Analysis of Web Service Performance Using Microservice and Monolithic Architectures

Haris Hamdani Latif [1], Siti Amatullah Karimah [2], Sidik Prabowo [3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]harishl@students.telkomuniversity.ac.id, [2]karimahsiti@telkomuniversity.ac.id,
[3]pakwowo@telkomuniversity.ac.id

Abstract

Container-based virtualization technology is currently popularly used across all cloud platforms and is predicted to continue for the next few years. The use of this container technology will make it easier and save resources used for services. Coupled with the development of the current web architecture which is increasingly being developed and used for commercial purposes, including Microservice and Monolithic. This Microservice architecture divides its services into smaller parts based on functionality. Meanwhile, Monolithic Architecture is referred to as conventional architecture because in it services become a unified whole. For this reason, a test scenario was carried out to determine the performance of the two web architectures. In this study, load testing was carried out with the number of requests 50, 100, 500, and 1000 on Microservice and Monolithic to show scalability. The results show that the Monolithic service is superior with an average CPU usage on AWS of 83% while Microservice is at 99%. CPU in Monolithic Docker Container is 92% while Microservice is 30% for each service. For Memory Usage, Microservice gets an average of 14% while for Monolithic services it is 12%. Response Time was obtained at 1497.31 ms for Microservice and 89.02 ms for Monolithic. In testing the availability by terminating/stopping the service in the Microservice service then it is reactivated and takes 2 seconds, while in the Monolithic service it takes 3 seconds to restore the service. When the service is turned off, the Microservice service can still run normally, only the dead service will experience interference, this is inversely proportional to the Monolithic service which will be completely dead when the service is turned off.

Keywords : Microservice, Monolithic, Container, Virtualization, Scalability, Availability