time state monitoring, periodic sampling, and single-tenant monitoring as their core monitoring functions, we propose that Monitoring as a Service (MaaS) should extend its capabilities. In addition to real-time state monitoring, it should also incorporate window-based monitoring. Along with periodic sampling, it should integrate violation-probability-based state collection and monitoring. Furthermore, instead of being limited to single-tenant monitoring, MaaS should support multi-tenant state monitoring. [10] The performance of streaming applications can be severely affected by over- or under-provisioning of resources under peak loads due to a statically provisioned Stream Processing Engine (SPE) installed on a small number of processing nodes. SPEs can adapt to grow or shrink cloud resources to manage dynamic data streams. Various scaling techniques are used to optimize resource allocation and reduce deployment costs in cloud environments. [11] The infrastructure, called the "cloud", allows users and organizations to access application services from anywhere in the world whenever they need them. Thus, cloud computing is a contemporary paradigm for the dynamic and adaptive delivery of computing services, typically powered by state-of-the-art data centers composed of groups of connected virtual machines. [12] Furthermore, multiple cloud management tools, including constructors, orchestrators, and monitoring systems, can be integrated with the recommended smart cloud engine and solution. A cloud service provider (CSP) that offers cloud services across multiple tiers, including SaaS applications from various vendors, has validated this solution, developed as part of the ICARO Cloud project. The tools in the organization's comprehensive cloud environment include multi-tier solutions for CRM (customer relationship management), ERP (enterprise resource planning), workflow automation, marketing, business analytics, cultural heritage, social media, and other areas. [13] In the era of cloud computing, service providers no longer need to set up their own IT infrastructure. Instead, they can use Infrastructure as a Service (IaaS) to quickly deliver the resources they need. Providers can create, configure, scale, and remove virtual machines (VMs), storage, and load balancers using APIs from platforms such as Google Compute Engine (GCE) and Amazon Web Services (AWS). [14] Thanks to cloud computing, scientists can now use computer infrastructure and applications in a new way. Algorithms and computational resources can be centralized and delivered as on-demand services tailored to specific application needs through the use of virtualization.. [15] Businesses typically purchase their own computing resources and handle hardware updates and maintenance, which increases costs. The emergence of cloud computing, powered by virtualization technology, allows for pay-as-you-go resource allocation. By outsourcing their computing needs to the cloud, companies can eliminate the burden of maintaining their own infrastructure. [16] Cloud computing enables tenants to rent resources on a pay-as-you-go model, providing a more cost-effective alternative to on-premises computing by eliminating the need to manage complex infrastructure. To fully utilize this advantage, cloud applications must obtain an

appropriate allocation of computing resources. However, resource requirements are rarely constant and fluctuate due to variations in overall workload, workload composition, and internal application phases and transitions. [17] Clouds face many of the same challenges as distributed systems, including load balancing and scalability, but they also include unique features such as virtual server provisioning and image management. In particular, clouds share similarities with grids, which are networks of computing resources. However, while grids focus primarily on providing raw computing power, clouds prioritize providing services to a broad user base, including those without technical expertise. [18] The use of cloud computing is still relatively new. Infrastructure vendors focus on virtual machine designs, databases, monitoring tools, cloud management frameworks, and identity solutions, while cloud service providers (CSPs) are largely involved in deploying cloud solutions and exploring potential business models. Rapid resource provisioning and flexible, dynamic, on-demand access are the goals of these organizations. [19] The dynamic allocation of automatically provisioned resources to cloud workloads is called cloud resource planning. Automated resource management strategies based on different scheduling criteria are the main topic of this work. It explores the advances in automated resource management of cloud computing. [20]

2. MATERIALS AND METHOD

2.1 WASPAS Method

For parameter optimization, WASPAS is a weighted sum product estimation technique. By combining the weighted sum model (WSM) and the weighted product model (WPM), it can determine one or more optimal responses. The first step in the multi-stage implementation of the WASPAS algorithm is to generate a decision matrix based on the formula. Equation (3) is related to unfavorable criteria, where the value is low, while equation (2) is designed for favorable criteria, where the maximum value is required depending on the specific criterion. The rule x_{norm} represents the normalized value of x. Next, using the WSM approach, the overall i-variance. The entropy approach is used to estimate the relative weight of the j-criterion, denoted by the symbol w_{ji}. Equation (5) should be used to determine the overall relative importance of the i-variable when using the WPM approach. Formulas (4) and (5) provide an initial assessment of the computational results in the WASPAS approach. Higher scores indicate a more favorable variance. Since no approach can be considered particularly robust in a categorical manner, they should be integrated into an overall weighted criterion (6) using the formulas given for the Weighted Sum Model (WSM) and the Weighted Product Model (WPM). The value of Q is then used to identify possible alternatives; the best alternative is represented by the highest value. The next step is to find the minimum variance (Q). This number ensures a more accurate estimate by eliminating errors in the criteria considered, which are usually random and show some variation. Finding the peak of the function gives the best value for λ, which is also necessary for choosing the correct coefficient of variation. By setting the derivative of equation (7) with respect to λ to zero, this peak can be found. Equation (8) is then used to determine the best λ value.

Step 1: The performance of several alternatives relative to several criteria is displayed in a decision matrix X.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

After normalization, the decision matrix is modified to account for favorable and unfavorable criteria.

$$w_j = [w_1 \quad \cdots \quad w_n] \tag{2}$$

were,

$$\sum_{j=1}^{n} w_j = 1$$

Step 2: Using the WSM approach, the weighted normalized decision matrix is calculated as follows:

$$n_{ij} = \begin{cases} \dfrac{x_{ij}}{\max x_{ij}} & j \in B \\ \dfrac{\min x_{ij}}{x_{ij}} & j \in C \end{cases} \tag{3}$$

Where n_{ij} is the normalized value of i^{th} alternative for the j^{th} section, $\max x_{ij}$ and $\min x_{ij}$ are the maximum and minimum values of x_{ij} in the j^{th} column for benefit (B) and cost criteria (C) respectively.

Step 3: The following is how the WSM approach determines the weighted normalized decision matrix.

$$w_{ij} = w_i \cdot n_{ij} \tag{4}$$

Normalized weighted decision matrix:

$$W(n_{ij}) = (n_{ij})^{w_j} \tag{5}$$

Step 5: Using the WSM technique, the priority score for a particular alternative is determined as follows:

$$S_i^{WSM} = \sum_{j=1}^{n} w_j n_{ij} \tag{6}$$

Step 6: Using the WSM technique, the priority score for a particular alternative is calculated as follows:

$$S_i^{WPM} = \prod_{j=1}^{n} (n_{ij})^{w_j} \tag{7}$$

Step 7: The priority score of the WASPAS method is determined by using equations (6) and (7).

$$S_i^{WASPAS} = \lambda S_i^{WSM} + (1-\lambda) S_i^{WPM}$$

$$S_i^{WASPAS} = \lambda \sum_{j=1}^{n} w_j n_{ij} + (1-\lambda) \prod_{j=1}^{n} (n_{ij})^{w_j}$$

where $0 \leq \lambda \leq 1$.

Finally, the alternatives are ranked based on the S_i^{WASPAS} values. The best alternative has the highest S_i^{WASPAS} value. If the value of λ is 0, WASPAS method is transformed to WPM and if λ is 1, it becomes WSM.

2.2 Parameters

Alternative:

Rule-Based Auto-Scaling: Rule-based autoscaling dynamically adjusts cloud resources based on predefined thresholds, such as CPU usage, memory consumption, or network traffic. When a metric exceeds a specified threshold, additional resources are allocated; when usage decreases, resources are released. This approach ensures optimal performance, cost efficiency, and scalability in cloud environments.

Predictive Auto-Scaling with AI/ML: Predictive autoscaling with AI/ML uses machine learning algorithms to predict resource requirements based on historical data and real-time metrics. Unlike rule-based scaling, it proactively adjusts cloud resources before demand increases or decreases, improving performance, reducing costs, and ensuring seamless scaling for applications in dynamic cloud environments.

Container-Based Scaling: The number of container instances is dynamically changed according to workload requirements through container-based scaling. It efficiently manages resource allocation by scaling containers up or down in response to traffic spikes or drops, using orchestration tools like Kubernetes. This approach improves flexibility, improves resource utilization, and ensures high availability in cloud environments.

Serverless Computing: Developers can run applications using serverless computing without having to worry about maintaining the underlying infrastructure. Cloud providers automatically take care of scaling, maintenance, and resource allocation. Operations are executed on an event-by-event basis, and users pay only for actual operational time. This model improves scalability, reduces operational complexity, and optimizes costs for cloud-based applications.

Hybrid Scaling: Hybrid scaling combines vertical and horizontal scaling to improve cloud resource management. It dynamically adjusts resource capacity by scaling up (adding more power to existing instances) or scaling down (adding more instances) based on workload demands. This approach balances performance, cost-effectiveness, and flexibility in dynamic cloud environments.

Evolution parameter:

- Scalability (SE) - Measures how effectively the approach scales resources.
- Resource Utilization (RU) - Optimal use of computing, storage, and network resources.

- Overprovisioning Risk (OPR) - The likelihood of over-allocating resources.
- Implementation Complexity (CI) - The difficulty of deploying and managing the strategy.

3. ANALYSIS AND DISCUSSION

TABLE 1. Cloud Management for Enhanced Resource Scaling

Alternative	Scalability Efficiency	Resource Utilization	Over-Provisioning Risk	Complexity of Implementation
Rule-Based Auto-Scaling	6.5	6	5	6.5
Predictive Auto-Scaling with AI/ML	8.5	9	8.5	7.2
Container-Based Scaling	7.8	8	7	6.8
Serverless Computing	9	9.5	8	7.5
Hybrid Scaling	8.2	7.8	7.8	7

The dataset (table 1) evaluates five cloud management strategies based on four key parameters: scalability, resource utilization, overprovisioning risk, and implementation complexity. Each alternative presents different strengths and trade-offs, which impact its performance in dynamic cloud environments. Rule-based autoscaling exhibits moderate scalability (6.5) and resource utilization (6), making it a viable option for predictable workloads. However, it has a relatively high overprovisioning risk (5), meaning that resources can sometimes be allocated inefficiently. Its implementation complexity (6.5), while straightforward, indicates that it still requires careful configuration. Predictive autoscaling with AI/ML scores the highest on scalability (8.5) and resource utilization (9), indicating its ability to dynamically optimize cloud resources. However, its overprovisioning risk (8.5) is the highest of all alternatives, likely due to its reliance on predictive models that can overprovision resources. Additionally, its implementation complexity (7.2) is high, reflecting the need for AI integration and advanced configurations. Container-based scaling provides a balance between performance and management. It has strong scalability (7.8) and resource utilization (8), while maintaining a relatively moderate overprovisioning risk (7). Its implementation complexity (6.8) is slightly lower than predictive scaling, making it a viable option for modern container applications. Serverless computing achieves the highest resource utilization (9.5) and scalability performance (9), demonstrating its ability to allocate resources only when needed. However, the overprovisioning risk (8) is still significant, and its implementation complexity (7.5) is relatively high due to the need for activity-based implementation models. Finally, hybrid scaling combines multiple approaches, leading to strong scalability (8.2) and resource utilization (7.8). Although its overprovisioning risk (7.8) is slightly higher than container-based scaling, it remains a versatile and effective strategy. Its implementation complexity (7) suggests a balanced trade-off between flexibility and management overhead. Overall, the dataset highlights how different scaling strategies impact cloud resource management, helping organizations choose the approach that best suits their specific needs.

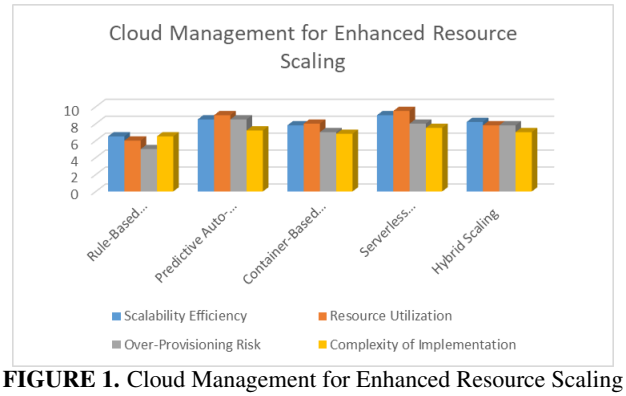

FIGURE 1. Cloud Management for Enhanced Resource Scaling

Figure 1 displays the comparative performance of five cloud management strategies based on four key evaluation parameters: scalability, resource utilization, overprovisioning risk, and implementation complexity. Each approach demonstrates different strengths and weaknesses, impacting its suitability for different cloud environments. Rule-based autoscaling shows moderate performance across all parameters, with scalability and resource utilization scores in the mid-range. Its overprovisioning risk and implementation complexity are relatively lower than AI-based approaches, making it a stable but less dynamic option for predictable workloads. Predictive autoscaling with AI/ML stands out with its high scalability and resource utilization, indicating its ability to effectively optimize cloud resources. However, it also shows the highest overprovisioning risk, which may be due to its reliance on predictive models that can sometimes over-allocate resources. The implementation complexity is also relatively high, reflecting the challenges of integrating AI into cloud management. Container-based scaling offers a balanced approach, demonstrating strong scalability and resource utilization, although slightly lower than AI-driven scaling. Its overprovisioning risk is moderate, suggesting a more controlled resource allocation strategy. Its implementation complexity is slightly lower than predictive scaling, making it a viable option for containerized applications. Serverless computing achieves the highest resource utilization because it dynamically provisions resources based

on real-time requirements. Its scalability is also one of the highest, reinforcing its suitability for dynamic workloads. However, its overprovisioning risk and implementation complexity are relatively high, indicating challenges in effectively managing serverless environments. Finally, hybrid scaling combines multiple approaches, resulting in strong scalability and resource utilization while maintaining moderate overprovisioning risk. Its implementation complexity is significant, suggesting a trade-off between flexibility and ease of management.

TABLE 2. Performance value

Rule-Based Auto-Scaling	0.72222	0.63158	1.00000	1.00000
Predictive Auto-Scaling with AI/ML	0.94444	0.94737	0.58824	0.90278
Container-Based Scaling	0.86667	0.84211	0.71429	0.95588
Serverless Computing	1.00000	1.00000	0.62500	0.86667
Hybrid Scaling	0.91111	0.82105	0.64103	0.92857

Table 2 provides the performance scores of various cloud management strategies, assessing their performance based on key performance metrics. These scores are normalized between 0 and 1, where higher scores indicate better performance in a given category. Rule-based autoscaling shows moderate performance in most areas, with a scalability score of 0.72222 and resource utilization of 0.63158. However, it has very high scores (1.00000) for overprovisioning risk and implementation complexity, indicating that while it is easy to implement, it tends to allocate too many resources, leading to inefficiency. Predictive autoscaling with AI/ML performs exceptionally well in scalability (0.94444) and resource utilization (0.94737), demonstrating its ability to effectively scale cloud resources. However, its overprovisioning risk (0.58824) is relatively lower than rule-based scaling, but is still a concern. The implementation complexity (0.90278) is high, reflecting the challenges of deploying AI-driven solutions. Container-based scalability offers balanced performance with strong scalability (0.86667) and resource utilization (0.84211). Its overprovisioning risk (0.71429) is moderate, indicating that resource allocation can be effectively optimized. The implementation complexity (0.95588) is one of the highest, indicating that while container-based solutions provide flexibility, they require advanced management. Serverless computing stands out with the highest scalability (1.00000) and resource utilization (1.00000), demonstrating its effectiveness in effectively handling dynamic workloads. However, its overprovisioning risk (0.62500) is significant, and its implementation complexity (0.86667) is relatively high due to the need to manage function-based implementation models. Hybrid scaling provides strong scalability efficiency (0.91111) and resource utilization (0.82105), with moderate over-provisioning risk (0.64103). Its implementation complexity (0.92857) is slightly lower than container-based scaling, but reflects the difficulty of integrating multiple scaling strategies.

TABLE 3. Weight

Rule-Based Auto-Scaling	0.25	0.25	0.25	0.25
Predictive Auto-Scaling with AI/ML	0.25	0.25	0.25	0.25
Container-Based Scaling	0.25	0.25	0.25	0.25
Serverless Computing	0.25	0.25	0.25	0.25
Hybrid Scaling	0.25	0.25	0.25	0.25

Table 3 presents the weight distribution assigned to different cloud management strategies across four key performance parameters: scalability, resource utilization, overprovisioning risk, and implementation complexity. In this dataset, each alternative is assigned an equal weight of 0.25 for all parameters. An even weight distribution indicates that all performance criteria are considered equally important when evaluating cloud management approaches. This means that no single factor—be it scalability, resource utilization, risk, or complexity—is given a greater influence in determining the overall performance of a scaling strategy. For rule-based autoscaling, equal weighting indicates that its moderate performance across all criteria makes it a balanced, but less dynamic approach. Since this method operates on predefined rules rather than real-time optimization, it is not superior in any one category, but remains a reliable option for predictable workloads. Predictive autoscaling with AI/ML benefits from equal weighting because, while it has high scalability and resource utilization, it also has significant complexity in implementation and overprovisioning risk. Equal weighting ensures that its advantages and disadvantages are considered fairly compared to other methods. Similarly, container-based scaling, serverless computing, and hybrid scaling all receive equal weighting in each performance metric. While serverless computing excels in scalability and performance, its overprovisioning risk and implementation complexity remain a concern. Hybrid scaling provides a mix of approaches, and equal weighting ensures that its trade-off between flexibility and complexity in management is properly assessed. Overall, the equal weighting approach provides an unbiased comparison of the five cloud management strategies, allowing for an objective assessment of their overall performance. This method ensures that no single parameter disproportionately influences decision-making, leading to a well-rounded evaluation framework.

Table 4 adjusts the performance values of different cloud management strategies by applying their weights. This change allows for a fair comparison of alternatives by ensuring that each performance metric contributes proportionally to the final score. Rule-based autoscaling has relatively low scalability (0.18056) and resource

TABLE 4. Weighted normalized decision matrix 1

Rule-Based Auto-Scaling	0.18056	0.15789	0.25000	0.25000
Predictive Auto-Scaling with AI/ML	0.23611	0.23684	0.14706	0.22569
Container-Based Scaling	0.21667	0.21053	0.17857	0.23897
Serverless Computing	0.25000	0.25000	0.15625	0.21667
Hybrid Scaling	0.22778	0.20526	0.16026	0.23214

utilization (0.15789), indicating its limited adaptability in dynamic cloud environments. However, it scores the highest (0.25000) for overprovisioning risk and implementation complexity, indicating that while it is easy to implement, it tends to allocate excess resources inefficiently. Predictive autoscaling with AI/ML shows higher scalability (0.23611) and resource utilization (0.23684) compared to rule-based scaling, demonstrating its ability to dynamically optimize cloud resources. However, it has a lower over-provisioning risk (0.14706), which is beneficial, but it comes with a moderately high implementation complexity (0.22569) due to the need for AI integration and data-driven decision making. Container-based scaling provides balanced performance between AI-based and rule-based approaches with scalability (0.21667) and resource utilization (0.21053) scores. Its over-provisioning risk (0.17857) remains moderate, while its implementation complexity (0.23897) is slightly higher, reflecting the challenges of managing containerized workloads. Serverless computing has the highest scalability efficiency (0.25000) and resource utilization (0.25000), which reinforces its effectiveness in effectively handling dynamic workloads. However, its overprovisioning risk (0.15625) and implementation complexity (0.21667) indicate that while it improves resource utilization, managing serverless environments remains challenging. Hybrid scaling scores well across all metrics, with scalability (0.22778) and resource utilization (0.20526) indicating strong adaptability. Its overprovisioning risk (0.16026) and implementation complexity (0.23214) reflect the tradeoffs of combining multiple scaling approaches. Overall, the table highlights the strengths and weaknesses of each approach, guiding organizations in choosing the most appropriate cloud management strategy.

TABLE 5. Weighted normalized decision matrix 2

Rule-Based Auto-Scaling	0.92187	0.89147	1.00000	1.00000
Predictive Auto-Scaling with AI/ML	0.98581	0.98657	0.87577	0.97475
Container-Based Scaling	0.96486	0.95795	0.91932	0.98878
Serverless Computing	1.00000	1.00000	0.88914	0.96486
Hybrid Scaling	0.97700	0.95190	0.89479	0.98164

Table 5 presents the final weighted normalized decision matrix, which refines the evaluation of cloud management strategies by combining their respective weights and normalized values. This approach ensures that each alternative is evaluated on a fair and standardized scale, which allows for a clear comparison of their overall performance. Rule-based automatic scaling shows the lowest scores for scalability (0.92187) and resource utilization (0.89147) compared to the other methods. However, it maintains the highest values for overprovisioning risk (1.00000) and implementation complexity (1.00000). This also means that while rule-based scaling is simple to implement, it is also more prone to resource waste and inefficiency, making it a less optimal choice for dynamic cloud environments. Predictive Auto Scaling with AI/ML shows significant improvement in scalability (0.98581) and resource utilization (0.98657), highlighting its ability to intelligently allocate cloud resources. However, its overprovisioning risk (0.87577) is slightly lower than rule-based scaling, but still significant, reflecting the unpredictability of AI-driven predictions. Implementation complexity (0.97475) is high due to the technical challenges associated with AI integration. Container-based scaling maintains balanced performance with scalability (0.96486) and resource utilization (0.95795). Its overprovisioning risk (0.91932) is relatively well-managed, indicating that containerization helps control resource overprovisioning. Implementation complexity (0.98878) is high, indicating that deploying and managing containers requires considerable expertise. Serverless computing achieves the highest scalability (1.00000) and resource utilization (1.00000), reinforcing its superior ability to dynamically handle fluctuating workloads. However, its overprovisioning risk (0.88914) and implementation complexity (0.96486) remain concerns, reflecting the challenges of managing a functionally-based operation. Hybrid scaling performs consistently across all metrics, with strong scalability (0.97700), resource utilization (0.95190), and moderate overprovisioning risk (0.89479). Its implementation complexity (0.98164) is slightly lower than other advanced methods, making it a versatile option for organizations looking for a balanced, adaptive approach to cloud management.

TABLE 6. Preference Score WSM & Preference Score WPM

	Preference Score WSM	Preference Score WPM
Rule-Based Auto-Scaling	0.82182	0.83845
Predictive Auto-Scaling with AI/ML	0.83025	0.84571
Container-Based Scaling	0.84018	0.84473
Serverless Computing	0.85789	0.87292
Hybrid Scaling	0.81688	0.82544

Table 6 presents the Priority Score 1 and Priority Score 2, which indicate the overall ranking of cloud management strategies based on their weighted performance values. These scores help identify the most effective resource scaling approach by combining multiple evaluation parameters into a single preference score. Rule-based autoscaling has relatively low preference scores, 0.82182 (Priority Score 1) and 0.83845 (Priority Score 2), making it the least preferred option. While it is easy to implement, this indicates that it is inefficient in handling dynamic workloads and suffers from high overprovisioning risks. Organizations that rely on this method may struggle with resource waste and suboptimal performance. Predictive autoscaling with AI/ML performs better, receiving scores of 0.83025 (Priority Score 1) and 0.84571 (Priority Score 2). The slight improvement in preference scores 2 indicates that AI-driven scaling is more adaptive and efficient, but its complexity and potential risk factors prevent it from being the best choice. Container-based scaling maintains a balanced position with scores of 0.84018 (preference score 1) and 0.84473 (preference score 2). It ranks slightly below AI-based scaling, but is a strong contender due to its efficient resource utilization and scalability. However, its complexity in implementation remains a major challenge. Serverless computing emerges as the most preferred option, with maximum scores of 0.85789 (preference score 1) and 0.87292 (preference score 2). These values reinforce its superiority in scalability, performance, and cost-effectiveness, making it ideal for modern cloud environments. Despite its complexity, its advantages outweigh its disadvantages. Surprisingly, hybrid scaling has the lowest priority scores (0.81688 and 0.82544), indicating that while it combines multiple scaling techniques, it may introduce additional management overhead and implementation issues.

TABLE 7. WASPAS Coefficient

	WASPAS Coefficient
Rule-Based Auto-Scaling	0.83013
Predictive Auto-Scaling with AI/ML	0.83798
Container-Based Scaling	0.84246
Serverless Computing	0.86540
Hybrid Scaling	0.82116
lambda	0.5

Table 7 presents the Weighted Aggregate Sum Product Assessment (WASPAS) coefficient, a decision metric used to rank different cloud management strategies based on multiple evaluation parameters. The WASPAS method combines both the weighted sum and weighted product models to provide a more comprehensive ranking. A lambda (λ) value of 0.5 indicates that both models are given equal importance in the evaluation process. Rule-based autoscaling has a WASPAS coefficient of 0.83013, which places it among the lower ranked alternatives. This confirms that although rule-based methods are simple and easy to implement, they lack efficiency in dynamic workload management and can lead to resource waste due to fixed scaling rules. Predictive autoscaling with AI/ML shows improvement with a WASPAS coefficient of 0.83798. This reflects its ability to analyze real-time data and dynamically adjust resources, making it a more adaptive and efficient scaling approach. However, its complexity and potential implementation challenges prevent it from ranking at the top. Container-based scaling achieves a slightly better WASPAS coefficient of 0.84246, making it one of the most consistent and efficient strategies. Containers offer flexibility and better resource utilization, but they require efficient management and bandwidth, which may explain why it did not achieve a high score. Serverless computing ranks very high with a WASPAS coefficient of 0.86540, reinforcing its superiority in scalability, resource efficiency, and cost-effectiveness. Despite its complexity in implementation, its advantages in automatic scaling and cost savings make it a more optimal choice. Surprisingly, hybrid scaling has the lowest coefficient at 0.82116, indicating that while it combines multiple scaling techniques, its complexity and management overhead reduce its overall performance.

TABLE 8. Rank

	RANK
Rule-Based Auto-Scaling	4
Predictive Auto-Scaling with AI/ML	3
Container-Based Scaling	2
Serverless Computing	1
Hybrid Scaling	5

Table 7 presents the final rankings of various cloud management strategies based on their overall performance evaluation. The rankings are derived from multiple decision-making techniques, including WASPAS coefficients, preference scores, and weighted decision matrices, which ensure a comprehensive assessment of the performance of each method in cloud resource scaling. Serverless computing ranks first (rank = 1), making it the most effective approach to resource scaling. This result is consistent with previous findings that serverless computing excels in scalability, resource utilization, and cost optimization. Its ability to dynamically allocate resources without manual intervention makes it a highly adaptive and cost-effective method for modern cloud environments. Container-based scaling secures the second rank, reflecting its strong balance between performance and flexibility. Containers allow for efficient resource allocation and workload distribution, especially when organized using platforms such as

Kubernetes. While it lags behind serverless computing in terms of fully automated scaling, it remains a viable option for organizations with a structured but scalable cloud architecture. Predictive autoscaling with AI/ML ranks third, indicating that while AI-driven techniques provide superior forecasting and optimization, they require complex implementation and significant computational overhead. This approach is valuable for businesses that prioritize intelligent workload management but need the infrastructure to support machine learning-based decision-making. Rule-based autoscaling ranks fourth, showing its limitations in handling dynamic workloads. While it is easy to implement, it lacks the adaptive capabilities required for efficient cloud resource scaling, leading to overprovisioning and resource waste. Finally, despite combining multiple scaling techniques, hybrid scaling ranks last (5th). Its complexity and management overhead outweigh its benefits, making it a less desirable choice for organizations looking for streamlined cloud management solutions.

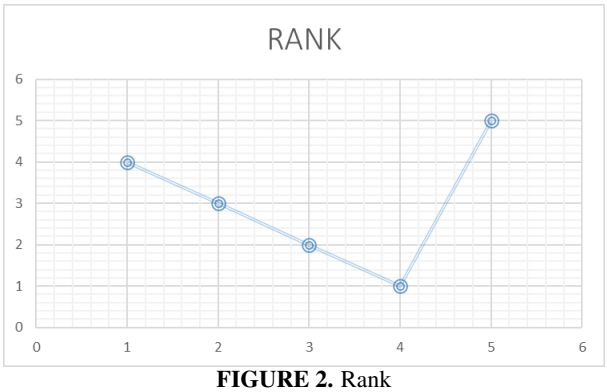

FIGURE 2. Rank

Figure 2 displays a ranking of different cloud resource scaling strategies based on overall performance evaluation. The x-axis represents five different scaling approaches, while the y-axis represents their ranking position. A lower ranking value indicates better performance, while a higher-ranking value indicates poorer performance compared to other alternatives. The graph clearly shows that serverless computing (rank = 1) is the most preferred method for cloud resource scaling. Its position at the lowest point on the graph reflects its high performance in terms of scalability, cost optimization, and adaptability. This is consistent with its ability to dynamically allocate resources based on real-time demand, reducing operational overhead, and ensuring seamless workload management. Container-based scaling (rank = 2) follows closely, highlighting its strong performance in resource allocation and flexibility. The small increase in ranking compared to serverless computing indicates that while containers are useful for scaling, they require careful orchestration and management, making them slightly less optimal than a serverless approach. Predictive autoscaling with AI/ML (rank = 3) is the next best approach, demonstrating a balance between automation and performance. However, its higher complexity and computational requirements for AI-driven predictions contribute to its lower ranking compared to serverless and container-based solutions. Rule-based autoscaling (rank = 4) has a higher ranking (poorer performance) than the previous three, indicating its limited adaptability and potential inefficiency. While it is easy to implement, it lacks dynamic scaling capabilities, making it less suited to handling fluctuating cloud workloads. Finally, hybrid scaling (rank = 5) appears at the very top of the chart, making it the least preferred option. This suggests that although it includes several scaling techniques, its complexity and management overhead outweigh its benefits, resulting in low overall performance.

4. CONCLUSION

Cloud computing continues to evolve, requiring advanced resource scaling algorithms to improve performance, cost, and efficiency. This study analyzed various cloud scaling techniques, including rule-based autoscaling, predictive autoscaling with AI/ML, container-based scaling, serverless computing, and hybrid scaling. Each approach offers unique advantages and tradeoffs in terms of scalability, resource utilization, overprovisioning risk, and implementation complexity. The findings indicate that serverless computing has emerged as a more efficient scaling approach, providing greater scalability and optimal resource utilization. However, its implementation complexity and overprovisioning risks must be effectively managed. Container-based scaling offers a balanced alternative, combining flexibility with robust resource efficiency, making it suitable for modern cloud applications. Predictive autoscaling with AI/ML demonstrates more adaptive and intelligent resource allocation, but its complexity and computational overhead pose challenges. Rule-based autoscaling, while simple and predictable, lacks the dynamic adaptability required for modern cloud environments, leading to inefficiencies. While combining multiple techniques, hybrid scaling introduces additional management overhead, reducing overall performance. The comparative assessment highlights the importance of choosing the appropriate scaling strategy based on an organization's specific needs. Businesses with fluctuating workloads may benefit from predictive or serverless scaling, while structured environments may favor container-based approaches. The research also underscores the need to balance scalability with cost-effectiveness and complexity to achieve optimal cloud resource management.

Future work should focus on refining predictive models for autoscaling, leveraging AI-driven resource allocation, and exploring hybrid solutions that combine multiple scaling strategies without excessive complexity. Additionally, incorporating real-world cloud workload datasets can further validate these findings and provide deeper insights into the practical implementation of scaling techniques.

REFERENCES

[1]. Mardani, Abbas, Mehrbakhsh Nilashi, Norhayati Zakuan, Nanthakumar Loganathan, Somayeh Soheilirad, Muhamad Zameri Mat Saman, and Othman Ibrahim. "A systematic review and meta-Analysis of SWARA and WASPAS methods: Theory and applications with recent fuzzy developments." Applied soft computing 57 (2017): 265-292.

[2]. Radomska-Zalas, Aleksandra. "Application of the WASPAS method in a selected technological process." Procedia Computer Science 225 (2023): 177-187.

[3]. Kutlu Gundogdu, Fatma, and Cengiz Kahraman. "Extension of WASPAS with spherical fuzzy sets." Informatica 30, no. 2 (2019): 269-292.

[4]. Zavadskas, Edmundas Kazimieras, Shankar Chakraborty, Orchi Bhattacharyya, and Jurgita Antucheviciene. "Application of WASPAS method as an optimization tool in non-traditional machining processes." Information Technology and Control 44, no. 1 (2015): 77-88.

[5]. Baykasoğlu, Adil, and İlker Gölcük. "Revisiting ranking accuracy within WASPAS method." Kybernetes 49, no. 3 (2020): 885-895.

[6]. Stanujkić, Dragiša, and Darjan Karabašević. "An extension of the WASPAS method for decision-making problems wić intuitionistic fuzzy numbers: a case of website evaluation." Operational Research in Engineering Sciences: Theory and Applications 1, no. 1 (2018): 29-39.

[7]. Lukić, Radojko, Dragana Vojteski Kljenak, Slavica Anđelić, and Milan Gavrlilović. "Application of WASPAS method in the evaluation of efficiency of agricultural enterprises in Serbia." Economics of Agriculture 68, no. 2 (2021): 375-388.

[8]. Turskis, Zenonas, Edmundas Kazimieras Zavadskas, Jurgita Antuchevičienė, and Natalja Kosareva. "A hybrid model based on fuzzy AHP and fuzzy WASPAS for construction site selection." (2015).

[9]. Turskis, Zenonas, Nikolaj Goranin, Assel Nurusheva, and Seilkhan Boranbayev. "A fuzzy WASPAS-based approach to determine critical information infrastructures of EU sustainable development." Sustainability 11, no. 2 (2019): 424.

[10]. Perec, Andrzej, and Aleksandra Radomska-Zalas. "WASPAS optimization in advanced manufacturing." Procedia Computer Science 207 (2022): 1193-1200.

[11]. Addis, Bernardetta, Danilo Ardagna, Barbara Panicucci, Mark S. Squillante, and Li Zhang. "A hierarchical approach for the resource management of very large cloud platforms." IEEE Transactions on Dependable and Secure Computing 10, no. 5 (2013): 253-272.

[12]. Liu, Xiaolong, Shyan-Ming Yuan, Guo-Heng Luo, Hao-Yu Huang, and Paolo Bellavista. "Cloud resource management with turnaround time driven auto-scaling." IEEE Access 5 (2017): 9831-9841.

[13]. Wang, Qingyang, Hui Chen, Shungeng Zhang, Liting Hu, and Balaji Palanisamy. "Integrating concurrency control in n-tier application scaling management in the cloud." IEEE Transactions on Parallel and Distributed Systems 30, no. 4 (2018): 855-869.

[14]. Jennings, Brendan, and Rolf Stadler. "Resource management in clouds: Survey and research challenges." Journal of Network and Systems Management 23 (2015): 567-619.

[15]. Al-Sharif, Ziad A., Yaser Jararweh, Ahmad Al-Dahoud, and Luay M. Alawneh. "ACCRS: autonomic based cloud computing resource scaling." Cluster Computing 20 (2017): 2479-2488.

[16]. Manvi, Sunilkumar S., and Gopal Krishna Shyam. "Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey." Journal of network and computer applications 41 (2014): 424-440.

[17]. Forell, Tim, Dejan Milojicic, and Vanish Talwar. "Cloud management: Challenges and opportunities." In 2011 IEEE international symposium on parallel and distributed processing workshops and Phd forum, pp. 881-889. IEEE, 2011.

[18]. Sahni, Jyoti, and Deo Prakash Vidyarthi. "Heterogeneity-aware adaptive auto-scaling heuristic for improved QoS and resource usage in cloud environments." Computing 99 (2017): 351-381.

[19]. Moens, Hendrik, and Filip De Turck. "A scalable approach for structuring large-scale hierarchical cloud management systems." In Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), pp. 1-8. IEEE, 2013.

[20]. Meng, Shicong, and Ling Liu. "Enhanced monitoring-as-a-service for effective cloud management." IEEE Transactions on Computers 62, no. 9 (2012): 1705-1720.

[21]. Runsewe, Olubisi, and Nancy Samaan. "Cloud resource scaling for time-bounded and unbounded big data streaming applications." IEEE Transactions on Cloud Computing 9, no. 2 (2018): 504-517.

[22]. Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services." In Algorithms and Architectures for Parallel Processing: 10th International Conference, ICA3PP 2010, Busan, Korea, May 21-23, 2010. Proceedings. Part I 10, pp. 13-31. Springer Berlin Heidelberg, 2010.

[23]. Bellini, Pierfrancesco, Ivan Bruno, Daniele Cenni, and Paolo Nesi. "Managing cloud via smart cloud engine and knowledge base." Future Generation Computer Systems 78 (2018): 142-154.

[24]. Liu, Xiao-Long, Shyan-Ming Yuan, Guo-Heng Luo, and Hao-Yu Huang. "Auto-scaling mechanism for cloud resource management based on client-side turnaround time." In Genetic and Evolutionary Computing: Proceedings of the Ninth International Conference on Genetic and Evolutionary Computing, August 26-28, 2015, Yangon, Myanmar-Volume II

9, pp. 209-219. Springer International Publishing, 2016.
[25]. Wang, Lizhe, Yan Ma, Jining Yan, Victor Chang, and Albert Y. Zomaya. "pipsCloud: High performance cloud computing for remote sensing big data management and processing." Future Generation Computer Systems 78 (2018): 353-368.
[26]. Beloglazov, Anton, and Rajkumar Buyya. "Energy efficient resource management in virtualized cloud data centers." In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826-831. IEEE, 2010.
[27]. Gong, Zhenhuan, Xiaohui Gu, and John Wilkes. "Press: Predictive elastic resource scaling for cloud systems." In 2010 International Conference on Network and Service Management, pp. 9-16. Ieee, 2010.
[28]. Citron, Daniel, and Aviad Zlotnick. "Testing large-scale cloud management." IBM Journal of Research and Development 55, no. 6 (2011): 6-1.
[29]. Radhakrishnan, Ganesan. "Adaptive application scaling for improving fault-tolerance and availability in the cloud." Bell Labs Technical Journal 17, no. 2 (2012): 5-14.
[30]. Singh, Sukhpal, and Inderveer Chana. "QoS-aware autonomic resource management in cloud computing: a systematic review." ACM Computing Surveys (CSUR) 48, no. 3 (2015): 1-46.

