CONTEXT AWARE DATACENTER LOAD BALANCER

Thesis

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

ASHWIN KUMAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,
SURATHKAL, MANGALORE - 575025

June 2020

DECLARATION

by the Ph.D. Research Scholar

I hereby **declare** that the Research Thesis entitled **Context Aware Datacenter Load Balancer** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfilment of the requirements for the award of the Degree of **Doctor of Philosophy** in **Computer Science and Engineering** is a **bonafide report of the research work carried out by me**. The material contained in this Research Thesis has not been submitted to any University or Institution for the award of any degree.

ASHWIN KUMAR

(**CS13P01, Ashwin Kumar**)
(Register Number, Name & Signature of Research Scholar)
Department of Computer Science and Engineering

Place: NITK, Surathkal.
Date: June 29, 2020

CERTIFICATE

This is to *certify* that the Research Thesis entitled **Context Aware Datacenter Load Balancer** submitted by **Ashwin Kumar**, (Register Number: **CS13P01**) as the record of the research work carried out by him, is *accepted as the Research Thesis submission* in partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy**.

Dr. Annappa B
Research Supervisor
(Name and Signature with Date and Seal)

Dr. Alwyn Roshan Pais
Chairman - DRPC
(Signature with Date and Seal)

Acknowledgments

In this incredible journey of my Ph.D., I had the fortune to meet and work with many outstanding people, without whom I could never have done it. Their encouragement has allowed me to push the envelope beyond what I initially thought to be feasible. For this reason, I would like to mention the individuals to whom I owe my genuine admiration and thankfulness.

I wouldn't know what was research and its depth if I was not given an opportunity to do it. A significant share of the credit is due to my research supervisor, **Prof.Annappa** who gave me a chance to work with him in the significant field of cloud computing. There are absolutely no words how much I am grateful to my research supervisor who supported me throughout my journey of ups and downs. His constant support, patience, guidance, and supervision accorded me to focus on my research work. His unrelenting passion for quality and soundness in research, and his incredible professionalism, work ethic, and forgiving nature have been a defining inspiration for my endeavors. Thank you, sir, for always leading me the right path.

I am enormously thankful to the Research Progress Assessment Committee members **Dr. Mohit P. Tahiliani** and **Dr. R. Madhusudhan** for their insightful comments, critical questions, and valuable ideas. Their continuous interest in my research progress and their sharp and quick feedback in all matters has greatly helped me in achieving research-related objectives.

I humbly thank **Dr. Alwyn Roshan Pais**, HoD, and Chairman(DRPC) and **Dr. Basavaraj Talwar**, Secretary (DRPC) for helping me in research related aspects. I should be thankful for the support and advice received from **Dr. Shashidhar G Koolagudi**.

My journey during Ph.D. wouldn't have been exciting if I had not met seniors

Dr. Likewin Thomas, **Dr. Sumith** and **Dr. Manoj**. They did the hand holding during my initial duration of the course, and I also share some memorable fun moments with them.

I should be thankful to my friends **Mrs. Keerthi Shetty** and **Mr. Vishnu** for helping with many institutional procedures and guiding me with much course-related information. The Feedbacks and bits of advice given by them also helped me in improving the quality of my research related documents to many folds. I want to thank also my juniors **Mrs. Saraswati** and **Mr. Manjunath** for their help in reaching people and resources when I am away from institution. I should also thank **Dr. Priyaranjan Sharma** for sharing a room in the hostel and also being there for me when I need something during course work.

I should be thankful from the bottom of my heart to colleagues at Sony **Mr.Uday Kiran A** and **Mr.Prashanth P** for supporting me throughout my journey of Ph.D. The amount of trust they have in me encouraged me to further my research with any worries. I should also thank **Mr.Jaison Joseph** and **Dr.Kanchana Gopinath** for motivating me to take up research and their inspiring thoughts to excel in my profession.

I want to thank my close friends **Mr. Vishnu**,**Mr. Malleshi**,**Mr. Bahubali**, **Mr. Rudragouda**, **Mr. Harish**, **Mr. santosh** for supporting and motivating me throughout my journey of Ph.D.

I would humbly thank the non-teaching staff of my department. **Mr. Dinesh Kamath**, and **Mrs. Yashawanthi** were supportive in ensuring that research-related seminars go well and uninterrupted.

I want to thank my parents, **Dr. Dattatraya Kulkarni**, **Mrs. Maya Kulkarni** and my wife **Mrs.Priyanka** for being the main pillars of this journey and for constant and unconditional love towards me. They understood and encouraged me in all my decisions. Without their support, this thesis would not have been possible. Without their constant encouragement, this journey would be left incomplete. I should thank my brother **Dr.Chetan** for being there for me during my difficult phases. I should also thank my little son **Avnish** for helping me get rid of tiredness with his smiles. I would like to thank my aunt **Mrs. Chaya Kulkarni**, she has been a friend, guide and mentor throughout my studies. With this, I would like to thank all of my family members for all the support and encouragement I have got always.

In the end, all that remains is to thank you, dear reader. If you have found at least a small part of this thesis useful or interesting, you have made all my work worthwhile

Finally my inmost gratitude towards almighty for helping me get through this!

Place: Surathkal **Ashwin Kumar**

Date: June 29, 2020

Abstract

The ever increasing demand for cloud adoption is prompting researchers and engineers around the world to make the cloud more efficient and beneficial for all the stakeholders that include cloud service providers and cloud service users. Cloud computing will bring profits for all when the cloud resources are used efficiently, and its services are made affordable for businesses by reducing its cost. Managing cloud data center incurs a high cost, which includes capital expenditure for procuring necessary IT infrastructure at the beginning and recurring operational expenditures for data center management which includes power, manpower and maintenance. Data center owners need to reduce the data center management cost by employing efficient resource provisioning techniques to save energy and reduce cost without affecting the service level agreements.

Load balancing is one of the critical aspects of cloud data centers that can significantly improve resource utilization, performance, and save energy by properly assigning/reassigning computing resources to the incoming requests. Therefore, how to schedule user tasks to virtual machines and virtual machines to physical servers effectively by considering various dynamic parameters is an evolving research problem in cloud computing.

The proposed work investigates contextual parameters such as physical machine characteristics, data center load conditions, and electricity prices in the geo-distributed data center locations to propose energy and cost-efficient load balancing technique for cloud data centers. The physical machine characteristics such as performance to power consumption profile are utilized for virtual machine placement decisions in data centers to optimize total energy consumption and improve throughput. The context of peak and non-peak load conditions is used to avoid virtual machine placement optimization overheads and efficient utilization of power-efficient physical servers. The electricity price varies according to geographical locations throughout the globe. The electricity price, along with response times, is considered to distribute data center loads optimally in geo-distributed data centers to save total power costs. Proposed work also investigates current challenges for efficient graphical processing units resource utilization in virtualized environments.

The work proposes a context-aware load balancing technique that ensures better power-efficient resource utilization, enhances performance by avoiding overheads, and also saves total power costs of the data centers. The experimental results indicated that our proposed context-aware load balancer helps to save around 2-10% of power for synthetic workloads and 1-3% for real workload traces in the data centers. The experimental results also attested that our proposed cost-aware cloud service broker load distribution technique for geo-distributed data centers can save around 15-23% of power costs of the data centers.

Table of Contents

List of Figures

List of Tables

Abbreviations and Nomenclature

Abbreviations

CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DC	Data Center
DC Id	Data Center Identifier
ESCE	Equally Spread Current Execution Load
FCFS	First Come, First Serve
FF	First Fit Policy
FFI	First Fit Increasing Policy
GPU	Graphical Processing Unit
GWS	Global Workload Scheduler
HPC	High Performance Computing
IT	Information Technology
LCM	Local Context Manager
Mbps	Megabits per second
MIPS	Million Instructions Per Second
NAS	Network Attached Storage

OpenCL	Open Computing Language
OS	Operating System
PCIe	Peripheral Component Interconnect express
PM	Physical Machine
QoS	Quality of Service
RAM	Random Access Memory
ROI	Return On Investment
SIMD	Single Instruction Multiple Data
SLA	Service Level Agreement
SM	Symmetrical Multiprocessor
vCPU	Virtualized Central Processing Unit
VDI	Virtual Desktop Interface
vGPU	Virtualized Graphical Processing Unit
VM	Virtual Machine
VMM	Virtual Machine Manager
vRAM	Virtualized Random Access Memory

Chapter 1

Introduction

1.1 Cloud computing

Cloud computing, a long-held dream of offering computing as a utility, has the potential to transform the way a major portion of IT businesses and internet services work. It is an idea that enabled software, hardware, and other services to be rented to a large base of customers with added flexibility. Now businesses with innovative ideas for internet services need not invest large capital for computing hardware or software at the beginning to deploy their services to end-users. This means cloud computing benefits small scale businesses to start operations with minimum capital in setting up IT infrastructure and with pay-as-you-go like model.

Cloud computing(Armbrust et al., 2009) eliminates the necessity to predict the application load in advance for provisioning computing resources. The computing resources in the cloud are scaled as per demand to save wastage of resources by over-provisioning and loss of business by under-provisioning. The human effort to maintain computing hardware and software is also avoided by offloading maintenance to cloud providers. The term cloud computing refers to both software(platform software and application software) offered as internet service along with computing hardware in data centers.

It is estimated that enterprises will spend 33% more on cloud services or solutions in 2019 and also it is predicted that 80% of IT businesses will rely on the cloud instead of conventional infrastructure by 2025. Cloud computing is the fastest-growing market with its investments expected to cross $214 bn in 2019(Jain, 2019).

1.1.1 Definition

The standard definition of cloud computing provided by NIST(Mell and Grance, 2011) is as below;
"Cloud computing is a model for enabling ubiquitous, 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."

1.1.2 Characteristics

Cloud computing infrastructure is made up of powerful computing nodes (composed of physical hardware and software entities) and large storage units which are connected by a high-speed network. The cloud services are made available through internet protocols for users located anywhere in the globe. The abstraction layer called hypervisor(or VMM) sitting above the physical layer decouples user workloads from underlying physical resources. The abstraction layer manifests the typical cloud characteristics. The five important characteristics typical to any cloud computing infrastructure are described briefly in this section.

1. **On demand self-service** - Consumers can provision resources from cloud unilaterally without the need for human intervention from cloud service providers.

2. **Broad network access** - Cloud computing capabilities are available over a network and can be accessed using standard network protocols from heterogeneous client devices.

3. **Resource pooling** - Cloud service providers pool resources to multiple customers using a multi-tenant model, wherein different virtual and physical resources are assigned and re-assigned dynamically as per the changing demands

from customers. Examples of resources are storage, memory, compute power, and network bandwidth.

4. **Elasticity** - Cloud capabilities can be provisioned elastically and released automatically. The feature allows the application to scale up or down as per current demand, often providing the user an illusion of provisioning capability of unlimited resources.

5. **Measured service** - Cloud resources provisioned are monitored, controlled, and reported providing transparency for both users and cloud providers to enable fair costing as per terms and usage.

1.1.3 Service models

The cloud computing services are offered in three distinct service models(Mell and Grance, 2011) to suit different customer requirements, as shown in figure 1.1.

1. **Software as a Service (SaaS)**

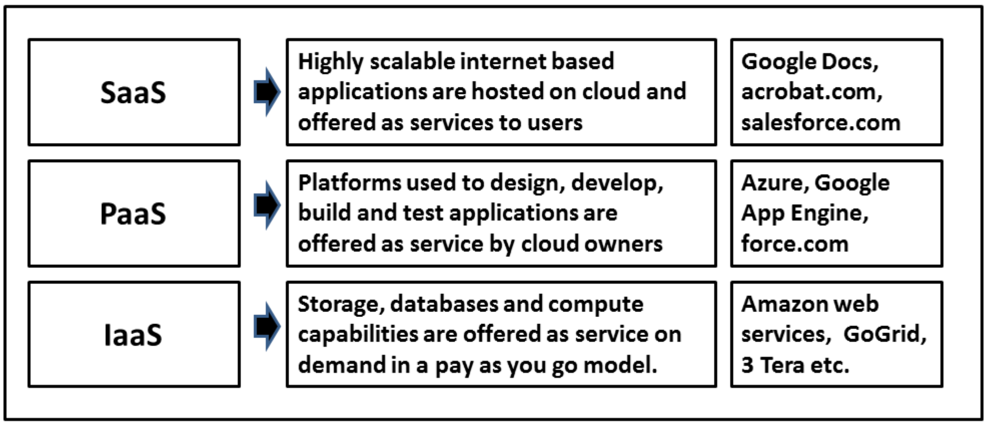
   The applications deployed on the cloud are offered as service to customers. Cloud users can access these applications using thin clients through web browsers or software tools. Except for limited user-specific application configuration settings, customers need not bother about managing or controlling application software or underlying cloud infrastructure.

2. **Platform as a Service (PaaS)**

   The development environment encapsulated into a software layer and is offered as service, upon which other higher levels of service can be built by the customers. Users have the freedom to build, configure, and run their applications making use of the abstracted APIs provided by the platform. Customers need not worry about managing the software platform offered as service by cloud providers.

3. **Infrastructure as a Service (IaaS)**

   In this model, fundamental resources such as compute power, memory, storage, and network are provisioned to customers as a service. The customer can typically deploy his or her software on the provisioned infrastructure to build and offer any application services to their clients. Such provisioned resources can be accessed through a simple command-line tool or a lightweight user interface. Customers need not be bothered about managing the underlying cloud infrastructure offered to them.



SaaS	Highly scalable internet based applications are hosted on cloud and offered as services to users	Google Docs, acrobat.com, salesforce.com
PaaS	Platforms used to design, develop, build and test applications are offered as service by cloud owners	Azure, Google App Engine, force.com
IaaS	Storage, databases and compute capabilities are offered as service on demand in a pay as you go model.	Amazon web services, GoGrid, 3 Tera etc.

Figure 1.1: Cloud Service Models

With IaaS used to host, PaaS used to build, and SaaS used to consume, three of these cloud computing models enable networked access to a pool of shared configurable resources such as servers, networks, storage, applications and services on demand.

1.1.4 Deployment models

The cloud infrastructure is set up and accessed using one of the four deployment models(Armbrust et al., 2009) as per the business needs. The cloud deployment model represents a specific genre of cloud environment, distinguished primarily by size, type of ownership, and access. The four popular deployment models are explained below.

1. **Public clouds**

   The public clouds are publicly accessible cloud environment hosted by a third party cloud providers. The cloud services are offered on demand for a defined cost. Here cloud providers are responsible for the creation and management of public clouds and its information technology(IT) resources. Public clouds are characterized by elasticity and utility pricing in the provisioning of their resources.

2. **Community clouds**

   Community clouds are in a way similar to public clouds except that the access to community clouds is restricted to a specific group of cloud users involved in a shared goal. It can be owned, managed, and operated by one or more organizations in the community or by a third party cloud provider.

3. **Private clouds**

   The private clouds are provisioned and managed for the exclusive use of a single organization and used by cloud consumers of different departments. The private clouds are either owned and managed by a single organization or by a third party cloud provider. The private cloud can exist on or off the company premises.

4. **Hybrid clouds**

   The hybrid clouds are a composition of two or more of the deployment models governed by a set of business rules. The hybrid clouds can be complex architectures for creation and management because of heterogeneous cloud environments and split cloud management responsibilities between public cloud providers and private cloud owners. Hybrid clouds are used when organizations restrict movement or storage of sensitive data into public clouds.

The cloud deployment models are shown in figure 1.2, and the differences between each of the models(Jain, 2019) are presented in table 1.1.

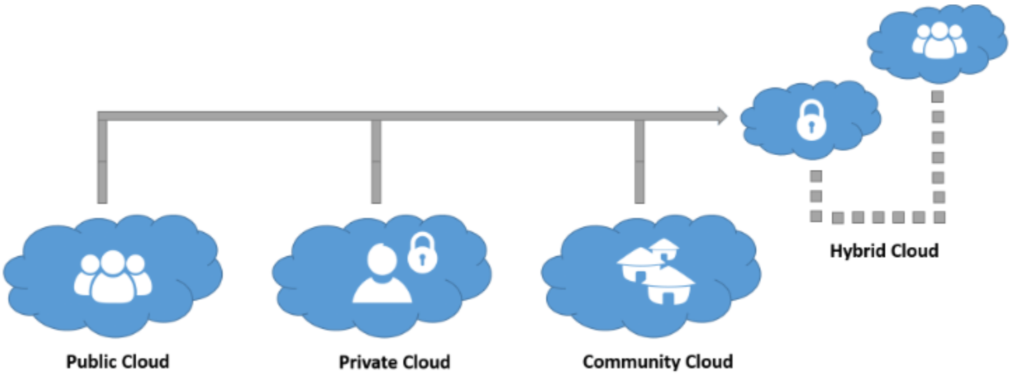


Figure 1.2: Cloud Deployment Models

Table 1.1: Comparison Of Cloud Deployment Models

Parameter	**Private**	**Public**	**Community**	**Hybrid**
Data security and privacy	High	Low	Comparatively High	High
Scalability and Flexibility	High	High	Fixed Capacity	High
Ease of setup and use	Requires IT expertise	Easy	Requires IT expertise	Requires IT expertise
Reliability	High	Vulnerable	Comparatively High	High
Cost Effectiveness	Most Expensive	Cheapest	Cost is shared among community	Cheaper than private but costlier than public

1.1.5 Advantages

Some of the major benefits associated with using cloud computing services for businesses are mentioned below,

1. **Investment cost**

   The cloud computing services free businesses from high capital investments for the hardware and software at the beginning. Lower initial investments help smaller businesses to start operations with smaller capital.

2. **Availability**

   Cloud providers ensure round the clock reliable services to the customers by maintaining 99.9% uptime for servers.

3. **Scalable capacity**

   The services provided by the cloud can be scaled both upwards and downwards as per dynamically changing resource demands. The scalability helps businesses rapidly increase service capacity with an increase in demand and optimize costs by reducing capacity during non-peak seasons.

4. **Carbon footprint**

   Cloud services help organizations to reduce carbon footprints by allocating computing resources that are just sufficient to meet current demands and avoiding any over-provisioning.

5. **Maintainability**
   Cloud computing exempts users from IT maintenance worries and provides simplified ways to manage and control the rented services. The quality and continuity of user services are guaranteed by the SLA agreements.

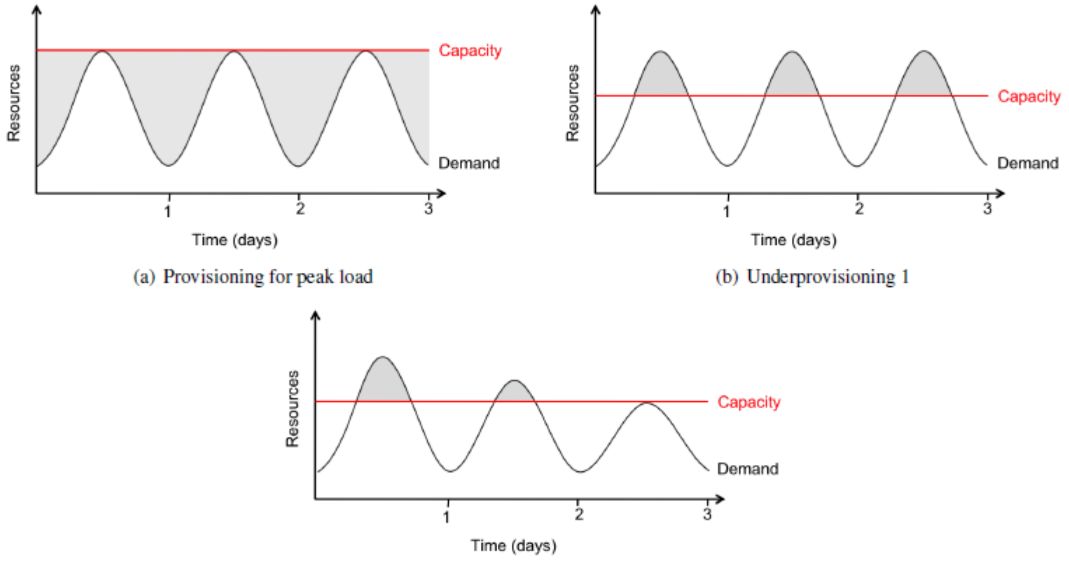
1.2 Virtualization in cloud computing

Cloud computing can exist without virtualization, but it would be difficult and inefficient. Cloud computing without virtualization can then be referred to as a situation in which computing resources, software, or platforms are delivered as a service and on-demand over the Internet. Virtualization is a key enabler technology and a vital factor in the success story of cloud computing. Virtualization(Vmware, 2019) technology makes cloud infrastructure elastic, efficient and fault-tolerant.

Consider three cases shown in figure 1.3, though the peak load is accurately determined and resource provisioning is done, there is a resource wastage in (a) case. The cases (b) and (c) show how changing resource demands cause loss of business through under-provisioning and also resource wastage. Virtualization helps companies to mitigate mismatches caused by resource demands and allocation in run time.

1.2.1 Virtualization

Virtualization(Vmware, 2019) is the process of creating a software-based, or virtual, representation of something, such as virtual servers, storage, and networks. Figure 1.4 shows a typical system stack of a virtualized physical host. The hypervisor(or virtual machine manager) is a thin software layer that enables multiple virtual machines(VM), each running its own copy of the guest operating system(OS) to run simultaneously on a single physical machine. The hypervisor provides an abstracted hardware version to each of the running virtual machines and multiplexes underlying hardware resources efficiently. OS running inside each virtual machine(VM) assumes complete control of the underlying hardware and the virtualization framework through a hypervisor layer provides this illusion to VMs. Each VM runs independently and in



(a) Provisioning for peak load

(b) Underprovisioning 1

(c) Underprovisioning 2

Figure 1.3: Instances Of Over And Under Provisioning

isolation, so that run time problems in one VM do not affect other co-located VMs on the same physical host.

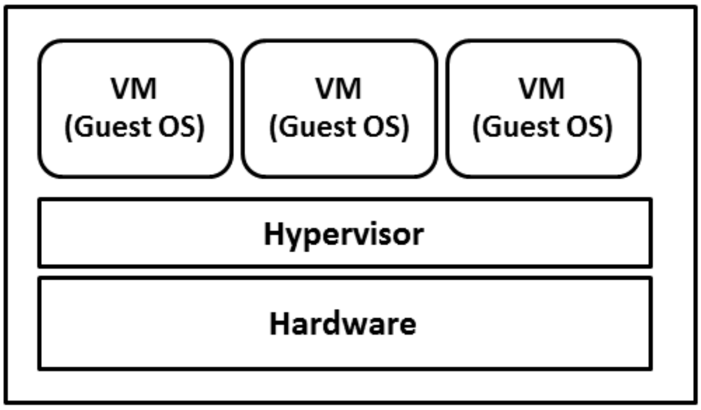


Figure 1.4: Virtualized Physical Machine

Virtualization is most effective in reducing IT expenses and helps improving resource efficiency and agility in the business operations of all scales. Virtualization enables data center providers to manage resource demands from users with a fewer number of physical machines to save power and reduce cost.

1.2.2 Characteristics of virtualization

The following are the key characteristics of virtualization(Vmware, 2019) technology.

A Partitioning

Virtualization enables multiple operating systems to run on the same physical machine. The underlying system resources are divided between multiple virtual machines.

B Isolation

VMs run in isolation and failure of a VM have no impact on other co-located VMs. Virtualization technology provides security and fault isolation at the hardware level.

C Encapsulation

The state of the virtual machines can be saved to a file. It is also possible to copy or move virtual machines as easily as moving and copying a file.

D Hardware independence

Virtual machines can be provisioned or migrated on to any physical host(server). It helps in server consolidation and load balancing data center workload.

1.2.3 Benefits of virtualization

The following are some of the benefits of virtualizing resources in data centers.

1. Instant provisioning and on-demand scalability
2. Live migration support
3. Optimization of resource utilization
4. Server consolidation to save power and load balancing physical resources for better response times.

5. Ease of maintenance and low downtime

6. Security and fault isolation

7. Simplified data center management

Virtualization enables data centers to self manage computing infrastructure in ever-changing load conditions using dynamic load balancing techniques.

1.3 Load balancing in cloud data centers

A distributed system such as cloud data centers can be viewed as a collection of heterogeneous computing, storage, and network resources shared between active users. The users of such a distributed system have different goals, specific objectives, and business-driven strategies, and their behaviors are complex to characterize. In such a complex system, the management of hardware resources and software platforms /applications is a very intricate task. The goal of load balancing is to improve the performance and efficiency of such a distributed system through uniform and fair distribution of the application load across available computing nodes. Load balancing is a critical aspect in the cloud computing environment that helps to improve resource utilization, enhances performance, and saves energy by efficiently assigning/reassigning computing resources to the user workloads/requests.

A general formulation(Grosu and Chronopoulos, 2004) of the load balancing problem is, given a large number of tasks, find the allocation of tasks to computing infrastructure optimizing a given objective function (e.g., total execution time). Load Balancing in the cloud is a method(Geeta and Singh, 2014) to distribute workloads across many servers, network interfaces, hard drives, or other computing resources. Cloud data centers are composed of large, powerful (and expensive) computing servers, storage and are connected by the network infrastructure. These resources are associated with usual risks of hardware failures, power interruptions, and resource overloads during high demands.

Load balancing in cloud computing differs from classical thinking on load-balancing architecture(S.Jyothsna, 2016). Load balancing virtualized resources in cloud data centers offers new opportunities and also a new set of unique challenges. Load balancing in cloud data centers is used to make sure none of your resources remain idle(or underused) while others are being overused. To balance the load distribution, some of the workloads may be migrated from overloaded source nodes to relatively lightly loaded destination nodes. When resource demands are not high, the load balancing technique may choose to power off some servers by migrating its workloads to other nodes in the data center to save energy.

Load balancing algorithms(Nadeem and Mohammed, 2015) are broadly classified into two types static and dynamic. When load distribution decisions are carried out during runtime considering the current state of the system, the process is called dynamic load balancing. If load variations are low in the systems, static load balancing is usually employed. Static load balancing requires prior information about the system resources to make load distribution decisions. The static load balancer does not consider the dynamic state of the system into account for decision making.

The goals of load balancing mechanism in cloud data centers can be summarized as below,

1. Improvement of the overall throughput substantially with optimal resource utilization.

2. Save energy when the load on the data center is not high by server consolidation.

3. Backup plan in case the system fails even partially.

4. Maintain system stability by monitoring server overload conditions.

5. Accommodate run time changes in the data center's load(demand) and resource availability(capacity).

The load balancer technique used in the cloud data centers needs to be very sophisticated and intelligent to consider various parameters to meet the given objective function. Load balancing decisions in the cloud environment are carried out at three different levels, as explained below.

1. **Cloud broker**

   In geo-distributed data centers or in a multi-datacenter setup, the cloud broker is responsible for routing user requests to a particular data center(DC) for processing. Load balancing at cloud broker may need to consider parameters such as proximity of DC, network latency to DC or any other business-related constraint.

2. **VM-PM mapping process**
   When the user requests new virtual machines(VMs) in a data center to meet current business demand, system creates new VMs and places them on a suitable physical host. Multiple VMs are placed on a single physical host to share underlying resources. The load balancer in the DC is responsible for physical machine(PM) selection for initial VM placement and VM migration to another PM during host overload or server consolidation process.

3. **Task-VM mapping process**
   When user requests arrive at the data center, it has to be assigned to a VM for processing. The selection of a particular VM for request(task) assignment is done by the task load balancer in DC. The task load balancer may take into account, the parameters such as VM state(idle, busy) or number of requests assigned, etc. for making assignment decisions.

1.4 Background for research

Cloud computing is growing at an overwhelming rate, with many internet-based applications being migrated to cloud data centers at an ever-increasing pace. The companies like Amazon, Microsoft, Google are expanding their cloud data centers for the services to their vast spread user bases across the globe. The setting up of cloud data centers need lot of investments at the beginning for IT hardware and software along with few non-IT expenses and later incur ongoing costs such as data center administration costs and huge power costs to keep the data center up for the 24x7 operations.

Table 1.2 lists the share of the costs of various components used in building cloud data centers. The cost is amortized to obtain a common cost run-rate metric that can be used for one-time costs(for purchase of servers etc.) and ongoing maintenance expenses(for power costs). Though these cost shares may vary slightly with time and

Table 1.2: Cost For Cloud Data Center Owners

Cost share in % (Amortized)	**Component type**	**sub-components**
45%	Servers	Physical resource such as CPU, Memory and storage
25%	Infrastructure	Power distribution lines and cooling
15%	Power Consumption	Electrical utility costs
15%	Network	Links, transits and other equipment

geographical positions, these are the overall major costs involved for data center owners. It can be noted from the table 1.2 that power consumption costs also contribute to a significant share in the overall data center management or ongoing costs and any saving in power costs can help reduce significant cost for data center owners in the long run.

It is noted that 59% of the total power consumption in the data center is attributed to servers and even small amount of decrease in power consumption of servers will certainly have the largest impact in total power costs of data centers. Additionally, it may save cooling costs.

According to the United States data center energy usage report(Berkeley, 2016), in the year 2014 alone data centers in the U.S. consumed an estimated 70 billion kWh which was equal to about 1.8% of total U.S. electricity consumption. The electricity usage by U.S. data centers is expected to reach 73 billion KWh in 2020.

Electricity prices vary from one geographical location to another. The electricity price depends on several factors and governed by the domestics rules of each geographic location. The factors that may have an impact on electricity price may be the technology employed, raw materials used and output volume involved in the generation of the electricity. It can be noted that various cloud providers are building data centers at geographically dispersed locations across globe to ensure availability and performance for their user applications.

It is vital for cloud providers to reduce data center management costs to offer competitive pricing for users. Power costs are one of the significant portions of the overall data center management costs, and it is going to be beneficial for both cloud providers and users to optimize the cost of power consumption without violating service level agreements of its customers.

1.5 Motivation

The following are the most important facts and observations that compelled us to explore our curiosity in this direction.

1. **Power efficiencies of heterogeneous physical servers**

   The data center is a server farm consisting of a large number of heterogeneous physical machines connected by a high speed shared network. These physical machines often tend to vary in terms of their computing capacity, composition, and also in their power consumption characteristics at different load conditions. Such heterogeneity in the composition of physical machines results in some of these machines being more power-efficient than others during their operation in the data center. It is feasible to optimize power consumption in the data center by detecting and efficiently scheduling workloads to power-efficient physical servers.

2. **Non-uniform electricity costs across geographical locations**

   Electricity(power) price varies with geographical locations across the globe, and many cloud providers are setting up data centers at multiple geographical locations to cater to their users. It is possible to optimize the cost of serving each request by geo-distributed data center network by routing requests to the cost-effective yet quickest data center among many geo-distributed data centers available at that time.

3. **Non-uniform load conditions(peak and non-peak) in data centers**

   The data centers experience varying load conditions at different times of the day. One of the goals of the load balancer in virtualized environments like cloud data centers is to adjust the workloads to available physical resources(VM placement optimization) as per changing load conditions to enhance resource utilization and also to improve performance. It is usually done considering the load conditions at each physical server(overload or underload). It is possible to improve the load balancing algorithms to consider global(intra-DC) load conditions(load context) to make optimal workload placement decisions.

4. **Increase in demand for supporting efficient GPU computing in cloud** Many cloud providers have begun offering GPU-enabled services for their customer applications where GPUs are essential or when high computational power is needed to meet the desired QoS. Though virtualization solutions for CPU are matured well to use in data centers, the same conventional virtualization techniques do not apply for GPUs because of the inherent differences in architecture and operations. There is a need to study various existing issues with GPU enabled VM provisioning, replacement, and power optimizations from the perspectives of resource management and also investigate difficulties posed by application developers to design their algorithm for efficiently utilizing virtualized GPUs(vGPU) in the cloud.

1.6 Research contributions

The following contributions of this research work are available to the research community in the form of journal and conference publications.

- **Information of contextual parameters for power and cost-saving in cloud environment:** It provides information about various parameters that can constitute the context of cloud DCs including physical machine power and performance characteristics in heterogeneous DCs, varying electricity costs in multi-DC set-up across the globe, and dynamic load conditions in DCs.

- **Framework for detection of context in DCs:** The context of the DC is classified as the local and global context. The detection techniques of the local(at each host) and global(overall load) contexts in DC are proposed.

- **Physical machine characteristics and load conditions aware VM placement optimization:** A new VM placement optimization technique considering power and performance characteristics of each physical host and overall load condition in the data centers is proposed.

- **Electricity cost-aware request routing technique in geo-distributed DCs:** A cost optimizing request routing cloud broker technique considering varying electricity costs across the globe in the geo-distributed multi data center setup is proposed.

- **Peak hour performance improvement of task load balancer(ESCE):** Modification to the existing ESCE algorithm is proposed to improve the peak hour processing efficiency. The proposed method overcomes the over-allocation problem in the algorithm to manage uniform allocations in the current state of the system.

- **Identifying research challenges for efficient GPU computing in the cloud:** Existing research challenges concerning resource management and programming for GPUs in virtualized environments is discussed.

1.7 Outline of the thesis

This section briefly describes each chapter of this thesis, to give a brief overview of the structure.

- **Chapter 1**, the current chapter introduces the general domain of cloud computing and motivates the need for new load balancing techniques that are capable of considering contextual parameters for cost and energy saving for data center owners.

- **Chapter 2** describes the literature survey related to the problems and past solutions for the objectives addressed in this thesis.

- **Chapter 3** explains the proposed context-aware VM placement optimization technique for power saving in cloud data centers.

- **Chapter 4** describes our proposed cost-aware request routing technique in a geo-distributed data center scenario.

- **Chapter 5** explains our proposal for peak hour performance improvement for Equally Spread Current Execution (ESCE) load balancing algorithm, a task to VM load balancer used in data centers.

- **Chapter 6** mentions our study of current infrastructure for supporting GPU computing in cloud data centers and existing challenges concerning resource provisioning and programming for virtual GPUs in a cloud setup.

- **Chapter 7** concludes the thesis and summarizes the contributions in more detail. Furthermore, the possible directions in which the proposed methods and techniques that can be improved further are also briefly discussed.

1.8 Summary

The chapter covered the introduction to the concepts of cloud computing, virtualization, and load balancing briefly. Then, the chapter presented the background for the proposed research, motivation and contributions of the reported work. The chapter concluded with the outline of the thesis.

In the next chapter, the important literature that is relevant to the research problem addressed in this thesis is discussed along with the research gaps identified, problem definition, and research objectives.

Chapter 2

Literature Survey

In the previous chapter, we have introduced concepts of cloud computing, virtualization, and load balancing in cloud environments. This chapter presents the set of scientific literature we have referred to in this thesis. There are several contributions done by researchers all over the world that have helped us in identifying the research gaps and addressing them by proposing suitable solutions. The problem of reducing the data center management costs is addressed in this thesis.

2.1 Overall data center management costs

The setting up of data center incurs huge capital investment at the beginning and later cloud providers have to pay on-going operational expenses for power bills and other maintenance tasks at regular intervals. It is noted by a study(Hamilton, 2019) that power consumption cost is the second most contributor to the overall data center management costs after servers. It is also estimated that power costs are going to dominate in maintenance costs of large scale modern data centers in the future. A study(Greenberg et al., 2009) conducted for the cost estimation of cloud service data centers observed that power costs contribute a share of about 15% in the overall data center management costs and 59% of power consumption costs are attributed to the power consumed by the data center servers. It is noted that any decrease in the power consumption of servers will have the largest impact on the overall data center power costs. Also any decrease in the power consumption by the data center servers will lead to reduced cooling costs.

Clearly, the data center management costs can be significantly reduced if power costs are optimized in the data centers. Any reduction in data center management costs will increase the return on investment(ROI) for data center owners and it will benefit both cloud providers and also, in turn, can reduce costs for the cloud services for users.

2.2 Power and cost optimization

The power and operation cost optimization in cloud data centers is an evolving research area considering the problem of stochastic nature. The overall power cost optimization can be addressed by any one or all of the following techniques.

1. Optimize the overall power consumption of servers.

2. Utilize resources from data centers where power is relatively cheap.

3. Improve resource utilization and throughput to avoid resource wastage in the data center.

Many important techniques have been proposed to optimize the cost of data center management by improving resource utilization and exploiting power-saving opportunities. This section discusses some of the noted past works that have motivated our research.

2.2.1 VM placement optimization

Although the notion of VMs and virtualization has been a game-changer for the IT industry, the VM placement brings many challenges that need to be addressed in cloud computing(Abdelsamea et al., 2014). VM placement needs to be optimal to meet performance goals, optimize network usage, reduce resource costs, and also save energy. The VM placement optimization strategy can be QoS-aware, power-aware, cost-aware, network-aware, GPU-aware or a combination of these.

The VM placement schemes can be broadly classified into two types(Masdari et al., 2016) as shown in figure 2.1.

1. **Static VM placement:** The mapping between VMs and PMs are fixed throughout the lifetime of the VMs based on the application-specific requirement. The VM-PM mappings may not undergo changes for long time. Static VM placement will not involve VM migrations. The static VM placement is generally not power efficient as they do not adapt to changing conditions in the data center.

2. **Dynamic VM placement:** The initial mapping of VM and PM is changed based on the state changes in the load of the system.

The dynamic VM placement schemes can be further classified into two types based on when the VM placement is initiated.

1. **Proactive VM placement:** The initial mapping of VM to PM is changed before the system reaches a certain condition.

2. **Reactive VM placement:** The initial mapping of VM to PM is changed after the system reaches a certain condition. The change in mapping may be induced by several factors such as performance, maintenance, power, or load situations.

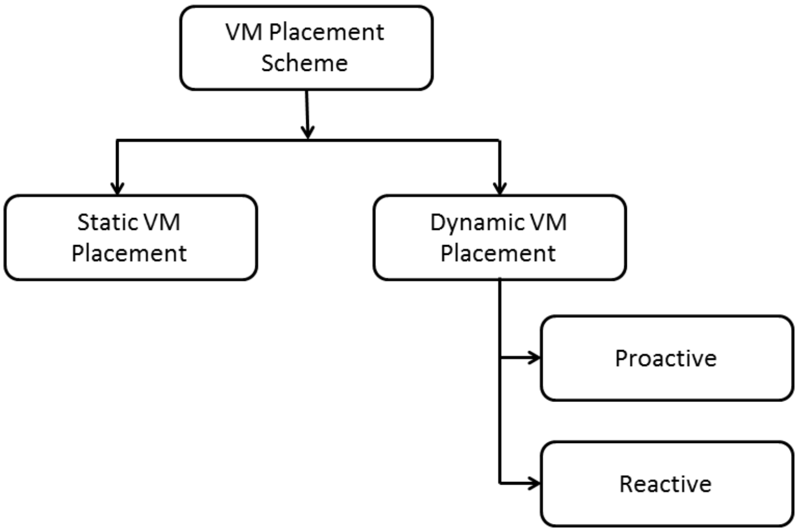


Figure 2.1: Classification Of VM Placement Schemes

We are not interested in static VM placements in our reported work as they do not help to save power in ever-changing load conditions in the data center. The reported work focuses on the objective of power cost minimization by power saving through dynamic VM placements and VM placement optimizations.

A Power and cost-saving in data center

Some of the noted past work that deals with power consumption minimization and energy saving are discussed in this section.

An adaptive heuristics-based performance efficient and energy-saving technique (Beloglazov and Buyya, 2012) for dynamic consolidation of VMs in cloud data centers is proposed. The authors presented a competitive analysis and proved competitive ratios of optimal online deterministic algorithms. The authors addressed the problems of VM migration and dynamic VM consolidation. Paper proposed a novel solution for dynamic consolidation of VMs based on the analysis of historical data from the resource usage by VMs and power consumption statistics of the host machines to arrive at the VM placement decisions.

A novel technique(Chiang et al., 2014) to utilize server idle power in the data center to minimize operational costs is proposed. The authors first studied the problem of controlling service rates and optimizing the operational cost of data centers. The authors then formulated a three-parameter cost function that takes into account the costs of power consumption, system congestion, and server startup. A green control algorithm was proposed to solve the constrained optimization problem of cost-saving and to make costs versus performances tradeoffs in physical machines with different power-saving policies without violating the performance SLAs promised to users.

A performance interference aware virtual machine placement strategy(Moreno et al., 2013) to avoid performance bottlenecks caused by non-compatible VMs co-hosted on the same servers is proposed. The paper proposes a novel technique for workload allocation for energy efficiency by considering the VM workload characteristics and host internal interference levels to select the suitable physical host for the given workload.

A technique(Guo and Fang, 2013) to utilize energy storage available in data centers to reduce the overall electricity costs in the wholesale electricity markets is proposed. The authors considered the scenario where the price of electricity varies both spatially and temporally. The technique proposed integrates center-level load balancing with the server-level configuration, and battery management and also at the same time ensures the quality-of-service(QoS) for users. The paper utilizes Lyapunov optimization to achieve a tradeoff between energy storage and cost-saving.

Energy and SLA-aware VM placement strategy(Mosa and Paton, 2016), which dynamically assigns virtual machines to physical servers in a cloud environment is proposed. The authors formulated the VM placement problem using utility functions and proposed a genetic algorithm to search VM-PM assignments that maximize the utility function formulated for the VM placement problem. The technique proposed co-optimizes SLA violations and power consumption.

A evolutionary game theory based VM placement and optimization technique(Xiao et al., 2014) for dynamic VM placement and server consolidations in the data center is proposed. The proposed work addressed the challenges with VM placement for energy saving by building a computational model for energy consumption in data center.

An energy-aware scheme for VM placement optimization is proposed for power consumption reduction and improving load balance in the data centers. A technique based on genetic algorithm and tabu search algorithm called GATA(Zhao et al., 2019) is proposed. The goal of the proposed technique is to achieve optimal VM placements and energy saving in the data centers.

A variant of Particle swarm optimization (PSO)(Dashti and Rahmani, 2015) to address the problem of incompatibility between user requests and physical machine specification causing the performance degradation and power wastage in data centers is proposed. A modified PSO algorithm is proposed to migrate the VMs from the overloaded hosts and also a dynamic server consolidation technique to save power is presented. They demonstrated that the proposed solution can reduce power consumption and improve performance.

The virtual machine placement problem with the goal of minimizing the power consumption in the data center is addressed using the heuristics-based approach(Li et al., 2013). Authors studied the wastage of resources in the physical machines due to imbalance created in utilization of multi-dimensional resources of the host machines. Authors proposed a multi-dimensional space partition model called EAGLE to overcome the imbalance in resource utilization and reduce power consumption in the data center.

A profit-maximizing technique(Toosi et al., 2014) for cloud service providers by optimizing the allocation of data center capacity to each pricing plan utilizing

Table 2.1: Summary Of Related Past Works In VM Placement And Optimization

Serial No	**Primary Mechanism**	**Authors,Year**	**Goal of proposed work**	**Limitations**
1	Adaptive Heuristics	Anton and Rajkumar,2012	Minimizing total energy consumption of datacenter.	Does not consider Performance characteristics of physical hosts.
2	Green control	Yi-Ju Chiang et. al,2015	Optimizes operational cost of datacenter and ensures SLA guarantee.	Technique considers only idle power in DC to save cost.
3	Dynamic programming	Adel Nadjaran Toosi et.al,2015	Maximizing profit for data center owners.	Does not consider energy saving.
4	PSO based	Seyed Ebrahim Dashti and Amir Masoud Rahmani,2015	Minimizing energy consumption and ensures QoS for users.	Technique does not consider power efficiency of PMs.
5	Heuristics based	Li, X.et. al,2013	Minimizing total energy consumption.	Does not consider power efficiency of PMs and does not guarantee QoS.
6	Best fit decreasing	Noumankhan Sayeedkhan, P. and S. Balaji,2014	Minimizing performance degradation due to interference.	Does not consider energy saving.
7	Graph theory based	Xiao, Z., et al., 2015	Minimizing energy consumption.	Technique is not power and Qos aware.
8	ACO based	Dong, J.-k., et al.,2014	Reduce communication traffic in DC network.	Technique is not power-aware.
9	Greedy algorithm based	Kanagavelu, R., et al.,2014	Reduces inter-VM traffic and network load.	Technique does not address energy saving.
10	Integer programming	Li, W., J. Tordsson, and E. Elmroth,2012	Ensures QoS for users.	Technique does not address power saving.
11	Automata-based	Liu, C., et al,2014	Maximize resource utilization and minimize communication traffic.	Technique does not consider power efficiency of PMs.
12	Lyapunov Optimization	Yuanxiong Guo and Yuguang Fang,2013	Minimizing power costs in the variable pricing market.	Technique does not address power consumption reduction.
13	Genetic algorithm based	Abdelkhalik et.al,2015	Minimizing overall cost and SLA violations.	Technique does not consider power efficiency of PMs.
14	Interference aware algorithm	Ismail Solis Moreno et.al,2013	Minimizing energy consumption and performance aberrations.	Technique does not consider power efficiency of PMs and QoS.
15	Affinity aware VM placement	Sujesha and Kulkarni, 2011	Minimizing network resource utilization.	The technique only considers network latency.
16	power-aware VM placement	Zhao et.al, 2019	Minimizing power usage by host shutdown.	Technique is not power and QoS aware.

admission control for resource reservations is proposed. The authors proposed an optimization technique based on the formulation of stochastic dynamic programming and two heuristics that consider trade-offs between computational complexity and optimality. The proposed technique is evaluated using real workload traces of Google to prove the effectiveness of the solution.

The problem of performance degradation due to resource contention with disk i/o when two or more disk intensive VMs are co-hosted on a physical server is discussed. Authors(Sayeedkhan et al., 2014) proposed a best fit decreasing(BFD) allocation technique based on the static disk threshold-based migration scheme for disk-intensive task scheduling in a cloud computing environment to overcome the problem.

Some of the past works also attempted to solve VM placement optimization for network traffic minimization in the data center using techniques such as Ant colony optimization(Dong et al., 2014), network affinity aware scheme(Sudevalayam and Kulkarni, 2011) and greedy based schemes(Kanagavelu et al., 2014). The VM placement optimization problem is also addressed for ensuring QoS for users at all times by using Integer programming(Li et al., 2011) technique and to also meet hybrid objectives such as maximizing resource utilization and reduce communication traffic using automata-based schemes(Liu et al., 2014).

The problem of VM placement optimization has been addressed in the past using different approaches/algorithms to achieve different desired objectives as discussed above. Table 2.1 summarizes these important related works with their primary mechanism and goals achieved by each one of them.

2.2.2 Load balancing in geo-distributed data centers

Many cloud providers are setting up geographically dispersed data centers to cater to increased computing demands from user applications and also reduce response times. When multiple DCs are serving user requests, it is vital to determine which DC and which PM to assign to fulfill the request for computation. It is also important to meet additional constraints like minimum cost, optimal power, etc. We have investigated the issue of load distribution among available geographically distributed data centers considering the operational expenses involved. Some of the noted literature that is relevant to our study are discussed in this section.

A study(Ashikur et al., 2014) of power management problem of data center operations and various aspects that influence the power costs is reported. The authors discussed the current state of art technologies and proposed methods to improve the power management in the data centers. The paper also proposes to utilize smart grid environment to ensure efficient and dynamic power management solution for the data centers.

A priority-based round-robin(Mishra et al., 2014) is proposed to schedule the requests from the user bases to the data center when there are multiple data centers are available in the same region. The data centers are assigned a priority and requests are assigned based on round-robin strategy to improve the performance compared to proximity-based routing service broker algorithm(Wickremasinghe et al., 2010).

A DVFS based operational cost optimization solution(Gu et al., 2015) is proposed for the geo-distributed data center scenario. The proposed technique exploits the dynamic frequency scaling technique for power consumption management and an optimization problem is formulated and solved that reduces the operational expenses of the data center without affecting the quality-of-service for the user tasks.

A game theory based algorithm(Tripathi et al., 2017) for load balancing is proposed to optimize the operating cost in the geo-distributed data centers. Authors modeled the load balancing problem as a non-cooperative game and operating expenditures are modeled as a linear combination of power and latency costs. The proposed technique models the load balancing as a cost optimization game and obtains a nash equilibrium structure. Based on the obtained structure a novel algorithm is proposed to minimize operating expenses.

The cloud service broker is responsible for routing requests from users to one of the cloud data centers in the geographically dispersed data centers. A proximity-based request routing technique(Wickremasinghe et al., 2010) is proposed that routes users to the nearest available data center in terms of transmission delay. The authors also proposed a best response time service routing policy that estimates the response times for all the available data centers for the current request and DC with smallest estimated response time is allocated for the user request.

A framework(Nadjaran Toosi et al., 2017) for reactive load balancing to distribute requests for web application among multiple available data centers is proposed. The load balancing algorithm routes the user requests based on the renewable energy source available in the location of the data centers. The authors suggest that the proposed technique can reduce power costs by reduced utilization of brown energy.

A response time-sensitive load balancing solution is proposed for distributed, heterogeneous data centers scenario. The offline solution is proposed based on force-directed scheduling technique(Goudarzi and Pedram, 2013) that can determine the application placement on a particular DC over a long period of time. The offline algorithm is further extended to support online application placement in a distributed DC with migrations. A prediction about application lifetimes, workload volumes, renewable energy sources are considered for decision making.

The authors proposed a fuzzy-based algorithm(Toosi and Buyya, 2015) to exploit the temporal variations of power costs, renewable energy available to reduce power costs and increase utilization of renewable energy. The proposed algorithm is tested with real workload traces of National Renewable Energy Laboratory and Energy Information Administration and found to improve the reduction in cost to a significant extent.

2.2.3 VM level load balancing policies in CloudAnalyst

The scheduling of user requests in the cloud data centers is an NP-hard optimization problem. Load balancing of tasks on VMs is an important aspect in cloud computing to meet several objectives like uniform utilization, power and cost-saving. Effective load balancing strategies can avoid conditions like overload, underload of VM resources causing system failures or wastage of power. There is lot of literature is available for load balancing on VM in cloud computing domain, we will discuss some of these algorithms which are relevant to our work.

CloudAnalyst(Wickremasinghe et al., 2010) is an open-source, graphical user interface(GUI) based simulator for the cloud environment. The CloudAnalyst offers simulation and modeling of all important entities in cloud and offers flexibility to add and evaluate a new resource provisioning policy in cloud before being deployed on to real cloud. The CloudAnalyst provides 3 different VM level load balancing strategies

Table 2.2: Important Past Work Related To Geo-distributed Data Center Load Balancing

Authors,year	**Primary Mechanism**	**Problem Addressed**	**Limitations**
(Wickremasinghe et al., 2010)	Proximity based	Distribution of load based on DC location	Dynamic electricity pricing is not used for request routing.
(Nadjaran Toosi et al., 2017)	Renewable energy utilization	Reduce power cost data centers through renewable energy	Response times for users is not considered.
(Le et al., 2017)	Advance energy procurement in multi-timescale electricity market	To reduce power procurement costs	Technique does not consider response time for users.
(Wickremasinghe et al., 2010)	Response time based	To improve response time for users	Dynamic electricity pricing is not considered.
(Goudarzi and Pedram, 2013)	Force-directed scheduling	To improve response time for online service applications	Dynamic electricity pricing is not considered.
(Gu et al., 2015)	DVFS based	Operational cost minimization but ensure QoS	Technique does not consider electricity cost for processing.
(Toosi and Buyya, 2015)	Fuzzy logic-based	Reduce power cost and carbon footprint	Response times for users is not considered.
(Tripathi et al., 2017)	Game theory based	To minimize the operating cost and obtain the structure of Nash equilibrium	Work does not consider dynamic electricity cost and QoS.
(Mishra et al., 2014)	Priority-based round-robin	To address request routing in multi-DC situation in same region	Technique does not consider electricity cost for routing.
(Ashikur et al., 2014)	Global load balancing technique	Power and cost management in the smart grid environment	Technique does not consider response time for users.

for users. A round-robin policy allocates user requests to available VMs in a circular fashion. The algorithm starts request allocation with a random VM in the data center at the beginning. The round-robin load balancer has a simple implementation with less computational overhead. However, the round-robin policy does not consider the current load on the VM for allocation.

Throttled load balancing policy considers the state of the VMs to assign new requests from users. A VM is associated with two states idle and busy, when a new request arrives at the data center, an idle VM is searched for allocation, if VM with the idle state is found, the request is assigned. If none of the VMs are idle, the request is moved to the waiting queue. Though throttled load balancing policy considers the state of the VM, the requests may need to wait for long time in the single waiting queue.

Equally spread current execution load balancing policy(ESCE) offers a minimum waiting time for the requests by allocating a VM with the least number of assigned requests/tasks. The ESCE ensures uniform request allocation to the VMs in the data center. The ESCE load balancer maintains an allocation table to keep track of requests and state of the allocation table is updated with notifications from data center controller about request allocations and de-allocations to VMs. However the ESCE load balancer does not ensure uniform request allocations to VM when request frequency is very high(peak load situation). Our proposed work offers a solution to the problem of non-uniform request allocation during peak load conditions for ESCE load balancer in this thesis.

A detailed analysis of contemporary VM load balancing algorithms in Cloud-Analyst is presented. Further a Weighted Signature-based load balancing (WSLB) algorithm(Ajit and Vidya, 2013) is proposed to reduce response time for the requests. WSLB calculates the load assignment factor for each host and assigns the VMs based on the factor value.

A comprehensive survey of important VM level load balancing algorithms is discussed in (Mishra et al., 2018). Authors present a taxonomy of load balancing schemes and cover most of the important work done in the domain of VM level task scheduling in the cloud. An evaluation of the heuristic-based algorithms for some of the vital performance metrics is carried out using Cloudsim(Calheiros et al., 2011) and a systematic comparative study of evaluation results is presented.

2.3 GPU enabled computing resource management

The GPU computing in the cloud is an emerging trend as more and more compute-intensive, HPC, graphics applications are hosted on the cloud datacenters. Though enough research has been done on CPU virtualization and their efficient resource management techniques, the GPU virtualization and issues with the management of GPU resources in the cloud is still a growing research area. In this section, we would mention some noted past works in GPU provisioning that are in line with our research direction.

A disengaged scheduling technique(Menychtas et al., 2014) for the provisioning of GPU to vGPUs is proposed. The authors utilize disengaged timeslice with an overuse control mechanism that ensures fairness in the allocation and disengaged fair queuing is used to limit resource idle states, but the method used is probabilistic. Schedulers ensure a fair share of GPU among all application even when applications are non-cooperative and adverse to each other.

A GVim(Gupta et al., 2009) scheme is proposed that utilizes both round-robin(RR) and Xeno credit-based scheduling(XC) techniques of the Xen hypervisor for task scheduling on GPU. RR scheduling sequentially selects a vGPU for every fixed timeslice and monitors the call buffer of the vGPUs during this period. XC uses a credit concept, which is time allocated for each vGPU. XC processes call buffer of vGPU for a variable time, which is proportional to the credit amount to ensure weighted fair sharing between guest vGPUs.

A Rain(Sengupta et al., 2013) framework is proposed for load balancing GPU requests across GPUs fitted on distributed machines. The work suggests a two-level hierarchical scheduling policy. The top-level module of the framework distributes the load across all GPU equipped server machines. The bottom level module is responsible for GPU device level scheduling of vGPUs.

A GPUvm(Suzuki et al., 2014) scheme that uses a BAND scheduler and solves the issue with a credit-based scheduling scheme is proposed. The proposed technique solves the miscalculation of credit when GPU idle time is included in credit amount,

Table 2.3: Important Past Work Related To GPU Provisioning Policies

Authors, year	**GPU provisioning policy**	**Limitation**
Menychtas et al. 2014	Fair Queueing and Round robin	Framework does not consider GPU memory transfers.
Gupta et al. 2009	Round Robin and Credit-based	Technique includes GPU idle time for credit calculation.
Sengupta et al. 2013	Priority-based and Credit-based	Framework does not support heterogeneous GPUs.
Suzuki et al. 2014	Credit-based	Technique induce unnecessary context switches due to credit value.
Farooqui et al. 2016	Affinity-based	Technique cannot be applied to applications with device-specific codes.
Gupta et al. 2011	FCFS	Technique does not address virtual environments.
Zhang et al. 2014	SLA based	Framework is specific to a mixture of time-constrained applications.
Siavashi and Momtazpour, 2018	Fair-share based	Technique does not consider memory transfer during a context switch.

which may lead to inappropriate GPU share for certain vGPUs. GPUvm solves this issue by first transforming the CPU time of GPU scheduler into credit value and then subtracts the total credit value from the current vGPU.

Investigation(Farooqui et al., 2016) of current work-stealing algorithms is conducted and observations are reported. Existing algorithms are found to be unaware of the CPU and GPU characteristics, and such a situation results in degradation of performance in OpenCL like applications that are capable of running on both CPU and GPU platforms. To overcome this issue, the authors proposed a framework named Libra, which first derives the device affinity scores for applications. Application is assigned to the device with the highest affinity score.

A new Pegasus(Gupta et al., 2011) framework that addressed one of the challenges in GPU scheduling is proposed. GPU virtualization technique has no access to impose scheduling policy because the multiplexing of GPU is integrated into the device drivers. Pegasus proposed a concept called VCPU with which GPUs are made basic scheduling entities. The Pegasus includes proportional fair share, FCFS, credit-based scheme, and SLA feedback based schedulers. The objective of Pegasus is to meet the different requirements set forth by applications using different GPU schedulers.

A framework VGASA(Zhang et al., 2014) including adaptive scheduling policies is proposed. These adaptive algorithms include a dynamic feedback control loop. VGASA consists of three scheduling policies, SLA-aware algorithm receives FPS(frame per second) information and adjusts the sleep time per frame time. Fair SLA-aware algorithm take away GPU from fast running applications and allocates to slow running ones, and enhanced SLA-aware algorithm allows all VMs to possess the same frame rate under 100% GPU utilization.

A fair-share GPU provisioning policy is proposed by GPUCloudSim(Siavashi and Momtazpour, 2018) to share physical GPU among multiple vGPUs. The technique allows all competing vGPUs to receive a slice of time on GPU. If the overall processing power of co-located vGPUs exceed that of physical GPU, then the processing power of vGPUs is scaled.

2.4 Research gaps identified

After the study of past work in the domain of resource management in cloud computing, we have found following research gaps, and an honest attempt is made to address these research gaps in our reported work in this thesis.

1. **Consideration of performance to power ratio of physical machines for power saving in DC**

   Though Power consumption profile is considered for physical machines in past work, performance to power ratio is the most appropriate indicator of power efficiency of physical machines. The performance to power ratio calculated from the SPECPower benchmark(SPEC, 2011), an industry-standard benchmark is considered to denote the power efficiency of PMs in DC. The optimal utilization of power-efficient machines is proposed in our reported work to save power in data centers.

2. **DC load conditions in the data center to improve underutilized hosts management**
   The overall load conditions(context) in data centers can be considered to improve the underutilized host management in DC. The DC load(peak and non-peak) condition can be used to avoid overheads and resource wastage caused due to host power off sequence and VM migrations during peak load conditions in DC.

3. **Response times and electricity price for power cost optimization in geo-distributed data centers**
   Some of the past literature proposed solutions for renewable energy usage, electricity procurement in non-peak price duration, etc to reduce the power costs for the data center owners. The varying electricity price across geographical locations is also suggested for request routing but estimated response time from the data center to the user request is a vital parameter to minimize SLA violations.

4. **Problem with ESCE algorithm during Peak load situation in DC**
   A performance problem regarding uniform VM utilization for ESCE load balancer is observed when the request frequency is high in the data center. The state information related to request allocations to each VM is incorrectly updated and used in peak load conditions causing non-uniformity in user task allocation to available VMs in DC.

5. **Scope for further investigation of efficient resource management and programming challenges for GPU computing in cloud**
   Conventional techniques of virtualization do not hold good for GPUs because of the inherent differences in terms of architectures, driver software, and distributed program/memory models. These differences make GPU provisioning in the virtualized environment more complex and can cause inefficiency in resource utilization. There is scope for further investigation of underlying resource challenges for efficient GPU processing in the cloud.

2.5 Problem statement

Design a context-aware load balancing strategy for the cloud to optimize energy consumption/cost, performance and resource utilization using physical machine, cost and load characteristics.

2.6 Research objectives

Our research work attempts for power consumption and cost optimization based on contextual parameters such as physical machine characteristics, data center load conditions, and electricity pricing at that point in time. Our proposed work also proposes peak hour performance improvement for data centers by an additional modification to existing solutions and investigates efficient GPU enabled computing problems in cloud from resource management perspective.

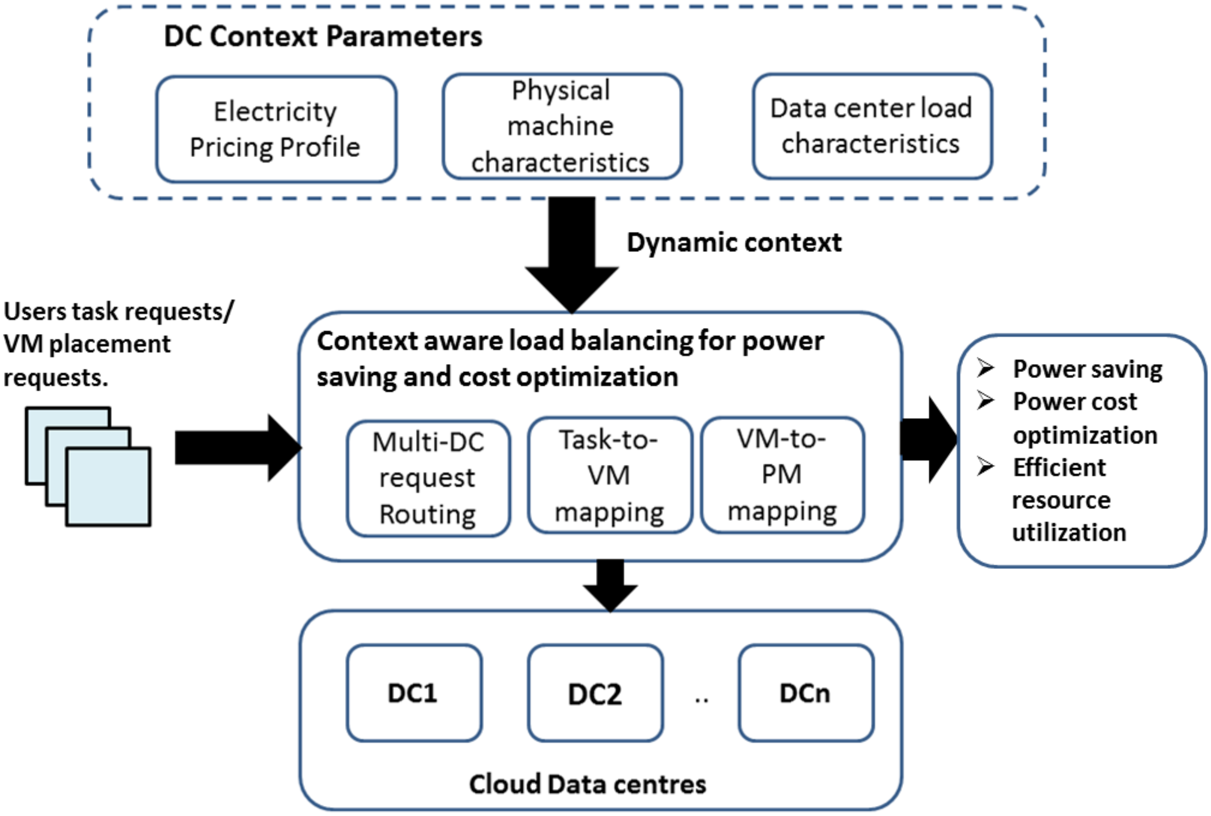

DC Context Parameters
Electricity Pricing Profile
Physical machine characteristics
Data center load characteristics
Dynamic context
Users task requests/ VM placement requests.
Context aware load balancing for power saving and cost optimization
Multi-DC request Routing
Task-to-VM mapping
VM-to-PM mapping
➢ Power saving
➢ Power cost optimization
➢ Efficient resource utilization
DC1
DC2
..
DCn
Cloud Data centres

Figure 2.2: Overview Of The Proposed Work

The overview of the proposed work is presented in figure 2.2.

1. **Power and performance characteristics aware energy saving**
   Analyze the power consumption Vs system throughput ratio for the physical machines in the data center to prioritize PMs for VM placements and also to switch-off machines during non-peak hours.

2. **Electricity cost-aware request routing**
   Analyze electricity cost in various geographical locations and response time for routing of user requests/tasks in the multi-datacenter scenario for power cost savings.

3. **Peak hour performance improvement**
   Detecting peak hours, non-peak hours in data centers, and suitably change the goal of load balancing to match the current situation. Also to propose modifications to existing algorithms to improve their performance during high load situations.

4. **GPU enabled computing in cloud**
   Investigate current gaps in resource management policies and programming with respect to virtualized GPUs.

2.7 Summary

The chapter presented the literature review for the problem of data center management cost minimization. The chapter discussed the details of the overall data center management costs, the impact of power consumption cost on the operating expenses is investigated. Then a literature review involving some of the relevant past work for VM placement optimization, load balancing geologically dispersed data centers setup, task-level load balancing algorithms in CloudAnalyst, and GPU provisioning in cloud are discussed. Finally, the chapter presented the research gaps identified, problem definition and research objectives addressed in this thesis.

In the next chapter, a novel context-aware VM placement optimization technique for heterogeneous cloud data centers is proposed with an objective of power-saving.

Chapter 3

VM Placement Optimization

The rapid expansion of cloud adoption by businesses of all scales has created the necessity of making the cloud more efficient and beneficial for both cloud service providers and their clients. Managing a cloud data center incurs a huge capital at the beginning and also a high maintenance cost for keeping it running at all times. The power cost forms a major share in the maintenance cost and any reduction in power usage will benefit to a great extent to cloud data center owners in the long run.

It is noted that 59% of total power consumption of the data center is attributed to the power consumed by computing servers(Greenberg et al., 2009). Any decrease in power consumption of physical servers in data centers will certainly have the largest impact on the data center maintenance cost. Data centers usually house a large number of servers connected by a high-speed network and provided with massive storage units. The servers(physical hosts) used in DC are heterogeneous in type, purchased from different vendors, and offer a distinct compute capability. These heterogeneous physical servers often exhibit variability in their power consumption and performance characteristics making some servers more power-efficient than others.

Many existing VM placement optimization techniques(Masdari et al., 2016) do not consider the power efficiency of heterogeneous physical hosts and current prevailing load conditions in data centers for VM provisioning and server consolidation process. The power efficiency can be described by variability in power consumption and throughput of two distinct machines at the same load levels. The data center will experience changing load conditions through its 24x7 operation, and it is also vital to optimize the task of VM placement to adapt to the current load conditions.

In this chapter, we propose a VM placement optimization technique for the reduction in total power consumption of the data center by considering the power efficiency of heterogeneous physical servers and dynamically changing load conditions. The rest of the chapter is organized as follows. Section 3.1 introduces the task of VM placement optimization briefly, section 3.2 presents the objective of our proposed technique. The system architecture of proposed VM placement optimization is described in section 3.3. The mechanism to model the power efficiency of physical machines is described in 3.4, and the technique for data center load condition based adaptation is explained in 3.5. Section 3.6 describes proposed algorithms for the VM placement optimization technique, and finally, the experimental setup, configurations, and discussion on results obtained are presented in 3.7.

3.1 Background study

The VM placement optimization is a vital step in data center(DC) operations to re-adjust the VM to PM mappings according to changing resource demands of applications and the physical resource availability in data centers. VM placement optimization is also helpful for server consolidation to save power during non-peak situations in DC. The goal of VM placement optimization is to ensure that, resource demands of user VMs are met with an optimal number of physical resources. The task of dynamic VM placement optimization can be generally split into 4 sub-tasks,

1. Host overload detection: It is the process of detecting physical server overuse where the performance of one or more VMs residing on it starts getting affected. Hence it requires one or more VMs to be migrated out of it.

2. Host underload detection: It is the process of detecting physical server underutilization. Host under-utilization causes power wastage because of idle resources in the system. The situation requires the consolidation of servers by migrating VMs to other appropriate physical hosts(PMs), which enables switching off of some of the servers to save power.

3. VM selection: VM selection is a process of selecting a VM to be migrated from a set of VMs residing on an overloaded host for VM migration.

4. VM re-placement: VM re-placement is a process of searching a new suitable host(PM) for migrating a VM from an overloaded host.

The task of VM placement optimization is invoked at fixed scheduled intervals in the data center. The scheduling interval of 5 minutes is used in the distributed resource scheduler (DRS) of VMware(Mosa and Paton, 2016).

3.2 Research objective

The proposed work in this chapter investigated vital contextual parameters that can constitute the overall context of the data center. The following contextual parameters are considered.

1. Physical machine's performance and power characteristics.

2. Prevailing load conditions in the data center.

With the help of these contextual parameters, we proposed an efficient VM to PM load balancing technique to optimize the overall power consumption in the data center. The objective of our proposed solution is shown as a block diagram in figure 3.1.

The proposed solution considers physical machines performance to power ratio, which signifies the power efficiency of physical hosts and load conditions(peak and non-peak) in data center for VM provisioning and server consolidation. We formulated the power consumption optimization problem as follows.

$$Ptotal(t) = \sum_{i=0}^{N} Po(t)(i)(l) \tag{3.1}$$

Where,
Ptotal(t) denotes the total power consumption of cloud datacenter at time t.
N represents the number of physical machines at the data center.
Po(t)(i)(l) corresponds to power consumption of i$^{\text{th}}$ machine having CPU load of *l*% at time t.

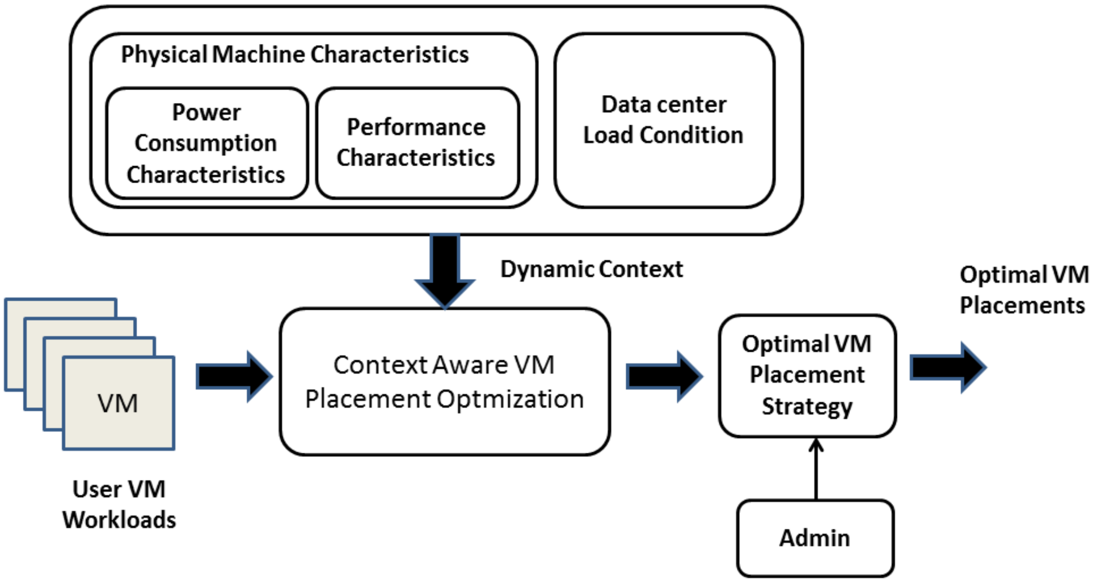

Physical Machine Characteristics
Power Consumption Characteristics
Performance Characteristics
Data center Load Condition
Dynamic Context
VM
User VM Workloads
Context Aware VM Placement Optmization
Optimal VM Placement Strategy
Optimal VM Placements
Admin

Figure 3.1: System Block Diagram For Proposed VM Placement Optimization

The objective of the proposed technique is to optimize the value of Ptotal without affecting the response time of user applications and meet SLAs.

3.3 Proposed system architecture

The target environment of our proposed system in this chapter is a cloud IaaS service model in a large scale data center with N heterogeneous machines. Each node is composed of major system resources such as CPU, main memory, network, and connected to network-attached storage(NAS) for storage. The proposed system has no prior knowledge of user application workloads and VM placement details. The geographically distributed users of such a cloud system can submit their VM placement requests, which may comprise a dynamic mix of distinct application workloads. These dynamic mixes of application workloads wrapped in VMs may be co-located on a single physical server in the cloud data center. The software architecture for the proposed solution consists of two distributed modules. These modules help to capture context information of the data center at both the physical machine level(local) and also at the data center level(global) for efficient VM provisioning and re-placements.

3.3.1 Local context manager

The local context manager(LCM) is designed to work at every physical host and at the same layer of software where hypervisor(VMM) is placed. The block diagram of the LCM is shown in figure 3.2. The LCM is responsible for the collection of information related to physical hosts and all co-located VMs residing on it.

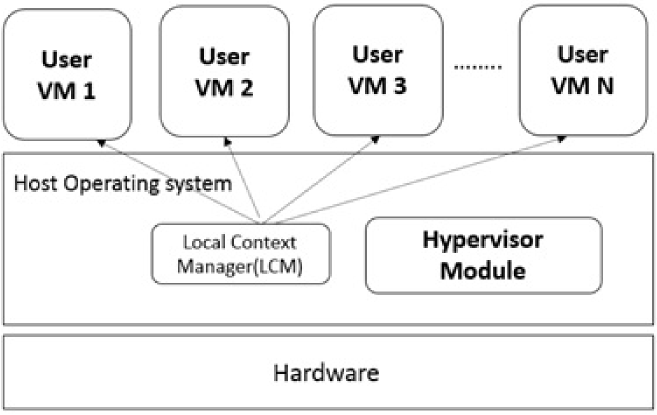


Figure 3.2: Local Context Manager Architecture

The following information is collected by the LCM at each physical host and is regarded as the local context at each physical host in the data center.

1. Resource utilization details of all VMs(CPU, network, and memory).

2. Physical machine remaining resource capacity at run time.

3. Run time power consumption information.

Each host maintains the information about the performance and power characteristics obtained by the SPEC power benchmark(SPEC, 2011). The resource utilization details of VMs are used in determining the overall resource utilization statistics of the physical host and in the determination of overload/underload conditions. The physical resource remaining capacity is needed to check the feasibility of the placement of new VMs on the physical host. The run-time power consumption of the physical host is needed to compute overall power consumption details of the data center at any point in time. The local context manager shares details(local context) with the global workload scheduler(GWS) for optimal VM provisioning decisions.

3.3.2 Global workload scheduler

The global workload scheduler(GWS) is designed to work at a central resource management server or a central load balancing server in each data center. The GWS module works in tandem with LCMs at each physical host to derive the current global context and local context for dynamic VM load balancing in the data center. The block diagram of the GWS module is shown in figure 3.3.

The following are the functions of the GWS module in the data center,

1. Detection of load conditions (peak or non-peak) in the data center(called load-/global context)
2. Invoking VM placement optimization at regular intervals in the data centers to re-adjust the VM-PM mappings to achieve power and performance efficiency.

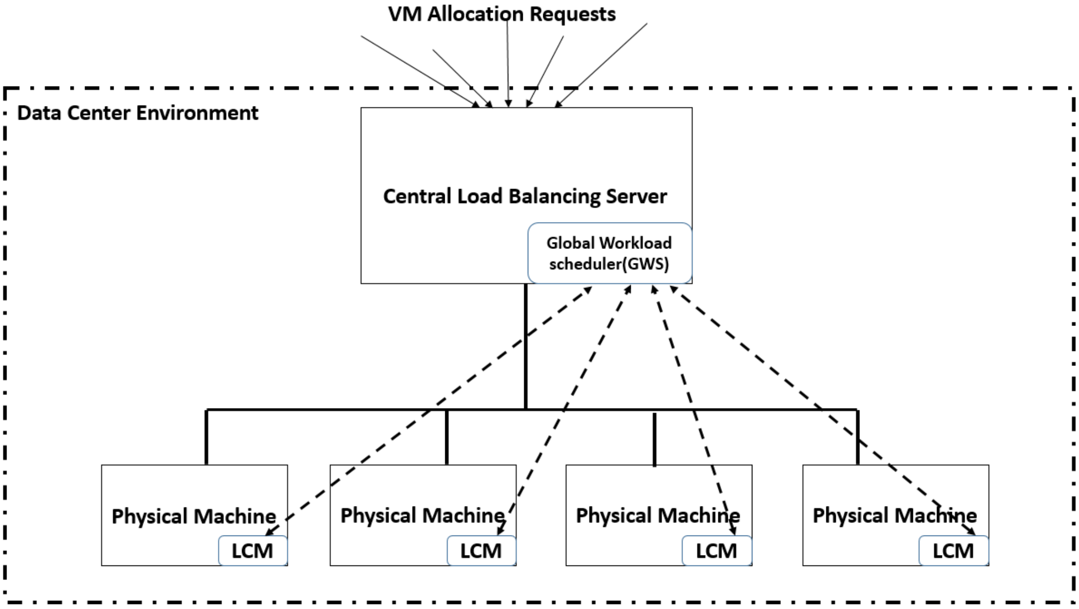


Figure 3.3: Global Workload Scheduler Architecture

3.4 Power efficiency of physical machines

One of the objectives of our work reported in this thesis is to consider the power efficiency of physical machines for VM provisioning and server consolidation decisions of VM load balancer in data centers. The power consumption of a physical machine in nothing but the collective sum of the power consumption of its sub-components such as CPU, memory, disk, power supply unit, and cooling equipment. But some past studies(Fan et al., 2007)(Kusic et al., 2008) have noted that there exists a linear relationship between power consumption and CPU utilization. However, because of evolving modern servers containing multi-core CPUs and support for virtualization, servers are fitted with large RAMs. These large RAMs start to consume a significant share of power in the total power consumption of physical servers. Also, the difficulties in modeling power consumption of multi-core CPUs makes building an accurate analytical model for power consumption analysis a complex research problem(Beloglazov and Buyya, 2012). So instead of relying on an analytical model for power consumption, proposed work reported in this thesis uses real benchmark results for power consumption and performance metrics provided by the SPECpower benchmark(SPEC, 2011).

The data centers consist of physical machines(servers) of varying configurations and from different vendors. These physical machines will not exhibit homogeneity in their power consumption and throughput profiles. We can measure the power efficiency of a physical machine by taking a ratio of throughput(NumOps) to the power consumed(Pc) at different defined load levels. An average of the values noted at different load levels is considered as performance to power ratio of the physical machine.

$$PerfToPowerR(\mathit{Load}\%) = NumOps(\mathit{Load}\%)/P_c(\mathit{Load}\%) \tag{3.2}$$

The ratio of performance to power consumption at different load levels of CPU utilization for a given physical machine is represented by equation (3.2).
Where, *Load %* is calculated as a ratio of current CPU utilization of the physical machine to the total CPU capacity in MIPS and then multiplying the CPU utilization fraction obtained by 100 as indicated in following equation (3.3),

$$\mathit{Load}\% = (cpuUtilizationMIPS(PM)/TotalCpuMips(PM)) * 100 \tag{3.3}$$

Then using data of different load levels of *PerfToPowerR* ratio, an average *PerfToPowerR* can be calculated as in (3.4), where N indicates total number of distinct load levels considered and *PerfToPowerR(Li)* specifies the performance to power ratio of physical host at specific load level of CPU at instance i calculated from (3.2).

$$AverPerf2Pow = 1/N \sum_{i=0}^{N} PerfToPowerR(Li) \tag{3.4}$$

The *AverPerf2Pow* is considered as a metric for power efficiency of the corresponding physical machine; a higher value of *AverPerf2Pow* indicates higher power efficiency of the physical machine. The reported work in this thesis relies on the Spec Benchmark(SPEC, 2011) data published for several types of servers for calculation of the power efficiency of physical machines. The SPEC power benchmark is the first industry-standard benchmark that evaluates the power and performance characteristics of the single server and multi-node servers. It can be used to compare power and performance among different servers and serves as a toolset for bringing about improvements in servers usage and efficiency.

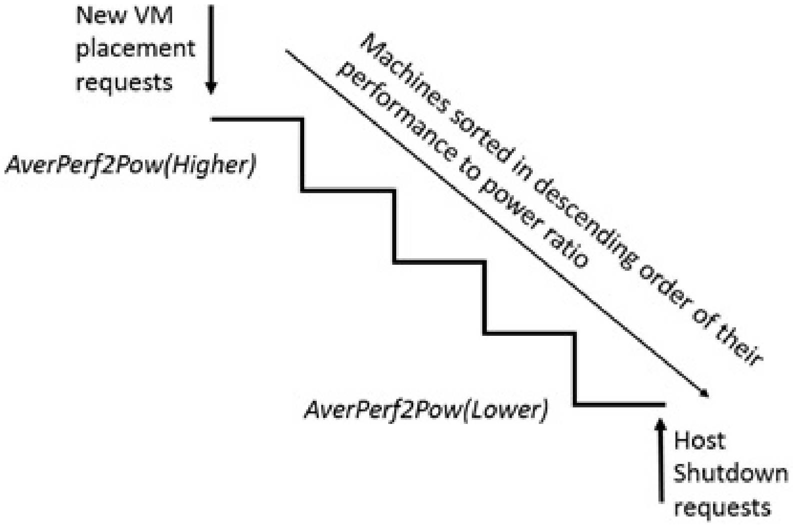

New VM placement requests
AverPerf2Pow(Higher)
Machines sorted in descending order of their performance to power ratio
AverPerf2Pow(Lower)
Host Shutdown requests

Figure 3.4: Proposed Host Selection Technique For VM Placement And Host Shutdown.

Table 3.1 lists a set of server's power and performance metrics reported in Spec Power Benchmark(SPEC, 2011). These server configurations(types) are used for evaluation of our proposed work reported in this thesis.

The P and P2P in table 3.1 represents power consumption and performance to power ratios at different load levels. The Avg P2P indicates average performance to power ratio(*AverPerf2Pow*) of the corresponding physical machine(server) type. The proposed technique prioritizes physical machines with higher *AverPerf2Pow* for physical host provisioning during VM allocation/re-allocation requests. During non-peak hours, physical machines with lesser *AverPerf2Pow* are prioritized for power-off to ensure power-efficient machines are used most to save power. Figure 3.4 describes the prioritizing process of physical hosts based on their power efficiency for new VM placement requests and also when host shutdown requests for power saving are processed.

3.5 Load condition based adaptations

The VM placement optimization process has to check each physical machine(server) for load conditions (overload and underload) at regular intervals in the data center. It is done to re-map VMs to PMs as per prevailing load conditions to ensure performance SLAs for user applications and also to save power. In the process of achieving its goals, the VM placement optimization algorithm also consumes significant computing power and time of the CPU of the servers involved. It is essential to improve the algorithm for VM placement optimization to consider overall load conditions(context) of the data center to eliminate some of the VM placement optimization sub-tasks to optimize power consumption in the data center. The host power off and power on sequences will consume significant power and also CPU time. Also, VM migrations arising out of host power off sequence will place demands for additional resources from both source and destination physical machines.

The work reported in this thesis proposes modifications to the VM placement optimization algorithm to skip acting on host underload conditions for PMs when the data center is experiencing peak traffic(high load) situation. The proposed modifications avoid unnecessary host power-offs and VM migrations during high load situations to help the data center save significant amount of power and CPU time. Figure 3.5 illustrates the modifications proposed to the VM placement optimization technique in

Table 3.1: Power And Performance Metrics From SPECPower Benchmark

Target Load %	**10%**		**20%**		**30%**		**40%**		**50%**		**60%**		**70%**		**80%**		**90%**		**100%**		**Avg**
PM Type	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P**	**P2P**	**P2P**
HP ProLiant ML 110 G4	89.4	63.9	92.6	116	96	170	99.5	222	102	262	106	313	108	350	112	394	114	430	117	467	268
HP ProLiant ML 110 G5	97	102	101	195	105	282	110	354	116	426	121	494	125	554	129	618	133	679	135	731	431
HP ProLiant ML 110 G3	112	47.9	118	89.4	125	128	131	160	137	191	147	218	153	241	157	268	164	285	169	309	190
IBM Server x3250	46.7	665	52.3	1205	57.9	1621	65.4	1930	73	2143	80.7	2361	89.5	2466	99.6	2539	105	2685	113	2767	2098
IBM Server x3550 XeonX5675	98	917	109	1651	118	2274	128	2793	140	3201	153	3497	170	3680	189	3805	205	3929	222	4009	3093
IBM Server x3550 XeonX5670	107	861	120	1528	131	2094	143	2568	156	2933	173	3200	191	3363	211	3491	229	3603	247	3694	2843

the data center. The data center load context is calculated based on the average CPU utilization of all active physical hosts. The load context parameter is configured to take two states called peak and non-peak states based on a static threshold technique. The powering off sequence and migrating VMs residing on the underloaded host are skipped in the data center when load context is set to peak situation to save power.

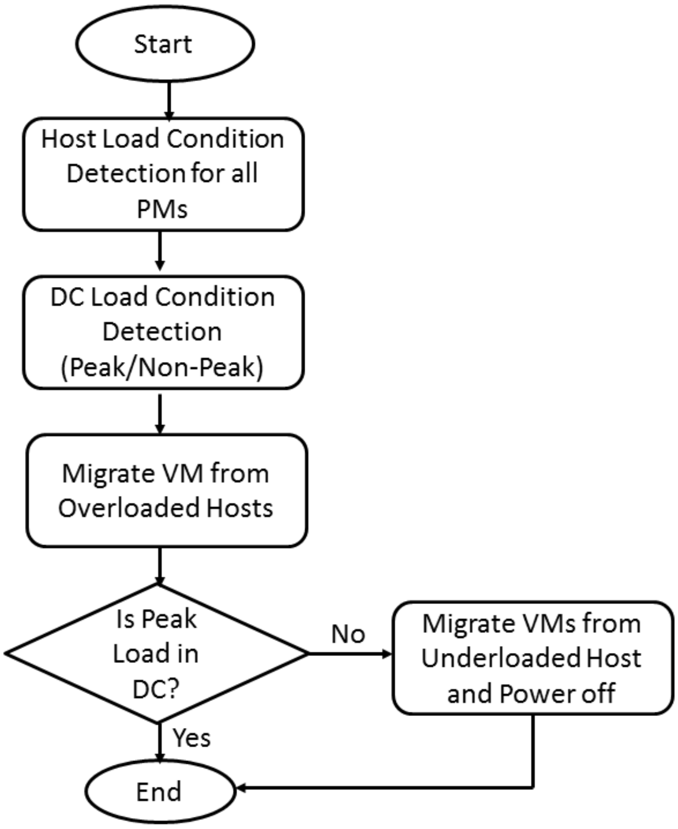

Start
Host Load Condition Detection for all PMs
DC Load Condition Detection (Peak/Non-Peak)
Migrate VM from Overloaded Hosts
Is Peak Load in DC?
No
Migrate VMs from Underloaded Host and Power off
Yes
End

Figure 3.5: Load Context Aware VM Placement Optimization Process.

3.6 Proposed context-aware VM placement optimization

In this section, a context-aware VM placement optimization technique is described. The objective of the proposed solution is to reduce the overall power consumption of the data center without any performance penalty for user applications. In our reported work, the global workload scheduler(GWS) is responsible for initiating the VM placement optimization process. The proposed technique regularly checks for host utilization(load) conditions by communicating with local context managers(LCM) of each physical machine. The proposed work defines the VM optimization scheduling interval as 5 minutes, which is a similar interval used in distributed resource scheduler (DRS) of VMware(Mosa and Paton, 2016).

The proposed work considers the power efficiency of physical machines for VM placement and server consolidation decisions. Also, the technique for detecting the load context of the data center and based on the load context, an alternative method to handle host underload conditions, is defined.

3.6.1 VM placement optimization process

The algorithm 3.1 presented in this section describes the steps of the proposed context-aware VM placement optimization process. In our reported work, the VM placement optimization algorithm is invoked by the global workload scheduler(GWS) module at fixed regular intervals in the data center(invocation interval is set as 5 minutes in the proposed work).

The proposed algorithm for VM placement optimization at first handles the host overload condition for all the active physical machines in the data center. The host load detection process checks each physical host for overload condition and selects one or more VMs from each of the overloaded hosts that need to be migrated out of it to reduce its load. A new suitable destination physical host is searched for one or more VMs that need to be migrated out of an overutilized host. The new (VM,PM) pair for VM migration is added into the *migrationList*. The context-aware algorithm queries the current load context in the data center. If the data center is experiencing peak load situation, the steps to handle the host underload condition for each physical host is skipped to avoid unnecessary power-offs and VM migrations as these physical hosts may need to be powered on again to meet surging resource demands. If the data center load condition is non-peak, the underutilized hosts are switched off after migrating all VMs residing on it to save power. The time complexity of the algorithm 3.1 is O(n^2).

Algorithm 3.1: Context-aware VM placement optimization

```
Input  : pmList
Output: migrationList
vmsForMigration = 0;//initialize list of migrating VMs
foreach pm ∈ pmList do
    /* Identify VMs to be migrated from Overloaded Hosts        */
    if isHostOverutilized(pm) then
        Include VMs from overutilized host into the list of VMs considered for
          migration
        Find suitable destination host for VM migration
        Add (VM, destination PM) pair into migrationList
    end
    /* Query current DC load context                             */
    if isNonPeakSituationInDc() then
        /* select VMs from underloaded Hosts for migration       */
        foreach pm ∈ pmList do
            if isHostInUnderloadedCondition(pm) then
                Include all VMs residing on underloaded host into list of VMs
                  considered for migration
                Find suitable destination host for migrating all VMs
                Add (VM, destination PM) pairs for all VMs into migrationList
            end
        end
    end
end
return migrationList;
```

3.6.2 VM placement algorithm(PPABFD)

The algorithm for VM placement called power and performance-aware best fit decreasing VM placement technique, a modified version of the power-aware best fit decreasing (PABFD) algorithm (Beloglazov and Buyya, 2012) is presented in algorithm 3.2.

The proposed algorithm first sorts the list of VMs for migration in descending order of CPU utilization and also the list containing all physical hosts in the data center are sorted in descending order of their average performance to power ratios. This is done to consider the average performance to power ratio *AverPerf2Pow* of physical hosts(PM) for prioritizing PMs for VM placement requests. For each VM in the migration list, the physical hosts are checked for placement suitability and the estimated power consumption after VM placement is calculated for all the suitable physical hosts. The energy and performance efficient physical machine among all the

Algorithm 3.2: Power and performance aware BFD(PPABFD)

```
Input  : pmList,vmList
Output: VMAllocationList
Sort vmList in the descending order of CPU utilization
Sort pmList in descending order of their average performance to power ratios
foreach $vm \in vmList$ do
    minPower = MAX_ VALUE;
    PMAssigned = NULL;
    foreach $pm \in pmList$ do
        if isSuitablePM(pm,vm) then
            power = estimatedPower(pm,vm);
            if $power < minPower$ then
                PMAssigned = pm;
                minPower = power;
            end
        end
    end
    if PMAssigned != NULL then
        VMAllocationList.add(vm, PMAssigned);
    end
end
return VMAllocationList;
```

suitable physical hosts is selected.

The algorithm ensures that the host machines(PM) with higher *AverPerf2Pow* are prioritized for VM allocation to maximize utilization of the power-efficient physical machines for power saving in DC. The algorithm PPABFD returns new VMs to PMs allocations, which are efficient in terms of power and performance efficiency. The time complexity of the algorithm 3.2 is O(n^2).

3.6.3 Host underload condition

The host underload detection and switch off process is essential in data centers to save power when the data center is not experiencing a heavy load. The host underload detection process detects physical hosts with CPU utilization lesser than a defined static threshold and selects underutilized physical hosts for power off. The VMs residing on these selected underloaded hosts are migrated out before the host power-off. However, the underloaded host selection technique should also consider the power efficiency of the underutilized physical hosts for selecting a particular host for power off.

When the data center is experiencing lesser workload requests, the proposed VM placement optimization algorithm considers switching off the host machines with lower power efficiency to maximize power saving benefit. The algorithm 3.3 presents our proposed host underload detection and underutilized host selection technique for power off. The proposed technique takes into account the performance to power ratio *AverPerf2Pow* of the underutilized host machines for host selection to power off. Algorithm 3.3 ensures that the physical host(PM) with CPU utilization lesser than minUtilization and which is least power-efficient is switched off, thereby saving power in a non-peak duration in the data center. The power-off of the physical host is performed only after all the VMs residing on it are migrated out successfully. The time complexity of the algorithm 3.3 is O(n).

Algorithm 3.3: Underloaded host detection algorithm

```
Input  : pmList
Output: underUtilizedHost
/* Initialize to static threshold value for under utilization
   check                                                        */
minP2PRatio = MAX_VAL;
minUtilization = LOWER_TRESHOLD;
/* Select power inefficient Host with lower than threshold CPU
   utilization                                                  */
foreach $pm \in pmList$ do
    utilization = getCurrentUtilizationOfCpu(pm);
    if (utilization > 0) && (utilization < minUtilization) then
        power2PerfRatio = getPerf2PowerRatio(pm);
        if power2PerfRatio < minP2PRatio then
            underUtilizedHost = pm;
            minP2PRatio = power2PerfRatio;
        end
    end
end
return underUtilizedHost;
```

3.6.4 Load context detection in datacenter

One of the objectives of the reported work in this thesis is to consider the global data center load conditions for the VM placement optimization process. We present a load context detection algorithm in algorithm 3.4 with a defined static threshold utilization. The algorithm accesses the information stored by a local context manager (LCM) at each physical host such as VMs running on hosts and their MIPS utilization to arrive at the overall host CPU utilization. Once host utilization data is summed up for all hosts in the data center, the proposed solution calculates the average CPU utilization of data center servers. If the data center has an average CPU utilization of over MAX_UTIL_THR_DC then the proposed algorithm designates the current load context as peak load duration. Otherwise, it is considered as the normal/non-peak duration in the data center. The time complexity of the algorithm 3.4 is O(n^2). The total host utilization(TotalHostUtilization) is calculated by summing up all VMs MIPS utilization stored at LCM at each physical host and TotalHostUtilization for all hosts is used to calculate the average host utilization in the data center (AverageHostUtilizationsInDc). The average host utilization in the data center(AverageHostUtilizationsInDc) is compared against a defined threshold value of CPU utilization to trigger the peak load condition(to set isPeakSituationFlag). The algorithm 3.4 is invoked in algorithm 3.1 to get the current load context in DC.

Algorithm 3.4: DC load context detection algorithm

```
Input   : pmList
Output: isPeakSituationFlag
TotalHostUtilizationsInDc= 0;
AverageHostUtilizationsInDc =0;
isPeakSituationFlag = FALSE;
/* Measure total DC MIPS utilization                                  */
foreach $pm \in pmList$ do
    utilization = 0;
    foreach $vm \in VMListOf(pm)$ do
        utilization = utilization + getMipsUtilization(vm);
    end
    TotalHostUtilization = TotalHostUtilization + utilization;
end
AverageHostUtilizationsInDc =
TotalHostUtilization / NumHostsInDc;
if $AverageHostUtilizationsInDc > MAX\_UTIL\_THR\_DC$ then
    isPeakSituationFlag= TRUE;
end
return isPeakSituationFlag;
```

3.7 Experimental evaluation

The experimental evaluation of proposed context-aware VM placement optimization technique for power saving has been carried out against another well known adaptive heuristics-based technique for dynamic consolidation of VMs(Beloglazov and Buyya, 2012).

3.7.1 Performance metrics

Various performance metrics used in the evaluation of the proposed solution are described in this section.

1. **Energy consumption**

   The metric denotes the total power consumption of all the physical hosts operating in the data center. Any decrease in the total power consumption in the data center implies reduced power costs for the data center owners.

2. **Overall SLA violations**

   SLA Violations occur because of the performance degradation caused by non-optimal mappings of VMs-PMs. Performance degradation is due to the resource shortages for co-located VMs often caused by server over-utilization and also because of frequent migrations involving the same VM.

3. **Total VM migrations**

   Total VM migrations denote the number of re-mappings of VMs to available physical machines(PM) done during the given time. A very high number of VM migration may mean performance degradation and wastage of network bandwidth, computing resources on the source and destination nodes. A small number of VM migration may mean non-adaptability to dynamical situations in the data center.

4. **Total host(PM) shutdowns**

   The metric denotes the number of times the host machines(PM) are shut down in a given duration. The physical machines(PM) are switched off for power saving or any maintenance in data centers. Though PM shutdowns save power for the data center, frequent shutdowns may mean additional power consumption because of PM start-up or shutdown procedures and may also lead to hardware component failures in the physical machines over time.

3.7.2 Experimental setup

The CloudSim(Calheiros et al., 2011) is used for the evaluation of the proposed context-aware technique of VM placement optimization for power and cost-saving. CloudSim is a popular toolkit for simulation and modeling of the cloud environment and its applications among the research community. CloudSim provides both behavioral and system modeling of cloud components. Simulation can help to evaluate the performance of proposed architectures, algorithms, and applications prior to their deployment in a highly dynamic, scalable and distributed environment like cloud.

CloudSim helps cloud developers to test the accuracy and performance of their resource management and provisioning policies in a highly repeatable and controlled environment without any cost burden. CloudSim also helps to overcome any bottlenecks and any issues in runtime before deployment on the real cloud. CloudSim provides essential classes for modeling of data centers, service brokers, computing resources(CPU, RAM, network, etc), virtual machines, users, applications, and also policies for management of various system-level components such as resource scheduling and provisioning. Using the simulated cloud components, it is possible to evaluate new techniques governing the use of cloud resources by utilizing existing or adding new scheduling policies, load balancing algorithms, etc. It can also be used to testify the competence of proposed techniques from various perspectives such as cost, power consumption, and execution time. The layered architecture of CloudSim toolkit is shown in figure 3.6.

For the evaluation of our proposed solution, the heterogeneous data center is simulated using a composition of six different types of physical hosts(PM) with configurations listed in table 3.2.

The experiments are conducted on an HP Probook computer with a compute capability of Core i5 CPU and 8 GB RAM. The computer is driven by the Windows 7 operating system. The duration of the simulation is set to one day, which is a similar

Table 3.2: Physical Machine Configurations In Data Center

Serial No	**Machine Model Name**	**MIPS**	**Number of cores**	**RAM Size(in MBs)**	**Network BW(in GBs)**	**Type**
1	HP Proliant ML 110 G4	1860	2	4096	1	Small
2	HP Proliant ML 110 G5	2660	2	4096	1	Small
3	HP Proliant ML 110 G3	3000	2	4096	1	Medium
4	IBM server x3250	3067	4	8192	1	Medium
5	IBM server x3550[Xeon-X5675]	3067	6	16384	1	Big
6	IBM server x3550[Xeon-X5670]	2933	6	12288	1	Big

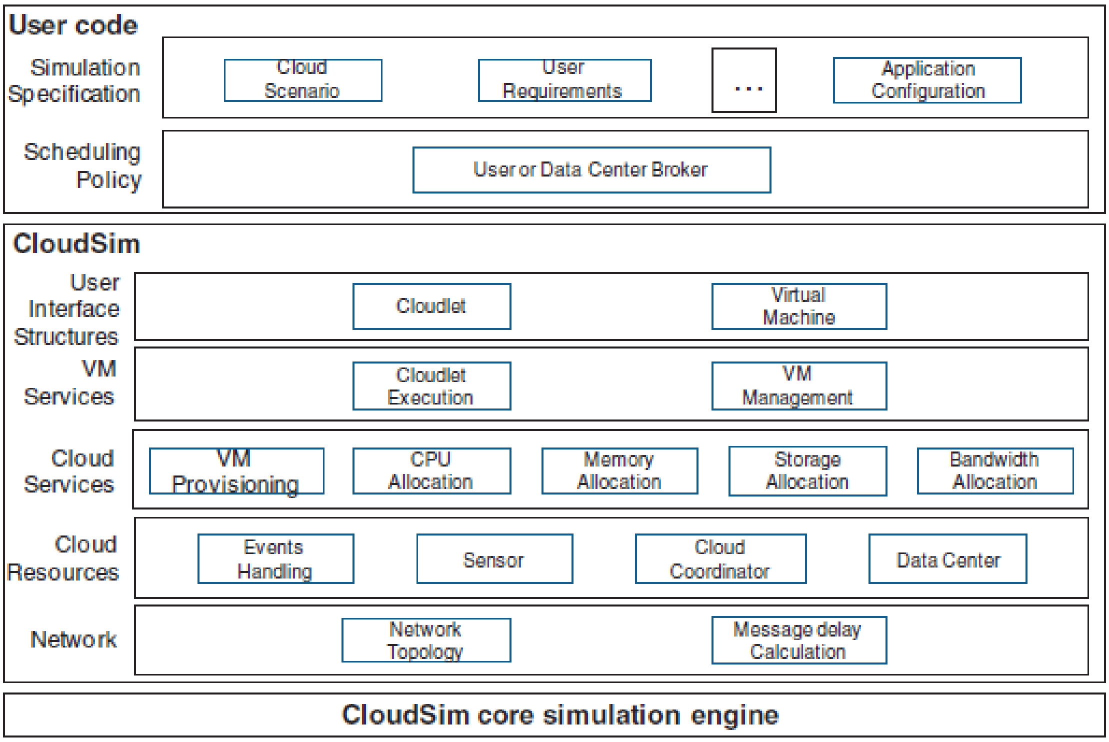


Figure 3.6: CloudSim Layered Architecture

duration used in the evaluation of another heuristic-based solution(Beloglazov and Buyya, 2012). The proposed context-aware VM placement optimization technique is invoked every 5 minutes once in the data center, which is a duration used in the VMWare distributed resource scheduler(Mosa and Paton, 2016) called DRS to adjust the VM-PM mappings.

The proposed solution is evaluated using two experiments carried out with two different natures of workloads and multiple distinct resource configurations in the data center. The objective of the first experiment is to testify the competence of

Table 3.3: Virtual Machines(VM) Configurations Used In DC

Serial No	**VM Type**	[CPU_MIPS,num_cores, RAM_in_MBs, VM_Size_in_GBs]
1	Type 1 [Extra Big]	[2500,1,870,2.5]
2	Type 2 [Big]	[2000,1,1740,2.5]
3	Type 3 [Small]	[1000,1,1740,2.5]
4	Type 4 [Extra-Small]	[500,1,613,2.5]

our proposed solution against synthetic workloads with a variable number of VMs to simulate a lightly loaded scenario to heavily loaded scenarios in the data center. The experimental configuration is chosen to ensure that our proposed solution is useful in different load conditions. The aim of the second experiment is to appraise the competence of our proposed context-aware solution against a real PlanetLab workload traces(PlanetLab, 2011) containing CPU utilization data of 1033 VMs. The data center configuration composing 400 PMs of six different host types is used.

3.7.3 Experiment 1: Synthetic workload with a variable number of VMs

The objective of the experiment is to testify the proposed VM placement optimization technique at different load conditions(lightly to heavily loaded) in a data center. The lightly loaded and heavily loaded situations indicate the overall load on the PMs considering available physical resource capacity of hosts and resource demands of co-located VMs. Five different configurations are chosen for testifying different load conditions.

- *Configuration 1.1: 100 VMs to be allocated to 100 PMs*
- *Configuration 1.2: 200 VMs to be allocated to 100 PMs*
- *Configuration 1.3: 250 VMs to be allocated to 100 PMs*
- *Configuration 1.4: 300 VMs to be allocated to 100 PMs*
- *Configuration 1.5: 400 VMs to be allocated to 100 PMs*

The simulation of the cloud data center is done composing of two physical machine types of HP ProLiant ML110 G4 and IBM server x3250 with configurations shown in table 3.2 and all types of VM configurations shown in table 3.3 are used to create VMs. Cloudlets are programmed to create utilization data every 5 minutes based on the stochastic model(Calheiros et al., 2011). The energy consumption results of the proposed solution, along with a heuristic-based solution(Beloglazov and Buyya, 2012) for five configurations of the synthetic workload are plotted in Figure 3.7.

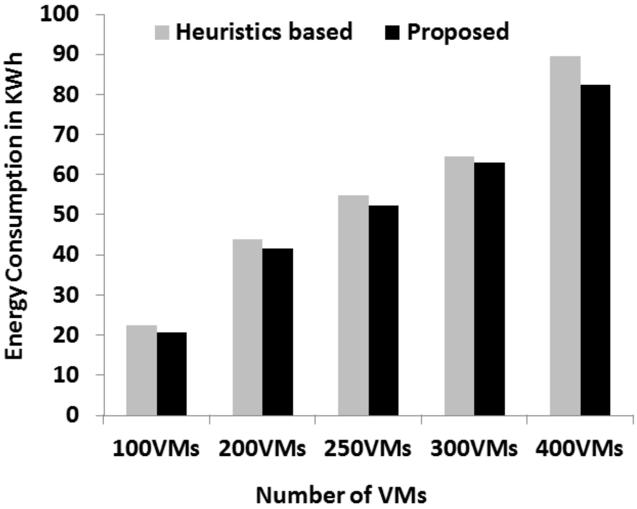


Figure 3.7: Comparison Of Power Consumption Results For Synthetic Workload

Results suggest that the proposed solution saves approximately 8-10% energy during lightly and heavily loaded cases and 2-6% during moderately loaded cases in the data center. The power-saving achieved can be attributed to the power efficiency aware VM placement and load context-based optimizations of VM placement. The results of all of the performance metrics for the experiment 1 are tabulated in table 3.4. The total VM migrations plotted in a graph in figure 3.8 for experiment 1(synthetic workload) indicate an high increase with lightly loaded to heavily loaded scenarios with heuristics based technique(Beloglazov and Buyya, 2012) due high number of host shutdowns and non-optimal VM-PM mappings. However, with the proposed solution, the number of VM migrations witness a small increase with lightly loaded to heavily loaded scenarios. This indicates that the proposed context-aware technique is able to generate better allocation strategy because of power efficient prioritization of hosts for VM allocation and DC load aware server consolidation strategy.

Figure 3.9 and figure 3.10 indicate that the overall SLA violations and total host shutdowns recorded for experiment 1 are much smaller in case of the proposed solution when compared with the heuristic-based past work for all the configurations. The proposed technique can avoid unnecessary host shutdowns and VM migrations by considering load context in the data center.

Table 3.4: Evaluation Results For Performance Metrics For Synthetic Workload

Number of VMs	**Energy consumption (in KWh)**		**Number of VM migrations**		**SLA perf degradation due to migration**		**Overall SLA violation**		**Number of host shutdowns**	
	Heuristics based	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**
100	22.51	20.69	6102	412	0.15%	0.01%	0.04%	0.01%	1155	175
200	43.96	41.57	11156	921	0.13%	0.01%	1.07%	0.02%	1969	277
250	54.79	52.24	13001	866	0.12%	0.01%	0.83%	0.01%	2304	249
300	64.72	63.14	15107	1070	0.12%	0.01%	0.96%	0.02%	2607	289
400	89.55	82.50	22068	1100	0.14%	0.01%	1.84%	0.01%	3003	292

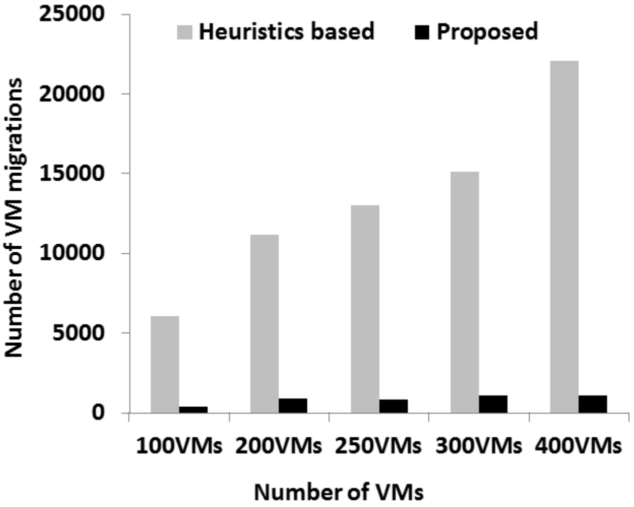


Figure 3.8: VM Migrations Results For Synthetic Workload

3.7.4 Experiment 2: Real-world workload with multiple PM types

The objective of experiment 2 is to evaluate the competence of the proposed VM placement optimization solution using a real-world workload in the data center. Three configurations are chosen for testifying different levels of heterogeneity with host machine types of varying power and performance characteristics.

- *Configuration 2.1: DC with 2 host machine types*
- *Configuration 2.2: DC with 4 host machine types*
- *Configuration 2.3: DC with 6 host machine types*

The cloud data center is simulated for experiment 2 using six types of physical machines with configurations listed in table 3.2 consisting of 400 PMs. The experiment utilizes a real-world workload consisting of resource utilization data of 1033 VMs(PlanetLab, 2011) captured in PlanetLab servers. The physical machine(PM) types used in all the three configurations are tabulated in table 3.5, and their configurations can be found in table 3.2.

Figure 3.11 shows the power consumption details of the proposed solution and adaptive heuristics-based technique(Beloglazov and Buyya, 2012). It can be noted from the figure that our proposed solution saves approximately 1-3% power compared to the heuristics-based technique.

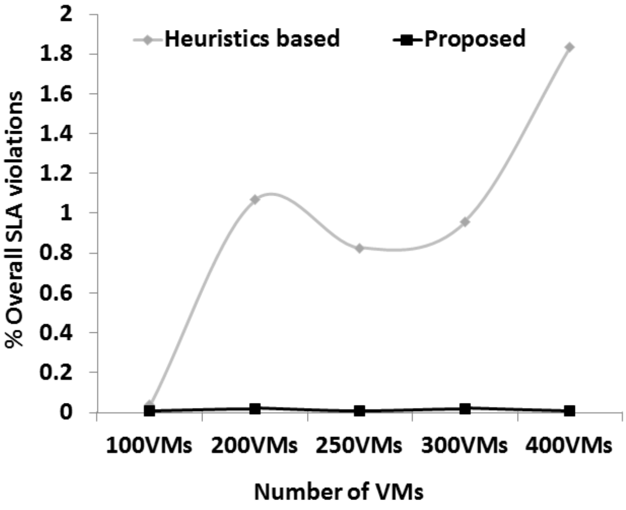


Figure 3.9: Overall SLA Violations Results For Synthetic Workload

The results of all the performance metrics for real world workload(experiment 2) are tabulated in table 3.6.

The graphs plotted for total VM migrations in figure 3.12 indicates that the VM migrations for the proposed context-aware technique are much smaller in number compared to the heuristics-based method for all configurations. The overall SLA violations and total host shutdowns recorded during experiment 2 are also much smaller in the case of the proposed solution for experiment 2 as shown in figure 3.13 and figure 3.14. The proposed technique can avoid a higher number of VM migrations by providing the near-optimal VM placement and by adopting a load aware underutilized host management technique than the adaptive heuristics-based technique proposed earlier(Beloglazov and Buyya, 2012).

Performance evaluation results suggest that the proposed context-aware VM placement optimization solution performs better than the heuristics-based technique for power consumption minimization and improves the efficiency of the operation by reducing VM migrations, host shutdowns, and SLA violations in both the experiments conducted. The proposed context-aware VM placement optimization technique can reduce power consumption by 2-10% for synthetic workloads and 1-3% for real workload traces in the data centers.
The key differentiating factors between proposed context-aware solution and heuristics-based technique are, using the performance and power characteristics of physical ma-

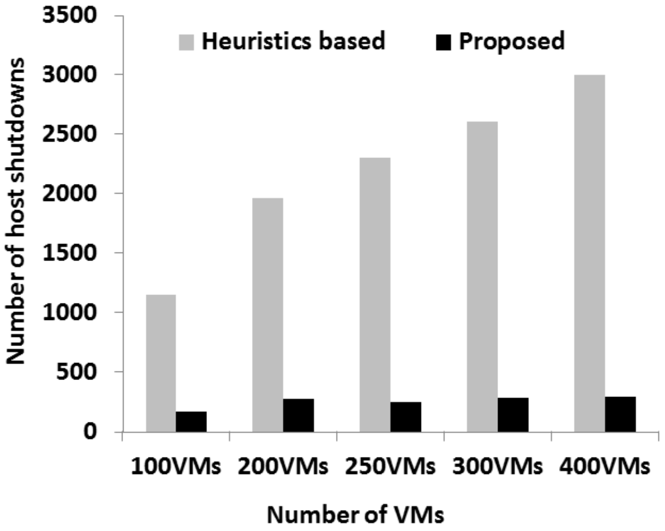


Figure 3.10: Number Of Host Shutdowns For Synthetic Workload

chines and detecting the global load context of the data center to improve the VM placement optimization efficiency.

Table 3.5: Physical Machines Types Used In Experiment 2

Configuration name	**Host(PM) types**
Configuration 2.1	HP ProLiant ML110 G4 IBM server x3250
Configuration 2.2	HP ProLiant ML110 G4 IBM server x3250 HP ProLiant ML110 G5 IBM server x3550 [XeonX5675]
Configuration 2.3	HP ProLiant ML110 G4 IBM server x3250 HP ProLiant ML110 G5 IBM server x3550 [XeonX5675] HP ProLiant ML110 G3 IBM server x3550 [Xeon X5670]

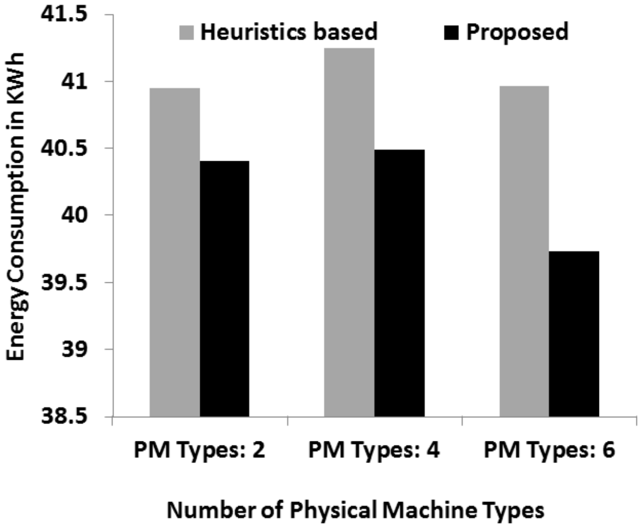


Figure 3.11: Power Consumption Results For Real Workload

Table 3.6: Evaluation Results Of Performance Metrics For Real World Workload

Types of Physical Servers	**Energy consumption (in KWh)**		**Number of VM migrations**		**SLA perf degradation due to migration**		**Overall SLA violation**		**Number of host shutdowns**	
	Heuristics based	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**	**Heuristics based**	**Proposed Solution**
2	40.95	40.41	16102	1875	0.07%	0.00%	0.08%	0.00%	2211	377
4	41.25	40.49	15855	1828	0.07%	0.00%	0.08%	0.00%	2228	382
6	40.97	39.73	15941	1716	0.07%	0.00%	0.08%	0.00%	2217	378

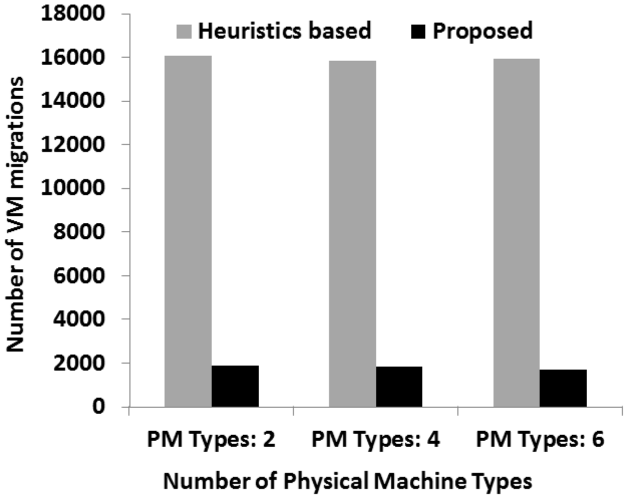


Figure 3.12: VM Migrations Results For Real Workload

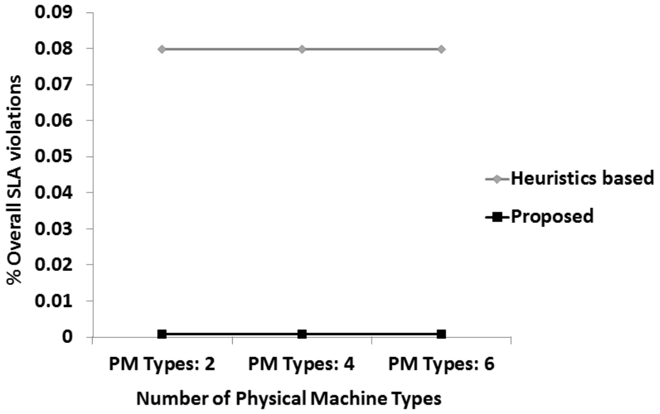


Figure 3.13: Overall SLA Violations For Real Workload

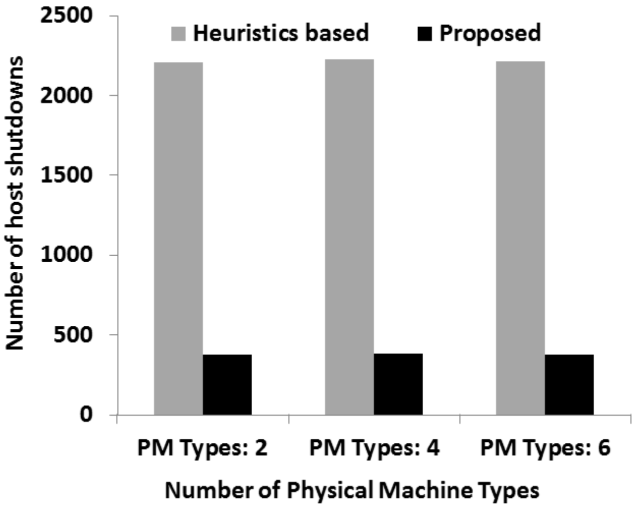


Figure 3.14: Number Of Host Shutdowns Reported For Real Workload

3.8 Summary

The chapter first introduced the technique of VM placement optimization and its sub-tasks. Then the chapter presented the research objective for the overall power saving in the data center, described the physical machine characteristics and its application to power saving, load condition detection and its consideration for underutilized host management. The proposed algorithms for VM placement optimization, VM placement, underutilized host selection, and DC load context detection are presented.Finally, the evaluation of the proposed technique with both synthetic and real-world workloads is described. The results obtained for proposed technique suggested that power saving of 2-10% for synthetic workloads and 1-3% with real-world workloads is achieved.

In the next chapter, an electricity cost-aware request routing(load distribution) algorithm for cloud service broker is presented for the power cost optimization in the geographically distributed data centers scenario.

Chapter 4

Electricity cost-aware load balancing in geo-distributed data centers

The adoption of cloud services by businesses across the globe is overgrowing, and many new services and customers consuming these services are added at an ever-increasing pace. Because of this growth, cloud providers like Amazon, Microsoft, and Google have set up many geographically dispersed data centers, and they are continuing to build more to support computing demands of their user bases. In the arena of internet applications like those hosted on cloud, the speed and latency are of utmost importance. The necessity creates a motivation for building geographically distributed data centers around the world to reduce speed-of-light delays for user applications hosted on cloud and accessed around the globe. But such a distributed set up of data centers over various geo-locations create a new set of research problems and opportunities. One such research problem addressed in this chapter is determining how to distribute the user application traffic(load) across geographically dispersed data centers to minimize the cost for data center providers.

The data centers need huge capital investments at the beginning of setting up IT and non-IT infrastructure and later incur management costs for data center maintenance and power(electricity) consumption to keep data center up for 24x7 operations. It is noted that 15% of overall data center amortized costs(Greenberg et al., 2009) corresponds to power/electricity cost.

Electricity is generated using various methods across the world, and its availability and volume are not uniformly distributed. The cost of electricity at a geographical location depends on various factors like availability of natural resources, technology involved for generation, and cost of infrastructure needed for generation. Electricity costs also found to vary based on the time of the day, total units consumed, etc based on the domestic rules of each country.

It is essential to minimize data center management costs for the cloud providers to help reduce the cost of ownership of a large scale computing facility like cloud data centers. The distributed data centers provide an opportunity to utilize the electricity price variability across the globe to optimize power costs. The rest of the contents in this chapter is organized as follows, section 4.1 introduce the functions of the cloud service broker briefly; section 4.2 presents the objective of our proposed work. The power cost-aware technique to load balance user requests among geographically dispersed data centers is described in 4.3, the experimental setup and configurations are presented in section 4.4, and experimental results are discussed in section 4.5.

4.1 Background study

Cloud service broker is responsible for controlling traffic routing between users and data centers in a geographically distributed data center set up. Cloud service broker distributes the user requests for cloud applications across multiple available DCs based on a load balancing algorithm/policy. The figure 4.1 shows the functions of service broker module in a cloud computing environment.

The cloud service broker routing policies(Wickremasinghe et al., 2010) that are commonly used are listed below.

- **Proximity based routing** - The closest data center in terms of transmission delay is considered for routing.

- **Performance optimized routing** - The performance of all data centers is monitored, and traffic is routed to the data center which is estimated to give the best response time to the user.

- **Dynamically re-configuring routing** - It is very similar to proximity-based routing, but it has an additional responsibility of scaling a load of a data cen-

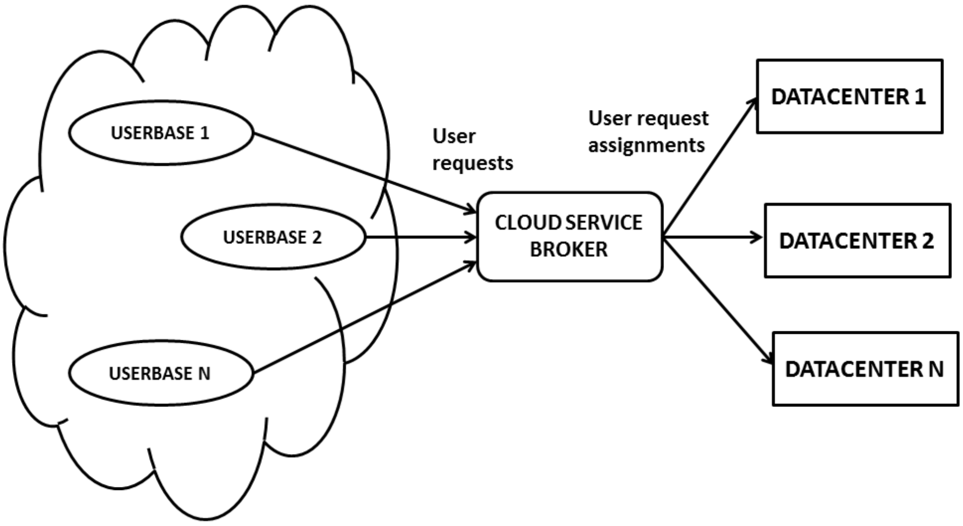


Figure 4.1: Cloud Application Service Broker

ter by increasing or decreasing VM allocation based on current performance comparing against best performance ever achieved with that data center.

The proposed work in this chapter describes a new task/request distribution algorithm for the service broker to optimize the cost of power consumption for data center owners without affecting the performance of user applications.

4.2 Research objective

The objective of the proposed work is to distribute more requests(load) on data centers where electricity/power cost is cheaper at that point of time to optimize the total power cost and also ensuring the response time is the same as or better than the closely located data center. The proposed power cost optimization problem can be mathematically presented as follows,

$$EC(N) = \sum_{i=0}^{N} n(i)E(it)Pc \tag{4.1}$$

In equation 4.1,
EC(N) denotes the total cost of electricity(power) for N data centers,
n(i) represents the number of user requests processed by i-th data center,
E(it) is the electricity cost at i-th data center location at time t,
Pc denotes electricity consumed by server per unit request which can be considered

as a constant value.
The objective of the proposed work is to minimize the value of EC(N).

4.3 Electricity cost-aware cloud service broker policy

The proposed technique aims to leverage the varying electricity price around the world to optimize the power costs for data center owners. The proposed technique for cloud service broker distributes user compute workload(requests) among available data centers by incorporating electricity prices prevailing in the DC regions as a decision parameter. The electricity price is modelled as a two dimensional context variable that varies with both place and time or amount of consumption. The 2-D table used to represent electricity price is referred to as Electricity cost matrix(or EC matrix) in this report and can be represented as shown in table 4.1. This EC matrix will have one row for each of the data centers and each of the columns indicating another parameter with which electricity cost varies for that geo-location, for example time of the day as shown in table 4.1. The EC Matrix should be updated by administrator based on domestic rules and made available to the cloud service broker at all the time. The proposed cost-aware algorithm placed at the cloud service broker accesses

Table 4.1: Electricity Cost Matrix Representation

Geo Location	**00:00-5:00**	**5:01-9:00**	**9:01-19:00**	**19:01-23:59**
DC Location X	x1	x2	x3	x4
DC Location Y	y1	y2	y3	y4
DC Location Z	z1	z2	z3	z4
...	...	...	...	...
DC Location N	n1	n2	n3	n4

the following details about all the available geographically dispersed data centers.

1. The closest data center available to the request in terms of its transmission delay and its estimated response time.

2. The updated EC Matrix containing the electricity price of all DC locations.

3. The estimate of response times for all data centers for the current request.

The criteria of the proposed cost-aware service broker algorithm for allocating requests to a cheaper data center in terms of electricity prices are as follows.

1. The data center should have a response time lower than the closest data center.

2. The electricity cost of the selected data center should be lesser than other available data centers that satisfy the first criteria.

The algorithm for the cost-aware algorithm in the cloud service broker is presented in algorithm 4.1.

The proposed algorithm 4.1 accesses the details of available geo-distributed data centers($allDataCenters$) and the updated EC matrix($allDccosts$) during initialization process. The algorithm accepts user base location of the incoming request as input and finds the closest DC($closestDc$) located to the corresponding user base location using the transmission delay matrix. The algorithm then calculates estimated response times for all the available data centers($allDcEstTime$) for the current request using the network delay and last recorded response time($bestRecordedresponseTime$) from the corresponding DC. The estimated response time for closest DC($closestEstResponseTime$) for current request is also calculated. Once required parameters such as estimated response time for all candidate DCs including closest DC($allDcEstTime$), electricity cost matrix($allDccosts$) for processing current request are recorded, the cost-aware algorithm finds the data center Id($dest$) for which the estimated response time is smaller than closest DC estimated response time($closestEstResponseTime$) and electricity price(per unit price for power) is lesser than closer DC. The selected data center ID($dest$) for which there exists an estimated response time($leastEstRespTime$) lower than closest DC and having the electricity price advantage is returned for request assignment otherwise closest data center Id($closestDc$) is returned for the incoming

Algorithm 4.1: Electricity cost-aware request routing technique

```
Result: Finds cheaper DC with best response times
Input  : src- Source location of request
Output: dest- Datacenter id for request routing
/* Initialization                                                   */
allDataCenters= getlAvailableDatacenterIds();
allDccosts =getECCostsforDCs(allDataCenters);
allDcEstTime = MAXTIME;
closestDc= findClosestDc();
/* Calculate estimated response times for all DCs                   */
foreach DataCenterId ∈ allDataCenters do
    nwDelay = getNetworkDelay(src,DataCenterId);
    bestRecordedresponseTime = getBestResponseTime(src,DataCenterId);
    currEstResponseTime = nwDelay + bestRecordedresponseTime;
    allDcEstTime[DataCenterId] =currEstResponseTime;
    if DataCenterId == closestDc then
        closestEstResponseTime = currEstResponseTime;
    end
end
dest = closestDc;
leastEstRespTime = closestEstResponseTime;
/* Find fastest and cheapest DC                                     */
foreach DataCenterId ∈ allDataCenters do
    if EstResponseTime(DataCenterId) < leastEstRespTime then
        if (getECCost(DataCenterId) < getECCost(dest)) then
            dest = DataCenterId;
            leastEstRespTime = EstResponseTime(DataCenterId);
        end
    end
end
return dest;
```

request. The time complexity of the algorithm 4.1 is O(n). The proposed cost-aware technique ensures that response time for the request processing is optimal than closely located DC with no degradation of service quality and also geo-distributed data center owners will save power cost.

4.4 Experimental setup

The proposed technique is evaluated using the CloudAnalyst(Wickremasinghe et al., 2010) toolkit widely used by researchers as a simulation tool for evaluating the competence of the cloud computing resource management policies and applications. The section provides a brief introduction of the CloudAnalyst tool and experimental configuration parameter settings used for our evaluation.

4.4.1 CloudAnalyst

CloudAnalyst(Wickremasinghe et al., 2010) is a GUI based open source cloud simulation tool to support simulation and visual modeling of large scale cloud applications. CloudAnalyst is built on top of CloudSim and provides many additional extended features to describe application workloads, geographically dispersed data centers, distributed user bases, and also supports configuring numbers and settings of hardware/software resources in data centers. With CloudAnalyst, application developers and researchers can develop and evaluate resource provisioning, scheduling and application deployment strategies for distributed data centers and users.
The block diagram of CloudAnalyst is shown in figure 4.2. It is built on the Cloudsim framework and extends some of its classes to model complex internet and application parameters. The CloudAnalyst provides a GUI layer to aid in conducting quick and complex experiments with high degrees of flexibility and ease. Because CloudAnalyst is an open-source simulator and built using a modular design, it is easy to extend the tool to support a new feature or modify its behavior to support a new perspective like cost of service.

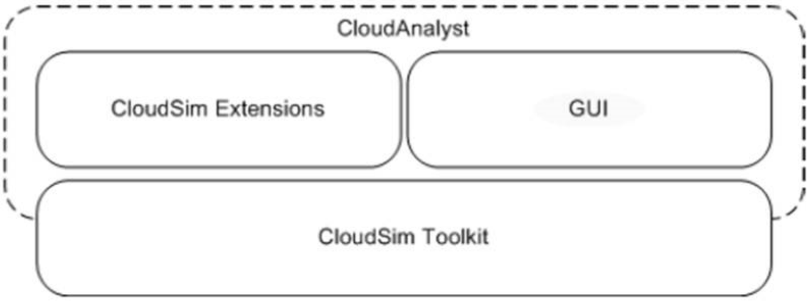


Figure 4.2: Block Diagram Of CloudAnalyst

4.4.2 Experimental configurations

The section explains the experimental configurations used for the evaluation of the proposed cost-aware technique for request routing in a geo-dispersed data center setup.

A Electricity price for all DC locations

The price of electricity at various geo-regions where data centers are set up is mentioned in table 4.2. The sample electricity cost/price values used in the experiments are based on Wikipedia source(Wikipedia, 2017) available on the web. The table 4.2 is referred to as EC Matrix. The EC Matrix considered for the evaluation shows variability based on geo-locations but does not change with any other parameter like time of day, units consumed, etc for the experiments.

Table 4.2: Electricity Cost Table

Data center Name (Location)	**Electricity Cost (in $/kWh)**
DC1(USA)	0.17
DC2(Brazil)	0.25
DC3(UK)	0.21
DC4(China)	0.24
DC5(Africa)	0.13
DC6(Australia)	0.22

B Data centers and user bases

The evaluation of the proposed technique is done considering users of six different geographical locations accessing cloud services from data centers located at six geographic locations around the world. The data centers have configurations shown in table 4.3 for the experiments.

Table 4.4 lists the user base configurations of six geographical regions used for the experiments. The rest of the settings like hypervisor, OS, memory, hardware

Table 4.3: Data Center Configurations

DC Name	Region	Number of VMs	Bandwidth(in mbps)
DC1	USA	500	1000
DC2	Brazil	500	1000
DC3	UK	500	1000
DC4	China	500	1000
DC5	Africa	500	1000
DC6	Australia	500	1000

configuration are considered uniform for all DCs. The experiment duration is set as 60 hours.

Table 4.4: User Base Configurations

DC Name	Region	Req/Hr	Req Size	Avg Peak Users	Avg Non-Peak Users
UG1	USA	60	100	1000	100
UG2	Brazil	60	100	1000	100
UG3	UK	60	100	1000	100
UG4	China	60	100	1000	100
UG5	Africa	60	100	1000	100
UG6	Australia	60	100	1000	100

The transmission delay matrix is given in table 4.5 is used in the experiments to search the closest located data center for any request received from user bases at the cloud broker for DC assignment.

4.5 Experimental results and analysis

The experiments are conducted using multiple combinations of the user bases, and data centers and results are presented in this section. The experiments are performed for five different categories of user groups and the data centers. The format used to represent each category is as follows.

Table 4.5: Transmission Delay Matrix Between Regions(in msec)

Regions	**USA**	**Brazil**	**UK**	**China**	**Africa**	**Australia**
USA	25	100	150	250	250	100
Brazil	100	25	250	500	350	200
UK	150	250	25	150	150	200
China	250	500	150	25	500	500
Africa	250	350	150	500	25	500
Australia	100	200	200	500	500	25

$$(Usergroups), (geo - distributedDatacenters)$$

For example,

$$(UG1), (DC3, DC4, DC5, DC6)$$

implies user group 1 located in the USA region can access the services from data centers located in the United Kingdom(U.K), China, Africa and Australia for this category of the experiment.

Table 4.6 tabulates the request assignments for the closest data center and cheapest data center corresponding to five categories of experiments. It can be observed from table 4.6 that for experiments E4 and E5, the proposed technique is able to find cheaper data center with estimated response time smaller than closest data center and assign significant number of incoming requests.

Table 4.7 summarizes the total power costs for closer DC and cheaper DC assignments for proposed cost-aware technique and also power costs of closest DC only(proximity based routing technique) assignments for five categories of experiments. The power consumed per unit request is considered as a constant(Pc) of 0.1KWh for the experiments and power costs are calculated using EC Matrix per geo-location as shown in equation 4.1. It can be noted from table 4.7 that the request assignment to cheaper data centers in case of experiments E4 and E5 has reduced the power costs by 15-23%.

Table 4.6: Proposed Service Broker Request Assignments

Exp Name	Experimental Combination	Total requests received	Assignments to Closest DC	Assignments to Cheapest DC
E1	(UG1),(DC3,DC4,DC5,DC6)	69425	69109	316
E2	(UG2), (DC3,DC4,DC5,DC6)	69425	65340	4085
E3	(UG1,UG2), (DC3,DC4,DC5,DC6)	139063	134820	4243
E4	(UG4),(DC2,DC5,DC6)	69365	35524	33901
E5	(UB3,UB4),(DC2,DC5,DC6)	139063	105040	34023

Table 4.7: Summary Of Power Costs For Proposed Technique

Experiment	**ClosestDC Cost**	**CheaperDC Cost**	**TotalCost (Cost-aware)**	**TotalCost (ClosestDc only)**	**Cost Saving**
E1	$1520.39	$6.63	$1527.03	$1527.34	0.02%
E2	$1437.47	$85.68	$1523.16	$1527.34	0.27%
E3	$2966.03	$88.99	$3055.03	$3059.38	0.14%
E4	$888.09	$440.71	$1328.81	$1735.62	23.43%
E5	$1790.34	$442.29	$2232.64	$2640.91	15.45%

Figure 4.3 summarizes the percentage-wise assignment of proposed technique for load(request) distribution to available data centers. The data center selection criterion of the proposed cost-aware algorithm is to find a cheaper data center with better response time than the closely located data center. It can be observed from the results that the E1 category has fewer assignments(0.4%) for cheaper data centers because the data center DC6, which is located close(in terms of transmission delay) to the user base of requests is also having estimated response time smaller than other competing data centers for most of the requests. The E2 and E3 categories have 3-6% request assignments for cheaper data centers whenever closely located data center DC6 has higher estimated response time for request than cheaper data center DC3. It can be noted from experiment categories E4 and E5 that cost-aware service broker technique is able to allocate 24-49% of requests to cheaper data centers with significant power cost-saving for data center owners.

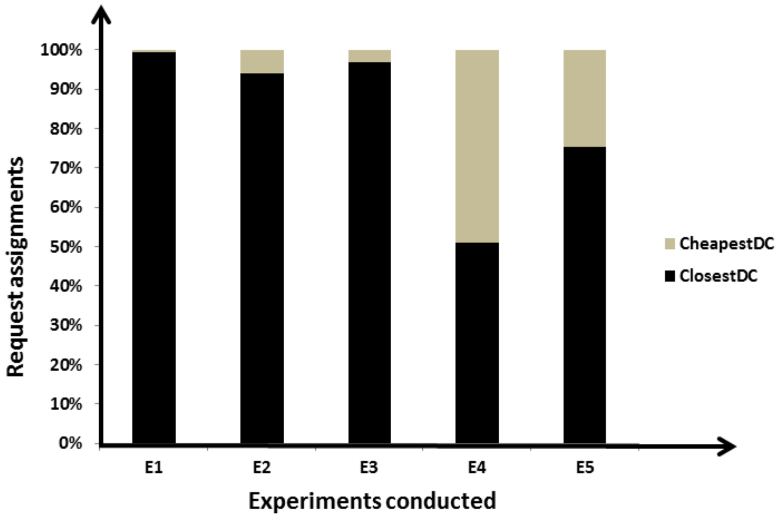


Figure 4.3: Request Assignment Percentage

Figure 4.4 presents the power costs for both proposed cost-aware technique and closest DC only allocations to indicate the total power cost saving achieved. It can be noted from experiment categories E4 and E5 that, the proposed cost-aware technique for cloud service broker can save 15-23% of power cost for cloud data center owners.

It is evident from the evaluation results that the proposed cost-aware request routing algorithm saves power cost of 15-23% for data center owners when there

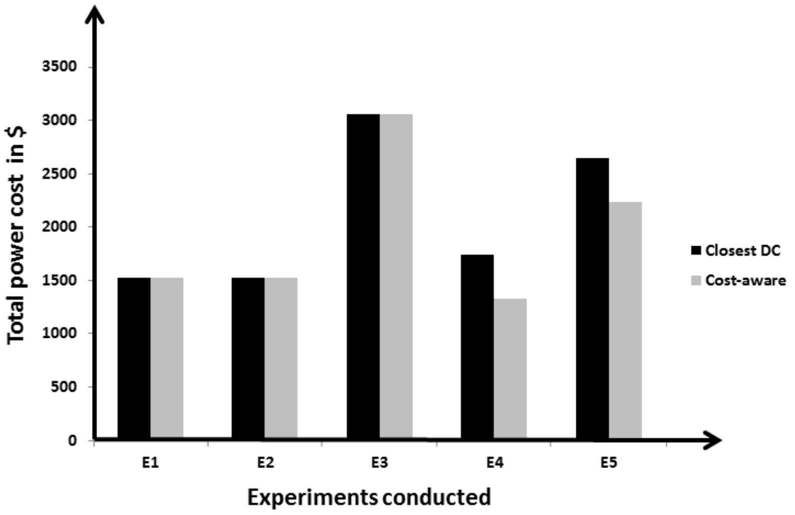


Figure 4.4: Comparison Of Power Costs

exists an opportunity of routing the requests(processing load) to cheaper data centers with no degradation in response times for user requests.

4.6 Summary

The chapter proposed an electricity cost-aware request routing technique to distribute tasks to data centers in a geographically distributed data center setup. The chapter introduced the cloud service broker module and three of the well-known request routing techniques employed by the cloud service broker. Then, the chapter presented the research objective addressed, described the proposed cost-aware request routing algorithm in detail. The experimental setup and a discussion on results are presented to prove the effectiveness of the proposed solution for power cost saving.

The next chapter describes the equally spread current execution(ESCE) load balancing algorithm and then a problem observed with it during the peak load condition is discussed. The chapter proposes a resolution to the problem and the experimental evaluation of the proposed solution is presented at the end.

Chapter 5

Peak hour Performance Improvement for ESCE Algorithm

In recent years, cloud computing has witnessed explosive growth because of the advancement of networking technology and ease with which the cloud services(computing hardware and software) can be rented and operated. Cloud computing has finally made the idea of offering computing as a utility a reality, and since then cloud has been embraced by millions of users across the world and also by giant IT companies like Amazon, HP, IBM, Microsoft, Apple, Google, Oracle, and others. The scalability and efficiency features of the cloud can only be achieved by proper management(utilization) of cloud resources. The essential characteristic of the cloud is the ability to manage and access the cloud resources in virtual form. The users access the cloud resources by submitting their work requests to virtual entities of computing called virtual machines(VMs) on rental basis. It is vital to balance the work requests(load) from users across available virtual machines to achieve resource efficiency through optimal utilization of underlying computing resources.

The load balancing in cloud data centers is done over both physical hosts or VMs. In the case of VMs, the load balancing algorithm distributes the cloud users dynamic workload equally among all the VMs. The performance of load balancing mechanism is critical during peak hours in data centers to meet stringent performance SLAs through optimal utilization of computing resources in the data centers. The over and under allocation of load to even few VMs can cause performance degradation and cause SLA violations. Our work reported in this chapter investigates the Equally spread current execution load(ESCE) algorithm for a problem with managing uniform resource utilization during high traffic situations and proposes a solution to address the problem.

5.1 Background study

The section explains the user task scheduling model in cloud data centers and briefly describes the ESCE algorithm for task load balancing.

5.1.1 Task scheduling in cloud data centers

The model used for task scheduling in the cloud data center is shown in figure 5.1. The cloud system contains N hosts and each running more than one VMs. The load balancing is required in a system where there is a huge number of inputs tasks submitted to cloud need to be assigned to a finite set of virtual machines. The VM manager(Mishra et al., 2018) receives the input tasks submitted to cloud system from the task queue. The VM manager has the information about the active VMs available in the cloud data centers and available resources with different hosts. If available resources are enough to complete the submitted tasks, the tasks are forwarded to the task scheduler called task load balancer. If enough resources are not available to process input tasks, new VMs are created in the data center to cater to additional resource demands. The task scheduler acts as a load balancer to map each task to the available VMs based on the resource requirements of each task and current load on each active VM.

5.1.2 Equally spread current execution load algorithm(ESCE)

Equally spread current execution (ESCE) load algorithm(Mali and Vidya, 2013) also known as active VM load balancer is a tasks-to-VM load balancer. The objective of the ESCE algorithm is to equally spread the execution load on different VMs in a data center to achieve uniform resource utilization. Active VM load balancer maintains a VM table with VM id and the number of requests currently allocated to

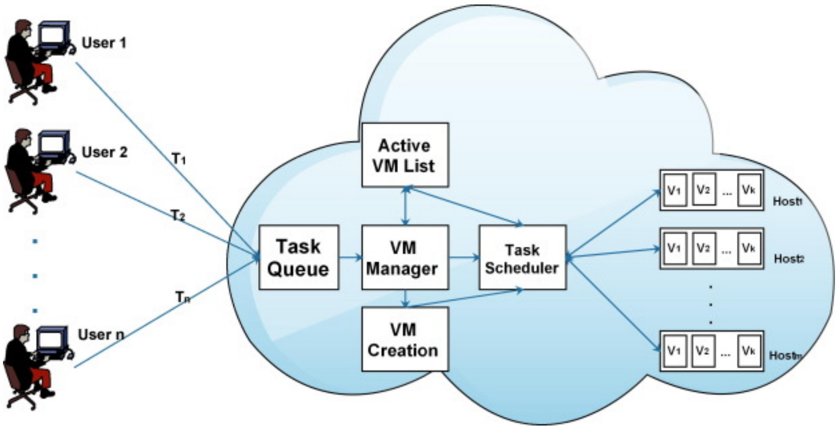


Figure 5.1: Task Scheduling Model In Cloud

each VM id. If a task(request) is submitted to the data center task queue for execution, the load balancer will search the VM table for least loaded VM(VM with least number of request assignments). If more than one VM is found with equal number of task assignments, first identified VM is selected and mapped for the task execution. The load balancer updates the VM table by increasing the allocation count of identified VM. When VM finishes the execution of allotted task, load balancer again update the VM table by decreasing the allocation count for identified VM by one. The steps of VM identification and VM allocation count update are done based on event triggers by the data center controller.

Table 5.1: Example Of VM Allocation By ESCE Algorithm

Task ID	**VM ID 0**	**VM ID 1**	**VM ID 2**
Init	0	0	0
0	1	0	0
1	1	1	0
2	1	1	1
3	2	1	1

The example of VM allocations by the ESCE algorithm is shown in table 5.1. The table shows a scenario where tasks are allocated to 3 VMs in the data center using the ESCE VM allocation algorithm. Initially, all VM Ids contain zero allocations as indicated in the first row in table 5.1. It can be noted that the ESCE algorithm found

VM Id with the minimum allocation(Zero in this case) in case of task ids 0,1 and 2. When task id 3 is requested for allocation, all VM IDs are having an equal number of allocations. Hence in case of task id 3, the VM Id 0 is allocated to the task.

The problem in uniform utilization of VMs is observed with active VM(ESCE) algorithm during the high traffic situations. The VM table update process in ESCE is invoked by the data center controller(DCC) task allocation and de-allocation events. When the data center controller requests ESCE algorithm for the least loaded VM id for allocation, the VM id is found and returned to DCC. However, the VM table update process is deferred until the VM id is mapped to the task by the data center controller and notification for the allocation is sent to ESCE(active VM load balancer). During the VM identification, VM allocation, and VM table allocation process, if any new requests for VM identification for tasks are received by ESCE algorithm, the VM table does not reflect the current state of the system and causes state inconsistency. The VM table state inconsistency problem is frequently observed during peak hours when huge number of tasks are submitted to the cloud system for processing. The work discussed in this chapter offers a resolution to this problem.

5.2 Research objective

The objective of the proposed work in this chapter is to achieve uniform resource utilization of all VMs at all times in the data center for the ESCE task load balancing algorithm. The proposed work investigates the problem of over-allocation of VMs with ESCE algorithm during high traffic situations and proposes a solution to overcome the problem.

5.3 Proposed VM load balancer

The proposed VM load balancer is a modified version of the ESCE algorithm(Mali and Vidya, 2013) to solve the problem of over-allocation of VMs in peak hour situations. The proposed algorithm uses an intermediate VM reservation table to record identified VM recommendations to the data center controller until the identified VM is allocated to the input task by data center controller and notification is received by the proposed VM load balancer. The proposed load balancer takes into account both VM table and reservation table entries for the identification of least loaded VM in the set of available VMs in the data center. The proposed VM load balancer algorithm is presented in algorithm 5.1.

Algorithm 5.1: Modified active VM(ESCE) algorithm

```
Result: Finds a least loaded VM for the given request
Input  : VMList- List of active VMs
Output: VMid- Datacenter id for request routing
/* Initialization                                          */
VMAllocTable= getVMAllocTable();
VMReserveTable =getVMReserveTable();
LeastLoadedVMid = INVALIDID;
minVMCount = MAXVALUE;
/* Find a VM with zero request allocations                 */
foreach  VMid ∈ VMList do
    if VMAllocTable[VMid] == 0 then
        VMReserveTable[VMid] +=1;
        return VMid;
    end
end
/* Find a VM with least request allocations                */
foreach  VMid ∈ VMList do
    currVMCount = VMAllocTable[VMid] + VMReserveTable[VMid];
    if currVMCount < minVMCount then
        LeastLoadedVMid = VMid;
        minVMCount = currVMCount;
    end
end
VMReserveTable[LeastLoadedVMid] +=1;
return LeastLoadedVMid;
```

The proposed load balancer returns the VM id for allocation to the data center controller, and once task is allocated to suggested VM id, a notification is sent to the proposed load balancer to increment the VM table entry meant for allocations and decrement the count of reservation table of the allocated VM id as shown in the call flow diagram in figure 5.2.

The proposed VM load balancer unlike the ESCE algorithm maintains an internal reservation table(*VMReserveTable*) to maintain the information of VM reservations suggested by the load balancer to the data center controller but not updated in allocation table until the notification arrives of allocation. The proposed load balancer takes into consideration both reservations table entry(*VMReserveTable*) and allocation statistics table entry(*VMAllocTable*) for particular VM id by the load balancer for VM selection for the next request. The line numbers 2, 7 and 11-17 in algorithm 5.1 represent the modifications done to the ESCE algorithm by our proposed load balancer. The modifications proposed to the ESCE algorithm avoids overloading of VMs during peak hours and also help reduce response time for tasks waiting on an overloaded VM. The time complexity of the algorithm 5.1 is O(n). The experimental results with the proposed load balancer indicating uniform resource utilization is shown in table 5.3 and table 5.4 with description provided in next section.

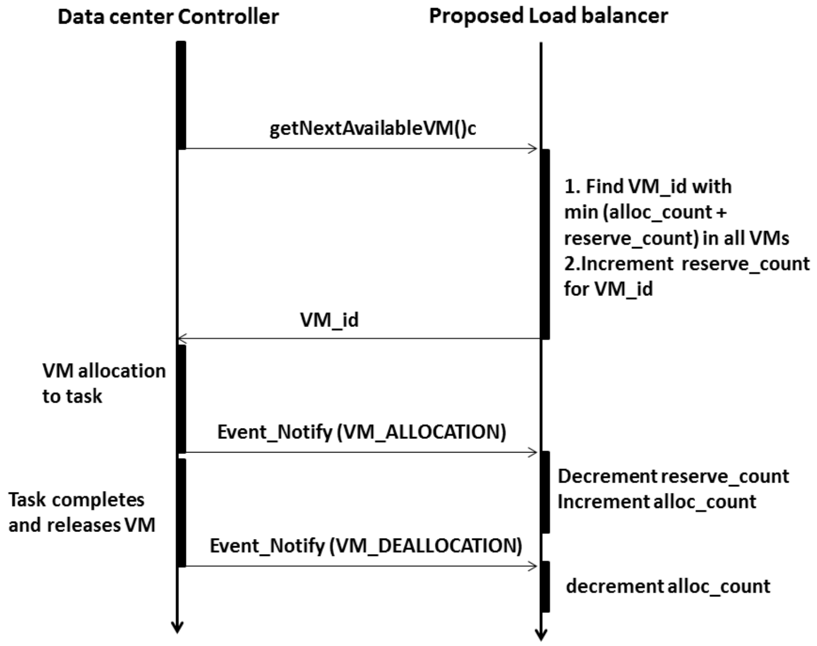


Figure 5.2: Call Flow Diagram For Proposed Load Balancer

5.4 Experimental setup

The experimental evaluation of the proposed VM load balancing algorithm has been carried out on a well-known simulator called CloudAnalyst(Wickremasinghe et al., 2010). CloudAnalyst is a simulation tool based on cloudsim library, developed using java and provides a GUI interface to configure various user and data center parameters to perform the experimental work with ease. The experiments to evaluate the competence of the proposed algorithm have been carried out using the configuration of internet users at four different continents, i.e. four user bases along with peak and non-peak users configurations used are given in the table 5.2. The requests(tasks) of having unit length are considered for simplicity. Data center hosts homogeneous physical machines having hardware resources with configurations of 100GB of storage, 4 GB of RAM with each physical machine(PM) equipped with 4 core CPU having 10K total MIPS(million instruction per second).

Table 5.2: User Bases: Regionwise Statistics Of Users

Region No	**Region Name**	**Peak Time Users**	**Off-Peak Users**
0	North America	35000	3500
1	South America	25000	2500
2	Europe	15000	1500
3	Asia	5000	500

5.5 Experimental results and analysis

The experiments are conducted with two simple configurations of 5 VMs hosted on two physical machines and 25 VMs hosted on 10 physical machines having configurations described in the previous section. The results are analyzed with uniform resource utilization criterion as the primary focus to check for non-uniform requests assignment conditions with any of the virtual machines in the data centers. The request(task) allocations for each VM for both current active VM(ESCE) algorithm and proposed VM Load balancer are tabulated in table 5.3 and plotted in the figure 5.3 for the case of the data center with the configuration of 5 VMs. It can be observed from the experimental results that initial VM ids are allocated with higher number of requests in case of ESCE algorithm because of the inconsistent VM allocation table data during allocation request processing. However it can be noted that proposed load balancer is able to allocate requests to the VMs uniformly.

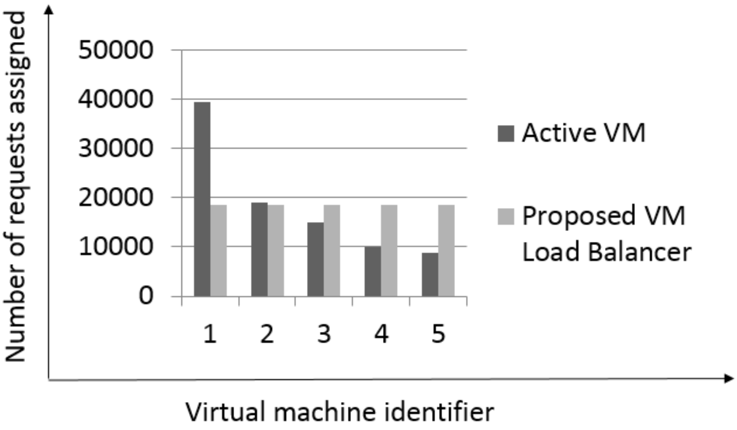


Figure 5.3: Comparison Results For 5 VMs In DC

Table 5.3: Comparison Results For 5 VMs Case

VM Id	Number of allocations (ESCE)	Number of allocations (Proposed LB)
0	39554	18502
1	19112	18507
2	14902	18503
3	10097	18504
4	8855	18504

Figure 5.4 plots the task assignments for 25 VMs for both the active VM(ESCE) algorithm and the proposed VM Load balancer. The task assignment numbers for ESCE and proposed VM load balancer are also tabulated in table 5.4. It can be observed from the experimental results that, for 25 VM case also ESCE algorithm does non-uniform allocation of requests to VMs and the proposed load balancer is able to distribute requests uniformly over 25 VMs.

It can be noted from the results that the ESCE(active VM) algorithm allocates tasks to VMs unevenly by over-allocating initial VM ids and under allocating remaining VMs because algorithm refers to the inconsistent VM table during VM allocation

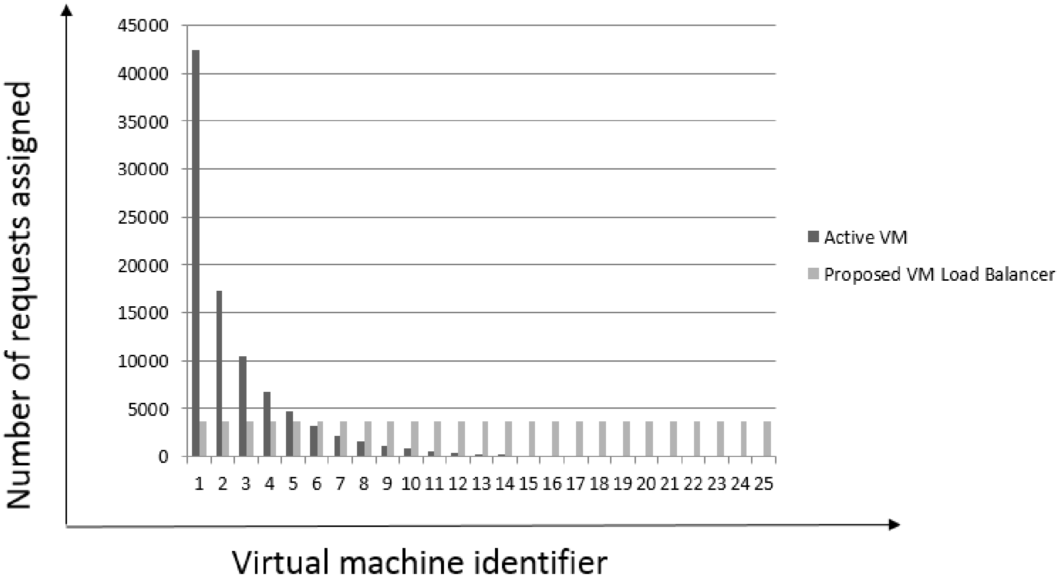


Figure 5.4: Comparison Results For 25 VMs In DC

for requests. The results also suggest that the proposed VM load balancer allocated the requests(tasks) to VMs evenly by overcoming the problem of ESCE(active VM) load balancer in all load conditions.

Table 5.4: Comparison Results For 25 VMs Case

VM Id	**Number of allocations (ESCE)**	**Number of allocations (Proposed LB)**
0	42374	3680
1	17324	3705
2	10477	3703
3	6783	3704
4	4683	3705
5	3249	3703
6	2229	3705
7	1613	3702
8	1191	3702
9	835	3702
10	609	3699
11	424	3703
12	299	3703
13	195	3700
14	120	3703
15	59	3702
16	36	3702
17	17	3703
18	8	3703
19	3	3700
20	3	3703
21	5	3702
22	1	3702
23	1	3702
24	2	3702

5.6 Summary

The chapter introduced the task scheduling in the cloud system and explained the ESCE load balancing algorithm and its problem with uniform VM task allocations during peak load situations. Then the chapter presented the research objective, proposed load balancing algorithm to overcome the problem with ESCE. The experimental set up used to evaluate the proposed load balancer for uniform VM allocations is described. It can be noted from the experimental results that the proposed load balancer is able to solve the VM over-allocation problem observed in ESCE load balancer.

In the next chapter, the existing solutions available to access GPU computing in the cloud are described. Then, the chapter discusses the existing research challenges and opportunities with load balancing in GPU enabled cloud. The chapter also describes the current hurdles to efficient utilization of GPU under the virtualization layer.

Chapter 6

Load balancing in GPU enabled Cloud: Challenges and Opportunities

In recent years graphical processing units(GPUs) are gaining importance for their massively parallel computing capability. The popularity is so much so that most of the commercial computing platforms(devices) sold in the market have a variant of GPU installed in it. Though GPUs were initially used only for graphics applications like gaming, display, etc., from past few years, GPUs are regarded as a high throughput parallel computing platforms suitable for general purpose computing such as high-performance computing(HPC), machine learning, medical imaging, inference generation and to support computing for smart cities and infrastructure development. The neural network (machine learning) extensions for deep learning and artificial intelligence built inside GPU hardware have only added to its growing popularity.

The ever increasing demand for GPUs has compelled cloud providers to enable GPU processing inside data centers to support hosting complex HPCs, real-time applications, and virtual desktop applications for its users. Today, almost all mainstream cloud providers like Amazon, Microsoft, Google have one, or many types of GPU enabled instances to offer. The GPU enabled VM instances can accelerate user applications significantly by offloading compute-intensive part of application logic onto the block of parallel threads in GPU. However, conventional techniques of virtualization do not hold good for GPUs because of the inherent differences in terms of architectures, control software, and distributed program/memory models. These differences make GPU provisioning in the virtualized environment and also GPU enabled VMs(gVMs) placement more complex and can cause inefficiency in resource utilization.

The reported work in this chapter examines the current infrastructure(software and hardware) available to support GPU inside cloud data centers and investigates existing challenges concerning efficient workload allocation(involving GPU computing) and provisioning techniques of virtualized GPUs. The reported work also examines the issues/challenges related to the effective utilization of GPU resources from applications when accessed from the virtualization layer.

6.1 Background study

6.1.1 GPUs and cloud datacenters

The typical block diagram of a GPU is shown in Figure 6.1(a). The GPUs consist of several thousands of single instruction multiple data(SIMD) cores packaged into streaming multiprocessors(SMs). SMs are responsible for executing GPU tasks. Each GPU has its own local memory called GDDRAM (graphics double data rate) and has two copy engines that can transfer data from GDDRAM to the main memory of the server in both directions simultaneously. The Giga Thread Engine is responsible for scheduling GPU threads onto streaming multiprocessors(SMs) for execution. The GPU tasks are usually submitted as a group of threads called blocks to GPU for execution. GPUs are usually fitted inside Video cards, and each video card can host multiple GPUs inside it. The GPUs are interconnected inside a video card using a PCIe(peripheral component interconnect express) switch, and the video card is connected to the host machine using PCIe connector. The PCIe connector is connected to system bus using which data flow between GPU and Main memory is carried out, as shown in figure 6.1(b).

The threads are distributed and scheduled inside SMs for execution. GPUs employ cooperative multitasking based on a leftover policy(Siavashi and Momtazpour, 2018) for the scheduling of thread blocks onto SMs. The steps followed in GPU task execution in an application is represented in figure 6.2.

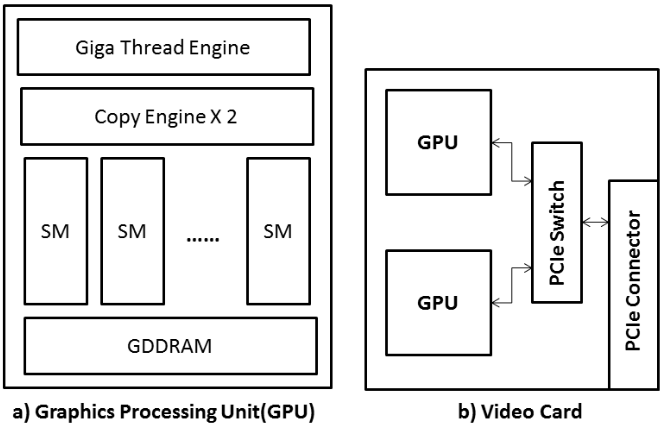


Figure 6.1: Block Diagram Of Typical GPU And Video Card

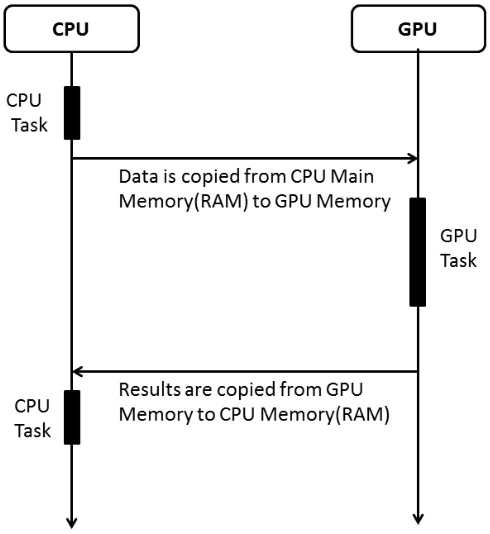

CPU
GPU
CPU Task
Data is copied from CPU Main Memory(RAM) to GPU Memory
GPU Task
Results are copied from GPU Memory to CPU Memory(RAM)
CPU Task

Figure 6.2: Typical Flow Of GPU Task Execution In An Application

Initially, the data required for GPU tasks are transferred from main memory (RAM) to GPU device memory (GDDRAM). Once the data transfer is complete, the GPU tasks are bundled into blocks of threads and launched onto GPU for execution. After the execution of all threads is completed, results are copied back to the main memory from GPU memory(GDDRAM).

6.1.2 GPU virtualization in cloud

Virtualization is employed with GPU to share the same GPU device with multiple user applications residing inside separate VMs. Virtualization helps to use resources efficiently by sharing unused computing power among different tasks.

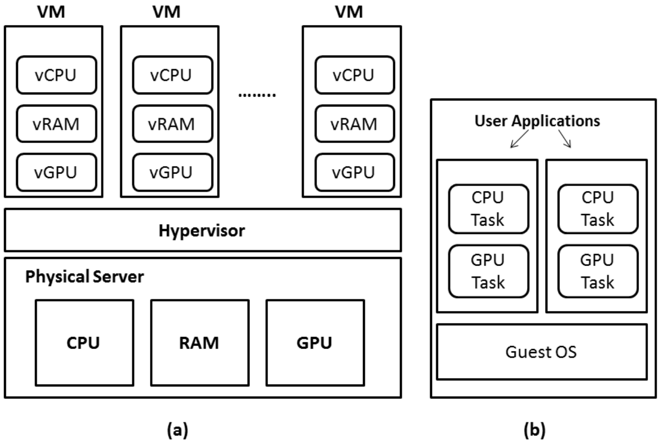


Figure 6.3: System View Of GPU Enabled Virtualized Server And User VM With GPU And CPU Tasks

The servers fitted with GPUs are offered by multiple vendors such as NVIDIA, AMD, Intel, etc. Due to the unavailability of GPU virtualization support from the vendor side in earlier days, user VMs were given direct pass through to GPU device(using vendor driver inside VM) for executing GPU tasks. However, in recent years, vendors have begun to offer virtualization support to GPUs for use inside cloud data centers such as NVIDIA Grid technology(NVidia, 2019). Figure 6.3(a) shows a virtualized view of a GPU enabled server inside the data center, and figure 6.3(b) depicts a typical user VM containing GPU tasks. There are multiple virtualization methods used for supporting GPU access inside cloud data centers with each having its pros and cons. Following is a brief discussion of some of the prominent GPU virtualization techniques.

i **API remoting:** The approach virtualizes GPU at API level where calls to GPU are intercepted at the API level in the host and are forwarded to a remote machine with GPU device for processing.

ii **Full virtualization:** The approach virtualizes GPUs at the device driver level where a GPU driver is installed inside user VM to communicate with virtual GPU. It will incur a penalty in performance. The hypervisor is responsible for scheduling virtualized GPUs(vGPUs).

iii **Paravirtualization:** The approach is similar to full virtualization. However, the guest OS driver is modified to avoid performance degradation to some extent.

The hypervisor is responsible for scheduling virtualized GPUs(vGPUs) in this approach too.

iv **Hardware-assisted virtualization:** The approach is supported by special hardware extensions provided by hardware vendors. These hardware extensions are responsible for VM to GPU mappings and parallel (multiplexing) executions of multiple VMs over GPU. The hypervisor may be involved to a minimal extent.

6.2 Research objective

The objective of the reported work in this chapter is to investigate current research gaps in GPU resource management policies and also study the challenges related to programming for GPUs in virtualized environments.

6.3 GPU resource provisioning techniques in cloud

The GPU provisioning to vGPU enabled VMs in cloud data centers is done at four different levels.

A Video card allocation

The video card allocation technique is responsible for allocating a GPU to a vGPU in the physical server. GPUs are housed in video cards, and the selection of video cards to be allocated for the given vGPU is taken care of by the video card allocation technique. There are three video card allocation techniques that are suggested(Siavashi and Momtazpour, 2018). The simple allocation policy follows first-fit policy wherein the first found video card with GPU that satisfies vGPU resource needs is selected. Breadth-first policy sorts available video cards in the ascending order of their GPU loads and returns lightly loaded video card and depth-first policy sorts the video card in descending order of GPU loads and returns the video card which is just enough to satisfy the vGPU resource needs.

B GPU allocation

The GPU allocation technique is responsible for allocating a GPU in a video card(with multi-GPUs) to a requested vGPU. Three allocation techniques, simple first Fit, breadth-first, and depth-first, employed in video card allocation are also used for GPU allocation.

C GPU enabled VM placement

VM placement policy for the VMs(with vGPU attached) is responsible for allocating a physical server in a data center. There are two types of placement policies proposed.

i **First fit Policy:** The technique used in VMware Horizon(VMware, 2019), where all the hosts are iterated until VM in question is accepted by a physical host considering VMs resource requirements.

ii **First fit increasing:** The technique(Siavashi and Momtazpour, 2018), first finds the bottleneck resource between each host-VM pairs. Then sorting of all VMs in ascending order is done based on their resource requirements and allocation to physical hosts is done using the first-fit policy.

D GPU provisioning

The GPU provisioning technique is responsible for defining the sharing policy of a physical GPU among multiple vGPUs. Some of the most commonly used GPU provisioning schemes (Siavashi and Momtazpour, 2018)(Hong et al., 2017) are listed below.

i **Space shared:** one vGPU occupies physical GPU till completion. The second vGPU is allocated only once the first vGPU is completed.

ii **Time shared:** The vGPUs share a physical GPU until co-executing vGPU does not exceed the total MIPS of a given GPU.

iii **FCFS:** First-come, first-serve policy allocates vGPUs in the order that they arrive.

iv **Round-robin:** Round-robin is similar to FCFS but assigns a fixed time slice to vGPU. This policy is also called a fair share scheme.

v **Priority-based:** Priority-based provisioning assigns a priority to every vGPU, and the provisioning logic executes vGPUs in the order of their priority.

vi **Fair queuing:** Fair queuing assigns a start tag to every vGPU and schedules them for execution in increasing order of the start tags. The accumulated usage time of a GPU is determined by the start tag value.

vii **Credit-based:** The algorithm periodically distributes credits to vGPUs, and each vGPU consumes credits when it is executed on the CPU for exploiting the physical GPU. The policy selects a vGPU with a positive credit value.

viii **Affinity-based:** The algorithm generates affinity scores for a vGPU to estimate the performance impact when it is allocated on a specific resource.

ix **SLA-based:** SLA (Service Level Agreement) is an agreement between a cloud service provider and a user about the quality of service(QoS) requirement and the price to be charged. The objective of the SLA based policy is to meet the SLA requirement while allocating GPU resources.

E Memory and PCIe bandwidth

The GPU device memory and PCIe bandwidth are two resources inside the GPU device that needs to be shared by co-running vGPUs. Each GPU can transfer data in two opposite directions simultaneously. PCIe bandwidth is provisioned on an equal share basis to all co-executing vGPUs. Device memory is one of the essential bottleneck resources inside GPU, which may have an impact on the performance of co-executing vGPUs.

6.4 Current challenges with GPU computing in the cloud

Virtualization is a crucial technology for the efficient utilization of server(PM) resources in cloud data centers. The virtualization solutions for CPU, memory, and network have attained sufficient maturity to be used in data centers for the benefit of both cloud providers and users. However, the same conventional technologies in CPU virtualization do not apply for the GPU virtualization because of inherent differences in architecture, programming models, and vendor-specific device driver software.

The reported work in this chapter investigates various research challenges/issues concerning efficient physical GPU utilization by current resource management and provisioning techniques in the cloud environment and also examines several problems with existing frameworks and technologies that limit user applications or VMs ability to exploit the real power of GPUs wrapped under the virtualization layer.

6.4.1 Challenges with GPU resource management in cloud

The section discusses various system-level issues that prevent efficient resource management of GPUs and GPU attached servers inside cloud data centers.

A GPU enabled VM migration

VM migration process is the re-placement of VM from the source physical host to a destination physical host in the data centers. The VM migrations are usually done for performance optimization, avoiding resource contentions, and to perform server consolidation during non-peak load situations inside cloud data centers. Though there are many proven algorithms(Choudhary et al., 2017) existing for the VM migration process, the VM migration becomes complicated when GPU enabled VM is to be migrated. The vGPU attached to the VM will have its process state and data inside GPU memory. The VM migration process has to wait till application finishes its GPU tasks or the ongoing GPU tasks need to be aborted in source machine and resumed in the destination machine. The extra computation or the extra delay caused by vGPU computation causes inefficiency in the VM migration process.

When GPUs in cloud servers are virtualized using hardware-assisted virtualization technology like NVIDIA Grid(NVidia, 2019), such a virtualization technology bypasses the core virtualization layer in the server for creating and managing virtual images of GPU. When vGPU has to be live migrated from such hardware-assisted virtualized GPU, retrieving GPU task states and restoring it on remote GPU is a complex task. There is a need to establish novel mechanisms to live migrate hardware-assisted virtualized GPU images from one GPU to another remote GPU efficiently.

B Power modeling of GPUs

Various vendors manufacture the GPUs, the components and architecture of GPUs are inherently different from one another. Because of their hardware composition, the power consumption and performance characteristics will vary from one another. Unlike CPUs, the power consumption and performance benchmarks(SPEC, 2011) for server scale GPUs are not available yet. The power-aware resource provisioning policies for GPUs have to rely only on the mathematical model for power consumption estimation. The mathematical equation(Siavashi and Momtazpour, 2018) for power consumption analysis is given by equation 6.1.

$$P(f, U) = a3.f.U + a2.f + a1.U + a0 \tag{6.1}$$

It is suggested that there is a linear correlation between power consumption and frequency and utilization of SMs in GPU. In equation 6.1, the frequency f and utilization U determine the power consumption approximation where a1,a2 and a3s are constants. Because the power consumption approximation is based on a mathematical model, the inherent physical composition of GPUs contributing to the power consumption factor is not considered. The factor may impact the efficiency of the resource provisioning algorithms.

C Power saving strategies involving GPUs

The power-saving in the data center is carried out by shutting down some of the servers with lower CPU utilization during non-peak hours of data center operations. However, when a physical server is equipped with one or more GPUs, the power saving strategy that selects a physical host has to consider additional parameters such as GPU utilization and state of GPU tasks to make host power-off decisions. The performance and power characteristics of both CPU and GPU can be considered for the selection of an underutilized host for power-off. The underutilized servers with relatively less power efficient GPUs can be prioritized for power-off to maximize the utilization of power-efficient GPUs.

D DC load aware GPU allocation policies

The current video card allocations, GPU allocation policies do not consider the DC load conditions. The GPU allocation schemes have to adapt to changing load conditions in data centers. The allocation techniques employed have to be aware of data center load (peak or non-peak conditions) state to make optimal decisions for allocations. During peak hours, the allocation policy can follow the breadth-first scheme or first fit technique for GPU allocations. During non-peak hours, the allocation technique can employ a depth-first strategy for the video card or GPU allocations. Such adaptive schemes can make GPU allocation models more responsive to the goal of power-saving during non-peak hours and also enables high performance during peak hours in the data centers.

E GPU memory pollution with fair-share policy for GPU provisioning

The fair share policy for GPU provisioning allocates a time slice of physical GPU to many vGPUs in round-robin fashion. GPU provisioning policies like fair share will have to make many hosts to device and device to host data transactions. If vGPU memory footprint(size) is not small, the vGPUs switching makes the processing very inefficient because of multiple data transfers involving the main memory and GPU device. Memory transfers are considered as major bottlenecks to achieve high throughput, and GPU device memory size limits the level of multitasking on GPU if data associated with GPU tasks is large. It is vital to consider data/memory transactions between host and GPU for GPU provisioning decisions and also GDDRAM memory size for scheduling GPU tasks on vGPUs.

6.4.2 Challenges with programming vGPUs

The section describes various user/programmer side issues that prevent exploiting the true power of GPUs by accessing them from the top of the virtualization layer.

A Target GPU generations

The GPUs available in the market possess different computing capabilities and support a varying degree of features because of the generation and type. If applications are ignorant of the target GPU generation, type or version, the program design may not be able to truly exploit the power of physical GPU. For instance, the device memory sizes, shared memory size, and the number of SIMD cores will vary between generations of GPU, and such details will impact the design of data structures and thread block sizes in algorithms. To overcome performance issues, the GPU allocation and provisioning techniques can prioritize higher generations of GPUs over lower generations for allocation decisions.

B Heterogeneity in GPUs and multiple frameworks

Multiple vendors manufacture data center-class GPUs in the market, and they possess different hardware extensions to support various features such as deep learning, AI solutions. Such heterogeneity in GPUs makes VM re-placement a complicated process. If VMs with GPU Tasks include demands for such additional features supported inside GPU, then such additional constraints need to be considered for VM placement. The GPGPU programs use different frameworks (CUDA, OpenCL, Vulcan, etc.) for accessing and computing on GPUs. Application using the CUDA framework for their GPU Tasks inside VMs can only be allocated to NVIDIA GPUs inside data centers. The resource management module for GPU provisioning needs to consider such hardware and software related constraints for making VM placements.

C Security aspects

Some cross-platform GPU frameworks like OpenCL delay the GPU specific logic (source code) compilations and GPU executable binary generation till run time if

target GPU device makes, version or generation is not known beforehand. Such VMs with delayed compilation process should be placed in secured environments to avoid cross VM attacks because application logic may be prone to leakage and may be used with malicious intent. The GPU allocation and provisioning policy have to consider such constraints for VM placements.
Some VM applications(Hong et al., 2017) may pose a denial of service attack for GPU by submitting a massive number of GPU tasks to underlying GPU devices and deny GPU resources for other co-allocated VMs. There is a need for a novel control mechanism in GPU virtualization layer to detect and control such VMs from overusing the GPU device.

The current research challenges and opportunities with GPU computing in virtualized environments discussed in this chapter can be addressed for designing an efficient GPU resource management framework in cloud data centers to improve performance and efficient GPU resource utilization.

6.5 Summary

The chapter describes the underlying architecture of GPU, current GPU virtualization software, and hardware infrastructure available in cloud data centers and then discusses various challenges investigated from GPU resource management and programming virtual GPUs(vGPU) perspectives to motivate further research in the load balancing techniques in GPU enabled cloud. Further research is needed to focus on solving some of the resource management issues discussed in this chapter to improve GPU enabled VM placement, GPU resource provisioning, and power/cost optimization algorithms.

In the next chapter, we summarize our research contributions and provide directions for future work.

Chapter 7

Conclusions and Future Work

To conclude this thesis, we first summarize the research contributions of the work reported in this thesis. Although the techniques and concepts presented in this thesis take a step forward in addressing some of these factors, several challenges remain to be addressed to improve the existing resource management techniques used for the cloud data centers in general. We list some of these extensions to our reported work in this thesis and provide some directions for future work.

7.1 Summary of contributions

The techniques proposed in this thesis for leveraging the contextual parameters to improve the load balancing decisions at multiple levels can be used as standalone concepts. These techniques can be thought of as of-the-shelf entities for enhancing already available and upcoming load balancing algorithms in the cloud. Following are the brief descriptions of the contributions through this thesis,

- The very first problem addressed in this thesis is to consider physical machine performance to power characteristics(power efficiency) and data center load characteristics for the VM placement optimization process. We have presented a framework for the collection and sharing of contextual information in data centers. Further, algorithms are presented for load context detection, VM placement, host consolidation and VM optimization tasks for power saving. It can be noted from the experimental results that our proposed context-aware VM placement optimization framework can save approximately 8-10% of power during lightly and heavily loaded cases and 2-6% during moderately loaded cases for synthetic workloads. With real-world workload traces, a power-saving of 1-3% is achieved by the proposed solution.

- The electricity prices vary with different geographical locations across the globe. We have addressed the problem of cost-saving in geographically dispersed data centers by considering electricity price and response time as parameters. We have presented a novel algorithm in cloud broker for load balancing user traffic among available geo-distributed data centers. The experimental results suggest that our proposed technique is able to distribute the load to cheaper data centers ranging from 3-6%(in case of experiment category E2 and E3) to about 50% (in case of experiment category E4) when there exists a cheaper data center(with lower electricity price) with relatively smaller estimated response times to closer data center.

- The peak hour performance is critical for data centers to meet the high demands of computing resources. The over-allocation and under-allocation of user tasks to VMs can cause performance degradation for cloud applications. We have investigated an existing ESCE(active VM) load balancing algorithm for uniform resource utilization and proposed a solution to solve the performance inefficiency in ESCE(active VM) load balancer during peak load situations. The experimental results attested that proposed VM load balancer allocated the tasks to available VMs evenly by overcoming the limitation of ESCE VM load balancer.

- GPU computing in cloud data centers is gaining momentum swiftly because of the massive parallel computing demands from applications like HPC, deep learning, and VDI applications. Though CPU virtualization techniques are matured enough, GPU virtualization and resource management is still a budding area of research. The VM placement techniques involving GPU allocation and provisioning needs further consideration for additional parameters and constraints. We have presented a summary of current GPU resource management techniques available in cloud data centers. Further, we have presented the remaining challenges/issues concerning GPU resource management and programming virtual GPUs(vGPU) to motivate further research in the related domain.

7.2 Directions for future work

Though the techniques and concepts presented in this thesis take a step forward in addressing some of the relevant factors in the domain of resource management in cloud computing, there are few extensions to our reported work possible to further improve the load balancing process in the cloud environment. In this section, we present some extensions and future directions as below.

- The co-located VMs on host machines can cause performance degradation due to conflicting resource demands. This can have an impact on the overall power consumption of the data center. There is a thorough investigation needed to understand the impact of interference and affinity of co-located VMs on the host resources from the perspective of power consumption. The affinity and interference can be modeled as an additional contextual parameter for VM placement decisions.

- The framework proposed for VM placement optimization has two modules, GWS(Global workload scheduler) at a master node and LCM(Local context manager) at each physical host to achieve context-aware VM placement optimization in the data centers. The proposed framework can be extended to hierarchical GWS to support a multi-DC setup or to support logical scaling of the resource management framework.

- VM placement optimization process can consider the user geo-locations in a geographically dispersed multi data center scenarios for VM placement decision to improve performance and overall power cost.

- The GPU enabled computing in the cloud is a relatively new area in cloud computing. The resource management techniques are yet to attain maturity to be used with GPUs wrapped in the virtualization layer efficiently. The research challenges reported in our reported work can be considered for further work.

On a closing note, the domain of cloud computing had a remarkable journey so far. The benefits of cloud computing have to lead more and more organizations to look up to cloud computing as a solution for deploying their applications ranging from a simple webserver to complex HPC applications. The contributions made in this thesis extends the journey by enabling contextual parameters like power efficiency, varying electricity price and load situations for load balancing at multiple levels in cloud data centers. Nonetheless, there are many evolving challenges for cloud computing researchers to address making the journey ahead one of discovery.

References

Abdelsamea, A., Hemayed, E., Eldeeb, H. and Elazhary, H. (2014). “Virtual Machine Consolidation Challenges: A Review.” *International Journal of Innovation and Applied Studies*, 8.

Ajit, M. and Vidya, G. (2013). “VM level load balancing in cloud environment.” *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT).* 1–5.

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. and Zaharia, M. (2009). “Above the Clouds: A Berkeley View of Cloud Computing.” *University of California at Berkeley UCB/EECS-2009-28, February*, 28.

Ashikur, R., Liu, X. and Kong, F. (2014). “A Survey on Geographic Load Balancing Based Data Center Power Management in the Smart Grid Environment.” *Communications Surveys Tutorials, IEEE*, 16, 214–233.

Beloglazov and Buyya (2012). “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.

Berkeley (2016). “United States Data Center Energy Usage Report.” https://datacenters.lbl.gov/resources/united-states-data-center-energy-usage (Accessed on Oct, 2019).

Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C. and Buyya, R. (2011). “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms." *Software Practice and Experience*, 41, 23–50.

Chiang, Y.-J., Ouyang, Y.-C. and Hsu, C.-H. (2014). "An Efficient Green Control Algorithm in Cloud Computing for Cost Optimization." *IEEE Transactions on Cloud Computing*, 3, 1–1.

Choudhary, A., Govil, M., Singh, G., Awasthi, L., Pilli, E. and Kapil, D. (2017). "A critical survey of live virtual machine migration techniques." *Journal of Cloud Computing*, 6, 23.

Dashti, S. and Rahmani, A. (2015). "Dynamic VMs placement for energy efficiency by PSO in cloud computing." *Journal of Experimental Theoretical Artificial Intelligence*, 1–16.

Dong, J.-k., Wang, H.-b., Li, Y.-y. and Cheng, S. (2014). "Virtual machine placement optimizing to improve network performance in cloud data centers." *The Journal of China Universities of Posts and Telecommunications*, 21, 62–70.

Fan, Weber, W.-D. and Barroso, L. A. (2007). "Power Provisioning for a Warehouse-sized Computer." *SIGARCH Comput. Archit. News*, 35(2), 13–23.

Farooqui, N., Barik, R., Lewis, B., Shpeisman, T. and Schwan, K. (2016). "Affinity-aware work-stealing for integrated CPU-GPU processors." *ACM SIGPLAN Notices*, 51, 1–2.

Geeta and Singh, C. (2014). "Load Balancing in Distributed System Using FCFS Algorithm with RBAC Concept and Priority Scheduling." *International Journal of Recent Development in Engineering and Technology*, 3(6), 33–39.

Goudarzi, H. and Pedram, M. (2013). "Geographical Load Balancing for Online Service Applications in Distributed Datacenters." *2013 IEEE Sixth International Conference on Cloud Computing*. 351–358.

Greenberg, A., R. Hamilton, J., Maltz, D. and Patel, P. (2009). "The Cost of a Cloud: Research Problems in Data Center Networks." *Computer Communication Review*, 39, 68–73.

Grosu, D. and Chronopoulos, A. T. (2004). "Algorithmic mechanism design for load balancing in distributed systems." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1), 77–84.

Gu, L., Zeng, D., Barnawi, A., Guo, S. and Stojmenovic, I. (2015). "Optimal Task Placement with QoS Constraints in Geo-Distributed Data Centers Using DVFS." *IEEE Transactions on Computers*, 64, 2049–2059.

Guo, Y. and Fang, Y. (2013). "Electricity Cost Saving Strategy in Data Centers by Using Energy Storage." *Parallel and Distributed Systems, IEEE Transactions on*, 24, 1149–1160.

Gupta, V., Gavrilovska, A., Schwan, K., Kharche, H., Tolia, N., Talwar, V. and Ranganathan, P. (2009). "GViM: GPU-accelerated virtual machines." *Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing, HPCVirt'09.*

Gupta, V., Schwan, K., Tolia, N., Talwar, V. and Ranganathan, P. (2011). "Pegasus: Coordinated Scheduling for Virtualized Accelerator-based Systems." *Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference*. USENIX-ATC'11, USENIX Association, Berkeley, CA, USA, 3–3.

Hamilton, J. (2019). "Overall Data Center Costs." https://perspectives.mvdirona.com/2010/09/overall-data-center-costs/ (Accessed on Oct, 2019).

Hong, C.-H., Spence, I. and Nikolopoulos, D. (2017). "GPU Virtualization and Scheduling Methods: A Comprehensive Survey." *ACM Computing Surveys*, 50, 1–37.

Jain, M. (2019). "How Perficient are Cloud Deployment Models for N/W Storage Needs?" https://www.konstantinfo.com/blog/cloud-deployment-model (Accessed on Sep. 04, 2019).

Kanagavelu, R., Lee, B., Le, N., Mingjie, L. and Aung, K. (2014). "Virtual machine placement with two-path traffic routing for reduced congestion in data center networks." *Computer Communications*, 53, 1–12.

Kusic, Kephart, J., Hanson, J., Kandasamy, N. and Jiang, G. (2008). “Power and Performance Management of Virtualized Computing Environments Via Lookahead Control.” volume 12. 3–12.

Le, T. N., Liang, J., Liu, Z., Sitaraman, R. K., Nair, J. and Choi, B. J. (2017). “Optimal Energy Procurement for Geo-distributed Data Centers in Multi-timescale Electricity Markets.” *SIGMETRICS Perform. Eval. Rev.*, 45(2), 58–63.

Li, W., Tordsson, J. and Elmroth, E. (2011). “Virtual machine placement for predictable and time-constrained peak loads.” *Lecture Notes in Computer Science*, 7150, 120–134.

Li, X., Qian, Z., Lu, S. and Wu, J. (2013). “Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center.” *Mathematical and Computer Modelling*, 58, 1222–1235.

Liu, C., Shen, C., Li, S. and Wang, S. (2014). “A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center.” *2014 IEEE 5th International Conference on Software Engineering and Service Science*. 272–275.

Mali, A. and Vidya, G. (2013). “VM level load balancing in cloud environment.” *2013 4th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2013*. 1–5.

Masdari, M., Nabavi, S. and Ahmadi, V. (2016). “An Overview of Virtual Machine Placement Schemes In Cloud Computing.” *Journal of Network and Computer Applications*, 66, 106–127.

Mell, P. and Grance, T. (2011). “The NIST definition of cloud computing.” *Communications of the ACM*, 53.

Menychtas, K., Shen, K. and Scott, M. L. (2014). “Disengaged Scheduling for Fair, Protected Access to Fast Computational Accelerators.” *SIGPLAN Not.*, 49(4), 301–316.

Mishra, R. K., Kumar, S. and Sreenu Naik, B. (2014). “Priority based Round-Robin service broker algorithm for Cloud-Analyst." *2014 IEEE International Advance Computing Conference (IACC)*. 878–881.

Mishra, S., Sahoo, B. and Parida, P. (2018). "Load Balancing in Cloud Computing: A big Picture." *Journal of King Saud University*.

Moreno, I., Yang, R., Xu, J. and Wo, T. (2013). "Improved energy-efficiency in cloud datacenters with interference-aware virtual machine placement." 1–8.

Mosa, A. and Paton, N. (2016). "Optimizing virtual machine placement for energy and SLA in clouds using utility functions." *Journal of Cloud Computing*, 5.

Nadeem, S. and Mohammed, F. (2015). "Static Load Balancing Algorithms In Cloud Computing: Challenges and Solutions." *International Journal of Scientific and Technology Research*, 4, 353–355.

Nadjaran Toosi, A., Qu, C., de Assuno, M. D. and Buyya, R. (2017). "Renewable-aware Geographical Load Balancing of Web Applications for Sustainable Data Centers." *Journal of Network and Computer Applications*, 83(C), 155–168.

NVidia (2019). "NVIDIA GRID Technology." https://www.nvidia.com/en-us/data-center/virtual-gpu-technology/(Accessed on Oct, 2019).

PlanetLab (2011). "PlanetLab Workload Traces." https://github.com/beloglazov/planetlab-workload-traces (Accessed on Sep, 2019).

Sayeedkhan, P. N., Nanded, V., S, M. S. B. and Nanded, V. (2014). "Virtual Machine Placement Based on Disk I/O Load in Cloud." *International Journal of Computer Science and Information Technologies*, 5.

Sengupta, D., Belapure, R. and Schwan, K. (2013). "Multi-tenancy on GPGPU-based Servers." *Proceedings of the 7th International Workshop on Virtualization Technologies in Distributed Computing*. VTDC '13, ACM, 3–10.

Siavashi, A. and Momtazpour, M. (2018). "GPUCloudSim: an extension of CloudSim for modeling and simulation of GPUs in cloud data centers." *The Journal of Supercomputing*, 2535–2561.

S.Jyothsna (2016). “Distributed Load Balancing in Cloud using Honey Bee optimization.” *International Journal of Emerging Trends Technology in Computer Science*, 5(6), 102–106.

SPEC (2011). “The SPECpower benchmark.” http://www.spec.org/power_ssj2008/ (Accessed on Sep, 2019).

Sudevalayam, S. and Kulkarni, P. (2011). “Affinity-Aware Modeling of CPU Usage for Provisioning Virtualized Applications.” *Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011*. 139–146.

Suzuki, Y., Kato, S., Yamada, H. and Kono, K. (2014). “GPUvm: Why Not Virtualizing GPUs at the Hypervisor?” *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*. USENIX ATC’14, USENIX Association, Berkeley, CA, USA, 109–120.

Toosi, A., Vanmechelen, K., Ramamohanarao, K. and Buyya, R. (2014). “Revenue Maximization with Optimal Capacity Control in Infrastructure as a Service Cloud Markets.” *IEEE Transactions on Cloud Computing*, 3, 1–1.

Toosi, A. N. and Buyya, R. (2015). “A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers.” *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. 186–194.

Tripathi, R., Vignesh, S., Tamarapalli, V., Chronopoulos, A. and Siar, H. (2017). “Non-cooperative power and latency aware load balancing in distributed data centers.” *Journal of Parallel and Distributed Computing*, 107.

VMware (2019). “Horizon—virtual desktop infrastructure.” https://www.vmware.com/products/horizon.html (Accessed on Oct, 2019).

Vmware (2019). “Virtualization.” https://www.vmware.com/in/solutions/virtualization.html (Accessed on Sep, 2019).

Wickremasinghe, B., Calheiros, R. and Buyya, R. (2010). “CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications.” 446–452.

Wikipedia (2017). “Electricity Price at Geographical locations.” https://en.wikipedia.org/wiki/Electricity_pricing/ (Accessed on Jan, 2017).

Xiao, Z., Jiang, J., Zhu, Y., Ming, Z., Zhong, S. and Cai, S. (2014). “A Solution of Dynamic VMs Placement Problem for Energy Consumption Optimization Based on Evolutionary Game Theory.” *Journal of Systems and Software*, 101.

Zhang, C., Yao, J., Qi, Z., Yu, M. and Guan, H. (2014). “vGASA: Adaptive Scheduling Algorithm of Virtualized GPU Resource in Cloud Gaming.” *IEEE Transactions on Parallel and Distributed Systems*, 25(11), 3036–3045.

Zhao, D.-M., Zhou, J. and Li, K. (2019). “An energy-aware algorithm for virtual machine placement in cloud computing.” *IEEE Access*, 07, 55659–55668.

List of publications

Journal publications

1. Ashwin Kumar Kulkarni and Annappa B. (2019).“Context aware VM placement optimization technique for heterogeneous IaaS cloud”, IEEE Access, Volume 7, Issue 1, pp 89702-89713. (DOI:10.1109/ACCESS.2019.2926291)

2. Ashwin Kumar Kulkarni and Annappa B. “GPU computing in cloud:Resource management and programming perspectives in virtualized environments”. (Under Review in Journal of King Saud University - Computer and Information Sciences, Elsevier).

Conference publications

1. Ashwin Kumar Kulkarni and Annappa B. (2017). “Cost aware service broker algorithm for load balancing geo-distrubuted data centers in cloud”. In IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, Kollam, India, pp 1-5. (DOI:10.1109/SPICES.2017.8091337).

2. Ashwin Kumar Kulkarni and Annappa B. (2015). “Load balancing strategy for optimal peak hour performance in cloud datacenters”. In IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems, Calicut, India. pp 1-5. (DOI:10.1109/SPICES.2015.7091496)

Brief Bio-Data

Mr.Ashwin Kumar
Research Scholar
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
P.O. Srinivasnagar
Mangalore - 575025
Phone: +91 9980156977
Email: ashwin.sony@gmail.com

Permanent Address

Ashwin Kumar
S/o Dr. D.V. Kulkarni
Noorandeshwar Colony
Moratagi - 586123
Sindagi (Tq.), Vijayapura (Dist.)
Karnataka, INDIA

Qualification

M. Tech. in Computer Science and Engineering, Visvesvaraya Technological University, Belgaum, Karnataka, 2011.

B. E. Computer Science and Engineering, Visvesvaraya Technological University, Belgaum, Karnataka, 2004.