F3: Fault Forecasting Framework for Cloud Systems

Pu Zhao*, Chuan Luo*, Bo Qiao*, Youjiang Wu†, Yingnong Dang†, Murali Chintalapati†, Susy Yi‡,
Paul Wang‡, Andrew Zhou‡, Saravanakumar Rajmohan‡, Qingwei Lin*, Dongmei Zhang*
*Microsoft Research †Microsoft Azure †Microsoft 365
{puzhao, chuan.luo, boqiao, yow, yidang, muralic, chayi, miwan, azhou, saravar, qlin, dongmeiz}@microsoft.com

***Abstract*—In recent years, the development of cloud systems (*e.g.,* Microsoft Azure) has grown explosively, and a variety of software services have been deployed on cloud systems. As cloud systems are required to serve customers on a 24/7 basis, high service reliability is essential to them. To reduce the number of the faults in cloud systems, many machine learning based fault forecasting methods have been proposed. Those forecasting methods aim to predict faults in advance so that proactive actions can be taken to avoid negative impact, and they mainly focus on a specific hardware (*e.g.,* disk, memory and node). In cloud systems, many fault forecasting tasks have similar characteristics: 1) they are based on the temporal monitoring data and 2) they usually suffer from similar challenges (*e.g.,* the extreme data imbalance problem). In this work, we present a unified fault forecasting framework for cloud systems, dubbed F3. In particular, F3 introduces an end-to-end pipeline for a variety of fault forecasting tasks in cloud systems, and the pipeline underlying F3 consists of several critical parts (*i.e.,* data processing, fault forecasting, prediction result interpretation and action decision). In this way, when a new fault forecasting task arrives, F3 can be easily and effectively utilized to handle the new task with adaption. Besides, F3 is able to overcome other challenges, including the extreme data imbalance problem, data inconsistency between online and offline environments, as well as model overfitting. More encouragingly, F3 has been successfully applied to Microsoft Azure and has helped significantly reduce the number of virtual machine interruptions.**

I. Introduction

Faults are common in cloud systems, and they are the main cause of service downtime. In practice, service interruption can adversely affect customer experience and even cause huge financial loss. For example, it is recognized that every minute of downtime costs about $9,000 [1]. Since high service reliability is critically important for cloud systems, fault prediction have been well studied, *e.g.,* disk failure prediction [2], memory failure prediction [3] and node failure prediction [4]. Each of them aims to deal with the fault of one specific hardware in cloud systems. Actually, the fault prediction tasks in cloud systems have similar characteristics. For instance, the monitoring status data of each task is recorded every minute or every hour; in another words, those fault prediction tasks are based on temporal data, so each of them can be formulated as a temporal information based binary classification problem, which aims to predict whether the hardware will fail within a given time. Besides, those fault forecasting tasks mainly suffer from similar challenges, including the extreme data imbalance problem, data inconsistency between online and offline environments, as well as model overfitting.

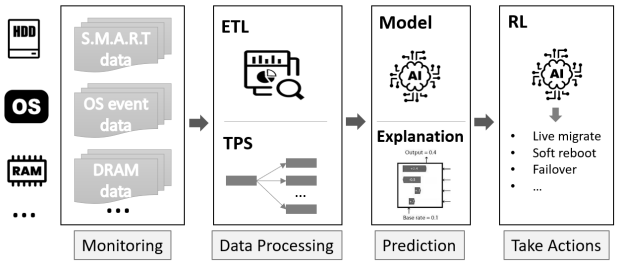

Fig. 1. The overview of F3.

In this work, we present a fault forecasting framework for cloud systems, dubbed F3. In particular, F3 is an unified, end-to-end framework to handle faults in cloud systems, and we will introduce it in the following sections.

II. F3: Fault Forecasting Framework

A. Overview

The entire fault handling pipeline underlying our F3 framework consists of four critical parts, including data processing, fault forecasting, prediction result interpretation and action decision. First, F3 collects the necessary data, including the hardware's own signal and its contextual signals. Then, F3 applies pre-processes the collected data. In particular, F3 utilizes feature engineering to select useful features, processes the missing feature values, and transforms the processed data into the format required by the fault forecasting model. After that, the fault forecasting model underlying F3 takes the processed features as its input, and outputs the fault probability to indicate whether current hardware will occur fault within a given period. The forecasting model underlying F3 is a contextual-temporal attention based deep learning model, which is able to fit various fault forecasting scenarios in cloud systems. Finally, we identify those hardwares with high fault probabilities and take mitigation actions (*e.g.,* live migrate and soft reboot) through reinforcement learning based action decider.

B. Data Processing

In practice, the data processing part underlying F3 contains data extraction, transformation and loading (ETL), as well as an effective method to deal with the data imbalance problem. For data ETL, we mainly focus on the data quality. In the online environment, due to data delay and unstable

data transmission, data missing is common in production. However, when we train a model using historical data in an offline environment, those missing data may have been filled, which means that the data quality in the offline environment would be better than that in the online environment. The difference of data quality between the online and offline environments would result in the forecasting performance degradation in online environment. To address this problem, we have made efforts in several ways. For example, we equip the data processing part with a data quality monitor, which can compute the KL divergence between current online data and historical offline data. Also, we employ data masking mechanism to make the missing rate of offline data align with online environment.

Besides, in the context of fault prediction for cloud systems, since the number of healthy hardwares is much greater than that of the failed ones, both traditional machine learning approaches and deep learning approaches suffer from the extreme data imbalance problem. In order to address this problem, we employ an effective method called Temporal Progressive Sampling (TPS) [5] to generate more failed samples to complement the data distribution of failed disks. Through generating more failed samples by TPS, the ratio between the number of healthy samples and that of failed samples would achieve a better balance.

C. Fault Forecasting

F3 aims to predict the faults as early as possible. In practice, if a system only takes mitigation actions after a fault is detected, users may have already experienced unstable service as the system runs in a degraded mode (although not completely failed). However, predicting faults ahead would cause lower precision than fault detection. To improve the prediction performance, when identifying whether one hardware will fail, F3 not only uses that hardware's own status information, but also utilizes the contextual information, *e.g.,* the status information of neighboring hardwares and OS status information, which have shown effectiveness in enhancing prediction [6].

We propose a novel deep learning based approach, named Contextual-Temporal Attention Model (CTAM), including a contextual information encoder, a temporal information encoder and a fully connected network. The contextual information encoder employs positional-attention to calculate the weight of each neighbor and the weighted sum of all neighbors' features, and then concatenate the OS status information to represent contextual information. The temporal information encoder consists of positional encoding, self-attention, and location-based attention layers, which can better capture the temporal information. The fully connected network is treated as fusion layer to do binary classification. We use the binary cross entropy as the loss function to train CTAM.

D. Prediction Result Interpretation

In production, prediction result interpretation is necessary, which can help us find the root cause of the fault. Therefore, in F3 we employ the Shapley additive explanation, which is a game theoretic approach, to explain the outputs of the forecasting model. It would help us quantify the the contribution of each feature to the output score. Besides, since the core forecasting model CTAM is based on the attention mechanism, we can easily identify the key time point that caused fault through the weight score calculated by location-based attention in the temporal information encoder underlying F3.

E. Action Decision

In order to mitigate a potential fault, the common practice is ad-hoc, which means that developers use static policies that prescribe actions based on the symptoms and domain knowledge. Although this approach works for simple systems, it does not work well for large-scale cloud systems. With multi-tenancy, heterogeneous infrastructures and diverse customer workloads, it is difficult to comprehensively categorize different fault scenarios in cloud systems and determine effective mitigation actions (or their hyper-parameters). Moreover, as the cloud system is constantly evolving (*e.g.,* software updating, hardware updating and customer workload changing), a number of mitigation action that worked well in the past might no longer be optimal. As a result, developers keep reactively adjusting the actions based on hind sights from service incidents. Therefore, cloud systems urgently call for smart and adaptive fault mitigation actions. In practice, for a mitigation action, it is infeasible to know whether or not the mitigation action is effective without trying it. Based on this insight, inspired by the success of the Narya system [6], F3 provides a reinforcement learning based decision engine to explore the best mitigation action through A/B testing, which would measure the benefits of each action and iteratively optimize the decision engine through policy gradient.

III. Application in Practice

We have successfully applied F3 to Microsoft Azure and Microsoft 365, in order to help improve the service reliability. In our industrial practice, the prediction task runs as an hourly job. After the deployment of F3, the number of virtual machine interruptions has been significantly reduced.

References

[1] Ponemon Institute, "Cost of data center outages," in *Data Center Performance Benchmark Series*, 2016.

[2] Y. Xu, K. Sui, R. Yao, H. Zhang, Q. Lin, Y. Dang, P. Li, K. Jiang, W. Zhang, J. Lou, M. Chintalapati, and D. Zhang, "Improving service availability of cloud systems by predicting disk error," in *Proceedings of USENIX ATC 2018*, 2018, pp. 481–494.

[3] X. Du and C. Li, "Memory failure prediction using online learning," in *Proceedings of MEMSYS 2018*, 2018, pp. 38–49.

[4] Q. Lin, K. Hsieh, Y. Dang, H. Zhang, K. Sui, Y. Xu, J. Lou, C. Li, Y. Wu, R. Yao, M. Chintalapati, and D. Zhang, "Predicting node failure in cloud service systems," in *Proceedings of ESEC/SIGSOFT FSE 2018*, 2018, pp. 480–490.

[5] C. Luo, P. Zhao, B. Qiao, Y. Wu, H. Zhang, W. Wu, W. Lu, Y. Dang, S. Rajmohan, Q. Lin, and D. Zhang, "NTAM: Neighborhood-temporal attention model for disk failure prediction in cloud platforms," in *Proceedings of WWW 2021*, 2021, p. To appear.

[6] S. Levy, R. Yao, Y. Wu, Y. Dang, P. Huang, Z. Mu, P. Zhao, T. Ramani, N. Govindaraju, X. Li, Q. Lin, G. L. Shafriri, and M. Chintalapati, "Predictive and adaptive failure mitigation to avert production cloud VM interruptions," in *Proceedings of OSDI 2020*, 2020, pp. 1155–1170.