Log-Driven Autonomic Auto-Scaling with LSTM Forecasting: An Industrial Case in On-Premises Containerized Systems

Rafael Xavier
Dep. of Technology Information,
Federal Center of Minas Gerais
Belo Horizonte, Brazil
rafael.xavier@cefetmg.br

Rafael Durelli
Department of Computer Science,
Federal University of Lavras
Lavras, Brazil
rafael.durelli@ufla.br

Bruno Cafeo
Institute of Computing,
University of Campinas
Campinas, Brazil
cafeo@unicamp.br

Elder Cirilo
Department of Computer Science,
Federal University of São João del-Rei
São João del-Rei, Brazil
elder@ufsj.edu.br

ABSTRACT

This work presents an autonomic auto-scaling solution for a containerized system deployed on on-premises infrastructure. The solution addresses the lack of native autoscaling — the automatic adjustment of computing resources to match workload demand — in private environments by implementing a self-managing architecture based on the MAPE-K (Monitor-Analyze-Plan-Execute) control loop, an established methodology where the system continuously monitors itself, analyzes changes, plans reactions, and executes them using existing knowledge. It incorporates a Long Short-Term Memory (LSTM) neural network to proactively forecast workload spikes, enabling informed scaling decisions before performance degradation occurs. A key innovation is the non-intrusive, log-driven approach, where system logs serve as the primary knowledge source for analysis and scaling decisions. Preliminary evaluation using a dataset collected from a production deployment at a Brazilian public institution reveals improved system responsiveness and resource utilization during peak demand. These results, derived from a real-world, industrial-scale scenario, demonstrate the practical applicability of the proposed solution and indicate that the LSTM-driven auto-scaler can maintain quality of service under variable workloads while optimizing resource usage.

KEYWORDS

Auto-scaling, Autonomic Computing, Time Series Forecasting, On-Premises Containerized Systems

1 Target Audience

This work primarily targets *DevOps engineers* and *infrastructure managers* responsible for maintaining on-premises software systems, particularly those evaluating cloud repatriation strategies or seeking to implement advanced resource optimization techniques without complete cloud migration. *Software engineers* working on system modernization can also benefit from understanding how machine learning techniques can be applied to infrastructure management problems. Additionally, *researchers in software engineering* with industry connections might find significance in the practical application of theoretical concepts such as autonomic computing and deep learning for time series forecasting in industrial-grade environments.

2 Presentation Content

The cloud computing paradigm has revolutionized how organizations deploy and manage their IT infrastructure by offering scalability, flexibility, and operational efficiency [2]. Despite the overall adoption of cloud computing, organizations are migrating their software systems and resources back to on-premises environments [11]. This movement, primarily driven by concerns related to cost control, operational flexibility, and data sovereignty requirements, is known as cloud repatriation [10]. During cloud repatriation, organizations strive to replicate a critical capability in on-premise environments: auto-scaling [1] (the ability to adjust computational resources in response to varying workloads dynamically). While cloud platforms offer built-in auto-scaling services (e.g., AWS Autoscaling[1]), implementing equivalent capabilities in on-premises environments might be challenging or prohibitive in some cases [1]. Therefore, traditional auto-scaling solutions rely on scaling strategies triggered manually or by predefined thresholds (CPU utilization or request rates) [3]. However, these reactive approaches often suffer from significant limitations [5]. For instance, they cannot anticipate sudden workload variations, resulting in periods of resource contention (affecting user experience) or inefficient over-provisioning (increasing costs). In our study at a Brazilian public institution, these challenges manifested as an increased administrative overhead for manual scaling operations to avoid resource wastage during off-peak periods and occasional service degradation during peak usage.

In this work, we present and discuss the implementation and evaluation of a non-intrusive, log-driven autoscaler that forecasts future workload using Long Short-Term Memory (LSTM) neural networks [6]. Instead of modifying the existing software system or incorporating new agentic properties, an autonomous autoscaler continuously mines existing logs to predict short-term future demand. These predictions feed into a MAPE-K [9] control loop that adjusts container instances proactively. Consequently, organizations can benefit from predictive resource management typically

[1] https://aws.amazon.com/autoscaling

associated with cloud platforms without invasive changes or additional complexity.

2.1 Proposed Solution

Our solution implements a MAPE-K-based autonomic auto-scaling architecture with four interconnected modules. The **Monitor** module continuously collects service quality metrics, such as the average HTTP response time, which is computed by processing access logs. These metrics are timestamped and stored in a central Knowledge Base. This log-driven approach requires no changes to the software system, only access to its existing logs. The *Analyze* module takes the workload time series from the Knowledge Base and runs the LSTM prediction model. In practice, the module maintains a sliding window of the most recent 48 observations (each 10 minutes apart) and normalizes them. This LSTM-based predictor outperformed simpler statistical methods (e.g., ARIMA – AutoRegressive Integrated Moving Average [4] and ETS – Exponential Smoothing State Space [8]) in our study, giving accurate multi-step forecasts (around 20% mean absolute error). In the **Plan** module, a planner service decides how many instances should run to meet the predicted load within service-level objectives. It reads the current resource counts (i.e., the number of active container instances), the average response time, and the forecasted request rate. The planner computes the new target number of replicas using a simple policy table (pre-calibrated by a synthetic load generator). The **Execute** module, in turn, enforces the scaling decisions. Whether more instances are needed, the executor calls Docker Swarm commands to scale out container replicas. Conversely, excess containers are terminated when the load subsides.

2.2 Evaluation

Experiments were conducted on a system equipped with an Intel Core i7-1165G7 processor, 20 GB of RAM, and integrated Intel Iris Xe graphics. The evaluation encompassed four distinct scenarios, each driven by authentic HTTP traffic traces to emulate industrial-grade conditions. Key metrics [7] collected included RMSE (Root Mean Square Error), MSE (Mean Squared Error), MAE (Mean Absolute Error) MAPE (Mean Absolute Percentage Error), peak-demand prediction accuracy, and resource usage. Compared to a statically provisioned baseline, the autonomic auto-scaler reduced aggregate resource usage by an average of 37.9%, absorbing sudden spikes with minimal overprovisioning. The LSTM predictor anticipated 62.5% of peak-demand events in live-operation tests, while SLA violations remained very low—on par with or better than the baseline. Finally, the forecasting module achieved an RMSE of approximately 1,041 requests and a MAPE of 19.8% on actual traffic, demonstrating that a log-driven LSTM auto-scaler can substantially improve efficiency without compromising service quality.

3 Industry Applications and Lessons Learned

Implementing our proactive, non-intrusive auto-scaling system for on-premise container services yielded four practical insights. First, workload monitoring and prediction without modifying the deployed software is essential to minimize adoption risk. Second, predictive auto-scaling proves effective even with MAPE around 20%: our LSTM forecaster correctly anticipated 62.5% of demand peaks, reducing over-provisioning without affecting user experience. Third, a modular MAPE-K architecture simplifies development and maintenance by keeping the Monitoring, Analysis, Planning, and Execution modules decoupled via a lightweight knowledge base. Finally, the design is platform-agnostic – though validated on Docker Swarm, it seamlessly extends to Kubernetes [12] or other container orchestrators. Our findings also suggest that log-driven LSTM auto-scaling can (1) deliver cloud-like elasticity on-premise without complete system redesign, (2) translate a 37.9% resource reduction into significant cost savings, (3) improve operational stability by mitigating SLA violations under sudden peak usage, and (4) bridge academic research and industrial practice.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from FAEPEX (grants No. 3404/23 and No. 2382/24) and from CAPES for its support of the Graduate Program in Computer Science at UFSJ.

REFERENCES

[1] Saleha Alharthi, Afra Alshamsi, Anoud Alseiari, and Abdulmalik Alwarafy. 2024. Auto-Scaling Techniques in Cloud Computing: Issues and Research Directions. *Sensors* 24, 17 (2024), 5551.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A View of Cloud Computing. *Commun. ACM* 53, 4 (2010), 50–58.

[3] Dariusz Rafal Augustyn. 2017. Improvements of the Reactive Auto Scaling Method for Cloud Platform. In *Computer Networks*, Piotr Gaj, Andrzej Kwiecień, and Michał Sawicki (Eds.). Springer International Publishing, Cham, 422–431.

[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

[5] Javad Dogani, Reza Namvar, and Farshad Khunjush. 2023. Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. *Computer Communications* 209 (2023), 120–150.

[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[7] Rob J Hyndman and Anne B Koehler. 2006. Another look at measures of forecast accuracy. *International journal of forecasting* 22, 4 (2006), 679–688.

[8] Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. 2002. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of forecasting* 18, 3 (2002), 439–454.

[9] IBM Corporation. 2006. *An Architectural Blueprint for Autonomic Computing* (4 ed.). Technical Report. IBM Corporation. https://www.research.ibm.com/autonomic/manifesto/

[10] Kiran Jewargi. 2023. Public Cloud to Cloud Repatriation Trend. *Scholars Journal of Engineering and Technology* 1 (2023), 1–3.

[11] Natalya Yezhkova. 2024. *Assessing the Scale of Workload Repatriation: Insights from IDC's Server and Storage Workloads Surveys, 1H23 and 2H23*. Technical Report US50903124. International Data Corporation (IDC). https://www.idc.com/getdoc.jsp?containerId=US50903124 IDC Survey Report.

[12] Zhiqiang Zhou, Chaoli Zhang, Lingna Ma, Jing Gu, Huajie Qian, Qingsong Wen, Liang Sun, Peng Li, and Zhimin Tang. 2023. AHPA: adaptive horizontal pod autoscaling systems on alibaba cloud container service for kubernetes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 15621–15629.