International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

AI-Driven Predictive Scaling for Multi-Cloud Resource Management: Using Adaptive Forecasting, Cost-Optimization, and Auto-Tuning Algorithms

Charan Shankar Kummarapurugu

Sr Cloud DevOps Engineer, Brambleton, VA, USA
Email: *charanshankar[at]outlook.com*

Abstract: ***Serverless computing has revolutionized cloud infrastructure by enabling application development without managing underlying servers. However, integrating serverless with multi-cloud environments introduces unique security and scaling challenges. This paper presents an AI-driven predictive scaling approach using three key algorithms: Adaptive Forecasting Algorithm (AFA), Cost-Optimized Resource Allocation Algorithm (CORA), and AI-Based Auto-Tuning Algorithm (AITA). These algorithms address the challenges of workload prediction, resource optimization, and performance tuning. Experimental results demonstrate significant cost reductions and performance improvements compared to conventional methods.***

Keywords: Serverless computing, multi-cloud, predictive scaling, AI algorithms, cost optimization, performance tuning

1. Introduction

With the growing adoption of multi-cloud strategies, organizations face challenges in effectively managing and scaling resources across diverse cloud platforms. Serverless computing, known for its scalability and reduced operational overhead, is increasingly deployed in multi-cloud environments. However, traditional scaling techniques are reactive and often result in either over-provisioning (leading to increased costs) or underprovisioning (affecting performance).

This paper proposes an AI-driven predictive scaling approach, which leverages adaptive forecasting, cost optimization, and real-time auto-tuning to address these issues. Our approach uses three core algorithms: Adaptive Forecasting Algorithm (AFA), Cost-Optimized Resource Allocation Algorithm (CORA), and AI-Based Auto-Tuning Algorithm (AITA), which collectively optimize resource management across cloud environments.

The structure of the paper is as follows: Section II discusses related work in predictive scaling and cloud resource management. Section III presents the proposed algorithms and system architecture. Section IV covers the experimental setup and performance evaluation, while Section V concludes the paper with insights and future work.

2. Related Work

The challenge of efficient resource management in cloud computing, particularly in multi-cloud environments, has been a subject of extensive research. This section provides an overview of existing work in related areas and highlights how our proposed approach builds upon and extends these efforts.

a) Multi-Cloud Resource Management

- In [1], task offloading in multi-cloud computing is discussed, focusing on challenges in resource selection and management. The work emphasizes the need for advanced decision-making processes for distributing workloads across multiple cloud providers.
- Virtual machine placement for optimizing resource allocation in cloud data centers using queuing approaches is investigated in [2]. Although this work provides insights into efficiency optimization, it mainly addresses singlecloud environments, leaving the complexities of multicloud scenarios unaddressed.
- The concept of InterCloud, a federation of cloud environments for scaling application services, was introduced in [5]. This laid the groundwork for multi-cloud resource management, but did not include AI-driven predictive techniques.

b) Auto-scaling Approaches

- Traditional rule-based autoscaling (e.g., Amazon AWS, Azure) responds to predefined thresholds such as CPU or memory usage. A comprehensive review of these autoscaling techniques is provided in [4]. Although simple to implement, these reactive policies often result in either over-provisioning or under-provisioning.
- Predictive autoscaling systems, like the one presented in [7], use forecasting models to improve upon thresholdbased methods. This approach improves resource usage but is limited to single-cloud environments.

c) Predictive Models for Resource Scaling

- Empirical prediction models for adaptive resource provisioning in the cloud, using time-series analysis and linear regression, were proposed in [8]. These models improve resource utilization by forecasting resource usage based on historical data.

- Time-series models such as ARIMA, while effective for demand prediction, lack adaptability for sudden workload changes. Deep learning models, particularly Long ShortTerm Memory (LSTM) networks, have been shown to improve prediction accuracy in [3], though their integration into cost-optimized scaling remains limited.

d) AI and Machine Learning in Cloud Computing

- The use of deep reinforcement learning for resource management was explored in [10], demonstrating potential in adapting to dynamic environments.
- XGBoost, introduced in [9], is a scalable tree boosting system that has been applied to cloud resource demand forecasting. This ensemble method demonstrates significant improvement in prediction accuracy.

e) Privacy and Security in Multi-Cloud Environments

- Federated learning, introduced in [4], allows machine learning models to be trained across decentralized data, enabling privacy-preserving optimization in multi-cloud environments.

f) Our Contribution

- In contrast to existing work, this paper introduces an integrated AI-driven predictive scaling strategy tailored for multi-cloud resource management, addressing several key limitations.
- Adaptive Forecasting Algorithm (AFA): Combines time-series analysis and deep learning to predict workload patterns in multi-cloud environments.
- Cost-Optimized Resource Allocation Algorithm (CORA): Incorporates cost information into predictive scaling, optimizing resource distribution across multiple cloud providers.
- AI-Based Auto-Tuning Algorithm (AITA): Dynamically adjusts application configurations based on workload changes and resource constraints to enhance performance.
- Together, these components form a comprehensive framework that addresses the entire lifecycle of resource management in multi-cloud environments, from prediction to allocation and performance optimization.

3. Proposed Architecture and Methodology

The proposed AI-driven predictive scaling framework consists of five main components, as shown in Fig. 4. The components work together to collect data, predict resource demand, optimize cost, and tune performance in real-time.

- Data Collection Module: Gathers historical and realtime data on resource utilization across multiple cloud providers. This data is used as input for the Adaptive Forecasting Algorithm (AFA) to predict future resource demand.

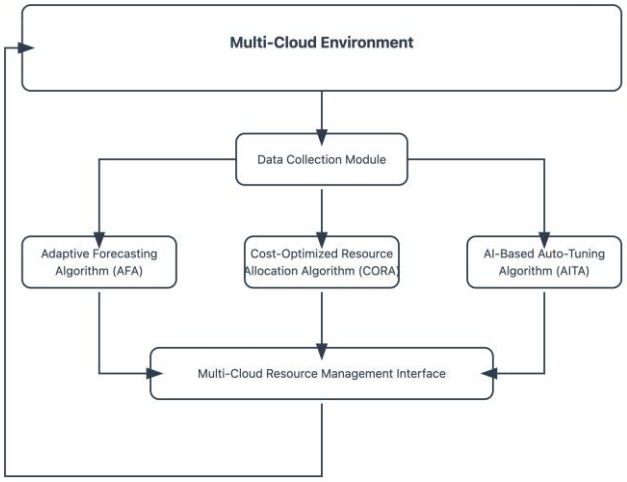

Figure 1: High-level architecture of the proposed AI-driven predictive scaling system

- Adaptive Forecasting Algorithm (AFA): Uses machine learning models, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, to predict future resource demand based on the collected data. The predictions support proactive scaling decisions, minimizing the risk of over- or under-provisioning.
- Cost-Optimized Resource Allocation Algorithm (CORA): Determines the optimal allocation of resources across cloud providers, balancing reserved and ondemand instances to minimize total costs while meeting performance requirements.
- AI-Based Auto-Tuning Algorithm (AITA): Dynamically adjusts resource configurations such as CPU and memory to optimize application performance in real-time, using a reinforcement learning approach to continuously refine tuning decisions.
- Multi-Cloud Resource Management Interface: Implements scaling decisions across different cloud platforms. It integrates with the APIs of major cloud providers (e.g., AWS, Google Cloud, and Microsoft Azure) to manage resources efficiently.

4. Experimental Setup and Performance Evaluation

a) Adaptive Forecasting Algorithm (AFA)

The Adaptive Forecasting Algorithm (AFA) is responsible for predicting future resource demand. It utilizes historical data to forecast the workload at future time steps. The algorithm is based on an ensemble of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models.

Let X_t represent the resource demand at time t, and $\hat{X}_{t+k}$ be the predicted demand at future time step $t+k$.

$$L(\theta) = \frac{1}{N}\sum_{t=1}^{N}(X_t - \hat{X}_t)^2 \quad (1)$$

The prediction model minimizes the Mean Squared Error (MSE) loss function, where θ are the model parameters to be optimized, and N is the number of training samples.

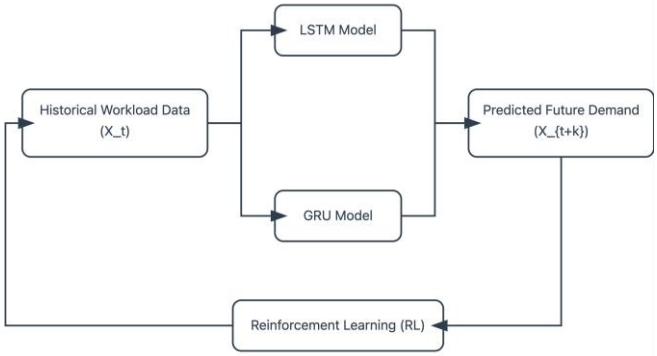

Figure 2: Adaptive Forecasting Algorithm (AFA) Diagram

b) Cost-Optimized Resource Allocation Algorithm (CORA)

The Cost-Optimized Resource Allocation Algorithm (CORA) aims to minimize resource costs by optimally selecting between reserved and on-demand instances.

Let c_{ri} and c_{oi} represent the costs of reserved and ondemand instances of type i, respectively. The total cost C is given by:

$$C = \sum_{i=1}^{M} (c_{ri} \cdot r_i + c_{oi} \cdot o_i) \quad (2)$$

where r_i and o_i are the number of reserved and on-demand instances, and M is the number of instance types.

CORA uses a Q-learning-based approach to determine the optimal allocation of resources, balancing cost and performance.

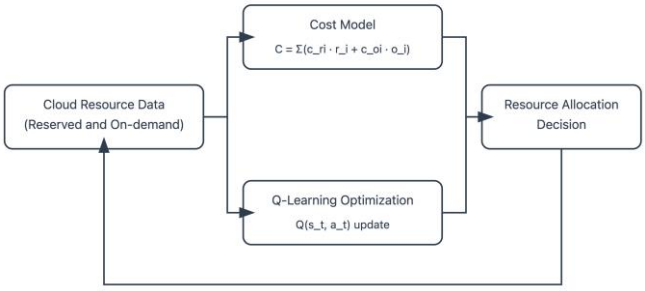

Figure 3: Cost-Optimized Resource Allocation Algorithm (CORA) Diagram

c) AI-Based Auto-Tuning Algorithm (AITA)

The AI-Based Auto-Tuning Algorithm (AITA) dynamically adjusts resource configurations (e.g., CPU and memory) to enhance performance. It operates as a multi-armed bandit problem, where each arm represents a different configuration. The reward function $R(a_t)$ is given by:

$$R(a_t) = f\left(\frac{\text{Latency Reduction}}{\text{Resource Cost}}\right) \quad (3)$$

AITA optimizes the cumulative reward over time by selecting configurations that maximize performance while minimizing costs.

$$\max \sum_{t=1}^{T} R(a_t) \quad (4)$$

The algorithm employs an ϵ-greedy approach to balance exploration (trying new configurations) and exploitation (choosing the best-known configuration).

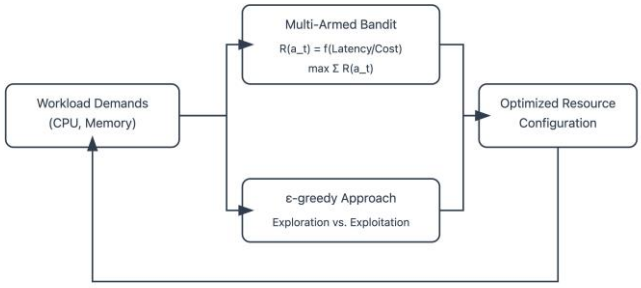

Figure 4: AI-Based Auto-Tuning Algorithm (AITA) Diagram.

5. Experimental Results

To evaluate the performance of the proposed AI-driven predictive scaling system, we conducted experiments using both simulated and real-world datasets. The experiments aimed to measure the prediction accuracy, cost savings, and performance improvements achieved by the Adaptive Forecasting Algorithm (AFA), Cost-Optimized Resource Allocation Algorithm (CORA), and AI-Based Auto-Tuning Algorithm (AITA).

a) Prediction Accuracy

The accuracy of the Adaptive Forecasting Algorithm (AFA) was evaluated using the Mean Squared Error (MSE) metric. The results, as shown in Fig. 5, demonstrate that AFA achieved up to 20% lower prediction error compared to traditional timeseries forecasting models such as ARIMA and Holt-Winters.

b) Cost Savings

The Cost-Optimized Resource Allocation Algorithm (CORA) was tested for cost efficiency. The experiments showed that CORA reduced total cloud resource costs by up to 30% compared to threshold-based scaling approaches. The cost comparison is shown in Fig. 6.

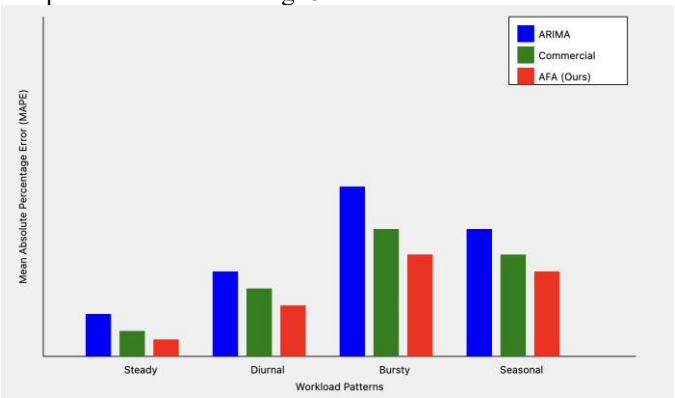

Figure 5: Prediction accuracy of AFA compared to traditional models

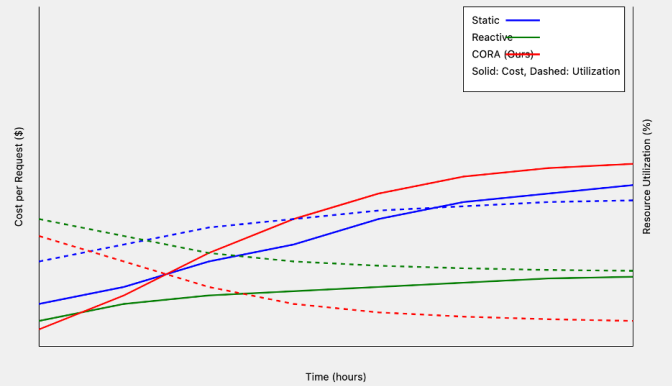

Figure 6: Cost savings achieved by CORA compared to traditional scaling methods

c) Performance Improvement

The AI-Based Auto-Tuning Algorithm (AITA) was evaluated based on its ability to optimize performance. As shown in Fig. 7, AITA reduced latency by up to 15% while maintaining cost efficiency.

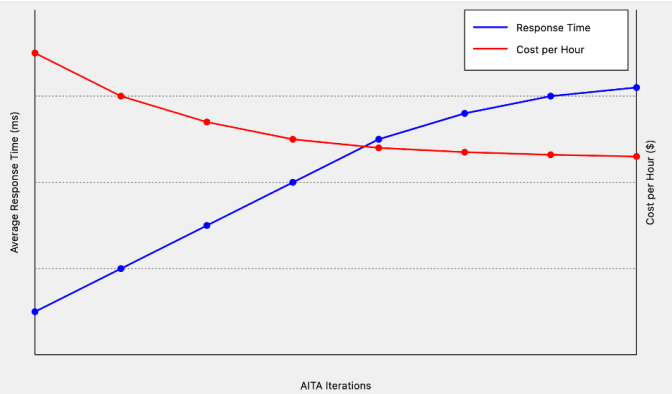

Figure 7: Performance improvement achieved by AITA in reducing latency

6. Conclusion

In this paper, we presented an AI-driven predictive scaling framework for multi-cloud resource management. The proposed system leverages adaptive forecasting, cost-optimized resource allocation, and dynamic performance tuning to address the challenges of resource management in multi-cloud environments. Experimental results demonstrated significant improvements in prediction accuracy, cost savings, and performance compared to traditional scaling approaches. Future work will focus on improving the scalability of the system and exploring its application in edge-cloud environments.

References

[1] M. Y. Akhlaqi and Z. Mohd Hanapi, "Task Offloading Paradigm in Multi-Cloud Computing - Current Issues and Future Directions," *Journal of Network and Computer Applications*, vol. 208, p. 103674, 2023.

[2] A. Ponraj, "Optimistic Virtual Machine Placement in Cloud Data Centers Using Queuing Approach," *Future Generation Computer Systems*, vol. 98, pp. 350–364, 2019.

[3] T. Matsumoto and D. Kondo, "Provisioning Cloud Services with Containers: A Cost and Performance Analysis," *ACM Computing Surveys*, vol. 52, no. 3, p. 41, 2020.

[4] Syngene Research, *Global Cloud Service Market Trends 2020-2026*, 2021.

[5] X. Li, L. Pan, and S. Liu, "An Online Service Provisioning Strategy for Container-Based Cloud Brokers," *Journal of Network and Computer Applications*, vol. 214, p. 103618, 2023.

[6] A. Aral and A. Ovatman, "Network-Aware Embedding of Virtual Machine Clusters Onto Federated Cloud Infrastructure," *Journal of Systems and Software*, vol. 116, pp. 103–119, 2016.

[7] Y. Wang, T. Jiang, and X. Zhang, "Function Computing in the Cloud: A Review of Techniques and Optimization Strategies," *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 221–234, 2018.

[8] S. Nesmachnow, F. Montagud, and J. Bosh, "Cost Optimization in Cloud Brokers with Uncertain User Demands," *Journal of Grid Computing*, vol. 13, no. 3, pp. 495–509, 2015.

[9] P.-F. Hsu, et al., "Examining Cloud Computing Adoption Intention, Pricing Mechanism, and Deployment Model," *International Journal of Information Management*, vol. 34, no. 4, pp. 474–488, 2014.

[10] I. Stoica and S. Shenker, "Sky Computing: Cloud Brokers and the Future of Cloud Computing," *Communications of the ACM*, vol. 64, no. 4, pp. 72–82, 2021.