ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Identification, Diagnosis of Errors in Sporadic Operations on Cloud Application

Pramod[1], Venkateshwar[2]

M. Tech Student, Dept. of Computer Engineering, BITM, Ballari, Karnataka, India[1]

Associate Professor, Dept of Computer Engineering, BITM, Ballari, Karnataka, India[2]

ABSTRACT: Modern software engineering represents a fast-paced, collaborative task of extreme complexity. To succeed in a world where technologies, requirements, ideas, tools, and timelines are constantly changing and information must be accurate, readily available, easily found, and delivered constantly, in real-time, to all team members. To meet these challenges devops are used. Devops aims to maximize the predictability, efficiency, security, and maintainability of operational processes. large scale applications in the cloud are subject to sporadic changes due to operational activities such as upgrade, redeployment, and on-demand scaling. Traditional anomaly-detection-based diagnosis techniques are less effective during sporadic operation periods. Diagnosing operation errors in such applications has always been difficult. The recent emergence of Devops gives a solution for sporadic changes.

In this paper, we propose two software development methodologies (1) agile (2) Devops and Process Oriented Dependability (POD) an approach that explicitly models these sporadic operations as processes and uses the process context to locate errors, filter logs, visit fault trees, and perform on-demand assertion evaluation for online error diagnosis and root cause analysis. Using devops concept we can Diagnosis of Errors in Sporadic Operations.

KEYWORDS: Software engineering; Software development; Devops; cloud; deployment; process mining; error detection; error diagnosis;

I. INTRODUCTION

Software development processes are fundamentally based on efficient and effective communication. large-scale applications in cloud systems may consist of thousands of nodes with complex software stacks inside each node and dependencies among nodes. Diagnosing operation errors in such applications has always been difficult. as information changes or new information becomes available, existing information must be updated immediately to avoid I) injecting defects into the software being developed and II) wasted effort spent working with incorrect information. Two modern software developments provide partial solutions to this problem they are data-driven development and automated devops. Data-driven development is a software development methodology promoting the continual collection of data throughout the SDLC to inform prioritization, tasking and adaptations of the process and product to ensure project success within temporal and budgetary constraints. devops integration targets product delivery, continuous testing, quality testing, feature development, and maintenance releases in order to improve reliability and security and provide faster development and deployment cycles. Many of the ideas involved in devops came from the enterprise systems management and agile software development movement. Devops aids in software application release management for an organization by standardizing development environments. in the past, major sporadic changes to large-scale applications would be infrequent and often done during scheduled downtime with careful execution. now, with high frequency continuous deployment, sporadic changes are being automated using devops. Process oriented dependability (pod)-diagnosis, an approach that explicitly models these sporadic operations as processes. These models allow us to (I) determine orderly execution of the process, and (II) use the process context to filter logs, trigger assertion evaluations, visit fault trees and perform on-demand assertion evaluation for online error diagnosis and root cause analysis. Error diagnosis during operation time heavily relies on logs. In addition to the unique challenges due to the high frequency and sporadic nature of the operations, there are also log analysis challenges exacerbated by the uncertain cloud environment, multiple distributed log sources and simultaneous confounding operations. Our pod-diagnosis approach also deals with them specifically.

II. RELATED WORK

Operation process is automated through scripts using exception handling mechanisms. This mechanism is best suited for single language. For example error handlers in Asgard, AWS API error code and Chef's fault handlers. After that, configuration error detection and diagnosis tools are used. This approach is largely for diagnosing a single program configuration rather than configuration errors introduced during operation. Both mechanisms are not efficient for large scale applications. log analysis tools are used for diagnosing errors in Sporadic Operations.

III. THE SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life cycle model used to describe a process for planning, designing, testing and deploying an information system. Various methodologies exist to manage the SDLC, also called as Water fall model, which views the process as a sequence of stages moving steadily downward to project completion. This sequence describes the output of each stage to be the input of the next. Though simple to conceptualize and plan, this approach is viewed as having significant shortcomings when applied to the practice of software development, specifically the difficulty in adjusting to change or new information throughout the development process. The inability to adopt to change has been seen to increase the risk of failure in many projects. This deficiency has led to other methodologies to gain footing in the modern software industry.

IV. AGILE TECHNOLOGIES

Some drawbacks of the waterfall model and other similar change-resistant software development methodologies, the Agile software development model was proposed. Agile methodologies are software development methods that focus on iterative and incremental development, often emphasizing direct and constant communication with stakeholders, adaptive planning, and ever-evolving requirements. Practitioners believe that designing processes to adapt to change and new information effectively and efficiently leads to reduced project risk and significantly enhanced project outcomes. To constantly adapt to change, teams implementing Agile require highly effective communication. These teams often leverage specialized tools and techniques to ensure rapid and robust communication both within teams and between teams and stakeholders.

V. DEVOPS

Devops is a culture that emphasizes the collaboration and communication of both software developers and other information technology professionals while automating the process of software delivery and infrastructure change. It establishes a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably.

Agile and devops are similar but differ in a few important aspects. Agile is a change of thinking whereas devops is actual organizational cultural change. One goal of devops is to establish an environment where releasing more reliable applications faster and more frequently can occur. Release managers are beginning to utilize tools such as application release automation and continuous integration tools to help advance this goal, doing so though the Continuous delivery approach. Continuous and devops are similar in their meanings and are often conflated, but they are two different concepts devops has a broader scope, and centres around the cultural change, specifically the collaboration of the various teams involved in software delivery as well as automating the processes in software delivery. Continuous Delivery, on the other hand, is an approach to automate the delivery aspect, and focuses on bringing together different processes and executing them more quickly and more frequently. They have common end goals and are often used in conjunction to achieve them. Devops and Continuous Delivery share a background in agile methods and lean thinking: small and quick changes with focused value to the end customer.

VI. PROCESS ORIENTED DEPENDABILITY(POD) DIAGNOSIS METHODOLOGY

Logs are our primary sources of information. Some logs are generated by the cloud infrastructure while other logs are generated by an “operation node” that orchestrates the operation. As we operate in the cloud, some logs are hidden by cloud providers but can be requested through API calls. The process model is created offline using process mining techniques. The assertions and their evaluation are developed based on the process model, also offline. As the rolling upgrade proceeds, it generates log lines that trigger assertion evaluation and conformance checking. Assertion evaluation is also triggered from non-log sources such as timers and diagnosis. Assertions evaluate the state of the environment to detect anomalies. Detection of an anomaly triggers error diagnosis, which is reported to the operator.

- Log Processor:

Local log processor running on operation node. The local log processor is a pipeline connecting a set of log-processing components. Once the log processor detects a new log line in the operation log file, the log line goes through the processing components within the pipeline.

- Conformance Checking:

Conformance checking in process mining refers to methods and algorithms for comparing if an event log fits a process model. Say, the log contains event A followed by B; conformance checking tells us if the process model permits this, or not.

- Assertion Evaluation:

The expected outcomes of each intermediary step are captured as assertions and evaluated at different times for different purposes.

- Error Diagnosis:

When an assertion evaluation fails, a process non-conformance is detected, or an error/failure is reported by other monitoring systems, Error Diagnosis is triggered to diagnose the causes at runtime. Error Diagnosis searches for the root causes when a failure occurs. The errors could be caused by the operations process itself, the cloud infrastructure, co-located applications or other simultaneous operations. The contribution of our diagnosis comes largely from on-demand assertions defined for each type of error/failure. Other diagnostic information can come from third party monitors or configuration repositories, which may provide data on who changed the configuration, when, and why.

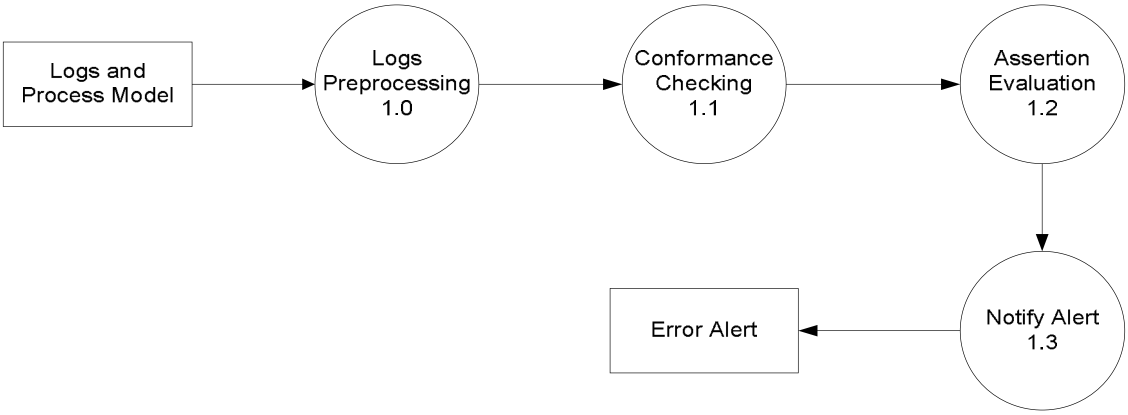

Fig 1. POD diagnosis

DOI: 10.15680/IJIRCCE.2016. 0404261

VII. CONCLUSION

Traditional anomaly-detection-based diagnosis approaches are built for use during normal and simple operations. These approaches are best suited for best suited for a single language environment. Error diagnosis during sporadic operations is difficult using these approaches. So, POD-Diagnosis which treats a sporadic operation as a process detects and diagnoses intermediate step errors through conformance checking, assertion evaluation, and diagnosis errors from root cause.

REFERENCES

[1] S. Kavulya, K. Joshi, F. Giandomenico, and P. Narasimhan, "Failure Diagnosis of Complex Systems," Resilience Assessment and Evaluation of Computing Systems, K. Wolter, A. Avritzer, M. Vieira and A. van Moorsel, eds., pp. 239-261: Springer Berlin Heidelberg,2012

[2] X. Xu, I. Weber, L. Bass, L. Zhu, H. Wada, and F. Teng, “Detecting Cloud Provisioning Errors Using an Annotated Process Model,” in The 8th Workshop for Next Generation Internet Computing (MW4NG2013), Beijing, China, 2013.

[3] W. v. d. Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes: Springer Verlag, 2011.

[4] A. Oliner, A. Ganapathi, and W. Xu, “Advances and challenges in log analysis,” Communications of the ACM, no. 55, pp. 2, 2012.

[5] Etsy, "Infrastructure upgrades with Chef," Code as Craft, 2013.

BIOGRAPHY

Pramod is a PG Scholar Studying in Ballari institute of Technology & Management, Ballari affiliated to VTU, Belgavi. He received his Bachelor of Engineering (BE) degree in 2014 from Acharya institute of technology, Bangalore

Venkateshwar is an associate professor at Ballari institute of Technology & management, Ballari affiliated to VTU,Belgavi