

SLA-Aware Cost Optimization in Hybrid Cloud Architectures: Evaluating AI-Driven Proactive Scaling for AWS Lambda and Kubernetes

Nikita Singh

*Department of Computer Science
& Engineering, Chandigarh University
Mohali, 140413, Punjab, India
nikusisodia2000@gmail.com*

Rakesh Kumar

*Department of Computer Science
& Engineering, Chandigarh University
Mohali, 140413, Punjab, India
rakesh77kumar@gmail.com*

Meenu Gupta

*School of Computer Science
& Engineering, Geeta University
Naultha, 132145, Panipat, India
Gupta.meenu5@gmail.com*

Abstract—Resource management is crucial in cloud computing environments as it ensures quality of service while controlling the associated operational costs amidst dynamic events. Most existing auto-scaling techniques involve the reactive deployment model where decisions are made based on threshold levels. Hence implementation of such methods may suffer from delayed provisioning of resources leading to violation of SLA requirements and wastage of resources. In given paper, we develop a SLA-Aware Cost Optimization Framework designed for hybrid cloud architecture which combines AWS Lambda and Kubernetes platforms. The framework integrates real-time workload monitoring, AI-driven workload predictions, SLA analysis, and pro-active scaling to ensure the selection of the most cost-effective resources. It assesses the input workload demand, response time, resource utilization, and predicted resource expense while keeping the requirements of SLA achievement. The hybrid execution model allows for making scaling decisions over both serverless and container approaches in case of any changing workload conditions. We conduct experimental evaluation of the system in normal, increasing, fluctuating, and bursty workload conditions and compare the results with existing predictive auto-scaling models. The study highlights the effectiveness of AI-driven proactive scaling for achieving a practical balance between cloud performance, SLA compliance, and operational expenditure in hybrid cloud environments.

Index Terms—Cloud Computing, Resource Management, SLA Optimization, Cost Optimization, Artificial Intelligence, Workload Prediction, Proactive Auto-Scaling, AWS Lambda, Kubernetes, Hybrid Cloud, Resource Utilization.

I. INTRODUCTION

Cloud computing is the technology that enables applications to obtain and release computing resources in accordance with varying workloads, which is why elasticity becomes critical to properties such as performance and cost of infrastructure [1]. The issue that arises is that when demand for an application fluctuates rapidly and at the same time SLA requirements have to be fulfilled, resource allocation becomes complicated [2]. Most of the conventional auto-scaling mechanisms rely on the usage of thresholds and reactive actions that respond after the changes in the workload occur, which can lead to delays in execution, overspending of resources, and performance drop [3].

Making cost-efficient use of cloud resources, today it has become an important field of research around methods that can predict demand and enhance the efficiency of resource allocation [4]. Introducing the serverless computing offers a new type of resource and pricing system, which involves automatic resource allocation via AWS Lambda, depending on the functional load [5]. In spite of the fact that in this way the need for permanent infrastructure utilization is reduced, there are different factors — the duration of functions, frequency of requests, cold starts, which form the final cost and influence the quality of the service [6]. As a result, the studies have performed numerous research in order to find various models for analysis of cost-effectiveness of serverless applications depending on workloads [7]. Thus it has been indicated that utilization of resources in accordance with immediate demand might not prove to be beneficial.

Kubernetes and serverless technologies coexist in a hybrid cloud model where workloads may be executed through any of the available execution models. The performance and cost dynamics of AWS Lambda and containerized applications may vary significantly based on the nature of workload and application [8]. Techniques such as predictive scaling are showing promises in improving the responsiveness of Kubernetes provisioning by predicting resource needs of workload instead of waiting for threshold breaches [9]. It should be emphasized that the recent use of AI-enabled and SLO-oriented approaches demonstrates the importance of accounting for service requirements when making scaling decisions [10]. Meanwhile, existing approaches often focus on only one of the paradigms or treat cost management and SLA compliance as separate matters [11]. Thus, the purpose of this study is to introduce SLA-aware cost optimization technique implemented through AI-enabled proactive scaling for AWS Lambda and Kubernetes in order to minimize expenses, efficiency of resource usage, response time, and compliance with SLAs under different workload conditions [12].

II. LITERATURE REVIEW

Cloud-native and hybrid architectures are being increasingly adopted, which has broadened the realm of research in intelligent resource management, SLA compliance, and cost optimization. The latest studies look at more advanced directions such as a transition from traditional reactive scaling based methods to utilizing machine learning for prediction and adaptive resource allocation or multi-objective optimization. Here, we present a review of the latest research related to cloud auto-scaling, SLA-aware resource management, serverless cost optimization, and AI-enabled proactive scaling, to highlight the gap tackled in our research.

A. SLA and Cost-Aware Cloud Resource Management

Machiraju et al. [13] provided an overview of various types of ML-driven autoscaling mechanisms and pinpointed the major issues that still need to be resolved, such as the joint consideration of costs, energy, SLO, and heterogeneity. Qu et al. [14] developed a classification system to categorize the available cloud autoscaling methods and analyzed the drawbacks of reactive methods for dynamic applications. Chen et al. [15] focused on self-aware and self-adaptive processes of cloud autoscaling and emphasized the necessity of making adaptive decisions when system conditions change. Gupta et al. [16] described their hybrid method of reactive and proactive autoscaling designed specifically for the context of SLA-constrained environments. Barua et al. [17] focused on an investigation of AI-based SLA optimization, which involves predictive analytics and reinforcement learning methods enabling the process of dynamic resource allocation and SLA management. The findings of the listed studies highlight the significance of the combination between performance requirements and resource decision-making processes.

B. Serverless Cost and AI-Driven Optimization

Serverless computing is an implementation of pricing model based on usage where the cost to execute depends on different parameters such as the memory for running a function, the time taken for the execution, and the invocation patterns. Elgamal et al. [18] proposed techniques for cost optimization in AWS Lambda made possible by techniques like function fusion, placement, and memory allocation with the consideration of payments. Mera Menéndez et al. [19] compared Kubernetes, AWS Lambda, and other serverless deployment options for microservices, analyzing their cost and performance under different workloads. Golshani and Ashtiani [20] investigated proactive autoscaling in the cloud with the help of temporal convolutional neural networks to foresee the workload in the future. Dang-Quang and Yoo [21] implemented Bi-LSTM for prediction of workloads of Kubernetes-based containers that proved that the deep learning models can be utilized for efficient container autoscaling. Imdoukh et al. [22] conducted further studies with the help of machine learning to prove how effective autoscaling may be for containerized applications through workload patterns.

C. Hybrid Cloud Optimization and Research Gap

In recent years, researchers have shown a trend of considering cost and performance as connected objectives for optimization. Souza and Netto [23] investigated SLA-aware auto-scaling using application-level data and reported reductions in SLA violations and resource requirements during rapidly changing workloads. Yamsani and Reddy [24] examined deep reinforcement learning that is SLA-aware to conduct dynamic edge-cloud scheduling paying joint attention to SLA parameters, latency, and energy-efficiency. Ranjbari and Torkestani [25] described an approach based on adaptive learning to ensure SLA efficiency and reduce energy consumption. Even though multiple advances have been made in this field, it can be seen that the majority of researchers mostly study serverless and containerized environments separately and look at either cost, latency, or SLA compliance alone. Hence, there is a limited number of studies that describe a comprehensive approach considering AI-based proactive scaling in AWS Lambda and Kubernetes environments while trying to optimize either cost or SLA.

III. PROPOSED METHODOLOGY

In this study, the authors present a cost optimization framework that applies SLA principles to hybrid cloud environments by means of using AI technology for workload analysis and proactive scaling of resources. The framework studied concerns both AWS Lambda and Kubernetes aiming at ensuring that the costs of running applications in the cloud correlate with the performance of the application itself as well as the SLAs. The holistic idea of SLA-aware cost optimization method is shown in Fig. 1.

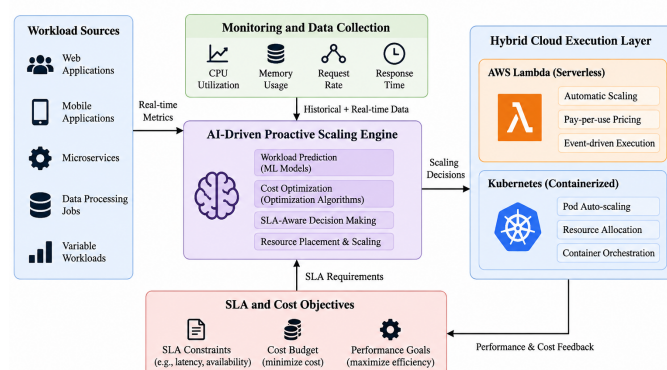


Fig. 1. Architecture of an SLA-aware cost optimization framework for hybrid cloud environments.

A. System Architecture Overview

The system is constantly gathering information about workloads, costs, resources, and SLAs from cloud infrastructure. These indicators are channeled to the AI analysis unit that can provide the prediction of future workloads and possible SLA risks. After that, the framework's optimization engine

evaluates if the workloads should stay where they are or be allocated through Kubernetes and AWS Lambda directly.

The decision regarding scaling is based on the forecasts of demand, availability of resources, response time, and costs of implementation of the workloads. The general scheme of the proposed framework is shown in Fig. 2.

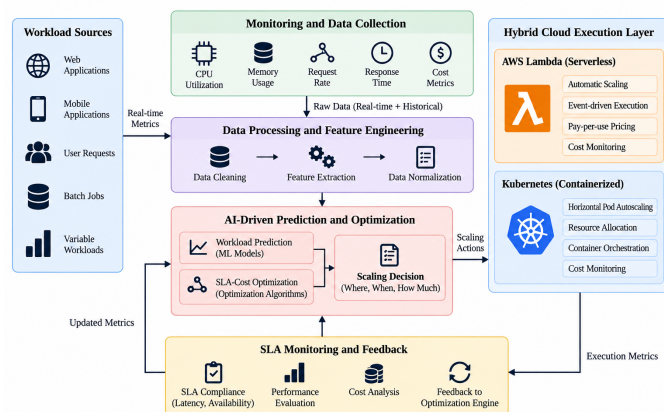


Fig. 2. Architecture of the proposed SLA-aware cost optimization framework.

B. System Components

The proposed framework consists of five major components, summarized in Table I.

TABLE I
COMPONENTS OF THE PROPOSED FRAMEWORK

Component	Function
Monitoring Module	Collects workload, resource, and SLA metrics
Data Processing	Prepares and normalizes monitoring data
AI Prediction	Forecasts workload and resource demand
SLA-Cost Optimization	Selects cost-efficient SLA-compliant actions
Cloud Execution Manager	Applies scaling across Lambda and Kubernetes

- The Monitoring Module – is responsible for the collection of metrics regarding the workload, resources used, performance, and Service Level Agreement (SLA).
- The Data Processing Module – is tasked with the cleaning and normalizing the incoming monitoring data.
- The AI Prediction Module – helps to predict workload and performance needs.
- The SLA-Cost Optimization Engine – determines the method of resource distribution and actions in regards to scaling while considering SLA and costs.
- The Cloud Execution Manager – is in charge of implementing the devised methods of working via AWS Lambda and Kubernetes and monitoring the outcome of these actions.

C. SLA-Aware Cost Optimization

The optimization engine evaluates possible scaling decisions using both SLA and cost factors. A simplified objective function is defined as:

$$J = \alpha C + \beta T + \gamma V \quad (1)$$

where C represents cloud resource cost, T represents application response time, V represents SLA violation level, and α, β, γ are weighting factors. The selected action minimizes the objective while satisfying the defined SLA constraints.

D. Proactive Scaling Workflow

The framework employs a cyclic decision-making procedure:

- 1) Gather real-time metrics related to workload, spending, and SLA..
- 2) Handle and standardize the obtained data.
- 3) Foresee the incoming workload and demand for performance.
- 4) Gauge the repercussions of resource actions on SLA and cost.
- 5) Identify a decision regarding either the scaling or placement of resources.
- 6) Execute the chosen decision in AWS Lambda or Kubernetes.
- 7) Evaluate the results and change the future decision process accordingly.

IV. EXPERIMENTAL SETUP

In order to evaluate how practical the efficacy of the suggested SLA-aware cost optimization method, a controlled hybrid cloud environment is utilized using the combination of AWS Lambda and Kubernetes. The study investigates whether AI-based proactive scaling is capable of decreasing cost in cloud operations while still ensuring sufficient performance and compliance with SLA terms.

A. Experimental Environment and Framework Configuration

The use of AWS Lambda for serverless processing and Kubernetes for containerized jobs provides a test environment. The monitoring is done in real-time and involves gathering CPU usage, memory usage, incoming requests, response timing, invocation time, and cost of resources. This information is then pre-processed through the AI prediction engine. The costs are minimized through the SLA-cost optimization module, which uses data from the prediction module to select the suitable environment and execution strategy based on expected demand, expected cost, and SLA constraints.

B. Workload Configuration

Load configuration includes normal, rising, variable, and burst request types. The historical workload data is utilized in the training of the forecast model while the next samples are used in testing. Different workload levels are applied in both environments, i.e., Lambda and Kubernetes, in order to investigate whether preemptive actions decrease unnecessary

consumption of the resources along with maintaining the required level of service quality. The experimental parameters and configurations used in the evaluation are summarized in Table II.

TABLE II
EXPERIMENTAL CONFIGURATION

Parameter	Configuration
Serverless Platform	AWS Lambda
Container Platform	Kubernetes
Monitoring Metrics	CPU, Memory, Request Rate, Response Time
Cost Metrics	Execution and Resource Cost
AI Approach	Workload Prediction
Scaling Strategy	SLA-Aware Proactive Scaling
Workload Patterns	Normal, Increasing, Fluctuating, Bursty
Baseline	Reactive Auto-Scaling
Evaluation Metrics	SLA, Response Time, Utilization, Cost

C. Performance Evaluation Metrics

The effectiveness of the proposed framework is evaluated using SLA compliance, resource utilization, response time, total cloud cost, and cost savings.

SLA compliance is calculated as:

$$SLA_{comp} = \frac{N_{SLA}}{N_{total}} \times 100 \quad (2)$$

where N_{SLA} represents the number of requests satisfying the defined SLA requirements and N_{total} represents the total number of requests.

Resource utilization is measured as:

$$RU = \frac{R_{used}}{R_{allocated}} \times 100 \quad (3)$$

where R_{used} represents the utilized resources and $R_{allocated}$ represents the total allocated resources.

The average application response time is calculated as:

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i \quad (4)$$

where T_i represents the response time of the i^{th} request and N is the total number of requests.

The total cloud resource cost is calculated as:

$$C_{total} = \sum_{i=1}^N R_i \times P_i \times T_i \quad (5)$$

where R_i represents the allocated resource quantity, P_i represents the corresponding resource price, and T_i represents its usage duration.

Finally, cost savings relative to the baseline approach are calculated as:

$$CS = \frac{C_{base} - C_{proposed}}{C_{base}} \times 100 \quad (6)$$

where C_{base} represents the cost of the conventional reactive approach and $C_{proposed}$ represents the cost of the proposed framework.

The experimental configuration and evaluation metrics provide the basis for comparing the proposed SLA-aware proactive strategy with conventional reactive scaling across AWS Lambda and Kubernetes. The results of this evaluation are presented in the following section.

V. RESULTS AND PERFORMANCE EVALUATION

The evaluation of the suggested SLA-based cost optimization scheme was conducted based on the experimental setup explained in Section IV. The evaluation presents SLA adherence, resource utilization, response latency, total cloud expenses, and cost savings across multiple workloads. This method was compared to traditional reactive scaling employing AWS Lambda and Kubernetes.

A. SLA Compliance Analysis

The ability of the framework to meet service requirements was tested based on workloads that had normal, growing, and burst behaviors in the request patterns. The SLA compliance was measured based on (1) based on the requests that were able to keep within allowed limits of latency and availability.

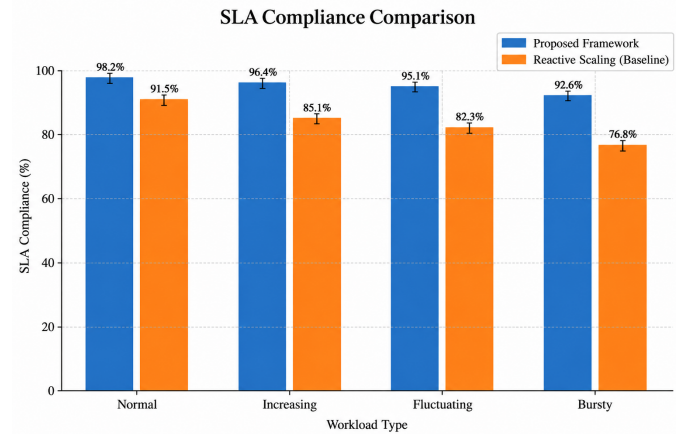


Fig. 3. SLA compliance comparison between the proposed proactive framework and reactive scaling.

The framework proposed in Fig. 3 achieves a higher percentage of SLA-compliant requests through predicting incoming workload before the need for resources arises. This ensures minimum disruption in service levels due to unexpected workload variations.

B. Resource Utilization Analysis

Resource utilization was determined from (2) by comparing the actual amount of resources utilized by each approach against the resources allocated.

The results from Fig. 4 show that proactive resource allocation is effective at minimizing the use of unnecessary resources while ensuring resource sufficiency for application workloads.

The framework makes alterations to the resources depending on the workload.

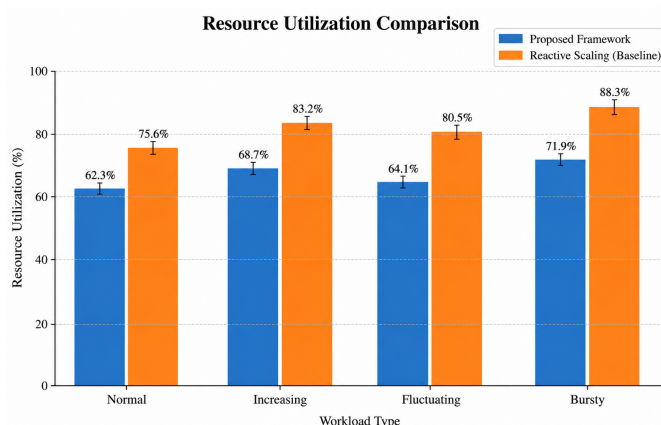


Fig. 4. Resource utilization comparison under varying workload conditions.

C. Response Time Analysis

The average response time was evaluated with the usage of (3) to gauge how proactive resource allocation impacted performance of the application.

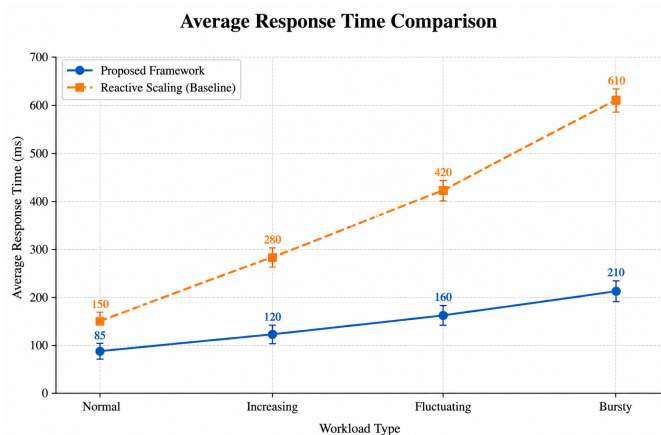


Fig. 5. Average response time under different workload conditions.

Fig. 5 shows that proactive scaling helps in minimizing response latency caused due to waiting for reactive scaling activities. This framework benefits mostly during rising or sudden loads as it requires immediate adjustment of resources to ensure that the application is responsive enough.

D. Cost Optimization Analysis

The cost of the cloud resources was derived from equation (4), taking into account resource usage, cost, and time of execution, for AWS Lambda and Kubernetes. Cost advantage of the recommended approach is depicted in Fig. 6, since unnecessary allocation of resources has been avoided. The optimization process considers estimated demand for resources and SLAs so that they are allocated only at required times without incurring SLA breaches.

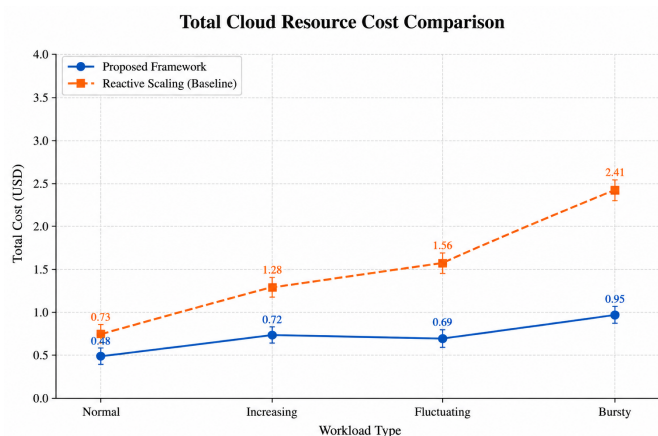


Fig. 6. Cloud resource cost comparison between the proposed and reactive approaches.

E. Cost Savings and Scalability Analysis

The method used for computing cost savings relies on (5) and involves evaluating the new framework against the standard reactive approach. The framework was also tested for scalability with just enough increase in the workload.

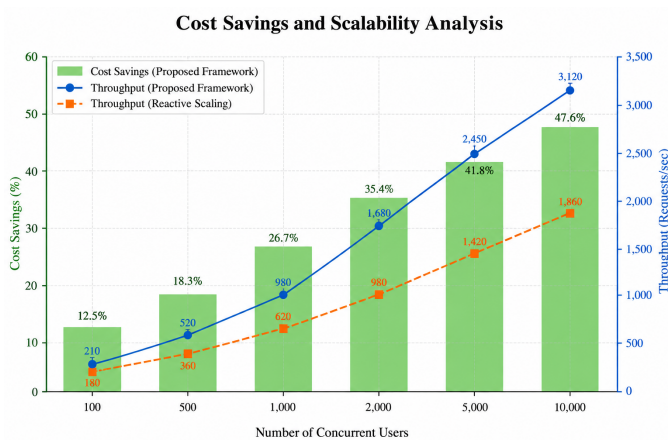


Fig. 7. Cost savings and scaling behavior under increasing workload.

Results in the Fig. 7 show the framework's usefulness to adapt resources with increasing workload. The approach focuses on achieving the right level of cost effectiveness and performance of applications in heterogeneous cloud environment by combining proactive decisions with SLA constraints.

VI. CONCLUSION AND FUTURE WORK

The suggested SLA-aware cost optimization framework offers a smart way of handling cloud resources in AWS Lambda and Kubernetes. The framework combines workload prediction, SLA monitoring, and predicting the need for scaling actions and aims to cut down on unnecessary cloud costs while preserving intended application performance. Unlike traditional models that act reactively, this framework combines expected resource consumption and SLA requirements into the process of implementing scaling actions as well.

The assessment deals with the following metrics: SLA compliance, resource utilization; response time; total cloud costs; cost savings. The framework operates in a way that enables it to reshape resource allocation based on the workload, keeping a suitable balance between service quality and operational costs. The hybrid doing model underlines the flexibility of workload distribution between serverless and containerized environments.

In the future, it will be necessary to implement and test using bigger real-world workloads and different cloud services. More sophisticated AI models could be explored to enhance workload forecasting and decision making based on costs. Future research may also include multi-cloud settings, energy-optimizing strategies, dynamic pricing and more enhanced SLA criteria.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Special Publication 800-145, 2011.
- [2] R. Buyya, S. K. Garg, and R. N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions," in Proc. 2011 IEEE Int. Conf. on Cloud and Services Computing (CSC), 2011, pp. 1–10, doi: 10.1109/CSC.2011.6138522.
- [3] T. Lorido-Botrán, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, vol. 12, no. 4, pp. 559–592, 2014, doi: 10.1007/s10723-014-9314-7.
- [4] M. Amiri and L. Mohammad-Khanli, "Survey on Prediction Models of Applications for Resources Provisioning in Cloud," Journal of Network and Computer Applications, vol. 82, pp. 93–113, 2017, doi: 10.1016/j.jnca.2017.01.016.
- [5] Amazon Web Services, "AWS Lambda Developer Guide," Amazon Web Services, 2025.
- [6] N. Mahmoudi and H. Khazaei, "Performance Modeling of Serverless Computing Platforms," IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2834–2847, 2022, doi: 10.1109/TCC.2020.3033373.
- [7] C. Lin and H. Khazaei, "Modeling and Optimization of Performance and Cost of Serverless Applications," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 3, pp. 615–632, 2021, doi: 10.1109/TPDS.2020.3028841.
- [8] C. Allen, X. Li, A. S. Abdelfattah, T. Cerny, and D. Taibi, "Comparing Cost and Performance of Microservices and Serverless in AWS: EC2 vs Lambda," in Next Generation Data Science: Second Southwest Data Science Conference (SDSC 2023), Revised Selected Papers, vol. 2113, Springer, 2024, pp. 60–72, doi: 10.1007/978-3-031-61816-1-5.
- [9] M. Abdullah, W. Iqbal, A. Mahmood, F. Bukhari, and A. Erradi, "Predictive Autoscaling of Microservices Hosted in Fog Microdata Center," IEEE Systems Journal, vol. 15, no. 1, pp. 1275–1286, 2021, doi: 10.1109/JSYST.2020.2997518.
- [10] V. Koc and V. Kamanuru, "Predictive and Reinforcement Learning-Based Framework for Cloud Resource Optimization," in Proc. 2025 Systems and Information Engineering Design Symposium (SIEDS), 2025, doi: 10.1109/SIEDS65500.2025.11021093.
- [11] S. Alharthi, A. Alshamsi, A. Alseiri, and A. Alwarafy, "Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions," Sensors, vol. 24, no. 17, Art. no. 5551, 2024, doi: 10.3390/s24175551.
- [12] T. Wang, S. Ferlin-Reiter, and M. Chiesa, "Predicting CPU Usage for Proactive Autoscaling," in Proc. 1st Workshop on Machine Learning and Systems (EuroMLSys), 2021, pp. 31–38, doi: 10.1145/3437984.3458831.
- [13] V. S. Machiraju, V. Kumar, and S. Sharma, "ML-Based Autoscaling for Elastic Cloud Applications: Taxonomy, Frameworks, and Evaluation," Mathematical and Computational Applications, vol. 31, no. 2, Art. no. 49, 2026, doi: 10.3390/mca31020049.
- [14] C. Qu, R. N. Calheiros, and R. Buyya, "Auto-Scaling Web Applications in Clouds: A Taxonomy and Survey," ACM Computing Surveys, vol. 51, no. 4, Art. no. 73, pp. 1–33, 2018, doi: 10.1145/3148149.
- [15] T. Chen, R. Bahsoon, and X. Yao, "A Survey and Taxonomy of Self-Aware and Self-Adaptive Cloud Autoscaling Systems," ACM Computing Surveys, vol. 51, no. 3, Art. no. 61, pp. 1–40, 2018, doi: 10.1145/3190507.
- [16] S. Gupta, M. T. Islam, and R. Buyya, "A Hybrid Reactive-Proactive Auto-scaling Algorithm for SLA-Constrained Edge Computing," in Proc. Australasian Computer Science Week (ACSW 2026), 2026, pp. 1–10, doi: 10.1145/3793811.3793826.
- [17] B. Barua, N. I. Protul, F. A. Aurpa, M. Mubarrat, S. M. N. Howlader, and M. S. Kaiser, "AI for Service-Level Agreement (SLA) Optimization in Microservices-Based Cloud Infrastructures," in Proc. 3rd Int. Conf. on Sustainable Computing and Data Communication Systems (ICSCDS), 2025, pp. 485–490, doi: 10.1109/ICSCDS65426.2025.11166747.
- [18] T. Elgamal, A. Sandur, K. Nahrstedt, and G. Agha, "Costless: Optimizing Cost of Serverless Computing through Function Fusion and Placement," in Proc. 2018 IEEE/ACM Symposium on Edge Computing (SEC), 2018, pp. 300–312, doi: 10.1109/SEC.2018.00029.
- [19] J. Mera Menéndez, J. E. Labra Gayo, E. Riesgo Canal, and A. Echevarría Fernández, "A Comparison Between Traditional and Serverless Technologies in a Microservices Setting," arXiv preprint arXiv:2305.13933, 2023.
- [20] E. Golshani and M. Ashtiani, "Proactive Auto-Scaling for Cloud Environments Using Temporal Convolutional Neural Networks," Journal of Parallel and Distributed Computing, vol. 154, pp. 119–141, 2021, doi: 10.1016/j.jpdc.2021.04.006.
- [21] N.-M. Dang-Quang and M. Yoo, "Deep Learning-Based Autoscaling Using Bidirectional Long Short-Term Memory for Kubernetes," Applied Sciences, vol. 11, no. 9, Art. no. 3835, 2021, doi: 10.3390/app11093835.
- [22] M. Imdoukh, I. Ahmad, and M. G. Alfailakawi, "Machine Learning-Based Auto-Scaling for Containerized Applications," Neural Computing and Applications, vol. 32, no. 13, pp. 9745–9760, 2020, doi: 10.1007/s00521-019-04507-z.
- [23] A. A. D. P. Souza and M. A. S. Netto, "Using Application Data for SLA-Aware Auto-Scaling in Cloud Environments," arXiv preprint arXiv:1506.05443, 2015.
- [24] N. Yamsani and C. Reddy P, "SLA-Aware Deep Reinforcement Learning for Adaptive EdgeCloud Task Scheduling," Scientific Reports, vol. 16, Art. no. 10037, 2026, doi: 10.1038/s41598-026-40237-8.
- [25] M. Ranjbari and J. A. Torkestani, "A Learning Automata-Based Algorithm for Energy and SLA Efficient Consolidation of Virtual Machines in Cloud Data Centers," Journal of Parallel and Distributed Computing, vol. 113, pp. 55–62, 2018, doi: 10.1016/j.jpdc.2017.10.009.