SERVERLESS PERFORMANCE IMPROVEMENT FOR KNATIVE USING PREDICTIVE AUTO SCALING

Kamindu Nanayakkara

A dissertation submitted in partial fulfilment of the requirement for MSc in Big Data Analytics

Department of Computing

Informatics Institute of Technology, Sri Lanka

in collaboration with

Robert Gordon University, Aberdeen, Scotland

2021

Abstract

Function-as-a-service (FAAS) is a capability Serverless computing platforms provide to the end-users with the promise of fine-grain scalability, high availability by abstracting away the infrastructure provisioning/management load balancing. This also comes with pay for only the execution times payment structure. These frameworks can scale down the services to zero (let it go cold) when the demand to the service falls and spin back the service instances when the demand regains.

Cloud providers have various products based on serverless architectures. These offerings use proprietary implementations and are tightly coupled to the cloud provider. Then there are also serverless architectures implementations on cloud-agnostic technologies such as Kubernetes. When it comes to Kubernetes-based frameworks for managing serverless workloads Knative is a popular technology. It has two main components called Eventing and Serving. To deploy serverless apps as Knative services as well the scaling of them is managed by the Serving primitive in the framework. However, the serving component only uses a moving average method to calculate the number of pods, that calculation is a reactive approach based on past data and may not properly be able to handle future changes in usage.

This study proposes a methodology to minimize the cold start by using an auto scaler that can scale by predicting the future workload based on past metrics. The proposed solution in this study used a predictive autoscaling component as a custom component for the Knative framework. This is deployed as a native component into the Kubernetes cluster and takes the Prometheus metrics server for the input. The predictive component is done using the ARIMA model which is a time series based analytical algorithm. The predictive component is created using python language while the Knative component is created using Golang. The implemented solution is tested under simulated workload conditions to evaluate its effectiveness. The evaluation of the new predictive component has proven to be effective in maintaining a good performance that could meet the expected quality of service.