An Optimal Broker Design for Inter-Cloud Systems

by

Shraddha Reddy Peesary

BTech, Jawaharlal Nehru Technological University, Hyderabad, India, 2012

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

An Optimal Broker Design for Inter-Cloud Systems

Shraddha Reddy Peesary
MSc, Computer Science, Ryerson University, 2015

Abstract

In the next generation of Cloud computing systems, it is expected that multiple Cloud Service Providers (CSPs) will cooperate together to advertise their services and prices to their end users, which may choose the one that best meets their budgetary and technical needs. Despite this benefit of having multiple CSPs to select from, several issues may arise. For instance, how does an IT entrepreneur select a CSP to offload his/her service request? How does the underlying Inter-Cloud system handle this service request? To address these questions, this thesis proposes a novel Optimal Cloud Broker design for Inter-Cloud Systems in the form of a Semi-Markov Decision Process (SMDP) based model. Under the long-run expected average cost criterion, the optimal policy is derived, which aim at maximizing the overall virtual machine utilization while giving the end users the best possible prices. The effectiveness of the proposed Broker design is validated by numerical results.

Acknowledgment

Foremost, I would like to express my sincere gratitude to my supervisor, Professor Isaac Woungang, and my co-supervisor, Dr. Glaucio H. S. Carvalho , for their continuous support, patience, motivation, enthusiasm and time throughout my graduate studies. Their guidance helped me throughout my research and in writing this thesis. It was a great privilege to work with them. My gratitude also goes to the Department of Computer Science and the School of Graduate Studies at Ryerson University for the timely financial assistance. A special thanks to my family for their support.

Contents

List of Figures

List of Abbreviations

API	Application Program Interface
CSP	Cloud Service Provider
IaaS	Infrastructure-as-a-Service
MCC	Mobile Cloud Computing
PaaS	Platform-as-a-Service
PAUG	Pay As You Go
QoS	Quality of Service
RAS	Resource Allocation Strategy
SaaS	Software-as-a-Service
SCGM	System Cloud Grey Model
SERA	Semantically Enhanced Resource Allocation
SIP	Stochastic Integer Programming
SLA	Service Level Agreement
SLO	Service Level Objective
SMDP	Semi-Markov Decision Process
VM	Virtual Machine

Chapter 1

Introduction

1.1 Context and Motivation

Cloud computing is an emerging technology in which computing resources such as memory, processing and storage are managed through the Internet. Despite the users of such systems, a cloud service provider (CSP) owns all the resources and manages them in order to accomplish all the tasks as requested by the user.

The cloud computing technology has become popular because of its intrinsic features such as affordability (in terms of the price that the user has to pay for utilizing the resources in the cloud, often added hardware or software costs), accessibility (in terms of availability of user's data in the cloud at any time, from any location, using devices such as laptops, desktops, tablets or mobile phones), scalability (in terms of the ability to reduce or increase the resources upon the user's request whenever the user wishes for). The only requirement is that of securing an Internet connection.

Based on the type of entity (person, machine, etc) who has requested access to the cloud resources and the ability of the system to provide specific kind of services, cloud computing operates in two kinds of models, so-called deployment model and service model.

- Deployment Model

A deployment model of cloud can be considered as the architectural cloud model at each stage of the cloud computing technology. The most common cloud deployment models are private cloud, public cloud, community cloud and hybrid cloud. According to [1], a private cloud is run by a single organization who has built it, installed its own infrastructure on it and maintained all its functions either by itself or via an external service provider. However, the services are solely run by the cloud itself, and for high performance, security, and reliability purposes, a virtualization technology is often used, which often incur higher costs in hardware and software installation . But this prospect is always been condemned for its process of encountering the costs incurred in hardware, software installation, maintenance, and administration. On the other hand, a public cloud offers the cloud computing services to the public or some group of users. It is built and managed by some external CSPs. These clouds may lack in data and network control. A community cloud model is the cloud which is shared by a combination of multiple organizations that have common policies and requirements. Its infrastructure is often controlled by its partner organizations. A hybrid cloud is nothing but a kind of inter-cloud which is a combination of two or more cloud deployment models (e.g.: a combination of private and public clouds, of a private and community clouds, to name a few). It is often managed by a unique organization, and it also maintains some standard technologies that allow the portability of the data and application between the constituent clouds. When compared to other models, a hybrid model is often designed with the goal to deliver a better security and a better flexibility of data transfer between the participant clouds.

- Service Model

  Cloud computing consists of several types of service models, and each service offered by a cloud computing comes under any of these models. The most prominent service models are known as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). According to [2], each model has its own benefits

and drawbacks.

- IaaS

  It offers the services that are associated with storage, usage of servers, operating systems, virtualization, to name a few. The infrastructure of a cloud computing system consists of data centers,a pool of VMs (resources) that are controlled by a CSP and the network that is required for communication purpose. The IaaS model is responsible for keeping track of the amount of resources used within a particular amount of time and for charging the users of those resources accordingly. Well known CSPs that offer IaaS services are Amazon EC2, HP Cloud, Joyent, Rackspace, to name a few [3].

- PaaS

  In this model, the CSPs provide access to the environment and the application program interfaces (APIs) upon the user's requests. This allows the users to develop their applications without incurring any installation or configuration cost. It also saves the development and maintenance time required by the users. Since the PaaS resides on the top of of the IaaS model, it has almost all the features of the IaaS model, which include dynamic allocating of the resources, reduced cost for investments and virtualization. The developers can also take advantage of the hardware virtualization feature of the PaaS model. Some challenges that may be encountered when operating the PaaS model are security and compatibility problems. Examples of PaaS service providers include Google App engine and Microsoft Azure [3].

- SaaS

  In this model, the CSPs allow the users to access the software or applications that exist in the cloud, but not on the user's personal device. To gain access of the application hosted by the cloud, the user needs to have access to the Internet and

a thin software such as a web browser. The challenges faced by the SaaS CSPs are security and the increase in the end-user data rate. Examples of standard SaaS applications include Gmail, Hotmail and Google Apps [3].

In this thesis, we consider a inter-cloud computing model, which consists of more than two CSPs that are combined to work together from a resource management perspective. In this model, various clouds consisting of different types of resources are inter-connected to each other and share resources between each other. According to [4], over-the-years there has been a vast increase of cloud users and a single cloud may not be able to offer all the services to a user. In order to fulfill these demands and also check for the required resources in other clouds, Inter-Cloud computing is needed. According to [5], some of the issues that have led to the introduction of Inter-cloud computing concept are briefly discussed below:

- Scalability of resources: With the increase in the size of the present applications and their demands for a service, there might be a need for additional resources in the cloud. This problem has been taken care of, by over provisioning the cloud capacity. This means, most of the times, the capacity of the cloud infrastructure is greater than the resources demanded by the system. This method may result in huge expenditures for the CSPs. The demands for various services may vary with the time. This may lead to overloading a cloud with unpredictable loads which in turn lead to service interruptions. If CSPs can dynamically scale up or down their resources through resource sharing with other clouds (Inter-cloud concept), this problem can be solved to a greater extent. This saves a lot of money, as a single CSP need not maintain any additional computing servers or resources for unexpected loads. For example, a private cloud with limited capacity can always share the resources of a public cloud as per users demand.

- Interoperability: In general, certain applications are restricted to a single enterprise cloud. In such a case, due to lack of interoperability; if a customer who is relying on a

vendor for their services is not able to move to another vendor because of some technical effort and cost, a vendor lock-in situation may occur. Inter-cloud computing model avoids this situation by allowing multiple CSPs to cooperate and deploy cross-cloud applications, thereby achieving cloud interoperability.

- Cost Efficiency and energy consumption reduction: Utilization of the resources of a CSP may always vary. Sometimes, the CSP may be underutilized and sometimes it might be over utilized. This leads to high cost and energy consumption. Using Inter-cloud computing model, during the times the CSP is underutilized, it can lease its idle resources to other CSPs; and during the times it is over utilized, it can purchase/rent the available resources from the other CSPs. This saves both cost and energy of the system.

- Legal Issues: Some CSP customers may have specific requirements on legal boundaries in which their applications can be hosted. One of the main concerns for a CSP in this case will be delivering the resources in the specific geographical locations to meet the requirements as specified by the customers. Using the interoperability feature of the Inter-Cloud Computing, a CSP can find another CSP that can meet the customer's requirements due to its datacenters location.

- Disaster Recovery: Unexpected failures in the availability of services of a cloud system may lead to service interruptions and sometimes can lead to disasters. Inter-cloud computing can overcome this problem by deploying multiple CSPs with high available services; and in case one CSP has insufficient resources, then it can share the resources from other CSP.

One of the major challenges of the Inter-Cloud Computing model is resource provisioning. The concept of resource allocation refers to the process of assigning the resources to the user's requests for processing purpose. In a inter-cloud architecture, each CSP consists of various types of resources, and the type and number of resources that should be allocated to an

incoming user's request depend on several parameters such as the user's requirements, the size of the request, the number of available resources in the CSP, to name a few. Therefore, designing a resource allocation model in inter-cloud environment is a challenging task. For this purpose, a component of the inter-cloud architecture, referred to as broker, has been introduced, with a goal to reduce the request processing time, maximizing the rewards [6], [7]. Such entity is meant to work with the individual CSPs in order to manage their identity, access and delivery of requests between them, and to ensure that the availability and performance requirements from both the users and the CSP are achieved in a timely and effective manner. The broker role is similar to that of a decision maker that uses several parameters such as geographical location of each CSP, cost incurred for the processing by the CSP of the user's request, CSP's business needs, CSP's task offload, security, to name a few, to decide on which CSP the user will be directed to request for resources in the cloud. Typically, the decision maker is in the form of resource management algorithm that assumes the existence of a service level agreement (SLA) among the considered CSP, both for operations and business perspectives. The broker also helps putting policies in place such as run only in public cloud, cannot run outside organization domain, run only in specific geographical region, to name a few. The broker can decide to run the load based on capacity-cost trade offs such as run where cheap, run on platform with minimum run time,run at high bandwidth with performance expectations.

To the best of our knowledge, there has been very few works that addresses the problem of resource allocation in a inter-cloud computing architecture from a Markov decision modeling perspective. In this thesis, we proposed an Semi-Markov Decision Process(SMDP)-based resource allocation scheme that not only assigns the resources from the cloud, but also selects the best cloud among the available ones from which the resources have to be allocated to satisfy the user's incoming requests.

1.2 Research Problem

Cloud computing is a recent technology/platform that is expected provide on-demand computing resources and services at low cost for a variety of applications such as database applications and storage services. The infrastructure (IaaS), platform (PaaS) and software (SaaS) offered by a CSP only allow its users to benefit the resources of that cloud, excluding any user external to it. To better take advantage of the versatility and availability of the computing resources originated from various CSPs, it is required that a inter- cloud architecture be designed, with the assumption that the users of each of the component CSPs may benefit from such architecture in terms of task completion time, cost constraints, to name a few, by having the possibility of selecting the machines (VMs) that are appropriate for handling their requests. The challenge is to design a resource allocation model that will select the best possible cloud from the available ones to satisfy the user's request. This kind of problem has been investigated in the literature using different decision making approaches such as Stochastic Integer Programming (SIP) [8], System Cloud Grey Model (SCGM(1,1)), Markov model and combination of both (SCGM(1,1)-Markov) [9], Fuzzy logic [10], Greedy model [11], etc,. Most of these approaches have been proposed in the form of optimization problems for allocating the best resources from different CSPs to fulfill the user's request, based on constraints such as reducing the request processing time, reducing the cost of offloading the request in the cloud, reducing the energy consumption, to name a few. Unlike previous works, this thesis addresses the same problem by taking advantage of the concept of optimal broker to design an SMDP-based resource allocation scheme for an inter-cloud ecosystem.

1.3 Proposed Approach

Our approach consists of using the SMDP framework to formulate the resource allocation problem in inter-cloud system as an Optimal Cloud Broker design for Inter-Cloud System.

Under the long-run expected average cost criterion, the optimal control problem is analyzed and the optimal policy is derived, which maximizes the overall virtual machine utilization while giving the end users the best possible prices.

1.4 Thesis Contributions

The contributions of this thesis are twofold:

- We have formulated the resource allocation problem in inter-cloud as an SMDP-based framework, and have derived the optimal policy that determines the broker's selection of the best CSP to satisfy the user's request.
- We have validated the effectiveness of the proposed scheme and analyzed the structure of its derived optimal policy, using a inter-cloud architecture composed of two CSPs, running two types of service classes.

1.5 Thesis Outline

This thesis is organized as follows:

- **Chapter 1** introduces the motivation and contributions of this research.
- **Chapter 2** presents some background information and related works.
- **Chapter 3** describes the proposed SMDP-based resource allocation scheme.
- **Chapter 4** is devoted to the performance evaluation of the proposed resource allocation scheme.
- **Chapter 5** concludes the thesis and highlights some future works.

Chapter 2

Background and Related Works

This chapter briefly discusses about the cloud computing and the importance of resource allocation in it. It also discusses what lead to step of using a inter-cloud environment for the purpose of resource allocation. This chapter also reviews some related work regarding the discussed topics.

2.1 Resource Allocation in Inter-Cloud Computing

In todays world, cloud computing [12] is considered to be a most promising and efficient technology, which can add a further value for administration, business and also society. This paradigm can be applied to various sensitive applications/scenarios such as medical applications, governmental systems, social services and some enterprises' businesses. But, in such an environment, if the computing services are provided by only by a single cloud, then it may lead to overloaded traffic and unexpected loads which may further lead to interrupted and specious services. In order to achieve reliable service levels in such circumstances, there evolved a concept of managing multiple single clouds which can complement each others limitations. This approach is called as Inter-Cloud computing.

The concept of Inter-cloud computing was first emerged at Cisco Systems which meant the interoperability of different clouds. While suggesting a set of Inter-cloud protocols,

Bernstein et al. (2009) [13] has mentioned the term "Inter-Cloud", which meant connecting the resources of multiple CSPs. Later on, many works were contributed on this concept in order to achieve better reliability and quality of services.

In practice, a single CSP cannot offer all kinds and number of resources as required by the users. To overcome this issue, inter-cloud computing concept is been introduced, where various CSPs consisting of different types of resources are inter-connected to each other, share resources between each other, etc.

Inter-Cloud, which will be the next generation of Cloud computing systems, consists of a multi-Cloud service provider ecosystem forming what is called *Cloud market*. In a Cloud market, despite the fierce competition, CSPs collaboratively and/or cooperatively advertise their services and the associated prices to their end-users, who may choose the one that best meets its budgetary and technical needs. Despite the appeal of having multiple CSPs to choose from, which naturally solves the vendor lock-in situation, several issue may arise. For instance, how does an IT entrepreneur select a CSP to offload his/her service request. How does the underlying Inter-Cloud system handle this incoming service request?

In [14], Grozev et al. reported that depending only on a single CSP may not provide all the desired functionalities for the users who are distributed world-wide. Hence, they try to achieve better reliability, cost efficiency, flexibility and Quality of Services (QoS) of the entire system through multi-cloud concept. The most important advantage of an inter-cloud system is that distributing the load and synchronizing dynamically among a set of clouds. A CSP cannot establish all its data center in various geographical locations and accommodate the requested resources to the user's requests that easy. Instead, it can be inter-connected with other CSPs that are present in various geographical locations and share those resources. This reduces the complexity to access the resources and accommodating the resources to a greater extent.

By using multiple CSPs, the users can easily get through the vendor lock-in situation and also can easily transfer from one cloud to the other if in case they don't like the present

cloud's policies and pricing. In this way, the CSP can migrate the load from one cloud to another at the time of scarcity of resources. In order to achieve high performance, flexibility and responsiveness in inter-cloud computing system, the concept of Cloud Broker has been introduced.

The role played by the Cloud Broker is to mediate CSPs' offers and end user requirements and find a satisfactory match for both parties. Typically, a Cloud Broker has the capability to maintain multiple clouds and also participates in the allocation/deallocation of resources for the given request. A Cloud Broker also has the capacity to balance the load among the available multiple clouds.

Before brokering concept was introduced, users used to directly interact with the CSPs, where the users have to decide the type and number of resources that are to be allocated to fulfill their request. A broker takes this burden from the user. In Inter-cloud computing environment, a broker is placed between the user and multiple CSPs. Based on the certain criteria like users QoS requirements, type and price of service, the broker will decide which CSP has the suitable resources for that particular users request and the resources are allocated accordingly.

Cloud brokers have been widely studied in literature as well as the optimization of virtual machines (VM) allocation. For Cloud Brokers, the work in [14] surveys the literature while Liang *et al.* in [15] investigate an optimal VM allocation for mobile cloud computing (MCC) service providers. Although service migration between cloud computing centers is supported in the model, they neglected the fact that the VM occupancy fluctuates over time in each neighbor CSP. [16] proposes an architecture for MCC over wireless networks where Cloudlets are integrated to base stations to provide traffic conformance between the Internet and wireless networks as well as to support the multimedia session handoff between different cells. In [17], trustworthiness and competence are the tenets of the so-called SelCSP framework.

In [18] an inter-cloud computing architecture is proposed that allows the user to utilize multiple cloud systems. Because there is a high flexibility of exchanging the resources in an

inter-cloud computing environment, during any failure, any failed resource or cloud can be replaced by the another immediately. As per the author's comments, this concept provides reliability and quality in the infrastructure.

According to [19], the Cloud Brokers act as intermediates between the cloud users and CSPs. There may exist different relationships between the CSPs and a broker, cloud user and a broker like one-to-one, one-to-many, many-to-one and any-to-many. A cloud broker doesn't own or change any services of a CSP, it only manages the already present services and the relationship between each entity in the multiple cloud environment. Cloud brokers can be classified into three groups based on the functionality provided by them, namely:

- Service Aggregation: A cloud broker can combine different services of a CSP into a single service as per the user's requirements and it can also allow a secure data transfer between the CSPs and the users.

- Service Intermediation: A cloud broker can provide the increased potentials of a service to the user. It can also provide additional service values to the users.

- Service Arbitrage: This is similar to the aggregation of service. Using this feature, the cloud broker can integrate some services together.

The cloud broker can provide additional service values such as:

- Managing and monitoring some tasks such as resource management, workload, scheduling the user's requests and policy-based automation.

- Allocating the resources from a CSP to a user, and ensuring the security and privacy of the data.

- Offering the services based on the location-specific and domain-specific requirements of the cloud environment.

- Ensuring the Service Level Agreement (SLA) management. A SLA acts like a contract between a CSP and a user. The contracts tell us what kind of service is offered by the

CSP, negotiations, costs, etc. Every CSP will have their self-defined SLAs. Whenever a user wants to use the resources of a particular CSP, he/she should first go through their SLAs. If they are satisfied with the agreement, then the users can proceed further. SLAs also consists of the QoS requirements of the users, which indeed helps the CSPs to assign the resources that are suitable to those QoS and hence can achieve the higher user satisfaction.

According to [20], the SLA provides some guarantees in terms of performance to the users in the form of Service Level Objectives (SLOs) such as throughput, response time, availability, various pricing models, etc. The amount of cost paid to the CSP by the user depends on the level of performance gained by them. One of the important challenges here is to determine the number of resources that are required to satisfy the user's requirements specified under the SLA.

According to [5], CSPs present what they guarantee in a SLA. It consists of many details like description of a service, QoS expectations, penalties implied on CSPs if they does not provide services as per QoS requirements. In Inter-cloud computing environments, each CSP should have their own SLA management procedures. Since users share resources from various CSPs in such environments, there is a need of enforcing a global SLA. Here, a global SLA means extensive SLAs between Inter-cloud (includes SLA for each CSP) and the user. In a dynamic environment like Inter-cloud, depending on the demand for resources, the service provided to the user might be the combination of multiple services from different CSPs. In such dynamic environments, having protocols for negotiating the SLAs is must. Despite having a service as an invariable property, this appeals that the SLAs are dynamically enacted at the time of request for service.

There are various challenging aspects in cloud computing that are to be dealt with such as resource allocation, handling enormous amounts of data, managing host servers, minimizing the energy consumption, Security and Privacy issues, migration of virtual machines, traffic management and analysis, etc. Among all, the aspect discussed in this thesis is resource

allocation.

According to [21], if the allocation of resources is not managed efficiently, then it would lead to the starvation of the requests. In order to avoid this, an efficient process called Resource Allocation Strategy (RAS) that considers both utilization and allocation of resources within the environment of cloud is designed. In order to accomplish the user's request, RAS also examines the type and amount of resources that are needed by each application request. The time and sequence of resources allocation are also taken as an input for optimal RAS [22]. An optimal RAS must avoid the clauses like offering more or less number of resources than demanded by the user. A situation like scarcity of resources should never arise. This generally occurs when the demand of resources are higher than the number of resources. Sometimes, even though there are enough number of resources, they cannot be allocated to the users requests because of the fragmentation of resources. Care must be taken to eradicate such states. RAS should also censure resource contention which generally occurs when two application requests try to access the same resources at the same time.

According to [23], the performance, cost and functionality are directly or indirectly affected by the resource management of the system. To deal with the complexity of the system and maintain the sharing of resources, certain policies need to be maintained. In [23], resource management policies can be classified into five classes, namely, admission control, balancing the load, Quality of Service (QoS), optimization of energy and resource allocation. Admission control will make sure that the system does not accept over workload, which in turn goes against the system policies. Load balancing and energy optimization policies are inter-related with each other. These both will affect the cost consumption of the system to a great extent. In general, load balancing means, the entire load should be equally disseminated among all the CSPs. The main concern about the cloud computing is to minimize the cost of accessing a resource and minimize the energy consumption of the system. In such a case, for a system, to work efficiently, the load balancing policy distributes the load in such a way that the minimum number of CSPs are used and each CSP is utilized to a maximum

extent. The QoS policy deals with the overall performance of the system.

From the above discussions, it is obvious that the process of allocating the resources to the user's requests plays an important role in cloud computing. At the same time, we should also remember that selecting the CSP for which the resources are to be allocated also plays a major role in the process of resource allocation. This selection may depend on various factors such as SLAs, energy consumed by a CSP, cost incurred by a CSP.

According to [24], cloud computing services can be purchased directly on-line. Different cloud service providers will have different access prices. These prices depend upon some criteria like various pricing plans (monthly/yearly/pay-as-you-go), SLA, number of data centers ad resources offered, certifications provided, scaling up/down of the resources, APIs, data transfer - inbound/outbound, etc,.

In this thesis, we investigate a key Cloud Broker function-Cloud selection. By formulating the Broker as a SMDP model, we propose a model that maximizes the Cloud virtual machines (VM) utilization while delivering the best prices to the end users. The optimal control problem is analyzed under the long-run expected average cost criterion, which by means of the Value Iteration Algorithm [32], the optimal policy is determined.

2.2 Related Work

Many researchers have contributed to the resource allocation problem in cloud computing. Some of them are discussed as follows.

In [15], the main concern is how to manage the resources of the entire cloud environment that are distributed across inter-cloud domains. This paper proposed a decision model called SMDP for inter-domain transfer service in order to balance the computational loads among the various cloud domains. The proposed model also considers the maximization of rewards for both the user and the cloud by minimizing the number of service rejections. Decisions for transferring the requests are based on the system's incomes and expenses. When compared

with the greedy model, the simulation results proved that the proposed decision model has been successful in increasing the rewards and decreasing the rejection of requests.

In [25], Wu et al. proposed a resource allocation model that is based on broker. Their model is mainly concerned on fulfilling the user's Quality of Service (QoS) requirements while taking the SLA into account. In their model, the broker primarily buys the resources from the CSPs and then allocates the resources to the users. Each user will have its own QoS requirements. The broker will assign the resources to the users' request based on SLA which has all the deails on each user's QoS. That means, in order to accommodate the resources, the broker considers the SLA as an on-demand aspect. Some numerical results are presented to validate the proposed model.

In [26], a resource allocation model is proposed for networked CSPs. The resource allocation problem in cloud computing is defined as a mixed integer optimization problem. Depending on the QoS requirements of various users, the resources are mapped (allocated) to their requests, well-structured and also gives high performance. The proposed model was been added over the platform for resource virtualization called FEDERICA, an European Internet test-bed.

According to [27], in cloud computing a single CSP may not be able to provide sufficient resources to all the users at all times as required. This may result in the losing of users and business for that particular CSP. To overcome this problem, the inter-cloud federation has been initiated in which a CSP is connected with the other CSPs, and the CSP which has insufficient resources can use the other cloud's resources. The resources are allocated based on the QoS requirements of the end users.

In inter-cloud computing model, there might be various kinds of resources in each cloud. In [28],the following question has been raised: when a user sends a request, how can a broker decide which resource from which serves the right purpose as required by users. As an answer, they proposed a inter-cloud model called "Inter-cloud Resource Provisioning System" that uses a resource ontology which consists of the semantic description of every resource and

the incoming requests. The requests are allocated depending on the semantic scheduler and some inference rules. The proposed model was implemented using a semantic framework called as Sesame, then validated their proposal by conducting some simulations.

In [29], a meta-broker model in inter-cloud computing environment is proposed that coordinates with all the cloud brokers considering both resources and SLAs. In the proposed model, for each user, meta-broker is generated that coordinates with local and the remaining meta-brokers. Meta-broker consists of all the infrastructure details about the characteristics of CSPs and resources. When the request arrives into the system, depending on its specifications and the availability of resources, the meta-broker allocates the resources to the request. This decreases the complexity of the system as the resource provisioning is dependent on the meta-brokers instead of the CSPs. Experimental results are provided to validate the effectiveness of their model.

In [30], an resource provisioning algorithm is proposed based on a graph clustering algorithm in a inter-cloud computing environment. In this proposed algorithm, the cloud subgraph that incurs minimum provisioning cost is selected. Then the mapping cost of virtual nodes and links is calculated. The proposed algorithm is compared with the traditional round robin algorithm, showing its superiority in terms of achieving the desired QoS requirements from the users.

By utilizing the knowledge management systems, a model for operating the inter-clouds is proposed in [31] based on a resource provisioning mechanism and some predefined rules defined by the cloud broker.

In [32], a model in which a meta-broker makes the decision of selecting a best cloud among all the coordinated inter-clouds is introduced. This decision making depends on some criteria such as energy efficiency, execution time of a service, etc,. When compared to other centralized models, the proposed decentralized model offers more flexibility and scalability.

In [33], a networking manager that allows the distribution of cloud resources was pro-

posed. The proposed system also provides the configuration capabilities and control on network. The resources that are been collected from various CSPs are interconnected by the manager based on the user requests. For the interpretation in the multiple CSPs environment, the manager was unified with the cloud broker. The proposed model extends and also supports the present state-of-art in the cloud with the control on connectivity and management.

In [34], a basic architecture for Inter-cloud computing was presented. Here, handling of various key challenges of multimedia in Inter-cloud computing environment is discussed. This paper also provides solutions to those challenges. It also discussed few design considerations for the storage on media cloud. Some initial results on storage size efficiency were presented in this paper.

In the present days of inter-cloud computing environment, there is a need for adopting the concept of schedulers for the workloads management [35]. Bessis et al. (2011) presents the needs and requirements of inter-cloud computing. They evaluated various schedulers for inter-cloud computing considering certain multi cloud environments like flexibility, distributing geographically, various SLA's compatibility, the heterogeneity of resources, etc.

To the best of my knowledge, there has been very few works that addresses the problem of resource allocation in an inter-cloud computing architecture from a Markov decision modeling perspective. In this thesis, we proposed an SMDP-based resource allocation scheme that not only assigns the resources from the cloud, but also selects the best cloud among the available ones considering maximum utilization of the resources and minimum access price.

Chapter 3

Methodologies

This chapter discusses the proposed model, its functionalities and also the decision model that is used for optimizing the proposed model.

3.1 System Model

An Inter-Cloud ecosystem having L CSPs working in collaboration is assumed. Each CSP whose capacity is L_j VMs, for all $j \in \{1, 2, ..., L\}$, supports K service classes. Additionally, assume that the service class $i \in \{1, 2, ..., K\}$ arrives into the system according to an independent Poisson process with parameter λ_i, requires b_i VMs to meet its service requirement, and demands a service time that is exponentially distributed with mean $1/b_i\mu_i$. The proposed optimal broker design model for Inter-cloud environment is given in figure 3.1.

The figure depicts the proposed system model for inter-cloud computing, which is based on optimal broker who decides to which cloud among the available interconnected clouds the request of the user to be sent. Each component of the above figure is explicitly discussed as below.

- User

  A user in the proposed model, can be any device like laptop, tablet, PC or a mobile

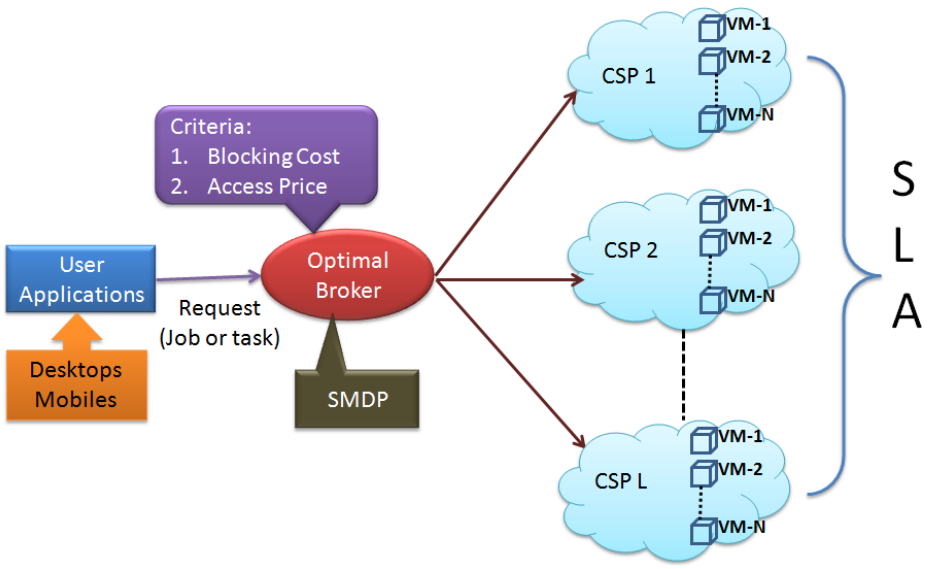

Figure 3.1: System model of Optimal Broker Design for Inter-Cloud Systems

phone from which an individual will be able to send his/her request to the cloud and also get the required result to that device from the cloud.

- CSP

  As seen in figure 3.1, the inter-cloud consists of various CSPs (they might be public, private or community clouds). Here, each CSP consists of two service classes. Each service class can accept different kinds of resources based on the user's request.

- SLA

  Various CSPs work with each other based on their Service Level Agreement (SLA) for better business. A SLA is a contract between a user and the CSPs that enumerates the amount and types of VMs and their performances, which are offered by the cloud to the users. The SLA determines the cost for the services offered by the cloud and the user pays it in a pay-as-you-go manner.

- Optimal Broker

  In this environment, the Cloud Broker collects the incoming request and selects the Cloud service provider that best matches end user needs. In this model, the optimal

broker considering the blocking cost of request and access price of the cloud, takes decision based on SMDP.

- Request

  A request is made by a user to a cloud for various services like retrieve a audio/video file, store something on cloud, etc,.

- VM

  The virtual machines/resources in the proposed model can either be CPU, database that can allow to process or store the user's data. The term 'bandwidth' in this model gives the number of VMs that has to be allocated for a single incoming request.

3.2 Optimal Control Problem

The optimal control problem relies on the use of an SMDP, a Markov model where the decision epochs are state transition epochs with random lengths. SMDP is a continuous decision making process in which exponential distribution is not mandatory. In this process, the distribution of the next state depends only on the current state. The proposed SMDP is made of the following components: system states, the actions, the expected time until next decision epoch, the state transition and the cost function.

3.2.1 System State and State Space

Let $\boldsymbol{S}$ be the state space and $s(t) \in \boldsymbol{S}$ be a state of the Inter-Cloud system at time t, where $t \in \mathbb{R}_+$. The state matrix of the considered system is

$$s(t) = [n_{ij}, \boldsymbol{\nu}]_{L+1\times K} \in \mathbb{Z}_+^{L+1\times K}, j \in \{1, \ldots, L+1\}, i \in \{1, \ldots, K\}, \tag{3.1}$$

in which n_{ij} is the number of allocated VM to the i^{th} service class into the j^{th} Cloud Computing Center. Given the service requirement of the i^{th} service class, the maximum

value of n_{ij} is given by $\lfloor \frac{L_j}{b_i} \rfloor$ [1]. $\boldsymbol{\nu}$ is a vector of size $K+1$ that specifies the last occurred event. Thus, $\nu(1) = 0$ denotes the service completion, $\nu(2) = 1$ denotes the arrival of the service class 1, and so on until $\nu(K+1) = K$. As long as each Cloud Center has a finite capacity, the summation of all ongoing service classes being supported by it must not exceed its capacity. Thus, the state space $\mathbf{S}$ is given by:

$$\boldsymbol{S} = \left\{ s \in \mathbb{Z}_{+}^{L+1 \times K} : \sum_{j=1}^{L} \sum_{i=1}^{K} b_i n_{ij} \leq L_j, \forall j \in \{1, 2, \ldots, L\} \right\}. \tag{3.2}$$

3.2.2 Decision Epochs and Actions

In a SMDP environment, the optimal broker has to make a decision after a random period of time, the so-called *decision epochs*. The natural decision epochs are the arrival instances; nonetheless, a service completion also leads to a change in the system state. Thus both events are defined as decision epochs.

Assume the system starts to operate at $t = 0$. Thus, let the events in $\boldsymbol{\nu}$ take place at the time instances $t, t = 0, 1, 2, ...$ At each decision epoch t, the optimal broker selects an action $u(t)$. In each state, the following actions are allowed: acceptance (Cloud selection) or rejection. Action $u(t) \in \boldsymbol{U}(s)$, where $\boldsymbol{U}(s)$ is the action space, is then defined as

$$u(t) = \left[u_{ij}(t) \right]_{L \times K} \in \{0, ij\}_{L \times K}, j \in \{1, \ldots, L\}, i \in \{1, \ldots, K\}, \tag{3.3}$$

where $u_{ij}(t) = 0$, for all $\nu(l) \in \{l = 1, \ldots, K+1\}$, denotes the rejection of the i^{th} service class request into the j^{th} Cloud Center while $u_{ij}(z) = ij$, for all $\nu(l) \in \{l = 2, \ldots, K+1\}$, stands for the acceptance of the i^{th} service class request into the j^{th} Cloud Center. For the sake of simplicity, the action $u_{ij}(t) = 0$ will be always chosen when $\nu(1) = 0$. In this case, however, rather than blocking it has to be interpreted as *do nothing* in as much as the corresponding event refers to as a service completion. Note that an extra action could be easily specified

[1] $\lfloor g \rfloor$ is the largest integer not greater than g

$$P_{sq}(u(t)) = \begin{cases} \lambda_i \tau_s(u(t_z)), & s = [n_{ij}, \boldsymbol{\nu}[l]], u(t) = 0, q = s, l \in \{1, \ldots, K+1\}, j \in \{1, \ldots, L\}, i \in \{1, \ldots, K\}; \\ \lambda_i \tau_s(u(t)), & s = [n_{ij}, \boldsymbol{\nu}[l]], u(t) = 1, q = s + \alpha_{ij}, l \in \{2, \ldots, K+1\}, j \in \{1, \ldots, L\}, i \in \{1, \ldots, K\}; \\ b_i n_{ij} \mu_i \tau_s(u(t)), & s = [n_{ij}, \boldsymbol{\nu}[1]], u(t) = 0, q = s - \alpha_{ij}, j \in \{1, \ldots, L\}, i \in \{1, \ldots, K\}; \\ 0, & \text{Otherwise.} \end{cases} \tag{3.5}$$

to cover such situation that despite making the SMDP model more "pedagogical" will lead to an unnecessary increase in its size.

3.2.3 Expected Time Until the Next Decision Epoch

If the system is in the state $s(t) \in \boldsymbol{S}$ and the action $u(t) \in \boldsymbol{U}(s)$ is chosen, then the expected time until the next decision epoch is given by:

$$\tau_s(u(t)) = \frac{1}{\sum_{i=1}^{K} \lambda_i + \sum_{j=1}^{L} \sum_{i=1}^{K} b_i n_{ij} \mu_i}. \tag{3.4}$$

3.2.4 Transition probabilities

Let $\alpha_{ij} \in \{0,1\}_{L \times K}$ denote a matrix containing only zeros except for the (i,j) position, which is one. In this respect, the operation $s \pm \alpha_{ij}$ represents an increase/decrease in the state variable located at the position (i,j) in s. Based on α_{ij}, the system dynamic may be completely described by determining the transition probabilities of the embedded Markov chain. To this end, let $P_{sq}(u(t))$ denote the probability that in the next decision epoch the state will be $q = s \pm \alpha_{ij}$ given that the current state matrix is $s = [n_{ij}, \boldsymbol{\nu}]$ and the action $u(t) \in \boldsymbol{U}(s)$ is taken. For all feasible $s, q \in \boldsymbol{S}$, $P_{sq}(u(t))$ is specified in Eq.(3.5).

In the proposed model, for every state (c1, c2, c3, c4, ev) and decision $a \in A(x)$, there might be three different transitions (B or AC1 or AC2).

where

c_1= Number of VMs in class 1 cloud 1

c_2= Number of VMs in class 2 cloud 1

c_3= Number of VMs in class 1 cloud 2

c_4= Number of VMs in class 2 cloud 2

$AC1$= Accepting into cloud 1

$AC2$= Accepting into cloud 2

B = Blocking the incoming request

ev = event

For example if the state is (2, 2, 1, 4, 2) and the action is AC1, then the possible transitions of the state are identified as below:

- Firstly, change the state according to the given event and action: Since the event here is 2 and the action is AC1, this means, arrival in class 2 and accepted by cloud 1. This changes the state to (2, 3, 1, 4, 0)

- Now, the possible state transitions are calculated as below:

  If next event = 1, then the possible state will be (2, 3, 1, 4, 1)

  If next event = 2, then the possible state will be (2, 3, 1, 4, 2)

  If next event = 0 and c_1>0, then the possible states will be (1, 3, 1, 4, 0)

  If next event = 0 and c_2>0, then the possible states will be (2, 2, 1, 4, 0)

  If next event = 0 and c_3>0, then the possible states will be (2, 3, 0, 4, 0)

  If next event = 0 and c_4>0, then the possible states will be (2, 3, 1, 3, 0)

3.2.5 Policy, Optimality Criterion, Cost Function, and Value Iteration Algorithm

For a given state $s(t) \in \boldsymbol{S}$, an action $u(t_z) \in \boldsymbol{U}(s)$ is selected according to a policy $\pi(s) \in \Pi$, where Π is a set of admissible policies defined as

$$\Pi = \left\{\pi : \boldsymbol{S} \rightarrow \boldsymbol{U} | \pi(s) \in \boldsymbol{U} \forall s \in \boldsymbol{S}\right\}. \tag{3.6}$$

In this thesis, we consider the average cost criterion as the optimality criterion, which is expressed as

$$g(\pi, s) = \lim_{T\to\infty} \frac{1}{T} E_s^{\pi}\left\{ \int_0^T c(s(t), u(t))dt \right\} \tag{3.7}$$

where E_s^{π} the expectation operator when the initial state $s_0 = s \in \boldsymbol{S}$ and the policy π is used. In Equation (3.7), $c(s(t), u(t))$ is the cost function. In this work, we assume that in the cloud-to-cloud market, each service provider announces the price of its service to the cloud broker whose task is to perform the cloud selection based on the users' requirement and the advertising price. Based on it, we assume that the *service price* represents a criterion on the cost function. At the same time, Cloud computing is based on the tenet that the resource sharing leads to an increase in the system resource utilization that in turn results in a cost reduction to the end users. Thus, an important design criterion is to minimize the rejection of service requests in an environment with multiple service classes and diverse QoS profile while prioritizing the higher priority QoS service requests. Thus, we include in the cost function the *blocking cost* as a way to reinforce the need to accept incoming service requests. By doing so, we define the following cost function

$$c(s(t), u(t)) = (1 - \rho) b_i p_{ij} + \rho r_{ij} \tag{3.8}$$

where p_{ij} is the price of i^{th} service class into the j^{th} Cloud Computing Center, r_{ij} is the blocking cost of the i^{th} service class into the j^{th} Cloud Computing Center, and ρ is a weighing factor that could be used to tune the relative importance between the used criteria.

Considering Eq.(3.7), the corresponding long-run expected average cost is given by

$$g^*(s) = \inf_{\pi\in\Pi} g(\pi, s). \tag{3.9}$$

Bearing Eq.(3.9) in mind, the optimal control problem may be stated as to find a control

policy π^* such that $g(\pi^*, s) = g^*(s)$ for all $s_0 = s \in \boldsymbol{S}$. In this thesis, the value iteration algorithm [36] is applied to derive the optimal policy. The principle behind this method is to approximate the minimal average cost through a sequence of value functions $V_n(s)$ for all $s \in \boldsymbol{S}$. The value functions provide lower and upper bounds on the minimal average cost, which iteratively converge to the minimal average cost. The value iteration algorithm is specified as follows [36]:

Step 0: Choose $V_0(s)$ such that

$$0 \leq V_0(s(t)) \leq \min_{u(t)} \{c(s(t), u(t))/\tau_{s(t)}(u(t))\}, \forall s \in \boldsymbol{S}.$$

Choose a number τ with $0 < \tau < \min_{s(t),u(t)} \tau_{s(t)}(u(t))$. Let $n := 1$.

Step 1: Compute the recursive function $V_n(s), s \in \boldsymbol{S}$, from

$$\begin{aligned} V_n(s(t)) = \min_{u(t) \in U(s)} \Bigg[& \frac{c(s(t), u(t))}{\tau_{s(t)}(u(t))} \\ & + \frac{\tau}{\tau_{s(t)}(u(t))} \sum_{q(t) \in \boldsymbol{S}} P_{sq}(u(t)) V_{n-1}(q(t)) \\ & + \left(1 - \frac{\tau}{\tau_{s(t)}(u(t))}\right) V_{n-1}(s(t)) \Bigg]. \end{aligned}$$

Let $\pi(n)$ be a stationary policy whose actions minimize the right-hand side of the recursive function.

Step 2: Compute the bounds

$$\begin{aligned} m_n &= \min_{q(t) \in \boldsymbol{S}} \{V_n(s(t)) - V_{n-1}(s(t))\} \text{ and} \\ M_n &= \max_{q(t) \in \boldsymbol{S}} \{V_n(s(t)) - V_{n-1}(s(t))\}. \end{aligned}$$

The algorithm is stopped with policy $\pi(n)$ when $0 \leq \frac{M_n - m_n}{m_n} \leq \epsilon$ where ϵ is a prespecified accuracy number. In this paper, $\epsilon = 10^{-12}$. Otherwise, go to Step 3.

Step 3: $n \coloneqq n+1$ and go to Step 1.

After a finite number of iterations, the algorithm terminates and outputs a policy $\zeta(n)$ whose the average cost function $g(\pi(n), s)$ satisfies $0 \leq \frac{g(\pi(n),s)-g^*(s)}{g^*(s)} \leq \varepsilon$ for all $s \in \boldsymbol{S}$.

The optimal policy π^* is a decision rule $f : \boldsymbol{S} \to \boldsymbol{U}$ that dictates the action $f(s) \in \boldsymbol{U}(s)$ each time the system is observed in the state $s \in \boldsymbol{S}$ [36]. Under π^*, the underlying continuous time Markov chain model is solved. To this end, its infinitesimal generator matrix $\boldsymbol{Q}$ is built following the specifications of the optimal policy. From that point on, taking into account the normalization condition $\sum_{s \in \boldsymbol{S}} \varpi(s) = 1$, one can compute the steady-state probability vector $\boldsymbol{\varpi}$ by solving the system of linear equations $\boldsymbol{\varpi Q} = 0$ using standard numerical techniques. In this paper, we have used the successive over-relaxation (SOR) method [37].

Chapter 4

Performance Evaluation

4.1 Simulation Setup

We consider an Inter-Cloud system with two Cloud Centers, a private and a public. This might be a typical scenario of a mobile service provider that has its private CSP, but also a service level agreement (SLA) with a public Cloud service provider to handle the limitation of a CSP specially when the peak demand exceeds the CSP's capacity. In such a context, it is mandatory to determine when to resort to the public Cloud to ensure the QoS provisioning at the lowest cost at long-run. The system still supports two service classes: a high priority (#1) and a low priority (#2). For numerical computation, an Inter-Cloud with the following parameters was considered:L_1 = 20 VMs, L_2 = 10 VMs, $r_{11} = r_{12} = 1$, $r_{21} = r_{22} = 0.8$, p_{11} = 0.05 monetary units (MU),p_{21} = 0.025 MU, $p_{21} = p_{22}$ = 0 MU, ρ = 0.8. λ_1 = 10 request/s, λ_2 = 15 request/s, $\mu_1 = \mu_2 = 6.6$ s^{-1}.

4.2 Performance Metrics

Let $O^{ij}_{u(t)=1}$ be the mean service completion rate of the i^{th} service class in the j^{th} Cloud Center.

$$O^{ij}_{u(t)=ij} = \sum_{s\in \boldsymbol{S}} (\sum_{i=1}^{K} \lambda_i + \sum_{j=1}^{L}\sum_{i=1}^{K} n_{ij} b_i \mu_i)\varpi(s) \tag{4.1}$$

In order to analyze the performance of the system in the proposed model, we considered performance metrics like blocking probability of service class 1 and 2, cloud utilization of VMs in private cloud and public cloud. These are discussed below:

- Blocking probability is the probability of blocking the incoming requests, which in turn means, processing the requests in the users' device itself once they get rejected from the cloud. This occurs whenever there are insufficient resources available for an incoming request. This means, blocking probability mainly depends upon the number of VMs that are available in a cloud. As a result, the request is returned to the user and the task is completed in the users' device itself. Given $O^{ij}_{u(t)=1}$, the blocking probability of the i^{th} service class P^i_B is derived as

$$P^i_B = 1 - \frac{O^{ij}_{u(t_z)=1}}{\lambda_i}. \tag{4.2}$$

- Utilization of VMs is defined as the percentage of VMs that are being utilized in a cloud. How well the cloud is utilized, that much the proposed model is efficient. The j^{th} Cloud Center VM utilization is defined as the ratio between the mean number of occupied VMs and the total number of VMs, which gives

$$X_j(u(t) = ij) = \frac{1}{N_j}\left(\sum_{n_{1j}=1}^{N_j} \sum_{n_{2j}=1}^{N_j} \cdots \sum_{n_{Lj}=1}^{N_j} (b_1 n_{1j} + b_2 n_{2j} + \cdots + b_{L-1} n_{Lj-1} + b_L n_{Lj}\right)\varpi(s). \tag{4.3}$$

4.3 Numerical Results

In this Section, we evaluate our proposed resource allocation scheme under several scenarios as follows.

4.3.1 Impact of Bandwidth Requirement of Service Class 1 on the Blocking Probability of Service Classes 1 and 2

The bandwidth requirement of service class 1 is increased and its impact on the blocking probability of the service class 1 (resp. service class 2) is determined. The results are captured in Fig. 4.1 and Fig. 4.2 respectively.

As shown in Fig.4.1 and Fig.4.2, the blocking probabilities of service classes #1 and #2, respectively, are quite sensitive to an increase in the number of requested VM by both service classes. It occurs because the higher b_1 and b_2 the fewer VMs are left to be used by new incoming requests. As a consequence, the blocking probabilities go up.

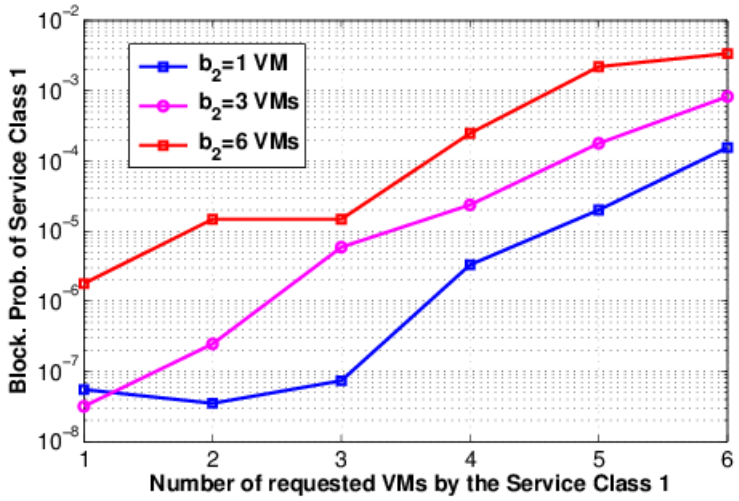

Figure 4.1: Blocking probability of the service class #1 versus the bandwidth requirement of the service classes #1.

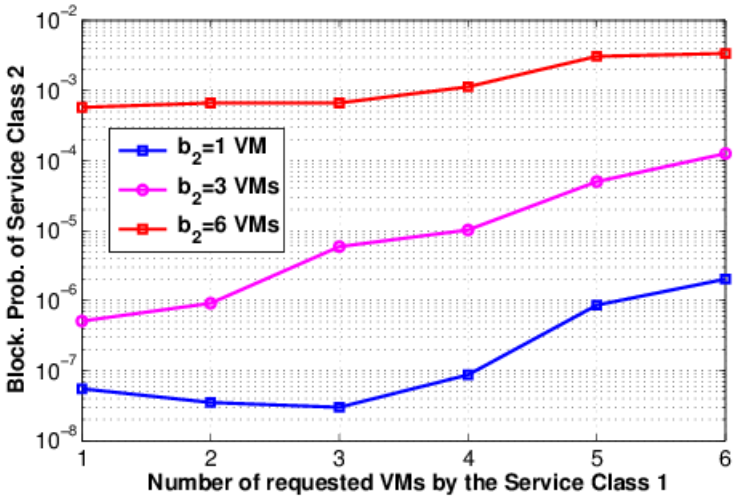

Figure 4.2: Blocking probability of the service class #2 versus the number of requested VM by the service class #1.

4.3.2 Impact of Bandwidth Requirement of Service Class 1 on the VM Utilization of Public Cloud and Private Cloud

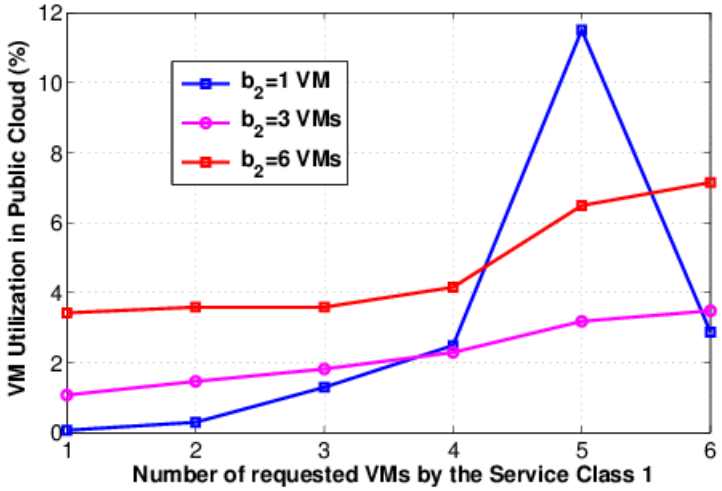

Figure 4.3: VM utilization in the Public Cloud versus the number of requested VM by the service class #1.

The bandwidth requirement of service class 1 is increased and its impact on the VM utilization of the public cloud (resp. the private cloud) is determined. The results are captured in Fig. 4.3 and Fig. 4.4 respectively. Please note that, in fact, Fig.4.3 shows the

VM utilization due to the requests the optimal broker sends to the public Cloud and not the total VM utilization since its VMs are potentially shared to meet the demand of other clients. Comparing both figures, we realize that the public Cloud starts to be selected when the number of requested VMs by both service classes increase. In such a case, the optimal broker routes the incoming service requests more often to the public Cloud to ensure the QoS provisioning. The setting $b_1 = 5$ VMs and $b_2 = 1$ VM unveils an interesting point. In such a case, the private Cloud rapidly becomes congested and the public one comes to be demanded by the optimal Broker. Because of that, there is a considerable drop in the VM resource utilization in the private Cloud compensated by a significant increase in the public Cloud.

4.3.3 Impact of Bandwidth Requirement of Service Class 1 on the Optimal Cost

The bandwidth requirement of service class 1 is increased and its impact on the optimal cost is determined. The results are captured in Fig. 4.5. As the number of requested VMs by both service classes increase, the service provision becomes more expensive and the end user has to pay more to use the Cloud. Additionally, as Fig.4.3 and Fig.4.4 uncover, the optimal broker selects the public cloud more often since it is unable to efficiently deal with the growing demand. Thus, the blocking cost increases. As a consequence, the optimal cost also goes up as shown in Fig.4.5.

4.4 Analysis of the Optimal Structure

This section shows how the proposed system allocates the resources from the available CSP (Cloud1 and Cloud2) under varying loads to achieve optimal policy. In this section, we derived a structure on selection of a cloud by which a user's service class #1 request will be accepted under different loads of Cloud2.

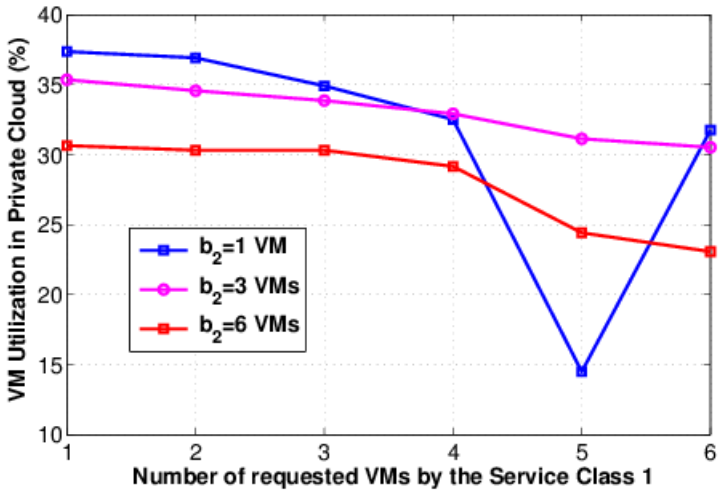

Figure 4.4: VM utilization in the Private Cloud versus the number of requested VM by the service class #1.

We have considered the following notations to analyze the optimal policy's structure for class 1 requests:

a) ▼ denotes class-1 requests accepted into Cloud1.

b) ■ denotes class-1 requests accepted into Cloud2.

c) ♦ means class-1 call which should be usually accepted into Cloud1, but are accepted by Cloud2.

d) ★ denotes blocking of a request.

Figures 4.6 and 4.7 depicts the results where bandwidth of class 1 is considered as 2Mbps and class 2 as 6Mbps. As seen in the figures, when the private cloud (cloud 2) load is between 0 and 5, a new class1 request is accepted into cloud2. And when the load of cloud 2 is between 6 and 10, the class1 request is accepted by public cloud (cloud1). This means that, as long as the resources are available and sufficient in the private cloud, the new requests are accepted by it. When the private cloud's load increases, the requests are then accepted into the public cloud. If there are insufficient number of resources in both clouds, then the requests are blocked.

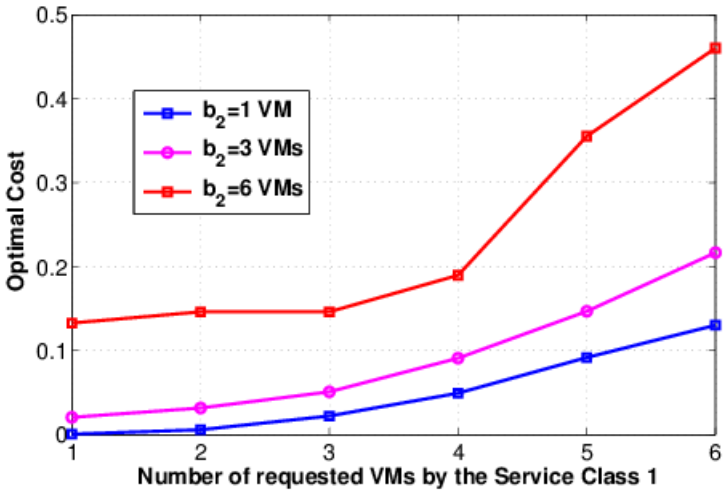

Figure 4.5: Optimal cost versus the number of requested VM by the service class #1.

Figures 4.8 and 4.9 depicts the results where bandwidth requirement of class 1 is considered as 4Mbps and class 2 as 6Mbps. Similar to the figures 4.6 and 4.7, in Fig. 4.8 and Fig. 4.9, it can be observed that the new requests are primarily accepted by the private cloud till its number of resources become insufficient. Then the new requests are accepted into the public cloud. The requests starts getting blocked when there are insufficient number of resources in both clouds.

Figures 4.10, 4.11, 4.12 and 4.13 presents the results where bandwidth requirement of class 1 is considered as 5Mbps and 6Mbps; and that of class 2 as 6Mbps which is constant for both class 1 bandwidth requirements. All these figures show that, as the bandwidth requirements increases for a incoming request, there will also be the increase in the number of allocated resources. This means, there will decrease in the number of available resources. Until there are sufficient resources in the private cloud, resources are accepted into it. When the private cloud starts getting congesting and to avoid 100% blocking of the future incoming requests, despite having few available resources, the present incoming requests are transferred to the public cloud. The requests are blocked when both private and public clouds have insufficient number of resources.

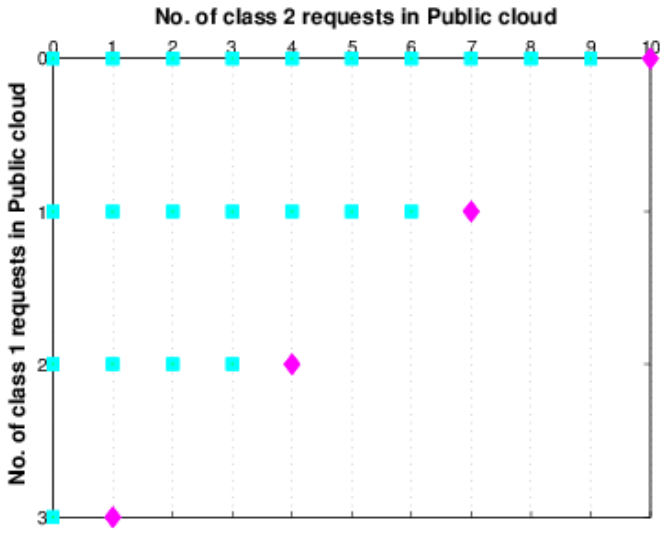

Figure 4.6: When Private cloud load is between 0 and 8
b1=2 and b2=6

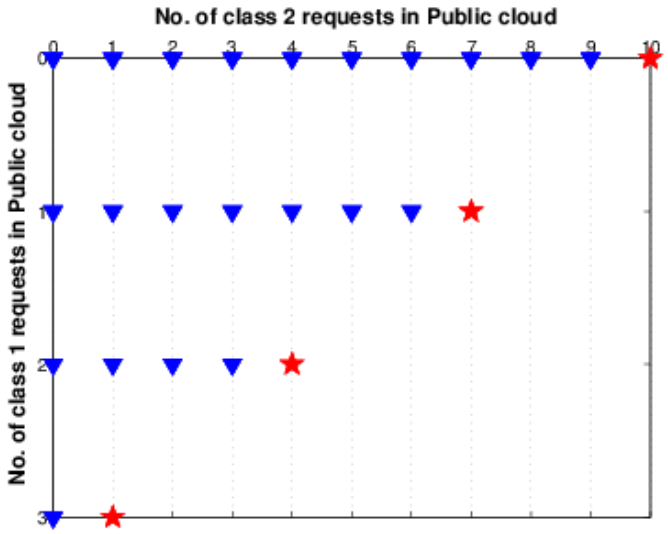

Figure 4.7: When Private cloud load is between 9 and 10
b1=2 and b2=6

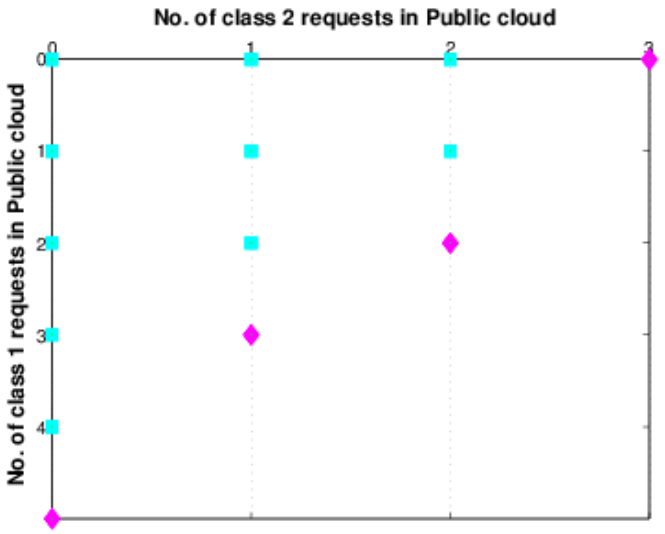

Figure 4.8: When Private cloud load is between 0 and 6
b1=4 and b2=6

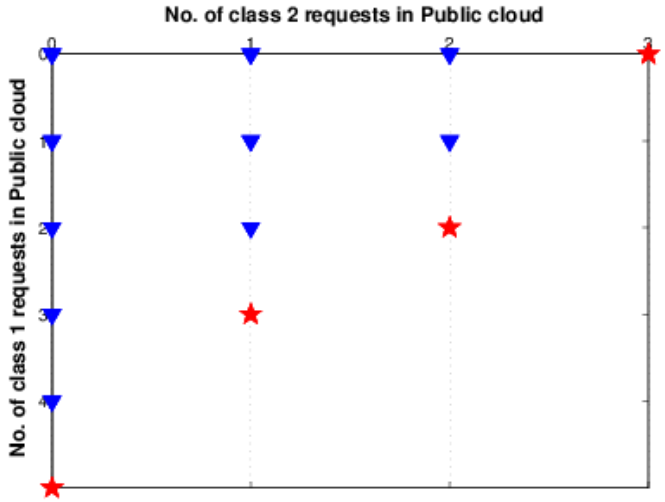

Figure 4.9: When Private cloud load is between 8 and 10
b1=4 and b2=6

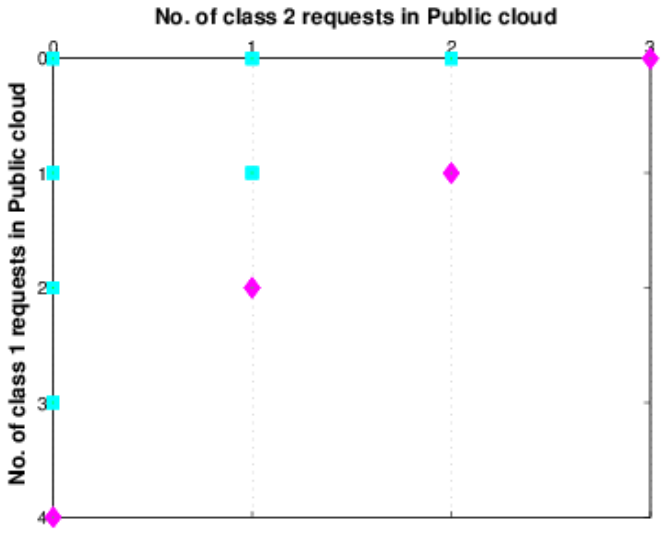

Figure 4.10: When Private cloud load is between 0 and 5
b1=5 and b2=6

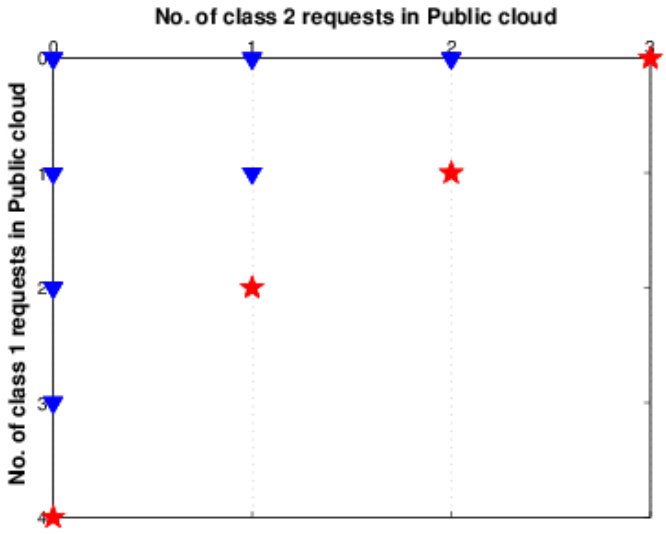

Figure 4.11: When Private cloud load is between 6 and 10
b1=5 and b2=6

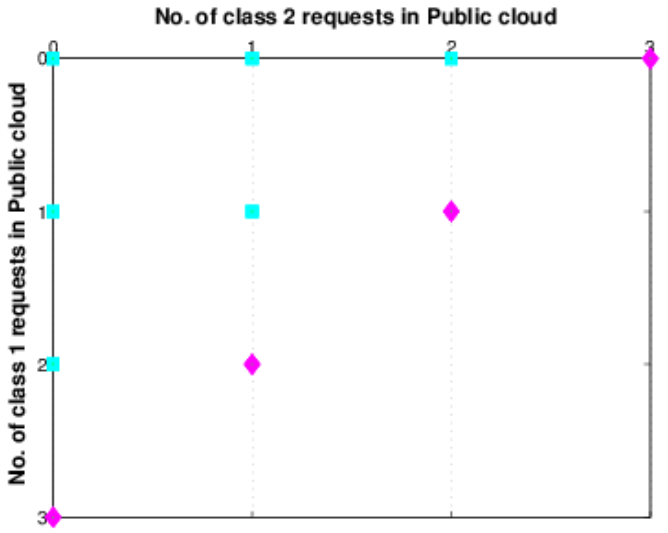

Figure 4.12: When Private cloud load is less than 6
b1=6 and b2=6

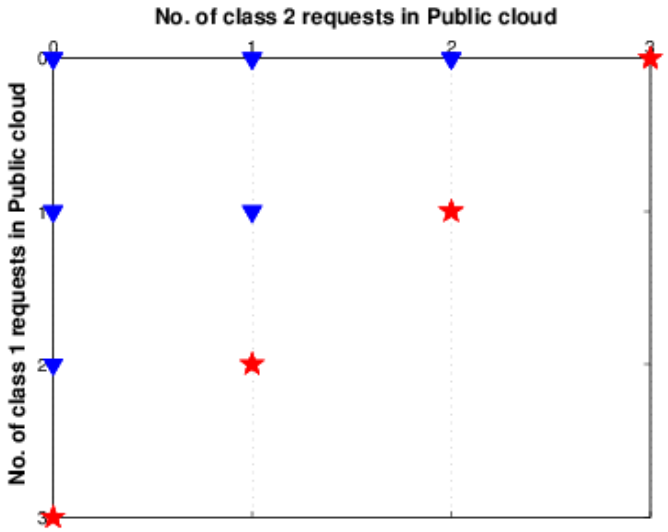

Figure 4.13: When Private cloud load is 6
b1=6 and b2=6

Chapter 5

Conclusion

In this thesis, we have presented an optimal Broker for Inter-Cloud ecosystem. The optimization problem was formulated according to the SMDP framework. The presented cost function allows the Cloud selection considering the service price and the blocking cost of each service class and Cloud center. This way, the designed Broker is encouraged to maximize the VM utilizations while supporting end users with less expensive services. Considering a scenario with a private and public Clouds, the optimal Broker admits incoming service requests in the private cloud whenever possible and resorts to the public cloud specially when the private one is unable to meet the peak demand.

In a Cloud market, eco-friendly operation is not only a necessity to maintain a long run sustainable development, but also a powerful advantage in the fierce Cloud market to attract new clients. For future works, we intend to design a Broker that takes energy-efficiency into account in the cloud selection problem.

In future, the proposed model can be used in various other scenarios. One of them can be E-Health Applications, in which a hospital can be a private cloud and any other third party can be a public cloud. The user in this scenario will be a patient. Every patient of a hospital will be a registered member in the private cloud of the hospital. For various limitations such as storage capacity required for the enormous data of patients, processing speed, etc.,the

hospital may tie up with a third party cloud, called as public cloud. When dealing with the management of resources such as important information on patients related to blood pressure, heart information, etc, these can be stored in the private cloud; and information not judged as important can be stored in the public cloud. In doing so, the proposed resource management scheme can be employed to systematically allocate the resources between the aforementioned CSPs in an efficient manner. In various other scenarios where the resources are required to be managed across various different heterogeneous entities, our proposed resource management scheme can also be applied.

Bibliography

[1] G. Motta, N. Sfondrini, and D. Sacco, "Cloud computing: An architectural and technological overview," in Service Sciences (IJCSS), 2012 International Joint Conference on, pp. 23-27, May 2012.

[2] J. Gibson, R. Rondeau, D. Eveleigh, and Q. Tan, "Benefits and challenges of three cloud computing service models," in Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on, pp. 198-205, Nov 2012.

[3] "Cloud Computing," in http://en.wikipedia.org/wiki/*Cloud_computing#Service_models*.

[4] M. Aazam and Eui-Nam Huh, "Media inter-cloud architecture and storage efficiency challenge," in Cloud and Autonomic Computing (ICCAC), 2014 International Conference on, 2014, pp. 206-211.

[5] Adel Nadjaran Toosi, Rodrigo N. Calheiros, and Rajkumar Buyya. 2014. Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey. ACM Comput. Surv. 47, 1, Article 7 (May 2014), 47 pages. DOI=10.1145/2593512 http://doi.acm.org/10.1145/2593512

[6] Y. Kessaci, N. Melab, and E.-G. Talbi, "A pareto-based genetic algorithm for optimized assignment of VM requests on a cloud brokering environment," in Evolutionary Computation (CEC), 2013 IEEE Congress on, pp. 2496-2503, June 2013.

[7] M. Nordin, A. Amin, and S. Shah, "Agent based resource broker for Medical Informatics Application in Clouds," in Computer Information Science (ICCIS), 2012 International Conference on, vol. 2, pp. 802-807, June 2012.

[8] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific, pp. 103-110, Dec 2009.

[9] F. Su, Z. C. Dong, and B. Chaolun, "An Application of Optimal SCGM(1,1)-Markov Model for Simulation and Prediction on Indexes of Water-saving," in Information and Computing (ICIC), 2010 Third International Conference on, vol. 3, pp. 82-84, June 2010.

[10] S. Frey, C. Luthje, C. Reich, and N. Clarke, "Cloud QoS Scaling by Fuzzy Logic," in Cloud Engineering (IC2E), 2014 IEEE International Conference on, pp. 343-348, March 2014.

[11] B. Nandi, A. Banerjee, S. Ghosh, and N. Banerjee, "Dynamic SLA based elastic cloud service management: A SaaS perspective," in Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, pp. 60-67, May 2013.

[12] Aoyama, Tomonori, and Hiroshi Sakai, "Inter-Cloud Computing," in Business & Information Systems Engineering 3.3 (2011): 173-177.

[13] Bernstein D, Ludvigson E, Sankar K, Diamond S, Morrow M, "Blueprint for the Intercloud protocols and formats for cloud computing interoperability," in International conference on internet and web applications and services (2009), pp 328336.

[14] N. Grozev, R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," in Software: Practice and Experience, vol. 44, no. 3, pp. 360-390, 2012.

[15] H.Liang, L.X.Cai, D.Huang, X.Shen, D.Peng, "An SMDP-Based Service Model for Interdomain Resource Allocation in Mobile Cloud Networks," in IEEE Transactions on Vehicular Technology, vol. 61, no.5, pp.2222-2232, 2012.

[16] M.Felemban,S. Basalamah, A. Ghafoor, "A distributed cloud architecture for mobile multimedia services," in IEEE Network, vol. 27, no. 5, pp.20-27, 2013.

[17] N. Ghosh, S.K.Ghosh, S.K. Das, "SelCSP: A Framework to Facilitate Selection of Cloud Service Providers," in IEEE Transactions on Cloud Computing, vol.3, no.1, pp.66-79, 2015.

[18] K. Hato, B. Hu, Y. Murata, and J. Murayama, "Designing inter-cloud system architecture," in Optical Internet (COIN), 2012 10th International Conference on, pp. 75-76, May 2012.

[19] Z. Mahmood, "Cloud Computing: Methods and Practical Approaches," Springer Publishing Company, Incorporated, 2013.

[20] G. Feng, S. Garg, R. Buyya, and W. Li, "Revenue Maximization Using Adaptive Resource Provisioning in Cloud Computing Environments," in Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on, pp. 192-200, Sept 2012.

[21] R. Patel and S. Patel, "Survey on Resource Allocation Strategies in Cloud Computing," in International Journal of Engineering Research and Technology, vol. 2, ESRSA Publications, 2013.

[22] V. Vinothina, R. Sridaran, and P. Ganapathi, "A Survey on Resource Allocation Strategies in Cloud Computing," International Journal of Advanced Computer Science & Applications, vol. 3, no. 6, 2012.

[23] D. Marinescu, "Cloud Computing: Theory and Practice," Elsevier Science, 2013.

[24] "11 cloud iaas providers compared." in http://www.techrepublic.com/blog/the-enterprise-cloud/11-cloud-iaas-providers-compared/ (Last visited Nov. 7, 2014).

[25] Q. Wu, Q. Zhu, X. Jian, and F. Ishikawa, "Broker-based SLA-Aware composite service provisioning," Journal of Systems and Software, vol. 96, no. 0, pp. 194-201, 2014.

[26] C. Papagianni, A. Leivadeas, S. Papavassiliou, V. Maglaris, C. Cervello-Pastor, and A. Monje, "On the optimal allocation of virtual resources in cloud computing networks," Computers, IEEE Transactions on, vol. 62, pp. 1060-1071, June 2013.

[27] M. Salama and A. Shawish, "A QoS-oriented inter-cloud federation framework," in Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual, pp. 642-643, July 2014.

[28] V. Nelson and V. Uma, "Semantic based resource provisioning and scheduling in inter-cloud environment," in Recent Trends In Information Technology (ICRTIT), 2012 International Conference on, pp. 250-254, April 2012.

[29] S. Sotiriadis, N. Bessis, and N. Antonpoulos, "Decentralized Meta-Brokers for Inter-Cloud: Modeling brokering coordinators for Interoperable Resource Management," in Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, pp. 2462-2468, May 2012.

[30] T. Choi, Y. Kim, and S. Yang, "Graph Clustering based provisioning algorithm for Optimal Inter-Cloud Service Brokering," in Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific, pp. 1-6, Sept 2013.

[31] G. Kecskemeti, M. Maurer, I. Brandic, A. Kertesz, Z. Nemeth, and S. Dustdar, "Facilitating Self-Adaptable Inter-Cloud Management," in Parallel, Distributed and Network-Based Processing (PDP), 2012 20th Euromicro International Conference on, pp. 575-582, Feb 2012.

[32] S. Sotiriadis, N. Bessis, P. Kuonen, and N. Antonopoulos, "The Inter-Cloud Meta-Scheduling (ICMS) framework," in Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, pp. 64-73, March 2013.

[33] Mechtri, M.; Zeghlache, D.; Zekri, E.; Marshall, I.J., "Inter-cloud Networking Gateway Architecture," Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on , vol.2, no., pp.188,194, 2-5 Dec. 2013

[34] Aazam, M.; Eui Nam Huh, "Inter-cloud Media Storage and Media Cloud Architecture for Inter-cloud Communication," Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on , vol., no., pp.982,985, June 27 2014-July 2 2014

[35] Sotiriadis, S.; Bessis, N.; Antonopoulos, N., "Towards Inter-cloud Schedulers: A Survey of Meta-scheduling Approaches," P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2011 International Conference on , vol., no., pp.59,66, 26-28 Oct. 2011

[36] H.C Tijms, "A first course in Stochastic models," John Wiley and Sons Ltd, 2003.

[37] G. Bolch, S.Greiner, H.de Meer, and K.S.Trivedi, "Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications," Wiley, 2006.