A dissertation submitted to the
Department of Computer Science & Software Engineering,
International Islamic University, Islamabad
as a partial fulfillment of the requirements
for the award of the degree of
Doctor of Philosophy in Computer Science

Accession No TH-26360

PhD
004.36
AFO

Cloud computing

Infrastructure as a service.

Revenue management

Customer satisfaction

Resource allocation (Computer science)

Plagiarism Undertaking

I take full responsibility of the research work conducted during the PhD Thesis titled "Optimizing IaaS Provider Revenue Through Customer Satisfaction And Efficient Resources Provision". I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

I understand that the management of International Islamic University Islamabad has a zero-tolerance policy towards plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my own original work and I have positively cited the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my PhD degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

Afzal Badshah _____

Date _____

Department of Computer Science & Software Engineering

International Islamic University Islamabad

Date: *December 8, 2021*

Final Approval

It is certified that we have examined the thesis report submitted by *Mr. Afzal Badshah*, Registration No. *120-FBAS/PhDCS/F15*, and it is our judgment that this thesis is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the Doctor of Philosophy in Computer Science.

Committee:

External Examiners

Dr. Hassan Mehmood, Professor
Department of Electronics
Quid-e-Azam University Islamabad

Dr. Moneeb Gauhar, Associate Professor
Department of Computer Science
Bahria University, Islamabad

Internal Examiner

Dr. Imran Khan, Assistant Professor
Department of Computer Science & Software Engineering
International Islamic University Islamabad

Supervisor

Dr. Anwar Ghani, Lecturer
Department of Computer Science & Software Engineering
International Islamic University Islamabad

Declaration

I hereby declare that this thesis, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that no portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Afzal Badshah

Dedication

Dedicated to my family who supported me a lot during this journey. They provided every support which I have needed.

Afzal Badshah

Acknowledgments

This thesis would not have been possible without the inspiration and support of several wonderful individuals — my thanks and appreciation to all of them for being part of this journey and making this thesis possible. Foremost, I would like to express my sincere gratitude to my supervisor, *Dr. Anwar Ghani*, for giving me guidance and counsel and having faith and confidence in me. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. He read and correct my early attempts at writings.

Besides my supervisor, I would like to thank the rest of my thesis committee: Dr. Hassan Mehmood, Dr. Moneeb Gauhar, and Dr. Imran Khan, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I sincerely acknowledge the contribution of different collaborators: Dr. Shahabodin Shamshirband, Ton Duc Thang University, Ho Chi Minh City, Viet Nam, Dr. Anthony Theodore Chronopoulos, University of Texas, San Antonio, TX 78249, USA, Dr. Giuseppe Aceto, University of Napoli Federico II, Italy, Dr. Antonio Pescape, University of Napoli Federico II, Italy and Dr. Ali Daud, College of Computer Science and Engineering, University of Jeddah. I am extremely grateful to these stalwart personalities who guided me.

This achievement would not have been possible without the pure love and support of my family. They back me in every hurdle I faced during this travel. I am forever indebted to them for giving me the opportunities and experiences that have made me who I am.

List of Publications

Publications from thesis

1. **Afzal Badshah**, Anwar Ghani, Shahaboddin Shamshirband and Anthony Theodore Chronopoulos "Optimizing IaaS Provider Revenue through Customer Satisfaction and Efficient Resource Provisioning in Cloud Computing" IET-Communication, 2019, Volume 13, Issue 18, p. 2913 – 2922, 2019. *(Impact factor 1.77)*

2. **Afzal Badshah**, Anwar Ghani, Azeem Irshad, Husnain Naqvi and Saru Kumari "Smart Resources Allocation on External Cloud Service Providers (CSPs) to Minimize Delay, Running Time and Transfer Cost", International Journal of Computer Systems (IJCS), Wiley, 2021, Volume 34, Issue 03, (Impact factor 1.34)

3. **Afzal Badshah**, Anwar Ghani, Giuseppe Aceto, Antonio Pescap'e, and ShahabShamshirband "Performance based Service Level Agreement in cloud computing to maximize providers' revenue" IET-Communication, 2019, Volume 14, Issue 07, p. 1102–1112, 2020. (Impact factor 1.77)

4. **Afzal Badshah**, Ateeqa Jalal, "SLAMaaS: Service Leverl Agreement monitoring through Monitoring as a Service" Big Data, [In production] (Impact factor 2.6)

5. **Afzal Badshah**, Anwar Ghani and Ali Daud "Comprehensive survey on revenue maximization: Research challenges and approaches" Research challenges and approaches " Transactions on Emerging Telecommunications Technologies, Elsevier,[Accepted for publication], (Impact factor 1.6) (2161-3915)

Others

1. **Afzal Badshah**, Ateeqa Jalal and Ghani Ur Rehman "Academic use of Social Media in Learners' Engagement in Underdeveloped Countries' Schools" Education and Information Technologies, volume 26, pages 6319–6336 (2021),[http://dx.doi.org/10.1007/s10639-021-10619-8] (Impact factor 2.04).

2. **Afzal Badshah**, Anwar Ghani, Ahsan Qureshi and Shahaboddin Shamshirband "Smart Security Framework for Educational Institutions using the Internet of Things (IoT)" Computers' Materials & Continua (CMC), Vol. 61, No. 1 ,pp. 81-101, 2019. *(Impact factor 3.24)*

3. **Afzal Badshah** and Anwar Ghani "Review of Internet of Things (IoT) in Education: Challenges, Opportunities and Possible Use of Smart Devices" ACM Computing Surveys. [Accepted for publication] *(Impact factor 7.990)*

4. **Afzal Badshah** and Anwar "Towards Smart Institutions by using the Internet of Things (IoT), Fog and Cloud Computing" IEEE transaction on Education. [Under review] *(Impact factor 2.27)*

5. **Afzal Badshah**, Ateeqa Jalal, Tauseef Ur Rehaman, "Performance-based Service Level Agreement in Cloud Computing." Research Journal of Science and IT Management, vol. 4, no. 4, pp. 20– 31, 2015. *(Impact factor 3.5)*

6. **Afzal Badshah**, Ateeqa Jalal, Tauseef Ur Rehaman, "SLA based Infrastructure resource allocation in Cloud Computing to increase IaaS revenue" Research Journal of Science and IT Management, vol. 4, no. 3, pp. 37-44, 2015. *(Impact factor 3.5)*

7. Anwar Ghani, **AfzalBadshah**, Saeedullah Jan, and Ali Daud "Issue and challenges in Cloud Storage Architecture: A Survey",Researchpedia Journal of Computing (RpJC) , Volume 1, Issue 1, Article 6, Pages 50–64, Jun 2020

8. Muhammad Asif, Muhammad Abid, **Afzal Badshah** "Stackelberg Game for Heterogeneous Traffics Management in Next-Generation Cellular Network", IET Communication, volume 15, issue 13, doi: https://doi.org/10.1049/cmu2.12185, 2021 .

9. **Afzal Badshah**, Ateeqa Jalal "Use of Regional Computing for Social Big Data to Minimize the Big Data Effects", Big data, [Under review]

10. **Afzal Badshah**, Ateeqa Jalal "Collaboration of Mobile, Edge and Cloud Computing to Minimize the Social Big Data Effects" Big data, [Under review]

Abstract

With *limited resources*, it is quite challenging to meet dynamic and massive cloud customers' demands. Overutilization or refusing any Service Level Agreement (SLA) drives to penalties, which play an uncertain role in the cloud business. Furthermore, *cost, performance*, and *penalties* are the key factors to revenue generation and customer satisfaction. However, they have a complex correlation which becomes more complicated in the absence of a suitable framework that defines these factors clearly. *Service Level Agreement (SLA)* is an initial document which negotiate these parameters before business initialization. Due to the massive workload and Internet activities, a lot of automation is needed. Therefore, it is essential to have scalable resources , clear-cut SLA, and efficient resource provision policies to avoid disastrous consequences.

Various studies have been conducted in this regard, however, improvements are still needed. To address the resource scalability and SLA violation issues, the proposed model uses scalable resources with suitable prices from external providers. Despite a federated cloud, providers are not compelled to hire from a specific alliance. In case of overload, the provider can engage external resources wherever they are available conveniently and economically. For customers' satisfaction, the proposed model uses joint prices model. Algorithms are used to optimize different metrics to maximize providers' revenue. This study investigated the cost, performance, and penalties issues and proposed *Performance-based Service Level Agreement (PerSLA)* framework to optimize these values for revenue optimization. PerSLA optimizes these parameters and maximizes both providers' revenue and customers' satisfaction. For workload migration to external resources, this study proposed smart resources allocation to minimize the transfer delay and cost.

CloudSim simulation is used to evaluate the functioning of the proposed framework. Experimental results show that other systems starts SLA violation as the workload increases by 500 cloudlets. However, the proposed structure effectively manages huge workloads of up to 1200 cloudlets without generating single violation. By offering joint prices on customer choice and outsourcing the overloaded workload to external resources, maximize revenue generation. The results show that this framework generates revenue from different pricing strategies. With the current workload, it earns $ 1494 from reserved customers, $ 2694 from on-demand customers, and $ 528 from negotiated customers. Hiring external resources earns external revenue as well as maximizes internal revenue. The results show that this earn $2100.

The above discussion validates that the proposed framework is adequate in revenue generation and customers satisfaction. Customers and providers monitor the business concerns to agreed terms and conditions. This framework is efficient in handling massive workloads, revenue generation, and customers' satisfaction. On violation, the provider is penalized. This agreement increases the trustworthy relationship between provider and consumer.

Contents

List of Figures

List of Tables

Acronyms

CC Cloud Computing
SLA Service Level Agreement
IaaS Infrastructure as a Service
PaaS Platform as a Service
SaaS Software as a Service
PerSLA Performance based Service Level Agreement
AI Artificial Intelligence
NADRA National Data Base and Registration Authority
VPN Virtual Private Network
QoS Quality of Service
CAGR Compound Annual Growth Rate
CLV Customer Lifetime Value
VM Virtual Machine
CLs Cloudlet
CSP Cloud Service Provider
RT Response Time
WT Waiting Time
SV SLA Violation
SS Service Scalability
ET Execution Time
UT User Tasks
ASF Assurance Satisfaction Factor
RSF Response Satisfaction Factor
DC Data Centers
MIPS Memory Instructions Per Second
MBPS Mega Bit Per Second
PM Physical Machines
MB Megabyte
GB Gigabyte
ZB Zeta byte
ESRM Efficient Scheduling for Revenue Maximization
PORM Price Optimization for Revenue Maximization
OSRM Optimizing Scheduling for Revenue Maximization
MDRM Migration Decision for Revenue Maximization
RAM Random Access Memory

Sim No Simulation Number

IoT Internet of Things

CloudSim Cloud Simulator

AWS Amazon Web Services

ICT Information and Communication Technology

IT Information Technology

Chapter 1

Introduction

The recent advances in smart technology (e.g, Internet of Things (IoT), Artificial Intelligence (AI) and 5G), generate a *massive data* traffic. The 51 billion devices forecast is a big number; even seven times greater than the world population [1]. These devices will increase the annual size of the global data-sphere up to 175 ZB up to 2025 [2, 3]. Another report states (as shown in Figure 1.6) that more than 331 billion dollars will be invested in the cloud up to 2023 [2]. Similarly, as shown in figure 1.7 the internet users are expected to increase up to 5.5 billion up to 2023 [4]. It needs special techniques and infrastructure to process the incoming big data [5]. Furthermore, integrating AI in smart devices makes the network more complicated. With this rapid development in smart technology, cloud computing is getting more and more attention and attraction [6, 7].

The above background shows that today markets are swiftly shifting towards the cloud. It is becoming crucial for today's business to migrate to the cloud. Rather than buying infrastructures, operators, licenses, and software; customers easily hire cloud services on more affordable charges. Cloud computing is moving desktop services to doorsteps via internet. Virtualization, parallel processing and distributed processing approaches are used to provide services on the network. It is highly successful paradigm for utility computing. Cloud technology is the need of time due to increasing number of devices and data [8, 9] to provide the desktop services on the network.

Though cloud was coined in 1996, however, it has been in discussion since 1950 when terminals were connected to the mainframe computers. Hence resources sharing idea was present at that time too. In 1960, John Mclarthy gave the idea that computing resources can be delivered at door step such as other utility services. In 1970, the concept of virtualization developed. In 1990, telecommunication companies started the Virtual Private Network(VPN). Before that, dedicated lines were used for each customer. In 1999, "salesforce.com" was the first cloud services provider. Now-a-days, Amazon is the leading cloud service

provider. Google started cloud services in 2009 and got a leading place in the cloud competition. IBM and Oracle started their cloud services in 2011 and 2012 respectively [10, 11].

Cloud computing classifies desktop services into three primary categories: *Infrastructure as a Service (IaaS)*, *Platform as a Service (PaaS)* and *Software as a Service (SaaS)* [12]. These services are provided in three different models: *private cloud*, *public cloud* and *hybrid cloud* [13], as shown in Figure 1.1. *Infrastructure as a Service (IaaS)* provides physical resources online (e.g., computing, storage, and networking). It provides servers, network connections, storage, and other related resources. Amazon Web Services (AWS) is most popular IaaS service provider [14]. Apart from AWS, Micro Soft Azure, Google Cloud Platform, Ali Baba Cloud and IBM Cloud are the leading IaaS providers in the market [15–18]. *SaaS* provides online applications (e.g., monitoring, finance, and communication) to consumer, running on provider infrastructure. Oracle is one of the popular SaaS service provider [10]. Apart from Oracle, SAP , Cobweb , MuleSoft and SalesForce are the leading SaaS providers in the cloud market [11, 19–21]. *PaaS* provides online development tools (e.g, testing, analysis, and deployment services) for software development. Users and customers design software using programming languages, libraries, and other tools. Oracle Cloud Platform is one of the popular PaaS service provider. Apart from Oracle, AWS, Google Cloud Platform, Microsoft Azure, and SalesForce are the leading PaaS providers in the market [9, 22].

Similarly to cloud services categorization "as a services", they are provided in three different models, as shown in Figure 1.2. These models are *private cloud*, *public cloud* and *hybrid cloud*. ***Private cloud*** resources are visible only for the organization's users. It is expensive and only large organizations utilize this, have high security concerns. National Data Base and Registration Authority Pakistan (NADRA-Pakistan) [23] is daily life use of a private cloud computing in Pakistan. ***Public cloud*** computing resources can be hired by anyone from public after getting registration and signing SLA. These services may be provided by the business, academic or government organizations. Google, Microsoft and Amazon are the example of a public cloud provider, popularly used in pubic. ***Hybrid cloud*** computing is a combination of both private cloud computing and public cloud computing. This requires special technology to enable the portability of data and applications [13]. Universities, providing their services to public are examples of the hybrid cloud.

As discussed, *IaaS* provides physical resources (e.g., computing, memory, storage, and networking) online as services, as shown in figure 1.3 . The traditional way of using physical resources has several limitations. First of all, computer infrastructure cost is high; secondly, there are many issues in configuration, management, and maintenance [24]. Therefore, small and medium level organizations cannot invest capital on IaaS at the initial stages of business. They simply hire IaaS services to start their business. IaaS clouds are growing rapidly than other cloud services. The Compound Annual Growth Rate (CAGR) is 20.4 percent over the 2015-2020 forecast period.

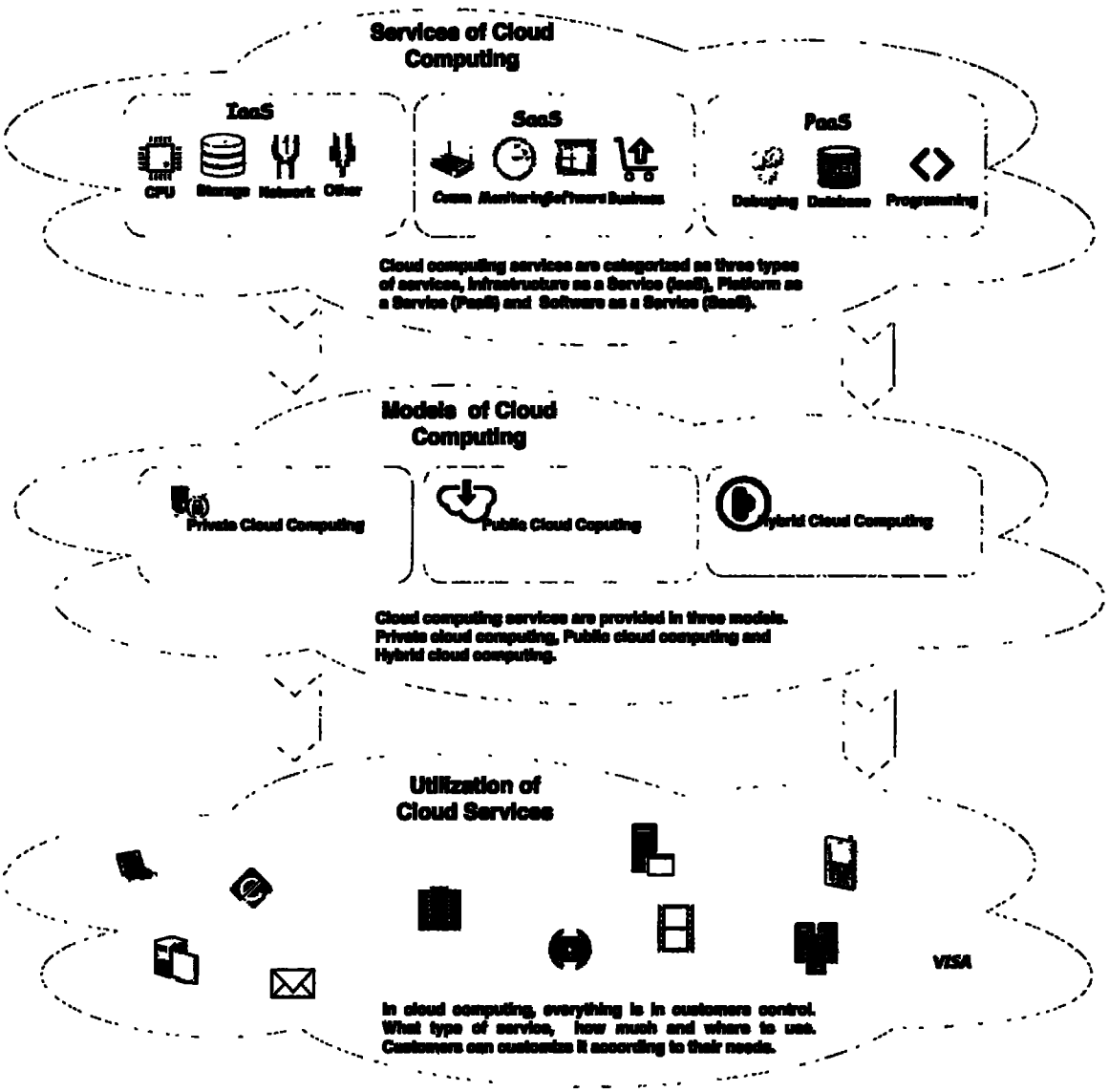


Figure 1.1: Structure of cloud computing

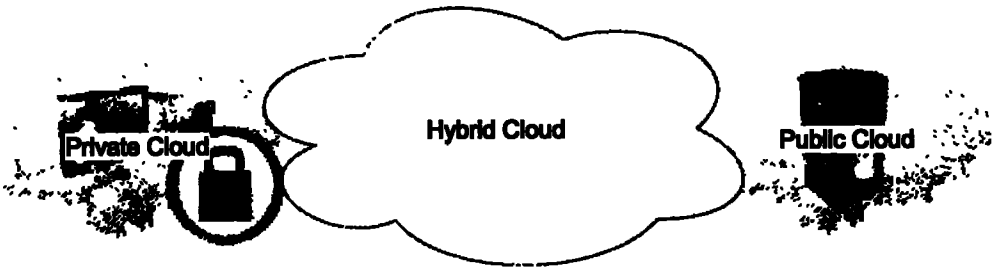


Figure 1.2: Models of cloud computing

Generally, *IaaS utilization* is the primary method by which cloud business success is determined. In basic terms, it is a measure of the actual revenue earned by assets against the potential revenue they could have earned. In IaaS clouds, virtualization, parallel and distributed processing techniques are used to improve the utilization [25]. In virtualization, single hardware is shared with many users. In parallel processing, many applications are run in parallel simultaneously. In distributed processing, heavy workloads are processed on different servers [26]. Utilization plays a decisive role; in case of high utilization, the revenue increases otherwise resources remain underutilized and cannot be claimed in future and are wasted [27, 28].

To efficiently allocate the resources, different IaaS *scheduling* strategies are used to handle the customers' workload. These policies greatly affect the resource's utilization, customers' satisfaction and providers' revenue. Widely used policies are static and dynamic scheduling. In static scheduling, a prior request is sent for resources, while in dynamic scheduling, resources are allocated according to customer needs [29]. In case of inefficient resources allocation, it creates under or over-utilization issues.

To address the under and over-utilization issues, federated cloud concept was introduced. *Federated cloud* is the union of different cloud service providers. They hire and share their services for higher resources utilization and customers' satisfaction. In a federated cloud, providers provide internal resources to the customers or they rent external resources from other providers to satisfy the customers' needs [30]. Federated cloud concept was introduced to handle the resources scalability and load balancing issues. It helps the limited resources providers to extend their business. It efficiently utilizes the providers' resources both in under and overutilization. Limitation of the federated cloud is that the providers are compelled to hire from the union. They have to follow the particular rules and restrictions of the federation [31].

Virtual services crates untruest situation in customers, therefore, an Service Level Agreement (SLA) is agreed between the provider and customers. *SLA* is an agreement to build trust between the service provider and consumer. It enhances customers' satisfaction by achieving Quality of Service (QoS) and improves the relationship between stack holders. Penalties are imposed on defaulter [32, 33]. As discussed earlier, massive transit is shifting towards the cloud. Not only the organizations, public is also using the cloud

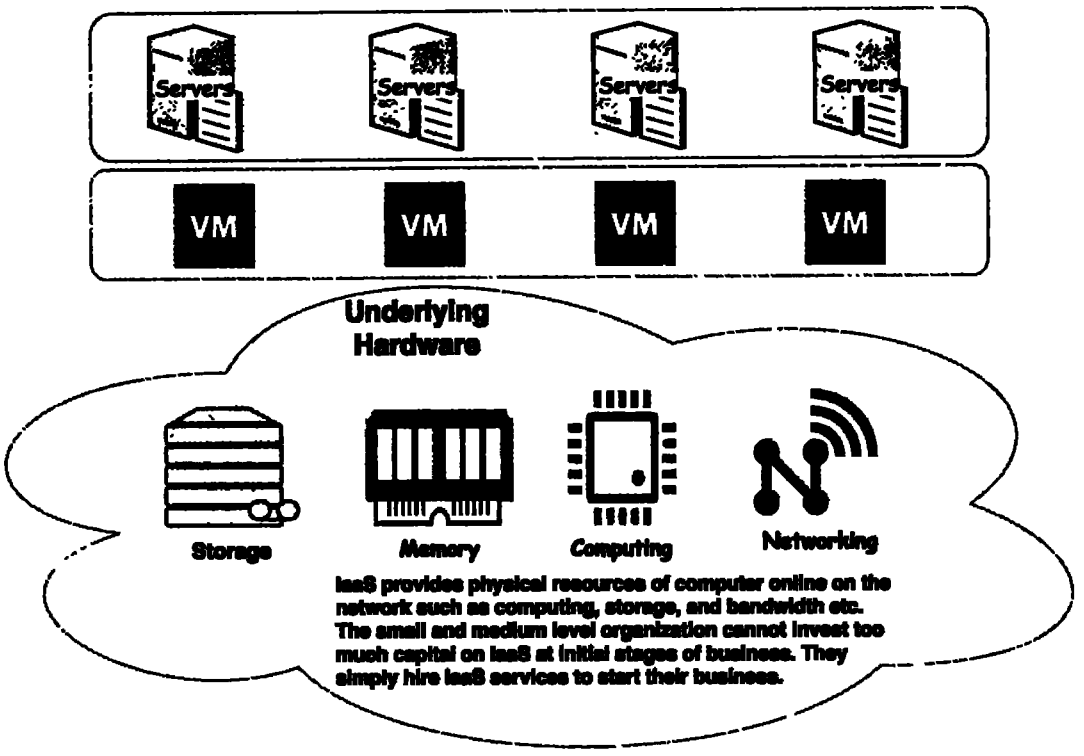


Figure 1.3: Infrastructure as a Service

in different ways. Therefore, It is extremely necessary for the cloud business to have clear-cut SLA for customers satisfaction and quality assurance [34, 35].

Likewise, the scheduling model, *price models* are very important part for customer attraction and satisfaction. Numbers of prices models are used in cloud computing. In tiered pricing, services are divided into different tires, having different prices. In unit pricing, customers are charged on a unit of space or bandwidth used. This pricing mechanism is more flexible than a tiered pricing mechanism. In subscription base pricing, customers are charged according to their subscription. Customers get a discount by early reservations. The downside of this pricing model are that the provider gives guaranteed services to customers, also underutilization wastes the resources. Usage-based pricing is also used by Amazon, charging according to customer's usage [36]. Introducing different pricing models in one business is complex however, it attracts more customers, which improves the resources utilization [37].

This thesis main focus is the providers' *revenue maximization*. We have investigated performance management, SLA and penalties management, resources scalability, customer satisfaction, resources management, cost management, and prices management. The proposed techniques handle four different scenarios i) efficiently handling the massive dynamic workloads with limited underlying resources ii) managing perfor-

mance in case of high utilization iii) customer satisfaction to attract more customers to increase the resources utilization and iv) efficient resources management and provision.

1.1 Motivation

Revenue is the main concern of any business. Maximizing the revenue and having perfect market place, is a goal of every business. IaaS market is growing up very quickly. *Total investment* in cloud computing, as shown in Figure 1.4, is 47 billion dollars in 2013, 96 billion dollars in 2015, 107 billion dollars in 2017, 176 billion dollars in 2018, 241 billion dollars in 2020 and expected 331 billion dollars in 2022 [2]. Similarly, as shown in figure 1.7, the internet users are expected to increase up to 5.5 billion up to 2023 [4]. IaaS is more demanding than others cloud services. Its Compound Annual Growth Rate (CAGR) is 20.4 percent in the 2015-2020 forecast periods. Further, as shown in Figure 1.5, more than 51 billion devices are expected to be connected to the Internet by the end of 2023. This is a big number, even seven times greater than the population of the whole world. These devices will increase the annual size of the global data-sphere up to 175 ZB, shown in Figure 1.6 [2, 3]. This attractive cloud background motivates the researchers to further investigate the field. There are number of interesting factors, as discussed below, which need to be investigated for providers' revenue maximization.

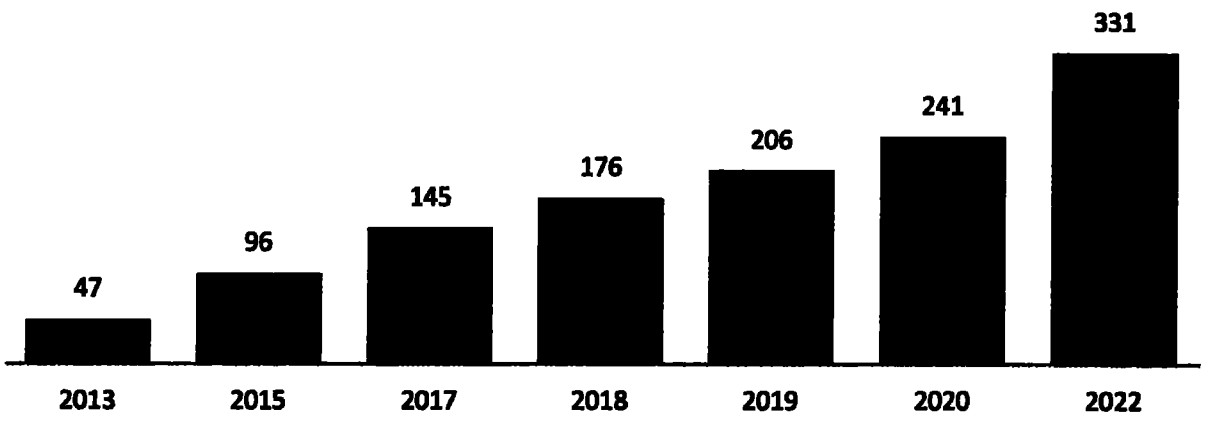


Figure 1.4: Investments forecast in cloud market (in million)

Service *performance* directly interferes the providers' revenue. Cloud resources performance covers the number of parameters. It includes running time, waiting time, availability, reliability and security etc. These parameters thresholds are agreed during the SLA negotiation. On violation of these thresholds, defaulter pays penalties. SLA violation not only increases cost but also creates dissatisfaction among customers. This needs a clear cut SLA to avoids performance degradation. The part of this dissertation addresses a

Performance-based Service Level Agreement (PerSLA) to efficiently manage the performance parameters of the services without violating them.

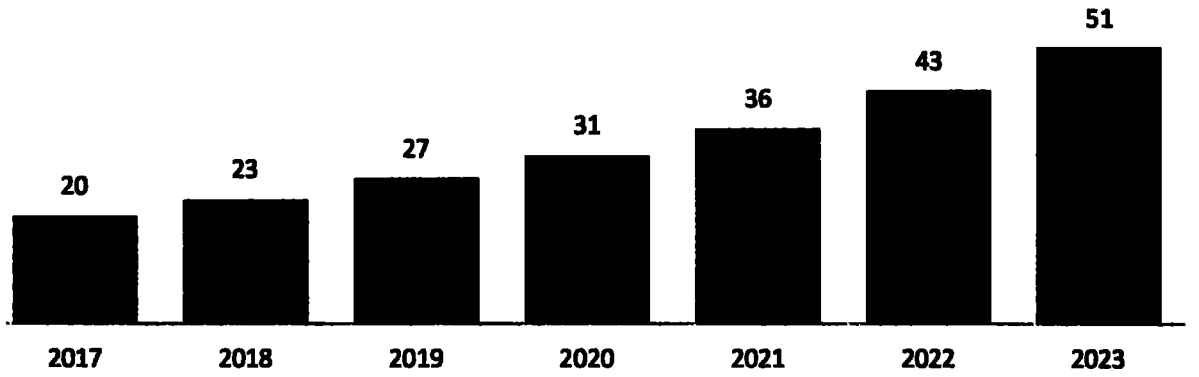


Figure 1.5: Devices forecast in cloud market (in billion)

Cloud resources are *not storable*, if not utilized on time, revenue is not generated. Also, efficient use of cloud resources is a critical issue. Where lower utilization minimizes the providers' revenue, overutilization also creates issues for cloud providers. In extreme utilization, providers reject some of the existing customers, having massive workloads. Refusing the customers' request creates dissatisfaction. Moreover, rejecting massive workload dispossess providers from higher revenue. These issues with the cloud resources motivate the providers to excellently utilize the resources within time and efficiently provide the resources according to customers' requirements. To handle these issues, part of this thesis discusses the efficient and smart resources scheduling in cloud computing.

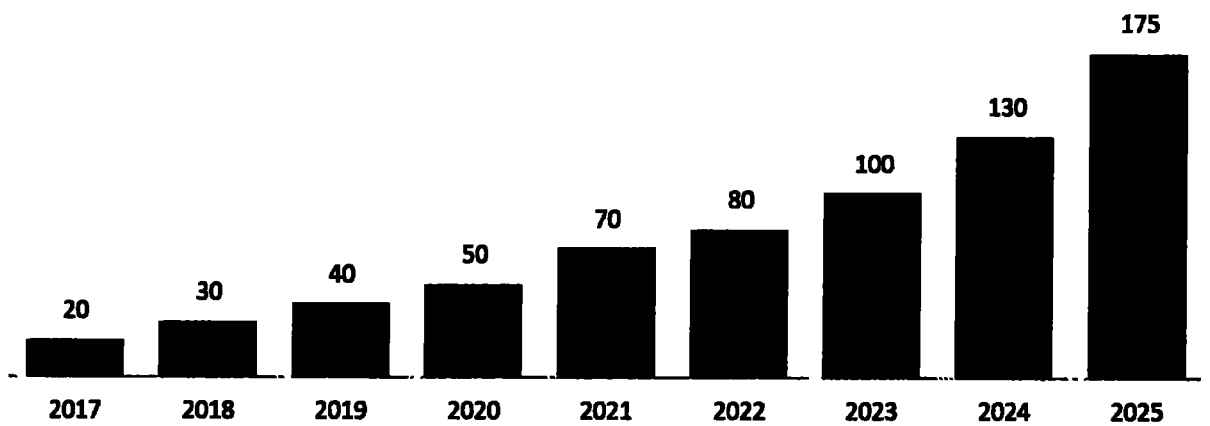


Figure 1.6: Data sphere forecast in cloud market (ZB)

Prices play an active role in customers' satisfaction and attraction. Where price is directly proportional to performance, it is also inversely proportional to customers' satisfaction. Market needs such like pricing model which address the need of lower prices and higher performance customers [38, 39].

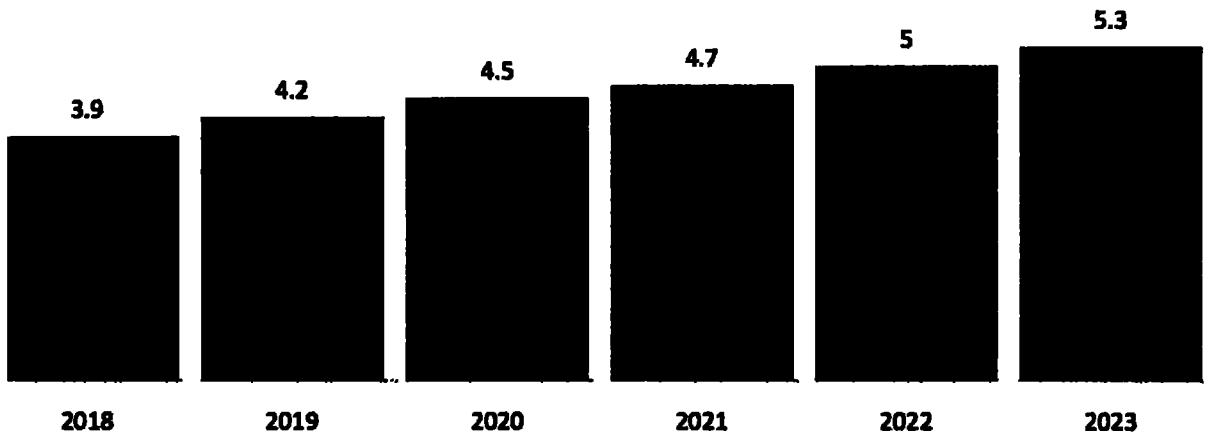


Figure 1.7: Forecast of users, connecting to the internet (in billion)

Customer satisfaction is a primary concern in business, which shows how much services are fulfilling the customers' needs. Customers are the measures of repurchases. It is more suitable economically to retain existing customers than to make new ones. Usually, companies spend millions of dollars on customer's attention, however, a small investment in customers' retention benefits more. Customers are the potential of any company. A global survey by Accenture Global Customer Satisfaction (AGCS) report (2008) shows that prices are not the most important concern, the most important is customer satisfaction service. Successful customer satisfaction services increase Customer Lifetime Value (CLV) which enhance the company's profit. According to McKinsey [40], 13 percent of unhappy customers tell about 9-15 people about their experiences. Customers' satisfaction is most important for revenue and repurchases. Customer's retention, repeat purchase and Likelihood to tell others about the services are the key customers' satisfaction matrices.

1.2 Scope of the Research

This thesis investigated the revenue maximization of the provider, having limited resources to handle heavy workloads (big data). Connecting to this, three different techniques have been proposed to maximize the providers' revenue. The scope of this study is given below. Table 1.1 also presents the scope of this thesis.

- As discussed earlier, this thesis cover cloud providers' revenue maximization and customers' satisfaction.

- Resources limitation and attached challenges are explored in this study.
- Resource scalability, performance, penalties, cost, prices, customer dis-satisfaction, and revenue are the main challenges which are directly affected by resource limitation. This study investigated all these complexities.

Table 1.1: Research scope

Characteristics	Research Scope
Parties	IaaS cloud providers and consumers
Objective	Providers' revenue maximization
Methodology	Resources scalability and customers' satisfaction
Workload	Cloudlet as custoemr's workload
Resources	VMs as providers data centers

1.3 Research Aims and Objectives

Cloud service providers, provide desktop services online. These services may be in terms of hardware or software. Cloud computing is the future of the coming world. In the next generation, it will be the part of every field of life, as today computer is an essential part of every field. Large transactions on the cloud will extend its market .

The main objectives of the proposed model are.

- *-How to optimize resource utilization for maximum revenue?*

  Cloud resources are not storable. Resources not utilized on time, are wasted. Therefore, under-utilization of cloud computing resources is a critical issue, which directly minimizes the providers' revenue. Furthermore, the customers hire some resources but later on, they do not use them, such like resources are also wasted. This topic has been investigated but needs further exploration.

- *-How to satisfy the heterogeneous customers' demands with limited resources?*

  Resource limitation is the barrier to revenue generation. The concept of federated cloud was introduced to deal with limited resources. However, Federated Cloud also has restrictions on hiring resources from specific providers registered with the Federated cloud. Even though, such resources

may be easily and moderately available from providers outside the federated cloud. This results in a kind of monopoly, where a hirer is compelled to hire from a specific seller.

- *-How to create a good SLA for customer satisfaction?*

  Cloud computing is not face to face business. Providers provide services online on the internet. There is no direct communication between the provider and consumer. In such a scenario, for the customer and provider, it is hard for both to trust each other. It is the need of the day the providers have to negotiate a clear-cut SLA to attract more customers and to increase their business.This satisfies both customers as well as providers.

- *- How to optimize prices and pricing policies to earn maximum revenue?*

  Prices play a vital role in any business. In the cloud, some customers need high performance and do not care about prices, while some care for prices but not for performance. Higher prices dissatisfy the customers, however, for optimum performance, higher prices are paid. It is challenging to satisfy diverse customers for prices and performance. The aim of this research study is to further investigate the price policies to satisfy the customers so that the revenue may be maximized.

- *-With limited IaaS resources and maximum utilization, how to minimize the SLAs violation to maximize revenue?*

  Penalties play a worse role in the cloud computing business. It also minimizes the customer retention. Penalties are usually caused when non-scalable resources are overutilized. Service providers accept the loaded SLAs for revenue maximization but later on, cannot run these workload. Such like decisions lead to SLA violations and penalties. The aim of this study is to handle the massive workload with limited resources.

- *- How to efficiently manage the resources provision for heavy loaded SLA to avoid SLA rejection?*

  SLA termination is a major concern in the cloud computing business. The main factors to the SLA termination are the resources non-scalability and over-utilization. Furthermore, most of the resources are wasted due to inefficient resource scheduling polices. Resources scheduling and resource management needs further investigation for better revenue and performance.

1.4 Research Contributions

This dissertation covers the challenges towards the providers revenue maximization. We proposed a framework to effectively optimize performance, penalties, cost, and revenue. The developed framework is incor-

porated in simulation and the system was tested in various topological and temporal conditions resulting a great deal. The detail contributions of this study are given below.

- To handle the heterogeneous customers demanded with limited resources, we intend to design an approach where a service provider, if overloaded or having customers more than the available resources, can serve or *hire an external resource* anywhere it is available conveniently and moderately. With external resources, the provider having limited resources may extend the business. Furthermore, it also facilitates providers to outsource underutilized resources.

- To overcome the resources utilization issue, the proposed approach tries to efficiently utilize the resources. The resources, which are underutilized, may be given to customers using *negotiation-based pricing*. The benefit of negotiation-based pricing is that it generates some revenue instead of wasting all underutilized resources. Furthermore, penalties are also not imposed on SLA violations. The proposed framework strives to minimize resources wastage by offering negotiation-based pricing.

- Customer satisfaction is primary concern of any business. Cloud business totally depends on the SLA signed by both parties. To maintain performance and customer satisfaction, the proposed approach uses *Performance based Service Level Agreement (PerSLA)* to provide good performance services to the customers. This model optimizes the performance, cost, and prices to satisfy both the provider and consumer.

- Price plays a vital role in customers' satisfaction. The proposed approach uses customers' satisfaction techniques to attract and retain them. *Customers are divided* according to their demands and joint prices are offered so every customer may be satisfied as per their needs.

- With limited IaaS resources and maximum utilization, it is challenging to maintain performance. The proposed framework tries to maximize the performance and resource scalability by *hiring external resources* and using performance-based Service Level Agreement. With optimum performance and scalable resources, SLA violation will be minimized.

- To minimize SLA rejection, in the proposed model, *SLA does not lead to termination directly*. With earlier violations, prices are decreased, which are the initial indicator to the provider to adjust the performance. The initial decrease in performance does not drop performance and customer satisfaction. Customers are also reimbursed for lower performance with respect to downtime. Therefore, the initial reduction in performance do not demotivate them.

1.5 Problem Statement

Towards the IaaS cloud provider's revenue maximization, *penalties* and *customer dissatisfaction* play a critical role. Cloud computing resources are perishable. Maximum revenue can be earned by maximum utilization. In the cloud, most of the provider's revenue waste in penalties payment. Cloud providers also lose their customers due to their dissatisfaction. Customer dissatisfaction and rejection means lots of wastage of revenue.

This thesis addresses research problems arise from the following question:

> ***With limited resources availability and heterogeneous customers' demands, how to maximize the providers' revenue and performance by minimizing SLA violation and customers' dissatisfaction in IaaS clouds?***

- **P-1:** With *limited resources*, it is quite challenging to meet dynamic and massive customer demands. Higher utilization or refusing any SLA (customer) drives to penalties, which play a hazardous role in the cloud business. In the overutilization phenomenon, instead of maximizing the revenue, a provider wastes most of the revenue in paying penalties. Providers having limited resources are not able to accept heavy loaded SLAs. All incoming requests must be less than or equal to the available capacity.

$$\sum_{k=0}^{n} CustomersWorkLoad \leq TotalResources \tag{1.1}$$

  This condition creates dissatisfaction among customers and providers heaving massive workload. SLA is violated or rejected due to limited resources.

$$Penalties \propto NumberofSLAViolation \tag{1.2}$$

  Where penalties increases as the SLA violation increases (equation 1.2)

- **P-2:** *Performance, penalties, cost,* and *revenue* are the key factors to revenue generation. They have a complex correlation and it gets more complicated in the absence of a proper framework to clearly define these factors. SLA is an initial document which negotiates these parameters before business initialization. Usually, SLA violation occurs due to the overutilization of limited resources. The provider is penalized for each SLA violation which wastes most of the revenue.

$$Per \propto Rev \tag{1.3}$$

$$Per \propto 1/\eta \tag{1.4}$$

$$Per \propto \chi \tag{1.5}$$

$$Rev \propto 1/\chi \tag{1.6}$$

The above comparison, equation no 1.3, 1.4, 1.5, and 1.6, explains that an increase in performance (*per*) maximize the revenue (*Rev*) and minimize the penalties (η), however, performance is also proportional to the cost (χ), which is inversely proportional to the revenue. These parameters have a complex correlation. The situation goes worse and complicated if there is no proper framework, which clearly defines them.

- **P-3:** ***Under and overutilization*** **is the major concern of IaaS providers. In expectation of high revenue, providers accept more SLAs which lead to overutilization. Some times, in case of lower quality of marketing, providers are not able to get proper customers to utilize all their resources. Furthermore, usually, in cloud computing resources are reserved for customers, which are not utilized and wasted.**

$$ResourcesUtilization \propto NumberOfSLAs \tag{1.7}$$

$$Revenue \propto ResourcesUtilization \tag{1.8}$$

Resources utilization (equation no 1.7) and revenue (equation no 1.8) are directly proportional to the number of customers, and the numbers of customers are directly proportional to resources utilization and revenue (equation no 1.9 and 1.10).

$$ResourcesUtilization \propto NumberOfCustomers \tag{1.9}$$

$$Revenue \propto ResourcesUtilization \tag{1.10}$$

- **P-4:** *Pricing* plays a vital role in customers' satisfaction, retention, and attraction. Different price models are used in cloud computing. Some customers need high performance and pay more, however, some customers cannot pay high and accept lower performance resources with suitable prices. In such like situation, there must be an attractive cost and pricing framework.
- **P-5:** The recent advances in Information Technology (IT) and infrastructure fueled a massive transition from in-house Information and Communication Technology (ICT) services to cloud computing. Furthermore, new data processing paradigms (e.g., Big Data) have opened new business models, creating new technological requirements and increasing the need for cloud services. Due to the large scale of data and internet business, extensive automation is required. Therefore it is essential to have *clear-cut SLA* to avoid disastrous consequences in the customer business.
- **P-6:** *Resources scheduling* on external Cloud Service Provider (CSPs) plays a very important role in cloud data centers. A good scheduling policy maximizes resource utilization and customer satisfaction. Bad scheduling severely affects the performance of the provider. This not only affects the performance, but it also increases the cost, energy consumption , and customer dissatisfaction. These are the main reasons that scheduling policies are the major concerns for providers [41].

Towards the IaaS cloud provider revenue and performance optimization, P-1, P-2, P-3 , P-4, P-5, and P-6 clearly show that under and over utilization, penalties, performance, resource scalability, cost, and prices management, proper SLA management, customer retention, and attention and customer dissatisfaction plays a critical role in revenue maximization.

1.6 Thesis Organization

The core chapters of this thesis are derived from research papers written during the PhD. This thesis is divided into three parts. First part discussed the hiring external resources, 2nd discussed the performance optimization, and the third part discussed the efficient resources scheduling on external resources for revenue maximization. The thesis structure is shown in figure 1.8.

Chapter 2 explains in detail the revenue maximization preliminaries. This explored the main parameters that increase the suppliers' revenues, directly or indirectly. This chapter is partially derived from the article

.

> ***Afzal Badshah**, Anwar Ghani and Ali Daud "Comprehensive survey on revenue maximization: Research challenges and approaches" Research challenges and approaches " Transactions on Emerging Telecommunications Technologies, Wiley, [Accepted for publication], (Impact factor 1.6)*

Chapter 3 reviews the literature and provides the background relevant for the context of the thesis. The associated techniques and literature are classified into seven different categories. This section helps to identify research gaps, challenges, and directions for revenue maximization. This chapter is derived from

> ***Afzal Badshah**, Anwar Ghani and Ali Daud "Comprehensive survey on revenue maximization: Research challenges and approaches" Research challenges and approaches " Transactions on Emerging Telecommunications Technologies, Wiley, [Accepted for publication], (Impact factor 1.6)*

Chapter 4 discussed the proposed methodology. CloudSim and Cloud Analytic are used to simulate the proposed techniques. The CloudSim was extended to evaluate the effectiveness of the proposed model. The experimental configuration is coded in Java to evaluate the operation of this model.

Chapter 5 investigates the revenue optimization through customer satisfaction and efficient resources utilization. In extreme utilization, the high workload is outsourced to external resources, which extends the provider business having limited resources. This chapter is derived from

> ***Afzal Badshah**, Anwar Ghani, Shahaboddin Shamshirband and Anthony Theodore Chronopoulos "Optimizing IaaS Provider Revenue through Customer Satisfaction and Efficient Resource Provisioning in Cloud Computing" The IET -Communication, 2019, Volume 13, Issue 18, p. 2913 – 2922, DOI: 10.1049/iet-com.2019.0554, 2019*

Chapter 6 discusses Performance based Service Level Agreement (PerSLA) for customer satisfaction and trusty business. PerSLA optimizes the SLA parameters to an optimum point. PerSLA specified the parameters, their thresholds, and fines. Algorithms monitor services and try to improve performance in the case of a failure. This chapter is derived from

> ***Afzal Badshah**, Anwar Ghani, Giuseppe Aceto, Antonio Pescap'e, and Shahab Shamshirband ""Performance based Service Level Agreement in cloud computing to maximize providers' revenue" IET-Communication. Volume 14, Issue 07, p. 1102 – 1112, 2020 (Impact factor 1.77)*

Chapter 7 discussed the delay and running time minimization to maximize the revenue. The issue with hiring external resources is that it increases the cost in terms of energy consumption. To handle this issue, such like external CSPs are selected which have the best running and delay time. This chapter is derived from

> ***Afzal Badshah**, Anwar Ghani, Azeem Irshad, Husnain Naqvi and Saru Kumari "Smart Resources Allocation on External Cloud Service Providers (CSPs) to Minimize Delay, Running Time and Transfer Cost", International Journal of Computer Science (IJCS) 2021, Volume 34, Issue 03, (Impact factor 1.34)*

Chapter 8 proposes a reliable framework for monitoring provider's services by adopting third party monitoring services with clearcut SLA and penalties management. Since, this framework monitors SLA as a cloud monitoring service, it is named as Service Level Agreement Monitoring as a Service (SLA-MaaS). This chapter is derived from

> ***Afzal Badshah**, Ateeqa Jalal, "SLAMaaS: Service Leverl Agreement monitoring through Monitoring as a Service" Big Data, [In production] (Impact factor 2.6)*

Chapter 9 is the comparative analysis of the proposed framework with others. This chapter is partially derived from

> ***Afzal Badshah**, Anwar Ghani, and Ali Daud "Comprehensive survey on revenue maximization: Research challenges and approaches" Research challenges and approaches " Transactions on Emerging Telecommunications Technologies, Wiley, [accepted for publication], (Impact factor 1.6)*

Chapter 10 concludes the thesis with a summary of the main findings and a discussion of future research directions.

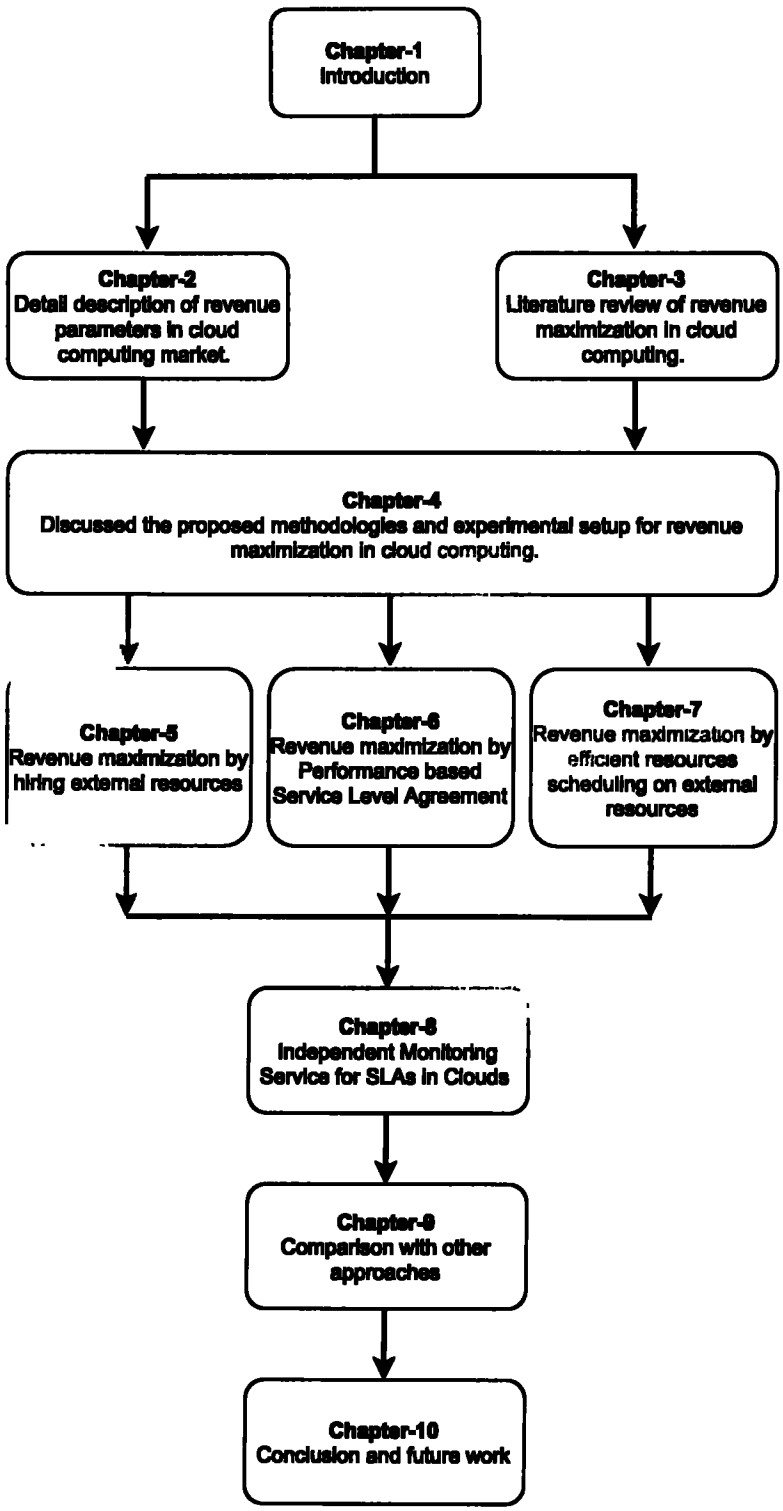


Figure 1.8: Thesis organization

Chapter 2

Revenue Maximization Preliminaries

Generating high revenue is the ultimate goal of every service provider. In any type of business, revenue is the main concern. In cloud computing, major parameters that affect revenue are, the performance of the services, SLA and penalties management, resources scalability, resources utilization and scheduling, customer satisfaction, cost and price management, and auction and advertisement [16]. This chapter covers the preliminaries of the thesis. Table 2.1 shows the terminologies used in this thesis.

In cloud computing, data is *stored in more than one places*. Therefore, if it is lost or some storage device crashes at one place, it can be recovered from other places as shown in Figure 2.1. Name node works as a server to keep the address of data stored on data nodes. Data is replicated on different data nodes for security and recovery purposes. Cloud computing provide large data storage capacity. Customers only pay for data storage as per their use. Data storage infrastructure is physically invisible to customers and it does not occupy their office spaces [42]. Google cloud uses hadoop file system to store and handle the big data [43].

Location independence is one of the major edge of cloud computing. Customers find the same desktop service on their move. They can access the services everywhere in the world where there is an internet connection. Due to the increase of hand-held devices, cloud computing is growing rapidly and is going more popular. It automatically manages and repairs itself from time to time. Cloud computing provides service on demand, always at any time, and anywhere [44].

Table 2.1: Terminologies used in the thesis

Term	Definition
Cloud computing	Cloud computing provides the desktop computing services online on the network. These services are categorized as IaaS, SaaS, and PaaS. Cloud providers charge the customer as per usage like utility services. virtualization techniques are used to distribute the resources among the users [45].
Service Level Agreement	Cloud is a virtual market, therefore, the services agreement is its crucial part. SLA is an agreement, agreed between service provider and consumer. Detailed Service Level Objective (SLO)s and parameters are discussed and signed. Penalties are enforced in case of violation [46].
Quality of Service	The cloud services purely depends on QoS. This covers the level of services discussed in SLA. The defaulter is penalized as per the agreed penalties structure. In cloud, delay, throughput and resources availability are considered as QoS [47].
Penalties	In case of QoS degradation or SLA violation, the provider is penalized as per the agreed penalty structure. Penalty is paid in cash or decrease in prices. This attract customers to trust the provider [48]. Penalties are the main reason towards revenue degradation.
Service Level Objectives	The SLO is the influential element of SLA. These are the main objectives the customer demands and agreed with Cloud Service Provider (CSP) [48].
Services Scalability	Service scalability is the crucial part of the cloud services. If the provider is able to scale the services as per the customers' workload, the services are called scalable. Horizontal and vertical scalability is used in cloud computing [49].
Physical Machine	Physical machine is a hardware based computer. This terminology is used to differentiate the computer from virtual machines. The virtual machines are created on physical machine [50].
Virtual Machine	It is not possible to provide individual physical machine to ever customers. Therefore, the physical machine services are virtually divided among the available workload. A separate virtual machine is created for every customer [50].
Cloud Service Provider	Cloud Service Provider is a company which provides the cloud services to the customer using virtualization. Google, Amazon and Salesforce are the leading CSPs [2].

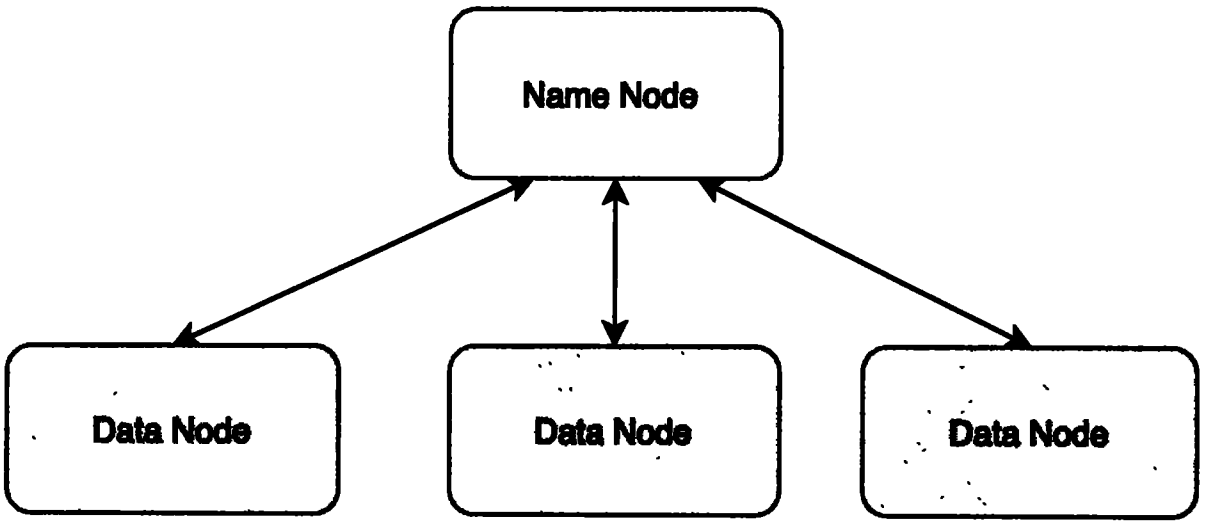


Figure 2.1: Structure of cloud storage

Term	Definition
Cloud Information System	Cloud Information System (CIS) contains the detail information of the cloud services and SLAs [51]. The detail information about workload and the resources are stored on it.

The most attracting feature of cloud computing is that it *converts capital expenses to operating expenses.* Organizations have not to consume capital expenses at the start of the business. Its services can be available on very low prices. Due to," *pay as you go*" feature, small and medium level companies can hire high quality services. Services are provided at customers' door step. It is *utility service* just like electricity, gas and telephone. Customers get connected with any cloud provider. They generate bill according to customers uses [52].

In cloud computing, *everything is in customers' control* and according to their needs. What , where and how to use, are customers' decisions. Services can be customized according to their needs. Due to customization, they do not need to pay for irrelevant functions and applications and also increase the speed of application. Maintenance is one of the major problem and budget affecting in traditional IT world. Cloud computing reduce the need of IT experts for maintenance of infrastructure. Provider is responsible for every type of maintenance [53].

Cloud computing is *highly scalable.* Customers can increase or decrease resources according to their organizational needs. Infinite resources are available on cloud servers. Virtualization is the main enabler of

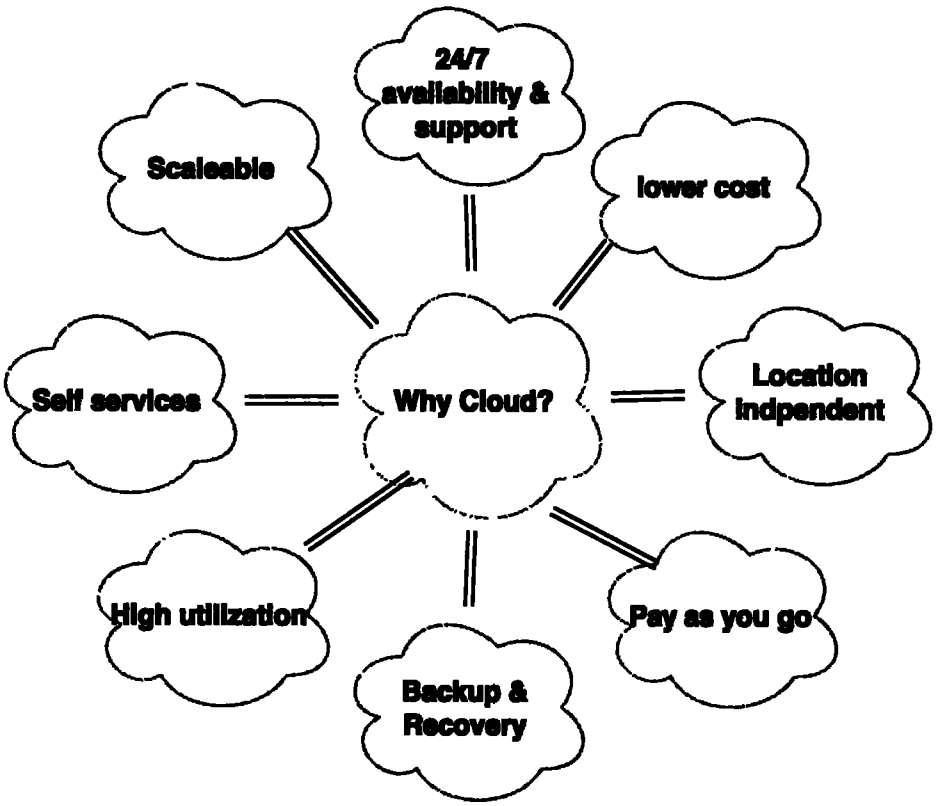


Figure 2.2: Why cloud computing

cloud computing. It shares the underlying infrastructure among many users. Every user thinks that only s/he is using those services. All this is possible due to virtualization, parallel and distributed processing techniques. Therefore, it provides very high-level utilization. Cloud computing provides unlimited space. Customers only pay for that storage which is used by them. Cloud providers give 99.99 % commitment of availability of stored data [54]. The characteristics of cloud computing is shown in figure 2.2.

2.1 Performance Management

To keep the end users satisfied, it is crucial to provide *high-performance* services. Customers buy services on cloud computing for performance. It is very complicated issue for the cloud provider. It is also challenging to convert the performance matrices to quantitative matrices for measurement. Cloud performance discusses the response time, running time, reliability and availability. SLA implements the agreed performance parameters. The performance of the services is tested by load test, stress test, functional test, and latency test [55]. Table 2.2 show the abbreviations and notations used in the chapter.

Table 2.2: List of abbreviations and notations used in the thesis

Abbreviations	Description	Abbreviations	Description
ICT	Information and Communication Technology	CC	Cloud Computing
SLA	Service Level Agreement	IaaS	Infrastructure as a Service
PaaS	Platform as a Service	SaaS	Software as a Service
AWS	Amazon Web Services	CAGR	Compound Annual Growth Rate
SLOs	Service Level Objectives	QoS	Quality of Service
IT	Information Technology	SS	Services Scal-ability
Eff	Efficiency	CLV	Customer Lifetime Value
PMs	Physical Machines	VMs	Virtual Machines
SPE	Service Provisioning Engine	RQE	Request Queue Engine
RSF	Response Satisfaction Factor	ASF	Assurance Satisfaction Factor
CSPs	Cloud Service Providers	ILP	Integer Linear Program
CIS	Cloud Information System	PSLA	Personalized Services Level Agreement
PerSLA	Performance based Service Level Agreement	CS	Customer Satisfaction
CSP	Cloud Service Provider	prof	Profit

2.1.1 Execution Time

Execution time shows the total time taken to execute the customers' workload. This depends on the request type and resources which is to be executed. If the resources are not appropriate, it takes longer than usual [56].

$$Per \propto 1/\tau_{run} \tag{2.1}$$

$$V_n \propto 1/Per \tag{2.2}$$

$$\eta \propto V_n \tag{2.3}$$

The above expression shows that performance (Per) is inversely proportional to total running time (τ_{run}). Further, the total number of SLA violations (V_n) is inversely proportional to performance. Wheres, penalties (η) are directly proportional to the number of SLA violation (V_n). These penalties have worse affects on cloud business.

2.1.2 Response Time

Response time is the waiting time of customer request in the waiting queue. Response time depends on the underlying resources utilization. If the underlying resources are heavily utilized, it takes longer to execute new tasks [57].

$$\tau_{res} \propto \upsilon \times SS \tag{2.4}$$

$$Per \propto 1/\tau_{res} \tag{2.5}$$

The above expression shows that response time (τ_{res}) is directly proportional to total customers' request (υ) and services scalability (SS). Further, performance is inversely propositional to total response time (τ_{res}). Total number of SLA Violation (V_n) is inversely proportional to performance. Whereas penalties are directly proportional to number of SLA violation (V_n).

2.1.3 Availability

Availability is defined as the presence of the agreed resources when they are required. Availability covers these resources which are discussed in SLA [57].

$$Avail \propto \frac{\tau_{avail} - \tau_{down}}{\tau_{com}} \tag{2.6}$$

$$avail \propto 1/fail \times SS \tag{2.7}$$

$$\chi \propto avail \tag{2.8}$$

In the above expression *Avail* shows the availability of resources, τ_{avail} shows the total availability, τ_{down} shows the down time, τ_{com} shows the total agreed time. Furthermore, availability is directly proportional to resources scalability (SS) and inversely proportional to system failure ($fail$). The cost χ is directly proportional to services availability.

2.1.4 Resources Reliability

Resources reliability is defined as the resources performing of the predefined functionalities for the agreed time under agreed terms and conditions. The resources are reliable if they are fault-tolerant and automatically recoverable. Reliability also includes the fault tolerance, recover-ability and resources constancy. Lower reliability reduces customers' retention which leads to lower revenue [57].

$$Per \propto Reliability \tag{2.9}$$

The above mathematical expression shows that reliable resources minimize the number of penalties.

2.2 SLAs and Penalties Management

Service Level Agreement (SLA) is an understanding, negotiated between a provider and a consumer. Detailed Service Level Objectives (SLOs) are addressed, expected services, Quality of Service (QoS) and performance are agreed and approved [5]. Both provider and consumer monitor services with agreed terms and conditions. If violations occur in agreed terms and conditions, penalties are imposed on provider [13]. Clearly explained SLA improves the customers' satisfaction and guarantee the continuous provision of services [30].

SLA violations leads to *penalties* that are applied in the form of lower prices during service failure or direct sanction. Usually, cloud providers accept loaded SLAs, but later on, they cannot provide resources in accordance with the agreement. As a result, they have to pay a large portion of their income in fines.

Performance, penalties, costs and revenues are complexly related. Their interdependence is explained in the following expressions.

$$Per \propto Rev \times \chi \times 1/\eta \tag{2.10}$$

$$Rev \propto 1/\chi \tag{2.11}$$

The above comparison explains that an increase in performance (*per*) maximizes the revenue (*Rev*) and minimizes the penalties (η), but performance is also proportional to the cost (χ), which is inversely proportional to the revenue. They have a complex correlation. The situation becomes more complicated if there is no proper framework, which clearly defines them.

Recent advances in Information Technology (IT), attracted more transition towards cloud computing. Furthermore, data ware and data mining techniques also attracts the market. Due to a large scale of data and internet business, it is very essential to have clearly defined SLA, otherwise provider will be disruptive with disastrous consequences in business.

2.3 Resources Scalability

Resource scalability is vital for QoS. Non scalable resources lead to penalties and revenue degradation. Most of the performance parameters directly depend on the resources scalability [58, 59].

$$V_n \propto \frac{1}{SS} \times \frac{1}{QoS} \times \frac{1}{Eff} \tag{2.12}$$

The above expression explains the relation of SLA violation (V_n) on services scalability (SS), Quality of Services (QoS), and services' efficiency (Eff).

The federated cloud concept was introduced for processing overloaded systems [30]. However, the federated cloud also imposes restrictions on the leasing of resources from specific providers registered with the federated cloud. Although, such resources can be easily and moderately available from non-federated cloud providers. The result is a kind of monopoly, in which a supplier is bound to get resources from a specific seller. We intend to design an approach whereby a service provider, in the case of an overload or with more

customers than the available resource, can easily or moderately operate or rent an external source where it is available conveniently and cheaply.

2.4 Customer Satisfaction

Customer satisfaction is extremely important,, which shows fulfilling of customers' needs. Customers are the indicators of repurchase. It shows the point of differences. It is cheaper to retain existing customers than to bring in new ones. Usually, companies spend millions of dollars on customers' attention but small investment on retaining them.

$$CS \propto SS \times Eff \times QoS \tag{2.13}$$

Customers are the potential of any company. A global survey by Accenture Global Customer Satisfaction report (2008) [60], shows that prices are not the most important concern; the most important are the customers' satisfaction. Successful customer satisfaction services increase Customer Lifetime Value (CLV) which increases the company's revenue. According to McKinsey 13 % unsatisfied customers give briefings about 9-15 people about their experiences. Customers are the true potential of any business, therefore, their satisfaction is most important for revenue and repeat purchases [36].

2.5 Resources Utilization and Provision

Services usage is the most important method for evaluating the performance of assets and determining the success of the company. Basically, it is a measure of the real income generated by the assets in relation to the potential income they could have earned. The cloud uses virtualization, parallel processing and distributed processing to optimize the use of the underlying resources [61]. In the following equation,

$$Rev \propto \rho \times \mu \times \nu \tag{2.14}$$

Wheres, Rev is revenue earned, ρ is prices, μ is resources utilization, ν is number of customers.

Usually, in cloud computing, resources are reserved. If reserved resources are not used by reserved customers, they are *underutilized* and wasted.These resources may be utilized with the permission of customers

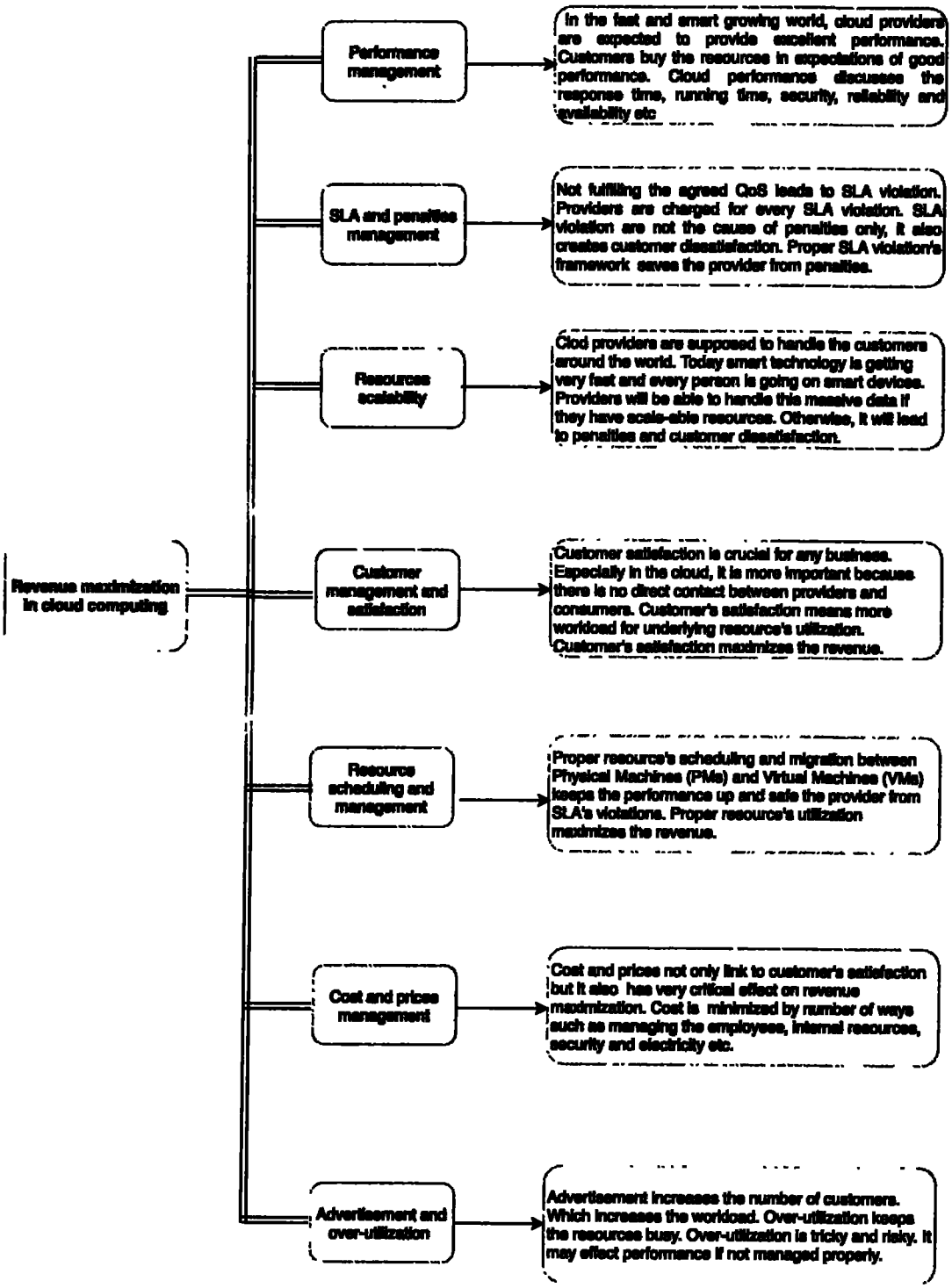


Figure 2.3: A taxonomy for revenue maximization approaches

for higher revenue. This benefits both provider as well as consumer. Resources utilization (μ) may be expressed as

$$\mu = \frac{Running(\sum_i^n VM)}{Available(\sum_i^n VM)} \tag{2.15}$$

Overutilization also creates issues for cloud providers. In extreme utilization, providers refuse some of the existing customers, having massive workloads. Refusing customers create dissatisfaction. Also, rejecting massive workloads dispossess providers from higher revenue. Customer satisfaction is very important, which shows that how much customers' needs are fulfilled [62].

Different IaaS *scheduling policies* are used to allocate resources to different customers. Scheduling policies greatly affect the resources utilization, customers' satisfaction and providers' revenue. Mainly used scheduling policies are; static scheduling and dynamic scheduling. In static scheduling, a prior request is sent for resources while in dynamic allocation resources are allotted as per customer's requirements [63].

2.6 Cost and Prices Management

Prices play a very important role in generating revenue. The cloud market uses different types of pricing. In the reservation pricing plan, customers reserve resources for a specific period, such as a month or a year. Resources are sold to customers with a reasonable discount. Customers pay the registration fee. In on-demand pricing, customers are billed individually. In this pricing system, prices are higher, however, providers are charged for breaching the SLA. In the spot pricing, prices are negotiated between customers and suppliers. Negotiation-based prices are used for underutilized resources. In differentiated pricing, cloud services are divided into different types of tier. Each tier has different prices. In unit pricing, customers are charged on a unit of space or bandwidth used. This pricing mechanism is more flexible than the tiered pricing mechanism. In the basic pricing of the subscription, customers are billed according to their subscription. Users receive a discount on early booking. The disadvantages of this pricing model are that the provider provides guaranteed services to customers and underutilization wastes resources. Usage-based pricing is also used by Amazon in which customers are charged based on usage [36, 64].

For provider, the basic cost is calculated as

$$\chi_b = \chi_{hu} + \chi_n + \chi_m^{ii} + P + \chi_{sec} + \kappa \tag{2.16}$$

Chargeable prices are calculated as

$$\rho = \chi_{cost}^{ii} + \Delta \tag{2.17}$$

Wheres, χ_b is the total basic cost of the resources, χ_{hu} is the hired unit cost, P is power consumption, χ_n is network cost, χ_m^{ii} is migration costs, χ_{sec} is security cost, Δ is margin, and κ is constant.

Introducing different pricing models in one business is complex but it attracts more customers, which increases the resources utilization.

2.7 Advertisement and Overutilization

Advertisement spread positive prospective and change negative impact. It attracts new customers and increases the utilization of under-laying resources. Auction is also used in cloud business to increase the resources utilization. Auction is usually used for the underutilized resources. Instead of wasting, auction gives some revenue. It is very tricky because in case of SLA violation, penalties are imposed on provider party which minimize the provider revenue [65].

TH-26376

2.8 Summary and Conclusion

In this chapter, the main parameters of revenue maximization are presented in detail (as shown in figure 2.3). In cloud computing, the key indicators affecting revenue are service performance, SLA management, resource scalability, utilization and availability, customers satisfaction, cost and price management, as well as auctions and advertisement. Service performance is tested by load tests, stress tests, performance tests, and latency tests to calculate running and waiting times. Performance, penalties, costs and revenues are complexly linked. Most of the performance parameters depend directly on the scalability of resources. The third parameter, the scalability of resources, deals with the problems posed by evolving resources. Customers' satisfaction is very important because it shows how many services meet the needs of customers. The use of resources is the key method for measuring the success of a business. Cost and prices mechanisms are discussed to offer different pricing to satisfy every type of customers. The higher payer is provided high-performance services while low payer customers is provided lower performance services. Advertisement spread positive perspective and change negative impact about any business. It attracts more customers to maximize resources utilization. The last parameter deals with such issues.

Chapter 3

Literature Review

The literature on revenue maximization is diverse and does not depend only on few parameters. Different authors have used different parameters and techniques for revenue maximization. In this chapter, the literature on revenue maximization is classified (as shown in Table 3.3) into performance management, SLA and penalties management, resources scalability, customers' satisfaction, resource utilization and provision, cost and prices management, advertisement and overutilization. For these parameters, research articles published between 2012 to 2019 are included in this study.

3.1 Methodology Used for Literature

The recent development in smart systems and a massive increase in smart devices, encourages to review the revenue maximization in cloud computing. Exploring the literature shows that extensive research articles are available on revenue maximization, however, they have not been analysed deeply. Therefore, we have analyzed the literature in depth to thoroughly discuss the opportunities and challenges of income maximization in cloud computing.

Table 3.1: Research string used to search the related literature

Area	Keywords	Synonyms in literature
Population	Cloud computing	Mobile computing
Methodology	Techniques	Resources OR prices OR SLA OR Penalties
Outcomes	Revenue Maximization	Profit Maximization

To get a more comprehensive view of cloud computing, we explored the literature from 2012 to 2020. For that period, we collected around 200 articles in which 70 were finally selected. We used Google Scholar [66], Elsevier [67], IEEE Xplore [68], and Science Direct to search for target articles. Google Scholar provides access to all articles published in a journal, and research libraries provide access to limited, high-quality articles published in associated journals.

To search the digital world, the search string is needed and the quality of the search purely depends on this string. The search string (shown in table 3.1) combines the keywords and includes population, methodology and results. The methodology of this research paper is divided into various phases and sections, mainly the planing, implementation and reporting. The remaining part of this section explore these phases.

3.1.1 Planning the Review

The planing phase includes two main objectives; (i) the importance and necessity of the study that distinguishes it from other related studies; and (ii) development of the research protocol and inclusion and exclusion criteria. Therefore, in the first phase, we designed the study protocol to search the journals and related articles. We further developed the inclusion and exclusion criteria. The development of the research protocol is crucial and also critical. The right protocol leads toward the best review; however, the defective protocols lead authors in other directions and leave the main focus. Therefore, emerging research questions, search strategies and selection criteria are discussed and identified at this stage.

Recent technologies have revolutionized the way of living. Every perspective of life is getting smarter such as smart cities, smart health, smart agriculture, smart power plants, etc. This is how huge devices connect to the internet, which is expected to reach 75 billion by 2025. These devices will provide major investment in cloud revenue. In order to do this, cloud computing needs an optimized framework to generate handsome revenue. This article covers the same domain to bring all the information together on one page.

3.1.2 Conducting the Review

In this phase, the study is conducted according to the protocol designed in phase 1. Most crucial is the identification of the study and to this end, each study is analyzed for three important checks.

1. The first one is the *population* of the research. This research covers the revenue maximization in cloud computing, therefore, the population of this article is cloud computing or mobile clouds.
2. The second check is the *methodology* or *technique* which is used to get the desired outcome. In this article, various revenue maximization techniques such as SLA and penalty management, resource scalability, customer satisfaction and management, resource utilization and provision, cost and price management, and advertising and auction are covered.

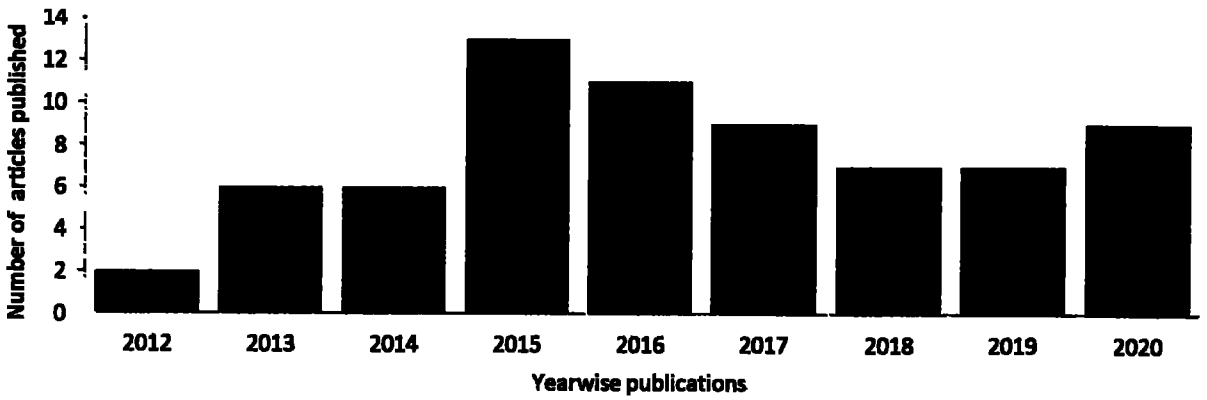


Figure 3.1: Number of year wise publications

3. The third one is the *outcome* achieved at the end of the research. In this case, the outcome is *provider revenue maximization.*

Pursuing these checks, the next most important phase is to design research questions. In this context, the research questions are;

1. How many research papers are published covering the revenue maximization in cloud business?
2. What are the main influential factors towards revenue maximization in cloud computing market ?
3. What are the main challenges and resistance towards revenue maximization in cloud computing?
4. What are the possible solutions to the issues towards revenue maximization in cloud computing?

The RQ1 deals with population, the RQ2 deals with methodology and RQ3 and RQ4 deals with the outcome.

For complete string we use "AND" to combine them (as shown in Table 3.1) for example;

Population AND Methodology AND Outcomes.

Now putting the related literature synonyms using OR logical operator.

(Cloud computing OR Mobile computing) AND (Resources OR prices OR SLA OR Penalties) AND (Revenue Maximization OR Profit Maximization)

3.1.3 Quality Assessment

The quality assessment of this study depends on the number of parameters. To cover this study, we have taken the following parameters to ensure the quality of the papers.

Inclusion criteria

1. The papers, cover the methodologies in cloud computing such as performance, SLA and penalties, scalability, customer satisfaction, resources utilization, cost and prices and advertisement and auction etc.
2. Discussing the cloud, fog and IoT for revenue maximization.
3. The presentation of the methodology and results in proper way
4. Full filling the above requirements along with 2 citation per year.
5. The research articles published since 2012.

Exclusion criteria

1. The research papers discussing the cloud computing and revenue maximization separately and there is not link between them.
2. The research papers, not properly presenting the results and methodology used for desired outcomes.
3. The research papers, failed to get two citation in last year.
4. The research article not published between 2012 and 2020.

3.1.4 Reporting the Review

In the final step, the meaning-full articles, covering the keywords and the research question, is extracted and presented in this study. The success of the review depends entirely on how the final review is presented in the document. Table 3.2 and Fig. 3.1 shows the year wise and Fig. 3.2 shows the category wise publications since 2012.

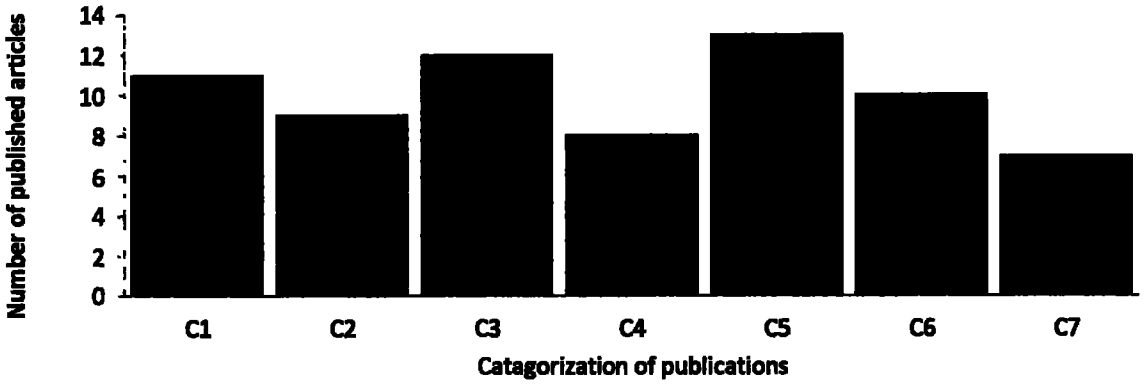


Figure 3.2: Number of category wise publications

3.2 Categorization of Literature

At this section, the literature on revenue maximization is classified (as shown in Table 3.3) into performance management, SLA and penalties management, resources scalability, customers' satisfaction, resource utilization and provision, cost and prices management, advertisement and overutilization.

Table 3.2: Detail summary of year wise publications since 2012

Category	2012	2013	2014	2015	2016	2017	2018	2019	2020	Total
Performance management	0	1	1	2	1	0	3	2	1	11
SLA and penalties management	1	0	0	1	0	3	2	1	1	9
Resources Scal-ability	0	1	2	2	1	1	1	2	2	12
Customer satisfaction	1	2	0	1	1	2	0	0	1	8
Resources' utilization and management	0	0	2	2	2	3	1	1	2	13
Cost and prices management	0	0	1	3	4	0	0	1	1	10
Advertisement and auctions	0	2	0	2	2	0	0	0	1	7
Total	**2**	**6**	**6**	**13**	**11**	**9**	**7**	**7**	**9**	**70**

3.2.1 Performance Management

Without reliable performance, services are useless in cloud computing market competition. No one can deny its importance. Investment in low performance services will give no benefits. Such type of issues detract customers rather than to attract. The following research studies investigated the performance towards the revenue maximization. Table 3.4 summarizes the performance related studies for achievement and limitations.

Performance depends on different parameters. Sinung Suakanto and Saragih [37] investigated the *primary performance parameters* in cloud computing. They measured performance metrics using empirical methods. Average response time and time out of customers' requests were calculated in the cloud environment. Their results showed that an increase in the number of customers increases the average response time. Similarly, an increase in the number of users, increases the request time out. This study considered the performance measurement of cloud computing and achieved C1 criteria.

Ran and Xi [69] advances the *performance study*. They worked on resource provisioning strategy with QoS constraints. The proposed framework used dynamic computing resources provision, cost-saving, and QoS guaranteed services. An algorithm Service Provisioning Engine (SPE) and Request Queue Engine (RQE) were used to efficiently provide the resources. They tried to maintain QoS while minimizing the total cost. With performance, resources and cost management, this study achieved C1, C5 and C6 criteria. Danilo Ardagna and Wang [70] collected and analyzed the detailed literature about *QoS*. The aim of their investigation was to study the QoS modeling area, categorizing contributions according to relevant areas and methods used. This study achieved performance management C1 criterion.

Over provision of Virtual Machine (VM) degrades the performance. Underutilization also wastes resources. Kundu et al. [71] addressed this challenge by *efficient resources allocation*. Resources were allocated dynamically. Three types of algorithms were proposed in this model. The first algorithm is MaxRevenue which searches VM for maximum revenue. The second algorithm searches MaxGain and MaxLoss in all available VMs. The third algorithm compare the MaxGain and MaxLoss. They search MaxRevenue and MaxLoss VM. With these properties, it achieves performance management C1, resources scalability C3 and resources utilization and scheduling C5 criteria.

The same issue was further investigated by Feng and Buyya [72]. Revenue-oriented resources allocation was used for revenue maximization. Two types of solutions were proposed for revenue maximization: (i) optimizing resources allocation and (ii) optimizing pricing mechanism. The main objective of their work was to find the proper allocation of servers among all instances to maximize the provider revenue. Two types of functions were discussed in this article: (i) Assurance Satisfaction Factor (ASF) and (ii) Response Satisfaction Factor (RSF). Both of these functions defined the achieved performance. In the pricing model, if agreed performance (ASF & RSF) is achieved then customers are charged on regular price otherwise the

h

Table 3.3: Literature evaluation criteria

Symbol	Criteria	Criteria Definition
C1	Performance management	In the fast and smart growing world, cloud providers are expected to provide outstanding performance. Customers buy the resources in expectation of good performance. Cloud performance includes the response time, running time, security, reliability and availability .
C2	SLA and penalties management	Not fulfilling the agreed QoS leads to SLA violation. Providers are charged for every SLA violation. SLA violation is not the cause of penalties only, it also creates customer dissatisfaction. A proper SLA violation framework saves the provider from penalties .
C3	Resources Scalability	Cloud providers are supposed to handle customers around the world. Today smart technology is getting very fast and every person is going on smart devices. Providers handle this massive data if they have scalable resources. Otherwise, it leads to penalties and customer dissatisfaction
C4	Customer Satisfaction	Customer satisfaction is crucial for any business. Especially in the cloud, it is more important because there is no direct contact between providers and consumers. Customers' satisfaction means more workload for resources utilization. Customers' satisfaction maximizes the revenue.
C5	Resources' utilization and management	Proper resources scheduling and migration between Physical Machines (PMs) and Virtual Machines (VMs) keeps the performance up and saves the provider from SLAs' violations. Proper resources' utilization maximizes the revenue.
C6	Cost and prices management	Cost and prices not only link to customers' satisfaction; it also critically affects revenue maximization. Cost is minimized by a number of ways such as managing the employees, internal resources, security and electricity.
C7	Advertisement and auctions	Advertisement increases the number of customers, which increases the workload. Overutilization keeps the resources busy. Overutilization is tricky and risky. It may affect performance if not managed properly.

provider is penalized and low prices are charged. With prices, performance and resources management, this study meets C1, C5 and C6 criteria.

Federation enhances the scalability, and results in increased performance. Nazanin Pilevari and Sanaei [73] explained the *federated CSP* to optimize the service quality and providers' revenue. They proposed an algorithm using an Integer Linear Program (ILP) to form the CSP federation. They also proposed a heuristic-based algorithm for cloud federation formation following the ILP. This study achieved performance management C1 and service scalability C3 criteria.

Number of studies contributed to improve this domain. Koziris [74] proposed an approach to overcome transient cloud failures that happen during the application deployment. Apon [75] gave a systematic evaluation of Amazon Kinesis and Apache Kafka for the highly demanding applications. Bauer [76] proposed a framework which keeps the performance up automatically. Gerndt [77] proposed the auto-scaling performance evaluation for two-layered virtualization in cloud computing. Wang [78] proposed the Virtual Machine Placement Algorithm for high performance.

Delay balancing, with today massive devices, is a hot issue. Gadey [79] investigated the energy consumption and delay balancing in IoT, Fog and cloud project. They focused on two parameters, energy consumption and quality of service and used evolutionary algorithm to resolve this issue. The same issue was further studied by Duan [80]. They introduced a general framework for IoT, fog and cloud. They proposed a delay-minimizing policy for IoT devices to minimize the service delay for IoT, fog and fog applications.

These investigations and research optimized the revenue and performance, however, there is a complex correlation between performance and revenue which is missing in the existing literature. For example, performance increases the customers' attention, however, on the other hand, this also increases the prices. Increase in prices, detracts the customers. Furthermore, with good performance, heavy workloads are expected. This workload affects performance and SLA. All these queries need thorough examination and investigation.

Table 3.4: Contributions and limitations of performance related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Feng and Buyya [72]	Authors in this paper proposed Assurance Satisfaction Factor (ASF) and Response Satisfaction Factor (RSF). The customers are charged for regular prices if performance (ASF & RSF) is achieved. In case of defaulter, the prices are minimized.	Customer satisfaction is very complex. The main challange is the satisfaction measurement. Authors failed to explore it in detail that how did they measure the customer satisfaction [81].
Kundu et al. [71]	Efficient resources allocation to dynamic requests for revenue maximization. MaxGain, MaxLoss and MaxRevenue algorithms were used to select the best economical VMs. Incoming requests were run on globally selected VMs	Over provision of VM violates SLA and decreases services performance. Performance badly affects incoming customers [82] .
Nazanin Pilevari and Sanaei [73]	Authors proposed the federated CSP to retain the performance. The federated providers shares the resources and workloads are migrated to other server in case of performance degradation, resulting a performance retention.	The challenge over here is federated cloud. Furthermore, the provider depends on third party for performance. If the third party performance decreases, this will degrade the system performance [83].

3.2.2 SLA and Penalties Management

SLA is a *contract* negotiated between a provider and consumer. Detailed service parameters are discussed and signed before starting the business. SLA creates a trusty relationship among business parties. If the signed parameters are violated, the defaulter is penalized. Penalties have worse effects on the provider side. With proper management, they may be minimized to maximize the revenue. Table 3.5 summarizes SLA and penalties management related studies for achievement and limitations.

Efficient resources management can save providers from SLA violation and penalties. Macas et al. [84] explored revenue maximization in cloud computing using Economically Enhanced Resource Manager (EERM). This used bi-directional data between the market and tried to increase sales through dynamic pricing mecha-

nisms. If the provider is unable to respond to the customer's request, a list of SLA violations is made. These SLAs are violated, resulting in a smaller loss. SLA that have lower revenues are also cancelled. If a VM is overloaded, the workload is shifted to other VMs to reduce the performance reduction. This SLA model gives a good idea about increasing revenue but it also rejects high loaded SLAs, which creates un-trusty situations in business. With prices, resources and SLA management, this framework achieved SLA violation management C2, resources scheduling and management C5, and cost and prices management C5 criteria.

Wu et al. [85] further studied the *resource allocation* to avoid SLA violations. The focus of this study was the dynamic changing customers' demands with SLA. Services are provided according to SLA. The customers can also change their requirements. Providers try to increase the profit by ensuring QoS to broaden their business. The software is delivered in standard, professional and enterprise and accounts are created in a group, team, and department. The contract is signed between a provider and a customer, if anyone violates, the defaulter pays penalties. The main objective of this work was to maximize the profit for the provider by minimizing the cost of VMs. By providing individual VMs to every user, no QoS degradation occurs, which minimizes the penalties. This paper achieved C1 for managing performance, achieved C2 and C3 by managing resources to avoid SLA violations.

Automatic Service Level Agreement (ASLA) was proposed by Christpher Redl and Schahram [86]. It uses past knowledge, user requirements and job evaluation to automatically meet every SLA. Mapping is determined by the market participant. Before SLAs are made in the cloud market, customers and suppliers submit their SLAs. These templates include service level contract statistics, SLA parameters and service level objectives. In the market, users associate their private SLA with a public SLA that is closest to their needs. SLA assignment is used to assign two SLAs. It automatically searches for similar SLAs on the market. ASLA lowers the costs of the market. With SLA and time management, this research resulted in C2 SLA management. Automatic SLA saves time and money, unfortunately, it is only an agreement and does not guarantee the quality of the service and performance.

Investigations were limited to get all the cloud resources from any single point. Jennifer Ortiz and Balazinska [87] worked on a *Personalized Service Level Agreement (PSLA)*. It acts as a broker between the service provider and customer. PSLA rents out different types of services from different service providers and then offers them to other customers based on their needs. PSLA has solved this major issue. The user does not have to translate his requirements. They only upload their needs to the broker and he provides services according to the needs of the client. PSLA has solved the major problem of service provision, but it is a combination of different services and each service has a different service quality and service level. This enabled C2 criteria for service level management.

For SLA, it is important to define measurable parameters. Emeakaroha et al. [88] advanced this idea by proposing *LoM2HiS (Low Metric to High-Level Services)*. It was part of the FoSII infrastructure. This

SLA framework provides a platform for converting low-level statistics into high-level metrics statistics. This infrastructure contains the assigned metric repository and an agreed service level agreement. When a new client request arrives at the system, this infrastructure has assigned it to an assigned metric repository. LoM2HiS is an automatic framework for managing and maintaining service level agreements. This framework informs about future threats. With SLA and resource management, this study met the SLA and penalty C2, and resources management C3 criteria. This framework is the first step to measure the performance of cloud computing. However, it does not describe how these metrics should be measured or how they should be analyzed and integrated into a service level agreement for implementation.

To analyze the SLA violations, Iyer [89] proposed analysis and diagnostic framework. Their study is based on 283 days of operational logs of the platform. They received workloads from 43 customers, spread around 22 countries. They developed tools to analyze this workload. This study showed that about 93 % SLA violation is caused by system failure. This study achieved SLA and penalties management C3 criterion.

Many authors and scholars contributed to enhance the SLA in cloud. Shivani and Singh [33] reviewed the detailed literature about *SLA violation and its minimization*. Shahin Vakilinia and Elbiaze [34] proposed an integrated platform to detect and predict conditions where improving decisions are required. They used neural networks to minimize SLA violations. Their results showed that this improve web request response time by up to 7 % and decreases SLA violation by 79 % in the context of the web application. Gargouri [35] used advanced SLA management strategies to provide good quality services. This reasoning technique minimizes SLA violation. Alayat Hussain et al. [90] proposed a Risk Management-based Framework for SLA violation abatement (RMF-SLA). This framework detects the SLA threat and recommended action is taken to avoid violation. Singh and Elgendy [91] proposed three approaches, namely, gradient descent-based regression (Gdr), maximize correlation percentage (MCP), and bandwidth-aware selection policy (Bw), that can mostly reduce power losses and violations. Jin [92] proposed a privacy-based SLA violation detection approach for cloud computing using Markov decision process theory. Zhonghai [93] worked on availability commitment in cloud computing to minimize SLA violations.

SLA and *penalties* are deeply investigated in literature. Different mechanisms are reviewed to minimize the SLA violations. Penalties are also well investigated to keep this burden minimum. The main drawback which needs improvement that most of the provider cancelled the SLA as their workload increases. Most of the provider with limited resources also have admission control and heavy loaded workloads are cancelled. These issues need further exploration.

Table 3.5: Contributions and limitations of SLA and penalties related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Macas et al. [84]	Economically Enhance Resource Manager (EERM) is used for revenue maximization. In extreme utilization, those SLAs are rejected whose penalties are lower.	Rejecting existing customers and not accepting heavy loaded SLAs have long term bad impact on business [94].
Adil Maarouf and Haqiq [95]	SLA's penalties functions, strengths and weakness were explored. A novel penalty framework was proposed for calculating the penalty of the violations and presented a formulation for this penalty definition.	Only a penalty framework is discussed. Further exploration is required to minimize the penalties [96].
Christpher Redl and Schahram [86]	Automatic Service Level Agreement (ASLA) automatically match the resources as per the need of SLA. This minimizes the market cost and increase the providers revenue	It saves time and money, but unfortunately, it is only an agreement. It does not guarantee the quality of the service and performance [96].
Jennifer Ortiz and Balazinska [87]	Authors proposed PSLA, acts as broker between provider and consumer and rent different type of services from single point as per the customers need.	The broker hires the services from various providers. If provider services performance goes down the whole system be affected [97].

3.2.3 Resources Scalability

Resources scalability is directly proportional to performance, which is directly proportional to revenue. With scalable resources, more customers are entertained with excellent performance. With non-scalable resources, performance decreases which leads to SLA violations. Therefore, for the cloud business, resources must be scalable. Table 3.6 summarizes resources scalability related studies for achievement and limitations.

Scalable resources are directly proportional to revenue generation. For resources scalability, Gao et al. [98] proposed Cloud Bank Service Level Agreement (CBSLA). In CBSLA, services are used by the service provider and these services are stored in a service pool. Two types of SLAs are used. The first SLA is signed by the service provider and the cloud bank, while the second is signed between the service consumer

and the cloud bank. Cloud banking works as a cloud service broker. Various SLAs are being negotiated with various cloud service providers and customers. Pooling services is also a difficult task, so their use and implementations are very complex. This investigation achieved the SLA management C2 and partially resources management C5 criteria.

Insourcing and *outsourcing* is the first step to deal with resources scalability. Hadji and Zeghlache [99] utilized these techniques in federations. The provider uses outside federation only when its cost is lower than internal cost, and also insourcing is used when internal utilization is lower. Mathematical programming approach is used for optimum outsourcing and insourcing decisions. For minimum cost and maximum revenue, four possible actions are discussed. An optimal number of machines are activated for any request. On maximum utilization, some requests are outsourced. In limited utilization, some of the internal resources are insourced. Nodes which are not in use, are turned off to save power. The main limitation of this framework is that insourcing and outsourcing can be done only with registered providers. With insourcing and outsourcing capabilities, this study achieved resources scalability C3 criterion.

Resources migration is one solution to handle the overutilization challanges. Upadhyay and Lakkadwala [100] advanced the resources migration in cloud computing. Migration is used in distributed systems, when data and applications are transferred from overloaded systems to underutilized systems. The proposed framework used two types of algorithms. The first algorithm checks the overloaded and underutilized systems. Only if the target system has enough space to run the burden of the overloaded system, the workloads are transferred to it. The second algorithm works for effective resources allocation in the cloud system. With migration property, this study achieved resources scalability and management C3 criterion.

Migration work was further advanced by Li et al. [101]. A framework was proposed to migrate data to other systems. Migration may be as a whole application work, partial migration, component replacement or codify. Those methods need different architecture and environment to migrate the data. They used the Eucalyptus platform to evaluate their framework. This study discussed the cost and prices C6 and resources scalability C3 criteria.

Mobile cloud is getting a big share in utility computing. Their reliability increased with the usage of fog computing. Samanta and Chang [102] investigated mobile cloud revenue. It is not possible to perform every application task on mobile because it requires huge space and memory. They have designed an approach to make some of the cloud sources and some sources work on the peripherals. They proposed an adaptive release system for Mobile Edge Computing (MEC) services to maximize total revenue. The execution of certain processes on the edge and some on the cloud server influences the performance due to network delays. They consider the delay-sensitive and delay-tolerant edge services by designing an offloading algorithm. By migration task between edge devices and a cloud server, this study achieved resources scalability C3 criteria. A challenge with this scheme is that migration of live data between edge devices and cloud

servers decreases the performance. It may also create security issues.

User distance from server decreases the performance and increases the delay time. Hou Deng [103] addressed this gap to maximize the revenue of *geographically distributed* data centres. The solution for this is to build geographical data centres but It needs too much investment to build new data centres. The authors proposed a solution for this issue to hire geographical resources from universities or other institutions to process the data locally. This minimizes the cost of the provider. The challenge with this approach is that getting geographical resources may need many SLAs as per the region. It is also not possible to hire these resources around the world. Secondly, it may have worse effects on performance. With cost minimization, this study achieved C6 criterion.

As per the above discussion, authors offered different solutions to handle the *resources scalability* challenges. Federated cloud partially overcome this issue by sharing resources within the union. However, the drawback of this proposal is that providers are compelled to hire resources from particular providers with fix rates. Another study suggests that workloads should be accepted from only the surrounding area. This study has some positive directions that with a lower distance, performance of the system may increase, however, it needs capital investment to build data centers.. All these questions need to be addressed and require further focus in future research.

Table 3.6: Contributions and limitations of resources scalability related studies to maximize the providers' revenue in Iaas cloud

Papers and Authors	Major Contributions	Limitations
Hadji and Zeghlache [99]	Insourcing and outsourcing were investigated in the federation to maximize the cloud provider's revenue. The provider used outside federation only when its cost was lower than internal cost, and also insourcing was used when internal utilization was lower.	They did not discuss efficient algorithms for insourcing and outsourcing. Insourcing and outsourcing minimizes the performance [104].
Gao et al. [98]	Cloud-based transcoding is a new way, which saves time and resources. In cloud computing, with numbers of VMs availability, parallel transcoding is used, which greatly reduced resources and time wastage.	This framework is only for live video transmission and cannot be utilized in other data.

Papers and Authors	Major Contributions	Limitations
Upadhyay and Lakkadwala [100]	The proposed framework checks the overloaded and underutilized systems. Only if the target system has enough space to run the burden of the overloaded system, the workloads are migrated to it.	Migration increases the resources scalability, however, it also have bad impact on performance [105].
Samanta and Chang [102]	Mobile Edge Computing (MEC) is positively affecting both the revenue and performance. Instead of transferring all the data on cloud, some of its portion is processed on edge.	Though, this minimizes cost and delay, however it needs capital revenue to invest on edge devices [106].

3.2.4 Customers' Satisfaction

For every business, *customer satisfaction* is the top priority. How good quality and scalable resources do providers have but if customers are not satisfied, revenue may not be earned. Customers' satisfaction is important in any business but in cloud computing, it is much important because there is no direct communication between provider and consumer. Customers' satisfaction is based on the performance parameters discussed in the performance management portion. If these parameters are achieved with agreed SLAs, customers will be satisfied. The following research studies discussed customers' satisfaction and classification in the cloud computing business. Table 3.7 summarizes customers' satisfaction related studies for achievement and limitations.

Customers' satisfaction is hard to deal with in terms of measurement and keeping them satisfied. Nazanin Pilevari and Sanaei [73] developed conceptual criteria to *measure customers' satisfaction*. These criteria were based on previous studies and expert opinions. The criteria consist of security, efficiency and performance, adaptability and cost. Secondly, they developed a fuzzy logic, both, human and machine to fulfill the above criteria. With the customer satisfaction investigation, this study achieved C4 criterion. R. A. Asaka and Ganga [107] extended the customers' satisfaction to Software as a Service (SaaS) cloud. This model followed the survey and statistical analysis of client accounts from one of the world's largest IT companies. According to their findings, quality of the execution, quality of the implementation, and relationship are factors with higher influence on client satisfaction. They achieved customers satisfaction management C3 criterion.

With limited resources, it is hard to satisfy customers. Dividing customer satisfaction parameters into different layers may ease the work of the providers. Hamsanandhini and Mohana [108] *categorized* all clients into different groups. They used different policies to maximize revenue. The policies selectively violate the

SLA. Overselling resources, hybrid pricing, booking already used resources, and client priority is artificially added. Clients are classified on different parameters such as the clients' relation to the provider and quality of service required by the customers. With the client classification framework, this study achieved customer satisfaction C4 criterion.

Customer classification was further studied by Huu and Tham [109]. They worked on SLA enforcement by client classification. They introduced a set of policies to manage SLA during its operations. They classified the clients according to their affinity and QoS. According to these policies and classification, high-priority clients are selected. According to their classification services are provided. They achieved performance management C1 and customer satisfaction C4 criteria. Manzoor et al. [110] worked on *customers centred approach* for IaaS cloud. The proposed framework works in three phases. In the first phase, customers submit their requirement specification to the CSP and providers start service provision. In the second phase, services are monitored according to CIS. In the third phase, monitoring reports are compared with CIS. With a customer-centred approach, this framework achieved C4 criterion.

Mei et al. [111] discussed two customer satisfaction parameters, which affect the revenue most, *QoS* and *Price of Services (PoS)*. QoS shows the expected performance and PoS shows the comparison between the predefined price and the actual price. They developed a model which optimizes the QoS and prices for customers' satisfaction. With customers' satisfaction, the number of customers increases which increases the revenue of the provider. With QoS, customer satisfaction, and prices management this work achieved C1, C4 and C6 criteria.

The above studies investigated customer satisfaction challenges and suggested different frameworks to satisfy customers. The main contributions of these studies are to classify the customers into different layers and according to these layers, providers create a customer satisfaction layer. Questions about what to do in case of lower resources with higher workloads and also to optimize the performance and prices still remain open. These need further investigations to optimize performance and prices and also to handle massive resources with limited resources.

Table 3.7: Contributions and limitations of customers' satisfaction related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Mei et al. [111]	Customers' satisfaction was explored for revenue maximization. A model was developed which optimized QoS and prices for customers' satisfaction. With customers' satisfaction, the number of customers increases which increase the revenue of the provider.	Customers' satisfaction increases the customers, however, to handle these customers' workload with limited resources, is still an issue [112].
Wu and Buyya [113]	This study focused on the dynamic changing customers' demands and QoS concerning SLA. The services are delivered in standard, professional and enterprise and accounts were created in a group, team, and department. This attracted more customers.	Rejecting existing customers and not accepting heavy loaded SLAs have long term bad impact on business [114].
Hamsanandhini and Mohana [108]	They categorized the clients into different groups and proposed to selectively violate the SLAs for minimum penalties.	Violating the SLAs is not a good approach. Customers never trust on such providers who reject customers [94].
Huu and Tham [109]	They classified the customers as per their priority and as per their priority services are provided.	Priority creation means that have to use different prices for different customers. This will complex the process [115].

3.2.5 Resources Utilization and Provision

Cloud resources are not store-able and wasted, if not utilized in time. Resources utilization extend the cloud provider business. For proper resources utilization, it is necessary to have a good utilization and scheduling framework. The following studies investigated these cores to increase provider revenue. Table 3.8 summarizes resources utilization and provision related studies for achievement and limitations.

Cloud resources need to be utilized on time. Similar to other utilities (e.g. power or water), cloud resources cannot be stored to be used later on. To fill this gap, Shin et al. [116] proposed an algorithm which enhances *deadline guaranteed resources utilization*. All jobs are sorted according to their arrival time, each job worse execution time is calculated. All VMs resources information is sorted in a CIS. VMs are allotted to different

jobs using worse case execution time and deadline sorting. With these properties, this research achieved resources scheduling and management C5 criterion.

In cloud computing, resources and workloads are geographically distributed. In this situation, it is very difficult to perfectly match virtual machines with different workloads. Balagoni and Rao [117] worked on the task planning policy in a heterogeneous cloud environment. This study investigated the locality predictor that increases the matching factor and the performance of cloud computing. They developed an algorithm that worked as basic functions on location, and load prediction.With these characteristics, this study achieved resources management and scheduling C5 criteria.

The previous studies stressed admission control, however, Yuan et al [118] proposed *Profit Maximization Algorithm (PMA)* with delay tolerance. They proposed temporal task scheduling for profit maximization in hybrid clouds. They addressed the problem, handling all the incoming tasks with limited private cloud computing resources. Private cloud workloads are scheduled to the hybrid cloud. The temporal task scheduling algorithm allows running the private task on the private and public cloud. With scheduling properties, this work achieved the resources scheduling and management C5 criterion.

Ibrahim et al. [119] worked on *task scheduling* in cloud computing. They proposed an enhancing task scheduling algorithm, which calculates all available resources and task request for processing. Groups of users are allotted to different VMs according to the ratio of needed power. With resources scheduling algorithm, this study achieved the resources scheduling and management C5 criterion.

Live cloud migration was utilized by Mansour et al. [120]. This work is divided into three phases. In the first phase, permission is granted to every VMs for migration. In the second phase, the required information for resources is gathered to decide either to migrate the resources or not. In the third phase resources utilization are monitored to avoid overutilization. This study discussed the resources scheduling C5, and cost and prices C6 criteria. Santikarama and Arman [121] developed an architecture framework for non-cloud to cloud migration. They used Economic Customer Relationship Management (ECRM) to efficiently migrate the data. This study achieved the resources scheduling C5 criterion.

Live migration was further investigated by Tsakalozos et al. [122]. They proposed a framework which reduces SLA violation by migrating the resources on time. They proposed a scalable and distributed network for customers. The migration is done within time windows. This study achieved resources scheduling C5 and SLA management C4 criteria. Gao et al. [98] worked on *transcoding* in the cloud for profit maximization. Transcoding is widely used for online video streaming. They proposed time scale stochastic optimization framework to maximize provider profit and also service performance. Transcoding in normal condition, waste about 30 % resources and time. Cloud-based transcoding is a new way, which saves time and resources. In cloud computing, with numbers of VM availability, parallel transcoding is used, which greatly reduced resources and time wastage. This work achieved resources scheduling C5 criteria.

The above review explains the optimum utilization of cloud resources, its challenges and suggested different frameworks to maximize resources utilization. The main challenges toward resources utilization are admission control and SLA violation. Providers do not overload their resources due to the fear of SLA violation. These complexities need further investigations to optimize resources utilization and SLA violation.

Table 3.8: Contributions and limitations of resources utilization related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Qi Zhang [123]	They developed a framework which uses a market analyzer and capacity planner to maximize the providers' revenue. Capacity planner prepares the machines and resources capacity according to the reports of a market analyzer.	Rejecting existing customers and not accepting heavy loaded SLAs may have long term bad impact on business [94].
Tevi Yombame Lawson [124]	They proposed an On-Off model for servers to save power for revenue maximization. Only limited PMs are turned on to meet the customers' requirements. Power consumption minimizes the costs.	Switching off some servers increases the workload on other servers. This may affect the performance of the services [125].
Hou Deng [103]	A solution was proposed for distance users issue to hire geographical resources from university or other institutions to process the local data. This minimizes the cost of the provider by decreasing the delay time.	The challenges with this approach are: getting geographical resources and the SLAs managing as per each region [126].
Amit et al. [102]	An approach was designed to run some of the resources on the cloud and some resources on the mobile devices. They proposed an adaptive service offloading scheme for Mobile Edge Computing (MEC) to maximize the total revenue.	A challenge with this scheme is migration live data between edge devices and cloud server may decrease the performance. It may also create security issues [127].
Yuan et al [118]	Previous work focuses on admission control but they proposed Profit Maximization Algorithm (PMA) with delay tolerance. The workload was scheduled from private cloud to hybrid cloud.	How to handle these customers with limited resources is still a big issue [126].

3.2.6 Cost and Prices Management

Prices have direct effects on customers. Some customers do not care about prices but want high performance, some customers do not care much about performance, they are not able to pay high prices. Therefore, there must be a framework, which will manage the prices according to customers' requirements. Table 3.9 summarizes cost and prices management related studies for achievement and limitations.

For better utilization and to increase the providers' revenue, nowadays a dynamic pricing mechanism is used for dynamic customers' requests. Ran and Xi [69] investigated a model to increase the revenue of the cloud computing provider. E.g. Amazon EC2 is also offering dynamic pricing since 2009. Dynamic pricing mechanism in IaaS causes many problems. They formulated a program which deals with such problems and handles infinite horizon cases. This study worked on resources scalability C3, and pricing C6 criteria.

Market analysis plays a good role to prepare the resources according to the incoming demands. Zhang and Boutaba [128] investigated a model to maximize the revenue of cloud providers. The market analyzer is used to analyze the market incoming request briefly. Then, they are using capacity planner which prepare the machines and resources capacity according to the reports of a market analyzer. This model is using both price mechanisms, dynamic and static. Different algorithms are used to predict the situation to use suitable prices mechanism technique which would be suitable for certain situations. This study discussed the cost and prices management C6 criterion.

Admission control is also discussed in cloud literature to maximize the providers' revenue. Toosi et al. [129] worked on optimizing admission decisions to accept only those contracts whose revenue is higher. In the proposed model three types of pricing mechanisms are used to maximize provider revenue in limited resources availability. These pricing plans are spot market, on-demand, pay as you go and reservation. Two types of algorithms are used in this model. Reservation contract is applied first and then the remaining capacity is utilized using spot instances. Revenue is earned from the upfront reservation, revenue from reserved, on-demand and spot instances respectively. Live reservation and running on-demand SLAs are kept within the provider capacity so that to control SLAs violations. With customer classification and capacity planner, this study achieved C4 and C5 criteria.

To *optimize the resources utilization with prices*, Chi et al. [130] proposed profit maximization using pricing methodology in cloud infrastructure. They worked on efficient resources utilization and pricing models to increase the number of customers. As customers' requests are accepted, they can easily and fairly be fulfilled. Higher pricing is used for those customers, whose jobs are difficult to fulfil. Two steps are used for pricing calculation, unit price redistribution and revenue redistribution. This study achieved resources utilization and management C5 and cost and prices C6 criteria.

Cloud cost was discussed by Zhou et al. [24]. They worked on cost optimization in IaaS clouds. Dyna, a scheduling system was developed, to minimize the monetary cost. A (*) based search transition is used

in this framework to search best price VMs. Finding the best price VMs maximizes revenue. This study achieved the cost and prices C6 criterion. Xu and Li [131] worked on *hybrid cost* and priority-based scheduling in the cloud environment. In this framework, they proposed a new hybrid economic algorithm which takes both the cost and priority scheduling to maximize the resources utilization. With scheduling and prices management they achieved C2 and C6 criteria.

As profit is directly affected by costs, Zhao et al. [132] tried to minimize the cost of the resources to maximize the provider profit. Furthermore, they worked on individually fulfilling the objectives. Their objective A is to minimize the cost, objective B is to start the queries execution at the earliest time and objective C is to combine objective A and B. With cost, SLA and resources management characteristics, this study achieved C5 and C6 criteria.

About 80 percent of all the power of data centers are consumed by the server. Power consumption is the major consumer of cloud revenue. Tevi Yombame Lawson [124] proposed an economic framework for resources management. This proposed framework minimizes the usage of power. They proposed an On-Off model for servers to save power and to maximize profit. The total power consumed by the Data Centers is $\propto * p$, where $\propto$ is the total CPU cores and P is the power consumed during the extreme time. Power consumption reduction, minimizes the costs. With cost minimization, this study achieved C6 criterion.

Tang and Chen [133] worked on *pricing and capacity planning*. They discussed two types of models, monopoly IaaS providers market, and multiple IaaS provider market. The optimal solution is searched in dynamic and static pricing for profit maximization. With prices and capacity planning properties, this study achieved the resources scheduling and management C5 and cost and prices management C6 criteria.

Mehiar Dabbagh and Rayes [38] tried to answer two main questions, i) where to place the incoming workload? and ii) how many resources should be allocated to this workload? This decision matters much in profit maximization because wrong placement wastes the resources and delays the tasks. A challenge with this framework is that running too many workloads only on a single machine may decrease the performance which leads to penalties. With resources and cost management, this study achieved C5 and C6 criteria.

Sharing common resources among VMs reduces the cost. To advance this idea, Rampersaud and State [134] worked on Sharing Aware Virtual Machine Revenue Maximization (SAVMRM) problem. They used a greedy algorithm to share the common memory and resources among the VMs hosted on one physical machine. This algorithm result shows a great deal toward revenue maximization. Sharing resources among different VMs may cause security and performance issues. With cost management discussion, this study achieved C6 criterion.

Authors in [135], worked on the novel revenue optimization model to address the operation and maintenance cost of the cloud servers. Authors used an algorithmic and analytic approach to solve the issues of optimal utilization of the resources. These algorithms minimize the power and operational cost to maximize the

profit. With cost management discussion, this study achieved C6 criterion.

Lower prices attract customers, however, it also creates performance issues. The above studies investigated how to optimize the cost, prices and performance, however, this needs further investigation to optimally determine these parameters.

Table 3.9: Contributions and limitations of cost and prices management related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Toosi et al. [129]	Optimal capacity was utilized to maximize the providers' revenue. Only those SLAs are accepted, whose revenue is higher. For customers' attraction, they used different pricing schemes.	Those customers are rejected, whose revenue is lower. Rejection of customers has a very bad impact on business [94].
Mehiar Dabbagh and Rayes [38]	They worked on cloud profit's maximization by efficient resources allocation, costing and pricing. This maximizes the utilization of a single physical machine instead of running many physical machines for the work having a capacity of a single physical machine.	A challenge with this framework is that running too many workloads only on a single machine may decrease the performance which leads to penalties [136] .
Rampersaud and State [134]	They used a greedy algorithm to share the common memory and resources among the VMs hosted on one physical machine. Sharing common resources among VMs reduces the cost.	Sharing resources among different VMs may cause security and performance issues [26] .
Snehanshu Saha and Roy [135]	They worked on a novel revenue optimization model to address the operation and maintenance cost of the cloud servers. They used an algorithmic and analytical approach to solve the issues of optimal utilization of the resources. These algorithms and analysis minimize the power and operational cost to maximize the profit.	The challenge with this framework is that performance increases the service cost [83].

Papers and Authors	Major Contributions	Limitations
Hong Xu [137]	Dynamic prices were utilized to attract more customers. Resources utilization increases providers' revenue.	More customers require salable resources. No solution was discussed for penalties and heavy loaded SLAs [138].

3.2.7 Advertisements and Auctions

Advertisement attracts more customers and obviously, it means more utilization. Overutilization is another technique to increase the underutilized resources' utilization, however, an issue is that it may lead to SLA violation. This may decrease performance and also customer satisfaction. Table 3.10 summarizes advertisement and auction related studies for achievement and limitations.

Over-commitment of resources is a complex decision. This increases the resources utilization but in case of risk miss calculation, it also increases the SLA violation. Dabbagh et al. [139] worked on resources utilization through cloud resources over-commitment. They used over commitment for minimizing Physical Machine (PM) overload and minimizing the number of VMs. In over-commitment, instead of initializing new VMs and to migrate the overloaded resource, simply resources are transferred to the PM. This saves VMs migration and initialization costs. The proposed framework uses different types of predictor such as VM utilization predictor and over- loaded predictor to increase resources utilization. With cost management and resources overutilization, this study achieved C6 and C7 criteria.

Metwally and Ahmed [140] worked on IaaS *resources allocation*. The main problem with cloud service providers is to handle a large number of requests of IaaS customers. Authors proposed Integer Linear programming technique and a mathematical programming model to find the optimal solution. They used large scale optimization tools and column generation formulation to allocate resources in a large data-centre. They worked on resources over-utilization C7 criterion.

Auction can attract customers. Samimi and Mukhtar [141] proposed combinatorial double auction model for revenue maximization. They extended already two models for double auction. The proposed model uses different phases and algorithms to maximize provider revenue. In the first phase, the cloud provider advertises its resources to the Cloud Information System (CIS). Every broker gets information from the CIS. The second phase generates the resources bundles according to user requirement; thereafter, auctioneers are informed. In the fourth phase winners and losers are determined. In the fifth phase resources are allocated to the winners. In the sixth phase, the payable amount is determined by the price model. With these properties, this study achieved resources auctioned C7 criterion. Hammoudi et al. [142] worked on *load balancing* in

cloud computing. With load balancing and parallel processing, large tasks are completed within a short time limit. They implemented this framework in the JADE platform. With load balancing characteristics, this study achieved C7 criterion.

The above investigation shows that advertisement increases the numbers of customers. People also take interest in the auction and it attracts more customers. More customer may overutilize the resources which may lead to SLA violation. These complexities need further research and exploration.

Table 3.10: Contributions and limitations of advertisement and auction related studies to maximize the providers' revenue in IaaS cloud

Papers and Authors	Major Contributions	Limitations
Zhao et al. [132]	This study investigated individually fulfilling of the objectives. Their objective A was to minimize the cost, objective B was to start the queries execution at the earliest time and objective C was to combine objective A and B. They proposed profit optimization algorithm.	They did not discuss how to entertain heavy loaded SLAs, and what to do in extreme utilization [94].
Hong Zhang and Liu [143]	They worked on revenue maximization by online auction for heterogeneous users' demands. Their approach works on two main functions. The first one is the payment function and the second one is the resource's allocation rule. The payment function works on the request allocation result and submission time. The allocation rule tries to maximize the bidder's utility.	The challenge with this is that customers do not trust online auctions[144].

3.3 Summary and Conclusion

This chapter described the related mechanisms and literature to IaaS cloud revenue maximization. We explored and analyzed the literature and classified it according to proposed parameters, as discussed in chapter 2. We reviewed and analyzed related findings to every parameter. This helped us to identify the literature gaps and future directions. We analyzed, how the performance, penalties, SLA management, resources utilization, resources scheduling, customer satisfaction and auction affect the revenue. We further

investigated these issues in the light of this literature to efficiently mange the resources and workload for better performance and revenue. Table 3.11 covers the year wise approaches and summarizes the relates studies. This chapter shows that massive research has been done in this field, however, some of the gaps still need further investigations. Dynamic resources provision with limited resources is till an open question to be investigated. Customers' satisfaction and resources scalbility still needs to be explore further.

Table 3.11: Year and approach wise summary of literature

Papers and Authors	Pub. Year	Approach used	Area
Hong Xu [137]	2012	Dynamic cloud pricing	C6
Qi Zhang [123]	2012	Dynamic resources allocation	C5
Wu and Buyya [113]	2012	SLA management	C2
Hong Zhang and Liu [143]	2013	Online auction	C7
Rongdong Hu [145]	2014	Resources provisioning	C5
[illegible]ndhini and Mohana [108]	2015	Client classification	C4
Adil [illegible]rouf and Haqiq [95]	2015	Novel penalty model	C2
Kundu et al. [71]	2015	Resource management framework	C1
Toosi et al. [129]	2015	Admission control for reservation contracts	C6
Ha[illegible] Zeghlache [99]	2015	Cloud federation	C5
Snehanshu Saha and Roy [135]	2015	Cost management	C6
Tevi Yombame Lawson [124]	2016	Power consumption minimization	C6
Zhao et al. [132]	2016	Optimization scheduling algorithm	C5
Mei et al. [111]	2017	Customer satisfaction	C4
Gao et al. [98]	2018	Transcoding in clouds	C1
Mehiar Dabbagh and Rayes [38]	2018	Price heterogeneity	C6
Rampersaud and State [134]	2019	Greedy approximation algorithm	C5
Hou Deng [103]	2019	Resources scalability	C3
Hong Zhang and Liu [146]	2019	Resources scalability	C3
Snehanshu Saha and Roy [102]	2019	Services performance	C3

Chapter 4

Methodology

This thesis explores the research problems related to the following research question.

> *With limited resources availability and heterogeneous customers' requests, How to maximize the providers' revenue and performance by minimizing SLA violation and customers' dissatisfaction in cloud computing?*

This is an *experimental research* which needs cloud lab for big data processing. Formation of cloud labs requires huge capital and expertise. A suitable alternative for cloud lab is *cloud simulators*. Several cloud simulators are used for different purposes. *Green cloud* is specifically used to simulate energy consumption in cloud projects [147]. Due to the broad functionalities of the proposed work, this is not suitable to help in our proposed methodology. *ICanCloud* was specially designed to calculate the cost and performance of network projects. However, as this research work covers a wide set of functionalists, ICanCloud would not be a better choice [148]. *Cloud analyst* is used to simulate the real environment of the databases. Cloud analyst provides more functionalists than Green Cloud and ICanCloud, further it also support GUI, which make it easy to use. We used it in part of our experimentation. *CloudSim* is a Java command line simulator which is widely used for simulating cloud environments [149]. This facilitates the virtual creation of servers, data centres and virtual machines. Cloud implementation and different scheduling algorithms can be implemented to plan different cloudlets on different virtual machines. The use of CloudSim in the proposed framework is due to its versatility. It can be used to create virtual machines, cloudlets, data centres, and migration scenarios. Cloud implementation and different scheduling algorithms can be implemented to schedule different cloudlets on different virtual machines. CloudSim was developed by Raj Kumar Buyya. He is affiliated with CLOUDS Lab, School of Computer Science and Information Systems, University of Melbourne, Australia [149].

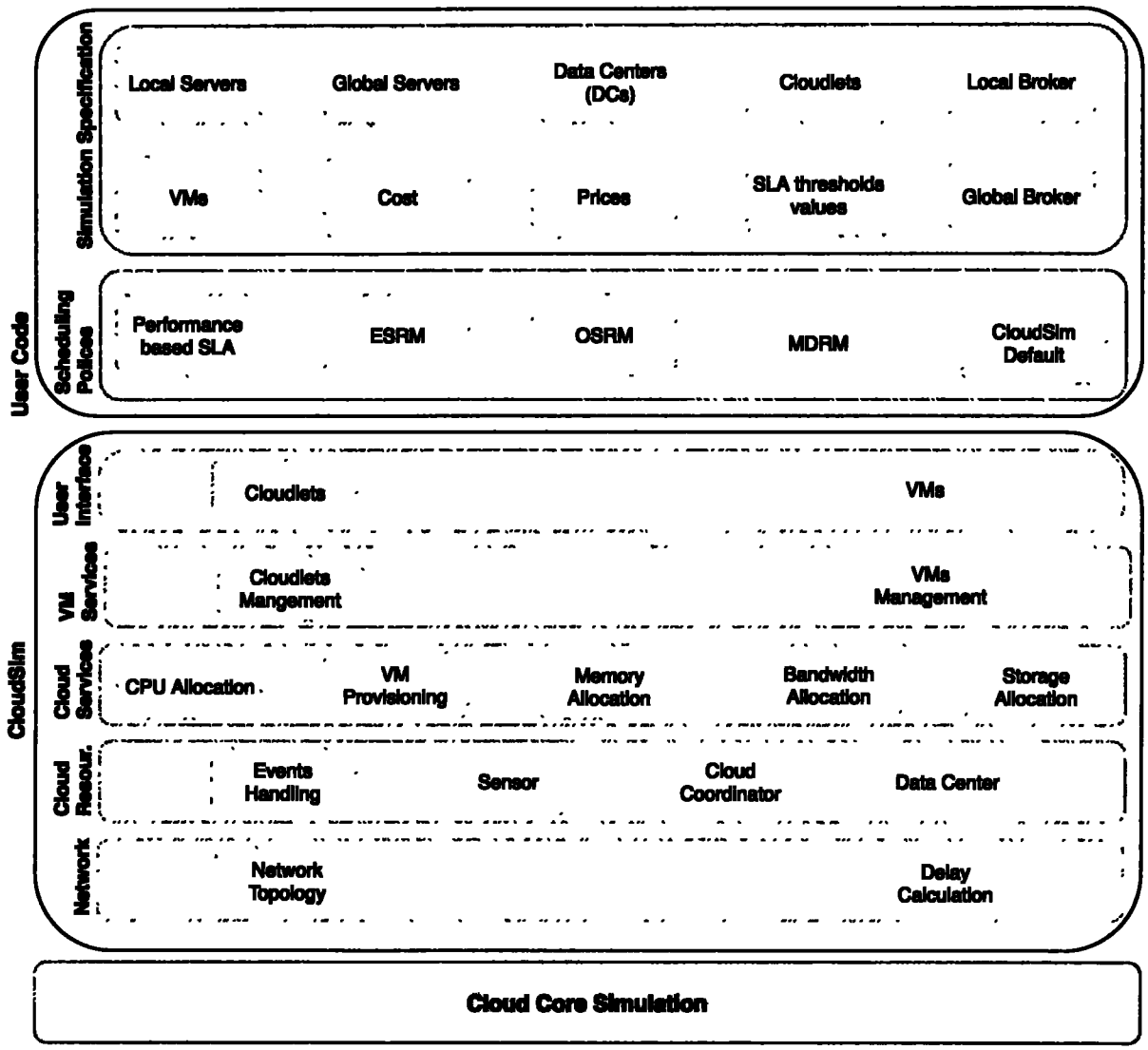


Figure 4.1: Structure of the CloudSim simulator

We extended the CloudSim simulator to evaluate the efficiency of the proposed model. Figure 4.1 shows the structure of CloudSim. Experimental setup was coded using java to evaluate the working of this model. In the first experimental setup, the important parameters which are considered and measured are, total number of *cloudlets*, total number of *VMs* on which these cloudlets are run, *response time*, *running time*, *waiting time*, *SLA violation*, *penalties*, *revenue generated* and *profit earned*. These parameters were calculated on both, running these resources on local servers and global servers.

4.1 System Model

This section discusses the general working of the proposed model. Algorithms are used to generate coudlets, Data Centers (DCs) and Virtual Machines (VMs). In CloudSim, cloudlet is used as workload to run it on VMs. In proposed simulation, the length of cloudlet is 800 MB, file size is 30000 MB, and out put size is 30 MB. VMs present the real VMs to run the cloudlets. In proposed simulation, Random Access Memory (RAM) is 2000 MB, MIPS are 250 MB, bandwidth is 1000 MBPS, and number of processors are 2. Where MIPS is memory instruction per second. DCs represent the data-centers to host the VMs. In proposed simulation, the architecture of datacenter is "x86", operating system is "Linux", vmm is "Xen", computing cost per second is 0.0040, cost per memory is 0.0030, cost per storage is .0035, and cost per bandwidth is .002. Two types of brokers are used in this model. Local brokers run the resources on local servers while global broker run the resources on global servers [150].

Three different cloud setups were created to measure the working of the proposed model. In the first setup, performance is measured for local and global servers, as shown in figure . In the second setup, simulation is created to check the working of PerSLA to minimize the SLA violations. In the third setup, delay and migration cost is calculated for different scenario.

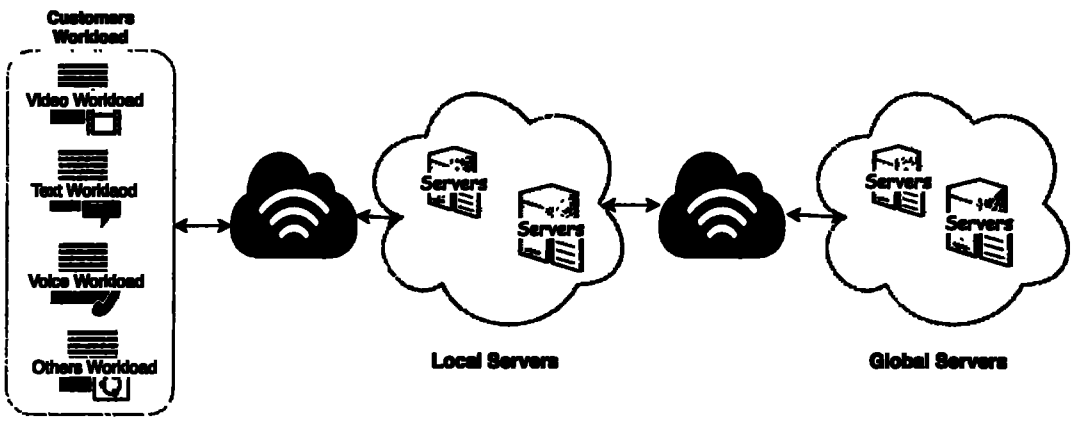


Figure 4.2: Running workload on local and global servers

- In the first simulation setup, *local brokers with local servers* were created as shown in figure 4.2. Number of VMs are initialized on local Physical Machines (PMs). Local servers take less running and transfer time comparatively to global servers. Cloudlet works as workload in CloudSim. Different numbers of cloudlets are generated using algorithms. In proposed simulation, the size of the single cloudlet is 30000 MB. The number of VMs, on which this workload is run, are gradually changed from 100 to 1500 [151]. Selecting this set of workload is to implement it reliably on CloudSim. Tak-

ing small simple would not give proper results while larger would be difficult to implement. During simulation, we observed running time, waiting time, SLA violation, penalties, revenue generation, and profit [150].

In the second part of the first setup, *global broker with global servers* were created. Global servers take more running and transfer time comparatively to local servers. Specification of cloudlets and VMs are the same as created for local broker for comparative study. Numbers of VMs are gradually increased from 100 to 1500 for same workload as in local broker setup. We observed, running time, waiting time, SLA violation, penalties, revenue generation, and profit during simulation.

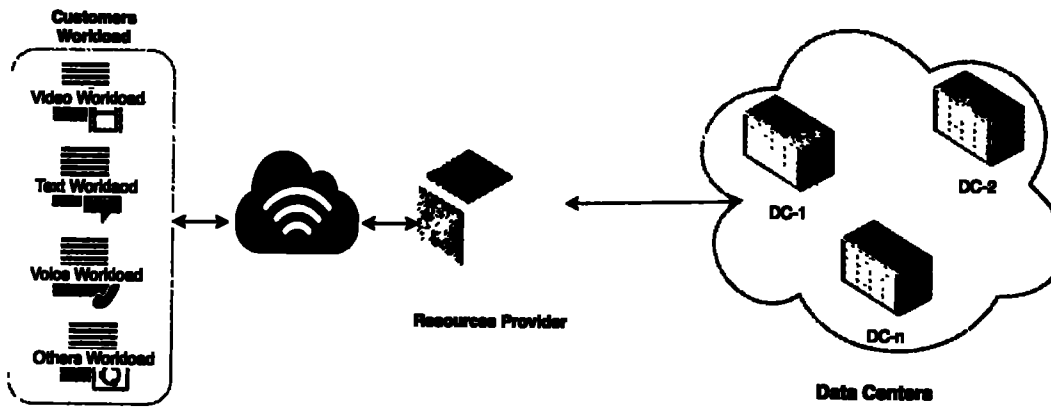


Figure 4.3: Running workload on different data centers

- In the second experimental setup, the CloudSim simulator was extended to evaluate the performance of Performance-based Service Level Agreement (PerSLA). Structure of the simulation is show in figure 4.3. We added new classes, "Penalties and Prices" to calculate and adjust performance and prices. Penalties are automatically calculated based on entries in the PerSLA class. Prices are also adjusted based on PerSLA entry and penalty classes. In the proposed scenario, 1,700 virtual machines were created as resources and 2,000 cloudlets as client workload [150, 151].
- In the third experimental setup, as shown in figure 4.4, Cloud Analyst was extended to evaluate the performance of efficient resources scheduling on external resources approach. For this, several CSPs have been created around the world on all continents. To calculate the delay, running time, and transfer cost, three different scenarios were created with different characteristics. In the first scenario, primary providers are required to hire resources from a single region. In the second scenario, providers hire local resources but there is no care of delay time and migration cost. In the third scenario, the proposed approach "Efficient Resources Scheduling on External Resources" is used to migrate the resources to external resources.

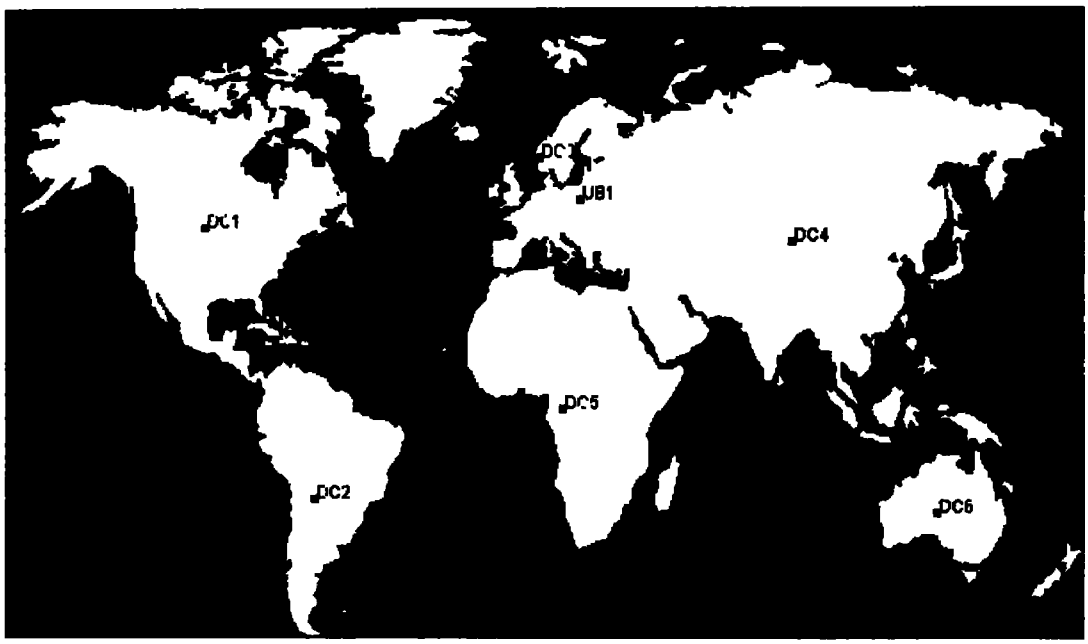


Figure 4.4: Migrating workload to different regions

4.2 Parameters

In the proposed model, running time, waiting time, SLA violations, penalties, revenue and profit is calculated. For *running time* calculation, CloudSim calculate the total time taken to fully execute the cloudlet. For *waiting time* calculation, CloudSim calculate the total time of cloudlet, it spent in waiting queue to access VM. To calculate the *SLA violations*, the above discussed performance parameters are measured against the agreed threshold values. Here in this model, three threshold values are introduced for running time, waiting time and availability. As the performance go down from these values, SLA violation accrues. The proposed model measure SLA violations in *three different layers*. In the proposed simulation, every parameter has its own weightage for penalty calculation. Penalties are calculated according to the intensity, time and parameter. These are calculated according to the proposed formula and deducted from the prices. In some situation, prices are reduced according to the SLA violations.

Revenue is generated from *demand customers, reserved customer, negotiated customer* and *external resources*. All the revenue are added to calculate the total revenue.

4.3 Proposed Scenarios

This section discusses the different experimental scenario to handle the research challenges discussed in chapter one.

1. How efficiently we utilize the provider resources?

Algorithms, *Efficient Scheduling for Revenue Maximization (ESRM)*, receive customers' tasks. ESRM decides where the incoming workload will be executed. If the execution and latency of the workload increase compared to the agreed threshold values, the workload is migrated to underutilized sources. If incoming requests exceed the supplier's available scale, resources are migrated to external providers. ESRM saves providers from fines. This algorithm also brings the workload back to the provider's resources as space becomes available. This minimizes external payments and also increases the use of provider sources.

2. How we control the resources scalability issue?

To solve problems with the scalability of resources, external sources are hired by an external supplier. In the proposed scenario, the cloud environment is created and managed by local and global brokers. The local broker carries out the workload on local sources, while the global broker carries it on worldwide resources. The same number of virtual machines are initialized on local and global servers. The same number of cloudlets are run on both servers to calculate the results of the simulation. The proposed algorithms are used to generate the number of cloudlets with a file size of 300,000 MB and to execute them on both servers. The simulation is performed multiple times, changing virtual machines from 100 to 1500 and cloudlets from 100 to 1500. During simulation, we observed the running time, waiting time, SLA violation, penalties and revenue generation by both servers to calculate the performance issues on transferring workload from local servers to global servers. The global servers increases the scalability of local servers. As the workload increased form local resources, workload is transferred towards the global servers. These external resources increases the providers resources scalability. Due to this, SLAs are not violated and heavy workloads are not rejected.

3. How we create a good SLA for customers satisfaction?

To handle the performance, prices, penalties and revenue issue, a clear cut SLA is needed. We proposed *Performance based Service Level (PerSLA)* to handle SLA related issues. PerSLA ensure the performance and suitable prices of the services. Clearly stated SLAs, with understandable links to the customer's business interests, are the key to help suppliers to provide the best service for customer requirements and maximizing customer satisfaction.

4. How to optimize prices and pricing policies to earn maximum revenue?

Joint prices are used in this model for customers' satisfaction and efficient use of resources. Fixed rates are used for high-quality reserved customers. Demand based prices are used by dynamic customers who are looking for high performance. Negotiated-based prices are used for underutilized resources. In negotiated-

based, prices are negotiated between customers and providers. Negotiated-based prices are used for under-utilized resources. As the proposed framework uses external resources, therefore, they are paid the agreed prices.

Joint prices are offered to attract more customers to the cloud business. Customers decide according to their scenario. Somewhere, they pay high but do not support lower performance. Somewhere, they may negotiate performance but cannot pay high prices. Furthermore, cloud resources are not storable. They are wasted if not utilized on time. Therefore, negotiated prices are best option to overcome the issue of underutilization. Some revenue is generated instead of complete wastage.

5. How we optimize the SlA violations and performance?

SLA violation is the main cause of penalties. Penalties are applied to decrease the prices during resources downtime or in the direct sanctions. In this model prices are reduced for downtime. Three penalty layer structure is used in this framework. Three threshold values are used to apply penalties on violation of SLA. It increases the customer retention and satisfaction. Further, it saves the provider from SLA violations.

4.4 Summary and Conclusion

Creating cloud lab requires large capital and high expertise. In academia, the easy solution for cloud lab is cloud simulators. CloudSim is widely used in academia to simulate the big data environment. Cloud Analyst is also a part of the CloudSim project. To evaluate the efficiency of the proposed approaches, we extended CloudSim and Cloud Analyst. The whole simulation is divided into three parts. In the first part, hiring external resources and implementing joint prices was simulated. In the second setup Performance based Service Level Agreement (PerSLA) was implemented. In the third simulation, Cloud Analyst was used to simulate the efficient migration to external resources. All the simulation setup efficiently performed in various environments.

Chapter 5

Revenue Maximization by Hiring External Resources

Revenue is the prime focus of every business. Cloud providers want to maximize revenue and manage their business up to mark [152]. For revenue maximization, we consider performance, costs, penalties and revenue. These parameters show that increase in performance, maximizes revenue and reduces penalties. However, the performance is also proportional to the costs and prices, which is inversely proportional to the profit in terms of cost [107].

Underutilization of cloud resources is a critical issue as it reduces providers' revenue [22, 100]. Sometimes, customers reserve certain resources but later on do not use them. As a result, these resources are wasted. Underutilized resources can be delivered based on negotiation-based pricing. Negotiation-based pricing generates revenue for underutilized resources instead of wasting them. Also, no penalties are imposed for the SLA breaches [26]. The proposed framework aims to reduce the waste of resources by offering negotiation-based pricing. Overutilization annoys the providers in different ways. They have to refuse customers having massive workloads because of limited resources. This dissatisfies customers. Also, the rejection of massive workloads deprives providers from higher revenue generation [38]. Customer satisfaction is measured by the degree of satisfaction of their needs. It is more economical to retain existing customers than to attract new ones. Customers satisfaction services improve customer value, increasing the long-term revenue of the company [14, 27].

Table 5.1: Symbols used in formulation

Symbol	Definition	Symbol	Definition
χ_{hu}	Cost of resources per unit price	χ_{ser}	Server costs
χ_{sec}	Security charges	χ_m^{ij}	External migration costs
χ_{mint}	Maintenance costs	χ_{mon}	Monitoring charges
χ_n	Network cost	χ_m^{ii}	Internal migration costs
χ_{cost}^{ii}	Internal costs	χ_{cost}^{ij}	External costs
VM_f	Free virtual machines	υ_h	Unit hired
ρ_r	Reserved prices	ρ_{db}	Demand based pricing
ρ_n	Negotiation based pricing	P	Power consumption
Rev_{ii}	Revenue from internal resources	Rev_{ij}	Revenue from external resources
μ	Number of customers	V_n	Number of violated SLA
α	Customer attention costs	Tx	Taxes
$\jmath$	Customer retention costs	η	Penalties
κ	Constant	ω	Human resources costs
Δ	Markup margin	σ	Discount on reservation
Φ	Upfront reservation	$Prov_{max}$	Provider having maximum revenue
∂_{ii}	Internal resources	UT	User task
∂_{ij}	External resources	VM_{max}	VM having maximum revenue
CR	Customers request	VM_u	Virtual machine utilized
$Prof$	Profit	γ	Minimum guaranteed revenue
Per	Performance	υ_{ii}	Resources utilization
QoS	Quality of services	SS	Scalability of services
ET	Execution time	Eff	Efficiency of services
Θ	Monitoring charges	Rel	Reliability of Services

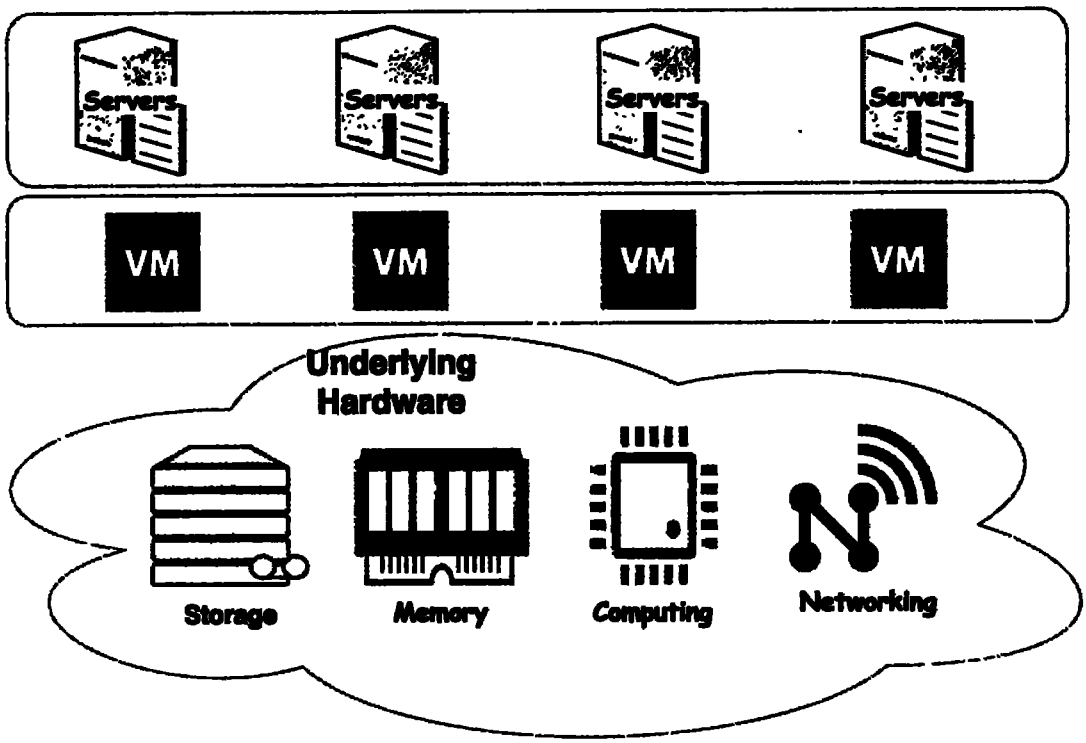


Figure 5.1: Infrastructure as a Service

Pricing plays a major role in customer satisfaction and attraction. Different price models are used in cloud computing [36]. In demand-based pricing, customers are charged per unit consumed. In reservation-based pricing, customers are billed based on their subscription level. Users also benefit from a discount on early booking. In the proposed framework, a combination of different prices is used for customers satisfaction and attraction [103, 119].

In case of SLA violation, sanctions are imposed on the provider. It plays a vital role in revenue maximization. Most of the providers' revenue is wasted in penalties [90]. To reduce rejection of service level agreements and customer dissatisfaction, the proposed model proposes the hiring of external resources. External resources optimize the scale of the providers' resources. With scaleable resources, massive workloads are accepted for execution, also it do not overload the system [34]. Insourcing and outsourcing techniques are applied to under-utilized and overutilized resources [32, 122].

The federated cloud concept was introduced to handle overloaded systems [30]. However, the federated cloud imposes restrictions to hire the resources from specific providers, registered with the federation. Even though such resources can be easily and moderately available from suppliers outside the federation. This results in a kind of monopoly, in which a provider is forced to hire resources from specific external providers.

This study intends to design an approach where a service provider, if overloaded, or has more customers than the available resources, can serve or hire external resources anywhere it is available conveniently and at low cost.

This chapter covers effective scheduling, scalability, optimized pricing, as well as provider optimized performance to maximize revenue. The most important factors for customer satisfaction are the prices and performance of services [111]. In the limited resources scenario, external resources are hired to extend the provider resources scalability, which optimizes pricing and performance. Local and global migration techniques are used to avoid over-utilization of internal resources. The table 5.1 shows the notation used in this chapter.

Contributions of this chapter are listed as follows

- The proposed approach hires resources from external Cloud Service Providers (CSPs) to extend the providers' business. As a federated cloud, providers are not forced to buy from specific allies.
- To optimize the performance and revenue, algorithms are proposed for local and global migration strategies to save internal resources from under and over utilization.
- The proposed approach offers different prices. Algorithms and formulas are used to select prices according to incoming customers' request. Dynamic prices attract more customers.

With limited resources, it is very difficult for cloud service providers to respond to the dynamic and massive demands of the customers. Massive use or denial of any service level agreement may result in penalties that play a critical role in the cloud industry. Over utilization of resources, instead of maximizing revenue, can result in lower revenues due to breaches of SLAs as providers are subject to penalties. Various studies have been conducted to study these problems, but improvements are still needed. In this chapter, we have proposed a model to solve the problem of resources scalability and service level agreement violations by recruiting resources from external suppliers at low prices. The figure 5.2 shows the detailed structure of the proposed model.

5.1 System Model

In this section, we discussed cloud prices, customer satisfaction, hiring external resources and how to formulate them to maximize revenue.

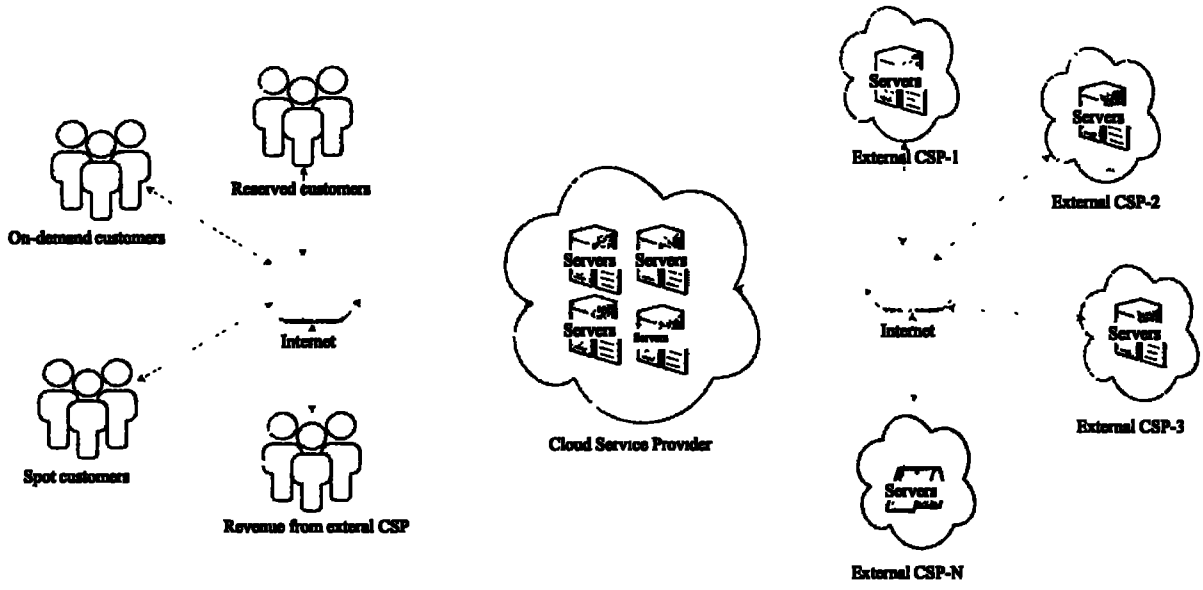


Figure 5.2: Proposed system model for hiring external resources

5.1.1 Cloud Pricing

Price models play a big role in revenue generation. The cloud market uses different types of price models. In the reservation price plan, customers reserve resources for a specific period, such as a month or a year. Resources are given to customers with a substantial discount. Customers pay the upfront registration costs. With on-demand prices, customers are billed as peruse. In this price system, prices are higher, but providers are charged for the SLA violations. In negotiated pricing, prices are negotiated between customers and providers. Negotiated prices are used for under-utilized resources.

As the proposed framework uses resources from external providers, therefore, they are paid agreed prices. As discussed, external resources are hired in case of extreme utilization to scale the limited resources and to manage the massive SLAs. The combination of all prices is used according to incoming customers' request. In this framework, internal and external migration costs are computed, as both the provider and consumer suffer a lot from this. Security and safety are important but they also increase the costs [152].

For provider, the basic cost is calculated as:

$$\chi_t = \chi_{hu} + \omega + \chi_n + \chi_m^{ii} + P + \chi_{sec} + \kappa \tag{5.1}$$

Here χ_t shows the total cost per unit use. According to the equation no 5.1, total cost (χ_t) of the resources

are the resources prices per unit time (χ_{hu}), human resources needed (ω), networking cost (χ_n), Internal migration (χ_m^{ii}), power consumption (P), security charges (χ_{sec}), and others related charges (κ).

Internal resources cost is calculated as:

$$\chi_{cost}^{ii} = \chi_t + \kappa \tag{5.2}$$

Where (χ_{cost}^{ii}) are the total internal cost

$$\eta \propto V_n \tag{5.3}$$

$$Per \propto Rel \tag{5.4}$$

The above computation and equations no 1.5 and 2.2 describes important composition in this framework. Number of penalties (η) are directly proportional to SLA violations (V_n) . SLA violations are inversely proportional to performance (Per) and execution time (Et). Performance (Per) is directly proportional to costs and reliability (Rel). In this section,these parameters are adjusted to improve revenue.

For the proposed framework, join prices are used. Fixed rates are used for high-quality, reserved customers. Demand-based prices are used by dynamic customers who are looking for high performance. negotiated-based prices are used for under-utilized resources. In the real cloud environment, this increases resource utilization and customer satisfaction.
Prices are calculated as:

$$\rho_r = \chi_{cost}^{ii} \times \upsilon_h \times t - \sigma + \Delta \tag{5.5}$$

In reservation based pricing (ρ_r), upfront reservation (σ) is deducted from prices because it is returned to the customer at the end of a successful business. After subtracting the upfront reservation, a total optimized margin (Δ) is added which is what the provider wants to earn.

$$\rho_{db} = \chi^{ii}_{cost} \times \upsilon_h \times t + \Delta \tag{5.6}$$

In demand based pricing (ρ_{db}), customers are charged for units consumed, however, they are not charged for upfront reservation.

5.1.2 Maximizing Providers' Resource Utilization

Utilization plays a leading role in the cloud computing market. If the use of resources increases to the maximum, the revenue will also increase. Negotiated-based prices are used for underutilized resources, to increase resources utilization.

$$Rev_{ii} \propto \rho \times \upsilon_{ii} \times \mu \tag{5.7}$$

Total revenue generated from internal resources (Rev_{ii}) is directly proportional to the product prices (ρ), internal resources utilization (υ_{ii}), and total number of customers (μ).

Usually, in cloud computing, resources are kept reserved. If customers reserve the resources but later on do not use them then these resources go underutilized and wasted. In the proposed framework, these resources are rented to other customers with the prior permission of buyer customer for higher revenue. This benefits both, provider as well as the buyer. Resource utilization (υ_{ii}) is denoted as:

$$\upsilon_{ii} = \frac{Running(\sum_i^n VM)}{Available(\sum_i^n VM)} \tag{5.8}$$

Where denominator shows the total available Virtual Machines (VM) and nominator shows the total running VMs.

The main reason behind negotiated pricing is that cloud resources are not store-able. Consequently, resources go wasted if not utilized. Therefore, it is better to get some revenue instead of complete wastage or to go into loss.

$$\rho_n = \chi + \Delta_{neg} \tag{5.9}$$

In negotiation based pricing (ρ_n), providers negotiate prices keeping margin (Δ_{neg}) in mind.

Where,

$$\Delta > \gamma \tag{5.10}$$

Equation no 5.9 attracts the low budget customers, which increase the resource utilization.

5.1.3 Hiring External Resources

To maximize revenue, the provider must improve customer satisfaction and minimize SLA breaches. The SLA violation occurs when a provider admits more customers than the available resources in the hope of improving its revenue. As a result, it is possible that the provider cannot provide the agreed services, which can lead to penalties. As a result, the provider wastes instead of increasing revenue by paying penalties.

With limited resources and maximum use, the major issue is that cloud service providers reject existing customers whose penalties are lower than the revenues of new customers. High-income SLAs are adopted, while low-income SLAs are rejected. As we know that:

$$\mu \propto SS \tag{5.11}$$

Number of customer (μ) is directly proportional to services scale-ability (SS).

$$Rev_{ii} \propto \mu \tag{5.12}$$

Revenue from internal resources are (Rev_{ii}) are directly proportional to the number of customers (μ).

Penalties negatively affect the cloud market. Usually cloud computing accepts massive workload but later on, they can not provide resources as per the agreement. As a result, they pay much of their revenue in penalties.

Similarly, equation 2.12 shows that total number of SLA violations (V_n) is inversely proportional to Service Scalability (SS), Quality of Service (QoS), and Efficiency of Service (Eff).

$$\eta \propto V_n \tag{5.13}$$

$$Prof \propto 1/\eta \tag{5.14}$$

Whereof penalties (η) are directly proportional to SLA violations (V_n) and total profit of the provider ($Prof$) is inversely proportional to penalties.

Rejecting customers, lower performance, and resources limitations lead to customer dissatisfaction. Generally, providers give little attention to customers' satisfaction and retention. Retention is easier than attracting new customers. Customer dissatisfaction means losing a lot of revenue.

$$CS \propto SS \times Eff \times QoS \tag{5.15}$$

Here (CS) shows the customers' satisfaction which is directly proportional to Service Scalability (SS), Quality of Service (QoS), and Efficiency of Service (Eff).

To minimize the SLA rejection and penalties of running SLAs, the proposed approach hires external resources. Hiring external resources maximizes Scale of Service (SS), Qualify of Service (QoS), and Efficiency (Eff) of services. Increasing these parameters, minimize penalties. Three types of parties take part in this framework. The first is the provider party, which provides cloud services. The second is the consumer party, which hires the resources from the provider. The third is the external cloud service provider. Monitoring services may be hired for smooth service delivery and customer satisfaction.

Providers hire resources from external providers to maximize the scale of services and to provide reliable services to consumers in extreme use. Two types of SLAs have been established. A service level agreement is signed between the provider and the consumer and the second between the service provider and the external cloud service provider. End users are charged based on their use of resources.

$$CS \propto SS \times Eff \times QoS \times \frac{1}{\rho} \tag{5.16}$$

Customer Satisfaction (CS) is inversely proportional to prices (ρ).

The equation 5.16 shows that limited scalability and high prices are the kay causes of SLA violation. The proposed approach tries to minimize this by hiring external resources to scale the business. Customers are permitted to select prices according to their choice which increases the customers' satisfaction.

External resources cost is calculated as:

$$\chi_{cost}^{ij} = \chi_{hu} \times t \times U_h + \chi_m^{ij} + \kappa \tag{5.17}$$

5.1.4 Revenue Maximization

From equations 5.5, 5.6 and 5.9 the total revenue in terms of prices is calculated as:

$$\rho_t = \rho_o + \rho_{db} + \rho_n \tag{5.18}$$

According to the equation 5.18, total incoming revenue (ρ_t) from prices is the sum of all prices. Revenue earned from reserved customers:

$$Rev_{res} = \sum_{i=0}^{n} (\rho_r s \mu_s i + \rho_r p \mu_p i + \rho_r m \mu_m i)\, t + \sum_{i=0}^{n} (\Delta_i + \phi_i) - \sum_{i=0}^{n} (\eta_{ri} + \sigma_i) \tag{5.19}$$

Where Rev_{res} denotes revenue earned from reserved customers, $\rho_r s$ shows the reserved prices for storage, $\rho_r p$ shows the reserved prices for processing power, $\rho_r m$ shows the reserved prices for memory, t shows the time, Δ_i shows the margin, ϕ_i shows the upfront registration, η_{ri} shows the penalties, and σ_i shows the discount.

For example, a customer C1 utilized storage of 5 GB, 1 VM as processing power , and memory of 2 GB for 12 hours where the prices for reserved customers (for one year) are: $\rho_r s$= 0.05, $\rho_r p$= 0.0192 , and $\rho_r m$= 0.002 [153]. The revenue generated from the above customer will be calculated as:

$Rev_{res} = (5 \times 0.05 + 12 \times 0.0192 + 2 \times 0.0192) \times 12 + (0.05284 + 0.010528) - (0 + 0.05284)$

$Rev_{res} = \$3.21408$

Revenue earned from demand based customers is calculated as:

$$Rev_{db} = \sum_{i=0}^{n} (\rho_{db}s\mu_s it + \rho_{db}p\mu_p it + \rho_r m\mu_m it) + \sum_{i=0}^{n} (\Delta_i) - \sum_{i=0}^{n} (\eta_{ri}) \quad (5.20)$$

Where (Rev_{db}) shows revenue earned from demand based customers. Equation 5.20 indicates that it is the sum of total prices (ρ_{dp}), utilization (μ_{db}), time of utilization (t), and margin for profit (Δ). Total penalties (η_{db}) are detected from the revenue.

For example, a customer C1 utilized storage of 5 GB, 1 VM as processing power , and memory of 2 GB for 12 hours where the prices for demand based customers are: $\rho_{db}s$= 0.05, $\rho_{db}p$=0.031 , and $\rho_{db}m$= 0.004 [153]. The revenue generated from the above customer will be calculated as:

$Rev_{db} = (5 \times 0.05 \times 12 + 12 \times 0.031 \times 12 + 2 \times 12 \times 0.004) + (0.0718) - (0)$

$Rev_{db} = \$3.82272$

Revenue earned from negotiated based customers are calculated as:

$$Rev_n = \sum_{i=0}^{n} (\rho_n s\mu_s it + \rho_n p\mu_p it + \rho_r m\mu_m it) + \sum (\Delta_i) \quad (5.21)$$

Where Rev_n means revenue earned from negotiated customers.

For example, a customer C1 utilized storage of 5 GB, 1 VM as processing power , and memory of 2 GB for 12 hours where the prices for demand based customers are: $\rho_n s$= 0.04, $\rho_n p$=0.025, and $\rho_n m$= 0.002 [153]. The revenue generated from the above customer will be calculated as:

$Rev_n = (5 \times 0.04 \times 12 + 12 \times 0.025 \times 12 + 2 \times 12 \times 0.002) + (0.0718)$

$Rev_n = \$2.6784$

Where,

$$\Delta > \psi$$

and

$$\rho_n = \chi_t$$

Total revenue earned from internal resources is:

$$Rev_{ii} = Rev_r + Rev_{db} + Rev_n \tag{5.22}$$

Total revenue earned from internal resources (Rev_{ii}) is sum of the revenue earned from reserved customers (Rev_r), revenue earned from demand based customers (Rev_{db}), and revenue earned from negotiated based customers (Rev_n).

As per the above examples, the total revenue earned from internal resources are:

$Rev_{ii} = 3.21408 + 3.82272 + 2.6784$

$Rev_{ii} = \$9.714$

The revenue earned from external resources is calculated as:

$$Rev_{ij} = \sum_{k=0}^{n} (\rho_e s \mu_s it + \rho_{ep} \mu_p it + \rho_r m \mu_m it) + \sum_{k=0}^{n} (\Delta_k) - \sum_{k=0}^{n} (ExCh_k) \tag{5.23}$$

Where $ExCh$ shows the external charges.

For example, the provider utilized external storage of 5 GB, 1 VM as processing power , and memory of 2 GB for 12 hours where the prices for demand based customers are: $\rho_n s$= 0.04, $\rho_n p$=0.025, and $\rho_n m$= 0.002 [153]. The revenue generated from the above customer will be calculated as:

$Rev_{ij} = (5 \times 0.04 \times 12 + 12 \times 0.0192 \times 12 + 2 \times 12 \times 0.002) + (0.013392) - (2.6784)$

$Rev_{ij} = \$0.013392$

Where,

$$\Delta > \psi$$

From equation 5.22 and 5.23 the total revenue earned by cloud providers is calculated as

$$Rev_t = Rev_{ii} + Rev_{ij} \tag{5.24}$$

Where Rev_t shows the total revenue generated by system. (Rev_{ii}) shows the revenue from local resources, (Rev_{ij}) shows the revenue from external resources. For above example $Rev_t = 9.714 + 0.0133$

$Rev_t = \$9.727$

$$\mu_t \propto 1/\rho \tag{5.25}$$

To set the prices is an issue for the cloud provider. Definitely, from equation 5.7, high rate of price maximize the revenue and on the other hand, equation 5.16, shows that it increase customers dissatisfaction, which is inversely proportional to revenue.

From equations 5.1, and 5.24, the total profit of provider is

$$Prof = (Rev_t) - (\chi_t + \partial + \beta + Tx + \chi_{min} + \Theta) \tag{5.26}$$

$Prof$ shows the total profit earned by the provider after deducting all cost (χ_t), customer attention cost (∂), customer retention cost (β), taxes (Tx), maintenance (χ_{min}), and monitoring charges (Θ).

5.1.5 Proposed Algorithms

The proposed pricing optimization and customer satisfaction framework maximizes the revenue. We proposed different algorithms which maximize the providers' revenue by customer satisfaction and efficient resource provision. Algorithm 1, Efficient Scheduling for Revenue Maximization (ESRM), takes free VMs (VM_f) and user tasks (UT) as input. Free (VM_f) are stored in ∂_{ii} as internal resources array list.

Experiments conducted in CloudSim show that under normal circumstances the waiting time remains less than 3 seconds. Therefore (if the thresholds are taken as 5s) if a virtual machine gives a response time of more than 5 seconds and there are no other sources to create virtual machines, the workload is migrated to external sources. Table 5.2 presents a detailed strategy for creating new virtual machines and migrating the workload to external resources.

Algorithm 1 ESRM decides where to run the workloads. It optimizes user tasks with virtual machines to improve revenue and performance. If the execution and response time of the workload exceeds the agreed threshold values, the workload is migrated to underutilized resources. If the incoming requests increase above the provider's available resources (VM_f), workloads are migrated to external providers. ESRM save providers from penalties. This algorithm also brings the workload back to the provider's resources as space becomes available. This minimizes the external payments and also increases the providers' resource utilization.The algorithm 1 takes $O(n)$ as running time on using sequential looping structure to match every

Table 5.2: Migration policy

SLA Parameter	Thr (φ_i)	Remarks
Response time (sec)	5 s	Creating new VMs and in the case of resources non availability migrate the workloads to external provider.
Execution time (sec)	7s	Creating new VMs and in the case of resources non availability migrate the workloads to external provider.
Storage Availability (%)	100 %	Creating new VMs and in the case of storage non availability migrate the workloads to external provider.
Memory Availability (%)	98 %	Creating new VMs and in the case of memory non availability migrate the workloads to external provider.

workload to every VM.

Algorithm 2, Price Optimization for Revenue Maximization (PORM), categorize the customers workload into on-demand (UT_{db}), reserved (UT_{res}) and negotiated (UT_{neg}) groups. The workloads from reserved and ondemand customers are accepted because the proposed framework uses external resources to retain heavy workload customers. Negotiation based customers are accepted only in the underutilization scenario. Incoming workloads are charged according to the customer type and equations number 5.5, 5.6 and 5.9. The algorithm 2 needs $O(c)$ as running time due to conditional operations and mathematical calculation.

Where UT_{db} represents the demand based user tasks, UT_{res} shows the reserved based user tasks and UT_n shows the negotiated based user tasks.

Algorithm 3, Optimizing Scheduling for Revenue Maximization (OSRM), receives the list of free VMs and users tasks (CR) which have to be run on these resources. OSRM look for the best combination of tasks and VMs for best pricing and performance. It compares the user tasks with available VMs and selects the VM for deployment which has a best fit for execution in terms of revenue and performance.The algorithm 3 takes $O(n)$ as running time on using sequential looping structure to match every workload with every VM for best VM selection.

Where VM_{max} is a VM having maximum revenue and performance.

Algorithm 4, Migration Decision for Revenue Maximization (MDRM), get requests for migration from ESRM algorithm. MDRM first looks the internal free VMs for best execution time and prices. If none of

Algorithm 1 Efficient Scheduling for Revenue Maximization

Input: Free VMs (VM_f) and Users' Tak (UT)

Output: Optimized Pair of VM_f and UT

1: $\partial_{ii}(=) VM_1, VM_2, VM_3, VM_4, ...VM_n$
2: $UT(=) User - task$
3: **if** $\partial_{ii} >= UT$ **then** ▷ If placing UT does not violate deadline given in Table 5.2
4: RESOURCE-OPTIMIZATION($VM[n], UT$);
5: $VM_{max} \leftarrow UT$
6: PRICING-OPTIMIZATION(UT_k) ▷ Calling algorithm II
7: **end if**
8: **if** $Deadline \geq Exetime || Deadline \geq ResTime$ **then**
9: LOCAL-MIGRATION(UT_i)
10: $VM_{max} \leftarrow UT$
11: PRICING-OPTIMIZATION(UT_k)
12: **end if**
13: **if** $UT \geq Threshold - \partial_{ii}$ **then**
14: GLOBAL-MIGRATION(UTk)
15: $Prov_{max} \leftarrow UT$
16: PRICING-OPTIMIZATION(UT_k)
17: **end if**
18: **if** $\partial_{ii} \geq UT_o$ **then**
19: Migrate the resources back
20: **end if**

the local VMs is capable to run the workload with the reliable performance, global providers are searched. External resources not only extend the resources scalability of a local provider but also increase its performance and revenue. These resources minimizes the SLA violation and overutilization. The algorithm 4 takes $O(n)$ as running time on using sequential looping structure searching optimized VMs and provider.

5.2 Performance Evaluation

Recent developments in cloud computing technology have attracted academia and the market. The formation of cloud computing laboratories for research experimentation requires considerable capital and expertise. A

Algorithm 2 Price Optimization for Revenue Maximization

Input: User Tasks

Output: Optimized Prices

if $UT_k = UT_{db}$ **then**

status = accepted

$\rho_{db} = \chi^{ii}_{cost} \times U_h \times t + \Delta$ ▷ Equation # 3

$\rho = \rho_{db}$

end if

if $UT_k = UT_{res}$ **then**

status = accepted

$\rho_r = \chi^{ii}_{cost} \times \upsilon_h \times t - \sigma + \Delta$ ▷ Equation # 4

$\rho = \rho_r$

end if

if $UT_k = UT_n$ **then**

if $VM_f >= threshold$ **then**

status= accepted

$\rho_n = \chi + \Delta_{neg}$ ▷ Equation # 7

$\rho = \rho_n$

end if

end if

good alternative is to use simulation tools such as CloudSim. CloudSim is a Java-based command-line simulator that is widely used for simulating cloud environments [93]. It facilitates the virtual creation of servers, data centres, virtual machines and many other elements related to cloud simulation. Cloud implementation and different algorithms can be implemented to schedule different cloudlets on different virtual machines [74].

5.2.1 Experimental Setup

The CloudSim simulator has been extended to evaluate the effectiveness of the proposed work. For this, new classes have been added for effective scheduling, pricing and generating revenue. In the proposed scenario, the cloud environment is created and managed by local and global brokers. A local broker runs the workloads on local sources, while a global broker runs it on external sources. A number of virtual machines

Algorithm 3 Optimizing Scheduling for Revenue Maximization

Input: Free VMs (VM_f) and Users' Taks (UT)

Output: Selecting best VM VM_{max}

```
1: RESOURCE-OPTIMIZATION(VM[n], UT);
2: VM_max ← 0
3: for int j = 1; j ≤ numbervm; j++ do
4:     Searching VM having optimum Execution time and Response time   ▷ Policy: table 5.2
5:     if VM[n] ≠ UT then
6:         if ∂_ii ≥ ReqSpace then
7:             space ← requiredspace   ▷ Space means the required processor, memory and storage
8:             CreateVM(space);
9:         end if
10:    end if
11:    return VM_max
12: end for
13: if Scheduling not successful then
14:    Restart from step 1
15: end if
```

are initialized on local and global servers. The same number of cloudlets is run on both servers to calculate the results of the simulation.

It is assumed that the capacity of the local data centre is limited and unscalable. The results of the simulation are calculated by changing the cloudlets from 100 to 1500. Each cloudlet has a file size of 300,000 MB. Due to limited resources, the number of virtual machines remains constant at 100. Each virtual machine has 3 GB of RAM and a 3.2 GHz processor. For the global server, algorithm 1 generates the number of cloudlets with a file size of 300,000 MB and runs it on a global server. The simulation runs multiple times, modifying the virtual machines from 100 to 1500 and the cloudlets from 100 to 1500 [34]. Each virtual machine has 3 GB of RAM and a 3.2 GHz processor. During the simulation, run time, wait time, SLA violation, penalties, and revenue generation for both servers are changed. Runtime is the total time taken to run the customer workload. It is an important parameter of the performance. The running time is represented as

Algorithm 4 Migration Decision for Revenue Maximization

Input: Free VMs (VM_f), Users' Taks (UT) and Prov

Output: Optimize Pair (VM,UT) AND (Prov,UT)

1: **if** Local migration request receives **then**
2: $VM_{max} \leftarrow 0$
3: $VM_{[n]}(=) VM_1, VM_2, VM_3, VM_4, ...VM_n$
4: RESOURCE-OPTIMIZATION($VM_{[n]}, UT$);
5: $VM_{max} \leftarrow UT_j$
6: **if** Scheduling not successful **then**
7: Restart from step 1
8: **end if**
9: **end if**
10: **if** Global migration request receives **then**
11: $prov_{max} \leftarrow 0$
12: $Prov_{[n]}(=) Prov_1, Prov2, Prov3, Prov4, ...Prov_n$
13: Searching provider using algorithm no 7.1
14: **return** $Prov_{max}$
15: **if** Scheduling not successful **then**
16: Restart from step 1
17: **end if**
18: **end if**

$$\tau_{run} = \tau_f - \tau_i \tag{5.27}$$

Where τ_{run} shows the total running time, τ_i shows the request initialization time and τ_f shows the workload finalization time. Waiting time shows the total time in queue waiting for execution. Waiting time is another important parameter of performance. It is represented as

$$\tau_{wait} = \tau_i - \tau_{sub} \tag{5.28}$$

Where τ_{wait} shows the total waiting time, τ_{sub} shows the request submission time and τ_i shows the workload initialization time.

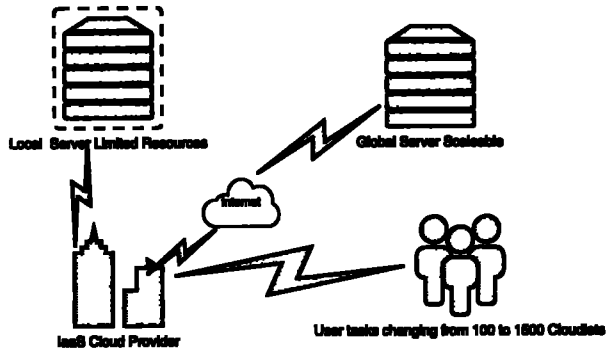


Figure 5.3: Simulation structure

5.2.2 Evaluation Results

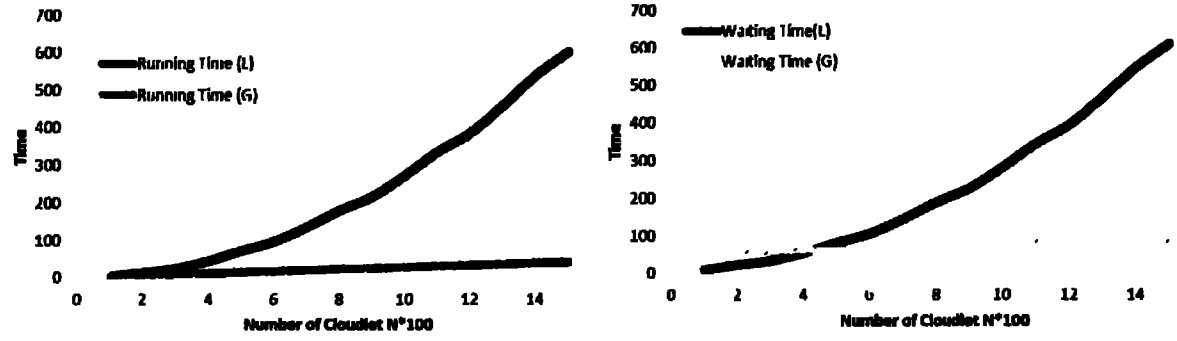
In this section, various experiments are conducted to demonstrate the contribution of the proposed works of customer satisfaction and revenue maximization. Depending on the experimental configuration, the results are divided into three parts: the results of the local server, the results of the global server and their comparative analysis.

Here in table 5.3 and 5.4, CL shows the number of cloudlets, VMs shows the number of Virtual Machines, RT shows the number of running time in seconds, WT shows waiting time in seconds, SV shows the number of SLA violations, Rev shows the total revenue earned and $Prof$ shows the total profit earned.

It is assumed that the capacity of the local data centre is limited and nonscalable. Simulation results are calculated by changing cloudlets from 100 to 1500. Every cloudlet has 300000 MB of file size. Because of limited resources, numbers of VMs remain constant at 100. Every VM has 3 GB of RAM and 3.2 GHz processor.

Table 5.3: Running time, waiting time, SLA violations, revenue and profit results for running the workload on local servers

CLs(*n*)	VMs(*n*)	RT(*ms*)	WT(*ms*)	SV(*n*)	Rev($)	Prof($)
100	100	3	10	0	1596	319
200	100	13	23	0	3192	638
300	100	24	34	0	4788	958
400	100	45	55	0	6384	1277
500	100	72	82	2	7980	1585
600	100	96	106	26	9576	1759
700	100	134	144	64	11172	1849
800	100	179	189	109	12768	1899
900	100	216	226	146	14364	1998
1000	100	272	282	202	15960	1982
1100	100	334	344	264	17556	1927
1200	100	384	394	314	19152	1946
1300	100	457	467	387	20748	1827
1400	100	537	547	467	22344	1666
1500	100	600	610	530	23940	1608



(a) Performance with respect to running time

(b) Performance with respect to waiting time

Figure 5.4: Performance with respect to running and waiting time (seconds)

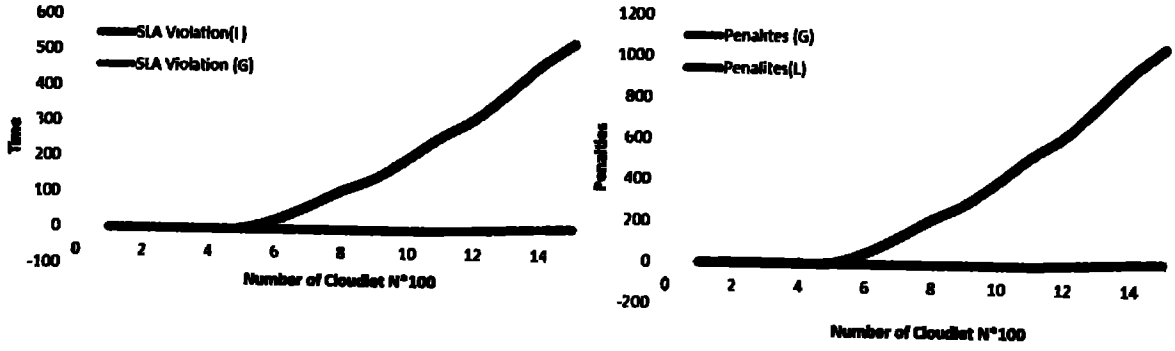
Table 5.4: Running time, waiting time, SLA violations, revenue and profit results for running the workload on global servers

CLs(*n*)	VMs(*n*)	RT(*ms*)	WT(*ms*)	SV(*n*)	Rev($)	Prof($)
100	100	3	53	0	1400	140
200	200	6	56	0	2800	280
300	300	8	58	0	4200	420
400	400	11	61	0	5600	560
500	500	14	64	0	7000	700
600	600	16	66	0	8400	840
700	700	19	69	0	9800	980
800	800	22	72	0	11200	1120
900	900	24	74	0	12600	1260
1000	1000	27	77	0	14000	1400
1100	1100	30	80	0	15400	1540
1200	1200	32	82	2	16800	1676
1300	1300	35	85	5	18200	1810
1400	1400	38	88	8	19600	1943
1500	1500	40	90	10	21000	2080

Figure 5.4a shows how long the workload ran on local and global servers. On a local server, the dynamic number of workloads are run on a fixed number of virtual machines. We found that the execution time peaked as the workload increased. To maintain performance, the provider must only accept limited workloads and refuse higher tasks due to non-scalable resources. Above equations show that the scalability of resources is directly proportional to customer satisfaction and minimizing SLA violations. In a real situation, such as in this scenario, the workload will cause customer dissatisfaction if it is increased. In a global server, new virtual machines are initialized for each new task. The results show that the execution time of the global server is not changed instead of the local server. Few variations occurred in an extreme workload, but this was due to the migration time.

Figure 5.4b shows the waiting time of local and global servers. During the first execution, the waiting time of the global server is greater than that of the local server. But, as we can see, the waiting time of the

local server greatly increases when the workload is increased by 400. Penalties are also increasing. It also maximizes customer satisfaction, which has a greater impact on the cloud market. The overall results of the server show that there is no effect on the waiting time. it varies only slightly in terms of extreme use due to the migration time.

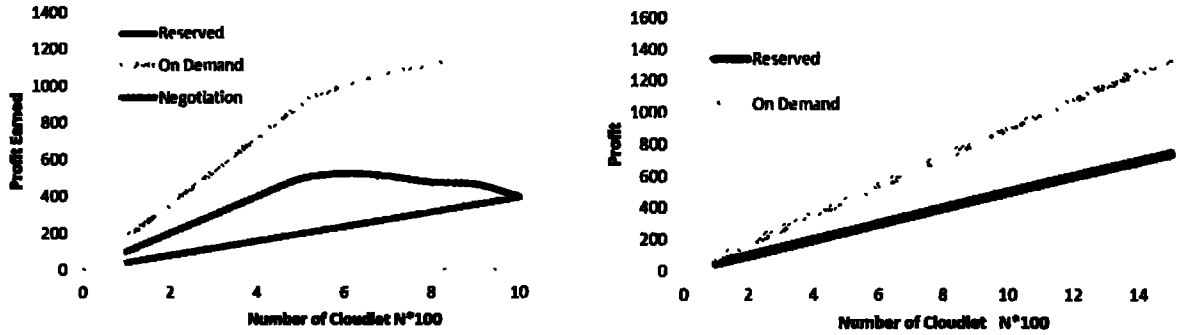


(a) SLA Violation (b) Penalties

Figure 5.5: SLAs violation and penalties

Figure 5.5a shows the number of SLA violations. The results of the local server show that the SLA violation increases considerably concerning the workload. The formulation shows that V_n is inversely proportional to the performance, a performance that is directly proportional to the scalability of the service. In such circumstances, most providers' income wastes in penalties. Excessive use has perverse effects on the cloud. The formulation shows that SLA violation is inversely proportional to scalable services. Global resources are scalable and new virtual machines are initialized for each new task. The global server results show that only a few ALS violations take place under extreme working conditions. These violations are negligible in the cloud market.

Figure 5.5b shows the penalties of local and global servers. The result explains that during the first simulation, both servers remain in the safe region, however, when the workload increases by 500, the local server penalties increase considerably, while the global server remains stable. These penalties have an extremely negative impact on the cloud computing business. Customers end their business when penalties pass thresholds values. The results of the local server show that the increase in workload also leads to penalties because resources are not scalable.



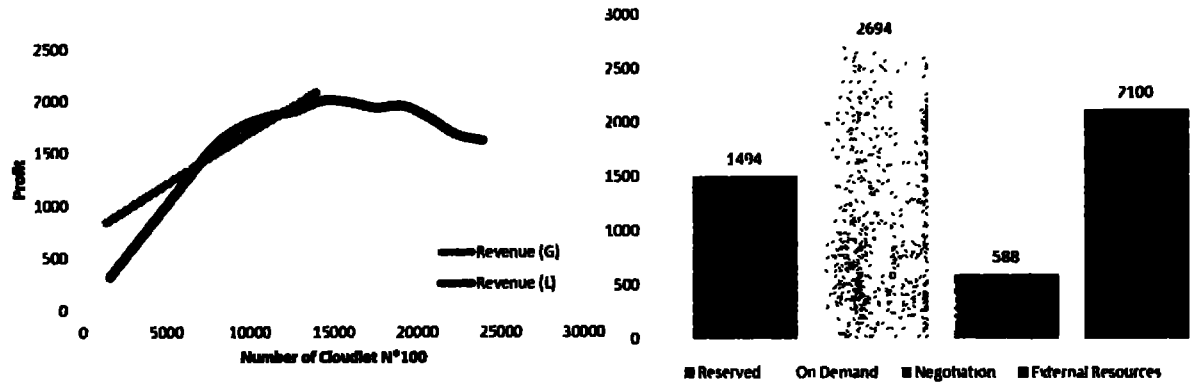
(a) Revenue from Local Data Center

(b) Revenue from Global Data Center

Figure 5.6: Revenue from Local and Global Data Centers

The figure 5.6a shows the providers' profit. In this chapter, three types of customers are discussed. These are reserved, on demand and negotiated. On-demand customers are the main revenue generators. Negotiated customers are only accepted in the underutilization scenario. Results of on-demand and reserved profit show that as the workload is increased, revenue generation drops below threshold value. In such situation, instead of increasing revenue, provider wastes it in paying penalties. To solve this problem, the proposed framework hires resources from external providers to handle the heavy workload with appropriate prices and performance. The goal of offering different prices is to attract customers. Customers select prices based on their own choices. The results show that negotiated based pricing revenue is lower yet at least some revenue is generated instead of wasting the underutilized resources.

Figure 5.6b shows revenue generated by external providers. The workload of the negotiated customers is not transferred to external resources. Their requests are accepted only in case of under-utilization. The overall revenue generated by reserved and on-demand customers increases providers' profits as well as customer satisfaction. External resources extend the providers' business. They facilitate the providers in terms of scaling resources, increasing customer satisfaction and quality of services. Global server revenue increases constantly relative to the local server due to resource scalability. There is less risk of breach of service level agreement and penalties.



(a) Total Revenue Generation

(b) Total Profit Generation

Figure 5.7: Revenue and Profit

Figure 5.7a shows the revenue generation in peak workloads. The same workload is forwarded towards both local and global servers. The results show that in the initial simulation, both servers generate the same revenue. But as the workload increases by 500, revenue generated by the local data server begins to go downward. Instead of increasing the revenue, provider wastes revenue in penalty payments due to resource limitation. On the contrary, we can see that the revenue generated by the global server grows upward with the increase in workload. It is due to the resource scalability. Results show that to save limited internal resources from over-utilization and to satisfy customers, hiring external resources is the only good option for providers.

The revenue is maximized by generating it in different ways. Figure 5.7b, shows the total revenue generated by the system. Revenues generated by reserved, on-demand, negotiated and external customers are 21%, 39%, 9% and 30% respectively. The results show that these resources contribute greatly to revenue generation.

These investigations are different from previous works in many ways.

- We have taken the scalability of the resources as a top priority. Limited resources are the main reason of penalties. Resource scalability not only increases the local resource utilization, but it also increases the number of new customers and their retention.
- Different prices options and paying back to customers in case of violation attracts more customers. More customers mean more resources utilization

The simulation results show that the proposed model is able to effectively manage the dynamic requirements of customers. The proposed model uses external resources, to minimize violations and rejections of SLAs.

These external resources help the provider in terms of customer satisfaction, customer attraction and revenue maximization. These resources help the provider to adapt services to the needs of the customer. The results show that the proposed model is greatly contributing to revenue maximization and customer satisfaction.

5.3 Summary and Conclusion

In this chapter, the main focus was to handle the massive demands of customers with limited resources in order to maximize the providers' revenue. In extreme use, the high workload is outsourced to external resources, extending the provider's limited resources. However, as a federated cloud, the proposed framework does not require providers to use a specific alliance. Common prices are used and optimized to maximize revenue and customer satisfaction. The proposed framework is able to effectively handle the massive demands of customers. The results of the simulation show that it contributes greatly to revenue maximization and customer satisfaction. Due to the disbursed, massive data and virtual scenario, it is hard and challenging to measure it. Our future intentions are to work on a framework to measure customer satisfaction in the cloud business.

Chapter 6

Revenue Maximization by Performance based Service Level Agreement

Cost, performance, and *penalties* are the key factors to revenue generation and customer satisfaction. They have a complex correlation, that gets more complicated when missing a proper framework that unambiguously defines these factors. Service Level Agreement (SLA) is the initial document discussing selected parameters as a precondition to business initialization. The clear definition and application of the SLA is of paramount importance as for modern *as a Service* online businesses no direct communication between provider and consumer is expected. For the proper implementation of SLA, there should be a satisfactory approach for measuring and monitoring Qualify of Service (QoS) metrics.

Cloud computing is a successful and widespread paradigm, effectively delivering desktop services over the internet like other common utilities as telephone, gas, and electricity distribution [33]. Rather than investing capital amount on infrastructure, software, registrations and IT experts, customers pay cloud providers for the use of the services or for the leasing of resources [8]. The characteristics of the service that is paid for are agreed upon in the form of Service Level Agreements (SLA), legal contract negotiated and signed by consumer and provider party [26]. In the frame of an SLA, complete Service Level Objectives (SLOs) are discussed, and desired Quality of Service (QoS) variables are defined and agreed upon as well [9]. Service provision is monitored with agreed terms and conditions: on SLA violation, defaulter is penalized [13]. Given these practices, it is self-evident that clearly explained SLAs, with understandable ties to the customer's business interests, are the key to help providers offer the best service for customer's requirements and maximize customer satisfaction [30].

Indeed, ***Business Intelligence*** tools and data-driven automation are the more and more becoming central to any commercial activity, and having a significant part of the ICT infrastructure outsourced to cloud providers

makes harder to apply insight-driven automation in the management (and re-negotiation) of rented cloud services [22, 100].

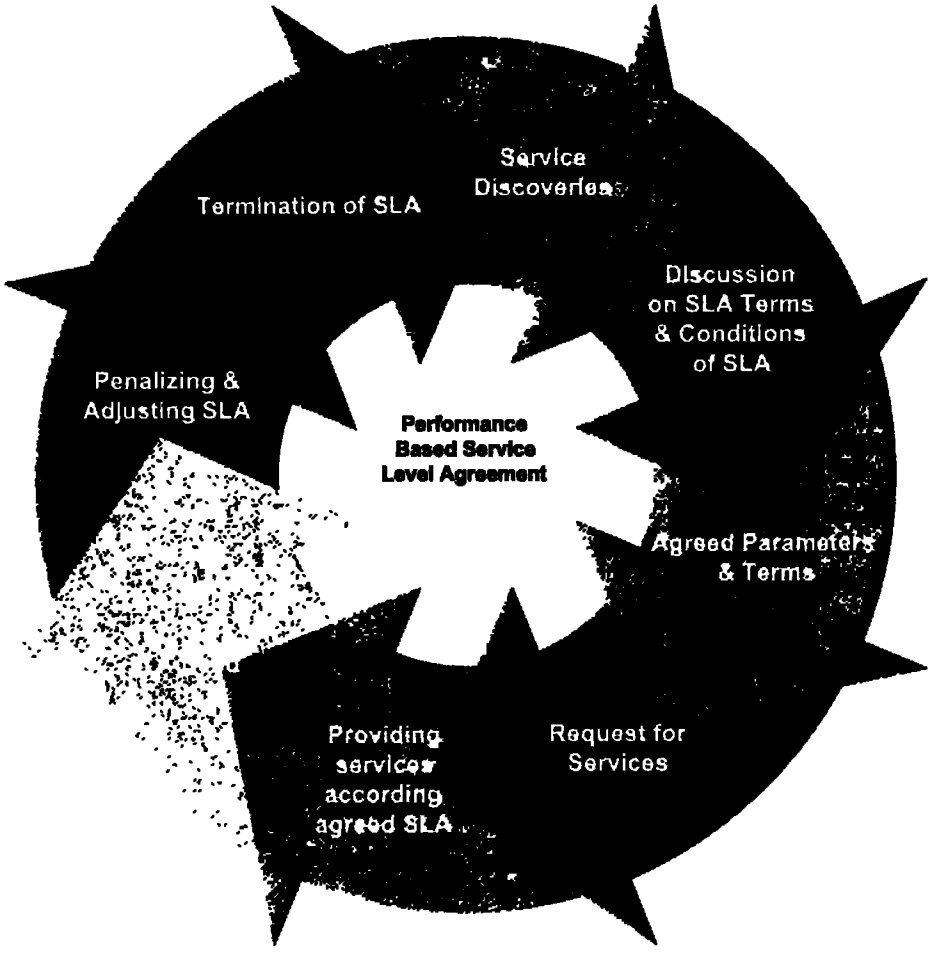


Figure 6.1: Performance based Service Level Agreement in Cloud Computing

Performance, penalties, cost and revenue are interrelated in a complex way, with both direct and inverse proportional effects [103, 119]. Actually, by assuming a temporary intuitive definition for such key parameters from the point of view of the cloud provider, we consider the performance (of the provided service), costs (payed for the service), penalties (payed by the provider for SLA violation), and revenues (earned from offering the service). These parameters show the following behavior: a raise in performance maximizes the revenue and minimize the penalties, although, performance is also directly proportional to the cost, which is inversely proportional to the revenue. Further complication is implied as there is no generally adopted framework clearly defining such parameters in relation to the SLA life cycle. Fig. 6.1 shows the life cycle of the proposed PerSLA.

Table 6.1: Symbols used in formulation

Symbol	Definition	Symbol	Definition
τ_{run}	Workload running time	τ_{res}	Workload response time
τ_i	Task initialization time	τ_f	Task finalization time
τ_{avail}	Resources availability time	τ_{down}	Resources down time
τ_{ag}	Total agreed time	τ_{exe}	Resources execution time
Req_t	Request type	Res_t	Resources type
SS	Service Scalability	Vn	Number of SLA violations
η	Penalties due Vn	Per	Resources performance
QoS	Quality of Services	Eff	Efficiency of Services
CS	Customer satisfaction	χ_t	Total costs
χ_{hu}	Total costs for unit hired	χ_n	Network costs
χ_{sec}	Security costs	$\chi_m ii$	Internal migration costs
$\chi_m ij$	External migration costs	$avail$	Resources availability
ρ	Prices	ρ_r	Prices for reserved services
ρ_{db}	Prices for demand based	ρ_n	Negotiated based pricing
$fail$	Resources failure	per	Performance
Δ	Margin	κ	Constant
φ_i	First threshold	φ_{ii}	Second threshold
φ_{iii}	Third threshold	η_{rate}	Penalty rate
BW_{up}	Uploading bandwidth	BW_{down}	Downloading bandwidth
$Prof$	Total profit earned	μ	Agreed QoS objectives

Although SLA, performance, and penalties are widely investigated in the scientific literature, we could not find an analysis of SLAs aimed at joint optimization of revenue for the provider and satisfaction for the customer. Authors in [154], [86], [87], [91] , [89], [92] investigated different types of SLAs but did not pursue our stated objective. From the point of view of maximizing revenue generation for the cloud provider SLA for optimal resources scheduling have been discussed [108] separately, also QoS management for cloud customer satisfaction has been explored [95, 102] .

Recent advances in Information Technology and infrastructures fueled a massive transition from in-house ICT services to cloud computing. Furthermore, new data processing paradigms (e.g. Big Data) have opened new business models, creating new technological requirements, and increasing need for cloud services. Due to the massive customers workloads and internet business, extensive automation is required. Therefore it is essential to have clear-cut SLA to avoid disastrous consequences in the customer business. This chapter investigated an SLA framework aimed at optimizing both customer satisfaction and provider revenues, considering performance and penalties. Moreover, a threshold-based approach is proposed to avoid SLA termination, that is a radical resolution strategy in cases of SLA violations and major business concern for the cloud computing industry [90]. In the proposed approach, SLA violation does not leads directly to termination, but on earlier small violations the service prices are decreased, constituting the initial incentive to the provider to adjust the performance. It can be hypothesized that an initial small decrease in performance is unlikely to result in the customer dropout and lose of trust. Moreover, the reduced QoS that has been actually delivered is automatically compensated for with the price reduction. The proposed mechanism therefore allows for more flexibility in SLA management and prevents unnecessary SLA termination. The feasibility and efficacy of the proposed approach have been validated in simulation, using the CloudSim simulator to evaluate the modeled cloud system in various topological and temporal circumstances. Table 6.1 shows the symbols used in the formulation.

Contributions of this chapter are:

- The design of a SLA framework that can maximize performance (hence user satisfaction) and increase the revenue for the provider;
- In the proposed framework, a layered structure of SLA penalties, to incentivize the cloud provider to offer the best performance while avoiding unnecessary SLA termination;
- An experimental evaluation of the proposed framework within a simulated environment with multiple scenarios
- A determination of optimum values for monitored parameters to maximize provider revenues and customer satisfaction in the simulated scenarios.

In Performance based Service Level Agreement (PerSLA), SLA does not leads directly termination. On earlier violations, prices are decreased, which are the initial indicator to the provider to adjust the performance. An initial decrease in performance is not much to results the customer dropout and trust. They are also paid back for lower performance with respect to downtime. So the initial reduction in performance does not demotivate them. The proposed approach optimize performance, penalties, cost and revenue. These parameters are also monitored. CloudSim simulator is used to check the system within the various topological and temporal circumstances. Results proved the supremacy of the proposed work

6.1 System Model

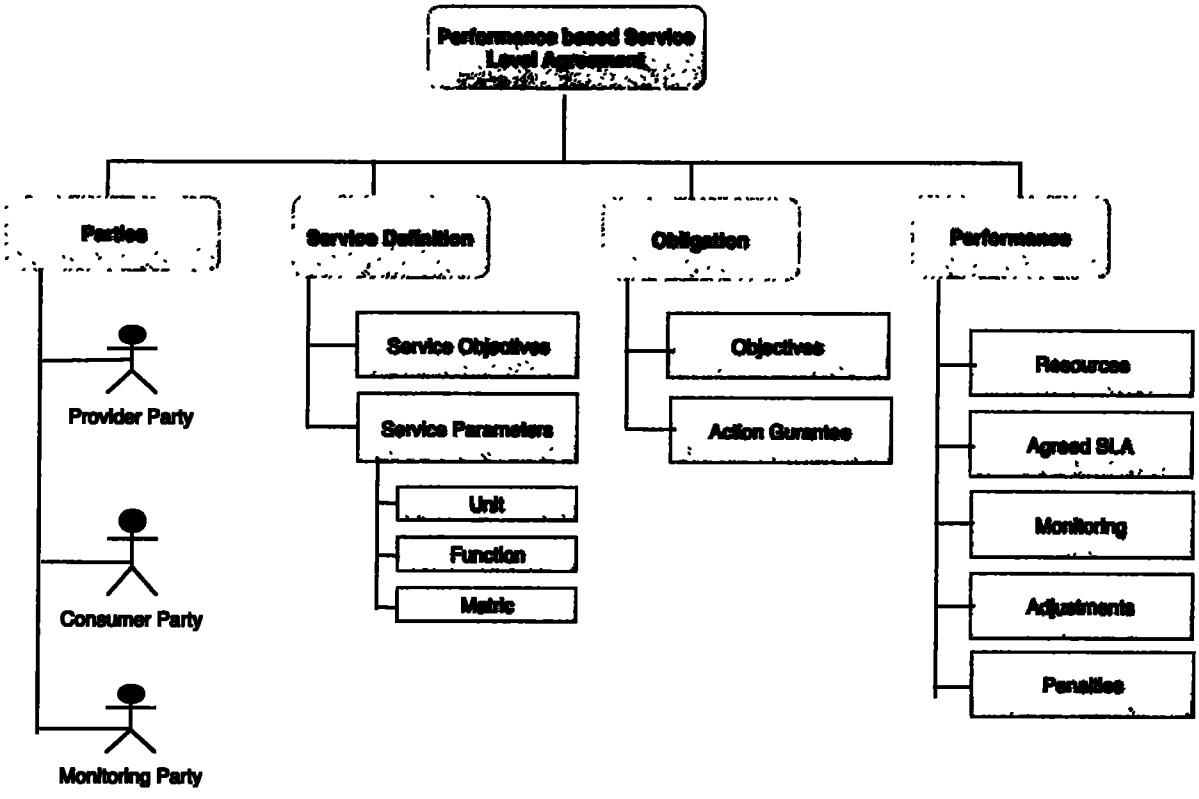


Figure 6.2: PerSLA Structure in Cloud Computing

Three types of parties participated in this model. Provider provides the services, consumer rents the services, and monitoring party monitors the services according to agreed terms and conditions to ensure the Quality of Service (QoS). In proposed model, service definition explains the services which are to be provided. Parameters are defined by metrics. In our scenario, we discussed response time in seconds, execution time in seconds, availability of memory in MBs, availability of storage in GB and bandwidth in Mbps per seconds. Figure 6.2 shows the structure of PerSLA.

Matrices and variables are used to measure services. The obligation is actually a conditional structure which check the services provision with the agreed terms and conditions. Here in this model we proposed two threshold values to check the services. To determine the SLA violation, all services are monitored with respect to the agreed terms and conditions. Performance analyzer examines the performance of the services. Penalties are enforced on the failing party in case of unsatisfactory services.

6.1.1 Service Level Agreement

SLA violation is the main cause of penalties. Violations occurs in case of QoS degradation. This may be due to overload or network delay [84] . Penalties are applied to on provider to satisfy the customers. This penalties may be in terms of direct cash or price reducing. In this model, prices are reduced for downtime.

Three penalty layer structure is used in this framework. Three threshold values are used to apply penalties on violation of SLA. It increases the customer retention. Table 6.2 shows the threshold values used in this study. Symbols φ_i, φ_{ii}, and φ_{iii} represents the threshold.

Table 6.2: Threshold values for agreed parameters

SLA Parameter	Thr (φ_i)	Thr (φ_{ii})	Thr (φ_{iii})
Response time (sec)	2 s	5 s	10 s
Execution time (sec)	3 s	7 s	15 s
Availability (%)	99 %	97 %	95 %
Bandwidth (Mbps)	9.9	9.8	9.5

6.1.1.1 Execution Time

Execution time shows the time used to complete the customer's demand. Execution time is an important parameter of SLA. This depends on request type and also the resources on which the request is executed. If the resources are not appropriate, it takes longer than usual [116].

$$\tau_{run} \propto Res_{type} \tag{6.1}$$

$$\tau_{run} \propto Req_{type} \tag{6.2}$$

$$\tau_{run} = \tau_i - \tau_f \tag{6.3}$$

Where τ_{run} shows the running time, Res_{type} shows the resources type, τ_i task initialization time and τ_f shows the task finalization time.

6.1.1.2 Response Time

Response time is the waiting time of customer request in waiting queue. Response time depends on the underlying resources utilization and network bandwidth. If the under laying resources are heavily utilized, it takes longer to execute new tasks [95].

$$\tau_{res} \propto SS \tag{6.4}$$

$$\tau_{res} \propto \upsilon \tag{6.5}$$

$$\tau_{res} = \tau_i - \tau_s \tag{6.6}$$

Where τ_{res} shows the response time of the services, and SS shows the services scalability.

6.1.1.3 Resources Availability

Availability define the presence of the agreed resources when they are required. Availability discusses all these resources which are discussed in SLA [95].

$$avail \propto 1/fail \tag{6.7}$$

$$avail \propto S \tag{6.8}$$

$$avail \propto \chi \tag{6.9}$$

We calculated the availability as

$$Availability \propto \frac{\tau_{avail} - \tau_{down}}{\tau_{com}} \tag{6.10}$$

Where τ_{com} shows the total computation time agreed in SLA, τ_{avail} shows the total availability of computation time and τ_{down} shows the down time during running.

6.1.1.4 Resources Scalability

Resource scalability is vital for QoS. If the resources are non-scalable, they will be over utilized leading to penalties and revenue degradation. Most of the performance parameters directly depend on the resources scalability [32, 122].

If the resources are not scaleable, execution and response time may increase. To address this issue, external resources may be hired from third party to maximize the resources scalability.

Resources scalability depends on total capacity of the provider. SLA violation V_n happen due to non-scalable resources. Below equation discuss the negative effects of non-scalable resources.

We know that (equation 2.2, 2.13, and 2.12) penalties (η) are directly proportional to SLA violations (V_n), where SLA violation is inversely proportional to performance (per), services scalability (SS), Quality of Service (QoS), and efficiency (Eff).

6.1.1.5 Resources Reliability

Resources reliability is defined as that resources perform the predefined functionality for the agreed time under agreed terms and conditions. The resources are reliable if they are fault tolerant and automatically recoverable. Reliability also discusses the fault tolerance, recover ability and resources constancy. Lower reliability minimizes the customer retention which leads to lower revenue [95]. In our case, reliability depends on performance, and availability. We consider the system reliable, if its performance is good and available on demand.

$$Per \propto Reliability \tag{6.11}$$

$$Per \propto \frac{1}{\tau_{run}} \tag{6.12}$$

6.1.2 The Cost & Prices Model

Pricing methods have a significant part in revenue generation. Different pricing methods are used in cloud markets. In the reservation pricing, customers reserved the resources for a particular time such as a month or year etc. Due to the reservation and massive use, customers get a heavy discount. However, they pay the prior registration fee.

In ondemand pricing, customers pay for units they consumed. In this scheme, prices are comparatively high, however, customers are paid for SLA violations [129].

The primary cost is calculated as

$$\chi_t = \chi_{hu} + \omega + \chi_n + \chi_m^{ii} + \chi_{sec} + \kappa \tag{6.13}$$

This study uses joint prices for customers' satisfaction. Prices are fixed for high performance and reserved customers. On-demand prices are employed for dynamic customers, demanding high performance. In the case of underutilization, negotiation-based prices are used. In the actual cloud background, it will increase the resources utilization and customer satisfaction. Prices are calculated as

$$\rho_r = \chi_{cost}^{ii} \times \upsilon_h \times t - \sigma + \Delta \tag{6.14}$$

$$\rho_{db} = \chi_{cost}^{ii} \times \upsilon_h \times t + \Delta \tag{6.15}$$

$$\rho_{Neg} = \chi_{cost}^{ii} + \Delta \tag{6.16}$$

Where

$$\Delta > \psi$$

Where χ_t shows the total cost, χ_{hu} unit cost, ω shows the human resources, χ_n shows the network cost, χ_m^{ii} shows the internal cost, χ_{sec} shows the security cost and κ represent the constant.

6.1.3 Penalties

Penalties greatly affect the cloud business. Usually providers admit weighted workloads but later on, they fail to provide resources as per the agreement and in the result, they have to pay much of their revenue in penalties. Table 6.3 shows the penalties structure for SLA violations.

To put a limit check on penalties, it must not exceed from 10% of the service charges [151]. If rate of penalty increases from 10% of the services charges, SLA should be terminated. This will help to maintain business between provider and consumer. If penalties increases from 10 %, it will have a very bad impact on provider [95].

Penalties decreases prices or may have direct sanctions on provider party. It also decreases reputation of provider and in future, customers would not trust on such like provider [30].

Table 6.3: Penalties structure for SLA violations

SLA Parameter	(η_i)	(η_{ii})	(η_{iii})
Response time	5%	10%	SLA Termination
Execution time	5%	10%	SLA Termination
Availability	5%	10%	SLA Termination
Bandwidth	5%	10%	SLA Termination

When first threshold is violated, prices are decreased by η_i and if 2^{nd} threshold is crossed prices are charged η_{ii} less, and if performance goes down from third threshold, business is terminated. Penalties are calculated automatically. Claim may be registered for higher loss. Market implementer may change or extend it according to their scenario.

6.1.3.1 Execution Time

For execution time (τ_{run}) three types of threshold values are declared. if the resources τ_{run} is below the φ_i no penalties are imposed on the provider. But, if the τ_{run} increases by φ_i and φ_{ii} provider is penalized by

Pen_i and Pen_{ii} respectively. SLA is terminated if τ_{run} increases by Pen_{iii}. Pen_{run} shows the penalties of running time.

$$Pen_{run}(x) = \begin{cases} 0 & \tau_{run} \leq \varphi_i \\ Pen_i & \varphi_i < \tau_{run} \leq \varphi_{ii} \\ Pen_{ii} & \varphi_{ii} < \tau_{run} \leq \varphi_{iii} \\ S_{ter} & \tau_{run} > \varphi_{iii} \end{cases} \tag{6.17}$$

Where $Pen_{run}(x)$ shows the total penalties of running time, Pen_i, shows the penalties of initial threshold, Pen_{ii} shows the penalties of the second threshold, Pen_{iii} shows the penalties of third threshold and S_{ter} shows the SLA termination.

6.1.3.2 Response Time

Response time has three threshold values. if the resources τ_{res} is below the φ_i no penalties are imposed on the provider. But, if the τ_{res} increases by φ_{ii} and φ_{iii}, provider is penalized by pen_i and pen_{ii} respectively. SLA is terminated if τ_{run} increases by pen_{iii}. Pen_{res} shows the penalties due to response time.

$$Pen_{res}(x) = \begin{cases} 0 & \tau_{res} \leq \varphi_i \\ Pen_i & \varphi_i < \tau_{res} \leq \varphi_{ii} \\ Pen_{ii} & \varphi_{ii} < \tau_{res} \leq \varphi_{iii} \\ S_{ter} & \tau_{res} > \varphi_{iii} \end{cases} \tag{6.18}$$

Where $Pen_{res}(x)$ shows the total penalties of running time, Pen_i, shows the penalties of initial threshold, Pen_{ii} shows the penalties of the second threshold, Pen_{iii} shows the penalties of third threshold and S_{ter} shows the SLA termination.

6.1.3.3 Availability

Availability refer to available time of resources with respect to agreed time of particular resource. It also discusses the storage and memory availability according to agreed SLA.

$$Pen_{avail}(x) = \begin{cases} 0 & \tau_{res} \leq \varphi_i \\ Pen_i & \varphi_i < \tau_{res} \leq \varphi_{ii} \\ Pen_{ii} & \varphi_{ii} < \tau_{res} \leq \varphi_{iii} \\ S_{ter} & \text{otherwise} \end{cases} \tag{6.19}$$

Where $Pen_{avail}(x)$ shows the total penalties of running time, Pen_i, shows the penalties of initial threshold, Pen_{ii} shows the penalties of the second threshold, Pen_{iii} shows the penalties of third threshold and S_{ter} shows the SLA termination.

6.1.3.4 Memory

Memory refer here the primary memory used during processing. if the resources mem violation is below the φ_i no penalties are imposed on the provider. But, if the mem increases by φ_i and φ_{ii}, provider is penalized by Pen_i and Pen_{ii} respectively. SLA is terminated if mem violation increases by Pen_{iii}.

$$Pen_{mem}(x) = \begin{cases} 0 & \sigma \geq \varphi_i \\ Pen_i & \varphi_i < mem \geq \varphi_{ii} \\ Pen_{ii} & \varphi_{ii} < mem\tau_{res} \geq \varphi_{iii} \\ S_{ter} & \text{otherwise} \end{cases} \tag{6.20}$$

Where $Pen_{avail}(x)$ shows the total penalties of running time, Pen_i, shows the penalties of initial threshold, Pen_{ii} shows the penalties of the second threshold, Pen_{iii} shows the penalties of third threshold and S_{ter} shows the SLA termination.

Penalty calculation

$$Pen = \sum_{k=0}^{n} (Pen_{rate} \times (\tau_{tst} - \tau_{at})) \tag{6.21}$$

Where Pen_{rate} shows the rate of penalty and $(\tau_{tt} - \tau_{at})$ shows the non-availability time. For example, customers C1, C2, C3, C4 and C5 utilized storage of 5, 10, 59 and 100 GB respectively, processing power

of 1 VM each , and memory of 2,2,2,2, and 3 GB for 12 hours where the prices are: $\rho_r s$= 0.05, $\rho_r p$= 0.0192 , and $\rho_r m$= 0.002 [153]. We assume that running time goes down from φ_{ii} for an hour. Below table 6.4 shows the penalties calculation for one hour down time for processing power and memory.

Table 6.4: Penalties calculation

Cus	Us	Up	Um	T	Re	Mar	Dis	Vup	VM	Pen	Dis	Revt
C1	5	1	2	12	2.678	0.267	0.535	1	1	0.002	0.267	3.211
C2	10	1	2	12	5.078	0.507	1.015	1	1	0.002	0.507	6.090
C3	50	1	2	24	48.55	4.855	9.711	1	1	0.002	4.85	58.26
C4	100	1	3	36	144.9	14.49	28.98	1	1	0.002	14.49	173.8
Total					201.2	20.12	40.24			0.009	20.12	241.4

Where **Cus** shows customers, **Us** shows the utilized storage unit, **Up** represents the processing usage in terms of VM, **Um** shows the memory unit in GB, **T** is total utilization time, **Re** represents revenue, **Mar** shows the margin earned, **Dis** shows the discount, **Vup** shows the time for which processing threshold is violated, **VM** shows the time for which memory threshold is violated, **Pen** shows the penalties imposed, **Dis** shows the discount, and **Revt** shows the revenue.

Total Penalty

$$Pen_{total} = \sum_{k=0}^{n} (Pen_{run}n + Pen_{res}n + Pen_{avail}n + Pen_{mem}n) \tag{6.22}$$

Pursuing the the equation no 6.22 the total penalties are calculating by adding the penalties of running time ($Pen_{run}n$), response time ($Pen_{res}n$), non availability of the services ($Pen_{avail}n$), and penalties of memory ($Pen_{mem}n$).

The final profit is calculated is

$$\chi_{total} = \sum_{k=0}^{n} \chi_k + Pen_{total} \tag{6.23}$$

$$prof_{total} = \sum_{k=0}^{n} Rev_k - \chi_{total} \tag{6.24}$$

If the resources are not properly scheduled on VMs, they leads to penalties. To overcome these issues, algorithms create mapping between all free VMs and customers' request. Such VMs are selected which are best fit in terms of QoS and cost.

6.1.4 Monitoring

Monitoring is the main component of SLA. Monitoring is important for the Quality of Service (QoS) and tenants trust. It uses the formulation as discussed earlier, agreed SLA parameters and the current status of the services. Cloud Service Providers (CSP), as well as tenants, need reliable and efficient monitoring to ensure SLA on both sides. Reliable monitoring increases trust between CSP and tenants.

6.1.5 Proposed Algorithms

Two algorithms are proposed for Performance based Service Level Agreement. The first algorithm properly implement the SLA and the second algorithm optimize the scheduling. Three stages for threshold are defined. On the first threshold violation, prices are reduced by 5%, on second threshold violation prices are reduced by 10%, and on third violation SLA is terminated.

The working of the the Algorithm 5 is;

1. Line 1 receives the tasks.
2. Line 2-3 define the parameters and their units.
3. Line 5-6 defined the penalty stages and penalty quantity.
4. Line 7-14 evaluate the performance for different threshold values to calculate the penalties status.
5. Line 15-22 implement the penalties as per the penalties status.
6. And finally, line no 23 calculates the total penalties.

The first threshold is optimized at such a point which not much hurt the customers but notify the provider to adjust the resources performance. Algorithm 5 uses table number I and II values and monitoring report statistics to applies penalties on failure party. Every parameter of SLA has different threshold and penalties

Algorithm 5 PerSLA Algorithm

Input: SLA parameters , Penalties description and Customers workloads (UT)

Output: Optimized prices

RECEIVE SLA REQUESTS

DEFINING PARAMETERS(τ_{run},τ_{res}, Through)

PARAMETERS DESCRIPTION(Unit, Function and Metrics)

PENALTY STAGE(1, 2, 3)

PENALTY DESCRIPTION(5 % , 10 %, SLA termination)

if $\tau_{run} \leq \varphi_i$ **then** ▷ Threshold I

    $Pen_{run} = 1, Pen_{status} = yes$

end if

if $\varphi_i > \tau_{run} \leq \varphi_{ii}$ **then** ▷ Threshold II

    $Pen_{run} = 2, Pen_{status} = yes$

end if

if $\varphi_{ii} > \tau_{run} \leq \varphi_{iii}$ **then** ▷ Threshold III

    $Pen_{run} = 3, Pen_{status} = yes$

end if

if $Pen_{status} = yes$ **then** ▷ Penalties Structure

    $Pen_{run} = \sum_j^n \{(\tau_i - \tau_f) \times penrun_j\}$

    $Pen_{res} = \sum_j^n \{(\tau i - \tau_s) \times penres_j\}$

    $Pen_{avail} = \sum_j^n \{\frac{\tau_{avail} - \tau_{down}}{\tau_{com}} \times penavail_j\}$

end if

$Pen_{total} = Pen_{run} + Pen_{res} + Pen_{avail} + k$ ▷ Total Penalty

values. Penalties are imposed according to the nature of violations. The algorithm 5 takes $O(c)$ as running time on calculating the penalties.

Algorithm 6 is called by the threshold call of I and II. It works for performance adjustment.

The working of the the Algorithm 6 is;

1. Line 1-2 checks the QoS and workload with threshold values and available space required on the VMs.
2. Line 3-5 searches the best fit VM in terms of low cost and high performance.
3. Line 10 checks the local resources availability and in case of lower availability, line 12 searches the external providers.

If system receives reports regarding performance degradation, this algorithm tries to adjust the performance by migrating the workload from over utilized to under utilized VMs. It maximizes the performance and revenue. In case of limited resources, algorithm 6 searches the global provider to scale the resources. The algorithm 6 takes $O(n)$ as running time on using a nested and sequential looping structure searching optimized VMs, and provider.

Algorithm 6 Scheduling

Input: Free VMs (VM_{free}) and Customers workloads (UT)

Output: Optimized Combination of VM_{free} and UT

1: **if** $OoS \leq threshold$ **then** ▷ To maintain the QoS
2: **if** $workload \geq VM$ **then**
3: $VM_{max} \leftarrow 0$
4: RESOURCE-OPTIMIZATION(VM_n, UT)
5: $VM_{max} \leftarrow UT$
6: **if** Scheduling not successful **then**
7: Restart from step 1
8: **end if**
9: **end if**
10: **if** Local Resources Decreases **then** ▷ To maintain the Scale of Services
11: $prov_{max} \leftarrow 0$
12: Searching provider using algorithm no 7.1
13: **return** $Prov_{max}$
14: **if** Scheduling not successful **then**
15: Restart from step 1
16: **end if**
17: **end if**
18: **end if**

6.2 Evaluation

This section explains the experimental setup to assess the performance and working of this model. It is very costly and complicated to create cloud labs. To overcome this issue, different simulations tools are used to simulate the cloud environment. CloudSim, Cloud Analyst, Green Cloud and ICanCloud are the leading cloud simulation tools. To measure the performance of the proposed PerSLA (Performance-based Service Level Agreement) model and algorithms, CloudSim simulator is used.

6.2.1 Experimental Setup

In the proposed scenario, 1700 VMs are created as resources and 2000 cloudlet as customers workload. Algorithm 5 generates the number of cloudlets having file size: 300000 MB and runs it on both servers. Simulation is run for the number of time, changing VMs from 50 to 1700 and cloudlets from 50 to 1700. During simulation, various factors like the running time, waiting time, SLA violation, penalties and revenue generation by both servers has been observed.

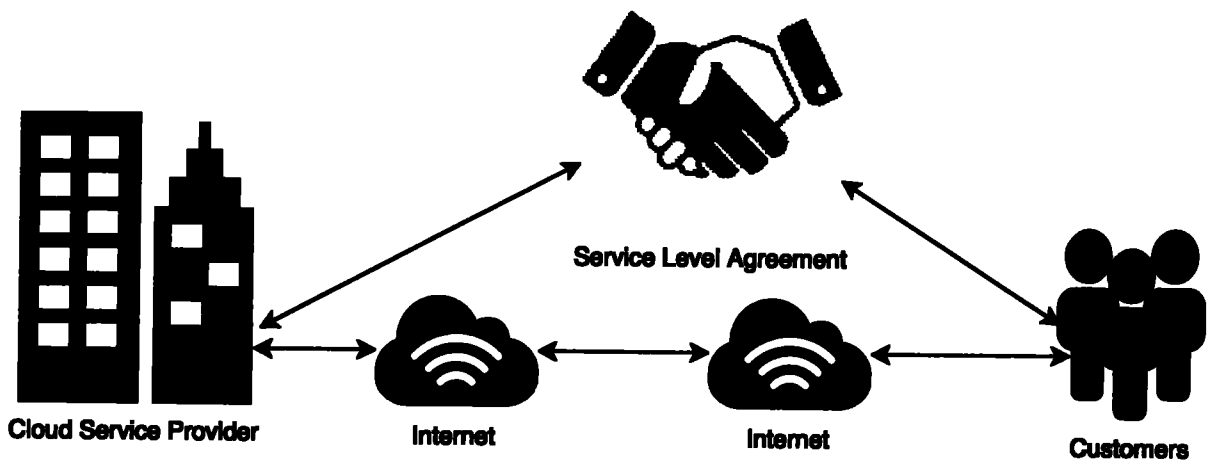


Figure 6.3: Performance based Service Level Agreement experimental setup

CloudSim is widely used simulator for cloud architectures simulation. It uses java as a command line. It helps to virtually build servers, datacentres and virtual machines etc. Workloads deployment and various scheduling algorithm can be easily executed. CloudSim uses cloudlet as user workload. Different algorithms can be implemented to schedule different customers workload on different resources. Figure 6.3 shows the experimental structure of PerSLA.

The CloudSim simulator has been extended to assess the performance of the PerSLA. We added new classes, PerSLA, Penalties and Prices to calculate and adjust the performance and prices. Penalties are automatically calculated according to the PerSLA class inputs. Prices are also adjusted according to the PerSLA and penalties classes input. In the proposed scenario, 1700 VMs are created as resources and 2000 cloudlet as customers workload [151][150].

Algorithm 5 produces cloudlets having file size: 300000 MB and execute it on different servers. Simulation is executed for the number of times, changing VMs from 50 to 1700 and cloudlets from 50 to 1700 [150]. During simulation, various factors like the running time, waiting time, SLA violation, revenue generation and penalties of both servers were observed.

6.2.2 Evaluation Results

In this portion, numbers of tests are presented which were carried out in order to understand how the proposed study contribute to the core parameters optimization. Numbers of VMs and cloudlet were created and ran during experimentation. It is assumed that the capacity of the provider resources is scalable. Simulation results are calculated by changing data-centers from 1 to 5, cloudlets from 50 to 1700 and VMs from 50 to 1700. Every cloudlet has 300000 MB of file size and every VM has 3 GB of RAM and 3.2 GHz processor.

Experiments were conducted in three different rounds. In the first turn, the dynamic number of cloudlets and VMs were run on a single data centre as shown in table 6.5. We changed the cloudlets and VMs from 50 to 400 and run it on a single data-center. In the second round, data centre were changed according to the requirement of Cloudlets, keeping VMs constant on 500. In the tables below $SimNo$ shows the simulation no, DC shows the data centers, CL shows the cloudlets, ET shows the execution time in ms, WT shows the waiting time in ms, φ_i represent the first threshold, φ_{ii} represent the second threshold, φ_{iii} represents the third threshold, pen shows the penalties ($) and Rev shows the revenue ($).

Table 6.5: Penalties ($) and revenue ($) calculation by running dynamic workload(cloudlet) on single data center

Sim No	DC	CL	VMs	ET	WT	φ_i	φ_{ii}	φ_{iii}	Pen	Rev
1	1	50	50	3	10	0	0	0	0	35
2	1	100	100	8	18	0	0	0	0	70
3	1	200	200	16	26	6	0	0	7	132
4	1	300	300	23	33	0	14	0	20	188
5	1	400	400	32	42	0	0	22	277	0

Table no 6.6 shows the second round results. In the third round, we used Performance based Service Level Agreement algorithms to adjust the resources according to the performance violation. Table no 6.7 shows the third experiment results.

Table 6.6: Penalties ($) and revenue ($) calculation by running dynamic workloads(cloudlet) on constant VMs

Sim No	DC	CL	VMs	ET	WT	φ_i	φ_{ii}	φ_{iii}	Pen	Rev
6	5	500	500	8	10	0	0	0	0	347
7	5	600	500	9	19	0	0	0	0	417
8	5	700	500	11	21	0	0	0	0	386
9	5	900	500	14	24	4	0	0	31	594
10	5	1100	500	17	27	0	7	0	76	688
11	5	1300	500	19	29	0	10	0	90	812
12	5	1500	500	23	33	0	13	0	104	937
13	5	1700	500	27	37	0	0	17	1180	0

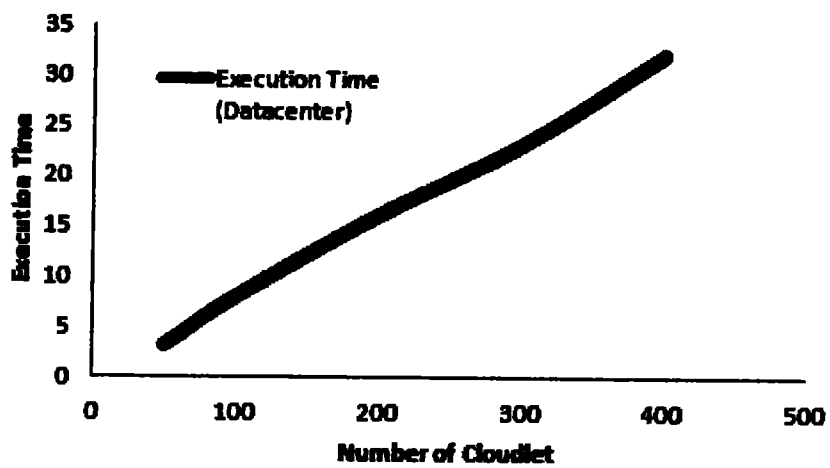


Figure 6.4: Execution time (Seconds) by running workloads (cloudlets) on constant data-centers

Figure 6.4 shows the execution time by running equal numbers of VMs and cloudlet on a single data-center. The results show that as the cloudlet reaches 200, first threshold violation happens. second threshold violation occurs on 300 cloudlets and on increasing the workload up to 500 terminate the SLA.

Table 6.7: Penalties ($) and revenue ($) calculation by running workloads (cloudlet) on VMs using Performance based Service Level Agreement algorithm

Sim No	DC	CL	VMs	ET	WT	φ_i	φ_{ii}	φ_{iii}	Pen	Rev
14	2	100	100	3	10	0	0	0	0	70
15	2	200	200	8	18	0	0	0	0	139
16	2	300	300	13	23	0	0	0	0	208
17	2	400	400	14	24	6	0	0	7	278
18	3	500	500	12	22	0	0	0	0	347
19	3	600	600	16	26	6	0	0	20	417
20	5	700	700	11	21	0	0	0	0	486
21	5	800	800	12	22	0	0	0	0	555

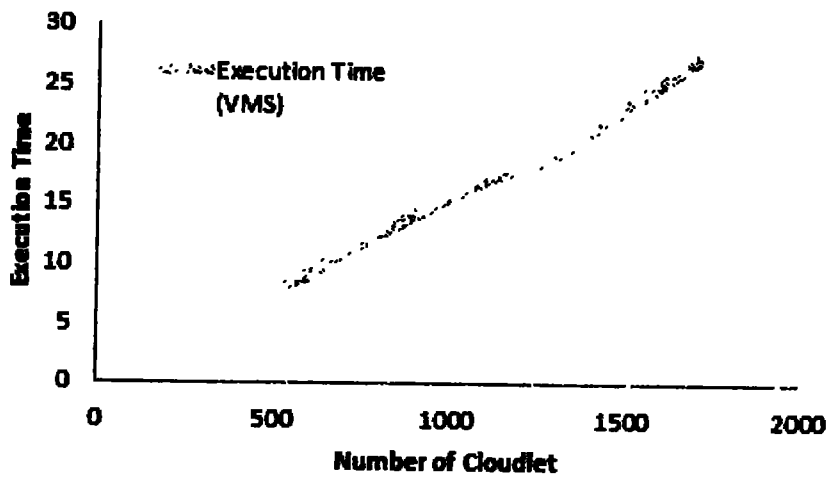


Figure 6.5: Execution time (Seconds) by running workloads (cloudlet) on constant VMs

Figure 6.5 shows the execution time by running dynamic cloudlets from 500 to 1700 keeping VMs constant on 500. Datacenters were increased as the increase in workload. Figure 6.5 shows that violation starts as the workload increases from 900 cloudlets. From 100 to 1500 cloudlets, the system remains in second threshold violation. SLA is terminated as it reaches to 1700 cloudlets.

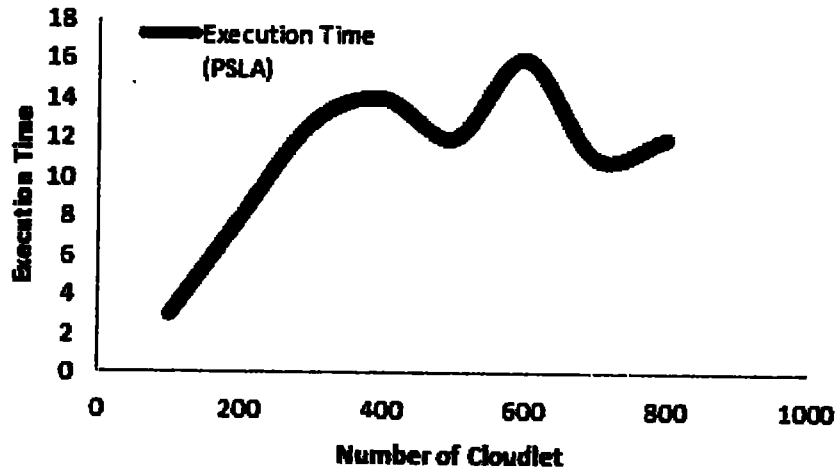


Figure 6.6: Execution time (Seconds) by running workloads (cloudlets) using PerSLA

In the third phase, the proposed algorithmic structure is used to scale the VMs and data centres as the first violation occurs. This leads to reliable services provision. We can notice in figure 6.4 that as the execution time increases from the particular point, new VMs and data centres are initialized to keep it under safe region.

The result shows that increasing the workload from a particular point on constant data-centers or VMs increases the execution time. Increase in execution time leads to SLA violations. As we assumed that the provider has scalable resources, so, as the first violation accrues, new VMs are initialized. Initialization of new VMs protects the provider from further violation. This preserves provider from SLA violations and also customers drop out.

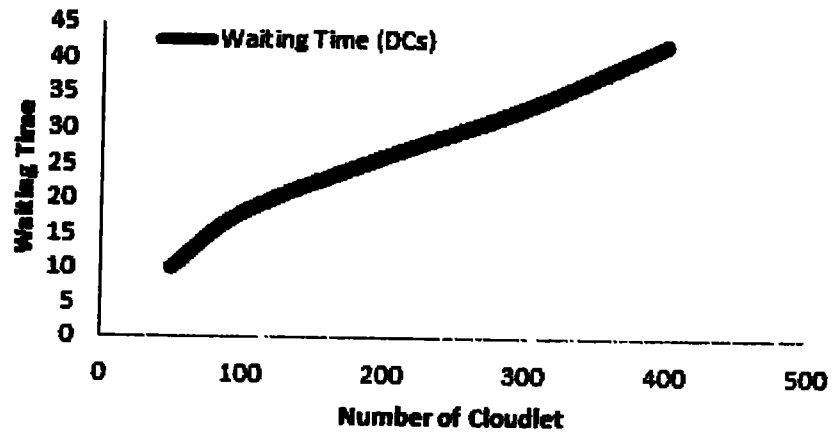


Figure 6.7: Waiting time (Seconds) by running workloads (cloudlets) on constant data-centers

Figure 6.7 explain the waiting time by running dynamic workload, dynamic VMs on a single data centre. By changing workload and VMs from 50 to 400 cloudlets shows that the waiting time changes 10, 18, 26, 33 and 44 seconds sequentially. Due to continuously increasing in waiting time, SLA is terminated.

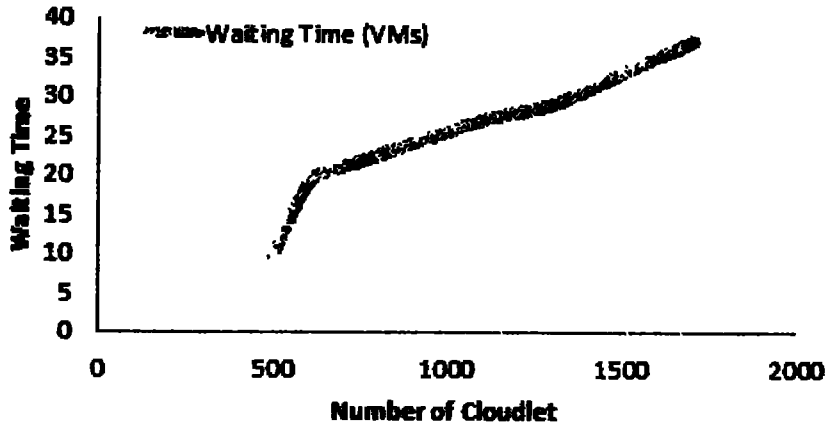


Figure 6.8: Waiting time (Seconds) by running workloads (cloudlets) on constant VMs

Figure 6.8 discuss the waiting time by running dynamic workload on constant VMs, changing the data centres according to the need. By changing workload form 500 to 1700 cloudlets shows that waiting time also increases as 10, 19, 21, 24, 27, 29, 33 and 37 seconds respectively. On more workload, SLA is terminated.

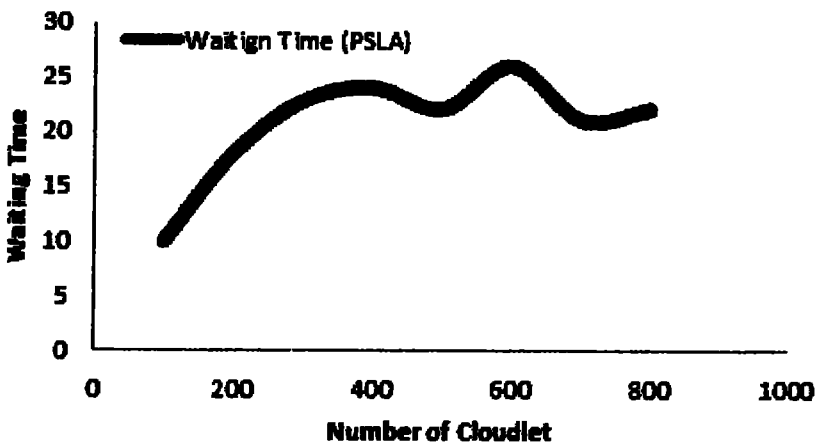


Figure 6.9: Waiting time (Seconds) by running workloads (cloudlets) using PSLA

Figure 6.9 discuss the waiting time by running dynamic workload on dynamic resources using the proposed algorithmic structure. By changing workload form 100 to 800 cloudlets, waiting time is 10, 18, 23, 24, 22, 26, 21 and 22 seconds respectively. In this run, SLA is not terminated and as you can see the Figure 6.9 of waiting time not increases much as in previous experimentation.

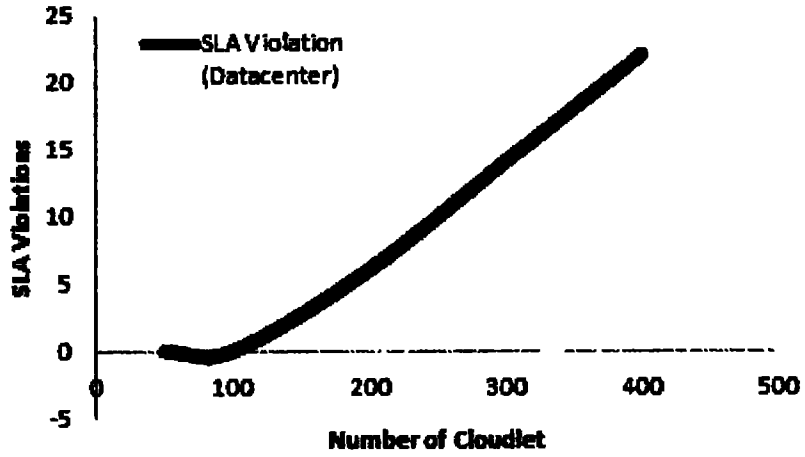


Figure 6.10: SLA violations (Seconds) by running workloads (cloudlets) on constant data-centers

To find the number of violations in total execution time, we executed the simulation for numbers of times. Figure 6.10 shows that as the cloudlets reach up to 200, first threshold violation occurs. Second threshold violation occurs on 300 cloudlets. Increasing the workload up to 100 cloudlets terminate the SLA.

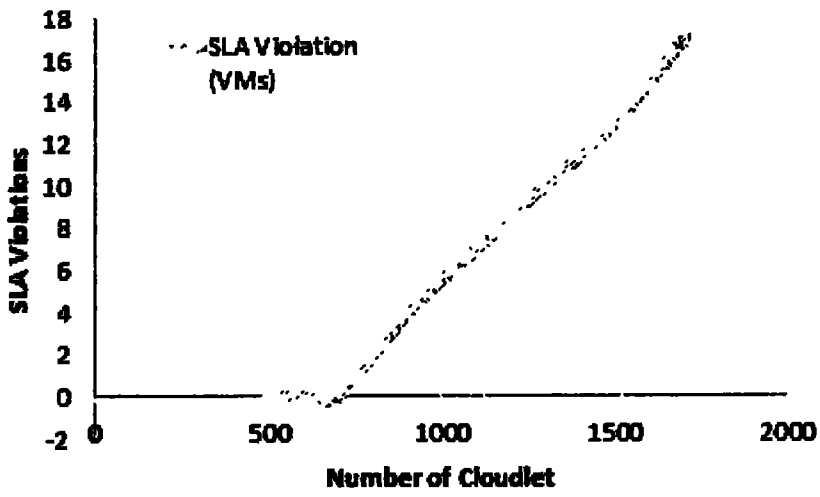


Figure 6.11: SLA violations (Seconds) by running workloads (cloudlets) on constant VMs

In the second experiment, two data centres were initiated keeping VMs constant at 500 cloudlets. Figure 6.11 shows that violation starts as the workload exceeds 900. For a workload of 100 to 1500, the system remains in second threshold violation. SLA is terminated as the workload reaches 1700.

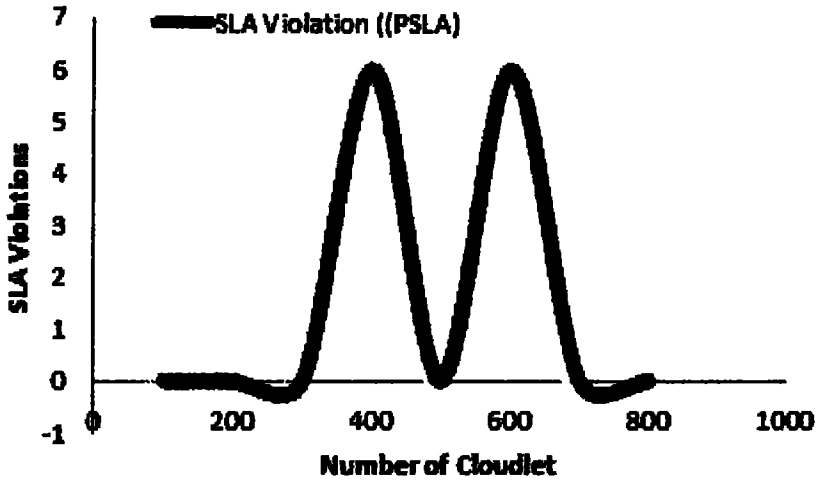


Figure 6.12: SLA violations (Seconds) by running workloads (cloudlets) using PerSLA

In the third stage, the proposed algorithms are used to scale the VMs or data-centres as the first violation occurs. It leads to reliable services provision. It can be seen from the figure 6.12 that if SLA violations exceed a certain point, new VMs and data-centres are initialized to keep it under safe region.

The result shows that increasing the workload from the particular point on constant data-centers or VMs, increases the SLA violations. As we assumed that we have scalable resources, so as the first violation accrues, new VMs are initialized. Initialization of new VMs saves the provider from further violation.

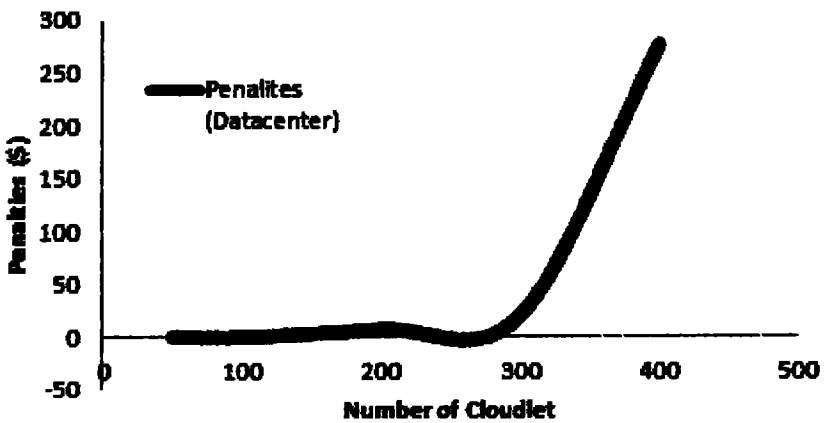


Figure 6.13: Penalties ($) by running workloads (cloudlets) on constant data-centers

Figure 6.13 discusses the penalties of the experiment by running dynamic workloads on dynamic VMs on a single data centre. We can see that as the VMs increases, penalties also increases. First penalty of $ 7 is

imposed during first threshold violation on 200 cloudlets. On 300 workload, the second threshold is violated and penalty increases to $ 20 . SLA is terminated on 400 workloads.

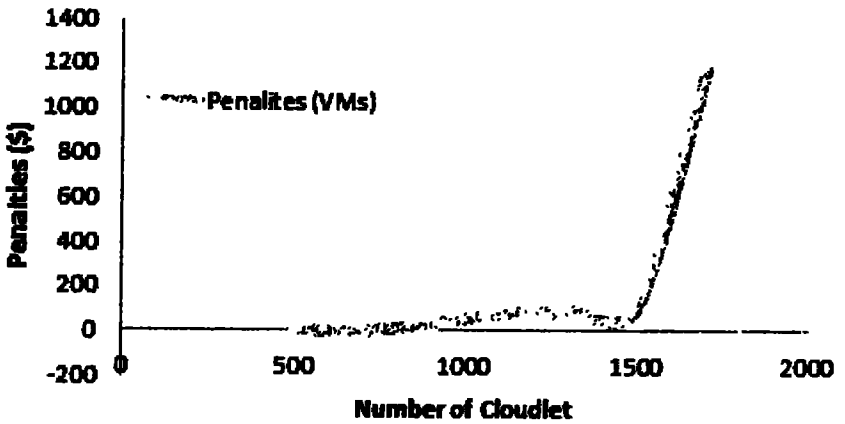


Figure 6.14: Penalties ($) by running workloads (cloudlets) on constant VMs

Figure 6.14 shows the penalties of experiments by running dynamic workloads on constant VMs of 500 and changing the data centres according to the requirements. Results show that the first threshold is violated on 900 workloads and $ 31as penalty was imposed. The system remains in second threshold violation from 1100 to 1500 having $ 76, $ 90, $ 104 penalties respectively. SLA is terminated as the workload reaches to 1700 cloudlets.

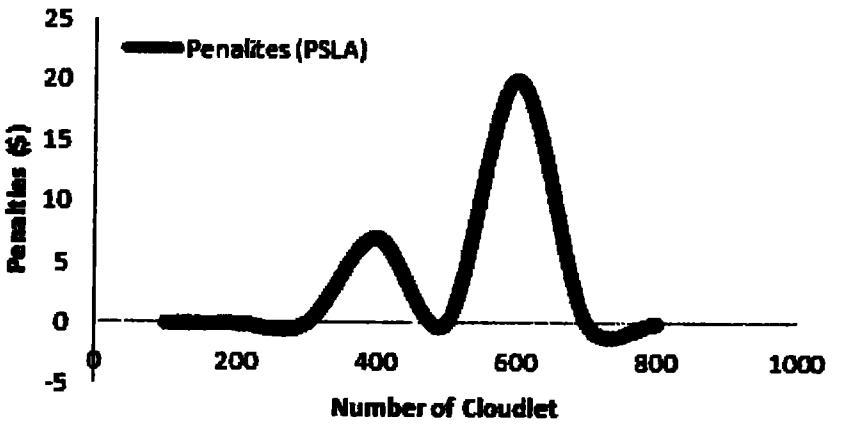


Figure 6.15: Penalties ($) by running workloads (cloudlets) using PerSLA

Figure 6.15 shows the penalties of experiments by running dynamic workloads on dynamic resources using PerSLA algorithmic structure. The result shows that as the first threshold is violated, new VMs and data

centres are initialized.

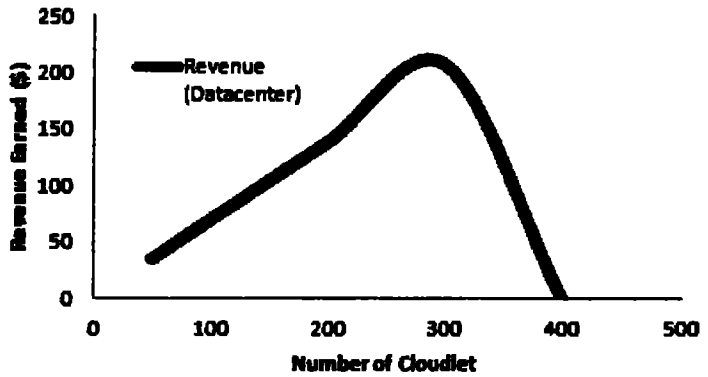


Figure 6.16: Revenue earned ($) by running workloads (cloudlets) on constant data-centers

Figure 6.16 shows the revenue earned by running dynamic workload on dynamic VMs on a single data centre. By changing workloads and VMs from 50 to 300, shows that the system earned $ 35, $ 70, $ 132 and $ 188 respectively, however, SLA is terminated as the more workload is directed toward that centre.

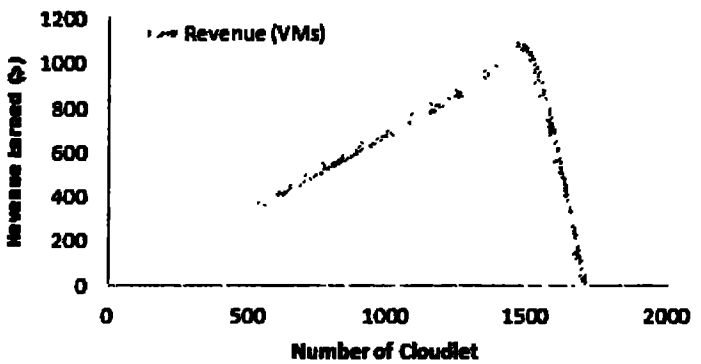


Figure 6.17: Revenue earned ($) by running workloads (cloudlets) on constant VMs

Figure 6.17 discussed the revenue earned by running dynamic workloads on constant VMs, changing the data centre according to the need. By changing workloads form 500 to 1700 shows that the system earned $ 347, $ 417, $ 486, $ 594, $ 689 and $ 937 respectively. But SLA is terminated as more workload is directed toward that centre.

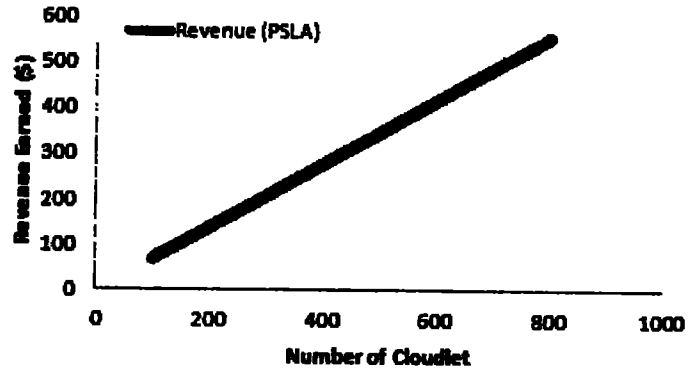


Figure 6.18: Revenue earned ($) by running workloads (cloudlets) using PerSLA

Figure 6.18 shows the revenue earned by running dynamic workload on dynamic resources using the proposed algorithmic structure. By changing workload from 100 to 800 cloudlets shows that the system earned $ 70, $ 139, $ 208,$ 278, $ 347,$ 417, $ 486 and $ 550 respectively. In this run SLA is not terminated and as you can see the revenue increases with respect to the workload.

6.3 Summary and Conclusion

Cost, performance, penalties, and revenue are very essential parameters in the cloud market. This correlation is very complex and has not been comprehensively investigated yet. The proposed framework discussed the Performance-based Service Level Agreement (PerSLA) which optimizes these parameters to one optimum point. PerSLA clearly specifies the parameters, their threshold values and penalties. Algorithms monitor the services and try to enhance the performance if it goes down. On the first two thresholds, SLA is not terminated but prices are decreased which is actually a call to the service provider to improve the performance. SLA is terminated on third threshold violations. The result shows that the proposed framework optimizes the cost, performance, penalties and revenue, also increases the customers' satisfaction. The main challenge in cloud computing business and in this article is customer satisfaction' measurement. Our future aims are to explore more customers' satisfaction measurement.

Chapter 7

Revenue Maximization by Efficient Resources Scheduling on External Resources

Cloud computing, fog and *Internet of Things (IoT)* have revolutionised the approach of communication and computation. People around the world are connected like a global village. It is expected that more than 75 billion devices will be connected to the internet up-to 2025 [2, 155]. This number is 10 times greater than the world population. This growing figure of data traffic has severe affects on performance which indirectly affects the cost and security. To optimize the performance in terms of *delay*, *running time*, *security* and *cost*, the proposed model considered three important parameters. In this article, investigations are carried out to select the target CSPs concerning its delay, running time and transfer cost. Such providers are selected, whose delay, running time and cost are less than the others. This minimizes the SLA violations and penalties.

In terms of penalties, there is a big difference between social communication and workload migration for processing. On social sites communication, an individually very little volume of data is transferred comparatively to migrating professional data. Further, in social sites, there are very less no of reserved customers and delay does not cause any penalties. In professional and reserved customers, delay affects the business badly [30]. Providers have to pay back to the customer for every violation. SLA violation may be minimized by using efficient scheduling policies to select good CSPs [107, 152].

Resources scheduling on external CSPs plays a very important role in cloud data centres [22]. A good scheduling policy maximizes resources utilization and customers satisfaction. Poor scheduling severely

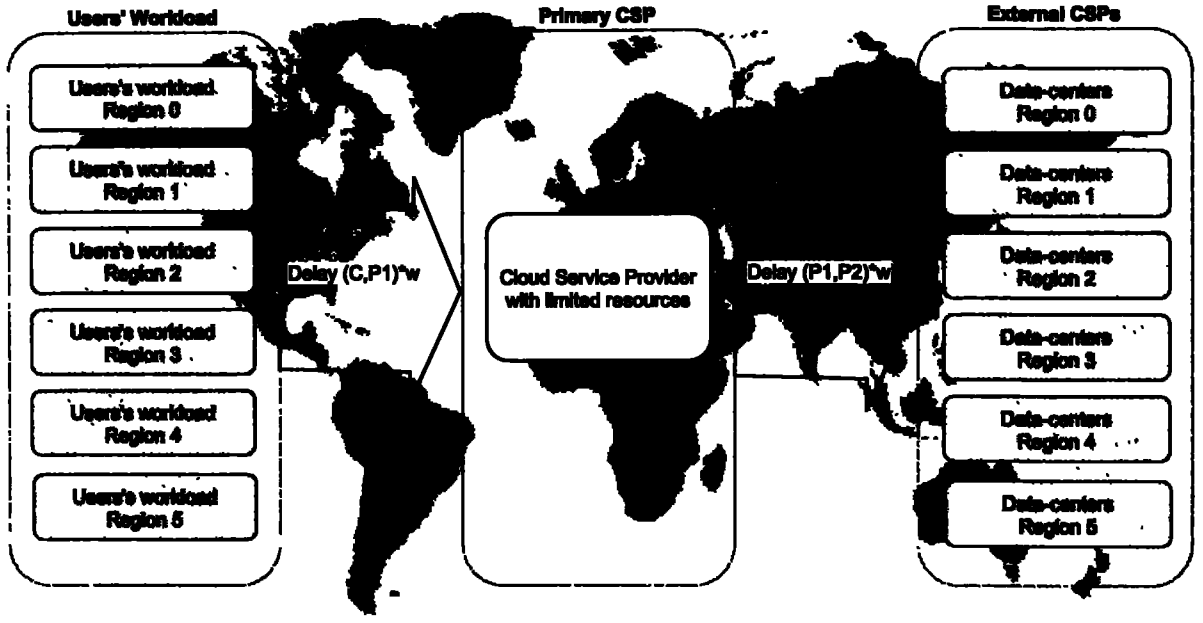


Figure 7.1: The proposed structure for delay and cost minimization

affects the performance of the services. This not only affects the performance, but it also increases the cost, energy consumption and customers dissatisfaction. These are the main reasons that scheduling policies are the primary concerns for providers [41]. A huge number of devices are connected to the internet and the cloud. This business is growing very rapidly. In such a big data, massive demands are forwarded to the cloud provider for processing. To meet the customers' demands, the proposed model depends on external CSPs. As discussed in the earlier article [146], external resources were hired to scale the provider resources. In this chapter, we discussed, how to efficiently schedule the workload towards the most suitable CSPs to run the workload with minimum delay and cost. Figure 7.1 shows the proposed structure for delay and cost minimization

Microsoft Azure is the leading cloud service provider. It also provides online services, to check the delay and uploading speed among their servers. This platform was used to calculate the file uploading and delay timing between different regions. Figure 7.2 shows the structure of this phenomenon. A 100 KB file was uploaded to different regions and upload, download and delay were calculated. Table 7.1 shows the total time taken to upload the 100 KB file to a different region and table 7.2 shows the total delay among the different regions around the world. The above Microsoft Azure results show that migrating resources to external CSPs badly decrease the performance and increase the cost [156, 157].

In this chapter, a framework is proposed (as shown in Figure 7.1) to select the optimized CSPs in terms of

Table 7.1: Upload time around the world

Source Region	Target Region	File Size	Upload Time
Pakistan, South Asia	Lowa, Central America	100 KB	3.02 sec
Pakistan, South Asia	Singapore, Southeast Asia	100 KB	2.37 sec
Pakistan, South Asia	Toronto, Canada	100 KB	4.21 sec
Pakistan, South Asia	Ireland, Europe	100 KB	3.53 sec
Pakistan, South Asia	Johannesburg, South Africa	100 KB	2.55 sec

Table 7.2: Total delay around the world

Source Region	Target Region	Delay (ms)
Pakistan, South Asia	Lowa, Central America	500 ms
Pakistan, South Asia	Singapore, Southeast Asia	297 ms
Pakistan, South Asia	Toronto , Central Canada	600 ms
Pakistan, South Asia	Ireland, Europe	365 ms
Pakistan, South Asia	Johannesburg, South Africa	422 ms

delay, running time and power consumption around the world to minimize the response time, delay, running time and cost. The proposed structure uses an algorithm which selects optimum external CSPs in terms of the above parameters.

The key contributions of this chapter are

- Efficient scheduling of customers' workload on external CSPs to minimize the providers' costs, response time, delay and running time.
- Selecting the optimum CSPs in terms of delay and energy consumption in need of external resources to maximize the services' performance.

Rest of the chapter is organised as follows. Section 2 explained hiring external CSPs and formulation to minimize the delay and running time. Section 3 presented the proposed algorithms for delay minimization, resources scheduling and migration decisions. Section 4 discussed the experimental setup and a series of

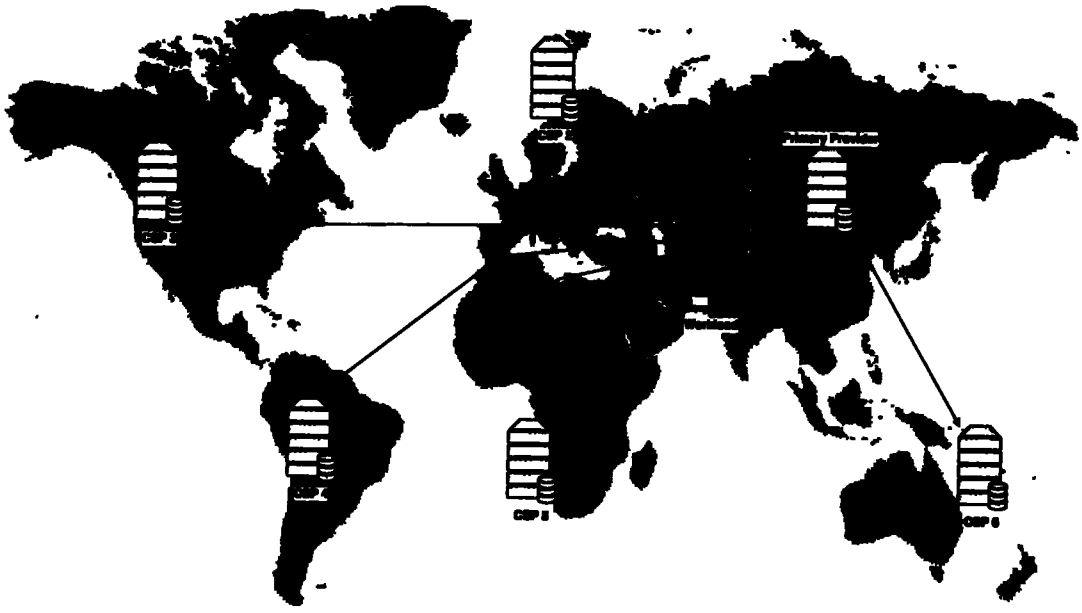


Figure 7.2: External Cloud Service Providers (CSPs) around the world

mathematical calculation and analysis. A series of measurements are conducted to confirm the domination of the proposed approach. Finally, section 5 concludes the study.

7.1 System Model

To minimize the delay, running time and cost, a framework is proposed to efficiently migrate the workloads to external CSPs. In this section, the challenges of overloaded systems and migrating workloads to external resources are investigated. The key challenges are performance degradation, cost maximization and customer dissatisfaction. Mathematical formulation and algorithm are used to deal with these challenges.

7.1.1 Overloaded CSPs

Overloaded systems not only reduce the performance which leads to penalties, but it also increases the cost in terms of electricity consumption, communication, and computation.

$$\chi \propto \gamma \tag{7.1}$$

Here χ shows the cost and γ represents the overloading of the system.

$$\tau_{run} \propto \frac{1}{\gamma} \tag{7.2}$$

Where τ_{run} is the running time

$$\gamma \propto \rho \times \lambda \tag{7.3}$$

Where ρ shows the power consumption and λ shows the heat.

Therefore, to overcome the revenue wastage in cooling the systems and other electricity consummation, proper threshold values should be selected for resources utilization and migration.

7.1.2 External CSPs

As we discussed in [146], external resources were hired to overcome the resources scalability issues. Hiring external resources increases the risks attached to external migration. It may be in terms of delay, running time, and power consummations. To overcome these issues, this model is proposed to select such external providers which minimizes all the associated challenges. The decision to migrate the workload to any other external provider is tricky. An optimized provider is searched, which will not affect the performance and cost of the primary provider.

The delay experienced by customers by migrating it to external providers are

$$D_{mig} \propto D(C, P_1) + D(P_1, P_2) \tag{7.4}$$

Where D is the delay, C is the customer, P_1 is the primary provider and P_2 is the secondary provider, from where primary provider hires resources.

Similarly, the power consumption in migration is

$$P_{mig} \propto P(C, P_1) + P(P_1, P_2) \tag{7.5}$$

Where P_{mig} is the total power consumed in migration , C is the customer, P_1 is the primary provider and

P_2 is the secondary provider, from where primary provider hires resources.

This analysis explains that searching (P_1, P_2) pair is a challenge. For external migration, primary provider searches underutilized services. Likewise, underutilized providers are in search of such customers who may utilize their resources to save the resources from wastage.

$$Per \propto \frac{1}{D} \tag{7.6}$$

Here *per* shows the performance. Performance degradation means that workload running time and response time increases

$$D \propto \nu \tag{7.7}$$

Where ν shows the computational tasks.

$$D_{total} \propto D_{com}\frac{S_{req}}{Cap} + D_{mig}(P_1, P_2) \tag{7.8}$$

Where D_{total} shows the total delay , D_{com} shows the delay cause during workload processing, S_{req} shows the total requested resources, Cap shows the total capacity of CSP and D_{mig} shows the delay cause during migration.

The above equation shows total delay depends on distance and resources availability of target servers. As we know that

$$Per \propto \frac{1}{U} \tag{7.9}$$

Where U is the resources utilization. This equation shows that such like CSPs should be searched, which are not extremely engage.

secondly,

$$Per * Cst \propto \frac{1}{D} \tag{7.10}$$

$$Rev \propto per * rel * sec \tag{7.11}$$

Where Rev is the revenue, C_{st}, is the customer satisfaction, rel, is reliability, and sec is services privacy and security. Reliability and security attract customers which increase resources utilization. In results, this improves the revenue generation.

$$CS \propto per * rel * sec \tag{7.12}$$

Where CS is the customer satisfaction.

The above mathematical formulation explains that hiring external resources is affecting the cost, performance and customer satisfaction. Instead of these concerns, we have to hire external resources due to resources scalability challenge. If long-distance and over-utilized resources are hired, it will have worse consequences on performance and cost . To overcome this issue, a geographical radius may be selected in which resources can be hired. Secondly, such CSPs are selected whose running and waiting time is lower to get good performance.

7.2 Proposed Algorithm

The above discussion and formulation shows that delay , running time and transfer cost are the major challenges to migrate the workload to the external providers. As the equations no 7.6, 7.7 and 7.9 shows that these parameters are directly proportional to target CSP and also the current workload of target CSP. The total delay faced by the customer is,

$$D_{com} = [2D_{com}(C, P_1) \times w] + [2D_{com}(P_1, P_2) \times w] \tag{7.13}$$

$$D_{proc} = D_{comp} \times w + D_{stor} \times w + k \tag{7.14}$$

$$D_{total} = D_{com} + D_{mig} + D_{proc} \tag{7.15}$$

Where w is the total workload, D_{com} is the communication delay, D_{comp} is the computational delay, D_{total} is the total delay, and k is the constant. For example the customer C is in Pakistan and the primary provider P_1 is in Tokyo, Japan then total delay $D_{com}(C, P_1)$ is 181 ms. The provider has limited resources and hire resources from London, UK. The delay between those two providers $D_{com}(P_1, P_2)$ is 217.02ms. If the workload is 100 KB than the total delay face by customer is calculated as.

$D_{total} = 2(2.37) + 2(3.53) + .5 + 0\ D_{total} = 15seconds$

The delay caused by $2D(C, P) \times w$ is not in the control of the provider. The provider may control the second, third and fourth portion of the above formula by using good scheduling polices and CSPs in the list.

As discussed in equation 7.5 that power consumption is directly proportional to distance communication and workload. Total energy consumption is calculated as.

$$P_{com} = 2P_{com}(C, P_1) + 2P_{com}(P_1, P_2) \tag{7.16}$$

$$P_{other} = P_{comp} + P_{stor} + P_{col} + k \tag{7.17}$$

$$P_{total} = P_{com} + P_{other} \tag{7.18}$$

Here P_{total} shows the total energy consumption, P_{com} shows the total energy consumption in communication. This include the wires, switches and routers etc. P_{comp} shows the total energy consumption in computation, P_{stor} shows the total energy consumption in storage and P_{col} shows the total power consumption to cool the data centers.

The proposed algorithm (algo no 7) receives numbers of users' tasks as USB, number of Cloud Service Providers as CSP, number of Data Centers as DCs and number of Virtual Machines on each data centres as VMs. In the first step, the algorithm searches the CSP for optimum cost, delay and running time. After making the optimum pair of primary and secondary provider, algorithm pop-up the available data centres of the provider. The available tasks are compared with available data centres to select the optimum data centre.

Algorithm 7 Migration decision for delay, running time and transfer cost optimization

Input: User tasks , List of VMs, List of CSPs, List of Data centers

Output: Optimized performance and cost

1: list of User Tasks: USB1, USB2.....USBn
2: List of external providers CSP1,CSP2,.....CSPn
3: **for** $intj = 1; j \leq numberCSP; j++$ **do**
4: Searching CSP having minimum prices with respect to others providers
5: **return** CSP_{min}
6: **end for**
7: List of Data Centers DC1, DC2,.....,DCn
8: **for** $intj = 1; j \leq numberDC; j++$ **do**
9: Searching DC having minimum delay and transfer cost
10: **return** DC_{min}
11: **end for**
12: Agreeing SLA
13: Starting resources
14: **if** Scheduling not successful **then**
15: Restart from step 1
16: **end if**

After selecting the optimum data centre, it compares VMs with user tasks (USB) for optimum cost, delay and running time. The algorithm ?? takes $O(n^2)$ as running time on using a nested and sequential looping structure searching optimized VMs and provider.

The algorithm calculates the expected migration cost, delay, running time and energy consumption as per the conditions of equations no 7.15 and 7.18. After searching the best pair of primary and secondary provider, SLA is agreed and initiated. It is supposed that they have already negotiated SLAs for resources sharing. After successful completion of resources scheduling, business is initiated.

7.3 Performance Evaluation

To measure the working of the proposed approach, Cloud Analyst simulator is used. It is a java based GUI simulator. It facilitates the researcher to make changes in its internal code. New algorithms may be written and existing policies may be updated according to the scenario. It is a java based command line and GUI

based simulator, which is widely used for big data simulation. It virtually creates servers, data centres and VMs around the world. Custom algorithms may be implemented to schedule different customers workload on different resources. To evaluate the execution of the proposed approach, we used this to create different data centres on a different continent. Different users' workloads are also created to run it on different cloud centre to find the best pair of (P1, P2) in terms of running time, delay and cost. We extended Cloud Analyst simulator to assess the performance of the proposed model.

7.3.1 Experimental Setup

This section furnished comparative results for uploading workload to different data centers. For this, several CSPs were created around the world in every continent. The total delay and execution time was calculated for the static workloads on these centers.

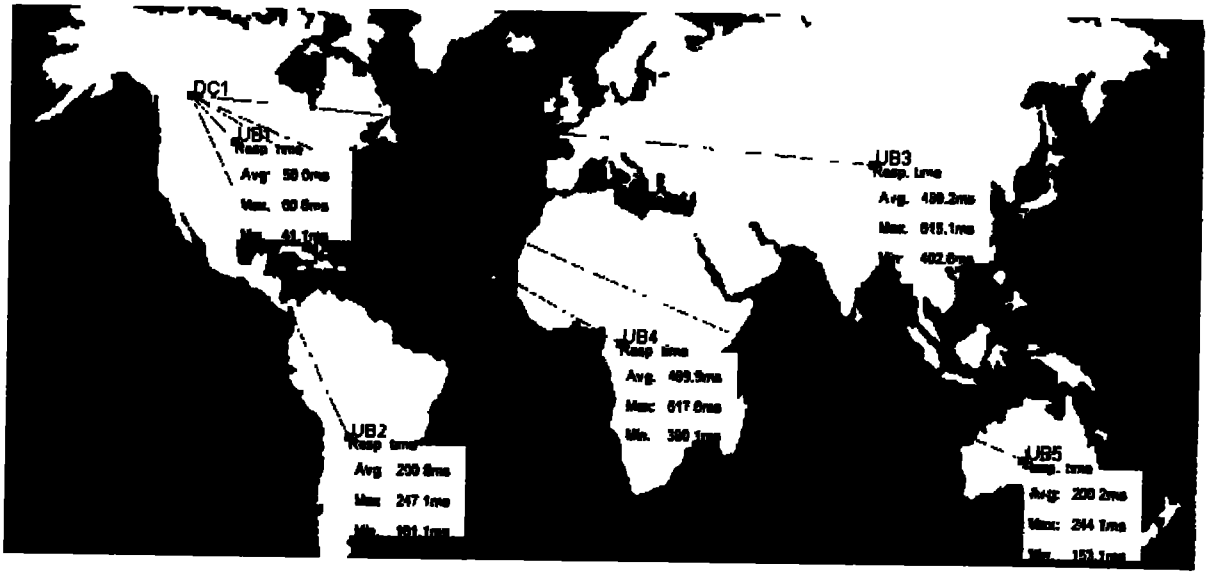


Figure 7.3: Delay, running time and transfer cost calculation for scenario 1

To calculate the delay, running time and transfer cost, three different scenarios were created having different characteristics. In the first scenario, figure 7.3, primary providers are bound to hire the resources only from one region. In the second scenario, figure 7.4, the primary provider may hire the resources from anywhere but there is no policy for optimum resources selection. In the third scenario, figure 7.5, primary providers place workload for optimum values of delay, running time and transfer cost. The scenarios are

7.3.1.1 Scenario 1

In the first scenario experimental setup, as shown in figure 7.3, numbers of data centres were created as external CSPs at Canada and 6 users' workloads in different regions, USB1 at North America, USB2 at South America, USB3 at Asia, USB4 at Africa, and USB5 at Australia. The characteristics of CSPs data-centers were that every datacenter (DC) had 204800 MB of RAM, 100000000 MB of storage, 1000000 of bandwidth, number of processors was 4, and processor speed was 10000 Memory Instructions Per Seconds (MIPS). Every data centre had 5 VMs which ran on the under-laying resources. The numbers of requests from a single USB were 60 per hour with 100000 instructions. We ran these workloads on these data centres and calculated the total delay and running time. Table 7.3 shows the total delay and execution time.

7.3.1.2 Scenario 2

In the second experimental setup, as shown in figure 7.4, several datacenters were created as external CSPs, DC1 at Canada, DC3 at South America, DC4 at Europe, DC 5 at Asia, DC6 at Africa, and DC7 at Australia. Similarly, six users' workloads were created in a different region, USB1 at North America, USB2 at South America, USB3 at Asia, USB4 at Africa, and USB 5 at Australia. The characteristics of CSPs datacenter were that every DC had 204800 MB of RAM, 100000000 MB of storage, 1000000 Mbps of bandwidth, numbers of processors were 4, and processor speed was 10000 MIPS. Every data centre had only 1 VMs available instead of 5. Numbers of requests from a single USB were 60 per hour with 100000 instructions. These workloads were run on these data centres and total delay and running time was calculated. Table 7.4 shows the total delay and execution time.

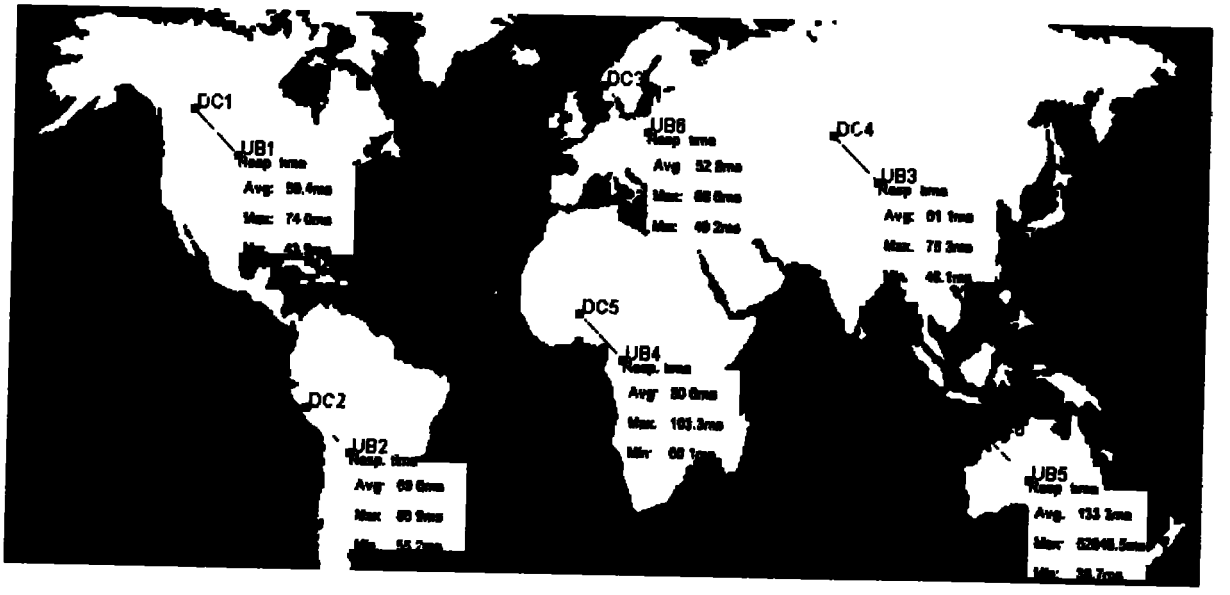


Figure 7.4: Delay , running time and transfer cost calculation for scenario 2

7.3.1.3 Scenario 3

In the third experimental setup, as shown in figure 7.5, several datacenters were created as external CSPs, DC1 at Canada, DC 3 at South America, DC4 at Europe, DC 5 at Asia, DC6 at Africa and DC7 at Australia. Similarly, six workloads were created in a different region, USB1 at North America, USB2 at South America, USB3 at Asia, USB4 at Africa, and USB 5 at Australia. The characteristics of CSPs datacenter were that every DC had 204800 MB of RAM, 100000000 MB of storage, 1000000 of bandwidth, numbers of processors were 4, and processor speed was 10000 MIPS. Every data centre had 5 VMs which was run on the under-laying resources. Numbers of requests from single USB were 60 per hour with 100000 instructions. These workloads were run on these data centres and calculated the total delay. Table 7.5 shows the total delay and execution time.

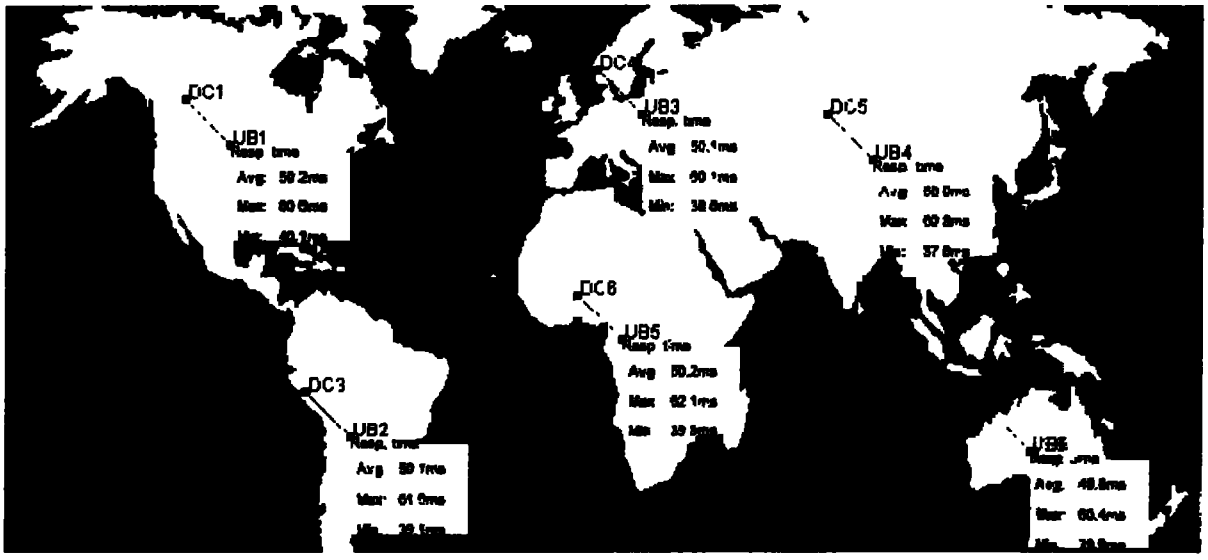


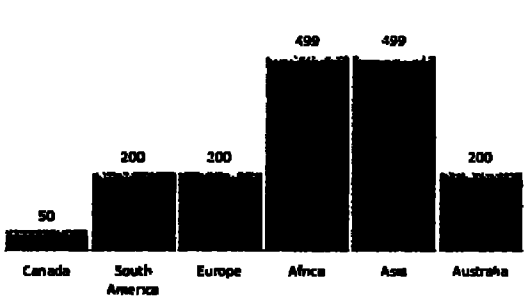
Figure 7.5: Delay , running time and transfer cost calculation for scenario 3

7.3.2 Evaluation Results

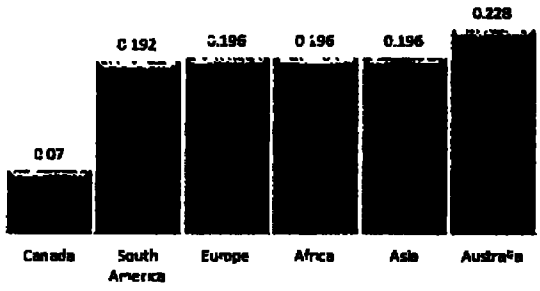
The calculated values are written in tabular form for each simulation run. In the first scenario, as displayed in figure 7.3, the workloads of 100*100 requests per minute were forwarded to the cloud provider and simulated for 60 minutes. Scenario 1 result shows the maximum delay for running workloads on Canada, and North America data centres.

Table 7.3: Delay , running time and transfer cost calculation for scenario 1

User Location	Data Center Location	Workload	Delay (ms)	Execution Time (ms)	Transfer cost ($)
North America	North America	100 * 100	50 ms	0.237	0.07
South America	North America	100 * 100	200 ms	0.237	.192
Europe	North America	100 * 100	200 ms	0.237	.192
Asia	North America	100 * 100	499 ms	0.237	.196
Africa	North America	100 * 100	499 ms	0.237	.196
Australia	North America	100 * 100	200 ms	0.237	0.228



(a) Delay for scenario 1



(b) Transfer cost for running resources on far-away data centers

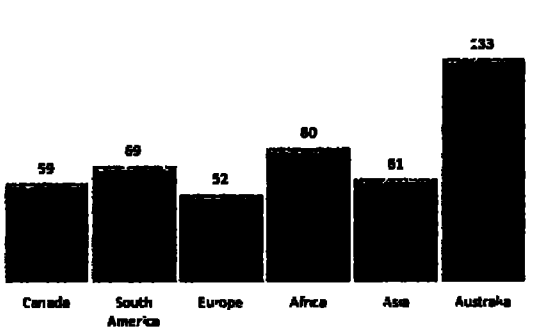
Figure 7.6: Delay and transfer cost for scenario 1

The delay to run resources at North America from Canada, South America, Europe, Africa, Asia and Australia is 50 ms, 200 ms, 200 ms, 499 ms, 499 ms and 200 ms. The results in figure 7.6a, shows that the maximum delay is 499 ms and the minimum is 50 ms. As the workloads and DCs had the same specification, therefore, the running time is constant at 0.237 seconds. The transfer cost to North America from Canada, South America, Europe, Africa, Asia and Australia is 0.07 $, 0.192 $, 0.196 $, 0.196 $, 0.192 $, and 0.228 $.

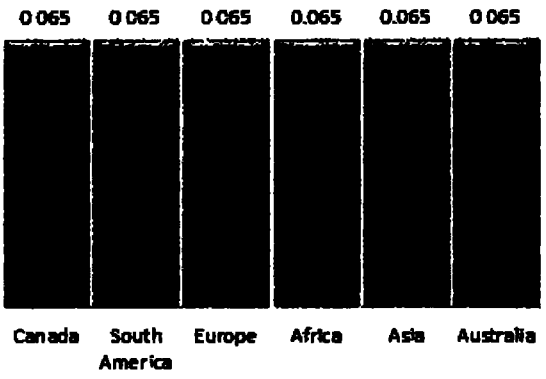
In the second scenario, as explained in figure 7.4, the closest data centres were selected for resources deployment without taking care of the optimum placement. Comparatively, to scenario 1, these data centres were busy and had only one VMs free on each data centre. The simulation was executed with the same characteristics as before. The result shows in figure 7.7a that the maximum delay is 133 ms and the minimum delay is 59 ms. The delay to run resources at Canada, South America, Europe, Africa, Asia and Australia from the local consumer was 60 ms, 55ms, 60 ms, 61 ms, 80 ms and 52 ms. As local data centres were selected,

Table 7.4: Delay , running time and transfer cost calculation for scenario 2

User Location	Data Center Location	Workload	Delay (ms)	Execution Time (ms)	Transfer cost ($)
South America	South America	100 * 100	60 ms	0.500	0.065
North America	North America	100 * 100	55 ms	0.500	0.065
Europe	Europe	100 * 100	60 ms	0.500	0.065
Asia	Asia	100 * 100	61 ms	0.500	0.065
Africa	Africa	100 * 100	80 ms	0.500	0.065
Australia	Australia	100 * 100	52 ms	0.500	0.065



(a) Delay for scenario 2



(b) Transfer cost scenario 2

Figure 7.7: Delay and transfer cost for scenario 2

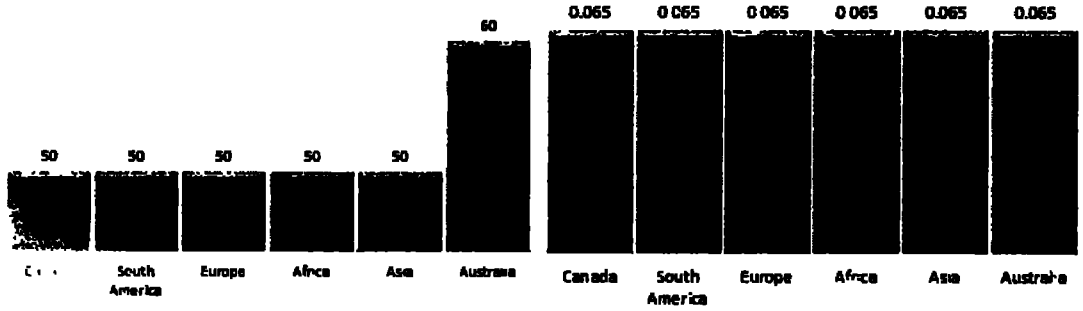
therefore, the migration cost was the same, 0.065 ms for every data centre. With the same specification, the running time was constant at 0.500 ms. Owing to busy VMs, the running time increased as compared to scenario 1, however, the delay and transfer cost decreased.

In the third scenario, as shown in figure 7.5, the same workloads were used in the same region. This time resources were run on the optimum selected data centers having free VMs and give optimum running time, delay and transfer cost. The simulation was run for 60 minutes forwarding 100*100 requests toward each data centre. The result shows in figure 7.8a that the maximum delay was 50 ms and the minimum delay was 49 ms. The delay to run resources at Canada, South America, Europe, Africa, Asia and Australia from the local consumer was 50 ms, 50 ms, 50 ms, 50 ms and 49 ms. As local data centres were selected, therefore, the migration cost was the same, 0.065 ms for every data centre. With the same specification, the running

time was constant at 0.237 ms. Owing to optimal VMs selection, the running time, delay and transfer cost decreased.

Table 7.5: Delay , running time and transfer cost calculation for scenario 3

User Location	Data Center Location	Workload	Delay (ms)	Execution Time(ms)	Transfer cost ($)
South America	South America	100 * 100	50 ms	0.237	0.065
Europe	Europe	100 * 100	50 ms	0.237	0.065
Asia	Asia	100 * 100	50 ms	0.237	0.065
Africa	Africa	100 * 100	50 ms	0.237	0.065
Australia	Australia	100 * 100	49 ms	0.237	0.065



(a) Delay for scenario 3 (b) Transfer cost scenario 3

Figure 7.8: Delay and transfer cost for scenario 3

The simulation results in figure 7.9 and 7.10 shows the comparison of scenario 1, 2 and 3. It shows that delay, and transfer time increases as the target distance increases. Second scenario result explains that delay and transfer time increases when workloads are run on the closest busy resources. In the third scenario, optimum CSPs are selected having minimum delay, running time and transfer cost.

The proposed model is tested on Cloud Analyst Simulation. Users' requests were handled on the algorithms to efficiently migrate the resources to next CSP keeping in mind running time, delay and transfer cost. Results shows a great scope for every parameter. Hence it can generate more profit and more customer satisfaction. Therefore, the proposed approach can be generalized to all the above parameters. Migration or transfer of data is the heart of IoT, fog and cloud computing. In the smart devices, usually servers are used as computational storage resources. The proposed framework can be adapted to any of such like project and migration environment.

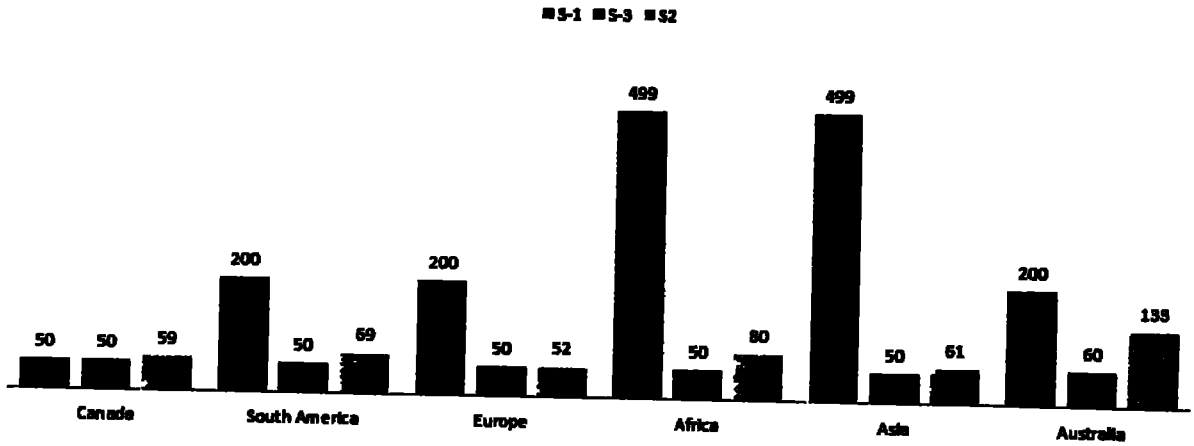


Figure 7.9: Comparative graph between three scenarios for delay

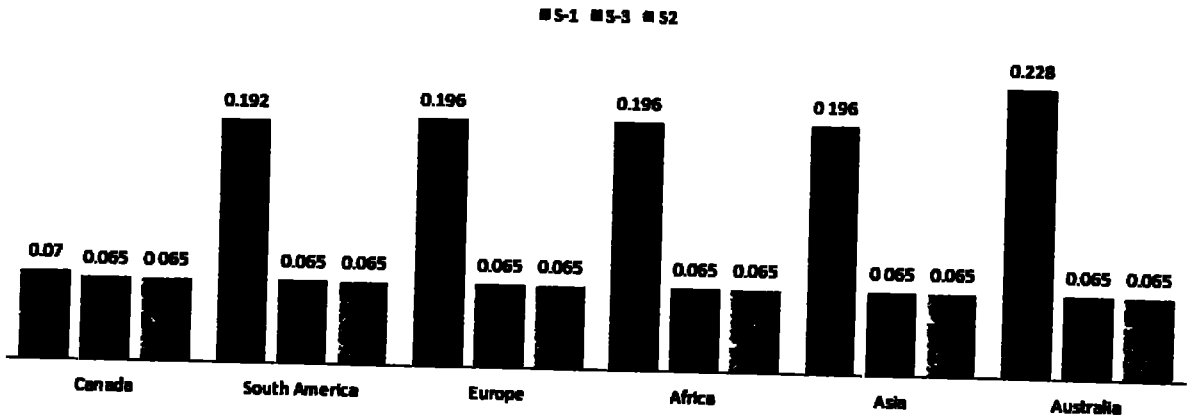


Figure 7.10: Comparative graph between three scenarios for transfer cost

7.4 Summary and Conclusion

Providers with limited resources, face many challenges. The major challenge is that the provider business remains limited. External resources and federated cloud are the solutions to handle these challenges. These are the best solutions however it gives birth to further challenges. The main challenges are the performance and cost of the services. These hurdles need to be addressed. The issue with hiring external resources is that it increases the cost in terms of energy consumption and performance degradation. To handle this issue, the proposed model select such like external CSPs for best running time, delay and transfer cost. The simulation result shows that the selection of best provider dramatically minimizes the external cost, delay and running time.

Chapter 8

Independent Monitoring Service for SLAs in Clouds

The recent integration (e.g, IoT, AI, and fog computing) to the cloud services, has completely changed the directions of this market [158, 159]. With this integration, experts believe that more than 75 billion appliances will be connected to the network by 2025 [160]. These devices will produce 175 ZB data annually [161]. The fear of cloud (in terms of security and privacy), has been dropping due to the wide study and massive cloud migration. For such a huge market, there must be a universal monitoring system (to ensue the agreed SLA) to maintain the relationship of trust among the parties. Thje table 8.1 shows the symbols and table 8.2 shows the parameters and units used in the chapter .

Cloud monitoring assess and manage the cloud services, applications and infrastructure under the terms agreed in SLA [162]. It is crucial for QoS, customer satisfaction and retention, as well as resource scaling [163]. QoS can only be guaranteed when any check is imposed on SLA which is followed up on both sides (i.e, the provider and consumer ends). Reliable monitoring develops the trust between both parties. Provider party requires monitoring to avoid the penalties and consumer party require it to monitor provider services against the agreed SLA [164].

The existing provider based monitoring frameworks (e.g, Amazon Cloud Watch [165], Paraleap Azure Watch [166], Rack Space Cloud Kick [167], etc) offer static monitoring and their services are limited to the specific providers only. *Portability*, *reliability* and *intractability* with other clouds are the main lacks in these frameworks. Some other existing monitoring frameworks (e.g, Nagios [168], Zabbix [169], Icinga [170] and Zenoss [171] etc) are general purpose frameworks, allowing administrators to monitor the servers etc, has the same limitation.

Table 8.1: Symbols and notations used in the formulation

Notation	Description	Notation	Description
D_{prov}	Total delay faced by provider	D_{cus}	Total delay faced by consumer
P	Provider	M	Monitoring party
D_{pro}	Propagation delay	C	Customer
D_{trans}	Transmission delay	D_{que}	Queuing delay
$D(C,M)$	Delay between customer and monitoring party	$D(M,P)$	Delay between monitoring and provider party
w	Monitoring overhead	ν	Monitoring frequency per 60 seconds
S_{req}	Services requested	S_{avail}	Services available
D_{total}	Total delay caused by the system	C_{total}	Total cost
C_{tran}	Transfer cost	$C(C,M)$	Transfer cost between customer and consumer
C_{tran}	Transfer cost	C_{proc}	Processing cost
C_{total}	Total cost	$Avail$	Resources availability
τ_{avail}	Up time	τ_{down}	Down time
τ_{com}	Computation time	(V_n)	SLA violation
τ_{run}	Running time	τ_{res}	Response time
ρ_{run}	Penalties of running time		
ρ_{res}	Penalties of response time	ρ_{BW}	Penalties of bandwidth
ρ_i	Initial violation threshold penalty	ρ_{ii}	Second violation threshold penalty
ρ_{ter}	Penalty of SLA termination		

To cope with the limitations related to monitoring actions performed by the provider side, *non-cooperative* strategies have also been proposed to integrate the knowledge with information gathered from customers' perspective[172]. Indeed, non-cooperative approaches were leveraged to monitor a variety of parameters mostly related to network aspects, such as cloud-to-user network throughput [173] and latency [174, 175] or network intra-cloud QoS (with focus on either inter-datacenter[176] or intra-datacenter[177] performance).

Though the Monitoring as a Service concept was coined in 2012 [178], however, this was not deeply investigated to implement.

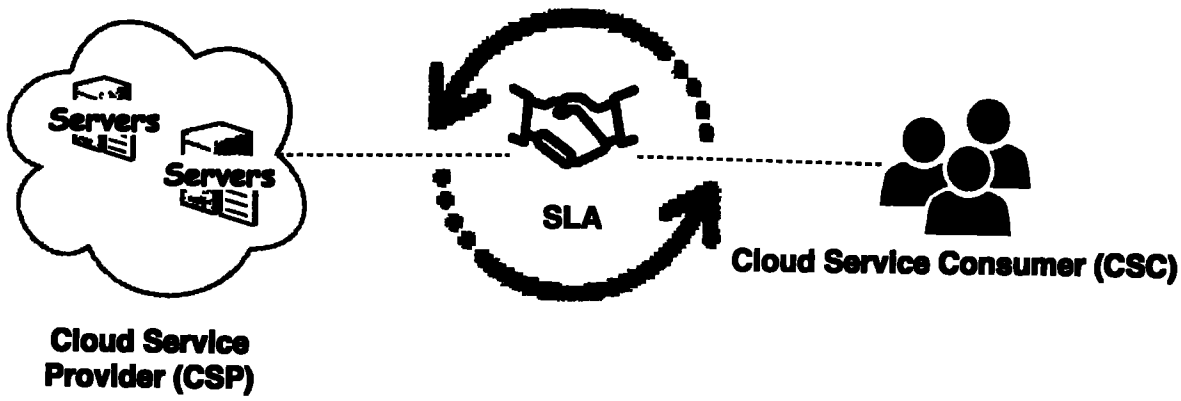


Figure 8.1: Service Level Agreement (SLA) for cloud services

Most of the aforementioned frameworks (i.e, provider based and general purpose monitoring frameworks), lacks in *portability*, *reliability* and *intractability* with other systems. These controlled frameworks do not use third-party help to monitor services against SLA which is the main reason behind un-trusted situation. Furthermore, every supplier has its monitoring framework, using its monitoring matrices that do not meet *international standards* [179]. These frameworks do not present a comprehensive solution towards trusted monitoring and penalties management. Additionally, the supplier and the consumer waste *time* and *costs* for monitoring services. From the discussion above, it is clear that clearly defined SLA, closely related to the the customer interests, is very important in helping the supplier to provide the optimum service for customer satisfaction [180].

Cloud computing needs such monitoring services that are *portable*, *reliable*, *intractable* and *internationally recognized*. Furthermore, this should be comprehensive in terms of monitoring and penalties management. This monitoring framework (i.e, SLA-MaaS) responds to all these concerns and creates a relationship of trust among partners.

The functionalities of the services coincide in the form of SLA, a legal agreement settled and endorsed by cloud partners [181]. In the context of an SLA, the Service Level Goals (SLG) are explained and the agreed Quality of Service (QoS) is fixed and signed [182]. The service are monitored according to the agreed conditions; in case of violation, the offender is penalized [183]. Figure 8.1 describes the services of SLA.

The cloud has been ending the era of desktop services and now all the computational resources are available *as a service*. Customers hesitate to trust on the providers' monitoring system. There is a gap and need of an

Table 8.2: Parameters their symbols and units used in this article

Symbol	Definition	Unit
D_{trans}	Transmission delay (Transmitting one bit to the channel)	second
D_{pro}	Propagation delay (Propagating one bit to the destination)	second
$D(C, M)$	Delay between customer and monitoring servers	second
$D(P, M)$	Delay between provider and monitoring servers	second
$D(P, M)$	Delay between provider and monitoring servers	second
τ_{res}	Response time (the time between the start and the completion of a task (in time units))	second
τ_{run}	Running time (total time a CPU spends to execute the given task)	second
BW	Bandwidth (data transfer rate through a network)	bps

independent third party monitoring framework which monitors the services as per the agreed terms. When every service in the cloud is provided "as a service" then why not monitoring (e.g, SLA-MaaS) which is the need of the day. Therefore, the main objective of *SLA-MaaS* is to offer comprehensive monitoring as an online service, just like other cloud computing services. Trust is paramount for every company. To develop trust among the parties, there must be international monitoring standards. SLA-MaaS builds trust among the parties and minimizes *implementation costs*. This service is offered as a service and paid on usage basis. The limitation of the proposed framework is its overhead, therefore, the second objective of this article is to *optimize the overhead* to optimally utilize the SLA-MaaS to the cloud platform.

In the proposed model, a third-party establish its server and starts providing monitoring services online. Before the initialization of the business, both parties (i.e, provider and consumer), submit the agreed SLA to the SLA-MaaS. It installs plugins to both ends to get real-time information of service provision and consumption. We used the Performance based SLA (PerSLA)[184], to test the working of this framework. In the case of services degradation, SLA-MaaS notifies the provider for services adjustment. On failure, fines are forced on the provider.

In this chapter, a cloud monitoring structure is proposed (*SLA-MaaS*) to provide online monitoring services by third party. The proposed structure uses third party services to monitor the SLA, agreed between both parties. For SLA, we substantially extended our previous work *PerSLA* [184]. Figure 8.2 shows the proposed structure.

The key contributions of this chapter are follows:

- *SLA-MaaS* is a framework which dynamically monitors the services at both ends. Monitoring party installs plugins (i.e, software agents) on both ends. These plugins collect the data and submit status to the SLA-MaaS server.
- A three layer penalties structure for customer attraction, satisfaction, and retention. Instead of directly jumping to the SLA termination, this start from warning and lower penalties.
- The parameters status are compared with the agreed SLA and monitoring reports are generated. The defaulter is penalized as per these reports.

The remaining of this chapter is structured as follows:

Section 8.1 explores the proposed SLA-MaaS framework, along with the proposed SLA and penalties structure; section 8.2 explains the proposed algorithms to provide online monitoring services; section 8.3 describes and analyzes the experimental setup and finally, section 8.5 concludes the work.

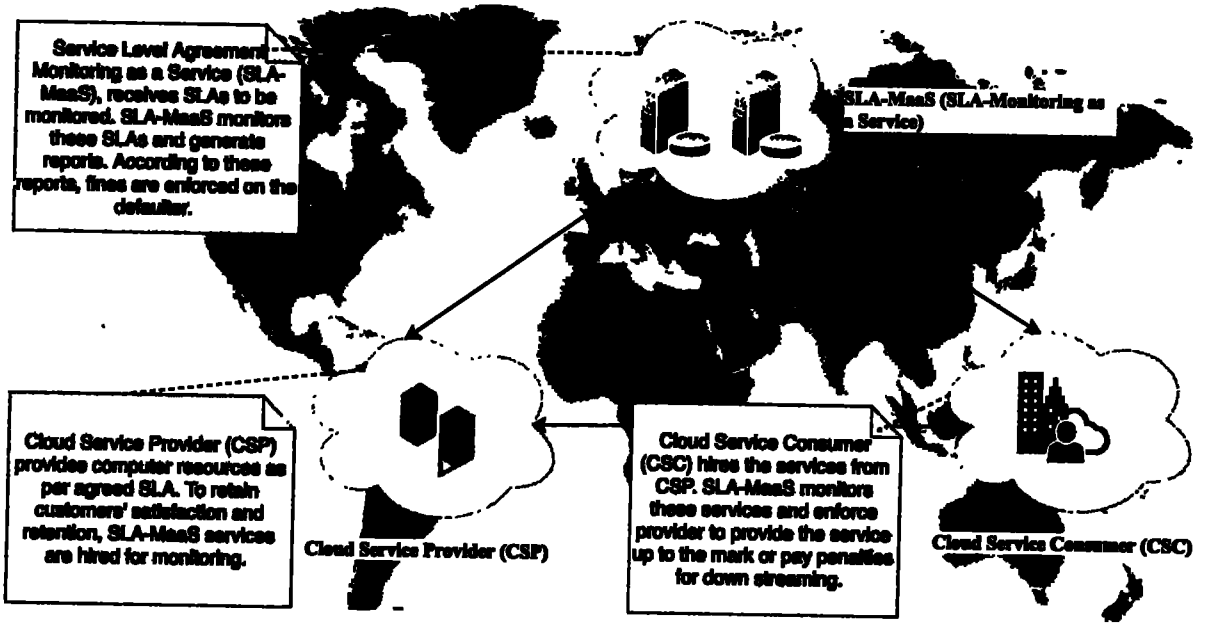


Figure 8.2: The proposed framework of Service Level Agreement-Monitoring as a Service (SLA-MaaS)

8.1 System Model

To address the aforementioned issues (discussed in overview), this section proposed a monitoring framework named as SLA-MaaS. Figure 8.2 explains the architecture of *SLA-MaaS*. Three parties participates in SLA-MaaS : (i) cloud provider, supplies the cloud services; (ii) cloud consumer, hires the services from the cloud provider and (iii) SLA-MaaS , monitors the services on both ends to ensure the agreed SLA. In the first step (as per the SLA life cycle), the consumer discover the services. In the second step, they agreed on terms and conditions and signed the SLA. In the third step, SLA-MaaS services are hired to monitor the business.

Table 8.3: Upload and delay time among different servers round the globe

Source Domain	Target Domain	Data Size	Upload Time	Delay (ms)
Pakistan, South Asia	Lowa, Central America	100 KB	3.02 sec	500 ms
Pakistan, South Asia	Singapore, Southeast Asia	100 KB	2.37 sec	297 ms
Pakistan, South Asia	Toronto, Canada	100 KB	4.21 sec	600 ms
Pakistan, South Asia	Ireland, Europe	100 KB	3.53 sec	365 ms
Pakistan, South Asia	Johannesburg, South Africa	100 KB	2.55 sec	422 ms

8.1.1 SLA-Monitoring as a Service (SLA-MaaS)

Microsoft Azure is one of the leading cloud service provider [185, 186]. For their customer satisfaction and attraction, they provide online tools to check their regional services, and the delay and uploading time among their servers. For the preliminary research, we used this framework to calculate the delay and upload time around the different regions of the world. A 100 KB data was uploaded to separate domains and upload, download and delay time were calculated. Table 8.3 shows the time it takes to upload the 100KB of data in an individual region and the total delay between regions around the world. The Microsoft Azure online tools results show that monitoring overhead decrease the performance and increase the cost [179] [187]. These articles investigated the effects of this overload on business in respect of delay time, processing cost and transfer cost.

The main challenges with the *SLA-MaaS* is the extra *delay* and *cost* of overhead. Here we calculate the delay due to overhead.

Calculating the total delay faced by provider by transferring the monitoring overhead.

In networks delay D faced by bit to transfer it from source to destination is

$$D = D_{tran} + D_{prop} + D_{proce} + D_{que} \tag{8.1}$$

Where D_{tran} shows the transmission delay, D_{pro} shows the propagation delay, D_{proce} shows the processing delay and D_{que} shows the queuing delay at the destination.

To calculate the transmission delay (D_{tran}),

$$D_{tran} = w/BW \tag{8.2}$$

where w is the workload (overhead) and the BW is the bandwidth capacity. Calculating the propagation delay ($D_{pro}4$),

$$D_{pro} = D/t_s \tag{8.3}$$

where D is the distance and the t_s is the transmission speed.

$$D_{prov} = D_{tran} + D(C, M) + D(M, P) \tag{8.4}$$

Where D_{tran} is the transmission delay to upload the overhead w, $D(C, M)$ is the propagation delay (D) between customer (C) and monitoring (M) servers and $D(M, P)$ is the propagation delay between monitoring and provider party. Table 8.2 shows the parameters and units used in this article.

The above equation 8.4 explain that the total transfer delay (D_{prov}) faced by the provider is the transmission delay and the propagation delay between customer (C) and monitoring party (M) and then transferring the report to the provider party (P) from monitoring party (M).

Delay faced by the customer

$$D_{cus} = D_{tran} + D(C, M) + D(M, C) \tag{8.5}$$

The above equation 8.5 explain that the total transfer delay (DT_{cus}) faced by the customer is the transmission delay and delay between customer (C) and monitoring party (M) and then transferring the report to the customer party (C) from monitoring party (M).

The total processing delay is calculated as:

$$D_{proc} \propto w \times \nu \tag{8.6}$$

Where w is the monitoring overhead and ν represents the sending frequency in 60 seconds.

Therefore the total delay faced by the provider or consumer is:

$$DT_{cus} = D_{cus} + D_{proc} \tag{8.7}$$

$$DT_{prov} = D_{prov} + D_{proc} \tag{8.8}$$

Where D_{proc} is the processing delay, DT_{cus} total delay faced by customer and DT_{prov} is the total delay faced by the provider.

For instance, the consumer C is in Pakistan and the SLA-MaaS server M is in Tokyo, Japan then the acquired delay $D(C, M)$ is 180 ms. Where, the provider is in London, United Kingdom. The delay between those two locations is $D(P, M)$ is 220.02 ms. If the workload is 100 KB then the total delay faced by the system is calculated as;

$D_{total} = 180ms + 220ms + 500ms$
$D_{total} = 0.9second$

where 500 ms is processing time.

The total cost consumed on the monitoring overhead transfer is calculated as:

$$C_{tran} = C(C, M) \times w + C(M, P) \times w + C(M, C) \times w \tag{8.9}$$

Where $C(C, M) \times w$ is the transfer cost between customer and SLA-MaaS and $C(M, C) \times w$ is the transfer cost between SLA-MaaS and customer.

The above equation 8.9 shows the total cost by transferring the monitoring overhead. The total cost (*cost*) is equal to the cost among (C, M), (M, P) and (M, C).

The total processing cost is calculated as:

$$C_{proc} = w \times \nu \tag{8.10}$$

Where w is the monitoring overhead and ν represents the sending frequency in 60 seconds.

Therefore the total overhead cost is:

$$C_{total} = C_{tran} + C_{proc} \tag{8.11}$$

For instance, the consumer C is in Pakistan and the SLA-MaaS M is in Tokyo, Japan then total transfer cost is $C(C, M)$ is 0.5 \$. The provider is in London, United Kingdom (UK). The cost between those two location is $C(P, M)$ is 0.5 \$. If the load is 100 KB then the total cost is calculated as;

$C_{total} = 0.5\$ + 0.5\$ + 1\$$
$C_{total} = 2\$$

where 1\$ is processing cost.

The inner complexity of the SLA-MaaS is encapsulated in Application Programming Interface (APIs) which facilitates the other applications or users to communicate with the system. The cloud consumer/provider or any other host interacts with SLA-MaaS by API for registering or submitting the respective SLA's.

SLA-MaaS uses agents that autonomously collect data from provider and consumer sides. Agents perform three types of tasks; collecting, processing and reporting. Agents are installed on VM and they are initiated when VM starts services provision. Agents monitor and collect the data and send the monitoring report to the repository. It monitors the specific matrix (i.e, CPU, utilization, storage, network bandwidth) and report any failure to agent's coordinator.

SLA repository is a database in SLA-MaaS that contains SLAs, submitted by cloud provider and consumer. It also store the SLA, established among SLA-MaaS , cloud provider and consumer. When monitoring is

performed, the running SLA is the SLA, submitted by the provider and consumer.

Monitoring repository is the database that stores the data about metrics to be monitored specified by party through SLA, collected by agents and received through plugins. It also stores the monitoring result. Penalty repository stores all the records of penalty imposed on the providers.

8.1.2 Service Level Agreement (SLA)

SLA is the treaty (agreed within the cloud partners) and monitored by SLA-MaaS. To evaluate this framework, we extended PerSLA [184] for algorithm initialization and monitoring. Fines are imposed on the supplier to satisfy customers. These fines can be in cash or reduction in prices. In this paradigm, rates are reduced for downtime. The three-layer fine formation is used for this SLA. Table 8.4 present the threshold conditions employed in this investigation. Symbols first threshold (T_i), second threshold (T_{ii}), and third threshold (T_{iii}) describes the threshold.

Table 8.4: Threshold Values for Service Level Agreement (SLA).

SLA metric	T_i	T_{ii}	T_{iii}
Response time (sec)	2 sec	5 sec	10 sec
Execution time (sec)	3 sec	7 sec	15 sec
Availability (%)	99 %	97 %	95 %
Bandwidth (Mbps)	9.9	9.8	9.5

Execution time is the time to respond to customers' request. Execution time is also known as running time. The execution time is a crucial criterion of SLA. It depends on the nature of the request and the services on which the workload is deployed. In case of not suitable resources, it gets longer than usual [188].

$$\tau_{run} \propto Res_{type} \times Req_{type} \tag{8.12}$$

$$\tau_{run} = \tau_f - \tau_i \tag{8.13}$$

The above equation explains that running time (τ_{run}) depends on resources type (Res_{type}) and request type (Req_{type}). Where running time is calculated by subtracting task starting time from task finish time.

Response time refer to the (waiting) time for the consumer's demand in the line. It depends on the availability and bandwidth of the sources. If the resources are used intensively, it takes longer to complete new tasks [189].

$$\tau_{res} \propto SS \times v \tag{8.14}$$

$$\tau_{res} = \tau_i - \tau_s \tag{8.15}$$

The above equation explains that response time (τ_{res}) depends on services scalability (SS) and services utilization (V). Where response time is calculated by subtracting task submission time by task starting time.

Availability describes the availability of agreed services when required. Availability focused on resources considered and agreed in the SLA [189].

The availability is calculated by equation no 2.6 where τ_{com} denotes the computation time agreed in SLA, τ_{avail} represents the computation time availability and τ_{down} represents the execution downtime.

In case of resources nonscalability, execution and response time escalate. The scalability factors depends on resources availability of the provider. SLA break (V_n) accures due to services nonscalability. The resources scalability affect the business in terms of SLA violation, penalties, lower performance and customers' dissatisfaction.

Reliability means the performance of the resources, compared to the predefined agreed conditions. The resources, supporting fault tolerance and automatic recovery are considered as reliable resources. lower reliability leads to revenue reduction [189].

8.1.3 Penalty Structure

On violation of SLA, fines are imposed on the supplier, which has a negative impact on the company. Usually, at the start of the business, suppliers get heavy workloads, however, later on they fails to deliver the agreed resources, which leads to fines. We extended our previous penalty structure PerSLA [184] for the SLA-MaaS penalty.

To have a limit check on fines, this should not surpass 10% of the charges [190]. In case of increase by 10% , SLA should be terminated. This assist in maintaining business between partner. Furthermore, if fines crosses the mentioned limit, this will have a very bad impact on the supplier's business [189].

Algorithm 8 SLA-Monitoring as a Service (SLA-MaaS)

Input: SLA parameters , Penalties, CSP, MaaS and Consumer
Output: Monitoring report

SLA REQUESTS
DEFINING CRITERION(τ_{run},τ_{res}, throughput)
CRITERION DESCRIPTION(Function, Units, and Metrics)
FINE DEGREES(1, 2, 3)
FINE STRUCTURE(2 %, 5 %, SLA cancellation)

if $\tau_{run} \leq T_i$ **then**
$\rho_{run} = 1, \rho_{status} = yes$
end if
if $T_i < \tau_{run} \leq T_{ii}$ **then**
$\rho_{run} = 2, \rho_{status} = yes$
end if
if $T_{ii} < \tau_{run} \leq T_{iii}$ **then**
$\rho_{run} = 2, \rho_{status} = yes$
end if
if ρ_{status} **then**
τ_{run} = calculating execution time penalties using equation no 8.16
τ_{res} = calculating response time penalties using equation no 8.17
τ_{avail} = calculating memory and storage availability penalties using equation no 8.18
τ_{BW} = calculating network availability penalties using equation no 8.19
end if
Streaming data to both ends
$\rho_{total} = \rho_{run} + \rho_{res} + \rho_{avail} + \rho_{BW} + k$ using equation no 8.19
Notifying provider and customer
Enforcing penalties

Penalties can be paid in lower prices or in cash. The worst thing about the SLA violation is the challenges to the company's reputation and future. The customers never relies on faulty suppliers [180]. We used the penalties structure, as shown in table 8.5 from our last study, PerSLA [191].

As explained in Table 8.5 , on violation of the first threshold, prices are reduced by ρ_i; if the second threshold is broke, rates are reduced by ρ_{ii}, and if performance decreased by third check, SLA is terminated. Fines for SLA violation is automatically calculated, in case of any issue, claim may also be registered with SLAM-aaS.

Running Time: For running time (τ_{run}), three layers threshold values are declared, used in our publication [184]. If the resources running time τ_{run} is lower than the first threshold T_i, no fines are applied. However, if the running time crosses the first threshold T_i or the second threshold T_{ii}, supplier is penalized by ρ_i and

Table 8.5: Penalties structure for SLA violation

SLA metric	(ρ_i)	(ρ_{ii})	(ρ_{ter})
Response time	5%	10%	SLA Termination
Execution time	5%	10%	SLA Termination
Availability	5%	10%	SLA Termination
Bandwidth	5%	10%	SLA Termination

Table 8.6: Calculating the overhead delay for the situation 1.

User Location	Data Center Location	Overhead	Sim time	D_{CP}	D_{proc}	D_{total}
North America	Australia	100KB	1	195	0.11	195
North America	Australia	100KB	2	197	0.11	197
North America	Australia	100KB	3	187	0.09	187
North America	Australia	100KB	4	204	0.09	204
North America	Australia	100KB	5	204	0.09	204
North America	Australia	100KB	6	206	0.11	206
North America	Australia	100KB	7	204	0.11	204
North America	Australia	100KB	8	205	0.11	205
North America	Australia	100KB	9	205	0.11	205
North America	Australia	100KB	10	203	0.11	203

ρ_{ii} respectively. Business is cancelled if running time increases by ρ_{iii}.

$$\rho_{run}(x) = \begin{cases} 0 & \tau_{run} \leq T_i \\ \rho_i & T_i \geq \tau_{run} \leq T_{ii} \\ \rho_{ii} & T_{ii} \geq \tau_{run} \leq T_{iii} \\ S_{ter} & \text{otherwise} \end{cases} \tag{8.16}$$

Response Time: If the resources response time τ_{res} is below the first threshold T_i, resources supplier gets full prices without any penalty. However, if the response time τ_{res} increases by the second threshold T_{ii} or

Table 8.7: Simulation results for the overhead cost for Scenario 1.

User Location	Data Center Location	Overhead	Sim time	C_{CP}	C_{proc}	C_{total}
North America	Australia	100KB	1	0.02	0.5	0.52
North America	Australia	100KB	2	0.05	1.09	1.14
North America	Australia	100KB	3	0.06	1.54	1.6
North America	Australia	100KB	4	0.09	2.05	2.14
North America	Australia	100KB	5	0.12	2.55	2.67
North America	Australia	100KB	6	0.17	3.05	3.22
North America	Australia	100KB	7	0.20	3.56	3.76
North America	Australia	100KB	8	0.22	4.08	4.3
North America	Australia	100KB	9	0.25	4.5	4.75
North America	Australia	100KB	10	0.29	5.07	5.36

third threshold T_{iii}, supplier is penalized by ρ_i and ρ_{ii} respectively. SLA is terminated if running time τ_{run} crossed the third threshold ρ_{iii}.

$$\rho_{res}(x) = \begin{cases} 0 & \tau_{res} \leq T_i \\ \rho_i & T_i < \tau_{res} \leq T_{ii} \\ \rho_{ii} & T_{ii} < \tau_{res} \leq T_{iii} \\ \rho_{ter} & \text{otherwise} \end{cases} \tag{8.17}$$

Availability: Resource availability means resources provision under the terms of the SLA if required. It includes storage, memory and bandwidth, etc. In the equation below, σ shows the resources availability .

$$\rho_{\sigma}(x) = \begin{cases} 0 & \sigma \geq T_i \\ \rho_i & T_i < \sigma \geq T_{ii} \\ \rho_{ii} & T_{ii} < \sigma\tau_{res} \geq T_{iii} \\ \rho_{ter} & \text{otherwise} \end{cases} \tag{8.18}$$

Bandwidth: The bandwidth three thresholds are follows: if the bandwidth BW break is below the first threshold T_i, no fines are imposed on the supplier, however, if the bandwidth BW exceeds by second threshold T_i and third threshold T_{ii}, supplier is penalized by ρ_i and ρ_{ii} respectively. SLA is aborted if bandwidth BW breakage passes by ρ_{iii}.

$$\rho_{BW}(x) = \begin{cases} 0 & BW \geq T_i \\ \rho_i & T_i < BW \geq T_{ii} \\ \rho_{ii} & T_{ii} < BW \tau_{res} \geq T_{iii} \\ \rho_{ter} & \text{otherwise} \end{cases} \tag{8.19}$$

Penalty is calculated as follows;

$$\rho = \sum_{k=0}^{n} (vp \times \rho_{rate} \times (\tau_{avail} - \tau_{dt})) \tag{8.20}$$

8.2 Proposed Algorithms

For the algorithm 8, we extended the *Performance bases Service Level Agreement (PerSLA)* [184] structure as agreed SLA among parties. PerSLA uses three threshold values to ensure the optimum performance and cost.

The proposed algorithm works as follow:

- Cloud provider and consumer register to SLA-MaaS server interface and submit a request for SLA monitoring (line 1).
- Joining SLA-MaaS means that cloud partners (i.e, provider and consumer) agree with SLA-MaaS terms and conditions and another SLA places among these three parties (i.e, provider, consumer and SLA-MaaS). This SLA terminates when the monitoring process completes or one of the party violates. The agreed terms and conditions along with penalties are submitted to SLA-MaaS (line 2-5).
- Monitoring party start monitoring of specified metrics that have been agreed (i.e, CPU usage, storage, memory and network etc) (line 6-16). The monitoring data is send back to the monitoring repository.

Table 8.8: Calculating the overhead delay for scenario 2 (SLAMaaS).

User Location	MaaS Location	Data Center Location	Overhead	Sim time	D_{CM}	D_{MP}	D_{proc}	D_{total}
Australia	Europe	North America	100KB	1	195	98	0.1	293
Australia	Europe	North America	100KB	2	197	99	0.1	296
Australia	Europe	North America	100KB	3	187	94	0.1	281
Australia	Europe	North America	100KB	4	204	103	0.1	306
Australia	Europe	North America	100KB	5	204	102	0.1	306
Australia	Europe	North America	100KB	6	410	299	0.1	309
Australia	Europe	North America	100KB	7	206	103	0.1	306
Australia	Europe	North America	100KB	8	204	103	0.1	308
Australia	Europe	North America	100KB	9	205	102	0.1	308
Australia	Europe	North America	100KB	10	205	301	0.1	305

Table 8.9: Simulation results for the overhead cost for Scenario 2

User Location	MaaS Location	Data Center Location	Overhead	Sim time	C_{CM}	C_{MP}	C_{proc}	C_{total}
Australia	Europe	North America	100KB	1	0.04	0.04	0.06	0.14
Australia	Europe	North America	100KB	2	0.05	0.05	1.09	1.19
Australia	Europe	North America	100KB	3	0.11	0.11	1.16	1.38
Australia	Europe	North America	100KB	4	0.17	0.17	2.05	2.39
Australia	Europe	North America	100KB	5	0.25	0.25	2.55	3.05
Australia	Europe	North America	100KB	6	0.34	0.34	3.05	3.73
Australia	Europe	North America	100KB	7	0.40	0.40	3.56	4.36
Australia	Europe	North America	100KB	8	0.44	0.44	4.08	4.96
Australia	Europe	North America	100KB	9	0.50	0.50	4.50	5.50
Australia	Europe	North America	100KB	10	0.57	0.57	5.07	6.21

It processes this data and summarizes it. In graphical form, this data presents the results and reports of the monitoring on both sides (line 19).

- In case of violation, SLA-MaaS enforces penalties on defaulter and makes sure penalties payment (line 20).

The algorithm 8 takes $O(n)$ as running time for monitoring the services.

8.3 Evaluation

To evaluate the operation of the proposed scheme, the Cloud Analyst simulator [192] was used. Cloud Analyst is often used for simulation in the cloud and virtually formulates servers, data centres and virtual machines around the world. Custom algorithms can be coded to implement customer load to different

sources. To assess the performance of the proposed strategy, we created several data centres on different continents. Multiple client workloads were also formulated to run on separate cloud centres to find the best combination of (P1, P2) in terms of delay, execution time, and transfer costs.

8.3.1 Experimental Setup

For the proposed structure, the experimental setup was created in three scenarios. In the first scenario, the traditional monitoring system was formed; in the Scenario 2, SLA-MaaS structured was created; and in the Scenario 3, Scenario 2 is extended by reducing monitoring data frequency.

Situation 1: In situation 1, experimental structure, Data Center (DC) was created in North America (Region 1). The user task (USB 1) was created in Australia (Region 5). The characteristics of Data Center (DC), that each DC had 204800 MB of RAM, 100000000 MB of storage, 1000000 of bandwidth, 4 processors, having speed of 10000 Memory Instructions Per Seconds (MIPS). DC had 5 VMs, running on the under-laying resources. 30 requests per minute with 100000 instructions were received by a single USB. We forwarded the monitoring overhead as 100 KB per 2 seconds. We ran this experiment for an hour on these data centres and calculated the overall delay, transfer cost,VMs cost and response time. Tables 8.6 and 8.7 show the detail findings.

Situation 2: In the Scenario 2 experimental structure, Data Center (DC) was created at North America (Region 1). The user task (USB 1) was created at Australia (Region 5). SIA-Monitoring as a Service (SLA-MaaS) server was created at Europe (Region 2). The characteristics of Data Center (DC), that each DC had 204800 MB of RAM, 100000000 MB of storage, 1000000 of bandwidth, 4 processors, having speed of 10000 Memory Instructions Per Seconds (MIPS). DC had 5 VMs, running on the under-laying resources. 30 requests per minute with 100000 instructions were received by a single USB. We forwarded the monitoring overhead as 100 KB per 2 seconds and ran this experiment for 1 to 10 hour on these data centres and calculated the delay, transfer cost,VMs cost and response time. Tables 8.8 and 8.9 shows the detail findings.

Situation 3: The Scenario 3, actually extend the Scenario 2 and has the same specification except the monitoring overhead and sending frequencies. The data overhead was reduced to 50 KB instead of 100 KB, similarly, the sending frequency was reduced to 15 per minute instead of 30 per minute.

Table 8.10: Criteria for studies (literature) assessment.

Symbol	Criteria	Criteria Definition
C1	Reliability	The monitoring framework is increases the customer trust by providing independent monitoring (i.e, third party independent monitoring).
C2	Scalability	Monitoring platform may easily be scaled to add more business SLAs for monitoring and penalties enforcement.
C3	Interoperatability	The monitoring framework is able to operate with different hardware and operating systems in different environment.
C4	Agent based	The monitoring party installed agent into the target computers and servers to transparently monitor the services provisions.
C5	Multi-clouds	The framework is able to work with multi cloud providers and cloud setup (i.e, the underlying structure of resources provision).

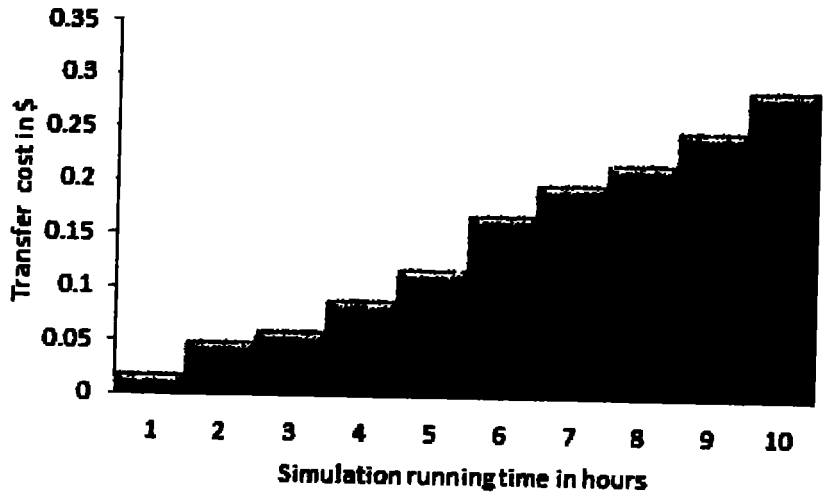


Figure 8.3: Transfer cost for Scenario 1

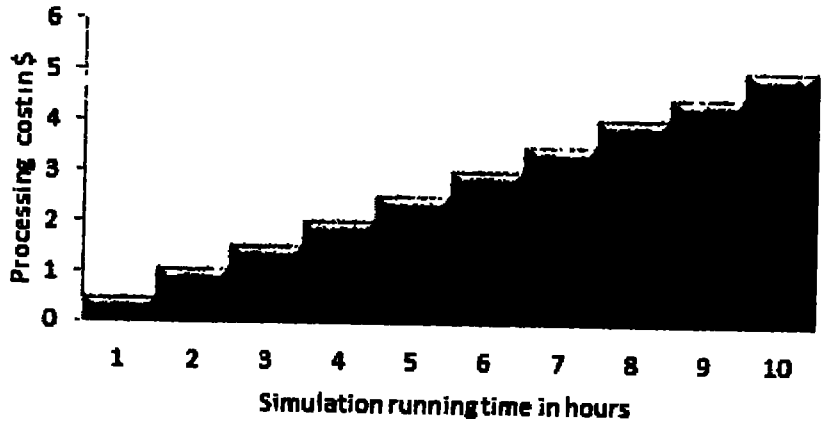


Figure 8.4: Processing cost for Scenario 1

8.3.2 Evaluation Result

The results of the above discussion are organized in the tabular form for each simulation run.

In the situation 1, the workload of 100 KB requests per 10 seconds was routed to the provider and simulated for ten times (1 to 10 hours). The average delay (as shown in Table 8.6) for directly forwarding the file to the cloud provider was 201 ms; the average processing delay was 0.11 ms and the average delay faced by the consumer or provider was 201 ms. The transfer cost (as shown in Table 8.7 and Figure 8.7) for each simulation run is 0.02 $, 0.05 $, 0.06 $, 0.09 $, 0.12 $, 0.17 $, 0.20 $, 0.25 $, and 0.29$ respectively; similarly the processing cost (as shown in Figure 8.8) is 0.5 $, 1.09 $, 1.54 $,2.05 $, 2.55 $, 3.05 $, 3.56 $, 4.08 $, 4.50 $, and 5.07$ respectively. Therefore the total cost (processing and transfer) of each hour is 0.52 $, 1.14 $, 1.6 $, 2.14 $, 2.67 $, 3.22 $, 3.76 $, 4.3 $, 4.75 $, and 5.36 $ respectively.

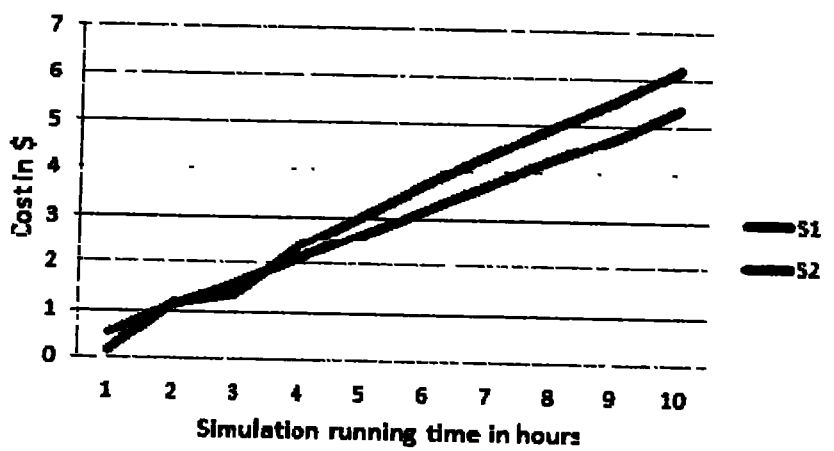


Figure 8.5: Cost comparison for Scenario 1 and 2

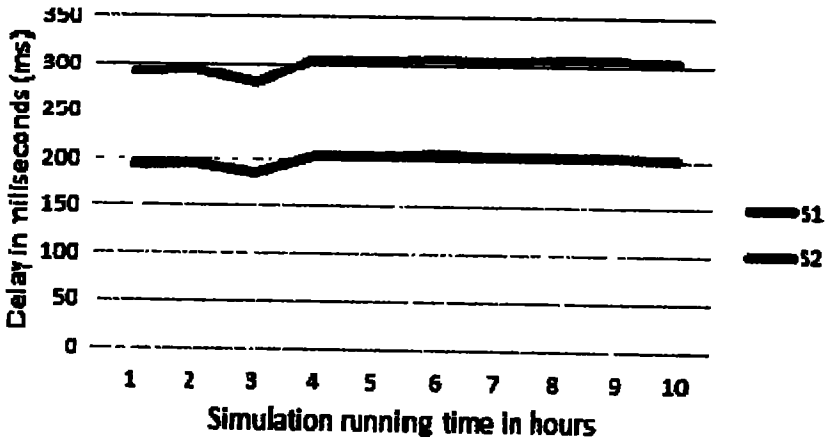


Figure 8.6: Delay comparison for Scenario 1 and 2

In the situation 2, the workload of 100 KB requests per 10 seconds was routed to the SLA-MaaS and later on to the provider and consumer. The simulation was run for ten times (1 to 10 hours for each run). The average delay (as shown in Table 8.8) for directly forwarding the file to the cloud provider was 302 ms ; the average processing delay was 0.11 ms. The total average delay faced by the consumer or provider was 302 ms. The transfer cost (as shown in Table 8.9 and Figure 8.7) for each simulation run is 0.04 $, 0.05 $, 0.11 $, 0.17 $, 0.25 $, 0.34 $, 0.40 $, 0.44 $, 0.50 $, and 0.57 $ respectively; similarly the processing cost is (as shown in Figure 8.8) 0.6 $, 1.09 $, 1.16 $,2.05 $, 2.55 $, 3.05 $, 3.56 $, 4.08 $, 4.50 $, and 5.07 respectively. Therefore, the total cost (i.e, processing and transfer) of each hour is 0.14 $, 1.19 $, 1.38 $, 2.39 $, 3.05 $, 3.73 $, 4.36 $, 4.96 $, 5.50 $, and 6.21$ respectively.

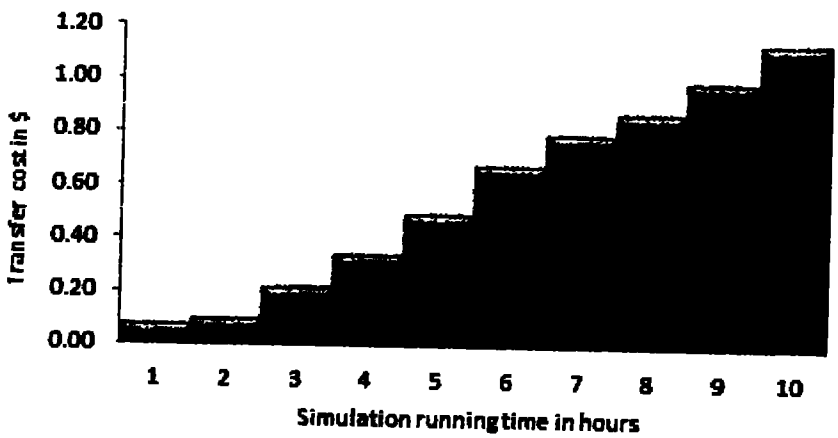


Figure 8.7: Transfer cost for Scenario 2

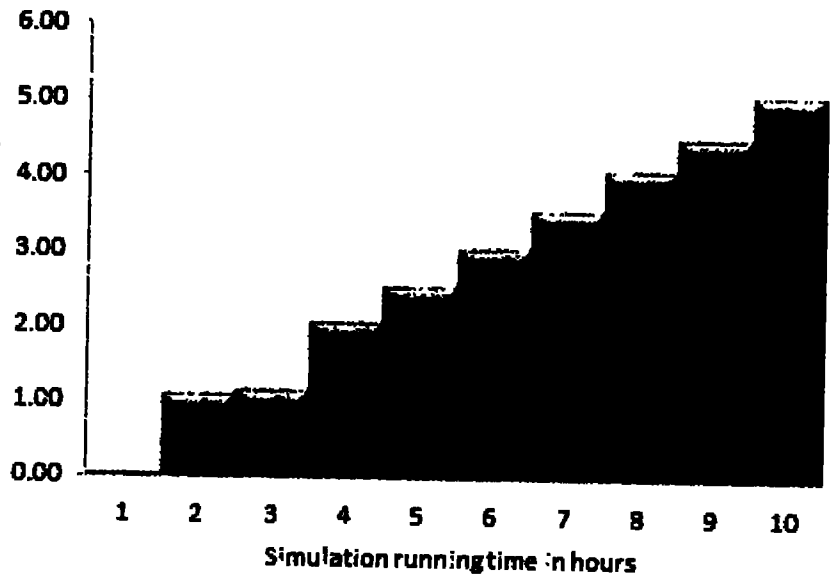


Figure 8.8: Processing cost for Scenario 2

The Figure 8.5 and 8.6 shows the comparison of both the scenarios. The Figure 8.6 shows that the overall delay of Scenario 2 is slightly greater than the Scenario 1. The Figure 8.5 shows that the overall cost of Scenario 2 is slightly greater than the Scenario 1. The benefit of the Scenario 2 lists the *customer satisfaction, retention* and *trustworthy relationship*. Furthermore, for such benefits, we slightly lose the revenue and delay, however, this is not so high to affect the whole revenue or to decrease the overall performance. Furthermore, the forthcoming 5G network, will minimize the network delay (ultra latency delay) [193].

To overcome the above challenges (exist in the above both scenarios), the following measures have been investigated to minimize the delay (as per Scenario 3):

- Minimizing the size of overhead by minimizing the monitoring parameters
- Minimizing the frequency of monitoring reports
- Maximizing the SLA-MaaS DC powers.

Therefore in the Scenario 3, we reduce the overhead size from 100 KB to 50 KB and monitoring data frequency from 30 to 15. The result comparison with the previous results are; The transfer cost for reducing the overhead size (as shown in Figure 8.9) for each simulation run is 0.08 $, 0.10 $, 0.12 $, 0.18 $, 0.24 $, 0.34 $, 0.40 $, 0.44 $, 0.50 $, and 0.58 $. This cost is half of the previous costs (when overhead was 100 KB). Similarly the transfer cost for minimizing the sending frequency (as shown in Figure 8.10) is 0.08 $, 0.10 $, 0.12 $, 0.18 $, 0.24 $, 0.34 $, 0.40 $, 0.44 $, 0.50 $, and 0.58 $. Furthermore, the same results were

taken (i.e, reduced processing cost) by increasing the processing power of the SLA-Monitoring as a Service (SLA-MaaS).

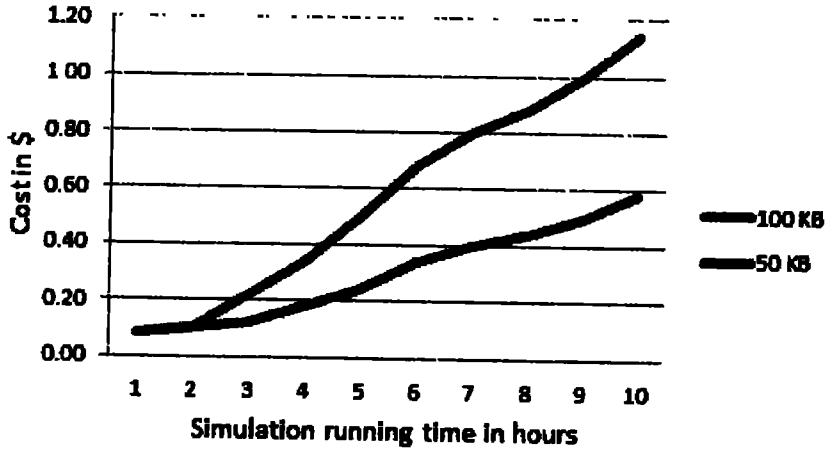


Figure 8.9: Cost comparison for reducing overhead

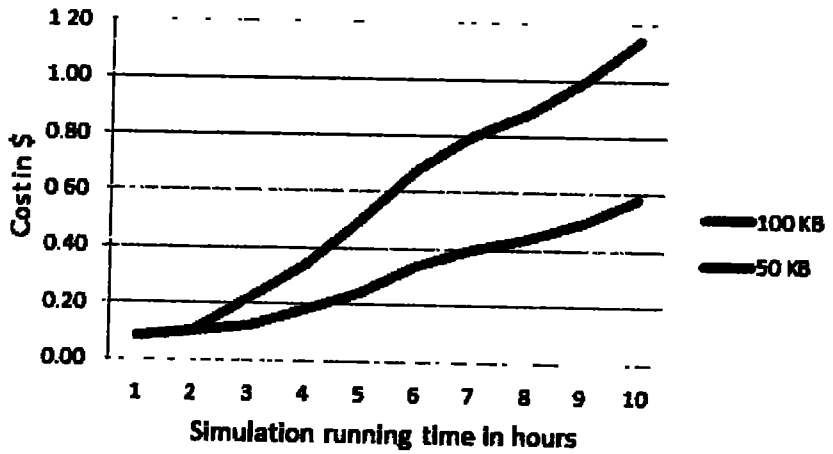


Figure 8.10: Comparison for reducing monitoring frequency

8.4 Comparative Analysis

Being an important part of the cloud business, massive investigation has been carried out on cloud services monitoring. The big difference between the traditional monitoring services and SLA-MaaS is, that the *SLA-MaaS uses third party monitoring systems.* To check the supremacy of the proposed structure, the model is compared with [194],[195], [196], [197], [198], [199], [200], and [201] research studies.

The most challenging with the SLA-MaaS is the *communication overhead.* The comparative results (as shown in Figure 8.5 & 8.6) of monitoring cost shows that SLA-MaaS adds a monitoring overhead to the communication which slightly increase the delay and also the transfer cost. However, comparing this model with the existing studies (in Table 8.10) shows that this framework increase the customer trust by providing third party monitoring. The customer trust means, customers' attention and retention.

Customers' attention and retention is the big goal of every service provider. This increases the underlying resources utilization, increasing the revenue of the providers. This also provides a relaxation and trust worthy environment to the consumer. Which shows that the proposed framework benefits both parties (i,e. provider and consumer).

8.5 Conclusion

Lack of *portability*, *reliability*, and *intractability* in existing monitoring system leads to customer dissatisfaction. This article addresses such issues by providing third party monitoring having clear-cut SLA and penalties structure. Results show that network overhead slightly delays the response time and increase the transfer cost. Comparatively to the benefits of the proposed system (e.g, customer satisfaction and retention), this delay and cost overhead are negligible. Response delay and cost are not so higher to interfere with the smooth running of processing. SLA-MaaS is a first step towards the standardized cloud monitoring system. Therefore, our future directions are to extend this work with unique standards for cloud monitoring.

Chapter 9

Comparative Analysis

An extensive literature exists on *revenue maximization* in cloud computing. To check the supremacy of this study, we compared it with [95], [129], [71], [99], [124], [84], [137], [132] and [123]. For comparative analysis, the criteria given in the table 8.10, were used to evaluate the functioning of previous and the proposed approach. Table 9.1 shows the detailed comparison study of previous studies.

Table 9.1: Comparative analysis of related studies

Paper and Authors	C1	C2	C3	C4	C5	C6	C7
Adil et al. [95]	✓	✓	✓	✗	✗	✗	✗
Kundu et al. [71]	✗	✓	✗	✗	✓	✗	✗
Toosi et al. [129]	✗	✓	✗	✓	✓	✓	✗
Hadji and Zeghlache [99]	✗	✗	✓	✗	✓	✗	✗
Tevi Yombame and Zbigniew [124]	✗	✗	✗	✗	✗	✓	✗
Macas et al. [84]	✓	✓	✗	✗	✗	✗	✗
Hong and Baochun [137]	✗	✗	✗	✗	✓	✓	✗
Zhao et al. [132]	✗	✓	✗	✗	✗	✓	✗
Qi et al. [123]	✗	✗	✓	✗	✓	✗	✓
Wu and Buyya [113]	✓	✓	✓	✗	✓	✗	✗
Mei et al. [111]	✗	✓	✗	✓	✓	✗	✗
Rongdong et al. [145]	✓	✗	✗	✗	✗	✗	✗

Table 9.1: Continued on next page

Table 9.1: continued from previous page

Paper and Authors	C1	C2	C3	C4	C5	C6	C7
Hamsanandhini and Mohana [108]	✗	✗	✓	✗	✓	✗	✗
Gao et al. [98]	✗	✗	✓	✗	✗	✓	✗
Mehiar et al. [38]	✗	✗	✗	✗	✓	✓	✗
Safraz and Wayne [134]	✗	✗	✓	✗	✗	✓	✗
Amit and Zheng [102]	✗	✗	✓	✗	✗	✓	✗
Hou et al. [103]	✓	✗	✓	✗	✗		
Afzal et al. [57]	✓	✓	✓	✓	✓	✓	✓
Hong et al. [143]	✗	✗	✗	✗	✗	✓	✓
Snehanshu et al. [135]	✗	✗	✗	✗	✗	✓	✗

Table 9.1: It ends from the previous page.

In terms of maximizing IaaS provider revenue, *customer dissatisfaction* and *penalties* play a crucial role. IaaS resources are not storable, they are wasted if not utilized on time. Maximum revenue can be generated by increasing service utilization. In the cloud market, the revenue of most providers is wasted in penalties. Cloud service providers also lose most of their customers because of their dissatisfaction. The rejection of customers is a waste of revenue. In cloud computing, there are many opportunities to maximize revenue. Cloud services are available everywhere and at any time. If manage carefully, it can generate more revenue than any other businesses. To maximize IaaS provider revenue, this thesis propose a solution to the above-discussed issues.

9.1 Performance Management

Several research refer *performance* for revenue maximization. Ran et al. [69] used QoS constrains, Feng and Buyya [72] used efficient resources allocation and Kundu et al. [71] worked on the revenue driven resources allocation. The major limitation over here is, the performance degradation due to *limited resources* or overutilization. They did not discuss any proper framework to maximize the resources scalability according to the incoming requests.

These investigations and research have greatly improved revenue and performance optimization. However, there is a complex correlation between performance and revenue, which is lacking in the existing literature. For example, performance attracts more customers, however, it raises prices. Rising prices hurt the customers. Also, with good performance, heavy workloads are expected. This workload affects performance and SLA. All of these issues require further exploration and investigation.

To overcome these problems, a framework is needed that can handle dynamic and loaded SLAs. Also, cloud performance is directly dependent on the scalability of resources. Resource scalability and a clear cut Service Level Agreement can protect the provider from performance degradation. The proposed framework *hires external resources* in extreme usage. Hiring external resources solves the problem of scalability. The provider will be able to entertain more customers and expand their business. Performance degradation due to overuse can be managed to some extent.

9.2 SLA and Penalties Management

Macas et al. [84] worked on SLA violations and their cancellation, Wu et al. [85] worked on resources provision according to incoming SLAs, and Emeakaroha et al. [88] worked to lower SLA metrics to higher metrics to be measured. The Major challenges over here is the revenue wastage in penalties payments also the rejection of SLA in extreme utilization. *Penalties* greatly affect the cloud business. Usually cloud computing accepts loaded SLAs but later on, they cannot provide resources as per the agreement and in the result, they have to pay much of their revenue in penalties.

SLA and *penalties* are thoroughly investigated. SLA parameters are measured. Different mechanisms are reviewed to minimize the violations of SLA. Penalties are also well studied to maintain this burden minimum. The main disadvantage that needs to be improved is that most providers cancel the SLA as their workload increases. Most providers with limited resources also have admission control and the heavy workloads are canceled. These issues need to be explored further.

The cloud business attracts customers. Their server workloads increases with time. It is expected that more than 331 billion dollars will be invested in cloud computing up to 2023. For such a large business, it is extremely necessary to have a clear-cut SLA (*PerSLA*) to provide resources according to agreed parameters. The proposed architecture propose three layers threshold structure. On first threshold violation, very less penalties are imposed on provider. This does not affect the provider or consumer. This is only the notification to the provider to adjust the resources.

9.3 Resources Scalability

Cloud bank was proposed to make the resources scalable according to incoming requests [98], Jennifer et al. [87] worked on the personalized services level agreement to provide the services according to a customers' demands, Hadji and Zeghlache [99] used in-sourcing and outsourcing techniques in federated cloud to make the resources scalable, Upadhyay and Lakkadwala [100] used the migration techniques, Li et al. [101] used the concept to run private cloud resources on public cloud, Mansour et al. [120] worked on live cloud migration, Santikarama and Arman [121] used the Economical Customer Relationship Management (ECRM)

techniques, Hadji and Zeghlache [99] worked on live cloud migration, Amit and Zheng [102] worked on resources scalability for mobile applications, and Hou et al. [103] worked to scale the resources using geographically distance servers.

Consistent with the above discussion, the authors proposed different solutions for managing *scalability* issues. The federated cloud partially overcomes this problem by sharing resources within the union. However, the disadvantage of this proposal is that suppliers are forced to hire resources from specific fixed providers. Another study suggests that workloads are only accepted from the surrounding area. This study presents some positive directions: with a smaller distance, system performance may increase, however, this narrows the concept of cloud computing. All of these issues need to be address and require further study.

The proposed framework solved this problem by *hiring external resources*. In this case, two service level agreements are implemented. First service level agreement is signed between the service provider and the consumer and the second service level contract is signed between the service provider and the external provider. External resources also affect performance and security. The cloud provider should not commit full resources from an external cloud service because they pay for external cloud services based on the usage of the services. The external cloud provider billed the IaaS provider based on resource usage.

9.4 Customers' Satisfaction

Hamsanandhini and Mohana [108] investigated customer satisfaction for revenue maximization, Manzoor et al. [110] used the customers centered approach, and Mei et al. [111] discussed the customer satisfaction by filling the Quality of Services and Prices of Services parameters.

The above studies have investigated *customer satisfaction* issues and suggested different frameworks to satisfy them. The main contributions of these studies are to classify customers into different layers, depending on these layers, providers create a customer satisfaction layer. Up to certain extent, it was a good contribution, but what to do in case of limited resources with a larger workload and to optimize performance and prices. These complexities require additional investigations to optimize performance and pricing, as well as to manage huge workload with limited resources.

With the quality, customer satisfaction is also proportional to service scalability. The proposed framework increases resources scalability by hiring external resources. Customer satisfaction can be increased by providing good quality services. Good quality services needs scalable resources. Prices also strongly affect the customers. Some customer prefer performance while some prefer lower prices. A good pricing framework may affect more customers. Prices are offered according to customers choices. Performance is optimized by *Performance based Service Level Agreement*. Customer support is the main reason for customer satisfaction. They feel confident with proper customer support. PerSLA gives proper feedback and support.

9.5 Resources Provisioning and Management

Shin et al. [116] worked on deadline guaranteed resources utilization, Balagoni and Rao [117] discussed the scheduling policies for heterogeneous clouds, Yuan et al. [118] proposed temporal task scheduling in the hybrid cloud, and Gao et al. [98] worked on trasncoding video streaming. The issue with the above studies are that with limited resources and maximum resource utilization, cloud providers reject such existing customers whose penalties are lower than new customers' revenue. There are different QoS SLAs, the combination of those SLAs are adopted which are having higher revenue and those are canceled which are having lower revenue.

The above studies explain the optimal use of cloud resources, the challenges and the different frameworks proposed to maximize the use of resources. The main resource utilization challenges are *admission control* and *SLA violation*. Providers do not overload their resources because of the fear of violation of the service level agreement. These complexities require further investigation to optimize the use of resources and the violation of SLAs.

Rejecting any customer is a great loss in the cloud business. Such providers will never be trusted in future. Resources utilization discusses the total revenue earned by total available resources. Computer resources are not storable and get wasted if not utilized on time. Efficient resources utilization depends on customer satisfaction, attraction, retention and accepting dynamic SLAs. Proposed framework worked on the resources *scalability* and *dynamic prices* to avoide SLA violation and customer rejection. This maximizes the provider business. .

9.6 Cost and Prices Management

Ran et al. [69] worked on the dynamic pricing model in cloud computing, Zhang and Boutaba [128] used market analyzer, capacity planer and dynamic pricing scheme, Toosi et al. [129] use optimal capacity and different pricing schemes, Chi et al. [130] used efficient resources scheduling and prices models, Zhou et al. [202] worked on cost optimization , Ibrahim et al. [119] worked on hybrid cost and priority based scheduling, Tevi Yombame and Zbigniew [124] worked on cost minimization to maximize the profit, Tang and Chen [133] proposed economic framework for resources management, Mehiar et al. [38] worked on prices and capacity planning and Safraz and Wayne [134] worked on efficient resources allocation and costing.

Lower prices attract customers, however, it also creates performance problems. The above studies have looked in-depth at how to optimize costs, prices and performance. However, additional research is needed to manage these settings optimally.

The proposed framework used a special framework to *optimize costs, prices, and penalties* to maximize vendor revenue. Prices are set according to the total cost. Costs are minimized by effectively managing

energy consumption and human resources. Joint prices are used for cloud business. Fixed rates are used for high-performance customers. Spot prices are used for underutilized resources. This increases the use of resources.

9.7 Advertisement and Overutilization

Dabbagh et al. [139] used the over commitment techniques to keep the resources busy, Metwally et al. [140] used the resources optimization tool, Hammoudi et al. [142] used the multi agent architecture for load balancing, Samimi et al. [141] proposed the double auction model, and Deng et al. [203] worked on online auction.

The above investigation shows that *advertising* increases the number of customers. People also take interest in the *auction*, which attracts more customers. More customers can overutilize the resources, which can lead to SLA violation. These complexities require further research and exploration.

The major challenges over here is that advertisement may requires much cost. Which have a major impact on the cloud profit. Recent marketing and advertisement techniques may be used to reach and attract new customers.

9.8 Summary and Conclusion

Our work differentiates from previous works in many ways. Table 9.1 state this difference clearly. The difference is: (i) we have taken the *customer satisfaction* on top priority. Customer satisfaction not only increases the underlying resources utilization, however, it also increases the number of new customers and retention. Different prices options pays back to customers in case of violation. This attracts more customers. (ii) *Penalties* waste most of the revenue. Limited resources are the reason for SLA violation and penalties. *Federated cloud* is the solution towards limited resources, however, the problem with that is customers are compelled to hire from a particular provider. There is no liberty with the provider to hire from the open market. To handle this issue, we hire third provider resources to provide uninterrupted resources to customers. (iii) *PSLA's* threshold works in layers. Penalties are not imposed all in sudden. The first threshold is actually a notification to the provider to adjust the performance. (iv) In case of *migrating* resources to external cloud provider, smart migration techniques are used to select optimal providers and VMs for the workload in terms of performance and prices. The comparison results

Chapter 10

Conclusions and Future Directions

Cloud computing is *rapidly changing* the way of market operation. It got a considerable attraction in the past few years. Every year, a large part of traditional cloud market migrates to the cloud. It is expected that more than 51 billion devices will be connected to the internet up to the end of 2023 [2, 155]. This figure is seven times higher than the total human population of the planet. Smart devices and sensors are already producing massive data. This gigantic transaction to the cloud, makes it overcrowded. Cloud business works on the internet. There is no face-to-face communication between suppliers and customers. Therefore, special attention and research are needed to effectively manage the cloud sector.

Connecting to this, the major objective of this thesis was to *handle this massive customers' data and maximize the providers revenue with limited resources.* With limited resources, it was a big issue to extend the providers' business. Limited resources also cause performance degradation and customers dissatisfaction. Related objectives to maximize the provider revenue is listed in chapter 1. In order to achieve these objectives, we proposed and investigated a set of parameters and economics-inspired mechanism for IaaS cloud providers, including services performance management, SLA and penalties management, resources scalability, customer satisfaction, resources utilization and management, cost and prices management, advertisement, and overutilization.

The investigations are divided into three parts. First part discussed the revenue maximization through customers' satisfaction and hiring external resources. Second part discussed the Performance based Service Level Agreement to efficiently optimize the performance, customer satisfaction and prices. Third part discussed the efficient resources scheduling on external resources searching provider for best delay, running time and cost.

Chapter 2 presented the detail discussion of cloud computing, related parameters to revenue maximization and migration needs of desktop resources to cloud computing. Cloud computing has many advantages

over the traditional desktop services, due to that, extensive workload is migrating towards it. This chapter discussed the main parameters which increases the providers' revenue directly or indirectly.

Chapter 3 reviewed the related studies. Related techniques and literature is classified into seven different categories. The parameters toward revenue maximization, as discussed earlier, are derived from this literature. This section has helped us to identify gaps, challenges, and the research direction for providers' revenue maximization.

Chapter 4 discussed the proposed methodology. CloudSim and Cloud Analytic are used to simulate the proposed techniques. We have extended the CloudSim simulator to evaluate the effectiveness of the proposed model. The experimental setup was coded using Java to evaluate the functioning of this model.

Chapter 5 major focus was on revenue maximization by hiring external resources. In extreme utilization, the high workload is outsourced to external resources which extends the provider business having limited resources. Prices also play a decisive role in customer satisfaction. Joint prices are used and optimized at such a point to maximize revenue and customer satisfaction. Results show that the proposed model is able to efficiently handle massive customers' requests.

Chapter 6 proposed the Performance based Service Level Agreement (PerSLA). PerSLA optimizes these parameters to one optimum point. PerSLA clearly specified the parameters, their threshold values and penalties. Algorithms monitor the services and try to enhance the performance if it goes down. On first two thresholds, SLA is not terminated, however, prices are reduced which is actually a call to the service provider to enhance the performance. SLA is terminated on third threshold violations.

Providers with limited resources, faces many challenges. The major challenge is that provider business remain limited. In previous chapter, we proposed a framework to hire external resources. This was a best framework toward this major challenge, however, the issue with hiring external resources is that it increases the cost in terms of energy consumption and delay. To handle this issue, chapter 7 proposed to select those external CSPs having best running and delay time. Simulation results show that best selection of (P1,P2) dramatically minimize the external cost, delay and running time.

In order to check the supremacy of this framework, chapter 8 compared this work with other related techniques. Comparative analysis results show that this model attracts more customers due to multi pricing offers. Furthermore, it also provides high performance services for higher payers.

10.1 Summary of Key Findings

The proposed model uses a multi-price structure, which means more customers will be attracted. For best performance requests, reserved prices are used. For lower usage and higher performance, on demand prices

are used. For underutilized resources, negotiated prices are used. This attracts more customers to the cloud market. The simulation results show that revenue and profit increase with multiple prices and external revenues. The table 10.1 shows the objectives, challenges and potential solutions of the studies.

In the case of a high workload, parts of the workload are migrated to external services to allow scalability of resources. If virtual machines become overloaded and reach extreme utilization, their workload is migrated *(Local migration)* to other virtual machines for better results. If all local virtual machines are busy and there are no other local resources, the workload is migrated *(Global migration)* to global resources. When space becomes available on local resources, workload is migrated back to local resources to minimize external costs. The results of the simulation show that the proposed model is able to effectively manage the dynamic demands of customers. This model greatly contributes to revenue maximization and customer satisfaction.

Cost, performance, penalties and *revenue* are very essential parameters to the cloud market. Their co-relation has not been comprehensively investigated yet. The proposed framework proposed the Performance based Service Level Agreement (PerSLA) which optimizes these parameters to one optimum point. For performance management, two algorithms are used. The first algorithm tries to implement the SLA according to the agreed terms. It notifies the provider to optimize the performance if it goes down. SLA is not canceled directly on first violation. In this case, SLA threshold is divided into three different layers. Penalties are accordingly divided. The second algorithm control the scheduling according to the first algorithm directions. Results show that the level of SLA violations and penalties are controlled with SLA. This increases the providers' revenue and profit.

Resources scheduling on external CSPs play very important role in cloud data centers. Migration resources to external resources not only affect the performance, it also increase the cost, energy consumption and customer dissatisfaction. That is the main reason that scheduling policies are the primary concerns for providers. A huge number of devices are connected with internet and cloud business is increasing very rapidly. In such a big data, massive demands are forwarded toward cloud provider for running. To meet these customers' requirement, we rely on external resources.

To optimize the performance in terms of delay, running time, and cost, we consider two important parameters. We selected such like external providers, whose delay time and its internal utilization is low. External CSP resource management plays a very important role in the cloud data centres. The migration workloads to external resources not only affect performance but also increase costs, energy consumption and customer dissatisfaction. This is the main reason why scheduling strategies are the main concern of the providers. Very large number of devices are connected to the Internet and cloud business is increasing very rapidly. In such a large volume of data, massive requests are transmitted to the cloud provider for execution. To meet the needs of these customers, we rely on external resources.

The results of the simulation show that the delay time and the execution time increase with the target dis-

tance. When we deploy the resources on the nearest resources but are not free as needed, the delay time increases. In the third scenario, we select the best CSPs to run the workload. The results of the simulation show that this gave a very appropriate execution time and response time.

Table 10.1: Potential and challenges of concern parameters

Evaluation Criteria	Research objectives	Related work	Challenges	Potential solution
C1 Performance management	If considered various performance parameters such as waiting time, running time, response time, security, reliability and availability	[69] [72] [71] [73] [74] [75] [76] [77] [78]	In heavy load, issues in timing, reliability and availability	Resources scalability and Performance based Service Level Agreement (PSLA) are a good steps toward performance management [57].
C2 SLA and penalties management	If acquired a clear cut SLA to provide above agreed performance parameters according to Agreed SLA	[84] [86] [88] [33] [34] [35] [90] [91] [92] [93]	Not filling the agreed SLA parameters in heavy load	Automatic SLA worked toward Performance based Service level agreement [146].
C3 Resources scalability	If questioned the provider about its resources to execute higher load SLAs	[98] [87] [99] [100] [101] [102] [103]	Limited resources. Cancellation of SLA in heavy load	Hiring external resources are getting attached with federated cloud [57]
C4 Customer satisfaction	If discussed the customers' satisfaction in terms of customer attraction and retention	[108] [110] [111] [109] [204] [73] [107]	Dissatisfaction in extreme load and prices	Offering dynamic prices on customer choices. High performance services. Authors in [146] worked toward customers' satisfaction.
C5 Resources utilization and management	If considered the total resources in use with respect to total available resources	[116] [117] [118] [98] [120] [121] [99]	Lower utilization of resources and wastage	Customers' satisfaction, customers, attraction and retention may increase the resources utilization [146].

Table 10.1: Continued on next page

Table 10.1: continued from previous page

Evaluation Criteria	Research objectives	Related work	Challenges	Potential solution
C6 Cost and prices management	If considered cost minimization by using various methods and reliability of prices	[69][128] [129] [130] [202] [119] [132] [124] [133] [38] [134]	High cost so high prices. Wastage of costs on physical and human resources.	Cost may be reduce by various methods. Suitable prices attract more customers. Dynamic pricing is the best solution for prices issues [57].
C7 advertisement and auctions	If considered different parameters such as to reach new customers, to get good auction and also sell underutilized resources.	[139] [140] [142] [203] [143] [205] [109]	Minimum attention toward new customer attraction	To attract new customers and to sell under utilized resources it is better to do advertisement and auction.

Table 10.1: It ends from the previous page.

10.2 Suggestions and Future Directions

Cloud computing, fog computing and the *Internet of Things (IoT)* are great interest in academies and the marketplace. It is expected that more than 51 billion devices will be connected to the Internet up to 2023. These massive devices will need a perfect provider to manage all resources and satisfy customers. This framework can easily be adapted to the limited resource provider for large data management. Cloud, fog and IoT work on the network and the delay is important. This framework can easily be extended to fog projects and the Internet of Things to minimize delays, costs and increase performance.

10.2.1 Outsourcing the Services

In Chapter 5, we hired the external resources for resources scalability. In the case of *underutilization*, resources can be wasted. We plan to investigate the outsourcing to provide resources to external providers in case of underutilization.

10.2.2 Customers' Satisfaction Measurement

In Chapter 6, we partially discussed the importance of customer satisfaction and how it works. Customer satisfaction is the most important parameter in any business. In the future, it is planned to work in depth to *measure the customers' satisfaction* for cloud activities.

10.2.3 Power Consumption

Most of the prices are wasted on energy consumption. This consumption increases costs and prices, hence the dissatisfaction of customers. In Chapter 7, we partially discussed energy consumption, but we did not properly propose a solution to minimize this waste. We plan to investigate *energy wastage* in data centers to minimize costs and prices. This will have a direct positive affects on revenue and customer satisfaction.

10.2.4 IoT and Fog Computing

Billion of devices are connecting to the cloud network. This massive traffic seriously affects performance. To solve these problems, future research will need to investigate and integrate IoT and Fog technologies into cloud computing so that some of the data can be processed locally. Therefore, our future directions are to investigate this issue further to minimize the load on the network.

10.3 Final Remarks

Cloud computing is the key technique towards utility computing. It provides all types of desktop services on the network. Massive investment is being spent in this market. To maximize the provider profit in cloud market, this dissertation investigated related mechanism and derived key parameters to providers' revenue.

Bibliography

[1] G. Davis, "2020: Life with 50 billion connected devices," in *IEEE International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, USA: IEEE, Jan 2018, pp. 1–1.

[2] "Forbes: Cloud computing forecast," https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts2017/#5c42322c31e8/, 2017.

[3] "Annual forecast of data-sphere," www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/#59c815325459//, 2019.

[4] "Cisco users forcast," https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2021.

[5] X. Wang, B. Wang, and J. Huang, "Cloud computing and its key techniques computer science and automation engineering (csae)," in *International Conference on Cloud Computing*. Shanghai, China: IEEE, 2011, pp. 404–410.

[6] C. Lalit and B. Rabindranath, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 208 – 219, 2020.

[7] R. Ghani-Ur, G. Anwar, M. Shad, S. Madhusudan, and S. Dhananjay, "Selfishness in vehicular delay-tolerant networks: A review," *Sensors*, vol. 20, no. 10, pp. 1–19, 2020.

[8] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges," in *24th International Conference on Advanced Information Networking and Applications (AINA)*, vol. 4. Perth, WA, Australia: IEEE, 2010, pp. 27–33.

[9] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," *Algorithms and architectures for parallel processing*, vol. 4, pp. 13–31, 2010.

[10] "Oracle cloud," https://cloud.oracle.com/home, 2019.

[11] "Sap cloud platform," https://cloudplatform.sap.com/index.html/, July 20, 2019.

[12] X. Lei, X. Liao, T. Huang, and H. Li, "Cloud computing service: The case of large matrix determinant computation," *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 688–700, 2015.

[13] H. Ziglari and S. Yahya, "Deployment models: Enhancing security in cloud computing environment," in *22nd Asia-Pacific Conference on Communications (APCC)*, IEEE. Shanghai, China: IEEE, 2016, pp. 1–6.

[14] "Amazon web services," http://aws.amazon.com/ec2/pricing/, 2019.

[15] "Microsoft azure cloud," http://azure.microsoft.com/en-us/, 2019.

[16] "Google cloud computing services," https://cloud.google.com/, 2019.

[17] "Alibaba cloud," https://www.alibabacloud.com/, 2019.

[18] "Ibm cloud," https://www.ibm.com/cloud/, 2019.

[19] "Cobweb cloud solutions," https://cobweb.com/, 2019.

[20] "Mulesoft cloud computing services," https://www.mulesoft.com/resources/cloudhub/cloud-computing-service, 2019.

[21] "Salesforce cloud computing services," https://www.salesforce.com/eu/products/what-is-salesforce/, 2019.

[22] J. Ateeqa, B. Afzal, and R. Tauseef, "Sla based infrastructure resources allocation in cloud computing to increase iaas provider revenue," *Research Journal of Science and IT Management*, vol. 4, no. 3, pp. 37–44, 2015.

[23] "National data base and registration authority pakistan," https://www.nadra.gov.pk//, 2019.

[24] A. C. Zhou, B. He, and C. Liu, "Monetary cost optimizations for hosting workflow-as-a-service in iaas clouds," *IEEE transactions on cloud computing*, vol. 4, no. 1, pp. 34–48, 2016.

[25] Q. Lu, J. Yao, H. Guan, and P. Gao, "gqos: A qos-oriented gpu virtualization with adaptive capacity sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 4, pp. 843–855, 2020.

[26] N. Matrazali, M. H., R. Uda, T. Kinoshita, and M. Shiratori, "Vulnerability analysis using network timestamps in full virtualization virtual machine," in *International Conference on Information Systems Security and Privacy (ICISSP)*. Angers, France: IEEE, 2015, pp. 83–89.

[27] A. Kaushik and A. Chaturvedi, "A review of efficient data utilization schemes in cloud computing,"

in *3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. New Delhi, India: IEEE, 2016, pp. 1–1.

[28] R. Ghani ur, G. Anwar, Z. Muhammad, S. Muhammad Imran, and S. Dhananjay, "Sos: Socially omitting selfishness in iot for smart and connected communities," *International Journal of Communication System*, 2020.

[29] L. Shakkeera, L. Tamilselvan, and M. Imran, "Improving resource utilization using qos based load balancing algorithm for multiple workflows in iaas cloud computing environment," *ICTACT Journal On Communication Technology*, vol. 4, no. 02, pp. 750–757, 2013.

[30] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2010, vol. 87.

[31] B. El Zant, I. Amigo, and G. Maurice, "Federation and revenue sharing in cloud computing environment," in *International Conference on Cloud Engineering (IC2E)*. Boston, MA, USA: IEEE, 2014, pp. 446–451.

[32] L. Wu, R. Buyya *et al.*, "Service level agreement (sla) in utility computing systems," *IGI Global*, vol. 15, 2012. [Online]. Available: https://arxiv.org/abs/1010.2881

[33] Shivani and S. Ajmer, "Taxonomy of sla violation minimization techniques in cloud computing," in *Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. Coimbatore, India: IEEE, April 2018, pp. 1845–1850.

[34] V. Shahin, J. K. Catherine Truchan, and E. Halima, "Automated enforcement of sla for cloud services," in *IEEE 11th International Conference on Cloud Computing (CLOUD)*. San Francisco, CA, USA: IEEE, July 2018, pp. 49–56.

[35] L. Taher, M. Achraf, G. Walid, T. Samir, and G. Faiez, "Cloud sla modeling and monitoring," in *IEEE International Conference on Services Computing (SCC)*. Honolulu, HI, USA: IEEE, June 2017, pp. 338–345.

[36] N. C. Nguyen, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: a survey," *IEEE Communications Surveys & Tutorials*, vol. 9, no. 2, pp. 954–1001, 2017.

[37] S. Sinung, S. Suhono, Suhardi, and S. Roberdm, "Performance measurement of cloud computing services," *International Journal on Cloud Computing: Services and Architecture(IJCCSA)*, vol. 2, no. 2, pp. 9–22, 2012.

[38] D. Mehiar, H. Bechir, G. Mohsen, and R. Ammar, "Exploiting task elasticity and price heterogeneity

for maximizing cloud computing profits," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 85 – 96, 2018.

[39] R. Ghani, G. Anwar, Z. Muhammad, G. Shahbaz Ahmed Khan, and M. Shad, "Honesty based democratic scheme to improve community cooperation for iot based vehicular delay tolerant networks," *Transaction on Emerging Telecommunication Technologies*, 2020.

[40] "Mckinsey theory," https://beyondphilosophy.com//, 2019.

[41] "The cloud service industrys 10 most critical metrics," https://guidingmetrics.com/content/cloud-services-industrys-10-most-critical-metrics/, Jan. 2019.

[42] G. Johnu, C. Chien-An, S. Radu, and X. Geoffrey G., "Hadoop mapreduce for mobile clouds," *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 224 – 236, 2019.

[43] "Hadoop," https://hadoop.apache.org//, 2019.

[44] K. D. Foote, "A brief history of cloud computing," https://www.dataversity.net/brief-history-cloud-computing/, 2019.

[45] X. Wang, Y. Han, V. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[46] Shivani and A. Singh, "Taxonomy of sla violation minimization techniques in cloud computing," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 1845–1850.

[47] S. Dhariwal and H. Makwana, "Maximize the cloud profit to improved qos in cloud computing: Design and analysis," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, 2019, pp. 279–285.

[48] S. Benbrahim, A. Quintero, and M. Bellaïche, "Live placement of interdependent virtual machines to optimize cloud service profits and penalties on slas," *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 237–249, 2019.

[49] L. B. Bhajantri and T. Mujawar, "A survey of cloud computing security challenges, issues and their countermeasures," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 376–380.

[50] H. Shen and L. Chen, "A resource usage intensity aware load balancing method for virtual machine migration in cloud datacenters," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 17–31, 2020.

[51] F. Liu and F. Hao, "Web service integrated cloud computing management information system," in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, 2015, pp. 288–293.

[52] H. Yilong, S. Zhou, and G. Song, "Utility based data computing scheme to provide sensing service in internet of things," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 2, pp. 337–348, 2019.

[53] P. S. Challagidad and M. N. Birje, "Determination of trustworthiness of cloud service provider and cloud customer," in *5th International Conference on Advanced Computing Communication Systems (ICACCS)*. IEEE, Mar. 2019, pp. 839–843.

[54] S. Jin, Z. Yi, W. Zebin, Z. Yaoqin, Y. Xianliang, D. Zhongzheng, W. Zhihui, P. Javier, and P. Antonio, "An efficient and scalable framework for processing remotely sensed big data in cloud computing environments," *IEEE Transactions on Geo science and Remote Sensing*, vol. 57, no. 7, pp. 4294 – 4308, 2019.

[55] C. Ing-Ray, G. Jia, W. Ding-Chau, J. P. T. Jeffrey, A.-H. Hamid, and Y. Ilsun, "Trust-based service management for mobile cloud iot systems," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 4294 – 4308, 2019.

[56] B. Varghese, O. Akgun, I. Miguel, L. Thai, and A. Barker, "Cloud benchmarking for maximising performance of scientific applications," *IEEE Transactions on Cloud Computing*, vol. 7, no. 1, pp. 170–182, 2019.

[57] B. Afzal, G. Anwar, and S. Shahaboddin, "Performance based service level agreement (psla) in cloud computing to optimize penalties and revenue," *IET Communications*, vol. 14, no. 7, 2020.

[58] Z. Lei, F. Anmin, Y. Shui, S. Mang, , and K. Boyu, "Data integrity verification of the outsourced big data in the cloud environment: A survey," *Journal of Network and Computer Applications*, vol. 112, pp. 1–15, 2019.

[59] S. Sayed Chhattan, C. Sajad Hussain, and B. Ali Kashif, "A centralized location-based job scheduling algorithm for inter-dependent jobs in mobile ad hoc computational grids," *Journal of Applied Sciences*, vol. 10, no. 3, pp. 174–181, 2010.

[60] "Accenture global customer satisfaction report," www.accenture.com//, 2019.

[61] G. Siqian, Y. Beibei, Z. Zheng, and C. Kai-Yuan, "Adaptive multivariable control for multiple resource allocation of service-based systems in cloud computing," *IEEE Access*, vol. 16, pp. 13 817–13 831, 2019.

[62] J. B. Abdo, J. Demerjian, H. Chaouchi, and T. Atechian, "Enhanced revenue optimizing sla-based admission control for iaas cloud networks," in *2015 3rd International Conference on Future Internet of Things and Cloud*. IEEE, 2015, pp. 225–230.

[63] A. Vahid, B. Kris, and N. Bryan, "Budget and deadline aware e-science workflow scheduling in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 29 – 44, 2019.

[64] A. Masoud, Y. Saleh, and N. Dusit, "Pricing strategies of iot wide area network service providers with complementary services included," *Journal of Network and Computer Applications*, vol. 147, p. 102426, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804519302784

[65] A. Ankita, V. Gregory, H. Seghbroecka, F. Morab, T. De, and V. Bruno, "Spech: A scalable framework for data placement of data-intensive services in geo-distributed clouds," *Journal of Network and Computer Applications*, vol. 142, pp. 1–14, 2019.

[66] "Google scholar," https://scholar.google.com/, Jan. 2020.

[67] "Elsevier," https://www.elsevier.com/, Jan. 2020.

[68] "Ieee journals," https://ieeexplore.ieee.org/, Jan. 2020.

[69] Y. Ran, Z. Jian, and H. Xi, "Dynamic iaas computing resource provisioning strategy with qos constraint," *IEEE Transactions on Services Computing*, vol. 10, no. 2, pp. 190–202, 2015.

[70] A. Danilo, C. Giuliano, and C. Michele, "Quality-of-service in cloud computing: modeling techniques and their applications," *Journal of Internet Services and Applications*, vol. 5, no. 1, pp. 5–11, 2013.

[71] S. Kundu, R. Rangaswami, M. Zhao, A. Gulati, and K. Dutta, "Revenue driven resource allocation for virtualized data centers," in *International Conference on Autonomic Computing (ICAC)*. IEEE, 2015, pp. 12–21.

[72] G. Feng and R. Buyya, "Maximum revenue-oriented resource allocation in cloud," *International Journal of Grid and Utility Computing*, vol. 7, no. 1, pp. 12–21, 2016.

[73] P. Nazanin, T. Abbas, and M. Sanaei, "A model for evaluating cloud-computing users satisfaction," *African Journal of Business Management*, vol. 7, no. 16, pp. 1405–1413, 2013.

[74] G. Ioannis, T. Dimitrios, and K. Nectarios, "Towards an adaptive, fully automated performance modeling methodology for cloud applications," in *IEEE International Conference on Cloud Engineering (IC2E)*. Orlando, FL, USA: IEEE, April 2018, pp. 148–158.

[75] N. Dung, L. Andre, D. Edward, K. Ken, and A. Amy, "Evaluation of highly available cloud streaming systems for performance and price," in *18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. Washington, DC, USA: IEEE, May 2018, pp. 360–363.

[76] E. Bauer, "Cloud automation and economic efficiency," *IEEE Cloud Computing*, vol. 5, no. 2, pp. 26–32, May 2018.

[77] P. Vladimir, J. Anshul, and G. Michael, "Iaas reactive autoscaling performance challenges," in *IEEE 11th International Conference on Cloud Computing (CLOUD)*. San Francisco, CA, USA: IEEE, July 2018, pp. 954–957.

[78] D. Songtai, Z. Ao, and W. Shangguang, "The performance evaluation of virtual machine placement algorithm based on webcloudsim," in *IEEE 11th International Conference on Cloud Computing (CLOUD)*, vol. 1. IEEE, July 2018, pp. 950–953.

[79] G. Anup H., "A survey paper on cloud computing and its effective utilization with virtualization," *International Journal of Scientific and Engineering Research,*, vol. 4, no. 12, p. 363–375, 2013.

[80] D. Qiang, "Cloud service performance evaluation: status, challenges, and opportunities – a survey from the system modeling perspective," *Digital Communications and Networks*, vol. 3, no. 2, pp. 101–111, 2017.

[81] H. Artail, M. A. R. Saghir, M. Sharafeddin, H. Hajj, A. Kaitoua, R. Morcel, and H. Akkary, "Speedy cloud: Cloud computing with support for hardware acceleration services," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 850–865, 2019.

[82] G. Anwar, B. Afzal, and J. A. D. Saeedullah, "Issue and challenges in cloud storage architecture: A survey," *Researchpedia Journal of Computing (RpJC)*, vol. 1, no. 1, p. 50–64, 2020.

[83] A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, "Providing performance guarantees for cloud-deployed applications," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 269–281, 2020.

[84] M. Macas, J. O. Fit, and J. Guitart, "Rule-based sla management for revenue maximization in cloud computing markets," in *International Conference on Network and Service Management (CNSM)*. Niagara Falls, Canada: IEEE, 2010, pp. 354–357.

[85] L. Wu, S. K. Garg, and R. Buyya, "Sla-based resource allocation for software as a service provider (saas) in cloud computing environments," in *11th International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*. Newport Beach, CA, USA: IEEE, May 2011, pp. 195–204.

[86] R. Christpher, I. Ivan Breskovic, and Schahram, "Automatic SLA matching and provider selection in grid and cloud computing environments." Las Vegas, NV, USA: ACM, Sep 2012, pp. 85–94.

[87] O. Jennifer, T. A. Victor, and M. Balazinska, "A vision for personalized service level agreements in the cloud," in *Proceedings of the Second Workshop on Data Analytics in the Cloud.* ACM, 2013, pp. 1–5.

[88] V. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level metrics to high level slas-lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments," in *International Conference on High Performance Computing and Simulation (HPCS).* Caen, France: IEEE, July 2010, pp. 48–54.

[89] D. M. Catello, S. Santonu, G. Rajeshwari, K. Zbigniew T., and I. Ravishankar K., "Analysis and diagnosis of sla violations in a production saas cloud," *IEEE Transactions on Reliability*, vol. 66, no. 1, pp. 54–75, 2017.

[90] H. Alayat, K. H. Farookh, H. Omar, B. Ravindra, C. Elizabeth, and Alexan, "Risk-based framework for sla violation abatement from the cloud service provider's perspective," *The Computer Journal*, vol. 6, no. 9, pp. 1306–1322, 2017.

[91] Y. Rahul, Z. Weizhe, K. Omprakash, S. Prabhat Ranjan, and E. Ibrahim A., "Adaptive energy-aware algorithms for minimizing energy consumption and sla violation in cloud computing," *IEEE Access*, vol. 6, no. 9, pp. 55 923–55 936, 2018.

[92] Z. Shengli, W. Lifa, and C. Jin, "A privacy-based sla violation detection model for the security of cloud computing," *China Communications*, vol. 15, no. 9, pp. 155–165, 2017.

[93] X. Yuan, Y. Li, T. Jia, T. Liu, and Z. Wu, "An analysis on availability commitment and penalty in cloud sla," in *IEEE 39th Annual Computer Software and Applications Conference.* Taichung, Taiwan: IEEE, July 2015, pp. 914–919.

[94] A. Kumar and S. Bawa, "A comparative review of meta-heuristic approaches to optimize the sla violation costs for dynamic execution of cloud services," *Soft Computing*, p. 3909–3922, 2020.

[95] M. Adil, E. q. Bouchra, M. Abderrahim, and H. Abdelkrim, "A novel penalty model for managing and applying penalties in cloud computing," in *IEEE ACS 12th International Conference of Computer Systems and Applications (AICCSA).* Marrakech, Morocco: ACM, Nov 2015, pp. 85–94.

[96] K. S. Kumar and N. Jaisankar, "An automated resource management framework for minimizing sla violations and negotiation in collaborative cloud," *International Journal of Cognitive Computing in Engineering*, vol. 1, pp. 27 – 35, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2666307420300036

[97] A. Barnawi, S. Sakr, W. Xiao, and A. Al-Barakati, "The views, measurements and challenges of elasticity in the cloud: A review," *Computer Communications*, vol. 154, pp. 111 – 117, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366419319516

[98] G. Gao, H. Hu, Y. Wen, and C. Westphal, "Resource provisioning and profit maximization for transcoding in clouds: A two-timescale approach," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 836–848, 2017.

[99] M. Hadji and D. Zeghlache, "Mathematical programming approach for revenue maximization in cloud federations," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 99 –111, 2015.

[100] A. Upadhyay and P. Lakkadwala, "Migration of over loaded process and schedule for resource utilization in cloud computing," in *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*. Noida, India: IEEE, December 2015, pp. 1–4.

[101] Y. Li, J. Zhang, Q. Hu, and J. Pei, "Research and practice on the theory of private clouds migration," in *13th International Conference on Signal Processing (ICSP)*. Chengdu, China: IEEE, Nov 2016, pp. 1813–1818.

[102] S. Amit and C. Zheng, "Adaptive service offloading for revenue maximization in mobile edge computing with delay-constraint," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3864 – 3872, 2019.

[103] D. Hou, H. Liusheng, and X. Hongli, "Revenue maximization for dynamic expansion of geo-distributed cloud data centers," pp. 1–13, 2018.

[104] A. Shakarami, M. Ghobaei-Arani, and M. Masdari, "A survey on the computation offloading approaches in mobile edge/cloud computing environment: A stochastic-based perspective," *Grid Computing*, vol. 4, p. 3909–3922, 2020.

[105] A. Jyoti, M. Shrimali, and S. Tiwari, "Cloud computing using load balancing and service broker policy for it service: a taxonomy and survey," *Journal Ambient Intell Human Computing*, no. 11, p. 4785–4814, 2020.

[106] S. Olariu, "A survey of vehicular cloud research: Trends, applications and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2648–2663, 2020.

[107] A. Raqual. A., M. Glaucu. H., and M. D. G. Gilberto., "Factors influencing customer satisfaction in software as a service (saas): Proposal of a system of performance indicators," *IEEE LATIN AMERICA TRANSACTIONS*, vol. 15, no. 8, pp. 1536–1541, 2017.

[108] S. Hamsanandhini and R. Mohana, "Maximizing the revenue with client classification in cloud com-

puting market," in *International Conference on Computer Communication and Informatics (ICCCI)*. Coimbatore, India: IEEE, Jan 2015, pp. 1–7.

[109] T. H. Tram and T. Chen-Khong, "An auction-based resource allocation model for green cloud computing," in *IEEE International Conference on Cloud Engineering*. Redwood City, CA, USA: IEEE, March 2013, pp. 269–279.

[110] S. Manzoor, A. Taha, and N. Suri, "Trust validation of cloud iaas: A customer-centric approach," in *Trustcom/BigDataSE SPA, IEEE*. Tianjin, China: IEEE, August 2016, pp. 97–104.

[111] J. Mei, K. Li, and K. Li, "Customer-satisfaction-aware optimal multiserver configuration for profit maximization in cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 1, pp. 17–29, 2017.

[112] A. Hussain, J. Chun, and M. Khan, "A novel customer-centric methodology for optimal service selection (moss) in a cloud environment," *Future Generation Computer Systems*, vol. 105, pp. 562 – 580, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X19300950

[113] L. Wu and R. Buyya, "Service level agreement (sla) in utility computing systems," *IGI Global*, vol. 15, pp. 1–26, 2012.

[114] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: Challenges, approaches, and open issues," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020.

[115] M. Sharma and S. N., "A review and comparative study of framework for e-commerce application service on to the public cloud environment," *Rising Threats in Expert Applications and Solutions. Advances in Intelligent Systems and Computing*, vol. 1187, pp. 111 – 117, 2020.

[116] S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA: IEEE, Jan 2015, pp. 814–819.

[117] Y. Balagoni and R. R. Rao, "Locality-load-prediction aware multi-objective task scheduling in the heterogeneous cloud environment," *Indian Journal of Science and Technology*, vol. 10, no. 9, pp. 1–9, 2017.

[118] H. Yuan, J. Bi, W. Tan, and B. H. Li, "Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 337–348, 2017.

[119] E. Ibrahim, N. A. El-Bahnasawy, and F. A. Omara, "Task scheduling algorithm in cloud computing environment based on cloud pricing models," in *World Symposium on Computer Applications & Research (WSCAR)*. IEEE, 2016, pp. 65–71.

[120] I. Mansour, E. A., K. Cooper, and H. Bouchachia, "Effective live cloud migration," in *4th International Conference on Future Internet of Things and Cloud (FiCloud)*. Vienna, Austria: IEEE, Aug 2016, pp. 1–1.

[121] I. Santikarama and A. A. Arman, "Designing enterprise architecture framework for non-cloud to cloud migration using togaf, ccrm, and crmm," in *International Conference on ICT For Smart Society (ICISS)*. Surabaya, Indonesia: IEEE, July 2016, pp. 32–37.

[122] K. Tsakalozos, V. Verroios, M. Roussopoulos, and A. Delis, "Live vm migration under time-constrains in share-nothing iaas-clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2285–2298, 2017.

[123] Z. Qi, Z. Quanyan, and B. Raouf, "Dynamic resource allocation for spot markets in cloud computing environments," in *Fourth IEEE International Conference on Utility and Cloud Computing*. Marrakech, Morocco: ACM, Nov 2011, pp. 85–94.

[124] L. Tevi Yombame and D. Zbigniew, "Economic framework for resource management in data centers," in *IEEE International Conference on Communication Systems (ICCS)*. Shenzhen, China: ACM, Dec 2016, pp. 1–4.

[125] I. Hamzaoui, B. Duthil, and V. Courboulay, "A survey on the current challenges of energy-efficient cloud resources management," *SN Computer Science*, vol. 73, no. 1, 2020.

[126] B. Afzal, G. Anwar, A. Azeem, and K. Saru, "Smart workload migration on external cloud service providers to minimize delay, running time and transfer cost," *International Journal of Communication Systems*, vol. 34, p. 3, 2020.

[127] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186 080–186 101, 2020.

[128] Q. Zhang and R. Boutaba, "Dynamic workload management in heterogeneous cloud computing environments," in *Network Operations and Management Symposium (NOMS), IEEE*. Krakow, Poland: IEEE, May 2014, pp. 1–1.

[129] A. N. Toosi, K. Vanmechelen, K. Ramamohanarao, and R. Buyya, "Revenue maximization with optimal capacity control in infrastructure as a service cloud markets," *IEEE transactions on Cloud Computing*, vol. 3, no. 3, pp. 261–274, 2015.

[130] Y. Chi, X. Li, X. Wang, V. C. Leung, and A. Shami, "A fairness-aware pricing methodology for revenue enhancement in service cloud infrastructure," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1006 – 1017, 2015.

[131] X. Hong and L. Baochun, "Dynamic cloud pricing for revenue maximization," *IEEE Transaction on Cloud Computing*, vol. 1, no. 2, pp. 158–172, 2013.

[132] Y. Zhao, R. Calheiros, J. Bailey, and R. Sinnott, "Sla-based profit optimization for resource management of big data analytics-as-a-service platforms in cloud computing environments," in *International Conference on Big Data (Big Data)*. Washington, DC, USA: IEEE, December 2016, pp. 432–441.

[133] L. Tang and H. Chen, "Joint pricing and capacity planning in the iaas cloud market," *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 57 – 70, 2014.

[134] R. Safraz and S. Wayne, "An approximation algorithm for sharing-aware virtual machine revenue maximization," in *IEEE Transactions on Services Computing*. IEEE, Dec 2018, pp. 1–1.

[135] S. Snehanshu, S. Jyotirmoy, D. Avantika, and R. Ranjan, "A novel revenue optimization model to address the operation and maintenance cost of a data center," *A journal of cloud computing*, pp. 1–23, 2016.

[136] A. Deldari and A. Salehan, "A survey on preemptible iaas cloud instances: challenges, issues, opportunities, and advantages," *Journa of Computer Science*, p. 489–507, 2020.

[137] X. Hong and L. Baochun, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case," in *IEEE International Conference on Communications (ICC)*. Ottawa, ON, Canada: ACM, June 2012, pp. 2929–2933.

[138] P. Rimba, A. Tran, and I. Weber, "Quantifying the cost of distrust: Comparing blockchain and cloud services for business process execution," *Information Systems Frontiers*, vol. 22, p. 489–507, 2020.

[139] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Efficient datacenter resource utilization through cloud resource overcommitment," in *Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Hong Kong, China: IEEE, April 2015, pp. 330–335.

[140] K. Metwally, K. Abdallah, and Ahmed, "A cost-efficient qos-aware model for cloud iaas resource allocation in large datacenters," in *4th International Conference on Cloud Networking (CloudNet)*. Niagara Falls, ON, Canada: IEEE, October 2015, pp. 38–43.

[141] P. Samimi, Y. Teimouri, and M. Mukhtar, "A combinatorial double auction resource allocation model in cloud computing," *Information Sciences*, vol. 357, pp. 201–216, 2016.

[142] S. Hammoudi, A. Benaouda, S. Harous, and Z. Aliouat, "Load balancing in the cloud using special-

ization," in *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual.* New York, NY, USA: IEEE, October 2016, pp. 1–7.

[143] Z. Hong, J. Hongbo, and L. Bo, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," *IEEE transaction on cloud computing*, vol. 65, no. 3, pp. 805–818, 2016.

[144] P. Prashant, R. Naela, K. Naghma, and R. Sharmistha, "A review on sla-based resource provisioning in cloud," *Advances in Intelligent Systems and Computing*, vol. 1101, pp. 449–463, 2020.

[145] H. Rongdong, J. Jingfei, and L. Guangming, "Efficient resources provisioning based on load forecasting in cloud," *The scientific world journal*, vol. 2014, pp. 1–12, October 2014.

[146] B. Afzal, S. Shahaboddin, G. Anwar, and C. Anthony, "Optimizing iaas provider revenue through customer satisfaction and efficient resource provisioning in cloud computing," *IET Communications*, vol. 13, no. 9, p. 2913–2922, 2019.

[147] "Greencloud," https:/greencloud.gforge.uni.lu//, Jan. 2020.

[148] "Icancloud," https://icancloud.org/, Jan. 2020.

[149] "Cloud bus," [https://www.cloudbus.org/cloudsim//, February 3, 2019.

[150] "Source code," https://sourceforge.net/projects/cloudanalystnetbeans/, February 3, 2019.

[151] "Arcos," https://www.arcos.inf.uc3m.es/old/icancloud/Home.html, February 3, 2019.

[152] B. Afzal, J. Ateeqa, and R. Tauseef, "Performance based service level agreement in cloud computing," *Research Journal of Science and IT Management*, vol. 4, no. 4, pp. 20–31, 2015.

[153] "Google cloud prices," https://cloud.google.com/compute/all-pricing//, 2019.

[154] K. Alexander and L. Heiko, "The web service level agreement: Specifying and monitoring service level agreement for web services," *Journal of Network and Systems Management*, vol. 11, no. 1, pp. 57–81, 2003.

[155] "Sales force cloud computing," [https://www.salesforce.com/products/sales-cloud/pricing/, July 4, 2017.

[156] R. Benay Kumar, S. Avirup, K. Sunirmal, and R. Sarbani, "Toward maximization of profit and quality of cloud federation: solution to cloud federation formation problem," *The Journal of Supercomputing*, vol. 75, no. 2, p. 885–929, 2019.

[157] P. PritiKumari, "A survey of fault tolerance in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, pp. 1–18, 2018.

[158] S. Kinza, K. Bilal, S. Farah, Q. Sameer, and M. Muhammad, "Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios," *Antenna and Propagation for 5G and Beyond*, vol. 8, no. 1, pp. 2169–3536, 2020.

[159] C. Lalit, Majitar, and B. Rabindranath, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16 – 32, 2020.

[160] G. Davis, "2020: Life with 50 billion connected devices," in *IEEE International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, USA: IEEE, Jan 2018, pp. 1–1.

[161] "Annual forecast of data-sphere," www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/#59c815325459//, 2019.

[162] G. Lian, W. Chen, and S. Huang, "Cloud-based online ageing monitoring for iot devices," *IEEE Access*, vol. 7, no. 1, pp. 135 964–135 971, 2019.

[163] J. H. Anajemba, Y. Tang, J. A. Ansere, and C. Iwendi, "Performance analysis of d2d energy efficient iot networks with relay-assisted underlaying technique," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 3864–3869.

[164] S. Jian, Z. Tianqi, H. Debiao, Z. Yuexin, S. Xingming, and X. Yang, "Block design-based key agreement for group data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 6, pp. 996 – 1010, 2020.

[165] "Amazon cloud watch," http://aws.amazon.com/cloudwatch/, 2019.

[166] "Rack space cloud monitoring," http://goo.gl/4BfqVf, 2019.

[167] "Paraleap azurewatch," https://www.paraleap.com/AzureWatch/, 2019.

[168] "Nagios monitoring platform," http://www.nagios.org/, 2019.

[169] "Zabbix monitoring platform," http://www.zabbix.com/, 2019.

[170] "Icinga monitoring platform," https://www.icinga.org/, 2019.

[171] "Zenoss monitoring platform," http://www.zenoss.com/, 2019.

[172] V. Persico, A. Montieri, and A. Pescapé, "Cloudsurf: A platform for monitoring public-cloud networks," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 2016, pp. 1–6.

[173] A. Badshah, A. Ghani, A. Irshad, H. Naqvi, and S. Kumari, "Smart workload migration on external cloud service providers to minimize delay, running time, and transfer cost," vol. 34, no. 3. Wiley Online Library, 2021, p. e4686.

[174] F. Palumbo, G. Aceto, A. Botta, D. Ciuonzo, V. Persico, and A. Pescape, "Characterizing cloud-to-user latency as perceived by aws and azure users spread over the globe," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[175] F. Palumbo, G. Aceto, A. Botta, D. Ciuonzo, V. Persico, and A. Pescapé, "Characterization and analysis of cloud-to-user latency: The case of azure and aws," *Computer Networks*, vol. 184, no. 1, pp. 1–13, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128620312962

[176] V. Persico, A. Botta, A. Montieri, and A. Pescape, "A first look at public-cloud inter-datacenter network performance," in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7.

[177] A. Ghani, A. Badshah, S. Jan, A. A. Alshdadi, and A. Daud, "Issues and challenges in cloud storage architecture: a survey," 2020.

[178] S. Meng and L. Liu, "Enhanced monitoring-as-a-service for effective cloud management," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1705–1720, 2013.

[179] "Micro soft azure speed," http://www.azurespeed.com/, March 20, 2019.

[180] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2010, vol. 87.

[181] N. Matrazali, M. H., R. Uda, T. Kinoshita, and M. Shiratori, "Vulnerability analysis using network timestamps in full virtualization virtual machine," in *International Conference on Information Systems Security and Privacy (ICISSP)*. Angers, France: IEEE, 2015, pp. 83–89.

[182] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," *Algorithms and architectures for parallel processing*, vol. 4, pp. 13–31, 2010.

[183] J. Seol, S. Jin, D. Lee, J. Huh, and S. Maeng, "A trusted iaas environment with hardware security module," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 343–356, 2016.

[184] A. Badshah, A. Ghani, S. Shamshirband, G. Aceto, and A. Pescapè, "Performance-based service-level agreement in cloud computing to optimise penalties and revenue," *IET Communications*, vol. 14, no. 7, pp. 1102–1112, 2020.

[185] V. Persico, P. Marchetta, A. Botta, and A. Pescapè, "On network throughput variability in microsoft azure cloud," in *2015 IEEE Global Communications Conference, GLOBECOM*

2015, San Diego, CA, USA, December 6-10, 2015. IEEE, 2015, pp. 1–6. [Online]. Available: https://doi.org/10.1109/GLOCOM.2014.7416997

[186] V. Persico, A. Botta, P. Marchetta, A. Montieri, and A. Pescapè, "On the performance of the wide-area networks interconnecting public-cloud datacenters around the globe," *Comput. Networks*, vol. 112, pp. 67–83, 2017. [Online]. Available: https://doi.org/10.1016/j.comnet.2016.10.013

[187] K. Priti and K. Parmeet, "A survey of fault tolerance in cloud computing," *Journal of King Saud University - Computer and Information Sciences*, pp. 1–18, 2018.

[188] S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV, USA: IEEE, Jan 2015, pp. 814–819.

[189] M. Adil, E. q. Bouchra, M. Abderrahim, and H. Abdelkrim, "A novel penalty model for managing and applying penalties in cloud computing," in *IEEE ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. Marrakech, Morocco: ACM, Nov 2015, pp. 85–94.

[190] B. Anton and B. Rajkumar, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Wiley InterScience (www.interscience.wiley.com)*, vol. 24, no. 13, pp. 1397–1420, 2012.

[191] S. Singh and M. Arora, "Monitoring and controlling multi level sla in cloud environment using agent," in *International Journal of Advanced Research in Computer Science and Software engineering*, vol. 3. New York, NY, United States: ACM, 2013, pp. 1–7.

[192] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, 2010, pp. 446–452.

[193] P. Matthias, "5g is coming around the corner [mobile radio]," *IEEE Vehicular Technology Magazine*, vol. 4, no. 1, pp. 4 – 10, 2019.

[194] G.-A. Juan, A. C. Jose, and D. V. Wladimiro, "Tasmon: Monitoring architecture for public cloud computing data centers," *Journal of Grid Computing*, pp. 283–297, 2016.

[195] A. C. Jose and G. A. Juan, "Monpaas: An adaptive monitoring platformas a service for cloud computing infrastructures and services," *IEEE Transactions on Services Computing* , vol. 8, no. 1, pp. 283–297, 2015.

[196] T. Demetris, P. George, and D. Marios, "Monitoring elastically adaptive multi-cloud services," *IEEE Transactions on Cloud Computing* , vol. 6, no. 3, pp. 800 – 814, 2018.

[197] A. Khalid, R. Rajiv, P. J. Prem, M. Karan, L. Chang, R. Fethi, G. Dimitrios, and W. Lizhe, "Cross-layer multi-cloud real-time application qos monitoring and benchmarking as-a-service framework," *IEEE Transactions on Cloud Computing* , vol. 7, no. 1, pp. 48 – 61, 2019.

[198] D. Min and L. Feifei, "Cross-layer multi-cloud real-time application qos monitoring and benchmarking as-a-service framework," *IEEE Transactions on Cloud Computing* , vol. 28, no. 8, pp. 2172 – 2189, 2017.

[199] S. Hassan Jamil, G. Abdullah, N. Fariza, Hanum, and N. Anjum, "Cloudprocmon: A non-intrusive cloud monitoring framework," *IEEE Access* , vol. 6, pp. 44 591 – 44 606, 2018.

[200] S. Sahil, I. Canturk, K. Ricardo, and L. Eyal de, "Touchless and always-on cloud analytics as a service," *IBM Journal of Research and Development*, vol. 60, no. 2-3, pp. 11:1 – 11:10, 2016.

[201] A. Ali, S. Anca, and B. Ali Raza, "Anatomy of cloud monitoring and metering: A case study and open problems," in *6th Asia-Pacific Workshop on Systems*, vol. 6. New YorkNYUnited States: ACM, 2015, pp. 1–7.

[202] A. C. Zhou, B. He, and C. Liu, "Monetary cost optimizations for hosting workflow-as-a-service in iaas clouds," *IEEE transactions on cloud computing*, vol. 4, no. 1, pp. 34–48, 2016.

[203] X. Deng, T. Xiao, and K. Zhu, "Learn to play maximum revenue auction," vol. 7, no. 4, 2019, pp. 1057–1067.

[204] M. Mario and G. Jordi, "Client classification policies for sla enforcement in shared cloud datacenters," IEEE, Ottawa, ON, Canada, pp. 156–163, May 2012.

[205] X. Wang, J. Sun, H. Li, C. Wu, and M. Huang, "A reverse auction based allocation mechanism in the cloud computing environment," *Applied Mathematics & Information Sciences*, vol. 7, no. 1, pp. 75–84, 2013.

[206] J. Seol, S. Jin, D. Lee, J. Huh, and S. Maeng, "A trusted iaas environment with hardware security module," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 343–356, 2016.

[207] H. Xu and B. Li, "Maximizing revenue with dynamic cloud pricing: The infinite horizon case," in *IEEE International Conference on Communications (ICC)*. Ottawa, ON, Canada: IEEE, June 2012, pp. 2929–2933.

[208] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed systems*, vol. 22, no. 6, pp. 931–945, 2011.

[209] C. Redl, I. Breskovic, I. Brandic, and S. Dustdar, "Automatic sla matching and provider selection in

grid and cloud computing markets," in *13th International Conference on Grid Computing*. Beijing, China: IEEE, Sep 2012, pp. 1–1.

[210] F. Zhu, H. Li, and J. Lu, "A service level agreement framework of cloud computing based on the cloud bank model," in *International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 1. Zhangjiajie, China: IEEE, May 2012, pp. 1–1.

[211] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau, "Online auctions in iaas clouds: Welfare and profit maximization with server costs," vol. 25, no. 2, pp. 1034 – 1047, 2015.

[212] "Green cloud," [https://greencloud.gforge.uni.lu/, February 3, 2019.

[213] M. Jing, L. Kenli, T. Zhao, L. Qiang, and L. Keqin, "Profit maximization for cloud brokers in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 190 – 203, 2019.

[214] M. Hero, S. Rosli, and H. Amir, "A survey on cloud computing security," *Archives des Science*, vol. 56, no. 6, pp. 2–9, 2012.

[215] J. M. Shah, K. Kotecha, S. Pandya, D. Choksi, and N. Joshi, "Load balancing in cloud computing: Methodological survey on different types of algorithm," in *International Conference on Trends in Electronics and Informatics (ICEI)*. IEEE, 2017, pp. 100–107.

[216] W. Ma and J. Zhang, "The survey and research on application of cloud computing," in *7th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2012, pp. 203–206.

[217] "Amazon web services," https://aws.amazon.com/, 2019.

[218] W. Schools, "History of cloud computing," https://www.w3schools.in/cloud-computing/history-of-cloud-computing/.

[219] I. R. Joseph and S. B. Keri. (2019) Cloud computing in advertising and marketing. https://www.internationallawoffice.com/.

[220] A. Sidra, U. Saif, K. Abid, A. Mansoor, A. Adnan, and K. K. Muhammad, "Information collection centric techniques for cloud resource management: Taxonomy, analysis and challenges," *Journal of Network and Computer Applications*, vol. 100, pp. 80–94, 2019.

[221] Y. Mengdi, C. Donglin, and J. Shang, "Optimal overbooking policy for cloud service providers: Profit and service quality," *IEEE Access*, vol. 7, pp. 96 132 – 96 147, 2019.

[222] D. George, K. Iordanis, and S. George D, "Cloud federations: Economics, games and benefits," *IEEE/ACM Transactions on Networking*, vol. 27, pp. 2111 – 2124, 2019.

[223] S. Mehdi, T. Hamid, A. Ejaz, G. Abdullah, and K. Muhammad Khurram, "A review on remote data auditing in single cloud server: Taxonomy and open issues," *Journal of Networks and Computer Applications*, vol. 43, no. 14, pp. 121–141, 2014.

[224] H. Deng, L. Huang, H. Xu, X. Liu, P. Wang, and X. Fang, "Revenue maximization for dynamic expansion of geo-distributed cloud data centers," vol. 8, no. 3, 2020, pp. 899–913.

[225] H. Tabrizchi and M. Kuchaki Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *Super computer*, p. 9493–9532, 2020.

[226] C. Peijin, X. Guo, W. Tongquan, and L. Keqin, "A survey of profit optimization techniques for cloud providers," *ACM Computing Surveys*, no. 26, 2020.

[227] A. Mohammad and H. Eui-Nam, "Cloud broker service-oriented resource management model," *Transaction on Emerging Telecommunication Technology*, vol. 28, no. 2, 2017.

[228] A. Mahboobeh, T. Hassan, and T. Ali, "A proposed computation, which benefits from the cooperation of dew, edge, and cloud computations," *Transaction on Emerging Telecommunication Technology*, vol. 31, no. 2, 2019.

[229] B. Denis M., G. Alexei A., and N. Per Jonny, "Optimization-based profitability management tool for cloud broker," *Transaction on Emerging Telecommunication Technology*, vol. 30, no. 3, 2019.

[230] S. Rayane El, G. Nader, A. Jacques Bou, and D. Jacques, "A survey on access control mechanisms for cloud computing," *Transaction on Emerging Telecommunication Technology*, vol. 30, no. 2, pp. 1–21, 2020.

[231] F. Shakeel and S. Sharma, "Green cloud computing: A review on efficiency of data centres and virtualization of servers," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. ieee, 2017, pp. 1264–1267.

[232] M. Kandpal, M. Gahlawat, and K. Patel, "Role of predictive modeling in cloud services pricing: A survey," in *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*. IEEE, 2017, pp. 249–254.

[233] V. Persico, A. Montieri, and A. Pescapè, "On the network performance of amazon S3 cloud-storage service," in *5th IEEE International Conference on Cloud Networking, Cloudnet 2016, Pisa, Italy, October 3-5, 2016*. IEEE, 2016, pp. 113–118. [Online]. Available: https://doi.org/10.1109/CloudNet.2016.16

[234] G. Aceto, A. Botta, W. de Donato, and A. Pescapè, "Cloud monitoring: A survey,"

Comput. Networks, vol. 57, no. 9, pp. 2093–2115, 2013. [Online]. Available: https://doi.org/10.1016/j.comnet.2013.04.001