AIOps Innovations in Incident Management for Cloud Services

Zhuangbin Chen,[1] Yu Kang,[2] Feng Gao,[3] Li Yang,[3] Jeffrey Sun,[3] Zhangwei Xu,[3]
Pu Zhao,[2] Bo Qiao,[2] Liqun Li,[2] Xu Zhang,[2] Qingwei Lin,[2] Michael R. Lyu,[1]
[1]The Chinese University of Hong Kong, Hong Kong, China, {zbchen, lyu}@cse.cuhk.edu.hk
[2]Microsoft Research, Beijing 100090, China, {kay, Pu.Zhao, boqiao, Liqun.Li, xuzhang2, qlin}@microsoft.com
[3]Microsoft, Redmond, WA 98052, USA, {fgao, lilyan, jeffsun, zhangxu}@microsoft.com

Abstract

While remarkable advances have been achieved in cloud computing infrastructure, the way incidents (unplanned interruptions or outages of a service/product) are managed needs to be as agile and dynamics as the cloud itself. In practice, incident management is conducted through analysing a huge amount of monitoring data collected at the runtime of services. Given its data-driven nature, we deem AIOps innovations as essential to empowering cloud systems to provide more reliable online services and applications by incorporating more intelligence into the entire workflow of incident management. This paper presents a project showcase of our AIOps practices towards these goals at Microsoft. First, we brief the incident management procedure and its corresponding real-world challenges. Then, we elaborate the ML & AI techniques used for mitigating such challenges and share some application results to demonstrate the intelligence and benefits conveyed to Microsoft service products.

Incident Management of Cloud Services

Incident Management Procedure

A typical procedure of incident management is shown in Figure 1, which consists of the following three steps. Correspondingly, the time costed in different phases is defined as Time to Detect (TTD), Time to Engage (TTE), and Time to Mitigate (TTM). The goal of improving incident management is to minimize these TTx, efficiently mitigate the incident impact, and reduce operation loads.

1) *Incident Reporting*. Incident reporting is the process of detecting service violations or performance degradation and creating a ticket to record relevant information. In cloud systems, incident can be detected via manual ways (i.e., reported by customers or engineers) or auto alerts (i.e., generated by health monitors).

2) *Incident Triage*. Upon the creation of an incident, the responsible service team should be quickly engaged for problem investigation, which is called incident triage. However, due to cloud systems' high complexity and dependencies, incidents are frequently assigned to wrong responsible teams, which significantly prolongs service downtime.

3) *Incident Mitigation*. Incident mitigation is the process of bringing problematic service back to normal, so it can continue to serve customers. In practice, some temporary workaround solutions (e.g., service redeployment and server rebooting) will be applied first to mitigate the impact as a short period of downtime could become an expensive drain on company revenue and user trust (Chen and others 2019a).

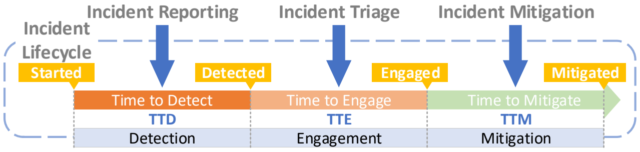

Figure 1: Incident management procedure

Pain Points and Challenges

Incident management is a challenging task due to the ever-increasing customer volume and complex physical infrastructure. Based on our practice and experience at Microsoft, we have identified the following challenges.

Large-volume and non-homogeneous data. During daily operations, services can produce terabytes or even petabytes of monitoring data. However, most of the monitoring data is irrelevant to a particular incident, e.g., plain system runtime behaviors. On the other hand, one failure could trigger multiple incidents due to its cascading effect along service dependency chain. This problem is further compounded by the fact that different service teams and monitors have distinct standards on rendering incident reports.

Complex failure-indicating signals. There are many potential causes that may incur a service incident, such as code defects, hardware failures, networking issues, resource competition, and configurations (Lou and others 2013). Therefore, there is no simple rule or single metric that can detect all incidents in a straightforward manner. Various types of monitoring data (e.g., temporal and spatial signals) should be collected and analysed simultaneously as they can potentially contribute to problem identification.

AIOps Innovations for Incident Management

To incorporate the power of AIOps into cloud services, we have designed and initiated an AIOps-oriented incident management project called BRAIN targeting for real-world scenarios in Microsoft online services. In this section, we first envision the impact brought by AIOps, then elaborate the techniques of BRAIN, and finally we share the initial application experience and feedback of BRAIN deployment.

Our Vision of AIOps in Incident Management

High Intelligence and Automation. Currently, severe and critical incidents are still tackled in a labor-intensive manner, which is inefficient and error-prone. In AIOps-powered incident management system, service violation may be alerted before its actual occurrence by predicting service's future status based on its historical behaviors, workload patterns, and underlying infrastructure activities, etc. (Dang and others 2019). When incident indeed happens, it can be auto-routed to the right responsible service team and a solution or workaround will be provided for quick service restoration.

High Engineering Productivity. Provided with powerful tools, software and service engineers can effectively and efficiently build and operate services throughout their whole lifecycle. Engineering efforts are shifted from repeated issue investigations (e.g., manual data collection/cleansing and identical issue fixing) to more intelligence-intensive initiatives, necessary architecture modifications, and service adaption strategy changes, etc. (Dang and others 2019).

Techniques of Our AIOps Solutions

A series of data-driven techniques has been developed in BRAIN, which targets different phrases of incident management. Specifically, incident detection improves the quality of incident reporting, while incident correlation and summary together promote incident triage and mitigation.

1) *Incident Detection*. To pursue a high accuracy of prediction, BRAIN gathers a variety of signals (e.g., service health data, infrastructure monitoring data, manual signal, customer input, etc.) that can potentially contribute to incident detection. However, having just resource health signals is not enough. What is missing is the resource relationship that helps understand topologies, resiliency models, and dependencies among the entire cloud system. Therefore, system topology with resource hierarchy information is also leveraged in BRAIN. Particularly, by modeling the relationship between outages (i.e., impactful incidents) and alerting signals with Bayesian network, our model (Chen and others 2019b) has achieved 0.89 F1 score in predicting outages from 8k incidents (obtained from tens of datacenters over 1 year of operation) with a gradient boosting tree based model.

2) *Incident Correlation*. Due to incident's cascading effect, being able to correlate related and identical incidents is critical to the operational efficacy and efficiency of both incident triage and mitigation. Meanwhile, it can assist us in customer communication by precisely locating the impacted users. Towards this end, incident title that briefly describes the cloud issue is utilized to do correlation. Specifically, we first cluster incidents by leveraging the resources tagged in each incident and then adapt previous log parsing approaches to identify incident topics in a fine-grained manner. Finally, incidents are correlated by following historical links among incidents, which are manually marked by On-Call Engineers (OCEs) during incident investigation. Moreover, BRAIN also exploits end-to-end solutions to discover new incident correlations by feeding incident's property information (e.g., title and discussion) to models. Particularly, our first attempt (Chen and others 2019a) has confirmed the effectiveness of facilitating incident triage with such information by showing a notable accuracy of 0.64∼0.73, which outperforms the state-of-the-art bug triage approach by a significant margin of 12.2%∼35.5%.

3) *Incident Summary*. Directly providing OCEs with a bunch of related incidents sometimes may not be useful as the size can be very large. Therefore, to quickly give OCEs a big picture of the ongoing issues and further facilitate post-mortem analysis, we propose to summarize the related incidents by extracting valuable information from their titles and discussions, which is cast to a text summarization problem using deep learning. However, state-of-the-art models fail to achieve notable results, because a significant portion of incidents are generated by machines with semi-structured languages. To tackle this challenge, our approach first iteratively differentiates several categories of entities (e.g., categorical variable and and error message) from both machine-generated and human-written incidents and then embeds these entities separately. Finally, the embedding representations are combined with the preserved structural relationship and fed into a summarization model (Hu and others 2018).

Project Deployment

A small-scale pilot implementation of BRAIN has been initiated in the existing incident management system for several popular online services at Microsoft, which serve hundreds of millions of users on a 24/7 basis. To assess its real-world benefits, we analyse daily usage data, measure TTx reduction, and conduct field communications with OCEs. Although BRAIN is at the early application stage, we have seen it shedding lights on all three incident management phases and received many positive feedback. Specifically, for many incidents reported via BRAIN, OCEs confirmed the difficulty of their auto-detection in the existing monitoring system. Moreover, when diagnosing incidents, the recommended related incidents and summaries successfully assisted OCEs in reducing their investigation scope.

Acknowledgement

The work was supported by the National Natural Science Foundation of China (Project No. 61702107).

References

Chen, J., et al. 2019a. Continuous incident triage for large-scale online service systems. In *Proc. of ASE'19*.

Chen, Y., et al. 2019b. Outage prediction and diagnosis for cloud service systems. In *WWW'19*.

Dang, Y., et al. 2019. Aiops: real-world challenges and research innovations. In *Proc. of ICSE'19: Companion Proc.*

Hu, X., et al. 2018. Deep code comment generation. In *Proc. of ICPC'18*.

Lou, J.-G., et al. 2013. Software analytics for incident management of online services: An experience report. In *Proc. of ASE'13*.