ok, 310325

Master Thesis Expose

Predictive Auto-Scaling for Cloud Applications

Chaithra Jagannatha Rao Telkar

Matriculation ID: 11037491

Official Start date: 01.04.2025

SRH Hochschule Heidelberg

Internal Supervisor 1: Gerd Moeckel

Internal Supervisor 2: Paul Tanzer

1 Introduction

1.1 Background

Cloud computing has become a critical component of modern IT infrastructure, enabling organizations to deploy applications and services with high scalability, flexibility, and cost-effectiveness. The ability to dynamically allocate resources based on demand has significantly improved operational efficiency and performance. However, traditional auto-scaling mechanisms primarily rely on reactive approaches, adjusting resources only after demand fluctuations occur. These reactive methods often lead to inefficiencies, such as delayed scaling responses during traffic surges or unnecessary over-provisioning during low-usage periods, leading to increased operational costs.

Predictive auto-scaling offers a more advanced solution by leveraging machine learning (ML) and statistical models to forecast workload variations. Instead of reacting to changes, predictive models anticipate demand shifts based on historical patterns, enabling proactive resource allocation. This approach improves performance, reduces latency in scaling actions, and optimizes cost efficiency.

1.2 Motivation

With the increasing adoption of cloud-native applications, real-time performance and reliability have become critical concerns. Traditional scaling strategies that rely on static threshold values (e.g., CPU utilization reaching 80%) are often insufficient in dynamically fluctuating environments. A proactive approach, where resources are pre-allocated based on demand predictions, can significantly enhance system efficiency and user experience. The integration of AI-driven predictive models in auto-scaling frameworks addresses these challenges by ensuring resource availability in advance, thus preventing performance degradation and reducing cloud infrastructure costs.

2 Problem Statement

Despite the advancements in cloud auto-scaling, most existing techniques fail to adapt to sudden workload fluctuations efficiently. Current reactive models introduce latency, causing either performance bottlenecks during peak loads or excessive idle resources during underutilized periods. The primary challenges include:

- Latency in Scaling: Delayed response to workload changes results in degraded user experience.
- Inefficient Resource Utilization: Over-provisioning leads to unnecessary costs, while under-provisioning results in service degradation.
- Lack of Adaptive Forecasting: Traditional models fail to account for complex and nonlinear workload patterns.

This proposal aims to address these limitations by developing an intelligent, predictive auto-scaling framework that dynamically provisions cloud resources based on historical workload patterns and real-time telemetry data.

3 Research Methodology

The methodology for predictive auto-scaling in cloud applications focuses on leveraging historical workload data and machine learning techniques to optimize cloud resource allocation. The goal is to develop models that can predict demand variations and scale resources proactively, ensuring high availability and cost efficiency. The research is based on publicly available datasets from Kaggle (borg_traces_data.csv and vmCloud_data.csv), which provide real-world cloud workload traces.

Data Collection and Preprocessing

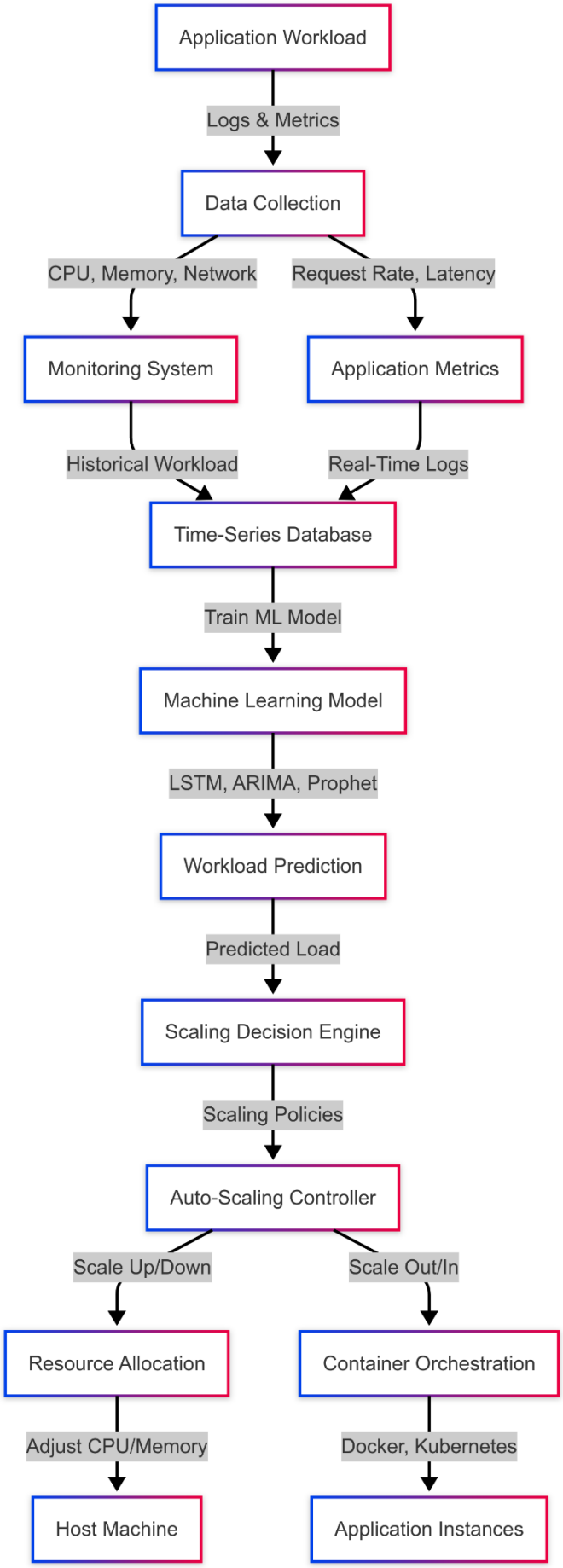
The first step involves gathering relevant cloud workload data. The borg_traces_data.csv dataset includes CPU and memory usage, scheduling events, and priority levels. The vmCloud_data.csv dataset contains metrics related to virtual machine performance, including CPU usage, memory consumption, network traffic, and power utilization. To ensure data consistency, preprocessing steps include handling missing values by applying mean, median, or interpolation techniques, normalizing numerical values using Min-Max scaling, encoding categorical variables with one-hot encoding, and detecting anomalies using statistical methods such as Z-score analysis. These preprocessing steps ensure that the dataset is clean, structured, and ready for further analysis.

Feature Engineering

Feature engineering is a critical step in predictive modeling as it helps extract meaningful insights from the dataset and enhances model performance. The study identifies and constructs various features, including time-series trends, lag features, network traffic correlations, anomaly detection indicators, and scaling events. Time-series decomposition is used to separate workload patterns into seasonal, trend, and residual components. Lag features incorporate previous workload data points to help predict future resource demands. Correlations between network traffic and system resource usage are analyzed to determine their impact on workload fluctuations. Anomaly detection techniques such as Isolation Forest and DBSCAN are employed to identify and mitigate outlier workload patterns that could affect model accuracy. By extracting and engineering these features, the dataset becomes more informative and suitable for training predictive models.

Model Selection and Training

To develop a robust predictive auto-scaling framework, this study utilizes **LSTM (Long Short-Term Memory Networks)** and **Random Forest Regression**. LSTM is a deep learning model well-suited for time-series forecasting, as it captures long-term dependencies in sequential data. The LSTM model processes historical workload traces, learning temporal relationships between CPU/memory usage and network traffic fluctuations. The model is trained using backpropagation through time (BPTT), and hyperparameters such as the number of LSTM units, dropout rates, and learning rates are optimized using Grid Search. Additionally, Random Forest Regression, an ensemble learning model, is employed to predict workload demand by analyzing multiple decision trees that evaluate different aspects of the dataset. This model is highly effective for structured workload data, as it reduces variance and prevents overfitting. Random Forest Regression is trained using historical workload features, and hyperparameter tuning techniques such as cross-validation are applied to improve its predictive accuracy. The dataset is split into an 80-20 ratio, where 80% of the data is used for training and 20% for testing. Both models are evaluated based on their ability to predict future workload variations accurately, and their performance is compared against traditional auto-scaling methods.


Application Workload
Logs & Metrics
Data Collection
CPU, Memory, Network
Request Rate, Latency
Monitoring System
Application Metrics
Historical Workload
Real-Time Logs
Time-Series Database
Train ML Model
Machine Learning Model
LSTM, ARIMA, Prophet
Workload Prediction
Predicted Load
Scaling Decision Engine
Scaling Policies
Auto-Scaling Controller
Scale Up/Down
Scale Out/In
Resource Allocation
Container Orchestration
Adjust CPU/Memory
Docker, Kubernetes
Host Machine
Application Instances

4 Expected Outcome

Predictive auto-scaling can transform cloud resource management by enabling:

- Enhanced Performance: Reduced response times and improved service reliability.
- Cost Efficiency: Reduction in cloud infrastructure expenses through optimized resource allocation.
- Scalability & Flexibility: Ensuring system adaptability to varying workloads with minimal human intervention.
- Proactive Fault Tolerance: Anticipating and mitigating potential resource shortages before they impact service quality.

The outcomes of this study will contribute to the development of more intelligent cloud management systems, paving the way for automated, AI-driven scaling solutions in enterprise cloud applications.

5 Result Evaluation

To assess the performance of the predictive models, various evaluation metrics are employed. Mean Absolute Error (MAE) is used to measure the average absolute differences between predicted and actual workload values. Root Mean Squared Error (RMSE) quantifies prediction accuracy by penalizing larger errors more heavily, ensuring precise forecasts. Mean Squared Error (MSE) evaluates prediction variance, while R-Squared (R^2) determines how well the model explains fluctuations in workload demand. These evaluation metrics provide a comprehensive assessment of model performance, allowing for a detailed comparison between predictive auto-scaling and traditional threshold-based scaling approaches. The results are visualized using graphical techniques such as scatter plots and trend analysis to illustrate the predictive capabilities of the models.

6 References

1. Rathinam, Anantha & Vathani, B. & Komathi, A. & Lenin, J. & Bharathi, B. & Murugan, S.. (2023). Advances and Predictions in Predictive Auto-Scaling and Maintenance Algorithms for Cloud Computing. 395-400. 10.1109/ICACRS58579.2023.10404186.
2. Alharthi S, Alshamsi A, Alseiari A, Alwarafy A. Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions. *Sensors*. 2024; 24(17):5551. https://doi.org/10.3390/s24175551
3. Yunda Guo, Jiake Ge, Panfeng Guo, Yunpeng Chai, Tao Li, Mengnan Shi, Yang Tu, and Jian Ouyang. 2024. PASS: Predictive Auto-Scaling System for Large-scale Enterprise Web Applications. In Proceedings of the ACM Web Conference 2024 (WWW '24). Association for Computing Machinery, New York, NY, USA, 2747–2758. https://doi.org/10.1145/3589334.3645330
4. Radhika, E. G., & Sudha Sadasivam, G. (2021). A review on prediction based autoscaling techniques for heterogeneous applications in cloud environment. *Materials Today: Proceedings, 45*, 2793-2800. https://doi.org/10.1016/j.matpr.2020.11.789
5. Xue, S., Qu, C., Shi, X., Liao, C., Zhu, S., Tan, X., ... & Zhang, J. (2022, August). A meta reinforcement learning approach for predictive autoscaling in the cloud. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 4290-4299).
6. Biswas, A., Majumdar, S., Nandy, B., & El-Haraki, A. (2015). Predictive Auto-scaling Techniques for Clouds Subjected to Requests with Service Level Agreements. *World Congress on Services*, 311–318. https://doi.org/10.1109/SERVICES.2015.54

7. Ivanovic, M., & Simic, V. (2022). Efficient evolutionary optimization using predictive auto-scaling in containerized environment. *Applied Soft Computing, 129*, 109610. https://doi.org/10.1016/j.asoc.2022.109610
8. Ivanovic, M., & Simic, V. (2022). Efficient evolutionary optimization using predictive auto-scaling in containerized environment. *Applied Soft Computing, 129*, 109610. https://doi.org/10.1016/j.asoc.2022.109610
9. Lanciano, G., Galli, F., Cucinotta, T., Bacciu, D., & Passarella, A. (2021, December). Predictive auto-scaling with OpenStack Monasca. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing* (pp. 1-10).
10. Flunkert, Valentin & Rebjock, Quentin & Castellon, Joel & Callot, Laurent & Januschowski, Tim. (2020). A simple and effective predictive resource scaling heuristic for large-scale cloud applications. 10.48550/arXiv.2008.01215.
11. Giannakopoulos, Ioannis & Papailiou, Nikolaos & Mantas, Christos & Konstantinou, Ioannis & Tsoumakos, Dimitrios & Koziris, Nectarios. (2015). CELAR: Automated application elasticity platform. Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014. 23-25. 10.1109/BigData.2014.7004481.