UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUILHERME DA CUNHA RODRIGUES

Unveiling The Interplay Between Timeliness and Scalability in Cloud Monitoring Systems

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Lisandro Zambenedetti Granville
Coadvisor: Profa. Dra. Liane Margarida Rockenbach Tarouco

Porto Alegre
June 2016

CIP — CATALOGING-IN-PUBLICATION

Rodrigues, Guilherme da Cunha

Unveiling The Interplay Between Timeliness and Scalability in Cloud Monitoring Systems / Guilherme da Cunha Rodrigues. – Porto Alegre: PPGC da UFRGS, 2016.

103 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2016. Advisor: Lisandro Zambenedetti Granville; Coadvisor: Liane Margarida Rockenbach Tarouco.

1. Monitoring. 2. Management. 3. Cloud computing. 4. Cloud monitoring requirements. 5. Timeliness. 6. Scalability. I. Granville, Lisandro Zambenedetti. II. Tarouco, Liane Margarida Rockenbach . III. Título.

To my family

“We don’t matter where we go because we always find the way back home.”

— Unknown Author

AGRADECIMENTOS

A Deus pela vida e saúde.

A minha família pelo amor e apoio incondicional.

Aos professores Lisandro Zambenedetti Granville e Liane Margarida Rockenbach Tarouco pela oportunidade de integrar um grupo de excelência e pela paciência e dedicação no processo de orientação.

Ao amigo Rodrigo Neves Calheiros pelo suporte técnico e pelo exemplo profissional e humano.

Ao professor Rajkumar Buyya pela oportunidade de compartilhar conhecimentos junto a um grupo de referência internacional.

Aos amigos Fábio Diniz Rossi, Vinícius Tavares Guimarães e Glederson Lessa dos Santos pelas inúmeras contribuições tanto no tocante a questões técnicas quando de ordem pessoal.

Aos colegas Roberto Irajá Tavares da Costa Filho, Marcelo Caggiani Luizelli, Weverton Luis da Costa Cordeiro, Rafael Hansen da Silva, Gustavo Miotto, Leonardo Richter Bays, Rafael Pereira Esteves, Jeferson Campos Nobre, Carlos Raniery Paula dos Santos, Flávio Roberto Santos e Márcio Barbosa de Carvalho pelo agradavel convívio diário, etc.

ABSTRACT

Cloud computing is a suitable solution for professionals, companies, research centres, and institutions that need to have access to computational resources on demand. Nowadays, clouds have to rely on proper management of its structure to provide such computational resources with adequate quality of service, which is established by Service Level Agreements (SLAs), to customers. In this context, cloud monitoring is a critical management function to achieve it.

Cloud monitoring requirements are properties that a cloud monitoring system need to meet to perform its functions properly, and currently there are several of them such as timeliness, elasticity and scalability. However, such requirements usually have mutual influence, which is either positive or negative, among themselves, and it has prevented the development of complete cloud monitoring solutions.

From the above, this thesis investigates the mutual influence between timeliness and scalability. This thesis proposes a mathematical model to estimate such mutual influence to enhance cloud monitoring systems. The methodology used in this thesis is based on monitoring parameters such as monitoring topologies, the amount of monitoring data, and frequency sampling. Besides, it considers as important metrics network bandwidth and response time. Finally, the evaluation is based on a comparison of the mathematical model results and outcomes obtained via simulation.

The main contributions of this thesis are divided into two axes, namely, basic and key. Basic contributions of this thesis are: (i) it discusses the cloud monitoring structure and introduced the concept of cloud monitoring focus (ii) it examines the concept of cloud monitoring requirement and proposed to divide them into two groups defined as cloud monitoring requirements and cloud monitoring abilities (iii) it analysed challenges and trends in cloud monitoring pointing research gaps that include the mutual influence between cloud monitoring requirements which is core to the key contributions. The key contributions of this thesis are: (i) it presents a discussion of timeliness and scalability that include: the methods currently used to cope with the mutual influence between them, and the relation between such requirements and monitoring parameters (ii) it identifies the monitoring parameters that are essential in the relation between timeliness and scalability (iii) it proposes a mathematical model based on monitoring parameters to estimate the mutual influence between timeliness and scalability.

Keywords: Monitoring. Management. Cloud computing. Cloud monitoring requirements. Timeliness. Scalability.

Desvelando a relação mútua entre escalabilidade e oportunidade em sistemas de monitoramento de nuvens computacionais

RESUMO

Computação em nuvem é uma solução adequada para profissionais, empresas, centros de pesquisa e instituições que necessitam de acesso a recursos computacionais sob demanda. Atualmente, nuvens computacionais confiam no gerenciamento de sua estrutura para fornecer recursos computacionais com qualidade de serviço adequada as expectativas de seus clientes, tal qualidade de serviço é estabelecida através de acordos de nível de serviço. Nesse contexto, o monitoramento é uma função crítica de gerenciamento para se prover tal qualidade de serviço. Requisitos de monitoramento em nuvens computacionais são propriedades que um sistema de monitoramento de nuvem precisa reunir para executar suas funções de modo adequado e atualmente existem diversos requisitos definidos pela literatura, tais como: oportunidade, elasticidade e escalabilidade. Entretanto, tais requisitos geralmente possuem influência mútua entre eles, que pode ser positiva ou negativa, e isso impossibilita o desenvolvimento de soluções de monitoramento completas.

Dado o cenario descrito acima, essa tese tem como objetivo investigar a influência mútua entre escalabilidade e oportunidade. Especificamente, essa tese propõe um modelo matemático para estimar a influência mútua entre tais requisitos de monitoramento. A metodologia utilizada por essa tese para construir tal modelo matemático baseia-se em parâmetros de monitoramento tais como: topologia de monitoramento, quantidade de dados de monitoramento e frequencia de amostragem. Além destes, a largura de banda de rede e o tempo de resposta também são importantes métricas do modelo matemático. A avaliação dos resultados obtidos foi realizada através da comparação entre os resultados do modelo matemático e de uma simulação.

As maiores contribuições dessa tese são divididas em dois eixos, estes são denominados: Básico e Chave. As contribuições do eixo básico são: (i) a discussão a respeito da estrutura de monitoramento de nuvem e introdução do conceito de foco de monitoramento (ii) o exame do conceito de requisito de monitoramento e a proposição do conceito de abilidade de monitoramento (iii) a análise dos desafios e tendências a respeito de monitoramento de nuvens computacionais. As contribuições do eixo chave são: (i) a discussão a respeito de oportunidade e escalabilidade incluindo métodos para lidar com a mútua influência entre tais requisitos e a relação desses requisitos com parâmetros de monitoramento (ii) a identificação dos parâmetros de monitoramento que são essenciais na relação entre oportunidade e escalabilidade (iii) a proposição de um modelo matemático baseado em parâmetros de monitoramento que visa estimar a relação mútua entre oportunidade e escalabilidade.

Palavras-chave: Monitoramento, Gerenciamento, Computação em Nuvem, Requisitos de monitoramento em computação em nuvem, Oportunidade, Escalabilidade.

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
APP	Application
AWS	Amazon Web Services
CPU	Central Processing Unit
EC2	Amazon Elastic Computing Cloud
CMS	Cloud Management System
CO	Cloud Operator
CP	Cloud Provider
CPs	Cloud Providers
CSP	Cloud Service Provider
FlexACMS	Flexible Automated Cloud Monitoring Slices
IaaS	Infrastructure as a Service
ICT	Information and Communication Technologies
InP	Infrastructure Provider
InPs	Infrastructure Providers
GB	Gigabyte
GBps	Gigabyte Per Second
GHz	Giga-hertz
IP	Internet Protocol
KVM	Kernel Based Virtual Machine
MaaS	Monitoring as a Service
MRTG	Multi Router Traffic Grapher
MS	Milliseconds
NIST	National Institute of Standards and Technology

PaaS	Platform as a Service
PCMONS	Private Cloud Monitoring System
QoS	Quality of Service
RAM	Random Access Memory
REST	Representational State Transfer
RMCM	Runtime Model for Cloud Monitoring
RRDToll	Round-Robin Database Tool
RT	Response Time
RVision	Remote Vision
SaaS	Software as a Service
SLA	Service Level Agreement
SLAs	Service Level Agreements
SP	Service Provider
SPs	Service Providers
TCP	Transmission Control Protocol
VM	Virtual Machine
VMs	Virtual Machines

LIST OF SYMBOLS

a_{symb}	Amount of monitoring data
ac_{symb}	Amount of channels
b_{symb}	Network bandwidth
fs_{symb}	Frequency sampling
h_{symb}	Amount of hosts in a same level
$l2_{symb}$	Amount of hosts in the level 3
rt_{symb}	Response time

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Cloud computing is a groundbreaking solution to acquire computational resources on demand (BUYYA et al., 2009) (ARMBRUST et al., 2010). Nowadays, big companies are following this breakthrough solution to increase their gains. For example, Amazon and Microsoft introduced profitable cloud systems such as Amazon EC2 (Amazon, 2015b) and Azure (Microsoft, 2015). From the customer point of view, cloud computing has been standing out for some features such as reduced costs, accessibility, and flexibility (MARSTON et al., 2011) (STANOEVSKA-SLABEVA; WOZNIAK; RISTOL, 2009).

According to NIST (National Institute of Standards and Technology) (MELL; GRANCE, 2011), there are five essential *characteristics* of a cloud, namely, on demand self-service, broad network access, resources pooling, rapid elasticity, and measured service. This work is focused on measure service. However, all essential characteristics are described in Chapter 2 to provide a broad vision of cloud computing.

To NIST, measured services define that "*a cloud system automatically controls resource usage by leveraging a metering capability at some level of abstraction appropriate to the type of service provided (*e.g., *processing, bandwidth, active user accounts). Thus, resource usage can be monitored, controlled, and reported, providing transparency for both the provider and customer of the utilized service*". Measured service is one of the essential characteristics that provides support to cloud operators to deliver high-quality cloud services, usually based on Service Level Agreements (SLA's), to customers. To perform measured service, a cloud, like any other computational system, needs to be properly managed in accordance with its characteristics (*e.g.,* scalability, elasticity, availability). In this context, cloud monitoring becomes an important issue because it provides to cloud operators (*e.g.,* service providers, infrastructure providers) means to manage (*e.g.,* analyse, control) a cloud computing environment (RODRIGUES et al., 2012).

Cloud monitoring serves as support to management activities. It presents information from multiple resources (*e.g.,* network, processing) and services (*e.g.,* analysis, notifications), enabling cloud operators to perform control activities, which allows for the cloud to offer predictable performance to customers (Amazon, 2015a) (CHAVES; URIARTE; WESTPHALL, 2011). Cloud monitoring systems need to meet several requirements/abilities such as scalability, comprehensiveness, adaptability, accuracy, elasticity, and timeliness (ACETO et al., 2013) (CLAYMAN et al., 2010).

When a cloud monitoring system tries to accomplish a specific requirement, another requirement is either negative or positive affected (ACETO et al., 2013). Thus, the development of cloud monitoring systems is focused on specific characteristics of a cloud, providing only partial solutions for cloud monitoring (MONTES et al., 2013). Besides, the lack of acquaintance about the exact influence among cloud monitoring requirements restricts the capacity of integrating cloud monitoring systems. Therefore, the balance among cloud monitoring requirements becomes a challenging and significant task to enhance cloud monitoring systems. To solve this problem, first there are a necessity to understand the relationship between different cloud monitoring requirements, *i.e.,* how much a specific cloud monitoring requirement influence others.

Scalability and Timeliness are important requirements because scalability allows a cloud to grow to cope with the necessity of more resources, and timeliness is closely related to cloud providers profits. Besides, regardless the trade-off between timeliness and scalability, they have a direct influence on all other requirements/abilities. For example, scalability and timeliness are closely related to requirements/abilities such as elasticity, accuracy, and adaptability (PARK et al., 2011) (ACETO et al., 2013) (RODRIGUES et al., 2014). For this reason, the trade-off between scalability and timeliness is a meaningful point to start the evaluation of the mutual influence of cloud monitoring requirements.

This thesis aims to investigate the trade-off between scalability and timeliness to propose a mathematical model to predict the mutual influence between both requirements in cloud monitoring systems. The awareness about such mutual influence may be used to improve clouds on issues such as SLAs, QoS (Quality of Service), and increase profits to cloud operators (*e.g.,* Cloud Providers (CPs), Infrastructure Providers (InPs)) as a consequence.

1.1 Hypothesis and Fundamental Questions

This thesis investigates how scalability has influenced timeliness and conversely. It proposes a mathematical model to estimates such influence in order to enhance cloud monitoring systems. Aiming to help in this investigation, it is defended the hypothesis as follows.

Hypothesis: **"How to provide mechanisms to predict the mutual influence between timeliness and scalability in cloud monitoring systems aiming to define accurate SLAs and increase profits to cloud operators."**

To predict such influence, this thesis proposes a mathematical model based on monitoring parameters such as the amount of monitoring data, and frequency sampling. To achieve this goal, the investigation follows fundamental questions related to the hypothesis. We defined such questions below.

- What are the requirements and abilities introduced by cloud monitoring systems? What are the differences between requirements and abilities?
- What are the monitoring parameters that must be considered to evaluate the influence of scalability over timeliness? What is the metric that must be used to assess such influence?
- Could cloud monitoring systems have the ability to predict the influence of scalability over timeliness and vice-versa, based on a mathematical model? Is it a suitable solution?

1.2 Goals

This thesis aims to propose a mathematical model to predict the mutual influence between cloud monitoring requirements, namely, scalability and timeliness. To address it, we consider as input metrics the monitoring parameters such as frequency sampling. The specific contributions expected as the outcome of this investigation are listed as follows.

- This thesis aims to distinguish the concepts of cloud monitoring requirements and cloud monitoring abilities;
- This thesis aims to define key monitoring parameters to evaluate the influence of scalability over timeliness;
- This thesis aims to demonstrate that the mutual influence between timeliness and scalability is liable to be mathematically modelled.

1.3 Organization

The remainder of this thesis is organized as follows.

- Chapter 2 firstly presents an overview of cloud computing. The most significant concepts related to cloud computing such as service models, deployment models, and essential characteristics are detailed. Secondly, this chapter provides a survey of the cloud monitoring. It discusses cloud monitoring structure, requirements, abilities, and solutions. As a complement are discussed the challenges and trends in cloud monitoring. After that, this chapter shows the most common network topologies for clouds. It presents a summary that includes some significant topologies to clouds such as conventional data center network, clos, and fat-tree.

- Chapter 3 starts presenting the related work. After that, it provides an extensive study of the interplay between cloud monitoring requirements, namely, scalability and timeliness. Next, it identifies essential monitoring parameters to evaluate the mutual influence between timeliness and scalability. Finally, Chapter 3 presents a simulation to demonstrate the influence of monitoring parameters over timeliness and scalability using response time as output metric.

- Chapter 4 demonstrates that the mutual influence between timeliness and scalability is liable to be mathematically modelled based on monitoring parameters. Next, it presents a mathematical model to predict such mutual influence. After that, it compares the results obtained via simulation, performed in Chapter 3, with results provided by the mathematical model to evaluate the effectiveness of the proposed solution to monitoring topologies based on conventional data center networks. Besides, it compares the outcomes obtained via simulation with results provided by the mathematical model to topologies based on fat-tree aiming to stretching the evaluation. Finally, it discusses the applicability of the mathematical model.

- Chapter 5 describes the conclusions and future works related to this thesis. Firstly, the answers to the fundamental questions are exposed and justified. Next, the main contributions of this thesis are listed. Finally, directions for future work are identified and detailed.

2 STATE-OF-THE-ART

The goal of this chapter is to present the background of the three fundamental research topics in this thesis, namely, cloud computing, cloud monitoring, and network topologies for cloud computing. It starts introducing the fundamental concepts that currently have supported cloud computing. Thus, service models are presented in details as well as deployment models. After that, essential characteristics of clouds are discussed. In a second moment, it is focusing on cloud monitoring fundamentals. It discusses the significance of monitoring to clouds, and it shows an example of cloud monitoring structure. It introduces the concept of cloud monitoring abilities and discusses it along with cloud monitoring requirements. Next, it presents the current research status of this area, followed up by a discussion about challenges and trends in cloud monitoring. Finally, network topologies for clouds are presented. It includes significant topologies for clouds such as conventional data center network, clos, and fat-tree.

2.1 Cloud Computing

Cloud computing has become an important and groundbreaking paradigm because it efficiently reduces costs of ICT (Information and Communication Technologies) infrastructures by offering computer resources as services (VAQUERO et al., 2008). Companies, universities, and governments, for example, can enjoy high availability and scalability by leasing cloud services offered by providers, with reduced prices if compared to traditional ICT models.

Cloud computing has some different definitions. Buyya defines it *"as a parallel and distributed system composed of clustered virtual machines interconnected which are allocated dynamically and presented as an unified system of computing resources based on SLA's established through business trading between service providers and clients"* (BUYYA et al., 2009). On the other hand, Foster understands that *"the paradigm of cloud computing is not essentially a new concept, but the result of a merge among different paradigms such as grids, and clusters"* (FOSTER et al., 2008). In this scenario, one of the most relevant definitions to cloud computing is presented by NIST (National Institute of Standards and Technology) (MELL; GRANCE, 2011).

According to NIST, cloud computing is a model composed of three service models, four deployment models, and five essential characteristics (MELL; GRANCE, 2011). Figure 2.1 depicts all of them as required support to clouds. In the next sections, service models, deployment models and essential characteristics are explained in details.

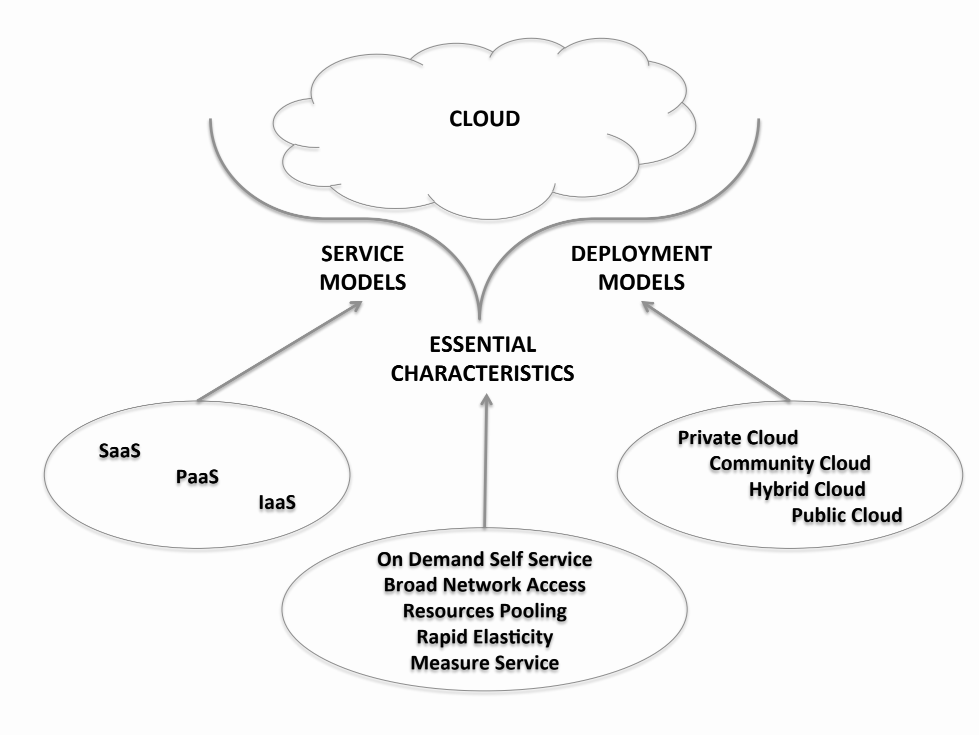


Figure 2.1: Cloud model

2.1.1 Service Models

Cloud computing have provided different levels of computer services. Such levels are defined as service models, namely, SaaS (*Software as a Service*), PaaS (*Platform as a Service*), and IaaS (*Infrastructure as a Service*) (VAQUERO et al., 2008) (MELL; GRANCE, 2011).

- SaaS: The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (*e.g.,* web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user- specific application configuration settings.

- PaaS: The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

- IaaS: The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (*e.g.,* host firewalls).

2.1.2 Deployment Models

Cloud computing have introduced different methods to deploy its resources. This methods varies from clouds infrastructures managed by a single organization to hybrid compositions of two or more organizations as described below (ARMBRUST et al., 2010) (MELL; GRANCE, 2011):

- Private cloud: The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (*e.g.,* business unit). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premisses.

- Community cloud: The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (*e.g.,* mission, policy). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premisses.

- Public cloud: The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed and operated by a business, academic, or government organization, or some combination of them. It exists on the premisses of the cloud provider.

- Hybrid cloud: The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (*e.g.,* cloud bursting for load balancing between clouds).

2.1.3 Essential Characteristics

A cloud computing environment can have a plenty of characteristics such as scalability, and reliability. Such characteristics are implemented depending on goals of cloud providers or users. For example, a goal to reduce costs or a goal to improve performance. However, to be defined as a cloud, according to NIST it must have five essential characteristics, namely, On-demand self-service, Broad network access, Resource pooling, Rapid elasticity, and Measured service (MELL; GRANCE, 2011).

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (*e.g.,* mobile phones, tablets, laptops, and workstations).
- Resource pooling: The resources of a cloud provider are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (*e.g.,* country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
- Rapid elasticity: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (*e.g.,* storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Considering the cloud composition based on such service models, deployment models and essential characteristics. Monitoring as a crucial part of an essential characteristic (*i.e.,* measured service) plays a key role to improve clouds in several ways. In this context, service models are the basis for cloud monitoring because each service model has its singular characteristics, as will be shown in Section 2.2.2. Such characteristics must have be considered by a cloud monitoring system that intend to oversee each service model.

In the next section are presented some vital concepts related to cloud monitoring such as cloud monitoring structure, cloud monitoring requirements and abilities. Also, it discusses some relevant cloud monitoring solutions along with current challenges and trends in this area.

2.2 Cloud Monitoring

Usually, a cloud has a large number of computational resources on data centers that are geographically spread. Such resources must be continuously monitored, since cloud computing entities (*e.g.,* SPs, InPs) need information related to these resources, mainly for two reasons. Firstly, to evaluate the current status of services hosted in the cloud. Secondly, to use information about resources to perform control activities (*e.g.,* allocation, migration, provisioning).

In general, cloud services are offered in different service models and are composed of different types of resources (*e.g.,* processing, network, storage). The effective management of a cloud depends on complete monitoring of its structure. In order to reach such effective management, cloud monitoring systems have to address features of a cloud through its requirements and abilities based on a suitable cloud monitoring structure.

In the next sections are presented the significance of monitoring to clouds, a cloud monitoring structure, cloud monitoring requirements and abilities along with a literature review based on cloud monitoring solutions. Finally, challenges and trends in cloud monitoring are presented and discussed.

2.2.1 Monitoring: Why is it so important to clouds?

Monitoring is an important function of clouds because it is crucial for cloud operators (*e.g.,* SPs, InPS) and customers. To cloud operators, monitoring is a fundamental function to control and manage its infrastructure and services. To customers, it provides information that supports issues such as resources usage control, and QoS oversight (NGUYEN et al., 2014). Besides, monitoring is significant to the commercial relation between cloud operators and customers because it is an essential function to guarantee the trustworthiness among them through measurements of resources, performance, SLAs and so on. In other words, monitoring allows to cloud operators and customers to have access to information which is key to their commercial relation as well as to support both of them on vital issues.

Monitoring is also significant to clouds as a support to the management in several ways which includes: resource management, SLA management, billing management, and security management.

To properly support the **resource management** on a cloud, a monitoring system must be able to capture its current state (AVERSA; TASQUIER; VENTICINQUE, 2012) (VIRATANAPANU et al., 2010). Monitoring systems are a crucial part of resource management because they have to guarantee to cloud operators means to quantify the number of resources (*e.g.,* storage, CPU) that they have available as well as provide support to estimate the workload to execute a specific application or service. It is challenging in many ways. For example, clouds have relied on virtualization technologies to overcome the heterogeneity of resources among other problems. Such virtualization has introduced challenges to cloud monitoring systems that have to cope with physical and virtual resources (FERRETTI et al., 2010) (LAKSHMANAN et al., 2010). Also, virtual resources can migrate from a physical resource to another at any time. For this reason, clouds are extremely dynamic, and the monitoring system must cope with it.

Another example of resource management is related to the trade-off between performance and energy management in clouds (ROSSI et al., 2014). In this case, cloud monitoring systems are an instrumental tool to assist customers that want to run applications that depend on performance whereas taking care of the energy waste.

To **SLA management**, monitoring is fundamental to certify SLA accomplishment to cloud operators and customers (ZHAO; SAKR; LIU, 2015) (EMEAKAROHA et al., 2012). To cloud operators, monitoring is also important when formulating SLA models who are seeking to present competitive market prices, performance and so forth (ROY; GANESAN; SARKAR, 2013). In the same way, monitoring plays a key role in **billing management** when supporting the model *pay per use* which is one of the basic concepts to clouds (VAQUERO et al., 2008).

Finally, monitoring is the basis for the **security management** because it provides means to audit the cloud environment as well as supports for surveillance activities (DOELITZSCHER et al., 2011). By the way, security management in cloud computing environments is considered a hot topic (KULKARNI; KHANAI, 2015). It happens because security is currently one of the obstacles to the spread of cloud computing, especially considering some kinds of applications (*e.g.,* business critical ones) and customers (*e.g.,* enterprises) (RAMACHANDRAN; CHANG, 2014).

2.2.2 Cloud Monitoring Structure

An effective management of a cloud depends on a complete monitoring of its structure. To provide a complete monitoring, we consider that a cloud monitoring structure is divided into three components defined as: cloud model, monitoring view and monitoring focus.

- *Cloud Model*: Clouds are offered on service models. They are Software as a Service (SaaS), when applications ready to be used are provided to customers; Platform as a Service (PaaS), when SPs are offered a platform where applications can be deployed. The InPs controls the allocation of underlying resources, and SPs have only to concern about writing the application; and Infrastructure as a Service (IaaS), where SPs have access to virtual machines where they can install their own platforms and applications (MONTES et al., 2013).

- *Monitoring View*: The view of cloud resources depends on who wants to obtain the information, *i.e.,* InPs, SPs, or customers. InPs are the owners of the infrastructure, and normally are concerned about the infrastructure's correct operation and efficient utilization. InPs may obtain information about both virtual and physical layers. In addition, InPs can make control activities over the layers. SPs are the guiding support to customers. SPs, in general, can obtain information about the virtual layer, such as response times and latencies observed in different elements of the platform, and how it relates to the performance observed by customers. Customers, in turn, can see information about the high level application/services they are using. Thus, monitoring view must be set so as to cope with different visions to different InPs, SPs, and customers (MONTES et al., 2013);

- *Monitoring Focus*: Design and implementation of monitoring solutions depend on the type of resource (*e.g.,* processing, network) or service (*e.g.,* SLA, QoS) to be monitored. Monitoring focus is the goal (resource type or service) defined by a specific monitoring solution or group of monitoring solutions so as to attend the specific requirements of InPs, SPs, and customers. Monitoring focus can be divided using two methods: by a cloud model or specific focus. The first one refers to the service model: SaaS, PaaS, or IaaS. The second refers to the specific focus of the monitoring performed by InPs, SPs, or customers (*e.g.,* SLA, billing).

Figure 2.2 shows a cloud monitoring structure, depicting the cloud models that compose a cloud, monitoring views to both SPs and customers, and monitoring focus. In this scenario,

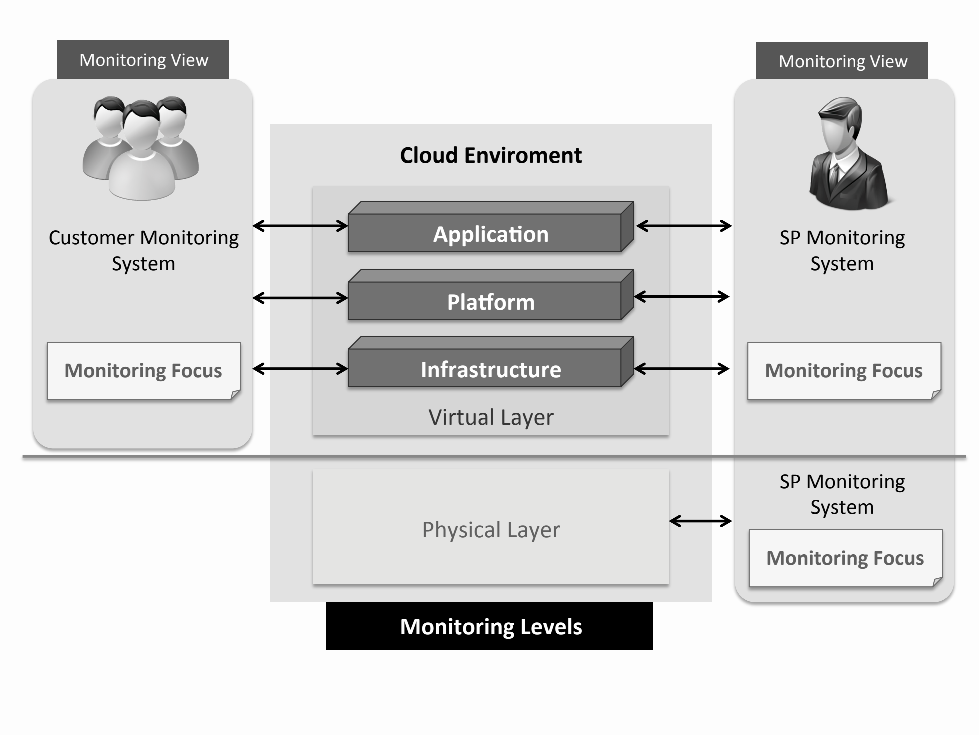


Figure 2.2: Cloud monitoring structure

monitoring focus has several goals based on either service models or specific focus as shown in Figure 2.3. In general, these goals are reached by monitoring solutions that are developed to address specific monitoring necessities. For example, monitoring the cloud model and/or achieve cloud monitoring requirements through specific focus.

Aiming to present a landscape to the development of monitoring solutions, we highlight the main goals of cloud computing solutions. The main goals of the solutions are defined according to their cloud model. Furthermore, in Section 2.2.5, we discuss research challenges based on specifics goal of each model.

In the **Infrastructure (IaaS)**, cloud resources are created on top of the bare hardware, which is often performed with the use of virtualization technologies. At IaaS, monitoring solutions acting on behalf of InPs monitor the actual hardware supporting the infrastructure, whereas SPs aim to get information about the virtual resources that are rented by them.

IaaS is either offered by public IaaS cloud providers such as Amazon EC2 (Amazon, 2015b), or it is built on private clouds by using solutions such as Eucalyptus (NURMI et al., 2009) and OpenNebula (SOTOMAYOR et al., 2009). Resources offered at IaaS are typically in the form of virtual machines (*e.g.,* Xen (BARHAM et al., 2003), KVM (KVM, 2014)). Virtual machines (VMs) are composed of resources such as processing, and storage. Therefore, at IaaS, cloud monitoring solutions have specific focus on monitoring basic metrics (*e.g.,* processor load

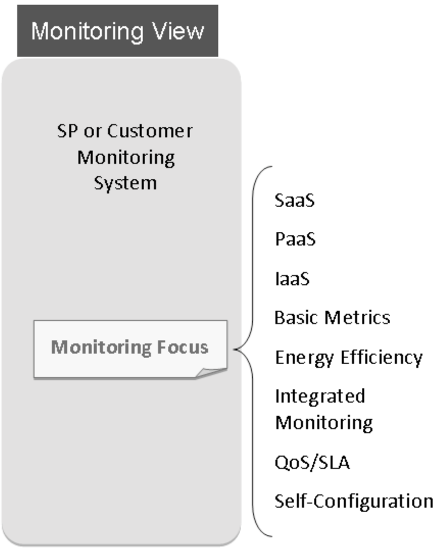


Figure 2.3: Cloud monitoring focus

and network usage) (CLAYMAN et al., 2010) (SHAO et al., 2010).

The **Platform (PaaS)** is composed of both, programming environments and execution environments. Commercially, Google App Engine (Google, 2014), and Heroku (Herokai Group, 2014) are example of PaaS.

PaaS aims to provide an environment for applications development (*e.g.,* programming language). Additionally, at PaaS services are provided to support the deployment and execution of applications, including features such as fault tolerance, server management, auto-scaling, and self-configuration (BRANDIC, 2009) (FELLER; RILLING; MORIN, 2012) (RAK et al., 2011).

At PaaS, cloud monitoring systems provide information to assist a given InP to deal with issues such as self-configuration and fault tolerance management. From a SP perspective, monitoring has the objective of ensuring that the platform is supporting a responsive application, as observed by customers.

At **Software or Application (SaaS)**, there are applications of interest to potentially millions of users that are geographically spread. An example of this is online alternatives for typical office applications such as word processors and spreadsheets (VAQUERO et al., 2008).

Besides, the diversity of applications in clouds is growing. To handling the diversity at SaaS, a cloud monitoring system needs to have unusual abilities such as coping with heterogeneous APIs and coping with different monitoring slices (CARVALHO et al., 2014) (CARVALHO et al., 2013).

Additionally, SPs and customers have defined SLA's to regulate the agreement between both. Therefore SLA's need to be respected and accomplished (BELOGLAZOV; BUYYA, 2013) (BRANDIC, 2009).

2.2.3 Cloud Monitoring Requirements and Abilities

Monitoring systems must be designed and developed to fit with the environment in which it will operate. Thus, some particular features of such environment have to be taken into consideration when building specialized monitoring systems. Clouds have introduced its particularities (*e.g.,* SLAs, massive use of virtualization) (BUYYA et al., 2009), and they have to be fulfilled. Aiming to solve it cloud monitoring systems have been developed taken into consideration a plenty of requirements that are not consensus in the literature.

The number of requirements has impaired the development of complete cloud monitoring systems in several ways. For example, it is complex either to accomplish all of them in a single cloud monitoring solution or to integrate some monitoring solutions to perform them. Besides, it has not yet the consensus in the literature because cloud monitoring solutions have been developed to perform specific functions that are only suitable to solve partial monitoring problems and, as a result, partial solutions just perform some requirements. On the other side, a considerable number of monitoring solutions currently used in clouds, were developed to work with other technologies such as grids (FOSTER; KESSELMAN, 1999) and clusters (HUANG et al., 2006). Therefore, these monitoring solutions are specialized in other environments that have their particularities. Even though the clouds particularities are similar to these other technologies. Clouds have particular characteristics, and they have to be performed in a properly way (FOSTER et al., 2008).

To support the development of specialized and complete cloud monitoring systems, we proposed to define cloud monitoring requirements based on cloud characteristics. Also, we suggest classifying as ability some requirements that are not key, but desirable and useful, to the development of cloud monitoring systems. The criteria used to define an ability is related to the number of cloud characteristics that it aim to support. In other words, cloud monitoring requirements have to support at least three cloud characteristics. Otherwise, we consider it less significant, and it is classified as an ability. For example, even though accuracy is important to the transparency of SLAs accomplishment, it is not vital to other cloud characteristics. Besides, it has directly compromised others requirements such as timeliness, and elasticity. Thus, we consider it as an ability. In this section, we present definitions to both requirement and ability along with cloud characteristics that are the basis for the classification of requirements and abilities that are defined one by one.

According to NIST (MELL; GRANCE, 2011), there are five essential characteristics of a cloud (See Section 2.1.3), namely, On-demand self-service, Broad network access, Resource pooling, Rapid elasticity, and Measured service. Besides, the literature has presented a significant discussion about other key characteristics which includes: SLA-Based, high availability, the massive use of virtualization, load balance, and multi-tenancy (ZHANG; CHENG; BOUTABA, 2010) (BUYYA et al., 2009). Such characteristics have been considered by us to define cloud monitoring requirements and cloud monitoring abilities.

A requirement is a characteristic that a cloud monitoring system must have to cope with features of a cloud. In other words, a requirement is mandatory for a cloud monitoring system notwithstanding they have mutual influence among themselves and with abilities. We highlight them below.

- Scalability: Scalability is the capacity to improve the performance of the system by increasing the computational resources. To fulfil this feature, the monitoring system needs to keep monitoring efficiently with potentially a plethora of probes to cope with cloud characteristics such as resource pooling, on-demand self-service, and rapid elasticity (CLAYMAN et al., 2010) (MONTES et al., 2013);

- Elasticity: Elasticity is the competence to increase and decrease computational resources on demand, according to the goal of a specific application or system. Elasticity aims to improve a cloud in terms of performance and cost. To support elasticity, the monitoring system needs to track virtual resources created/destroyed by expanding/contracting a cloud and to correctly handle expansion/retraction of the system. Elasticity is a requirement related to several cloud characteristics (*e.g.,* on-demand self-service, rapid elasticity, and massive use of virtualization) (CHAPMAN et al., 2010) (CLAYMAN et al., 2010) (COPIL et al., 2013);

- Migration: Migration is the capacity to change the location of computational resources according to the goals of a specific application or system. Migration has provided improvements to users in terms of performance, energy consumption, and costs. In migration, any virtual resource that moves from one physical host to another must be correctly monitored to ensure that no information is lost upon migration, and that the monitoring system is not negatively affected by the potential migration of a monitored resource. This requirement is related to cloud characteristics such as load balance, massive use of virtualization, and high availability (CLAYMAN et al., 2010) (FAN; WANG; CHANG, 2011) (RODRIGUES et al., 2016);

- Timeliness: Timeliness is the competence to detect events with available time for their intended use. In other words, it is a requirement in which a cloud monitoring system is able to supply information in the time that users (*e.g.,* service providers and customers) need to access such information in order to perform management activities. Timeliness is an important requirement for cloud monitoring systems because its activities (*e.g.,* collection, synchronization) need to be opportunely accomplished to avoid breaches in SLAs and consequent financial penalties cause by late detections of problems or events. Timeliness is vital to some cloud characteristics which includes: SLA-based, measured service, and resource pooling (WANG et al., 2011) (ACETO et al., 2013);
- Resilience: Resilience is the capacity to keep working when some components are failing. In other words, resilience is the requirement in which a monitoring system persists to deliver its services when facing changes or problems. It means that a cloud monitoring system must endure to failures while continue to operate. Resilience is significant to cloud monitoring systems because it provides support to cloud characteristics (*e.g.,* high availability, the accomplishment of SLAs, and on-demand self-service) related to the quality of services (ACETO et al., 2013) (TAN; VENKATESH; GU, 2013);
- Availability: Availability is the competence to provide monitoring services based on the cloud monitoring system design whenever customers, SPs, and InPs are requesting such services. Cloud monitoring systems must have availability because it is essential to guarantee crucial issues related to cloud characteristics such as the accomplishment of SLAs, on-demand self-service and high availability (MCGILVARY et al., 2013) (HADLEY et al., 2015).

Besides such requirements, cloud monitoring systems should be able to enhance themselves to handle with clouds features. For this reason, a complete cloud monitoring system has to have at least some abilities. This thesis introduces the concept of cloud monitoring ability based on a division of cloud monitoring requirements into two distinct groups, requirements, and abilities.

An ability is a characteristic that a cloud monitoring system should have to cope with features of a cloud. In other words, an ability is desirable for a cloud monitoring system notwithstanding they have mutual influence among themselves and with requirements. We highlight abilities as follows:

- Accuracy: Accuracy is the ability of monitoring systems to measure without making mistakes. In cloud computing environments, accuracy is critical because SLAs are an intrinsic part of the system. Therefore, a poor performance can lead to financial penalties to InPs and SPs and loss of confidence by customers, which may damage the reputation of the company and result in a reduction of the client base (MENG et al., 2012) (MIAN; MARTIN; VAZQUEZ-POLETTI, 2013) (RODRIGUES et al., 2014);

- Adaptability: Adaptability is the ability of a cloud monitoring systems to adapt to varying computational and network loads in order not to be invasive (*e.g.,* preventing other activities, resulting in performance losses). Thus, the workload of cloud monitoring systems cannot negatively affect other activities in a cloud. In this context, cloud monitoring systems have to be capable to reconfigured themselves ideally at runtime to perform adaptability (KUTARE et al., 2010) (TOUEIR; BROISIN; SIBILLA, 2013) (RODRIGUES et al., 2014);

- Autonomy: In clouds, dynamicity is a key factor because changes are intense, sharp and frequent. Autonomy is the ability of a monitoring system to self-manage its configuration to keep itself working properly in a dynamic environment. Enabling autonomy in a cloud monitoring system is complex since it requires the ability to receive and manage inputs from a plethora of probes (CLAYMAN et al., 2010) (BUYYA; CALHEIROS; LI, 2012);

- Comprehensiveness: Cloud computing environments encompass several types of resources (*e.g.,* different virtualization resources, various physical resources) and information. Comprehensiveness is the ability of a monitoring system to support several types of resources and information. Therefore, the cloud monitoring system should have the capacity to retrieve updated status from different kinds of resources, several types of monitoring data, and a large number of users (HASSELMEYER; D'HEUREUSE, 2010) (ACETO et al., 2013);

- Reliability: Reliability is the ability to perform a required function under stated conditions for a specified period presenting a trustworthy outcome. Reliability is critical to cloud monitoring systems in issues such as billing, and SLA accomplishment check. Reliability is critical because customers have to rely on monitoring information provided by cloud monitoring systems to verify that its rented resources have been honestly delivery by SPs and InPs, thereby as well as accuracy, reliability is closely related to the reputation of SPs and InPs (MENG et al., 2012) (ACETO et al., 2013).

In the next section, we will present, discuss, and classify some cloud monitoring solutions taking into consideration the requirements and abilities proposed.

2.2.4 Cloud Monitoring Solutions

We divide monitoring solutions for cloud in three types: generic solutions, cluster and grid solutions, and cloud-specific solutions.

Generic solutions have been created to monitor computational systems without concerns about specific peculiarities relating to each type of system. These solutions are widely used in computational systems to retrieve information about global hosted resources. However, generic solutions may not be suitable regarding some specific features/requirements/abilities of clouds such as virtualization, scalability, and comprehensiveness. Cluster and grid solutions, in turn, were created with these specific domains in mind, but they also lack support for some cloud-specific requirements. Thus, it is required the design and development of cloud-specific solutions.

Generic solutions are often designed without a specific context. Therefore, they are suitable to be used in computational systems, in general. Although generic solutions have been usually created before the emergence of clouds, we can find initiatives to explore the utilization of those solutions in clouds. At clouds, generic solutions can be used to monitor basic metrics (*e.g.,* memory, network). That is the case, for example, of Cacti (Cacti, 2014), MRTG (MRTG, 2014), and Nagios (Nagios, 2014).

Cacti and MRTG are solutions to create RRDTool graphs that are usually used to show bandwidth consumption in network links, but both can plot graphs to any monitoring metric such as processing, and storage. However, Cacti and MRTG do not provide features such as self-configuration or support for discovery to help in their configuration. In clouds, Cacti and MRTG are used to build graphs from basic metrics and aggregated metrics (*e.g.,* number of clouds slices, amount of available resources).

Nagios is a monitoring tool widely employed in traditional environments. One of the Nagios main features is the support for plugins that are used to collect monitoring information from the monitored objects. These plugins can be developed in an easily way and leverage the Nagios flexibility that allows to monitoring virtually any computational environment. Therefore, Nagios flexibility allows to the developers of plugins to collect basic metrics and aggregated metrics in clouds.

Cacti, MRTG and Nagios do not handle cloud monitoring requirements/abilities such as elasticity, and autonomy. For example, these solutions do not support self-configuration. Thus, new solutions focusing on the requirements of cloud monitoring are required.

On the other side, many monitoring systems were proposed to handle cluster and grid systems. Monitoring system specific to clusters include PARMON (BUYYA, 2000) and RVision (FERRETO; ROSE; ROSE, 2002), whereas grid monitoring systems include GridEye (FU; HUANG, 2006) and Ganglia (MASSIE; CHUN; CULLER, 2003).

There are clear overlaps between cluster and grid requirements and cloud requirements. For example, clusters, like clouds, are composed of many machines connected to local networks. However, clusters do not have SLA as a key priority like clouds have. Furthermore, clouds strongly depend on virtualization, whereas this is not always the case for clusters. Grids tend to be geographically distributed and belong to autonomous management domains, whereas clouds have a large scale infrastructure managed by a single organization. It is a reality in emerging InterCloud approaches where multiple (potentially distinct) cloud services are aggregated to provide a service to the final customer. Similarly to the cluster case, grid systems do not have SLAs as their key priority, and they tend to be cooperative environments rather than financially-driven services.

Cloud specific monitoring solutions have been created to be used in clouds. Currently, cloud-specific monitoring solutions are designed by academic researchers or commercial efforts.

Amazon CloudWatch (Amazon, 2015a) is a monitoring solution for Amazon Web Services (AWS). Amazon CloudWatch allows easy handling of basic metrics such as processing and storage. Additionally, it presents several types of statistics and self-configuration. Thus, Amazon CloudWatch is useful for users and managers of Amazon clouds. However, it is restricted to AWS products. Other commercial cloud monitoring solutions include Accelops (Accelops, 2014), Copperegg (Cooperegg, 2014), Zennoss (Zennoss, 2014), Monitis (Monitis, 2014), and Rackspace (Rackspace, 2014), that like CloudWatch, focus on specific proprietary platforms/solutions.

Private Cloud Monitoring Systems (PCMONS) (CHAVES; URIARTE; WESTPHALL, 2011) is a monitoring solution for private clouds. PCMONS is an open source solution that uses a layer called Integration to provide homogeneous access to users and managers that manipulates resources in a cloud. It provides a uniform monitoring of infrastructures, independently of the type of resource hosted in a cloud. In addition, other monitoring solutions can be used as support and complement to PCMONS, promoting an integration of monitoring solutions. On the other hand, the configuration of monitoring must be done manually, which compromises cloud monitoring requirements/abilities such as scalability, migration, and autonomy.

Cloud Management System (CMS) (HAN et al., 2009) aims to provide a monitoring solution based on RESTful Web Services. It employs REST to allow the development and integration of monitoring solutions. The REST system can design monitoring elements (*e.g.,* network, storage). The Get method in REST can replace the operations of monitoring because REST is widely used for web services in research centres and commercial applications (MCFADDIN et al., 2008). CMS can be easily integrated with other solutions services, and technologies that use REST approaches.

Runtime Model for Cloud Monitoring (RMCM) (SHAO et al., 2010) aims to monitor resources through abstract models, making possible homogeneous handling of heterogeneous resources. In this way, it is possible to work with different resources, such as platforms and virtual systems in a same approach. In addition, it generates customized models according to the needs of each agent that integrates a cloud. These models have been defined as a model for operators, a model for developers and a model for users. However, it requires a constant update of monitoring resources in order to maintain the model consistent. The main disadvantage of this solution is the manual installation and configuration of specific agents. For this reason, cloud monitoring requirements/abilities such as scalability, migration and autonomy are compromised.

The system of monitoring/warning that operates over the mOSAIC (RAK et al., 2011) platform generates warnings when a SLA is apparently in risky conditions. In others words, the monitoring system observes SLA rules, and when they are close to being violated, a message is sent to managers. It is a simple and efficient method to maintain control over monitoring resources.

Flexible Automated Cloud Monitoring Slices (FlexACMS) (CARVALHO et al., 2014) aims to integrate several monitoring solutions to provide a comprehensive cloud solution. FlexACMS allows cloud administrators to configure monitoring solutions automatically, providing flexibility and dynamicity to cloud providers. Flexibility and dynamicity improve operational aspects of cloud providers such as billing and SLA. To reach such automatic configuration, FlexACMS is based on rules defined by cloud administrators that determinate the metrics that must be monitored on each cloud slice and what monitoring solutions must be used to monitor each cloud slice.

2.2.5 Challenges and Trends in Cloud Monitoring

Monitoring plays a significant role in clouds and enables the delivery of services meeting agreed SLAs. Currently, there are several solutions with different purposes of monitoring such as target applications, basic metrics, and target infrastructure. However, some areas in clouds are growing and represent trends of research in monitoring. Therefore, research opportunities for monitoring in clouds have emerged. In this scenario, we highlight as open research opportunities in cloud monitoring integrated monitoring and energy efficiency.

Integrated monitoring is a trend for two main reasons. First, there are several consolidated monitoring solutions that integrate themselves is a natural way. Second, there are sundry goals to monitoring solutions depending on the service model and the role of the monitor (*e.g.,* InP, SP or customer). Thus, design and development of an overall monitoring solution is a hard task. Additionally, integrated monitoring is a goal of recent developments in the area of cloud monitoring (*e.g.,* PCMONS (CHAVES; URIARTE; WESTPHALL, 2011) and RMCM (SHAO et al., 2010)).

Furthermore, there is a trend in cloud computing towards energy management, green computing and reduction of operational costs. It confirms a trend towards energy efficiency. It is evidenced by recent works such as (BELOGLAZOV; ABAWAJY; BUYYA, 2012), (RODERO et al., 2012) and (WANG; DU; CHEN, 2012). In this context, monitoring is paramount to enable reduced energy consumption without compromising application performance and SLAs.

The issues above are generic *i.e.,* they are applied to all cloud models. In order to be more precise, we present trends and future directions for each particular cloud model as follows.

In the IaaS model, monitoring issues such as energy efficiency (ROSSI F.; ROSE, 2015) (BELOGLAZOV; ABAWAJY; BUYYA, 2012) (CARDOSA; KORUPOLU; SINGH, 2009) and integrated monitoring (HASSELMEYER; D'HEUREUSE, 2010) (CHAVES; URIARTE; WESTPHALL, 2011) are especially important. Energy efficiency concerns about reasonable consumption of power to operate services. It aims to reduce the energy waste. Integrated monitoring has concerns about integration of several monitoring solutions that operate in a cloud. It aims to develop comprehensive cloud monitoring solutions or promote integration among monitoring solutions.

In the context of IaaS monitoring, open challenges include translation of higher services objectives in useful lower-level metrics as observed in the infrastructure layer, virtualization and multi-tenancy-aware monitoring, and comprehensive monitoring solutions.

In the PaaS model, a gap exists in self-configuration. Self-configuration has addressed adaptation of cloud monitoring solutions to characteristics such as dynamicity and elasticity. However, to support these characteristics, cloud monitoring solutions demand more resources from infrastructure to cater for SLAs. Therefore, an emerging issue in the area of PaaS monitoring concerns improved techniques for self-configuration to reach a suitable balance between consumption of infrastructure resources and SLA constraints.

In the SaaS model, because of the diversity of customers, applications, and SLAs, cloud monitoring systems must handle different scenarios. There are different customers profiles with different SLAs using a plethora of applications. The monitoring system needs to identify the customers correctly and ensure that privileged customers are getting better services than ordinary customers.

In this scenario, emerging research questions include how to fulfill SLAs without being invasive (impairing other SLAs), and how to improve the profit of SPs without compromising quality of service to customers.

Additionally, when a cloud monitoring system aims to accomplish a specific requirement/ability, it is usually negative or positive affected by other requirement/ability. Therefore, the development of cloud monitoring systems have focused on improving specific aspects of cloud operation, providing only partial solutions for cloud monitoring. Thus, the balance among cloud monitoring requirements/abilities is a challenging and significant trend. Complementary, it is important to develop solutions that can be integrated. Thus, cloud monitoring solutions designed to cope with specific requirements/abilities could be complemented by other solutions, resulting in a complete solution for cloud monitoring.

Finally, several gaps remain and there are several challenging research directions to be explored in cloud monitoring such as creation of comprehensive monitoring solutions, improve techniques for self-configuration, translation of higher service objectives in useful lower level metrics as observed in the infrastructure layer, reduce the intrusiveness in accomplishment of SLAs, and unveil the interplay among requirements/abilities which is the focus of this thesis.

2.3 Network Topologies for Cloud Computing Environments

Data centers have become an efficient infrastructure for supporting data storage, and providing the platform for deployment of a plethora of services and applications such as cloud computing and video streaming (BARI et al., 2013). Clouds particularly have imposed multiple resource demands (*e.g.,* storage, distributed processing power). These demands are based on cloud characteristics and to reach them clouds have to rely on an efficient network.

The network of a data center is the infrastructure used as a support for communication among nodes. It is defined in the literature as either network topology or topology. Usually, a network topology is composed of network components (*e.g.,* routers, switches) and it is based on protocols (*e.g.,* Ethernet, TCP). Currently, there are some different ways to deploy a network topology to support a cloud, but the vast majority are structures based on trees (SUN et al., 2013) (BARI et al., 2013) (GREENBERG et al., 2009). In this section, we will highlight network topologies implemented in data centers used as a support to clouds.

2.3.1 Conventional Data Center Network Topology

Conventional data center networks topology usually have been built of either two or three level trees of switches or routers (FIORANI et al., 2014) (BARI et al., 2013) (SUN et al., 2013) (GREENBERG et al., 2009) (AL-FARES; LOUKISSAS; VAHDAT, 2008). Figure 2.4 depicts an example to this topology. A three-level network topology has a core level placed at the root of the tree. Underneath the core is placed the aggregation level. The edge level completes the structure of a three-level network topology placed at the bottom. On the other hand, a two-level network topology is shorter and composed by levels core and edge.

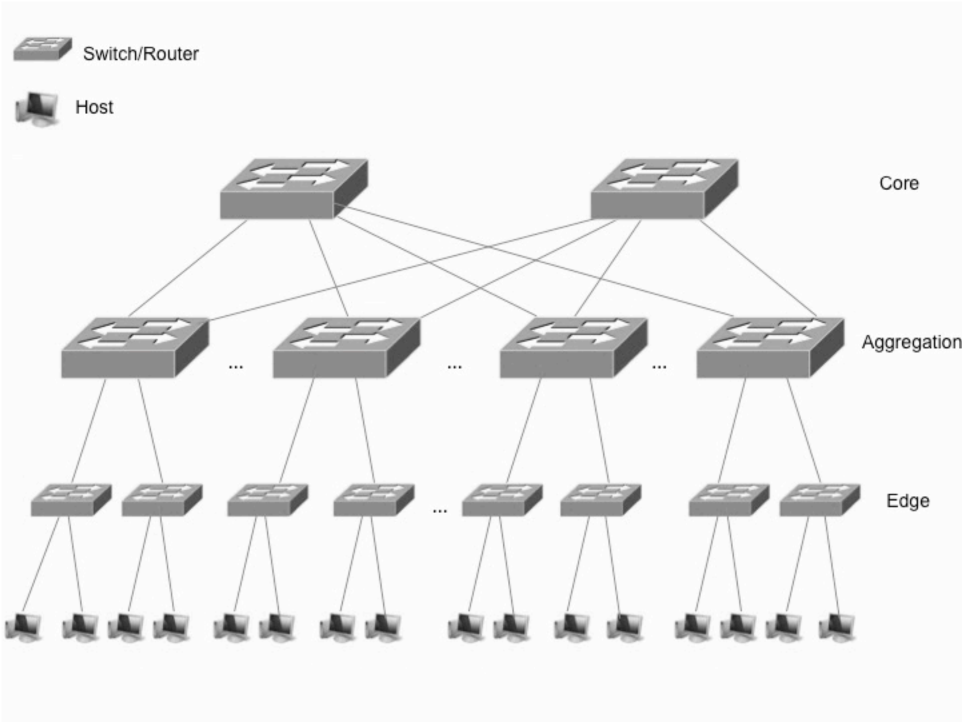


Figure 2.4: Conventional data center network topology

Whether considering a three-level network topology, the edge switches provide connectivity to hosts deployed in racks. The aggregation switches, at the aggregation level, are in charge to forward data from edge switches to the core level.

2.3.2 Clos Topology

Currently, the price differential between commodity and non-commodity switches provides a strong incentive to deploy large-scale networks based on a plenty of commodity switches rather than fewer larger and more expensive ones. A similar scenario in telephone switches stimulated to the development of a Clos topology that delivers high levels of bandwidth for a plethora of end devices by appropriately interconnecting smaller commodity switches (GUO; YANG, 2015a) (BARI et al., 2013) (AL-FARES; LOUKISSAS; VAHDAT, 2008) (DALLY; TOWLES, 2003).

Clos topology is built up from multiple stages of switches as depicted in Figure 2.5, which is an example of a Clos topology composed of three stages. This topology is easy to understand, each switch in a particular stage is connected to all switches in the next stage providing an extensive diversity of paths to data communication.

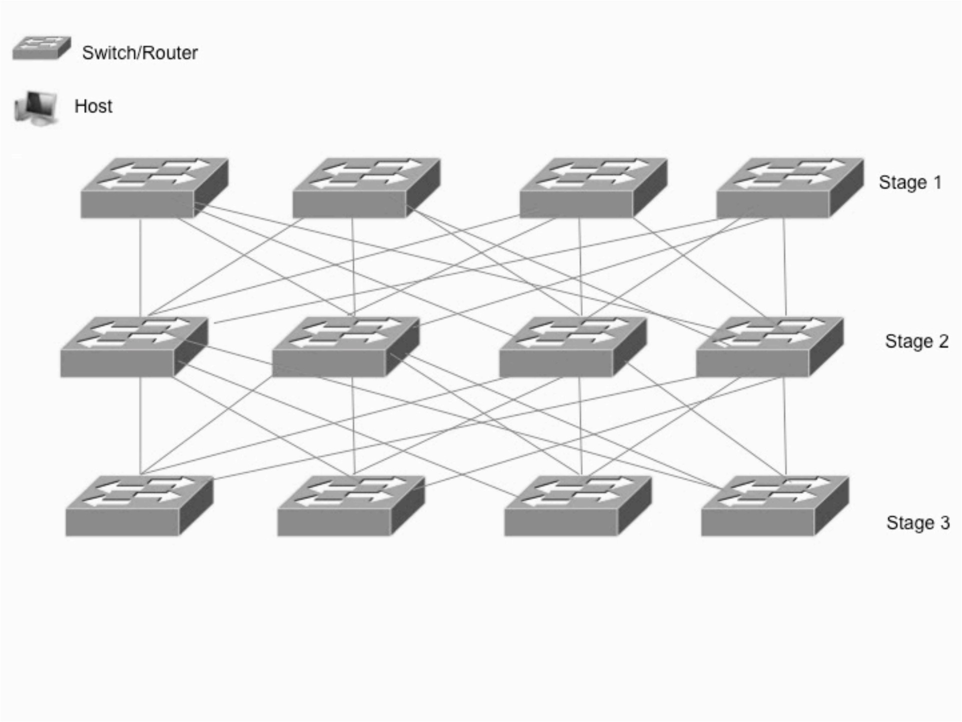


Figure 2.5: Clos topology

2.3.3 Fat Tree Topology

According to the literature, Fat-tree topology is a particular type of Clos topology that has its structure based on a tree as presented in Figure 2.6 (GUO; YANG, 2015b) (SUN et al., 2013) (BARI et al., 2013) (LI et al., 2009) (AL-FARES; LOUKISSAS; VAHDAT, 2008).

Fat-tree topology is built of *k*-port switches that contain *k pods*. Each *pod* has two levels of *k* / 2 switches, namely, aggregation and edge. Each of (*k* / 2) core switches has one port connected to each of *k* pods. The *i*-th port of any core switch is connected to pod *i* so that consecutive ports in the aggregation level of each pod switch are connected to core switches on *k* / 2 strides. Each edge switch is directly connected to *k* / 2 end-hosts. Each of the remaining *k* / 2 ports of an edge switch is connected to *k* / 2 ports of an aggregation switch.

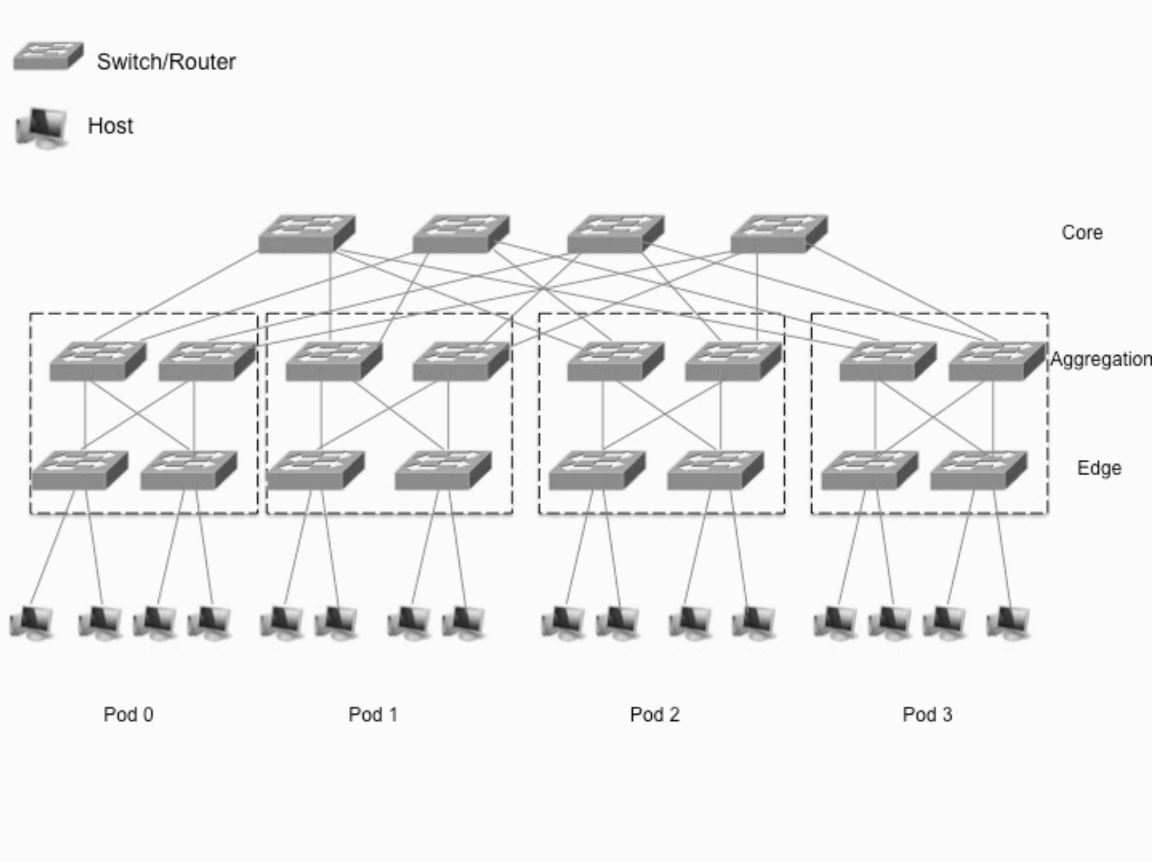


Figure 2.6: Fat-tree topology (k=4)

2.4 Summary

This chapter presented an overview of the state of the art regarding cloud computing, cloud monitoring, and network topologies for cloud computing. In order to maintain consistency, in this chapter cloud computing concepts are mainly based on the NIST definition of a cloud. In regard to cloud monitoring, this chapter presents some contributions as follows. It introduces the concept of cloud monitoring focus. This chapter distinguishes the concepts of cloud monitoring requirements and cloud monitoring abilities. Although, cloud monitoring requirements are not a new concept, cloud monitoring abilities presented in this chapter are a new concept and the separation between both is a breakthrough in cloud monitoring. In addition, this chapter presents some cloud monitoring solutions and discusses challenges and trends in cloud monitoring. In regard to challenges and trends, we highlighted the interplay among cloud monitoring requirements and abilities. In this context, scalability and timeliness are two cloud monitoring requirements that have mutual negative influence. The interplay between both will be discussed in details in the following Chapter. Finally, this chapter presented the most significant network topologies for clouds. Such topologies are the basis in which scalability and timeliness will be assessed.

3 THE INTERPLAY BETWEEN TIMELINESS AND SCALABILITY

This chapter provides an extensive study of the interplay between timeliness and scalability. It starts discussing some researches about cloud monitoring requirements and abilities to cloud monitoring systems to present a landscape in which the mutual influence between timeliness and scalability is a challenging and significant issue. After that, it presents a broad discussion about timeliness and scalability that include: the methods currently used to cope with the interplay between both requirements and the relation between such requirements and monitoring parameters (*i.e.,* frequency sampling). In a third step, it is concentrating on an evaluation of the mutual influence between timeliness and scalability. Such evaluation is based on monitoring parameters, namely, monitoring topology, amount of monitoring data, and frequency sampling. Next, it shows remarks about the results proposing a generic method to deal with the mutual influence between timeliness and scalability.

3.1 Related Work

Monitoring is a key function for cloud operators that intend to deliver high quality services to customers (CALERO; AGUADO, 2015a). Aiming to achieve it, researchers have investigated cloud monitoring requirements along with their mutual influence to define the properties that are necessary in a complete cloud monitoring system in order to enhance it. In this section, we highlight the most recent and relevant investigations in this area.

Aceto *et al.* (ACETO et al., 2013) presented an exhaustive study on cloud monitoring. The authors defined a set of properties that cloud monitoring systems should support, difficulties in supporting those properties, and related solutions currently available in the literature. Examples of the monitoring properties discussed by Aceto *et al.* include: scalability, elasticity, adaptability, and timeliness. Finally, the authors presented the current platforms and services available for monitoring cloud environments and discussed which properties are tackled by each platform.

Montes *et al.* (MONTES et al., 2013) proposed a cloud monitoring solution based on levels (*e.g.,* SaaS, PaaS, IaaS, physical) called GMonE. By using GMonE, both Cloud Service Providers and customers are able to visualize monitoring data. Also, Montes *et al.* presented an evaluation about the mutual relation between scalability and elasticity. The results demonstrated that monitoring solutions can be improved when monitoring requirements are considered together.

Clayman *et al.* (CLAYMAN et al., 2010) presented lattice, a cloud monitoring framework developed to monitor both resources and services in virtualized environments. The design and development of the framework, allowed Clayman *et al.* to identified the main requirements for cloud monitoring systems, *i.e.,* scalability, elasticity, migration, adaptability, autonomy, and federation. The authors also discussed federation problems and its impact on the monitoring activity.

Shao *et.al* (SHAO et al., 2010) proposed a cloud monitoring solution based on abstract models to allow a homogeneous management for heterogeneous cloud resources. Such solution is defined as Runtime Model for Cloud Monitoring (*i.e.,* RMCM). RMCM creates customized models according to the needs of each entity (*e.g.,* cloud operators) that compose a cloud computing environment. Through such models, RMCM wants to promote integration among cloud monitoring solutions tackling an important cloud monitoring ability which is comprehensiveness. However, RMCM requires a frequent update of monitoring resources to keep its models consistent. The main disadvantage of this solution is the manual installation and configuration of agents. Thus, cloud monitoring requirements such as scalability, elasticity and migration are impaired.

Hasselmeyer and Heureuse (HASSELMEYER; D'HEUREUSE, 2010) developed a cloud monitoring architecture that provides one single monitoring infrastructure. This cloud monitoring solution tackles some monitoring requirements such as scalability, multi-tenancy, dynamism, simplicity, and comprehensiveness. It is significant because the authors intended to solve multiple requirements using a single monitoring architecture. However, this monitoring solution has two major problems. First, simplicity is a sort of cloud monitoring requirement that is only defined by the authors, so it has lacked in depth discussion of the literature about the significance of this requirement. Second, Hasselmeyer and Heureuse omitted other requirements/abilities such as timeliness, adaptability, elasticity, and accuracy.

Chaves *et.al* (CHAVES; URIARTE; WESTPHALL, 2011) developed the Private Cloud Monitoring System (*i.e.,* PCMONS) a cloud monitoring solution that provides homogeneous access to customers and cloud operators through a layer called integration. By using PCMONS, cloud users can perform two key functions. First, they can uniformly monitor its infrastructures, regardless of the type of resources hosted. Second, PCMONS can be used as either support or complement to the integration of other monitoring solutions. By promoting integration, PCMONS tackles a significant cloud monitoring ability, namely, comprehensiveness. However, when using PCMONS, the configuration process must be done manually, and it commits some requirements which include: scalability, and migration.

Meng and Liu (MENG; LIU, 2013) introduced the concept of Monitoring-as-a-Service (*i.e.,* MaaS) and its principal components. The authors wanted to optimize the performance of some cloud monitoring requirements such as scalability. In order to deal with it, they argue that Monitoring-as-a-Service should support at least the conventional monitoring capabilities such as periodical monitoring and instantaneous violation detection. The results demonstrated that they lack in a depth evaluation to specifically prove the interplay between cloud monitoring requirements.

Povedano-Molina *et al.* (POVEDANO-MOLINA et al., 2013) proposed the distributed architecture for resources management and monitoring in clouds, called Dargos. Dargos is a cloud monitoring architecture to disseminate resources monitoring information. Povedano-Molina *et al.* presented an architecture that dealt with the mutual influence among cloud monitoring requirements such as scalability along with significant cloud monitoring abilities such as adaptability and accuracy. The results demonstrated that a cloud monitoring system can be enhanced even when cloud monitoring requirements and abilities are considered together.

Rodrigues *et.al* (RODRIGUES et al., 2014) presented an investigation of the mutual influence among scalability, adaptability, and accuracy. Results demonstrate the existence of mutual influence among these requirements and analyse the impact of each requirement over others. The authors show that the impact of scalability over adaptability is reduced by adopting specifics monitoring methods. For example, when a method of filtering and aggregation is used to decrease the amount of monitoring data, the mutual influence between adaptability and scalability is inverted. In other words, the mutual influence between them is usually negative, and by using filtering and aggregation it becomes positive.

Carvalho *et.al* (CARVALHO et al., 2014) developed the Flexible Automated Cloud Monitoring Slices (*i.e.,* FlexACMS). FlexACMS aims to provide means to integrate a bunch of monitoring solutions that together can deliver a comprehensive cloud monitoring solution. Also, it enables cloud operators to configure monitoring solutions automatically. To deliver such automatic configuration, this solution is based on rules defined by cloud administrators that determinate the metrics that must be monitored on each cloud slices and what monitoring solutions must be used to monitoring each cloud slice. FlexACMS tackles some important cloud monitoring abilities which include: autonomy, and comprehensiveness. It is also a solution that accomplishes cloud monitoring requirements such as scalability, and elasticity.

Calero and Aguado (CALERO; AGUADO, 2015b) compared a set of monitoring architectures for cloud computing infrastructures. The authors compared monitoring architectures in terms of performance, usage of resources, and security. Regarding cloud monitoring require-

Table 3.1: Cloud monitoring requirements discussed or tackle in the related work.

Paper / Requirements	**Scalability**	**Elasticity**	**Migration**	**Timeliness**	**Resilience**	**Availability**
(ACETO et al., 2013)	X	X		X	X	X
(MONTES et al., 2013)	X	X				
(CLAYMAN et al., 2010)	X	X	X			
(SHAO et al., 2010)						
(HASSELMEYER; D'HEUREUSE, 2010)	X					
(CHAVES; URIARTE; WESTPHALL, 2011)						
(MENG; LIU, 2013)	X					X
(POVEDANO-MOLINA et al., 2013)	X			X		
(RODRIGUES et al., 2014)	X					
(CARVALHO et al., 2014)	X	X				
(CALERO; AGUADO, 2015b)	X					

ments, scalability is analysed and evaluated. It is significant because the paper provides an extensive discussion along with comparison among cloud monitoring architectures based on qualitative requisites. Also, is important to highlight that, to perform the evaluation, the authors deployed the monitoring architectures in a real cloud computing infrastructure based on OpenStack (BELOGLAZOV et al.,) (Openstack, 2015). However, it lacks in terms of comprehensiveness to cloud monitoring requirements because the authors considered only scalability.

Table 3.1 and Table 3.2 present a comparative that summarize the cloud monitoring requirements and abilities discussed or tackled by the related work. It is important to highlight that more comprehensive works (*i.e.,* the papers that presented more requirements) are papers which only discussed cloud monitoring requirements. In other words, they do not want to accomplish any requirement through a cloud monitoring solution, they only discussed the significance of requirements. These works include, for example, Aceto *et.al.* On the other side, some works want to tackle cloud monitoring requirements through monitoring solutions. Usually, such works (*e.g.,* Montes *et.al*) fulfilled some requirements without taking care of others. In this scenario, we highlight that scalability is a subject that matters to 9 in 11 papers. Thus, scalability is the most significant requirement because it is the requirement which is more mentioned and tackled by the related work.

Despite the significance of the presented investigations, they lacked in an extensive discussion on the mutual influence of requirements. Such a discussion is vital once each requirement is affected by others (MONTES et al., 2013). In this area, a comprehensive discussion is necessary because there are several requirements. It imposes that at least some evaluations

Table 3.2: Cloud monitoring abilities discussed or tackle in the related work.

Paper / Requirements	**Accuracy**	**Adaptability**	**Autonomy**	**Comprehensiveness**	**Reliability**
(ACETO et al., 2013)	X	X	X	X	X
(MONTES et al., 2013)					
(CLAYMAN et al., 2010)		X	X		
(SHAO et al., 2010)				X	
(HASSELMEYER; D'HEUREUSE, 2010)				X	
(CHAVES; URIARTE; WESTPHALL, 2011)				X	
(MENG; LIU, 2013)	X	X			
(POVEDANO-MOLINA et al., 2013)	X	X	X		X
(RODRIGUES et al., 2014)	X	X			
(CARVALHO et al., 2014)			X	X	
(CALERO; AGUADO, 2015b)					

between them have to be performed to unveil the effects that a specific requirement have over other. Regarding timeliness and scalability, we have ourselves initiate an effort to understand the mutual influence between them (RODRIGUES et al., 2015). The results allowed us to identify the monitoring parameters (*e.g.,* frequency sampling, amount of monitoring data) that influence timeliness and scalability.

3.2 Scalability and Timeliness

Previous research provided a broad discussion about cloud monitoring requirements. However, there are several issues that require further investigation. In this section, we discuss the mutual influence between timeliness and scalability based on factors that influence both of them.

Timeliness is the competence that a monitoring system has to detect events in time to assist users to obtain information at the moment in which they need to use it. Timeliness is important to cloud monitoring systems because cloud systems are based on Service Level Agreements (SLAs) that regulate the deal among infrastructures providers, service providers, and customers. In this scenario, if monitoring data is not timely, an action to correct the violation in the SLA cannot be accomplished in time, resulting in penalties (costs) to a service operator (*e.g.,* cloud provider, infrastructure provider) (WANG et al., 2011) (ACETO et al., 2013).

On the other side, scalability is the competence to increase the amount of probes in a monitoring system to cope with resources increase in the cloud. Scalability is important to cloud monitoring systems because the cloud business model provides resources on demand (MONTES et al., 2013). Traditional monitoring systems developed to large distributed systems such as Ganglia (MASSIE; CHUN; CULLER, 2003), Nagios (Nagios, 2014), and GridICE (ANDREOZZI et al., 2005) do not easily handle with cloud features that are directly related to scalability such as dynamicity and autonomicity (CLAYMAN et al., 2010).

Scalable systems like clouds have the capacity to increase the amount of resources on demand quickly. However, it implies in challenges to cloud monitoring systems. For example, it can be challenging to assure that a cloud monitoring system will detect and respond within a previously agreed time interval (*e.g.,* 10ms, 50ms, 200ms) a virtual machine failure in a cloud computing environment with 10,000 virtual machines (timeliness). In this context, other aspects that can be suitably explored include: how to define SLAs in accordance with the size of a cloud, or how to provide support to a service provider in order to define SLAs based on its response time capacity, or how to give support to an infrastructure provider to deploy resources based on its response time aptitude.

Cloud monitoring systems currently have two methods to handle the mutual influence between timeliness and scalability. In the first method, the cloud monitoring system works to accomplish a specific requirement in detriment to another. In other words, it aims to achieve scalability without concerning its impact over timeliness or vice-versa. This method is widely used (MONTES et al., 2013) (MENG; LIU, 2013), although it is not efficient because it restrains the cloud monitoring system capacity to attend a specific requirement, creating incomplete cloud monitoring solutions.

In the second method, the cloud monitoring system aims to provide both requirements in a balanced way. This method is more complex and non-trivial and, to be achieved, the mutual influence between these two requirements has to be unveiled.

The acquaintance of mutual influence between timeliness and scalability assists to improve monitoring in clouds. Besides, the mutual influence between both is important because at the same time that a cloud monitoring system grows to monitor all resources in the system, the amount of probes and monitoring data also increase, impairing activities such as data collection and synchronization.

Data collection is impaired in this situation because there are more monitoring data to be gathered. Synchronization is impaired because there are more monitoring data to be analysed together. Thus, to fulfil data collection and synchronization, the cloud monitoring system spends more time. Therefore, the amount of monitoring data makes it difficult to handle it timely, causing a delay between event occurrence and notification. In this context, if scalability is fulfilled, the cloud monitoring system has more monitoring information to be managed, and it impairs timeliness because it induces communication delay (ACETO et al., 2013), as demonstrated in Section 3.3.2.

Usually, filtering and aggregation are implemented in cloud monitoring systems to reduce the amount of monitoring data. They reduce the communication delay and, as a result, timeliness and scalability are enhanced. To apply filtering, there are different methods (*e.g.,* by resource type, statistics), techniques (*e.g.,* compress, reduce), and monitoring architecture (WANG et al., 2011) (HASSELMEYER; D'HEUREUSE, 2010) (KUTARE et al., 2010). However, regardless the method, technique, or architecture, filtering and aggregation were shown to be harmful to others requirements such as accuracy (RODRIGUES et al., 2014).

Moreover, there are monitoring parameters such as frequency sampling, and resources placement, which need to be considered because in a scalable cloud monitoring system they have influence over timeliness.

Frequency sampling depends on the resource type (*e.g.,* CPU, memory) that is monitored. For example, to monitor CPU utilization, a monitoring system has to obtain samples at shorter intervals because this is a resource whose utilization constantly changes in tiny intervals of time. Sampling in higher frequency increases the amount of monitoring data in a network, causing communication delay as verified in Section 3.3.3.

Resource placement is a characteristic that contributes to increasing the communication delay in a scalable cloud monitoring system. It happens because the distance between resources and managers contributes to the time spent in communication, as shown in Section 3.3.1. Also, resource placement is crucial to activities such as synchronization because the managers placement contributes to communication delay.

Besides the above considerations, timeliness and scalability have issues that must be analysed to accomplish other cloud monitoring requirements/abilities such as accuracy, adaptability and elasticity. For example:

- Frequency sampling impairs both *timeliness* and *accuracy*, as shown by Park *et al.* (PARK et al., 2011);
- The method used to increase the amount of probes is important to reduce the influence of *scalability* over *adaptability* (RODRIGUES et al., 2014);
- The dynamic changes (*elasticity*) of monitored resources must be timely (*timeliness*) re-

ported (ACETO et al., 2013).

From the above, we conclude that the evaluation of the mutual influence between timeliness and scalability can assist in future works about the influence of other cloud monitoring requirements as well as it can support the design and development of cloud monitoring systems. In the next section, we perform a quantitative evaluation to timeliness and scalability based on monitoring parameters.

3.3 Quantitative Evaluation of Scalability and Timeliness

The experimental evaluation presented in this section aims to unmask the mutual influence between timeliness and scalability. It provides results such as how such requirements are affected by monitoring topologies and how timeliness and scalability are impaired by monitoring parameters such as the amount of monitoring data, and frequency sampling.

The experimental environment is built using Mininet system (LANTZ; HELLER; MCKEOWN, 2010) to simulate monitoring topologies. The evaluation is performed on an Intel 2.20Ghz Pentium 4 core 2 duo T6600 CPUs, 4GB of RAM, running Ubuntu Server 12.04LTS.

The monitoring parameters that we investigate are the amount of monitoring information per sampling (*i.e.,* 120 and 150 bytes), frequency sampling (*i.e.,* 1 and 10 seconds) and the amount of samples per experiment is 100 samples. Network links have 1 Gbps, and two monitoring topologies are evaluated. Response time is the output metric, and it is measured between probes in the edge hosts and core manager.

Monitoring topologies are based on a conventional data center network for cloud environments consisting of two and three level trees of routers or switches (See Section 2.3.1) (ANDREOLINI; COLAJANNI; TOSI, 2011) (AL-FARES; LOUKISSAS; VAHDAT, 2008). To each switch is added one aggregator, thus, the amount of aggregators depends on the topology. Monitoring topologies are depicted in Figure 3.1 and Figure 3.2, namely, Topology 1 and Topology 2.

Monitoring topologies are evaluated to timeliness based on the addition of hosts as well as aggregators in order to analyse the influence of scalability over timeliness. Topology 1 is extended to 64, 256, 576 and 1296 hosts, and respectively to 10, 18, 26 and 38 aggregators. Topology 2 is extended to 64, 216, 512 and 1331 hosts, and respectively to 21, 43, 73 and 133 aggregators.

In the next subsections, the evaluations are presented in accordance with monitoring

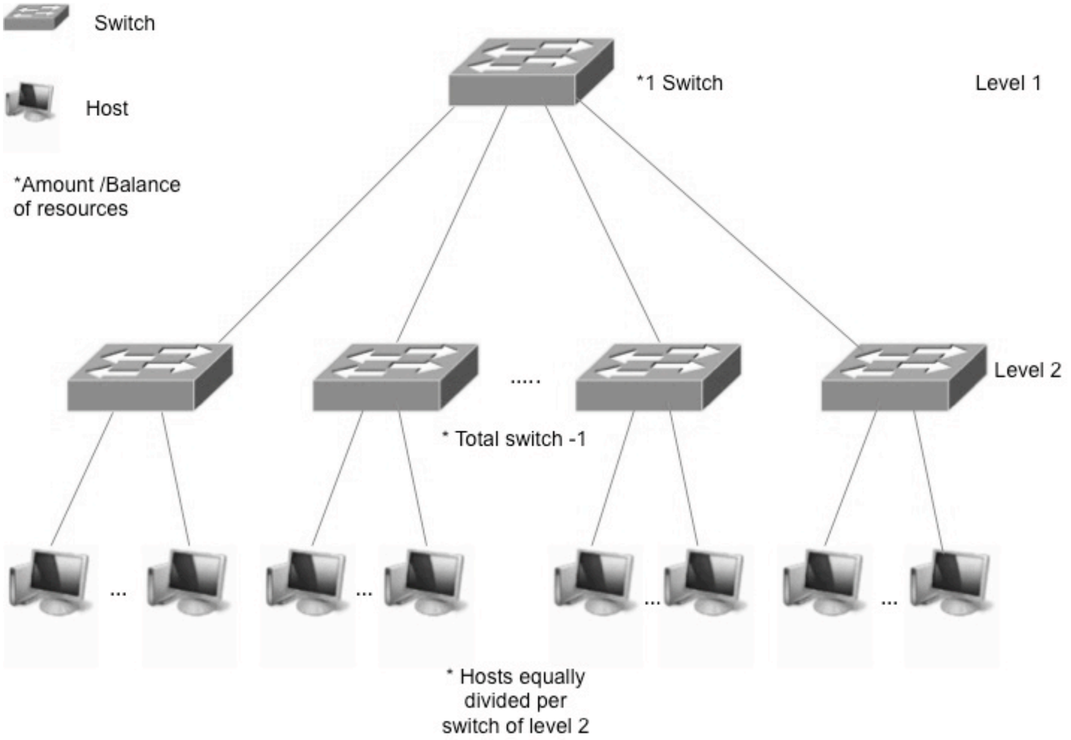

Switch
Host
*Amount /Balance of resources
*1 Switch
Level 1
.....
Level 2
* Total switch -1
...
...
...
...
* Hosts equally divided per switch of level 2

Figure 3.1: Topology 1, evaluated topology for until level 2 and extended to 64, 256, 576 and 1296 hosts.

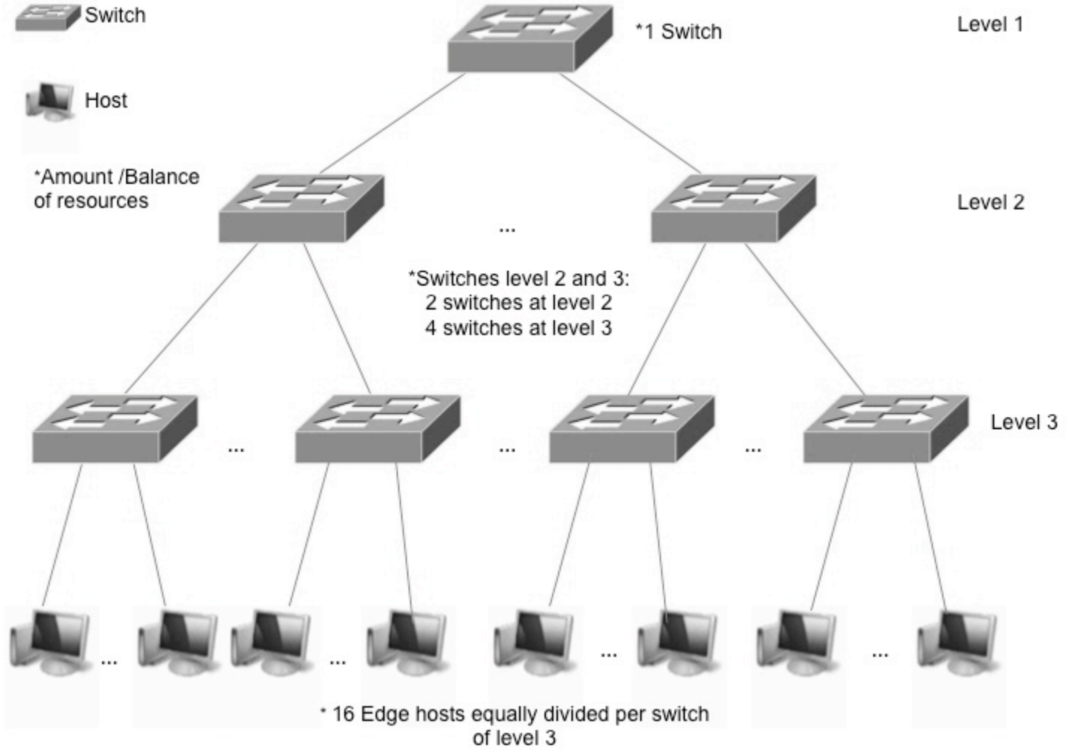


Figure 3.2: Topology 2, evaluated topology for until level 3 and extended to 64, 216, 512 and 1331 hosts.

parameters, namely, monitoring topologies, amount of monitoring data and frequency sampling.

3.3.1 Monitoring Topology

Monitoring topologies are structures used as a support for the collection and transferring of monitoring data. In our first set of experiments, we investigate the influence of monitoring topologies over the performance of a cloud monitoring system.

Evaluations are made by a comparison between Topology 1 and Topology 2 which were described in Chapter 3.3. Figure 3.3 presents the evaluations based on both the number of hosts and the average response time per topology to 120 bytes and 150 bytes of monitoring data. After all, evaluations to such monitoring topologies show that non-deep topologies (*i.e.,* Topology 1) present shorter response time than deep topologies (*i.e.,* Topology 2). Such affirmative is sustained when comparing monitoring topologies with two axes; firstly, based on the number of hosts, secondly, based on the amount of monitoring data.

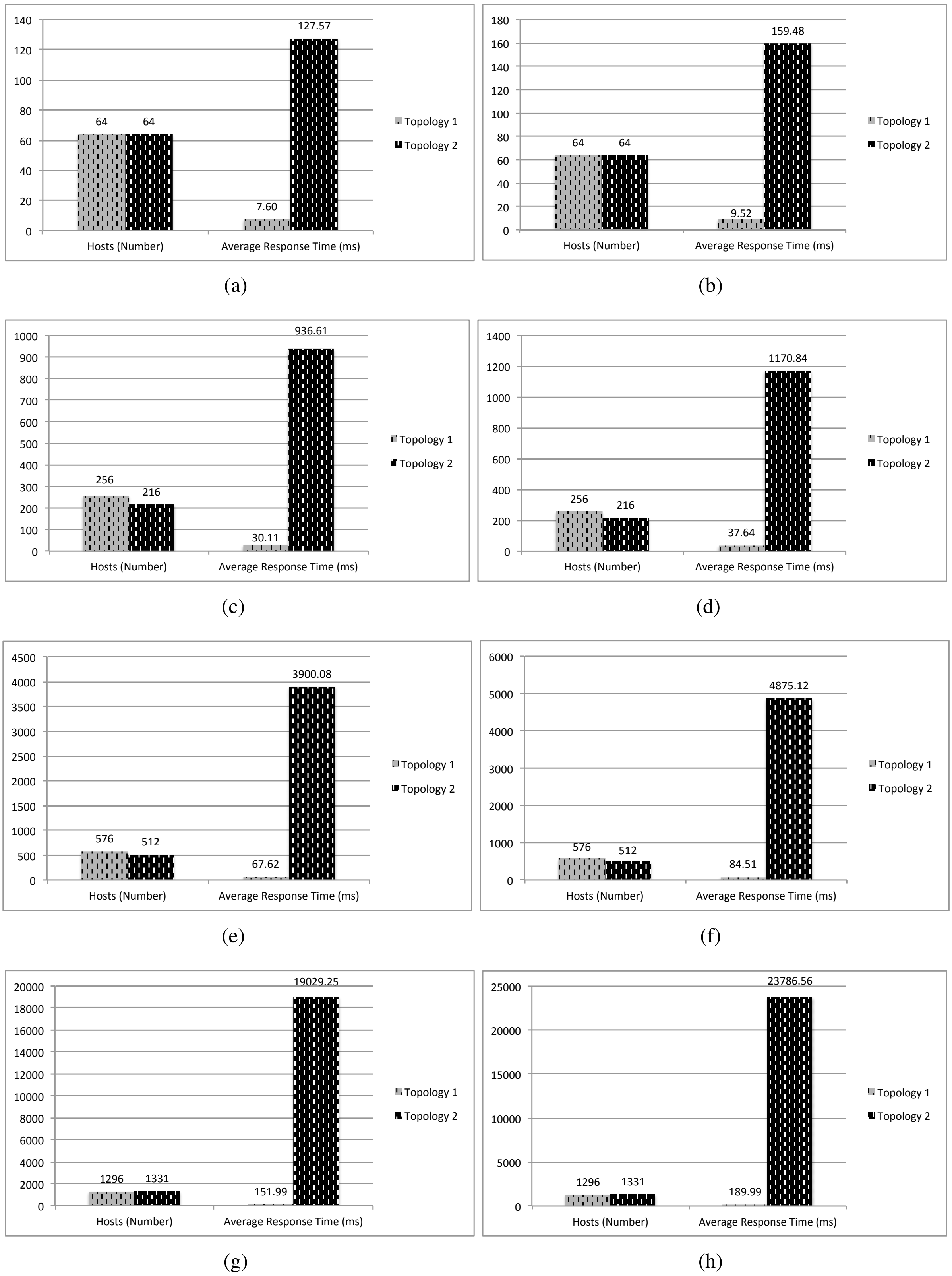
To explain the first axis let us analyse Figure 3.3. Figure 3.3 (a) and Figure 3.3 (g) show that the average response time is lower for Topology 1, regardless the number of hosts. In other words, both Figures show that the average response time is always lower to Topology 1 when comparing to Topology 2 because Topology 2 is the deepest among them. Such behaviour is confirmed when comparing other examples to the number of hosts such as between Figure 3.3 (c) and Figure 3.3 (e), and Figure 3.3 (b) and Figure 3.3 (d).

Figure 3.3 can also be used to analyse the second axis. When comparing Figure 3.3 (g) and Figure 3.3 (h) is observed that the average response time is lower for Topology 1, regardless the amount of monitoring data per sampling. It means that as well as to the first axis the performance of Topology 2 is worst when comparing to Topology 1 because Topology 2 is deep. This behaviour is strengthened when comparing other examples to the amount of monitoring data such as between Figure 3.3 (a) and Figure 3.3 (b), Figure 3.3 (c) and Figure 3.3 (d), and Figure 3.3 (e) and Figure 3.3 (f).

To explain this behaviour we sustain that non-deep topologies reduce replication of monitoring data in a network because core managers are closer to edge agents as well as close to intermediate managers. Besides, non-deep topologies reduce hops between edge agents and core managers which contributes to having less monitoring data on a network.

It is significant to highlight that monitoring topologies present a predictable increase for response time based on the amount of monitoring data per sampling.

For example, when analysing Figure 3.3 (a) and Figure 3.3 (b) is observed that the



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3.3: Evaluation of the number of hosts and average response time to 120 bytes and 150 bytes of monitoring data between Topology 1 and Topology 2.
To 120 bytes (a) (c) (e) (g) (*i.e.,* left side).
To 150 bytes (b) (d) (f) (h) (*i.e.,* right side).

average response time increased by 1,678.55% (*i.e.,* 7.60 ms to 127.57ms) between Topology 1 and Topology 2 with 120 bytes and 64 hosts as shown in Figure 3.3 (a). It is virtually the same increased in average response time (*i.e.,* 1,675.21%) to Topology 1 (*i.e.,* 9.52 ms) and Topology 2 (*i.e.,* 159.48 ms) with 150 bytes and 64 hosts (Figure 3.3 (b)).

Figure 3.3 (g) and Figure 3.3 (h) confirmed such behaviour because the average response time increased to 12,520.06% (*i.e.,* 151.99 ms to 19,029.25 ms) between Topology 1 and Topology 2 with 120 bytes and 1300 hosts as observed in Figure 3.3 (g). It is practically the same increased in average response time (*i.e.,* 12,519.90%) to Topology 1 (*i.e.,* 189.99 ms) and Topology 2 (*i.e.,* 23,786.56 ms) with 150 bytes and 1300 hosts as demonstrated in Figure 3.3 (h).

This increase happens continuously to others sets of experiments which include comparisons between Figure 3.3 (e) and Figure 3.3 (f), for example. Such experiments demonstrated that it is viable to predict the increase of response time using the amount of monitoring data per sampling as a parameter. Aiming to assess the impact of such increase in the next section we evaluated the amount of monitoring data in a particular way.

3.3.2 Amount of Monitoring Data

In this section, evaluations aim to unveil how the amount of monitoring data impairs response time in an isolated way. Such evaluations seek to verify the influence of monitoring data increase over timeliness without taking into account other parameters and, as a result, evaluations show how the monitoring data scalability impairs timeliness. Frequency sampling is not utilized because it impairs the evaluation to the amount of monitoring data as we explain in the Section 3.3.3.

Table 3.3, Table 3.4, and Table 3.5 present the results in terms of average response time for the amount of monitoring data for 120 bytes, 150 bytes, and 180 bytes without frequency sampling. Results show that the average response time is affected by the amount of monitoring data in accordance with the growing of a cloud monitoring system.

For example, the average response time is 30.11 ms for 120 bytes and it is 37.64 ms for 150 bytes of monitoring data to Topology 1 with 256 hosts as shown, respectively in Table 3.3 and Table 3.4. On the other side, to Topology 2 with 64 hosts, for example, the average response time is 127.57 ms for 120 bytes (Table 3.3), and it is 159.48 ms for 150 bytes of monitoring data (Table 3.4).

In other words, in both examples the average response time increased 25.01% virtually

Table 3.3: Average response time for 120 bytes of monitoring data without frequency sampling.

Topology	Hosts / Aggregators	RT Average (ms)
Topology 1	64 / 10	7.60
Topology 1	256 / 18	30.11
Topology 1	576 / 26	67.62
Topology 1	1296 / 38	151.99
Topology 2	64 / 21	127.57
Topology 2	216 / 43	936.61
Topology 2	512 / 73	3900.08
Topology 2	1331 / 133	19029.25

Table 3.4: Average response time for 150 bytes of monitoring data without frequency sampling and difference (%) to response time (RT Average) from Table 3.3.

Topology	Hosts / Aggregators	RT Average (ms)	Difference (%)
Topology 1	64 / 10	9.52	25.26%
Topology 1	256 / 18	37.64	25.01%
Topology 1	576 / 26	84.51	24.97%
Topology 1	1296 / 38	189.99	25.00%
Topology 2	64 / 21	159.48	25.01%
Topology 2	216 / 43	1170.84	25.00%
Topology 2	512 / 73	4875.12	25.00%
Topology 2	1331 / 133	23786.56	24.99%

Table 3.5: Average response time for 180 bytes of monitoring data without frequency sampling and difference (%) to response time (RT Average) from Table 3.3.

Topology	Hosts / Aggregators	RT Average (ms)	Difference (%)
Topology 1	64 / 10	11.48	51.05%
Topology 1	256 / 18	45.13	49.88%
Topology 1	576 / 26	101.41	49.97%
Topology 1	1296 / 38	227.97	49.99%
Topology 2	64 / 21	191.39	50.02%
Topology 2	216 / 43	1,404.87	49.99%
Topology 2	512 / 73	1,950.06	50.00%
Topology 2	1331 / 133	28,543.89	50.00%

the same difference of amount of monitoring data between 120 bytes and 150 bytes (*i.e.,* 25%). The behaviour is practically the same to all topologies and scenarios compared in Table 3.3 and Table 3.4, being the worst case to Topology 1 with 64 hosts where the average response time is increasing to 25.26%. Also, it happens when comparing Table 3.3 and Table 3.5.

For example, in Table 3.5, to Topology 1 with 1296 Hosts, the average response time is 227.97ms which means that it increased in 49.99% when comparing to the same Topology in Table 3.3. In this case, the average response time increased practically the same difference of amount of monitoring data between 120 bytes and 180 bytes which is 50%. The behaviour is practically the same to all topologies and scenarios compared in Table 3.3 and Table 3.5, being the worst scenario to Topology 1 with 64 hosts where the average response time is increasing to 51.05%.

This consistent behaviour demonstrates that the amount of monitoring data is increasing in a proportional way. It is significant because it allows the development of a mathematical model to estimates the mutual influence between timeliness and scalability using the amount of monitoring data as an input metric. In the next section, we evaluate other important monitoring parameter, namely, frequency sampling.

3.3.3 Frequency Sampling

Evaluations for frequency sampling aim to unmask issues such as: how the interval between data collection and response time impairs timeliness in accordance with the scalability of cloud monitoring systems. Frequency sampling is an important parameter because, depending on the frequency of data collection, the response time increases and, as a consequence, timeliness is impaired.

Table 3.6, Table 3.7, Table 3.8, and Table 3.9 present results in terms of average response time for variation of frequency sampling based on 120 bytes and 150 bytes. Evaluation results show that frequency sampling impairs response time when the frequency sampling is shorter that the response time. It happens when a new process of data collection starts, and the former data collection was not finished.

For example, Table 3.6 (*i.e.,* Topology 2, 512, and 1331 Hosts) shows that the average response time is increased by 290.00% and 1,803.00% respectively, comparing the average response time with the same configuration in Table 3.3 (*i.e.,* without considering frequency sampling). Also, it is observed in Table 3.8 (*i.e.,* Topology 2) which shows that the average response time is respectively increased by 17.00% to 256 Hosts, 387.50% to 512 Hosts, and 2,278.60% to 1331 Hosts when comparing to Table 3.4 (*i.e.,* without considering frequency sampling). At this point, it is important to highlight that Table 3.6 and Table 3.8 have presented the same interval sampling (*i.e.,* 1 second).

On the other hand, when the frequency sampling is bigger than the response time, the monitoring data in a network does not increase at the same time and, as a consequence, the response time practically remain constant.

For example, Table 3.6 (*i.e.,* Topology 2, 64 Hosts) has virtually the same average response time (*i.e.,* 127.93 ms) that the equivalent configuration in Table 3.3 (*i.e.,* 127.57 ms). This behaviour is the same to all topologies that present response time minor than 1000 mil-

Table 3.6: Average response time for 120 bytes of monitoring data with frequency sampling of 1 second and difference (%) between response time (RT Average) from Table 3.3.

Topology	**Hosts / Aggregators**	**RT Average (ms)**	**Difference (%)**
Topology 1	64 / 10	7.72	1.57%
Topology 1	256 / 18	30.07	- 0.13%
Topology 1	576 / 26	67.77	0.22%
Topology 1	1296 / 38	151.54	- 0.29%
Topology 2	64 / 21	127.93	0.28%
Topology 2	216 / 43	936.33	- 0.02%
Topology 2	512 / 73	15210.46	290.00%
Topology 2	1331 / 133	362127.72	1,803.00%

Table 3.7: Average response time for 120 bytes of monitoring data with frequency sampling of 10 seconds and difference (%) between response time (RT Average) from Table 3.3.

Topology	**Hosts / Aggregators**	**RT Average (ms)**	**Difference (%)**
Topology 1	64 / 10	7.67	0.92%
Topology 1	256 / 18	30.19	0.27%
Topology 1	576 / 26	67.54	-0.12%
Topology 1	1296 / 38	151.81	-0.12%
Topology 2	64 / 21	128.02	0.35%
Topology 2	216 / 43	936.32	-0.03%
Topology 2	512 / 73	3900.22	0.01%
Topology 2	1331 / 133	36155.62	90.00%

Table 3.8: Average response time for 150 bytes of monitoring data with frequency sampling of 1 second and difference (%) between response time (RT Average) from Table 3.4.

Topology	**Hosts / Aggregators**	**RT Average (ms)**	**Difference (%)**
Topology 1	64 / 10	9.44	-0.84%
Topology 1	256 / 18	37.71	0.19%
Topology 1	576 / 26	84.38	-0.15%
Topology 1	1296 / 38	189.67	-0.17%
Topology 2	64 / 21	159.77	0.18%
Topology 2	216 / 43	1369.91	17%
Topology 2	512 / 73	23766.35	387.50%
Topology 2	1331 / 133	565787.41	2,278.60%

liseconds (*i.e.,* 1 second) in Table 3.3 when comparing to both Table 3.6 e Table 3.8.

Table 3.7 and Table 3.9 present results when frequency sampling is set to 10 seconds. In Table 3.7 (*i.e.,* Topology 2, 512 Hosts), we noticed that the average response time (*i.e.,* 3900.22 ms) is virtually the same in Table 3.3 (*i.e.,* 3900.08 ms) because the response time is shorter than the frequency sampling.

In Table 3.7 (*i.e.,* Topology 2, 1331 Hosts), we observe that the average response time (*i.e.,* 36,155.62 ms) increases because the frequency sampling is shorter than the average response time (*i.e.,* 19,029.25 ms) in Table 3.3. It is also observed when comparing Table 3.9

Table 3.9: Average response time for 150 bytes of monitoring data with frequency sampling of 10 seconds and difference (%) between response time (RT Average) from Table 3.4.

Topology	Hosts / Aggregators	RT Average(ms)	Difference (%)
Topology 1	64 / 10	9.56	0.42%
Topology 1	256 / 18	37.51	-0.34%
Topology 1	576 / 26	84.59	0.09%
Topology 1	1296 / 38	189.91	-0.04%
Topology 2	64 / 21	159.11	-0.23%
Topology 2	216 / 43	1170.45	-0.03%
Topology 2	512 / 73	4875.58	-0.01%
Topology 2	1331 / 133	56374.18	137.00%

with Table 3.4. For example, in Table 3.9 (*i.e,* Topology 2, 216 and 512 Hosts) we observed that the average response time is practically the same presented in Table 3.4 whereas to Topology 2 with 1331 Hosts it increased in 137.00%. In other words, it increased from 23,786.56 ms (Table 3.4) to 56,374.18 ms (Table 3.9).

In addition, when comparing Table 3.7 (*i.e.,* Topology 2, 1331 Hosts) with the equivalent configuration in Table 3.6, we realized that the average response time (*i.e.,* 362127.72 ms) is bigger in Table 3.6 because the frequency sampling is 10 times short (*i.e.,* 1 second in Table 3.6 and 10 seconds in Table 3.7). This behaviour is confirmed when comparing Topology 2 with 1331 Hosts in both Table 3.8 and Table 3.9 that respectively presented 565,787.41 ms and 56,374.18 ms.

These results demonstrated that shorter intervals for frequency sampling impairs timeliness. The impact of frequency sampling over response time grows with the increase of number of probes. It happens because, when the monitoring system grows, the amount of resources to be monitored increases. Hence, we observed more precisely the frequency sampling influence in large topologies.

Finally, it is significant to highlight that the evaluations to the influence of frequency sampling as well as to both monitoring topologies and the amount of monitoring data present a predictable increase in response time which is liable to be mathematically modelled.

3.4 Remarks

The acquaintance of the mutual influence between timeliness and scalability provides support to significant issues such as assisting the design and development of comprehensive and integrated cloud monitoring systems, supporting future evaluations among others requirements or abilities (*e.g.,* adaptability, elasticity, accuracy), increasing profits for infrastructure and service providers based on predictions of resources usage in monitoring, assisting cloud providers to satisfy SLAs based on timeliness, and assisting infrastructure providers to assess its structure.

Evaluations demonstrated the mutual influence between timeliness and scalability in regard to different monitoring parameters such as monitoring topologies, amount of monitoring data, and frequency sampling. Thus, cloud operators (*e.g.,* SPs, InPs) could be more aware of the mutual influence between timeliness and scalability based on such monitoring parameters.

Results for monitoring parameters presented in this section demonstrated that cloud providers could reduce the mutual influence between timeliness and scalability via at least two different methods based on the balance between both requirements.

In the first method, cloud providers could reduce as much as possible the frequency sampling in accordance with SLAs as presented in Section 3.3.3. In the second method, when negotiating SLAs, cloud providers could take into account the depth of the infrastructure to set the response time. For example, to increase profit, a cloud provider could own different infrastructures with different topologies to support different and more restricts SLAs for top customers, thus, providing to them better response time and accomplishing timeliness.

Additionally, the results showed that mutual influence between timeliness and scalability can be quantified based on monitoring parameters such as monitoring topologies, the amount of monitoring data, and frequency sampling. Therefore, the mutual influence between timeliness and scalability is liable to mathematically modelled.

3.5 Summary

This chapter discussed the mutual influence between timeliness and scalability. Firstly, It presented research efforts that aim to propose monitoring requirements to cloud monitoring systems. Such research efforts have lacked in a depth discussion on the mutual influence of monitoring requirements. After that, this chapter introduced the mutual influence of timeliness and scalability as a starting point for research in this field. Secondly, it showed a quantitative evaluation based on monitoring parameters that simulate the influence of scalability over timeliness in a cloud monitoring system. The evaluation demonstrated that the mutual influence between both cloud monitoring requirements is liable to be mathematically modelled. In the next Chapter such mathematical model is proposed and evaluated.

4 PREDICTING THE INTERPLAY BETWEEN TIMELINESS AND SCALABILITY

Timeliness and scalability are two cloud monitoring requirements that have direct relation. To cloud operators, when unveiling the mutual influence between timeliness and scalability is useful to enhance their quality of service. For example, cloud operators can propose SLAs to customers based on the relation between such requirements to avoid breaches in SLAs, and as a consequence increase their profits. According to Rodrigues *et.al* (RODRIGUES et al., 2015), this mutual influence can be predicted based on monitoring parameters. However, it has lacked on a mathematical comprobation.

The aims of this chapter are propose and evaluate a mathematical model for estimate the mutual influence between timeliness and scalability. It begins discussing the reasons because each monitoring parameter is meaningful to the interplay between timeliness and scalability. After that, it demonstrates how the mathematical model is developed based on monitoring metrics along with network bandwidth and response time (RT) as output metric. Next, this chapter compares the results provided by a simulation that was described in Chapter 3 with results obtained via the mathematical model to evaluate the effectiveness of the proposed solution to conventional data center networks. Moreover, this chapter presents the results provided by a simulation based on Fat-tree topologies. It compares such results with outcomes obtained via the mathematical model to extend the analysis of the proposed solution. Also, this chapter discusses the applicability of the mathematical model based on three axes: Enhancing SLA's and Billing, Saving Energy, and Raising Profits to Cloud Operators and Reducing Costs to Customers. Finally, it shows final remarks about the obtained results.

4.1 Tackling The Interplay Between Timeliness and Scalability

Previous research presented a wide discussion about the interplay among cloud monitoring requirements/abilities such as adaptability, accuracy, elasticity, timeliness, and scalability (ACETO et al., 2013) (MONTES et al., 2013) (PARK et al., 2011) (RODRIGUES et al., 2014)

In Chapter 3, we demonstrated that the mutual influence between timeliness and scalability is liable to be mathematically modelled based on monitoring parameters, namely, monitoring topology, amount of monitoring data and frequency sampling. Moreover, network bandwidth must be considered along with response time as an output metric. In this section, we analysed all of these factors (*i.e.,* monitoring parameters, network bandwidth, response time) to discuss their significance on the interplay between timeliness and scalability.

Monitoring topology is vital because it contributes to boosting the communication delay in a cloud monitoring system. It occurs as a result of the distance from agents to managers, which increases the time spent in the process of communication in a network depending on both, the placement of such agents and managers, and the number of communication links.

The amount of monitoring data is critical because it stimulates delay between event occurrence and warning. It happens as a consequence of the growing of the cloud monitoring system that has more monitoring data to be gathered and managed. As a result, timeliness is impaired when the cloud monitoring system escalates to cope suitably with the cloud infrastructure.

Frequency sampling is an essential monitoring parameter because, when sampling in higher frequency, the amount of monitoring data is increased in a network, causing communication delay. Thus, timeliness is impaired when a cloud monitoring system is gathering samples in tiny intervals of time. It usually happens to resources such as CPU that has to be continuously monitored in narrow periods of time. In this scenario, to perform frequency sampling is important to unmask how the interval between data collection and response time impairs timeliness in accordance with the scalability of a cloud monitoring system.

Network bandwidth must be considered to calculate the mutual influence between timeliness and scalability because it consists of physical links among a plenty of resources provided by the cloud infrastructure. In others words, it represents the capacity that a cloud has in terms of data transmission among its peers.

Response time must be used as the output metric because it means the time spent on the data collection and notification. Therefore, it represents the quantification of the influence of scalability over timeliness and vice-versa.

In the next section, we propose to join all this metrics in a mathematical formulation to create a generic model to predict the mutual influence between timeliness and scalability in cloud monitoring systems. Such mathematical formulation is a step forward in cloud management because it provides support to both infrastructure and cloud providers to enhance its services and, as a result, to raise their profits and improve their quality of service (QoS).

4.2 Depicting The Mathematical Model

In this section, we demonstrate how the mathematical model is developed based on monitoring metrics along with network bandwidth and response time (RT) as output metric.

An instance of this mathematical model is response time that is considered the output metric or the result of such mathematical model. Let us define *rt* as response time. We show that *rt* is characterized by a, which is the amount of monitoring data gathered per sampling; *b*, which is the network bandwidth capacity to transferring bytes per second in the cloud; *h*, which is the amount of hosts in a same level; *fs*, which is the frequency sampling.

First of all, let keep apart the communications traffic generated by a cloud monitoring system from ordinary traffic in a cloud infrastructure. After that, we will divide the equation in short equations that are called steps aiming to present the whole equation in tiny parts. To maintain consistency, all equations in this section assume an amount of monitoring data of 120 bytes and a network bandwidth of 1Gbps, as an example.

First step, let us define *rt* as an instance that is the result of Equation 4.1, which is based on a that represents the amount of monitoring data to be transferred divided by *b*, which is the network bandwidth capacity to transfer bytes per second. As a result of Equation 4.1, we have response time (*rt*) based on the communication between 2 hosts shown in milliseconds (ms). Figure 4.1 depicts such communication between a manager placed at switch A and an agent placed at host B, as an example. To such example, shall the result is 0.1171 ms to response time (*rt*), which represents only the traffic generated by monitoring a cloud.

$$rt = \frac{a}{b} \quad [ms] \tag{4.1}$$

Second step, let us introduce *h*, which is the amount of hosts in a same level. The *rt* is the result of Equation 4.2, which is the result of step 1 (a divided by b) added by a multiplied by h and divided by b. If considered as an example, 16 edge hosts divided in 4 switches at the second level and 1 switch at first level. Shall the result of Equation 4.2 is 1.9921 ms, which represents response time *rt*, from agents deployed at edge hosts to a manager deployed on the switch at first level, to a topology of a two-level tree of routers or switches, without considering frequency sampling (*fs*). Figure 4.2 depicts this topology of a two-level tree.

$$rt = \frac{a}{b} + \frac{a * h}{b} \quad [ms] \tag{4.2}$$

Aiming to explain the Equation 4.2, let us use Equation 4.1, whose result is equal to *rt* between all hosts edge and their respective managers deployed in switches at level 2. Switches at level 2 gathered monitoring data from all agents deployed at edge hosts of their branches. It justified *a*h* in Equation 4.2 to find out the amount of monitoring data gathered in all switches deployed at level 2. This monitoring data is, posteriorly, forwarded to the core manager deployed in the switch at level 1.

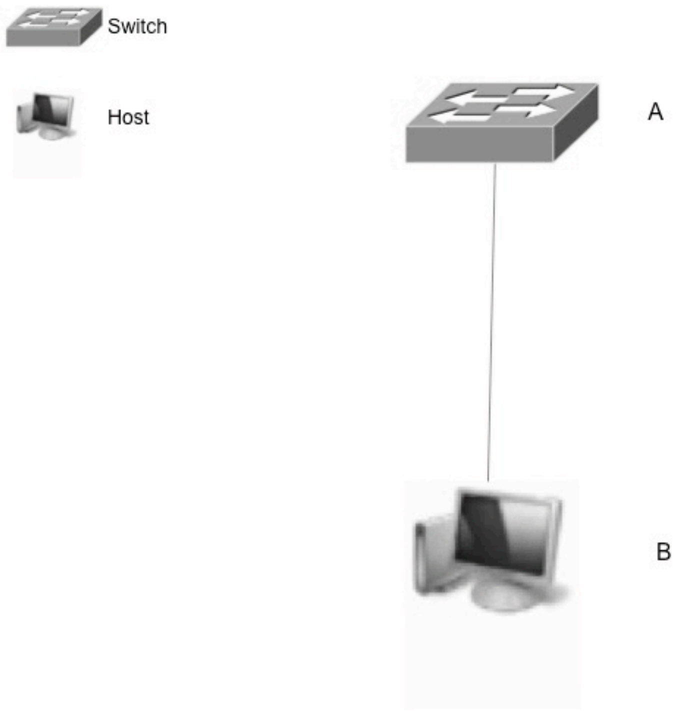


Figure 4.1: Link between switch A and host B.

From the above, the result of Equation 4.2 is a predictable response time (*rt*) from edge hosts to the manager deployed at level 1 without considering frequency sampling *fs*.

Third step, let us introduce *fs*, which is the frequency sampling. The *rt* is the result of Equation 4.3, which is a multiplied by *fs* divided by b. Considering as an example that three samples are processed at the same time. The result of Equation 4.3, is 0.3515 ms, which represents *rt* based on the communication between switch A and host B as depicted in Figure 4.1, including three as *fs*.

$$rt = \frac{a * fs}{b} \quad [ms] \tag{4.3}$$

To explain Equation 4.3, we have to take into consideration that frequency sampling is necessary to unmask how the interval between data collection and response time impairs timeliness. It happens when a sampling starts, and the previous samplings are still in the process (data collection still in process on the network). In this context, we considered as an example that three samples are collected in a same gap of time when the former sample is not finalized and the second and third are still in process.

Fourth step, let us introduce h along with *fs*. The *rt* is the result of Equation 4.4, which is the result of Equation 3 (a multiplied by *fs* divided by b) added by a multiplied by *fs* and h and, posteriorly, divided by b. If considered as an example, 16 edge hosts divided in 4 switches

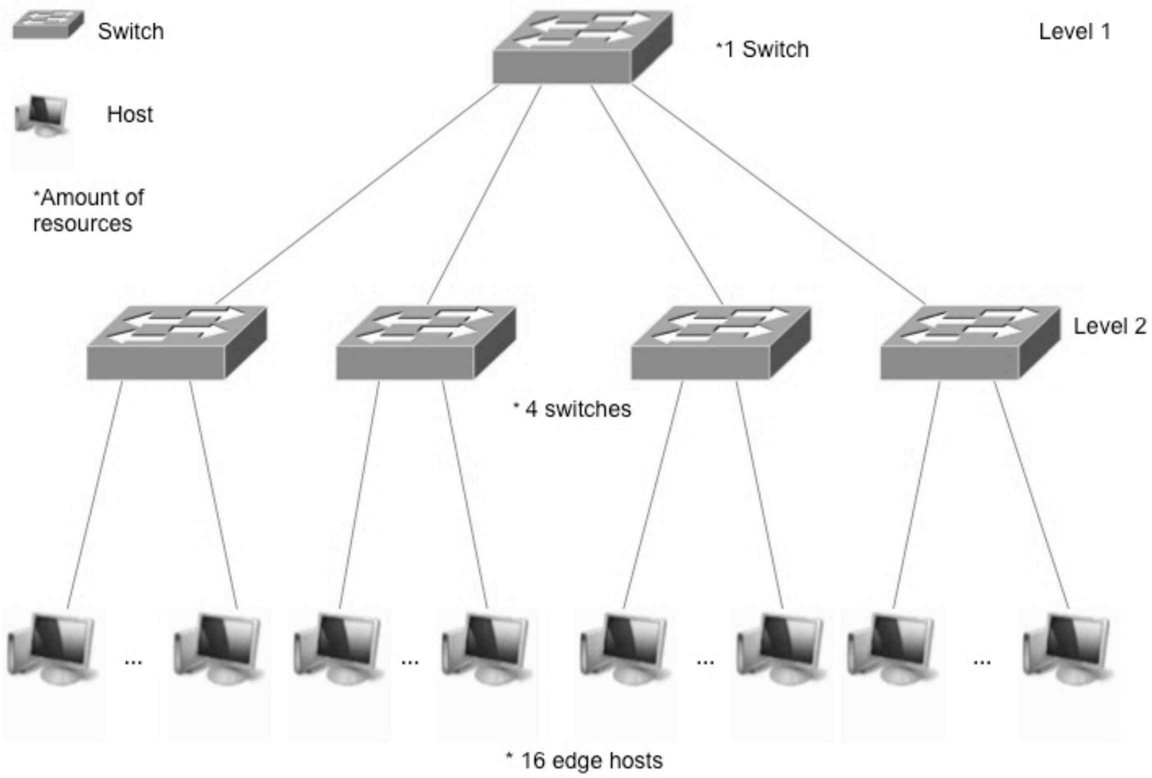


Figure 4.2: Topology 1 used as an example to second step with 16 edge hosts, 4 switches at level 2 and 1 switch at level 1.

at the second level and 1 switch at first level along with three samples that are processed at the same time. Shall, the result of Equation 4.4, is 5,9765 ms, which represents response time (*rt*), from edge hosts to a manager deployed on the switch at first level, to a topology of a two-level tree of routers or switches. Figure 4.2 depicts such topology of a two-level tree.

$$rt = \frac{a * fs}{b} + \frac{a * fs * h}{b} \quad [ms] \tag{4.4}$$

Aiming to explain the Equation 4.4, let us use Equation 4.3 because the result of Equation 4.3 is equal to response time *rt* between all hosts edge and their respective managers deployed in switches at level 2 considering frequency sampling (*fs*). Switches at level 2 gathered monitoring data from all edge hosts of their branches. It justified *a*fs*h* in Equation 4.4 to find out the amount of monitoring data gathered in all switches deployed at level 2, considering the impairing of frequency sampling at this level. This monitoring data is, posteriorly, forwarded to the core manager deployed in the switch at level 1. It explains the necessity to consider the traffic among the switches at level 2 and the switch at level 1.

From the above, the result of Equation 4.4 is the predictable response time (*rt*) from edge hosts to the manager deployed at level 1 considering all values used as an example in this section. Therefore, such result is useful, like the predicted response time (*rt*), to the topology depicted in Figure 4.2. Equation 4.4 presents a solution to topologies like two-level trees. However, to deep cloud infrastructures it is not useful at all. For example, topologies of three-level trees have to be exploited.

Fifth step stretches the mathematical model to solve three-level trees. Let us introduce l as a variant to h. Response time *rt* is the result of Equation 4.5, which is the result of Equation 4.3 (a multiplied by *fs* and divided by b) added by a multiplied by *fs* and *h* to level 3 (*l2*), which is the amount of hosts/switches in the level 3 (to represent the aggregated monitoring data at this point), posteriorly, divided by b and added by a multiplied *fs* and and h, which is the amount of hosts, posteriorly, divided by b.

If considered as an example, 16 agents deployed at edge hosts divided in 4 switches at third level, which are divided equally by 2 switches at second level and 1 switch at first level along with three samples that are processed at the same time. Shall the result of Equation 4.5, is 28,4765 ms, which represents response time (*rt*), from edge hosts to a manager deployed on the switch at first level, to topology of a three-level tree of routers or switches. Figure 4.3 depicts such topology of three-level tree.

$$rt = \frac{a * fs}{b} + \frac{a * fs * h_{l2} * h}{b} + \frac{a * fs * h}{b} \quad [ms] \tag{4.5}$$

To provide a readable version of the Equation 4.5, it is summarized in Equation 4.6. Summarizing, a and *fs* that appeared in all parts of the Equation 4.5 are put in evidence in the Equation 4.6, and they are multiplied by *1*, *h*, and *h*hl2* , respectively, each of them representing parts of the Equation 4.5. After that, *b* divided the result because it is dividing in all parts of the Equation 4.5.

$$rt = \frac{a * fs(1 + h + h * h_{l2})}{b} \quad [ms] \tag{4.6}$$

Conventional data center networks for clouds are based on either two or three level trees (See Section 2.3.1). Then, Equation 4.6 is suitable to estimate the mutual influence between timeliness and scalability in conventional data center networks to three level trees whereas Equation 4.4 is proper to deal with two level ones. However, it has to adjust to working properly with other common topologies for clouds such as Fat-tree.

Fat-tree is a special type of Clos Topology (See Section 2.3.3), and it is a topology based on trees. A usual Fat-tree topology is a three level tree with more communication channels (See

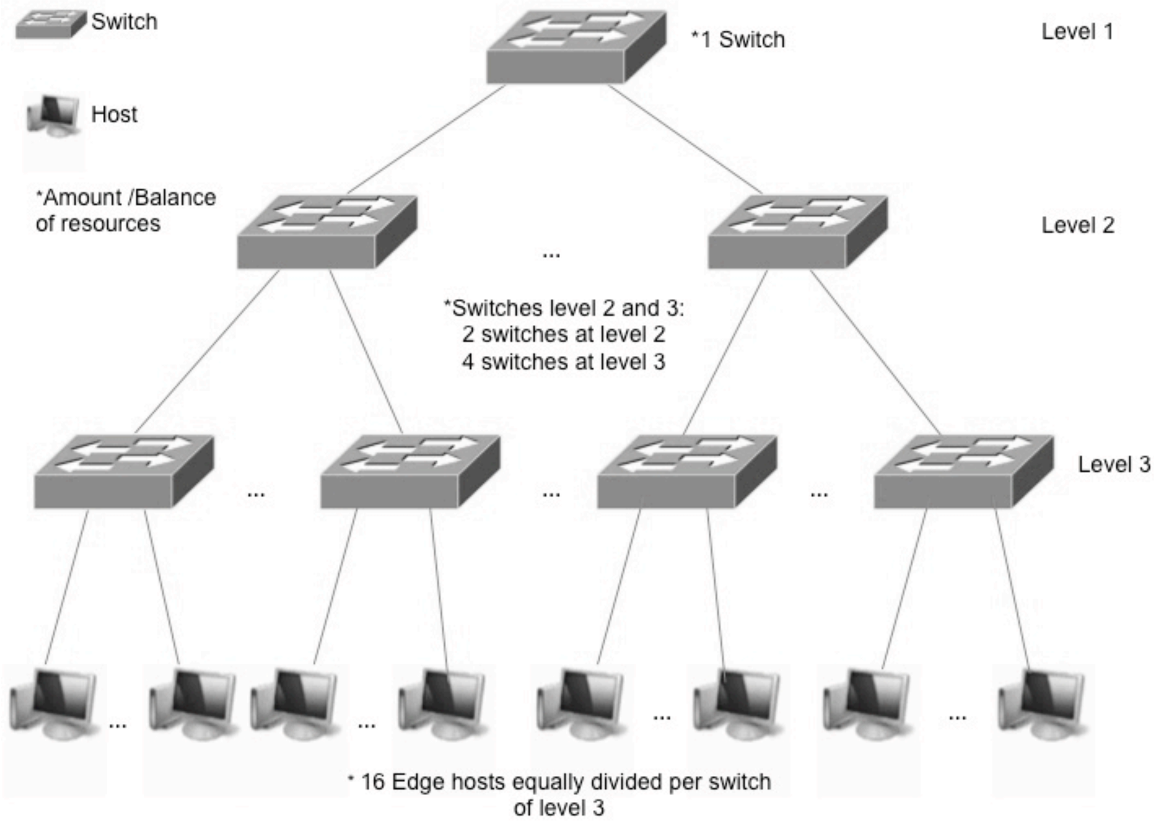


Figure 4.3: Topology 2 used as an example to fifth step with 16 edge hosts, 4 switches at level 3, 2 switches at level 2 and 1 switch at level 1.

Figure 2.6). Thus, to adjust the Equation 4.6 we have to take into consideration the amount of communication channels provided which is represented by *ac* (Amount of Channels).

$$rt = \frac{a * fs(1 + \dfrac{h}{ac} + \dfrac{h * h_{l2}}{ac})}{b} \quad [ms] \tag{4.7}$$

Equation 4.7 includes *ac* to levels Edge and Aggregation of a Fat-tree topology. It is necessary because both levels have at least two communication channels with their upper levels reducing the response time.

These mathematical models presented in Equation 4.6 and Equation 4.4 are useful to predict the mutual influence between timeliness and scalability and vice-versa in a conventional data center network. If necessary, it can be easily adjusted to estimate such mutual influence in Fat-Tree topologies as we demonstrated in Equation 4.7.

Aiming to assess these mathematical models in the next section, we compare such models with the results of a simulation presented in Chapter 3 to prove that it is suitable for conventional data center networks. Besides, we performed experiments in a Fat-Tree topology to evaluate the adjustment to fit the mathematical model to this topology.

4.3 Evaluating The Mathematical Model

In this section, we demonstrate that the mathematical model is useful to predict the mutual influence between timeliness and scalability. It is composed of two sets of evaluations. Firstly, this section evaluates the behaviour of the mathematical model in monitoring topologies based on conventional data center networks. In a second moment, it evaluates such behaviour in a monitoring topology based on Fat-tree.

4.3.1 Evaluating Monitoring Topologies Based on Conventional Data Center Network

In Chapter 3 we evaluated the interplay between timeliness and scalability. This evaluation was a simulation that provided results to such interplay based on response time (RT). The evaluation unmasked the mutual influence between timeliness and scalability to topologies based on trees like conventional data center networks with either two or three level trees. It provides results that are significant to the development of the mathematical model that aims to predict the mutual influence between such requirements. In this scenario, if the mathematical model presents similar results comparing to the simulation, it shows to be useful and suitable to conventional data center network topologies.

In this section, we compare the results provided by simulation with results obtained by the mathematical model. To organize such comparison in this section we define the results as follow:

- Predicted RT: It is the result for response time (RT) obtained by the mathematical model.
- Average RT: It is the result for average response time (RT) obtained by the simulation.

Table 4.1, Table 4.2, Table 4.3 and Table 4.4 present the results to predicted RT and average RT for the amount of monitoring data for 120 bytes and 150 bytes with frequency sampling based on an interval of 1 and 10 seconds.

Table 4.1 and Table 4.2 compare results to predicted RT and average RT for 120 bytes of monitoring data with interval sampling of 1 and 10 seconds. To Table 4.1 the results show that predicted RT and average RT are equal to topology 2 with 512 hosts. Moreover, we highlight that the most relevant difference among results is to topology 1 with 64 hosts that are 1.29%. Results demonstrate that the behaviour of the mathematical model is compatible with the simulation as observed in Figure 4.4 (a) which bars to predict RT considers 3% as a margin of

Table 4.1: Predicted and Average response time (RT) for 120 bytes of monitoring data with interval sampling of 1 second.

Topology	**Hosts/Aggregators**	**Predicted RT (ms)**	**Average RT (ms)**
Topology 1	64 / 10	7.62	7.72
Topology 1	256 / 18	30.12	30.07
Topology 1	576 / 26	67.62	67.77
Topology 1	1296 / 38	151.99	151.54
Topology 2	64 / 21	127.62	127.93
Topology 2	216 / 43	936.68	936.33
Topology 2	512 / 73	15210.46	15210.46
Topology 2	1331 / 133	362127.78	362127.72

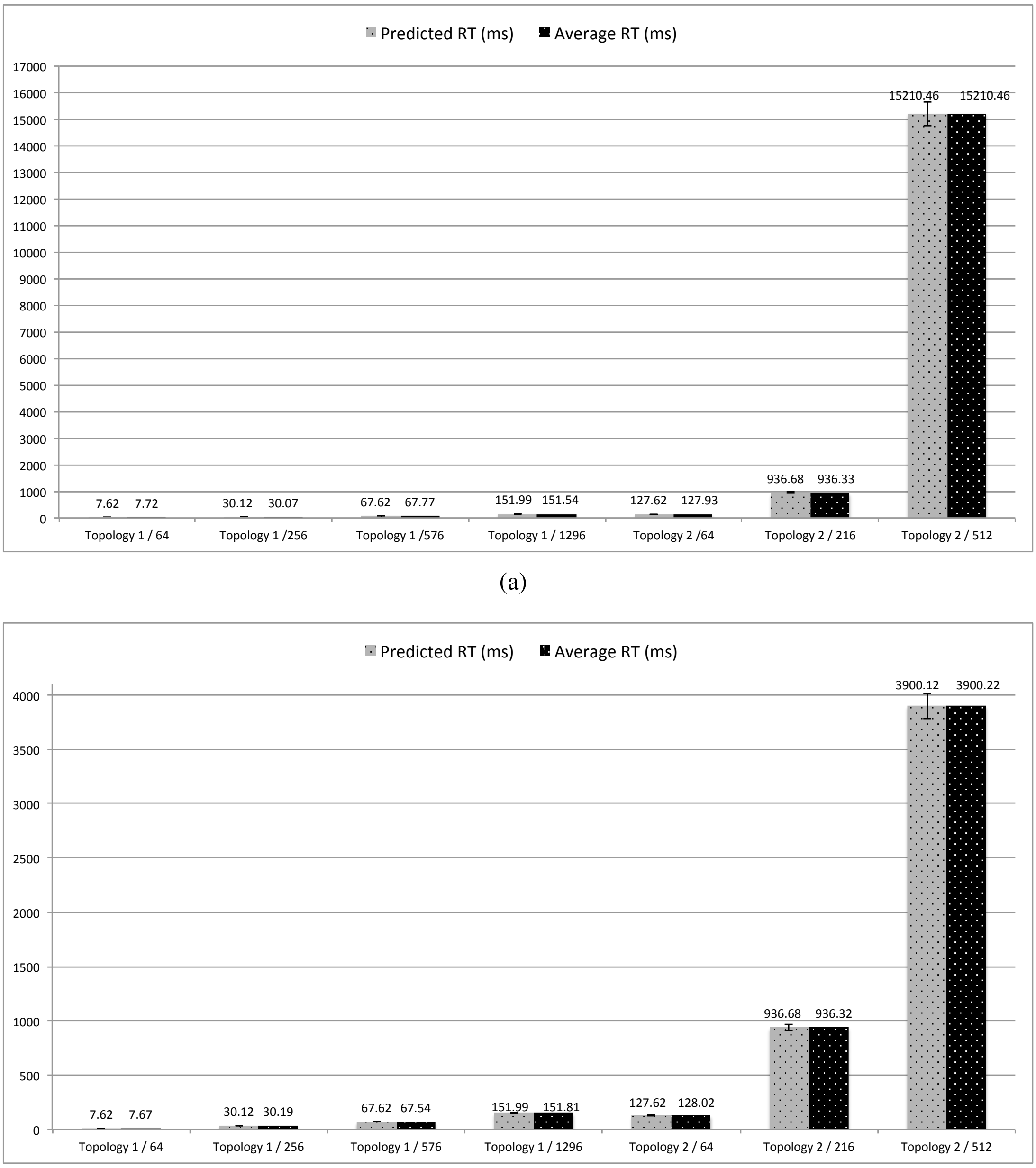
Table 4.2: Predicted and Average response time (RT) for 120 bytes of monitoring data with interval sampling of 10 seconds.

Topology	**Hosts/Aggregators**	**Predicted RT (ms)**	**Average RT (ms)**
Topology 1	64 / 10	7.62	7.67
Topology 1	256 / 18	30.12	30.19
Topology 1	576 / 26	67.62	67.54
Topology 1	1296 / 38	151.99	151.81
Topology 2	64 / 21	127.62	128.02
Topology 2	216 / 43	936.68	936.32
Topology 2	512 / 73	3900.12	3900.22
Topology 2	1331 / 133	36155.59	36155.62

error. It demonstrate that the mathematical model is useful to 120 bytes of monitoring data with interval sampling of 1 second, considering 1.29% as a maximum margin of error.

In Table 4.2 the results demonstrate that predicted RT and average RT are close in topology 2 with 216, 512 and 1331 hosts. It provides indications that to deep topologies the mathematical model reaches results with small margins of error, in this example 0.04%, 0.01% and 0.01%, respectively. Additionally, the most significant difference among results is to topology 1 with 64 hosts (0.65%). It evidences that the mathematical model results are consistent with the simulation as demonstrated in Figure 4.4 (b) which bars to predict RT considers 3% as a margin of error. Therefore, the mathematical model is useful to 120 bytes of monitoring data with interval sampling of 10 seconds, considering 0.65% as a maximum margin of error.

Table 4.3 and Table 4.4 assess results to predicted RT and average RT for 150 bytes of monitoring data with interval sampling of 1 and 10 seconds. Table 4.3 shows that predicted RT and average RT are closer in deep topologies when compared to results from topology 1 and topology 2. It strengthens that the mathematical model is more accurate to deep topologies and reiterates that this model is more efficient in larger environments if compared to smaller ones. Moreover, we highlight that the most relevant difference among results is to topology 1 with 64 hosts (0.85%), which is the smaller environment evaluated. The mathematical model results



(a)

(b)

Figure 4.4: Comparison between predicted and average response time to 120 bytes of monitoring data with interval sampling to 1 and 10 seconds. (a) 1 Second . (b) 10 Seconds.

(predicted RT) are coherent with the simulation results (average RT) as showed in Figure 4.5 (a) which bars to predict RT considers 3% as a margin of error. It attests that the solution proposed is useful to 150 bytes of monitoring data with interval sampling of 1 second, considering 0.85% as a maximum margin of error.

Table 4.4 confirms that predicted RT and average RT are closer to larger topologies such as topology 2 with 216 hosts (0.03%), 512 hosts (0.01%) and 1331 hosts (equal). It demonstrates that the mathematical model is more accurate when applied to larger topologies. On the other hand, if applied to smaller topologies the mathematical model presents results with

Table 4.3: Predicted and Average response time (RT) for 150 bytes of monitoring data with interval sampling of 1 second.

Topology	**Hosts/Aggregators**	**Predicted RT (ms)**	**Average RT (ms)**
Topology 1	64 / 10	9.52	9.44
Topology 1	256 / 18	37.65	37.71
Topology 1	576 / 26	84.52	84.38
Topology 1	1296 / 38	189.99	189.67
Topology 2	64 / 21	159.52	159.77
Topology 2	216 / 43	1369.89	1369.91
Topology 2	512 / 73	23766.34	23766.35
Topology 2	1331 / 133	565787.41	565787.41

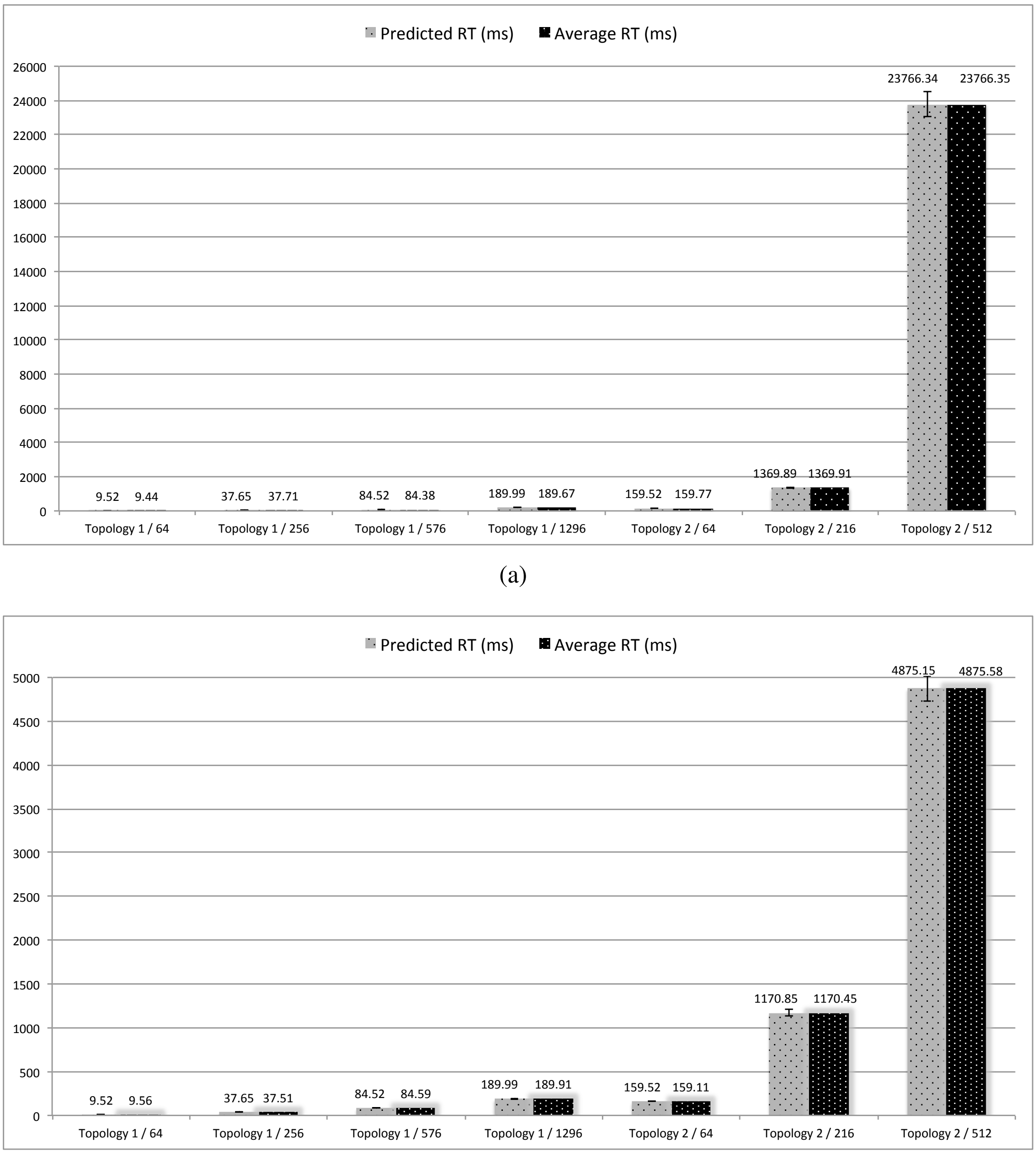
Table 4.4: Predicted and Average response time (RT) for 150 bytes of monitoring data with interval sampling of 10 seconds.

Topology	**Hosts/Aggregators**	**Predicted RT (ms)**	**Average RT (ms)**
Topology 1	64 / 10	9.52	9.56
Topology 1	256 / 18	37.65	37.51
Topology 1	576 / 26	84.52	84.59
Topology 1	1296 / 38	189.99	189.91
Topology 2	64 / 21	159.52	159.11
Topology 2	216 / 43	1170.85	1170.45
Topology 2	512 / 73	4875.15	4875.58
Topology 2	1331 / 133	56374.18	56374.18

small margin of error such as in topology 1 with 64 hosts (0.42%). The Figure 4.5 (b) shows the behaviour of the solution compared to the simulation considering 3% as a margin of error to predicted RT. Finally, the solution proves to be useful to 150 bytes of monitoring data with interval sampling of 10 seconds, considering 0.42% as a maximum margin of error.

In this section, the comparison between predicted RT and average RT demonstrates that the proposed mathematical model is useful to predict the mutual influence between timeliness and scalability to conventional data center network topologies. This model presents accurate results to larger and deep topologies when comparing to smaller and non-deep ones which is desirable to massive environments such as clouds. On the other side, to smaller environments the mathematical model shows to be useful because it reaches closer results with small margin of error.

Nevertheless, the mathematical model must be evaluated based on other common cloud network topology such as Fat-tree. In the next section, we perform a set of tests to assess the behaviour of the mathematical model in a Fat-tree topology to fulfil this gap.



(a)

(b)

Figure 4.5: Comparison between predicted and average response time to 150 bytes of monitoring data with interval sampling to 1 and 10 seconds. (a) 1 Second . (b) 10 Seconds.

4.3.2 Evaluating Monitoring Topologies Based on Fat-Tree

In the previous section, we compared the results provided by simulation with results obtained by the mathematical model to conventional data center networks. Aiming to expand the evaluation in this section we present experiments based on Fat-tree topologies and compare the results obtained with the mathematical model.

A third monitoring topology is built based on a Fat-tree topology consisting of three levels of switches (See Section 2.3.3) (BARI et al., 2013) to perform the experiments. To each switch is added one aggregator as deployed in Topology 1 and Topology 2. This monitoring topology is depicted in Figure 4.6, and it is defined as Topology 3. Topology 3 is evaluated to timeliness based on the addition of pods, and it was extended to 4, 6, 8, 10, and 12 pods (*e.g.,* K= 4). Moreover, Topology 3 was evaluated to both 120 and 150 bytes of monitoring data, and frequency sampling of 1 second as performed in the previous topologies to maintain consistency.

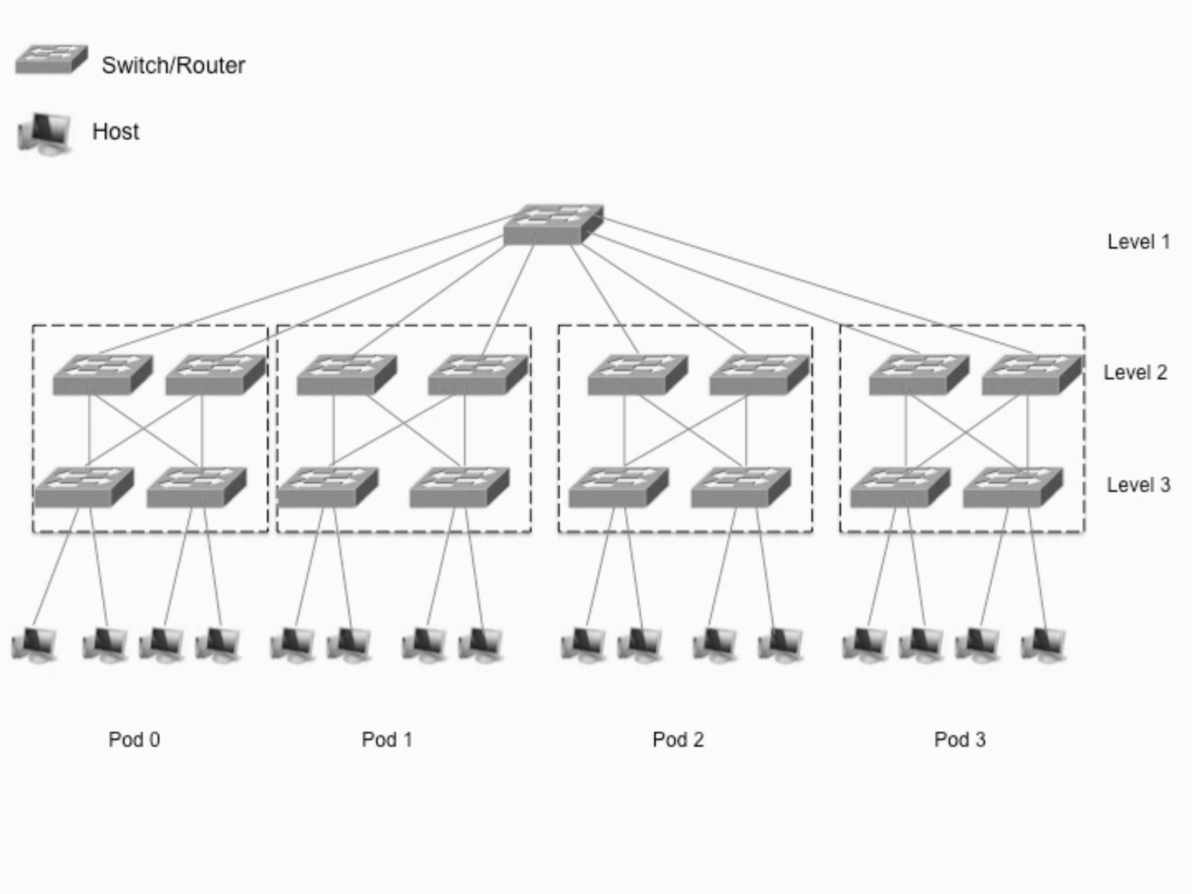


Figure 4.6: Topology 3, a monitoring topology based on a Fat-tree topology (k=4)

To organize the comparison between simulation and the mathematical model to Fat-tree topologies we define the results as follow:

- Predicted FAT-RT: It is the result for response time (RT) obtained by the mathematical model to Fat-tree topologies.
- Average FAT-RT: It is the result for average response time (RT) obtained by the simulation to Fat-tree topologies.

Table 4.5 and Table 4.6 present the results to predicted FAT-RT and average FAT-RT for the amount of monitoring data for 120 bytes and 150 bytes with frequency sampling based

Table 4.5: Predicted FAT-RT and Average FAT-RT for 120 bytes of monitoring data with interval sampling of 1 second.

Topology/Pods	Predicted FAT-RT (ms)	Average FAT-RT (ms)	Difference(%)
Topology 3 / K=4	34.92	32.57	- 6.72%
Topology 3 / K=6	74.76	70.17	- 6.13%
Topology 3 / K=8	125.80	118.72	- 5.63%
Topology 3 / K=10	199.45	189.60	- 4.94%
Topology 3 / K=12	284.30	272.27	- 4.23%

Table 4.6: Predicted FAT-RT and Average FAT-RT for 150 bytes of monitoring data with interval sampling of 1 second.

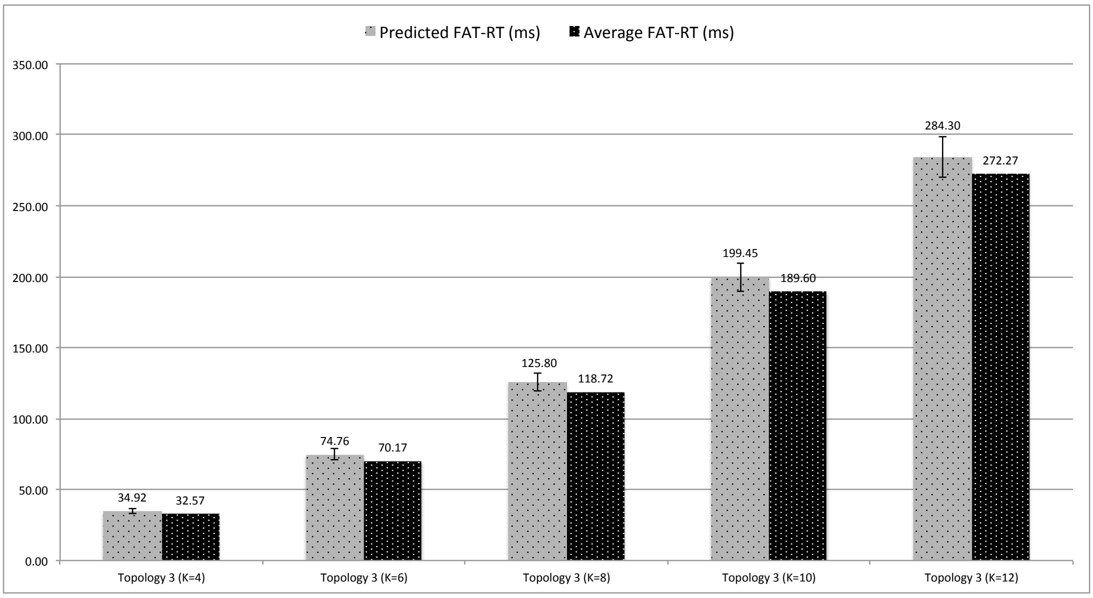
Topology/Pods	Predicted FAT-RT (ms)	Average FAT-RT (ms)	Difference (%)
Topology 3 / K=4	43.65	40.23	- 7.84%
Topology 3 / K=6	93.46	86.91	- 7.01%
Topology 3 / K=8	157.25	147.50	- 6.20%
Topology 3 / K=10	249.31	235.07	- 5.71%
Topology 3 / K=12	355.37	336.39	- 5.34%

on an interval of 1 second. Otherwise to conventional data center networks experiments, the experiments based on Fat-tree topologies does not exceed the time of 1 second to response time, so there is no need to extend the experiments to this topology to 10 seconds as we have done for topologies 1 to 4.

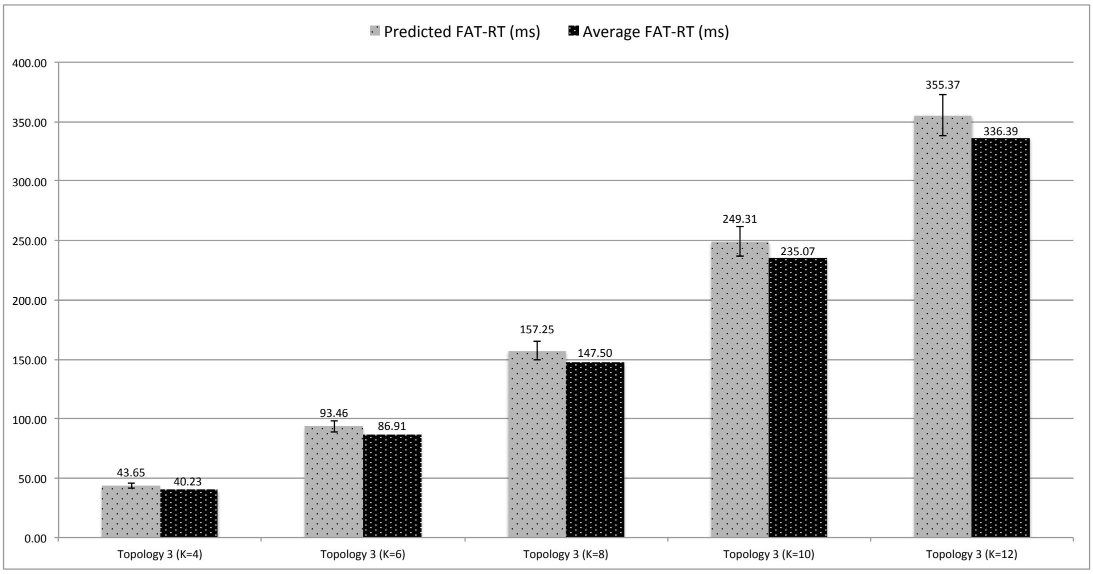
Table 4.5 compare results between predicted FAT-RT and average FAT-RT for 120 bytes of monitoring data with interval sampling of 1 second. In Table 4.5, predicted FAT-RT shows results that are useful to estimate the mutual influence between timeliness and scalability when considering - 6.72% as a maximum margin of error. Such margin of error is the result to the topology that has 4 pods (K= 4), which is the topology with fewer pods. On the other side, predicted FAT-RT has a minimum margin of error when considering 12 pods (K= 12) which is the topology with more pods.

Moreover, we highlight two significant issues from the Table 4.5. First, the margin of error was reduced, when more pods were added. In other words, when the monitoring topology escalates the margin of error decreases, which is significant to clouds that are environments usually composed of a plethora of probes and managers. Second, the margin of error is negative to Fat-Tree topologies which mean that the mathematical model can be useful to support the development of SLA's based on timeliness and scalability. It happens because the estimative provided by the mathematical model (Predicted FAT-RT) is always a bigger value when comparing to Average FAT-RT; thereby it is useful to avoid SLA breaches since it provides a big margin of tolerance.

Results demonstrate that the behaviour of the mathematical model is compatible with



(a)



(b)

Figure 4.7: Comparison between predicted and average response time to 120 and 150 bytes of monitoring data with interval sampling to 1 second to Fat-Tree Topologies. (a) 120 bytes . (b) 150 bytes.

the simulation as observed in Figure 4.7 (a) which bars to predict FAT-RT considers 6% as a margin of error. Such results demonstrate that the mathematical model is useful to 120 bytes of monitoring data when considering 8 pods or more because of the margin of error assumed. On the other hand, monitoring topologies with fewer pods present values that extrapolate the margin of error. However, the absolute values are closed owing to the small size of the structure; thereby the results are useful as a reliable reference.

Table 4.6 compare results between predicted FAT-RT and average FAT-RT for 150 bytes of monitoring data with interval sampling of 1 second. Table 4.6 shows that the results to predicted FAT-RT are useful to estimate the mutual influence between timeliness and scalability when considering - 7.84% as a maximum margin of error. This margin of error is the result to the topology that has 4 pods (K= 4), which is the topology with fewer pods. On the other side, predicted FAT-RT has a minimum margin of error when considering 12 pods (K= 12) which is the topology with more pods.

Moreover, we highlight two significant issues from the Table 4.6. First, the margin of error was reduced, when more pods were added. In other words, when the monitoring topology escalates the margin of error decreases. Second, the margin of error is negative to Fat-Tree topologies. The behaviour is practically the same if comparing to experiments performed to 120 bytes (Table 4.5). In this sense, we highlight that the amount of monitoring data accentuates the difference as observed when comparing Table 4.5 and Table 4.6. For example, in Topology 3 with 12 pods (K= 12), the difference is - 4.23% and - 5.34% respectively to 120 bytes and 150 bytes of monitoring data.

Results demonstrate that the behaviour of the mathematical model is compatible with the simulation as observed in Figure 4.7 (b) which bars to predict FAT-RT considers 6% as a margin of error. Such results demonstrate that the mathematical model is useful to 150 bytes of monitoring data when considering 10 pods (K= 10) or more because of the margin of error assumed. On the other hand, monitoring topologies with fewer pods present values that extrapolate the margin of error. However, the absolute values are closed owing to the small size of the structure; thereby the results are useful as a reliable reference.

4.4 Applying The Mathematical Model

Cloud monitoring systems must be designed and developed to fit with the characteristics of a cloud environment. In this way, particular characteristics of clouds have to be taken into consideration when building specialized monitoring systems. Cloud monitoring systems have introduced its requirements to accomplish its characteristics (See Section 2.2.3). Nevertheless, the current cloud monitoring systems have lack yet to perform all the requirements together, providing only partial solutions for monitor clouds. Thus, to develop a complete cloud monitoring system, such requirements must be all accomplished (MONTES et al., 2013). For this reason, the mutual relation among requirements must be unveiled, and it has to be taken into consideration when designing and developing comprehensive monitoring systems to clouds. Timeliness and scalability are two significant cloud monitoring requirements and the mutual influence between both are direct because at the same time that a cloud monitoring system grows to monitor all resources in the cloud, the number of probes and monitoring data also increase, impairing timeliness.

The mathematical model predicts the influence between timeliness and scalability unveiling the monitoring parameters that are involved in the mutual relation between such requirements. This model is useful as a support for the development of future cloud monitoring systems because it provides means to adjust cloud monitoring systems to cope with timeliness and scalability at the same time in a balanced way.

Although the mathematical model just takes into consideration timeliness and scalability. It is an innovative solution and the first step to understanding and solve the whole problem. According to literature (ACETO et al., 2013) (PARK et al., 2011), it is significant as a starting point to analysis the mutual influence among requirements, because timeliness and scalability have issues that must be taken into consideration to cope the mutual influence among others requirements (See Chapter 3.2). Thus, the mathematical model to predict the mutual influence between timeliness and scalability is helpful to understand their relation as well as is a basis for future research in this field. Moreover, it is important as a support to issues such as enhancing SLA's and billing, saving energy, raising profits to cloud operators, and reducing costs to customers.

4.4.1 Enhancing SLAs and Billing

Clouds have relied on SLAs to regulate the commercial relation between cloud operators and customers (EMEAKAROHA et al., 2012) (BRANDIC, 2009). Cloud monitoring is essential to certify that the SLA accomplishment is fair to both cloud operators and customers. In this scenario, the acquaintance of the mutual influence between timeliness and scalability provided by the mathematical model is support for significant issues such as assisting cloud operators to satisfy SLAs based on response time (timeliness), assisting infrastructure providers to assess its structures, assisting cloud operators to fulfil SLA without being invasive, and avoid SLA breaches to reduce penalties.

As discussed previously (See Chapter 3.2), timeliness is closely related to the accomplishment of SLAs based on response time. Predicting the influence of scalability over timeliness is important to avoid the development of weak SLAs which are vulnerable to the effects of scalability over response time and, as a consequence, do not suitable to clouds. In this way, the results presented by the mathematical model are meaningful to assist cloud operators for developing fair and consistent SLAs, based on timeliness.

Also, the mathematical model is useful to cloud operators that aim to manage its structure based on the forecast to resources usage. Predictions about the mutual influence between timeliness and scalability can be used to define the deployment of resources in data center networks to reduce the response time by adjusting the network topologies. It is useful in a plenty of ways to cloud operators which include: to reduce the response time to top customers, to improve the cloud performance. Thus, the mathematical model provides means to cloud operators to offer competitive SLAs because it is useful to improve the service rendered without increasing operational costs.

One of the most important issues when accomplishing a cloud monitoring requirement is do not being invasive. It means that at the same time in which such a requirement is fulfilled another requirement must not be prejudiced at all. To solve this problem, coherent and consistent SLAs are key. Predicting the mutual influence between requirements in this scenario is useful as a support to build such SLAs. In the same way, coherent and consistent SLAs are the basis to avoid SLAs breaches and, as a consequence, reduce the financial penalties to cloud operators.

4.4.2 Saving Energy

As highlighted in the Section 4.4.1, the mathematical model is useful to cloud operators as a support to manage its structure based on the forecast to resources usage. In the same way, it plays a vital role in saving energy in cloud data centers because proper management is fundamental to use energy in an efficient way.

Migration is a meaningful cloud monitoring requirement to save energy (SAHU; PATERIYA; GUPTA, 2013), and it is directly affected by timeliness. Migration is significant because it change the location of computational resources according to the goals (*e.g.,* save energy) of a specific application or system. Migration depends on timeliness to work properly because the monitoring data is useful to migration only when it is timely. It allows for cloud operators to perform actions in time to either correct or adjust the deployment of resource to save energy.

In this scenario, predictions based on timeliness and scalability are key to provide support to define the placement of resources to save energy because such predictions can estimate what size and architecture of network topology that is more adequate to migrate resources based on the response time.

4.4.3 Raising Profits to Cloud Operators and Reducing Costs to Customers

Predictions on timeliness and scalability provide advantages to both cloud operators and customers through competitive SLAs and saving energy. Moreover, when providing such gains, the predictions have offered means to cloud operators to improve their profits, and to reduce costs to customers.

Consistent SLAs are meaningful to raise the profits to cloud operators as well as to reduce costs to customers. It happens because consistent SLAs are trustworthy and, as a result, they increase the number of customers. It usually reduces the purchase costs to customers because there are more of them sharing services in an infrastructure. Thus, consistent SLAs normally means to get good services with less investment providing good cost-benefit services.

On the other side, the annual energy cost of the cloud data centers counts for billions of dollars, and it makes up to 23% of the total amortized costs of the cloud. For this reason, the energy saving is a vital issue when raising profits to cloud operators (JIANG et al., 2013). Moreover, the unused resources do not only means energy waste, but it also results in increasing the purchase costs to customers (QIU; SHEN; CHEN, 2015).

4.5 Final Remarks

This chapter discussed the mutual influence between timeliness and scalability to justify each monitoring parameter that is used in the mathematical model. Also, it described and evaluated the mathematical model. Finally, this chapter argued about the applicability of the mathematical model.

This mathematical model is a step forward in cloud monitoring and, as a consequence, in cloud management because it provides means to cloud operators (*e.g.,* CPs, InPs) to enhance their services.

In this context, cloud operators may use the estimation provided by the mathematical model to raise its profits, improve its quality of service and use it to develop fair SLAs based on timeliness, for example.

Aiming to demonstrate the mathematical model effectiveness to conventional data center networks, a comparison was performed comparing results provided by the mathematical model (predicted RT) and the results obtained by a simulation (average RT) which was performed in Chapter 3. Such results revealed some important issues such as the solution is useful, and it presents accurate results to larger and deep topologies, and in smaller topologies the mathematical model provides meaningful outcomes.

The usefulness of the mathematical model is verified because it was evaluated considering different monitoring metrics such as monitoring topology, the amount of monitoring data and frequency sampling. The mathematical model handles these monitoring metrics, and it presents consistent and coherent results that are close or even are equals at sometimes to the results obtained by the simulation.

Results have shown that the mathematical model presents accurate outcomes to larger topologies when comparing to smaller ones. It is somehow desirable because clouds are recognized as massive environments usually composed of a plethora of resources. These results are also observed when comparing deep topologies with non-deep ones. In this case, the former topology have presented more significant outcomes, which is desirable like to larger topologies compared to smaller ones.

On the other hand, for small and non-deep topologies the mathematical model does not present disappointing results at all. For example, to topology 1 with 64 hosts (*i.e.,* the smallest topology evaluated) the model showed considerable results considering a little margin of error (*i.e.,* 1.29% to 120 bytes of monitoring data and 0.85% to 150 bytes of monitoring data).

In a second moment, a comparison was performed to monitoring topologies based on Fat-tree. It compares results provided by the mathematical model (predict FAT-RT) and the results obtained by a simulation (average FAT-RT). Even that the results have not had the precision demonstrated for conventional data center networks, they confirmed that the mathematical model based on monitoring metrics is useful to another kind of monitoring topology commonly used in clouds.

Results have confirmed that the mathematical model presents best outcomes to larger topologies when comparing to smaller ones. It is demonstrated because when the amount of Pods increased the margin of error decreased providing more accurate outcomes.

From the above, we demonstrated that the mathematical model is adequate to predict the mutual influence between timeliness and scalability with a considerable level of accuracy. It is also a step forward regarding the study of the influence of cloud monitoring requirements, and the results are helpful in future researches in this area.

5 CONCLUSION AND FUTURE WORK

The goals of this chapter are to present the conclusion and future work of this thesis. Firstly, answers to the fundamental questions are discussed according to the hypothesis defended. After that, the main contributions of this thesis are presented. Finally, indications on future work related to the results of this thesis are identified.

5.1 Answer for the Fundamental Questions

The purpose of the fundamental questions was to define the main issues to be evaluated to pave the way to reach the hypothesis and, as a consequence achieve the contributions of this thesis. The description of fundamental questions along with answers is below.

Fundamental question I: What are the requirements and abilities introduced by cloud monitoring systems? What are the differences between requirements and abilities?

Answer: Cloud monitoring requirements are properties that cloud monitoring systems need to meet to perform its functions properly (ACETO et al., 2013) (CLAYMAN et al., 2010). According to the literature, there are several cloud monitoring requirements such as scalability, elasticity, comprehensiveness, adaptability, accuracy, and timeliness. However, there are not a consensus among researchers about them. In other words, some researchers claim by some requirements that are just ignored by others. In addition, cloud monitoring systems usually accomplish one or some of them, and there is not any cloud monitoring system that meets all requirements.

To tackle this problem in Chapter 2 we claimed that some requirements are more significant than others, and we defined them as cloud monitoring requirements. Therefore, to us a requirement is a property that a cloud monitoring system must have to handle properly with resources in a cloud. On the other hand, to others requirements with less significance we proposed a new taxonomy, namely, cloud monitoring ability. To us, an ability is a property that a cloud monitoring system should have to cope suitably with resources in a cloud.

Fundamental question II: What are the monitoring parameters that must be considered to evaluate the influence of scalability over timeliness? What is the metric that must be used to assess such influence?

Answer: Timeliness and scalability are requirements that have direct relation. The acquaintance of the mutual influence between them is a step forward to critical issues in cloud monitoring such as assisting the design and development of comprehensive and integrated monitoring system to clouds, increase profits to cloud operators based on estimations to resource usage, and supporting cloud operators to satisfy SLAs based on timeliness.

In Chapter 3, we demonstrated that the mutual influence between timeliness and scalability is liable to be mathematically modelled based on monitoring parameters, namely, monitoring topology, the amount of monitoring data, and frequency sampling. Besides, network bandwidth must be considered. In this context, response time is the output metric to represent such influence because it is the proper metric to represent timeliness.

Fundamental question III: Could cloud monitoring systems have the capacity to predict the influence of scalability over timeliness and vice-versa based on a mathematical model? Is it a suitable solution?

Answer: The prediction of the mutual influence between timeliness and scalability is useful to cloud operators and customers. It is significant to cloud operators mainly because it provides support to cloud operators to enhance their quality of service. Also, it is important to customers in several ways. For example, it assists cloud operators to satisfy SLAs based on timeliness, in this way providing guarantees to customers. However, it has lacked in deep research to unveiling this mutual influence based on mathematical comprobation.

To solve this problem in Chapter 4, we proposed a mathematical model to predict the mutual influence between timeliness and scalability. It provides a generic solution to cloud monitoring systems that wants to tackle the problem of mutual influence between such requirements based on a mathematical estimation. Aiming to demonstrate the mathematical model effectiveness a comparison was performed comparing results provided by the mathematical model and the results obtained via simulation. Such results raised meaningful outcomes that demonstrate the usefulness of the proposed solution.

Answering the fundamental questions that helped on the investigation of the hypothesis of this thesis, the main contributions obtained in this work are highlighted in the next section.

5.2 Main Contributions Obtained

The main contributions of this thesis are divided into two axes, namely, basic and key. The basic axe was identified during the investigation based on the current literature. They are contributions related to conceptual issues provided by this thesis. On the other hand, The key axe was identified during the analysis provided by the mathematical model development. Thus, keys are specific contributions associated with the core of this thesis. Basic and Key contributions are listed below.

- **Basic contributions of this thesis are:**
  - It discussed the cloud monitoring structure and introduced the concept of cloud monitoring focus.
  - It examined the concept of cloud monitoring requirement and proposed to divide them into two groups defined as cloud monitoring requirements and cloud monitoring abilities.
  - It analysed challenges and trends in cloud monitoring pointing research gaps that include the mutual influence between cloud monitoring requirements which is core to the keys contributions.
- **Key contributions of this thesis are:**
  - It presented a discussion of timeliness and scalability that include: the methods currently used to cope with the mutual influence between them, and the relation between such requirements and monitoring parameters.
  - It identified the monitoring parameters that are essential in the relation between timeliness and scalability.
  - It proposed a mathematical model based on monitoring parameters to estimate the mutual influence between timeliness and scalability.

Aiming to identify open issues to further researches. In the next section, future works that could derive from the contributions of this thesis are presented.

5.3 Future Work

The investigation performed on this thesis leads to the identification of further opportunities for research. These opportunities are listed as follows.

- **The economic impact of the mathematical model:** This thesis introduced a mathematical model to estimate the mutual influence between timeliness and scalability. A full research about the economic impact of the mathematical model in clouds will be helpful to assist cloud operators that would like to use it to increase their profits.

- **Tackle the mutual influence between cloud monitoring requirements:** This thesis is a step forward in the mutual influence between cloud monitoring requirements. It is helpful to researchers that have investigated the mutual influence between other requirements.

- **The development of comprehensive cloud monitoring systems:** Cloud monitoring systems usually accomplished one or some cloud monitoring requirements, and there is not any cloud monitoring system that meets all requirements. This thesis provides results that will be useful to solve it.

The list of future work discussed previously highlights the significant opportunities for research that can be directly derived from the work presented in this thesis. It is important describe that these possibilities can be listed because of the know-how acquired during the Ph.D. program.

REFERENCES

Accelops. **Accelops**. 2014. Disponível em: <http://www.accelops.com/> . Acesso em: 10 dez. 2014.

ACETO, G. et al. Survey cloud monitoring: A survey. **Comput. Netw.**, Elsevier North-Holland, Inc., New York, NY, USA, v. 57, n. 9, p. 2093–2115, jun. 2013. Disponível em:< http://dx.doi.org/10.1016/j.comnet.2013.04.001>. Acesso em: 10 dez. 2014.

AL-FARES, M.; LOUKISSAS, A.; VAHDAT, A. A scalable, commodity data center network architecture. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 38, n. 4, p. 63–74, ago. 2008. Disponível em: <http://doi.acm.org/10.1145/1402946.1402967>. Acesso em: 16 jan. 2016.

Amazon. **Amazon CloudWatch**. 2015. Disponível em: <http://aws.amazon.com/en/cloudwatch/>. Acesso em: 11 jan. 2015.

Amazon. **Amazon Elastic Compute Cloud: EC2**. 2015. Disponível em: <http://aws.amazon.com/documentation/ec2/>. Acesso em: 11 aug. 2015.

ANDREOLINI, M.; COLAJANNI, M.; TOSI, S. A software architecture for the analysis of large sets of data streams in cloud infrastructures. In: **IEEE 11th INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, 11., CIT'11, 2011, Pafos, Cyprus. Proceedings IEEE:**. [S.l.: s.n.], 2011. p. 389–394.

ANDREOZZI, S. et al. Gridice: a monitoring service for grid systems. **Future Generation Computer Systems**, v. 21, n. 4, p. 559 – 571, 2005. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X04001669>. Acesso em: 13 jan 2014.

ARMBRUST, M. et al. A view of cloud computing. **Commun. ACM**, ACM, New York, NY, USA, v. 53, n. 4, p. 50–58, apr 2010.

AVERSA, R.; TASQUIER, L.; VENTICINQUE, S. Management of cloud infrastructures through agents. In: **INTERNATIONAL CONFERENCE ON EMERGING INTELLIGENT DATA AND WEB TECHNOLOGIES, 3., EIDWT'12, 2012, Bucharest, Romania, Proceedings:**. [S.l.: s.n.], 2012. p. 46–53.

BARHAM, P. et al. Xen and the art of virtualization. In: **ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 19., SOSP '03, 2003, Bolton Landing, US. Proceedings:**. [S.l.: s.n.], 2003. (SOSP '03). Disponível em: <http://doi.acm.org/10.1145/945445.945462>. Acesso em: 23 jun. 2015.

BARI, M. et al. Data center network virtualization: A survey. **Communications Surveys Tutorials, IEEE**, v. 15, n. 2, p. 909–928, Second 2013.

BELOGLAZOV, A.; ABAWAJY, J.; BUYYA, R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. **Future Generation Computer System**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 28, n. 5, p. 755–768, maio 2012. Disponível em: <http://dx.doi.org/10.1016/j.future.2011.04.017>. Acesso em: 26 nov. 2015.

BELOGLAZOV, A.; BUYYA, R. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. **Parallel and Distributed Systems, IEEE Transactions on**, v. 24, n. 7, p. 1366–1379, July 2013.

BELOGLAZOV, A. et al. **Deploying OpenStack on CentOS Using the KVM Hypervisor and GlusterFS Distributed File System**. [S.l.]. Technical report. Disponível em: <http://www.cloudbus.org/reports/OpenStack-CentOS-KVM-glusterfs-guide-Aug2012.pdf>. Acesso em: 24 nov. 2015.

BRANDIC, I. Towards self-manageable cloud services. In: **33rd ANNUAL IEEE INTERNATIONAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, 33., COMPSAC'09, 2009, Seattle, US. Proceedings:**. [S.l.: s.n.], 2009. (COMPSAC '09). Disponível em: <http://dx.doi.org/10.1109/COMPSAC.2009.126>. Acesso em: 03 jan. 2016.

BUYYA, R. PARMON: a portable and scalable monitoring system for clusters. **Software: Practice and Experience**, Wiley, v. 30, n. 7, p. 723–739, jun. 2000.

BUYYA, R.; CALHEIROS, R.; LI, X. Autonomic cloud computing: Open challenges and architectural elements. In: **THIRD INTERNATIONAL CONFERENCE ON EMERGING APPLICATIONS OF INFORMATION TECHNOLOGY, 3., EAIT'12, 2012, Kolkata, India. Proceedings:**. [S.l.: s.n.], 2012. p. 3–10.

BUYYA, R. et al. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 25, n. 6, p. 599–616, jun. 2009. Disponível em: <http://dx.doi.org/10.1016/j.future.2008.12.001>. Acesso em: 11 jan. 2015.

Cacti. **Cacti RRDTool**. 2014. Disponível em: <http://www.cacti.net>. Acesso em: dez. 2014.

CALERO, J. A.; AGUADO, J. G. Monpaas: An adaptive monitoring platformas a service for cloud computing infrastructures and services. **Services Computing, IEEE Transactions on**, v. 8, n. 1, p. 65–78, Jan 2015.

CALERO, J. M. A.; AGUADO, J. G. Comparative analysis of architectures for monitoring cloud computing infrastructures. **Future Generation Computer Systems**, v. 47, p. 16 – 30, 2015. Special Section: Advanced Architectures for the Future Generation of Software-Intensive Systems.

CARDOSA, M.; KORUPOLU, M. R.; SINGH, A. Shares and utilities based power consolidation in virtualized server environments. In: **11th IFIP/IEEE INTERNATIONAL CONFERENCE ON SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 11., IM'09, 2009, New York, US. Proceedings:**. [S.l.: s.n.], 2009. (IM'09). Disponível em: <http://dl.acm.org/citation.cfm?id=1688933.1688986>. Acesso em: 18 jun. 2014.

CARVALHO, M. Barbosa de et al. A cloud monitoring framework for self-configured monitoring slices based on multiple tools. In: **9th INTERNATIONAL CONFERENCE ON NETWORK AND SERVICE MANAGEMENT, 9., CNSM'13, 2013, Zurich, Switzerland. Proceedings:**. [S.l.: s.n.], 2013. p. 180–184.

CARVALHO, M. Barbosa de et al. Efficient configuration of monitoring slices for cloud platform administrators. In: **IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, 19., ISCC'14, 2014, Madeira, Portugal. Proceedings:**. [S.l.: s.n.], 2014. p. 1–7.

CHAPMAN, C. et al. Elastic service definition in computational clouds. In: **IEEE/IFIP SYMPOSIUM ON NETWORK OPERATIONS AND MANAGEMENT, 11., NOMS'10, 2010, Osaka, Japan. Proceedings:**. [S.l.: s.n.], 2010. p. 327–334.

CHAVES, S. D.; URIARTE, R.; WESTPHALL, C. Toward an architecture for monitoring private clouds. **Communications Magazine, IEEE**, v. 49, n. 12, p. 130 –137, december 2011.

CLAYMAN, S. et al. Monitoring service clouds in the future internet. In: TSELENTIS, G. et al. (Ed.). **Future Internet Assembly**. [S.l.]: IOS Press, 2010. p. 115–126. Disponível em: <http://dblp.uni-trier.de/db/conf/fia/fia2010.htmlClaymanGCTRVNR10>. Acesso em: 12 nov. 2015.

Cooperegg. **Copperegg**. 2014. Disponível em: <http://copperegg.com/products/> . Acesso em: 17 nov. 2014.

COPIL, G. et al. Sybl: An extensible language for controlling elasticity in cloud applications. In: **13th IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING, 13., CCGrid'13, 2013, Delft, Holland. Proceedings:**. [S.l.: s.n.], 2013. p. 112–119.

DALLY, W.; TOWLES, B. **Principles and Practices of Interconnection Networks**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

DOELITZSCHER, F. et al. An autonomous agent based incident detection system for cloud environments. In: **IEEE THIRD INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECNOLOGY AND SCIENCE, 3., CloudCom'11, 2011, Fuzhou, China. Proceedings:**. [S.l.: s.n.], 2011. p. 197–204.

EMEAKAROHA, V. C. et al. Casvid: Application level monitoring for sla violation detection in clouds. In: **IEEE 36th ANNUAL COMPUTER SOFTWARE AND APPLICATIONS CONFERENCE, 36., COMPSAC'12, 2012, Izmir, Turkey. Proceedings:**. Washington, DC, USA: IEEE Computer Society, 2012. (COMPSAC '12), p. 499–508. Disponível em: <http://dx.doi.org/10.1109/COMPSAC.2012.68>. Acesso em: 27 jun. 2014.

FAN, C.-T.; WANG, W.-J.; CHANG, Y.-S. Agent-based service migration framework in hybrid cloud. In: **IEEE 13th INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING AND COMMUNICATIONS, 13., HPCC'11, 2011, Banff, Canada. Proceedings:**. [S.l.: s.n.], 2011. p. 887–892.

FELLER, E.; RILLING, L.; MORIN, C. Snooze: A scalable and autonomic virtual machine management framework for private clouds. In: **12th IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING, 12., CCGRID'12, 2012, Ottawa, Canada. Proceedings:**. [S.l.: s.n.], 2012. Disponível em:<http://dx.doi.org/10.1109/CCGrid.2012.71>. Acesso em: 09 jun. 2013.

FERRETO, T. C.; ROSE, C. A. F. D.; ROSE, L. D. RVision: An open and high configurable tool for cluster monitoring. In: **2nd IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING, 2., CCGRID'02, 2002, Berlin, Germany. Proceedings:**. [S.l.: s.n.], 2002.

FERRETTI, S. et al. Qos 150;aware clouds. In: **IEEE 3rd INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, 3., CLOUD'10, 2010, Miami, US. Proceedings:**. [S.l.: s.n.], 2010. p. 321–328.

FIORANI, M. et al. Energy efficiency of an integrated intra-data-center and core network with edge caching. **Optical Communications and Networking, IEEE/OSA Journal of**, v. 6, n. 4, p. 421–432, April 2014.

FOSTER, I.; KESSELMAN, C. (Ed.). **The Grid: Blueprint for a New Computing Infrastructure**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

FOSTER, I. et al. Cloud computing and grid computing 360-degree compared. In: **IEEE GRID COMPUTING ENVIRONMENTS WORKSHOP, 1., GCE '08, 2008, Austin, US. Proceedings:**. [S.l.: s.n.], 2008. p. 1–10.

FU, W.; HUANG, Q. GridEye: A service-oriented grid monitoring system with improved forecasting algorithm. In: **5th INTERNATIONAL CONFERENCE ON GRID AND COOPERATIVE COMPUTING WORKSHOPS, 5., GCCW'06, 2006, Washington, US. Proceedings:**. [S.l.: s.n.], 2006.

Google. **Google App Engine**. 2014. Disponível em: <http://cloud.google.com/products>. Acesso em: 11 nov. 2014.

GREENBERG, A. et al. Vl2: A scalable and flexible data center network. In: **ACM CONFERENCE ON DATA COMMUNICATION, 14., SIGCOMM'09, 2009, Barcelona, Spain. Proceedings:**. New York, NY, USA: ACM, 2009. (SIGCOMM '09), p. 51–62. Disponível em: <http://doi.acm.org/10.1145/1592568.1592576>. Acesso em: 10 jan. 2013.

GUO, Z.; YANG, Y. Exploring server redundancy in nonblocking multicast data center networks. **Computers, IEEE Transactions on**, v. 64, n. 7, p. 1912–1926, July 2015.

GUO, Z.; YANG, Y. On nonblocking multicast fat-tree data center networks with server redundancy. **Computers, IEEE Transactions on**, v. 64, n. 4, p. 1058–1073, April 2015.

HADLEY, B. et al. Phantom of the cloud: Towards improved cloud availability and dependability. In: **IEEE 4th INTERNATIONAL CONFERENCE ON CLOUD NETWORKING, 4., CloudNet'15, 2015, Pisa, Italy. Proceedings:**. [S.l.: s.n.], 2015. p. 14–19.

HAN, H. et al. A restful approach to the management of cloud infrastructure. In: **IEEE INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, 1., CLOUD'09, 2009, Bangalore, India. Proceedings:**. [S.l.: s.n.], 2009. ((CLOUD '09)). Disponível em: <http://dx.doi.org/10.1109/CLOUD.2009.68>. Acesso em: 14 jun. 2012.

HASSELMEYER, P.; D'HEUREUSE, N. Towards holistic multi-tenant monitoring for virtual data centers. In: **IEEE/IFIP SYMPOSIUM ON NETWORK OPERATIONS AND MANAGEMENT, 11., NOMS'10, 2010, Osaka, Japan. Proceedings:**. [S.l.: s.n.], 2010.

Herokai Group. **Heroku**. 2014. Disponível em: <http://www.heroku.com>. Acesso em: 01 nov. 2014.

HUANG, W. et al. A case for high performance computing with virtual machines. In: **20th ANNUAL INTERNATIONAL CONFERENCE ON SUPERCOMPUTING, 20., ICS'06, 2006, Cairns, Australia. Proceedings:**. New York, NY, USA: ACM, 2006. (ICS '06), p. 125–134. Disponível em: <http://doi.acm.org/10.1145/1183401.1183421>. Acesso em: 11 jan. 2012.

JIANG, Y. et al. Cloud analytics for capacity planning and instant vm provisioning. **Network and Service Management, IEEE Transactions on**, v. 10, n. 3, p. 312–325, September 2013.

KULKARNI, P.; KHANAI, R. Addressing mobile cloud computing security issues: A survey. In: **INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND SIGNAL PROCESSING,1., ICCSP'15, 2015, Melmaruvathur, India. Proceedings:**. [S.l.: s.n.], 2015. p. 1463–1467.

KUTARE, M. et al. Monalytics: Online monitoring and analytics for managing large scale data centers. In: **7th INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING, 7., ICAC'10, 2010, Washington, US. Proceedings:**. New York, NY, USA: ACM, 2010. (ICAC '10), p. 141–150. Disponível em: <http://doi.acm.org/10.1145/1809049.1809073>. Acesso em: 12 jan. 2013.

KVM. **Kernel Based Virtual Machine**. 2014. Disponível em: <http://www.linux-kvm.org>. Acesso em: 20 nov. 2014.

LAKSHMANAN, G. T. et al. A business centric end-to-end monitoring approach for service composites. In: **IEEE INTERNATIONAL CONFERENCE ON SERVICES COMPUTING, 1., SCC'10, 2010, Miami, US. Proceedings:**. [S.l.: s.n.], 2010. p. 409–416.

LANTZ, B.; HELLER, B.; MCKEOWN, N. A network in a laptop: Rapid prototyping for software-defined networks. In: **9th ACM SIGCOMM WORKSHOP ON HOT TOPICS IN NETWORKS, 9., SIGCOMM'10, 2010, Monterey, US. Proceedings:**. New York, NY, USA: ACM, 2010. (Hotnets-IX), p. 19:1–19:6. Disponível em: <http://doi.acm.org/10.1145/1868447.1868466>. Acesso em: 24 jan. 2015.

LI, D. et al. Ficonn: Using backup port for server interconnection in data centers. In: **INFOCOM, IEEE**. [S.l.: s.n.], 2009. p. 2276–2285.

MARSTON, S. et al. Cloud computing - the business perspective. **Decis. Support Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 51, n. 1, p. 176–189, abr. 2011. Disponível em: <http://dx.doi.org/10.1016/j.dss.2010.12.006>. Acesso em: 04 jul. 2015.

MASSIE, M. L.; CHUN, B. N.; CULLER, D. E. The ganglia distributed monitoring system: Design, implementation and experience. **Parallel Computing**, v. 30, p. 817–840, 2003.

MCFADDIN, S. et al. Modeling and managing mobile commerce spaces using restful data services. In: **THE NINTH INTERNATIONAL CONFERENCE ON MOBILE DATA MANAGEMENT, 9., MDM'08, 2008, Kansas City, US. Proceedings:**. [S.l.: s.n.], 2008. (MDM '08). Disponível em: <http://dx.doi.org/10.1109/MDM.2008.38>. Acesso em: 16 jan. 2013.

MCGILVARY, G. et al. C2ms: Dynamic monitoring and management of cloud infrastructures. In: **IEEE 5th INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECHNOLOGY AND SCIENCE, 5., CloudCom'13, 2013, Bristol, UK. Proceedings:**. [S.l.: s.n.], 2013. v. 1, p. 290–297.

MELL, P. M.; GRANCE, T. **SP 800-145. The NIST Definition of Cloud Computing**. Gaithersburg, MD, United States, 2011.

MENG, S. et al. Reliable state monitoring in cloud datacenters. In: **IEEE 5th INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, 5., CLOUD'12, 2012, Honolulu, US. Proceedings:**. [S.l.: s.n.], 2012. p. 951–958.

MENG, S.; LIU, L. Enhanced monitoring-as-a-service for effective cloud management. **IEEE Trans. Comput.**, IEEE Computer Society, Washington, DC, USA, v. 62, n. 9, p. 1705–1720, 2013. Disponível em: <http://dx.doi.org/10.1109/TC.2012.165>. Acesso em 27 jun. 2015.

MIAN, R.; MARTIN, P.; VAZQUEZ-POLETTI, J. L. Provisioning data analytic workloads in a cloud. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 6, p. 1452–1458, ago. 2013. Disponível em: <http://dx.doi.org/10.1016/j.future.2012.01.008>. Acesso em: 07 nov. 2014.

Microsoft. **Microsoft Azure**. 2015. Disponível em: <http://azure.microsoft.com/en-us/>. Acesso em: 18 jul. 2015.

Monitis. **Monitis Monitoring**. 2014. Disponível em: <http://portal.monitis.com> . Acesso em: 11 nov. 2014.

MONTES, J. et al. Gmone: A complete approach to cloud monitoring. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 8, p. 2026–2040, out. 2013. Disponível em: <http://dx.doi.org/10.1016/j.future.2013.02.011>. Acesso em: 01 jul. 2014.

MRTG. **Multi Router Traffic Grapher - MRTG**. 2014. Disponível em: <http://www.mrtg.com/> . Acesso em: 04 jan. 2014.

Nagios. **Nagios**. 2014. Disponível em: <http://www.nagios.org/>. Acesso em: 11 jan. 2014.

NGUYEN, T. A. B. et al. Role-based templates for cloud monitoring. In: **IEEE/ACM 7th INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING, 7., UCC'14, 2014, London, UK. Proceedings:**. [S.l.: s.n.], 2014. p. 242–250.

NURMI, D. et al. The eucalyptus open-source cloud-computing system. In: **9th IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER COMPUTING AND THE GRID, 9., CCGRID'09, 2009, Shanghai, China. Proceedings:**. [S.l.: s.n.], 2009. ((CCGRID '09)). Disponível em: <http://dx.doi.org/10.1109/CCGRID.2009.93>. Acesso em: 21 jan 2012.

Openstack. **Openstack: The Open Source Cloud Operating System**. 2015. Disponível em: <https://www.openstack.org/software/>. Acesso em: 23 jul. 2015.

PARK, J. et al. Markov chain based monitoring service for fault tolerance in mobile cloud computing. In: **IEEE WORKSHOP OF INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS, 1., WAINA'11, 2011, Biopolis, Singapure. Proceedings:**. [S.l.: s.n.], 2011. p. 520–525.

POVEDANO-MOLINA, J. et al. Dargos: A highly adaptable and scalable monitoring architecture for multi-tenant clouds. **Future Gener. Comput. Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 8, p. 2041–2056, out. 2013. Disponível em: <http://dx.doi.org/10.1016/j.future.2013.04.022>. Acesso em: 26 jul. 2014.

QIU, C.; SHEN, H.; CHEN, L. Towards green cloud computing: Demand allocation and pricing policies for cloud service brokerage. In: **IEEE INTERNATIONAL CONFERENCE ON BIG DATA, 1., Big Data'15, 2015, Santa Clara, US. Proceedings:**. [S.l.: s.n.], 2015. p. 203–212.

Rackspace. **Rackspace: cloud monitoring**. 2014. Disponível em: <http://www.rackspace.com> . Acesso em: 27 jan. 2014.

RAK, M. et al. Cloud application monitoring: The mosaic approach. In: **IEEE THIRD INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECHNOLOGY AND SCIENCE, 3., CLOUDCOM'11, 2011, Athens, Greece. Proceedings:**. [S.l.: s.n.], 2011. ((CLOUDCOM '11)). Disponível em: <http://dx.doi.org/10.1109/CloudCom.2011.117>. Acesso em: 17 jul. 2013.

RAMACHANDRAN, M.; CHANG, V. Recommendations and best practices for cloud enterprise security. In: **IEEE 6th INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECHNOLOGY AND SCIENCE, 6., CloudCom'14, 2014, Singapure City, Singapure. Proceedings:**. [S.l.: s.n.], 2014. p. 983–988.

RODERO, I. et al. Energy-efficient thermal-aware autonomic management of virtualized hpc cloud infrastructure. **Journal of Grid Computing**, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 10, n. 3, p. 447–473, set. 2012. Disponível em: <http://dx.doi.org/10.1007/s10723-012-9219-2>. Acesso em: 24 jul. 2015.

RODRIGUES, G. D. C. et al. The interplay between timeliness and scalability in cloud monitoring systems. In: **IEEE INTERNATIONAL SYMPOSIUM ON COMPUTERS AND COMMUNICATION, 20., ISCC'15, 2015, Larnaca, Cyprus. Proceedings:**. [S.l.: s.n.], 2015. p. 776–781.

RODRIGUES, G. D. C. et al. Monitoring of cloud computing environments: Concepts, solutions, trends, and future directions. In: **31st ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 31., SAC'16, 2016, Pisa, Italy. Proceedings:**. New York, NY, USA: ACM, 2016. (SAC '16), p. 378–383. Disponível em: <http://doi.acm.org/10.1145/2851613.2851619>.

RODRIGUES, G. D. C. et al. Network and services monitoring: A survey in cloud computing environments. In: **11th INTERNATIONAL CONFERENCE ON NETWORKS, 11., ICN'12, 2012, Saint Gilles, Reunion Island. Proceedings:**. [S.l.: s.n.], 2012. (ICN '12), p. 7–13.

RODRIGUES, G. D. C. et al. An architecture to evaluate scalability, adaptability and accuracy in cloud monitoring systems. In: **THE INTERNATIONAL CONFERENCE ON INFORMATION NETWORKING, 28., ICOIN'14, 2014, Phuket, Thailand. Proceedings:**. [S.l.: s.n.], 2014. p. 46–51.

ROSSI, F. D. et al. Green software development for multi-core architectures. In: **IEEE INTERNATIONAL SYMPOSIUM ON COMPUTERS AND COMMUNICATION, 19., ISCC'14, 2014, Madeira, Portugal. Proceedings:**. [S.l.: s.n.], 2014. p. 1–6.

ROSSI F., X. M. m. Y.; ROSE, C. D. On the impact of energy-efficient strategies in hpc clusters. In: **23rd EUROMICRO INTERNATIONAL CONFERENCE ON PARALLEL, DISTRIBUTED AND NETWORK-BASED PROCESSING, 23., PDP'15, 2015, Turku, Finland. Proceedings:**. [S.l.: s.n.], 2015. p. 17–21. Disponível em: <http://dx.doi.org/10.1109/PDP.2015.122>. Acesso em: 20 jan. 2016.

ROY, A.; GANESAN, R.; SARKAR, S. Keep it moving: Proactive workload management for reducing sla violations in large scale saas clouds. In: **IEEE 24th INTERNATIONAL SYMPOSIUM ON SOFTWARE RELIABILITY ENGINEERING, 24., ISSRE'13, 2013, Pasadena, US. Proceedings:**. [S.l.: s.n.], 2013. p. 421–430.

SAHU, Y.; PATERIYA, R.; GUPTA, R. Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. In: **5th INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS, 5., CICN'13, 2013, Mathura, India. Proceedings:**. [S.l.: s.n.], 2013. p. 527–531.

SHAO, J. et al. A runtime model based monitoring approach for cloud. In: **INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, 3., CLOUD'10, 2010, Miami, US. Proceedings:**. [S.l.: s.n.], 2010. ((CLOUD '10)). Disponível em: <http://dx.doi.org/10.1109/CLOUD.2010.31>. Acesso em: 24 jan. 2015.

SOTOMAYOR, B. et al. Virtual infrastructure management in private and hybrid clouds. **IEEE Internet Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 13, n. 5, p. 14–22, set. 2009. Disponível em: <http://dx.doi.org/10.1109/MIC.2009.119>. Acesso em: 28 jan. 2012.

STANOEVSKA-SLABEVA, K.; WOZNIAK, T.; RISTOL, S. **Grid and Cloud Computing: A Business Perspective on Technology and Applications**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009.

SUN, Y. et al. A high performance network architecture for large-scale cloud media data centers. In: **GLOBAL COMMUNICATIONS CONFERENCE, 13., GLOBECOM'13, 2013 Atlanta, US. Proceedings:**. [S.l.: s.n.], 2013. p. 1760–1766.

TAN, Y.; VENKATESH, V.; GU, X. Resilient self-compressive monitoring for large-scale hosting infrastructures. **Parallel and Distributed Systems, IEEE Transactions on**, v. 24, n. 3, p. 576–586, March 2013.

TOUEIR, A.; BROISIN, J.; SIBILLA, M. A goal-oriented approach for adaptive sla monitoring: A cloud provider case study. In: **LATIN AMERICAN CONFERENCE ON CLOUD COMPUTING AND COMMUNICATIONS, 2., LatinCloud'13, 2013, Maceió, Brazil. Proceedings:**. [S.l.: s.n.], 2013. p. 53–58.

VAQUERO, L. M. et al. A break in the clouds: towards a cloud definition. **SIGCOMM Computer Communications Review**, ACM, New York, NY, USA, v. 39, n. 1, p. 50–55, dez. 2008. Disponível em: <http://doi.acm.org/10.1145/1496091.1496100>. Acesso em: 02 jan. 2012.

VIRATANAPANU, A. et al. On demand fine grain resource monitoring system for server consolidation. In: **Kaleidoscope: Beyond the Internet? - Innovations for Future Networks and Services, 2010 ITU-T**. [S.l.: s.n.], 2010. p. 1–8.

WANG, C. et al. A flexible architecture integrating monitoring and analytics for managing large-scale data centers. In: **INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING, 8., ICAC'11, 2011, Karlsruhe, Germany. Proceedings:**. New York, NY, USA: ACM, 2011. (ICAC '11), p. 141–150. Disponível em: <http://doi.acm.org/10.1145/1998582.1998605>. Acesso em: 04 jun. 2013.

WANG, X.; DU, Z.; CHEN, Y. An adaptive model-free resource and power management approach for multi-tier cloud environments. **Journal of System and Software**, Elsevier Science Inc., New York, NY, USA, v. 85, n. 5, p. 1135–1146, maio 2012. Disponível em: <http://dx.doi.org/10.1016/j.jss.2011.12.043>. Acesso em: 23 jul. 2014.

Zennoss. **Zennoss: cloud Monitoring**. 2014. Disponível em: <http://www.zenoss.com>. Acesso em: 18 jan. 2014.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, Springer London, v. 1, p. 7–18, 2010.

ZHAO, L.; SAKR, S.; LIU, A. A framework for consumer-centric sla management of cloud-hosted databases. **Services Computing, IEEE Transactions on**, v. 8, n. 4, p. 534–549, July 2015.

APPENDIX A - PUBLISHED PAPERS AND CURRENT SUBMISSIONS

BRIEF DESCRIPTION OF THE PAPER PUBLISHED AT ICN 2012

This paper is entitled “Network and Services Monitoring: A Survey in Cloud Computing Environments”. It was published in the ICN conference and was the first paper regarding to the topics of this thesis. The paper presented an initial analysis about cloud monitoring based on frameworks such as Amazon EC2, Eucalyptus and OpenNebula. The main focus of this paper was to analyses methods used by such frameworks to cope with monitoring in cloud environments.

- **Title:**
  Network and Services Monitoring: A Survey in Cloud Computing Environments
- **Conference:**
  The Eleventh International Conference on Networks (ICN 2012)
- **URL:**
  <http://www.iaria.org/conferences2012/ICN12/>
- **Qualis**: B1
- **Date:**
  February 29 - March 05, 2012
- **Venue:**
  Novotel Hotel, Saint Gilles, Reunion Island

BRIEF DESCRIPTION OF THE PAPER PUBLISHED AT ICOIN 2014

This paper is entitled “An Architecture to Evaluate Scalability, Adaptability and Accuracy in Cloud Monitoring Systems”. It introduces the discussion about the mutual influence among cloud monitoring requirements. This paper evaluates the mutual influence among cloud monitoring requirements such as scalability, adaptability and accuracy based on local filters. Results proved a mutual influence among these requirements. The results show that filtering methods can be used in order to benefit Scalability and Adaptability. However, filtering methods are depreciating to Accuracy.

- **Title:**
  An Architecture to Evaluate Scalability, Adaptability and Accuracy in Cloud Monitoring Systems

- **Conference:**
  28th International Conference on Information Networking (ICOIN 2014)

- **URL:**
  <http://2014.icoin.org/main/>

- **Qualis**: B1

- **Date:**
  February 10 - February 12, 2014

- **Venue:**
  Graceland Resort, Phuket, Thailand

BRIEF DESCRIPTION OF THE PAPER PUBLISHED AT ISCC 2015

This paper is entitled “The Interplay Between Timeliness and Scalability in Cloud Monitoring System”. It was published in the ISCC Conference. It presents the discussion about the mutual influence between timeliness and scalability. In order to deal with the interplay between such requirements, this paper presents an evaluation based on monitoring parameters, namely, monitoring topology, amount of monitoring data and frequency sampling along with response time as the output metric.

- **Title:**
  The Interplay Between Timeliness and Scalability in Cloud Monitoring Systems
- **Conference:**
  The Twentieth IEEE Symposium on Computers Communications (ISCC 2015)
- **URL:**
  <http://www.ieee-iscc.org>
- **Qualis**: A2
- **Date:**
  July 06 - July 09, 2015
- **Venue:**
  Golden Bay Hotel, Larnaca, Cyprus

BRIEF DESCRIPTION OF THE PAPER TO APPEAR AT SAC 2016

This paper is entitled “Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions”. It is going to be published in the SAC Symposium. It presents a discussion of cloud monitoring based on four axes: Concepts, Solutions, Trends, and Future Directions. Firstly, it examines the fundamental concepts currently associated with cloud monitoring. In a second moment, it analysis some cloud monitoring solutions. After that, in discuss trends along with future directions in this area.

- **Title:**
  Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions
- **Conference:**
  ACM Symposium on Applied Computing (SAC 2016)
- **URL:**
  <http://www.acm.org/conferences/sac/sac2016/>
- **Qualis**: A1
- **Date:**
  April 04 - April 08, 2016
- **Venue:**
  University of Pisa and Scuola Superiore Sant’ana, Pisa, Italy

BRIEF DESCRIPTION OF THE PAPER SUBMITTED TO COMPUTER NETWORKS

This paper is entitled “Unveiling The Interplay Between Timeliness and Scalability in Cloud Monitoring System”. It was submitted to the Computer Networks. It extends the discussion about the mutual influence between timeliness and scalability. This paper presents a mathematical model to estimate such mutual influence aiming to provide suitable means to cloud operators that want to raise their profits and enhance their quality of service based on timeliness.

- **Title:**
  Unveiling The Interplay Between Timeliness and Scalability in Cloud Monitoring Systems
- **Journal:**
  Computer Networks
- **URL:**
  <http://www.journals.elsevier.com/computer-networks/>
- **Qualis**: A1

LIST OF COLLABORATIONS

1. Márcio Barbosa de Carvalho; Rafael Pereira Esteves; GUILHERME DA CUNHA RODRIGUES; Lisandro Granville; Liane Tarouco. A Cloud Monitoring Framework for Self-Configured Monitoring Slices Based on Multiple Tools. In: 9th International Conference on Network and Service Management - CNSM, 2013, Zurich. 9th CNSM and Workshops - Short Papers, 2013. p. 180-184. **Qualis**: B4.

Status: Published.

2. Márcio Barbosa de Carvalho; Rafael Pereira Esteves; GUILHERME DA CUNHA RODRIGUES; Clarissa Cassales Marquezan; Lisandro Granville; Liane Tarouco. Efficient Configuration of Monitoring Slices for Cloud Platform Administrators. In: IEEE Symposium on Computers and Communications, 2014, Madeira, Portugal. Proccedings of IEEE Symposium on Computers and Communication, ISCC 2014, Madeira, Portugal, 2014. **Qualis**: A2.

Status: Published.

3. Vinícius Tavares Guimaraes; Glederson Lessa dos Santos; GUILHERME DA CUNHA RODRIGUES; Lisandro Granville; Liane Tarouco. A Collaborative Solution for SNMP Traces Visualization. In: The International Conference on Information Networking, 2014, Phuket, Thailand. ICOIN 2014, 2014. p. 458-463. **Qualis**: B1.

Status: Published.

4. Glederson Lessa dos Santos; Vinicius Tavares Guimaraes; GUILHERME DA CUNHA RODRIGUES; Lisandro Granville; Liane Tarouco. A DTLS-based Security Architecture for the Internet of Things. In: IEEE Symposium on Computers and Communications (ISCC 2015), 2015, Larnaca,Chipre. Proccedings of IEEE Symposium on Computers and Communication, 2015. p. 809-815. **Qualis**: A2.

Status: Published.

5. Vinícius Tavares Guimarães, Oscar Caicedo Rendon, Gléderson Lessa dos Santos, GUILHERME DA CUNHA RODRIGUES, Carla Maria Dal Sasso Freitas, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville. A Reuse-based Approach to Promote the Adoption of Visualizations for Network and Service Management Tasks. The 40th IEEE Computer Society International Conference on Computers, Software and Applications (COMPSAC 2016). **Qualis**: A2.

Status: Submitted.

6. Vinícius Tavares Guimarães, Oscar Caicedo Rendon, Gléderson Lessa dos Santos, GUILHERME DA CUNHA RODRIGUES, Carla Maria Dal Sasso Freitas, Liane Margarida Rockenbach Tarouco, Lisandro Zambenedetti Granville. Improving Productivity and Reducing Cost Through the Use of Visualizations for SDN Management. In: IEEE Symposium on Computers and Communications (ISCC 2016). **Qualis**: A2.

Status: Submitted.

APPENDIX B - RESUMO DA TESE EM PORTUGUÊS

Resumo

Computação em nuvem é uma solução inovadora para prover recursos computacionais sob demanda. Atualmente, grandes empresas estão seguindo esta solução no intuíto de aumentar os seus ganhos. Por exemplo, Amazon e Microsoft tem se utilizado de sistemas de nuvem rentáveis, como Amazon EC2 e o Windows Azure. Do ponto de vista do cliente, a computação em nuvem tem se apresentado como uma tecnologia que fornece recursos tais como redução de custos, acessibilidade e flexibilidade.

De acordo com o NIST, existem cinco características essenciais em uma nuvem computacional, a saber, o auto-serviço sob demanda, o acesso à rede de forma ampla, recursos disponíves, elasticidade e serviço de medição. Este trabalho é focado em serviços de medição. No entanto, todas as características essenciais são descritas no Capítulo 2 para fornecer uma ampla visão da computação em nuvem.

Para o NIST, os chamados serviços de medição definem que "um sistema de nuvem controla automaticamente o uso de recursos, aproveitando uma capacidade de medição em algum nível de abstração apropriado para o tipo de serviço prestado (por exemplo, processamento, largura de banda, usuário ativo). Assim, o uso de recursos pode ser monitorado, controlado e reportado, oferecendo transparência tanto para o provedor quanto para o cliente do serviço utilizado ".

O serviço de medição é uma das características essenciais que presta apoio aos operadores de nuvem para fornecer serviços em nuvem de alta qualidade, normalmente baseados em acordos de nível de serviço, para os clientes. Para executar o serviço de medição, uma nuvem, como qualquer outro sistema computacional, precisa ser devidamente gerida de acordo com as suas características (por exemplo, escalabilidade, elasticidade, disponibilidade).

Neste contexto, o monitoramento nuvem torna-se uma questão importante porque fornece aos operadores em nuvem (por exemplo, prestadores de serviços, fornecedores de infra-estrutura) meios para gerenciar (por exemplo, analisar, controlar) um ambiente de computação em nuvem que geralmente é composto por um enorme número de recursos computacionais normalmente virtualizados e espalhados.

Portanto, o monitoramento da nuvem serve como apoio a atividades de gestão. Apresenta informações de vários recursos (por exemplo, rede, processamento) e serviços (por exemplo, análises, notificações), permitindo que os operadores nuvem possam executar as atividades de controle, o que permite a nuvem oferecer um desempenho previsível para os clientes.

No entanto, sistemas de monitoramento de nuvem precisam atender a diversos requisitos e habilidades tais como escalabilidade, abrangência, capacidade de adaptação, precisão, elasticidade e pontualidade.

Entretanto, quando um sistema de monitoramento de nuvem tenta cumprir uma exigência específica, outro requisito é negativamente ou positivamente afetado. Assim, o desenvolvimento de sistemas de monitoramento de nuvem, atualmente, está focado em características específicas de uma nuvem, fornecendo apenas soluções parciais para o monitoramento. Além disso, a falta de conhecimento sobre a influência exata entre os requisitos de monitorização nuvem restringe a capacidade de integrar sistemas de monitoramento de nuvem. Logo, o equilíbrio entre as exigências de monitoramento de nuvem torna-se uma tarefa desafiadora e significativa para aprimorar os sistemas de monitoramento de nuvem.

Para resolver esse problema, em primeiro lugar, existe a necessidade de compreender a relação entre os diferentes requisitos de monitoramento de nuvem, ou seja, o quanto um requisito de monitoramento de nuvem tem influencia sobre os outros.

Escalabilidade e Pontualidade são requisitos importantes porque escalabilidade permite uma nuvem a capacidade de crescer para lidar com a necessidade de mais recursos, e a oportunidade está intimamente relacionada os lucros dos provedores de nuvem. Além disso, independentemente da interrelação entre a oportunidade e a escalabilidade, ambos têm uma influência direta sobre todos os outros requisitos / capacidades. Por exemplo, a escalabilidade e a oportunidade estão intimamente relacionados com requisitos / capacidades, tais como a elasticidade, precisão e capacidade de adaptação. Por esta razão, a interelação entre a escalabilidade e a oportunidade é considerada um ponto significativo para iniciar a avaliação da influência mútua entre os requisitos de monitoramento nuvem.

Essa tese tem como objetivo investigar a interrelação entre escalabilidade e oportunidade para propor um modelo matemático para prever a influência mútua entre ambas as exigências em sistemas de monitoramento de nuvem. O conhecimento sobre essa influência mútua pode ser usado para melhorar as nuvens em questões como acordos de nível de serviço e qualidade de serviço, aumentando os lucros para os provedores de nuvem como uma consequência imediata e direta.

Visando realizar tal investigação, essa tese define a seguinte hipotese: **"Como prover mecanismos para estimar a influência mútua entre escalabilidade e oportunidade em sistemas de monitoramento de nuvens computacionais visando definir acordos de nível de serviço precisos e desse modo aumentar os ganhos financeiros para os operadores de nuvens computacionais"**.

Tal hipotese norteia a pesquisa dessa tese e auxilia na definição das seguintes perguntas de pesquisa que foram respondidas por essa tese.

- **Quais são os requisitos e habilidades introduzidas pelos sistemas de monitoramento de nuvem? Quais são as diferenças entre os requisitos e habilidades?**

  Resposta:

  Requisitos de monitoração de nuvens computacionais são propriedades que sistemas de monitoramento de nuvem precisam ter para desempenhar as suas funções adequadamente. De acordo com a literatura, existem vários requisitos de monitoramento de nuvem, tais como escalabilidade, elasticidade, abrangência, capacidade de adaptação, precisão, resiliência e oportunidade.

  No entanto, não há um consenso entre os pesquisadores sobre tais requisitos. Em outras palavras, alguns pesquisadores clamam por alguns requisitos que são simplesmente ignorados por outros. Além disso, sistemas de monitoramento de nuvem geralmente realizam um ou alguns desses requisitos, e não há qualquer sistema de monitoramento em nuvem que atenda a todos os requisitos.

  Visando contribuir no sentido de resolver esse problema no Capítulo 2, afirmamos e defendemos que alguns requisitos são mais significativos do que outros baseado em características essenciais de nuvens computacionais. Assim, procuramos caracterizar e definir com mais claridade quais são os requisitos de monitoramento de nuvens computacionais.

  Portanto, para nós um requisito é uma propriedade que um sistema de monitoramento de nuvem deve ter, obrigatoriamente, para lidar adequadamente com os recursos em uma nuvem.

  Por outro lado, para outros requisitos com menos importância propusemos uma nova taxonomia, definida como: habilidade de monitoramento de nuvens computacionais. Defendemos que uma habilidade é uma propriedade que um sistema de monitoramento de nuvem pode ter no sentido de aprimorar as suas habilidades para lidar com os recursos em uma nuvem. Entretento, essa capacidade não é obrigatória e imprescindivel.

- **Quais são os parâmetros de monitoramento que devem ser considerados para avaliar a influência da escalabilidade sobre a oportunidade? Qual é a métrica de que deve ser usada para avaliar essa influência?**

  Resposta:

  Oportunidade e escalabilidade são requisitos que têm relação direta. O conhecimento da influência mútua entre ambos os requisitos é um passo significante no intuíto de resolver questões críticas no monitoramento de nuvens computacionais, pois auxilia diretamente no projeto e desenvolvimento de sistemas de monitoramento abrangentes e integrados para nuvens, aumentando os lucros para os operadores de nuvem com base em estimativas do uso de recursos e apoiando operadores de nuvem no sentido de desenvolver acordos de nivel de serviço baseados no requisito de oportunidade.

No Capítulo 3, demonstramos que a influência mútua entre a oportunidade e escalabilidade é susceptível de ser matematicamente modelada com base em parâmetros de monitoramento, a saber, a topologia de monitoramento, a quantidade de dados de monitoramento, e frequência de amostragem. Além disso, demosntramos que a largura de banda de rede deve ser considerada. Neste contexto, o tempo de resposta é a métrica de saída para representar tal influência, porque é a métrica adequada para representar o requisito de oportunidade.

- **Poderiam sistemas de monitoramento de nuvens computacionais ter a capacidade de prever a influência mútua entre a escalabilidade e a oportunidade e vice-versa com base em um modelo matemático? Essa seria uma solução adequada?**

Resposta:

A previsão da influência mútua entre a oportunidade e escalabilidade é útil para operadores de nuvem e clientes. É significativo para operadores de nuvens computacionais principalmente porque tal influência oferece suporte a operadores de nuvem para melhorar a qualidade de serviço prestada. Além disso, é importante para os clientes de várias maneiras. Por exemplo, auxilia operadores de nuvem para desenvolver acordos de nível de serviço com base na oportunidade, desta forma fornecendo garantias aos clientes.

No entanto, ainda não existe uma investigação profunda para desvendar essa influência mútua com base em comprovação matemática.

Para resolver este problema no Capítulo 4, propusemos um modelo matemático para prever a influência mútua entre a oportunidade e a escalabilidade. O modelo fornece uma solução genérica para a sistemas de monitoramento de nuvens computacionais que pretendem resolver o problema da influência mútua entre esses requisitos com base em uma estimativa matemática.

Com o objetivo de demonstrar a eficácia do modelo matemático foi realizada uma comparação entre os resultados fornecidos pelo modelo matemático e os resultados obtidos através de simulação. Estes resultados levantaram resultados significativos, que demonstram a utilidade da solução proposta.

Respondendo às tais perguntas fundamentais de pesquisa nossa tese encontra em tais respostas as contribuições que foram possibilitadas por essa tese. As principais contribuições obtidos neste trabalho são destacadas a seguir.

As principais contribuições dessa tese são divididas em dois eixos, a saber, básico e fundamental. O eixo básico foi identificado durante a investigação com base na literatura atual. O eixo basico possui contribuições relacionadas com as questões conceituais fornecidas por esta tese. Complementarmente, o eixo chave apresenta contribuições que foram identificadas durante a análise fornecida pelo desenvolvimento modelo matemático. Assim, o eixo chave é composto por contribuições específicas associadas com o núcleo dessa tese. Tais eixos são descritos separadamente como segue:

Contribuições do eixo basico

- Essa tese discute a estrutura de monitoramento de nuvens computacionais e introduziu o conceito de foco de monitoramento.
- Essa tese examina o conceito de requisito de monitoramento de nuvens computacionais e propôe a divisão em dois grupos definidos como requisitos de monitoramento nuvem e capacidades de monitoramento em nuvem.
- Essa tese analisa os desafios e tendências de pesquisa sobre o monitoramento de nuvens computacionais apontando apontando as lacunas de investigação que a literatura possui atualmente. Tais lacunas incluem a influência mútua entre os requisitos de monitoramento de nuvens computacionais, o que é fundamental na definação do problema de pesquisa a ser investigado, bem como na fundamentação de literatura necessária para atingir as contribuições chave.

Contribuições do eixo chave

- Essa tese apresenta uma discussão sobre a oportunidade e a escalabilidade que incluem: os métodos utilizados atualmente para lidar com a influência mútua entre ambos os requisitos, e a relação entre esses requisitos e os parâmetros de monitoramento que os atingem diretamente.
- Essa tese identifica os parâmetros de monitoramento que são essenciais para a relação entre a oportunidade e a escalabilidade.
- Essa tese propoe um modelo matemático baseado nos parâmetros de monitoramento para estimar a influência mútua entre a oportunidade e escalabilidade.

Finalmente, baseado em todos os resultados atingidos por essa tese é realizada a identificação dos provaveis trabalhos futuros que podem derivar dessa tese. Elencamos e discutimos tais trabalhos futuros a seguir.

- O impacto econômico do modelo matemático: Essa tese introduziu um modelo matemático para estimar a influência mútua entre a oportunidade e a escalabilidade. A pesquisa completa sobre o impacto econômico do modelo matemático em nuvens computacionais seria de muita utilidade para ajudar os operadores de nuvem que gostariam de aplicar o modelo matemático para aumentar seus lucros.

- Combater a influência mútua entre os requisitos de monitoramento em nuvens computacionais: Essa tese é um passo em frente no estudo a respeito da influência mútua entre os requisitos de monitoramento em nuvens computacionais. O modelo proposto, portanto, é útil para pesquisadores que visam investigar a influência mútua entre outros requisitos de monitoramento.

- O desenvolvimento de sistemas de monitoramento de nuvem computacionais mais abrangentes: Sistemas de monitoramente de Nuvens computacionais geralmente cumprem apenas um ou alguns requisitos de monitoramento, e não há qualquer sistema de monitoramento em nuvem que atenda a todos os requisitos. Essa tese apresenta resultados que serão úteis para resolver esse problema, porque pel primeira vez considera os impactos do cumprimento de um requisito de monitoramento sobre os outros.