VIRTUAL MACHINE ALLOCATION IN FEDERATED CLOUDS

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer Science

By

Monica Gahlawat
Enrollment No. 119997493009

under supervision of
Dr. Priyanka Sharma

**GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD**

June – 2016

VIRTUAL MACHINE ALLOCATION IN FEDERATED CLOUDS

A Thesis submitted to Gujarat Technological University

for the Award of

Doctor of Philosophy

in

Computer Science

By

Monica Gahlawat
Enrollment No. 119997493009

under supervision of
Dr. Priyanka Sharma

**GUJARAT TECHNOLOGICAL UNIVERSITY
AHMEDABAD**

June – 2016

© Monica Gahlawat

DECLARATION

I declare that the thesis entitled Virtual Machine Allocation in Federated Clouds submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from July 2011 to June 2016 under the supervision of Dr. Priyanka Sharma and this has not formed the basis for the award of any degree, diploma, associate ship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of Research Scholar: Date: 23rd June, 2016

Name of Research Scholar: Monica Gahlawat

Place: Ahmedabad

CERTIFICATE

I certify that the work incorporated in the thesis Virtual Machine Allocation in Federated Clouds by Ms. Monica Gahlawat was carried out by the candidate under my supervision/guidance. To the best of my knowledge: (i) the candidate has not submitted the same research work to any other institution for any degree/diploma, Associateship, Fellowship or other similar titles (ii) the thesis submitted is a record of original research work done by the Research Scholar during the period of study under my supervision, and (iii) the thesis represents independent research work on the part of the Research Scholar.

Signature of Supervisor: Date: 23rd June, 2016

Name of Supervisor: Dr. Priyanka Sharma

Place: Ahmedabad

Originality Report Certificate

It is certified that PhD Thesis titled Virtual Machine Allocation in Federated Clouds submitted by Ms. Monica Gahlawat has been examined by me. I undertake the following:

a. Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced.

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as Author own work.

c. There is no fabrication of data or results which have been compiled / analyzed.

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.

e. The thesis has been checked using **https://turnitin.com** (copy of originality report attached) and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <=25%).

Signature of Research Scholar: Date: 23rd June, 2016
Name of Research Scholar: Monica Gahlawat
Place: Ahmedabad

Signature of Supervisor: Date: 23rd June, 2016
Name of Supervisor: Dr. Priyanka Sharma
Place: Ahmedabad

VM Allocation in Federated Clouds Monika

ORIGINALITY REPORT

8%	7%	7%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	onlinelibrary.wiley.com Internet Source	5%
2	Computer Communications and Networks, 2014. Publication	4%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 1%
EXCLUDE BIBLIOGRAPHY	ON		

PhD THESIS Non-Exclusive License to GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD Research Scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, Monica Gahlawat having Enrollment No. 119997493009 hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

a) GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media;

b) GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a);

c) GTU is authorized to submit the Work at any National / International Library, under the authority of their "Thesis Non-Exclusive License";

d) The Universal Copyright Notice (©) shall appear on all copies made under the authority of this license;

e) I undertake to submit my thesis, through my University, to any Library and Archives.

Any abstract submitted with the thesis will be considered to form part of the thesis.

f) I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license.

g) If third party copyrighted material was included in my thesis for which, under the terms of the Copyright Act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph (a) above for the full term of copyright protection.

h) I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my University in this non-exclusive license.

i) I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my University in this non- exclusive license.

j) I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism.

Signature of Research Scholar: Date: 23rd June, 2016
Name of Research Scholar: Monica Gahlawat
Place: Ahmedabad

Signature of Supervisor: Date: 23rd June, 2016
Name of Supervisor: Dr. Priyanka Sharma
Place: Ahmedabad

Seal:

Thesis Approval Form

The viva-voce of the PhD Thesis submitted by Shri. /Smt./ Kum. Monica Gahlawat (Enrollment No. 119997493009) entitled Virtual Machine Allocation in Federated Clouds was conducted on 23rd June, 2016(Thursday) at Gujarat Technological University.

(Please tick any one of the following option)

- ☐ We recommend that he/she be awarded the Ph.D. Degree.
- ☐ We recommend that the viva-voce be re-conducted after incorporating the following suggestions.

- ☐ The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the Ph.D. degree.

Name and Signature of Supervisor with Seal

External Examiner -1 Name and Signature

External Examiner -2 Name and Signature

External Examiner -3 Name and Signature

ABSTRACT

Cloud computing has revolutionized the computing paradigm by enabling ubiquitous, elastic on-demand computing over the network usually Internet. Nowadays, cloud computing is used in every real world activity e.g. healthcare, mobile communication, banks, firms, gyms, farming etc. The increased demand in computational power resulted in establishing large-scale data centers. The data centers consume a lot of electricity for continuous customer service support. Data center electricity consumption is projected to increase to roughly 140 billion kilowatt-hours annually by 2020, the equivalent annual output of 50 power plants, costing American businesses $13 billion annually in electricity bills. However, the developments in virtualization technology have resulted in increased resources utilization across data centers, but energy efficient utilization of resources becomes a challenge. There is a debate that cloud computing is green, contributing to sustainability or a risk of climate change because of the huge amount of carbon emission from the data centers every year. Optimum utilization of resources can reduce energy consumption resulting in minimizing overall energy consumption of the data centers and carbon emission.

The thesis presents novel model for Energy Efficient Virtual Machine (VM) Allocation in Federated cloud data centers. The goal is to improve the utilization of the resources and reduce energy consumption without violating Service Level Agreement (SLA) terms. The VM allocation is performed in two phases. In the first phase the energy efficient Initial VM allocation algorithm is proposed and in the second phase, VM Allocation is optimized by dynamic VM consolidation approach. A Support vector machine regression-based method is proposed for VM consolidation phase. The key contributions are:

1. Initial VM Allocation algorithm analogous to bin packing algorithm has been proposed to pack the maximum virtual machine on less number of hosts being active consuming less power.
2. VM consolidation algorithm based on support vector machine is proposed for energy efficient automatic VM consolidation.
3. The algorithm also monitors the consolidation ratio to maintain the quality of Service of the data center.

ACKNOWLEDGEMENTS

Ph.D. is a unique endeavor and a once-in-a-life time opportunity. First of all, I would like to extend my sincere gratitude to my guide Dr. Priyanka Sharma, Professor (IT), Raksha Shakti University for her invaluable guidance and encouragement throughout my Ph.D. candidature.

Any journey can't be consummated without the support of people who directly or indirectly lend their time or resources, in order to reach your goal. I would like to thank my Doctoral Progress committee members: Dr. D. B. Choksi, Professor, Department of Computer Science & Technology Sardar Patel University and Dr. H.B Bhadka, Dean, Faculty of Computer Science, C.U Shah University for their excellent reviews and suggestions to improve the thesis.

I am heartily thankful to LJK trust especially Dr. Manish Shah, Vice President Lok Jagruti Kendra for providing such a great platform and Prof Dipti Chudgar for her continuous encouragement. Special thanks to my colleagues Prof. Dhaval Patel, Prof. Sangeeta Rajole and Prof. Meetu Joshi for their suggestions.

Most important, I would like to thanks my family especially my father-in-law, mother-in-law, bother-in-law, sister-in-law, Abhimanyu, Himansh and my husband for their constant support in tough times. Thanks to my parents and brothers for their extraordinary love and inspiration in the complete Ph.D. tenure. Special thanks to my daughter Kritvi for managing her studies when I was extremely busy.

Last but not the least I thank GOD for all the blessings, strength and the people around me who makes my life more meaningful.

List of Tables

List of Figures

Table of Content

CHAPTER 1

Cloud Computing

1.1 Introduction

With the growing popularity of the Internet, web and various handheld devices like mobile, sensing devices, etc. are changing the way we communicate, access and deliver services. Although the predecessor technologies to cloud like parallel and distributed computing, www, SOA (service-oriented architecture), autonomic and utility computing were already existing but when used collaboratively to evolve the cloud computing concept, brought a comprehensive change in the industry. Cloud computing is a paradigm shift from traditional on-premise computing towards elastic on-demand, ubiquitous computing. With the advent of cloud computing, computing is transformed into a model of services, delivered like a utility rather than products. In this model, the users can access the services based on the requirements over the network typically Internet. The services are metered and payment for the services is on the usage basis like electricity and water. The consumer is not aware of where the services are hosted. They access the applications using a web browser or a lightweight desktop or mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance as if the software programs were installed locally on end-user computers and the customer has to pay on the use basis like electricity bills [1].

Definition

According to NIST (National Institute of Standards and Technology)

"Cloud computing can be defined as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models." [2]

According to Prof. Raj Kumar Buyya – Univ. of Melbourne, Australia:[1][16]

"A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers."

1.2 Enabling Technologies behind Cloud Computing

Cloud computing has been evolved by the advancement in various technologies e.g. Distributed computing (cluster, Grid Computing etc.), internet technologies (Service-oriented architecture (SOA), web 3.0 etc.), hardware Technologies (multi-core chips, virtualizations etc.) and system management technologies e.g. autonomic computing. Figure 1.1 shows the predecessor and the contributor technologies in the advent of cloud computing.

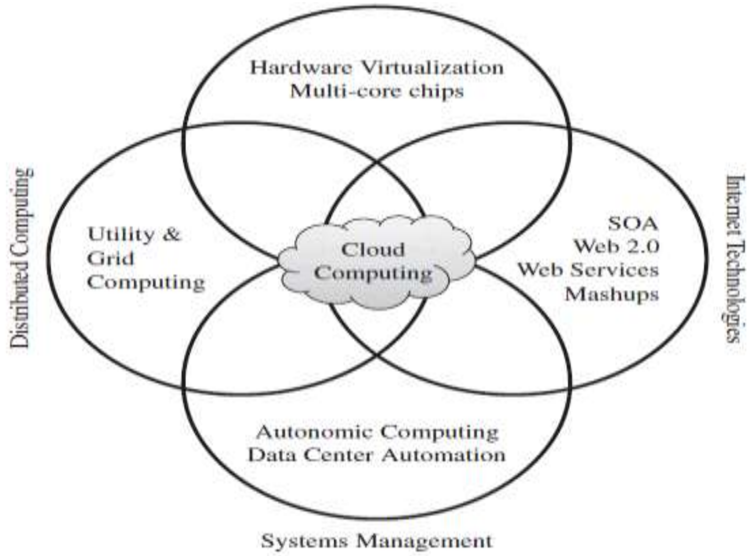


Figure1.1 The Contributor and Predecessor Technologies[3]

1.2.1 SOA, Web 2.0, Web services, Mashups:

Cloud computing is a paradigm shift from Application Oriented Architecture to Service Oriented Architecture (SOA). A Service Oriented Architecture is a set of collection of web services. Web services are self-contained, self-describing, platform independent programs that can be invoked over the Internet.[6] The Orchestration of web services in a particular manner can be exposed as a single web service to solve a particular activity over the Internet. The advent of web 2.0 contributed in gluing up the web services together with the help of a new programming technique called AJAX (Asynchronous JavaScript and XML), REST(REpresentational State Transfer), RSS(Really Simple Syndication). AJAX is a technique to update the web pages without reloading the complete page. RSS distributes the up-to-date information to the web pages dynamically. The data exposed by the REST and RSS are extracted by a technique called a mashup. A mashup is defined as a website or web application that uses content or services from more than one source to create a completely new service [4].

1.2.2 Autonomic Computing:

Autonomic Computing is a type of computing model in which the system is self-managing and adapt to the unpredictable changes. It is a collection of various existing technologies, including adaptive algorithms, human-computer interaction, machine learning algorithms, artificial intelligence algorithms, software agents, robotics, etc. autonomic computing means computing without or least human intervention. [5]. It is helpful to minimize the time required by the computer professional to resolve the system difficulties and the other maintenance work like software updates. [7]

1.2.3 Utility and Grid Computing:

The current model of cloud computing is an evolution of grid computing. Both the computing paradigms are based on utility computing. Utility computing is a service delivering model in which a service provider makes available the required resources to the customer and charges them for specific usage rather than a fixed rate. Cloud differs from grid computing in the sense that the cloud provides user-centric interfaces and does not require learning new commands and API as required in the case of grid computing.

1.2.4 Hardware Virtualization and Multi-core chips:

Virtualization is a technique to create a virtual version of operating system, network, CPU, server, storage devices, etc. virtualization is an integrated solution to increase the resource utilization in a data center[8]. Virtualization and cloud computing are used interchangeably but there is a significant difference between both the technologies, virtualization is a technology that manipulates hardware and cloud computing is a service that is a result of the manipulation done by virtualization. Virtualization is a basis for cloud computing that helps in improving resource utilization rate.

1.3 Characteristics, Service and Deployment Models of Cloud Computing

NIST (National institute of Standards and Technology) have defined 4 deployment models, 3 service models and 5 core key characteristics of cloud computing. Figure 1.2 shows the service, deployment models and the essential characteristics of cloud computing.

1.3.1 Essential Characteristics of Cloud Computing

A. **On-demand self-service:** On demand self-service is a process through which the consumers can provision the computing capabilities automatically as needed without human intervention.

B. **A broad network access:** Broad network access means that the resources are available to be accessed from a wide variety of devices such as mobile phones, tablets etc.

C. **Resource pooling:** Resource pooling allows cloud providers to scale up the computing infrastructure based on dynamic cloud consumers' requirements. Resource pooling can be achieved by applying multi-tenant model where a cloud provider pools its resources to serve multiple cloud consumers.

D. **Rapid elasticity:** Elasticity means the ability to scale up and down based on the consumers' resource requirements. Scalability can be horizontal or vertical, horizontal scalability means to add and remove the resources from the resource pool and the vertical elasticity means to add and remove resources from an existing node or server.

E. **Measured Service:** cloud computing is based on utility computing means that the resource usage is controlled and measured to support transparency between cloud service provider and the consumer. It is vital for billing, access control, resource management and capacity planning etc.

1.3.2 Service Models of Cloud Computing

Cloud computing is based on Service Oriented Architecture. Cloud service providers' offers their services to cloud consumers based on 3 service models shown in Figure 1.2. Based on this general categorization of the cloud services, SPI(**S**aaS,**P**aaS,**I**aaS) model was shaped and denotes the software, platform and infrastructure of the cloud respectively. SPI model is also known as the software stack.

Software–as-a-Service (SaaS) is on the top of cloud service stack and is also called a cloud application layer [10]. The targeted users of the cloud applications are end-users. Cloud applications are normally web applications with predefined functionalities and scope. The customization of the applications is also possible based on the end-user requirements. The costing model for the software as a service is either subscription based or follows the pay-per-use model. SaaS model is useful as compared to the traditional software purchase model when the application is required for a short span of time and the demand changes rapidly e.g. Tax and billing software and the collaboration software etc. The players of the SaaS industry are SalesForce.com, Abiquo, Cloud9, AccelOps, CloudTran etc.

Platform-as-a-service (PaaS) is a model of delivering platform (hardware and software) to the developers to develop, execute cloud applications. A PaaS provider hosts the platform on its own infrastructure or can use the outsourced Infrastructure. The consumer has to pay on the basis of the usage. The model frees the developers from purchasing the software, software licenses and installing the supporting hardware and software. It is also useful in the situation where multiple developers are working on a single application. This also reduces the overall development time. Swot analysis of PaaS concludes that though the model is useful in collaborative software development, but the cloud service consumers are dependent on the capabilities provided by the cloud provider [11]. The platform as a Service provider companies are openStack, Google app engine, Amazon web services, windows Azure etc.

Infrastructure-as-a-Service (IaaS) is a model of delivering processing power, storage, network and other fundamental resources so that the IaaS service consumer can deploy and run arbitrary software on the infrastructure provided by the cloud vendor[11]. The consumer does not have the control of infrastructure but has control over the operating system and the software installed over the infrastructure. Limited access to the networking resources is also provided to the consumers.

IaaS is a cost effective solution for companies starting a new endeavor as they do not have to invest in the infrastructure and can focus on the main business activities. IaaS companies in the market are hp, goGrid, Amazon, Datapipe, logic works etc.

1.3.3 Deployment Models of Cloud Computing

Categorization of deployment models depends on the way the cloud is installed, accessed and owned by the cloud consumer. Which deployment model to use depends on the necessity, purpose of the cloud consumer. Most of the companies are migrating to cloud for their cost effective business activities. The 4 types of cloud deployment models are:

A. Public Cloud: A public cloud as the name suggests, allows the resources to be accessed publicly to all. A fundamental characteristic of the public cloud is that it is meant to serve the multitude of users, not a single user. The cloud providers can establish their own infrastructure or can use the infrastructure owned by any third-party provider and allows accessing the resources free of charge or on a pay-as-usage policy. This model is the cost-effective solution as compared to other deployment models because the infrastructure is shared between various users at a movement. Popular public cloud providers are Google, Amazon Ec2, IBM Smart Cloud and SalesForce.com etc.

B. **Private Cloud:** Private cloud or internal cloud is a model which is implemented exclusively for one customer or organization in order to meet the high security, privacy, high customization needs of the customer. This type of cloud allows the organizations to use the cloud computing technology as a means of centralized access to its resources from different locations, departments of the organization. The private cloud can be installed on-site or outsourced.

1) O**n-site private cloud** : cloud is implemented at customer's premises

2) **Outsourced Private cloud**: server side is outsourced to a hosting company.

The popular private cloud providers are Amazon VPC(Virtual Private Cloud), VMware Cloud Infrastructure Suite and Microsoft ECI data center etc.

C. Hybrid Cloud: The hybrid model is an integration of two or more cloud deployment model i.e public, private and community cloud. This model combines the advantages of all deployment models, e.g. cost effectiveness, security, etc. Hybrid clouds can be used by the organizations to keep the sensitive data in the private cloud but private cloud suffers from an inability to scale on demand and to efficiently address peak loads. In this case, it is beneficial to use public cloud capabilities as needed.Hybrid deployment architecture can be difficult to construct because of the disparity in the cloud environment.

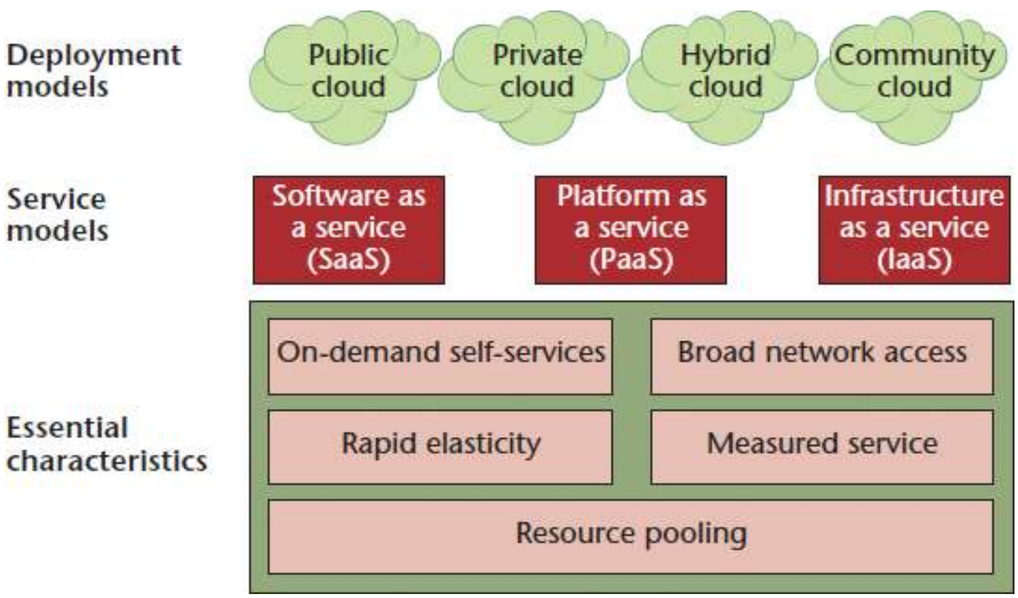


Figure 1.2 Service, Deployment model and cloud characteristics [9]

D. Community Cloud: community cloud is a model in which the setup is mutually shared among multiple organizations having similar policies and compliance considerations. For example, banks and trading firms have similar privacy, security and performance concerns so the single cloud can be shared between all these organizations and the multi-tenant storage can be used for the cost-effective solutions.

1) **On-site Community Cloud Scenario :** community cloud is implemented at the customer site.

2) **OutSourced Community Cloud:** Server side is outsourced to a hosting company.

   The examples of community clouds are Google Apps for Government, Microsoft Government Community Cloud etc.

1.4 Motivation and Research Objective

The evolution of the cloud computing paradigm has created an illusion of infinite resources for executing customers applications on-demand basis. To handle sudden spikes in the customer demands the cloud providers establishes the large cloud infrastructure. However, this vision raises severe issues with energy consumption. Figure 1.3 shows the power consumption estimates of the USA datacenters from 2000 to 2020. The huge amount of energy consumption not only increases the energy cost but also results in heavy carbon emission which increases health-related risks to the society [15].

The cloud data centers consume a considerable amount of energy even for powering the idle resources. The current trends are unsustainable with respect to resource utilization, carbon emission and overall energy consumption of the data centers. Some of the challenges are listed in Table 1.1. Energy Efficient approaches proposed in the literature based on processor level, process level or data center level are proved beneficial in a single cloud. On the other hand, the cloud infrastructure is federating outside the boundaries of the single cloud. There is a great need for the approaches those can be applied to federated clouds. There is an unexplored potential for energy aware operations in the federated clouds.

Though every IaaS provider is restricting the energy consumption at their own level but there is a great need for some software-based schemes to reduce the overall usage of the energy in the federation. The VM migration can be a costly operation in cloud federation so the migration between federation should also be a controlled activity. The work deal with the research challenges in relation to energy-efficient VM allocation and consolidation in IaaS federated cloud under Quality of service constraints. In particular, the following research problems are investigated:

A. **Which PM to select for VM Allocation.** Since the VM allocation process is a two-phase process, the first task is to select the appropriate PM to allocate VM so that less number of hosts remains active at a particular point in time.

B. **When and Which PM to switch on/off.** During VM allocation and consolidation the objective is to minimize the number of active hosts. But the VM consolidation ratio should also not be so high that it violates the SLA (service level agreements). So it is necessary to appropriately determine that which and when the PM should be off so that SLA should not be violated and the workload should also be allocated optimally.

C. **Determining the overloaded hosts.** Virtual machines need to be migrated is based on the host overload and underload criteria. If the host is overloaded then VMs from the host needs to be migrated to avoid SLA violations. Host overload detection is a crucial task in VM migration. The proper host should be determined to improve the VM consolidation process.

D. **Minimizing the Scheduling Delay:** Federated clouds are getting attention from cloud providers as well as the cloud consumers. Despite the advantages, Federated Clouds also presents some downsides as the resources are federated geographically distributed in different cloud domains. There is a crucial need of an algorithm which minimizes the scheduling delay in Virtual Machine allocation.

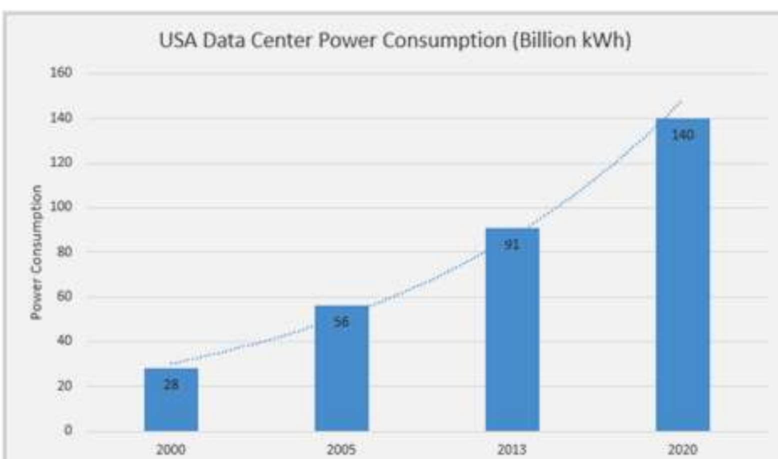


Figure 1.3: Power consumption estimates in data centers [14]

Table 1.1 Research Challenges of IaaS,PaaS and SaaS

Attribute System	Challenges
Infrastructure-as-a-Service	1. Temperature of cloud places need to be maintained 2. System should be power failure tolerant 3. Selection of infrastructure hardware is very important 4. Connection between cloud and hardware should be a high bandwidth channel 5. Storage of cloud should be able to fulfill the changing demands of large data size 6. Reduction in Energy Consumption and carbon emission.
Software-as-a-Service	1. Data security is highly preferred feature 2. High availability requirement & Data Backup 3. Authentication and authorization 4. Data integrity, Privacy 5. Deployment of cloud resources in different countries results in conflict of rules
Platform-as-a-Structure	1. Limited APIs 2. Data Lock-in 3. Auditability 4. Performance is unpredictable 5. Lack of control over low level security 6. Data inaccessibility between applications 7. Vulnerabilities of web applications and SOA

To deal with the above challenges , the following objectives are delineated.

- Analyze, explore the methods available in the energy efficient data center management to get the methodical understanding of the existing techniques and approaches.
- Conduct competitive analysis of algorithms for VM allocation & consolidation to obtain theoretical performance estimates and insights into the design of algorithms.
- Design and develop an efficient algorithm to allocate VM on PM in energy efficient manner with minimum possible scheduling delay.
- Propose an efficient algorithm which can predict accurately the overloaded host considering the current workload.

1.5 Problem Statement

The objective of the work is to optimize the virtual machine allocation in federated clouds. The optimization problem is divided into two phases 1) Initial VM allocation 2) VM consolidation. The problem statement for the same is as under:

Let $R(r_1, r_2.............r_m)$ be the set of resources in the data center i with a capacity c_i^k where $k \in R$. The energy consumption for the data center E_i is defined as $E_i(t) = (P_{\max} - P_{\min})*U_i(t) + P_{\min}$ where

$P_{\max} =$ Power Consumption at the peak load

$P_{\min} =$ Minimum Power Consumption in active mode

$U_i =$ Resource Utilization in Data center i where $U_i(t) = \sum_{j=1}^{m} CjXj$

$C_j =$ No of MIPS (Millions Instructions Per Second) executed by the CPU

$Xj =$ 1/Total capacity in terms of MIPS of CPU

The cost of the power per unit is $\cos t_p$ and the cost of the SLA violation is $\cos t_v$. If the cloud provider does not have enough cloud resources or the SLA will be violated, the cloud provider first contacts with another Cloud Provider and negotiates with it the execution of this task. In this case, the price of this execution is defined as:

$T\cos t = \cos t_p + \cos t_v$ $\qquad$ $\cos t_v = \lambda_t + \varepsilon_t$

$\lambda_t =$ Migration Cost of task t including Machine cost

ε_t = cost Energy Impact of task t on the environment

If all negotiations fail, the Cloud Provider finds the SLA whose violation implies in lower cost and executes the task. In this case, the price to execute the tasks is defined as

$N\cos t_v = T\cos t_v + \lambda + \delta$

where λ is the cost of violating the QoS requirements of other tasks and δ is the cost associated with energy consumption violation.

$$\text{Minimize } E = \sum_{t=1}^{T}\sum_{j=1}^{m} E_i(t) \qquad \text{subject to} \qquad E_i(t) = (P_{\max} - P_{\min}) * U_i(t) + P_{\min}$$

$U_i(t) \leq$ peak load at time t , ∀ Ri ϵ R and ∀ tj ε T

1.5 Contributions

Cloud Computing is an emerging paradigm promising to turn the vision of "computing utilities" into reality. According to the report presented in "Revolutionizing Data Center Energy Efficiency" [23], a typical data-center consumes as much energy as 25,000 households. Energy costs of powering a typical data center double every five years. Because energy costs are increasing, there is a need to shift focus from optimizing datacenter resource management for pure performance alone to optimizing for energy efficiency while maintaining high service-level performance. The Objective of the research is to optimize the virtual Machine allocation in federated clouds. It is quality based research which optimizes the VM allocation based on energy. The key contributions of the thesis are.

1. An organization & survey of the state of the art of the energy efficient VM allocation algorithms.
2. Design and develop an initial VM Allocation algorithm analogous to bin packing algorithm to pack the maximum virtual machine on less number of hosts being active consuming less power.
    a. A dynamic data structure (Divided KD tree) is used to search the optimum host for every particular VM request.
    b. Apart from exact match queries the KD trees also supports range queries and most importantly multi-attribute queries to make the searching fast as compare to other search data structures.

3. Design and develop VM consolidation algorithm based on support vector machine for automatic VM consolidation.
   a. Support vector machine analyses the CPU utilization of the host as a time series and predicts the overloaded hosts.
   b. Support Vector Machine is so effective in time series prediction that it has outperformed the artificial neural network based methods.
4. VM Selection Framework for market-oriented federated clouds is also proposed. The framework selects the virtual machines from the pool of virtual machines in market exchange on the basis of multiple attributes specified by the consumer. The parameter of searching includes cost, CPU, memory, disk , MIPS(Millions of Instructions per second etc.)

With the help of the Initial VM allocation algorithm and the VM Consolidation algorithm, the data centers operations are performed in fast and autonomous manner. In the first phase of VM allocation divided KD tree will pack the virtual machine in the appropriate host so that the total energy usage is minimized and in the second phase the support vector machine will predict the overloaded hosts so that load balancing and VM consolidations can be done effectively. Apart from this, the VM selection framework proposed in the third phase of the thesis will provide a rapid resource discovery in the federated environment.

1.6 Thesis Organization

The core chapters of the thesis are derived from the several conferences and journal papers published during the Ph.D. tenure. The remainder of the thesis is organized as follows:

Chapter 2 presents survey and taxonomy of inter-cloud infrastructures and the energy efficient VM allocation and VM consolidation algorithms present in the literature. This chapter is derived from [17-19]

Gahlawat, Monica, and Priyanka Sharma. "Survey of virtual machine placement in federated clouds." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.

Gahlawat, Monica, and Priyanka Sharma. "Green, Energy-Efficient Computing and Sustainability Issues in Cloud." Managing and Processing Big Data in Cloud Computing (2016): 206.

Gahlawat, Monica, and Priyanka Sharma. "Reducing the cost of virtual machine migration in the federated cloud environment using component based VM." Journal of Information Systems and Communication 3.1 (2012): 288.

Chapter 3 presents the importance of the virtualization technologies in VM allocation process. It explains how virtual machines are created and migrated in the cloud data-centers. **Chapter 4** explains the first phase of the virtual machine allocation model in the federated clouds. The chapter explains the federated cloud architecture and a space partitioning KD-tree based virtual machine allocation. The outcome of the chapter is energy efficient allocation of the virtual machine which is also contributed positively in improving the scheduling time. the chapter is derived from [20]

Monica Gahlawat and Priyanka Sharma. 2015. A Multi-Objective Initial Virtual Machine Allocation in Clouds using Divided KD Tree. In Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15), Indu Nair (Ed.). ACM, New York, NY, USA, 656-660. DOI=http://dx.doi.org/10.1145/2791405.2791560

Chapter 5 explain the support vector machine based host overload detection algorithm for VM consolidation. The VM consolidation is based on host overload/underload behavior and the VM selection algorithm. This chapter is derived from [21]

Gahlawat, Monica, and Priyanka Sharma. "Support Vector Machine-Based Model for Host Overload Detection in Clouds." Proceedings of International Conference on ICT for Sustainable Development. Springer Singapore, 2016.

Chapter 6 proposes a VM selection framework for the Market-oriented federated clouds. The VM selection framework is also based on the divided KD tree. Divided KD tree is a multi-dimensional space partitioning tree which supports fast and reliable multi-dimensional queries. The chapter is derived from [22]

Gahlawat Monica, and Priyanka Sharma. "VM selection framework for market-based federated cloud environment." Computing, Communication & Automation (ICCCA), 2015 International Conference on. IEEE, 2015.

Chapter 7 concludes the thesis by providing the summary of main findings, discussion of future directions and future remarks.

CHAPTER 2

Literature Review

2.1 Introduction

The computing systems were designed with the objective that the performance of scientific and business activities can be improved. With the advancements in the hardware and the technologies the performance of the computing systems increased. Although the performance per watt of the power is increasing, the total power consumption of the systems has not reduced. The severity of the problem is increasing in large scale distributed data centers. The objective of the cloud paradigm is to provide scalability of the infrastructure, platform and software on-demand. To satisfy the sudden spikes in the computing demand of the customers, the cloud vendors need to keep the data centers up-to-date resulting in high power consumption. It is projected that by 2020 the total power consumption of the USA data centers will be 140 billion kWh (kilowatts per hour) [81]. Energy efficient computing in the federated cloud is even more challenging because the resource allocation policies of the clouds participating in the federation may differ from each other.

2.2 Inter Clouds Architecture Taxonomy

Inter-cloud means the clouds of clouds. Inter clouds are advantageous to both cloud users and the cloud vendors. The cloud users have the advantage that they can avoid vendor lock-in if the provider is changing its policies which are affecting its performance or cost. It also helps in service outage situations; the workload can be safely transferred from the one cloud to another. Inter-cloud is also beneficial to cloud vendors in the sense that it can ensure enough resources at any time. [25] Inter-clouds are categorized into 2 types of cloud architectures, e.g. federated clouds and multi-clouds. Federated clouds are those types of architectures in which the cloud vendors voluntarily share their resources with each other. In multi-cloud architecture, multiple independent clouds participate in the cloud federation which is not volunteer based. The taxonomy of the inter-cloud architectures is shown in Figure 2.1.

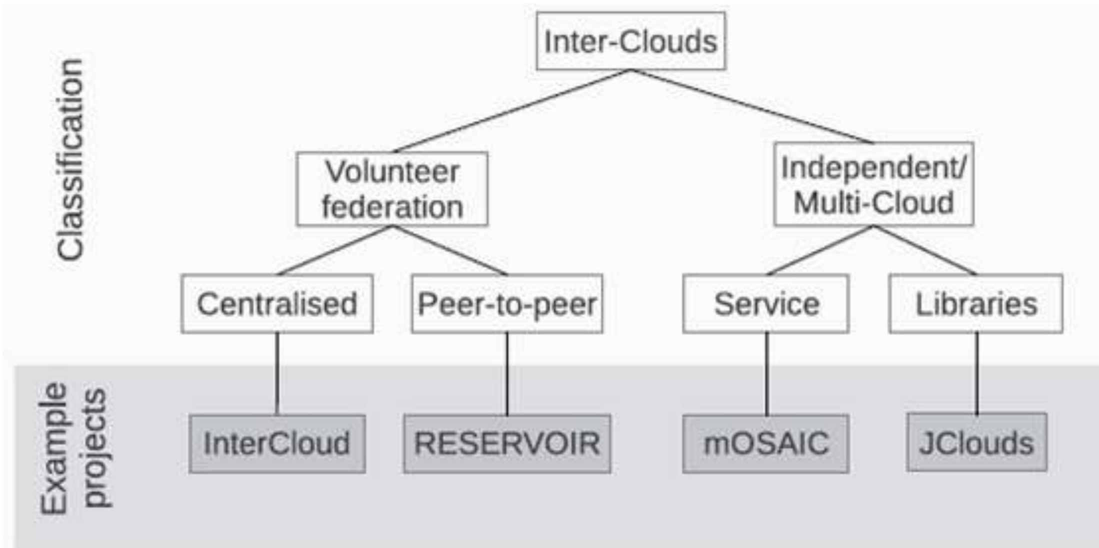
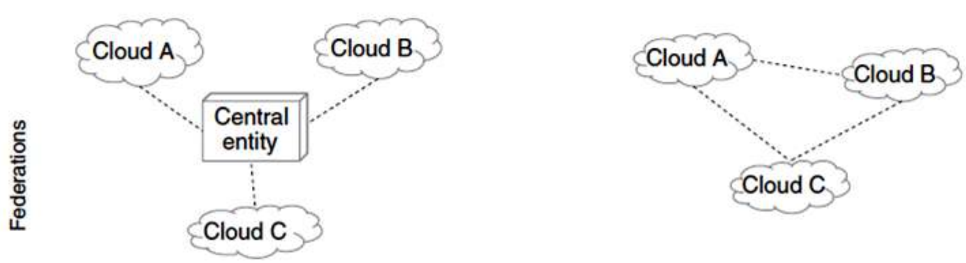


Figure 2.1: Taxonomy of Inter-cloud architectures [26]

Based on the architecture the volunteer federation can be further classified as:

Centralized Federation means there is a single entity, positioned centrally which performs or facilitates resource allocation. In *Peer-to-peer federation*, the participating clouds communicate and negotiate directly without any mediator. Figure 2.2 shows the architectures of centralized as well as the peer-to-peer federation.



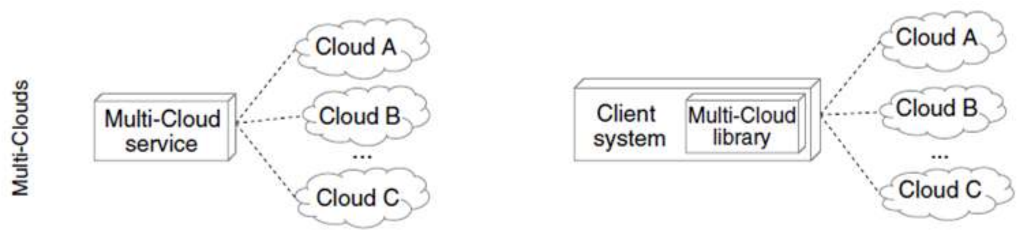
(b)

Figure 2.2 (a) centralized Federation (b) Peer-to-peer Federation [28]

The multi-cloud inter- cloud architecture are categorized as:

Service: Application provisioning is done with the help of services deployed externally or in-house at the client side. These services normally include a broker component which takes input from the client as provisioning rules and the service level agreement rules. These services run in the background to perform the provisioning work.

Libraries: Using Library based multi-cloud approach the applications are implemented by linking the multi-cloud libraries in the built path so that the applications can be deployed using any of the libraries and the migration of the applications from one environment to another is possible without redesigning the application. Examples of the multi-cloud libraries includes a Java-based j*clouds, JetS3t, Whirr*, Python based *boot, libcloud, PyStratus* and Ruby based *fog and deltaCloud* etc.[28] .Figure 2.3 shows the categories of multi-clouds.



(b)

Figure 2.3 (a) Multi-cloud Service (b) Multi-cloud Library [28]

Various inter-cloud projects have been proposed in the literature based on centralized, peer-to-peer, services or libraries.

2.2.1 OPTIMIS cloud Architecture

OPTIMIS project allows organizations to easily provision on federated cloud infrastructure and higher-level solutions, like platform as a service (PaaS). Their goal is to optimize the resources used by multiple providers in a transparent, interoperable and architecture- independent manner to support higher-level concerns and nonfunctional aspects to achieve a wider adoption of Cloud Computing. The architectural views of the OPTIMIS project are based on three scenarios(i) A service provider(SP) accesses an infrastructure provider(IP) which shares the resources from another IP. (ii) The second architecture is based on multi-cloud architecture, the different infrastructure providers are accessed separately by a service provider. (iii) The service provider accesses a private cloud and the private cloud can utilize the resources from different infrastructure providers. The architectural views are presented in Figure 2.4.

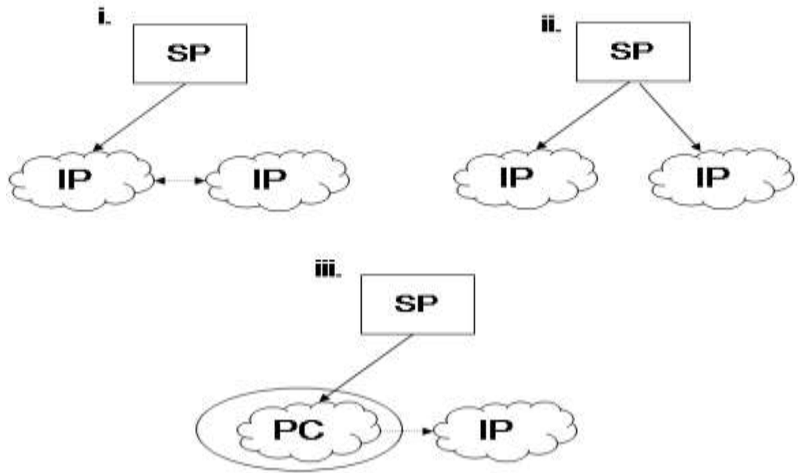


Figure 2.4 Architectures of OPTIMIS [74]

2.2.2 Reservoir Architecture

RESERVOIR project is European research initiative. The primary goal of the project is to deal with the scalability problem of the single provider cloud computing model. RESERVOIR explores the concept of a federated cloud in which computing infrastructure providers leases their extra capacity to fulfill transitory additional resources. RESERVOIR project comes under the category of volunteer federation in which two or more cloud providers voluntarily participate in the federation and share their resources when required. Each infrastructure provider (IP) is considered as a Reservoir site (RS). Each RS has resources that are partitioned into isolated Virtual Environment Execution (VEE). The applications are able to use VEE hosts from different RSs simultaneously. Virtual Execution Environment Managers (VEEM) interacts with VEEs, Service Managers and other VEEMs to enable federations to be formed. The architecture diagram is shown in Figure 2.5.

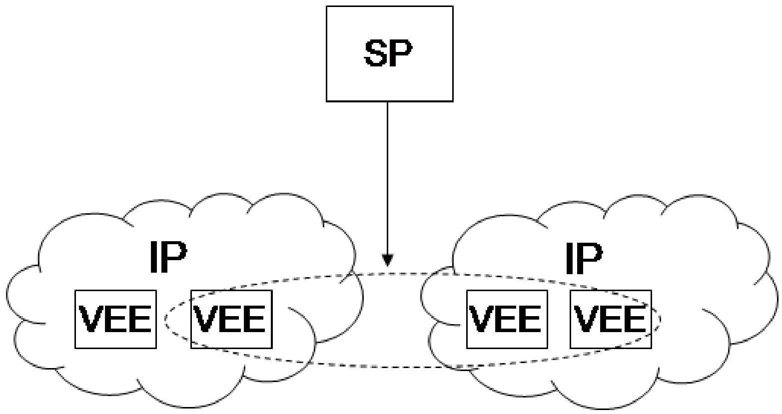


Figure 2.5 Architecture of RESERVOIR [74]

2.2.3 Contrail Architecture

Contrail enables interoperability amongst Cloud providers by eliminating the vendor dependency (data lock-in). Contrail supports both horizontal as well as vertical integration. The user requests are satisfied by assigning the best provider for the job, according to the requirements and type of application automatically. Two or more providers can be integrated for execution of a single or multiple applications. The computing by integrating multiple clouds is so smooth that they are considered as a single Cloud. The architecture of the contrail is shown in Figure 2.6.

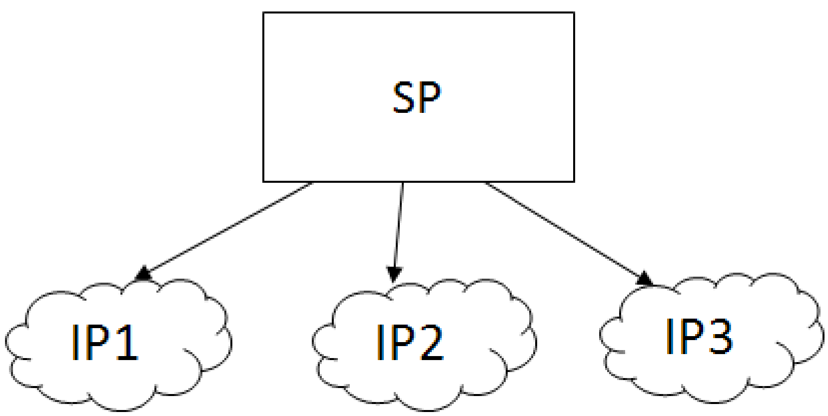


Figure 2.6: Architecture of CONTRAIL [74]

2.2.4 BonFIRE Architecture

BonFIRE combines seven testbeds(EPCC (UK), INRIA (France), PSNC (Poland), HLRS (Germany), iMinds (Belgium), Wellness Telecom (Spain) and HP (UK)) throughout the Europe which allows users to execute experiments that use compute, network, storage and router resources. BonFIRE also supports horizontal as well as vertical integration with the services. The BonFIRE user can create resources in AMAZON. The architectural diagram of BonFIRE is shown in Figure 2.7.

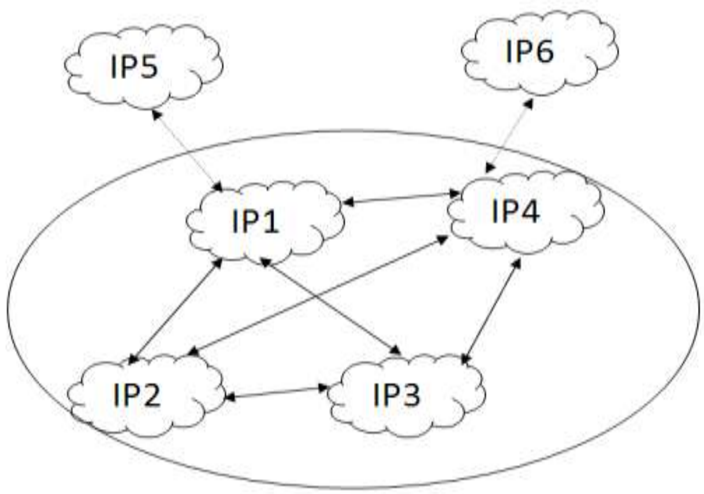


Figure 2.7 : BonFIRE Architecture [74]

2.2.5 mOSAIC Architecture

mOSAIC is an Open-source API and Platform for Multiple Clouds. It is service based API for the multi-cloud federation that is installed on the client machine to provide data and application interoperable and portability. It promotes competition between the cloud providers by enabling the selection of best-fitting cloud services to actual user needs and efficiently outsources computations. The architecture of mOSIC is presented in Figure 2.8.

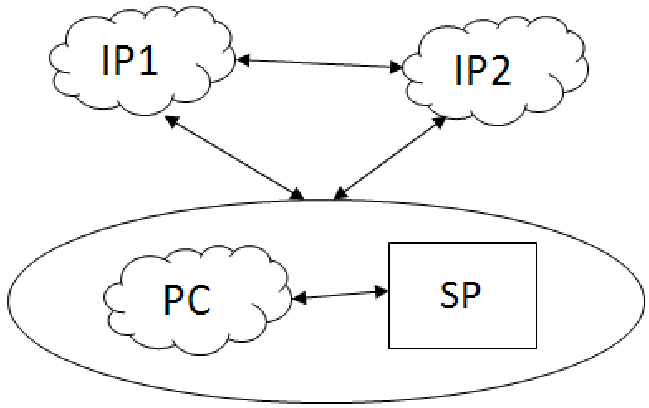


Figure 2.8: mOSIC Architecture [74]

2.2.6 EGI Federated Cloud Architecture

The European Grid Infrastructure (EGI) is a federation of academic private clouds and virtualized resources, focusing on the requirements of the scientific community. Currently, 16 cloud providers are participating in federation testbed using OpenNebula, OpenStack and StratusLab. The main objective of the federation is to securely federate and share the virtualized environments as part of EGI production infrastructure and to create a testbed to evaluate the performance. The EGI architecture is shown in Figure 2.9.

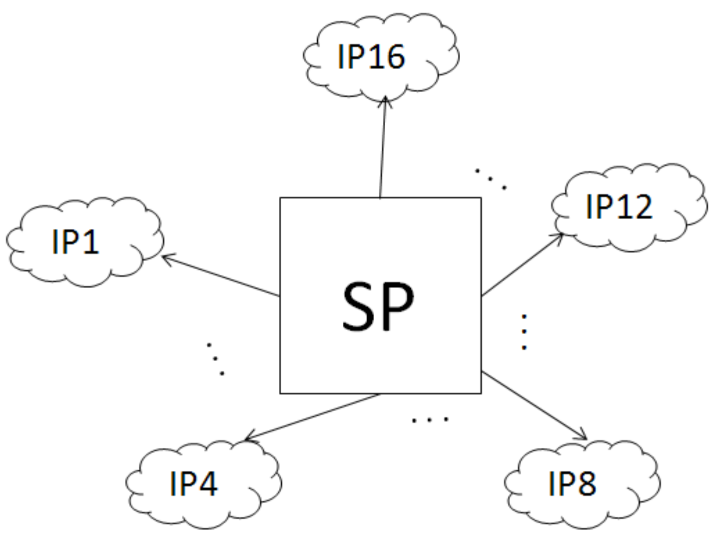


Figure 2.9 EGI Federation architecture [74]

Table 2.1 summarizes the existing projects with their architecture. Apart from these projects several simulation tools have also been proposed. Simulation tools are helpful to determine the accuracy and effectiveness of the design before the system is actually constructed in a faster and cheaper way. We have used SmartFED simulator based on CloudSim simulator for performing the experiments. The federation is modeled based on the 5 components, e.g. application model, Resource model, Application Queue Model, Resource Monitoring and Resource allocation model [31]. The applications are modeled in the form or undirected graph and the resources are allocated based on the constraints. The application is submitted as a graph to the federation and the allocation matrix is returned as a result of allocation. The applications are not necessarily allocated to a single cloud. The application can be distributed between the different data-centers and the even in different clouds. Allocation matrix defines the application distribution in different data centers. Internet estimator models the very basic and important property, e.g. network bandwidth between data centers. Figure 2.10 explains the package structure of SmartFED which is extending classes from CloudSim.

Table 2.1

Existing inter-cloud Projects [26]

Project	**Type Organization**	**Architectures**	**Brokering approach**	**Application able**	**Awareness**
InterCloud	Research project, the University of Melbourne	Centralized federation	SLA based and directly managed	Singular jobs	Geo-location, pricing
Contrail	Private and public European research organizations funded by EU	Centralized federation and Organizations independent service	SLA based	Singular jobs	Pricing
Dynamic Cloud Collaboration (DCC)	Academic research project supported by South Korean research funds	Centralized federation	SLA based	Singular jobs	Pricing

RESERVOIR	Private and public European research organizations funded by EU	Peer-to-peer federation	SLA based and Trigger-Action	Singular jobs	Pricing
Open Cirrus	Research testbed by academic and industry partners. Partially funded by US NSF	Peer-to-peer federation	Directly managed	Singular jobs	Data location
OPTIMIS	Private and public European research organizations funded by EU	Peer-to-peer federation and Independent service	SLA based	Singular jobs, periodical jobs, compute-intensive and data-intensive interactive application	Pricing
mOSAIC	Private and public European research organizations funded by EU	Independent service	SLA based	Singular jobs, periodical jobs, compute-intensive and data-intensive interactive Application	Pricing
STRATOS	York University. Supported by Canada's NSERC funds, Amazon and CA Inc.	Independent service	SLA based	Singular jobs, periodical jobs, compute-intensive and data-intensive interactive Application	Geo-location, pricing, legislation/policy and local resources
Commercial Cloud Management Systems (RightScale, EnStratus, Scalr, Kaavo)	Commercially owned	Independent service	Trigger-Action	Singular jobs	Geo-location, data location, pricing, legislation/policy and local resources
Libraries (JClouds, LibCloud, DeltaCloud, SimpleCloud, Apache Nuvem)	Open source projects	Multi-Cloud libraries	Directly managed	Singular jobs, periodical jobs, compute-intensive and data-intensive interactive Application	Geo-location, data location, pricing, legislation/policy and local resources

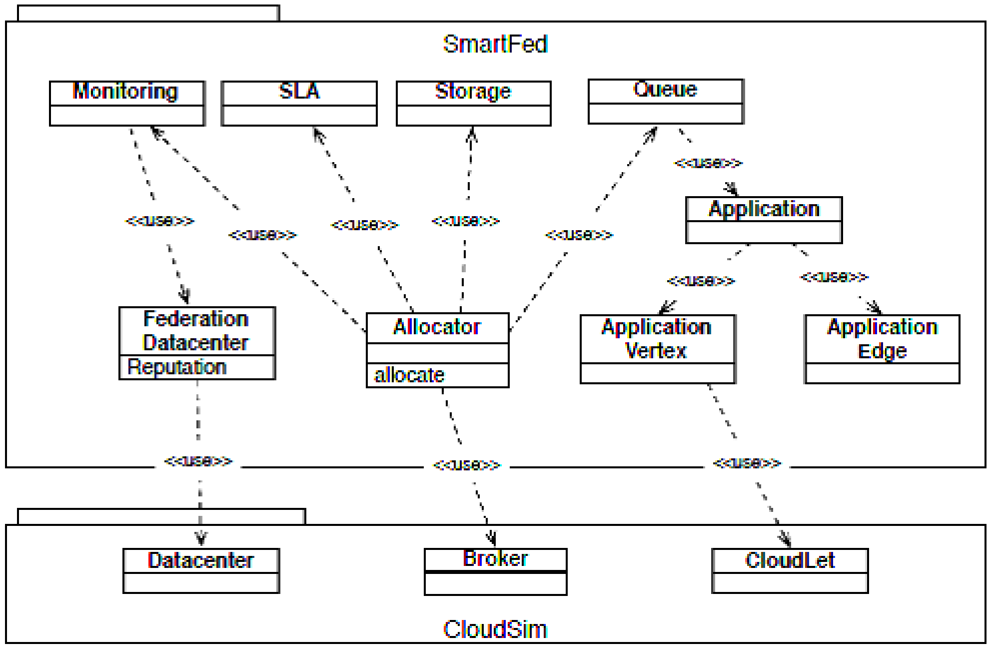


Figure 2.10 SmartFED main entities [31]

2.3 Cloud Federation Stack

The cloud federation stack is divided into three layers i.e. conceptual layer, logical and operational layer, infrastructure layer. Figure 2.11 shows the cloud federation reference model and the challenges faced by each layer so that the service providers can integrate their services in such a transparent manner that it is considered as a single cloud entity. The conceptual level addresses the challenges in presenting the cloud federation as a favorable solution for the cloud provider as well as the consumer. From the perspective of the cloud provider, the federation is favorable if it helps in increasing their revenue. The federation is also beneficial for the consumers as they can get the resources in a cost-effective manner and can avoid vendor lock-in.

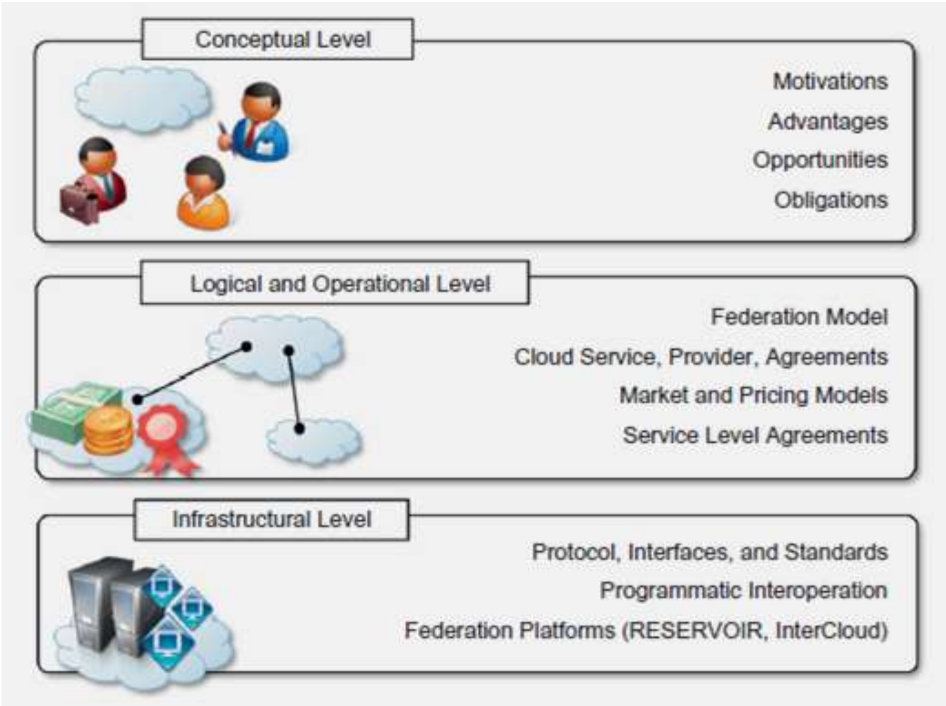


Figure 2.11: Cloud Federation Stack [1]

Logical and operational level addresses the challenges related to the policies and the rules for interoperations. This layer defines the context within which agreements among different providers are settled and services are negotiated. At the lowest infrastructure level addresses the technical challenges involved in enabling heterogeneous cloud computing systems to interoperate seamlessly. This layer deals with designing protocols and interfaces to overcome the barriers of the federation. We are designing an algorithm for the infrastructure layer so that energy efficient VM allocation can be performed with minimum scheduling delay.

2.4 Energy Efficient Resource Allocation

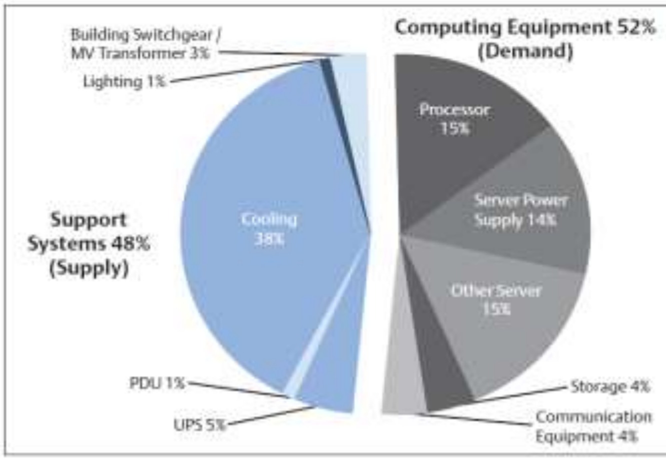
Recent development in the virtualization technology have resulted in its usage across data centers, because of the virtualization technology the virtual machines can be resized and the VM consolidation takes place to run the maximum number of VMs on a single server to increase the performance of the server and also to reduce the total power consumption of the data center. Before designing an energy efficient algorithm it is important to understand the power consumption patterns and identifying the components those are contributing to the considerable power consumption.

2.4.1 Power Consumption Patterns

Energy consumption patterns are important to be understood before designing energy efficient algorithms for large-scale data centers. Emerson Network Power analyzed 5000 square-foot data center to understand the power consumption pattern and concluded that the energy consumption is categorized either as “demand side” or at “supply side”. Figure 2.12 shows the energy patterns in the data centers. It shows that the power consumption of computing equipment is 52% and the support system of the data centers is 48%. The computing equipment are processors, server power supply, other server components and storage. The supply equipment includes cooling systems, UPS and lighting, etc. It has also been analyzed in Emerson network power that the effect of reduction in the power consumption at “demand side” cascades to the “supply side”. For example, for every 1-watt reduction on the demand side, there is an additional 1.84-watt reduction in the supply side. [30]

2.4.2 Static and Dynamic Power Management

The components contributing to total energy consumption can be either static or fixed energy consumption components, e.g. networking components or dynamic energy computation components e.g. servers , processors etc. The static power management techniques can reduce the total power consumption permanently by modifying the circuit design. Some modern low-power circuit uses dual-voltage for critical and non-critical paths. In some circuits, the different threshold voltage transistors are used for saving power. Another very popular method of the static power management is to shut off the current to that part of the circuit which is not in use. [32]. Likewise, thermal aware static management techniques are based on chip design and floor plan, etc. Floor planning consists of the structural planning of the empty room, e.g. windows, support columns, etc. [38] Equipment planning is a blueprint of placing the servers and the cooling equipment. Proper floor planning can reduce the energy consumption significantly. [33]



Category	Power Draw*
Computing	588 kW
Lighting	10 kW
UPS and distribution losses	72 kW
Cooling power draw for computing and UPS losses	429 kW
Building switchgear/MV transformer/other losses	28 kW
TOTAL	1,127 kW

Figure 2.12 Power consumption patterns [30]

Power and thermal management can be done dynamically using resource allocation algorithms in such a way that the total power consumption can be reduced by maintaining the healthy temperature. A lot of research has been done and is going on for designing and developing a wide range of algorithms for energy efficient computing dynamically. Dynamic energy efficient algorithms are based on optimum utilization of the resources in the data center. Virtualization played an important role in the optimum resource utilization. [36-37]. It is being used by a large number of organizations to reduce power consumption and cooling needs. It is a combination of software and hardware that creates virtual machines that allow a single machine to act as multiple machines resulting in scalable infrastructure on demand. Nearly all resources can be virtualized ranging from the server, CPU, desktop, storage, network, etc. No doubt using virtualization techniques resource utilization is increased which results in energy efficiency computing in data centers; but to gain more profit and for cost-effective computing, a lot of organizations are migrating to the cloud. To fulfill a large variety of scalable demand, the service provider companies are increasing their infrastructure. In such scenario energy efficient computing is a challenge. [34] On the basis of the analysis done by Emerson network power shown in Figure 2.12, the computing equipment and the cooling equipment of the data centers are consuming nearly equal ratio of total power consumption, so the algorithms should also focus on either the cooling management or on the optimum utilization of the resources. We have selected the resource utilization optimization for contributing in the reduction of the total power consumption resulting in energy efficient computing.

The dynamic resource allocation algorithm is one which is able to deal with dynamic and uncertain conditions, e.g., failure of network link or CPU hosting application, availability of bandwidth, memory and storage space, etc. Considering these uncertainties the dynamic resource adaption algorithms can be categorized into three categories 1) predictive 2) reactive and 3) hybrid

Predictive resource allocation algorithms dynamically predict the relationship between quality of service (QoS) and the level of energy efficiency, hardware resource allocation and workload patterns to optimize the resource allocation. The predictive algorithms are based on the past knowledge of the resource allocation patterns. Various Statistical and machine learning algorithms have been proposed in the literature to predict the behavior of the resource allocation in the future based on the past resource allocation pattern knowledge. The past resource allocation pattern is derived from the monitoring service. The task of the monitoring service is to (i) Ensure the cloud resources and application operations at peak energy efficiency levels; (ii) Identifying the deviation in the energy efficiency of the resources and QoS of hosted applications; and (iii) Track the malfunction of resources and applications. [39]. While performing the experiment we identified the limitation that; machine learning algorithms are based on the past resource allocation history and sufficient workload and the resource allocation history should be available to train the machine learning technique to get the desirable results. Apart from this, the algorithms may fail in the situation when the new application is deployed, having no resource allocation history or do not follow any specific distribution. Predictive algorithms are also expensive in terms of space complexity as well as time complexity. [40].

Reactive algorithms are based on the threshold-based approach. These techniques are based on continuous monitoring the state of the cloud resources and pre-configured corrective actions are triggered when a specific event occurs. For example utilization of the CPU increases certain threshold. Monitoring service also plays an important role in the effectiveness of the reactive algorithms. Monitoring service runs continuously for a specified time period and based on the feedback the reactive algorithms change the resource allocation. The advantage of the reactive algorithms is that these algorithms do not consider any resource allocation history as the basis of the algorithm. The limitation of the reactive algorithms includes 1) Lack of unpredictability 2) instability 3) High provisioning cost. [41]

To overcome the limitations of both the algorithms, **hybrid approach** for resource allocation can be applied to the data centers. The reactive algorithms come under the category of computational algorithms and are time-consuming. The predictive algorithms preserve long-term workload statistics besides predicting and allocating for the next few hours. Hybrid algorithms combine the advantages of both the techniques and accomplish considerable improvements in 1) Service level agreements 2) Reduction in the provisioning cost 3) Energy efficient allocation. [42]. Hybrid algorithms outperformed both the above techniques (predictive and reactive) when performance, power consumption and the number of changes required in the resource allocation is considered. [41] we have also proposed the hybrid algorithm which is based on support vector machine (predictive) and CPU utilization threshold (reactive algorithm).

2.4.3 Virtual Machine Allocation

VM(Virtual Machine) allocation is the process of selecting the appropriate host for allocating the virtual machines based on the QoS (Quality of Service) constraints. The VM allocation constraints can be 1) memory aware 2) energy aware 3) cost aware 4) network aware 5) security aware 6) scheduling time aware etc... When a new request arrives, the service request monitor and the admission control policy service verify the workload submitted by the user and forwards the workload to the scheduler. The scheduler takes a decision regarding the placement of virtual machine that the placement will be simply a placement or VM consolidation needs to be required to place the VM at the appropriate physical machine. If no physical machine is available the workload is shifted outside the boundaries of the cloud. Other cloud providers are attached to the cloud using a communication interface. The task of Energy efficient VM placement is to decrease the running physical machines under reliability constraints [76]. The goal of an efficient virtual machine allocation algorithm is to maximize the utilization of the available resources [75]. We can loosely group cloud resource management policies into five classes:

A. Admission Control Policy
B. Capacity Allocation
C. Load Balancing and energy Efficiency

D. Minimizing Cost
E. Quality of service

Admission control policies are high-level system policies that prevent the system from accepting the additional workload that cannot be executed without affecting the QoS. These kinds of policies works on the principle that the additional workload acceptance may lead to performance decrease resulting in inefficient computing. So the algorithm based on the admission control policies, monitors the data center status and the only accepts the workload that can be appropriately allocated to the data centers and can meet the deadlines for the same.

Capacity allocation means virtual machine allocation. Virtual machine allocation in the federated cloud is a challenging task. While allocating the VM the basic criteria for shifting the VMs to supporting cloud vendors is the network latency. If the network latency is high then there will be chances of the SLA violation. Selecting an appropriate host in federation requires you to search a large space when the state of individual systems is changing so rapidly. Load balancing and energy optimization is correlated and affects the cost of providing the services.

Load Balancing and energy efficiency factors are attracting major attention these days because of the global warming. Energy efficiency can be attained by managing the load in such a way that the active servers can be reduced so that remaining servers can take a power nap to save the electricity. This type of algorithms has the cascading effect if the electricity consumption is reduced the cost will automatically reduce and also carbon emission will be less resulting in green computing.

The cloud computing paradigm provides the services on demand as pay per use basis, so the basic objective of the paradigm is **minimizing the cost** for cloud users as well as the cloud providers. The cost can be minimized by efficient computing so that SLA violations can be prevented and the energy consumption can be reduced to cut the cost. Data centers can use the natural ways for cooling services. In this way the electricity bill will be reduced and also the carbon emission will be less which is also a challenge in cloud sustainability.

The Quality of service is defined in the SLA document between the cloud provider and the vendor. Capacity planning, minimizing cost are important factors to be stressed upon but to save electricity and to reduce the cost the cloud vendors can't compromise with the QoS. Every VM allocation algorithm must ensure the QoS apart from the factor on which the main stress is. Like, cost aware VM allocation will try to allocate the VM in such that the total cost will be reduced, the energy efficient VM allocation will concentrate on the energy efficiency factor, but all the algorithms are required to ensure the QoS factor otherwise the complete cloud computing paradigm will collapse. Resource management strategies often jointly target performance and power consumption.

VM placement process can take various factors into considerations like memory, energy efficient, CPU performance, network and security based allocation. We have selected the most prevailing factor i.e. energy efficient VM placement. Scheduling delay is also an important factor when allocating the VM in federated clouds. So we have considered both the factors and proposed an efficient algorithm which reduces the energy consumption as well as the scheduling delay. Energy efficient VM Allocation allocates the virtual machines to the physical machines so the power utilization can be minimized. Various virtual machine allocation algorithms have been proposed; Random Striping, Round Robin, Bin packing, free CPU count, free CPU ratio and watt per core etc. are some of them [66]. Striping discards all the hosts that do not have the required number of resources to allocate the virtual machine. The next step is to find the host with the least number of virtual machines. Random algorithm selects randomly the resources available across the zone. The advantage of this algorithm is that the time complexity is very low because there is no need of pre-processing or analysis is required. Round robin discards the physical machine which was selected in the previous allocation. [45] The first fit algorithm selects the first resource which fulfills the virtual machine requirements. The time complexity of the first fit algorithm is also low as no pre-processing is required. [46]. In contrast best-fit algorithm sorts the VMs in order of increasing CPU share and sort PMs in order of increasing available CPU share. The next step is to search a PM such that (Required VM = threshold utilization- VM allocation) [48]. In the modification of best fit allocation algorithm denoted as Power Aware best fit decreasing algorithm sorts all the VMs in the decreasing order of their current CPU utilization and allocates the host that provides the least increase of the power consumption caused by the allocation [49]. Watt per core is an energy efficient algorithm executes in two phases; In the first phase, the algorithm discards the physical machines having insufficient resources and in the next step selects the host which is taking minimum additional watt per core resulting in shrinking the overall power consumption. The Cost per core policy also minimizes the cost estimated by searching the host that would capture a minimum additional cost per core [47]. An energy efficient algorithm called "OptSched" is proposed in [56]. This scheduler overcomes the inefficiencies of the default schedulers of Eucalyptus, OpenNebula and OpenStack. The set of VMs and the hosts are divided into 2 categories 1) timed instance 2) on-demand instance. Timed instances and the on-demand instance of VMs occupy a non-intersecting set of timed and on-demand set of hosts. The algorithm selects the powerful hosts first for allocating the VMs and the less capacity; the physical machine is used when the VM does not fit in the PM capacity [56]. Bin packing based virtual machine algorithm discards all the hosts that do not have available resources to allocate the virtual machine. In the next step, the host having a maximum number of the virtual machine is selected to allocate the virtual machine if that host is having the sufficient resources [46]. The Bin packing problem is to pack different volumes of objects into a finite number of bins or containers that minimize the number of bins used [44]. Multidimensional Bin packing modified multidimensional Bin packing, parallel implementations of the bin packing algorithms have also been proposed in the literature [50-54]. Some of the researchers argue that the Bin packing models are not always sufficient because all the VM characteristics can't be modeled along the bin dimensions. For example, VMs only exists during its execution period. Modeling these time factors are beyond the classical Bin packing formulation [50]. Thus, VM Allocation can be considered as online Bin packing problem because, in online Bin packing algorithm, each object is assigned without knowledge of the next pieces. An approximation algorithm is designed that uses the minimum possible number of bins irrespective of the input [55]. The authors in [58] have proposed a multi-dimensional space partitioning model for VM allocation. The complete space is divided into 3 types of spaces 1) Acceptance Domain 2) Safety Domain 3) Forbidden Domain. In Acceptance domain, all D-dimensional resources are nearly exhausted and there are few resource fragments in the physical machine. This is the idle state in which if the virtual machine is allocated the total energy efficiency can be maintained. The second domain is the Safety domain in which there is no obvious uncertainty of D-dimensional resource utilization. This is a balanced state in which, if the virtual machine is allocated to the physical machine there is little increase in total energy consumption. The third domain is the forbidden domain that indicates that there is an obvious uncertainty of D-dimensional resource utilization and this case should be avoided. This algorithm is performing better that first fit algorithm and saves 15% more energy as compared to a first fit algorithm. Some studies [63-65] presents the techniques like voltage settings, processor speed and turning off the display, sleep mode. But these techniques cannot achieve substantial energy efficiency since the energy saved by these techniques is very less as compared to powering on a server. DVFS (Dynamic voltage and frequency scaling) technique is similar to these techniques. This technique dynamically alters the voltage and frequency levels of the CPU to the lowest possible level while maintaining the performance levels [66]. We have surveyed various VM allocation algorithms in the federated environment of Reservoir, open Cirus, clouDiA etc. reservoir solves the VM allocation algorithm by dividing the problem into the master problem and the sub-problem. First, the master problem is solved while considering less number of variables using linear programming; then the sub problems are solved by selecting the additional constraints. This process of VM allocation is called column generation. [79]. clouDiA allocates the VMs based on the communication graph in the first phase and in the second phase the scheduler checks the pair-wise latencies to improve the placement plan. Open cirus allocates the VMs through the zoni client i.e. the administrator or the cluster manager. The client queries the zoni server for the resources. A list of suitable resources is provided to the client. The user chooses machine attributes and submits a request for the resources for specified period of time. zoni processes the request and send it to Tashi cluster manager to move VM off from the selected node so that new allocation can be done.[79]

2.4.4 Virtual Machine Consolidation

Dynamic consolidation of virtual machines is a promising method of gaining energy efficiency. The idea is to pack the maximum number of VMs on a minimum number of physical machines. The process migrates VMs from one server to another in such a way that the idle servers can be switched off and the overall utilization of the data-center can be improved [64]. VM consolidation process is explained in Figure 2.13.

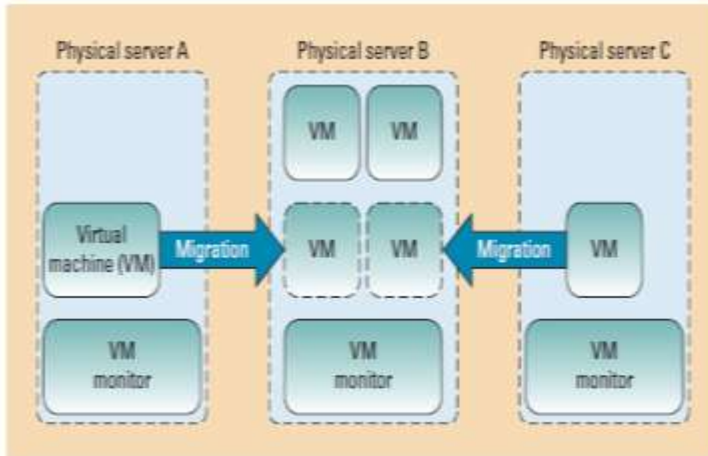


Figure 2.13: VM Consolidation Process [41]

The basic objective of the virtual machine consolidation is to monitor and select the physical machines which are having less number of virtual machine running on the server. Such virtual machines need to be migrated to other servers in such a way that number of inactive servers can be maximized. The idle server components are also consuming a significant amount of energy; so it is good practice to run the servers at their maximum possible level of utilization. The idle servers may be suspended to gain more energy efficiency benefit. Though VM consolidation increases the number of suspended servers; a variety of problems can exist to influence the cloud's energy efficiency and performance of the data center. The VM consolidation ratio should be considered while consolidating the virtual machines as high VM consolidation ratio without considering the resource optimization may lead to performance degradation and the increased execution time may decrease the energy savings originated from the server suspension. Another aspect is power cycling; Excessive power cycling a computer can cause thermal stress and can reduce the overall reliability of the server. Thirdly, turning resources off in a dynamic environment is risky from a QoS prospective. Due to the variability of the workload and aggressive consolidation, some VMs may not obtain required resources under peak load, so failing to meet the desired QoS [41]. Considering all the points described above controlled VM consolidation process should be followed in the datacenter to gain the maximum benefit. Virtualization and specifically live migration techniques played an important role in VM consolidation as the provisioning of multiple virtual machines on a single physical machine became possible after virtualization evolution. Most of the VM consolidation algorithms are based on optimization algorithms to decide which virtual machine should be placed on which physical machine. These optimization algorithms take input as the utilization of the resources and the total available capacity of the physical machine and try to allocate the maximum possible virtual machines on minimum possible physical machines.[65] [67]. The author in [73] allocates the virtual machines based on the parallel ANT colony optimization techniques. The relocation of VMs from one machine to another is possible because of live migration strategies. The author compares the effectiveness of the VM consolidation using two popular methods of Live Migration e.g. pre-copy live migration and post-copy live migration. The author concluded that post-copy live migration strategy is more reliable and suited for VM consolidation process. The post-copy live migration copies the memory pages after the execution host is switched and the migrating VM updates the pages only at destination node not at source and destination both, in-contrast to pre-copy live migration. With the help of post-copy migration algorithm the total transferred data is smaller than pre-copy and the total migration time is also less. [66]. All cloud vendors do not provide mature VM consolidation as explained in [68]. The author explains that there is a scheduler service which takes care of the scheduling activities. The load status of the servers can be fetched from the database. A hierarchical 2 phase scheduling algorithm based on the host overload and underload criteria is proposed in [57]. In the first phase the algorithm searches for the overloaded hosts and VMs from the overloaded hosts are migrated to the other hosts. In the second phase the underutilized PMs are selected and the VMs are migrated to another host and the PM is turned off to reduce the power consumption. [57]. OpenNebula cloud platform uses Free-CPU-Count policy for VM consolidation. In the first phase it discards all the physical machines which do not match the VM resource requirements. From the remaining hosts, the scheduler selects the hosts having maximum number of free CPUs available [47]. The modified variation of this algorithm is free CPU ratio. The procedure is same as above but here instead of free CPU the ratio of CPU is compared and the VM is allocated to the host having more free CPU ratio [47]. The author explains the phases of VM consolidation in [69]. The first phase of VM consolidation is data collection phase. This phase collects the details about server model, hostname, IP (Internet Protocol) address of each host and the physical location of the each host. On the software side this phase collects the data about the operating system, database management system and web server software etc. The second phase of the VM consolidation process is the live VM migration. This phase uses the data collected in phase 1. Live virtual machine migration can be performed by pre copy, post copy or hybrid live migration techniques. Out of these three post copy VM migration is most suitable for VM consolidation process [66]. The third phase is the post process after migration takes place. The process includes deleting all the remaining data from the source server and instance is activated and maintained by the controller. The decision, which VM should be selected, is also a challenging task. The cooperative VM consolidation technique is proposed in [61]. The author considers each physical machine in a data center as a peer connected with each other in an overlay network. Each peer in the network checks the status of its neighbor and triggers a decision function. The decision function performs the re-configuration based on two conditions. Both the peers checks for the possibility that if one peer node can contain or execute the load of both the peers, then the VMs from one peer are shifted to another and the idle peer can be set to power saving mode. The second condition is if the re-distribution of VMs increases the CPU and memory utilization of the peers then re-distribution of the VMs takes place [61]. Another VM consolidation approach based on Reinforcement learning (RL) is proposed in [62]. The algorithm is based on the learning agent who learns through Q-learning technique. The Q-learning technique learns on-line through experience from the environment and utilizes its knowledge for VM consolidation without prior knowledge of the environment. VM consolidation packs the maximum virtual machines in minimum possible physical machines under reliability constraints. But the idle servers are also consuming a considerable amount of energy that should also be reduced to manage the total power consumption of the data centers. By decreasing the power and frequency scaling of a multi-core processor the considerable amount of energy can be saved. The author also analyzed that instead of using idle state of the component say hard disk; it can be kept in standby or sleep mode to reduce the power consumption. Likewise the power consumption through fan can be reduced by reducing the RPM (rotation per minute). The more the rotation the more power it consumes [60]. The same kind of work is proposed in [59] where the author designed an algorithm to reduce the total power consumption by reducing the energy usage in the idle state of the server. Whenever the server completes all the assigned work, it transitions from an active state to a nap state. Nap state means the power consumption is extremely low and processing also not possible in this state. To perform the computation, the system components signals the arrival of the new task, the system awakes and transition back to the active state. When the work is completed, the system returns back to the nap state [59]. To predict the system behavior and to analyze the workload pattern various statistical and machine learning algorithms proposed in the literature e.g. Linear Regression, Median Absolute Deviation, Interquartile Range, Bayesian neural networks, K nearest neighbor regression, etc. CloudSim is also using statistical methods to predict the host overload and underload detection. The CPU utilization is considered as a univariate time series and then by predicting the future values using time series analysis methods, the host overloads and underload status can be determined. Various methods have been proposed in the literature out of which ARIMA (autoregressive integrated moving average model) outperformed all the traditional statistical methods. But ARIMA is not suited for the every kind of time series because the model is parametric and assumes that the time series being forecasted will be linear and stationary. ARIMA will not be suitable for the cloud workload pattern because the cloud workload is extremely non-stationary. Therefore, Artificial Neural Network (ANN) supported multivariate analysis has become a prevailing and admired tool in recent years. As compare to ARIMA neural network is more effective for non-stationary time series due to its unique non-parametric, non-assumable, noise-tolerant and adaptive properties. Neural networks are universal function approximators that can map any nonlinear function without a priori assumptions about the data [72]. However the main issue with neural network is the problem of over fitting. The neural network combines the unwanted noise with the useful information resulting in poor level of generalization. To combine the advantages of both, hybrid techniques also been proposed in the literature [71]. Support vector machine became popular because it improves the generalization property of the neural networks. It is a new way of training method which minimizes the structural risk minimization (SRM). We have proposed Support vector machine based host overload and underload detection method for efficient VM consolidation. The advantages of using SVM are underlined as 1) SVM provides a good generalization on the proper selection of c and r parameters. 2) There is no need of human experience or assumptions about the functional form of the transformation required to make the data linearly separable, the kernel implicitly contains the transformation. 3) SVM delivers more unique solution as compare to neural network, which has multiple solutions associated with local minima.

2.5 Monitoring and Metering Energy Efficiency

The emergence of Cloud computing has drastically changed the perception towards infrastructure, software delivery and development models. It is based on service oriented architectures and provision services over the Internet. Industries use a variety of metrics for measuring the performance and the energy efficiency of the cloud computing systems. The metrics for measuring the energy efficiency are shown in Table 2.2.

Table 2.2

Monitoring and Metering Energy Efficiency Metrics [58]

Metric	Explanation	Formula
Power usage Effectiveness (PUE)	It is the fraction of total energy Consumed by the service of a data centre to the total energy consumed by IT equipments.	$PUE = \frac{Total\ Equipment\ Energy}{Total\ Facility\ Energy}$
Carbon Usage Effectiveness (CUE)	It is a calculation of green house gases (CO_2, CH_4) release in atmosphere by the data centre	$CUE = \frac{Total\ CO2\ emission\ from\ total\ energy\ used\ for\ service\ of\ Datacenter}{Total\ energy\ consumed\ by\ IT\ equipments}$
Water Usage Effectiveness (WUE)	It is calculation of yearly water used by data centre like for cooling, energy Production.	$WUE = \frac{AnnualUsageOfWater}{TotalEnergyUsedByITEquipment}$
Energy Reuse Factor (ERF)	It calculates the reusable energy Like hydro power, solar power etc used by data center.	$ERF = \frac{\mathrm{Re}usedEnergyUsage}{TotalEnergyUsedByITEquipment}$
Energy Reuse Effectiveness (ERE)	It is a parameter for measuring the profit of reuse energy from a data centre.	$ERF = \frac{TotalEnergy - \mathrm{Re}usedEnergy}{TotalEnergyUsedByITEquipment}$
Data centre Infrastructure Efficiency (DCiE)	This factor is used to calculate The energy efficiency of a data Centre.	$DCiE = \frac{TotalITEquipmentPower}{TotalFacilityPower} * 100$
Data Centre Productivity (DCP)	It calculates the amount of useful work done by data centre.	$DCP = \frac{TotalUsefulWork}{Total\ \mathrm{Re}sourceUsedToDoTheWork}$

Compute Power Efficiency (CPE)	It determines the total amount of power is truly used for computing.	$CPE = \frac{ITEquimentUtilizationEnergy}{PUE}$
Green Energy coefficient (GEC)	It measure the amount of green energy used to provide services to data centre.	$GEC = \frac{GreenEnergyConsumed}{TotalEnergyConsumed}$
Space, Wattage and Performance (SWaP)	It is used for work out the space and energy required by the data centre.	$SWaP = \frac{Performance}{Space * power}$
DataCentre Energy Productivity (DCeP)	It calculates the quantity of useful work done by data centre as compare to total energy consumed to make this work.	$DCeP = \frac{TotalUsefulWorkDone}{TotalEnergyUsedToDoThisWork}$
Energy and SLA violation metric(ESV)	It calculates both energy consumption and levels of SLA(service level agreement) violations. *PDM=Performance Degradation Due to VM Migrations* *OTF= The time fraction during Which active hosts Experienced the 100% CPU Utilization*	$ESV = Energy * SLAV(SLAviolationMetric)$ $SLAV = OTF * PDM$ $OTF = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}}$ $PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{D_{dj}}{D_{rj}}$ T_{si} =time duration when host experienced maximum utilization T_{ai} = total duration when VM was allocated to host D_{dj} =performance degradation because of migration D_{rj} = total capacity requested by VM.

CHAPTER 3

Role of Virtualization in Efficient VM Allocation

3.1 Introduction

Todays' IT firms strive for agility, efficiency and flexibility in the operations. To satisfy these demands the organization requires to utilize the available infrastructure capacity flexibly and efficiently. Evolution in hardware and software technologies changed the way the infrastructure has been used in the past. The technologies like virtual memory create an illusion and executes an application that may not be in memory. [82]. Other techniques such as I/O device abstraction creates an illusion that each application is using all physical resources and even more, those are not physically existing. A Virtual Machine was originally defined by Popek and Goldberg [85] as an efficient, isolated duplicate of a real Machine, which allows the multiplexing of the underlying Physical Machine. Virtualization provides a great opportunity to build elastically scalable system because of which 50% of the hardware and operational costs of data centers are reduced and the energy consumption is also reduced by 80%. [89]. A recent survey shows that as computing demand increases, rather than adding additional server resources to existing data center facilities, over 60% of data center operators consolidated their workloads using virtualization technologies (encapsulated in virtual machines).[88].

3.2 Virtualization Techniques

Virtualization technology is one of the fundamental components of the cloud computing specifically in the case of IaaS(Infrastructure as a Service). It is an approach which refers to the abstraction of the computing resources to maximize the utilization. It offers many benefits such as server consolidation using live migration of virtual machines from one host to another, with minimum effect on the hosted services. Xen, VMware (VMotion) and Microsoft Hyper-V etc. support virtual machine migration on-the-fly. Live (Hot) virtual machine migration plays an important role in server consolidation which is the requirement of today's data centers. Virtual machine monitor (VMM) (also called a hypervisor) directly manages the underlying hardware(processors and memory). The components of the virtual machine monitor included hyper call interface (entry point to all partitions to run sensitive instructions), memory service routines(works as an interface to all the partitions to access the memory), advanced programmable interrupt controller(manages the signals coming from the underlying hardware), scheduler(schedules the virtual processors to run on available physical processors), address manager(manages the virtual network addresses allocated to each guest OS), partition manager(responsible for creating, deleting and configuring partitions).[90,91] Based on the hypervisor installation the virtualization can be categorized as shown in Figure 3.1 [87]

3.2.1 Hardware Virtualization

3.2.2 Software Virtualization

3.2.3 Virtual OS/Container

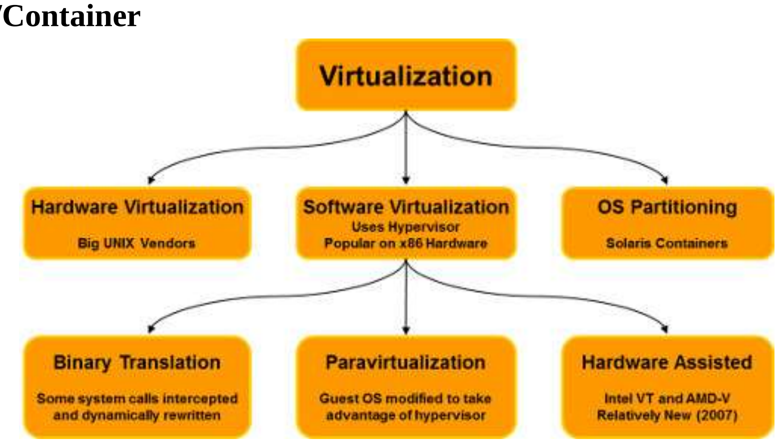


Figure 3.1 : Categories of Virtualization [92]

3.2.1 Hardware Virtualization:

In Hardware Virtualization, the hypervisor runs directly on the host's hardware to control it and to manage guest operating systems. A guest operating system thus runs on another level above the hypervisor. The task of this hypervisor is to handle resource and memory allocation for the virtual machines in addition to providing interfaces for higher level administration and monitoring tools as depicted in Figure 3.2(c). The hypervisors for hardware virtualization often referred as "bare-metal", native and embedded hypervisors. Type -1 or the bare-metal hypervisors' are gaining more attention as type-1 hypervisors are more secure, provides high availability and performance than type-2 hypervisors. Oracle VM Server for SPARC, Oracle VM Server for x86, the Citrix XenServer, VMware ESX/ESXi and Microsoft Hyper-V 2008/2012 etc are examples of type-1 hypervisors.[90-91]

3.2.2 Software Virtualization

In software virtualization, the hypervisors run within a conventional operating system environment. With the hypervisor layer as a distinct second software level, guest operating systems run at the third level above the hardware as shown in Figure 3.2 (b) . The examples of the type-2 hypervisor are VMware Workstation,VMware Player, VirtualBox and QEMU etc. Software virtualization can be achieved by :

A. Binary Translation

Though various virtualization technologies have been evolved for the optimum resource utilization to save power and energy cost. Some instructions like context sensitive can't be virtualized. To overcome this problem binary translation can be used. The sensitive instructions in the binary of Guest OS are replaced by either Hypervisor calls which safely handle such sensitive instructions or by some undefined opcodes which result in a CPU trap. Such a CPU trap is handled by the Hypervisor.

B. Para-virtualization

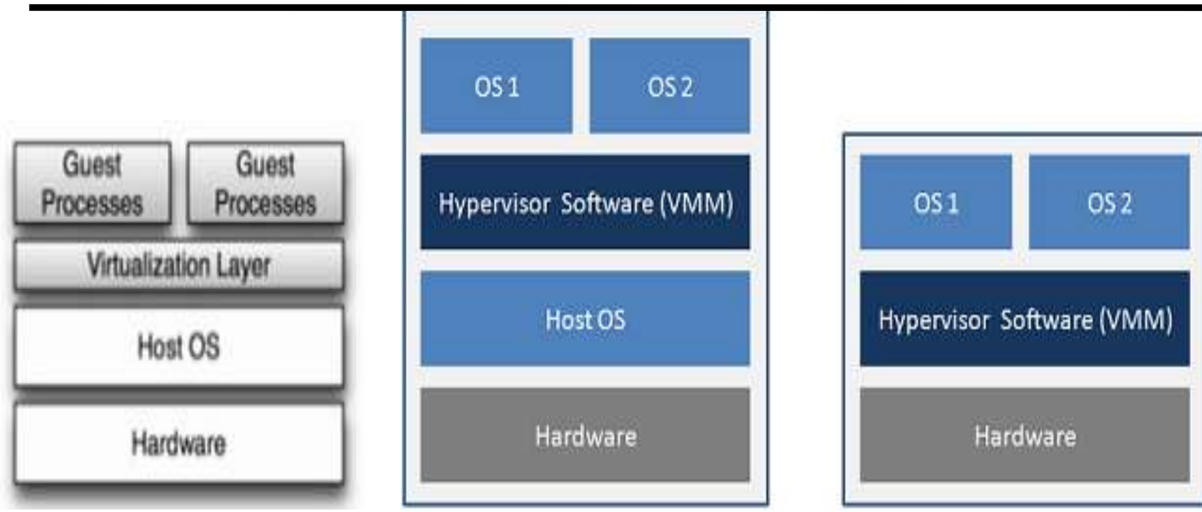
Para-virtualization is a lightweight and efficient virtualization technique introduced by Xen. In this technique, the guest operating system kernel is modified to replace the non-virtualizable instructions with hyper calls that directly communicates with the hypervisor and simply issues a command directly to the host operating system instead of an interrupt or trap as in full virtualization. This also includes memory and thread management as well, which usually require unavailable privileged instructions in the processor.

C. Hardware Assisted virtualization

Hardware vendors have started accepting virtualization and are developing new features to simplify virtualization techniques. Full virtualization and the para-virtualization is required because there are some instructions in the instruction set that cannot be virtualized . To overcome this problem the hardware vendors have made modifications to the hardware such that a new CPU execution mode feature is added that allows the virtual machine monitor to run in a new mode below ring 0. The privileged and sensitive instructions trap the hypervisor directly removing the need for either binary translation or paravirtualization.

3.2.3 Virtual OS/container:

virtual OS/containers provide a virtual environment in which the host operating system is partitioned into containers or zones (e.g., Solaris Zones, BSD Jail) as shown in Figure 3.2(a) . This approach can also improve performance because there is just one operating system taking care of hardware calls. A disadvantage of container-based virtualization, however, is that each guest must use the same operating system the host uses.



(a) Container Based Virtualization (b) Software Virtualization (c)Hardware Virtualization

Figure 3.2 : Types of Virtualization[87]

3.3 Role of Virtualization in VM Management

Virtualization has now become extremely popular and is largely used, especially in cloud computing. The primary reason for its wide success is the elimination of technology barriers that made virtualization not an effective and viable solution in the past. The most relevant barrier has been performance. Today , the capillary diffusion of the Internet connection and the advancement in the computing technology, have been made virtualization an interesting opportunity to deliver on-demand IT infrastructure and services. The role of virtualization in VM management is :

3.3.1 Optimum Resource Utilization:

A traditional data center with a single workload runs at only 10 to 15% of the total processing capacity wasting the remaining 85 to 90% compute capacity. By applying the virtualization techniques multiple virtual machines can reside on a single server and the available computing power, storage space and network bandwidth can be used much more efficiently. Rather than purchasing the physical hardware virtual machines can be created on the existing hardware to increase the utilization of the existing resources up to 70 to 80%. Virtualization provides an isolated environment to each user working on a separate virtual machine. Programmers can run software on independent virtual machines without worring about affecting other applications. Figure 3.3 shows that without virtualization the operating system was bound to the physical machine on which it was running. But after virtualization, the connection between the operating system and the physical hardware is interfaced by virtualization layer. With the help of virtualization layer or more specifically server virtualization the resource are utilized up to their optimum level as one server can host multiple virtual machines.

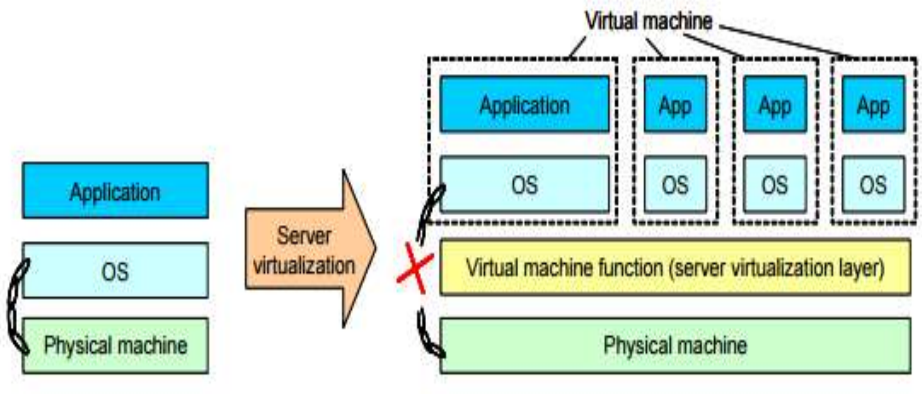


Figure 3.3 Server Virtualization[94]

3.3.2 Portability

Open virtualization Format(OVF) assists flexible, secure packaging and distribution of the virtual machines across multiple platforms. With the help of OVF, the vendors can deploy the virtual machines that follows the OVF specifications on any virtualization platform. OVF is not tied to specific host platform, virtualization platform or guest operating system. Though OVF does not guarantee universal portability across all hypervisors but research is going on for eliminating all barriers for complete portability. [93]

3.3.3 Green Computing

The primary goal of the virtualization is to maximize the utilization of the resources including energy. Rather than powering thousands of machines, most of which are running at very low utilization rates, server virtualization allows multiple virtual machines to resides on a single server so that the active number of servers are reduced switching off the remaining one resulting in lowering energy consumption. The downtime is eliminated because of the live virtual machine migration. The workload can be balanced dynamically. With the help of storage virtualization, the appropriate storage device can be selected for storing the data i.e. frequently accessed data and applications are stored on fast and costly storage devices consuming more power, less frequently used data is stored on less expensive devices that consume less power.

3.3.4 Cost Effective Computing

Traditionally, IT organizations have been constrained to offer an entire server to each workload to ensure stability and reliability. With the help of virtualization, multiple virtual machines can run on a single server allowing multiple workloads to be run on a single system in isolation and independence from each other resulting in reduced hardware cost, data center power consumption and cooling costs, data center floor space requirement and costs related to maintenance.

3.3.5 Consolidation

Server consolidation is a method of combining workloads from separate machines or applications into a smaller number of systems and applications. It involves decoupling a workload and its data from the functional details of the physical platform on which it is hosted. For organizations that own hundreds or thousands of servers, consolidation can dramatically reduce the need for floor space, A/C power, and co-location resources. This means the cost of ownership is reduced significantly, since less physical servers, floor and rack space are required, which in turn leads to less heat and power consumption, and ultimately a smaller carbon footprint.

3.3.6 Disaster Recovery & Maintenance

Because of server virtualization and the open virtualization format virtual machines can be transferred from one server to another with minimal downtime. It is also used for load balancing by shifting virtual machines from overloaded servers to other server in order to optimize the utilization of available CPU resources. One of the significant advantage of the live migration is the fact that it facilitates upbeatable maintenance. If an imminent failure is suspected, the potential problem can be resolved without disruption of running service .

3.4 Virtualization players

There are three major x86 server virtualization players: VMware, Citrix, and Microsoft. There are also vendors who deliver virtualization-specific to their own compute platforms such as HP.

A. **VMware:** VMware is the best-established virtualization provider with a large installed base of server virtualization customers. VMware's flagship product, ESX, uses hardware emulation.

B. **Citrix:** Citrix, which offers a server virtualization product called XenServer, uses para-virtualization. The privileged guest (called the Control Domain in Xen parlance) and the Xen hypervisor work in tandem to enable guest virtual machines to interact with the underlying hardware.

C. **Microsoft:** Microsoft recently began shipping its new server virtualization product, called Hyper-V. Its architecture is very similar to that of Xen. Instead of the term domain being used to refer to guest virtual machines, Hyper-V refers to them as partitions. The counterpart to Xen's Control Domain is called the Parent partition.

The current version of Hyper-V is high-performance, but as a relatively new offering, its functionality is somewhat less than the more-established Citrix XenServer and VMware ESX products. In particular, Hyper-V doesn’t currently have the ability to migrate virtual machines among physical servers, thereby reducing the ability to enable High Availability and Disaster Recovery. However, Microsoft is diligently working on improved functionality and is likely to offer a much richer product in the near future.

D. **Hewlett-Packard:** Hewlett-Packard provides a broad portfolio of virtualization products across server, storage, network, and client hardware, as well as management software, consulting services, and outsourcing services. It works closely with the other virtualization technology leaders to integrate these capabilities into deployment-ready solutions.

CHAPTER 4

Divided KD tree based Initial VM Allocation in Federated Clouds

4.1 Introduction

Cloud computing is a ubiquitous computing paradigm which provides on-demand services to the users with the help of Internet. To deal with the growing demand for the cloud services, the cloud vendors' established number of geographically distributed data centers. As the size of the data centers is growing with the increasing demand of the users, it becomes important to deal with the energy consumption issue at priority because huge energy consumption not only increases the maintenance cost of the data centers but also a sustainability issue. The massive amount of energy consumption from the data centers results in huge amounts of carbon emission which leads to climate change and a health-related risk to the society. Efficient virtual machine allocation is the core of the cloud data center. Virtual machine placement is a process of mapping virtual machines to physical machines. [95]. Virtualization played an important role in maximizing the resource utilization. Specifically, Server virtualization smashes the inflexible one-to-one link between the operating system (OS) layer and the physical machine layer. It enables the flexible construction of the virtual servers that can migrate from one physical machine to another with almost no hardware limitation. [96].

4.2 Energy-aware Cloud Federation Architecture

The cloud computing paradigm is evolving from centralized control over the resources under the administration of the single cloud provider to interconnections between multiple cloud providers to gain economic profits as well as maximization of the resources. Cloud federation is a preferred solution to lower down the adverse effects of vendor lock-in. It comprises services from the different providers combined in a single pool providing complementary resources, resource migration and resource redundancy services. [98]. Energy-aware cloud federation architecture is shown in Figure 4.1

4.2.1 Application Layer

The application layer provides services to the users to interact with the federation. The user sends a new request for executing applications in the federated cloud through the command line or web services. They can execute single applications or workflow by choosing among available services. Job controller collects all the requests and sends the input data to the core layer. This layer also shows the status of the running applications to the user.

4.2.2 Core Layer

The core layer is responsible for managing the federation environment. The task list of the core layer includes: 1) Identification of the new providers with their hardware and software resources 2) Service Level Agreement monitoring 3) Energy aware scheduling 4) Reconfiguration of VM for load balancing 5) Storage and management of input/out files etc.

A. Service Request Examiner and admission control

Service request examiner as the name suggest decides whether to accept the service request or not based on the status information provided by the discovery service as well as the virtual machine monitor. The main focus of this component is to prevent any overload on the resource. Service requests are rejected to prevent any overburden on the resources available.

B. Discovery Service

The discovery service manages the information of all the cloud providers participating in the federation and consolidates information about storage, processing capabilities, network latency and the availability of the resources. The cloud providers continuously update the data structure that keeps the resource information of the providers participating in the federation. This information guarantees the correct and updated task execution on the federated clouds.

C. Monitoring Service

The monitoring service verifies that the requested service is available by the cloud provider. If the service is not available in the current service provider; it searches for another cloud in the federation which fulfills the request. It receives the task to be executed and pass them to scheduling service that distributes the tasks guaranteeing that all the tasks successfully finish its execution. Monitoring service also gets the task execution status from the federation and passes this information to the users so that users can track the status of the each submitted task.

D. Storage Service

The storage service decides how to distribute and replicate data among the cloud providers participating in the federation. The storage service takes the federation participants information from the discovery service since the discovery service knows the actual storage conditions of each provider integrating the federation. Replication is done copying data to more than one cloud in order to ensure recovery in case of failure.

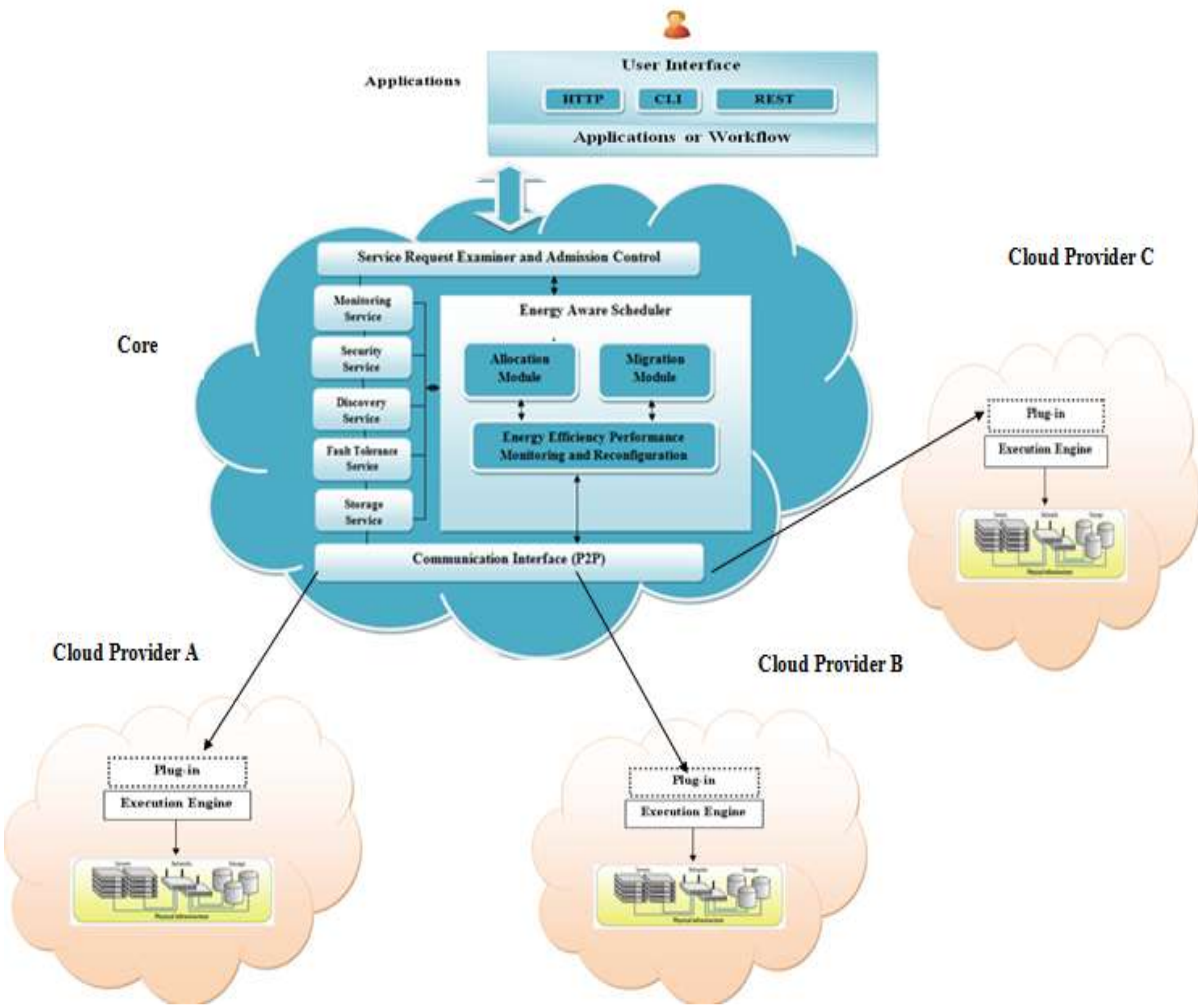


Figure 4.1: Energy-aware Cloud Federation Architecture

E. Fault Tolerance Service

Fault tolerance services guarantee the reliability of the federation. In a cloud environment when a failure occurs the fault tolerance service recovers the system from the failure. Therefore, a fault tolerance service is an essential part of our federated cloud and has the objective of providing high availability and resiliency against periodic or transient failures.

F. **Energy Aware Scheduler**

Energy-aware scheduler has 3 modules. 1) Allocation module 2) migration module 3) monitoring and re-configuration module. Initial VM allocation is the task of the allocator module. After initial VM allocation, the performance monitoring of the virtual machine starts and based on the host overload and underload events; the migration module migrates some of the VMs from the overloaded hosts avoiding the performance degradation and SLA violations in the federation.

4.2.3 Communication Interface

It provides an interface between the cloud service providers so that they can communicate with each other and can participate in the cloud federation.

4.3 Bin Packing Approach to VM Placement

Given a Bin S of size V and a list of n items with the sizes $a_1, ..., a_n$ to pack, find an integer number of Bins B and a $B - partition$ $S_1 \cup .... \cup S_B$ of the set $\{1.........n\}$ such that $\sum_{i \in S_k} a_i \leq V$ for all $k = 1, ..., B$.

A solution is optimal if it has minimal B. A standard Integer Linear programming formulation of the problem is:

$$Minimize \;\; B = \sum_{i=1}^{n} y_i \qquad \text{Eq..(4.1)}$$

$$\text{Subject to } \sum_{j=1}^{n} a_j x_{ij} \leq V y_{i,} \forall i \in \{1, ..., n\} \qquad \text{Eq.. (4.2)}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \forall j \in \{1, ..., n\} \;\; y_i \in \{0,1\}, x_{ij} \in \{0,1\},$$

$$\forall j \in \{1, ..., n\}, \forall i \in \{1, ..., n\}$$

Where $y_i = 1$ if Bin i is used and 0 otherwise $x_{ij} = 1$ if item j is assigned to Bin i.

The servers in the each data centers are considered as Bins in Bin packing problem. The objective of the Bin packing is to minimize the number of servers or Bins while arranging the VM or the items in the server/Bins. VM allocation problem adds 2 more constraints to Bin packing problem. The first constraint is that we can't place the VMs on top of another which is possible in Bin packing as items can be placed over the other. The second constraint is overlapping of VMs is not allowed. VM overlapping is possible only in the case of a shared memory in VMs. So the constraints on Bin packing problems are:

1) No VM can be placed on top of another.
2) Overlapping of VMs is not allowed except in the case of shared memory.

VM Allocation is also analogous to the floor tile placement problem because both the above constraints are fulfilled in tile placement problem. We can't place tiles on top of another. Also, no tile can overlap each other. So if we add these two constraints to the Bin packing problem, then it will be identical to VM placement problem in data centers.

4.4 System Model

The target system is IaaS federated distributed across multiple cloud providers. Each cloud participating in the federation communicates with each other through the communication link and has a large scale data centers represented by n heterogeneous physical servers. Each node in the system is defined by the number of CPUs, the size of the memory, bandwidth and the capability of the host in terms of millions of instructions per second (MIPS). The host machines are having SAN (storage area network) or NAS (Network attached Storage) to enable the live migration in clouds instead of having DAS (Direct attached storage). The multiple users based on their requirements submit a VM request through the command line or through a web interface. The VM request is defined by the CPU, memory, MIPS requirements. The service examiner and admission control service accepts or rejects the request based on the resource usage. If the request is accepted the discovery service providers the information of the free resource pool in a particular cloud. The cloud is selected based on the information updated by the cloud provider in the discovery service. After the cloud selection, the data center manager i.e. the second level manager selects the data center from where the resources can be fetched. The hosts in the datacenter are arranged in divided KD-trees. The search query is performed to fetch the appropriate best-fit host to allocate the host. The VMs are not categorized based on the workload means any VM can execute any type of workload if it is capable of containing the same.

The user and the cloud providers are binds with an agreement called SLA (Service level agreement). If the provider is not able to fulfill the service according to the standards specified in the SLA the provider pays the penalty to the user. The VM allocation problem is divided into 2 phases 1) initial VM allocation 2) VM consolidation. VM consolidation is further divided into 3 categories

A. Host overload detection
B. Host underload detection
C. VM selection

Host overload detection has a significant impact on the system performance as described in Section 5.3. The host overload detection is a challenging task. Various static as well as dynamic techniques employed to predict the overloaded host accurately. Based on the L3-managers the data center manager issues a VM migration request if the VM exists within the boundaries of a single cloud. If VM needs to be migrated to a host monitored under another cloud domain, the L2 manager sends a migration request to the L1 manager i.e. cloud manager to migrate the VM outside the boundaries of the cloud. The host with less utilization as compared to other hosts is selected as the underloaded host.

4.4.1 CPU Model

A physical server is equipped with n cores each executing m MIPS .In CloudSim, it is assumed that the multi-core CPU can be modeled as a single core CPU having MIPS. This is justifiable because the applications and the VMs are not attached with the CPU cores but can be executed on any core using time-shared scheduling policy. CloudSim supports two Scheduling policies 1) Space Shared 2) Time shared. Figure 4.2 explains the allocation of CPU cores to VMs as well as the task using both the algorithms. The only constraint that is applied is that the capacity of the virtual CPU core allocated to the VM must be less than or equal to the physical CPU core otherwise VM must be executing on multiple physical cores in parallel and that can't be assumed.

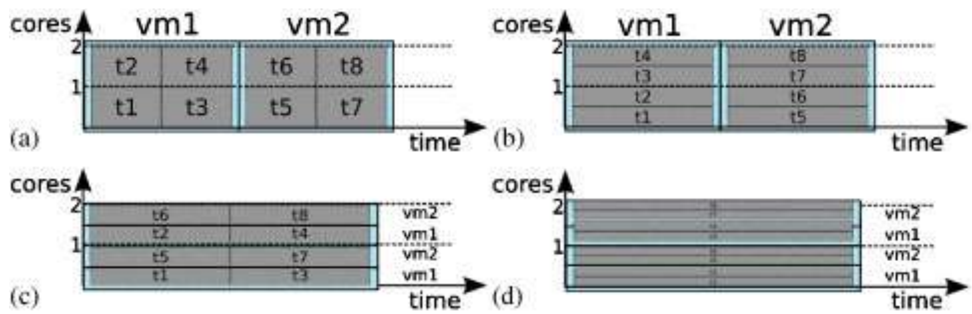


Figure 4.2 : (a) space-shared provisioning for VMs and tasks; (b) space-shared provisioning for VMs and time-shared provisioning for tasks; (c) time-shared provisioning for VMs, space-shared provisioning for tasks; and (d) time-shared provisioning for VMs and tasks.[109]

4.4.2 The Power Model

The recent research proved that power consumption by the server can be accurately described by the linear relationship between the CPU utilization and power consumption. [107]. CloudSim is using the real data of power consumption provided by the result of SPEC power benchmark published in February 2011. CloudSim simulates dual-core CPU with less power as lighter workload is required to overload a server and dual-core will be sufficient to simulate the multi-core CPUs.

4.5 VM Allocation Phases

VM allocation process is a subset of resource allocation which typically refers to allocation of virtual machines to physical machines which usually takes into consideration of CPU, storage and bandwidth requirements. The objective of the virtual machine placement is to find the best physical server to place the VM. The virtual machine allocation process is performed in 2 phases

4.5.1 Phase I-Initial VM Placement

Consider a federated cloud each having a set of data centers composed of a set of servers (physical machines) and a corresponding queue of client request for virtual machine allocation.[108]. Initial VM Placement can be considered as placing a VM with fixed resource requirements to n physical servers those are unoccupied or partially occupied while aiming the following objectives:

A. To minimize the total power consumption of the data center. As explained in Section 2.5 the cost of powering computing equipments is 52% of the total operational expenditure.
B. To minimize the scheduling delay so that the placement ratio can be improved.
C. To minimize the Service Level Agreement caused by virtual machine migration caused by VM consolidation.
D. We have designed a best fit initial virtual machine allocation algorithm based on divided KD tree which is selecting the best host to allocate VM while improving the placement ratio. The algorithm allocates the virtual machines in such a way that the active hosts are minimized consuming less power.

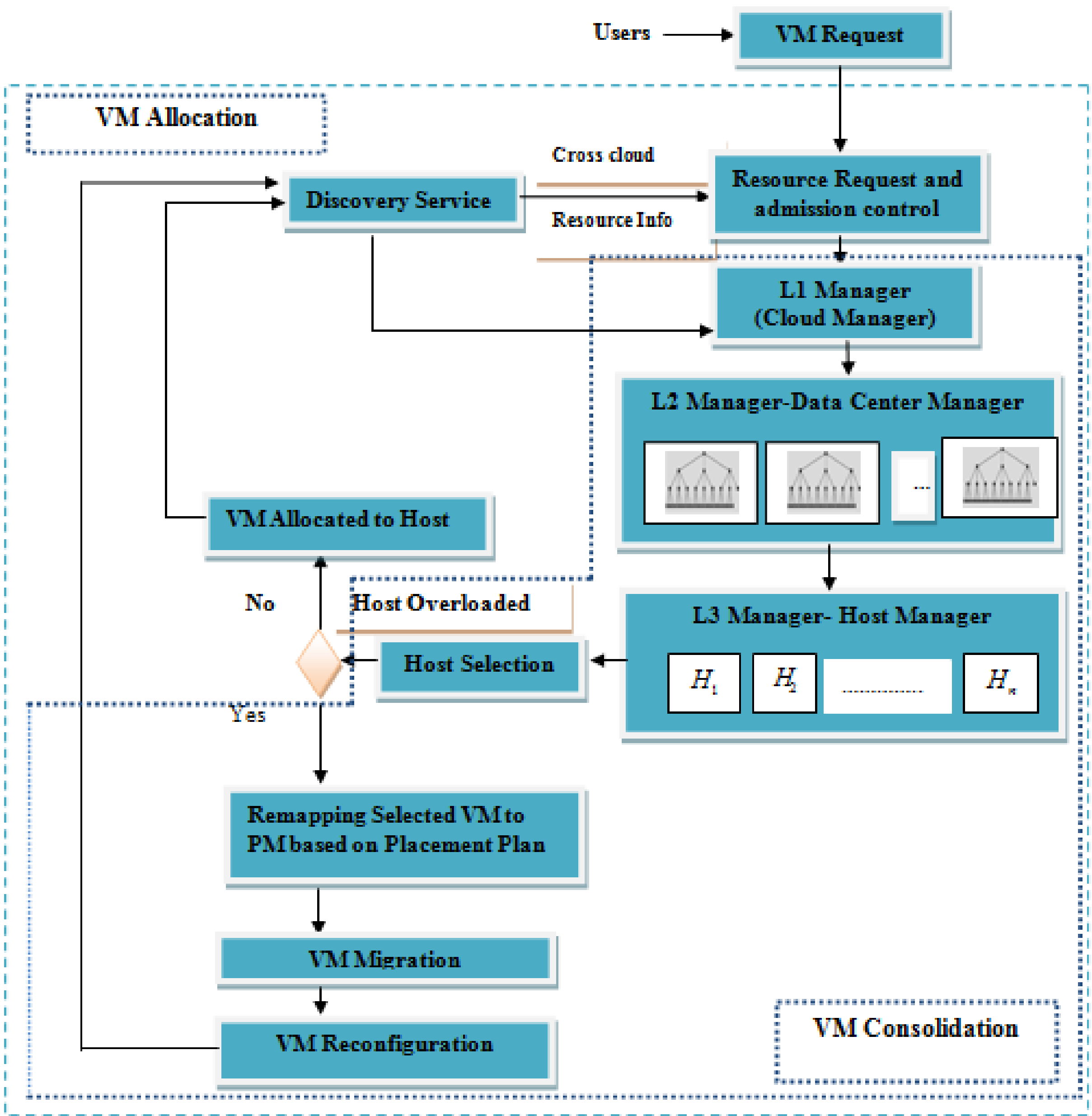


Figure 4.3: System Model of VM Allocation in Federated Clouds

4.5.2 Phase II-Virtual machine re-configuration with Consolidation

After initial VM allocation the second phase is to execute the optimization plan to consolidate the workload by re-configuring the Virtual machines. The objectives of the phase-ii are:

A. Deciding if the host is overloaded then select the virtual machines from the overloaded host those can be migrated to other hosts.
B. Deciding if the host is underloaded then migrating all the virtual machines from the host to shift the host to low power mode consuming less power.
C. Selecting the virtual machine from the overloaded host.

4.5.2.1 Host Overload Detection

Dynamic least square support vector machine based algorithm is proposed in this thesis to predict the overloaded host. Machine learning algorithms are subfield of Artificial Intelligence algorithms that are based on pattern recognition to enable the machine to learn and perform automated tasks and activities without human intervention. Statistical methods for predicting future values includes regression analysis, time series analysis, least squares, autoregressive moving average model and vector autoregressive model. Support vector machine algorithms overlaps statistical methods by providing the method to automate the information discovery process. Support vector machine is a neural network based method which over-performed the statistical methods and artificial neural network because SVM (Support Vector Machine) provides global optimum solution and doesn't even have the over fitting problem. As compare to neural network, support vector machine requires less training samples to make decisions.

Dynamic LS-SVM is a machine learning algorithm which outperformed all statistical and artificial neural network based algorithms presented in chapter 5. The algorithm is tested in SmartFED simulator based on CloudSim. The experiment results are presented in Section 7.2

4.5.2.2 Host Underload Detection

The Host underload detection algorithm is executed after the hosts overload detection and the VM migration phase. In this algorithm all the host utilization are arranged in order to get the host with lowest utilization in such a way that the VMs running on these hosts can be migrated to other host so that the host can be shifted to sleep mode. The host is shifted to the sleep mode when all the migrations are completed.

4.5.2.3 VM Selection for Migration

Once the overloaded host is detected the next step is to migrate some of the VMs from the overloaded host to another host in such a way that VM migration should not affect the QoS. We have used already implemented CloudSim methods for VM selection. The methods are

A. Minimum migration time

The minimum migration policy as the name suggests, migrates the VM requiring minimum time to migrate. The migration time is measured as the amount of RAM utilized to the VM divided by the spare network Bandwidth the host has.

B. Random selection

Random selection policy randomly selects the VM to be migrated from the host according to a distributed discrete random variable whose index i selects the i^{th} VM .

C. Maximum correlation

The maximum co-relation is based on the idea that the higher the correlation between the application running on an oversubscribed servers and the resource usage by the application, the higher the probability of the server overloading. So the maximum correlation algorithm selects the VMs for migration that have the highest correlation of the CPU utilization with the other VMs.

4.6 Proposed Model of Initial VM Allocation

Service oriented computing is increasing with cloud computing paradigm. It provides on-demand and scalable access to high-end computing capabilities to customers. However, the growing user demand drastically increased the energy consumption of datacenter, which has become a critical issue. High power consumption not only increases the energy cost but also contributes in carbon emission which increases a risk of sustainability to cloud computing paradigm. Thus, there is an urgent need for energy-efficient solutions to deal with the high energy cost and the sustainability issues. VM allocation problem is considered as Bin packing algorithm. The objective of the Bin packing problem is to pack the boxes in least possible bins; likewise in IaaS the VMs should be allocated in minimum possible number of bins so that active hosts can be minimized. To minimize the active hosts we have designed an algorithm based on Divided KD tree. Apart from this the scheduling delay will also be a major issue in the federated clouds as different cloud vendors with different policies will be participating in the federation. We have proposed a divided KD tree based algorithm for initial VM allocation.

4.6.1 Introduction to classical KD tree

KD tree is defined as an alternative to Quad-tree for storing the values or objects in multi-dimensional space. Quad trees were designed to develop an index structure in main memory. But KD tree is preferable over Quad-tree as Quad tree do not scale well in high dimensions and also the quad tree is not usually balance; the structure further depends on the sequence of the insertion of points. KD (k-dimensional) trees are binary search trees and space partitioning trees which represent a set of points in a multi-dimensional space. [99]. Due to the efficiency and the speed with which they can be traversed, k-d trees have been used in the variety of applications e.g. Fingerprint matching, floor layout, astronomy, ray tracing in graphical applications and the database applications etc. we have developed and algorithm which, for the first time, utilized space partitioning KD tree to efficiently allocate the virtual machines in the data centers.

The k-dimensional space is divided into a partition plane perpendicular to one of the co-ordinate axes. In 2-dimensional tree subdivision is done with the help of a line and in multi-dimensional tree a hyper-plane aligned to one-axis. [101]. Take an example of 2-dimensional tree, Let v is a value stored as a root of the tree. It splits the set of points into 2 subsets based on the value of the first co-ordinate of the points. The two subsets will be stored as two sub-trees. The values are inserted in the tree by comparing the values with the root, if the value is smaller that root is stored in the left sub-tree and the value greater than the root will be stored in the right sub-tree. If the splitting required second time it will be based on the second co-ordinate of the point. Third split will be on the basis of first co-ordinate again and fourth on the basis of second again and so on. Figure 4.1 shows the 2-dimensional KD tree.

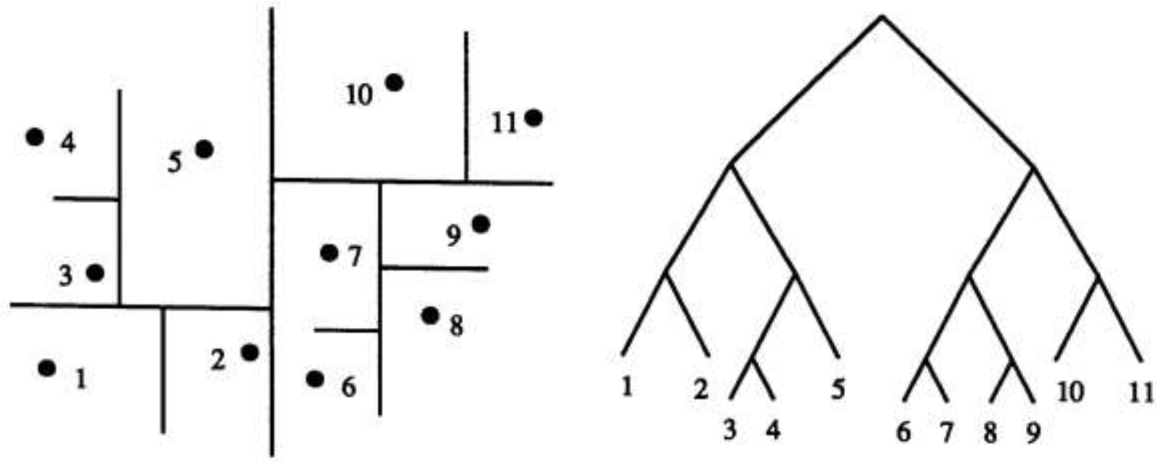


Figure 4.4: 2-d tree [8]

Issues with classical KD tree

There are various advantages of the KD tree e.g. exact match queries, nearest neighborhood queries etc. KD tree are proved beneficial in various database applications where data needs to be fetched based on the multi-dimensional data items. But the advantages of the KD tree are constrained with some issues given below:

A. Rebuilding is Time consuming:

When new point is inserted in KD tree, it may found either (i) empty child; in this case a single child is created. (ii) A child; then the new node is added as a second child. (iii) When the child node is full then splitting of node takes place and the rebalancing takes place. But there is no guarantee that a series of points added in this way will preserve the balance of the KD tree. Rebuilding operation for balancing the KD tree costs incremental additional cost $O(\log N)$. [102].

B. Long Query Time

In unbalanced tree the query time also increase. [103].

4.6.2 Dynamic KD tree Variation: Divided KD tree

Divided KD tree is a dynamic variation of the traditional KD tree. It is called dynamic KD tree because insertion, deletion and the exact match queries are performed in $O(\log n)$ in the worst case. The time complexity of the range queries is $O(\sqrt{n \log n} + k)$ where n is the no of nodes in tree and k is the number of the answers to the query. The comparison of the big O notations of the traditional and the divided KD tree is given in Table 4.1.

Table 4.1

Time Complexity in worst case

	Pseudo KD tree	Divided KD tree
Insertion	$O(n)$	$O(\log n)$
Deletion	$O(n)$	$O(\log n)$
Exact Match Query	$O(n)$	$O(\log n)$
Range Query	$O(\sqrt{n} + k)$	$O(\sqrt{n \log n} + k)$

n= number of nodes in the tree, k= no of objects returned in the query

Divided KD tree is based on 2-3 trees or AVL tree. We have used 2-3tree for constructing divided KD tree [12]. The characteristics of the 2-3 trees are as follows:

- Each node either has one value or two values.
- If the node has one value then either the node will be leaf node or will have exactly two children. The value in the left sub-tree will be less than the node's value or object and the value in the right sub-tree will be more than the node's value or object.
- The node having 2 values will be either the leaf node or will be having exactly 3 children. The left sub-tree has the smaller values than the node values and the right sub-tree will have the greater values than the node values. The middle child will have the values in between the first and second value of the node.
- All leaf nodes are at same level means the tree is balanced.

4.6.3.1 Insertion in Divided KD tree

The insertion algorithm in the divided KD is based on the 2-3 tree. The insertion operation is performed in such a way that the resulting tree is balanced. If the tree is empty, the value will be inserted in the root. Otherwise, a search operation is performed to find the leaf node where the value can be inserted. If the node has one value, then insert the second value into the node and exchange the values if the left value is not smaller than the right value. If the leaf node has more than one value, the value will be inserting after splitting the node. The node will split to promote the median of the tree values to the parent node. If the parent also has three values, the splitting & promoting process continues forming a new root node if required [104]. Algorithm for inserting the values in the divided KD tree is presented in Algorithm 4.1.

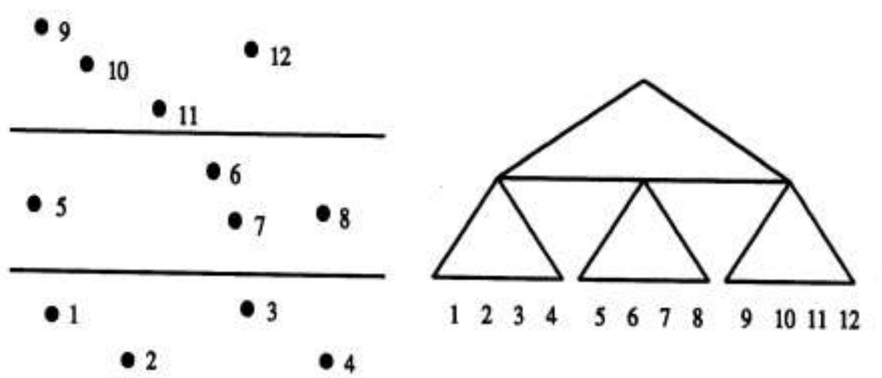


Figure 4.5 Divided KD tree [102]

4.6.3.2 Deletion in Divided KD tree

To delete a value or an object, search the node containing the value. To ensure the balanced divided KD tree, find the in-order successor of the value. The successor can be found by traversing through the edge immediately to the right of the value to be deleted (That is, the middle edge if the key was the left key or the right edge if the key was the right key.) Then, we always go down the left-most edge of all nodes, until we hit a leaf. Once the successor is found swap it with value to be deleted so that the tree will be balanced and no element could now be out of place---All the nodes in the right sub-tree will be greater than the successor. The deletion algorithm for the divided KD tree is presented in Algorithm 4.2.

4.6.3.3 Exact Match Query in Divided KD tree

The exact match query is same as searching an object in a divided KD tree. The searching process starts from the root. The required object is compared with the root, if the root is the required object then that object is returned otherwise the lower sub-tree is traversed to search the required object. Which sub-tree is to be searched is decided by comparing the object with both the objects in the node. If the object is greater than both the objects in the node; traverse through the right sub-tree. If the object is smaller than both the objects in the node, traverse through the left sub-tree. If the object is greater than first object and less than the second object in the node then traverse through the middle sub-tree. The process is continued while the object is found or the traversing stops at the leaf. The steps of the exact match query are explained in Algorithm 4.3.

Insertion in Divided KD tree Algorithm 4.1

Input: Hosts (H_1, H_2..............H_n) Object

Output: Divided KD tree

1) If tree is empty, create a root node and insert value in the root and exit.
2) Compare the host object to the root to search the position in the lower tree (left subtree, middle subtree, right subtree) in which the new host should insert.
   - 2.1 If the node is having one host object. Insert the new one as the second object. Rearrange the hosts in such a way that smaller host appears as the first value and the larger one as second value and exit.
   - 2.2 If the node is having more than one host then split based on the middle host Value and promote the middle host object to the parent and so on till the tree becomes balanced.
     - 2.2.1 If the parent is also having more than one value then splitting and promoting continues to create a new root if required and exit.
3) Rebalance, if necessary

Deletion in Divided KD tree Algorithm 4.2

Input: Root of Divided KD tree, host object to be deleted

Output: Divided KD tree after Deletion

1) If the host is at root. Swap the host object with the successor of the host.

   1.1 The successor can be found by traversing through the edge immediately to the right of the value to be deleted (That is, the middle edge if the object was the left object or the right edge if the object was the right key.)

   1.2 Traverse down the leftmost edge of all nodes, until a leaf. At that point, the leftmost host object in that leaf is the successor.

   1.3 swap the host object with the successor and delete the desired object from the leaf.

2) Rebalance, if necessary

Exact Match Query in Divided KD tree Algorithm 4.3

Input: Root, Host Object to be searched

Output: Host Node Object

1) Compare the host object to the root to search the position in the lower tree (left subtree, middle subtree, right subtree)
2) Search the host object in the lower subtree (left, middle or right) by comparing the Host objects.
3) Return the object of the host if found otherwise return not found.

4.6.3.4 Nearest Neighborhood Query in Divided KD tree

Nearest neighborhood query in divided KD tree is to search the next successor of the node in the tree. To search the successor of the node, find the in-order traversal of the node and the next node in the in-order traversal will be the successor of the node. Algorithm 4.4 explains the steps of nearest neighborhood query.

Nearest Neighborhood Query in Divided KD tree Algorithm 4.4

Input: Root of divided KD tree

Output: Host node Object

1) Compare the host object to the root to search the position in the lower tree(left subtree, middle subtree, right subtree).
2) Nearest neighbor of the host object which is next in the inorder traversal of the tree because the neighbor having less capacity then required will not be useful.
3) Return the Host object otherwise return NULL.

4.7 Initial VM Allocation Algorithm

The hosts in each data centers are arranged in Divided KD tree. When the request for the VM arrives the VM object is compared with the root. If the VM request object is less than the root then the left sub-tree is searched for the appropriate neighborhood node of the VM object. If the VM request object is greater than the root then right sub-tree of the tree is searched for the appropriate host. If the VM request object is less than the second value of the root object and greater than the first value of the root object then the middle sub-tree is searched for the appropriate node to place the VM object. Because only one sub-tree of the tree is searched the searching is fast as compared to list-based algorithms. The neighborhood node is searched because in exact match query there may be chances of host overload which leads to VM migration. The characteristics of the host selected are compared with the VM characteristics. If the VM characteristics are less than the Host characteristics the host is allocated to the VM and VM allocation map is updated. The VM characteristics are subtracted from the node and the node is deleted and reinserted in divided KD tree. If the appropriate host is not found then the host in another data center is initialized and the VM is allocated to the host. Flowchart of the Algorithm 4.5 is shown in Figure 4.3.

Initial VM Allocation Best Fit Algorithm: 4.5

Input: Hosts ($H_1, H_2, \ldots H_n$) arranged in a Divided KD Tree ($T_1, T_2 \ldots\ldots T_n$)and VMList ($V_1$, V2…….$V_n$)
OutPut: VM Allocation Map
Begin
    While all VMs not placed **do**
        Place a nearest Neighbor hood Query to find the best host (H) to place The VM

            if V_i (Resources)<= H_i(Resources) then
                {
                (i)VM is placed on the host H_i
                (ii)Delete the host H_i from the Tree T_i
                (iii)Deduct the characteristics of VM from the Host characteristics
                (iii)Insert the newly created Host with less characteristics in the KD tree again and balance the tree.
                }
                else
                initialize a new host.
                place the VM on the newly initialized host.
            end
    end
end

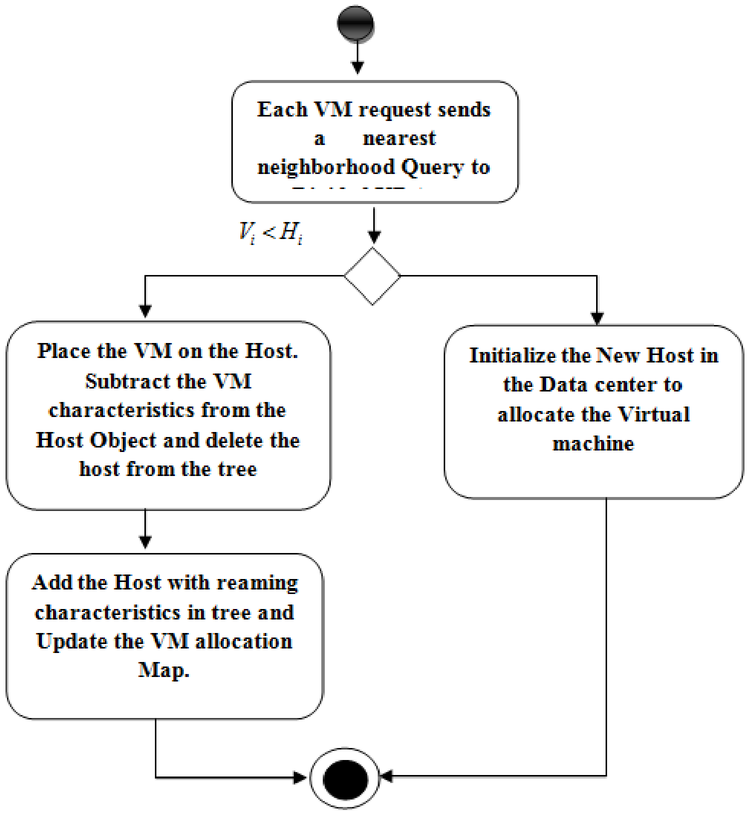


Figure 4.6: Initial VM Allocation

4.8 Experimental Setup and Results

We have used CloudSim at the core of Federation. To make a simulation evaluation we have used the workload traces from a real system. The data is taken from the CoMon project which is a monitoring infrastructure for platnetLab [140]. The data contains the CPU utilization of more than 1000 VMs from servers located at more than 500 places around the world. The interval of utilization measurement is 5 minutes. In CloudSim, each VM is randomly assigned a workload trace from one of the VMs from the PlatnetLab workload.

The initial VM allocation algorithm in the first phase of the model packs the VMs in less number of hosts as well as a sufficient amount of CPU utilizations data is collected to train the model. We have used first 50 allocations in initial VM allocation phase. Based on the CPU utilization of these 50 allocations the SVM model is trained and the future CPU utilizations are predicted for load balancing. The parameter for comparing the results of the various existing algorithms and the proposed algorithm is the total number of hosts (Bins) used to allocate the first 50 virtual machines. CloudSim simulates dual-core CPU with less power as lighter workload is required to overload a server and dual-core will be sufficient to simulate the multi-core CPUs. The experimental results are shown in Figure 4.7. The proposed algorithm DKD-LSSVM (Divided KD tree -Least Square Support Vector Machine) is compared with best fit decreasing algorithm. The results are promising as DKD-LSSVM uses less number of hosts as compared to best fit decreasing allocation.

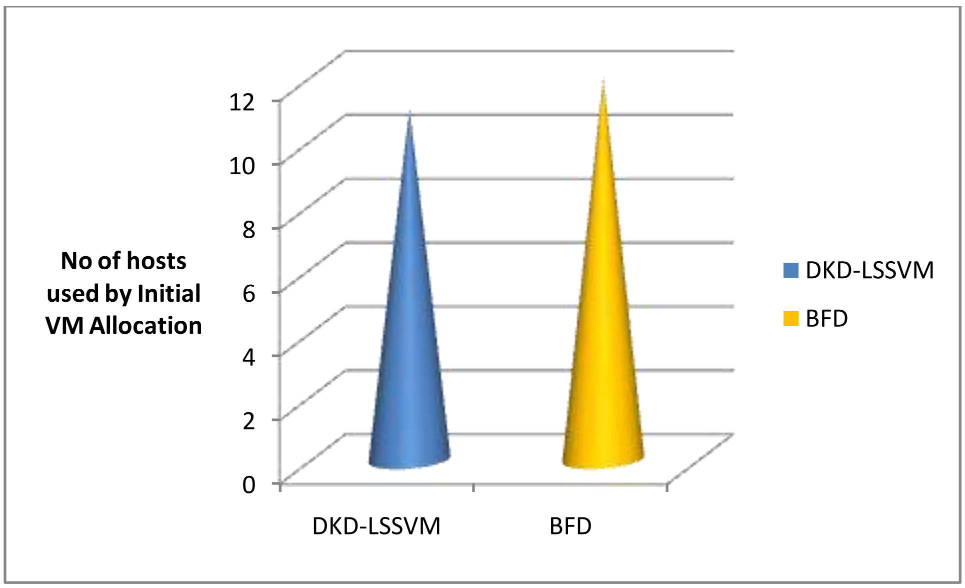


Figure 4.7 No of hosts used in Initial VM Allocation

DKD-LSSVM- Divided KD-tree Least Square Support Vector Machine

BFD- Best Fit Decreasing Algorithm

CHAPTER 5

Dynamic Least Square Support Vector Machine Regression Based Host Overload Detection algorithm for Dynamic VM Consolidation

5.1 Introduction

Dynamic Virtual Machine Consolidation is a process of re-allocating the virtual machines in a minimum possible number of servers, to improve the utilization of the physical resources and also to improve the total energy efficiency of the datacenter. To deal with the sudden spikes in the customer demand, the number of data centers is growing with the need which leads to high active server ratio resulting in extra energy emission and production of Carbon dioxide (CO^2).Virtualization played an important role in VM consolidation by applying the Live VM migration. The commercial products like Microsoft System Center Virtual Machine Manager (VMM),VMware vSphere Distributed Resource Scheduler (DRS) and Citirix XenServer offer VM consolidation as their chief functionality .[124].

The VM consolidation decision depends on the host overload/underload behavior. The host is overloaded or under-loaded is based on its utilization behavior. The VMs from the under-utilized and over-utilized hosts are relocated to other hosts by packing the VMs on a minimum number of hosts. The hosts having no virtual machine are shifted to the passive mode so that the total energy consumption can be reduced. Statistical methods played a great role in predicting the behavior of the host in a dynamic manner. The author [124] has proposed various statistical methods for host overload and underload behavior of the hosts in his thesis. These algorithms take input as the previous or current utilization of the hosts and predict the future based on the previous or current state of the system. He has proposed Local Regression, Median Absolute Deviation, Robust Local Regression and Markov Chain model for predicting the overloaded hosts [124]. All statistical models cannot be applied to all the environments. The choice of the statistical methods depends on the input data because every statistical model is based on some assumptions. Markov chain model assumes that the data will be stationary but complex and dynamic environment like cloud, experience highly variable non-stationary workload.

We have proposed a host overload prediction model based on the Least Square Support Vector Machine to analyze the host utilization to forecast the host overload behavior. Machine learning algorithms are the subfield of Artificial Intelligence algorithms that are based on pattern recognition to enable the machine to learn and perform automated tasks and activities without human intervention. Statistical methods for predicting future values includes regression analysis, time series analysis, least squares, autoregressive moving average model and vector autoregressive model. Support vector machine algorithms overlap statistical methods by providing the method to automate the information discovery process. Support vector machine is a neural network based method which over-performed the statistical methods and artificial neural network because SVM (Support Vector Machine) provides a global optimum solution and doesn’t even have the over-fitting problem. As compared to neural network, support vector machine requires less training samples to make decisions.

5.2 Related Work

Time series modeling and forecasting have essential importance in various practical domains such as stock market analysis. Accurate prediction of the finance related domains is essential to take better investment decisions with minimum risk. Various statistical based and artificial intelligence tools are proposed in the literature. There are plenty of successful applications have shown that artificial neural network (ANN) is very useful tool for time series modeling and forecasting which outperformed the traditional statistical methods. However, some studies shows that Artificial Neural Networks has some limitations in learning the patterns in complex multidimensional data and also suffers from difficulty in selecting the large number of controlling parameters which includes relevant input variables, size of the hidden layers and learning rate [112]. Unlike ANNs, Support Vector Machine (SVM) does not depend on the dimensionality of the input space and are less prone to over fitting. [113][114].

Support vector machine has been applied in many real life classifications based as well as regression based applications i.e. Geo- and Environmental Sciences, Prediction of Traffic Speed and Travel Time, Underground Cable Temperature Prediction and for various time series prediction applications like stock market price prediction, electricity price prediction and peak load prediction etc. [115]. The main focus of this chapter is on support vector regression. We have used Least Square Support Vector Machine Regression method for predicting the host overloading behavior. On the basis of which the re-configuration of the virtual machines takes place. SVM was used as a tool for load prediction in 2001 in EUNITE Competition [116]. After that Support Vector Machine has been applied to various forecasting applications. Some examples of load prediction are presented in [120-123]. The author in [120] combined the features of Support Vector Machine and Genetic algorithms to predict the regional electricity load. However, the paper [121] integrates the Support Vector Machine and ant colony optimization for predicting the power load. The author in [120-123] investigated the use of Support Vector Machine algorithm applicability for predicting short-term load forecasting. The improved versions of the classical SVM have also been proposed i.e. Least Square Support Vector Machine, SVMTorch, on-line Support Vector Machine, Dynamic Support Vector machine Regression etc.[117-119]. We have proposed a dynamic support vector approach which is suitable for predicting host overload detection in the dynamic environment like cloud.

5.3 The Objective of Host Overload Detection

The objective of the host overload detection algorithm is to improve the quality of the VM consolidation. Since Live VM migration supports dynamic VM consolidation but frequent VM migration can degrade the performance of the data center because most of the time will be wasted in VM migration. The VM consolidation algorithm should also consider the time between the two VM migrations and the objective should be such that the time between two migrations should be maximized. The quality of the VM consolidation is inversely propositional to H, the mean number of active servers over n time steps.

$$H = \frac{1}{n}\sum_{i=1}^{n} a_i \qquad \text{Eq..(5.1)}$$

Where a_i is a number of active hosts where $i = 1, 2, ..., n$. The value of H should be minimized to improve the quality of the VM consolidation. VM consolidation approach will migrate some of the VM from the overloaded host to another target host. Two cases can occur when a VM migration happens.

Case 1: VM is migrated to an active host.

Case 2: VM is migrated to an inactive host.

If the target host is active then there is no change in the mean number of active hosts. But if VM migration opens a new host for containing a VM then the number of active hosts increases. To understand the effect of the host overload detection algorithm it is assumed that no host will be switched off during the experiment i.e. once a host is activated it will be active until n. let p is the probability of the case 1 then the probability of the case 2 will be (1-p). Let T is a random variable denoting the time between two subsequent VM Migrations initiated by the host overload detection algorithm. The expected value of the number of VM migrations initiated by the host overload detection algorithm over n time steps is n/E(T) where E(T) is the expected time between the VM migrations.

Based on the above description, the number of extra hosts switched on due to VM migration based on the host overload detection algorithm can be defined as $X \sim P_0(\lambda)$ where λ = np, which is Poisson distribution random variable. The expected number of extra hosts switched on is defined by

$$E(X) = \lambda \qquad \text{Eq..(5.2)}$$

Let A is a random variable which denotes the time during which an extra node is activated between time steps between 1 to n. The expected value of A can be derived as

$$E(A) = \sum_{i=1}^{\frac{n}{E(T)}} \lambda^{'}$$

$$\lambda^{'} = (n-(i-1)E(T))p$$

$$= \frac{n}{E(T)} \frac{p}{2}(n+n-(\frac{n}{E(T)}-1)E(T)$$

$$\leq \frac{np}{2}(1+\frac{n}{E(T)}) \qquad \text{Eq..(5.3)}$$

Where $\lambda^{'}$ is the expected number of extra hosts to be activated. The Equation (5.1) can be rewritten as

$$H = \frac{1}{n} a_1 + \frac{1}{n}\sum_{i=1}^{n}(a_i - a_1) \qquad \text{Eq.. (5.4)}$$

Here a_1 denote the active hosts after initial VM allocation phase. The second term $H^* = \frac{1}{n}\sum_{i=1}^{n}(a_i - a_1)$ represents the mean number of servers being switched on because of the VM Migration in n steps of time. The objective of the host overload detection algorithm is to minimize H^*.

$$E(H^*) \propto \frac{\lambda}{n} E(A)$$

$$\leq \frac{1}{n} \frac{np}{E(T)} \frac{np}{2}(1+\frac{n}{E(T)})$$

$$= \frac{np^2}{2E(T)}(1+\frac{n}{E(T)}) \qquad \text{Eq.. (5.5)}$$

To improve the quality of the VM consolidation we can improve the E(T). E(T) is the average time between VM migrations which can be controlled by the host overload detection algorithm. The host overload detection algorithm should maximize the mean time between two migrations from an overloaded host.

5.4 Least Square Support Vector Machine Regression

Least Square Support Vector Machine (LS-SVM) is a class of kernel-based learning methods which is the simplified version of the classical support vector machine. In LS-SVM the inequality constraints of the objective function in classical SVM are transformed into the equality constraints and the quadratic programming problem is transformed into linear equations avoiding insensitive loss function and greatly reduces the complexity .[110]. The various characteristic of the LS-SVM are underlined as: [111]

A. The LS-SVM is computationally fast as compared to classical Support Vector Machine
B. It is able to combine support vector machine, neural networks and Gaussian process which can explore the nature of them.
C. The least squares support vector machine is able to expand to the autoregressive process so as to deal with the dynamic problems.

5.4.1 Mathematical Formulation

Host utilization is a univariate time series and the future values can be predicting by analyzing the past values. Support Vector Machine algorithms are based on the statistical learning techniques. The goal of Support vector Regression is to find a hyper plane that are close to as many as the data points as possible. The future values are predicted by splitting the time series $x_1, x_2, \ldots\ldots, x_n$ data into training inputs and the training outputs. Given training data sets of N points $\{x, y\}$, $i = 1, 2 \ldots N$ with input data $x_i \in X \subset R^n$ and $y_i \in Y \subset R$ output data .Assume a nonlinear function $f(x)$ as given below

$$f(x) = W^T \phi(X_i) + b \qquad \text{Eq...(5.6)}$$

w=weight vector, b=bias and $\phi(X_i)$ is a non-linear mapping to a higher dimensional space. The optimization problem is defined as:

$$\text{Minimize} \quad \frac{1}{2} W^T W$$

$$\text{Subject to:} \begin{cases} y_i - (W^T \phi(X_i) + b) \le \varepsilon \\ y_i - (W^T \phi(X_i) + b) \ge -\varepsilon \end{cases} \qquad \text{Eq.. (5.7)}$$

ε (≥ 0) is a user defined maximum error allowed. The above Eq (5.7) can be rewritten as

$$\text{Minimize} \quad \frac{1}{2} W^T W$$

$$\text{Subject to:} \begin{cases} y_i - (W^T \phi(X_i) + b) \le \varepsilon \\ W^T \phi(X_i) + b - y_i \le \varepsilon \end{cases} \qquad \text{Eq..(5.8)}$$

To solve the above equation slack variables needs to be introduced to handle the infeasible optimization problem. After introducing the slack variables the above equations take the form

$$\text{Minimize} \quad \frac{1}{2} W^T W + C \sum_{i=1}^{m} (\xi_i^+ + \xi_i^-)$$

$$\text{Subject to:} \begin{cases} y_i - W^T \phi(X_i) - b \le \varepsilon + \xi_i^+ \\ W^T \phi(X_i) + b - y_i \le \varepsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \ge 0 \end{cases} \qquad \text{Eq.. (5.9)}$$

The slack variables ξ_i^+, ξ_i^- define the size of the upper and the lower deviation as shown in the Figure 5.1 (a). Figure 5.1(b) represents the trade-off between training error and the flatness of the solution.

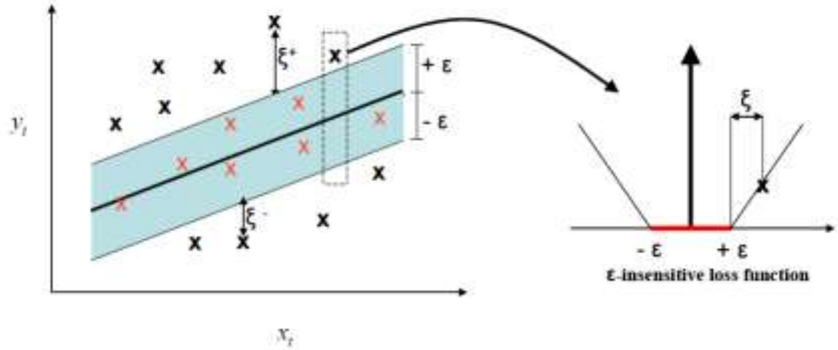


Figure 5.1 (a) The Accurate points inside ε Tube (b) Slope decided by C

The dual form of the Equation (5.9) after adding the Lagrangian multipliers $(\alpha_i^+, \alpha_i^-, \xi_i^+, \xi_i^-)$ can be rewritten as:

$$L_p = \frac{1}{2}W^T W + C\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-) - \sum_{i=1}^{m}(\eta_i^+ \xi_i^+ \eta_i^- \xi_i^-) \quad -\sum_{i=1}^{m}\alpha_i^+(\varepsilon + \xi_i^+ - y_i + W^T\phi(X_i) + b)$$

$$-\sum_{i=1}^{m}\alpha_i^-(\varepsilon + \xi_i^- + y_i - W^T\phi(X_i) - b) \quad s.t.\ \alpha_i^+, \alpha_i^-, \xi_i^+, \xi_i^- \geq 0 \qquad \text{Eq.. (5.10)}$$

Applying the conditions of the optimality, one can compute the partial derivatives of L with respect to w, b, ξ_i^+, ξ_i^- equating them to zero and finally eliminating w and ξ obtain the following linear system of equations

$$\begin{pmatrix} o & e^T \\ e & \Omega + C^{-1}I \end{pmatrix}\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \qquad \text{Eq.. (5.11)}$$

Here $y = [y_1, y_2, ...y_l]$, $e = [1,1,.......1]^T$, $a = (a_1, a_2, ..., a_l)^T$ and $k(x_i, y_i)(\forall i = 1, 2, ...l)$ is the radial base function based kernel matrix. ($K(x, y) = e^{-[\![x-y]\!]^2/2\sigma^2}$).

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & k(x_1, x_1) + 1/c & \cdots & k(x_1, x_l) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & k(x_l, x_1) & \cdots & k(x_l, x_l) + 1/c \end{pmatrix}\begin{bmatrix} b \\ a_1 \\ \vdots \\ a_l \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ \vdots \\ y_l \end{bmatrix} \qquad \text{Eq.. (5.12)}$$

The LS-SVM decision function is thus given by [112]. $y(x)$ is the future CPU utilization of the host based on the previous utilization data. LSSVM regression model is executed locally on each host and based on $y(x)$ future utilization is calculate for host overload detection. If the host is overloaded the VM on the host will be migrated to another less overloaded host for load balancing.

$$y(x) = \sum_{i=1}^{l} \alpha_i k(x_i, y_i) + b \qquad \text{Eq.. (5.13)}$$

The considerable advantage of the Equation (5.13) is that it is independent of the dimensionality of the input space and the sample size.

5.4.2 Selection of Kernel Function

$k(x_i, y_i)$ is defined as the kernel function. The kernel function is used in SVM techniques for mapping the input space to high dimensional space through some non-linear mapping to make a separating hyper-plane in a feature space. There are several types of kernel functions used in SVM. The type of SVM is constructed is based on the kernel function used. The kernel function also affects the computation time to implement the SVM. The most popular kernel functions used in SVM are:

Polynomial Kernel Function:

$$K(x, y) = (x, y)^d \text{, d=1}$$

A sigmoid kernel function

$$K(x, y) = \tanh(b(x.y) + c) \text{ where b is slope and c is the bias}$$

A Gaussian radial basis kernel function:

$$K(x, y) = e^{-[\![x-y]\!]^2/2\sigma^2} \text{ where } \sigma(>0) \text{ is the kernel width}$$

The radial base kernel method is widely used in the ANNs and SVM because it can handle linear and non-linear input-output mapping effectively. It requires less number of hyper-parameters, is numerically simple.

5.5 Online Host Overload Detection

In the dynamic environment of cloud, the number of samples will be incremental because the host utilization will be changing continuously. The model deals with these changes by modifying the sample points in the model continuously. The role of the previous utilization data of the host for training the model decreases gradually with the time. The farther the sampling moment away from the present moment, the smaller proportion those data account for in modeling; the closer the sampling moment to the present moment, the more consideration is given. So to train the model with the latest history of the CPU utilization of the host we have given a varying degree of weights to the sample data. The weighting function introduced in the model is given by

$$d_i = e^{-i^2/2\lambda^2}, i = 1, 2, ..., t \qquad \text{Eq.. (5.14)}$$

λ is the given parameter , the small value of λ can reduce the storage of the historical data but the value should not be chosen very small which can affect the training process of the model. After applying the weight function the Equation (5.9) can be rewritten as

$$\text{Minimize } \frac{1}{2}W^T W + Cd_{t-i}\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-) \qquad \text{Eq.. (5.15)}$$

$$\textit{Subject to}: \begin{cases} y_i - W^T\phi(X_i) - b \le \varepsilon + \xi_i^+ \\ W^T\phi(X_i) + b - y_i \le \varepsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \ge 0 \end{cases}$$

After applying the Lagrangian multipliers the equation becomes

$$L_p = \frac{1}{2}W^T W + Cd_{t-i}\sum_{i=1}^{m}(\xi_i^+ + \xi_i^-) - \sum_{i=1}^{m}(\eta_i^+\xi_i^+\eta_i^-\xi_i^-) \quad -\sum_{i=1}^{m}\alpha_i^+(\varepsilon + \xi_i^+ - y_i + W^T\phi(X_i) + b)$$
$$-\sum_{i=1}^{m}\alpha_i^-(\varepsilon + \xi_i^- + y_i - W^T\phi(X_i) - b) \quad s.t.\ \alpha_i^+, \alpha_i^-, \xi_i^+, \xi_i^- \ge 0 \qquad (5.16)$$

Applying the conditions of the optimality, one can compute the partial derivatives of L with respect to w, b, ξ_i^+, ξ_i^- equate them to zero and finally eliminating w and ξ obtain the following linear system of equations. Table 5.1 shows the description of the symbols used in the model.

$$\begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & k(x_1,x_1)+1/c & \cdots & k(x_1,x_t) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & k(x_t,x_1) & \cdots & k(x_l,x_l)+1/cd_0 \end{pmatrix} \begin{bmatrix} b \\ a_1 \\ \vdots \\ a_t \end{bmatrix} = \begin{bmatrix} 0 \\ y_1 \\ \vdots \\ y_t \end{bmatrix} \tag{5.17}$$

Table 5.1

Symbols used in the Mathematical formulation of Model

Symbol	**Description**
W	Direction Vector (angle with the axis)
b	Bias
$\phi(X_i)$	Non-linear mapping to a higher dimensional space
ε	Maximum Error Allowed
ξ_i^-, ξ_i^+	Slack Variables, defines the upper and lower deviations to ε tube
$(\alpha_i^+, \alpha_i^-, \xi_i^+, \xi_i^-)$	Lagrangian multipliers
$k(x_i, y_i)(\forall i = 1, 2, ...l)$	Radial base kernel matrix
C	Regularization Parameter
d_i	Weighting function

5.6 Working of the model

5.6.1 Data set pre-processing:

The data set pre-processing mainly includes normalization processing and divided the sample data into a training set and testing set because LSSVM is a supervised learning algorithm. Three common steps of the pre-processing are formatting, cleansing and sampling. The CPU utilization of the host is formatted in the matrix format and the data is separated in training as well as test data. Cleansing and sampling processes fix the missing values if any and we have selected last 50 utilization values so that latest utilization history is fed to the algorithm to get the desired results. The result of the data pre-processing is the final training and the testing set.

5.6.2 Initialize Kernel function and Initialize Parameters

As described in section 5.4.2; the kernel function is used in SVM techniques for mapping the input space to high dimensional space through some non-linear mapping to make a separating hyper-plane in a feature space. The selection of the kernel function affects the computation time. We have used radial basis kernel function for training our model as the radial basis function is it can handle linear and non-linear input-output mapping effectively. It requires less number of hyper-parameters, is numerically simple. The two control parameters are C and ε. C is a regularization parameter. If the value of C is selected too large the number of support vector will be high resulting in over fitting. If the very small value of the C is selected then it may result in under fitting. The value of ε determines the level of the accuracy of the approximation. In other words, its value signifies the maximum error allowed.

5.6.3 Training and Predicting future utilizations

Based on the training sets and the regularization parameters the online LSSVM model is trained and the CPU utilizations are predicted on the basis of it. Figure 5.2 shows the detailed working of the model.

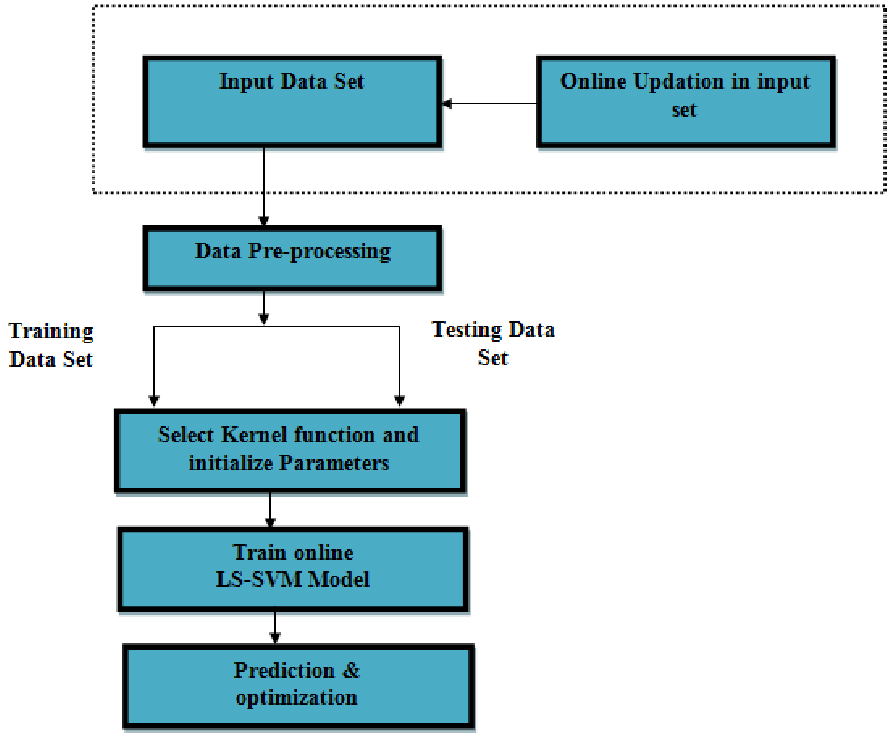


Figure 5.2 Host Overload Detection Model

5.7 Experimental Setup and Results

We have used CloudSim at the core of Federation. To make a simulation evaluation we have used the workload traces from a real system. The data is taken from the CoMon project which is a monitoring infrastructure for platnetLab [140]. The data contains the CPU utilization of more than 1000 VMs from servers located at more than 500 places around the world. The interval of utilization measurement is 5 minutes. In CloudSim each VM is randomly assigned a workload trace from one of the VMs from the PlatnetLab workload. We have compared the proposed algorithms with the existing algorithms proposed in [124]. The description of the algorithms is presented in Table 5.2.

To maintain the QoS requirements with energy efficiency, we have used a workload independent metrics to evaluate the performance of the federation. The proposed algorithm is evaluated mainly on energy consumption, SLA violations and Scheduling Delay. The SLA violation is measured by two metrics

Overload Time Fraction (OTF): The time fraction in which the active hosts experienced 100% CPU utilization.

$$OTF = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{si}}{T_{ai}}$$ Where n is number of hosts

T_{si} =time duration when host experienced maximum utilization
T_{ai} = total duration when VM was allocated to host

Figure 5.3 shows the overload time fraction of the different algorithms with the proposed algorithm. We have used different permutation combinations of the host overload detection algorithms with VM selection algorithms and compared the result with our proposed algorithm. The results are promising.

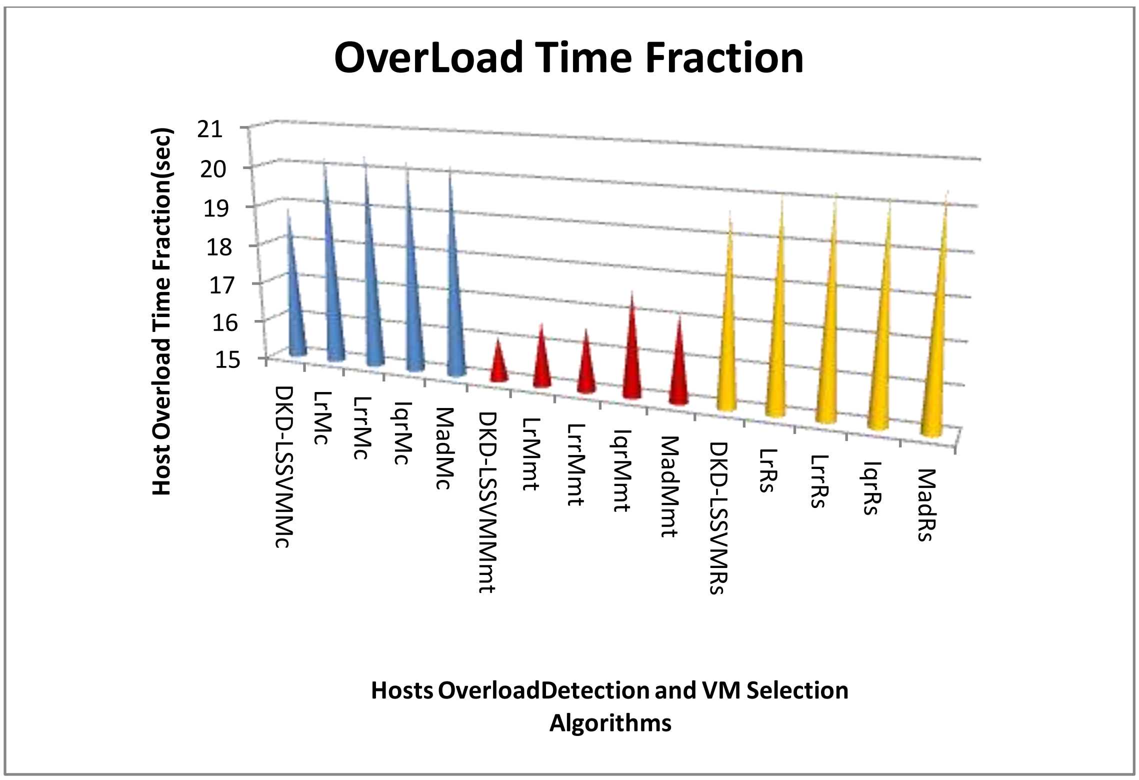


Figure 5.3 Overload Time Fractions

Performance Degradation due to Migration (PDM): The second metric PDM measures the performance degradation of the data center.

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{D_{dj}}{D_{rj}}$$ Where M is number of VMs

D_{dj} =performance degradation because of migration

D_{rj} = total capacity requested by VM.

Figure 5.4 shows the comparison of performance degradation of different algorithm with the proposed algorithm.

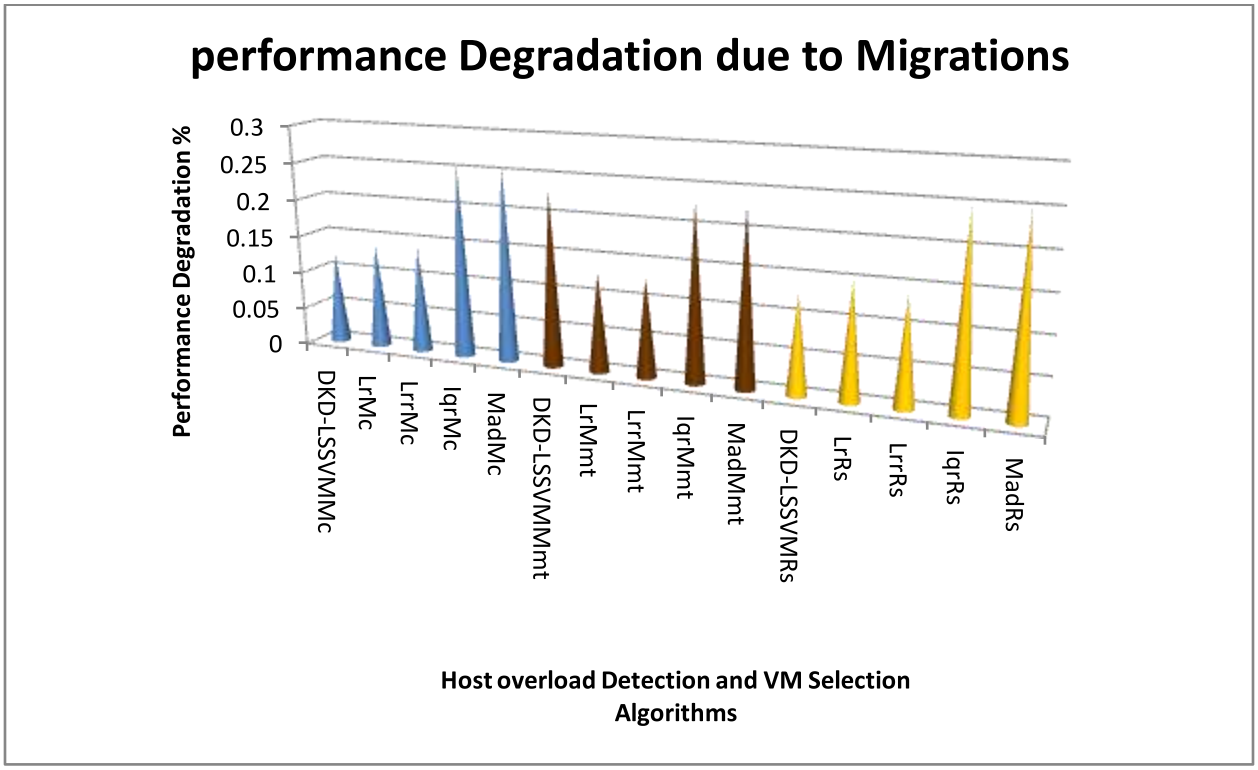


Figure 5.4 Performance Degradation percentages

No of Host Shut down: VM consolidation algorithm performs two basic activities: predicting the overloaded hosts and the underloaded host. Migrating the VMs from the underloaded hosts and delegating the VMs from the overloaded hosts in such a way that the active hosts can be minimized. The algorithm which minimizes the active hosts is considered as a good algorithm.

Figure 5.5 shows the comparison of host shut down by existing algorithm with the proposed algorithm.

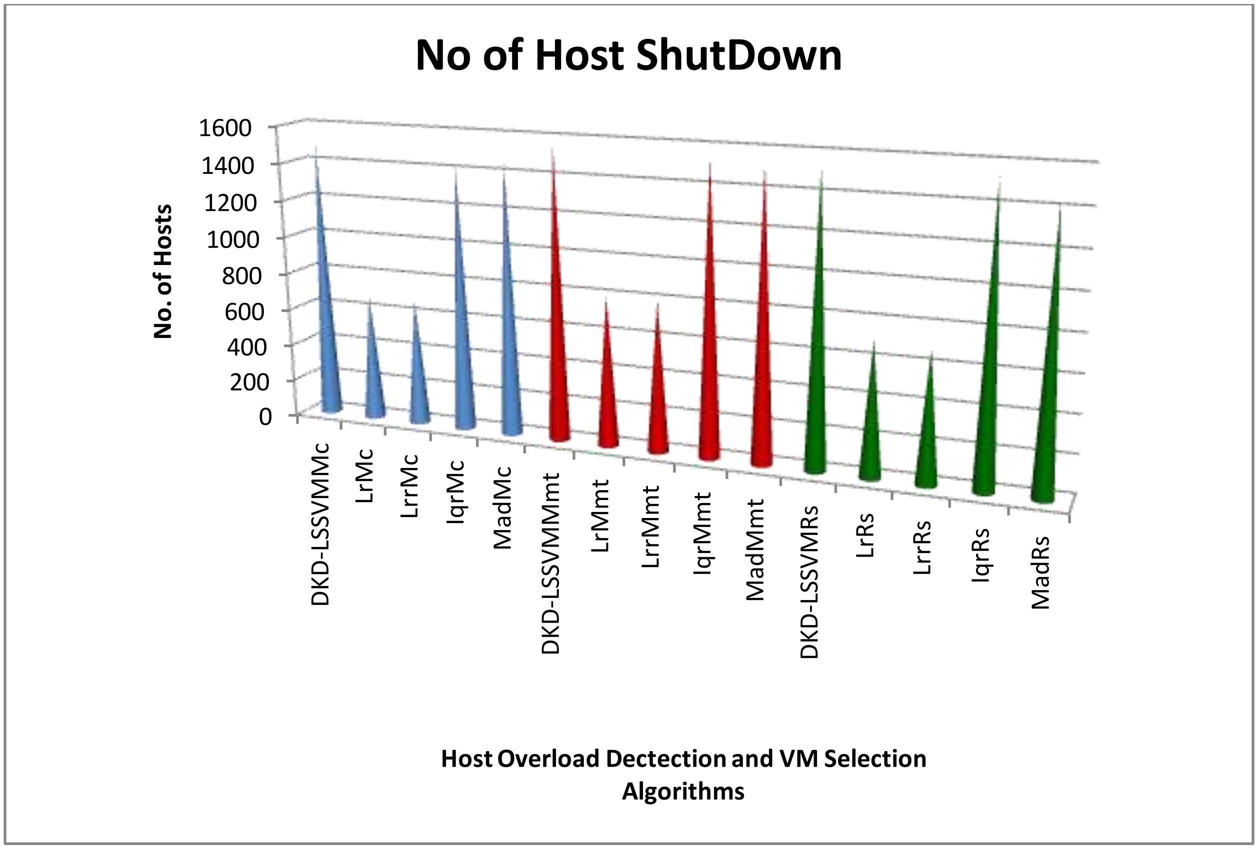


Figure 5.5 Number of hosts shut down

Host Selection Mean: Host selection mean is the time taken by the algorithm to search the appropriate host from the data center to place the VM. The best-fit decreasing algorithm sorts the list of hosts and then selects the appropriate host from the list by comparing each starting from the first node in the list. The proposed algorithm has arranged the hosts in the divided KD tree and it is already proven that searching query in trees provides fast results as compare to the list. Figure 5.6 shows the host selection mean of the proposed algorithm is better than the existing algorithms in the literature.

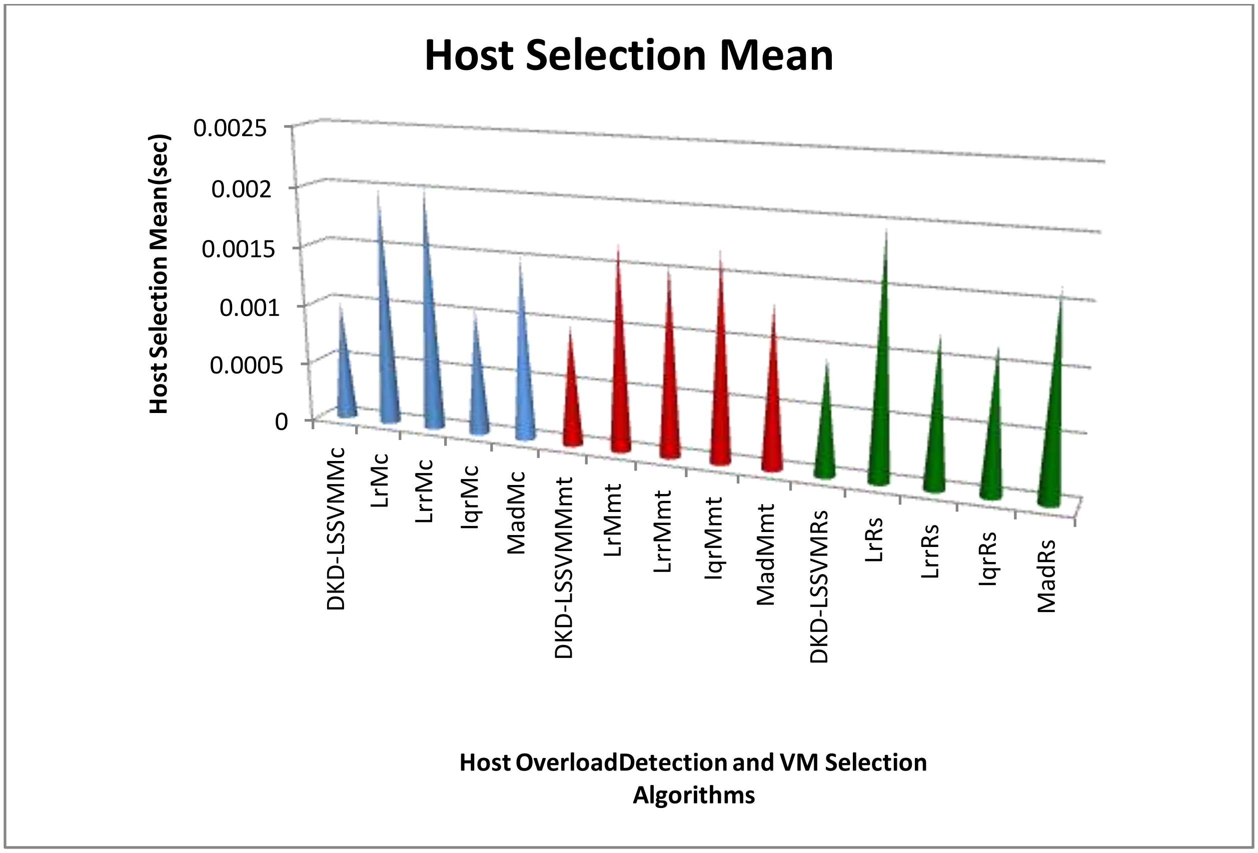


Figure 5.6 Host Selection Mean

SLA violations: Service Level Agreement is a contract between the service provider and the service consumer which defines the level of service expected from the service provider. Thus, the algorithm should not violate the SLA. Figure 5.7 shows that the SLA violation of the proposed algorithm is significantly less as compare to the algorithms available in the literature. The percentage of the SLA violation is represented on the y axis and the algorithms are on x-axis.

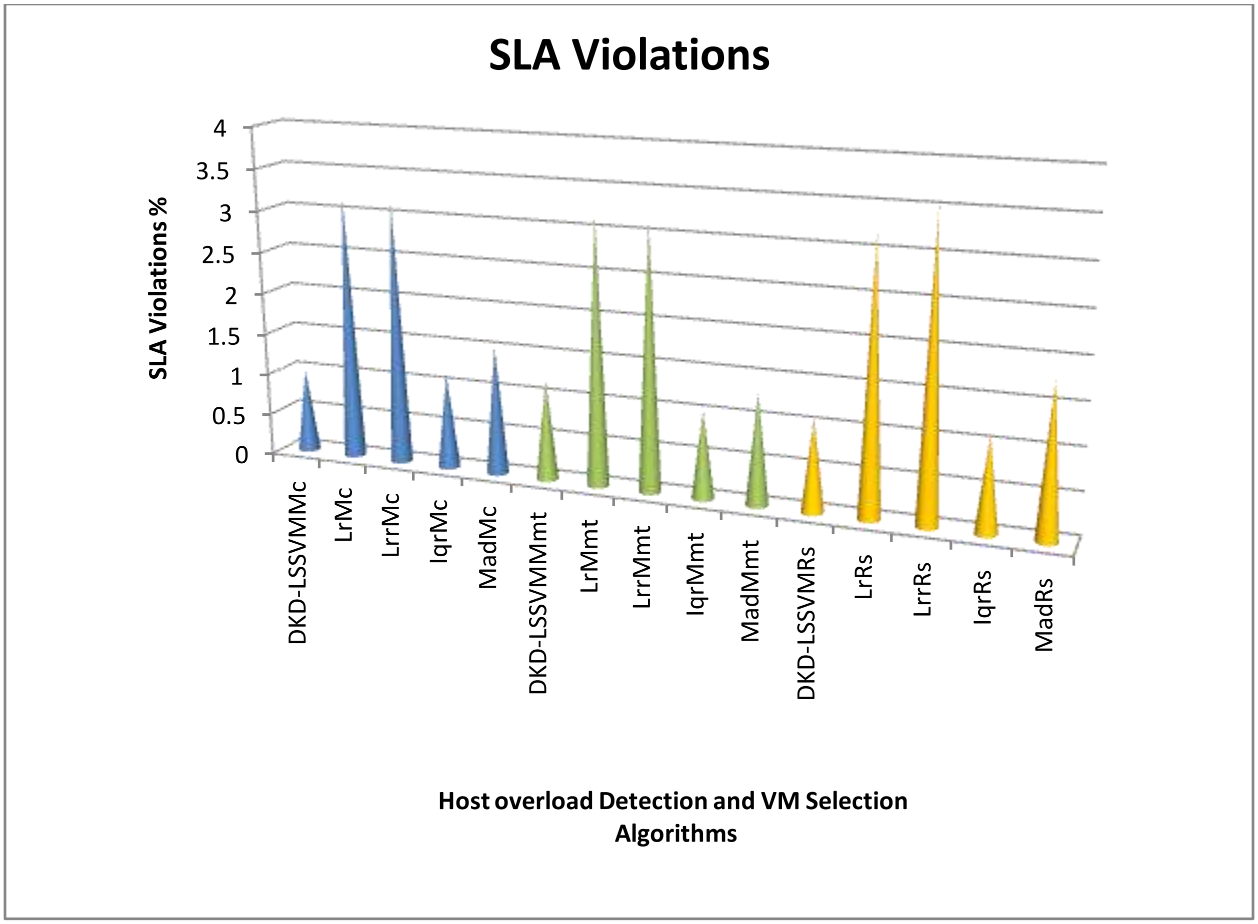


Figure 5.7: SLA Violations

Energy Consumption: The quality of the VM allocation and the consolidation algorithm depends on the energy consumption of the data center. Energy consumption is represented in Kilo Watt per hour and is reduces if the total number of active host reduces. Figure 5.8 compares the energy consumption of the proposed algorithm with the existing algorithms in the literature.

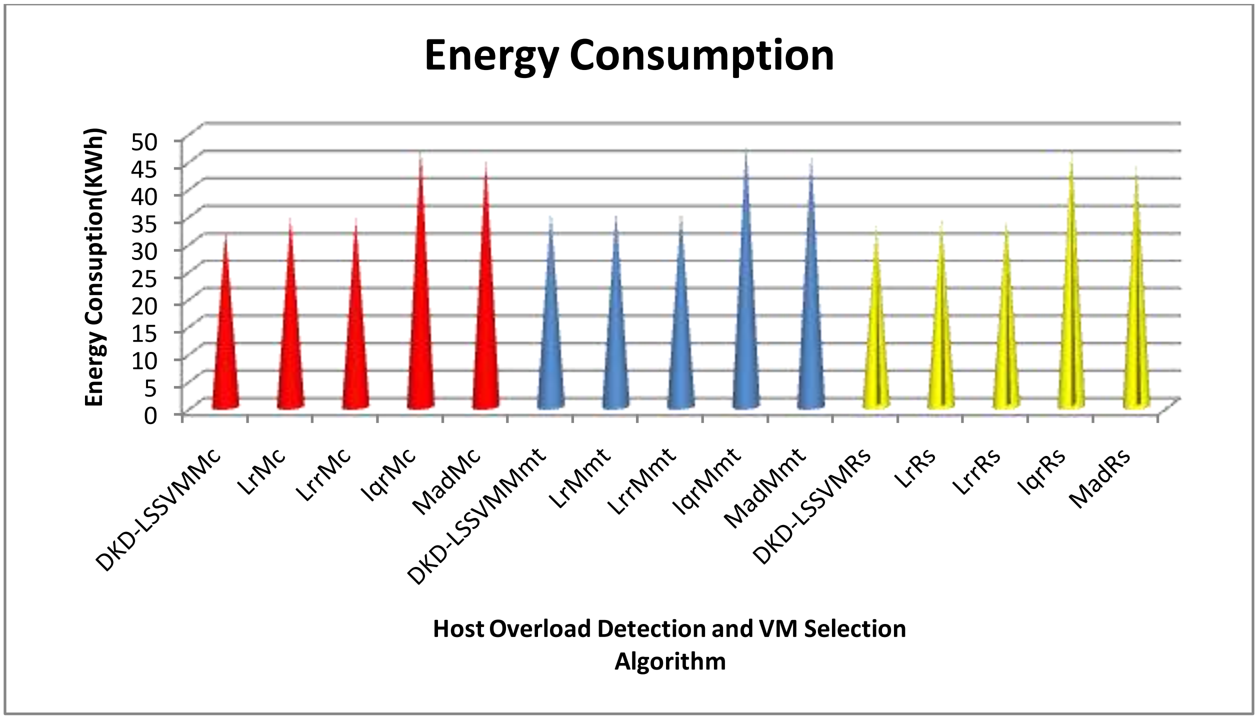


Figure 5.8: Energy Consumption (KWh)

Table 5.2

Algorithm Description

Host Overload Detection Algorithms		**VM Selection Algorithms**	
Name	**Description**	**Name**	**Description**
Lr (Local Regression Algorithm)	The main idea of the local regression method is fitting simple models to localized subsets of data to build up a curve that approximates the original data.	Mc(Maximum correlation)	The idea is that the higher the correlation between the resource usage by applications running on an oversubscribed server, the higher the probability of the server overloading. According to this idea, those VMs are selected to be migrated that have the highest correlation of the CPU utilization with the other VMs.
Lrr(Local Robust Regression)	This method is modified version of the Local	Mmt(Minimum Migration time)	The Minimum Migration Time (MMT) policy migrates

	regression algorithm. This is an iteration method. At each iteration a robustness weight is added to predict more precise values		a VM that requires the minimum time to complete a migration relatively to the other VMs allocated to the host.
Iqr(Interquartile range)	The interquartile range (IQR) is a measure of variability, based on dividing a data set into quartiles. Based on the variability the future values can be predicted	Rs(Random Selection)	The random Selection (RS) policy randomly selects a VM to be migrated from the host
Mad(Median Absolute Deviation)	In statistics, the median absolute deviation (MAD) is a robust measure of the variability of a univariate sample of quantitative data.		

Chapter 6

VM Selection Framework for Market Based Federated Clouds

6.1 Introduction

Cloud computing paradigm emerged as future generation technology to facilitate the customers with IaaS (Infrastructure as a service), PaaS (Platform as a service) and SaaS (Software as a Service). The customers can subscribe to the services according to their requirements and pay for the services on the basis of the usage. A growing number of Cloud Computing service providers and service offerings, in particular, Cloud Storage and Compute services have been observed recently. These offerings bind the user to a specific technology and the user can't switch from the service without providing significant switching cost. For example, Heroku is one of the first Platform-as-a-service provider and provides a fully prepared stack to deploy automatically Ruby, Node.js, Clojure, Java, Python and Scala applications and runs on top of Amazon Web Services (AWS). Since then many service providers came in the cloud market as Google with its App Engine (GAE) platform or Microsoft with Windows Azure, HP etc. Every cloud provider has their own particular features, framework and languages supported. Google App Engine is using BigTable while Windows Azure is using Azure Table to store data. [127]. Designing the application using web services and cloud federation solves the above issue significantly.

Cloud Federation is defined as "practice of interconnecting the cloud computing environments of two or more service providers for the purpose of load balancing traffic and accommodating spikes in demand." To handle the sudden spikes in the customer demands, instead of establishing a large infrastructure, the cloud providers integrate their services to serve their customers in un-interruptible manner. Integrating the services with each other evolved the concept of cloud Federation. Federation is beneficial to customers as well as the cloud service providers. The customers can choose a Federated cloud provider of choice, instead of selecting a handful of “global” cloud providers in the market today. The customers are free from vendor lock-in. On the other hand, cloud vendors are also getting benefits from cloud Federation as the local infrastructure provider can connect to the global marketplace that enables each participant to buy and sell the computing capacity on demand. The vendors having spare capacity in the data center can monetize it by submitting it to the marketplace for others to buy, creating an additional source of revenue. Three basic features of cloud federation are resource migration, resource redundancy and combination of complementary resources and services. [128]. Resource migration allows transferring the resources such as virtual machine images, data items, source code etc. from one service provider domain to another. Redundancy allows synchronized usage of similar services from different cloud provider domains, combinations of complementary resources and services allows integrating services from different cloud provider domain to work as a single service. The taxonomy of the cloud federation is presented in Section 2.2. According to the taxonomy the cloud vendors can participate in cloud federation using broker component, direct communication between cloud providers (in this case the communication component is embedded in the cloud itself) or with the help of libraries. Cloud Federation Broker is a gateway component through which all the requests go through. Broker based cloud federation architecture is better *w.r.t* future perspective of the cloud Federation. The benefits of the broker include increased security and control, reduction in IT cost to support growing federation infrastructure, troubleshooting configuration and setup issues, SLA negotiations, SLA violation monitoring, resource scaling during heavy load conditions etc. [130]

6.2 Cloud Federation Broking Schemes

The main task of the broker component is discovery and match-making needed for creation of the cloud Federation. The goal of the discovery service is to explore all the clouds that are available for taking part in the federation. The convenient way to accomplish the discovery process is based on the distributed approach by using publish-subscribe model. A cloud who wants to participate in federation publishes its information to the centralized entity i.e. broker and the broker after identity verification allows the cloud to enter in the federation. Other factors can be reputation mechanism. The match-making service selects the appropriate resource based on cloud request. Thus, Broker creates a trusted, governed and secure cloud management platform to simplify the delivery of complex cloud services to cloud service customers. The cloud federation based on different broking mechanisms is presented in Figure 6.1.

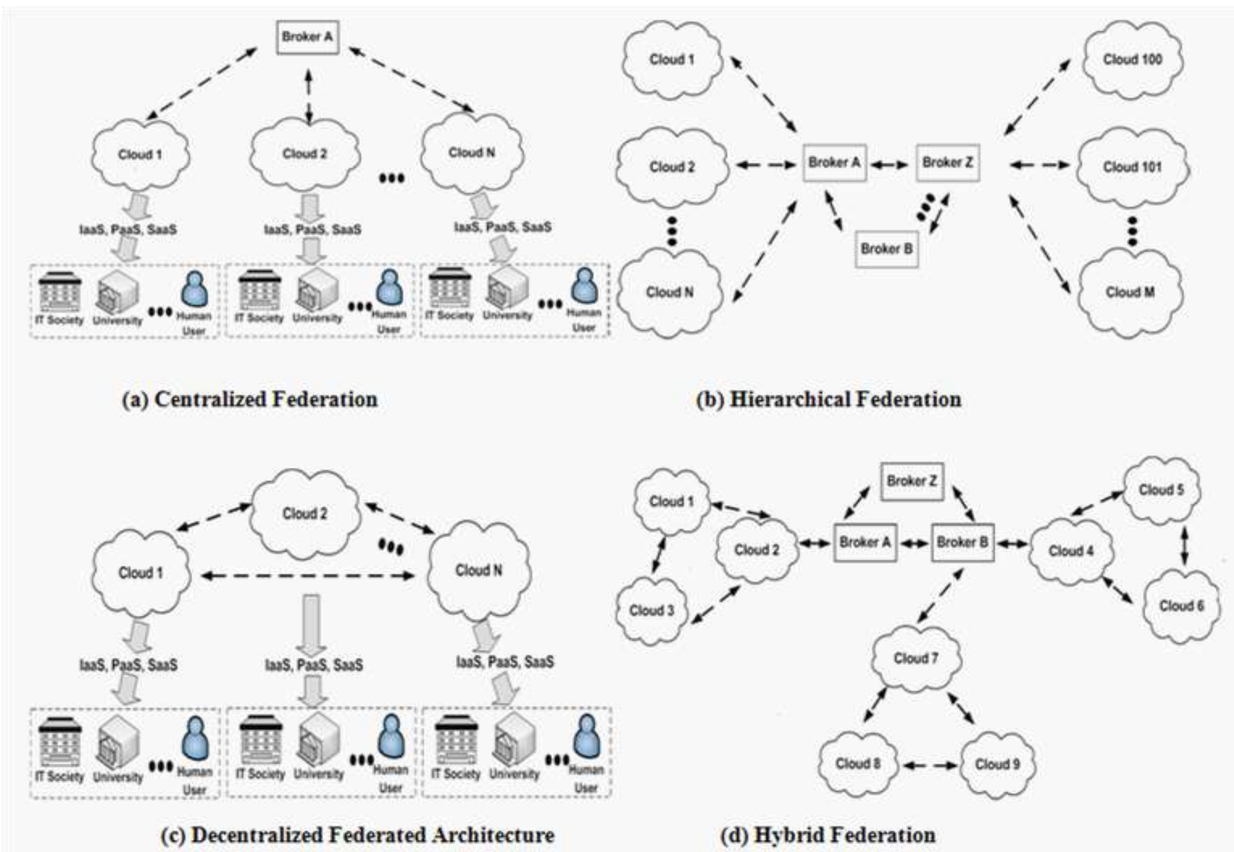


Figure 6.1: Cloud Federation Broking Schemes [129]

In a **centralized** architecture shown in diagram 6.1(a), a single centralized broker is in-charge of establishing the federation. The complete communication between clouds passes through the broker. The broker can be a third party entity allowing the cloud to discover the other clouds for federation according to its requirement. In the case of centralized broking architecture the broker has the load information of all the servers so the load distribution is balanced. As this type of systems totally rely on the centralized entity which creates a bottleneck.

The **hierarchical scheme** shown in Figure 6.1 (b) is an extension of the centralized architecture. The clouds are connected to the broker and the brokers are again connected with other brokers for resource discovery and match making. Single centralized broker is not overloaded for load distribution. The work is divided into multiple brokers but the load balancing decision is affected because no broker will have the complete load information of all the clouds.

In **decentralized architecture** the broker entity is absent and the clouds participating in the federation negotiates the partnership by themselves. All the tasks performed by the cloud brokers e.g. discovery, negotiations of agreements, communication, match making and selecting the best partner for federation are handled by the cloud themselves. This scheme appears to be simple and more flexible but is hard to achieve because it affects the cloud middleware as the middleware will be busy with the above listed tasks instead of the concentrating of the QoS.

In the **Hybrid Federation** or mixed architecture situation some of the clouds participating in the federation are relying on the brokers and others are managing the federation by themselves. In the Figure 6.1(d) cloud (1,2,3), (4,5,6), (7,8,9) are participating in federation without the broker and the cloud 2,4,7 are communicating with the brokers to communicate with the other group of clouds. The hybrid architecture is combining the advantages of both the architectures.

6.3 Related Work

Cloud Computing is adopted by the enterprises in order to deal with the sudden spikes in the customer demand. At present the service providers have inflexible pricing generally based on the usage based. Swapping from one service provider to another is possible only after paying the considerable amount. All these limitations have contributed to the evolution of Market based federated cloud computing model where the market directory allows the participants to locate providers or customers with the right offers. Such markets can bridge disparate Clouds allowing consumers to choose a provider that suits their requirements by either executing SLAs in advance or by buying capacity on the spot. The cloud providers can set the price based on the market condition and can pass the offer to cloud market. The cloud participating in federation can send the VM request to the Market Exchange. The cloud market fulfills the request after analyzing the bids, offers etc. The cloud controllers can negotiate with the market on price and the negotiation process continues until SLA is formed or the participant decides to break off. The idea of utility markets for computing resources has been around for a long time. Many research projects like SHARP, Tycoon, Bellagio, GRACE and Shirako [131] have come up with market structures for trading in resources.

Considerable amount of research has been done in academics for defining the models for trading computing utilities with a specific focus on the design of market oriented schedulers for grid computing systems. The schedulers proposed for the grid will also be operational for the cloud computing as it leverages pre-existing distributed computing technologies including grid computing. The taxonomy of the market-based schedulers is presented in Figure 6.2. The schedulers are categorized based on the allocation decision, objective, market model, application model and participant focus. Along the dimensions the scheduling can be categorized as 1) Game Theory 2) proportional share 3) commodity market 4) posted price 5) Contract-Net. 6) Bargaining 7) Auction. In Game theory based scheduler, the participants interact in the form of allocation game. Proportional schedulers are proposed to suit the market oriented architectures. The user request is directly proportional to the share of the cluster. In this case the user is allocated the resource strictly equal to the request. In the commodity market the resource providers specify the price and the consumer are charged on the basis of the specified price. The posted price is similar to the commodity market but it provides some discount options for the new customers. In Contract-Net model the users submits their advertisements and the providers responds with a bid if the request is favorable to them. The user consolidates all the bids and selects the most beneficial to him. In Bargaining based market model the negotiations between resource consumer and the resource provider is carried out until SLA is implemented or it is stopped as no party is interested in trading. In the market oriented scheduler based on the auction, the auctioneer decides the price of the resources based on the bids and the consumer with the highest bid will gain the resource. These schedulers are proposed in academic research and the industrial implementation of the techniques is limited. Some industries like AMAZON have done an experimental implementation of the spot instances. [135]. The concept of the spot instances allows the consumers to bid for the unused AMAZON EC2 capacity and run those instances as far as their bid is higher than the spot instance price. SpotCloud[136] is another virtual marketplace where buyers and sellers can trade for the compute resources. Buyers search for a specific resource and the seller makes available these resources to the buyer to gain profit out of it. Besides being the web portal the SpotCloud can be installed on the private premises and provides the resources based on money.

SORMA[133] supports automated bidding to participate in an auction to bargain with a resource provider that may lead to increased delays for consumers who urgently need the resources. To deal with this delay a fast data structure to store and retrieve the virtual machine information is required. Cost is not the only criteria to select the resource from the market, the virtual machine allocated by the market should be able to execute the task specified by the user. The user sends a request by specifying the CPU, memory, bandwidth etc. to search the virtual machine from the pool of VMs in the market. The market suggests a VM after searching all the VM offers and selects the suitable VM for the request. Distributed hash table is a data structure that is widely used to store and retrieve the data dynamically in clouds, but DHT is based on keyword search queries. The keys are searched for the values. Most the cloud providers are using keyword based queries.

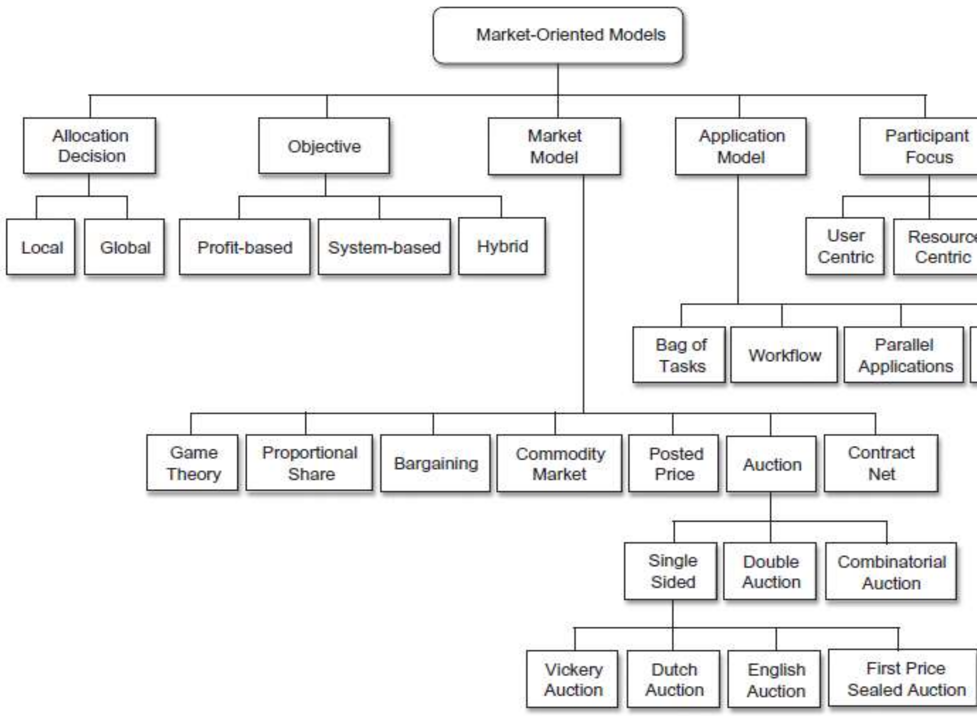


Figure 6.2: Market-Oriented Scheduler Taxonomy [134]

6.4 Market Oriented Cloud Federation

The Federated cloud is based on Utility Computing which promises the delivery of highly scalable and cost effective infrastructure on demand. The cloud providers can participate in the federation based on any of the architectures shown in Figure 6.1. The market based cloud computing is considered to be more suitable for profit oriented architecture. Every cloud participating in federation registers with the market entity, usually a third party entity apart from provider and the customer. The market gathers the preferences of the consumers and the providers to set the transaction prices. In case of the Auction based market the market is the auctioneer which provides the best deal to the consumers as well as the providers. With market oriented federated cloud the consumer is free from the tedious process of going through all the providers to find the best deal. The cloud controllers on the behalf of cloud communicates with the market to request for a resource if the cloud is in shortage of the resources; can offer cloud resources if the resources are idle, can bid for a resource. Auction based architecture is more profitable as the cloud users and the cloud providers both make profit out of it. The high level view of market based cloud architecture is explained in the Figure 6.3.

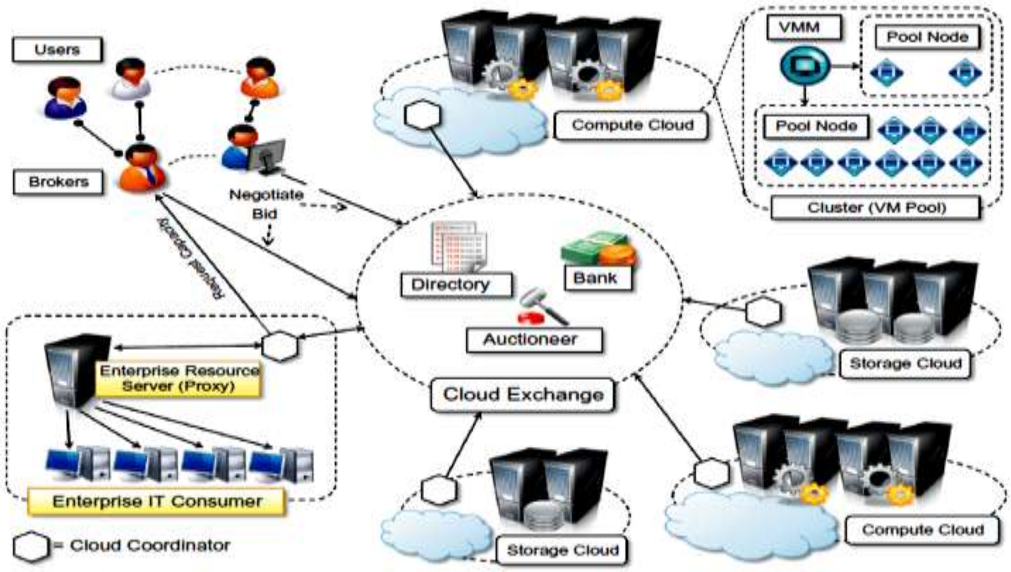


Figure 6.3: High Level View of Market Oriented Cloud Architecture [134]

As shown in the above diagram the market is central point of communication for all the clouds in the federation. The components of the cloud exchange are 1) directory 2) Bank 3) Auctioneer. Market Directory contains the information about all the services available in the cloud exchange. It also helps the cloud coordinator and the broker to search the appropriate service based on request criteria. Auctioneer is in-charge of keeping track of running auctions. The Bank takes care of the financial aspects of all the transactions happening in the marketplace.

6.5 VM Selection Framework for Market Oriented Clouds

The Market Oriented Federated cloud architecture is suitable for maximizing the profit of the cloud providers as well as the consumers. Apart from the profit maximization the objectives of any cloud computing architecture is to provide a flexible and effective scheduling environment, energy efficiency etc. The VM Selection framework described here focus on the selection delay that can occur if we use the traditional data structures. Most of the cloud environments are using distributed hash table for storing and retrieving the data in cloud, but the distributed hash table has the drawback that it can't store multi-dimensional data. VM characteristics are stored in a multi-dimensional object so an improved data structure is required to store and retrieve the VMs from the pool of VMs under reliability constraints.

6.5.1 VM Selection Framework Based on Divided KD tree

The detailed description of the Divided KD tree is presented in Section 4.5. Some of the characteristics of the Divided KD that suits VM selection framework are presented as follows:

A. Divided KD tree is a multi-dimensional tree, which can store the complete detail of the virtual machine as a multidimensional object.
B. Divided KD tree provides exact match queries and the nearest neighborhood queries quickly as compare to Distributed Hash tables; Distributed hash table (DHT) is a data structure that is widely used to store and retrieve the data dynamically in clouds, but DHT is based on keyword search queries. The keys are searched for the values. Most the cloud providers are using keyword based queries.
C. Based on recent information published by Amazon EC2 CloudWatch service, each Amazon Machine Image (AMI) instance has three dimensions associated with it [127]. In the current peer-to-peer literature, multi-dimensional data structures like R* tree, K D tree and Quad tree are used for exact match queries and the range queries in multidimensional databases.
D. Divided KD is dynamic; means the insertion and deletion of the VM objects can be performed at runtime without affecting the cloud Exchange operations. [137]
E. Divided KD tree supports parallel execution of the nearest neighborhood queries. Figure 6.4 shows the proposed VM Selection framework in market oriented federated clouds.

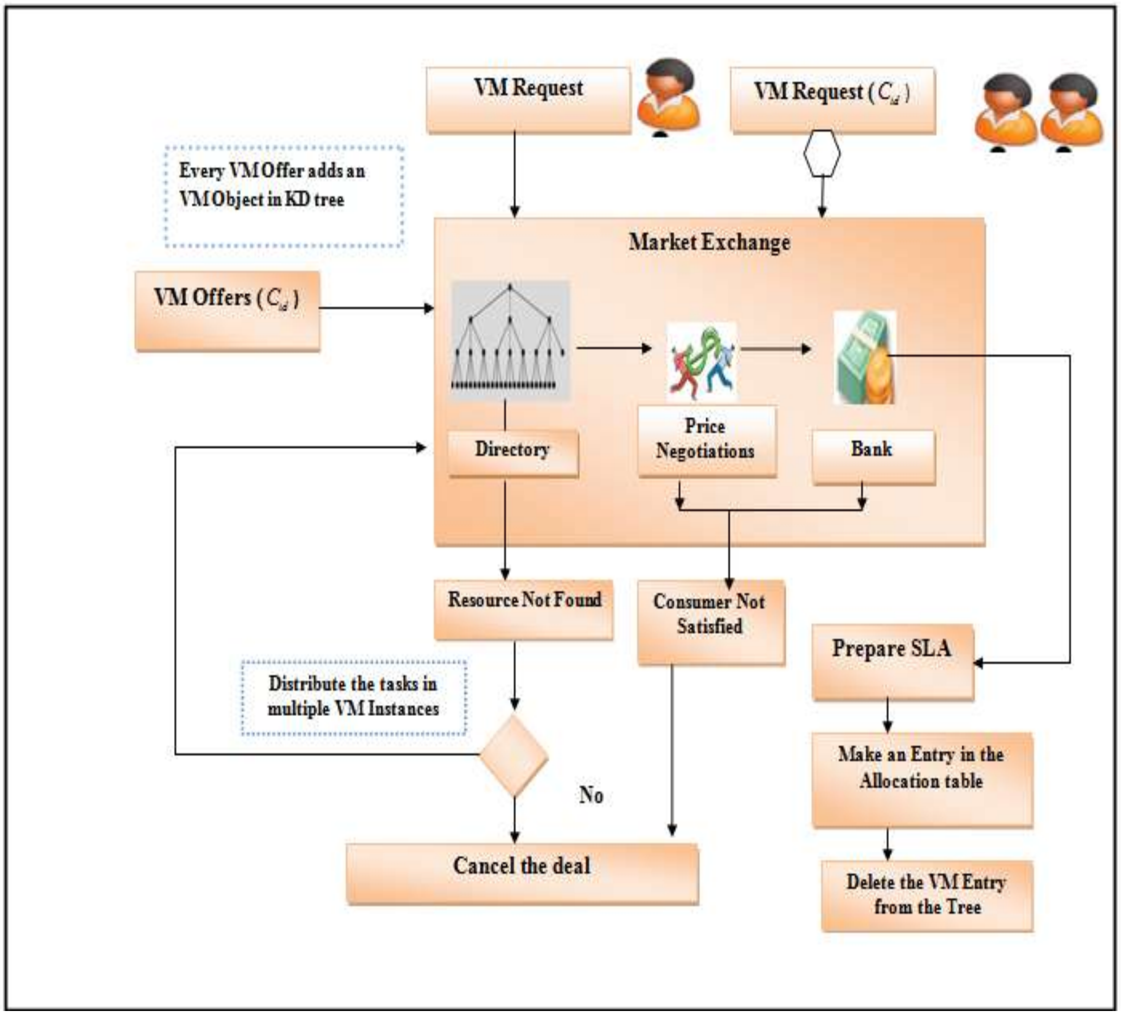


Figure 6.4 VM Selection Framework

The centralized market stores the VMs offered from the cloud providers in the directory as an object in Divided KD tree. The registered users and the cloud providers send a VM request to the market. The VM request is compared with the root if the VM object is smaller than the root then it is searched in the left sub-tree. If the required VM is greater than the root then the search takes place in the right sub-tree. If the required VM is greater than the first value in the root and smaller than the right value then the object is searched in the middle sub-tree. If the exact match is not found then the nearest neighborhood search is performed to get the similar VM to the requested VM. The search query generates two cases:

Case 1: Requested resource found:

The price negotiation between the market and the requesting party takes place. If both the parties are satisfied, SLA is signed between both the parties, an allocation table is updated with the current VM allocation and the VM object is deleted from the divided KD tree. If the negotiation fails then the deal between the parties is canceled.

Case 2: Requested resource Not Found

If the exact requested resource is not found, the resource requestor is asked to distribute the task in more than one virtual machine. If the resource requestor agrees the resource is allocated, allocation table is updated, the VM objects are deleted from the tree, negotiations and transactions take place for the request otherwise the deal is canceled.

Chapter 7

Conclusions and Future Directions

7.1 Introduction

The thesis has proposed and investigated novel technique of implementing Initial VM allocation and VM consolidation in Federated Clouds under workload-independent QoS constraints. The chapter records the results based on various parameters like overload time fraction of any host, SLA violations, performance degradation after migration, number of host shut down after VM consolidation, VM selection mean and Host selection mean for monitoring scheduling delay. The algorithms proposed in the thesis improved the energy efficiency of the cloud federation efficiently but the model can still be enhanced. The future directions are discussed in the Section 7.3.

7.2 Objectives Achieved

The key contributions of the thesis are.

1. Designed and developed an initial VM Allocation algorithm analogous to bin packing algorithm which packs the maximum virtual machine on minimum possible number of hosts.

a. A dynamic data structure (Divided KD tree) is used to search the optimum host for every particular VM request.

b. Apart from exact match queries the KD trees also supports range queries and most importantly multi-attribute queries to make the searching fast as compare to other search data structures.

2. Designed and developed VM consolidation algorithm based on support vector machine for automatic VM consolidation.

   a. Support vector machine analyses the CPU utilization of the host as a time series and predicts the overloaded hosts.

   b. Support Vector Machine is so effective in time series prediction that it has outperformed the artificial neural network based methods.

3. VM Selection Framework for market-oriented federated clouds is also proposed. The framework selects the virtual machines from the pool of virtual machines in market exchange on the basis of multiple attributes specified by the consumer. The parameter of searching includes cost, CPU, memory, disk, MIPS (Millions of Instructions per second etc.)

7.3 Conclusions

The thesis has proposed a novel model to solve immediate practical problem. The proposed approach improves the utilization of the data center resources in such a way that the total energy consumption can be reduced. The proposed algorithm is divided into 2 phases and is explained in chapter 4 and chapter 5 of the thesis. In chapter 4, initial VM allocation algorithm based on the divided KD-tree has been proposed. The algorithm arranges the hosts in the data center as a tree structure making the searching process faster so that scheduling delay can be reduced. In chapter 5, host overload detection algorithm is proposed and the results prove that the proposed algorithm outperformed the statistical algorithm available in the literature.

In chapter 6 we have proposed VM selection framework in market-based cloud. In market-based federated cloud, market is the centralized entity which fulfills the requests of all the service consumers. To fulfill the request a fast data structure is required to search the appropriate VM for each coming request. Divided KD-tee is used as a data structure to store the virtual machine information so that it can be searched faster. Today all famous service providers rely on keyword based search. With the help of Divided KD tree we can search the VM with attributes and multi-attribute queries makes the search faster.

7.4 Future Research Directions

Regardless of substantial contributions of the thesis, there are number of open challenges that need to be addressed in order to further advance the area.

7.4.1 Dynamic Resource Pricing in Auction Based Cloud

The current strategies for resource pricing are based on setting the fixed price of the resource and the user has to bid above the current price [138]. Amazon spot Instances are using the same strategy. But there is a need of more dynamic pricing scheme that monitors the previous bidding patterns and can select the dynamic price of the resource based on the user preference.

7.4.2 Minimizing Global Latency

Since various clouds with different networking properties will be participating in cloud federation. Network latency can be a major issue which can result in degrading overall user experience. Considerable amount of research is required to place the virtual machine near the location of the user eliminating the network traffic. A network aware VM allocation strategy needs to be investigated which automatically provisions the virtual machine near user location. Divided KD tree is also used to solve this problem because it is a space partitioning tree so the data centers of federated clouds can be arranged based on their location.

7.4.3 Governance laws for energy efficiency of data centers

Generally data centers do not publish the complete details about the carbon footprints and because of the lack of common governance laws the energy efficiency standards can’t be maintained. So an extensive research is required in this area to have standardized governance over the data centers energy efficiency.

7.4.4 Excessive Power Cycling vs Reliability

Power cycling means turning a piece of equipment usually computer on and off and then again on and do on. Frequent power cycling of equipment can results in causing thermal stress and also degrades the overall performance of the system. VM consolidation algorithm should be designed in such a way that frequent power cycling can be minimized so that it doesn't affect the reliability of the cloud

7.4.5 Consistency in federated environment

Since the database in federated clouds is also federated between various distributed geographic locations monitored by different cloud domains. In a centralized database system the consistency is easy to be maintained but in distributed environment it is a challenging task as the data is partitioned and distributed in federation. A novel consistency model needs to be developed to ensure consistency in federated clouds.

List of References

[1] Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. “Mastering cloud computing: foundations and applications programming”. Newnes, 2013.

[2] Grance, T., R. Patt-Corner, and J. B. Voas. "Cloud Computing Synopsis and Recommendations." NIST Special Publication (2012): 800-146.

[3] Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. Cloud computing: principles and paradigms. Vol. 87. John Wiley & Sons, 2010.

[4] Li, Songnian, and Jiansheng Gong. "Mashup: a new way of providing web mapping/GIS services." ISPRS Congress Beijing. 2008.

[5] Müller, Hausi A., et al. “Autonomic computing”. No. CMU/SEI-2006-TN-006. Carnegie-mellon univ Pittsburgh pa software engineering institute, 2006.

[6] Erl, Thomas. Service-oriented architecture: concepts, technology, and design. Pearson Education India, 2005.

[7] Huebscher, Markus C., and Julie A. McCann. "A survey of autonomic computing—degrees, models, and applications." ACM Computing Surveys (CSUR) 40.3 (2008): 7.

[8] Eisen, Morty. "Introduction to Virtualization." The Long Island Chapter of the IEEE Circuits and Systems (CAS) Society (2011).

[9] Kirsten Ferguson-Boucher, "Cloud Computing: A Records and Information Management Perspective", IEEE Security & Privacy, vol.9, no. 6, pp. 63-66, November/December 2011, doi:10.1109/MSP.2011.159

[10] Youseff, Lamia, et al. "Understanding the cloud computing landscape."Cloud Computing and Software Services (2010): 1.

[11] Lobo, Melvin, and Preeti Khanna. "Cloud Computing:“The Silver Lining”."International Journal of Computer Applications 56.9 (2012).

[12] Bhardwaj, Sushil, Leena Jain, and Sandeep Jain. "Cloud computing: A study of infrastructure as a service (IAAS)." International Journal of engineering and information Technology 2.1 (2010): 60-63.

[13] Dillon, Tharam, Chen Wu, and Elizabeth Chang. "Cloud computing: issues and challenges." Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. Ieee, 2010.

[14] http://www.quotecolo.com/how-to-choose-the-best-green-cloud-hosting-provider/

[15] Gandhi, Anshul, et al. "Minimizing data center sla violations and power consumption via hybrid resource provisioning." Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, 2011.

[16] Buyya, Rajkumar, James Broberg, and Andrzej M. Goscinski, eds. Cloud computing: principles and paradigms. Vol. 87. John Wiley & Sons, 2010.

[17] Gahlawat, Monica, and P. Sharma. "Survey of virtual machine placement in federated clouds." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.

[18] Gahlawat, Monica, and Priyanka Sharma. "Green, Energy-Efficient Computing and Sustainability Issues in Cloud." Managing and Processing Big Data in Cloud Computing (2016): 206.

[19] Gahlawat, Monica, and Priyanka Sharma. "Reducing the cost of virtual machine migration in federated cloud environment using component based vm." Journal of Information Systems and Communication 3.1 (2012): 288.

[20] Monica Gahlawat and Priyanka Sharma. 2015. A Multi-Objective Initial Virtual Machine Allocation in Clouds using Divided KD Tree. In Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15), Indu Nair (Ed.). ACM, New York, NY, USA, 656-660. DOI=http://dx.doi.org/10.1145/2791405.2791560

[21] Gahlawat, Monica, and Priyanka Sharma. "Support Vector Machine-Based Model for Host Overload Detection in Clouds." Proceedings of International Conference on ICT for Sustainable Development. Springer Singapore, 2016.

[22] Gahlawat, Monica, and Priyanka Sharma. "VM selection framework for market based federated cloud environment." Computing, Communication & Automation (ICCCA), 2015 International Conference on. IEEE, 2015.

[23] Kaplan J, Forrest W, Kindler N. Revolutionizing data center energy efficiency. San Francisco, CA, USA: McKinsey; 2008.

[24] Global Inter-Cloud Technology Forum. Use Cases and Functional Requirements for Inter-Cloud Computing.Technical Report, Global Inter-Cloud Technology Forum, 2010.

[25] Grozev, Nikolay, and Rajkumar Buyya. "Inter-Cloud architectures and application brokering: taxonomy and survey." Software: Practice and Experience 44.3 (2014): 369-390.

[26] Calcavecchia, Nicolò Maria, Antonio Celesti, and Elisabetta Di Nitto. "Understanding decentralized and dynamic brokerage in federated cloud environments." Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice: Theory and Practice 36 (2012).

[27] Kurze, T., Klems, M., Bermbach, D., Lenk, A., Tai, S., & Kunze, M. (2011). Cloud federation. CLOUD COMPUTING, 2011, 32-38.

[28] Gahlawat, Monica, and Priyanka Sharma. "Green, Energy-Efficient Computing and Sustainability Issues in Cloud." Managing and Processing Big Data in Cloud Computing (2016): 206.

[29] Power, Emerson Network. "Energy logic: Reducing data center energy consumption by creating savings that cascade across systems." A White Paper from the Experts in Business-Critical Continuity (2009).

[30] Anastasi, Gaetano F., Emanuele Carlini, and Patrizio Dazzi. "Smart cloud federation simulations with cloudsim." Proceedings of the first ACM workshop on Optimization techniques for resources management in clouds. ACM, 2013.

[31] https://en.wikipedia.org/wiki/Low-power_electronics

[32] Rasmussen, Neil, and Wendy Torell. "Data Center Projects: Establishing a Floor Plan." American Power Conversion White Paper 144 (2007).

[33] Sahoo, Jyotiprakash, Subasish Mohapatra, and Radha Lath. "Virtualization: A survey on concepts, taxonomy and associated security issues." Computer and Network Technology (ICCNT), 2010 Second International Conference on. IEEE, 2010.

[34] Hameed, Abdul, et al. "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems." Computing (2014): 1-24.

[35] Chase J, Anderson D, Thakur P, Vahdat A (2001) Managing energy and server resources in hosting centers. In: 18th symposium on operating systems principles (SOSP ’01), pp 103–116

[36] Hermenier F, Loriant N, Menaud J (2006) Power management in grid computing with Xen. Lecture notes in computer science. Springer, Berlin Han, Yongkui, Israel Koren, and Csaba Andras Moritz. "Temperature aware floorplanning." Workshop on Temperature Aware Computer Systems. Vol. 24. 2005.

[37] Ranjan R, Benatallah B (2012) Programming cloud resource orchestration framework: operations and research challenges. CoRR abs/1204.2204

[38] Kalyvianaki E (2009) Resource provisioning for virtualized server applications. Technical Report UCAM-CL-TR-762, Computer Laboratory, University of Cambridge

[39] Chen Y, Gmach D, Arlitt M, Marwah M, Gandhi A (2011) Minimizing data center SLA violations and power consumption via hybrid resource provisioning. In: Second international green computing conference (IGCC 2011), pp 1–8

[40] Xie T, Wilamowski B (2011) Recent advances in power aware design. In: 37th annual conference on IEEE industrial electronics society IECON 2011, pp 4632–4635

[41] Jansen, Ryan, and Paul R. Brenner. Energy efficient virtual machine allocation in the cloud. Green Computing Conference and Workshops (IGCC), 2011 International. IEEE, 2011.

[42] https://en.wikipedia.org/wiki/Bin_packing_problem

[43] Mohialdeen, Isam Azawi. "Comparative study of scheduling al-gorithms in cloud computing environment." Journal of Computer Science 9.2 (2013): 252.

[44] http://support.citrix.com/article/CTX135790

[45] Pal, A. Souvik, and B. Prasant Kumar Pattnaik. "Classification of virtualization environment for cloud computing." Indian Journal of Science and Technology 6.1 (2013): 3965-3971.

[46] Sharma, Viney, and Gur Mauj Saran Srivastava. "Energy efficient architectural framework for virtual machine management in IaaS clouds." Contemporary Computing (IC3), 2013 Sixth International Conference on. IEEE, 2013

[47] Beloglazov, Anton, and Rajkumar Buyya. “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers”. Concurrency and Computation: Practice and Experience 24.13 (2012): 1397- 1420.

[48] Bhuiyan, Mohammad Fozlul Haque, and Chun Wang. "Energy-Efficient Virtual Machine Management in Heterogeneous Environment: Challenges, Approaches and Opportunities." Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE, 2013.

[49] Bo Yin and Lin Lin, "Energy reducing dynamic multi-dimensional resource allocation in cloud data center," in Network Operations and Management Symposium (APNOMS), 14th Asia-Pacific, 2012, pp. 1-4.

[50] S. Srikantaiah, A. Kansal and F. Zhao, "Energy aware consolidation for cloud computing," in Proceedings of the 2008 Conference on Power Aware Computing and Systems, San Diego, California, 2008, pp. 10-10.

[51] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010, pp. 826-831.

[52] X. Li, Z. Qian, S. Lu and J. Wu, "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center," Mathematical and Computer Modelling, vol. In Press, pp. Available online, 2013.

[53] Seiden, Steven S. "On the online bin packing problem." Journal of the ACM (JACM) 49.5 (2002): 640-671.

[54] Knauth, Thomas, and Christof Fetzer. "Energy-aware scheduling for infrastructure clouds." Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on. IEEE, 2012.

[55] M. M. Taheri and K. Zamanifar, "2-phase optimization method for energy aware scheduling of virtual machines in cloud data centers," in Internet Technology and Secured Transactions (ICITST), 2011 International Conference for, 2011, pp. 525-530.

[56] Li, Xin, et al. "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center." Mathematical and Computer Modelling 58.5 (2013): 1222-1235.

[57] Meisner, David, Brian T. Gold, and Thomas F. Wenisch. "PowerNap: eliminating server idle power." ACM Sigplan Notices. Vol. 44. No. 3. ACM, 2009.

[58] Basmadjian, Robert, Florian Niedermeier, and Hermann De Meer. "Modelling and analysing the power consumption of idle servers." Sustainable Internet and ICT for Sustainability (SustainIT), 2012. IEEE, 2012.

[59] Sedaghat, Mina, et al. "Divide the task, multiply the outcome: Cooperative VM consolidation." Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on. IEEE, 2014.

[60] Farahnakian, Fahimeh, Pasi Liljeberg, and Juha Plosila. "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning." Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on. IEEE, 2014.

[61] Tibor Horvath , Tarek Abdelzaher , Kevin Skadron , Xue Liu, “Dynamic Voltage Scaling in Multitier Web Servers with End-to-End Delay Control”, IEEE Transactions on Computers, v.56, p.444-458, 2007

[62] Xiaotao Liu , Prashant Shenoy , Weibo Gong, “A time series-based approach for power management in mobile processors and disks”, Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video, June 16-18, 2004.

[63] David C. Steere, Ashvin Goel, Joshua Gruenberg, Dylan McNamee, Calton Pu, and Jonathan Walpole, “A Feedback-driven Proportion Allocator for Real-Rate Scheduling”, In Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI), pages 145–158, February 1999.

[64] Graubner, Pablo, Matthias Schmidt, and Bernd Freisleben. "Energy-efficient virtual machine consolidation." IT Professional 2 (2013): 28-34.

[65] Sekhar, Jyothi, Getzi Jeba, and S. Durga. "A survey on energy efficient server consolidation through vm live migration." International Journal of Advances in Engineering & Technology 5.1 (2012): 515-525.

[66] Hirofuchi, Takahiro, et al. "Making vm consolidation more energy-efficient by postcopy live migration." CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization. 2011.

[67] Lee, Sangmin, et al. "Validating heuristics for virtual machines consolidation."Microsoft Research, MSR-TR-2011-9 (2011).

[68] Corradi, Antonio, Mario Fanelli, and Luca Foschini. "VM consolidation: A real case based on OpenStack Cloud." Future Generation Computer Systems 32 (2014): 118-127.

[69] http://docs.media.bitpipe.com/io_10x/io_101335/item_439646/VE_ch1_v2_final.pdf

[70] Kandananond, Karin. "A comparison of various forecasting methods for autocorrelated time series." International Journal of Engineering Business Management 4 (2012).

[71] Zhang, G. Peter. "Time series forecasting using a hybrid ARIMA and neural network model." Neurocomputing 50 (2003): 159-175.

[72] Cao, Lijuan, and Francis EH Tay. "Financial forecasting using support vector machines." Neural Computing & Applications 10.2 (2001): 184-192.

[73] Xu, Gaochao, Yushuang Dong, and Xiaodong Fu. "VMs Placement Strategy based on Distributed Parallel Ant Colony Optimization Algorithm." Applied Mathematics & Information Sciences 9.2 (2015): 873.

[74] Kertesz, Attila. "Characterizing Cloud Federation Approaches." Cloud Computing. Springer International Publishing, 2014. 277-296.

[75] U. Bellur, C. S. Rao, and M. K. SD, “Optimal placement algorithms for virtual machines,” CoRR, vol. abs/1011.5064, 2010.

[76] Saldanha, Hugo, et al. Towards a hybrid federated cloud platform to efficiently execute bioinformatics workflows. INTECH Open Access Publisher, 2012.

[77] Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. Mastering cloud computing: foundations and applications programming. Newnes, 2013.

[78] Gahlawat, Monica, and P. Sharma. "Survey of virtual machine placement in federated clouds." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.

[79] Fiandrino, Claudio, et al. "Performance and energy efficiency metrics for communication systems of cloud computing data centers." (2015).

[80] Gahlawat, Monica, and Priyanka Sharma. "Green, Energy-Efficient Computing and Sustainability Issues in Cloud." Managing and Processing Big Data in Cloud Computing (2016): 206.

[81] http://www.quotecolo.com/how-to-choose-the-best-green-cloud-hosting-provider/

[82] http://idlebrains.org/virtual-memory/what-is-virtual-memory/

[83] https://docs.oracle.com/cd/E27300_01/E27309/html/vmusg-virtualization-reasons.html

[84] Kim, Jungsoo, et al. "Correlation-aware virtual machine allocation for energy-efficient datacenters." Proceedings of the Conference on Design, Automation and Test in Europe. EDA Consortium, 2013.

[85] Gerald J. Popek and Robert P. Goldberg, “Formal requirements for Virtualizable third generation architectures,” Commun. ACM, 17(7), pp. 412-421, 1974.

[86] Barham, Paul, et al. "Xen and the art of virtualization." ACM SIGOPS Operating Systems Review 37.5 (2003): 164-177.

[87] Ameen, Radhwan Y., and Asmaa Y. Hamo. "Survey of server virtualization."arXiv preprint arXiv:1304.3557 (2013).

[88] Liu, Ming, and Tao Li. "Optimizing virtual machine consolidation performance on NUMA server architecture for cloud workloads." Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on. IEEE, 2014.

[89] http://www.vmware.com/consolidation/overview

[90] Rosenblum, Mendel, and Tal Garfinkel. "Virtual machine monitors: Current technology and future trends." Computer 38.5 (2005): 39-47.

[91] Bugnion, Edouard, Scott W. Devine, and Mendel Rosenblum. "Virtual machine monitors for scalable multiprocessors." U.S. Patent No. 6,075,938. 13 Jun. 2000.

[92] https://oracle-base.com/articles/vm/a-cure-for-virtual-insanity

[93] https://www.vmware.com/pdf/ovf_whitepaper_specification.pdf

[94] Durairaj, M., and P. Kannan. "A study on Virtualization Techniques and Challenges in Cloud Computing." International Journal of Scientific &Technology Research 1.1 (2014).

[95] R. G. Chris Hyser, Bret Mckee and B. J. Watson, “Autonomic virtual machine placement in the data center,” in HP Labs technical report, 2007.

[96] Yamamoto, V. Yoshihiko Oguchi V. Tetsu. "Server virtualization technology and its latest trends." Fujitsu Sci. Tech. J 44.1 (2008): 46-52.

[97] Abdelsamea, Amany, et al. "Virtual Machine Consolidation Challenges: A Review." International Journal of Innovation and Applied Studies 8.4 (2014): 1504.

[98] Saldanha, Hugo, et al. Towards a hybrid federated cloud platform to efficiently execute bioinformatics workflows. INTECH Open Access Publisher, 2012.

[99] Frazer, J., Reptiles. Architectural Design, 1974(April): p. 231-239.

[100] Knecht, K., and R. Koenig. "Generating floor plan layouts with kd trees and evolutionary algorithms." Generative Art Conf. 2010.

[101] Van Kreveld, Marc J., and Mark H. Overmars. "Divided k-d trees."Algorithmica 6.1-6 (1991): 840-858.

[102] Moore, Andrew W. "An intodactory tutorial on kd-trees." (1991).

[103] https://cathyatseneca.gitbooks.io/data-structures-and-algorithms/content/2-3_trees/index.html

[104] http://www-bcf.usc.edu/~dkempe/CS104/11-19.pdf

[105] Samet, Hanan. Foundations of multidimensional and metric data structures. Morgan Kaufmann, 2006.

[106] Camati, Ricardo Stegh, Alcides Calsavara, and Luiz Lima Jr. "Solving the virtual machine placement problem as a multiple multidimensional knapsack problem." ICN 2014 (2014): 264.

[107] Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer." ACM SIGARCH Computer Architecture News. Vol. 35. No. 2. ACM, 2007.

[108] http://www.spec.org/power_ssj2008/

[109] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41.1 (2011): 23-50.

[110] Suzhen, Li, Liu Xiangjie, and Yuan Gang. "Supervisory predictive control of weighted least square support vector machine based on Cauchy distribution."Control and Decision Conference (2014 CCDC), The 26th Chinese. IEEE, 2014.

[111] Zhou, Min, and Qiwan Wang. "The on-line electronic commerce forecast based on least square support vector machine." *Information and Computing Science, 2009. ICIC'09. Second International Conference on*. Vol. 2. IEEE, 2009.

[112] Gui, Bin, et al. "Financial Time Series Forecasting Using Support Vector Machine." Computational Intelligence and Security (CIS), 2014 Tenth International Conference on. IEEE, 2014.

[113] Nguyen, Khanh, et al. "Least square Support Vector Machine for large-scale dataset." Neural Networks (IJCNN), 2015 International Joint Conference on. IEEE, 2015.

[114] Martínez-Álvarez, Francisco, et al. "A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting." Energies 8.11 (2015): 13162-13193.

[115] http://www.clopinet.com/SVM.applications.html

[116] Chen, Bo-Juen, Ming-Wei Chang, and Chih-Jen Lin. "Load forecasting using support vector machines: A study on EUNITE competition 2001." Power Systems, IEEE Transactions on 19.4 (2004): 1821-1830.

[117] Collobert, Ronan, and Samy Bengio. "SVMTorch: Support vector machines for large-scale regression problems." The Journal of Machine Learning Research 1 (2001): 143-160.

[118] Martin, Mario. On-line support vector machine regression. Springer Berlin Heidelberg, 2002.

[119] Chang, Yuqing, et al. "Soft sensing modeling based on stacked least square-support vector machine and its application." Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on. Vol. 1. IEEE, 2006.

[120] Pai, Ping-Feng, and Wei-Chiang Hong. "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms." Electric Power Systems Research 74.3 (2005): 417-425.

[121] Niu, Dongxiao, Yongli Wang, and Desheng Dash Wu. "Power load forecasting using support vector machine and ant colony optimization." Expert Systems with Applications 37.3 (2010): 2531-2539.

[122] Li, Qiong, et al. "Applying support vector machine to predict hourly cooling load in the building." Applied Energy 86.10 (2009): 2249-2256.

[123] LI, Yuan-cheng, Ting-jian FANG, and Er-keng YU. "Study of support vector machines for short-term load forecasting [J]." Proceedings of the Csee 6 (2003): 10.

[124] Beloglazov, Anton. "Energy-efficient management of virtual machines in data centers for cloud computing." (2013).

[125] Niu, D. X., Li, W., Cheng, L. M., & Gu, X. H. (2008, July). Mid-term load forecasting based on dynamic least squares SVMs. In Machine Learning and Cybernetics, 2008 International Conference on (Vol. 2, pp. 800-804). IEEE.

[126] Ojemakinde, Bukola Titilayo. "Support Vector Regression for Non-Stationary Time Series." (2006).

[127] Beslic, Alexandre, et al. "Towards a solution avoiding Vendor Lock-in to enable Migration Between Cloud Platforms." *MDHPCL@ MoDELS*. 2013.

[128] Kurze, Tobias, Markus Klems, David Bermbach, Alexander Lenk, Stefan Tai, and Marcel Kunze. "Cloud federation." CLOUD COMPUTING 2011 (2011): 32-38.

[129] Calcavecchia, Nicolò Maria, Antonio Celesti, and Elisabetta Di Nitto. "Understanding decentralized and dynamic brokerage in federated cloud environments." Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice: Theory and Practice 36 (2012).

[130] Nair, Srijith K., et al. "Towards secure cloud bursting, brokerage and aggregation." Web services (ecows), 2010 ieee 8th european conference on. IEEE, 2010.

[131] Buyya, Rajkumar, Chee Shin Yeo, and Srikumar Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities." High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on. Ieee, 2008.

[132] Broberg, James, Srikumar Venugopal, and Rajkumar Buyya. "Market-oriented grids and utility computing: The state-of-the-art and future directions." Journal of Grid Computing 6.3 (2008): 255-276.

[133] Garg, Saurabh Kumar, Christian Vecchiola, and Rajkumar Buyya. "Mandi: A market exchange for trading utility computing services." Cloud Computing and Distributed Systems Laboratory. The University of Melbourne, Australia (November 17, 2009) (2009).

[134] Garg SK, Buyya R. Cooperative Networking Market-oriented resource management and scheduling: a taxonomy and survey. New York, USA: Wiley Press; 2011.

[135] http://aws.amazon.com/ec2/spot-instances/

[136] www.spotCloud.com.

[137] van Kreveld, Marc J., and Mark H. Overmars. "Dividedk-d trees." Algorithmica6, no. 1-6 (1991): 840-858.

[138] Mihailescu, Marian, and Yong Meng Teo. "Dynamic resource pricing on federated clouds." *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2010.

[139] Cuadrado, Félix, et al. "Research challenges for cross-cloud applications."*Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014.

[140] K. S. Park and V. S. Pai, “CoMon: a mostly-scalable monitoring system for Planet- Lab,” ACM SIGOPS Operating Systems Review, vol. 40, no. 1, pp. 65–74, 2006.

List of Publications

Book Publication

1. Green, Energy-Efficient Computing and Sustainability Issues in Cloud, published in Managing and processing Big Data in cloud in IGI global publications, 2016.
2. An Insight into Cloud Computing –In Press.

Conference/Publications

1. Gahlawat, Monica, and Priyanka Sharma. "Support Vector Machine Based Model for Host Overload Detection in clouds" Proceedings of International Conference on ICT for Sustainable Development (ICT4SD 2015 Volume 1)
2. Gahlawat, Monica, and Priyanka Sharma. "A Multi-Objective Initial Virtual Machine Allocation in Clouds using Divided KD Tree." Proceedings of the Third International Symposium on Women in Computing and Informatics. ACM, 2015
3. Gahlawat Monica and Priyanka Sharma. "VM Selection Framework for Market Based Cloud Environment" Computing, communication & Automation (ICCCA), 2015 International Conference on. IEEE, 2015.
4. Gahlawat, Monica, and Priyanka Sharma. "Survey of virtual machine placement in federated clouds." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.
5. Gahlawat, Monica, and Priyanka Sharma. "Analysis and Performance Assessment of CPU Scheduling Algorithms in Cloud using Cloud Sim."Analysis 5.9 (2013).
6. International Conference on Reliability, Infocom technologies and Optimization (IEEE and CSI Supported) Jan 29-31, 2013
7. International conference on Advances in Cloud Computing(CSI Bangalore Chapter) July 26th-28, 2012

8. Gahlawat, Monica, and Priyanka Sharma. "Reducing the cost of virtual machine migration in federated cloud environment using component based vm." Journal of Information Systems and Communication 3.1 (2012): 288.