

An Architecture for Machine Learning-based Fault Prediction and Migration In Cloud Service Systems

Dinh Dai Vu, Young Han Kim*
Soongsil University

daivd@dcn.ssu.ac.kr, younghak@ssu.ac.kr*

Abstract

A cloud system typically contains a huge number of computing nodes with numerous containers running on. Nodes can fail in practice, leading to the downtime of services. Therefore, it is necessary to archive migration work to guarantee service availability in container-based environments. The ability to predict failure nodes acts as a mechanism, allowing container-based services to be migrated to healthy nodes, increasing service availability. To handle this problem, this paper proposes an architecture, which enables correct fault prediction of nodes by using machine learning and a migration module taking suitable action to maintain automatically services on Kubernetes platform.

I. INTRODUCTION

Cloud computing systems offer the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. A number of issues that cause the negative effect on the reliability and availability of system result in applications running on them are prone to failures which could affect a large number of customers and could even lead to massive financial loss.

In recent years, Shifting from virtual machines to containers allows users to deliver their applications more quickly. Kubernetes [1], the container orchestrator that has become recently the most popular cloud manager, offers the deployment, management, and execution of containerized applications with high ability to provide API for getting default or custom metrics. To guarantee service availability, Kubernetes has its recovery function by using replicaset controller to control service that can be mapped to many pod instances. In case of keeping the state of container applications, Kubernetes uses stateful-set for deployment. The applications have to design to separate the state from local-pod state by storing any necessary data into pod-independent persistent storage. However, the above mechanism does not cover all scenarios due to some applications can not be designed in this way. Hence, It results in the need for a container migration mechanism that can restore the container application with keeping its process in the other healthy worker node in Kubernetes.

To take advantage of migration work, It is necessary to determine the scenarios that need to use migration mechanism. A Pod needs to be migrated when the worker node is in overload situations, such as CPU or memory overload. In that case, Pods running inside will be automatically restarted or down and Service will point to another pod running on

another node, even though this worker node is still running. In addition, Kubelet, a component of Kubernetes, can detect a failed worker node, however, at that time, the master node already lost control to that worker node. In other words, we can not use the migration module in that case. Therefore, the cloud system needs to have a fault prediction mechanism to solve those problems

Failure prediction based on runtime system metrics is an important area of research that can help in improving the performance of the entire system. To predict the failure, Machine Learning(ML) approach has become one of the popular methods in comparison with traditional methods like using threshold. In [2], the authors propose MING, which can improve service availability by predicting node failure in cloud service systems. Through failure prediction, intelligent VM allocation and migration can be achieved. However, they only focus on VM cloud while container environments are prominent. In [3], the authors propose a framework for fault detection and prediction in edge cloud environments. For fault detection, they apply various models including RF, SVM. For fault prediction, they use deep learning techniques including LSTM and CNN with high accuracy. They are limited to a mechanism for migrating service automatically after predict correctly the faults.

In this paper, we proposed the architecture which contains a fault prediction module based on ML technique that helps to predict failure nodes before it already happened. Through failure prediction, migration for container service can be achieved.

II. THE PROPOSED ARCHITECTURE

A. GENERAL ARCHITECTURE

This section describes our design for migrating container service running on Kubernetes based on faults predicted by machine learning model. Figure 1 shows the main building blocks of the system and their

relationship. The proposed architecture consists of cloud infrastructure which deploys Kubernetes for container orchestrator, monitor framework, and fault prediction module. In general, monitor framework is responsible for monitoring the status of Kubernetes cluster and collecting real-time data from nodes. This data will be passed through fault prediction module to give a prediction. In case of any failure node in the cluster is predicted, migration controller will be called by migration API to migrate container service to another healthy node.

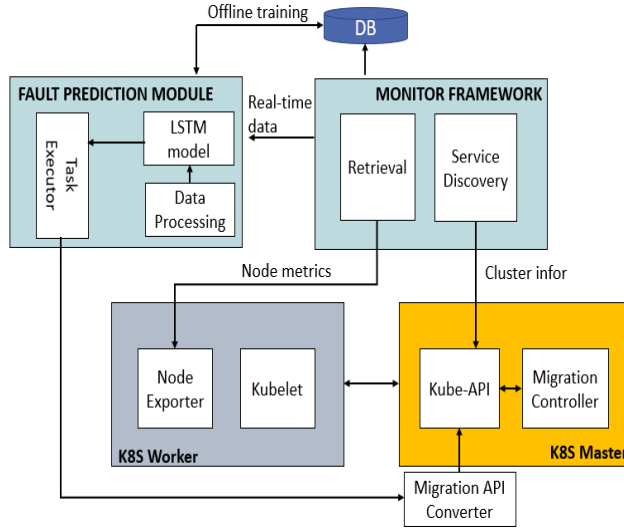


Figure 1. System architecture

B. FAILURE PREDICTION MODULE

This module consists of receiving cluster metric, a data processing unit that processes it, and checking the failure value by inputting the processed data into the learned model. For machine learning model, we choose to apply LSTM (Long Short-Term Memory) since LSTM can balance between retaining the previous state and memorizing new information. LSTM can better capture the patterns behind the time-series data.

A possible problem that the performance of model may decline over time in real-time system. This may be because the assumptions made and captured in the model are changing or no longer hold due to the new flow of data. Therefore, LSTM model has to update automatically on a periodic schedule. In our architecture, we use the real-time data for predicting and also store it in an external database for the strategy to retrain model on both new and old data.

In addition, when our system scale up or down, it needs a mechanism to detect and put data of existed nodes for fault prediction module automatically. To solve this problem, Service discovery in monitor framework updates cluster information frequently to give the list of worker nodes. Retrieval will use this list to gets real-time metrics from all of nodes in the system. In other words, the system can automatically predict the failure when we add new nodes or remove any nodes.

C. MIGRATION CONTROLLER

In our system, Prediction module acts as a mechanism to determine when we should migrate container service. If the result of prediction shows that a node is prone to be failed, Migration Controller will be called by Task Executor through Migration API to migrate pod to another healthy node.

Currently, relocating a pod in Kubernetes is only possible by disposing of the source pod, then recreating a new pod of the same type or template from scratch. We propose the live migration pod strategy that using checkpoint with CRIU (Checkpoint Restore in Userspace) [4]. If the container, even in a StatefulSet, is started again, it still needs to load all the data from storage. CRIU enables to have a stateful migration including everything which the process has loaded in memory. The running container can be written to disk and it can be restored in another pod while keeping its state and all data already loaded into memory.

III. CONCLUSION

This paper introduces a combination flow of fault prediction and migration for Kubernetes. The proposed architecture is expected to predict correctly failures of service based on machine learning model. By using pop migration mechanisms in case of node failure, container services can be maintained. The proposed architecture will be implemented and evaluated.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2017-0-01633) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

REFERENCES

- [1] Kubernetes, (<https://kubernetes.io/>)
- [2] L. Qingwei, H. Ken, D. Yingnong, Z. Hongyu, "Predicting node failure in cloud service systems." In Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 480-490, 2018
- [3] S. Mbarka, F. Chunyan, K. Foutse, "Infrastructure Fault Detection and Prediction in Edge Cloud Environments" Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, pp. 222-235, 2019
- [4] CRIU Authors, (https://criu.org/Main_Page)