e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:05/May-2023 Impact Factor- 7.868 www.irjmets.com

MONITORING PERFORMANCE COMPUTING ENVIRONMENTS AND AUTOSCALING USING AI

Ravi Pulle*[1], Gaurav Anand*[2], Satish Kumar*[3]

*[1]Principal Member Of Technical Staff, Salesforce, Inc., San Francisco, CA, USA.

*[2]Senior Principal Software Engineer, TDS Telecommunications LLC, Madison, WI, USA.

*[3]Partner Engineer, Apple Inc., Cupertino, CA, USA.

DOI : https://www.doi.org/10.56726/IRJMETS40883

ABSTRACT

Modern organizations face unpredictable demands and dynamic workloads, necessitating effective management of computing resources. Autoscaling in cloud computing offers a solution by enabling applications to autonomously adjust their capacity in response to changing demand. This paper explores the utilization of AI-powered algorithms to address the challenges encountered in monitoring and autoscaling, considering factors such as memory requirements, network traffic, CPU utilization, and custom metrics. AI-driven models offer numerous advantages, including enhanced resource utilization, scalability, reliability, reduced maintenance overheads, continuous availability, cost-effectiveness, and simplified management of the computing environment. However, complexities in configuration, potential performance degradation, inconsistent performance, security concerns, and increased costs are notable drawbacks. By comparing AI-powered techniques with other traditional methods, this research evaluates the role of AI in overcoming challenges faced by alternative approaches. Through experimental evaluation and comprehensive analysis, this study demonstrates the superiority of AI-driven techniques in diverse aspects such as CPU utilization, memory utilization, throughput, and response time. Moreover, the paper identifies multiple areas for further improvement, aiming to enhance efficiency and reduce computing costs.

Keywords: Performance Computing Environment, Autoscaling, Monitoring, Artificial Intelligence (AI), Resource Optimization, Workload Dynamics, Rule-Based Scaling, Threshold-Based Scaling, Predictive Scaling, Cost Optimization.

I. INTRODUCTION

1.1 Problem statement

In modern performance computing environments, the complexities posed by dynamic workloads, resource utilization, and efficiency present unprecedented challenges. Efficient monitoring and autoscaling techniques are crucial in ensuring optimal performance. Ineffective monitoring and scaling systems often result in increased computing costs. Researchers have recognized the necessity of addressing these challenges in complex environments through advanced techniques that balance cost reduction and high performance. This paper aims to validate the hypothesis that employing AI models for monitoring and autoscaling surpasses traditional methods in terms of efficiency.

1.2 Study objectives

The primary objective of this study is to conduct a comprehensive analysis of the challenges associated with monitoring performance and scaling in computing environments. The research will investigate various existing techniques such as predictive, rule-based, and threshold-based scaling and compare them with AI-driven systems. The study aims to demonstrate the potential of AI in improving monitoring and autoscaling by addressing the limitations encountered by conventional methods. To achieve this, experimental results from the comparison will be meticulously documented, analyzed, and interpreted. This research endeavor contributes significantly to the understanding and enhancement of resource utilization efficiency, reduction of computing costs, and improvement of system performance.

1.3 Elements of comparative analysis

Various elements are relevant in setting up an AI model for monitoring performance and autoscaling in a computing environment. Previous research indicates that while using AI to achieve this goal, it is significant to define performance metrics relevant to that environment in monitoring the system's performance, including network traffic, memory usage, and CPU utilization [12]. Data collection on performance metrics involves various tools such as DataDog, Prometheus, and Grafana. Since the AI model must learn the resource utilization patterns and predict future demands in computing resources, it requires effective training using the collected data.

According to research, deciding the computing resources required for efficient performance in a dynamic environment is difficult and requires a fast and timely understanding of workloads and services [11]. The trained AI model will be able to determine when autoscaling is necessary automatically. When AI predicts that resource utilization is likely higher, it initiates autoscaling to ensure adequate environmental computing resources. However, if the resource demand is likely to decrease, it reduces resources committed to the environment to save on costs. The AI model has alerts to notify the administrator of potential challenges and when performance metrics exceed predefined thresholds. Continuous monitoring and data collection improves AI's accuracy, scalability, resource optimization, and reliability in the computing environment.

1.4 Paper structure

This paper evaluates various aspects relevant to monitoring and autoscaling and unique perspectives incorporated by AI-powered systems in this role. The report is divided into major sections, including a comprehensive literature review on AI-powered monitoring and autoscaling elements. The main components of AI-powered techniques that improve autoscaling and monitoring performance are analyzed. It assesses the issues in monitoring performance and autoscaling in computing environments and the corresponding solutions for the challenges. The paper explores the merits and demerits of using AI in autoscaling and monitoring. The next section provides experimental analysis results for monitoring performance and optimal autoscaling by comparing AI-powered paradigms with other frameworks with similar functionality. The last section incorporates various recommendations for the use and improving AI-based monitoring and autoscaling.

II. LITERATURE REVIEW AND CHALLENGES

2.1 Literature review

AI techniques can mitigate the complexities and challenges faced in other methods of monitoring and autoscaling. It is capable of hyperparameter optimization, efficient deployment, and automated model selection. AI-powered techniques are associated with numerous advantages over other methods for monitoring and autoscaling. These are discussed in the following section.

2.1.1 Accurate workload prediction

AI analyzes computing environment historical data and accurately identifies patterns to predict future demand for computing resources. It can derive accurate patterns by considering resource demand elements like specific days, times of the day, seasons, and trends. Since it considers millions of data points, it performs more precisely. AI is perceived as a more accurate strategy than traditional methods. Continuous learning enables proactive autoscaling [9]. It enhances the adjustment of resources based on predicted future resource demand and utilization.

2.1.2 Response time optimization and proactive scaling

Research indicates that systems use AI-powered algorithms such as temporal convolutional neural networks in proactive autoscaling [2]. AI-powered monitoring and autoscaling reduce response to changing workload demands in the computing environment. AI models ensure faster responses to changes in resource demands than traditional reactive auto-scaling solutions [2]. They analyze underlying workload conditions to make informed decisions on autoscaling. After analyzing workload patterns, the system can reduce fluctuations in response time by a significant percentage and eliminate performance bottlenecks during peak times.

2.1.3 Anomaly detection and alerting

AI can detect real-time anomalies by comparing historical data to performance metrics in specific environments. The model can detect abnormal system behaviors such as unprecedented drops or spikes in resource demand and utilization, unusual network traffic, and deviations in response times. The system can trigger appropriate actions and alert the administrator of such anomalies to mitigate risks, prevent system downtime, improve performance, and ensure system stability.

2.1.4 Dynamic resource allocation based on real-time demand

Over-provisioning computing resources is wasteful and leads to additional unnecessary costs. Under-provisioning degrades system performance. AI-powered autoscaling systems can understand real-time resource demand and appropriately adjust resources committed to the computing environment [1]. It considers many factors, including workload patterns, service-level objectives, and user behavior. It compares the data with intelligence collected before or history to predict the optimal number of resources at any given time. Dynamic resource allocation incorporates the reduction or increase of resources and minimizes wastage.

2.1.5 Self-learning and Adaptation

AI-powered monitoring and autoscaling models continuously learn and adapt to changing system behavior and workload patterns [10]. It analyzes feedback and collected data to update and improve the ability to precisely predict the future, help make better decisions, and improve accuracy [15]. Self-learning allows the system to adapt and adjust to meet the dynamically changing needs while maintaining high-level performance.

2.1.6 Efficiency and cost saving

AI-driven models for monitoring performance and auto-scaling in cloud environments minimize costs while ensuring that needs are efficiently met. Research indicates a rise in adopting a cost-aware approach in autoscaling web applications [5]. Computing resources have considerable value. The more an organization utilizes resources, the higher the price of computing. The system considers various factors, including reservation, on-demand, business constraints, and spot instances, to make informed decisions and ensure cost efficiency. The model learns from historical data, including patterns in resource utilization, to recommend the most efficient autoscaling decision guided by efficient pricing with high performance.

2.1.7 Intelligent management of alerts and reduced manual intervention

The organization aims to reduce the manual and cognitive burden for efficient monitoring, reporting, and refocusing administrative personnel on more significant tasks. According to research, autoscaling enables the organization to adjust computing resources, automatic monitoring, creation or release of ECS based on policy, and configuration of RDS safelist and load balancers [6]. AI-powered autoscaling and monitoring help in the automatic management of alerts and alarms. It can be trained to understand their frequency, severity, and correlation to other incidents. Continuous learning enhances the ability to filter false alarms, group correlated events, and prioritize critical alarms.

2.2 Challenges faced

Monitoring and autoscaling require robust frameworks, advanced algorithms, real-time data analysis tools and techniques, and high-level domain expertise to interpret results and understand alarms. Feedback from previous challenges helps the AI model learn how to address future challenges. Continuous learning improves the performance of the AI as it gains more capability to understand workload patterns, commit resources, and optimize their usage.

2.2.1 Scalability challenges

Autoscaling systems have predefined minimum and maximum numbers of computing resources they can scale up or down partly due to infrastructure constraints, architectural limitations, or cost considerations. Upon reaching the limits, autoscaling fails to meet demand, often resulting in resource shortages or degrading performance. Dynamic changes in size and complexity can characterize modern computing environments. This change needs to be monitored and rapidly respond to resource demand by increasing or decreasing resource allocation. Monitoring and accurately handling changing workloads is a challenge for many system administrators.

2.2.2 Real-time monitoring and alerting

Real-time monitoring requires the model to have the capability to collect, process, analyze, and interpret data to ensure a timely and precise response. AI-driven models help administrators identify anomalies, analyze performance, and identify bottlenecks. However, developing and training the model with efficient capabilities to handle high-frequency data updates and processes and rapidly generate responses is challenging.

2.2.3 Data accuracy and noise

Dealing with data quality challenges is a considerable challenge in predictive computing. Monitoring systems perform better when the data they use is accurate and reliable. AI algorithms process and analyze such data to help make efficient and informed decisions. However, data collected during monitoring performance is often incomplete and noisy. If not cleaned, it leads to inaccurate predictions in resource demand, causes false alarms, and delays decisions. Data cleaning often results in computing overheads.

2.2.4 Dynamic workloads

Computing environments, especially those relying on cloud services and infrastructure, are characterized by dynamically changing demands and workloads. Fluctuating resource needs are a challenge to organizations. Predicting resource needs accurately, given dynamic scenarios, is challenging. AI-powered systems require comprehensive training to address the fluctuating demands. In addition, future demands may not always be aligned with historical patterns. For instance, the computing environment can experience a sudden spike or drop in workload, which still requires AI algorithms to make decisions and scale efficiently. Therefore, the changing needs are difficult to accurately predict and ensure optimal system performance while keeping prices as low as possible.

2.2.5 Cost optimization

Autoscaling has the objective of optimizing resource usage while maintaining low costs. However, achieving this goal is difficult since accurately forecasting resource demands is challenging. Often, there is a tradeoff when trying to avoid under-provisioning and over-provisioning computing resources [13]. There is a need to accurately analyze the data collected and understand business requirements and objectives to strike the right balance.

2.2.6 Complex dependencies

The heterogeneity of computing resources is a challenge among many experts and systems. Environments for performance computing encompass various resources, including virtual machines, hardware configurations, distributed computing systems, and containers. Monitoring highly integrated resources with different frameworks, protocols, and APIs is complex and challenging. In monitoring and autoscaling, tools lack resilience in handling the complexity of a dynamic computing environment; the system struggles to collect, process, analyze, and interpret performance data, leading to delays and low accuracy levels in autoscaling. Computing environments in modern societies are characterized by interconnected and integrated components and services, resulting in complex dependencies. Growth in this complexity can result in new bottlenecks and degrade system performance. It is significant to understand the impacts of complex dependencies and ensure that autoscaling decisions do not impact system performance negatively.

2.2.7 Security and compliance

Modern AI-powered systems are faced with ethical issues and security challenges. Research indicates that rule-based autoscaling and policy adherence require in-depth expertise and knowledge, which is difficult for many organizations [8]. As such, AI models must adhere to security and legal requirements and standards, including data security, privacy, and compliance with regulations. Research indicates that AI-powered monitoring and autoscaling systems aim to improve performance while maintaining participant data privacy [7]. The system handles sensitive data that guide decisions in monitoring performance and autoscaling. Collecting, processing, analyzing, and using sensitive data to monitor performance and optimize resource utilization requires compliance with regulatory policies and security standards.

2.3 Existing solutions

Addressing challenges in monitoring performance and autoscaling using AI-driven models requires robust operational frameworks and prior planning to address numerous capacity issues in the computing environment. Multiple solutions match every challenge expected in a computing environment. Organizations often combine these elements to ensure improvement and better decision-making for the AI-powered performance monitoring and autoscaling system.

2.3.1 Cloud-based autoscaling and monitoring services

Cloud service providers offer autoscaling and monitoring services. Examples of such instances include Google Cloud Monitoring, Amazon CloudWatch, and Azure Monitoring have built-in monitoring and autoscaling features. They assist their clients in managing computing services, networks, and resources.

2.3.2 AI-powered tools for anomaly detection

AI-powered Machine Learning (ML) algorithms can collect and analyze historical data and compare it with real-time data to detect inconsistencies and anomalies. They employ time-series analysis, statistical modeling, and unsupervised learning to identify inconsistencies. Once they detect anomalies, they trigger the alarm and appropriate autoscaling actions. Examples of tools used for anomaly detection include Kibana and Elasticsearch.

2.3.3 Dynamic scaling algorithms

Algorithms can help autoscaling and optimize resource utilization by learning from workload patterns. The determining elements for appropriate action to take by the algorithms include response time, workload predictions, and resource utilization. The algorithms fall into three major categories, i.e., hybrid approach, reactive, and proactive algorithms. According to a study, the current container orchestration, including Amazon EC2 and Kubernetes, uses autoscaling rules based on static thresholds and relies solely on infrastructure elements like memory and CPU states [4]. However, AI-powered systems for monitoring and autoscaling can change this by incorporating dynamic multi-level rules for applications.

2.3.4 Cost optimization strategies

Algorithm designers have in mind the cost objectives of the organization. The algorithms take into account various forms of business constraints in scaling decisions. The basis of such algorithms is spot instances, on-demand pricing, historical patterns of resource use, and reserved instances—the goal of considering these factors in ensuring the lowest costs possible while ensuring satisfactory or high-level performance. Examples of cost-driven algorithms include HPA and AWS Autoscaling.

2.3.5 Application-aware scaling

Scaling decisions must consider both application and infrastructure metrics and dependencies. Decision-making is complex in uncertain environments with changing demands in computing resources. Scholars have argued that Reinforcement Learning (RL) has great potential to help systems make decisions in such environments [3]. Many factors indicate application awareness, including que lengths, request rates, database connections, and response time. Application-aware tools that offer efficient monitoring and autoscaling capabilities include AppDynamics, DataDog, and New Relic.

2.3.6 Use of security and compliance frameworks

Cloud providers such as MS Azure help integrate compliance and security requirements for their clients. According to research, with great care, cloud providers can help specify autoscaling policies without performance violations [8]. Cloud services are often targets of malicious and threat actors. Therefore, employing best practices, meeting requirements and standards, and complying with legal regulations are significant. Various frameworks help in autoscaling and monitoring that have the best security and compliance. Guiding frameworks include GDPR guidelines, PCI DSS, and CIC benchmarks.

2.3.7 Continuous monitoring and improvements

Continuous monitoring is a critical component of RL, as it offers real-time feedback and observations to inform the decision-making process of RL agents. It empowers agents to adapt to dynamic environments, make informed choices, balance exploration and exploitation, and refine policies based on current states and observed rewards. To ensure optimal real-time actions, it is important to maintain metric data availability as close to real-time as possible. This ensures that RL agents can make timely and effective decisions, allowing them to navigate complex and ever-changing scenarios with maximum efficiency.

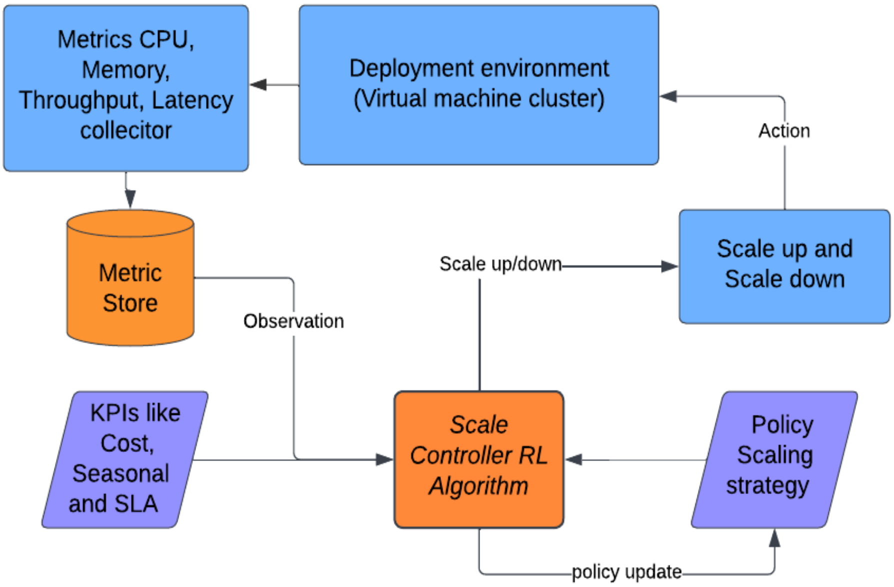
Metrics CPU, Memory, Throughput, Latency collector
Deployment environment (Virtual machine cluster)
Action
Metric Store
Scale up/down
Scale up and Scale down
Observation
KPIs like Cost, Seasonal and SLA
Scale Controller RL Algorithm
Policy Scaling strategy
policy update

Figure 1: Autoscaling framework based on a reinforcement learning model

III. METHODOLOGY

3.1 Research approach

The study conducted a simulation of AI-driven monitoring and autoscaling performance to demonstrate its superiority over other methods. The simulation incorporates ML algorithms [Fig. 1] based on reinforcement learning and neural networks to improve monitoring and autoscaling. A workload[14] of varying throughput and intensity is executed, and results are compared. The metrics are used in the comparative analysis of the performance of the AI-powered model as compared to rule based scaling and fixed count.

3.2 Experiment setup

The experiment involves setting up a performance computing environment on cloud infrastructure with multiple virtual machines (VMs) [Fig. 1]. The configurations of the VMs are similar, and they have equal processing power, network bandwidth, and memory. A workload generator executes dynamically changing workloads like a real-world or organizational scenario. Various metrics are considered for monitoring and autoscaling.

3.3 Data description

The input data consists of the current state, action taken, reward received, and next state. The current state represents the environment's situation, the action is the decision made by the agent, the reward is the immediate feedback, and the next state is the subsequent environment state. This input data enables RL algorithms to learn optimal strategies by iteratively interacting with the environment, maximizing cumulative rewards over time and effectively solving complex problems in dynamic environments. The current state is measured using metrics CPU (percentage), Memory usage, Throughput (requests per second) and Request latency (milli seconds) which are collected using an application monitoring agent. Seasonal history data is also considered as an input.

IV. PERFORMANCE METRICS RESULTS

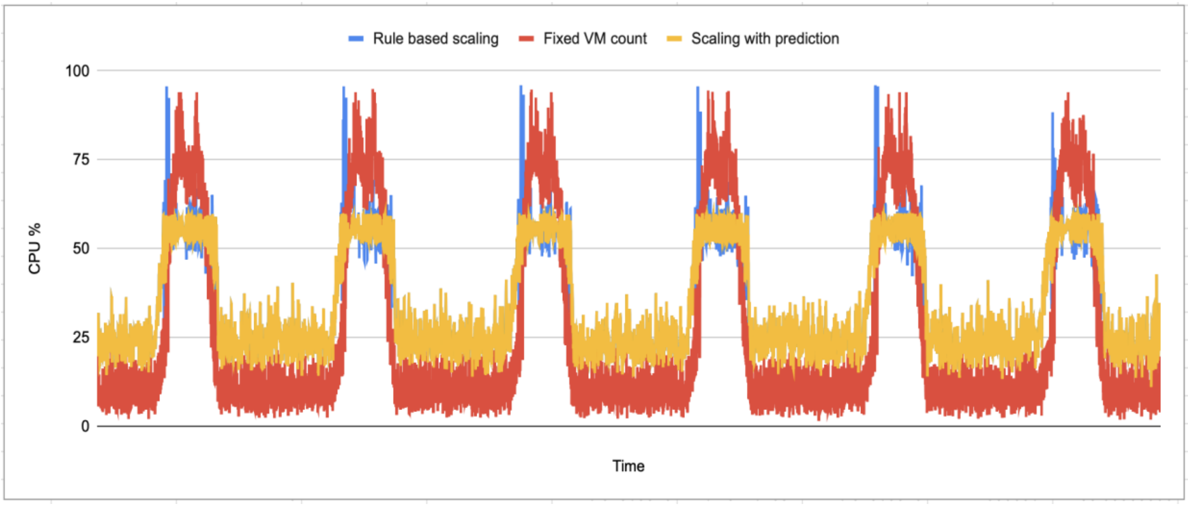

Figure 2: six days peak/off peak CPU time line series with different scaling approaches

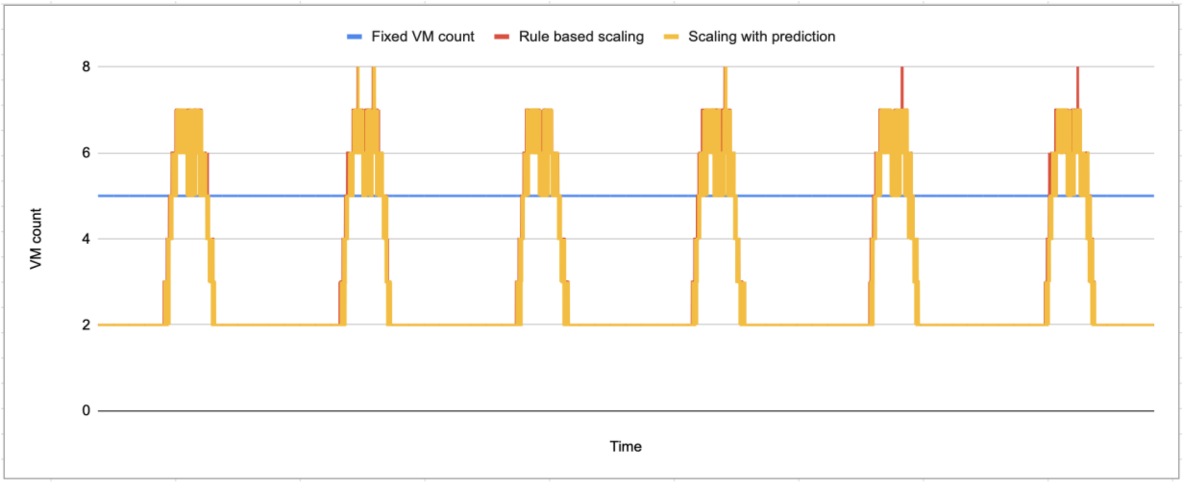

Figure 3: six days peak/off peak CPU time line series with different scaling approaches

Table 1. VM cost

Scaling technique	VM time (min)	AWS VM cost	Total Cost	% Saving
Fixed VM count	43200	$0.264/Hour	$190.08	0%
Rule based scaling	26311	$0.264/Hour	$115.77	39.1%
Scaling with Prediction	25108	$0.264/Hour	$110.48	42.1%

Table 2. Request latencies with scaling techniques

Scaling technique	Response time P90 ms	Response time P99 ms	Response time Avg ms
Fixed VM count	57.1	121.7	76.45
Rule based scaling	56.3	102.2	68.65
Scaling with prediction	55.9	96.9	62.23

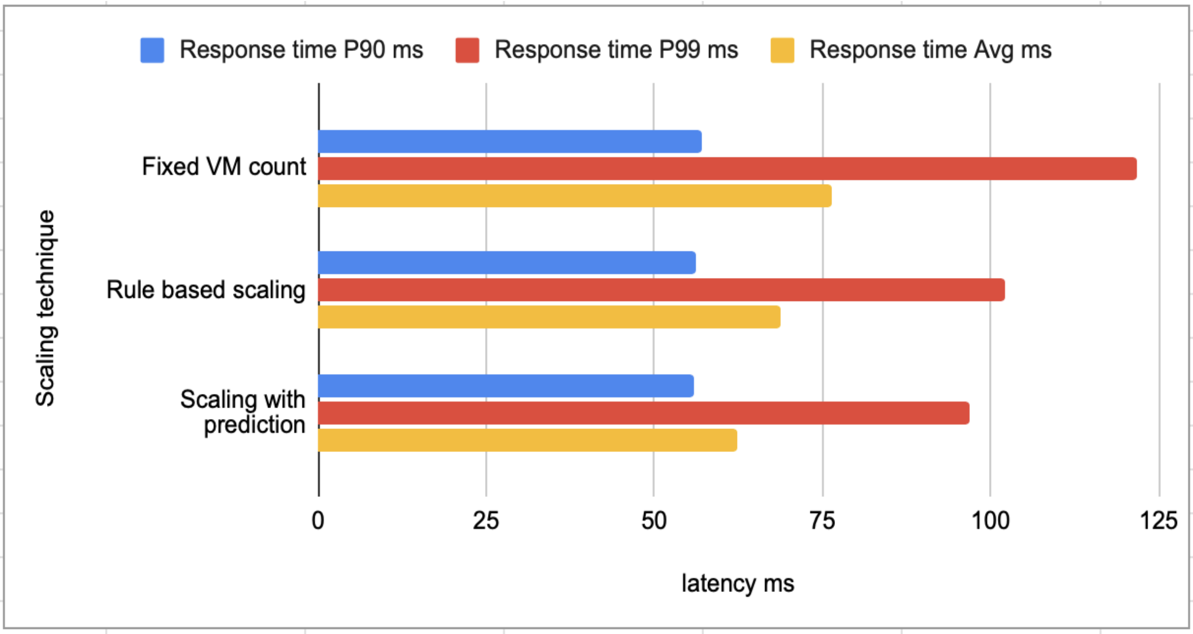

Figure 4: Request Latency performance comparison

V. DISCUSSION

Various metrics were taken into account to examine the application under test, and it was observed that the CPU% and request latency were particularly affected. The workload was simulated during both peak and off-peak hours, and the metrics CPU% and request latency were analyzed using different scaling techniques.

When a fixed number of virtual machines (VMs) were used, the CPU utilization was either low or spiked to a higher value (>75%), which had an impact on request latencies. On the other hand, with rule-based scaling, the CPU% still spiked temporarily until the VM became available to handle the load. This scenario can be seen in Fig. 2, and it resulted in additional VM requests, as depicted in Fig. 3. In some cases, this led to overusing VM resources.

However, with prediction-based scaling, VMs were requested well in advance to meet future demands. This approach allowed for handling requests within the expected CPU range and latencies, providing a more efficient solution.

The evaluation of VM costs revealed a significant saving [Table 1] of 39% compared to using a fixed VM count. Furthermore, an additional 3% cost reduction was achieved by implementing a predictive auto-scaling approach.

When examining request latencies, it becomes apparent that the fixed count approach results in significantly higher P99 values, primarily due to elevated CPU usage during peak hours. The rule-based scaling approach also has a slight impact on P99 latencies [Table 2]. However, it is worth noting that the predictive scaling approach demonstrated better latencies across various metrics such as P90, P99, and averages [Fig. 4]. In the case of rule-based scaling, the minor impact observed on latencies and CPU spikes can be attributed to the lag in container availability after request.

The experimental results indicate the effectiveness of AI-powered monitoring and autoscaling over other methods in performance computing environments. However, there are specific areas of improvement for more efficient capabilities. Well-defined metrics for assessing performance are significant in monitoring and accuracy evaluation. KPIs and metrics must align with user expectations and organizational goals. Specific metric examples include resource utilization, response time, throughput, user satisfaction, and error frequencies. The system must gather relevant monitoring data, process, and store results. Specific relevant data points in performance monitoring and autoscaling include performance counters, system logs, user interactions, and network traffic. A complex configuration framework for this model requires high-level expertise. Setting up AI-powered autoscaling and monitoring can be complicated and time-consuming for the IT team.

VI. CONCLUSION

The paper has explored challenges faced in monitoring performance computing environments and autoscaling, existing solutions to the challenges, and advantages and disadvantages of AI-powered models. The paper explores capacity issues and various methods of autoscaling, including rule-based, fixed, and predictive autoscaling. The experimental results evaluated the advantages of the AI-powered method over the other techniques in CPU utilization, memory utilization, throughput, and response time. The superiority of AI is demonstrated. However, AI-powered models still face complexities, scalability challenges, and dynamic workload changes. Future improvements can focus on enhancing the accuracy of predictive models and data quality to support decisions, security, and policy compliance.

VII. REFERENCES

[1] Alibaba Cloud, "Auto Scaling," Alibaba, 1 1 2023. [Online]. Available: https://www.alibabacloud.com/product/auto-scaling?spm=a3c0i.11847046.7364687560.1.b81945decxUkCc. [Accessed 17 5 2023].

[2] M. S. Al-Asaly, M. M. Hassan and A. Alsanad, "A cognitive/intelligent resource provisioning for cloud computing services: opportunities and challenges," Soft Computing, vol. 23, no. 19, p. 9069–9081, 2019.

[3] H. Alipour, "Model-Driven Machine Learning for Predictive Cloud Auto-scaling," 1 5 2019. [Online]. Available: https://core.ac.uk/download/pdf/241087136.pdf. [Accessed 17 5 2023].

[4] M. S. Aslanpour, M. Ghobaei-Arani and A. N. Toosi, "Auto-scaling Web Applications in Clouds: A Cost-Aware Approach," Journal of Network and Computer Applications, vol. 95, no. 1, pp. 26-41, 2017.

[5] N. Birari, "Autoscaling: Advantages and Disadvantages," ESDS Software Solution Ltd., 8 2 2023. [Online]. Available: https://www.esds.co.in/kb/autoscaling-advantages-and-disadvantages/. [Accessed 17 5 2023].

[6] T. Chen, R. Bahsoon and X. Yao, "A Survey and Taxonomy of Self-Aware and Self-Adaptive Cloud Autoscaling Systems," ACM Computing Surveys, vol. 1, no. 1, pp. 1-36, 2018.

[7] S. Chouliaras and S. Sotiriadis, "Auto-scaling containerized cloud applications: A workload-driven approach," Simulation Modelling Practice and Theory, vol. 121, no. 1, p. e102654, 2022.

[8] A. Evangelidis, D. Parker and R. Bahsoon, "Performance modeling and verification of cloud-based auto-scaling policies," Future Generation Computer Systems, vol. 87, no. 1, pp. 629-638, 2018.

[9] I. Fé, R. Matos, J. Dantas, C. Melo, T. A. Nguyen, D. Min, E. Choi, F. A. Silva and P. R. M. Maciel, "Performance-Cost Trade-Off in Auto-Scaling Mechanisms for Cloud Computing," Sensors (Basel), vol. 22, no. 3, p. e1221, 2022.

[10] Y. Garí, D. A. Monge, E. Pacini, C. Mateos and C. G. Garino, "Reinforcement learning-based application Autoscaling in the Cloud: A survey," Engineering Applications of Artificial Intelligence, vol. 102, no. 1, p. e104288, 2021.

[11] E. Golshani and M. Ashtiani, "Proactive auto-scaling for cloud environments using temporal convolutional neural networks," Journal of Parallel and Distributed Computing, vol. 154, no. 1, pp. 119-141, 2021.

[12] Y. Sun, H. Ochiai and H. Esaki, "Decentralized Deep Learning for Multi-Access Edge Computing: A Survey on Communication Efficiency and Trustworthiness," IEEE Transactions on Artificial Intelligence, vol. 3, no. 1, pp. 963-972, 2022.

[13] S. Taherizadeh and V. Stankovski, "Dynamic Multi-level Auto-scaling Rules for Containerized Applications," The Computer Journal, vol. 62, no. 2, pp. 174-197, 2019.

[14] SocialNetwork micro service https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork.

[15] Gaurav Anand, Sharda Kumari and Ravi Pulle, "Fractional-Iterative BiLSTM Classifier : A Novel Approach to Predicting Student Attrition in Digital Academia" SSRG International Journal of Computer Science and Engineering, Volume 10 Issue 5, 1-9, May 2023