Journal of Engineering Science and Technology
Vol. 19, No. 4 (2024) 72 - 79

SERVERLESS CLOUD COMPUTING DEPLOYMENT FOR PRE-TRAINED MACHINE LEARNING MODEL

GALURA MUHAMMAD SURANEGARA, VINA FUJIYANTI,
ADE GAFAR ABDULLAH, ENDAH SETYOWATI, DIKY ZAKARIA

Universitas Pendidikan Indonesia, Jl. Dr. Setiabudi no 229, Bandung 40154, Indonesia
*Corresponding Author: galurams@upi.edu

Abstract

This study examines the application of pre-trained machine learning models on serverless cloud computing platforms, specifically comparing three serverless services offered by Google Cloud Platform: Cloud Run, Cloud Functions, and Vertex AI. The research aims to evaluate and compare the performance and cost-effectiveness of these services for deploying machine learning models. The methodology involves using a pre-trained model, implementing it on each platform, and measuring key performance indicators such as CPU utilization, memory utilization, latency, and cost. Testing was conducted using Apache JMeter to simulate HTTP requests to the endpoints. The results show that all three services successfully implemented machine learning models with relatively low CPU and memory usage (less than 1% and 1.5%, respectively). However, Vertex AI exhibited much higher latency (17.32 ms) compared to Cloud Run (2.69 ms) and Cloud Functions (3.33 ms). In terms of cost, Vertex AI is significantly more expensive than the other two services. Thus, while all three services are capable of implementing pre-trained machine learning models effectively, each platform has distinct characteristics suited to different use cases. Cloud Run is ideal for containerized applications, Cloud Functions for simple tasks triggered by specific conditions, and Vertex AI for complex AI and machine learning workloads despite its higher latency and costs.

Keywords: Cloud computing, Machine learning, Pre-trained model, Serverless cloud.

1. Introduction

Machine learning (ML), a subset of Artificial Intelligence (AI), is increasingly utilized across various applications due to its reliability in identifying and analysing data patterns applicable to numerous fields. One significant challenge in leveraging ML models is the implementation process, which has evolved with advancements in technology. Initially, ML was typically implemented on local servers; however, this trend has shifted with the advent of cloud technology [1-3]. Cloud technology offers scalability, flexibility, and cost efficiency that are difficult to achieve with local server implementations. Currently, there are two cloud implementation models: server-based and serverless. The server-based model, similar to local server implementation, faces challenges related to configuration complexity and infrastructure management, given the high computing resources required by ML [4]. To address these challenges, the serverless model was developed.

Several studies indicate that the serverless model simplifies the complexity of server-based implementations [5, 6]. Additionally, the serverless model enhances efficiency during high-demand periods and reduces costs during low-demand periods [7, 8]. The implementation of serverless models for ML has been widely researched to explore their advantages and challenges. For instance, adopting a serverless model can reduce operational costs by up to 40% in uncertain usage scenarios [9]. It also simplifies application management by eliminating the need for server management, although challenges such as execution duration limitations and debugging complexity remain [10, 11]. Other studies have noted performance constraints in serverless models for tasks requiring intensive computing [12-14]. As the serverless model evolves, the number of services available for its implementation increases. Google Cloud Platform (GCP), one of the largest cloud platforms, offers three serverless services suitable for ML implementation: Cloud Run, Cloud Functions [15-17], and Vertex AI, a service focused on AI development and also classified as serverless.

With the growing number of services, users must carefully choose the appropriate serverless service for their ML implementations. Currently, there is no research comparing the best serverless services for ML implementation. This study aims to fill this research gap by comparing the serverless services offered by GCP for ML implementation. The contribution of this research is to provide insights into selecting the most suitable serverless service from GCP, as service selection often depends on specific project needs, the scale of operation, and overall system architecture considerations.

2. Methods

This study does not require a high-specification computer since the computing is done on the cloud. The primary requirement for the computer used is a browser and a good Internet connection to access Google Cloud Platform (GCP). Apache JMeter (v5.6.3) is used to test the request endpoints of the machine learning models implemented on each serverless service on GCP. For implementing machine learning models, API files are needed to handle HTTP requests and responses, along with dependencies such as Flask (v2.3.2), Gunicorn (v21.0.0), scikit-learn (v1.3.0), and functions-framework (v3.*). The pre-trained machine learning model developed locally using the Decision Tree algorithm in Scikit-learn v1.3.0, achieved a prediction accuracy rate of 98.41%. The model includes training code

and outputs in .joblib format. The API for handling HTTP requests and responses uses Python 3.11.

The implementation design diagram for all serverless services used in this study is shown in Fig. 1.

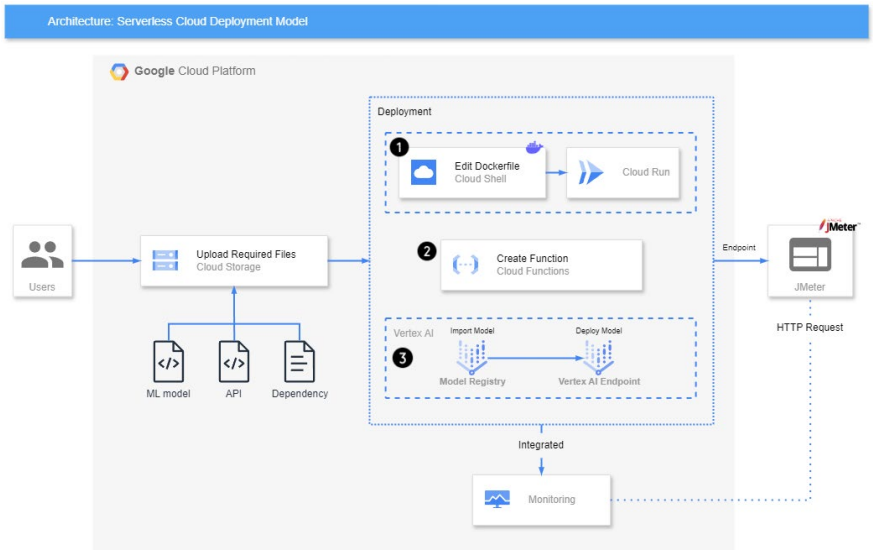

Fig. 1. Cloud run implementation architecture.

- **The Cloud Run:** Utilizes containers for deployment, requiring a Dockerfile but not deployment.yaml or service.yaml as in Kubernetes Engine. Once deployed, the application endpoint appears without needing to be exposed first.
- **Cloud Functions:** Differs slightly from Cloud Run by not using the Docker engine. It utilizes functions in the API file to handle requests and responses. Additional dependencies include functions-framework (v.3), with the API file named main.py. An endpoint is triggered by an HTTP request.
- **Vertex AI:** Utilizes the Model Registry to register and deploy the model using the Vertex AI endpoint. The model, API, and dependencies are input files, with the model file named appropriately according to the training framework format.

The resource specifications for each service are balanced to minimize bias during measurement, as shown in Table 1.

Table 1. Serverless services.

System Configuration	Serverless Services		
	Cloud Run	**Cloud Function**	**Vertex AI**
Instance Type	-	-	e2-standard-4
Number of vCPU	4	4	4
Memory	16 GB	16 GB	16 GB

The specifications of each serverless service are designed to ensure consistent performance measurements and an accurate assessment of service performance. Data collection involved testing the endpoints with JMeter for 10 sets, each making

10 requests over 30 minutes, totalling 100 requests. Performance parameters such as CPU utilization, memory utilization, and latency were monitored using Cloud Monitoring. CPU and memory utilization indicate service quality [18, 19], while cost calculations assess the economic efficiency of each service [20, 21].

3. Results and Discussion

This research has successfully implemented three serverless cloud services for machine learning. The machine learning model functions well on all serverless cloud services tested. For cost parameters, the exchange rate used when the calculation was carried out was IDR 16,209.99. To make it easier to understand, price data for the costs of using serverless cloud services is presented in rupiah. The measurement results for CPU and memory utilization parameters can be seen in Fig. 2.

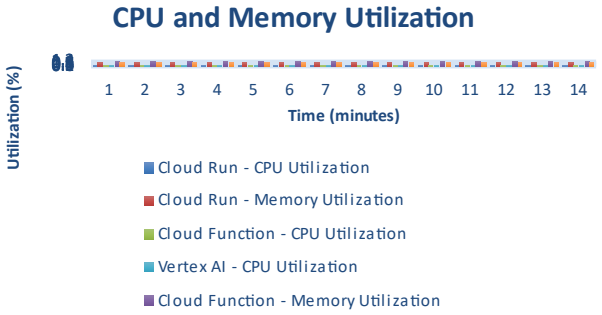

Fig. 2. Measurement results of CPU and memory utilization parameters.

From the measurement results illustrated in Fig. 2, the highest utilization reached 0.061%, while the lowest utilization was 0.044%. In Fig. 2 it can also be seen that there were fluctuating values during the test from the 4th to the 11th minute with a value range of 0.044% to 0.061%. However, these results can be said to be stable because the difference between the maximum and minimum values is 0.017% so it can be said to be insignificant. What causes fluctuations in the graph is caused by the use or switching of servers for internal tasks of the service infrastructure provider.

The measurement results for the Cloud Function service can also be seen in Fig. 2. The results for the Cloud Function also produce fluctuating values during testing over the time range used. However, the graph can be said to be stable because the difference between the maximum and minimum values is still very small and therefore not significant. Fluctuations occur for the same reason as the Cloud Function service, namely due to server switching in the internal tasks of the service infrastructure provider.

For the Vertex AI service, The measurement results can also be seen in Fig. 2. The CPU utilization and memory utilization parameters are very stable when compared to the other two services. However, for the latency parameter, the Vertex AI service has the worst score compared to the other two services. Complete latency measurement results can be seen in Fig. 3.

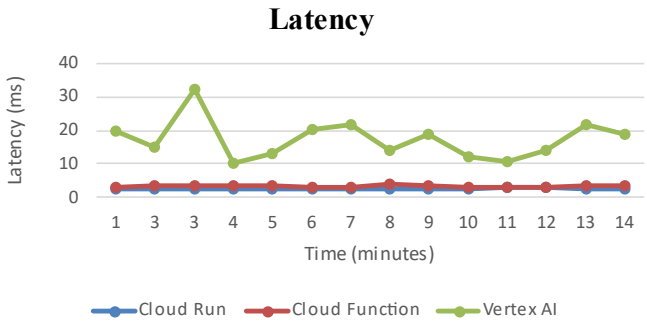

Fig. 3. Complete latency measurement results.

To see the big picture of the three services tested, the average results of the technical performance tests along with the prices charged by the service providers are presented in Table 2.

Table 2. Summary of Service Performance

Parameter	Services		
	Cloud Run	Cloud Functions	Vertex AI
CPU Utilization	0.05%	0.08%	0.10%
Memory Utilization	0.91%	1.37%	0.94%
Latency	2.69 ms	3.33 ms	17.32 ms
Pricing	Rp 6.493	Rp 6.493	Rp 2.559.270

From the average results of the measurements carried out, the result was that the CPU utilization parameters were all less than 1%. This shows that to serve the needs of the machine learning model implemented on this serverless system does not require high computing resources. This is because the machine learning model implemented is already a pre-trained model. This is in line with previous research which shows that pre-trained machine learning models can reduce the need for computing resources [22-24].

The memory utilization parameter is measured because of its crucial position for effective resource allocation [25]. Apart from that, these parameters are also useful for predicting load requirements which have a linear impact on predicting service costs [26]. The measurement results show that all memory utilization is still in the low category with insignificant differences. Therefore, the three serverless service models are reliable in terms of CPU and memory performance.

In terms of latency, Vertex AI recorded the worst time compared to the other two serverless services. This is because the processes that occur in Vertex AI require real-time AI processing so that the latency in Vertex AI is greater than in other serverless services [27]. Apart from that, if viewed from the infrastructure side, Vertex AI causes higher latency because Vertex AI abstracts some low-level details [28]. For price parameters, Vertex AI is far superior to both Cloud Run and Cloud Function. This is in accordance with one of the goals of developing Vertex AI.

This research certainly has limitations, firstly, the amount of sample data for latency parameters does not represent service performance because the measurement results only produce little data compared to other parameters. Second, it would be better if the time and number of test requests can be increased to produce more data samples. Third, this research only uses one monitoring tool, so there is no comparison of the measurement results, so it has the potential to cause errors during measurement.

4. Conclusion

The selection of cloud-based serverless services is crucial for machine learning applications that demand high computing resources. This research successfully implemented machine learning on three serverless cloud services, with minimal performance differences except for latency.

Each service has distinct characteristics: Vertex AI is ideal for AI and machine learning applications, Cloud Run is suited for containerized applications requiring complex or large data processing, and Cloud Functions is best for real-time, simple data processing and event-driven applications, offering integration with various cloud services and third parties.

Given the growing number of serverless cloud services capable of supporting machine learning, future research should broaden the range of services compared and include other platform providers.

Reference

1. Ali, A.; Zawad, S.; Aditya, P.; Akkus, I. E.; Chen, R.; and Yan, F. (2022). Smlt: A serverless framework for scalable and adaptive machine learning design and training. *arXiv preprint arXiv:2205.01853*.
2. Bisong, E.; (2019). *Google cloud machine learning engine (cloud mle).* In Bisong, E. (Ed.). *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners*. Apress.
3. Marculescu, D. (2021). When climate meets machine learning: Edge to cloud ML energy efficiency. *Proceedings of the* 2021 *IEEE/ACM International Symposium on Low Power Electronics and Design* (*ISLPED*), Boston, MA, USA, 1-1.
4. Li, J. (2022). Infrastructure-level design: Serverless and decentralized machine learning. In Guo, S.; and Zhou, O. (Eds.). *Machine learning on commodity tiny devices*. CRC Press, 83-98.
5. Lannurien, V.; D'orazio, L.; Barais, O.; and Boukhobza, J. (2023). *Serverless cloud computing: State of the art and challenges*. In Krishnamurthi, R.; Kumar, A.; Gill, S.S.; and Buyya, R. (Eds.). *Serverless computing: Principles and paradigms*. Springer International Publishing.
6. Mora, H.; Mora-Gimeno, F.J.; Jimeno-Morenilla, A.; Macia-Lillo, A.; and Elouali, A. (2022). Serverless computing at the edge for aiot applications. *Proceedings of the* 2022 *International Conference on Artificial Intelligence of Things* (*ICAIoT*)*,* Istanbul, Turkey, 1-6.
7. Bac, T. P.; Tran, M. N.; and Kim, Y. (2022). Serverless computing approach for deploying machine learning applications in edge layer. *Proceedings of the*

2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 396-401.

8. Mahmoudi, N.; and Khazaei, H. (2020). Performance modeling of serverless computing platforms. *IEEE Transactions on Cloud Computing*, 10(4), 2834-2847.
9. Bilal, M.; Canini, M.; Fonseca, R.; and Rodrigues, R. (2023). With great freedom comes great opportunity: Rethinking resource allocation for serverless functions. *Proceedings of the Eighteenth European Conference on Computer Systems,* New York, NY, USA, 381-397.
10. Li, Y.; Lin, Y.; Wang, Y.; Ye, K.; and Xu, C. (2022). Serverless computing: state-of-the-art, challenges and opportunities. *IEEE Transactions on Services Computing*, 16(2), 1522-1539.
11. Kumar, N.S.; and Samy, S.S. (2023). A survey and implementation on using a runtime overhead to enable serverless deployment. *Proceedings of the* 2023 9*th International Conference on Advanced Computing and Communication Systems (ICACCS),* Coimbatore, India, 497-501.
12. Assogba, K.; Arif, M.; Rafique, M.M.; and Nikolopoulos, D.S. (2022). On realizing efficient deep learning using serverless computing. *Proceedings of the* 2022 22nd *IEEE International Symposium on Cluster, Cloud and Internet Computing* (*CCGrid*)*,* Taormina, Italy, 220-229.
13. Nestorov, A.M.; Polo, J.; Misale, C.; Carrera, D.; and Youssef, A.S. (2021). Performance evaluation of data-centric workloads in serverless environments. *Proceedings of the* 2021 *IEEE* 14th *International Conference on Cloud Computing* (*CLOUD*)*,* Chicago, IL, USA, 491-496.
14. Tütüncüoğlu, F.; Jošilo, S.; and Dán, G. (2022). Online learning for rate-adaptive task offloading under latency constraints in serverless edge computing. *IEEE/ACM Transactions on Networking*, 31(2), 695-709.
15. Jambi, S. (2022). Serverless machine learning platform: A case for real-time crisis detection over social media. *Proceedings of the* 2022 *Second International Conference on Computer Science, Engineering and Applications (ICCSEA),* Gunupur, India, 1-6.
16. Paraskevoulakou, E.; and Kyriazis, D. (2023). ML-FaaS: Toward exploiting the serverless paradigm to facilitate machine learning functions as a service. *IEEE Transactions on Network and Service Management*, 20(3), 2110-2123.
17. Wu, H.; Deng, J.; Fan, H.; Ibrahim, S.; Wu, S.; and Jin, H. (2023). QoS-aware and cost-efficient dynamic resource allocation for serverless ML workflows. *Proceedings of the* 2023 *IEEE International Parallel and Distributed Processing Symposium (IPDPS),* St. Petersburg, FL, USA, 886-896.
18. Wang, Z.; Zhu, S.; Li, J.; Jiang, W.; Ramakrishnan, K.K.; Zheng, Y.; and Liu, A.X. (2022). Deepscaling: microservices autoscaling for stable CPU utilization in large scale cloud systems. *Proceedings of the* 13th *Symposium on Cloud Computing,* New York, United States, 16-30.
19. Zia Ullah, Q.; Hassan, S.; and Khan, G.M. (2017). Adaptive resource utilization prediction system for infrastructure as a service cloud. *Computational intelligence and Neuroscience*, 2017(1), 4873459.
20. Rodríguez, M.A.N.; and Martínez, F.U.I. (2022). Creation of serverless applications in the cloud. *Proceedings of the* 2022 11th *International*

Conference On Software Process Improvement (*CIMPS*), Acapulco, Guerrero, Mexico, 216-218.

21. Sedefoğlu, Ö.; and Sözer, H. (2021). Cost minimization for deploying serverless functions. *Proceedings of the* 36th *Annual ACM Symposium on Applied Computing,* New York, United States, 83-85.
22. Fang, J.; Zhu, Z.; Li, S.; Su, H.; Yu, Y.; Zhou, J.; and You, Y. (2022). Parallel training of pre-trained models via chunk-based dynamic memory management. *IEEE Transactions on Parallel and Distributed Systems*, 34(1), 304-315.
23. Wu, Z.; Cai, X.; Zhang, C.; Qiao, H.; Wu, Y.; Zhang, Y.; and Duan, H. (2022). Self-supervised molecular pretraining strategy for low-resource reaction prediction scenarios. *Journal of Chemical Information and Modeling*, 62(19), 4579-4590.
24. Diao, S.; Xu, R.; Su, H.; Jiang, Y.; Song, Y.; and Zhang, T. (2021). Taming pre-trained language models with n-gram representations for low-resource domain adaptation. *Proceedings of the* 59th *Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers), Online, 3336-3349.
25. Ouhame, S.; Hadi, Y.; and Ullah, A. (2021). An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. *Neural Computing and Applications*, 33(16), 10043-10055.
26. Spillner, J. (2020). Resource management for cloud functions with memory tracing, profiling and autotuning. *Proceedings of the* 2020 *Sixth International Workshop on Serverless Computing,* New York, United States, 13-18.
27. Fowers, J.; Ovtcharov, K.; Papamichael, M.; Massengill, T.; Liu, M.; Lo, D.; and Burger, D. (2018). A configurable cloud-scale DNN processor for real-time AI. *Proceedings of the* 2018 *ACM/IEEE* 45th *Annual International Symposium on Computer Architecture (ISCA),* Los Angeles, CA, USA, 1-14.
28. Ravi, B.S.; Madhukanthi, C.; Sivasankar, P.; and Prasanna, J.D. (2023). A novel technique to improve latency and response time of AI models using serverless infrastructure. *Proceedings of the* 2023 *International Conference on Inventive Computation Technologies (ICICT),* Lalitpur, Nepal, 428-433.