An Efficient and Secure Intercloud Framework

A Thesis submitted
for the award of degree of
DOCTOR OF PHILOSOPHY

By:
Lohit Kapoor
(951203010)

Under the Supervision of

Dr. Seema Bawa
Professor, CSED, Thapar University, Patiala
&
Dr. Ankur Gupta
Professor, CSE, MIET, Jammu

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA- 147004, INDIA

July 2016

To My Parents.....

Contents

List of Figures

List of Tables

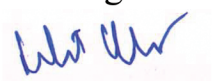
List of Abbreviations

FI	Federated Intercloud……..	2
CSP	Cloud Service Providers…….	3
SP	Service Providers……...	3
SITES	Service management in InTercloud for Efficiency and Security.....	12
SF	SITES Framework……..	42
SRS	Service Ranking and Selection……..	48
RM	Resource Manager.....	52
RRM	Remote Resource Manager.....	52
LG	Local Group……...	52
SG	Super Group……...	52
RA	Resource Availability…….	53
RR	Resource Request……...	53
RRR	Remote Resource Request…….	54
RSR	Request Service Rate.....	64
RT	Response Time……...	64
LRQ	Latency based Resource Query.....	66
CRQ	Cost based Resource Query.....	66
HRQ	Hybrid Resource Query.....	66
SC	Services Cloud.....	71
SISL	Single Instance Single Location.....	84

Certificate

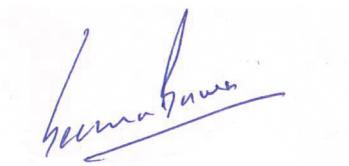
I hereby certify that the work which is being presented in this thesis entitled **"An Efficient and Secure Intercloud Framework"** in fulfilment of the requirement for the award of degree of **"Doctor of Philosophy"** submitted in **Computer Science and Engineering Department** of **Thapar University**, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Seema Bawa and Dr. Ankur Gupta, and refers other research works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

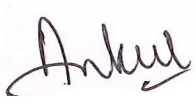


(Lohit Kapoor)
Regn. No. 951203010

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.



(Dr. Seema Bawa)
Professor, CSED,
Thapar University,
Patiala, 147004
Punjab, INDIA.



(Dr. Ankur Gupta)
Professor, CSE,
MIET, JAMMU,
J&K, INDIA

Acknowledgment

First of all, I express my gratitude to the Almighty, Who blessed me with the zeal and enthusiasm to complete this work successfully. The work presented in this thesis would not have been possible without my close association with many people. I take this opportunity to extend my sincere gratitude and appreciation to all those who made this Ph.D thesis possible.

First and foremost, I would like to extend my sincere gratitude to my research guide Dr. (Mrs.) Seema Bawa for introducing me to this exciting field of science and for her dedicated help, advice, inspiration, encouragement and continuous support, throughout my Ph.D. Her enthusiasm, integral view on research and her mission for providing high-quality work, has made a deep impression on me. I owe her lots of gratitude for having me shown this way of research. I am really glad to be associated with a person like her in my life.

My special words of thanks should also go to my research co-guide Dr. Ankur Gupta for his continuous support, guidance, cooperation, encouragement and for facilitating all the requirements, going out of his way. I have learnt extensively from him, including how to raise new possibilities, how to regard an old question from a new perspective, how to approach a problem by systematic thinking, data-driven decision making and exploiting serendipity. He has taught me another aspect of life, that, “hard work can never be defied and good human beings can never be denied”. His constant guidance, cooperation, motivation and support have always kept me going ahead. I owe a lot of gratitude to him for always being there for me and I feel privileged to be associated with a person like him during my life.

I am profoundly obliged to Dr. Maninder Singh, Associate Professor and Head, CSED, for his whole-hearted support and motivation. Also, I am thankful to my

Doctoral committee members for their constructive comments and regularly ensuring the progress of my research work.

This thesis would have been impossible without the support of my family. My deep regards to my father Mr. Kasturi Lal Kapoor and my mother Mrs. Chanchal Kapoor for their patience and love. Without them this work would never have come into existence. They have provided me with lessons on honesty and ethics and their humbleness and patience have always amazed me. I would also thank my sister Megha for her love, companionship, and support.

Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one.

July, 2016

Lohit

Abstract

The Intercloud represents the idea of a global ecosystem of collaborating CSPs offering potentially infinite compute resources enabling seamless resource provisioning and consumption. Various standardization bodies, working groups and research communities are working hard to provide a comprehensive framework but service deployment, orchestration, provisioning, Service Level Agreement (SLA) compliance and security present significant challenges in an intercloud environment. This research proposes an efficient and secure intercloud framework "Services management in InTercloud for Efficiency and Security (SITES)", for unified services management in the intercloud environment.

The SITES framework proposes a novel resource discovery mechanism, which provides direct CSP to CSP interaction, a contrast to traditional approaches where central entity is required for service scalability. A comparison of existing techniques with the proposed mechanism establishes the viability of the proposed mechanism.

SITES also proposes a services management mechanism which provides an on-demand ranking mechanism for service consumers. It allows greater optimization at an individual level rather than traditional one-size-fits-all approach. In this scenario users consume services by specifying cost and different Quality of Service (QoS) related constraints like latency, processing time, reliability, reputation and availability which are measured and tracked automatically within SITES. Individual users are therefore able to reduce their overall cost of service consumption compared to existing mechanisms while enjoying improved QoS-compliance. Real-world experimentation along with simulations show that SITES orchestrated services conceded significantly lower SLA violations even in the face of flash-crowd scenarios. This results in potentially increased profits for service providers over traditional mechanisms.

Finally SITES framework proposes a novel mechanism for detecting malicious applications based on application profiling and minimizing their impact using a containment-based security model. Encouraging results have been obtained by detecting malicious applications through comparison of resource consumption and performance time-series data pertaining to CPU usage, memory usage and inbound/outbound traffic of executing applications with known malicious applications.

The realization of the SITES framework for an intercloud environment results in enhanced capability to deliver services to end users in a customized manner while maximizing service provider revenue in a secure environment. This results in efficient management of cloud services, reducing cost for end users as well as service providers while increasing overall experience in a secured environment.

Chapter 1

Introduction

The Intercloud or the “cloud of clouds”, environment encompasses the deployment, configuration, provisioning, operation and portability of cloud resources across different Cloud Service Providers (CSPs). The intercloud provides a strong motivation for the deployment of planetary-scale services which need to cater to diverse geographical locations of their users besides optimizing latency and cost while ensuring quality-of-service and complying with service-level agreements. The intercloud has emerged as a logical evolution of the cloud computing paradigm allowing for the creation of a community of CSPs to offer greater value-add to the end consumer while facilitating enhanced elasticity, ensuring QoS even at peak loads, service/data portability and migration and collaborative services for mutual benefits. Compute resources/services provisioned and controlled by a single cloud CSP need to be orchestrated to meet demands for guaranteed end-to-end quality, compliance and other reliability issues. It is envisaged that if a cloud system experiences an unexpected overload or a natural disaster, spare resources shall be required to cope with the situation. In order to guarantee the required service quality, such as service availability and performance it is intuitive to consider a mechanism for flexibly reassigning resources from other CSPs under an overarching intercloud system. In particular, private clouds built by Small and Medium Enterprises (SMEs) are likely to collaborate with other clouds to effectively meet peak-load requirements, offer value-added services to their consumers or tap business opportunities in geographic areas where they do not

have a presence. The intercloud appears a promising business model in this context.

1.1 About Intercloud

According to Wikipedia [1]:

"The Intercloud scenario is based on the key concept that each single cloud does not have infinite physical resources or ubiquitous geographic footprint. If a cloud saturates the computational and storage resources of its infrastructure, or is requested to use resources in a geography where it has no footprint, it would still be able to satisfy such requests for service allocations sent from its clients".

If a cloud service provider gets resources shortage or the requirement for resources is demanded from a location where it has no footprints. It would still be able to satisfy such requests for service allocations sent from its clients. Research in the intercloud domain has picked up pace with large industry players having a sizeable cloud presence embracing the intercloud concept. "Cisco's vision for intercloud is a "cloud of clouds" that encompasses both Cisco data centers and those of its partners [2]. Cisco's vision is a perfect example of the “Federated Intercloud (FI)”, which is a close-knit eco-system, built around a large CSP, a set of SMEs and a developer community with the standards driven by the large CSP [3]. On the other hand is the concept of a non-federated intercloud or what we term the "Democratic Intercloud (DI)”, which represents a more open-market approach to CSP-interactions, built around a global standard. Both models shall be referred throughout the rest of the thesis and a clear distinction made wherever required.

While the intercloud is an evolutionary business model, it does have its unique technical challenges. When actually considering cross-cloud interactions, it is challenging to meet the demands from CSPs for guaranteed end-to-end service quality primarily during peak hours [4]. Existing large CSPs like Amazon, Google etc. are also reportedly facing the problem of predicting geographic

distribution of cloud users and providing QoS as per SLA's [5]. According to [6] an intercloud must ensure following functions:

a) Guaranteed end-to-end quality for each service
b) Guaranteed performance
c) Guaranteed availability
d) Convenience of service cooperation
e) Service continuity
f) Market transactions via brokers

These functions can be reliably achieved through adoption of common standards for cross-cloud communication, protocols for resource discovery and dynamic provisioning, resource/service orchestration, global trust and financial settlement mechanisms which are non-trivial to say the least. While technological solutions to the above issues do seem feasible in the near term, more complex business challenges need to be overcome before the intercloud can become a reality.

1.2 Intercloud Stakeholders

In this sub-section different stakeholders or key actors in the different intercloud paradigm are discussed:

1.2.1 Cloud Service Providers (CSPs)

A CSP provides the three cloud models Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) which is considered as unit in Intercloud, apart from implementing interfaces for flawless integration with the intercloud.

It is anticipated that the CSP will contribute a part of its resources to the intercloud. The participation of the amount of CSP resources can be fixed or dynamic in nature.

1.2.2 Service Providers (SPs)

A Service Provider is a entity which uses Intercloud infrastructure to inclement or deploy its own services primarily for users. It is a sort of third-party which uses different CSP's resources and offers its own value-added services. It can use single or multiple CSPs to deploy its service instances for potential consumers.

1.2.3 Service Consumers (SCs)

A user is the end-consumer of resources/services from the CSP/SP. Users can be potentially mobile and utilize services or access resources from geographically diverse locations, getting the same quality of service.

1.2.4 Brokers

The Broker is an intermediate that unite the intercloud entities like CSP's, SPs and users. As per Distributed Management Task Force (DMTF)'s Open Cloud Standards Incubator [10] a broker performs following functions:

a) Description of the cloud service in a template
b) Deployment of the cloud service into a cloud
c) Offering of the service to consumers
d) Consumer entrance into contracts for the offering
e) Provider operation and management of instances of the service
f) Removal of the service offering

1.3 Benefits and Challenges for Intercloud Stakeholders

Each stakeholder participated in intercloud to avail its benefits; however a number of challenges are also faced by them as discussed in Table 1.1 below:

Table 1.1: Benefits and Challenges for different stakeholder/actors in intercloud environment.		
	Benefits Foreseen	**Challenges**
For Users	Choice of CSPs	Highly abstracted, low control on data
	Access to diverse services, choice of Service Providers	Authenticity of service providers and quality of services
	Better service provisioning and response times based on geographical proximity	Transparency in accounting
For Cloud Service Providers	Access to additional resources on demand	Meeting intercloud resource commitments and performance guarantees along with those for traditional customers.
	Availability of datacenters in different geographical locations	Trust, migration, security, efficiency issues
For Broker	Boosting intercloud revenues	Geographical-aware auto-scaling of services
	Self-sustaining eco-system of producers and consumers	Interfacing with large number of different CSP brokers.

For Service Provider	Business benefits	SLA compliance across CSPs
	Greater access	Absence of governing body to handle legal cases

1.4 Classifications of Intercloud

We propose the definition of intercloud as,

"An ecosystem of CSPs offering standardized mechanisms for resource discovery and consumption involving resource, data and service migration in a secure and seamless manner across different CSPs based on well-defined economic principles"

A preliminary classification of the intercloud is presented in Figure 1.1.

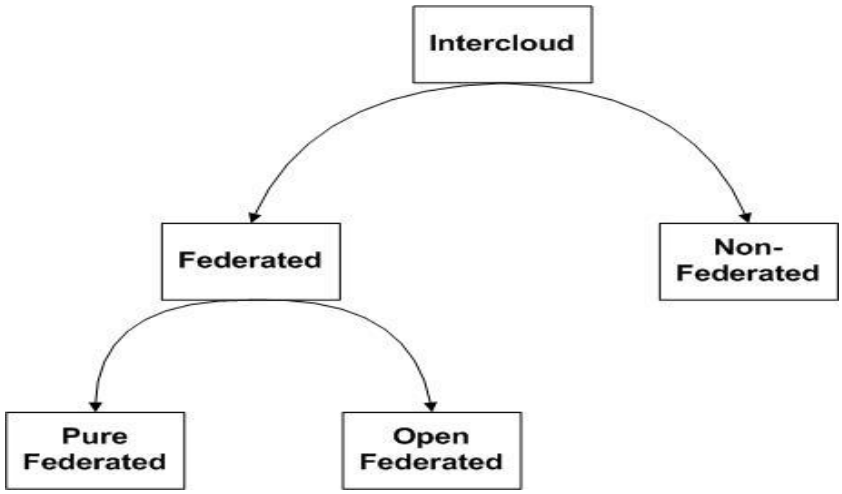


Figure 1.1 Intercloud Classifications

1.4.1 Federated Intercloud System

Authors in [11], [12] and [13] describe an intercloud to be in a federation when few CSPs interconnect their services or infrastructures with common objective and predefined/dynamic policies. In [14] authors define a federation as:

"An organizational structure where the parties concerned are autonomous but cooperate through agreement."

Being a part of a federation implies implementing common protocols and adopting a framework for:

a) Resource Discovery
b) Resource Requests
c) Negotiation
d) Resource Provisioning
e) Resource Utilization
f) Resource Release
g) Accounting and Settlement

We provide a more comprehensive definition of the federated intercloud as:

"An inter-connection of clouds following a set of common protocols, Application Programming Interfaces (APIs) and standards operating in a trusted environment with well established metering services"

Trust is implicit within a federation and a central entity within the federation is expected to provide authentication services to all participants. We provide a sub-classification of the federated intercloud as:

a) Pure Federated: In this model, a large CSP acts as a resource/service reservoir and smaller players typically consume resources/services from the larger CSP with very little likelihood of offering any resources in return, in that sense it is akin to a client-server model for the intercloud. Also, in such a system charges are typically pre-defined by the large CSP and in case of demand for resources, they are simply provisioned as per pre-defined

agreements. A common architecture for a pure federated model is a Hub and Spoke model as shown in Figure 1.2.

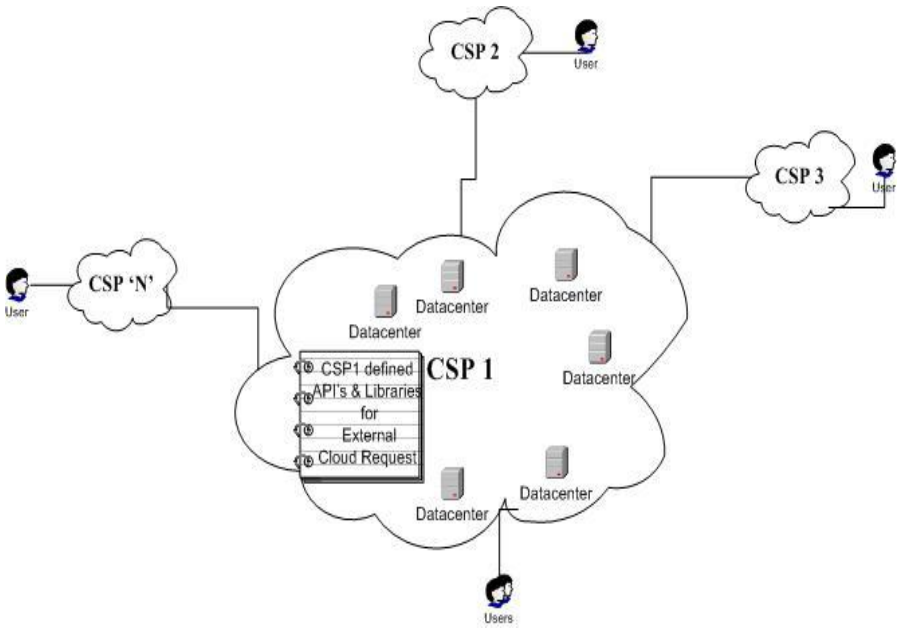


Figure 1.2 The Hub and Spoke organization of a pure federated intercloud

b) Open Federated: In this system, a resource discovery mechanism comes into play leading to determination of the best prospective CSP partner through negotiation. Here in the event of resource sharing, a new agreement is reached between participating CSPs. In that sense each CSP acts as both a resource provider and consumer. The model is depicted in Figure 1.3.

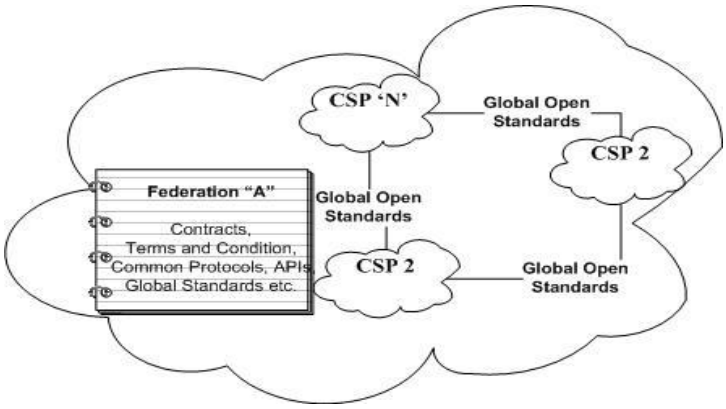


Figure 1.3 An example of an open federated intercloud

The central entity facilitating negotiation and resource provisioning is not shown for brevity. Critics of the federated intercloud still count data lock-in as an area of concern as the data of the user circulates within a certain set of CSPs and user has a limited choice to provision resources from outside the community.

1.4.2 Non-Federated Intercloud System

In a non-federated or democratic intercloud system all the CSPs interact with each other through Peer-to-Peer (P2P) model [4] or through a central hub/ exchange [5] (Figure 1.4). The major difference between democratic intercloud and federated cloud is that, in a federation the CSPs are fixed, while in the democratic intercloud CSPs are free to join and leave as per requirement.

A democratic intercloud is based on global open standards, setup. Moreover in democratic intercloud users have greater choice over service consumption and data migration due to performance, availability, cost issues etc. This migration can be done in two ways:

a) Transfer of data from one cloud to another in automated manner requires cross CSP Migration and due to performance issues, resource constraints at the CSP (without user-intervention).
b) Manual transfer initiated by the user due to deficient services.

We propose the definition of the democratic intercloud as:

"An intercloud in which users can dynamically choose and consume services/resources from any participating SP or CSP and CSPs can interact with other based on a global open standard supporting dynamic negotiation."

Intuitively, a democratic intercloud is more complex to envision with emphasis on trust management, authentication, non-repudiation, cross-cloud orchestration and metering services.

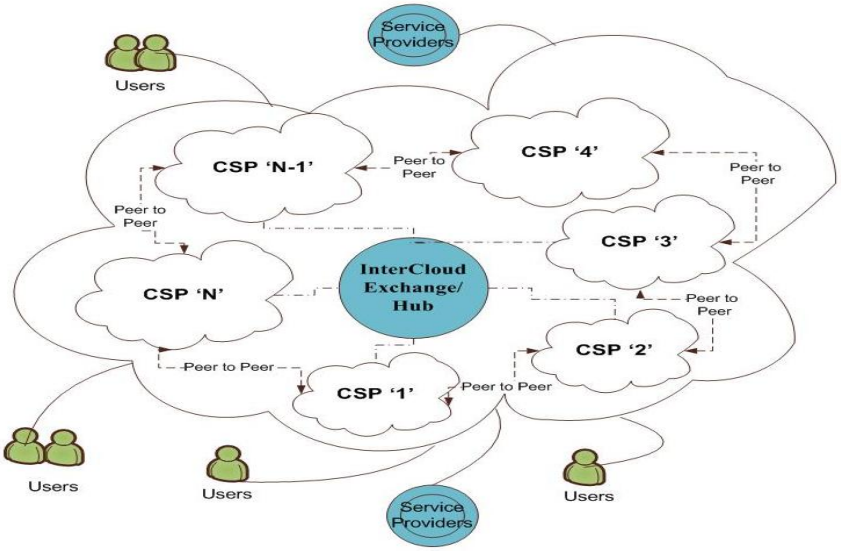


Figure 1.4 An Example of a Democratic intercloud based on Global Open Standards

While the democratic intercloud is in infancy stage, there are a broad spectrum of open source software like Eucalyptus [15], Openstack [16] etc. which provide the interfaces, protocols, programming models, and deployment options of the proprietary clouds. These might provide a viable approach to create a democratic intercloud in future.

1.5 Categories of Intercloud Operations

Various types of operations are performed by different stakeholders which can be further categorized based on above discussion:

1.5.1 Dynamic operations

In such kind of operations joining or leaving of new CSPs takes place, also referred as expansion or contraction process in Intercloud. The democratic intercloud supports both an expansion process to reflect new CSPs joining and the contraction process reflecting CSPs leaving [17]. Some researchers also refer to expansion and contraction in terms of the resources availability after each CSP to CSP transaction within a federation. These processes happen under

a set of procedures and standards agreed mutually by each of the participating CSP.

1.5.2 Resource/Service Integration Operations

Consider a case where CSPs or SPs can utilize services of other SPs or those deployed on other CSPs in order to offer value-added services to end users. Such kind of integration is termed as vertical integration. Example of vertical integration is present in [18] where authors propose architecture to provide a platform for sharing services for mutual benefits residing in different CSPs. On the other hand different CSP-level resource collaborations fall under the purview of horizontal integration. Tusa and Celesti et al. [19] [20] presented a heterogeneous Cloud federation model, which is known as "Horizontal Federation" for CLEVER [21] (a virtualized cloud environment). A Cross Cloud Federation Manager component (CCFM) is introduced that is integrated into every Cloud service provider while vertical service integrations are possible, they question its practicality. Hassan et al. [22] propose a game based distributed resource allocation scheme which works for horizontal federation.

1.5.3 Architectural Operations

Grozev et al. [7] provide the architectural classification for the intercloud, other classifications such as those in [8][9] classify the intercloud as

i) Centralized
ii) Peer-to-Peer
iii) Hybrid: In this type of system a central entity may perform the usual role while resource discovery/provisioning are done in a peer-to-peer manner.

Table 1.2, provides a summary of intercloud classification based on above discussed operations.

Table 1.2: Classification and Operations			
Type of Intercloud	**Dynamic Organization**	**Service Integration**	**Architecture**
Pure Federated	None	None	Centralized
Open Federated	Contraction	Vertical and Horizontal	Centralized
Democratic	Expansion and Contraction	Vertical and Horizontal	Centralized, Peer-to-Peer and Hybrid

1.6 Thesis Contribution

As per above discussion a number of forums, groups and researchers are working to develop an efficient and secure framework which provides seamless interaction between multiple Cloud Service Providers, beneficial for users, service providers. Without incorporating resource discovery and unified services management system in a secured manner, realization of such framework is not possible in real world scenario. Therefore, this thesis work is proposing a framework which incorporates resource discovery and services management in a secured manner without compromising the overall performance. It provides a common platform for various stakeholders like services consumers, services users and CSPs to manage the resources as per requirement.

Therefore the main contributions of the thesis are:

a) Facilitating direct CSP to CSP interaction without involving third party for resource discovery.

b) Efficient Resource Discovery for service scalability.
c) Seamless management of services in an intercloud environment.
d) A novel service ranking mechanism in an intercloud based on detailed performance monitoring and historical analysis.
e) A fine-grained control mechanism to the end user's for optimal service selection.
f) Á model of automated service scaling and deployment for service providers in an intercloud based on dynamic service consumption patterns.
g) Prevention of misuse of CSP infrastructure for launching DoS/DDoS attack.

1.7 Thesis Organization

Chapter 1: Introduction - This chapter presents an overview of the intercloud as an evolutionary distributed computing paradigm as a federation of individual Cloud Service Providers (CSPs). The key stakeholders in the intercloud system model and their respective roles, motivation for participation and expected benefits are discussed. Finally, a classification of the intercloud in terms of organization, functionality and nature of stakeholder interaction is provided. Based on these assessments an in-depth analysis of research gaps is discussed which forms the basis of problem formulation.

Chapter 2: Literature review - This Chapter provides a detailed discussion on the standardization efforts by industry consortia and protocols developed by various working groups, followed by related work in the areas of resource discovery, services management and security. Based on these assessments problem formulation and objectives of the thesis is discussed.

Chapter 3: Proposed Framework: Service management in InTercloud for Efficiency and Security (SITES) – This chapter begins from the introduction of services management and followed by framework requirements from the perspective of consumers, service providers and cloud service providers. It

represents proposed framework architecture which also includes its sequence of operation. The chapter highlighted the proposed mechanisms for resource discovery, services management and security to improve its efficiency.

Chapter 4: Proposed Resource Discovery Mechanism in SITES - This chapter presents a P2P-based resource discovery mechanism between individual CSPs in the SITES, which is the fundamental operation to support the services abstraction. This chapter describes its need and influence on SITES performance in terms of:

Detailed results and its analysis in terms of effectiveness and efficiency are presented. A comparison of existing techniques with the proposed mechanism establishes the viability of the proposed mechanism.

Chapter 5: Proposed Services Management Mechanism in SITES - This chapter introduces the problem of efficient service selection and consumption by geographically dispersed users in an intercloud scenario detailing the complexities involved. A novel fine-grained service selection mechanism is proposed which allows end user's complete control in service selection based on various quantitative and qualitative parameters.

Chapter 6: Proposed Security Mechanism in SITES - This chapter proposes a novel mechanism for effectively and efficiently preventing malicious applications hosted on SITES to execute. The potential of Intercloud as a "safe haven for DDoS attackers" is discussed in the beginning of the chapter. Further a real world experimental set up is presented which includes launching of DDoS attack by using virtual machines hired from top CSPs like Amazon EC2, Microsoft Azure and GoGrid. A recorded behavior in time series data base is presented. Further based on observed behaviour a scheme is discussed which identifies the malicious (DDoS) activities of application deployed in these VMs. This scheme is analyzed on the basis of experimental evaluation (effectiveness and efficiency).

Chapter 7: Conclusion and Future Scope - This chapter concludes the work done with a brief discussion on challenges faced for designing systems for establishing secure and efficient intercloud framework. It also presents conclusion of the thesis by highlighting the main contributions of this research work and specifically focuses light on the future work that can be undertaken to extend.

Chapter 2

Literature Review

This chapter explores standards and working groups, specialized services, business models and economic models followed by related work in the domain of resource discovery, services management and security. Therefore on the basis of this discussion, research gaps are evaluated and thereafter problem is formulated. After problem formulation objectives of the thesis is presented.

2.1 Standards, Services and Models in Intercloud

A number of standards/protocols have been offered by various working groups for interoperability in Intercloud. However their acceptability varies widely. This leads to different types of service offerings under different business/economic models.

2.1.1 Standards and Working Groups in Intercloud

There are various models in existence in the domain of Intercloud interoperability. These models are proposed by various industry consortia. Each of which representing its own framework to implement intercloud interoperability. However there is no uniform and standard model as suggested in Table 2.1. Further in Table 2.2, details of various interoperability standards, protocols and projects are presented followed by their status of development. The table also suggesting there is no universal acceptance of standards and

protocols as each of Working Group is coming with its own suggestion and implementation.

Table 2.1: Interoperability Models in Existence					
Model	**IaaS**	**PaaS**	**SaaS**	**Data**	**Networking**
Orchestration layer [40]	Yes	Yes	Yes	NA	NA
DMTF CIMI [41]	Yes	NA	NA	NA	NA
Adapters [40]	NA	NA	Yes	NA	NA
CMWG [41]	Yes	Yes	Yes	Yes	NA
Cisco Intercloud[38]	NA	NA	NA	NA	Yes

Table 2.2: Interoperability Working Groups		
Standard/Protocol/Project Name	**Aim**	**Defined/Developed**
Open Virtualization Format (OVF) [42]	VM migration	Standard developed
Cloud Data Management Interface (CDMI) [43]	To generate, access, update and modify data inside cloud	Standard defined
Open Cloud Computing	API for management	Protocol Defined

Interface (OCCI) [44]	tasks in cloud	
Topology and Orchestration Specification for Cloud Applications (TOSCA) [45].	Enables the interoperability with cloud service description. Creates the link between services instances and make the operational behavior.	Standard developed
Cloud Application Management for Platforms (CAMP) [46]	Definition of protocol for interoperability	Standard developed
Cloud Auditing Data Federation (CADF) [47]	Defines open standards for cloud auditing	Standard developed
LDAP [48], OpenID Connect [49]	Enable third party ID and Access Management functionality	Standard developed
US FIPS 140-2 [50]	Specifying the security requirements by a cryptographic module for sensitive information	Standard developed

2.1.2 Specialized Services

Recent Intercloud research and standardization efforts by several groups have led to the development of specialized services for facilitating intercloud functionality. This sub section details the work done on these specialized services for the intercloud.

i) Security-as-a-Service

Different types of security services are available such as:

a) Trust management (between various intercloud entities) - Trust-as-a-Service (TaaS) is an intermediate service which maintains trust ratings for different CSPs, SPs and individual services and even users. Most of the work in Trust Management in intercloud system is based on establishing the credibility of users based on feedback analysis [33]. This service plays a big role for selection of service or CSP based on its past behavior. Abawajy et al [34] present a reputation management framework which helps a service consumer to put a weightage for feedback a service provider. On the basis of this framework a technique for controlling falsified feedback ratings is generated.

b) Encryption services (secure communication) - Encryption-as-a-Service (EaaS), is subscription based model that permits customers to use encryption techniques without any need for installation of any kind of software [35]. Marc et al. [36] present the π-Cloud, a personal secure cloud that provides users with encryption services which can be used in an intercloud environment.

c) Identity and access management (one user with fixed privileges can access whole intercloud) - Consider a case, when users registered with a particular CSP want to use or access another CSP, for reasons such as performance, availability of a particular service, or purely as backup,

they would need to register or signup with the new CSP. A single sign-on system for the intercloud would solve this problem. Bernstein et al. in [30] present a trusted mediator model between elements in intercloud, providing identity and access management for users. Celesti et al. [37] present identity and access management in federated cloud to manage the authentication needed among clouds for federation establishment.

d) Secure workload migration (Porting workload/data from one cloud/service to other in a secure manner)-This service is useful in providing a secure connection between two different datacenters belonging to different CSPs. Cisco's Intercloud Fabric [38] builds hybrid clouds by increasing existing number of datacenters to public clouds while keeping network/security policies. This helps in creating highly secure connectivity across multiple clouds. They argue to provide secure workload migration by sustaining all network and security policies unambiguous to workload.

ii) Interoperability as a service

Due to potential large-scale heterogeneity in intercloud environment, interoperability is a major issue to enable seamless cross-cloud interactions. Interoperability would enable two heterogeneous cloud environments to collaborate by sharing compute resources [39]. The intercloud landscape inherits heterogeneous products and services ranging from IaaS, PaaS, SaaS and more. This heterogeneity of cloud services has led to increase the risk of vendor lock-in for customers. Therefore the aim of ideal cloud interoperability is very critical to the future success of the intercloud.

iii) Auditing as a service

Audit as a Service provides interface allowing various enterprises to automate the Audit, Assertion, Assessment, and Assurance of their

infrastructure. Cloud Auditing [54] presents the information about performance and security. Cloud Security Alliance (CSA) [55], The Open Group (TOG) [56] and the CloudAudit [57] is the cross-industry effort from the best minds and talent in Cloud, networking, security, audit, assurance and architecture backgrounds.

iv) Meta-scheduling as a service

Various classifications of schedulers are present which also includes in operating system, parallel and distributed computing etc. Among all these, meta-computing scheduling computing is considered to be most complex as there are a large numbers of local scheduler are present[58]. Meta-Scheduling is a strategy to manage and schedule user's requests in intercloud environment. It handles dynamic creation of VM, handles message exchanging, offers resources management while keeping SLA. The Inter-cloud Meta-scheduling (ICMS) Framework [59] uses meta-brokers that establish a central component for orchestration and to decide the best datacenter among collaborating clouds. The selection is based on performance criteria and the key requirements for a meta-scheduler are identified in [58] as:

a) The management of unpredictability (dynamics).
b) The heterogeneity of resources.
c) The geographically distribution of resources.
d) The variation of job requirements.
e) The compatibility on different SLAs.
f) The rescheduling support.

2.1.3 Business Models

The intercloud facilitates several business to business models encircling CSP to CSP interaction. In this interaction various different stakeholders comes with

different business expectations. Table 2.3 below presents a better understanding where each stake holder of intercloud comes with a specific need and each one should be dealt with a universally accepted intercloud framework.

Table 2.3: Stakeholder Expectations from Intercloud

Intercloud Stakeholder	Expectations
Users/Consumers	a) Cost optimization. b) Selection of Service. c) Consumption of Service. d) Availability of Service. e) Service and performance analytics. f) Migration of Service.
Service Provider (SP)	a) Maximizing profit. b) Availability of low cost resources. c) Scalability. d) Resource availability. e) Service visibility and access to geographically dispersed users. f) CSP performance analytics. g) Automated service management through service and user analytics.

Cloud Service Provider (CSP)	a) Availability of CSPs and unlimited access to resources when required. b) External resource-provisioning in low-cost. c) Control over resource provisioning in fine-grained manner as per demand scenario. d) Clear accounting and settlement. e) Performance guarantees for externally provisioned resources f) Maximize revenues through new business opportunities.
Intercloud Broker	a) Maximize revenues for CSPs and itself. b) CSPs should fulfill their contractual obligations. c) Large community of CSPs, SPs and users. d) Geographic spread.

On the basis of above discussion the following business models emerge:

i) Business to Business model (B2B): It is commerce between multiple CSPs to manage flash crowd scenario, exploiting geographical distribution of resources (datacenters) and to maintain service level agreements.

ii) Business to Consumer model (B2C or C2B): It refers to the commerce between broker/CSP, a user/SP and intercloud. It empowers SPs to:

a) Use resources from intercloud to provide better performance to the users.
b) Provide the option to deploy planetary scale services.

iii) Consumer to Consumer model (C2C): It implies interaction between users and SP. It facilitates users to:
a) Use multiple range of services for improved service selection
b) Cost vs Optimize performance
c) Access dynamic marketplace
d) Deployment of various services (Applications)

Authors in [24] propose a global service-oriented ecosystem based on the intercloud supporting large-scale, geographically-aware and dynamic service deployment, optimization and consumption.

2.1.4 Economic Models

Economics involves in intercloud when different stakeholders have different business interest. For example each CSP always wants to rent out all of its free resources in intercloud environment, therefore this cannot be achieved with following any of the following economic models [23] [25]:

i) Commodity Market: CSPs and SPs price their resources dynamically on the basis of demand-supply ratio.
ii) Posted Price Models: Also known as spot pricing performs advertising of special offers to attract customers.
iii) Bargaining model: To generate best deal.
iv) Contract-Net Model/Tendering: participants (CSP, SP, users) agree upon a contract.
v) Auction Model: Auctioning of resources across CSPs.
vi) Monopoly: when only one service provider/ cloud federation exist and price in non-negotiable.

vii) Bartering System – Resources are provided in exchange of resources and no financial transaction takes place

These economic models are equally applicable to the traditional cloud or in fact any distributed economic system. In that sense, the intercloud does not give rise to any new economic models.

The requirement in building a successful real world intercloud economic model includes:

i) A globally accepted naming service for different elements of intercloud [26] [27] and [28].
ii) A credible system which tracks SLA compliance/violations and maintains accounts [29].
iii) A globally recognized Trust Authority [30]
iv) A strong Auditing System [30].
v) Transparent and timely financial settlements [29].
vi) An arbitration mechanism [29] [31] and [32].

2.2 Resource Discovery in Intercloud

Existing techniques in resource discovery can be categorized into two approaches a) Centralized and b) Peer-to Peer approach.

2.2.1 Centralized Approach

Authors in [51] and [52] have presented a novel mechanism on service discovery based on negotiation using centralized exchange. However this central approach is prone to single point of failures. Net-Wide-Services (NWIRE) [53] is another resource discovery and scheduling technique which presents meta-computing scheduling architecture primarily depends on brokerage and trading. Global Inter-Cloud Technology Forum (GICTF) [54] which is a forum for intercloud principally focused on service discovery and is

based on collection of services in a centralized manner. In [55] authors present clustering of services for resource discovery based on past service performance. However, the clustering scheme offered is based on mutually transient services and undergo from overheads of creating and disbanding of the clusters. Moreover, monitoring past service experiences of each participant comes with overhead issues.

2.2.2 Peer-to-Peer Approach

A cross-grid cooperation architecture InterGrid [56] is composed of a set of InterGrid Gateways (IGGs) for managing peering provisioning between grids. A decentralized manner for efficient service discovery is used in the InterGrid Gateways for efficient service discovery. No fault-tolerance mechanism for the IGGs is available. Similar to InterGrid, authors in [57] proposed a completely decentralized peer-to-peer framework for dynamic service provisioning across cloud service providers. This scheme does not consider optimization which may come by considering geographical location of datacenters. Bessis et al. [58] also presented meta-scheduling model in intercloud environment to engage in drawbacks exist in centralized models. Along with it also undertake bottleneck in concurrent requests in intercloud environment during peak hours. Nelson et al [59] present an Inter-cloud Service Provisioning System (IRPS) in which each service and task represented semantically using service ontology. Further they use present a set of inference rules for discovery and semantic scheduler. Some instances of decentralized service discovery are available in grid computing.

2.3 Services management in Intercloud

Research in intercloud services management is in nascent stage. Existing literature in the field addresses some challenges of deploying and managing services in an intercloud environment, but the big picture seems missing.

Authors in [60] propose a broking mechanism in a multi cloud environment which looks at various aspects of pricing schemes, automatic decisions for service elasticity, optimization and finding the perfect cloud for new service deployment. However this work does not take into account latency between the service provider and the consumer which may result in sub-optimal service selection. The work presented in [61] figures out the latency benefits for optimal service deployment and minimizes the service response time to user. The experimental results in this work are based on only one real-world cloud. Hence, the comparison between implementation of service instances in different CSPs is not made.

TM forum (TMF) [62] presents a unified service delivery management model which focuses on deployment of federated services. It builds standards for services in intercloud environment known as Inter-cloud Services (ICS) which converges towards a singular approach to service orchestration of networks and datacenters of multiple providers. According to ICS, "*Service providers require an integrated, flexible, automated, service-focused intercloud management system*". As per Forrester Research's, Cisco commissioned research on Global Managed Services Opportunity (2009) the demand for managed "on-demand" services is identified as a key shift in user preferences [63]. Therefore the necessary intercloud infrastructure and middleware required to support service orchestration across CSPs needs to be in place to realize this market requirement.

Several brokers exist in literature which can operate across CSPs and focus on specific issues such as interoperability, performance monitoring of virtual machines, data migration and orchestration. RightScale [64] is a real world cloud broker around a middleware which is adaptable and automated. It analyzes past events for better cloud control, administration, and life-cycle management of applications across multiple clouds. It relies on monitoring the

performance parameters of instances of different clouds (in terms of virtual machine performance) but is service agnostic. Service performance can vary significantly in different production environments, but RightScale does not provide any mechanism to monitor the performance of different services across CSPs. Thus, optimal service selection remains a challenge. Zimory [65] cloud management platform enables cloud brokers to create a cloud eco-system including non-cloud providers to provide intercloud services. However, it does not provide any scheduling mechanisms or any decision making technique to manage services or aid the end user in selection of service instances which best meet their requirement. Aeolus [66] is an open source, Ruby-based cloud management software which allows users to choose between private, public or hybrid clouds, using DeltaCloud [67] cross-cloud abstraction library. But it is not aware of monitoring, scheduling and pricing schemes of different clouds, making it tough for the users to decide on the most economical service for their workload. Rackspace [68] cloud monitoring brokering service lets users monitor its websites whether located in Rackspace's own data centers or any other cloud and use graphs to analyze trends, outliners and patterns of their allotted servers but scheduling across CSPs is not handled. Cohesiveft [69] provides enterprise-grade virtualization and cloud migration services. Its main contribution is the transfer of applications comprising application template, operating system images, libraries and system components, across private, public or hybrid clouds in an automated manner but service-level monitoring and fine-grained control over service selection to the user is not considered.

Authors in [70] propose a "cloud coordinator" between multiple clouds which allows customers to dynamically scale their services for optimal performance. The introduction of middleware cloud coordinator allows a service to improve its performance, reliability and scalability. However this model does not

consider the dynamic change in market price of resources. Also, the view is service-centric and not user-centric.

2.3.1 Scheduling Techniques

Authors in [71] present a routing technique for managing service collaboration among different cloud providers. It works for the stability and efficiency of overall routing processes. This protocol deals with the changing configuration and traffic overheads in real cloud but fails to take decision based on internal performance of the service. Authors in [72] present a ranking mechanism of different services in order to provide a comparative view to users to chose a particular service over another under different use-cases. In this work authors proposes SMICloud Broker which performs service discovery and ranking on different Key Performance Indicators (KPIs). However this mechanism does not provide any autonomic technique to redirect the users request in the case of flash-crowd scenario, fault incidence etc. to ensure minimal SLA violations and revenue loss to the service provider.

Thus, most of the current cloud brokering mechanisms provide scheduling or routing of service requests based on monitoring the virtual machines on which the services are deployed. In most of the cases customers can select, monitor and migrate these virtual instances across CSPs without having a comprehensive service-view of the intercloud. The service-view of the intercloud is required to a) optimize service deployment and management for the service provider (auto-scaling, replication, migration) b) optimize service selection for the end-user (cost, response-time, QoS-compliance).

2.4 Security in Intercloud

Different cloud security working groups like [165] [166] clearly suggests a number of security issues in cloud computing. However these issues pose a high

degree of risks when we talk about Intercloud environment. If we take an example of a DoS/DDoS attack, which is an attempt to make the services assigned to the authorized users unavailable. The occurrence of a DoS/DDoS attack typically increases bandwidth consumption, causing congestion, making the service inaccessible to the users or floods the service with packets to overwhelm it. DDoS attackers may launch different types of attacks (TCP, UDP, http etc) from outside source to a cloud destination (Malicious Outsiders) or from inside as a normal user of Intercloud infrastructure to outside destination (Malicious Insider).

2.4.1 Malicious Outsiders

For malicious outsiders a number of devise mechanisms are used which collocated at the server hosting the service to quickly detect attacks and then attempt to recover from them. For large applications which are hosted in the cloud, the CSPs can provide several layers of security and high-availability features including firewalls, traffic analysis, redundant-hosting, dynamic service-replication and vm migration. However, such attack detection strategies and recovery can be expensive, both in terms of time and resources, while the impact on service performance cannot be entirely mitigated. Intrusion Detection System (IDS) is a famous defense method against these types of attacks [73]. In this method, network and/or system activities are monitored for malicious activity. Its main function is to identify malicious activity keep log information about this activity, block/stop it, and report it. Three major detection methods are used by any IDS: signature-based, statistical anomaly-based, and stateful protocol analysis. Authors in [74] use IDS to secure services in cloud from DDoS attack. SNORT technique is proposed where network traffic of inbound and outbound are audited. All the packets are scrutinized in real-time to identify a particular type of attack based on predefined rules by the

intrusion detection system. A similar approach is used in [75] in which each virtual machine is loaded by SNORT for sniffing all traffic. This success of the scheme has been shown in a Eucalyptus [76] cloud. All IDS-based schemes are designed to work at the recipient end which is the target of the attack. Further, attacks launched by well known DDoS toolkits like High Orbit Ion Cannon (HOIC) [77], a DDoS http-based attack toolkit, uses what it calls "Booster Scripts" to modify the packet headers and introduce variations in the attacks. By examining the valid header ordering possible abnormalities can be detected. However, inspecting network traffic designated for a hosted application (either at source or recipient) constitutes a violation of privacy in case of normal user traffic and would be unethical on part of the CSP. Hence, a privacy-preserving mechanism is clearly required.

In case of a federated cloud model, a defence federation has been proposed in [78] against DoS/DDoS attacks. A separate IDS is loaded on each cloud and on the basis of information exchange these IDS work. If any specific cloud is under attack an alert is generated for the whole system. Further trustworthiness of CSP is taken by voting without hampering overall system performance.

Some CSPs offer DDoS mitigation services [79], [80] to ensure the uptime of the service for customer in the event of DDoS attacks. Prolexic [81] claims to have an effective DDoS mitigation strategy with four DDoS traffic scrubbing centers. All in-bound traffic is routed to the nearest scrubbing center. It also relies on filtering techniques and anti-DoS hardware which closes the source of all the botnet activities. However, routing traffic for scrubbing introduces communication delays for normal traffic possibly impacting response time, SLAs and even user satisfaction.

Authors in [82] use Services Oriented Architecture (SOA) to ascertain the true identity of source of DDoS attack and present a model to filter these packets.

However it doesn't explain how to detect source if attacker is using cloud infrastructure for attack.

Thus, most of the work done on detecting DDoS attacks is based on:

a) Network traffic analysis at the recipient end i.e. at the server hosting the application under attack.
b) IDS Snort rule and attack signatures
c) Use of firewalls and puzzle servers.

All the above mechanisms are reactive in nature and entail significant processing costs and communication delays without any performance guarantees for applications under attack. Existing work also does not adequately address preventing the misuse of cloud computing infrastructure for launching DoS/DDoS attacks.

2.4.2 Malicious Insiders

Identifying and containing malicious insider without violating privacy is always a challenging job in Intercloud. Consider a case where an attacker with valid credit card registers and immediately launches DDoS attack. The traditional techniques like IDS seem to be a failure to cope up the issue effectively. Only few techniques have been used by most of the CSPs like enforcing strict registration and verification processes. Other CSPs are monitoring and inspecting customers network traffic which clearly violation clouds' privacy preserving conditions [165]. This area is very much open and big research effort is required in this part of Intercloud domain.

2.5 Problem Formulation and Objectives

Based on literature survey the problem focussing on resource discovery, service management and security explained as follows:

a) Discovering the required resources to avail the facility of scaling of services in an intercloud environment plays a critical role in order to implement a well coordinated federation of CSPs to avoid user request drop and delayed request response. Moreover, resource information in a federated environment should be up to date and each CSP in the federation should be aware of the resource status of the other CSPs. Small or medium-sized Cloud Service Providers (CSPs) are usually limited in terms of serving capability due to limited compute services in their data centers. The problem gets exacerbated during peak hours when the demand is very high increasing the probability of non-servicing of user service request. Due to the nature of cloud computing cloud vendors need to dynamically provision resources from other vendors to create the illusion of "on-demand elasticity". An intercloud architecture connecting different cloud-service providers therefore becomes unavoidable in this context.
b) A number of intercloud brokers have emerged as intermediaries between service providers, users and the CSPs. However, current cloud brokers do not provide advanced service management capabilities across CSPs to allow end users to select the service instances that best meet their requirements. To enable such a selection mechanism, detailed service monitoring on various performance parameters over sustained periods shall be necessitated. Traditional performance management techniques mainly focus on vm-level monitoring within a CSP, but in the intercloud scenario we need to have a complete performance view of services across CSPs. Clearly a more comprehensive service-level monitoring mechanism is required to take informed decisions on service selection which meet user-defined criteria.
c) Recently there has been a spate of security attacks using hired cloud computing infrastructure especially Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks.

DoS/DDoS attacks are some of the most widely prevalent attacks on the internet which target well known web-sites/applications/services by overwhelming them with malicious requests affecting their performance or causing them to crash. With intercloud offering huge amount of computing resources on a pay-per-use basis, it has become easier for malicious users to hire cloud resources and launch DoS/DDoS attacks from multiple locations. This makes it difficult to identify the source of origin of these attacks and contain their damage. Moreover, some malicious users have used cloud resources to host Attack-as-a-Service which allows third-party users to launch DDoS attacks by just specifying the intended targets. Thus, there is a need to detect malicious applications hosted on the cloud and prevent them from utilizing the vast computing infrastructure of the cloud to launch planetary-scale attacks which can potentially cripple the internet including critical business and government resources.

On the basis of above discussion following are the objectives of the thesis:

i) To study and analyze existing intercloud systems on various functional and performance parameters like security and resource discovery.
ii) To propose an intercloud framework to address resource discovery and security aspects efficiently.
iii) To design and implement the proposed framework.
iv) To validate the proposed framework.

Chapter 3

Proposed Framework: Services management in InTercloud for Efficiency and Security (SITES)

This chapter starts with the description of service management in intercloud paradigm, which further extended to present the basic requirements to develop an intercloud framework for efficiency and security. On the basis of these discussions framework architecture for the proposed framework is presented followed by its sequence of operations.

3.1 Services management

The intercloud has emerged as a logical evolution of the cloud computing paradigm allowing for the creation of a community of CSPs to offer greater value-add to the end consumer while facilitating enhanced elasticity, ensuring QoS even at peak loads, service/data portability and migration and collaborative services for mutual benefits. For small and medium CSPs this model is fairly intuitive from a resource sharing perspective. For Large CSPs the intercloud offers them the possibility of offering services in the geographical proximity of their customers and improving responsiveness. The intercloud also alleviates the issue of data/vendor lock-in. In any case end-users want flexibility to shift from one cloud to another or from one service to another, due to various reasons like performance degradation, high cost, legal issues etc. Service portability or redeployment can solve these issues [122] [123], but a large numbers of CSPs makes it difficult for service providers and end users to decide which CSP will be a best-fit for their requirements. Services management in an intercloud

environment encompasses the deployment, configuration, provisioning, operation and portability of cloud resources across different Cloud Service Providers (CSPs). The intercloud provides a strong motivation for the deployment of large-scale services which need to cater to diverse geographical locations of their users besides optimizing latency and cost while ensuring quality-of-service and complying with service-level agreements. Thus, a mechanism which allows user's customized and seamless access to services in an intercloud scenario is required. A major concern for any inter-cloud services management framework is the orchestration of services across CSPs and allowing end-users fine-grained control over how they select and consume services without knowing the deployment and location details. Cloud brokers have emerged as intermediaries between all stakeholders like service providers, users and the CSPs. However, current cloud brokers do not provide advanced service management capabilities across CSPs to allow end users to select the service instances that best meet their requirements [124]. To enable such a selection mechanism, detailed service monitoring on various performance parameters over sustained periods shall be necessitated. Traditional performance management techniques mainly focus on vm-level monitoring within a CSP, but in the intercloud scenario we need to have a complete performance view of services across CSPs. Clearly a more comprehensive service-level monitoring mechanism is required to take informed decisions on service selection which meet user-defined criteria. Consider a Service Consumer (end user) who requires access to a data-intensive service with constraints over latency and/or data transfer costs. Therefore it is logical to choose the CSP whose data-center (where the service resides) is located closest to the consumer of that service. Similarly, for compute-intensive services a user would like to choose a service instance which offers the lowest cost while delivering acceptable performance. Thus, the nature of services and the service

deployment models play an important part in service selection. However, the service selection process in an intercloud environment can be expected to throw up some counter-intuitive results. This is because network and service performance can vary significantly over time-zones and periods of peak-usage [124] [125]. There is a need to focus on how a service is performing over a period of time and at specific times. Moreover a performance comparison between different service instances would also provide greater insights for service selection facilitating delivery of the best possible cost to performance proposition to the end user. At the other end of the spectrum are the Service Providers who need insights into the consumption pattern of their services to facilitate dynamic deployment and scaling to maximize revenues while meeting customer requirements. Facilitating optimal service provisioning and consumption is therefore non-trivial and a major challenge in intercloud environments.

3.2 Framework Requirements

To better understand the requirements to develop an efficient and secure service management framework in Intercloud, the perspective of all its stakeholders (Service Consumer (SC), Service Providers (SP) and Cloud Service Providers (CSPs)) need to be considered. We identify the perspectives of these stakeholders while developing the framework.

3.2.1 Service Consumer Perspective

Service consumer requires a flexible environment for its service deployment. Following are the points from service consumers point of view.

i. Service Selection

Consider the case where a user wants to execute a job. In the intercloud a large number of similar services which meet the user requirements may be available.

Different services deployed across different CSPs will have different performance levels and might entail different costs. Therefore, to select the service which best meets the stated requirements of the end user is not trivial. The proposed framework should facilitate optimal service selection for the end-user.

ii. Changing Service Provider

Consider a case where the Service Consumer is not satisfied with the services offered by the current Service Provider and wants to shift to other Service Provider. The proposed framework should facilitate seamless service consumption across Service Providers without data lock-in.

iii. Fine-grained Control

An end-user might want greater control in how it consumes the services. For instance the end-user might impose cost constraints, time constraints, latency constraints, geographical constraints or other requirements on service characteristics which the proposed framework should be in a position to satisfy. The proposed framework should also provide detailed service-related information to allow the end-user complete control over the mechanism of service selection and consumption.

iv. Seamless Interaction

An end-user should not be concerned about the location of the CSP which hosts the service being consumed nor should it be aware of how its service requests are routed and serviced across the intercloud. The proposed framework should therefore abstract the underlying details of the CSPs and service deployment from the end-user, which should focus on just selecting and consuming services through a standardized and seamless interface.

3.2.2 Service Provider Perspective

Service Providers expectation from proposed framework is different from service consumer as given below:

i. QoS Compliance

The intercloud is a highly dynamic environment which needs to cater to flash-crowd scenarios and volatile resource requirements. From a Service Provider perspective it is imperative that the proposed framework ensures that deployed service instances are continuously monitored and dynamic load-balancing, fault-tolerance and elasticity be provided to ensure that the Service Provider agreed QoS thresholds with the end-user are not violated.

ii. Dynamic Service Deployment

Responsiveness of a service is an important performance parameter for any SP. Thus, a SP needs to be aware of the geographical distribution of its end-users and dynamically deploy service instances across the intercloud to improve service responsiveness. Proposed framework should facilitate such dynamic deployment in response to the geographical location of the end-users.

iii. Maximize SP's Return on Investment (RoI)

A Service Provider needs specific insights into how its service is being consumed (average and peak load, user location, average cost) and its performance (response time, reliability, SLA violations etc.) besides the various service hosting costs across the intercloud to maximize its RoI. The proposed framework should allow a Service Provider enough control to allow it to optimize its service deployment strategy.

iv. Geographical Aware Auto-Scaling

Geographical Location is very important when dealing with varied users requests. Further Service Provider have the option to deploy services such that they are near to user's location. A Service Provider may need to dynamically scale-up or scale-down service instances in different geographic regions

depending upon the number of users' requests form a particular region. Therefore the proposed framework should provide this facility.

3.2.3 Cloud Service Providers Perspective

Participation of CSPs in Intercloud depends mainly on following points:

i. Resource Discovery

Participation of CSP in any intercloud for resource sharing is very important when dealing with flash crowd scenario. Therefore proposed framework should facilitate the discovering of right resources in limited time so that high productivity can be achieve.

ii. Security

The main essence of any intercloud paradigm is to offer seamless resources for service deployment. However these services could be malicious in nature and may launch high volume attacks with such high resource availability. Therefore proposed framework should detect and contain these types of attack within a limited timeframe considering privacy of the service.

3.3 Framework Architecture

This work proposes a comprehensive "Services management in InTercloud for Efficiency and Security (SITES)", a distributed hybrid framework which encompasses resource discovery for services elasticity, service performance management, a service ranking and selection mechanism for end users, SPs and CSPs in a secure environment.

Consider SITES framework (SF) consists of a federation of N CSPs given by

$$SF = \{CSP_1, CSP_2, \ldots\ldots, CSP_N\}$$

Each CSP consists of multiple data-centers located in M different geographical locations across globe i.e.

$$CSP = \{DC_1, DC_2, ..., DC_M\}$$

Let S be the set of 't' total services offered by an Ic, such that ,

$$S = \{S_1, S_2, \ldots, S_t\}$$

Let 'x' be the total number of services components for each service 'S' deployed in CSP_N, such that

$$S_t = \{S_{c1}, S_{c2}, \ldots, S_x\}$$

A service 'S' can be deployed in multiple CSPs with individual components distributed in a way such that

$$\{S_t \mid (S_{c1} \varepsilon\ CSP_1), (S_{c2} \varepsilon\ CSP_2), \ldots, (S_{c(x-1)} \varepsilon\ CSP_{N-1}), (S_{cx} \varepsilon\ CSP_N)\},$$

or single monolithic service is deployed/replicated at multiple CSPs, such that 'y' multiple service instances exist at several CSPs, such that

$$\{S_t \mid S_{t1} \varepsilon\ CSP_1, S_{t2} \varepsilon\ CSP_2, \ldots, S_{ty} \varepsilon\ C_N\},$$

or a single service component is replicated at multiple CSPs

$$\{S_t \mid S_{c1} \varepsilon\ CSP_1, S_{c1} \varepsilon\ CSP_2, \ldots, S_{c1} \varepsilon\ CSP_N\},$$

and finally where a monolithic service or all components of a service are deployed at the same CSP and maybe even at the same data center within the CSP. The schematic of proposed SITES is shown in Figure 3.1, which consists of following three proposed mechanisms:

i. Resource Discovery:

Proposed resource discovery mechanism in SITES interconnects datacenters belonging to different CSPs in P2P manner as contrast to existing techniques discussed in [9] most of which depends heavily depends on a central entity. Since the resource information in a federated environment should be up to date and each CSP in the federation should be aware of the status of the other CSPs. Therefore, due to the geographical distribution of the data centers belonging to different CSPs communication latency can become a major performance bottleneck. Traditional architecture like centralized brokers and schedulers, it is not always possible to place them in close proximity to all data centers. Thus, some CSPs end up paying a higher communication cost than others each time

resources from other CSPs are requisitioned. Thus, proposed SITES attempts to minimize the communication latency by taking into account the geographical location of the data centers. It present a P2P-based distributed resource discovery mechanism based on spatial-awareness of cloud data-centers belonging to different Cloud Service Providers. The scheme is based upon exploiting location information of Data Centers and organizing them into DHT peers for optimal communication. It consists of two levels of groups local and global, inter-connected to each other. A JXTA [127] based implementation is done to provide some inherent benefits such minimized response time which is suited to such kind of environments. It thus allows for QoS-compliant resource/service provisioning across Cloud Service Providers (CSPs).

ii. Services Management:

Proposed services management mechanism in SITES provides a step towards a Unified Services Management Framework for an intercloud environment which facilitates a) optimal service deployment and geographically-aware auto-scaling and b) optimal service selection and consumption by the end-user in a seamless manner. It encompasses brokering services, service performance management and a service ranking and selection mechanism for end-users. The SITES receives user's request, provides a customized ranking of various available services (across CSPs) and allocates the service based on user selection. Further, in the case the user finds the services not meeting their expectations, it can select a different service instance on the intercloud. Moreover, a service provider can also initiate a migration of a service instance to some other CSP to better meet its SLAs with the end users. The aim of the proposed framework is to enable the users to consume customized services as per their choice in the intercloud ecosystem and present them a variety of services options which best meet their requirements. We evaluated the effectiveness of the proposed framework using a custom simulator based on real world service performance

measurements across different CSPs. The results indicate that from a service provider perspective achieving high performance at reasonable cost is dependent on the service deployment scheme. It was further observed that mere load-based auto-scaling is not effective to deal with flash crowd scenario but instead geographically-aware auto-scaling to exploit latency benefits in an intercloud environment is more efficient for global services. Further, a customized ranking mechanism for service consumers allows greater optimization at an individual level rather than with a one-size-fits-all approach.

iii. Security:

Proposed security mechanism in SITES manages to detect and contain the malicious activity of any deployed services inside any CSP which is a novel contribution compared to existing intercloud framework [38]. Attackers now a day are hiring resources to launch planetary scale DDoS attacks. The seemingly infinite compute resources on offer make cloud computing an attractive option for launching planetary-scale attacks. Cloud Service Providers (CSP) which rent out computing resources, need to ensure that their platforms are not used by malicious users/services in launching attacks.

SITES offer a novel security mechanism for detection and containment of malicious applications based on application profiling. Further, a global blacklist of malicious applications and their performance profiles is maintained and continuously updated to collaboratively aid in quick detection across CSPs. This privacy-preserving scheme effectively neutralizes malicious applications preventing them from misusing the large computational resources on offer.

The proposed framework architecture consists of a number of sequences of operations as described below.

3.3.1 Registration

Service Providers (SPs), CSPs and Service Consumers register their credentials through Registration/ Authentication module. SPs can offer their services with SITES through a well-defined interface. A service registration request contains complete service description including generic information, service components, related files and dependencies which facilitate automated service deployment. Upon successful registration services are reflected in the central service directory maintained by the SITES.

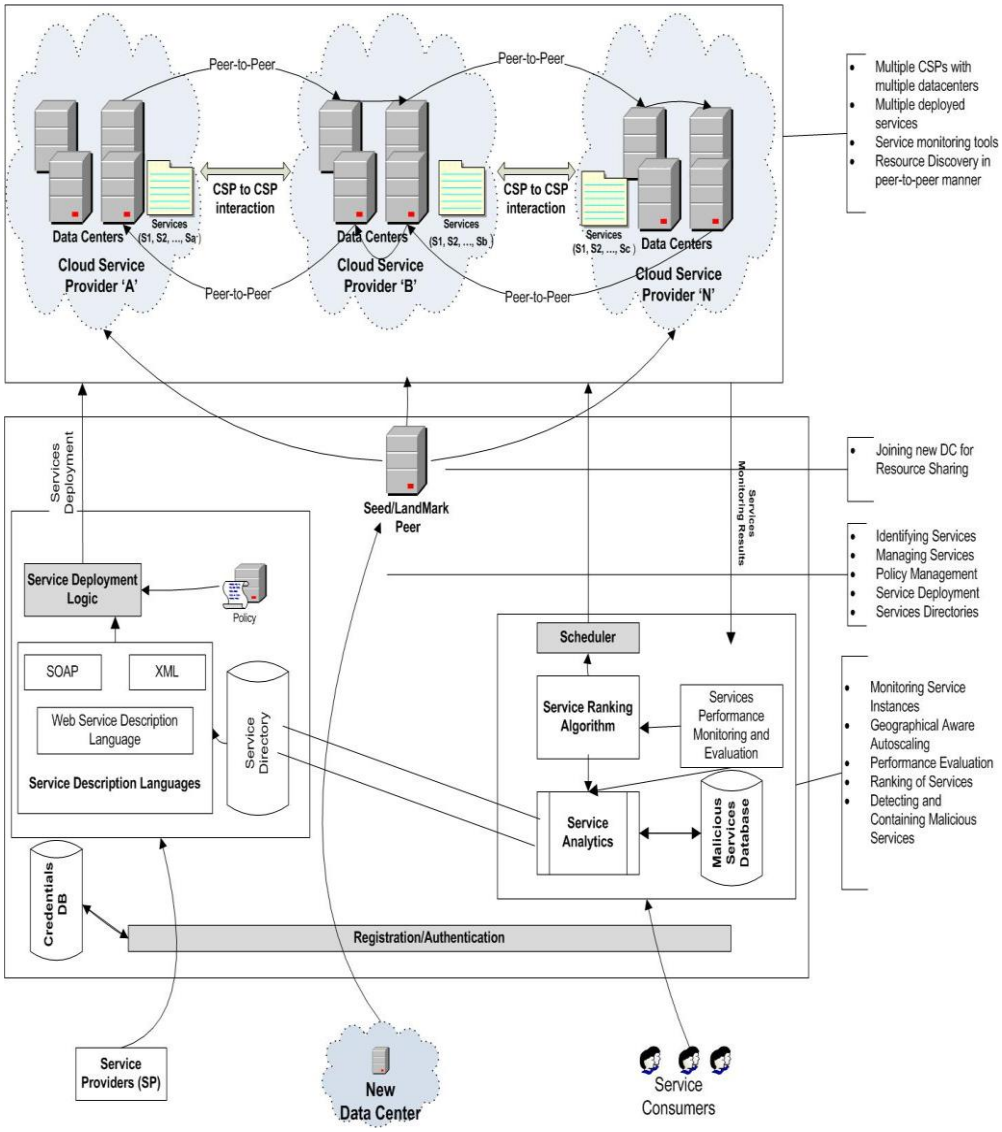


Figure 3.1: A Schematic View of SITES

Further, it also provides a seed/landmark peer for a new data center of existing/ new CSP willing to join the environment. This newly joined DC is put under a local group which enables it for contribution of its resources.

3.3.2 Service Deployment

The process of service deployment is based on the resource requirements and constraints of a particular service. For instance a specific platform may be a dependency for service deployment. Also, the physical resources (number and desired configuration of virtual machines), their location and cost ($ per hour per virtual machine) can be important constraints for the SP. In that case the SC deploys the service at the CSP which best meets the requirements of the SP. The SP can also enable the dynamic provisioning functionality in the SITES, which allows for auto-scaling (service duplication or provision of additional physical resources) to meet volatile service requests.

3.3.3 Service Monitoring and Profiling

Once a service is deployed by the SITES, the Service Monitor (a Real-time monitoring Tool) monitors each service. It obtains the service handle from the service directory and keeps sending heartbeat messages to the deployed service instance after a pre-defined interval for checking uptime, availability and response times. It also obtains detailed performance information from the local service monitor deployed at the CSPs servers which monitor the resource usage of each deployed service instance. Service monitor also entails keeping track of individual service queues maintained by the SITES. This gives indication of the service load and allows the correlation between load and service performance to be established. Thus, trend analysis and performance profiling of each service instance is performed. Thus, a strong basis for service ranking and selection and further optimization is created by the service monitor. These monitored results

also involves in creating detailed profiles of installed applications/services at the level of individual VMs and matching them against the profiles of known malicious applications. Unknown applications are classified based on observed deviations from normal behavior and inputs from expert system which captures classification of logic of expert human administrations. Further, the approach is privacy-preserving and proactive enabling faster detection of malicious applications.

3.3.4 Service Ranking and Selection

Since detailed performance profiles of each service instance are created by the service performance monitoring module it is straightforward for the SITES to rank comparable services (belonging to the same category). The service performance monitoring module is also responsible for providing feedbacks to Service Analytics for evaluating different services. This is done through a weighted formula based on different parameters such as latency, reliability, availability and user rating while remaining within the overall cost constraints specified by the consumer. Hence Service Ranking and Selection (SRS) algorithm ranks the services in ascending order keeping the parameters into consideration. Although the SITES maintains these internal rankings based on the weighted formula, the end-users are free to specify their own weighted formula allowing customized rankings to be computed on demand for individual end-users. This is a major contribution of the proposed framework. Subsequently, the user selects the appropriate service and negotiates the SLAs with the selected service. The details of SRS are presented in Chapter 5.

3.3.5 Service Consumption and Management

All the preceding operations contribute to major real-time optimizations during service consumption phase. If the SITES detects that the service request queue

has reached its threshold (can be determined by observing past correlation between service queue length and response time) then the SITES can perform auto-scaling including duplication of service instance. Similarly if service requests from a particular geographic region are pervasive and the SLA agreement contains a latency threshold then an additional service instance can be deployed at a CSP in that geographical region. Similarly, if the resource costs at the CSP hosting the deployed service goes beyond the cost constraint specified by the SP, the SITES can migrate the service to a CSP which offers resource cost within the specified cost constraint of the SP. Similar optimizations are possible at the user level as well allowing users to switch between service instances or even competing service providers depending upon which better meets their requirements. Thus, SITES allows multiple optimizations to be performed dynamically both in the context of the SP and the end-user giving all stakeholders flexibility in meeting their objectives.

3.3.6 Service Scalability

Service scalability involves the process of discovering resources, therefore SITES facilitates the process of resource discovery for a CSP in a P2P manner. This interaction eliminates the need of any third party or intermediate hence optimizing the resource discovery process.

Chapter 4

Proposed Resource Discovery Mechanism in SITES

Resource discovery for scaling of services is a major challenge in the successful implementation of intercloud environment. Efficient resource discovery is very important to avoid user request drop and delayed request response. For this resource information of every CSP in the intercloud should be available to other participants. Resource discovery can be done in centralized or decentralized manner; however majority of the research has focused on centralized mechanisms where each CSP interact with meta-broker and submits all its requirements. Meta-broker on the behalf of the requesting CSP searches and provision required resources. It coordinates with CSPs local brokers as local resource availability changes dynamically, thus making meta-broker to pre-assume the previous known state of the local services as the scenario becomes highly dynamic. In intercloud paradigm different CSPs consist of a number of datacenters located in different geographical locations. Therefore to deal with latency issue, placement of meta-broker is very important, but in centralized model it is not possible to place meta-broker in close proximity to all data centers.

Thus some CSPs have to suffer from higher communication cost. Along with this centralized approach comes with a number of other shortcomings like real time resource availability information, single point of failure, lack of trust between CSP and third party meta-broker, performance vs. scalability, security etc. In this chapter SITES presents a peer-to-peer based resource discovery mechanism which reduces the communication latency by taking into

consideration the geographical location of the data centers. It makes certain that communication latency is reduced and requests are serviced by most appropriate datacenters in terms of proximity and cost.

4.1 Resource Modelling for Proposed Mechanism

Each CSP participates in the federation and in any CSP multiple data-centers are located in different geographic locations. Each data-center of a CSP consists of Resource Manager (RM) which manages its resources and a Remote Resource Manager (RRM) which keeps resource information of other neighboring data centers. In this scheme the RRMs fit in a particular geographical location are arranged into the different Local Groups (LG). One RRM in each group assumes responsibility for acquiring all the required resource information from other peer RRMs located in the respective LG through resource availability advertisements. A virtual network overlay of all such RRMs is created to facilitate exchange of resource information. This virtual network overlay is called the Super Group (SG). Figure 4.1 provides a schematic of the proposed scheme in which different datacenter belonging to different CSPs forms different local groups with a chosen RRM from each LG participating in the global Super Group.

Let RRM_i (i=1, 2, 3…, M) represents 'M' Remote Resource Managers (RRMs) and each RRM_i fits in a LG comprising M data centers, which is represented by DC_{i1}, DC_{i2}.....DC_{ir}…. DC_{iM}. Theoretically M is variable which can increase or decrease with time as datacenters may join and leave the P2P network, but in this case we assume that all the datacenters go on with federation. Thus,

$$\| LG_i \| = \| RRM_{ir} \| = \| DC_{ir} \|$$

where $i>0$, $i = 1, 2…., M$ and $i \leq r \leq M$

Resource Availability (RA) status is generated by each datacenter periodically in the form of advertisements. The RA is articulated in terms of Resources (RES) and their associated cost (C), where each resource can be a virtual machine, platform or service. Thus, RA is the set of resource, cost tuples advertised by each RRM within the LG and cached by the super RRM which participates in the SG.

$$RA = \{(RES_{1,} C_1), (RES_2, C_2).\ldots\ldots(RES_N, C_N)\}$$

Consider 'X' is the number of resources offered by a particular data center at a particular time; hence X varies with time based on resource demand. Thus, other RRMs require caching merely the last RA's issued by RRMs of other data centers as it accurately signifies the position of available services. Moreover, the cost related with the resources is also a component of the RA. Other data centers which are desirous of availing services within the federation put out a Resource Request (RR) advertisement which is again expressed in terms of required RES and desired cost.

$$RR = \{(RES_{1,} C_1), (RES_2, C_2).\ldots\ldots(RES_K, C_K)\}$$

where K is the number of resources requisite by the requesting RRM.

The objective for the requesting RRM is to locate another RRM such that

$$RR_K \approx RA_X$$

Where K<=X, so that the number of resources available at the prospective partner RRM is more than or equal to the number of resources requested. The RR from a particular RRM is first attempted to be serviced within the LG. Each RRM already has the cached RA advertisements from other RRMs within the LG. If the resource availability within the LG is not met, the requesting RRM sends a "Remote Resource Request" (RRR) to SG. If the resources requested in the RRR are available at a particular RRM, the RRM sends the details of the RMs to the requesting RRM. If none of the RRMs within an LG meet the requested services, the RRR is propagated further within the SG until desired

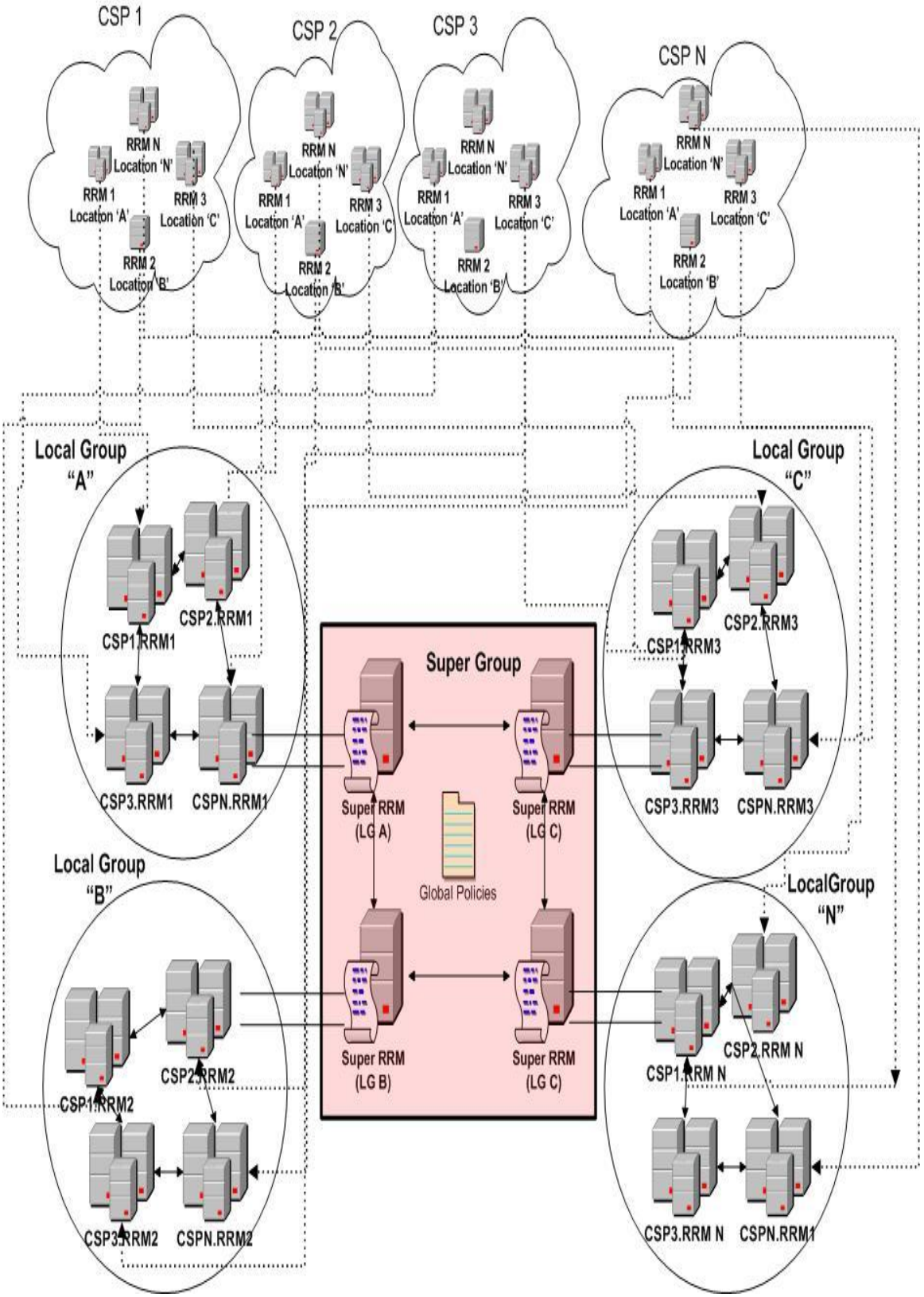


Figure 4.1 Schematic view of proposed resource discovery mechanism in SITES

request is met or all options are exhausted. Managing trade-off between cost and latency is a challenge in all resource discovery strategies in any intercloud paradigm. A case may arise where the cheapest resources might be available in the outermost datacenter due to which latency adds its own costs in terms of data transfer costs and communication overheads. This choice needs to be made by the requesting RRM.

4.2 Algorithms for Proposed Mechanism

Positioning of datacenter in an intercloud paradigm is one of the most important issues for efficient resource discovery. In the proposed mechanism placement of newly joined datacenter is based on the geographical locations. For example datacenter belonging to different CSPs but in same location are grouped together to form a Local Group (LG), thereafter, one datacenter holds the position of Super RRM. These Super RRMs forms a Super Group (SG) so that a communication can be established between various LGs.

4.2.1 Algorithm for Joining Process

If the RRM is first in the network, it assumes the role of the Super RRM. At initial stage each RRM uses seed peers or landmark peers which are provided by SITES to assist in the peer join process. For subsequent joins of RRM's the request is responded by the super RRM. With the increase in the size of LG, requests get cached on all intermediary RRMs that they pass through. Thus RRM join times are subsequently lowered. The newly entered RRM is now capable to receive RA and sent RR advertisement from/to other RRMs. Algorithm 1 shows RRMs joining the LG.

Algorithm 1: RRMs joining the LG

```
1:    For each RRM ε Peer Cloud do
2:       if (RRM does not ε to LG_i)
3:    locateSuperRRM(myRRMID, regionID)//find SuperRRM for
      my region
4:          if (!superRRM)   // no SuperRRM found
5:    newSuperRRMID = becomeSuperRRM(myRRMID)
6:    LG = createLG(myRegionID)
7:    joinSG(newSuperRRMID, regionID)
8:    else  // SuperRRM exists in my region
9:    registerWithSuperRRM(myRRMID)
10:   joinLG (LG)
11:   Exit
```

4.2.1.1 Super RRM Selection

For the selection of Super RRM, first come first serve technique is used. The first RRM to connect a LG recommend itself as Super RRM for a particular region. Succeeding RRMs keep hold of their joining rank in the LG.

The Super RRM performs as a Gateway to the SG by gathering resource advertisements from other RRMs within the LG and giving out it within the group of super RRMs. In the doubtful event that the Super RRM be unsuccessful, the next ranking RRM behaves as the Super RRM.

This procedure is started if the Super RRM does not send out a special status message during a designated time period.

Each RRM constantly produces an RA (resource availability) status message 5 minutes which holds the present position of resources and their related cost and circulates it inside the LG. Each RA message has a time-to-live parameter connected with it to make sure that older messages do not stay in distribution.

4.2.2 Proposed Algorithm for Resource Discovery Process

The process of resource discovery is covered by two types of constraints a) cost or b) resource specification which are part of the resource request advertisements.

Algorithm 2 represents the method of resource look up and its provisioning thereafter. Following are the steps involved:

i. Selection of resources by any RRM can be performed on the basis of "latency" (proximity) or "cost" or both.
ii. In advertisement both resource vector and constraint vector is passed as resource requirement.
iii. The obtained results are processed on the basis of responses and a RRM is selected after ranking based on lowest response time. A confirmation is send in the case of final decision.
iv. Specific requests which are not serviceable within the LG due to lack of resources or not meeting cost constraints are then put out in the SG for possible resource provisioning.
v. The Super RRM propagates the request to other Super RRMs in the SG which propagate the requests further within their respective LGs.
vi. RRMs which fulfil the resource criteria specified in the advertisement contact the advertising RRM directly.

A sample RRM resource request advertisement is shown in Figure 4.2 where the desired resource description is present.

Each RRM holds its own identification number (ID) with clear indication for the requirement (virtual machine (vm) or service).

For virtual machine request, parameters are required like number of virtual machines, virtual machine configuration etc. Vm configuration includes the CPU type (number of cores), storage capacity (GB), memory (GB) etc.

This desired configuration comes along with different constraint which includes time for the resource requirement, maximum affordable latency etc.

In the case of service request advertisement, service ID along with number of instances required is mentioned. It also includes the desired operating system or the platform.

Algorithm 2: Resource Lookup and Provisioning at each RRM

```
1:    advertiseResources(resourceVector, costVector) // RA
2:    advertiseRequirements(resourceVector, constraintsVector) //RR
3:    processResponse (responseVector)
4:    for each response in responseVector
5:    rankResponse(response)
6:    selectedRRM = getTopRRM()
7:    sendConfirmation(selectedRRM)
8:    processRequest (request)
9:    If (evaluateRequest(request))
10:   sendConfirmation(request.getRRM())
11:   Exit
```

```
<RRM:Resource Request Advertisement>

<Resource Descp=”Resource Description for Individual RRM”>

<RRM_ID type=”UUID” Descp =”RRM’s ID”/>
< Resource type =”String” Descp =Virtual Machine/Service/>
<Resource_quantity = “Uint32” Descp  = “Number of VMs”>
<Bandwidth_Type =”String” Descp =”Minimum bandwidth required”/>
<Platform type=”String” Descp =”Specific operating system/platform required”/>

<VM Config>
< CPU type =”Uint32” Descp =”Number of cores”/>
<Storage type =”Uint32” Descp = “Hard disk space “/>
<Memory type = “Uint32” Descp = “Minimum RAM”>
</VM Config>

<Constraints>
<Cost type = “double” Descp = “cost constraint for resource/hour”/>
<Cost Weight type = “double” Descp = “weight ”/>
<Latency type=”double” Descp = “desired latency”/>
<Latency Weight type=”double” Descp = “desired weight”/>
</Constraints>

<Service Descp = “Service Description for individual RRM”>
<Service_name=”String” Descp = “Service ID”>
<Service_instances=”Uint32”  Descp  = “Number of instances required”>
<Platform type=”String” Descp =”Specific operating system/platform required”/>
</Service Description >

</Resource Description>

</RRM:Resource Request Advertisement>
```

Figure 4.2: Sample RRM Resource Request Advertisement

4.3 Implementation

In this section the evaluation of proposed resource discovery mechanism is done. For this a real world experimental setup is established to conduct various experiments and analyzing their results.

4.3.1 Experimental Setup

To evaluate the effectiveness of mechanism, thirty physical machines each with configuration shown in Table 4.1 are deployed. Devstack [126] is used to create a local cloud which provides an option to install and run Openstack (software to control the cloud) on local systems. It enables user to create, control and destroy virtual machines. A number of one hundred and fifty virtual machines with configuration as shown in Table 4.2 are created.

Table 4.1: Physical Machine configuration

OS	CPU	HDD	Memory (MB)
Ubuntu 14.04 (Trusty)	Intel Core i7-2600 @3.4 GHz	500 GB SATA	4GB

Table 4.2: Virtual Machine configuration

OS	CPU	Cores	Memory(MB)
Windows Server 2012	Intel Xeon E52670	1	1024

For peer to peer deployment, we also implemented the Juxtapose (JXTA) [127] java based protocol for creation and maintenance of our P2P network. JXTA utilizes the Distributed Hash Table (DHT) for organizing the P2P overlay as a hierarchical topology.

However, it relies on rendezvous peers to maintain and distribute routing indices for normal peers and the resources/services that they provide. Queries are forwarded to rendezvous peers to locate the actual peer on which the desired resource/service resides.

The reason for using JXTA is:

a) Supports Interoperability required in intercloud
b) Platform and Language independence for heterogeneous environment in intercloud
c) Ubiquity (any virtual machine can be a peer)
d) Open standards (XML) for advertisement and communications

Each VM constitutes a JXTA peer which depicts a RRM corresponding to each data center. Therefore a P2P network of participating RRMs is created. LG and SG construction is done by using these results obtained from real world latency data mentioned in [128] by AT&T.

For LG construction local latency were used while in the case of SG, inter-continental network latency measurements were used to model communication delays within the SG.

Cloudsim 3.1 [129] is used to generate the workload in the form of cloudlets for each VM. These cloudlets are then converted in the form of resource queries for each RRM under following parameters (Table 4.3).

Table 4.3: Cloudlets/Queries parameters	
Parameters Name	**Ranges**
cloudletLength (the length or size (in mips) of this cloudlet to be executed per VM)	(1000 to 5000 mips)
pesNumber (CPU cores per VM)	1
Frequency of Resource Request (The number of requests per unit time)	3~6 per minute
Resource Usage Duration (Time to hold a resources)	35~65 minutes
Frequency for Flash-crowd (Peak hour time)	once every 2 hours
Duration of Flash-crowd scenario (Time duration of peak hour)	15 minutes
Frequency of Flash-crowd resource request (The number of requests per unit time)	15~20 per minute
resCost (Cost requested per resource)	0.20$ - 0.40$ per hour

4.3.2 Results Analysis

After deploying the required VMs on a physical infrastructure, a number of experiments were performed.

4.3.2.1 Evaluation of Startup Time

In the first experiment we measured the startup time for ten to fifty participating RRMs with one designated Super RRM in a Local Group. The aim of the experiment is to observe the cumulative time for the initial configuration and organization of a Local Group. It is clear from Figure 4.3 that as the number of participating RRMs increases the overall startup time per RRM reduces from 9.3 seconds/RRM (for ten RRMs) to 8.2 seconds/RRM (for fifty RRMs).This is due to the impact of super RRM startup time and resource aggregation on the overall time gets averaged out. The startup time includes the JXTA initialization time per peer/RRM as well.

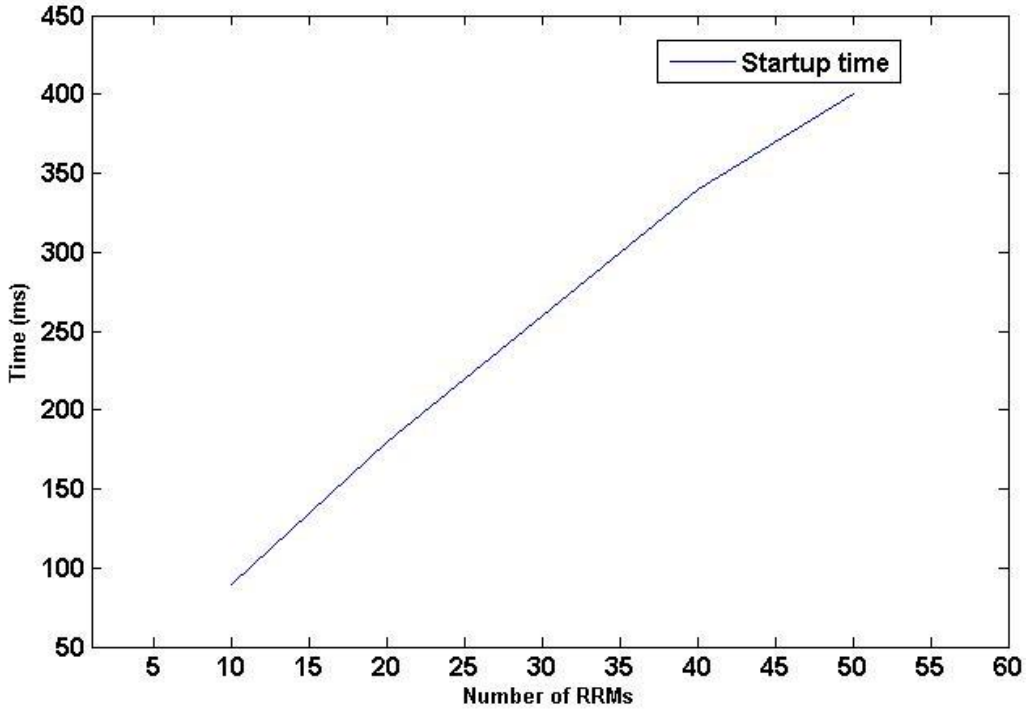


Figure 4.3: Startup time with varying number of RRMs

4.3.2.2 Average Joining Time

A variety of timing measurements for two different types of operations and resource discovery queries within the test setup were obtained for varying topology sizes.

Figure 4.4 provides the time taken for a new RRM to join the existing setup. Average time ranges from 770 to 860 ms for topologies with 10 to 50 RRMs within a LG.

The join process for a new RRM comprises initialization time plus JXTA peer join time plus the time taken for the RRM to connect with the Super RRM.

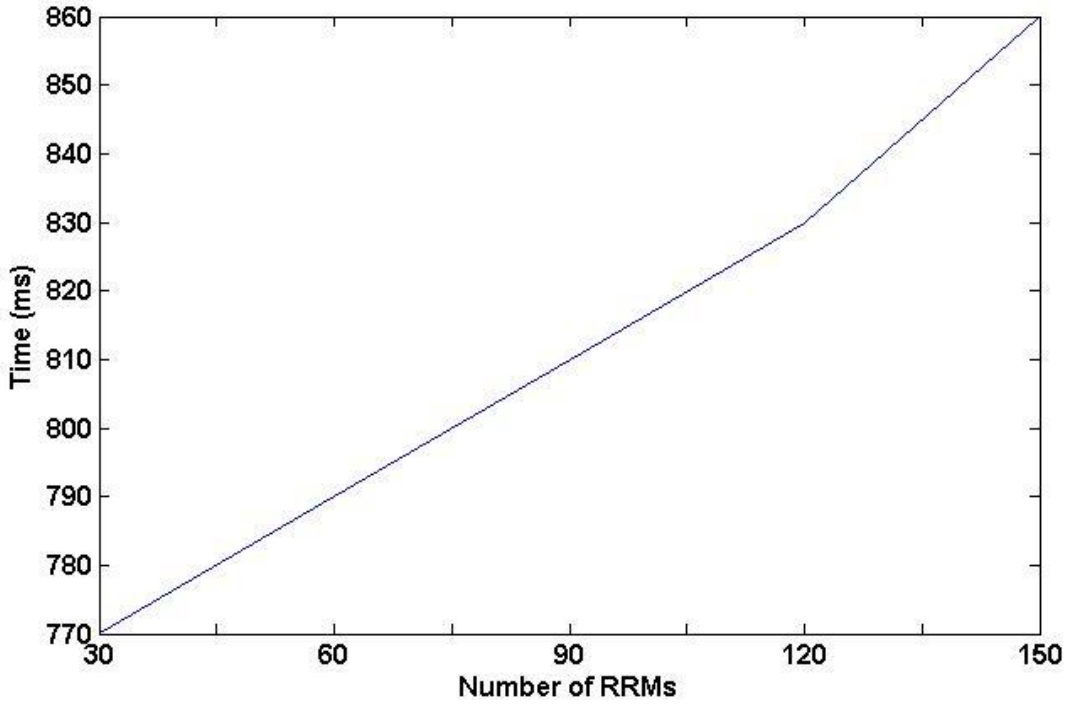


Figure 4.4: Average join time for a new RRM as a function of LG size

4.3.2.3 Comparison of Request Service Rate and Response Time

In Figure 4.5 to 4.6 we present the Request Service Rate (RSR) and Response Time (RT) within an LG for varying number of RRMs. We observe that the RSR remains linear with varying number of queries. The size of the LG has a direct bearing on RSR. Thus, a larger size of LG results in lower number of resource queries being forwarded to the SG.

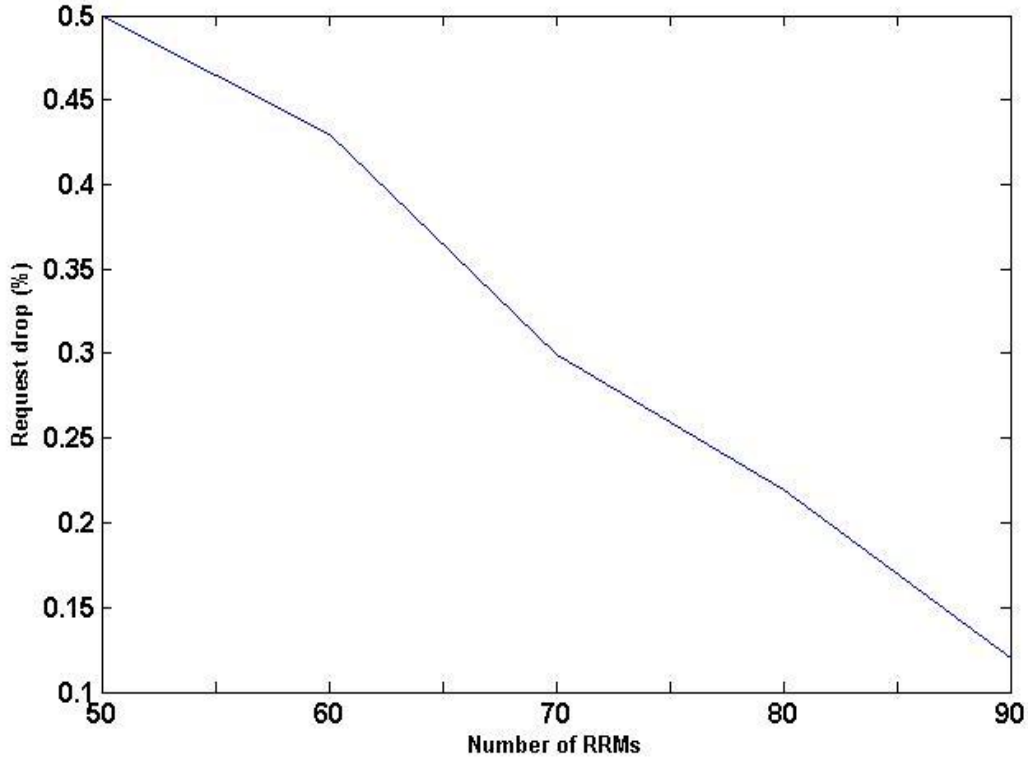


Figure 4.5: Request Service Rate

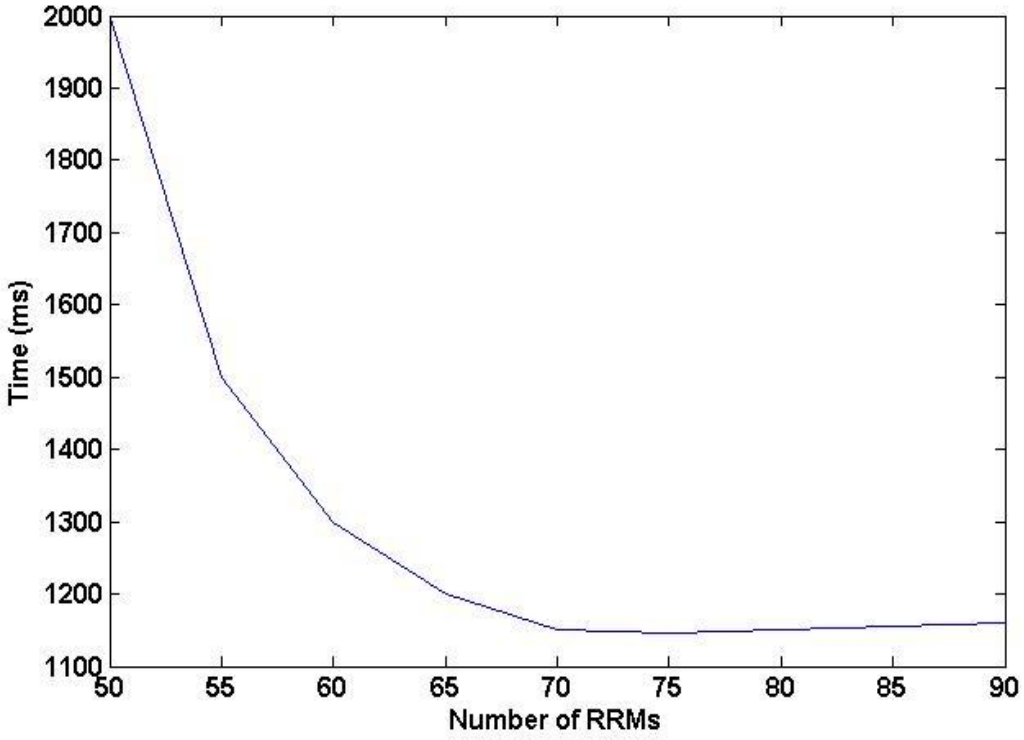


Figure 4.6: Response Time

4.3.2.4 Response time of Latency based Query

In the coming experiments we evaluated resources query responses from LG and SG under following preferences set by resource query generator/user:

a) Latency based resource query (LRQ): In this type of resource query there is an attempt to look out for resources which fall under pre-defined latency.
b) Cost based resource query (CRQ): In this type of resource query there is an attempt to look out for resources which falls under pre-defined cost.
c) Hybrid resource query (HRQ): It attempts to find resources which fall under the response time while maintaining the requested costs.

For LRQ, about 7% of the queries were handled by the SG and 93% of the queries were handled by the LG. Further there is an average increase of 41% in response time when the responses come from SG as compared to LG owing primarily to communication delays shown in Figure 4.7.

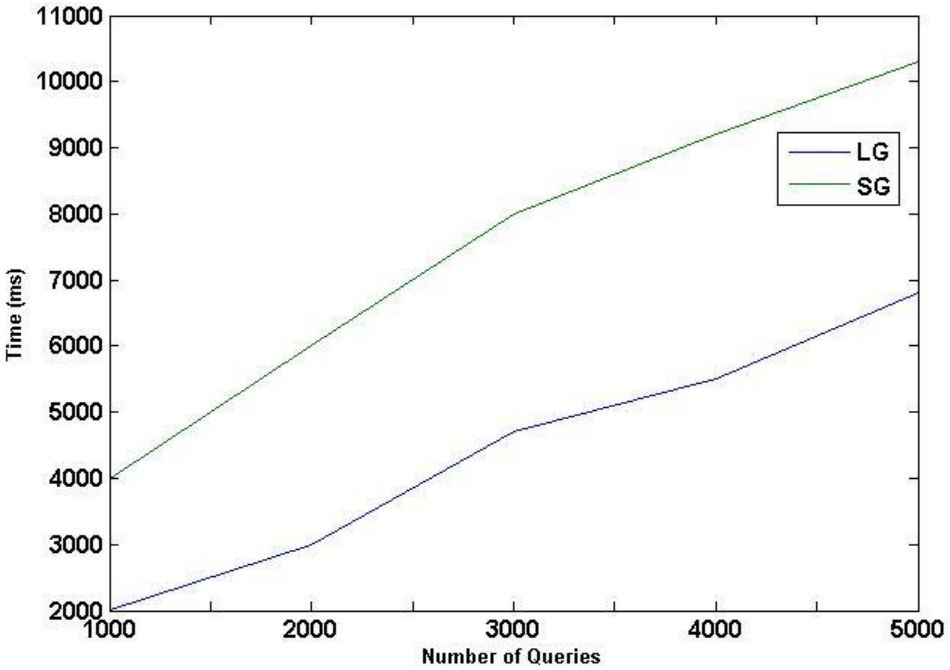


Figure 4.7: Average resource query response time for varying number of queries (LRQ)

4.3.2.5 Response time of Cost based Resource Query

For CRQ, about 43% of the queries were serviced by the SG and 57% of the queries were serviced by the LG.

As shown in the Figure 4.8, the queries serviced by SG suffers very high overhead (communication delay), resulting in high response time, where as in the case of LG communication delay is far less and hence responding in a better way.

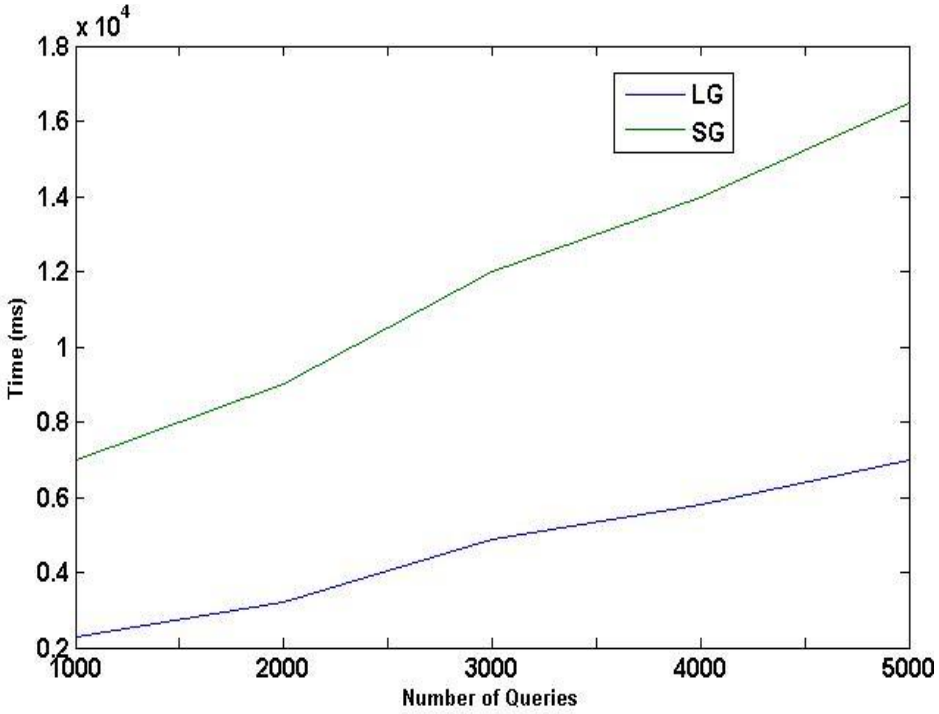


Figure 4.8: Average resource query response time for varying number of queries (CRQ)

4.3.2.6 Response time of Hybrid Resource Query

However for HRQ as shown in Figure 4.9, 93% of queries were serviced within LG and 7 % from SG and the resultant response time remains marginal high to LRQ and below CRQ.

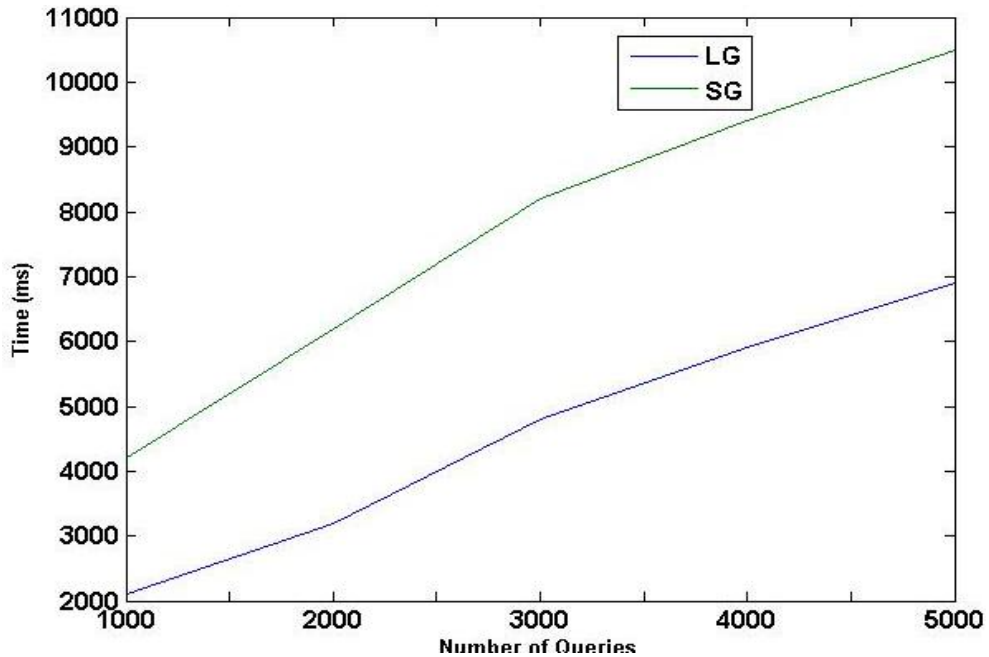


Figure 4.9: Average resource query response time for varying number of queries (HRQ)

4.3.2.7 Comparative view of CRQ, LRQ and HRQ

In Figure 4.10, a complete 24 hrs comparison result is displayed for the various resource techniques [55] [58] where we can observe that during flash crowd scenario (i.e. after every 3 hrs) CRQ responded in lowest time followed by HRQ and then LRQ.

This is due to the reason that in CRQ, 43% of requests are serviced by SG which hold sufficient resources for the requests, while in the case of LRQ 93% requests are serviced in LG which are insufficient during peak hours resulting in high waiting time for the requests.

However in normal conditions LRQ serviced the requests in lowest time if compare to CRQ and HRQ.

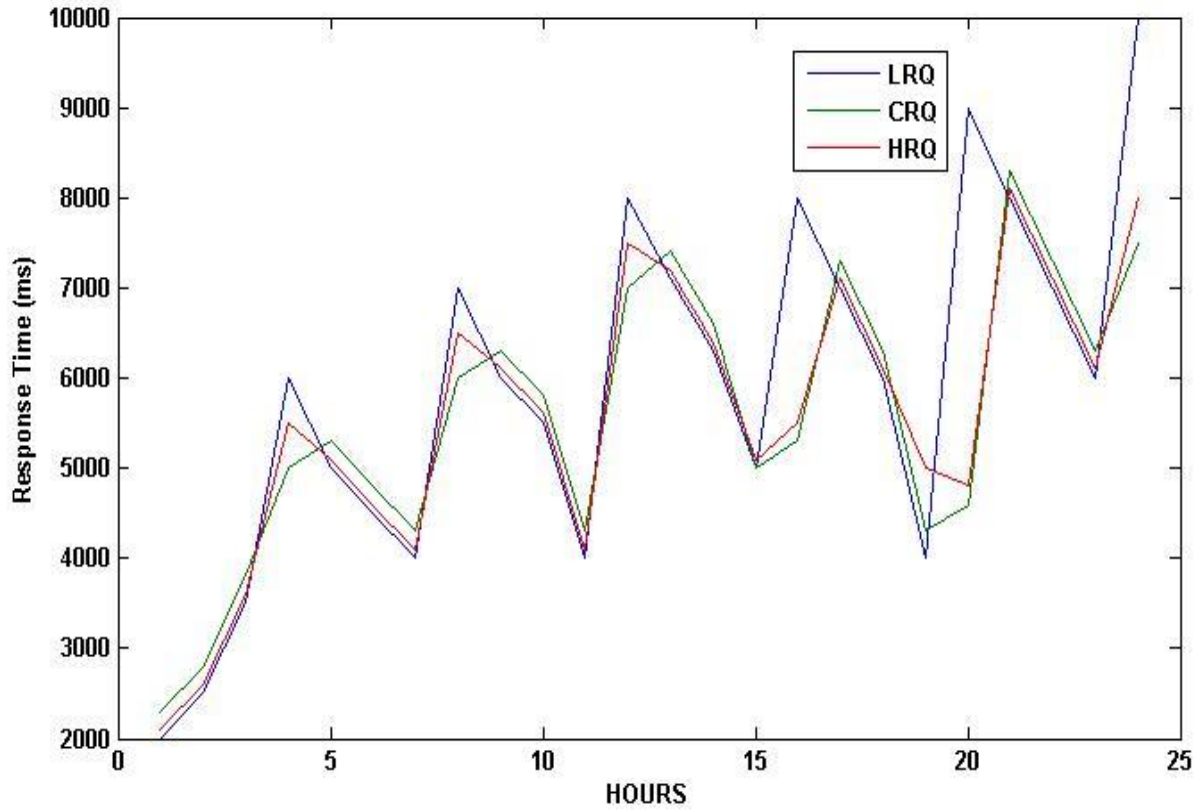


Figure 4.10: Comparative view of CRQ, LRQ and HRQ

4.4 Findings and Observations

This chapter presents a resource discovery mechanism in SITES which consists of two levels of groups local and super (global), inter-connected to each other. The application of P2P strategies for resource discovery in the intercloud environment has not been explored before. While overcoming the shortcomings of central mechanisms the use of JXTA based implementation provides some inherent benefits such minimized response time which is suited to the intercloud environment.

Chapter 5

Proposed Services Management Mechanism in SITES

Service deployment, orchestration, provisioning and SLA-compliance present a challenge in intercloud environments. A comprehensive and unified service management framework is required so that service providers and consumers can leverage the intrinsic benefits of services deployed across different cloud service providers exploiting latency and cost advantages while ensuring scalability, load-balancing and high-availability. This section presents architecture for unified services management in the intercloud environment allowing users seamless access to services through an optimal service selection mechanism providing on-demand ranking on several quality parameters.

5.1 Components of Proposed Mechanism

In this section the detailed system architecture of the proposed service management mechanism is discussed. A Cloud Service Provider (CSP) may encompass multiple data-centers at different geographical locations. Data-Centers belonging to a CSP are managed by a central broker which distributes the service requests within the CSP. The proposed service management mechanism in SITES framework is depicted in Figure 5.1. At a broad level the architecture consists of four entities: a) Services Cloud (SC) b) Cloud Service Providers c) Service Providers and d) Service Consumers. Services Cloud (SC) has its own broker for scheduling services across CSPs. Thus the SC broker

talks to individual CSP brokers for deploying/scheduling services across the intercloud. Each Service Provider which is desirous of hosting services in the intercloud needs to register its services with the Services Cloud (SC) by specifying the service characteristics (name, type, category, cost etc.), resource requirements, constraints and the deployment policy (fixed or geographically-aware auto-scaling). Each service may consist of multiple instances which can be deployed at any CSP in the intercloud. There are several standardized service description formats available such as [130] [131] [132]. Based on the inputs provided by the SP, the SC broker proceeds to deploy the service to a specific CSP or across multiple CSPs. The SC is also responsible for monitoring all the active deployed service instances in terms of their performance (load, resource usage, response times, latency, reliability, availability etc.) over sustained periods of time. There exists several standard service performance monitoring frameworks in practice [133]. This is needed to build a historical performance profile of each service and its usage patterns. These insights are used both by the SP in fine-tuning its deployment and provisioning strategies and by the end-user in selecting the services which best meet its requirements. Thus, the SC is responsible for orchestrating services across CSPs. Service orchestration can be achieved by using commonly used APIs of different CSPs. The proposed SC consists of compute resources contributed by each CSP participating in the intercloud and has its own master broker for scheduling services across CSPs. Thus the SC broker talks to individual CSP brokers for scheduling services across the intercloud. Each service may consist of a number of instances which can be deployed anywhere in the intercloud.

Further, SC employs caching for maintaining references to the most frequently-used services and the frequently-accessed data sets to speed up the service lookup time and minimize the service response time. The SC is not monolithic, but composed of several sub-components. Prominent among these are the user-

management module, services performance monitoring module, services deployment and management module and the scheduler module (for routing end-user requests to appropriate/selected service) through the CSPs broker.

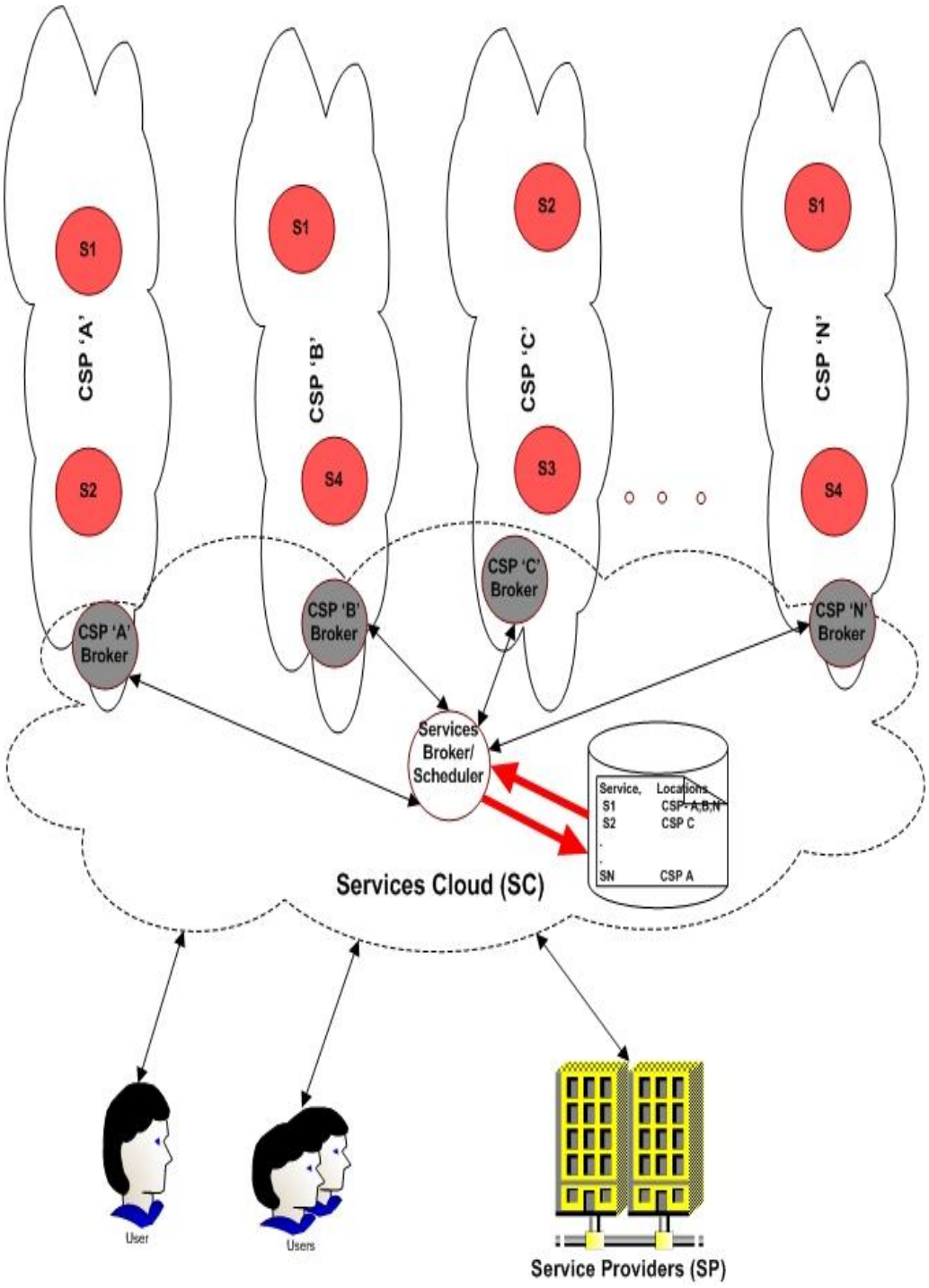


Figure 5.1: Conceptual Model of proposed Services Cloud (SC).

The SITES receives user's request, provides a customized ranking of various available services (across CSPs) and allocates the service based on user selection with the help of SC. Further, in the case the user finds the services not meeting their expectations, it can select a different service instance on the intercloud. Moreover, a service provider can also initiate a migration of a service instance to some other CSP to better meet its SLAs with the end users. The aim of the proposed mechanism is to enable the users to consume customized services as per their choice in the secure intercloud ecosystem and present them a variety of services options which best meet their requirements.

The main contribution of this section is to:

i. Proposes a Services Cloud for managing services in SITES.
ii. Presents a service ranking mechanism in SITES based on detailed performance monitoring and historical analysis.
iii. Presents a fine-grained control mechanism to the end users for optimal service selection based on different parameters.
iv. Presents model of automated service scaling and deployment for service providers in SITES based on dynamic service consumption patterns.

5.2 Algorithm of Proposed Mechanism

Proposed Service Ranking and Selection (SRS) algorithm requires performance metrics to evaluate any service. Service Measurement Index (SMI) [134] which is a set of business-relevant Key Performance Indicators (KPI's) that provide a standardized method for measuring and comparing a business service regardless of whether that service is internally provided or sourced from an outside company. SMI enables individual preferences to be the basis for what defines a good service [168]. We have identified five parameters as shown in Table 5.1 from SMI as a basis for defining service quality *(p)* for the proposed framework.

Table 5.1. Identified Parameters	
Service Quality Parameters	Standard Weightage
Network Latency (ϕ_1)	0.20 (wt_1)
Processing Time (ϕ_2)	0.20 (wt_2)
Reliability (ϕ_3)	0.20 (wt_3)
Reputation (ϕ_4)	0.20 (wt_4)
Availability(ϕ_5)	0.20 (wt_5)

i. Network Latency: We measure the round trip delay between sending a service request and receiving the response. Latency varies based on geographical location; therefore we continuously measure latency for each time slot 'n' as:

$$\phi_1 = ({}^{n}\sum_{t=0} (\alpha/\tau)) /n$$

where α = minimum latency observed during ideal time

τ = real latency observed at that time,

n = total number of requests.

ii. Processing Time: It is the measure of time a service takes to process a job request. This depends on several factors including the service architecture, speed of CPU and available cores, load on the sever hosting the vm which hosts the service and service load etc. Therefore, we calculated processing time as

$$\phi_2 = ({}^{M}\sum_{j=1} (\beta/\lambda)) /M$$

where β = minimum processing time observed during ideal time

λ = real processing time observed at any time,

M = total number of requests in a time period.

iii. Reliability: Reliability is an important factor in computing service quality. In this case we monitored service instances during peak and lean hours and observed the service responses received and calculated

the Mean Time Between Failures (MTBF). Once MTBF is calculated we can find out the reliability of a service by using following equation [135]:

$$\phi_3 = \exp(-t/MTBF) \leq 1$$

where e = exponential function,

t = minimum expected processing time for node to deal for any task's execution,

MTBF = the failure rate of the node at the give time.

iv. Reputation: Reputation of a service is a multi-faceted concept. Reputation is calculated based on the feedback provided by the user community about their previous experiences which is ranked between 0 and 1. On completion of every service usage cycle, users rate the service as '0' (not recommended), 0.25 (poor) '0.5' (acceptable), '0.75' (good) and '1' (excellent). Our Ranking algorithm calculates the reputation score similar to proposed by [136] which includes service ID, consumer ID, timestamp (used to determine the aging factor of a particular service rating).

Therefore,

$$\phi_4 = {}^{N}\sum_{i=1} S_i \lambda^{di} \leq 1$$

where N = number of ratings for a service,

S_i = ith service ratings

λ = inclusion factor i.e. $0 \leq \lambda \leq 1$

di = age of the ith service ratings in days.

The inclusion factor 'λ' used to signify the recent ratings of the service. For example smaller value of λ means more recent ratings which have more impact on reputation and larger λ means more of the reputation influences the reputation scores.

v. Availability: This parameter is simply obtained by observing the total time duration for which the service remains down relative to the total time the service is offered. Therefore,

$$\phi_5 = 1 - (t_{down} / t_{up} + t_{down}) \leq 1$$

t_{up} = time for which service remains up during a partuclar period of time

t_{down} = time for which a service remains down during a particular period of time

Based on their contribution under SRS, a weight age is given to all the Service Quality factors and as a final point aggregated to compute ranking score (R) of a service is given by:

$$\mathbf{R} = {}^{n}\sum_{i=1} \; \boldsymbol{wt_n \, \phi_n \leq 1}$$

The Service Monitor at each vm where the service is deployed sends performance information to the Service Performance Monitoring module in the Services Cloud every five minutes. Service Analytics uses this information to compute the Ranking Scores (R) for each service instance. By default equal weights for the five service quality parameters are used in the computation of ranks, although individual service consumers can define their own weights to derive customized service rankings as per requirement. After computing the rank, a service list is presented to the user and final negotiation process is initiated as shown in Algorithm 3. Service ranks can potentially be recomputed every five minutes when new data comes in from the service monitor. The service consumer is notified if the service ranks changes or a service quality parameter changes significantly since the last ranking cycle for taking suitable action including possible selection of a new service. Figure 5.2 presents an indicative snapshot of sample data and computed ranks for different service instances deployed at different CSPs/locations.

Algorithm 3: Service Ranking and Selection

```
1:  // Collecting users' preferences and discovering
2:  for each user
3:  getCustomWeights(userweights)
4:  getCostConstraints(cost)
5:  getServiceCategory(category)
6:  services =  ServicesDirectory.lookup(category)
7:  // Computing rank and negotiating SLAs
8:  for each service in services[i]
9:  topServices=computeRank(services[i], userweight)
10: user.showRank(topServices)
11: user.recordSelection(service)
12: service.negotiateSLAvalues(SLAvalues,cost)
13: end for
14: end for
```

	Date:10:15:2013 Time: 01:00:00 – 01:05:00						Date:10:15:2013 Time:01:06:00 - 01:10:00						Date:10:15:2013 Time:01:11:00-01:15:00					
	Individual Score Weight (Wt) = 0.20					Total Score	Individual Score Weight (Wt) = 0.20					Total Score	Individual Score Weight (Wt) = 0.20					Total Score
	$\phi_1.Wt$	$\phi_2.Wt$	$\phi_3.Wt$	$\phi_4.Wt$	$\phi_5.Wt$	R= $^{5}\sum_{i=1} wt\ \phi_n$	$\phi_1.Wt$	$\phi_2.Wt$	$\phi_3.Wt$	$\phi_4.Wt$	$\phi_5.Wt$	R= $^{5}\sum_{i=1} wt\ \phi_n$	$\phi_1.Wt$	$\phi_2.Wt$	$\phi_3.Wt$	$\phi_4.Wt$	$\phi_5.Wt$	R= $^{5}\sum_{i=1} wt\ \phi_n$
S1.L3	0.08	0.12	0.134	0.108	0.174	0.40	0.08	0.12	0.134	0.108	0.174	0.40	0.08	0.12	0.134	0.108	0.174	0.40
S1.L2	0.09	0.09	0.09	0.112	0.068	0.45	0.09	0.09	0.09	0.112	0.068	0.45	0.09	0.09	0.09	0.112	0.068	0.45
S1.L1	0.174	0.108	0.086	0.068	0.134	0.57	0.174	0.108	0.086	0.068	0.134	0.57	0.174	0.108	0.086	0.068	0.134	0.57
S1.L4	0.174	0.108	0.086	0.068	0.134	0.59	0.174	0.108	0.086	0.068	0.134	0.59	0.174	0.108	0.086	0.068	0.134	0.59
S1.L6	0.45	0.65	0.67	0.112	0.87	0.63	0.45	0.65	0.67	0.112	0.87	0.63	0.45	0.65	0.67	0.112	0.87	0.63
S1.L5	0.10	0.07	0.10	0.112	0.068	0.66	0.10	0.07	0.10	0.112	0.068	0.66	0.10	0.07	0.10	0.112	0.068	0.66

Figure 5.2: Indicative sample snap shot of hourly score on the basis of various parameters for service instances across CSPs.

5.3 Implementation

In this section the evaluation of proposed services management mechanism is done. For this a real world experimental setup is established to conduct various experiments and analyzing their results.

5.3.1 Experimental Setup

We collected real-world data pertaining to a sample deployed service across three popular CSPs -Amazon EC2 [137], Windows Azure [138], and GoGrid [139]. The experimental data served as a basis for designing our custom simulator built on top of Cloudsim[129]. The real-world test environment had a total of 3 CSPs, 6 data centers at 6 different physical locations and 10 service instances deployed at various CSPs locations. We evaluated a Photo Storage service, which is implemented on each CSPs datacenter location. We further consider only one type of virtual machine instance in different CSPs since in [140] authors have concluded that both medium and large vm instances gets overloaded with the similar level of concurrent requests. Therefore in order to save experimental cost without compromising on service performance measurements we use instances shown in Table 5.2 for service deployment. We replace actual CSP names with "CSP_1, CSP_2 and CSP_3" and replace actual data-center locations with generic locations "L_1, L_2, L_3, L_4, L_5 and L_6".

Table 5.2 Virtual Machine Configuration

OS	CPU	Cores	Memory(MB)
Linux/ubuntu	AMD Opteron 2 GHz	1	1024
Windows Server 2012	Intel Xeon E52670	1	1024

We used HTTP HEAD Requests using curl for latency measurements instead of ICMP ping command because MS Azure has barred incoming/outgoing ping requests [141]. We take measurements for every 5 minute interval and compute averages for each parameter every 1 hour. Intuitively network latency is dependent on the geographical locations of the user with respect to the data center, but routing inefficiencies, time zone differences and usage of multiple data centers in an intercloud environment can lead to counter-intuitive results. The results obtained in the test setup are shown in Figure 5.3 which concurs with those obtained in [142]. It can be seen that the average percentage variation for one location is up to 18% i.e. latency varies significantly over 24 hours for the same location which implies that any QoS-complaint service deployment scheme requires factor these latency variations to effectively meet defined SLAs.

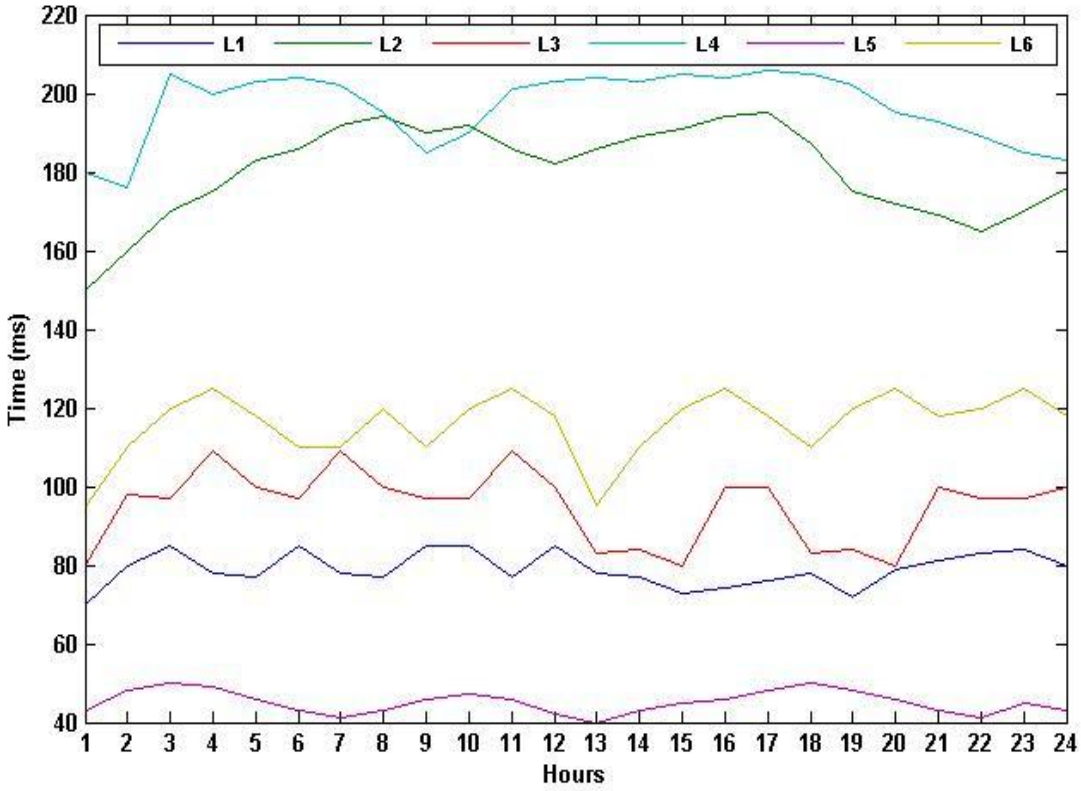


Figure 5.3: Observed latency for different service instances over a 24 hour period.

Further we use Manage Engine's Application Manager [143] to measure service performance in a virtualized environment. We also used httperf [144] as our benchmark to generate the workload for service instances deployed in different locations. Since our service is web-based, therefore detailed information on the number of connections, rate of connections, request size, request rate, reply time/size and rate etc. is required. We perform our benchmark tests on each deployed service instance in our test environment. We measure the variations in the number of requests processed per minute and results are depicted in Figure 5.4. The number of requests processed per minute is dependent on the latency between the user and location of the service and the variations of up to 60% are observed. Here the location of the user remains the same. In the real world varied users locations can result in greater variations in service instances performance. We use the photo-storage service to look-up and download a 5KB file in all these tests.

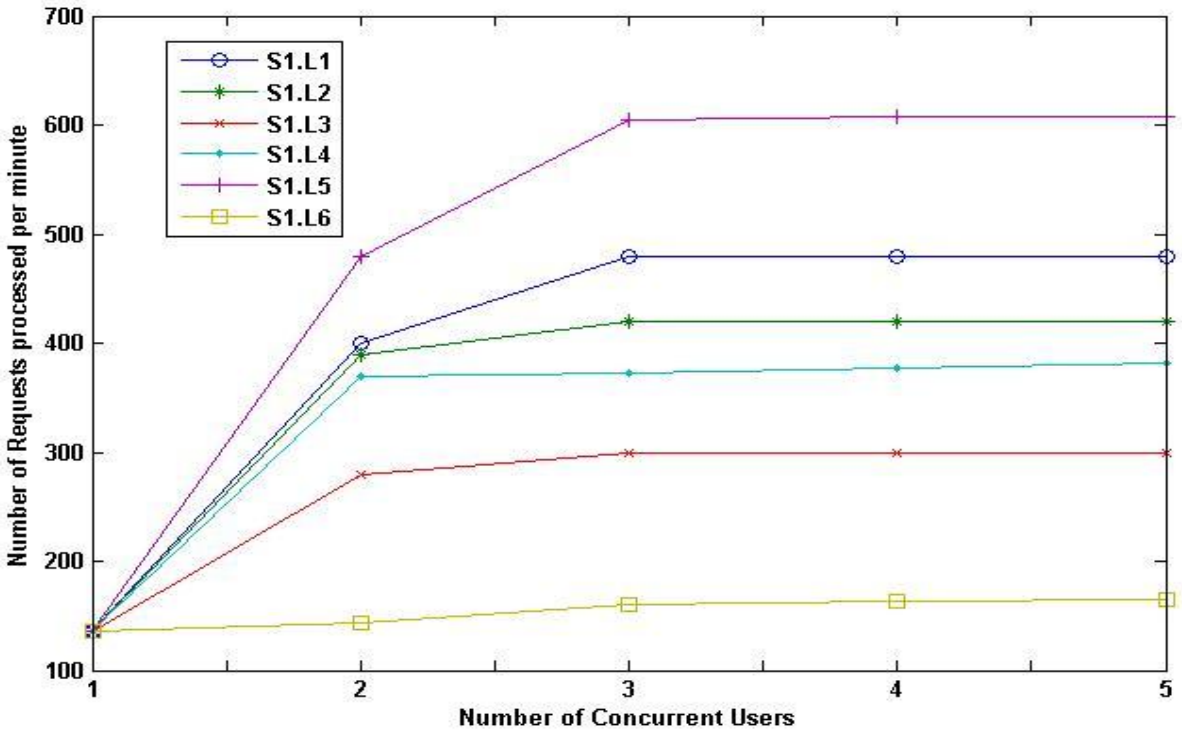


Figure 5.4: Observed Number of Requests processed/minute by service instances deployed in different locations.

This test setup served the following purpose:

a) Benchmarking service behavior with varying workloads in a real world scenario.
b) Provide strong basis for the simulation of service behavior in the proposed Services Cloud framework

5.3.2 Results Analysis

To validate the proposed framework we have developed a custom simulator which works on real-world data obtained in Section 4 as its base input. We have built the service ranking logic on top of cloudsim base classes and use the results obtained from service ranking as an input to route service requests to cloudlets in different datacenters. We conducted experiments to assess the impact of our scheme on both the Service Providers and Service Consumers We consider three CSPs with two data centers each for a total of 6 data centers. Each datacenter is located in a different geographic location. A total of 25 service instances with varied deployment strategies and 100 service consumers are considered for simulation purposes. The service consumers are located in '20' different geographical locations including the six locations of the data centers with different latencies. Each CSP follows different pricing policies for resource usage. Further, we assume that each vm can host one service instance with different processing time, reliability and availability and the vm cost per hour varies from 0.10-0.50.

5.3.2.1 Evaluating geographical implications for service usage

The aim of this experiment is to assess the impact of service location on the response time. We measured the average response time for six instances of the same service deployed in each of the 6 data centers. In Figure 5.5 we can observe that the average response time obtained by different instances of the

same service to download a 5KB image file for increasing number of user requests. We can see that the same service is responding differently if deployed in different locations and variations of upto 600% are observed in the response times. The major component in this variation is the actual latency between the service consumer and the service location, while variations due to differences in the processing capabilities of vm's belonging to different CSPs is only upto 10%. This experiment results agree with the conclusion mentioned in [145] which highlighted the latency factor for effective service response.

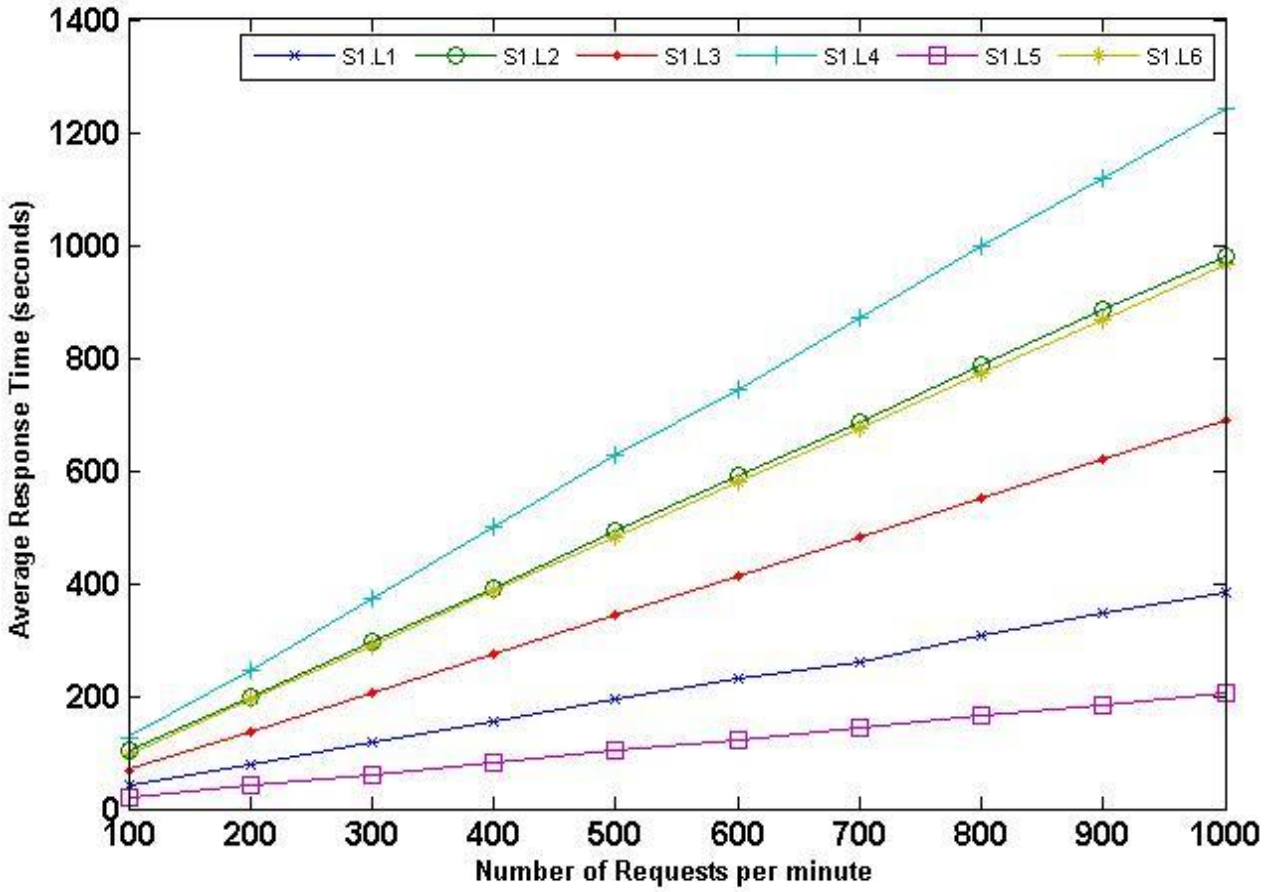


Figure 5.5: Average Response Time obtained for same service instances deployed at 6 different geographical locations.

5.3.2.2 Evaluating Deployment Scenarios

In this experiment we evaluated different deployment scenarios as discussed below and measure the impact on the observed average response times, requests drops and profits for increasing number of requests.

a. Single Instance Single Location (SISL): It represents a service which is deployed in only one location and has got only single service instance to serve users requests.
b. Multiple Instances Single Location (MISL): Same service and associated components replicated at single CSP and same data center (physical location). In this scheme all the requests comes to a single instance and when this instance get overloaded another instance is created on the same physical location to handle users requests.
c. Multiple Instances Multiple Locations (MIML): Multiple instances of service components (distributed replicated services) deployed at different data centers (physical locations) of 1 or more CSPs. In this scheme the requests are distributed across different instances in intercloud environment. Different strategies of load balancing can be applied to serve users requests due to huge availability of service instances. For example consider a scheme in which load balancing is done to exploit geographical proximity. However, initially this type of deployment scenario is costly as compared to SISL.

Here we use the same configuration service instance, but deploy them in different strategies (SISL, MISL and MIML). In the cases of MISL and MIML three service instances are used to cater to users' request. In Figure 5.6 shows the Cumulative Distribution Function (CDF) of average response time over the file size 5 KB that were generated during the experiment as results. It has been observed that in the case of MISL and MIML the response time has been decreased by 62% and 78.75% respectively as compared to SISL. This is

obviously due to SIML and MIML having more service instances and MIML is exploiting geographically proximity.

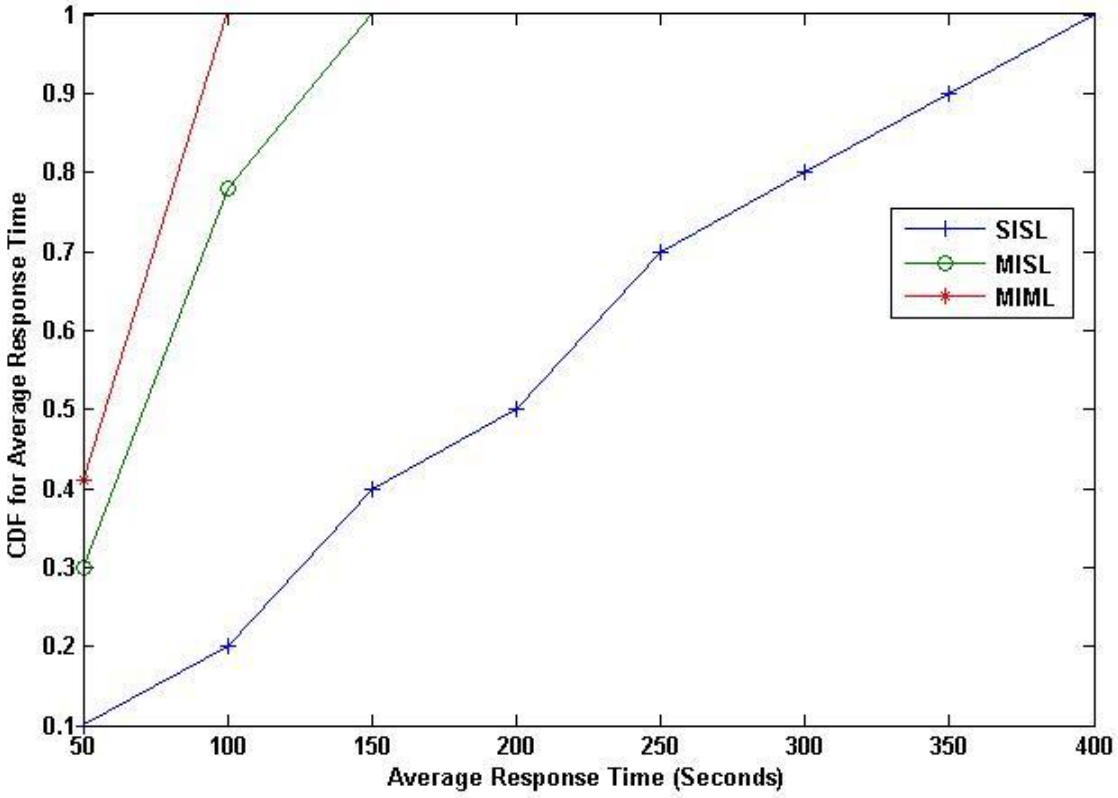


Figure 5.6: CDF for average response time obtained for different service deployment strategies

The processing time for a service request is dependent upon VM's operating system/hypervisor, hardware configuration of server, scheduling logic, web-server performance, service queue length etc., but shows a maximum variation of 10% between CSPs. In terms of latency since SISL and MISL are located at same location they offer almost same latency for each service instance but in case of MIML deployment the average observed latency is reduced significantly as service instances are deployed at different geographical locations to serve scattered user requests more efficiently. Therefore the main reason behind the performance degradation of MISL as compared to MIML is its static service location. This results in 37% lower average response time for a large number of geographically dispersed user requests.

We assume the maximum waiting time for the response is 100 seconds before the request is considered dropped, Therefore the request drops for increasing number of user requests were measured for each of the service deployment scenarios. Results are displayed in Figure 5.7. As expected, SISL suffers from high request drop due to its single instance getting overwhelmed sooner followed by MISL and MIML. MIML due to its lower overall response time and latency is able to service approximately 22% more requests compared to MISL without requests getting dropped.

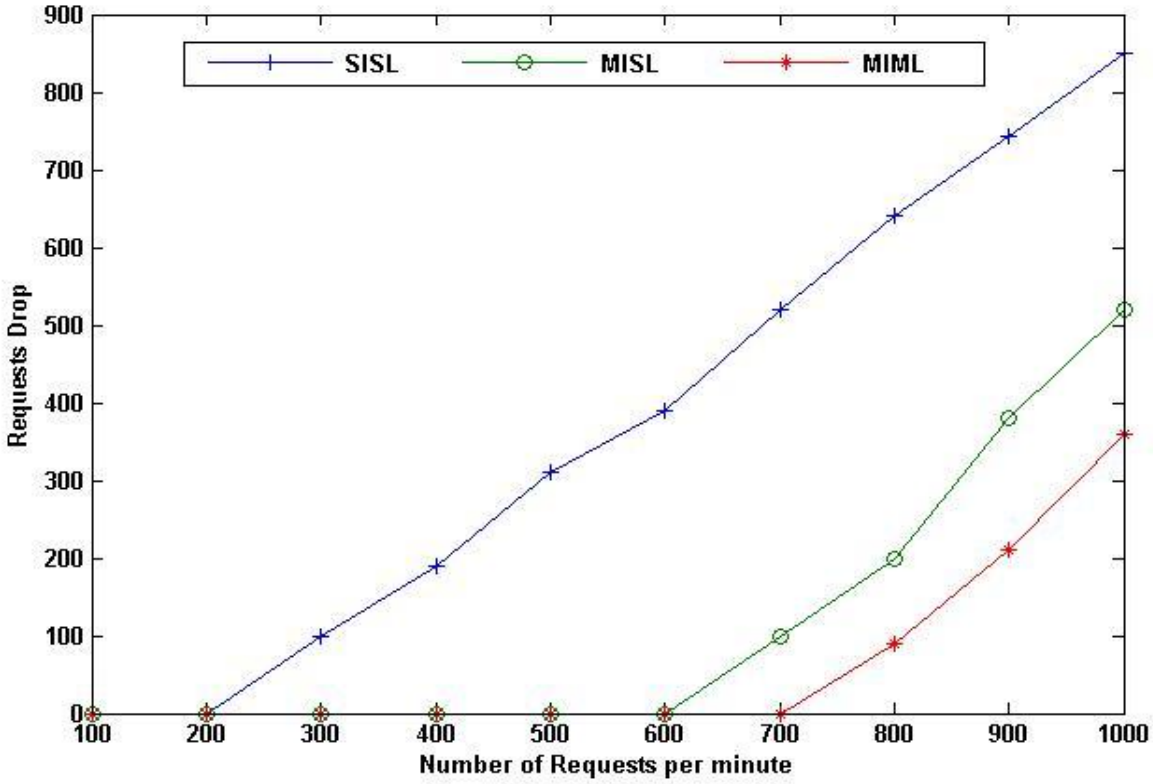


Figure 5.7: Number of request drops per minute for a single service under different deployment scenarios

It is noticeable that for up to 200 requests per minute, SISL, MISL and MIML have negligible request drops but beyond it SISL drops nearly 90% of received requests due to increasingly longer wait time. Further MISL with its 3 service instances starts dropping requests at close to 600 requests/minute but MIML remains stable till 700 requests per minute.

Moreover, the service deployment scenarios have a direct impact on the overall revenue generated for the SP. For up to 200 requests/minute, SISL is the most cost-effective deployment strategy, but when request/minute goes beyond 200 requests/minute SISL starts dropping requests. The impact on overall profit under various scenarios is depicted in Table 5.3. The potential revenue loss for a SP comprises of a) SLA violations and b) request drops since both these cases result in a penalty [146] [147].

Table 5.3 shows that SISL is cost effective as compared to MISL and MIML when the frequency of request is small (in our case up to 200 requests/minute) but when the frequency of requests increase MIML and MISL are naturally more effective as they provide more service instances and scale better. Therefore more the number of service instances lesser the response time since more instances are available to handle the requests. Further, with increase in the number of service instances in intercloud environment the profitability can be increased by a) exploiting lower latencies through geographical proximity b) selecting the most cost-effective CSPs. We can also notice that with more number of service instances, SLA violations decreases significantly. Further, auto-scaling by the Services Cloud provides a practical mechanism to increase service instances to handle peak-load and reduce service instances proportionately as requests drop while adhering to SLA agreements. Thus, optimal deployment from the SP perspective is ensured. Therefore the shifting from SISL to MISL or MIML is dynamic and depends upon the frequency and geographical origin of requests.

It was observed that MIML performance is up to 40% better when requests are from geographically diverse origins and request loads are high. MIML can thus exploit time-zone differences and benefit from heavily discounted non-peak hour prices for compute resources at different CSPs.

Table 5.3: Profit projections for a SP for varying number of requests/minute		
Case 1: For 100 requests/minute		
Deployment Scenario	SLA violations	Average Profit ($/hour) = (Total fee earned- Expenditure of deployment of service instance- Penalties)
SISL	0	59.9
MISL	0	59.7
MIML	0	59.75
Case 2: For 500 requests/minute		
Deployment Scenario	SLA violations	Average Profit ($/hour) = (Total Fee earned- Expenditure of deployment of service instance- Penalties)
SISL	30%	263.7
MISL	0	299.7
MIML	0	299.75
Case 3: For 1000 requests/minute		
Deployment Scenario	SLA violations	Average Profit = (Total Fee earned- Expenditure of

		deployment of service instance-Penalties)
SISL	80%	257.5
MISL	33%	551.7
MIML	26%	563.75

5.3.2.3 Evaluation of Scheduling Schemes

In this experiment we compare the SRS scheme with a Hybrid Service Selection (HSS) scheme which attempts to minimize response time while reducing service consumption costs. It is similar to the scheme proposed in [8] [148] in which broker focused on performance optimization with a cost constraint. This policy takes care of both cost and workload of a service. It uses cost as an upper bound and finds out the least loaded service. We have compared the performance of the HSS policy with our SRS policy on several parameters and evaluated the benefits of SRS in optimizing end-user service selection. Further we use MIML deployment strategy for both the SRS and HSS and observe the overall response time and SLA violations. For this we deploy six service instances across different CSPs with varying cost and workload characteristics. The service instances are also assigned values for availability, reputation and reliability. SRS is clearly the more advanced selection policy since it takes into account the availability, reliability and reputation of a service instance while trying to optimize communication and resource usage costs. In our tests HSS selected services with low availability for 7% of the requests, services with low reliability for 6% of the requests and services with low reputation for 10% of the cases. This result in significantly higher SLA violations compared to SRS selection policy.

We observe in the Figure 5.8 that poor selection choices by the HSS result in the higher response times compared to SRS to the tune of 17% on average. SRS makes more credible choices based on availability, reliability and reputation of service instances.

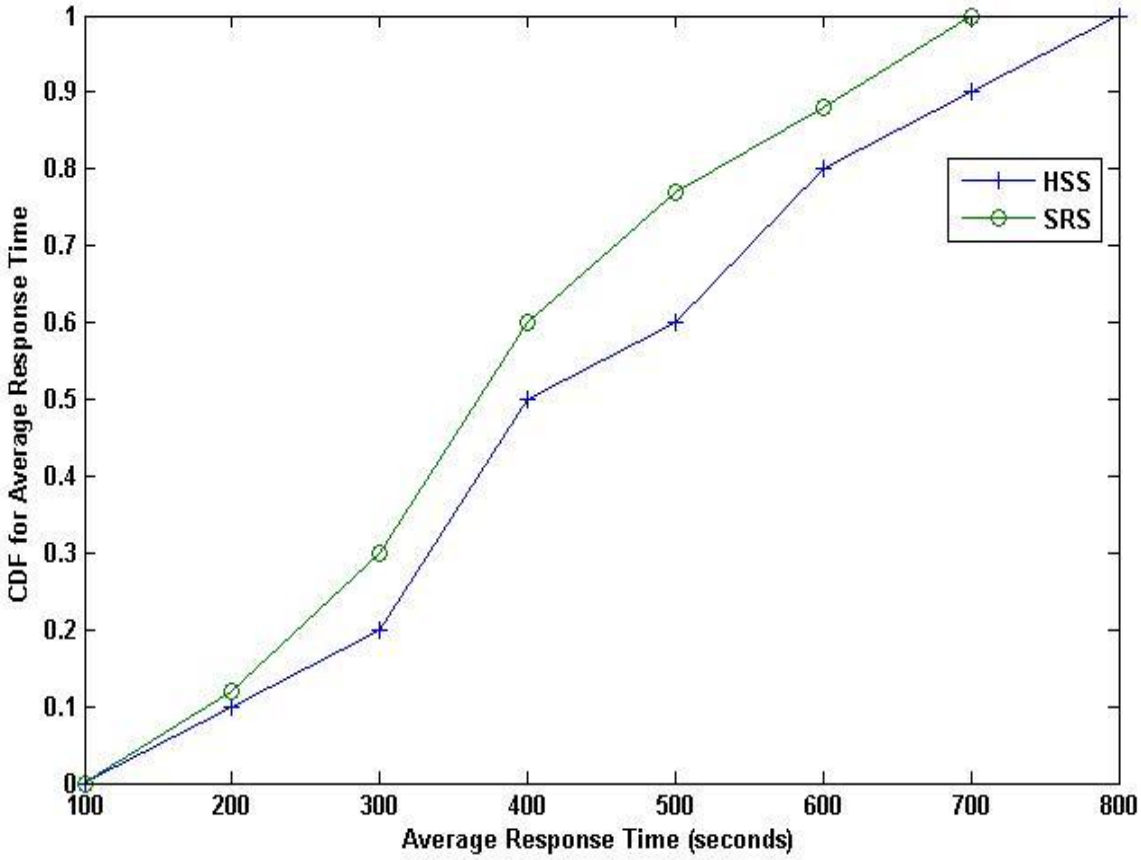


Figure 5.8: CDF for average response time for HSS and SRS service selection policies

Further, service selection schemes can also be based on:

a) Cost based Service Selection (CSS): In this type of selection policy scheduler looks out for the cheapest service offer (in case of similar competing services). It selects the least cost service and allocates it to the user.

This kind of service selection is best fit for the users who are not performance oriented and want to pay less. Here the probability of response-time based SLA- violations is very high.

b) Work load based Service Selection (WSS): It monitors the work load on each service deployed in datacenter and find out the least loaded service with less service requests pending. This type of policy is optimal when performance is paramount but very prone to cost based- SLA violations.

Therefore in Table 5.4 and Table 5.5 we compare HSS and SRS on cost optimization achieved and SLA violations observed with CSS and WSS selection policies as the base. SRS outperforms HSS due to: a) qualitatively better service selection and b) geographically-aware auto-scaling.

Table 5.4:Optimization achieved by HSS and SRS over CSS service selection policy

Parameters	**HSS**	**SRS**
SLA-violations optimization	31 %	58%
Cost optimization	9%	22 %

Table 5.5:Optimization achieved by HSS and SRS over WSS service selection policy

Parameters	**HSS**	**SRS**
SLA-violations Optimization	22 %	47%
Cost optimization	19 %	34

5.3.2.4 Service Instance Transition Behavior of SRS

In this experiment we tracked service instance transitions (Service S1 deployed at locations L1 through L6) for varying number of user requests/hour as shown in Table 5.6. SRS selects the appropriate service instance combinations while optimizing the service hosting cost for the Service providers. We begin with one service instance at each location. At hour 1, S1.L6 is sufficient to handle up to 6200 request/hour with the associated hosting cost of 0.20 $/hour. Between hours 2 to 4 due to flash crowd scenario (peak load) SRS performs service replication at L6 and L2 choosing a combination of service instances that lowers the hosting cost for the service provider in each case.

The replicated services are also automatically decommissioned when not required (Table 5.6).

Table 5.6: Service instance transitions in response to dynamic user requests.

Hour	**Requests/hour**	**Service Instances**	**Total Cost/ hour**
1	6000	S1.L6	0.20
2	10000	S1.L6+ S1.L6	0.40
3	17000	S1.L5+ S1.L2+ S1.L2	0.65
4	22540	S1.L6+S1.L5+ S1.L2+ S1.L2+ S1.L2	0.95
5	12000	S1.L6+ S1.L6	0.40
6	4000	S1.L2	0.10
7	3950	S1.L2	0.10
8	3900	S1.L2	0.10
9	4000	S1.L2	0.10
10	4799	S1.L2	0.10

11	4045	S1.L2	0.10
12	4000	S1.L2	0.10
13	4350	S1.L2	0.10
14	4298	S1.L2	0.10
15	4220	S1.L2	0.10
16	5700	S1.L6	0.20
17	6000	S1.L6	0.20
18	3900	S1.L6	0.20
19	5800	S1.L6	0.20
20	10000	S1.L6+ S1.L6	0.40
21	11000	S1.L6+ S1.L6	0.40
22	11098	S1.L6+ S1.L6	0.40
23	10086	S1.L6+ S1.L6	0.40
24	10035	S1.L6+ S1.L6	0.40

5.3.2.5 Computation Time for SRS

In this experiment computation time for SRS strategy is evaluated. The SRS strategy involves on-demand ranking of services based on customized weights provided by the end users which facilitates fine-grained control over service selection and consumption. This approach therefore involves computing service ranks in the context of the user-defined weights and incurs additional computational overheads compared to schemes which use a static weighted formula for determining service ranks. This is because the service ranks need to be recomputed for each user. Figure 5.9 provides the average processing time for computing service ranks for up to 500 service instances for a single user. The results show that with the increase in requests the computation time is

growing linearly. This computation however needs to be performed only once before the service consumption phase for each user session.

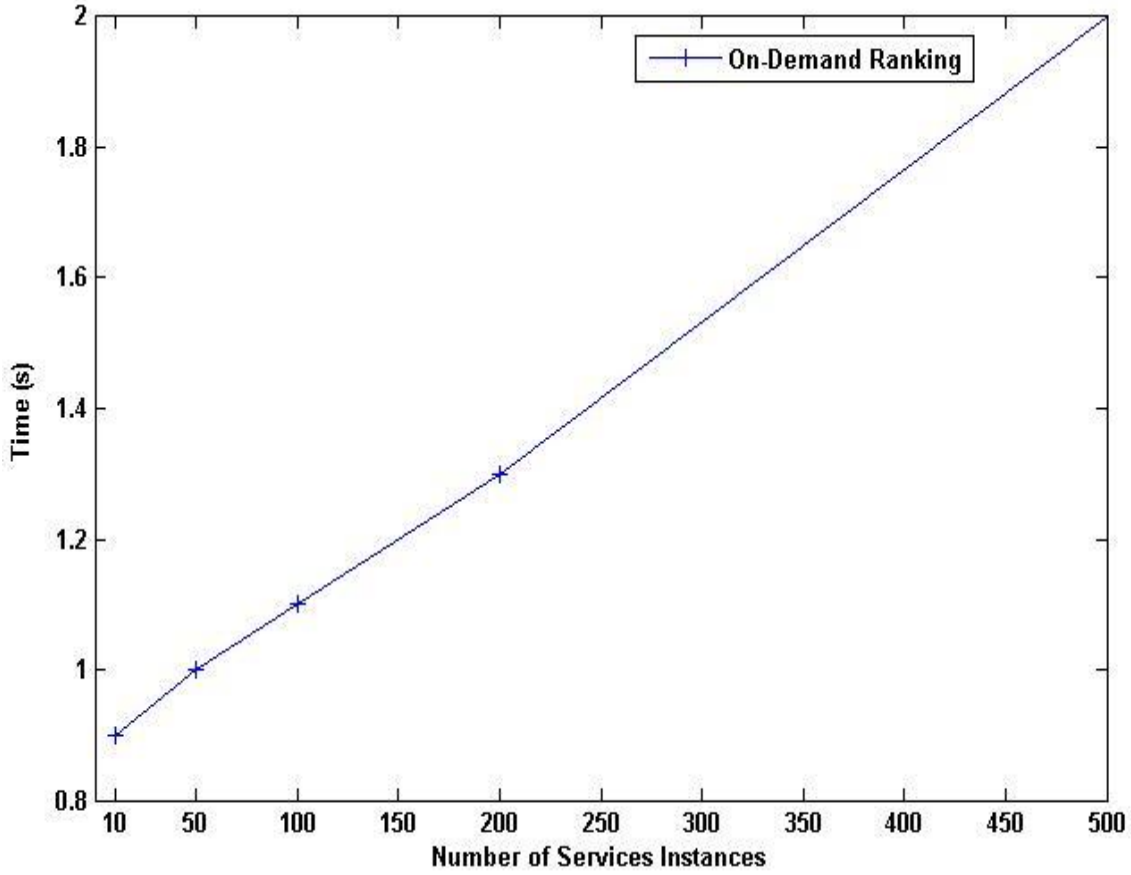


Figure 5.9: Average processing time for on-demand ranking of services instances

5.4 Findings and Observations

This chapter proposes a services management mechanism which is a step towards a Unified Services Management Framework for SITES which facilitates a) optimal service deployment and geographically-aware auto-scaling and b) optimal service selection and consumption by the end-user in a seamless manner. We evaluated the effectiveness of the proposed framework using a custom simulator based on real world service performance measurements across different CSPs. The results indicate that from a service provider perspective

achieving high performance at reasonable cost is dependent on the service deployment scheme. It was further observed that mere load-based auto-scaling is not effective to deal with flash crowd scenario but instead geographically-aware auto-scaling to exploit latency benefits in an intercloud environment is more efficient for global services. Further, a customized ranking mechanism for service consumers allows greater optimization at an individual level rather than with a one-size-fits-all approach.

Chapter 6

Proposed Security Mechanism in SITES

Distributed computing has shown a great prospect [168] and Intercloud can be seen as its future. However recently there has been a spate of security attacks using hired cloud computing infrastructure especially Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks. DoS/DDoS attacks are some of the most widely prevalent attacks on the internet which target well known web-sites/applications/services by overwhelming them with malicious requests affecting their performance or causing them to crash. With cloud computing offering huge amount of computing resources on a pay-per-use basis, it has become easier for malicious users to hire cloud resources and launch DoS/DDoS attacks from multiple locations. This makes it difficult to identify the source of origin of these attacks and contain their damage. Moreover, some malicious users have used cloud resources to host Attack-as-a-Service which allows third-party users to launch DDoS attacks by just specifying the intended targets. Thus, there is a need to detect malicious applications hosted on the cloud and prevent them from utilizing the vast computing infrastructure of the cloud to launch planetary-scale attacks which can potentially cripple the internet including critical business and government resources. As already discussed in Chapter 2, most of the work done on detecting DDoS attacks is based on network traffic analysis at the recipient end i.e. at the server hosting the application under attack. It typically involves creating a baseline of normal

incoming traffic patterns and detecting any abnormal variations from the known baseline. Once determined that the host is under attack, there is an attempt to drop/ignore packets from the source IP address from where the attack is originating. However, performance is definitely degraded and unless high-availability and load-balancing features are implemented to defray the attack, the application/service under attack is severely impacted [149].

This chapter proposes a novel mechanism for effectively and efficiently detecting malicious applications hosted on the cloud. It involves changing the perspective on detecting and containing DoS/DDoS attacks from the recipient to the source. As Cloud Service Providers (CSPs) rent out Virtual Machines (VMs) to customers for their computing needs and are responsible if malicious users use their resources to launch global DDoS attacks. Hence, CSPs need to check the malicious use of their resources and the proposed method allows CSPs to detect and contain DDoS attacks originating from their computing infrastructure. In an intercloud environment if each CSP can prevent DoS/DDoS attacks originating from their infrastructure, the entire intercloud becomes free of such attacks. This provides a strong motivation for the CSPs to participate in this collaborative process of detecting and containing malicious services. One conceivable method to detect malicious hosted applications in the cloud is packet sniffing to understand the kind of traffic being generated and transmitted by the hosted application. However, this would constitute a violation of privacy and be unethical on part of the CSP. Hence, a privacy-preserving mechanism is required which is still able to detect whether the hosted application is launching DoS/DDoS attacks using the CSP's compute infrastructure. The proposed mechanism is based on the concept of application/service profiling, which involves creating detailed performance and behavior profiles of hosted code. The run-time behavior of the hosted application is further compared to the performance profiles in a global database of known malicious applications for

quick detection. Unknown applications are referred to expert system for classification based on deviations from known normal behavior and the database expanded. This chapter flips the security focus from preventive attack techniques [150] [151] to originating attack. Detailed experimental analysis of known malicious applications is used as a basis for initial detection and determining the effectiveness of the proposed scheme.

6.1 Components of Proposed Mechanism

At the CSP, an application monitor module is instantiated per VM allocated by the cloud hypervisor to the customer as shown in Fig 6.1. Its job is to monitor the installed application image (static) and resource usage and behavior (dynamic) of the customer's application running on the VM. The static profile includes the number of files installed, their checksum and size in bytes etc. If the static profile (installed image) of the hosted application matches against the profile any of the known malicious application it is prevented from executing. The dynamic profile of the application includes dynamic resource usage such as CPU, Memory and I/O as a function of time. If this run-time behavior of the hosted application matches against that of any known malicious application (DDoS application) it is terminated preventing the DoS/DDoS attack from being propagated further. The profile matching is done through comparison of time-series data (CPU, Memory and I/O) in real-time with the data of known malicious applications.

If the behaviour profile of the application does not match any known malicious applications, but the run-time behavior (trend) exhibits anomalous behavior such as an immediate and sustained spike in outbound traffic, hitting a pleateau after some time and rapid decrease in inbound traffic (indicating that the target application is overwhelmed) it is referred to an expert system, which classifies

the application as malicious or non-malicious. For well-behaved applications no action is initiated and they are allowed to execute as normal.

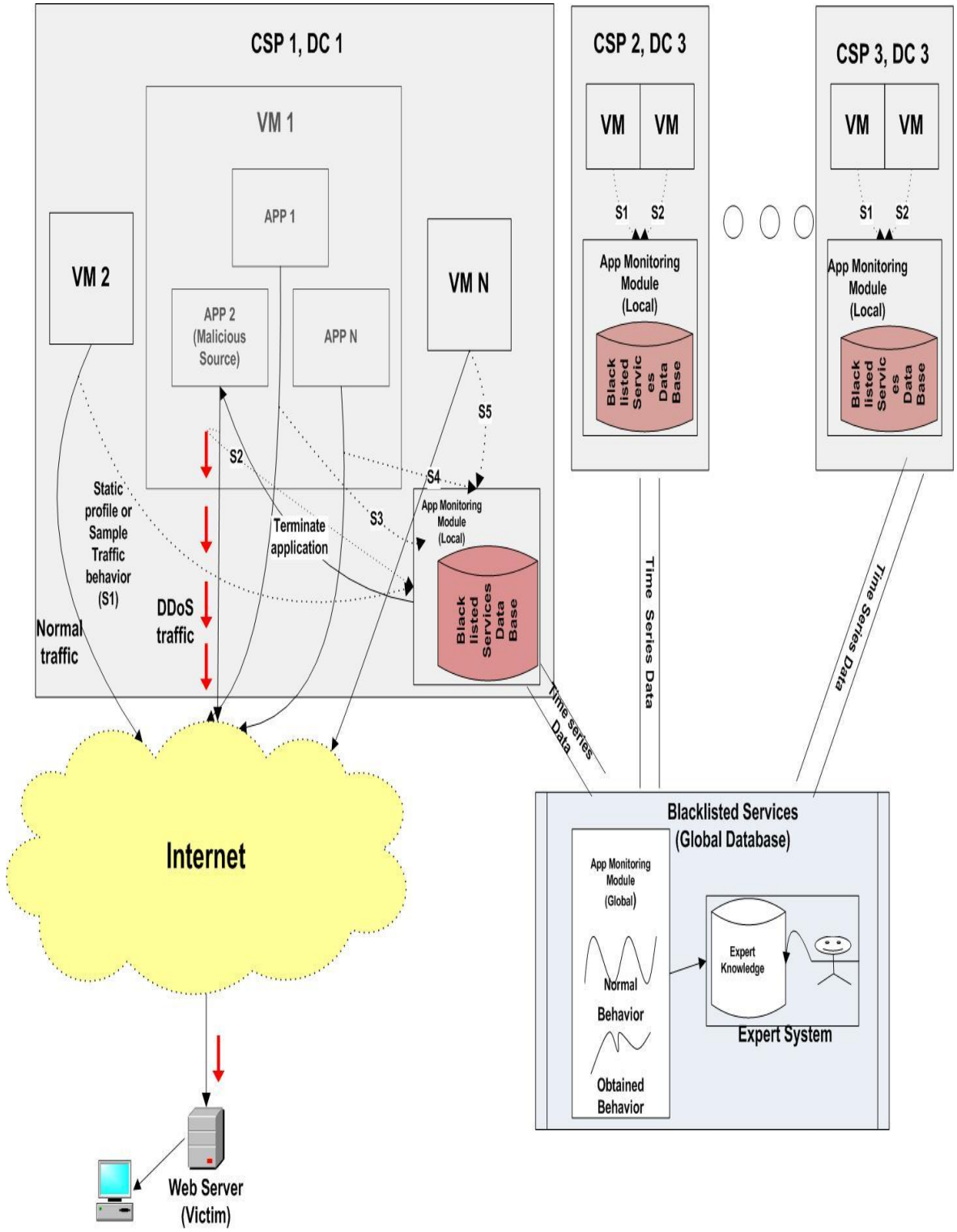


Figure. 6.1 A schematic of the proposed security mechanism in SITES

6.2 Classification of Deployed Malicious Services

Following are the sequence of steps involved for classification of services:

i. An "Application Monitor" is instantiated for every hosted application installed on an allotted VM at the CSP. Its job is to create a static profile (installed image) and dynamic profile (run-time resource usage), which is logged in a local log file and compared in real-time to the behavior profiles of known malicious applications stored in a local database. However the global database of known malicious applications is designed to create collaborative intelligence across CSPs on the behavior profiles of known malicious applications aiding in quick detection.

ii. If the static profile of the application matches, the application is prevented from execution, else it is allowed to execute.

iii. The run-time resource usage in terms of Memory and I/O usage (inbound and outbound) is stored as a 3-tuple in a time-series database such as rrdtool [150] to optimize data storage and retrieval. The data which is logged locally is also simultaneously compared with the global database storing time-series data for known malicious applications. The global database is initially seeded with the time-series data obtained from experimental analysis of available and known malicious applications and grows further as new malicious applications are detected and classified.

iv. If no matches are found, but network traffic generated exhibits anomalous behavior, the application is referred to an expert system to help classify the application as malicious or normal and the global database of malicious applications is updated for future reference if application is found to malicious.

6.3 Profiling of DDoS Services

In order to obtained the profiles of various deployed services we hired a total of nine virtual machines with different configurations i.e. three VMs each from Windows Azure [151], Amazon EC2 [152] and GoGrid [153] respectively. Further, we created our own Local Cloud (LC) using OpenStack [154] with Red Hat Enterprise and deployed three virtual machines with different configurations. Configurations of VMs used in the experiments are shown in Table 6.1. This was done to evaluate dynamic application behavior in different environments i.e. hardware configurations, platforms and under varying conditions.

Table 6.1: Configuration of VMs

CSP Name	Type of VM	Number of cores vCPU	Memory (GB)
Microsoft Azure (CSP1)	A series-A1 (VM1)	1	1.75
	A series-A2(VM2)	2	3.5
	A series-A2(VM3)	4	7
Amazon EC2 (CSP2)	m3.medium (VM1)	1	3.75
	m3.large (VM2)	2	7.5
	m3.xlarge (VM3)	4	15
GoGrid (CSP3)	small (VM1)	1	1
	medium (VM2)	2	2
	large (VM3)	4	4
Local Cloud (CSP4)	small (VM1)	1	2
	medium (VM2)	2	2
	large (VM3)	4	2

6.3.1 Services Monitoring Setup

We considered ten commonly available DoS/DDoS attack launching tools, as shown in Table 6.2, on each of the configured VM instances. These DoS/DDoS tools are capable of generating UDP, TCP SYN and HTTP flooding attacks. After installing these tools we first generated and store their MD5 hashes (checksum) by using "MD5 and SHA Checksum Utility" tool [155]. After this each of these DoS/DDoS tools are continuously monitored on parameters like CPU utilization, Memory consumption, Inbound Traffic and Outbound Traffic by perl scripts built on top of standard tools i.e. rrdtool [150] which is a database tool to work with time-series data.

Table 6.2: Tools and Attack Type

DDoS Attack Launching Tools	Attack Type
HOIC	HTTP
LOIC	(TCP, UDP, HTTP)
XOIC	TCP, UDP
Hulk	UDP
R-U-Dead-Yet	HTTP post
Tor's Hammer	HTTP post
PyLoris	HTTP, FTP
OWASP DOS HTTP POST	TCP
DAVOSET	TCP, HTTP
GoldenEye HTTP Denial Of Service Tool	TCP, HTTP

6.3.2 Observed DDoS Behaviour

After deploying and executing these DDoS tools we observed and recorded their behaviour patterns. For illustration purposes we present the detailed analysis of High Orbit Ion Cannon HOIC [156] and Low Orbit Ion Cannon (LOIC) [157].

The dynamic resource usage of HOIC for memory, CPU, outbound and inbound traffic is shown in Figures 6.2, 6.3, 6.4 and 6.5 respectively. We observe that the memory consumption trend in Figure 6.2 remains linear with average increase of 0.8 MB per second for different VMs with an observed variation of ±2%.

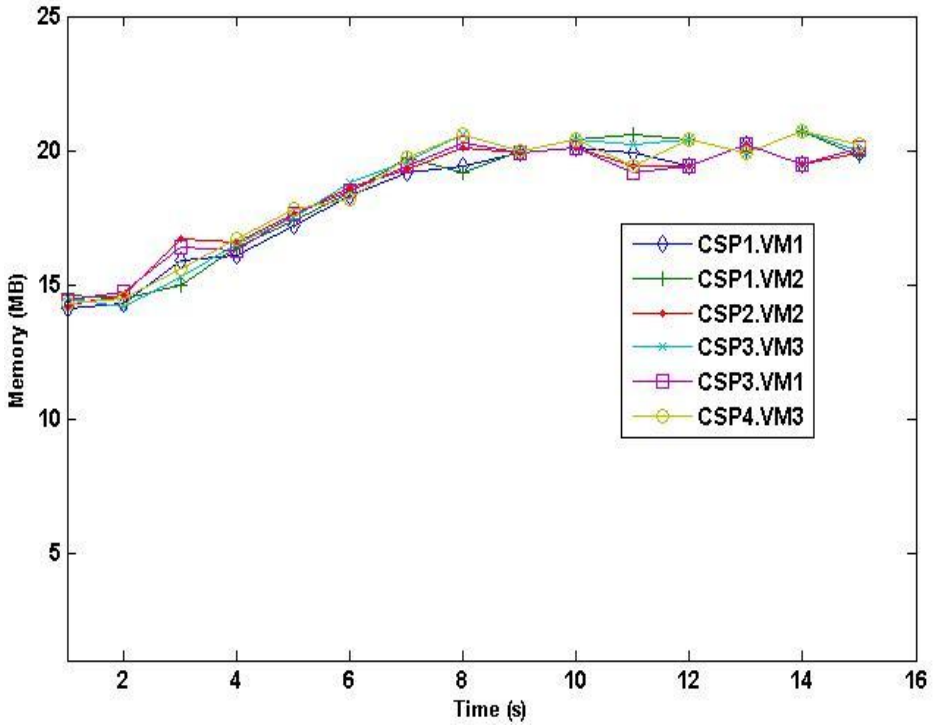


Figure 6.2: HOIC Memory Consumption

This is intuitive as any application executes same set of instructions and its memory growth can be expected to remain same if executed on similar architecture platforms. In the case of CPU utilization as shown in Figure 6.3, the trends remain the same but the percentage variation across different VMs is higher due to varying number of cores i.e. for 4 cores the average CPU utilization across CSPs is 6%, for 2 cores the utilization is around 12% and for 1 core the average CPU utilization is 17%.

Due to this reason CPU usage is not a definitive parameter to be used for dynamic profile matching.

HOIC consumes very high bandwidth as this is again expected since DoS/DDoS attacks rely on overwhelming their targets.

This is because the inbound traffic depends upon the number of participating users at that time. These trends are repeated across multiple experiments and hence bandwidth (inbound and outbound) emerges an important parameter in detecting malicious applications.

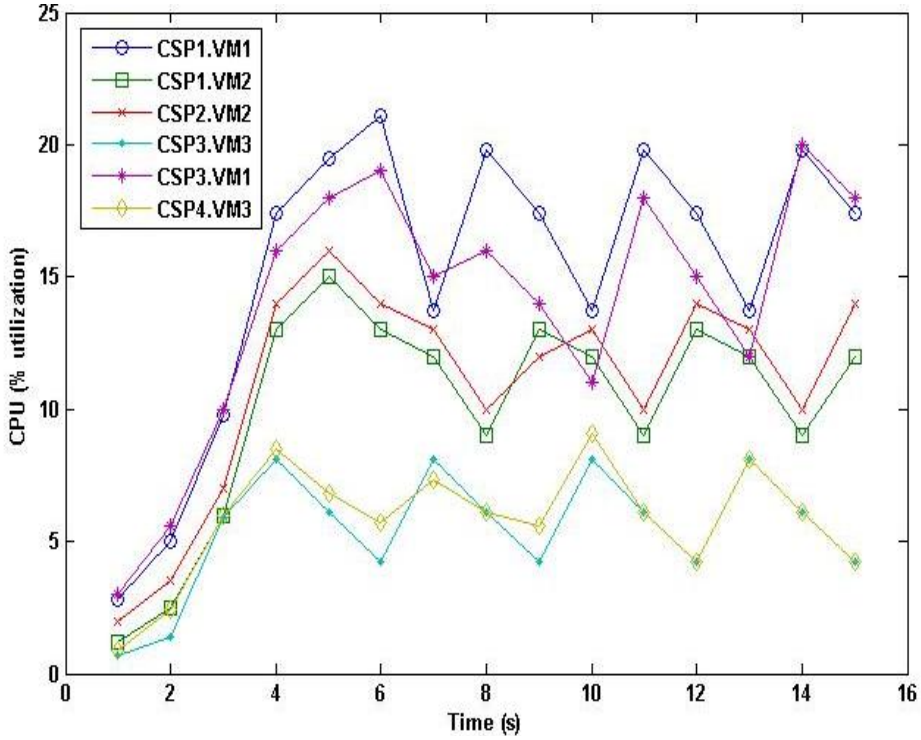


Figure 6.3: HOIC CPU Utilization

It is observed that in the outbound traffic for first three seconds the average traffic grows very high from 1200 B/s to 10000B/s and after that it comes down at the rate of 1000 B/s. The inbound traffic, during first three seconds grows exponentially i.e from 1000000 B/s to 1900000 B/s.

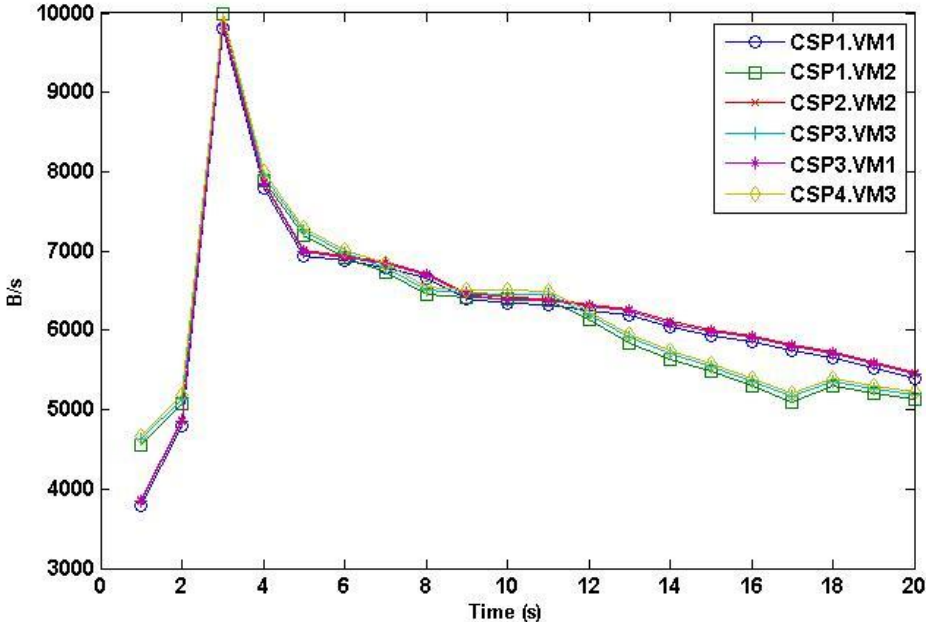


Figure 6.4: HOIC outbound traffic

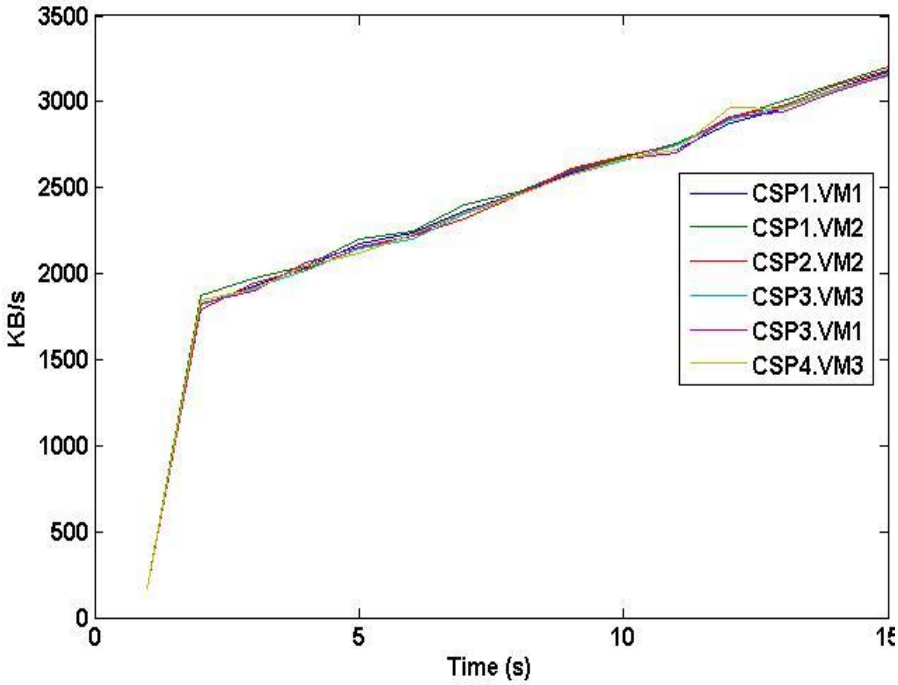


Figure 6.5: HOIC inbound traffic

The dynamic resource usage profile of LOIC (a TCP/UDP based DDoS attack tool) is presented in Figures 6.6 through 6.9. Its memory consumption trend is similar to HOIC with less consumption of memory i.e. on average 10 MB.

Further its CPU utilization varies significantly i.e. for any CSP's VM3 (4 cores) during first 7 seconds the average CPU utilization is 1.5% and it grows significantly to 15% thereafter. Further for each VM2 of any CSP (2 cores) the average CPU utilization is 3% initially, increases rapidly to 30% and then remains stable. For VM1 belonging to any CSP the average CPU utilization is 5% and it grows up to 63%.

In Figure 6.8, it has been observed that LOIC's outbound traffic is very high but predictable i.e. during first 2 seconds it reaches 500000 B/s, then in next 1 sec it shoots to 1900000 B/s with an increase of 100000 B/s for every consecutive second. In the case of inbound traffic it is observed that during first three seconds it is very high and reaches 22000 B/s. This is due to heavy requests sent to the victims' end and the responses received from the victim.

After three seconds the response decreases linearly at 500 B/s due to heavy incoming requests until it becomes unresponsive and the DDoS attack succeeds.

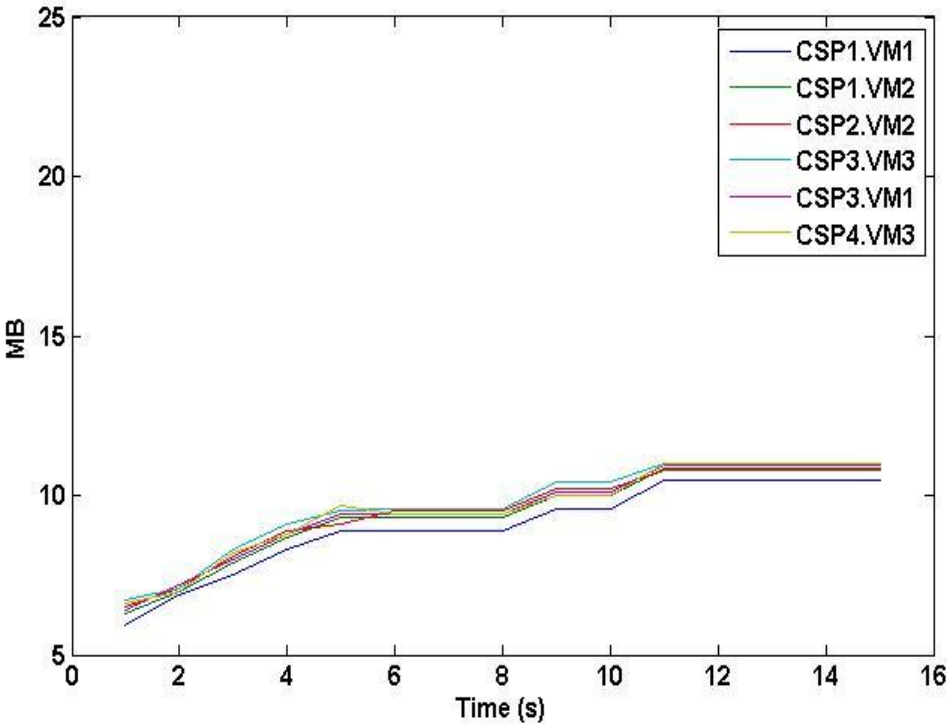


Figure 6.6: LOIC Memory

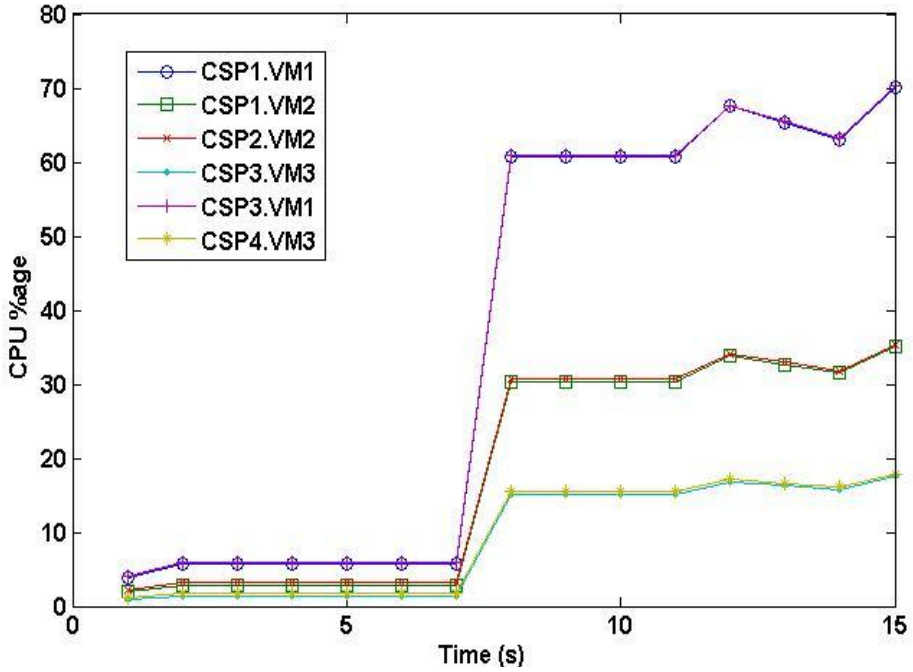


Figure 6.7: LOIC CPU Utilization

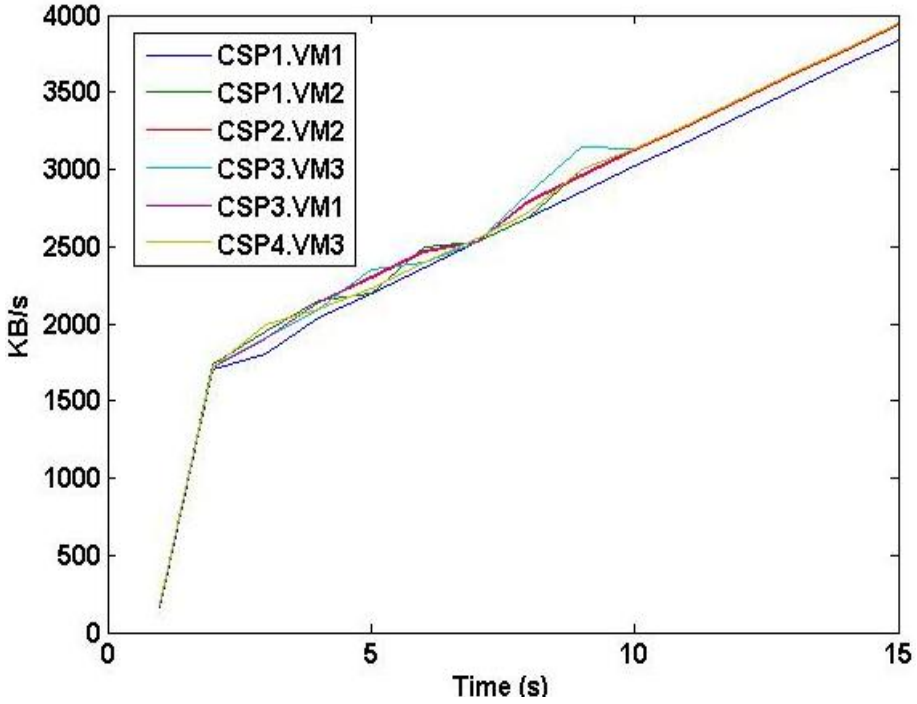


Figure 6.8: LOIC outbound

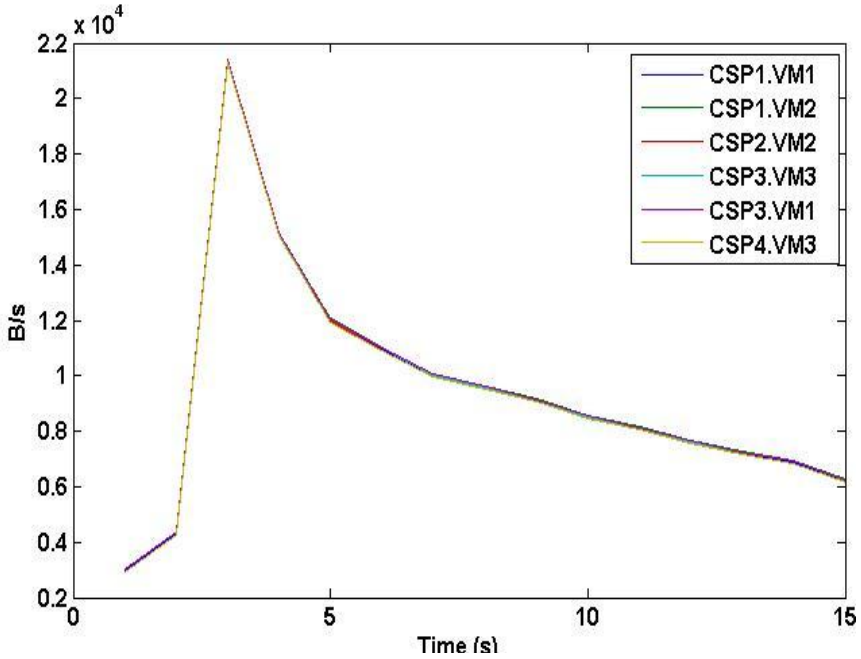


Figure 6.9: LOIC inbound

We also measured the impact of number of threads on resource usage (Figures 6.10 through 6.12) and found that if we double the number of threads, an average increase of 5% in consumption of memory and B/W is observed.

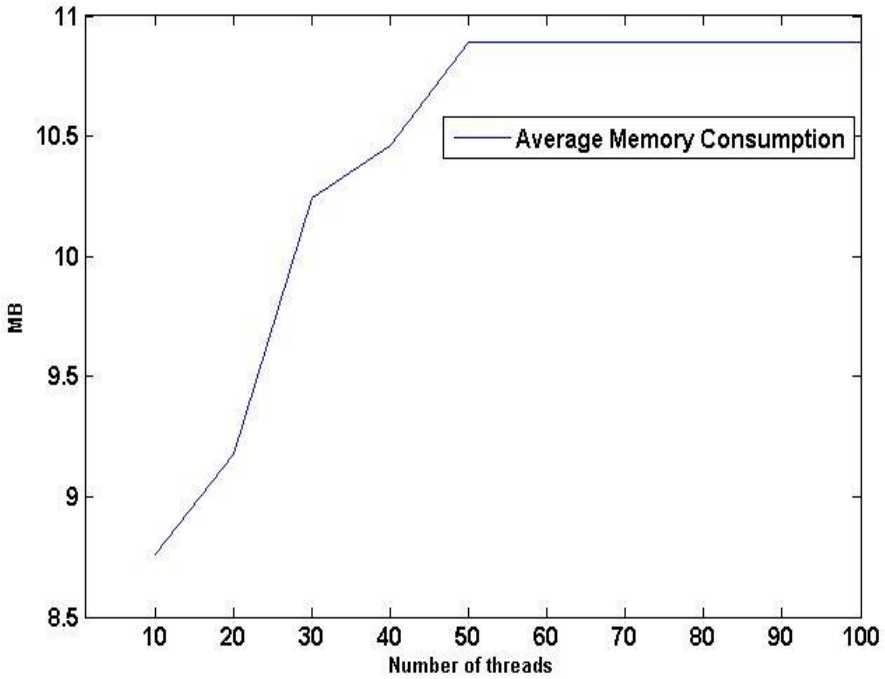


Figure 6.10: LOIC Memory consumption with variable threads

After 50 threads in each case, on average resource consumption on a VM flattens out.

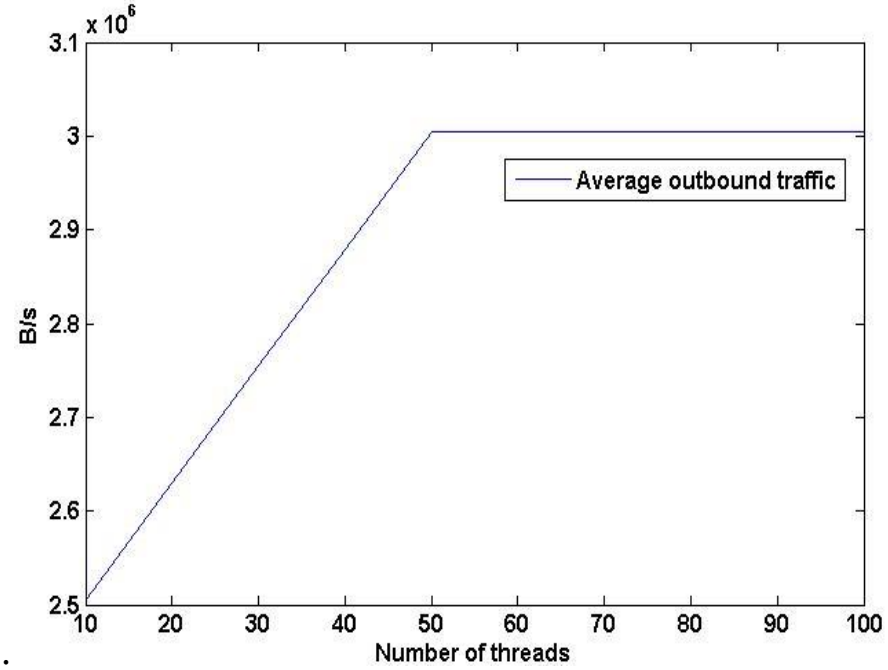


Figure 6.11: LOIC outbound traffic with variable threads

However, the trends of resource usage remain the same and serve as a strong basis for detecting malicious services

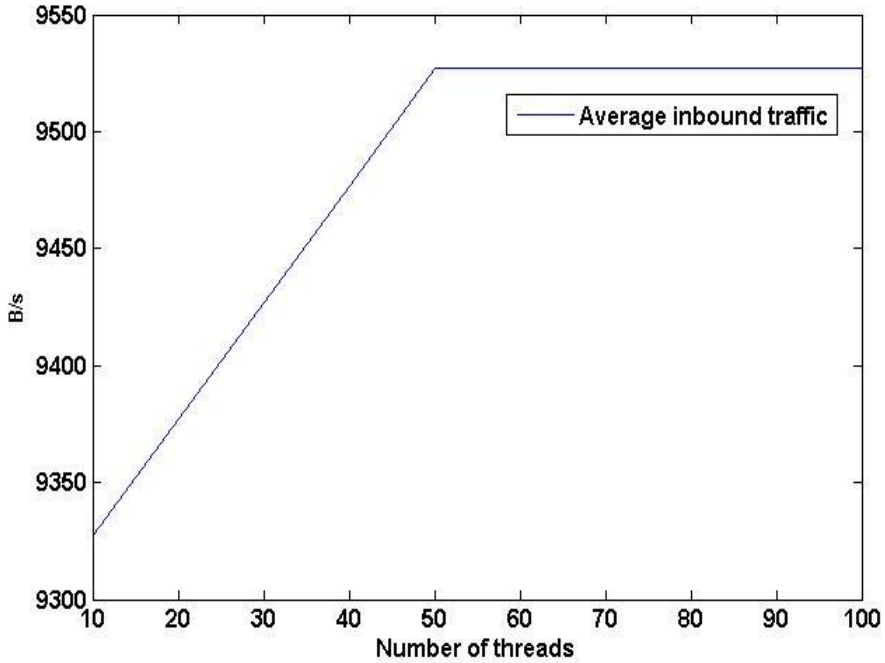


Figure 6.12: LOIC inbound traffic with variable threads

6.4 Proposed Mechanism

On the basis of the observed behaviour, the following strategy emerges for the proposed mechanism:

i. The trends of memory consumption and bandwidth usage (inbound or outbound) can be used as definitive and reliable parameters to identify a malicious application/service.CPU utilization varies significantly across CSPs and hence cannot be used reliably.

ii. Bandwidth usage (inbound or outbound) of every http, TCP/UDP based DoS/DDoS tool is very high and it grows exponentially during first five seconds hitting a long plateau later. Hence, bandwidth analysis is an effective indication of identifying malicious behavior.

iii. After executing a malicious application, a point comes when the resource usage stabilizes and flattens out after hitting a peak (this usually indicates that the target application has been overwhelmed). We term this as the Point of Confidence (PoC), where it can be safely concluded that the application is indeed malicious. For normal applications the resource usage varies and never hits a long plateau.

iv. We store the time series data using rrdtool as a 3-tuple (Memory, Inbound and Outbound traffic) representing observed values of these parameters by the Application Monitor module in the last one second interval. Each data point is stored by rrdtool as a double float (8 bytes) and hence each 3-tuple occupies 24 bytes of storage.

v. A sampling time of 1 second is determined for logging of application resource usage as time-series data. The 1 second window allows a meaningful trend to emerge and also provides enough time to log and compare data using rrdtool, which typically takes around 2 ms (for retrieval and comparison).In [40] authors observe an average latency of133ms between two geographical locations, 20000km apart which is an

extreme case. Even with this extreme latency factored in, a data matching operation against the global malicious database is expected to take an average time much lower than the 1 second window for comparing time-series data.

vi. We define "θ", the error interval as ±2% to account for variations in resource usage trends across different CSP platforms.

vii. We use the PoCas a reference point to determine the length "L" up to which values are to be stored in time series databases. This optimizes the data storage requirements further and speeds up the detection process. In our experiments a value of 5 for L was the minimum required to correctly match two time-series trends. Greater values of L would enhance the accuracy, but the delay might lead to the target application being irrevocably impacted.

viii. Similarity between two time series can be obtained by two methods: i) by measuring and comparing Euclidean distance [158]. For example in two dimensions the Euclidean distance is computed as:

$$\sqrt{\sum_{i=1}^{m}((T_{i,x} - T_{j,x})^2 + (T_{i,y} - T_{j.y})^2)}$$

where $T_{i,x}$ is the observed data point at time x and $T_{i,y}$ the observed data point at time y and ii) by comparing *absolute values* of both time series at the same timestamp. The observed average computational time to calculate and compare Euclidean sum is '2ms' and absolute value is 1ms [159]. Based on the experimental data obtained, we utilize the Euclidean Distance approach for comparing trends of bandwidth usage, while we use the absolute value comparison for comparing memory.

6.5 Implementation

We analyzed the behavior patterns of normal high bandwidth consuming applications like online gaming clients, HD TV, video (720p, 360p, 210p and

120p) uploading/downloading etc. We observed that despite of high B/W consumption per second, very stable and consistent behavior for both inbound and out bound traffic is observed. Results for online game Crazy Taxi [160] are shown in Figures 6.13 and 6.14 with an average inbound B/W usage of 64000B/s and outbound usage of 89600B/s.

Therefore, the traffic behavior pattern of DDoS is discernible against normal applications. This is linear, since the percentage increase is only 34 % per second in the case of inbound traffic and 42% in the case of outbound traffic which is very less as compared to observed DDoS tools traffic behavior i.e. on average rise 400% per second. There is an increase in first 5 seconds as well, but this increase is very low as compared to HOIC or LOIC. However after 6 seconds the traffic remains stable while in the case of DDoS tools its keeps on increasing with high volume.

To properly evaluate the effectiveness and efficiency of the proposed scheme we performed an experiment in which ten malicious DDoS tools and 20 different genuine applications were executed randomly on our cloud setup. Success of the proposed scheme is defined as the ability to correctly identify malicious applications, while false positives and negatives are recorded to evaluate its effectiveness.

Detection of false positive is given by

$$FP = \frac{Total\ number\ of\ observed\ false\ positives}{Total\ number\ of\ services\ executed\ (T)}$$

Similarly detection of false negatives is given by

$$FN = \frac{Total\ number\ of\ observed\ false\ negatives}{Total\ number\ of\ services\ executed\ (T)}$$

Also success ratio is given by

$$S_R = 1 - (FP + FN)$$

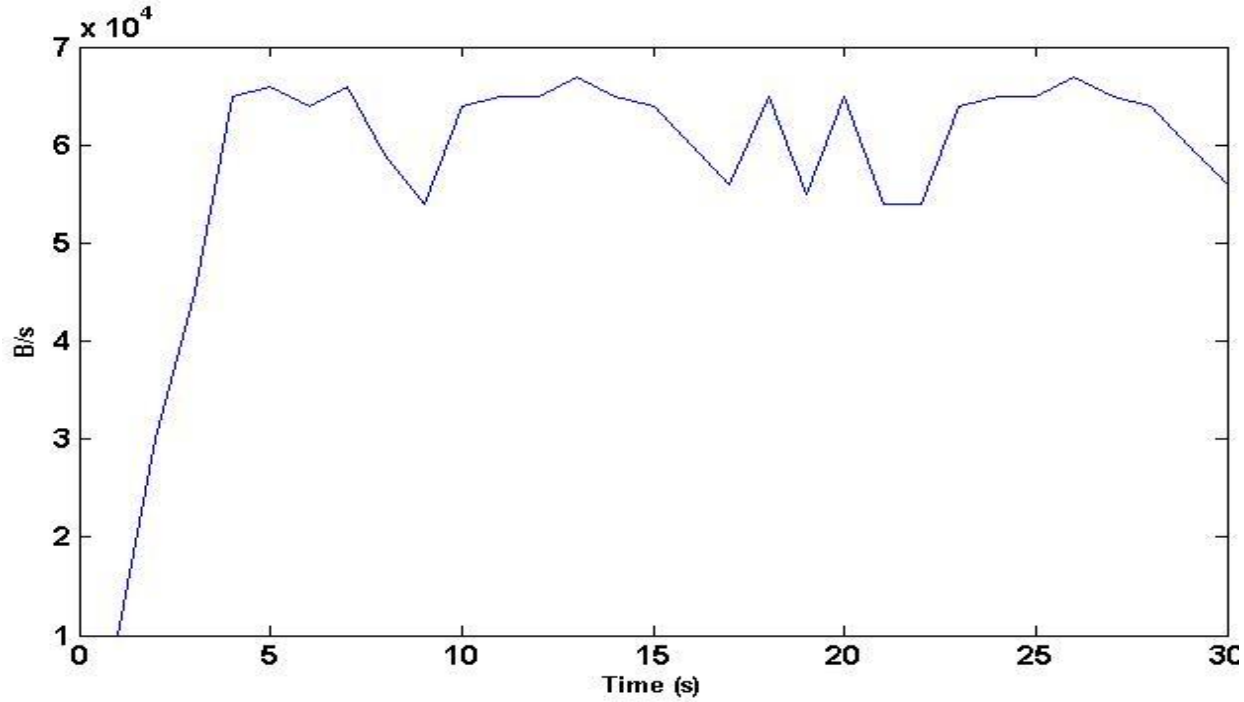


Figure 6.13: Online Game (Crazy Taxi) Inbound Traffic

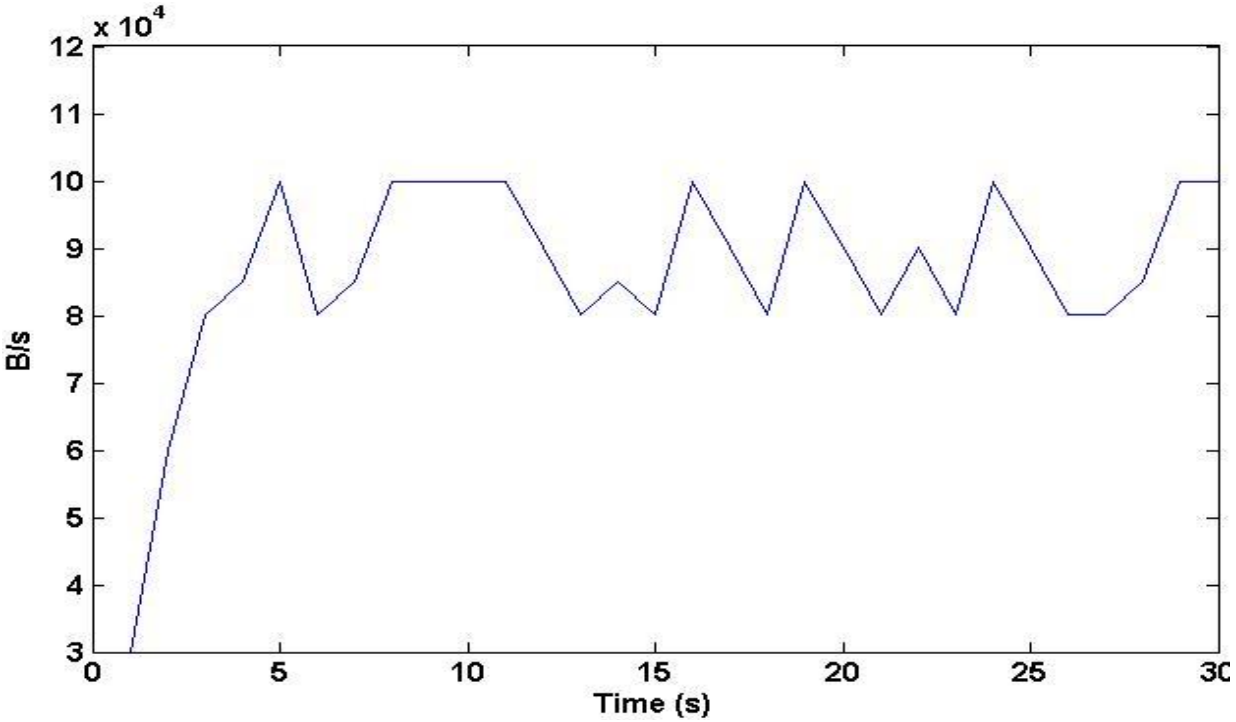


Figure 6.14: Online Game (Crazy Taxi) Outbound Traffic

We define $\theta=\pm 2$ and L=5 for all experiments, the comparison of time-series data of the executing application with the global database of known malicious applications is an iterative process with L iterations. Each successive iteration

focuses on a narrower set of matching profiles, disregarding applications where the trends of bandwidth usage and absolute values of memory usage do not match; Table 6.3 depicts the decision matrix depending upon the outcomes of the comparison process between the running application and the global database. If no match is found but service is consuming high bandwidth it is referred to the classification module which may entail human intervention.

Table 6.3: Decision Matrix

Memory-usage matches	**Outbound traffic trend matches**	**Inbound traffic trend matches**	**Traffic higher than normal**	
Yes	Yes	Yes	NA	Kill Application
No	Yes	Yes	NA	Kill Application
No	No	No	Yes	Classification Logic + Expert System
Yes	No	No	Yes	Classification Logic + Expert System
Yes	No	Yes	NA	Kill Process
Yes	Yes	No	NA	Kill Process
Yes	No	Yes	NA	Kill Process

6.5.1 Results Analysis

The decision matrix above is implemented for a total of 30 deployed applications, including 10 malicious applications. Figures 6.15 through 6.21 present the results (success ratio, false positives and negatives) of using a combination of different parameters (m) on the success ratios achieved. The best results are obtained with m = 3 i.e. memory usage, inbound traffic and outbound traffic with a success ratio of 100% achieved, while with m=2 success

ratio is on average 90%. This concludes that with the increase in the number of reliable parameters the effectiveness of the scheme increases.

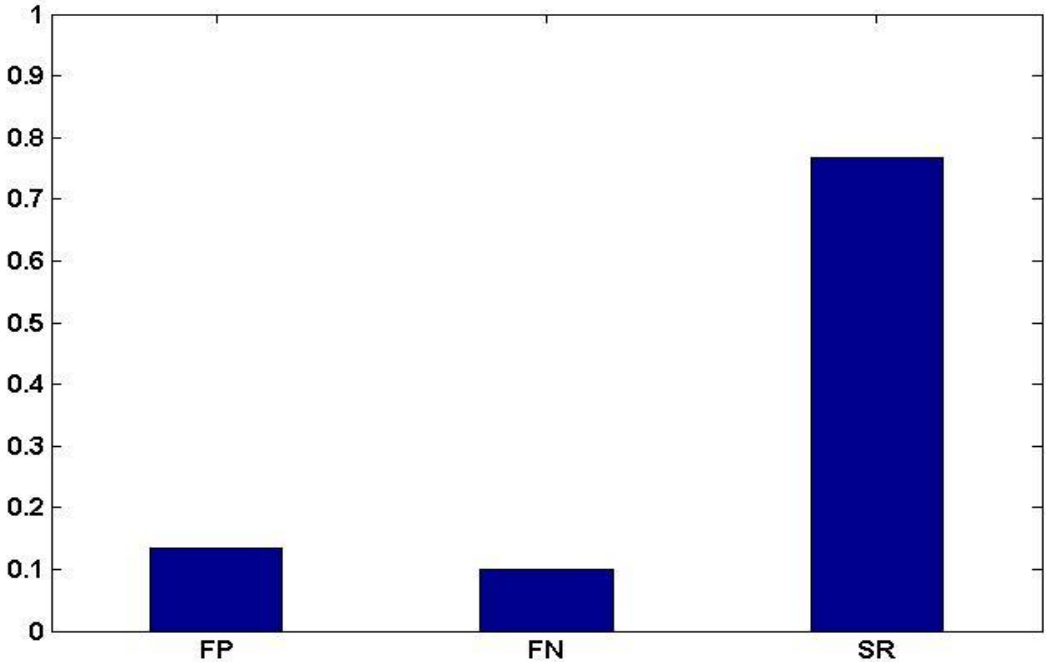


Figure 6.15: Results with m=1 (outbound traffic)

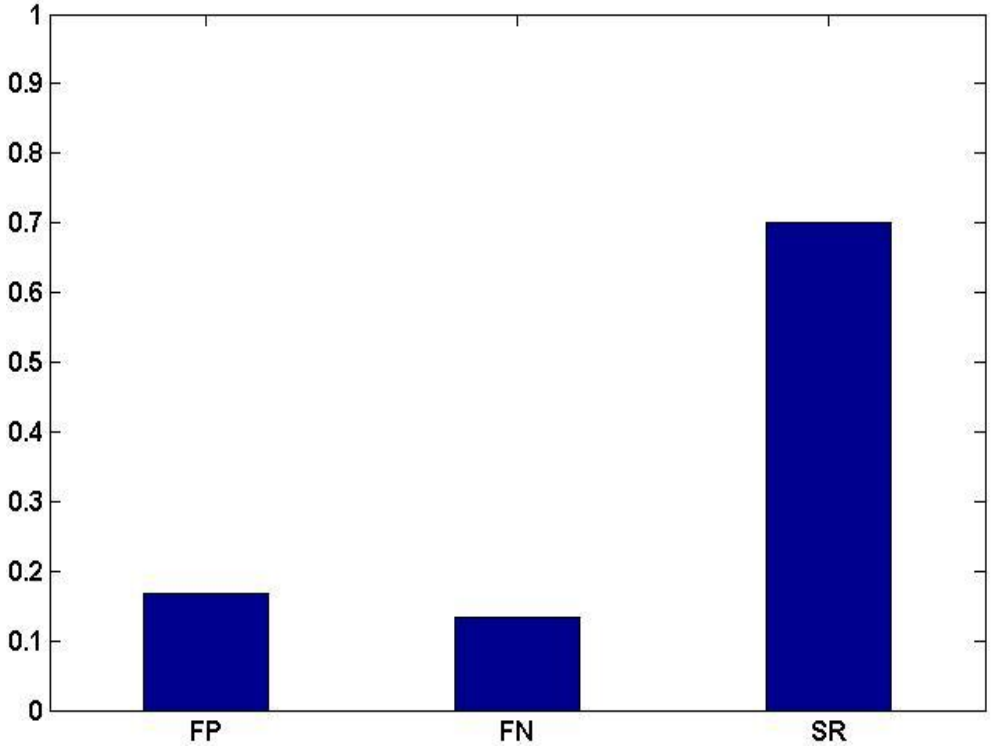


Figure 6.16: Results with m=1 (inbound traffic)

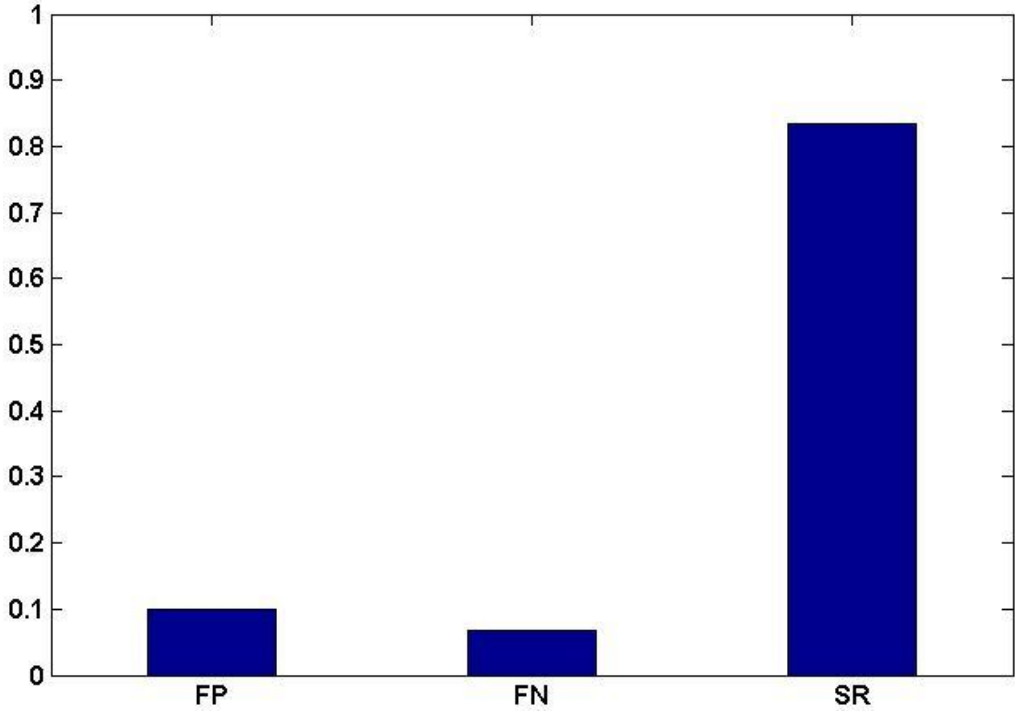


Figure 6.17: Results with m=1 (memory)

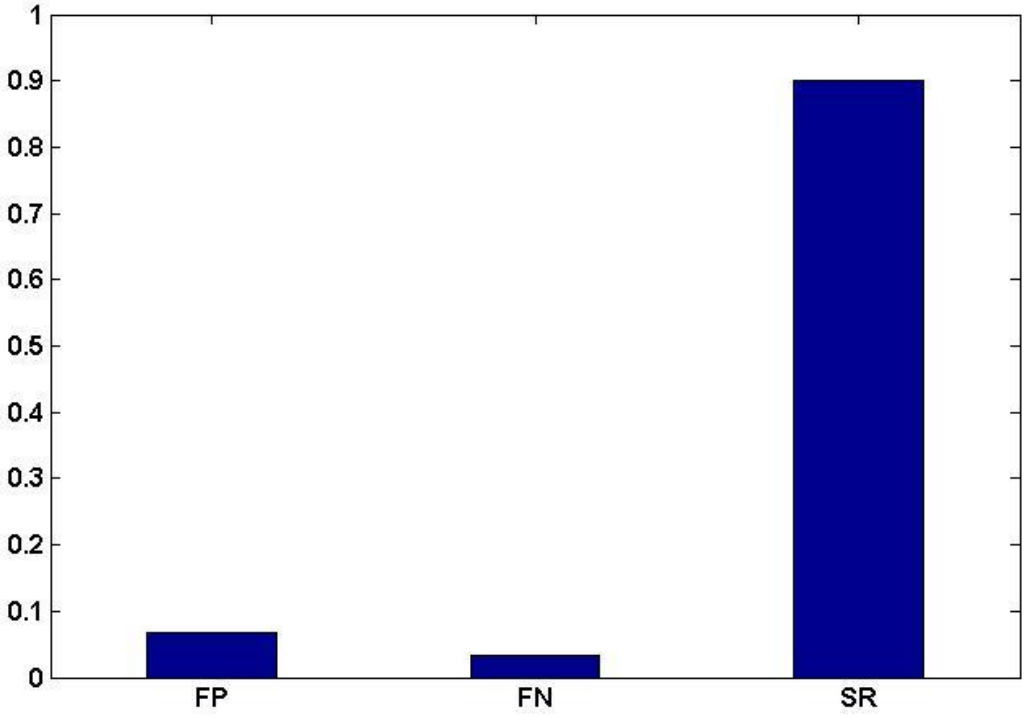


Figure 6.18: Results with m=2 (memory, inbound traffic)

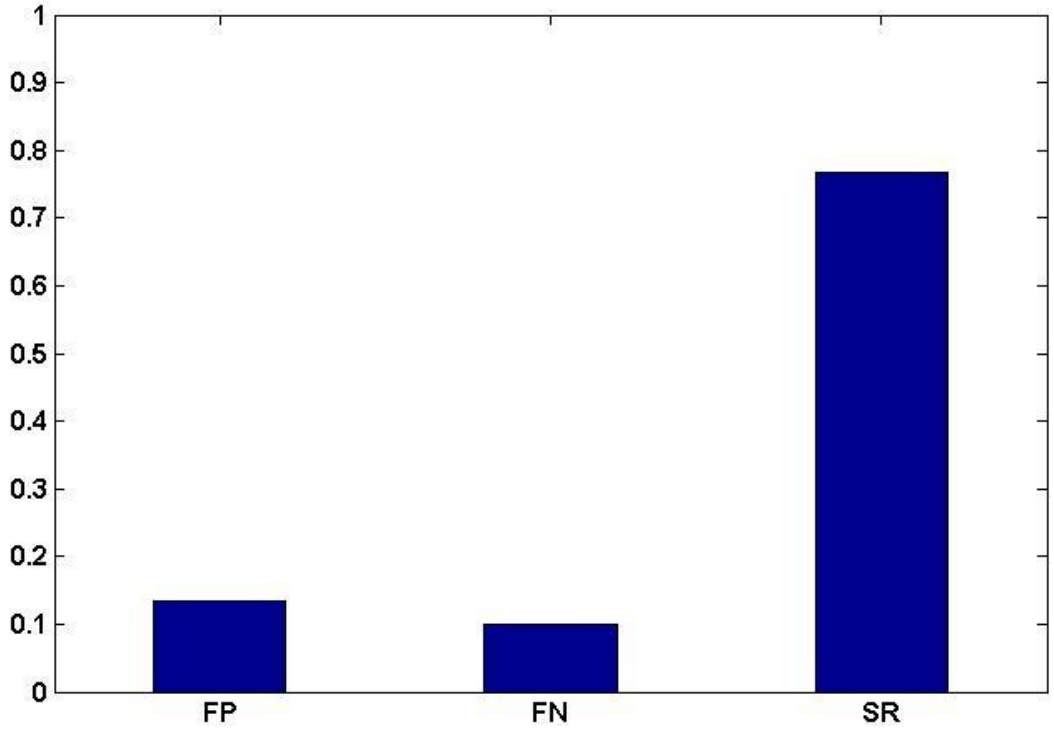


Figure 6.19: Results with m=2 (memory and outbound traffic)

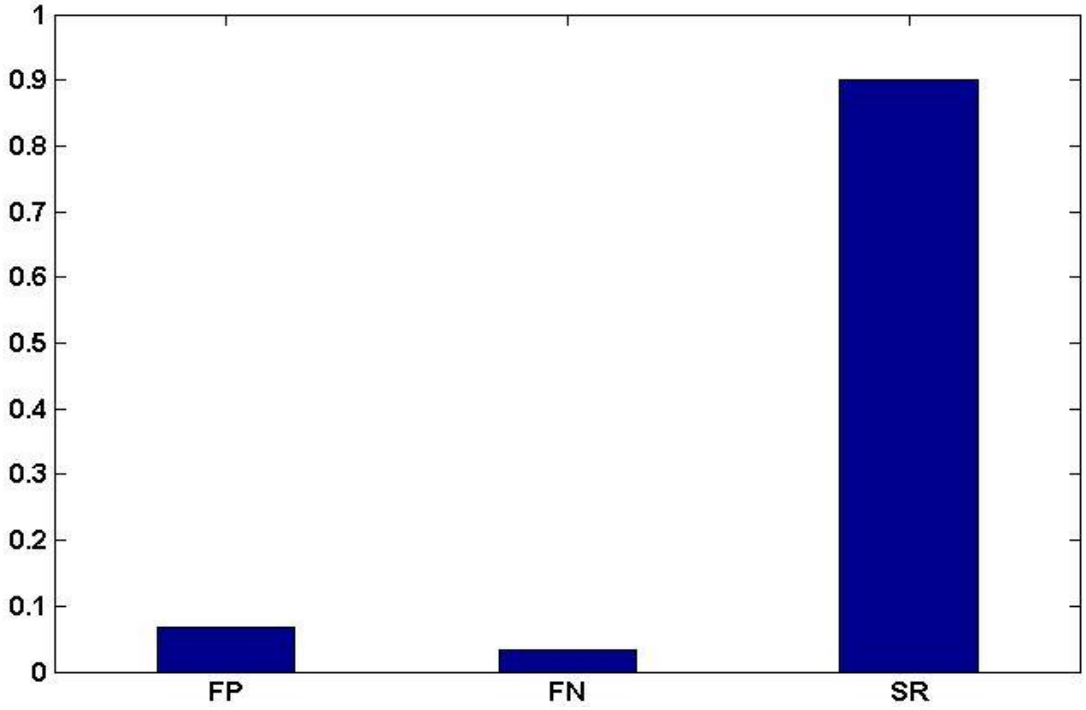


Figure 6.20: Results with m=2 (inbound and outbound traffic)

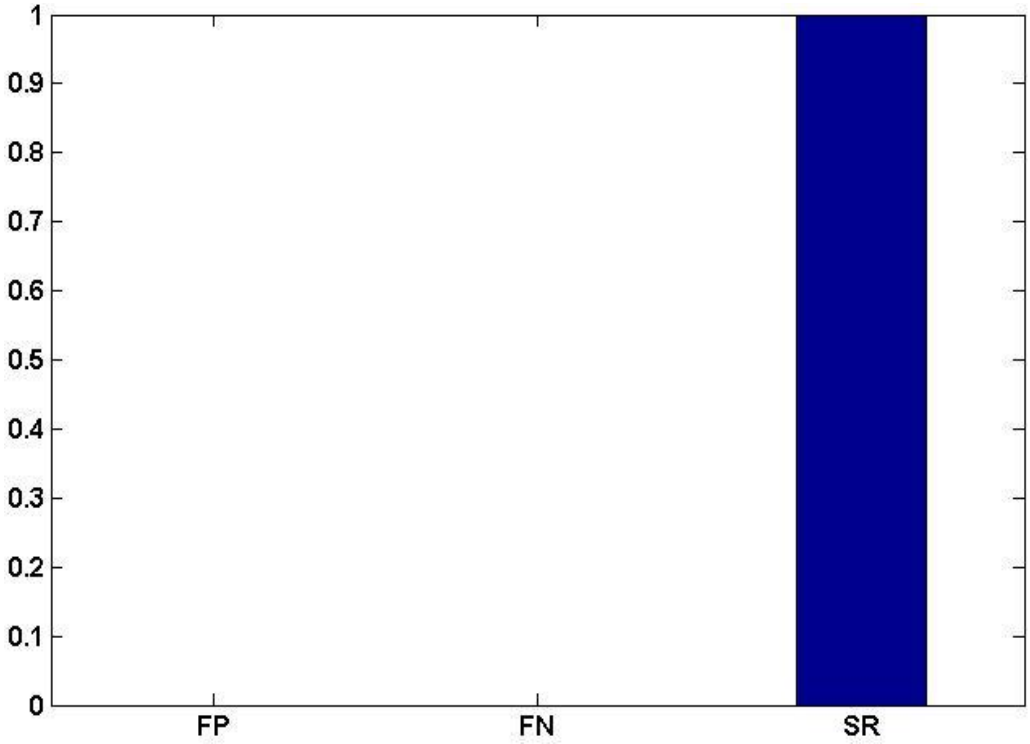


Figure 6.21: Results with m=3 (memory, inbound and outbound traffic)

Figure 6.22 and 6.23, depicts the trends of traffic for an online game application named warcraft [161] which incidentally is detected as a False Positive case by the proposed mechanism (for the case m=2 above) because its observed behaviour trend is similar to httpDoS tool (a low intensity http based DoS tool). This is due to the reason that trend remains same for the first five seconds. However if we increase the point of confidence to 8 sec (L = 8) the proposed scheme will correctly evaluate its non-malicious behaviour. But waiting for a longer duration may affect the target application irrevocably. Hence, the choice of L represents a tradeoff between early detection and accurate detection. However, when memory usage is also considered (m=3 case above) it is observed that memory consumption of these two applications is completely different i.e. for warcraft its 100 MB usage on average while for httpDoS its 6MB. Therefore, the proposed scheme successfully differentiates normal applications from malicious ones.

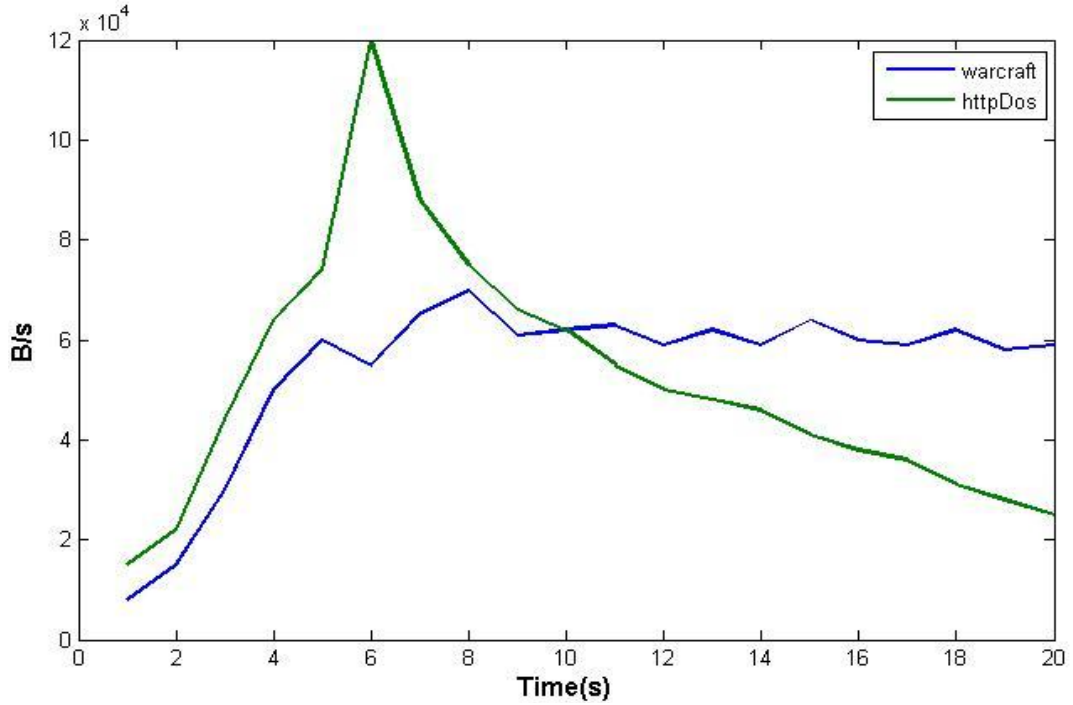


Figure 6.22: Inbound Traffic trends for warcraft (online game) and httpDos (malicious tool)

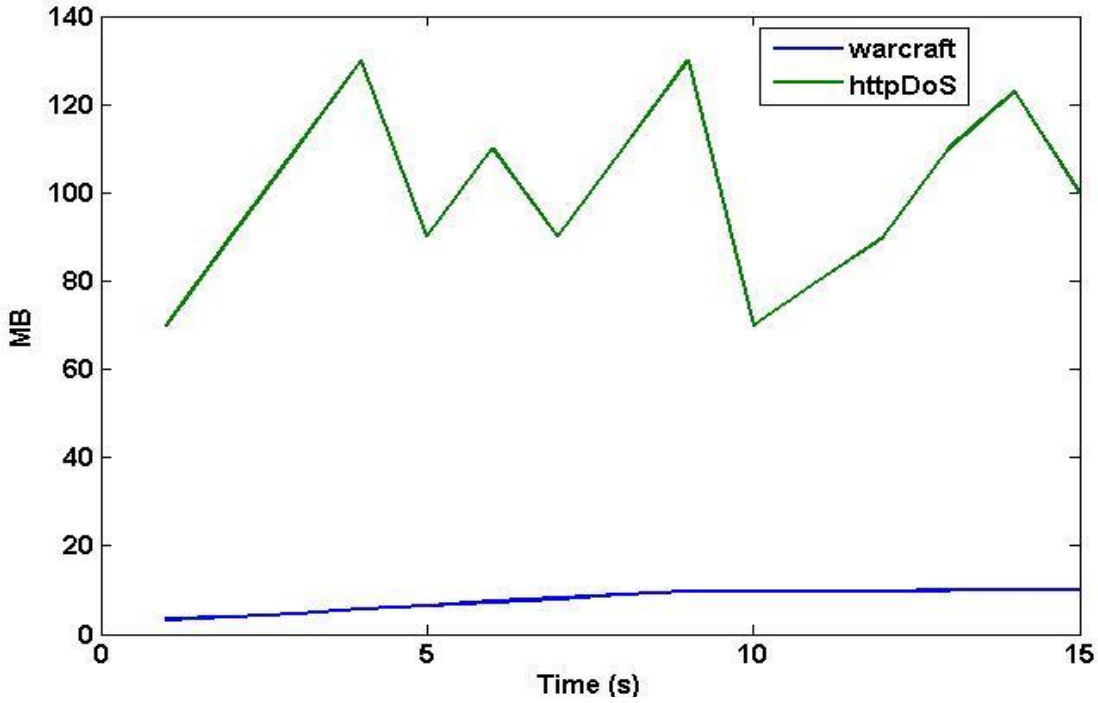


Figure 6.23: Memory Consumption trends for warcraft (online game) and httpDos (malicious tool)

The cost of finding the matches in rrdtool is O(P + 2m), where P is the total number of comparison operations between the elements of Time series 'R'and 'S' with m as the length of R and S. The cost to insert each element of R into the time series database is O(m). While this cost is $O(m^2)$ in the worst case, in the general case, P will be much less than 2 ms for common values.

The storage requirement for the time-series data for ten malicious applications in rrdtool for our experimental setup is 1200 Bytes with five data points per application, each occupying 24 bytes. As per [162] [163] and [164] there are around 30 types of known DDoS attacks many of which can be launched using tools discussed in [164]. These tools exhibit similar architecture with minor modifications; hence the behaviour or attack pattern remains largely similar. Thus, global storage requirements for a couple of hundred identified malicious applications which are capable of launching planetary scale DDoS attacks utilizing cloud infrastructure are not envisaged to exceed a few megabytes.

6.6 Findings and Observations

This chapter proposes a novel mechanism to prevent misuse of cloud infrastructure for launching DoS/DDoS attacks. It flips the traditional perspective by detecting and neutralizing DoS/DDoS applications at the source i.e. CSP rather than at the victim's end. We evaluated and observed the behaviour of existing DDoS launching tools to create detailed performance profiles of such applications. Encouraging results have been obtained by detecting malicious applications through comparison of time-series data pertaining to memory usage, inbound/outbound traffic of executing applications. Another important aspect of this mechanism is its privacy-preserving operation as it does not sniff or inspect packets like traditional methods.

Chapter 7

Conclusion and Future Scope

The chapter concludes the thesis by highlighting the main contributions of this research work. Further the chapter commences with discussing contributions of the proposed framework SITES for establishing an efficient and secure intercloud environment. Later, it presents future scope for the proposed work and intercloud environment.

7.1 Conclusion

The proposed framework SITES is found to be a step towards a Unified Services Management for an intercloud environment which facilitates a) optimal service deployment and geographically-aware auto-scaling b) efficient resource discovery and c) optimal service selection and consumption by the end-user in a seamless secure manner. Further, a customized ranking mechanism for service consumers allows greater optimization at an individual level.

It is observed that location based grouping of datacenters belonging to different CSPs in P2P manner produces very encouraging results as compared to traditional approaches.

It is also observed that mere load-based auto-scaling is not effective to deal with flash crowd scenario. Instead the geographically-aware auto-scaling which exploits latency benefits is more efficient for global services.

It was further observed that intercloud environment can be very beneficial for the attackers to launch malicious attacks like DDoS. Prevention of such type of misuse of intercloud by existing techniques is not feasible as they violate the privacy norms laid down by many CSPs. Therefore, the proposed privacy preserving security technique can be very helpful for keeping the confidence of all the intercloud stake holders.

The thesis contributes in the following ways:

a) The thesis presents detailed literature review of the work done in the area of intercloud and addresses challenges such as resource discovery, services management and security. The thesis presented SITES, a comprehensive and unified service management intercloud framework.
b) SITES proposed novel mechanisms for resource discovery. Its resource discovery mechanism provides incentive to CSPs by facilitating the process of resources trading and motivates SPs/Users to work with intercloud environment.
c) It also proposed a service management mechanism which offers service providers and consumers to leverage the intrinsic benefits of services deployed across different cloud service providers. It exploits latency and cost advantages while ensuring scalability, load-balancing, high-availability in a secure environment. It allows users' seamless access to services through an optimal service selection mechanism providing on-demand ranking on several quality parameters.
d) It also provides a novel mechanism to identify and prevents the malicious activities inside intercloud while maintaining privacy constraints.

e) The thesis demonstrates the applicability of the proposed mechanisms in intercloud paradigm, which may leverage many existing technologies and provides additional services for resource aggregation.

We believe that the realization of SITES for an intercloud environment will result in enhanced capability to deliver services to end users in a customized manner while maximizing service provider revenue at the same time. This contributes to a more efficient management of cloud services and also reduces cost for users as well as service providers through comprehensive measurement of services performance and their ranking, while maintaining a sense of security to all its stakeholders. Hence, the present work has potentially far-reaching consequences on the evolution of intercloud services.

7.2 Future Scope

There is immense scope of research in this area. The work can be further enhanced to enable intercloud users and scientists to achieve and utilize true potential of this massive infrastructure. Some of the future directions related to SITES are under:

i. For resource discovery challenges, the existing solutions considered geographical locations as a base for grouping of participated different CSPs' data centres to minimize latency. In future, introduction of a number of other performance metrics like reliability, reputation etc. can be considered to further optimize this resource discovery technique.

ii. Mechanism for management of services and its instances is presented in this thesis. Future work involves consideration of management of individual services components distributed across datacenters of SITES.

iii. The performance of framework can be further enhanced by introducing latest machine learning techniques in dynamic resource discovery, load balancing, rescheduling and other resource management challenges.

iv. This thesis has presented a mechanism to prevent misuse of SITES infrastructure to launch DDoS attacks. Intercloud also provides unprecedented storage capacity which can be used by attackers or malicious users to host and spread malwares. In future our proposed security mechanism work can be extended to prevent the abuse of such services.

The identified potential future scope for intercloud domain are:

i. Consolidation of the Federations

We foresee that the intercloud will thrive as *"Business Alliances"* centered on dominant players with deep pockets and not as a *"democratized model".* Each business alliance or federation will attract small cloud vendors which will follow their standards and protocols. This scenario can be related to todays "Andriod" and "IOS" market share. Similarly in intercloud a pure democratic model is far from reality and we envisage that only two or three federations will survive and all the new vendors will align with these federations with a limited set of hypervisors in every cloud. For interaction between clouds a pre-defined common standard would be followed by each cloud within a federation. These standards would involve a set of APIs and protocols which would be proprietary of the federation only. The democratic intercloud based on open standards seems destined to be limited to academic research and some showcase projects.

ii. P2P and Intercloud

We believe that the potential of P2P model remains underutilized in the intercloud. While Hybrid models have been proposed in literature

combining elements of centralized and P2P models, not much work has been done in this domain. The P2P model is a natural fit to requirements of large scale distributed interactions between CSPs, SPs and users, where a purely central model might lead to issues such as performance bottlenecks and single-point-of-failure. Intercloud functionality such as resource/service discovery and negotiation can be easily accomplished via P2P interactions between participating stakeholders and then the centralized agency notified for tracking and financial settlement. Moreover, the centralized agency can expose APIs for stakeholders to access trust ratings of stakeholders enabling decentralized interactions and making P2P interactions viable.

References

[1] Definition of Intercloud-Wikipedia.
http://en.wikipedia.org/wiki/Intercloud
(Last Accessed on 2016-6-15)

[2] Cisco's Intercloud future plans.
http://www.pcworld.com/article/2111300/ciscos-intercloud-could-supercharge-its-iot-plans.html
(Last Accessed on 2016-6-18).

[3] Cisco Partnership with cloud providers.
http://newsroom.cisco.com/release/1494528/cisco-adds-over-30-intercloud-partners-including-deutsc_2
(Last Accessed on 2016-6-18).

[4] A. Gupta, L. Kapoor and M. Wattal, "C2C (Cloud-to-Cloud): An Ecosystem of Cloud Service Providers for Dynamic Resource Provisioning", proceedings of Advances in Computing and Communication, Springer-Verlang, (2011), pp: 501-510.

[5] R. Buyya, R. Ranjan, R. Calheiros, "InterCloud: Scaling of Applications across multiple Cloud Computing Environments", Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing, Springer-Verlang, (2010), pp: 13-31.

[6] GICTF Use Cases and Functional Requirements for Inter-Cloud Computing.
http://www.dmtf.org/sites/default/files/20110518_ISO_JTC1_SG38_SGCC_GICTF_2.pdf
(Last Accessed on 2016-6-15)

[7] N. Grozev and R. Buyya, “Inter-Cloud architectures and application brokering: taxonomy and survey”, Software: Practice and Experience, Wiley, (2014), pp: 369–390.

[8] S. Sotiriadis, N. Bessis, N. Antonopoulos , “Towards inter-cloud schedulers: A survey of meta-scheduling approaches”, International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, (2010), pp: 59-66.

[9] A. N. Toosi, R. N. Calheiros, R. Buyya, “Interconnected Cloud Computing Environments: Challenges, Taxonomy, and Survey”, ACM Computing Surveys (CSUR), ACM, (2014), Article No. 7, DOI: 10.1145/2593512.

[10] A White Paper from the Open Cloud Standards Incubator. http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0101_1.0.0.pdf
(Last Accessed on 2016-5-12)

[11] A.J. Ferrer, F. Hernández, J. Tordsson, E . Elmroth, A .Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K . Djemame, W. Ziegler, T . Dimitrakos, S. K. Nair, G . Kousiouris, K . Konstanteli, T . Varvarigou, B . Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, C. Sheridan. “OPTIMIS: a holistic approach to cloud service provisioning.” Future Generation Computer Systems, Elsevier, (2012), pp: 66–77.

[12] Details of Cloud federation. http://searchtelecom.techtarget.com/feature/Cloud-federation-primer-The-coming-Intercloud
(Last Accessed on 2016-5-12).

[13] Working of Intercloud. http://www.networkworld.com/article/2221427/virtualization/prepare-ye-the-way-of-the-intercloud.html

(Last Accessed on 2015-6-11).

[14] Industry oriented federated Intercloud definition.
http://www.arjuna.com/what-is-federation
(Accessed 2016-5-12).

[15] Ecclyptus Project.
http://open.eucalyptus.com/
(Last Accessed on 2016-6-7).

[16] Openstack Project.
http://www.openstack.org/
(Last Accessed on 2016-6-7).

[17] R. Buyya, R. Ranjan, R. N. Calheiros. "InterCloud: utility-oriented federation of cloud computing environments for scaling of application services", In Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing. Springer-Verlag, (2010), pp: 13–31.

[18] M. Stihler, A. O. Santin, A. L. Marcon, and J. da Silva Fraga.. "Integral federated identity management for cloud computing", In New Technologies, Mobility and Security (NTMS), 5th International Conference, IEEE, (2012), pp: 1 –5.

[19] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation", in 3rd International Conference on Cloud Computing (CLOUD), IEEE, (2010), pp: 337-345.

[20] F. Tusa, A. Celesti, M. Paone, M. Villari, and A. Puliafito, "How clever-based clouds conceive horizontal and vertical federations" In Computers and Communications (ISCC), (2011), pp: 167 –172.

[21] F. Tusa, M. Paone, M. Villari, and A. Puliafito, "Clever: A cloud-enabled virtual environment", In Computers and Communications (ISCC), (2010), pp: 477-482.

[22] M. M. Hassan, M. Shamim Hossain, A. M. Jehad Sarkar, E. Nam Huh, "Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform", Inf Syst Front, Springer Science+Business Media, IEEE, (2012), pp: 523-542.

[23] https://www.cloudyn.com/blog/cloud-commodity/

[24] M. Al-Roomi, S. Al-Ebrahim, S. Buqrais and I. Ahmad, "Cloud Computing Pricing Models: A Survey", International Journal of Grid and Distributed Computing Vol.6, No.5, Elsevier, (2013), pp: 93-106.

[25] P. Saimi, A. Patel, "Review of pricing models for grid & cloud computing ", Computers & Informatics (ISCI), IEEE, (2011), pp: 634 – 639.

[26] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow. "Blueprint for the intercloud - protocols and formats for cloud computing interoperability", In Internet and Web Applications and Services, IEEE, (2009), pp: 328 –336.

[27] D. Bernstein, D. Vij, and S. Diamond, "An intercloud cloud computing economy technology, governance, and market blueprints", In SRII Global Conference (SRII), IEEE, (2011), pp: 293-299.

[28] D. Bernstein and D. Vij, "Intercloud exchanges and roots topology and trust blueprint", International Conference on Internet Computing, IEEE, (2011), pp: 286-289.

[29] Inercloud Broker.
http://www.arjuna.com/
(Last Accessed on 2016-07-01)

[30] D. Bernstein, D. Vij, "Intercloud Security Considerations", 2nd IEEE International Conference on Cloud Computing Technology and Science, (2007), IEEE, pp: 537-544.

[31] Cloud market place information website.
http://www.zimory.com/en/solutions/why-zimory/cloud-marketplace.html
(Last Accessed on 2016-06-05)

[32] Cloud Market Information for Spot cloud information.
http://www.spotcloud.com/sellers
(Accessed: 2016-07-01)

[33] T. Noor, Q. Sheng, **"**Trust as a service: a framework for trust management in cloud environments", Proceeding in 12th international conference on web information system engineering, IEEE, (2010), pp: 314-321.

[34] J. Abawajy, "Establishing Trust in Hybrid Cloud Computing Environments", 10th International Conference Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, (2011), pp: 118 – 125.

[35] Blog to explain of E-a-a-S.
http://whatis.techtarget.com/definition/encryption-as-a-service-EaaS
(Last Accessed on: 2016-06-05)

[36] M. Mosch, S. Gro and A. Schill, "User-controlled resource management in federated clouds", Journal of Cloud Computing: Advances, Systems and Applications, Springer, (2014), pp: 3-10.

[37] A . Celesti, F. Tusa, M. Villari, A. Puliafito, , "Security and Cloud Computing: InterCloud Identity Management Infrastructure", Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop, IEEE, (2010), pp: 263 – 265.

[38] Cisco's Intercloud product information.

http://www.cisco.com/c/en/us/products/cloud-systems-management/intercloud-fabric/index.html
(Last Accessed on 2016-06-05)

[39] Working group of intercloud.
http://www.opengroup.org/cloud/cloud/cloud_iop/cloud_port.htm
(Last Accessed on 2017-06-02)

[40] M. Kostoska , M. Gusev , S. Ristov , K. Kiroski , "Cloud Computing Interoperability Approaches – Possibilities and Challenges", Proceeding in BCI'12, ACM, (2012), pp: 30-34.

[41] Details of all DMTF standards.
http://dmtf.org/standards/cloud
(Last Accessed on 2016-06-05)

[42] Details of all DMTF OVF standard
http://www.dmtf.org/standards/ovf
(Last Accessed on 2016-06-05)

[43] Details of CDMI standards
http://www.snia.org/cdmi
(Last Accessed on 2016-06-05)

[44] Details of OCCI standard.
http://occi-wg.org/
(Last Accessed on 2016-06-05)

[45] Details of TOSCA.
https://www.oasis-open.org/committees/tosca/
(Last Accessed on 2016-06-05)

[46] Details of OASIS standards (CAMP).
https://www.oasis-open.org/committees/camp/
(Last Accessed on 2016-06-05)

[47] Details of all DMTF CADF standards.
http://www.dmtf.org/standards/cadf
(Last Accessed on 2016-06-05)

[48] Product for cloud integration by Rackspace.
http://www.rackspace.com/knowledge_center/article/rackspace-private-cloud-active-directory-and-ldap-integration
(Last Accessed on 2016-06-05)

[49] Identity layer
http://openid.net/connect/
(Last Accessed on 2016-06-05)

[50] NIST Standards.
http://csrc.nist.gov/groups/STM/cmvp/standards.html
(Last Accessed on 2016-06-05)

[51] R. Buyya, R. Ranjan, R. Calheiros , "InterCloud: Scaling of Applications across multiple Cloud Computing Environments" , in Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing, (2010), pp: 13-31.

[52] Nikolay, R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey", DOI: 10.1002/spe.2168, John Wiley & Sons, 2012.

[53] U. Schwiegelshohn and R. Yahyapour, , "Resource allocation and scheduling in metasystems in High-Performance Computing and Networking", Springer, (1999), pp: 851-860.

[54] GICTF White Paper.
http://www.gictf.jp/doc/GICTF_CloudIF_Protocol_WhitePaper_e_20120515.pdf
(Last Accessed on 2016-06-05)

[55] S. Sotiriadis, N. Bessis, P. Kuonen, "Advancing Inter-cloud Resource Discovery Based on Past Service Experiences of

Transient Resource Clustering", Third International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, (2012), pp: 38-45.

[56] Y. Huang, N. Bessis, P. Norrington, P. Kuonen, and B. Hirsbrunner, , "Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm", Future Generation Computer Systems, (2012), pp: 402-415.

[57] T. Nodehi, S.Ghimire, R. Jardim, "Toward a unified intercloud interoperability conceptual model for IaaS cloud service", Model-Driven Engineering and Software Development (MODELSWARD), IEEE, (2014), pp: 673-681.

[58] S. Sotiriadis, N. Bessis, N. Antonpoulos, "Decentralized Meta-brokers for Inter-Cloud: Modeling brokering coordinators for interoperable resource management", 9th International Conference on Fuzzy Systems and Knowledge Discovery IEEE, (2012), pp: 2462-2468.

[59] V. Nelson, V. Uma, "Recent Trends In Information Technology", ICRTIT, IEEE, (2012), pp: 250-254.

[60] J.L.L-Sumarro, R. Moreno, R. S. Montero, I. M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds", Future Generation Computer System, Elsevier, (2013), pp: 1431-1441.

[61] Z. Whu, H. V.Madhyastha, "Understanding latency benefits of multi-cloud Web-services deployments", ACM SIGCOMM Computer Communication Review, (2013), pp: 13-20.

[62] TM Forum Intercloud Services.
https://www.tmforum.org/InterCloudService/8480/home.html
(Last Accessed on 2016-06-05)

[63] Blog on Managed Services.
http://www.sbtpartners.com/_etc/managed_services_paradigm.pdf
(Last Accessed on 2016-06-05)

[64] RightScale Project.
URL: http://www.rightscale.com/
(Last Accessed on 2016-06-03)

[65] Zimory Project.
http://www.zimory.com/
(Last Accessed on 2016-05-05)

[66] Aelous Projects
http://www.aeolusproject.org/
(Last Accessed on 2016-06-03)

[67] Delta Cloud Project.
http://deltacloud.apache.org/index.html
(Last Accessed on 2016-04-01)

[68] Rackspace monitoring.
http://www.rackspace.com/cloud/monitoring/
(Last Accessed on 2016-06-05)

[69] Cohesiveft Project.
http://www.cohesiveft.com/
(Last Accessed on 2016-06-10)

[70] R. N. Calheiros, A. N. Toosi, Christian Vecchiola, Rajkumar Buyya, “A coordinator for scaling elastic applications across multiple clouds”, Future Generation Computer Systems (2012), pp: 1350–1362.

[71] W. Itani, C. Ghali, R. Bassil, A. Kayssi, A. Chehabb, “ServBGP: BGP-inspired autonomic service routing for multi-provider collaborative architectures in the cloud”, Future Generation Computer Systems, Elsevier, pp: 99–117.

[72] "S. K. Garg, S. Versteegb, R. Buyya"A framework for ranking of cloud computing services", Future Generation Computer Systems, Elsevier, (2013), pp: 1012–1023.

[73] A. Bakshi, Y. B. Dujodwala, "Securing cloud from DDoS attacks using intrusion detection system in virtual machine", In proceedings of the second international conference on communication software and networks, ICCSN'10, IEEE, (2010), pp: 260–264.

[74] M. Kretzschmar, M. Golling, "Security management spectrum in future multi-provider Inter-Cloud environments Method to highlight necessary further development, Systems and Virtualization Management (SVM)", 5th International DMTF Academic Alliance Workshop, IEEE, (2011), pp: 1 – 8.

[75] C. Mazzariello, R. Bifulco and R. Canonico, "Integrating a Network IDS into an Open Source Cloud Computing Environment", Sixth International Conference on Information Assurance and Security, USA, IEEE, (2010), pp: 265-270.

[76] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus open-source cloud-computing system", in Proceedings of the 9th International Symposium on Cluster Computing and the Grid (CCGRID '09), IEEE/ACM, (2009), pp: 124–131.

[77] HOIC: http://sourceforge.net/projects/highorbitioncannon/

[78] C. Mazzariello, R. Bifulco and R. Canonico, "Integrating a Network IDS into an Open Source Cloud Computing Environment", Sixth International Conference on Information Assurance and Security, USA, IEEE, (2010), pp: 265-270.

[79] Rackspace:
http://www.rackspace.com/managed_hosting/services/security/ddos mitigation/
(Last Accessed on 2016-06-05)

[80] Cloud Flare for DDoS.
http://www.cloudflare.com/ddos
(Last Accessed on 2016-06-05)

[81] Prolexic for DoS.
http://www.prolexic.com/why-prolexic-best-dos-and-ddos-scrubbing-centers.html
(Last Accessed on 2016-06-05)

[82] L.Yang, T. Zhang, J. Song, JinShuangWang, Ping Chen, "Defense of DDoS Attack for Cloud Computing", International conference on Computer Science and Automation Engineering, IEEE, (2012), pp: 626-629.

[83] David, Vij, Deepak, "Intercloud Security Considerations", Cloud Computing Technology and Science (CloudCom) International Conference, IEEE, (2010), pp: 537 – 544.

[84] L. Jian-Xin, B.Li, Z. Du, L.Meng, "CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration", Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 11th ACIS International Conference, IEEE, (2010), pp: 181 – 186.

[85] M. Kretzschmar, M. Golling, "Security management spectrum in future multi-provider Inter-Cloud environments Method to highlight necessary further development, Systems and Virtualization Management (SVM)", 5th International DMTF Academic Alliance Workshop, IEEE, (2011), pp: 1 – 8.

[86] M. G. Jaatun, G. Zhao, and C. Rong , "Identity-Based Authentication for Cloud Computing", LNCS 5931, Springer-Verlag, (2009), pp. 157-166.

[87] CNET Blog.
http://news.cnet.com/8301-19413_3-10133487-240.html
(Last Accessed on 2016-02-04)

[88] Narottam Chand, Ramesh Joshi, Manoj Misra,"Supporting cooperative caching in mobile ad hoc networks using clusters", International Journal of Ad Hoc and Ubiquitous Computing, ACM, (2006), pp: 58-72.

[89] David, Vij, Deepak, "Intercloud Security Considerations", Cloud Computing Technology and Science (CloudCom), second International Conference, IEEE, (2010), pp: 537 – 544.

[90] L. Jian-Xin, Bo Li, Du Zongxia, L. Meng, "CloudVO: Building a Secure Virtual Organization for Multiple Clouds Collaboration", Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), IEEE, (2010), pp: 181 – 186.

[91] Amazon Web Services LLC.
http://aws.amazon.com/ec2/
(Last Accessed on 2016-06-05)

[92] Rackspace Cloud.
http://www.rackspace.com/cloud/
(Last Accessed on 2016-06-05)

[93] RightScale website.
http://www.rightscale.com/
(Last Accessed on 2016-06-05)

[94] Eucalyptus cloud.

http://www.eucalyptus.com/
(Last Accessed on 2016-03-02)

[95] R. Buyya, D. Abramson, J. Giddy, H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing", Concurrency and Computation: Practice and Experience, Wiley and Sons, (2002), pp:1507–1542.

[96] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I.Brandic, "Cloud computing and emerging it platforms:Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Elsevier, (2009), pp: 599–616.

[97] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition", SIGCOMM Computer Communication Review, IEEE, (2008), pp: 50–55.

[98] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K.Pohl, "A journey to highly dynamic, self-adaptive service based applications", Automated Software Engineering, Springer-verlang, (2008), pp: 313–341.

[99] R. Ranjan, L. Zhao, X. Wu, and A. Liu, "Peer to peer cloud provisioning: Service Discovery and load balancing", Proceedings of international conference cloud computing and networks, Springer Verlang, (2010), pp: 195-217.

[100] S. Sotiriadis, "Towards Inter - Cloud Schedulers: A Survey of Meta-Scheduling Approaches", proceedings of 3PGCIC , IEEE , (2013), pp: 59-66.

[101] S. Sotiriadis, N. Bessis, P. Kuonen, "Advancing inter-cloud resource discovery based on past service experiences of transient resource clustering", Third International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, (2012), pp: 38-45.

[102] Mosaic Cloud.
http://www.mosaic-cloud.eu
(Last Accessed on 2016-01-01)

[103] J. J. Bahaman, R. Buyya ,"Failure aware resource provisioning for hybrid cloud infrastructure", Journal of Parallel and Distributed Computing, Elsevier, 2012, pp: 1318-1331,

[104] F. Xhafa, and A. Abraham, "Computational Models and heuristic methods for grid scheduling problems", In proceedings of Future Generation Computer System", Elsevier, (2010), pp: 608-621.

[105] V. C. Emeakorah, M. A. S. Netto, R.N. Calheiros, I. Brandic, R. Buyya, C. A. F. De Rose, "Towards Autonomic detection of SLA violations in Cloud infrastructures", Future Generation Computer System, Elsevier, (2012), pp: 1017-1029.

[106] DMTF-Interoperable Clouds: http://www.dmtf.org/sites/default/files/standards/documents/DSP-IS0101_1.0.0.pdf

[107] OMG: Cloud Interoperability Roadmaps Session, December 2009. Available at: http://www.omg.org/news/meetings/tc/ca/special-events/Cloud_Interop_Roadmaps.htm

[108] OGF, SNIA, ETSI.
http://www.cloudplugfest.org
(Last Accessed on 2016-03-05)

[109] IEEE 2301 Standard.
http://standards.ieee.org/develop/project/2301.html
(Last Accessed on 2016-04-05)

[110] IEEE 2302 Standard.
http://standards.ieee.org/develop/project/2302.html
(Last Accessed on 2016-02-05)

[111] CloudForum
http://www.cloudforum.org/

(Last Accessed on 2016-06-05)

[112] C. A. Lee, ”An open cloud computing interface status update (and roadmap dartboard) Cloud interoperability roadmaps sessions”, In: OMG Technical Meeting (2009)

[113] DMTF Standards
http://www.dmtf.org/standards/

[114] R. Buyya, R. Ranjan, R. Calheiros, “InterCloud: Scaling of Applications across multiple Cloud Computing Environments”, Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing, Springer-Verlang, (2010), pp: 13-31.

[115] *Abramowicz,* “Portability and Interoperability between Clouds: Challenges and Case Study”, LNCS, Springer-Verlang, (2011), pp: 62-74.

[116] P. Riteau, “Building Dynamic Computing Infrastructures over Distributed Clouds “First international symposium Network Cloud Computing and Applications (NCCA), IEEE, (2011), pp: 27 – 130.

[117] Sail Project.
http://www.sail-project.eu/
(Last Accessed on 2016-06-05)

[118] Narottam Chand, Ramesh Joshi, Manoj Misra, “Energy Efficient Cache Invalidation in a Disconnected Mobile Environment”, Distributed Computing and Internet Technology , LNCS, Springer, pp: 85-95

[119] Sail Project Documents.
http://www.sail-project.eu/wp-content/uploads/2012/06/D-D.1v2.0-final-public.pdf
(Last Accessed on 2016-06-05)

[120] J. J. Bahaman, R. Buyya , "Failure aware resource provisioning for hybrid cloud infrastructure", Journal of Parallel and Distributed Computing, 2012, Elsevier, pp: 1318-1331.

[121] S. Dowell, A. Barreto III, J. B. Michael and M. T. Shing, "Cloud to cloud interoperability", 6th International Conference on System of Systems Engineering (SoSE), IEEE**,** (2010), pp: 258 – 263.

[122] D. Petcu, G. Macariu, S. Panica, C. Craciun, "Portable Cloud applications—From theory to practice", Future Generation Computer Systems, Elsevier, (2013), pp: 1417–1430.

[123] C. Krintz, The AppScale Cloud Platform: Enabling Portable, Scalable Web Application deployment", Internet Computing, IEEE, (2013), pp: 72 – 75.

[124] J.L.L-Sumarro, R.Moreno, Ruben S.Montero, Ignacio M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds", Future Generation Computer System, Elsevier, (2013), pp: 1431-1441.

[125] Z. Whu, H. V.Madhyastha, "Understanding latency benefits of multi-cloud Web-services deployments", ACM SIGCOMM Computer Communication Review, ACM, (2013), pp: 13-20.

[126] Devstack.
http://docs.openstack.org/developer/devstack/
(Last Accessed on 2016-06-05)

[127] Juxtapose
https://jxta.kenai.com/
(Last Accessed on 2016-06-05)

[128] Network Latency.
http://ipnetwork.bgtmo.ip.att.net/pws/network_delay.html
(Last Accessed on 2016-06-05)

[129] CloudSim.
http://www.cloudbus.org/cloudsim/
(Last Accessed on 2015-07-03)

[130] Services Protocol fomat description website.
https://marinemetadata.org/conventions/services-protocols-formats
(Last Accessed on 2016-05-01)

[131] W3.
http://www.w3.org/TR/wsdl
(Last Accessed on 2016-06-05)

[132] TOSCA
https://www.oasis-open.org/committees/tosca/faq.php
(Last Accessed on 2016-06-05)

[133] TM Forum
https://www.tmforum.org/InterCloudService/8480/home.html
(Last Accessed on 2016-06-05)

[134] SMI.
http://www.cloudcommons.com/about-smi
(Last Accessed on 2016-06-05)

[135] Nidhi Bansal, T.P. Sharma, Manoj Misra, R.C. Joshi, "FTEP: A Fault Tolerant Election Protocol for Multi-level Clustering in Homogeneous Wireless Sensor Networks,"in the proceedings of 16th IEEE International Conference on Networking (ICON 2008), IEEE, (2008), pp:1-6.

[136] R. Wishart, R. Robinson, J. Indulska and A. Josang, "SuperstringRep: Reputation-enhanced Service Discovery". In proceeding of the 28th Australasian conf. on Computer Science, IEEE, (2012), pp: 49-57.

[137] Amazon EC2.

http://aws.amazon.com/ec2/
(Last Accessed on 2016-06-05)

[138] Windows Azure
http://www.windowsazure.com/en-us/
(Last Accessed on 2016-06-05)

[139] GoGrid.
http://www.gogrid.com/
(Last Accessed on 2016-06-05)

[140] J.L.L-Sumarro, R. Moreno, R. S. Montero, I. M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds", Future Generation Computer System, Elsevier, (2013), pp: 1431-1441.

[141] Outbound Ping on Azure VM.
http://social.msdn.microsoft.com/Forums/windowsazure/en-US/e9e53e84-a978-46f5-a657-f31da7e4bbe1/icmp-outbound-ping-on-azure-vm?forum=WAVirtualMachinesforWindows
(Last Accessed on 2016-06-05)

[142] Z. Whu, H. V.Madhyastha, "Understanding latency benefits of multi-cloud Web-services deployments", ACM SIGCOMM Computer Communication Review, ACM, (2013), pp: 13-20.

[143] ManageEngine App.
http://www.manageengine.com/products/applications_manager/
(Last Accessed on 2016-06-05)

[144] HTTPPERF.
http://www.hpl.hp.com/research/linux/httperf/
(Last Accessed on 2016-06-05)

[145] Z. Whu, H. V.Madhyastha, "Understanding latency benefits of multi-cloud Web-services deployments", ACM SIGCOMM Computer Communication Review, ACM, (2013), pp: 13-20.

[146] C. B. Lee and A. Snavely. “On the user-scheduler dialogue: Studies of user-provided runtime estimates and utility functions.” International Journal of High Performance Computer Applications, (2004), pp:495–506

[147] V. C. Emeakaroha, T. C. Ferreto, “CASViD: Application Level Monitoring for SLA Violation Detection in Clouds”, COMPSAC, IEEE, (2012), pp. 499-508.

[148] J.L.L-Sumarro, R. Moreno, R. S.Montero, I. M. Llorente, “Scheduling strategies for optimal service deployment across multiple clouds”, Future Generation Computer System (2013), pp. 1431-1441.

[149] Attack-as-a-Service.
http://cloudtimes.org/2013/06/22/attack-as-a-service-criminals-in-the-cloud/
(Last Accessed on 2016-06-05)

[150] RRD Tool.
http://oss.oetiker.ch/rrdtool/
(Last Accessed on 2016-06-05)

[151] Azure VM instance
http://azure.microsoft.com/en-in/
(Last Accessed on 2016-06-05)

[152] Amazon Instances.
https://aws.amazon.com/ec2/instance-types/

[153] GoGrid,
http://www.gogrid.com
(Last Accessed on 2016-06-05)

[154] OpenStack Project

https://www.openstack.org/software/project-navigator/https://raylin.wordpress.com/downloads/md5-sha-1-checksum-utility/
(Last Accessed on 2016-06-05)

[155] DDoS Tool HOIC.
http://sourceforge.net/projects/highorbitioncannon/
(Last Accessed on 2016-06-05)

[156] DDoS Tool LOIC.
http://sourceforge.net/projects/loic/
(Last Accessed on 2016-06-05)

[157] Eucledian Distance.
http://www.cut-the-knot.org/do_you_know/far_near.shtml#euclidean
(Last Accessed on 2016-06-05)

[158] L. Deri, S. Mainardi1, and F. Fusco, "TSDB: A Compressed Database for Time Series", LNCS, Springer, (2012), pp: 143–156.

[159] Crazy Taxi Game.
http://crazy-taxi.en.softonic.com/
(Last Accessed on 2016-06-05)

[160] Warcraft online game
http://world-of-warcraft.en.softonic.com
(Last Accessed on 2016-06-05)

[161] DDoS Attack Types.
https://www.rivalhost.com/blog/12-types-of-ddos-attacks-used-by-hackers/
(Last Accessed on 2016-06-05)

[162] DDoS Attack Types.
https://www.stateoftheinternet.com/types-of-ddos-attacks.html
(Last Accessed on 2016-06-05)

[163] C. Douligeris, A. Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art”, Computer Networks, Elsevier, pp: 643–666.

[164] Security Alliances.
https://cloudsecurityalliance.org/
(Last Accessed on 2016-06-05)

[165] Cloud Audit.
http://cloudaudit.org/CloudAudit/Home.html
(Last Accessed on 2016-06-05)

[166] M. Monil and R.M. Rahman, “VM consolidation approach based on heuristics fuzzy logic, and migration control”, Journal of Cloud Computing, Advances, Systems and Applications, Springer, (2016), pp: 5-8.

[167] J.H. Morris, M. Satyanarayanan, M. H. Conner, J.H. Howard, D.S. Rosenthal, F.D.Smith, “Andrew: a distributed personal computing environment”, Communications of the ACM - The MIT Press scientific computation series CACM Homepage, ACM, (1986), pp: 184-201.

[168] T.P. Sharma, R.C. Joshi, Manoj Misra, “Tuning Data Reporting and Sensing for Continuous Monitoring in Wireless Sensor Networks”, in the proceedings of 27th IEEE International Performance Computing and Communications Conference, IEEE, (2008), pp: 412-417.

Publications

International Journals

Peer Clouds: A P2P-Based Resource Discovery Mechanism For The Intercloud, "International Journal of Next-Generation Computing", Vol. 6, No. 3, pp 153-164, November, 2013

Detecting And Containing Malicious Services In An Intercloud Environment, Journal of Web Engineering, Vol.15 No.5&6, November, pp 521-538, 2016

Book Chapters/International Conferences

A novel mechanism for dynamic optimization of intercloud services, "Confederated International Workshops: OTM Academy ISDE", Published in Lecture Notes In computer Science, *Springer,* pp. 377–388, Amantea, Italy.

International Conferences

1. Hierarchical Chord based Resource Discovery Mechanism in Intercloud Environment, "6th International Conference on Utility and Cloud Computing (UCC)", 2013, pp: 464-469, *IEEE/ACM*, Dresden, Germany.