

Resumen

Existe una gran cantidad de proyectos realizados sobre una arquitectura monolítica, generando dificultades al momento de escalar o realizar actualizaciones, situación que no ocurre con la arquitectura de microservicios, ya que cada parte es un componente pequeño que se puede escalar de manera independiente.

Paralelamente, con el objetivo de mejorar el tiempo de desarrollo y escalabilidad de cada componente, surgen frameworks que permiten obtener un mejor rendimiento, como Quarkus, que es optimizado para contenedores e implementado en aplicaciones serverless.

En esta investigación se realiza la comparación del framework Quarkus con Spring Boot, este último es el más popular en el ecosistema Java, para la comparación se construyó una aplicación sobre la JVM y sobre GraalVM, para determinar qué framework brinda un mejor rendimiento al momento de recibir y atender una alta demanda. Dichas solicitudes fueron enviadas por Artillery, una biblioteca de JavaScript que permite realizar pruebas de carga y analizar los logs en AWS CloudWatch.

Al finalizar con los análisis de las métricas de AWS CloudWatch, los resultados indicaron que el framework Quarkus sobre GraalVM tiene un mejor rendimiento, ya que atiende 153526 peticiones de la prueba de carga, con una latencia de 10.4778ms, mientras que Spring sobre GraalVM atiende 78621 peticiones con una latencia de 35.7914ms.

Palabras Claves: Quarkus, Serverless, Lambdas, GraalVM, Imagen Nativa

Abstract

There are many projects based on a monolithic architecture, generating difficulties at the time of scaling or upgrades, a situation that does not occur with microservices architecture, since each part is a small component that can be scaled independently.

At the same time, to improve the development time and scalability of each component, frameworks have emerged that allow better performance, such as Quarkus, which is optimized for containers and implemented in serverless applications.

In this research a comparison of the Quarkus framework with Spring Boot, the latter is the most popular in the Java ecosystem, for the comparison an application was built on the JVM and on GraalVM, to determine which framework provides better performance when receiving and serving a high demand. These requests will be sent by Artillery, a JavaScript library that allows load testing and log analysis in AWS CloudWatch.

At the end of the AWS CloudWatch metrics analysis, the results indicated that the Quarkus framework on GraalVM has better performance, serving 153526 load test requests, with a latency of 10.4778ms, while Spring on GraalVM serves 78621 requests with a latency of 35.7914ms.

Keywords: Quarkus, Serverless, Lambdas, GraalVM, Native Image