FORMAL MODELING AND DYNAMIC VERIFICATION OF SERVICE LEVEL AGREEMENTS IN CLOUD COMPUTING

BY

ABDEL-RAHMAN F. AL-GHUWAIRI, B.S., M.S.

A dissertation submitted to the Graduate School

in partial fulfillment of the requirements

for the degree

Doctor of Philosophy

Subject: Computer Science

New Mexico State University

Las Cruces, New Mexico

February 2013

UMI Number: 3574545

"Formal Modeling and Dynamic Verification of Service Level Agreements in Cloud Computing," a dissertation prepared by Abdel-Rahman F. Al-Ghuwairi, B.S., M.S. in partial fulfillment of the requirements for the degree, Doctor of Philosophy, has been approved and accepted by the following:

Linda Lacey

Linda Lacey
Dean of the Graduate School

Jonathan C. Cook

Jonathan Cook
Chair of the Examining Committee

May 6, 2013

Date

Committee in charge:

Dr. Jonathan Cook, Chair

Dr. Muhammad Dawood

Dr. Satyajayant Misra

Dr. Inna Pivkina

DEDICATION

To my parents, my wife, and my kids.

ACKNOWLEDGMENTS

In the first place, many thanks to New Mexico State University and the Computer Science Department for giving me the opportunity to pursue my higher education and accomplish the PhD degree in computer science. The main person in this university who I greatly appreciate and thank is my advisor *Dr. Jonathan Cook* for his outstanding help and support in my PhD study and research. My parents had a hard time waiting for this great moment, this work would have never be done and accomplished without their prayers and support. I hope the best for them and may God protect and keep them in good health. Also, I thank my wife and kids for their patience, support, and sacrifice sharing me the effort to achieve this degree. Last but not least, I thank my dissertation committee for their participation and great effort.

VITA

December 1, 1968	Born at Berein, Zarqa, Jordan.
1986	Graduated from Zarqa Secondary School, Zarqa, Jordan.
1986-1990	B.S., Mu'tah University, Karak, Jordan.
1990-2004	Instructor, Programmer, and System Analyst, Royal Jordanian Air Force, Amman, Jordan.
2004-2006	M.S., The University of Jordan, Amman, Jordan.
2006-2008	Teaching Assistant and Instructor, College of Information Technology, Hashemite University, Zarqa, Jordan.
2008-2012	PhD., New Mexico State University, Las Cruces, New Mexico.

FIELD OF STUDY

Major Field: Computer Science.

ABSTRACT

FORMAL MODELING AND DYNAMIC VERIFICATION OF SERVICE LEVEL AGREEMENTS IN CLOUD COMPUTING

BY

ABDEL-RAHMAN F. AL-GHUWAIRI, B.S., M.S.

Doctor of Philosophy

New Mexico State University

Las Cruces, New Mexico, 2013

Dr. Jonathan Cook, Chair

A Service Level Agreement (SLA) defines the contract between a cloud provider and a cloud customer, detailing the resources being provided, the price the user will pay, and the Quality of Service (QoS) guarantees that the cloud provider ensures for the customer. If the QoS guarantees are not upheld, typically the cloud provider is assessed some penalties, such as a payment credit for the customer. Monitoring and enforcing the SLA is an area of open research, and in this research we present the foundations towards a full realization of an SLA monitoring infrastructure. We proposed a formal model that can precisely describe both the SLA QoS guarantees and the penalties assessed for violation. Also, we implemented the proposed model as a dynamic SLA monitoring infras-

tructure system which is used for automatic SLA enforcement efficiently. Our SLA monitoring system can be deployed in any cloud environment and it can be used to monitor any type of QoS parameters. The performance and efficiency of the proposed SLA framework was measured and proved that our SLA model is efficient and reliable.

TABLE OF CONTENTS

LIST OF TABLES . . . xiii

LIST OF FIGURES . . . xiv

1 INTRODUCTION . . . 1

2 CLOUD COMPUTING . . . 4

2.1 Cloud Computing History . . . 4

2.2 Cloud Computing Fundamental Models . . . 7

2.3 Benefits and Liabilities of Cloud Computing . . . 9

2.4 Migrating to the Cloud Platform: Issues . . . 12

3 MONITORING IN CLOUD COMPUTING . . . 17

3.1 Objectives and Goals of Cloud Monitoring . . . 17

3.2 Monitoring for Novel Purposes . . . 18

3.3 Commercial Efforts in Cloud Monitoring . . . 19

3.4 Open Issues in Cloud Computing Monitoring . . . 24

4 MONITORING FOR SLA VERIFICATION . . . 26

4.1 Overview . . . 26

4.2 IaaS Cloud Service Level Agreements . . . 29

4.2.1 Service Level Agreement Overview . . . 29

4.3 Issues Covered by Quality of Service Guarantees . . . 30

4.4 Service Level Agreement Examples . . . 31

5 MODELING AND VERIFICATION OF AN SLA . . . 38

5.1 Thesis Statement . . . 38

5.2 Monitoring and Modeling SLAs . . . 40

5.2.1 SLA Monitoring Framework 40
5.2.2 The proposed SLA Model 43
5.2.3 The Penalty Function 48
5.3 Example of the Proposed Model 51
5.3.1 Proposed Mapping Rules for an SLA 55
5.3.2 Real-World SLAs Examples 58
5.4 SLA Related Work 60
5.5 Agent-Based Verification of an SLA 62
5.6 Implementing an SLA using Policy Description Language (PDL) . 68
5.6.1 SLA Implementation Example 70
5.7 Proposed Implementation using PDL and oClingo 74
5.7.1 Why is oClingo not Used in the Implementation? 75
6 IMPLEMENTATION AND VERIFICATION OF AN SLA 78
6.1 Overview 78
6.2 Implementation Language and Database 79
6.2.1 Implementation Language 80
6.2.2 Database 81
6.3 SLA Monitoring System 83
6.3.1 SLA Monitoring System Design 83
6.3.1.1 Sockets and Multi-Threaded Programming . . . 84
6.3.2 SLA QoS Parameters 85
6.3.3 Infrastructure and Network Monitoring System (Instrumentation) 86

6.3.4 Network Availability . . . 88
6.3.5 Network Response Time . . . 90
6.3.6 Server (Virtual Machine) Availability . . . 91
6.3.7 CPU Performance . . . 93
6.4 SLA Penalty Monitor System . . . 93
6.4.1 How it Works . . . 94
6.4.2 SLA Monitoring Sytem Properties . . . 96
6.4.2.1 Dynamic . . . 97
6.4.2.2 Automatic . . . 97
6.4.2.3 Agent-Based . . . 97
6.4.2.4 High Efficiency . . . 98
6.4.2.5 High Availability . . . 98
6.4.2.6 Low Overhead . . . 98
6.4.2.7 Portability . . . 99
6.4.2.8 Flexibility . . . 99
6.4.2.9 High Accuracy . . . 99
6.4.2.10 Privacy . . . 100
6.4.2.11 Security . . . 100
6.4.2.12 Accessibility . . . 100
7 EXPERIMENT AND RESULTS . . . 101
7.1 Overview . . . 101
7.2 Experimental Hypothesis . . . 101
7.3 Experiment Requirements . . . 102

7.3.1 Hardware Requirements 102

7.3.2 Software Requirements 102

7.4 SLA QoS Parameters 103

7.5 Deployed Cloud Application 104

7.5.1 Light Load of Cloud Application 104

7.5.2 Heavy Load of Cloud Application 104

7.5.3 Performance Metrics to Measure 104

7.6 Experimental Types and Constraints 107

7.6.1 Experiment 1 108

7.6.2 Experiment 2 109

7.6.3 Experiment 3 109

7.6.4 Experiment 4 110

7.6.5 Experiment 5 111

7.6.6 Experiment 6 112

7.7 Experimental analysis and Results 114

7.7.1 Analysis by Performance Metric 117

7.7.1.1 Memory Usage 117

7.7.1.2 CPU Utilization 119

7.7.1.3 System Load Average 122

7.7.1.4 Input/Output Statistic 123

7.7.1.5 Network Statistic 126

7.7.1.6 Bandwidth Utilization 128

7.8 Statistical Analysis 130

7.8.1 First ANOVA Test 130
7.8.2 Human Gap Time 134
7.8.3 Second ANOVA Test 135
7.9 Efficiency and Performance Measurements 139
8 CONCLUSION 141
8.1 Future Work 144
REFERENCES 146

LIST OF TABLES

4.1 Rackspace SLA example 35

4.2 SLA of Amazon S3 36

4.3 SLA of Google App 36

5.1 SLA example 52

5.2 SLA summarized example 53

5.3 Example of mapping rules for the measurement functions 56

5.4 Example of mapping rules for the penalty functions 57

5.5 Rackspace SLA example 59

6.1 SLA Contract form example 86

7.1 Performance Commands 107

7.2 Experimental Types and Constraints 108

7.3 Used Memory Utilization 117

7.4 CPU Utilization 120

7.5 System Load Average 122

7.6 Input/Output Overhead 124

7.7 Network Traffic Statistic 126

7.8 BW Utilization 128

7.9 P-Vlaues for the first ANOVA test 131

7.10 P-Vlaues for the second ANOVA test 137

LIST OF FIGURES

5.1 SLA Monitoring Architecture. 41

5.2 SLA Domain Model. 44

5.3 Penalty Function 50

6.1 SLA Monitoring Dseign Architecture. 79

7.1 Used Memory Overhead. 119

7.2 CPU Utilization Overhead. 121

7.3 System Load Average Overhead. 123

7.4 Input/Output Overhead. 125

7.5 Network Traffic Overhead. 127

7.6 Bandwidth Overhead. 129

CHAPTER 1

INTRODUCTION

There are many definitions for cloud computing. One definition states that cloud computing is a model of on-demand network access to shared computing resources (networks, storage, servers, services, and applications) that can be released with a minimal effort and a reasonable performance [34]. Another definition claims that cloud computing is similar to the terms grid computing, utility computing, virtualization, and clustering [40]. Cloud fundamentals [62] defines cloud computing as an Internet based computing solution by using shared resources to deliver IT as a service. Computers are working together and all users, wherever they are, use the same computing power as if they are running on a single computer [62]. The term "cloud computing" probably came from the cloud diagram that represents the Internet in text books [1].

Revolution can be defined as a change in the way people think and behave. The same thing happened when using cloud computing and it is considered a revolution in the IT sector. Cloud computing creates a major and fundamental change in computer architecture, software tools, development, storage, and how to distribute and consume information [62].

Cloud computing monitoring is considered a very important issue for both parties who are involve in the process: the user and the provider of the cloud. Because of the on-line nature of cloud computing, the customer and the provider cannot see each other and build a normal business relationship based on the trust and privacy. Therefore, a need for a contract between those two parties arises and become a very important and essential part of the service. This contract is called a Service Level Agreement (SLA), where the two parties agree upon the Quality of Service parameters or guarantees (QoS) that will be maintained during the whole service.

We propose a Quality of Service model (QoS) that defines a formal Service Level Agreement (SLA). In this SLA the cloud parties can include any parameter they want and its corresponding value in addition to the penalty type and amount. The formal definition of this SLA is designed in a way to capture most aspects, parameters, and constraints that exist in real world SLAs. It can be used to cover all services and attributes they wish to monitor which includes a mapping rule to evaluate each parameter or attribute included in the SLA. In case of any violation, the penalty can be calculated easily by following the formal definition and the mapping rules of the SLA.

The proposed SLA model consists of tow main parts: the infrastructure monitoring system (instruments) and the SLA penalty monitor. This model was implemented using Java programming language to enforce the agreement efficiently and effectively. Also, MySQL database was used to store the monitoring, violation, and penalty data.

The proposed and implemented SLA monitoring framework can be used for dynamic, automatic, and efficient verification within the deployed cloud environment. To measure the performance and the efficiency of the SLA monitoring system, many experiments with different constraints were performed. As a result, the performance and the efficiency measurements of the proposed model give a good indication that the overall system is efficient and reliable.

This dissertation is organized as follows: Chapter 2 presents background in cloud computing and Chapter 3 presents issues and commercial efforts in cloud computing monitoring. Chapter 4 addresses background in the Service Level Agreement (SLA) monitoring. In Chapter 5 we present our formal SLA model and monitoring ideas with examples. Chapter 6 elucidates the implementation and the dynamic verification of our SLA model. Chapter 7 describes the SLA monitoring framework experiments and results with analysis, and finally Chapter 8 concludes this research with directions for the future of this work.

CHAPTER 2

CLOUD COMPUTING

2.1 Cloud Computing History

The cloud computing model has many characteristics: on-demand services, mobile network access, rapid deployment, minimal management effort, and high availability services without human interaction [1]. The user can access the cloud through any portable device (laptops, mobile phones, and PDAs). On the other hand, the provider's resources are pooled to serve many users in a multi-tenant technique. The user does not know the exact location of these resources; however, the consumer feels that the capabilities are unlimited and they can have rapid access to the cloud.

Cloud computing is spreading quickly these days and it is not considered new in the IT field. The idea of computing technology started in the 1960s, at a time when John McCarthy wrote that "computation may someday be organized as a public utility" [1]. In the early 1960s, grid computing started to make computer power as easy to access as electric power. Telecommunications companies started the idea by making a radical shift from point-to-point data circuits to Virtual Private Network (VPN) services in the 1990s [1]. To make their work efficient and inexpensive, they used load balancing to optimize resource utilization. 1997 was the first year when the term was started in its current context by Ramnath Chellappa, where he defined it as a new "computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone [1]."

As stated by [1], In 1999, Salesforce.com was one of the first companies that used cloud computing for delivering applications via a simple website. Amazon Web Services had been launched in 2002, and in 2006 Google Docs put cloud computing in the front of the IT sectors. Amazon Elastic Compute Cloud (EC2) also became a commercial web service in 2006, so small companies and individuals can rent and run their applications there. In 2007, collaboration began between Google, IBM and a number of universities in the United States in the field of Cloud Computing. In 2008, Eucalyptus appeared as the first open source platform for deploying the private clouds. After that Open Nebula was established, which is considered the first open source software for deploying private and hybrid clouds.

CloudTweaks [1], addressed that Microsoft entered cloud computing in 2009 with the launch of Windows Azure. Following Microsoft, Oracle, Dell, Fujitsu, Teradata, HP, and many other companies entered the field of Cloud Computing. According to recent reports, businesses in the United States will spend more than 13 billion dollars on cloud computing by 2014 in comparison to 3 billion dollars at present. Cloud computing takes into consideration the achievements done in cyber infrastructure and builds from there to make the applications easier to develop and deploy, and to reduce the cost of ownership. On the other hand, cyber infrastructure lets the scientists concentrate on scientific research rather than information technology development [61].

Computing applications in the past was nearly impossible without using the installation of suitable software locally in the user's computer. As an example, word processing or any other application needs to be installed in the computer to work. Also a license needs to be bought for each application to have the right to install. After the Local Area Network (LAN) and the concept of client-server model was used, computing became easier and more reliable than before. Servers with high capabilities and large storage could be used to host applications and data for a large working network. In the client-server model, the client version of the application needs to be installed on client computer to utilize the client memory and CPU for processing, while the data will be stored on the server. Also, the need for multiple licenses of an application still exists [40]. In Cloud Computing, there is no need to install the client version of the application; all we need is a server with applications installed on it and we can execute and manage the applications through the client's web browser [34].

2.2 Cloud Computing Fundamental Models

Cloud computing models are separated into service models and deployment models [51]. The service models are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The deployment models are private cloud, which functions only for an organization owned or leased; community cloud, which is a shared infrastructure for specific community or several organizations; public cloud, which is available to public or large organizations group; and hybrid cloud, which is a composite of two or more clouds [28].

The main concept of the cloud is that it is built on layers, and each layer is responsible for a specific functionality. A virtual machine (VM) is a separated operating system under a hosting operating system with the ability to act and behave as a physical machine [11]. The virtual machine monitor (VMM) is a program located in a host machine that lets one computer support multiple and identical execution environments. In cloud computing, the VMM lets users monitor and manage the data access of the process, data storage, encryption, addressing, topology, and workload movement [62].

There are three layers provided by the cloud: Infrastructure as a service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The IaaS contains the physical components of the cloud such as servers, network devices, and storage disks. One of the benefits of this layer is Print On Demand (POD) service, which is based on allowing the users to open shops and sell designs on products, and also provides unlimited storage [62].

The middle layer is PaaS, which is responsible for providing development platform for developers, so the users can write their code and it can be uploaded into the web by the PaaS providers [1]. Amazon Elastic Compute Cloud (EC2)is considered an example of PaaS providers [62]. In PaaS, the consumers build their applications that run on the provider's infrastructure and then will be delivered to the consumers' users from the provider's servers via the Internet. The provider constraints and capabilities are key factors which affect these services and the consumers do not have complete freedom but they have predictability and pre-integration. Examples of PaaS are: Salesforce.com, Coghead, Google App Engine, and Yahoo Pipes [45].

The top layer is SaaS, or the application layer where the applications run and are provided to the users on demand [62]. In SaaS, cloud computing delivers a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, there is no need to have servers or software licensing; on the provider side, costs are low because there is one application to maintain, while cost will be higher in conventional hosting. The best example for SaaS is Salesforce.com; it is also is common in HR applications, desktop applications like Google applications, and Zoho Office [45]. There are several providers for the SaaS, such as Google Pack (which includes Internet applications), tools such as Calendar, Gmail, Google Talk, and Docs [62].

2.3 Benefits and Liabilities of Cloud Computing

Cloud computing is online computing where users can access applications through a browser, where the applications and the data are installed on a server. In this way, users from all over the world are allowed to access something without having to download or install anything on their own computers. The main risk and challenge is security, though many companies use it such as Amazon, Yahoo, Google, Zoho, Microsoft, and Salesforce.com [19].

There are many benefits for cloud computing: cost reduction for both users and websites owners, power saving, increased agility in software deployment, easy access to users files from any computer, and the option of renting the server space instead of buying it. Cloud computing also provide more storage that can be used in reducing the need for upgrading computer memory and minimizing the cost. Another benefit of cloud computing is that owners no longer need to have more employees to update more than one server: it also helps the users who use the services to get all updates without do anything on their own computers. Flexibility of computing is much better than other network computing systems and it saves time and money. Mobility is also a benefit of cloud computing, as most networks allow users to connect from anywhere and at any time. No downloads are needed from the user and this will save money and storage space [19].