Feedback Driven Development of Cloud Applications

Feedback Driven Development of Cloud Applications
Master-Thesis von Harini Gunabalan aus
Tag der Einreichung:

1. Gutachten: Prof. Dr.-Ing. Mira Mezini
2. Gutachten: Dr. Guido Salvaneschi

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Software Technology Group

Feedback Driven Development of Cloud Applications
Feedback Driven Development of Cloud Applications

Vorgelegte Master-Thesis von Harini Gunabalan aus

1. Gutachten: Prof. Dr.-Ing. Mira Mezini
2. Gutachten: Dr. Guido Salvaneschi

Tag der Einreichung:

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-12345
URL: http://tuprints.ulb.tu-darmstadt.de/1234

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:
Namensnennung – Keine kommerzielle Nutzung – Keine Bearbeitung 2.0 Deutschland
http://creativecommons.org/licenses/by-nc-nd/2.0/de/

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 11th December 2015

(Harini Gunabalan)

Abstract

Over the last few years, the Cloud Computing Paradigm has gained a lot of importance in both the Academia and the Industry. The cloud has not only changed the IT Landscape from the user's perspective but has also changed how the Developers develop applications on the cloud. The increasing adoption of the DevOps approach has led to the removal of the boundaries between the development and the operations.

Among the the three levels of the Cloud Computing: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service(PaaS) and Software-as-a-Service(SaaS), Software Developers are mainly concerned with the PaaS which allows them to focus on the Application Development. Leveraging the fact that the Application is hosted on the cloud, there are additional metrics regarding the application available to Developers in different Log formats. There are also Cloud Monitoring Tools which consolidates these logs with a huge volume of data representing the run-time metrics etc. of the cloud applications. Though monitoring of the cloud applications has been done by many tools, most of the developers do not go through the cumbersome error/warning log data. These log data are not visually made available to the developers in their Development Environment.

This Thesis work aims in bridging this issue. The focus is mainly to map the run-time metrics to the source code artifacts. Mapping will involve log data aggregation, code analysis techniques etc.

Contents

List of Figures

A Introduction

Cloud Computing is one of the fields in Computer Science that has gained a rapid growth and importance in the recent years. The fact that the severs are remote hosted rather than local servers has led to innumerable small scale businesses. Start-ups no longer require high infrastructure, instead they just have to pay per use. This also significantly reduces the initial monetary setup costs of such start-ups. There has been extensive research in Cloud Computing areas such as auto-scaling of resources at the infrastructure, monitoring of metrics at both the application and infrastructure level. However, there is not much research done in how these monitored metrics are utilized by the Cloud Application Developers. Making the run-time metrics visibly effective for the developers in their Development Environment is the major issue this thesis is aiming to solve.

This chapter is structured into four sections. The first section provides the motivation of this thesis work. The second section gives an insight into the problem statement which this Thesis work aims to solve. The third section details the contribution of the work and the fourth section provides how the following chapters of the Thesis are structured.

A.1 Motivation

Software Engineering Practice in the industry has faced a phenomenal change since the advent of Cloud Computing. This is mainly due to the flexibility and the dynamic scaling up and scaling down of the infrastructure as required by the current workload. This proves to be not only elastic but cost-effective as well. It is quite obvious that this elasticity is achieved by the continuous monitoring of several metrics that indicate the demand at the moment, and provisioning the necessary resources to meet the monitored demand. Distributed, scalable enterprise-wide applications also mandate the monitoring of metrics for reasoning the effectiveness of the applications by engineers and business analysts [3].

The metrics that are being monitored vary widely. For instance, the low-level metrics such as memory consumption, CPU utilization, Network bandwidth utilization could be considered to be at the Infrastructure level, whereas higher level metrics such as Response times of methods/procedures, the number of users accessing the application, maximum number of users who can use the application simultaneously could be considered to be at the application level. Sometimes, the application level metrics could depend on the primitive metrics at the infrastructure level as well.

As Cloud Application Development has becoming more common, the run time monitoring metrics of these applications are available through several Application Performance Monitoring (APM) tools such as Amazon Cloudwatch, New Relic, etc. But, they do not provide any valuable and visible feedback to the developers, and hence most of the cloud developers do not use it. However, this run-time monitoring data could be used to provide useful analytic information such as performance hotspots that is taking a lot of execution time, and predictive information such as methods or loops that may become critical, even before the deployment. This type of analytic and predictive feedback should be provided to the developers in their IDEs which otherwise may not be explored by the developers. This technique of utilizing the monitoring data is known Feedback driven development. Feedback Driven Development that provides visual tools to the cloud developers is the focus of this thesis research.

A.2 Problem Statement

The Cloud computing paradigms are further classified into three service models which forms a stack as shown:

- Infrastructure-as-a-Service
- Platform-as-a-Service
- Software-as-a-Service

From the software engineering perspective, it is important to note how Cloud Computing impacts the development practices. Based on the research conducted, there are two important research issues [2].

- Impact of DevOps on Cloud Application Developers: The Cloud Developers are forced to look into the huge log data of the cloud applications.
- Data and Tools utilized by Cloud Developers: The data produced by the logs include Business Metrics, System-level Data etc. Sometimes the implementation changes may even have monetary consequences in the cloud, nevertheless most of the developers do not pay attention to this aspect.

A.3 Contribution

A.4 Structure of the Thesis

This thesis is structured in six chapters. Chapter 1 provides a short introduction into the topic and describes the goals of the thesis. Chapter 2 includes more background information and presents the state of the art of the important topics of this thesis: Cloud Application Performance Monitoring(AMP) tools, Feedback Driven Development(FDD) in general, and how FDD could be useful for Cloud Application Developers. The third Chapter describes an overview of the high-level System Design and the design decisions made in this research work. The fourth chapter explains the system on a lower fine-grained level. Interesting Implementation details are also provided here. Chapter 5 shows the various case studies, evaluates the developed system and illustrates its usage as well as possible applications. Finally, Chapter six provides the conclusion of the thesis and outlines the future work ideas.

B State of Art

B.1 Log Data Management

C System Design

C.1 System Design

D System Implementation

D.1 System Implementation

E Evaluation and Results

E.1 Evaluation

F Conclusion and Future Work

F.1 Conclusion and Future Work

References

[1] Cito J, Leitner P, Gall HC, Dadashi A, Keller A, Roth A. *Runtime Metric Meets Developer - Building Better Cloud Applications Using Feedback* . Onward! 2015 - Proceedings.

[2] Cito J, Leitner P, Gall HC, Fritz T. *The Making of Cloud Applications - An Empirical Study on Software Development for the Cloud*. ESEC/FSE'15 conference.

[3] Leitner, Philipp, et al. *Application-level performance monitoring of cloud services based on the complex event processing paradigm*. Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference on. IEEE, 2012.