AUTHOR QUERY FORM

ELSEVIER	**Book: Perspectives on Data Science for Software Engineering** **Chapter: 00012**	**Please e-mail your responses and any corrections to:** **E-mail: M.NATARAJAN@elsevier.com**

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

We were unable to process your file(s) fully electronically and have proceeded by

□ Scanning (parts of) your article □ Rekeying (parts of) your article □ Scanning the artwork

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

Location in Chapter	Query / Remark: click on the Q link to go Please insert your reply or correction at the corresponding line in the proof
Q1	The first paragraph has been treated as the Abstract. Please check and confirm.
Q2	Please check if the keywords suggested by the editor are appropriate.
	Please check this box or indicate your approval if you have no corrections to make to the PDF file □

Thank you for your assistance.

c0060 How to tame your online services

Q. Lin, J.-G. Lou, H. Zhang, D. Zhang
Microsoft Research, Beijing, China

CHAPTER OUTLINE

s0010 BACKGROUND

p0025 Online service systems, such as online banking systems and e-commerce systems, have become increasingly popular and important in our society. During the operation of an online service, a live-site service incident can happen: an unplanned interruption, outage, or degradation in the quality of the service. Such incidents can lead to huge economic loss or other serious consequences. For example, the estimated average cost of 1 h service downtime for Amazon.com is $180,000 [1].

p0030 Once a service incident occurs, the service provider should take actions immediately to diagnose the incident and restore the service as soon as possible. A typical procedure of incident management in practice (eg, at Microsoft and other service-provider companies) goes as follows. When the service monitoring system detects a service violation, the system automatically sends out an alert and makes a phone call to a group of on-call engineers to trigger an incident investigation. Given an incident, engineers need to understand what the problem is and how to resolve it. In ideal cases, engineers can identify the root cause of the incident and fix it quickly. However, in many cases, engineers are unable to identify or fix root causes within a short time, as it usually takes time to identify and fix the root causes, conduct regression testing, and re-deploy the new version to data centers. Thus, in order to recover the service as soon as possible, a common practice is to restore the service by identifying a temporary workaround solution (such as restarting a server component) to restore the service. Then, after service restoration, identifying and fixing the underlying root cause for the incident can be conducted via offline postmortem analysis. Incident management has become a critical task for online services. The goal is to minimize the service downtime and to ensure high quality of the provided services. In practice, incident management of an online service heavily depends on data collected at runtime of the service, such as service-level logs, performance counters, and machine/process/service-level events. Such monitoring data typically contains information that reflects the runtime state and behavior of the service. Based on the data collected, service incidents can be detected and mitigated in a timely way.

s0015 SERVICE ANALYSIS STUDIO

p0035 We formulated the problem of incident management for online services as a software analytics problem [2], which can be tackled with phases of task definition, data preparation, analytic-technology development, and deployment and feedback gathering. We carried out a 2-year research project, where we designed a set of incident management techniques based on the analysis of a huge amount of data collected at service runtime [3]. As a result of this project, we developed a tool called Service Analysis Studio (SAS), which targets real incident management scenarios of large-scale online services provided by Microsoft.

p0040 SAS includes a set of data-driven techniques for diagnosing service incidents. Each of these techniques targets a specific scenario and a certain type of data. Here we briefly introduce some of the major techniques SAS offers:

- u0025 *Identification of incident beacons from system metrics*: When engineers diagnose incidents of online services, they usually start by hunting for a small subset of system metrics that are symptoms of the incidents. We call such kinds of metrics "service-incident beacons." A service-incident beacon could provide useful information, helping engineers locate the cause of an incident. For example, when a resource-intensive SQL query blocks the execution of other queries accessing the same table, symptoms can be observed on monitoring data: the waiting time on the SQL-inducing lock becomes longer, and the event "SQL query time out failure" is triggered. Such metrics can be considered service-incident beacons. We developed data mining-based-techniques that helped engineers effectively and efficiently identify service-incident beacons from huge numbers of system metrics. The technical details can be found in [4,5].
- u0030 *Leveraging previous effort for recurrent incidents*: Engineers of an online service system may receive many similar incident reports. Therefore, leveraging the knowledge from past incidents can help improve the effectiveness and efficiency of incident management. The key here is to design a technique that automatically retrieves the past incidents similar to the new one, and then proposes a potential restoration action based on the past solutions. More details can be found in [6].
- u0035 *Mining suspicious execution patterns*: Transactional logs provide rich information for diagnosing service incidents. When scanning through the logs, engineers usually look for a set of log events that appear in the log sequences of

failed requests, but not in the ones of the succeeded requests. Such a set of log events are named as suspicious execution patterns. A suspicious execution pattern could be an error message indicating a specific fault, or a combination of log events of several operations. For example, many normal execution paths look like (task start, user login, cookie validation success, access resource R, do the job, logout). In contrast, a failed execution path may look like (task start, user login, cookie not found, security token rebuild, access resource R error). The code branch reflected by (cookie not found, security token rebuild, access resource R error) indicates a suspicious execution pattern. We proposed a mining-based technique to automatically identify suspicious execution patterns. The details of our technique can be found in [6].

s0020 SUCCESS STORY

p0060 We have successfully applied SAS to one of Microsoft's large-scale services (a geographically distributed, web-based service serving hundreds of millions of users). Similar to other online services, the Microsoft service is expected to provide high-quality service at all times. In the past, the service team faced great challenges in improving the effectiveness and efficiency of their incident management in order to provide high-quality service. SAS was first deployed to the datacenters of the service in June 2011. The engineers of the service team have been using SAS for incident management since then. The actual usage experience shows that SAS helps the engineers improve the effectiveness and efficiency of incident management. According to the usage data from a 6-month empirical study, about 91% of engineers used SAS to accomplish their incident management tasks and SAS was used to diagnose about 86% of service incidents. Now SAS has been successfully deployed to many Microsoft product datacenters and widely used by on-call engineers for incident management.

p0065 Our experience shows that incident management has become a critical task for a large-scale online service. Software analytics techniques can be successfully applied to ensure high quality and reliability of the service, utilizing the data collected at service runtime.

REFERENCES

[1] Patterson DA. A simple way to estimate the cost of downtime. In: Proc. of LISA' 02; 2002. p. 185–8.

[2] Zhang D, Han S, Dang Y, Lou J, Zhang H, Xie T. Software analytics in practice. IEEE Softw 2013;30(5):30–7. (special issue on the many faces of software analytics).

[3] Lou J-G, Lin Q, Ding R, Fu Q, Zhang D, Xie T. Software analytics for incident management of online services: an experience report. In: Proceedings of the 28th IEEE/ACM international conference on automated software engineering (ASE 2013), Palo Alto, California; 2013.

[4] Lim M-H, Lou J-G, Zhang H, Fu Q, Teoh ABJ, Lin Q, et al. Identifying recurrent and unknown performance issues. In: Proceedings of IEEE international conference on data mining 2014 (ICDM 2014), Shenzhen, China; 2014.

[5] Fu Q, Lou J-G, Lin Q, Ding R, Zhang D, Ye Z, et al. Performance issue diagnosis for online service systems. In: Proceedings of 31st international symposium on reliable distributed systems (SRDS 2012); 2012.

[6] Ding R, Fu Q, Lou J-G, Lin Q, Zhang D, Shen J, et al. Healing online service systems via mining historical issue repositories. In: Proceedings of the 27th IEEE/ACM international conference on automated software engineering (ASE 2012), Essen, Germany; 2012.

Non-Print Items

Abstract

Online service systems, such as online banking systems and e-commerce systems, have become increasingly popular and important in our society. During the operation of an online service, a live-site service incident can happen: an unplanned interruption, outage, or degradation in the quality of the service. Such incidents can lead to huge economic loss or other serious consequences. For example, the estimated average cost of 1 h service downtime for Amazon.com is $180,000. Q1

Keywords: Online service systems, Service incident, Incident management, Service Analysis Studio (SAS), Service-incident beacons, Transactional logs Q2