Comparison of Microservice Call Rate Predictions for Replication in the Cloud

Narges Mehran*, Arman Haghighi†, Pedram Aminharati*, Nikolay Nikolov‡
Ahmet Soylu§, Dumitru Roman‡§, Radu Prodan*
*Alpen-Adria-Universität Klagenfurt, Austria
†Azad University, Science and Research Branch, Tehran, Iran
‡SINTEF AS, Oslo, Norway
§OsloMet - Oslo Metropolitan University, Oslo, Norway

***Abstract*—Today, many users deploy their microservice-based applications with various interconnections on a cluster of Cloud machines, subject to stochastic changes due to dynamic user requirements. To address this problem, we compare three machine learning (ML) models for predicting the microservice call rates based on the microservice times and aiming at estimating the scalability requirements. We apply the linear regression (LR), multilayer perception (MLP), and gradient boosting regression (GBR) models on the Alibaba microservice traces. The prediction results reveal that the LR model reaches a lower training time than the GBR and MLP models. However, the GBR reduces the mean absolute error and the mean absolute percentage error compared to LR and MLP models. Moreover, the prediction results show that the required number of replicas for each microservice by the gradient boosting model is close to the actual test data without any prediction.**

***Index Terms*—Cloud computing, microservice, replication, linear regression, multilayer perceptron, gradient boosting.**

I. Introduction

The recent shift towards the increasing number of microservice-based applications in the Cloud-native infrastructure brings new scheduling, deployment, and orchestration challenges [1], such as scaling out overloaded microservices in response to increasing load.

1) Research problem: In our previous work [2], we explored microservice scheduling on provisioned resources. However, we did not inspect the scalability requirements of the containerized microservices by prediction models considering different request arrival rates from end-users acting as producers [3]. Traditional microservice scaling methods [4], [5] focus on the resource or application processing metrics without predicting the stochastic changes in user requirements, such as dynamic request rates.

2) Example: Table I presents an example involving three producers calling three microservices deployed on three resources. In this scenario, the microservices experience varying *call rates* initiated by the *producers*. Every producer request leads to interactions with its corresponding microservice on the specific resource within a specific time. Typically, microservices with higher execution times necessitate horizontal scalability to accommodate the call rate. In other words, a direct correlation exists between the microservice time and call rate, motivating the need to explore prediction models addressing their horizontal scaling [6]. Table I shows that during a $2\,\mathrm{s}$ execution, the microservices m_0, m_1, and m_2 receive the following number of calls:

TABLE I: Motivational example.

Microservice	*Resource*	*Microservice time* (s/call)	*Call rate* (calls/s)
m_0	r_0	0.7	2
m_1	r_1	1.5	2
m_2	r_2	2	3

$$\begin{aligned} m_0: &\quad 2\,\mathrm{s}\cdot 2\,\mathrm{calls/s} = 4\,\mathrm{calls};\\ m_1: &\quad 2\,\mathrm{s}\cdot 2\,\mathrm{calls/s} = 4\,\mathrm{calls};\\ m_2: &\quad 2\,\mathrm{s}\cdot 3\,\mathrm{calls/s} = 6\,\mathrm{calls}. \end{aligned}$$

However, at the end of the $2\,\mathrm{s}$ interval, the microservices m_0, m_1, and m_2 still respond to their third, second, and first calls. To reduce the bottleneck on the Cloud infrastructure [7], we need to scale the microservices based on the multiplication function between the correlated microservice time and call rate up to the following number of replicas:

$$\begin{aligned} m_0 \text{ on } r_0: &\quad 2\,\mathrm{calls/s}\cdot 0.7\,\mathrm{s/call} = 1.4 \approx 2;\\ m_1 \text{ on } r_1: &\quad 2\,\mathrm{calls/s}\cdot 1.5\,\mathrm{s/call} = 3;\\ m_2 \text{ on } r_2: &\quad 3\,\mathrm{calls/s}\cdot 2\,\mathrm{s/call} = 6. \end{aligned}$$

3) Method: We address the scalability problem through *microservices call rate predictions* employing ML models involving two features:

- *Microservice time* defining the processing time of each containerized microservice on the Cloud virtual machine;
- *Microservice call rate* defining the number of calls/requests invoking a microservice.

We apply ML models to predict microservices call rate based on the microservice time and estimate the number of microservice replicas to support stochastic changes due to the dynamic user requirements. Recently, there has been a growing interest in the applicability of deep learning models to tabular data [8], [9]. However, tree-based machine learning (ML) models such as bagging (e.g., RandomForest) or boosting (e.g., XGBoost [10], gradient boosting tree, and gradient boosting regression) are among the popular learners for tabular data that outperform deep learning methods [11]. Nevertheless, related work did not explore and evaluate the *gradient boosting*

regression (GBR) and *multilayer perceptron (MLP)* learning methods for microservice call rate prediction. Therefore, we apply and compare the GBR, neural network-based MLP, and traditional linear regression (LR) models to estimate the number of replicas for each microservice.

4) Contributions: A comparative evaluation of the ML models on trace data collected from a real-word Alibaba Cloud cluster [12] indicates that the GBR reaches a balance between the prediction errors, including the mean absolute error (MAE) and mean absolute percentage error (MAPE), and the training time compared to the MLP and LR methods.

REFERENCES

[1] Christina Terese Joseph and K Chandrasekaran. Intma: Dynamic interaction-aware resource allocation for containerized microservices in cloud environments. *Journal of Systems Architecture*, 111:101785, 2020.

[2] Narges Mehran, Zahra Najafabadi Samani, Dragi Kimovski, and Radu Prodan. Matching-based scheduling of asynchronous data processing workflows on the computing continuum. In *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 58–70, 2022.

[3] Shutian Luo, Huanle Xu, Chengzhi Lu, Kejiang Ye, Guoyao Xu, Liping Zhang, Yu Ding, Jian He, and Chengzhong Xu. Characterizing microservice dependency and performance: Alibaba trace analysis. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 412–426, 2021.

[4] Hamidreza Arkian, Guillaume Pierre, Johan Tordsson, and Erik Elmroth. Model-based stream processing auto-scaling in geo-distributed environments. In *ICCCN 2021-30th International Conference on Computer Communications and Networks*, 2021.

[5] Krzysztof Rzadca, Pawel Findeisen, Jacek Swiderski, Przemyslaw Zych, Przemyslaw Broniek, Jarek Kusmierek, Pawel Nowak, Beata Strack, Piotr Witusowski, Steven Hand, et al. Autopilot: workload autoscaling at google. In *Proceedings of the Fifteenth European Conference on Computer Systems*, pages 1–16, 2020.

[6] Angelina Horn, Hamid Mohammadi Fard, and Felix Wolf. Multi-objective hybrid autoscaling of microservices in kubernetes clusters. In *Euro-Par 2022: Parallel Processing: 28th International Conference on Parallel and Distributed Computing*, pages 233–250. Springer, 2022.

[7] Nikolay Nikolov, Yared Dejene Dessalk, Akif Quddus Khan, Ahmet Soylu, Mihhail Matskin, Amir H Payberah, and Dumitru Roman. Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers. *Internet of Things*, page 100440, 2021.

[8] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data, 2023.

[9] Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning, 2023.

[10] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[11] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022.

[12] Shutian Luo, Huanle Xu, Chengzhi Lu, Kejiang Ye, Guoyao Xu, Liping Zhang, Jian He, and Chengzhong Xu. An in-depth study of microservice call graph and runtime performance. *IEEE Transactions on Parallel and Distributed Systems*, 33(12):3901–3914, 2022.