Energy Efficient Resource Scheduling Algorithms for Cloud Computing

A Thesis submitted

for the award of degree of

Doctor of Philosophy

By:

Sudhir Goyal

(951003012)

under the guidance of:

Dr. Seema Bawa, Professor, CSED, Thapar University, Patiala

Dr. Bhupinder Singh, Senior Program Manager, Microsoft India (R&D) Ltd., Hyderabad

Computer Science and Engineering Department

Thapar University, Patiala – 147004, INDIA

March 2016

Dedication

To my mother, late Ms. Kamlesh Rani,

and my father Sh. SatPal Goyal

Contents

List of Figures ---------- vi

List of Tables ---------- viii

Certificate ---------- ix

Acknowledgement ---------- x

Abstract ---------- xii

1 **Introduction** ---------- 1

1.1 Cloud Computing: An Overview ---------- 2

1.1.1 Evolution of Cloud Computing ---------- 3

1.1.2 Types of Cloud ---------- 4

1.1.3 Benefits of Cloud Computing ---------- 7

1.2 Need of Energy efficient Computing in Clouds ---------- 8

1.3 Energy Efficient Cloud Computing Techniques ---------- 10

1.3.1 VM Management and Scheduling ---------- 11

1.3.1.1 Thermal Aware Scheduling ---------- 11

1.3.1.2 Energy efficient Resource Scheduling ---------- 12

1.3.1.3 Energy Aware Data Storage and Placement Strategies 12

1.3.2 Energy Efficient Network ---------- 13

1.3.2.1 Energy Efficient Network Devices ---------- 13

1.3.2.2 Traffic Engineering and Energy Efficient Routing --- 13

1.3.2.3 Energy Efficient Network Architecture ---------- 14

1.3.3 Advance Data Center Design ---------- 14

1.3.3.1 Energy Efficient Hardware ---------- 14

1.3.3.2 Renewable Energy Sources ---------- 15

1.3.3.3 Data Center Locations ---------- 16

1.3.4 Energy Aware User Contracts ---------- 16

1.3.4.1 Energy Aware SLA---------- 16

1.4 Thesis Organization ---------- 16

1.5 Thesis Contribution ---------- 18

2 **Literature Review ----------** 20

2.1 Energy Efficient Resource Scheduling in Cloud Computing ---------- 21

2.1.1 Consolidation based Heuristic Approaches ---------- 21

2.1.2 Greedy Heuristic Approaches ---------- 23

2.1.3 Bio-Inspired Heuristic Approaches ---------- 24

2.1.4 Power-Aware Heuristic Approaches ---------- 25

2.1.5 Miscellaneous Heuristic Approaches ---------- 26

2.2 Comparative Analysis of Energy Efficient Resource Scheduling Algorithms---------- 30

2.3 SLA Based Energy Aware Scheduling in Cloud Computing ---------- 32

2.3.1 Power-Aware Heuristic Approaches ---------- 32

2.3.2 Bio-Inspired Heuristic Approaches ---------- 33

2.3.3 Miscellaneous Heuristic Approaches ---------- 33

2.4 Comparative Analysis of Green SLA Scheduling Algorithms ---------- 35

2.5 Motivation ---------- 38

2.6 Problem Formulation and Objectives ---------- 39

2.7 Summary ---------- 40

3 **Proposed Energy-efficient Cloud Resource Scheduling (ECRS) Algorithm** 41

3.1 Formalization of the Proposed Energy-efficient Resources Scheduling Algorithm ---------- 42

3.2 Proposed Energy Efficient Cloud Service Framework: ACA-Cloud------- 43

3.2.1 Energy Model ---------- 43

3.2.2 Reservation Model ---------- 44

3.2.3 Prediction Model ---------- 46

3.3 Proposed Energy-efficient Cloud Resource Scheduling (ECRS) Algorithm 47

3.3.1 Initial VM Placement ---------- 47

3.3.2 Optimization the Current Workload ---------- 48

3.4 Summary ---------- 50

4 **Proposed Green SLA aware Cloud Resource Reservation (GSLACRR) Algorithm**---------- 51

4.1 Introduction ---------- 52

4.2 Proposed Green Service Level Agreement (GSLA) based Resource Management ---------- ---------- 52
4.2.1 Energy Based Resource Provisioning Policy (EBRPP): A Negotiation Approach ---------- 54
4.2.2 GSLA Workflow ---------- 56
4.2 Proposed GSLA aware Cloud Resource Reservation (GSLACRR) Algorithm ---------- 56
4.3 Advantages of the Proposed (GSLACRR) Algorithm ---------- 58
4.4 Summary ---------- 59
5 **Design and Implementation of Proposed Algorithms ----------** 60
5.1 Architecture of ACA-Cloud ---------- 61
5.1.1 Interface Services ---------- 61
5.1.2 Account Manager ---------- 61
5.1.3 Cloud Cluster Controller ---------- 61
5.1.4 Host Controller ---------- 64
5.2 Design Details of ACA-Cloud ---------- 65
5.2.1 Use Case Diagrams ---------- 65
5.2.2 Class Diagram ---------- 67
5.2.3 Activity Diagram ---------- 69
5.2.3.1 ECRS Algorithm's VM Handler for Initial VM Placement ---------- 69
5.2.3.2 ECRS Algorithm's Load Balancer for Optimizing Current Workload ---------- 69
5.2.4 Sequence Diagram ---------- 70
5.2.4.1 GSLACRR Successful User Negotiation ---------- 70
5.2.4.2 GSLACRR Renegotiation for User's Requirement -- 71
5.2.5 State Diagram ---------- 71
5.3 Implementation Details ---------- 73
5.3.1 Experiment Setup ---------- 74
5.3.2 User Interface ---------- 76
5.4 Summary ---------- 78

6. Test Results and Analysis ---------- 79
6.1 Performance Evaluation Criteria ---------- 80
6.1.1 Performance Metrics ---------- 80
6.1.2 Workload Data ---------- 80
6.2 Performance Evaluation of Proposed ECRS Algorithm ---------- 81
6.2.1 Test Case 1: Low Heterogeneity Performance Evaluation ------ 81
6.2.2 Test Case 2: High Heterogeneity Performance Evaluation ------ 83
6.2.3 Statistical Analysis of Results ---------- 83
6.3 Performance Evaluation of Proposed GSLACRR Algorithm ---------- 87
6.3.1 Analysis of Results ---------- 88
6.3.2 Effect of the number of VM Requests ---------- 89
6.3.3 Statistical Analysis of Results ---------- 92
6.4 Effects of the Number of Resources ---------- 92
6.5 Summary ---------- 94
7. Conclusion and Future Scope---------- 95
7.1 Conclusion ---------- 96
7.2 Future Scope ---------- 98
References 101
Publications 118
Appendix A: Implementation Details 119

List of Figures

1.1	Simple view of Cloud Computing----------	2
1.2	Convergence of various technologies advancement leading to the advent of cloud computing ----------	4
1.3	Cloud computing comprised a collection of technologies, solutions and models ----------	5
1.4	The cloud computing stack ----------	6
1.5	Energy distribution in a data centre ----------	8
1.6	Average CPU utilization of more than 5,000 servers for a six-month period----------	9
1.7	A taxonomy of Green Cloud Framework----------	11
2.1	Energy efficient resource scheduling taxonomy ----------	22
3.1	High level architecture of proposed energy efficient cloud service framework ----------	45
3.2	VM migration band for a host ----------	49
4.1	Proposed GSLA-based resource management ----------	53
4.2	Proposed Green Service Level Agreement----------	54
5.1	Architecture of proposed ACA-Cloud framework ----------	62
5.2	Communication protocol VMH-GE and LB-GE ----------	64
5.3	Use case diagram for user authentication and management ----------	66
5.4	Use Case Diagram for resource provision----------	66
5.5	Class diagram for ACA- Cloud ----------	67
5.6	GlobalExecutor class of ACA-Cloud----------	67
5.7	Host class of ACA-Cloud ----------	68
5.8	Activity diagram of VMHandler module of ECRS algorithm ----------	69
5.9	Activity diagram of LoadBalancer module of ECRS algorithm ---------	70
5.10	Sequence Diagram of successful negotiation process of GSLACRR----	71
5.11	Sequence diagram of GSLACRR renegotiation for user's requirement --	72
5.12	State diagram of user resource requests ----------	73
5.13	Authenticated user resource request screen ----------	75
5.13	GSLA negotiation screen----------	76
5.15	ACA-Cloud administration panel screen ----------	77

5.16	Development environment and output generated by GlobalExecutor for ACA-Cloud ----------	77
5.17	Snapshot of HostSupervisorGlobal for ACA-Cloud ----------	78
6.1	Accepted VM requests with low resource heterogeneity ----------	84
6.2	Average power consumption of a centre with low resource heterogeneity	85
6.3	Accepted VM requests with high resource heterogeneity ----------	86
6.4	Average power consumption of a centre with high resource heterogeneity ----------	86
6.5	Accepted VM requests ----------	89
6.6	Rejected VM requests ----------	90
6.7	Average power consumption of a centre----------	90
6.8	Number of PMs used with respect to VMs ----------	91
6.9 (a)	The experimental results with different types of workload (a) Test 1 with low number of VMs request ----------	91
6.9 (b)	The experimental results with different types of workload (b) Test 2 with high number of VMs request----------	92
6.10	Effect of number of resources/servers on the power consumption in a data center ----------	94

List of Tables

2.1	Energy efficient scheduling algorithms for cloud computing----------	27
2.2	Comparative analysis of energy efficient scheduling algorithms and policies ----------	31
2.3	Comparative analysis of green SLA Scheduling algorithms----------	36
3.1	Acronyms used in the algorithms----------	47
4.1	Acronyms used in the GSLACRR algorithm ----------	57
5.1	Implementation Technology used by the ACA-Cloud ----------	74
5.2	Experimental setup ----------	75
6.1	VMs deployed ----------	81
6.2	T-test: paired two sample for means----------	87
6.3	T-test: paired two sample for means with respect to MEC metric-------	93
6.4	T-test: paired two sample for means with respect to AVM metric--------	93

CERTIFICATE

I hereby certify that the work which is being presented in this thesis entitled "***Energy Efficient Resource Scheduling Algorithms for Cloud Computing***", for the award of degree of "***Doctor of Philosophy***" submitted to the Department of Computer Science and Engineering, Thapar University, Patiala, is an authentic record of my own work, carried out under the supervision of Dr. Seema Bawa and Dr. Bhupinder Singh. It refers to the work done by other researchers which are duly listed in the reference section.

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

Sudhir Goyal

(Sudhir Goyal)
Reg. No. 951003012

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

Seema Bawa
22/03/2016

(Dr. Seema Bawa)
Professor,
Computer Science & Engineering Department
Thapar University, Patiala 147004, Punjab, INDIA

(Dr. Bhupinder Singh)
Senior Program Manager,
Microsoft India (R&D) Ltd.,
Hyderabad, INDIA

Acknowledgement

First of all, I express my gratitude to that Almighty, Who blessed me with the zeal and enthusiasm to complete this work successfully.

My endeavors in this thesis have been made possible by several remarkable people who helped and supported me, and most importantly prayed for me. I would like to express my sincere appreciation and gratitude to them all in my humble acknowledgment.

Firstly, I would like to express my sincere thanks to my supervisors Prof. Seema Bawa, and Dr. Bhupinder Singh for their supervision, timely advice, and comments from the very early stage of this research which kept me stayed on the right track. It has been a great pleasure and a privilege to have the opportunity to work under their supervisions.

I also wish to extend my gratitude to the members of the PhD committee: Prof. Rajesh Khana, Dr. Inderveer Chana, and Dr. V.P Singh for their encouragement and insightful comments in relation to my research.I would like to many thanks all faculty and staff members of Thapar University, who have been kind enough to advise and help in their respective roles.

This research has profited from the friendship, advice, encouragement and support of a fine set of people. I am indebted to my dear friend Naveen Gupta, Panjab University, for continual counsel, willingness to support and helped me, with a smiling face, as and when required. I would also like to special thanks to Khushneet Jindal, Sunil Singla, Manmohan Chhibber, Ajay Kakkar, Ashish Aggarwal, Anuj Gupta, Arun Bansal for their friendship and help during my PhD. I have had a great time with them.

Last but not least, I am heartily thankful to all my family members for their love, encouragement and endless support at all times. I can never repay the greatness of my father's love for me. He has sacrificed everything for his sons, my younger brother and me. This thesis is a result of his endless love. I remain indebted to my divine mother, late Ms. Kamlesh Rani,who always believed in my capabilities. Thanks to my loving brother, Sandeep Goyal, and my sisters-in-law Meenu Goyal, their daughters for being supportive and caring. Finally, I thank to my wife Ritu and daughter Avani, for their support, love, inspiration,

patience, and for making my life filled with joy and happiness. They never give up on me, but love me from the bottom of their heart.

Sudhir Goyal

Abstract

Cloud Computing presents an exciting new horizon for the Information Technology (IT) industry. It provides a cost-effective solution, as it allows hosting of storage, computational and supported network services on a shared infrastructure of physical servers. It offers utility oriented IT services to the users worldwide. Cloud computing empowers companies to host engineering, scientific, and business applications, and makes them accessible across the world. However, to accommodate the increasing trend of online computing applications and the ever growing massive amount of data, the data centers are also continually expanding in size. This means a huge consumption of electrical energy that ultimately results in high operational costs and emission of green house gases into the environment. Therefore, to curb this unsustainable increase in energy consumption, research on "energy efficient computing" becomes a critical need and a roadblock of great magnitude with respect to energy efficient utilization of resources while preserving the desired users' Quality of Service (QoS) standards.

This thesis focuses on cloud resource management with special attention to energy efficient resource scheduling and Green Service Level Agreements (GSLAs). Initially, it compares, analyses and reports on the existing energy aware resource scheduling frameworks and heuristics on the basis of various aspects. From the literature survey, it is apparent that there is a great energy saving potential with respect to the system operations and workload specificities. This in particular holds for small and medium sized datacenters which can't afford expensive hardware and renewable energy sources to save energy. To address this challenge, this thesis presents novel techniques, models, and algorithms for the cloud environment.

To achieve the goal of improving resource utilization and reducing energy consumption, the Energy efficient Cloud Resource Scheduling (ECRS) algorithm has been designed. It gathers information such as host utilization level, power consumption, number of VMs and their state etc. regarding available resources. Using this information along with an estimate of resources required for future requests, the energy efficiency of cloud compute cluster is assessed. This assessment is used to manage resources energy efficiently. Further, to involve the users in eco-system cloud services, a Green Service Level Agreement (GSLA) aware

Cloud Resource Reservation (GSLACRR) algorithm has been proposed. It is an endeavor to incline the users towards sustainable computing through user negotiation strategies. It ultimately results in cost benefits for users as well as service providers and helps to minimize the energy consumption. To address the various cloud resource management challenges such as performance, energy efficiency etc., an energy efficient cloud framework, named ACA-Cloud has been proposed, designed, developed and tested, and further used to demonstrate the applicability of the proposed algorithms.

The proposed energy efficient cloud framework, ACA-Cloud, has four layer architecture, with Host Controller (HC) as the base layer, providing actual resources to a user. The second layer is Cloud Cluster Controller (CCC) which is responsible for monitoring the status of all the physical machines/hosts and making appropriate resource management decisions in response to the current workload and incoming user requests. The next layer is Account Manager, which renders user authentication management to the framework to handle different account types and authentication. The top most layer is the Web Portal Interface, through which users can submit their requests for cloud services.

This proposed framework has been installed at Thapar University, Patiala. The experimental results reveal the competitive performance and usage of the proposed algorithms implemented on this framework.

Finally, the proposed algorithms are compared with the existing one to validate the outcome. The results show that the scheduling algorithms successfully and collectively address the issues of energy efficiency and performance to establish an efficient Cloud.

Chapter 1

Introduction

Nowadays, the increased deployment of servers to facilitate High Performance Computing (HPC) for scientific, engineering and web applications, lead to huge consumption of energy. Cloud computing is a cost-effective solution, as it allows the hosting of storage, computational and supported network services on a shared infrastructure of physical servers. It offers utility oriented IT services to users worldwide. It enables the hosting of a large capacity of consumer, scientific, and business applications, and makes them accessible across the world.

This chapter foremost discusses cloud computing fundamentals, its evolution, service models and usage benefits as a powerful distributed computing paradigm. The popularity of cloud computing has carved a niche for itself as it is a cost-effective solution for the growing demand of computing resources. It has become a part of every computing paradigm due its success and usability. However, the drastically increased demand of Cloud infrastructure has led to consumption of huge amounts of energy (power and heat). It contributes to high operational costs and carbon footprints to the environment. So there is a need of energy efficient resource management of data centers in order to save energy and reduce the operational overhead cost. This chapter presents requirements, issues and challenges of energy efficient computing. Further, it highlights the current energy efficient cloud computing mechanisms and technologies available to address these challenges. It culminates with a discussion of thesis organization and its contribution.

1.1 Cloud Computing: An Overview

Cloud computing can generally be understood as delivered hosted services over the Internet. These hosted services are commoditized and delivered similar to other utilities such as power, gas, water, telephone etc. Cloud computing can be usually seen as a standardized cloud shaped diagram (Figure 1.1) that is a depicted metaphor for the network of resources (e.g. server, storage, networks, software and application services) which render some services that are invisible to the users, as if obscured by a cloud.

"Cloud computing is a type of parallel and distributed system consisting of a collection of interconnected and virtualized machines that are dynamically provisioned and presented as one or more unified computing resources based upon service level agreement " [1].

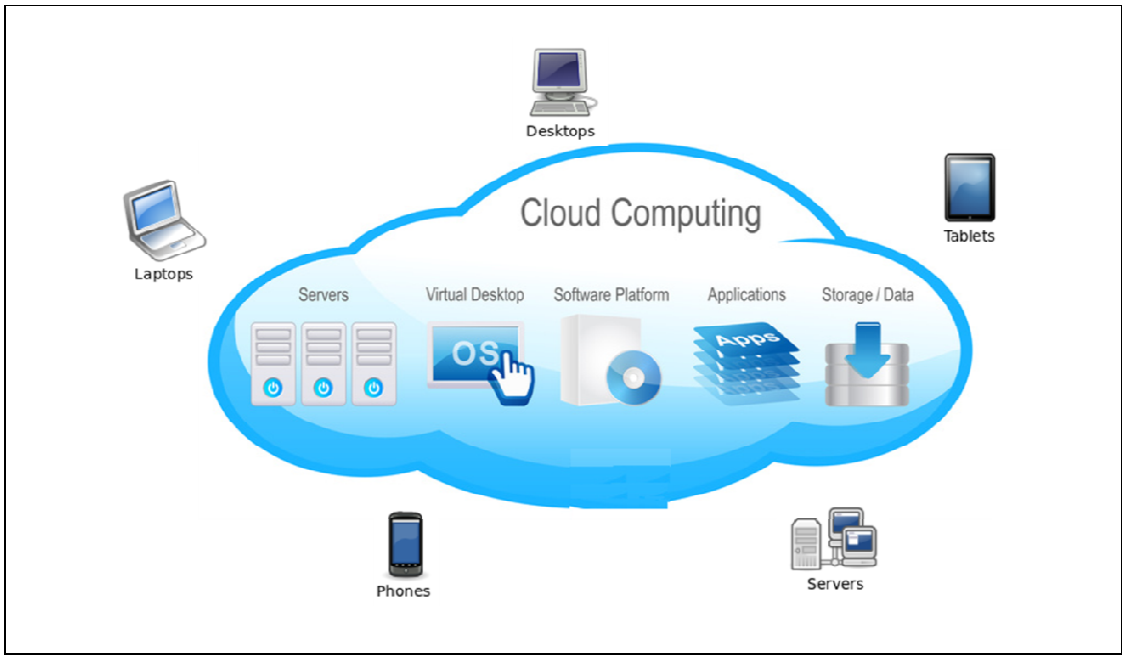


Figure 1.1: Simple view of Cloud Computing

It propels us into the next generation of data centers that are designed with a network of virtual resources (hardware, database, user interface, application logic) which offer services under a pay-as-you-go model with the desired Quality of Service (QoS) [2]. A user having access to the Internet can make use of cloud services anywhere, irrespective of the hosted services' location. It allows different sections of users to focus on their core business affairs rather than being hindered by IT obstacles such as capacity planning, procurement, management, software licensing fee etc. Innovative developers can focus on their business logic rather than the capital

outlay in hardware, software and human expense to operate [3]. An end user can access their data and documents from the cloud storage on any device (phone, tablet, laptop, desktop, servers etc.) that can connect to it. Large and small enterprise can translate their business ideas to results without upfront costs of IT infrastructure. As per the demand of storage and processing workload, additional resources can be instantly scaled and released [4].

The next section discusses the evolution of cloud computing.

1.1.1 Evolution of Cloud Computing

The evolution of cloud computing started around 1950 with the launch of mainframe computers. Figure 1.2 shows how technologies have significantly advanced and refined to the advent of cloud computing. Mainframe computers are the first examples of a large computation power that provides CPU computing on time shared basis [5] to thin clients or dumb terminals. It offers massive data processing, better utilization by eliminating inactivity period, and a highly reliable and greater return on investment.

With the invention and development of ARPANET (Advanced Research Project Agency Network) in 1969 Distributed Computing System (DCS) came into existence [6]. IN DCS, a number of minicomputers/servers are interconnected through networks and these provide a pool of processing power, data storage and fault tolerance. A large problem is divided into sub-problems and distributed among the interconnected servers. Each server solves the problem individually and communicates with each other over the network to solve/perform it as a single entity problem. The ultimate goal of DCS is to enhance the performance by connecting the distributed IT resources in a cost-effective, transparent and reliable manner.

With the advancement of network, Internet became an established technology in late 1985 [7]. Internet became popular among the researchers and developers for daily computer communication.

The large scale development of IT infrastructure and the innovation in new and scalable applications led to the emerging of Grid computing in 1990s [8]. Grid computing connects large geographically scatted computers, allows them to work as a single resource to solve large-scale computing and data intensive computing applications [9]. In Grid computing, the user can use computational resources just like other utilities such as water, power, gas etc.

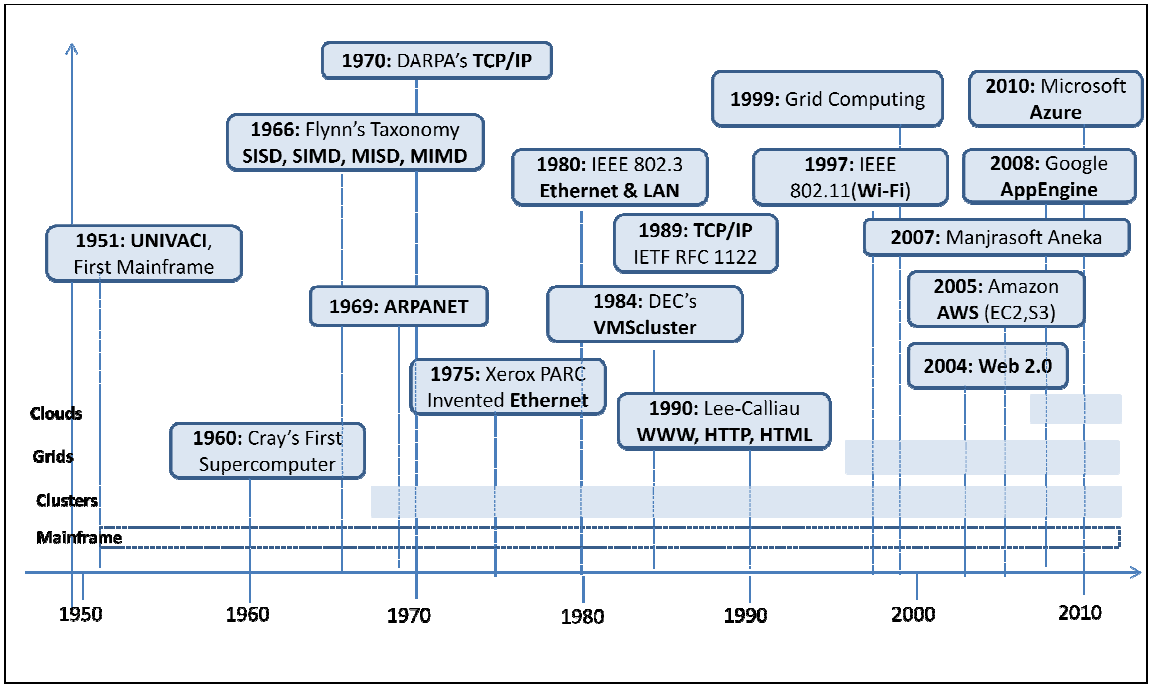


Figure 1.2: Convergence of various technologies advancement leading to the advent of cloud computing [4]

Grid computing is often considered as a root of cloud computing. In reality, cloud computing emerged as a result of maturity and comprised a collection of technologies, solutions and models such as distributed computing, hardware, Internet Technology and System Management as shown in Figure 1.3 [10]. In short, the advancement in technologies over a span of time lead to the advent of cloud computing and this new paradigm is a result of an evolution rather than a revolution.

1.1.2 Types of Clouds

Cloud computing is categorized based on the service model of providers and the deployment strategies to meet specific needs of the users. Different types of services models provide users different level of control, flexibility and management (Figure 1.4). There are three main service models of cloud computing [11].

I. Infrastructure as a Service (IaaS)

A cloud IaaS is a bottom layer of the cloud computing stack. It provides access to network, storage and computational resources in virtualized manner on basis of demand. Virtualized

servers are run based on a user's choice of operating system, customizable software stack, and demanded virtualized storage disks. Users have privileges to install software, configure different user access permissions and firewall rules. This layer provides the highest level of flexibility and management control over virtualized servers. Some of the leading cloud IaaS providers are Amazon EC2 [12], Amazon S3 [13], Microsoft [14], IBM [15], Google [16], Rackspace [17] etc.

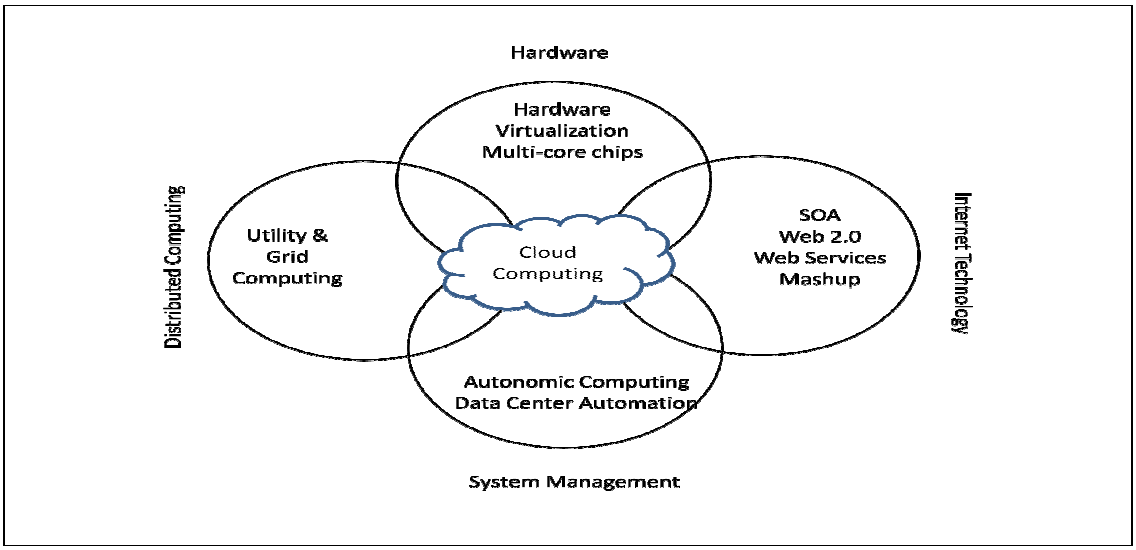


Figure 1.3: Cloud computing comprised a collection of technologies, solutions and models [10]

II. Platform as a Service (PaaS)

PaaS allows organizations or developers to design, deploy and manage their applications. They don't need to worry about the infrastructure capacity planning, software maintenance, security patches and other undesired activities such as firewall configuration etc. In addition, programming languages (such as Python, Java, C, C++ etc.), IDE (Eclipse, Netbeans, .net), and specialized services (payment gateway, mail service, instant messaging service, data access, authentication, image processing, SDKs for Android and Apple mobiles etc.) are offered to build new applications. Some of the leading cloud PaaS providers are AWS Toolkit for Eclipse [18], AWS CloudFormation [19], Google App Engine [20], Microsoft Azure [21], Salesforce.com [22] etc.

III. Software as a Service (SaaS)

SaaS is an end user application and lies at the top of the cloud computing stack. The user needs to know only how to use the finally delivered application such as online word processing, spreadsheet, email service etc., without the hindrance of server management, operating system, and software deployment. We can think of this as a desktop application's migration to online web application. Some of the examples are Google Gmail and Google docs [23], Microsoft office 365 [24], Dropbox [25], Salesforce.com's online Customer Relationship Management (CRM) [26], BlueJeans Network [27], Vidtel for Video conferencing [28]etc.

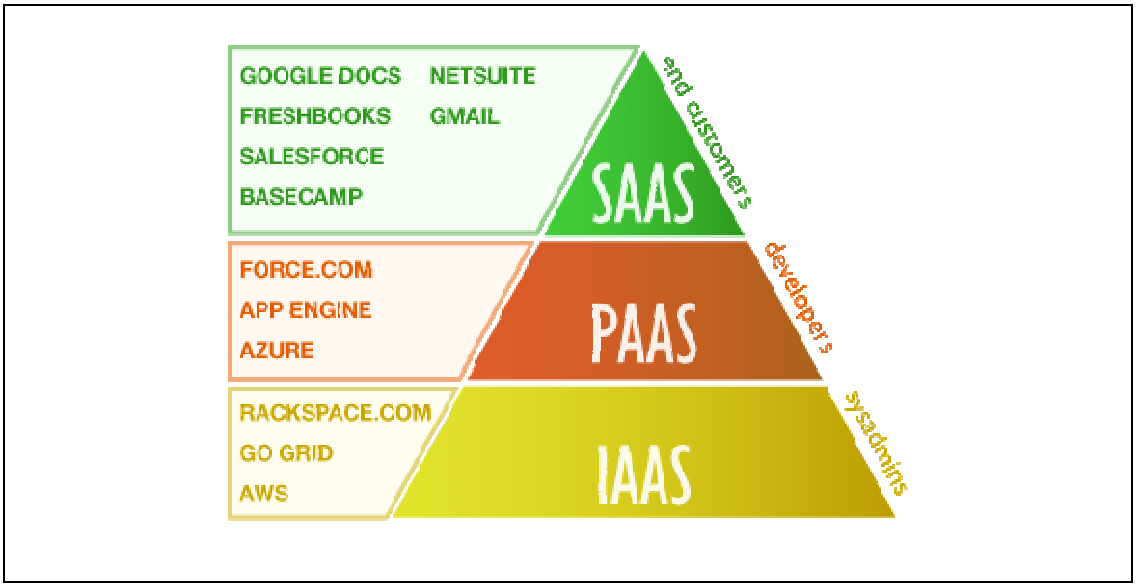


Figure 1.4: The cloud computing stack [11]

Apart from these industry standard cloud deployment models, there are more granular classifications by David Linthicum [29]

i. Storage-as-a-Service
ii. Database-as-a-Service
iii. Information-as-a-Service
iv. Process-as-a-Service
v. Application-as-a-Service
vi. Platform-as-a-Service

vii. Integration-as-a-Service

viii. Security-as-a-Service

ix. Management/Governance-as-a-Service

x. Testing-as-a-Service

1.1.3 Benefits of Cloud Computing

Self Service: Users can use cloud services without the assistance of any human operator. They just need to open the service provider's cloud web portal, submit the appropriate resource/software/application request, make Service Level Agreement, pay and use the services [30].

Pay as per usage: Users need to pay only for the necessary services based on the usage time (hourly, weekly, quarterly, or yearly). Client service metering is calculated based on the amount of storage, processing power, bandwidth and so forth. All this is done with transparently as the used resources are accounted, audited, and billed accordingly.

Elasticity: Cloud resources can be rapidly provisioned and scaled out based on the basis of increasing or decreasing application load. Large IaaS cloud providers (such as AWS) provide users with the illusion of infinite resources. Additional resources can be allocated and released automatically at any time and in any quantity, as per the agreement or on the basis. of requests.

Reliability: The ability to provide fault tolerance and failover make cloud computing services highly reliable. It achieves even more reliability than service that lies inside an organization [31].

Work from anywhere: You can access the cloud services all over the world provided you have access to the Internet. A knowledgeable worker wants to work as per his/her convenience to strike a balance between work, life, and productivity. A study shows that 42% of adults are ready to leave a part of their salary with the permission to work on telecommute.

Eco friendly: Business houses, through cloud computing use a part of the server space instead of using a wholesome IT infrastructure. Getting IT services through cloud computing consumes 30% less energy and results in lesser carbon footprint as compared to on-site servers [32].

Increase productivity and collaboration: Employees can sit at different geographical locations and share work simultaneously, synchronise, follow colleagues, and make critical updates in

real time. The Frost and Sullivan survey found that companies which worked in shared and joint working technology had a 400% return on investment [33].

Reduce capital cost: Organizations are looking for IT services for their operations and technical services for their clients that reduce their upfront cost on procurement, management and maintenance of hardware and software at largely. There is no need to spend on capacity planning, on buying expensive hardware and software licensing, and to invest in technical manpower to administer IT infrastructure.

1.2 Need of Energy efficient Computing in Clouds

The demand for cloud hosted applications among IT/non-IT companies and the end users is drastically increasing. It yields an expeditious proliferation of data centers. As per reports of Emerson 2011 [34], there were 509,000 data centers around the world. These data center consume a significant amount of energy and emit greenhouse gases. The Information and Communications Technology (ICT) industry itself contributes to approximately 2% of the global CO2 discharge, which is equal to the aviation industry [35]. As per a projection by the ClimateGroup [36], the CO2 emissions from data centers will be more than double in 2020 as compared to 2007.

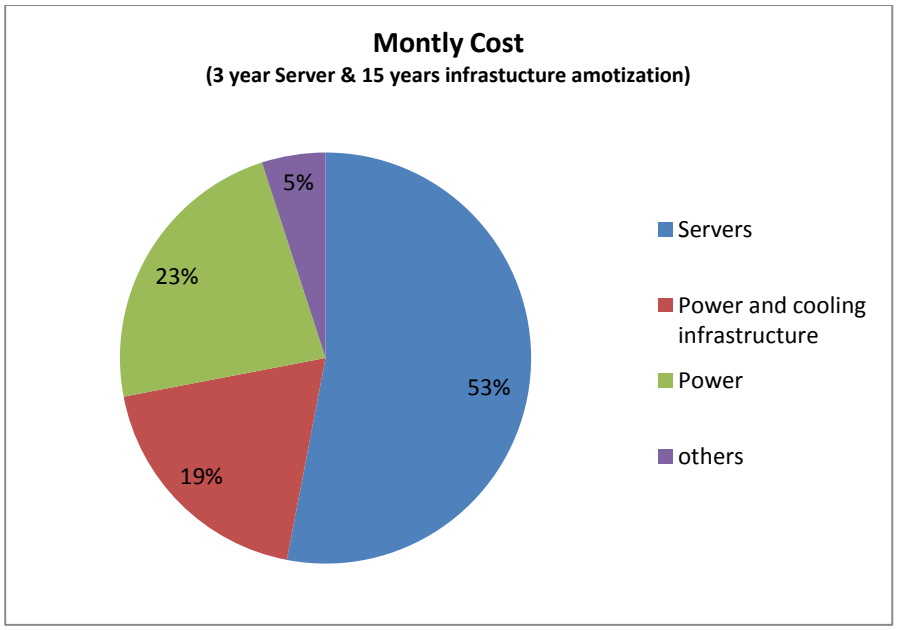


Figure 1.5: Energy distribution in a data centre [37]

Data centers allocate resources to user applications based on peak workload characteristics to meet the desired performance goals. This high performance and demands that need to be fulfilled without giving much weightage energy optimization have been the only the interest of data center deployments. However, for a data center, the energy expenditure plays a crucial role in running its day to day operations. According to Amazon estimates (Figure 1.5), the energy-related total cost of a data center is 42% that covers both electricity consumption (~19%) and the capital investment of a cooling system (23%) amortized over a fifteen year period. In another survey, it was found that a traditional data center having 1000 racks needs 10MWof power to operate [38]. A source of energy wastage at old data centres is due to the inefficient usage of resources. As per reports of Barroso et al. [39], servers operate most of the time at 10 to 50% of their maximum utilization level. This data has been reported for more than five thousand servers over the span of a six months period (Figure 1.6).

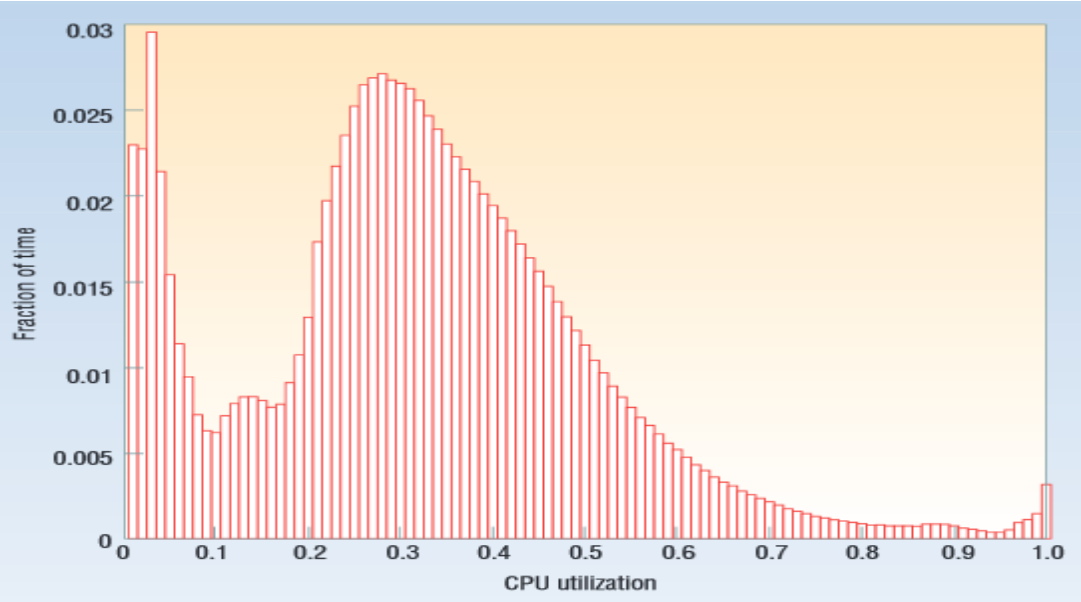


Figure 1.6: Average CPU utilization of more than 5,000 servers for a six-month period [39]

High energy usage is unacceptable since it yields a high energy cost. Data centers are not only expensive to maintain because of the high operational and energy cost, these are also not eco-friendly. Data centers in the US discharge more carbon than both Argentina and the Netherlands [40]. High energy costs and immense carbon emissions are produced due to the enormous amounts of electricity required to power and cool hosted servers in data centers. In addition to that, there is also the pressure from the governments in various countries to reduce the carbon footprint so that there is minimal effect on climate change. For example, the

Government of Japan has established a Japan Data Center Council to measure and curb this unsustainable risein carbon emission by data centers [41]. The Federal Data Center Consolidation Initiative (FDCCI) was setup in 2010 to initiate a drive for more efficient computing and to promote Green IT by reducing the overall energy consumption [42]. Even the leading worldwide ICT industries, technology providers and utility companies have setup an open industry consortium, namely Green Grid, for effective and accountable resource efficient ICT [43].

Curbing this unsustainable increase in energy consumption and release of carbon footprints into the environment is a challenging and complex issue because of the ever-growing trend of online computing applications and the massive amount of data involved. These online computing applications require a large IT infrastructure (servers, disk storage, network infrastructure, software applications etc.) to process data fast and within the stipulated time period. Thus, the research on "Energy efficient computing" is a critical need and is a problem of great magnitude, leading us towards efficient utilization of resources as well as minimization of energy consumption while satisfying the Quality of Service (QoS) requirements of the users.

"*Green Cloud computing is envisioned to achieve not only efficient processing and utilization of computing infrastructure, but also minimize energy consumption* [44]".

With this considerable discussion on today's need of sustainable computing, the next section presents existing techniques and mechanisms used for energy efficient cloud computing.

1.3 Energy Efficient Cloud Computing Techniques

To minimize the operating and procurement cost of IT infrastructure, many big and small cloud enterprises are focusing on energy conservation rather than the sole concern of pure performance. Energy conservation can be achieved through energy efficient servers, resource management, virtual network scheduling, tweaking communication devices, efficiency of applications running in the system, energy aware user contracts etc. In this section, the possible areas of energy conservation are discussed for a cloud infrastructure as shown in Figure 1.7.

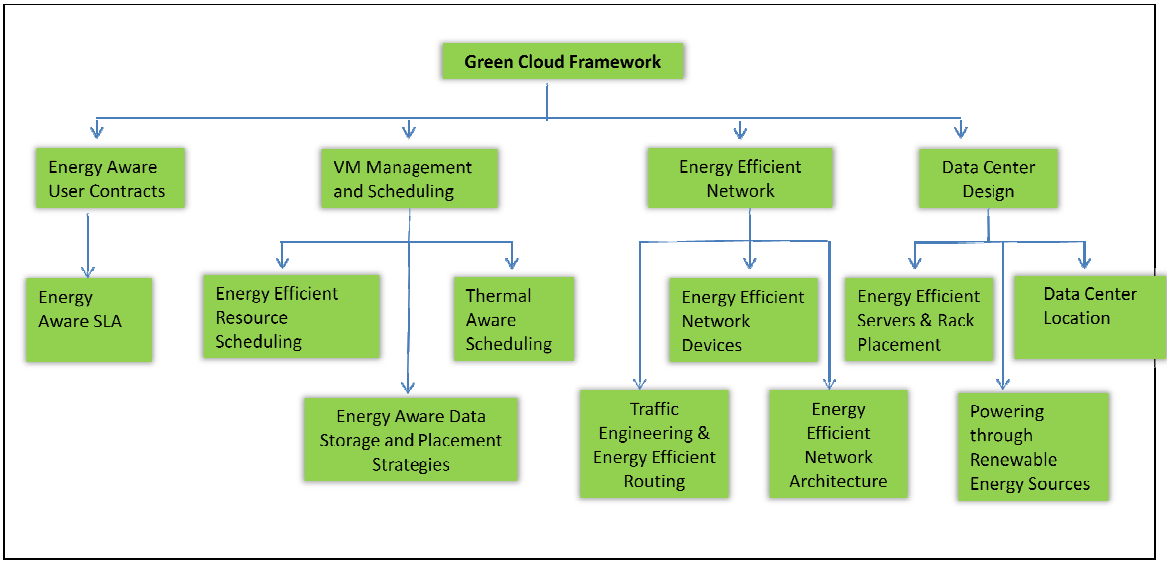


Figure 1.7: A taxonomy of Green Cloud Framework

1.3.1 VM Management and Scheduling

Cloud resources are provisioned and allocated to user applications in a way that improves system utilization, improves energy consumption and meets the desired QoS. VMs hosting user applications are scheduled according to the host status in terms of CPU, memory, disk utilization, network usage, data store read/write latency, thermal state, fault tolerance, criticality of applications, workload, operating cost etc. This section presents VM management and energy saving scheduling at various levels.

1.3.1.1 Thermal Aware Scheduling

The purpose of thermal aware scheduling is to minimize the overall temperature of a data center [45]and keep the systems at a safe operating temperature. In a thermal aware scheduling, thermal states of the systems are monitored periodically and jobs from overheated nodes are migrated [46]. The goal of thermal aware scheduling is to reduce the data center-wide thermal gradient, hotspots and cooling magnitude [47]. Significant parts of electrical energy consumed by the physical machines are transformed to heat. The high temperature of a data center causes many problems such as reducing the life time of the system components. Further, in modern 1 unit rack and blade servers lead to high density computing resources that complicate the heat dissipation. Thus, to reduce the energy needed by the cooling system to preserve the safe operating temperature of the data center, there is a need for thermal aware workload placements [48-49].

1.3.1.2 Energy efficient Resource Scheduling

Energy efficient Resource Scheduling or management means allocation of cloud resources in such a manner that the Quality of Service (QoS) requirements of the users are preserved via Service Level Agreement (SLA) and the energy usage is reduced [46]. Energy efficiency of a data center can be increased through server consolidation by service virtualization [50-51].

Virtualization partitions computational resources and allows the sharing of hardware. Many services usually need a small fraction of the available computational resources. However, even when run at a low utilization, servers typically need up to 70% of their maximum power consumption [52]. Such services can be virtualized and run within a VM to consolidate many services on single server. It results in an increase of resource utilization and a significant increase in the overall energy efficiency of a data center. Further, energy savings can be achieved by VM migration according to the current utilization of resources, virtual network topologies and using thermal state of computing nodes.

1.3.1.3 Energy Aware Data Storage and Placement Strategies

With the exceptional growth of web and social network data, there is a need to address various research issues like the life cycle of data management, large scale storage, data analysis, and processing, data archiving etc. One of the important reasons that data consumes power is because the data in databases requires operating storage disks to store, analysis, and process [53]. Here, data placement strategies, efficient data archiving, sharing and searching, efficient replication strategies etc. help to reduce the amount of data stored, that ultimately reduce the amount of power used to operate the data storage infrastructure. This emphasis on reducing the data movement at build time (Starting) and Running time (intermediate Stage) of the scientific workflow and design specific algorithms to automatically place and move application data (dynamically calculated dependency factor) to the node with the highest dependency factor [54]. Taking into account both user data locations and application properties, Liu et al. [55] designed a novel Distributed Energy efficient Scheduler (DEES) that aims at seamlessly integrating the process of scheduling tasks with data placement strategies to provide energy savings. DEES encompasses three main components: energy-aware ranking, performance aware scheduling, and energy-aware dispatching.

1.3.2 Energy Efficient Network

The network is one of the key components that consume significant energy since there are network connections in a data center, there is a fixed network among the data centers and there is an end user network that accesses various cloud services through smart phones, laptops, tablets etc.

1.3.2.1 Energy Efficient Network Devices

One of the important aspects of energy efficient operations in a data center is the installation of energy saving network devices. These network devices use energy as per the network traffic load and switch to power saving mode as they become idle. It offers energy efficient operations such as power saving modes through throttling of processors [37], turning off network ports, links, switches and rate adaptation. For inter-data center network, NetStitcher can be used to reduce the network idle time by bulk transfer between data centers [56] or by leveraging the edge of different data rates. The energy efficient Ethernet IEEE 802.3az reduces the energy consumption by 50% and it fully compatible with existing network devices [57]. It consumes less power during low data activity periods.

1.3.2.2 Traffic Engineering and Energy Efficient Routing

Energy can be conserved in case of wired/wireless networks through traffic engineering and routing methods. There are numerous studies on energy aware routing protocols for wireless ad-hoc networks [58-59], multi-hop wireless networks[60], wireless sensor networks [61-62] and other network technologies. The basic approach of energy efficient routing is to select proper transmission ranges and routes to save energy for multihop packet deliveries [63].

Kostic et al. [64] have proposed energy aware traffic engineering, in which the same traffic rate is maintained and energy is still conserved with rate adaption and by dividing the network load is among multiple paths. In a REsPoNse framework [65], energy critical paths are found, used and network traffic is diverted in such a manner that enables the network to enter a low power state at various levels. Mingui et al. [66] have proposed, GreenTE, an inter domain traffic engineering mechanism that put the maximum number of links to sleep mode while addressing performance guarantee.

1.3.2.3 Energy Efficient Network Architecture

Poor network architecture plays a pivotal role in significant energy loss. Due to poor network, some parts of a network are underutilized which ultimately leads to wastage of unnecessary energy in rerouting traffic. It puts additional stress on the cooling system in a data center. User application at a global scale results in VMs residing across various data centers and communicates over the network. Energy-aware communication among data centers needs to explore, based upon the communication pattern, time, data transfer overhead, and energy consumed by the network infrastructure. Due to large usage of the cloud application via wireless communication, the wireless traffic is drastically increasing. The wireless connection is not energy efficient due to interference, path loss, and processing of error detection and correction [67], which causes unnecessary energy consumption by supporting tasks as against the main task of data communications.

VMs communication in a data center can be energy optimized by establishing efficient virtual network topologies [46]. By monitoring network traffic and dynamic adaption of network communication links can reduce network traffic and data transfer overhead, thus reducing the energy consumption. For inter-data center network, solutions such as NetStitcher can be deployed for effective utilization of the network bandwidth and achievement of energy goals [68].

1.3.3 Advance Data Center Design

Energy efficient designs and operations are the primary concern of a data center to reduce electricity usage, cost, and carbon footprint and enhance the life cycle of the data center resources. Use of energy efficient IT equipment, effective air management, usage of renewable energy sources contribute in significant saving of energy consumption, cost saving and making the environment sustainable. This section addresses the issue of data center design from the energy efficiency perspective.

1.3.3.1 Energy Efficient Hardware

One of the best approaches to increase the energy efficiency of a data centre is to install and use of energy efficient hardware. Power can be saved at the server and cluster level through Dynamic Voltage and Frequency Scaling (DVFS) [69-73].

Under DVFS, to optimize the power consumption, the frequency of the CPU is adjusted according to the workload. Processor clock frequency and supply voltage are adjusted/reduced during some time slots, for example, idle or communication phases, lack of user–machine interaction etc., different redundant hardware parts can incrementally be turned off or put in hibernating mode (display, disc etc.). A DVFS-enabled cluster is a compute cluster where compute nodes can run at multiple power-performance operating points [74]. The DVFS techniques have been applied in the high performance computing fields, for example, in large data centers, to reduce power consumption and achieve high reliability and availability. These measures enable slowing down of the CPU clock speeds (clock gating), or powering off parts of the chips (power gating), if these are idle. By sensing the lack of user–machine interaction, different redundant hardware parts can incrementally be turned off or put in hibernating mode (display, disc etc.). A second method is selection of servers with variable fan speed rather than standard fan speed. A fan adjusts the speed according to the cooling requirement of the server components and thus consumes less power. Thirdly, power can be saved by consolidating IT system redundancy [75]. The power load efficiency can be increased by supplying rack mounted power to the entire rack as against supplying power to individual servers. It potentially improves power supply efficiency.

1.3.3.2 Renewable Energy Sources

Renewable energy for a data center means sourcing data center power from renewable sources such as sunlight, wind, rain, tides, waves and geothermal heat. It will place caps on carbon emissions. Google [76] is using 35% of power for their operations through renewable energy. It focuses on improving the energy efficiency of the data center by migration the workloads to better use renewable energy sources. Eduard Oró et al. [77] have presented work on renewable energy integration into data centers and throw light on present scenarios of data center, its environmental guidelines, business models, power distribution system and cooling system. Deng et al. [78] emphasized on the usage of a grid-tie device that integrates renewable energy into the data center power delivery system. A renewable-powered instance as a metric has been used to measure the concentration of renewable energy in data centers.

1.3.3.3 Data Center Locations

Data center locations are a contributory factor to bring energy efficiency at the data center level. A data center's energy efficiency depends upon many factors that vary with the geographical location because of the environmental temperature. Google established some of its data centers in cold locations such as Hamina and the Gulf of Finland [79]. These data centers use the cold water of the sea for cooling services. The installed pumps facilitate cold sea water to the facilitating centers and also transfer back the heat generated from data center operations to the sea water through a heat exchanger. So, the energy efficiency changes dynamically depending upon the energy cost, carbon emission rate, workload, environment climate, and the installed cooling system at different data center locations. The workloads of these centers are generally HPC/ compute intensive parallel applications with low data transfer requirements or less human interactions [80].

1.3.4 Energy Aware User Contracts

Energy-aware user contracts or Green SLAs urge the users to value sustainable computing. The data center's energy is conserved through user negotiated energy saving strategies and planning. It emphasizes to incorporate energy percentage into the users contracts and in lieu of that the users have an incentive to use more resources or cost benefits.

1.3.4.1 Energy-aware SLA

In a cloud environment, SLA is a negotiation between the cloud provider and the user under which the cloud provider must provide services within the ambit of QoS parameters such as uptime of service availability, security, cost, service incentives and penalties on violation of agreement. Energy reduction at the Service Level Agreement (SLA) level comprises energy saving strategies and negotiation. It offers different energy efficient negotiation solutions to the users. The users can see which offer is suitable to them [81]. GreenIT-SLA measures and monitors the eco-efficiency of green IT services offered by a data center. It uses energy metrics to monitor the eco-efficiency of the offered services [82].

1.4 Thesis Organization

Chapter 1 presents the basic introduction of cloud computing and relevant concepts, the rest of the thesis is derived from the basic objectives. The remaining chapters of the thesis are structured as follows.

Chapter 2: With a brief discussion of the need for energy efficient cloud computing and techniques in chapter 1, chapter 2 presents the existing literature on energy efficient cloud computing. It is primarily divided into Energy efficient resource scheduling algorithms and SLA Based Energy Aware Scheduling algorithms used for cloud environment. This chapter winds up with problem formulation and objectives. This chapter accomplishes Objective 1 and published in [P3, P4].

Chapter 3: presents the design of the proposed energy efficient resources scheduling algorithm. The proposed algorithm optimises the energy consumption through efficient resource scheduling, Virtual Machine (VM) consolidation and VM admission control policies. Further, this chapter presents the energy, resource reservation and load prediction models. A high level architecture of the energy efficient cloud service framework is also presented here. This chapter accomplishes Objective 2 and published in [P1].

Chapter 4: presents the design of the Green Service Level Agreement (GSLA) aware Cloud Resource Reservation (GSLACRR) algorithm based on the user and cloud service provider negotiation. Energy of a data center is conserved through user negotiated energy saving strategies and planning. This chapter presents the GSLA architecture that outlines the general negotiation structure and approach. This chapter accomplishes Objective 3 and published in [P2].

Chapter 5: This chapter provides the design and implementation details of the proposed algorithms. It covers the various components, architecture, design and development of the energy efficient cloud service framework. The detailed requirements and design have been analyzed through UML. The implementation details of the experimental testbed that carry out the proposed algorithms are provided. This chapter partly accomplishes Objective 4 and published in [P1, P2].

Chapter 6: The proposed algorithms have been validated and compared with existing ones. Several experiments were conducted with different number of VM requests, and different VMs types which were executed according to the different schedule patterns. Experimental results have been collected from the implementation explained in chapter 5. To get statistically significant differences of the existing and the proposed algorithms, paired samples t-test have been conducted. This work accomplishes Objective 4 and is published in [P1, P2]

Chapter 7: gives concluding remarks on the thesis by highlighting the main contributions of this research work. In the end, the future scope of work has been discussed.

1.5 Thesis Contribution

This thesis makes the following research contributions:

i. This thesis presents detailed literature review of the work done in the area of energy efficient cloud computing. It highlights the present requirements, key challenges and the mechanisms available to address these challenges.

ii. To address the challenges of energy efficient cloud resource management, this thesis presents novel techniques, models, and algorithms for the cloud environment as per the set of objectives delineated.

iii. The proposed Energy-efficient Cloud Resource Scheduling (ECRS) algorithm has been designed, developed, and validated in this thesis. Its main aim is to manage resources efficiently by striking a balance between performance and energy.

iv. Green Service Level Agreement (GSLA) parameters based resources scheduling algorithm, GSLA aware Cloud Resource Reservation (GSLACRR) algorithm, has been designed, implemented and validated. It offers cloud eco-system services to the users that result in cost benefits to the user and the cloud service providers.

v. To demonstrate the usability of the proposed algorithms, an energy efficient cloud framework, named ACA-Cloud, has been designed, developed, and tested on a heterogeneous testbed of physical servers.

vi. The experimental results show that the proposed resource scheduling algorithm outperforms in comparison to existing ones in all aspects such as performance and energy conservation.

vii. Statistical analysis of experimental results has been performed in order to assess the significant differences of the existing and the proposed algorithms. Paired samples t-test has been conducted on the outcome results of the existing and the proposed resource scheduling algorithms.

viii. The thesis demonstrates the applicability of the proposed model for small and medium cloud setups for many kind of organisation such as public sector units, government sectors, universities/institutes etc.

Chapter 2

Literature Survey

Cloud computing has emerged as an incredible technology to provide Internet computing services. Infrastructure as a Service (IaaS), a cloud computing service model is one of the significant emergent fields in which users get online computers in the form of virtual machines associated with storage, configured firewalls and network devices. Efficient resource management is a critical task to achieve performance in cloud computing for IaaS. Unlike other traditional parallel and distributed systems, resource management in cloud computing involves VM provisioning and scheduling. In addition to this addressing the issues of energy efficiency without severe loss of performance makes the whole process more challenging. Energy efficient resource management offers various benefits like energy conservation, scalability, Quality of Service (QoS), optimal utility, reduced overheads, improved throughput and cost effectiveness.

This chapter focuses on cloud resource management with special attention to energy efficient resource scheduling. Energy efficiency is addressed at the data center level through resource scheduling algorithms and user negotiation level through Service Level Agreement (SLA). At last, the chapter concludes with the gap in existing research work and finally lists down the objectives of the thesis.

2.1 Energy Efficient Resource Scheduling in Cloud Computing

Resource scheduling is a viable and effective solution to handle workload fluctuations, while guaranteeing Quality of Service (QoS) and increasing the predictability of the system. The computing and network resources are limited and have to be shared efficiently among the users, in a virtual manner. Energy efficient resource scheduling offers a number of advantages such as energy conservation, scalability, QoS, optimal utility, reduced overheads, improved throughput, reduced latency and cost effectiveness [83].This section explains the various frameworks and scheduling approaches/mechanisms used to achieve energy efficiency and performance in data centers. These are classified into five categories as shown in Figure 2.1. Table 2.1 highlights various aspects of these research papers such as the proposed algorithm, tools and technology, data center types etc. The terms green computing and energy efficiency are used interchangeably in the rest of the thesis.

2.1.1 Consolidation based Heuristic Approaches

Consolidation means to bring together the services of multiple machines into a single machine which helps to improve the work efficiency and reduce the energy consumption by assigning the same work to lesser number of machines and switching off unused machines [84].VM consolidation is the key aspect of virtualization to improve the server utilization and reduce the energy consumption of a cloud data center. Through virtualization, VMs are migrated among the servers and consolidated on a set of servers according to the requirements of resources and the desired user's QoS. It helps switch the idle servers, drawing power unnecessarily, on to low power mode [85-86].

Modified Best Fit Decreasing algorithm: Buyya et al. [44] have proposed an energy efficient architectural framework for cloud computing. It modelled VM placement to nodes as a multidimensional bin-packing problem in which the bins characterize nodes and the VMs are the items to be packed. To save the energy consumption of a data center, the authors proposed the Modified Best Fit Decreasing (MBFD) algorithm, in which VMs are organised in a declining order of current utilization and each VM is mapped to a node that increases the least power consumption of a data center.

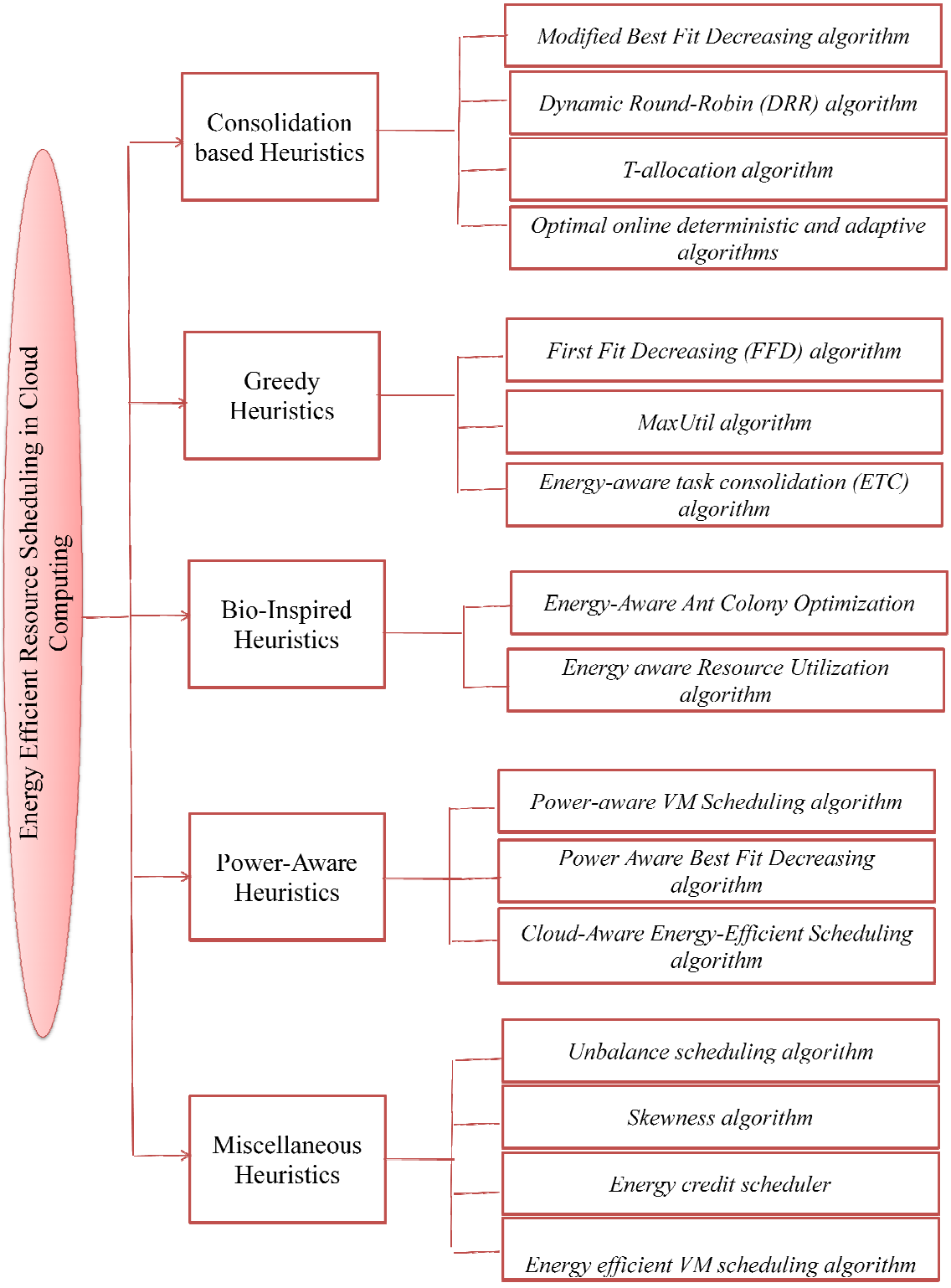


Figure 2.1: Energy efficient resource scheduling taxonomy

Dynamic Round-Robin (DRR) algorithm: Ching-Chi et al. [87] have proposed DRR for VM placement and migration. It is an extension of the Round Robin algorithm. The algorithm is based upon two rules. First, as a VM completes its execution on a server, there are no more VMs allocated on that server. That server is declared as in "retiring" state. As the rest of the running VMs complete their execution on that server, the server is put on a power sleep mode. Second, if a server has been waiting for a long time in a retiring state, all the running VMs are migrated and the server is switched into a power sleep mode.

T-allocation algorithm: Quan et al. [88] have proposed scheduling of tasks involving human efforts, to save the energy of a data center. In a data center, there are heterogeneous types of computing resources. The proposed T-allocation heuristic places heavy load applications to new generation servers having high performance, energy efficient and more number of processing cores. The light applications are placed on older servers. Older servers with small number of cores have a lower chance of full load. Hence, there are greater chances to put these servers on low-power modes to save energy.

Optimal online deterministic and adaptive heuristics: Beloglazov et al. [89] have proposed a set of heuristics (NPA, Dynamic Voltage Frequency Scaling (DVFS), THR-MMT, THR-MMT, IQR-MMT, MAD-MMT, LRR-MMT, and LR-MMT) to achieve energy and performance efficiency through dynamic VM consolidation. The problem of dynamic VM consolidation is divided into four parts (1) Selection of the overloaded host from where one or more VMs need to be migrated (2) Identifying the under loaded host whose all VMs need to be migrated for it to be put it in sleep mode (3) Recognition of the VM or VMs that need to be migrated from the overloaded host (4) Finding the destination host where the migration of VMs need to take place. Additionally, to that the authors define single VM migration problem, cost incurrence and competitive analysis of optimal online deterministic algorithms

2.1.1 Greedy Heuristic Approaches

Greedy algorithms are problem solving heuristics which make a greedy choice at each stage to achieve a global optimal solution [90]. These are constructive heuristics that produce local optimal solution by selecting the best choice at each step of construction. Then, the addition of a partial solution delivers a global optimal solution in a justifiable time frame.

First Fit Decreasing (FFD) algorithm: First Fit Decreasing is a well-known greedy algorithm for classic bin packing problems and used commonly for server consolidation [91]. It sorts the items (i.e VMs) by their decreasing order i.e. capacity usage level. Then the first VM is mapped to the first host. The second VM is mapped to the same host if it can accommodate the requested VM resource requirements. Otherwise, a new host is brought and the VM is mapped to this new host. The steps are repeated until all the VMs in a queue have been mapped to the hosts.

MaxUtil algorithm: Lee et al. [92] have presented an energy efficient cloud model which brings energy efficiency by maximizing resource utilization. The authors proposed two greedy approaches named as energy-conscious task consolidation (ECTC) and MaxUtil which made an effort to reduce energy consumption by consolidating as many tasks as possible, to a VM. The task consolidation decision is taken by monitoring the utilization, energy efficiency of a resource and the cost function.

Energy-aware task consolidation (ETC) algorithm: The MaxUtil [92] algorithm has been extended for large virtual clusters through the energy-aware task consolidation (ETC) algorithm [93]. ETC minimizes energy consumption by restricting the CPU use below a 70% threshold limit. The algorithm is specially designed for a data center where virtual clusters (VCs) and VMs reside on the same or different racks but the network transmission is relatively constant between the VCs.

2.1.2 Bio-Inspired Heuristic Approaches

Bio-Inspired heuristics are nature-based computing techniques that specify a set of simple rules, a set of simple organisms which follow to those rules and a method of repeatedly applying such rules. Biologically inspired computing encompassed many novel approaches such as Ant Colony Optimization (ACO), Bee Colony Optimization, Swarm Optimization, Fast bacterial swarming algorithm and many more such techniques to manage scalability, heterogeneity, network and energy problems [94-95].

Energy-Aware Ant Colony Optimization: Feller et al. [96] have proposed the Energy-Aware Ant Colony Optimization Based Workload Consolidation algorithm. It is a meta-heuristic based algorithm. In the proposed algorithm, each ant receives items (i.e. VMs), opens a bin Physical Machine (PM), and begins to place the items into the bin. This is accomplished by the

use of a probabilistic decision rule, which characterizes the allure for an ant to choose a particular item as the successive one to place in its current bin. This rule is set on the information available regarding the current pheromone strength on the item-bin pair. It leads the ants towards selecting the most promising items and finding better placement. At last, when the solutions of all the ants are ready, the quantity of pheromone combined with each item-bin pair is updated. Again, there is pheromone evaporation which intensifies item-bin pairs to find better solutions [97-98].

Energy aware Resource Utilization algorithm (ERU): ERU for cloud environments [99] has been based on artificial bee colony. An artificial bee colony is used to locate the best job-node pair to maximize the energy efficiency. In the presented work, the authors considered CPU, memory and energy thresholds to assign a task to a node. The proposed approach makes a decision of resource allocation based on the past resource usage and energy consumption data.

2.1.3 Power-Aware Heuristic Approaches

Power-aware heuristics reduce the power consumption of a server by regulating the voltage and frequency at appropriate times to optimize a combined consideration of performance and power consumption [100]. The technique of DVFS can control the dynamic frequency scaling, dynamic speed scaling and dynamic power scaling of many server components such as CPUs, main memories, local buses, expansion cards etc. This technique is used as the part of scheduling to reduce the energy consumption.

Power-aware VM Scheduling algorithm: Younge et al. [45] have presented a Green Cloud Framework to reduce the energy consumption for cloud environment. The authors present a power-aware VM scheduling algorithm which continually assigns each host with as many VMs as possible. After each specified interval of time 't', the algorithm executes for each waiting VM in the queue. The first host in the resource pool is selected and checked if it has enough virtual cores to fulfil the new VM. If it has the required virtual cores, it is mapped on to the selected host and this procedure is continued until the VM queue is empty. In addition to this, the authors provide a VM management solution using VM imaging and VM live migration.

Power Aware Best Fit Decreasing (PABFD) algorithm: The authors have presented the energy efficient resource management system [46] in two parts. The first part was to provision and

allocate a PM to a new VM request using PABFD algorithms. The second part focuses on optimization of the present VM allocation on active PMs. The proposed heuristic migrates a VM from a PM when the CPU utilization of a PM either goes above the upper threshold level or goes below the lower threshold level.

Cloud-Aware Energy efficient Scheduling algorithm: Calheiros et al. [101] have proposed a cloud aware energy efficient scheduling algorithm that exploits the DVFS capability of a CPU by coupling it with task scheduling. The basic idea is to schedule a task on VM that can meet its deadline and execute all assigned tasks with the lowest frequency level. The proposed approach is designed dedicatedly to urgent and CPU intensive tasks.

2.1.5 Miscellaneous Heuristic Approaches

All the heuristic approaches which could not be categorised into one of the aforementioned categories, are listed into miscellaneous heuristic approaches.

Unbalance scheduling algorithm: An energy-aware framework, namely Green Open Cloud (GOC), has been proposed [102-103] to manage cloud resources. The basic idea of energy conservation is to switch the servers to on/off modes based upon the users' resource usage prediction. This helps in making decisions to switch the server on or off in the near future. Secondly, GOC offers "green" advice to the users to make them energy aware [104-105].

Skewness algorithm: Xiao et al. [106] used the idea of "skewness" to measure the imbalance in the resource utilization of servers. To improve the overall utilization of servers, an optimized combination of different types of workloads on hosted VMs is required. The authors implement this idea by minimizing the skewness. Further, a load prediction algorithm has been designed that predicts the future trend of the application's resource usage pattern. It helps to improve the stability and the decision making of resource allocation that ultimately contributes in green computing by optimizing the number of active servers.

Energy credit scheduler: Kim et al. [107] have proposed an energy credit scheduler, a modified version of default credit scheduler of Xen [108], which estimates the energy consumption of each VM. Based on that estimation, resources are allocated according to the energy budget of a VM so that energy consumption rates remain below than the user defined values.

Energy efficient VM scheduling algorithm (EEVS): Ding et al. [109] have proposed EEVS algorithm to execute deadline constraint tasks. In the proposed approach, weights are assigned to each PM based on optimal performance-power ratio. The EEVS process is divided into some time scheduling intervals. First, the requested VMs are scheduled on a PM having higher performance-power ratio and also having each core operating at optimal frequency levels. After each interval, cloud resources are further reconfigured to squeeze the workload to save more energy.

Table 2.1: Energy efficient scheduling algorithms for cloud computing

Heuristics/ Meta-Heuristic	**Platform/toolkit**	**Technology**	**Data Center**	**Authors Conclusion/Remarks**
Modified Best Fit Decreasing algorithm (Buyya et al., 2010) [44]	CloudSim toolkit	Java	Heterogeneous	• Dynamic reallocation of VM provides higher energy saving as compared to static VM allocation. • Minimization of Migrations (MM) policy achieve better energy saving and less SLA violation percentage as compared with Non-Power Aware (NPA), Single Threshold(ST), DVFS policies. • Energy can be more conserved with SLA relaxation.
Unbalance scheduling (Lefèvre et al., 2010)[102]	Cloud infrastructure is configure with Xen virtualization technology		Homogenous	• Unbalanced scheduling with green scenario consume 25% less energy as compared to round robin scheduling with basic scenario
Power-aware VM Scheduling algorithm (Younge et al.,	OpenNebula		Homogenous	• The proposed power aware scheduling save 12% system's power in comparisons of default non power aware scheduling

Continue…

Heuristics/ Meta-Heuristic	Platform/toolkit	Technology	Data Center	Authors Conclusion/Remarks
2010)[45]				policy of OpenNeubla scheduler
Energy-Aware Ant Colony Optimization(Feller et al.,2011)[96]	Java based simulation toolkit	Java	Homogenous	• The proposed algorithm conserves 4.1% more energy as compared to First-Fit Decreasing (FFD) greedy algorithm for workload placement.
Dynamic Round-Robin Hybrid method (Lin et al., 2011)[87]	Eucalyptus Cloud		Homogenous	• DRR and the Hybrid method reduce 56.4% and 55.9% power consumption as compared to Round Robin policy. • The proposed algorithms save 3% power on an average basis as compared to PowerSave policy of Eucalyptus cloud.
Power Aware Best Fit Decreasing algorithm (Beloglazov et al., 2012)[46]	CloudSim toolkit	Java	Heterogeneous	• PABFD algorithm substantially reduces the energy consumption in comparison to static resource allocation techniques for cloud environment. • Proposed algorithm achieve 66% better energy saving and 100% performance delivery in comparison to non-migration aware DVFS policy.
T-allocation (Quan et al., 2012)[88]	Gnu Linear Programming Toolkit (glpk)	Mixed Integer Linear Programming (MILP)	Heterogeneous	• Algorithm is effective where compute intensive application requests come more frequently. • It is more useful with old data centers where many old servers are deployed with heavy workload and many new generation high performance servers are working with light workload.
ECTC, MaxUtil (Lee et al., 2012)[92]	Simulator based upon homogeneous Cloud model		Homogeneous	• Proposed algorithms ECTC and MaxUtil outperformed by 18% and 13% respectably as compared to random algorithms.

Continue…

Heuristics/ Meta-Heuristic	Platform/toolkit	Technology	Data Center	Authors Conclusion/Remarks
Adaptive Heuristics For Dynamic VM Consolidation (Beloglazov et al., 2012)[89]	CloudSim toolkit	Java	Heterogeneous	• The proposed Local Regression policy (a host overload detection policy) in combination with Minimum Migration Time policy (MMT: a VM selection policy) outperforms in terms of SLA violation and energy consumption.
SKEWNESS ALGORITHM (Xiao et al., 2013) [106]	Cloud infrastructure is configure with Xen virtualization technology		Homogeneous	• Proposed algorithm makes effective balance of system overloading and energy consumption of machines with multi-resource constraints.
Energy-aware task consolidation (ETC) (Hsu et al.,2014)[93]	Cloud Simulator composed of several virtual clusters in a data center.			• Proposed algorithm can save energy 17% as compared to MaxUtil[92] .
Cloud-Aware Energy efficient Scheduling (Calheiros et al., 2014) [101]	CloudSim toolkit	Java	Homogeneous	• Proposed algorithm improves energy consumption between 2% to 29%. It depends upon request pattern, number of requests and urgency of requests.
Energy-credit scheduler (ECS) (Kim et al., 2014) [107]	Xen Virtualized system[108]		Homogenous	• The energy estimation model is presented that estimates the energy consumption of a VM based on in-processor events of VM. This is useful in billing system. • An energy aware VM scheduling is proposed that allocates resources to VMs according to energy budget.
Energy aware Resource Utilization (Kansal et al., 2015)[99]	CloudSim	Java	Homogeneous	• The proposed algorithm ERU is compared with the existing FFD and ACO. It is found that ERU outperforms the existing techniques by means of

Continue…

Heuristics/ Meta-Heuristic	Platform/toolkit	Technology	Data Center	Authors Conclusion/Remarks
				minimizing energy consumption and execution time of applications.
Energy efficient scheduling of virtual machines (EEVS) (Ding et al., 2015)[109]	Physical cloud testbed		Heterogeneous	• The proposed algorithm EEVS compared with EEVS-N (EEVS algorithm without ranking the physical machines), Homogeneous [45] and MBFD [44]. • EEVS consumes 24.8% and 11% less energy as compared to Homogeneous and MBFD respectively. • EEVS has lesser number of failed VMs as compared to other algorithms. • The numbers of active PMs found are largest in MBFD. EEVS and EEVS-N have almost same number of active PMs and Homogeneous use less active PMs.

2.2 Comparative Analysis of Energy Efficient Scheduling Algorithms

A comparison of these policies and algorithms are made in Table 2.2 based on various aspects such as energy saving, virtualization, DVFS, SLA, deadline constraint, network, multiple data centers and load balancing.

Table 2.2: Comparative analysis of energy efficient scheduling algorithms and policies

	Buyya et al.[44], Beloglazov et al.[46][89]	**Lefèvre et al.[102]-[103]**	**Younge et al. [45]**	**Feller et al. [96]**	**Lin et al.[87]**	**Quan et al.[88]**
Energy Saving	✓	✓	✓	✓	✓	✓
Virtualization	✓	✓	✓	✓	✓	✓
DVFS						✓
SLA	✓					
Deadline constraint		✓				
Network		✓				
Multiple data center	✓					
Load balancing		✓				

continue..

	Lee et al.[92]	**Xiao et al.[106]**	**Hsu et al.[93]**	**Calheiros et al.[101]**	**Kim et al.**[107]	**Kansal et al.**[99]	**Ding et al.**[109]
Energy Saving	✓	✓	✓	✓	✓	✓	✓
Virtualization	✓	✓	✓	✓	✓	✓	✓
DVFS				✓			
SLA			✓				
Deadline constraint	✓			✓			✓
Network			✓				
Multiple data center							
Load balancing		✓					

2.3 SLA Based Energy Aware Scheduling in Cloud Computing

SLA specification, negotiation and enforcement with the energy conservation perspective is a relatively new concern in cloud computing. A SLA is a formal legal agreement between a cloud provider and a service user regarding the offered QoS and cost. As per SLA negotiation, the cloud providers must provide their services within the ambit of the QoS parameters such as uptime of service availability, security, cost, service incentives and penalties on violation of agreement. Energy can be conserved at a SLA level biased on promoting decreased resource allocation within the cloud customer's accepted limits. It consists of a round of negotiation offers to the party so that a balance between energy and performance can be made. It comprises of a number of incentives for the users to trade off the traditional performance parameters and matrices. This section discusses the details of some existing Green SLA (GSLA) framework and scheduling heuristics.

2.3.1 Power-Aware Heuristic Approaches

As discussed in section 2.1.4, power-aware heuristics reap advantage of the DVFS technique to control the power consumption. This section discusses the earlier work on power-aware heuristic approaches to satisfy the SLA parameters.

Modified best fit decreasing heuristic: Gao et al. [110] have proposed an energy efficient dynamic resource management framework which takes the advantage of DVFS and server consolidation. The proposed algorithm is based upon modified best fit decreasing heuristic which minimizes the total power consumption of all the hosts in addition to satisfying the response-time based SLA. VMs are mapped on a server based upon the operating CPU frequency for each active host and the increment of power consumption.

Energy Aware SLA (EASLA): In [111], Xuedi et al. have proposed the EASLA scheduling algorithm for precedence-constrained applications. This algorithm minimizes the energy consumption within the permissible limit of the makespan of tasks. Under the proposed architecture, the user negotiates with the service provider regarding the makespan extension of a task to reduce the energy consumption under the QoS. The service provider assesses the makespan and energy consumption based upon the user's input on the estimated computation time of the task and dependency relations, and the required number of computing nodes. After reaching an agreement, the QoS parameter is sent to the scheduler. The scheduler allocates

tasks to the appropriate computing nodes with the frequency and sequence information of a task. Finally, on the completion of a task, the results return to the user. The basic idea of the algorithm is to allot each slack to the tasks and to scale down the CPU frequencies to minimize the power consumption. Initially, it locates the set of independent sub tasks for each task and then iteratively assigns each slack to that independent set of sub tasks whose total energy can be reduced to a maximum level.

2.3.2 Bio-Inspired Heuristic

As discussed in section 2.1.3, Bio-Inspired heuristics focus on nature-based computing to solve the problems of different areas such as resource scheduling and provisioning, data center networks, energy optimization etc. This section discusses the Bio-Inspired heuristic approaches used in the negotiation process of SLAs.

Particle Swarm Optimization (PSO): PSO based negotiation for SLA has been proposed in[81]. PSO evolves populations of possible negotiation offers for each party about a new offer. This process leads to an approximate negotiation solution. The authors proposed negotiation work biased towards promoting decreased resource allocation within the cloud customer's accepted limits. It ultimately inclined the users to strike a balance between energy and performance.

Ant colony optimization: Gao et al. [112] have proposed integrated power management solutions for virtualized data centers which take the benefit of VM resizing and server consolidation for energy conservation in addition to meeting the QoS defined in the SLA. The proposed framework consists of performance controller and an energy optimizer. The performance controller maintains the demanded performance through dynamic VM resizing. The energy optimizer consolidates VMs onto the most power efficient servers for power saving. Further, the authors proposed the ACO algorithm to place and consolidate VMs on virtualized servers. In their work, two metrics, application SLA (e.g., response time), and power consumption have been used to evaluate the performance of the proposed framework.

2.3.3 Miscellaneous Heuristic Approaches

The rest of the approaches which couldn't be classified into the above discussed heuristic categories, are listed under the miscellaneous heuristic approaches. These comprise general energy saving negotiation strategies for SLAs.

FIT4Green: FIT4Green project [113] has proposed an energy-aware computing framework. It comprises of energy saving strategies and policies. These strategies were packaged and used in the context of data center control frameworks. The developed plug-in has two parts: monitoring and controlling. The monitoring module continuously updates the status of meta-model instance that contains the attributes of the data center's machines. The controlling module searches for the appropriate optimal deployment actions to reduce the energy consumption. In addition, it also takes care of the present status, SLA and rules framed by the user/operator. The authors claim that with this proposed model, energy can be saved from 10 to 30 percent. Similarly, in [114], the authors have proposed a SLA aware framework, to operate cloud infrastructure in an energy efficient way. It is an integrated approach for VM migration, (re-)configuration and power management to minimize the energy consumption.

GreenIT Service Level Agreement: Another approach for Green SLA was developed by Laszewski and Wang [82]. It was focused on measuring and monitoring the eco-efficiency of green IT services offered by the data centre. GreenIT-SLA uses energy metrics to monitor the eco-efficiency of the offered services. The goal of GreenIT-SLA is to bring down the environmental impact of the data center operations at the hardware (i.e computing infrastructure and cooling system) and software (sophisticated energy efficient scheduling algorithms) level. For this, proper metrics are placed and used in SLAs such as Data Center Temperature & Humidity [115], Data Center Infrastructure Efficiency (DCiE) [116], Power Usage Effectiveness (PUE) [117], Data Center energy Productivity (DCeP) [118], Space Watts and Performance (SWaP) [119]. The authors emphasize adding these metrics and scales to assess the entire environmental impact of a given task or scientific experiment.

Resource-aware SLAs: Resource-aware SLA or Green Service Level Agreement [120] was an energy optimization approach that enhanced text-based SLA by including semantic information regarding metrics and behavior. It collects energy data on a per service request basis. The framework defines an SLA validation facility (SLA Validator) that checks the SLA compliance of provider's activities and triggers (counter-) actions. In addition, the GSLA also offers incentives to users for compromising on the performance parameters and metrics.

OPTIMIS: Rasheed et al. [121] project OPTIMIS toolkit focused on optimizing the usage of Cloud infrastructure services based on the parameters such as Trust, Risk, Eco-efficiency

and Cost (TREC parameters) specified in SLA. Further, the authors developed and evaluated an approach for service manifesto and SLA on multiple cloud architectures. This trust model is an evaluation process at the user level where cloud providers are selected based on high level of trustworthiness. This is useful for critical business application and more sensitive information.

Green SLAs for High Performance Cloud Computing: In [122], the authors have proposed a GSLA for high performance cloud computing service providers, in which each client specifies the minimum percentage of green energy that must be used to run the job. On violation of the GSLA, the provider is penalized. The authors have also proposed a power distribution and control infrastructure to support the GSLA. Further, Simulated Annealing and Linear optimization scheduling polices have been evaluated to seek profit maximization and predict green energy that would be produced in the future.

Plug4Green:Plug4Green has an energy aware VM placement algorithm which reduces power consumption and greenhouse gas emission [123]. The proposed framework deals with technical SLA and energy related constraints through constraint programming (CP). The proposed solution triggers two events: Single Allocation and Global Optimization. The Single Allocation event triggers on a new VM allocation. It maps the VM to a server while taking care of VM characteristics, current state of servers, SLA constraints and current objectives of data center in terms of minimizing the power consumption or CO2 emissions. The Global Optimization event executes itself regularly and produces data center reconfiguration plans such as switching on or off a server, migrating VM etc., as output.

2.4 Comparative Analysis of Green SLA Scheduling Algorithms

A comparison of Green SLA scheduling algorithms with different attributes is summarized in Table 2.3. The comparison primarily focuses on their platform, technology, QoS and the outcome pointed by the authors.

Table 2.3: Comparative analysis of Green SLA scheduling algorithms

Heuristics/ Meta-Heuristic	Platform/toolkit	Technology	QoS Parameters	Authors Conclusion/Remarks
Fit4Green 2010[113]	Cloud architecture-based plug-in			• Energy can be saved from 10 to 30 percent of a site.
Energy Saving Cloud SLA Negotiation - A PSO Approach (Copil et al. 2012) [81]	JADE (Java Agent Development Framework)	Java	CPU, Memory, Price, Energy	• Each negotiation between service provider and user reaches close to pareto frontier. The last negotiation process is one close to Pareto-optimality. • The nogotiation results high social welfare and maximizing the sum of the negotiation parties fitness function.
Resource-aware SLA (Bunse et al. 2012)[120]	WS-Agreement	XML	Eco-efficiency integrated with traditional SLA	• A typical service consumes energy 5.8J and in total it sums up to 8.56KWh to complete the execution of sequence of requests. By optimization, it can be reduced to 4.4J and 6.51KWh while keeping SLA violations quite low.
OPTIMIS (Rasheed et al. 2012) [121]	WS-Agreement	XML	Performance requirements in terms of Hardware, Trust, Risk, Eco-efficiency, Cost, Data location, Data security, Elasticity.	• Service menifest has been defined and managed electronically in a multi-cloud environment.
Modified best fit decreasing heuristic (Gao et al. 2013) [110]	Cloud infrastructure is configured with Xen virtualization technology	----	Response time, energy consumption	• It effectively handles VM and physcial machine hetrogenity for dynamic workloads. • The proposed resource management scheme achieve 50.3 % power saving as compared to static provisioned scheme with strict SLA gurarntees.

Continue…

Heuristics/ Meta-Heuristic	Platform/toolkit	Technology	QoS Parameters	Authors Conclusion/Remarks
Ant colony optimization- Local Search (ACO-LS) (Gao et al. 2013) [112]	Hetrogenous Xen-virtualized environment	---	Response time, energy consumption	• The proposed algorithm ACO-LS has been compared with First-Fit Decreasing algorithm (FFD) [124], an ACO algorithm without local search strategy (ACO-VC) [96] and a random placement algorithm (RPA). • ACO-LS saves energy 17%, 8% and 3% in comparison to the RPA, FFD and ACO-VC algorithms respectivly.
Simulated Annealing(SA) Linear Programming(LP) Static Green-aware Placement (SGP) First-Fit(FF) (Haque et al. 2013) [122]	Simulator based upon HPC Cloud model	---	normalized profit, normalized green energy use, number of admitted jobs, and number of Green SLA violations	• A comprision has been done among the SA, LP, SGP, FF against the parameters (normalized profit, normalized green energy use, number of admitted jobs, and number of Green SLA violations) • Optimization policies (SA, LP) outperform against the greedy policies (SGP, FF). • Choice of optimization policy depends upon cloud provider's preference choice whether it wants to accept more jobs or violate fewer Green SLAs.
Energy aware SLA (EASLA) 2014 [111]	Homogenious cluster	----	Makespan, energy	• The proposed algorithm saves 22.68% and 12.01% energy consumption as compared to Greedy DVS and Evenly DVS algorithms.
Plug4Green (Dupont et al. 2015)[123]	Hetrogenous cloud data center configured with VMWare ESX hypervisor	Java, XML	Hardware, QoS, security, energy	• The proposed solution has been verified on 23 SLA specifications and 2 energy policies. • It reduces power consumpton and CO_2 emission 27% and 23% respectivly. • It also addresses the issues of hardware hetrogenty, workload particularities and scalability.

2.5 Motivation

In large organisations such as public sector units, government sectors, academic institutes /universities etc., IT infrastructure is usually non-centralised. There are a large number of commodity computers with different configurations used in different office sections. This leads to underutilization of resources, cost overhead, unorganised resources, and many other problems as discussed in the next section.

Cost overhead: In many organisations, the traditional followed approach to allocate IT infrastructure is to distribute resources (desktop PCs, servers, software, storage, network equipment, etc.) among the different departments. Purchasing of common hardware and software repeatedly for different departments leads to many problems. It increases the procurement cost, management cost, and electricity cost, and requires deployment of more technical staff.

Underutilization of resources: Commodity computers are a large part of the IT infrastructure in organisations. These commodity computers are used by various types of users from different domains. These computers have different performance levels because of different configurations, and architecture which have been purchased in separate spans of time. Their usage level is also non-uniform like in universities or academic institutes, these computers are used by students/research scholars in labs and teachers/staff into different office sections. Now, one student might be using a computer with a very good configuration for web-browsing or word-processing/spreadsheet applications or any such minor small computation applications, whereas another student (research scholar) might be using an averagely configured computer to run their scientific or large file compilation programs, thereby using 90% of the resources. Thus one person does not get up to the mark performance and the other is just wasting resources.

Non-optimal energy usage of IT equipments: The main sources of energy usage in a data center are categorised into IT equipment (server, storage, network) and supporting facilities (power, cooling and lighting). These IT resources run 24*7 hours to cater computational and storage services to the users. Typical and unplanned usage of IT resources leads to significant energy consumption and emission of greenhouse gasses into the environment. The feature of virtualization and optimization of user time usage slots can save a considerable amount of energy.

Unorganized resources: The computer systems are placed in different office sections. Now, each user has his/her own preferences of operating systems, applications, and personal storage files. It involves a lot of time and technical man power to install, configure, and maintain the systems. Taking the scenario of universities, where the computer systems placed in the labs are shared by a number of students. Now, each student has his/her own preference of operating systems, software, and personal storage files. Now, either the systems/computers need to be configured with all kinds of applications used by students or the systems need to be grouped and configured for specific student needs. It requires lot of time and efforts to install, configure, and maintain the systems.

These issues mainly have been the motivation and driving force behind this thesis and provide the basis of problem formulation as presented in next section.

2.6 Problem Formulation and Objectives

Cloud computing is a heterogeneous computing system with a number of resources of different types. Mapping a set of tasks on to a set of resources using conventional methods becomes infeasible as a result of the dynamic nature of resources, different administrative policies and Quality of Service (QoS) requirements. Additionally, to that addressing the issues of energy efficiency without severe loss of performance make the whole process even more challenging. A new area of research has been setup to tackle this problem.

Many algorithms exist in literature based on energy efficient resource scheduling. However, each data center is unique because of its hardware and workload specificities. A data center to host web applications or business applications cannot cater the services to High Performance Computation (HPC) scientific and engineering applications. A data center manages VMs according to conventional and established rules while taking an account of workload characteristics. The design of framework and heuristics need to take an account the particularities of hardware and workload to provide the efficient resource utilization and additional energy savings. The proposed algorithms try to provision data center resources to client applications in a manner that improves the energy efficiency of the data center. It is based on the case study of the real problems, stated above, faced by the universities. Secondly, the existing research work fails to take the user interests into consideration, which is one of the important QoS parameters. Energy can be conserved through rounds of negotiation and by

inclining the user to strike a balance between energy and performance. So, the proposed algorithms differ in their objective from the existing ones.

Objectives:

I. To explore and analyze energy efficient scheduling techniques in the cloud computing.

II. To design energy efficient resources scheduling algorithm(s) for cloud computing. The parameter being considered for energy minimization will be like CPU, memory, disk storage, network infrastructure etc.

III. To design Green Service Level Agreement (GSLA) parameters based energy efficient scheduling algorithm(s) for cloud computing.

IV. To test and validate the proposed algorithms in cloud computing.

2.7 Summary

This chapter provides the technological aspect of the existing energy efficient resource scheduling and GSLA framework and algorithms. It analyses the existing energy aware resource provisioning frameworks and heuristics based on various aspects such as energy saving, virtualization, DVFS, SLA, deadline constraint, network, data centers, load balancing, platform, technology etc.

The next chapter presents the design of the proposed energy efficient resources scheduling algorithm to address the issues identified in problem formulation, and to fulfill the objectives of this research work.

Chapter 3

Proposed Energy-efficient Cloud Resource Scheduling (ECRS) Algorithm

---

The previous chapter discussed the accomplished work in the area of energy efficient resource scheduling and SLA based energy aware scheduling algorithms used for cloud environment. The study depicted that the challenge of energy conservation from the context of many organisation has not been addressed yet. To provide a solution to the problem discussed in the previous chapter, an Energy-efficient Cloud Resource Scheduling (ECRS) algorithm has been proposed and designed.

This chapter presents a set of algorithms to solve the problem of energy and performance efficient resource scheduling. Further, the statistical analyses of the results for the system behavior to infer potential benefits are discussed in chapter 6. This chapter starts with the formalization of the energy efficient resources scheduling algorithm, which divides the problem into four sub-problems. The chapter continues with the proposed energy efficient cloud service framework. It introduces various models such as the energy model, the reservation model and the prediction model, used in the design of the proposed solution. The chapter concludes with the design aspects and the pseudo-code of the proposed energy efficient resources scheduling algorithm, which helps in accomplishing the second objective of this research work.

3.1 Formalization of the Proposed Energy-efficient Resources Scheduling Algorithm

Energy efficient resource scheduling has a direct impact on cloud performance and cost. It should meet user expectations with respect to resource requirements and QoS, improve resource utilization, execution efficiency and energy conservation, and reduce operational cost and carbon footprints to the environment [125].

Resource scheduler is an important part of the Cloud Resource Management System, which gathers information about the available resources such as host utilization level, power consumption, number of VMs and their state, an estimate the resources required for future requests etc. With this input, it provisions the resources to user's request. Energy efficient resources scheduling primarily comprises two steps: step one, requires placing the new VM request to a host that meets user requirements and to minimizing the overall power consumption with this VM allocation; step two, involves performing VMs migration and consolidation to optimize the current VM allocation on a set of hosts. Energy efficient resources scheduling is divided into the following four parts:

I. Initial placement: Initial placement concerns with the new VM allocation. The host server is selected for the allocation of new VM based upon the VM characteristics, the current status of the servers, the SLA constraints, and the current objectives of data center in terms of minimizing the power consumption or CO2 emissions.
II. Source host selection for VM migration: To optimize the data center's workload, the workload optimization event triggers itself regularly and executes a data center reconfiguration plan (switching on or off a server, migrating VM etc.) as an output. For VM migration, the source host is selected as per data center's reconfiguration plan.
III. Destination host selection for VM migration: A destination host is selected where the VM consolidation action needs to perform. Again, the destination host is chosen as per the data center's reconfiguration plan.
IV. VM migration: VM migration is a selected VM or set of VMs that need to be migrated from the source to the destination host.

A detailed discussion of all parts is covered in section 3.3. The terms node, host and Physical Machine (PM) are used interchangeably in the rest of thesis.

3.2 Proposed Energy efficient Cloud Service Framework: ACA-Cloud

The proposed energy efficient cloud service framework, named ACA-Cloud, takes care of user VM admissions and provisions the data center's resources in an energy efficient way, with desired QoS constraints. It presents the architectural framework and various models used in the design of the proposed solution.

A number of independent users submit VM requests characterized by VM type (small, medium, large), operating system, execution time etc. Resource requirement depends upon the applications/software being used. The users negotiate with the resource provider through SLAs to formalize the QoS requirements, awards, penalties, service terms, energy reduction incentives and the cost model. Figure 3.1 shows the high level architecture of the energy efficient cloud system model. The detailed design and implementation of the framework has been discussed in chapter 5. In the subsequent sections, the energy model, the VM admission control policy, the prediction model and the Energy-efficient Cloud Resource Scheduling (ECRS) algorithm have been discussed.

3.2.1 Energy Model

The power consumed by the servers contributes to 75% of the total energy consumption of a data center [126]. Thus, the objective is to minimize the total power consumption of all the servers in a data center. There are various components in a server that consume power, such as CPU, memory, disk storage, system components, network interface and the power supplies. However, CPU consumes a significant power of a server, as per data provided by Intel Labs [52]. Therefore, CPU utilization rate is used to model the energy consumption of a server. There is a linear relationship between power consumption and CPU utilization [88, 89, 92, 127] as shown below [46].

$$P(U_{i.node}) = k \cdot P_{i.max} + (1 - k) \cdot P_{i.max} \cdot U_{i.node} \qquad (1)$$

$U_{i.node}$ is the CPU utilization of node i. $P_{i.max}$ is the maximum power consumed by the i^{th} node when the server is fully utilized; k is the fraction of power consumed by the idle server.

In a multi core system, the total utilization of the CPU of a server is the summation of the utilization of all the cores or virtual CPUs

$$U_{i.node} = \sum_{j=1}^{m_i} (U_{i.jvcpu}) \qquad (2)$$

$U_{i.jvcpu}$ is the utilization of virtual CPU_j *and* m_i is number of cores of the server i.

Therefore the total power consumption of all the servers in a data center is presented as

$$P_{Total} = \sum_{i=1}^{n} (k \cdot P_{i.\max} + (1-k) \cdot P_{i.\max} \cdot \sum_{j=1}^{m_i} (U_{i.jvcpu})) \qquad (3)$$

P_{Total} is the total power consumption of all the servers in a data center.

The energy consumption of a server depends on its power consumption function over a period of time. Therefore, the total energy consumption E of a data center is defined as follows:

$$E(T) = \int_0^T P_{Total}(t)dt \qquad (4)$$

3.2.2 Reservation Model

This model describes how reservation requests of users for VMs are made and allocated. In the proposed framework, user requests are classified into two categories: high priority user requests (teachers) and low priority user requests (students/researchers). A user submits a VM request in the form of the VM's configuration, operating system, submission date/time and the execution time. All the VM requests are submitted via the following procedure:

I. VM requests submitted by a teacher are stored in a Resource Allocating Queue (RAQ), a high priority queue. A decision on the number of VMs and their types is made according to the students' strength and the type of applications required by them to conducting the laboratory class.

II. A Resource Waiting Queue (RWQ) takes VM requests submitted by students or researchers for their practice/research work. Researchers generally require large VMs to run their applications; for example physics scholars use Quantum Molecular Dynamics (QMD), Isospin dependent Quantum Molecular Dynamics (IQMD), Boltzmann-Uehling-Uhlenbeck (BUU) and Statistical Multifragmentation Model (SMM).

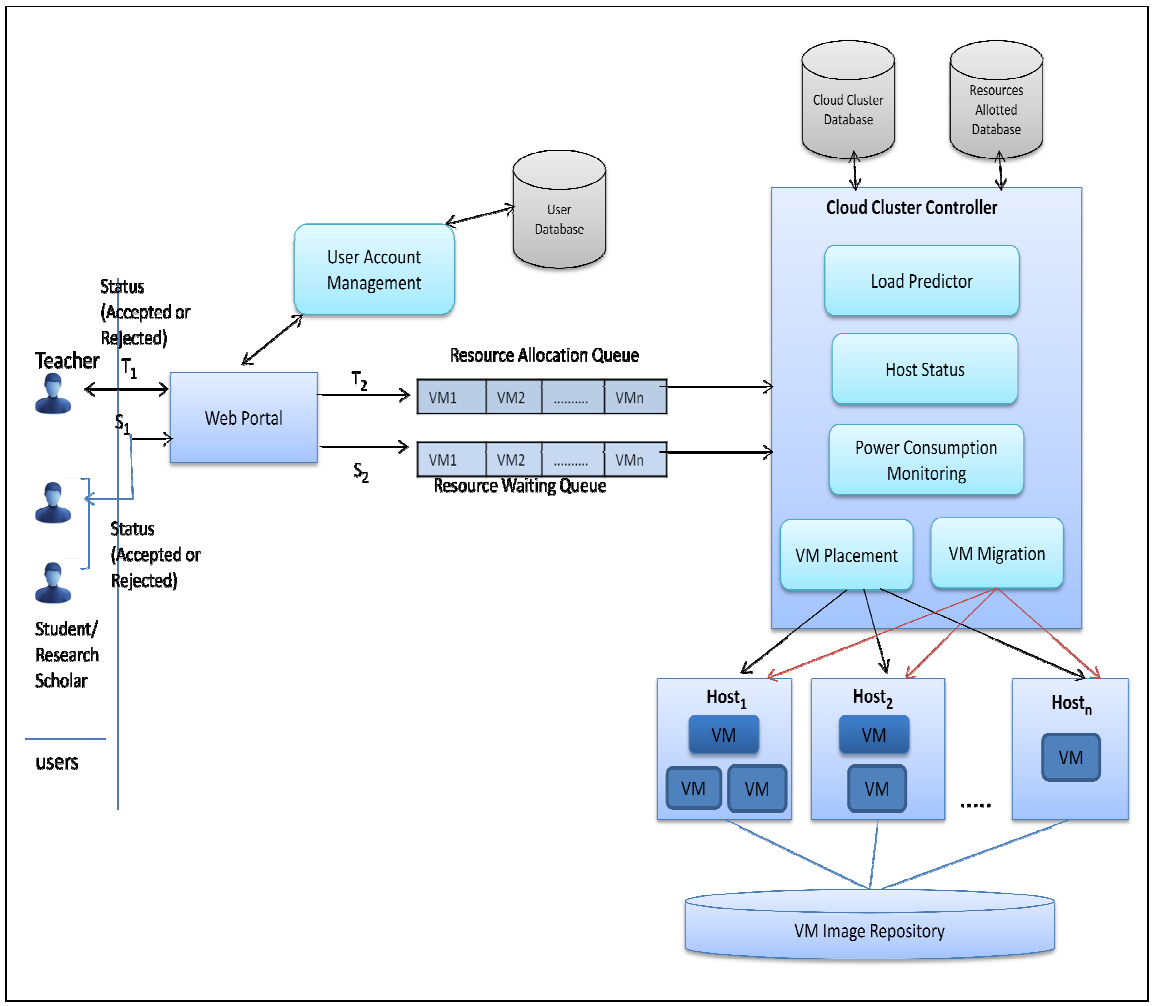


Figure 3.1: High level architecture of proposed energy efficient cloud service framework

Initially, all user VM requests are stored in either RAQ or RWQ as shown in Figure 3.1. Once a request is approved by the cloud admin, it is duly notified to the user. There are two scenarios of resource allocation:

i. On a student's request, if the requested resources are available, resources are allocated.

ii. In case of unavailability of resources, there are two options:

   a. The available time slot nearest to the user's requested time slot will be allocated to him. or

   b. The user is granted a time slot for the weekend.

The objective of the scheduling algorithm is to aggregate all the reservations into one part of the slot, from big free time slot, and switch off resources in the other parts of the time slot

[104]. The detailed discussion of each component of the proposed energy efficient cloud service framework is covered in chapter 5.

3.2.3 Prediction Model

Prediction helps to estimate the required number of resources for the current workload plus a margin of the growth. It guides the on-off cycles of the hosts. It calculates the required number of resources for the current load based on the analysis of the previous day's demand during the same period plus a margin of the growth. The numbers of resources are estimated based on the average of the previous day's resource requirement during that time frame.

$$R_e = R_{req} + 20\% \text{ of } (1/6 * \sum_{i=1}^{6} R_i) + R_{feedback} \qquad (5)$$

R_e is the estimated number of resources reserved to cater to the present requests plus a margin of the growth.

R_{req} is the resources required as per the forthcoming VM requests from the load predictor

R_i is the resources reserved in the previous six days during that time

$R_{feedback}$ is a corrective factor, calculated based on the three previous errors. It allows the algorithm to take the measurement of load variations.

$$R_{feedback} = 1/3 * \sum_{i=1}^{3} Err(i) \qquad (6)$$

Err(i) is the error in estimating the required resources which is (R_a-R_e).

R_a is the actual resources required.

The accuracy of the prediction for a set of active hosts to address current and future resources needs are essential for optimal power consumption. An incorrect combination of active and inactive hosts leads to wastage of energy in rebooting the hosts and a unavailability of resources, which ultimately results in loss of performance and SLA violations.

3.3 Proposed Energy-efficient Cloud Resource Scheduling (ECRS) Algorithm

As discussed in section 3.1, the proposed ECRS algorithm focuses on initial VM placement and optimizes the current workload on the hosts with the energy conservation perspective. Also, the issue of VM migration overhead has been addressed. The next section discusses energy efficient VM placement and migration along with their pseudo-code.

3.3.1 Initial VM Placement

As the new VM request comes to a cloud system, there is need to place the VM on a host. VMs are mapped on hosts based on host ranking. A source host with the highest number of VCPUs, greatest power and performance efficiency, has the highest rank. The idea is that the number of resources are estimated (current plus a margin of the growth) and reserved in a decreasing order, starting from a host with a greater number of VCPUs onto ones with fewer VCPU numbers. A 16 core host accommodating 13 VMs consumes less power compared to switching on two 8 core hosts or four quad core hosts. Also, hosts with fewer numbers of VCPUs have a lower chance of full load as the requested VMs have already been accommodated by hosts with greater number of VCPUs; thus, there is a higher chance to low-power modes for hosts with small number of VCPUs. The complexity of the algorithm is m*n, where m is the number of hosts and n is number of VMs. The pseudo-code for the algorithm is presented in Algorithm 1. The acronyms used in the algorithms are listed in Table 3.1.

Table 3.1: Acronyms used in the algorithms

symbol	meaning
PM_{active}	A set of active physical machines/servers
$PM_{inactive}$	A set of inactive/sleeping physical machines
VM_{system}	User VM request need to allocate by the system
$pm_{i.inactivitytime}$	Physical machines i's inactivity time is a time during which the physical machine remains idle. Through resource estimation the required active and inactive PMs are determined.
$pm_{i.therasoldtime}$	Physical machines i's switching threshold time is the time interval when energy consumption of two possible cases are equal a) the resource is switched off, stays off for a while and then switches it on again, b) the case where the resource stays idle for the entire interval.
$VM_{j.remainingcompletiontime}$	VM's remaining completion time is the time left to complete the VMs execution from the present time. it is $VM_{j.duration}$- ($VM_{j.presenttime}$- $VM_{j.startedtime}$)
$VM_{j.migrationtime}$	Time taken to migrate a VM from one physical machines to another.

Algorithm 1: Energy efficient VM Placement

Input: Set of PM, VM **Output:** VM mapping on PM

1. Assign rank to each PM, a PM assigned higher rank having more number of VCPUs.
2. for all $vm_i \epsilon VM_{system}$ do
3. for all $pm_j \epsilon PM_{active}$ do
4. pm_j.Status()
5. If (pm_j *has enough resources for* vm_i) then
6. $pm_j \leftarrow vm_i$
7. VM=VM- {vm_i}
8. break // breaks the loop on successful mapping of vm to pm
9. End if
10. End for
11. if (vm_i= NULL) then
12. for all $pm_j \epsilon PM_{inactive}$ do //When vm_j didn't get resources from PM_{active}
13. If (pm_j *has enough resources for* vm_i) then // Activate pm_j which can fulfill vm_i request
14. $pm_j \leftarrow vm_i$
15. VM=VM- { vm_i}
16. break
17. End if
18. End for
19. End if
20. if (vm_i= NULL) then // when vm_i neither get resources from PM_{active} nor $PM_{inactive}$
21. UnallocatedVM[k] $\leftarrow vm_i$
22. INCREMENT k
23. End if
24. End for

3.3.2 Optimization of Current Workload

Second function of the ECRS algorithm is to assess the current load plus a margin of the growth and make decisions for VM migration so that there is optimal energy consumption. VM migration involves three components a) Source host selection for VM migration b) Destination

host selection for VM migration c) VM migration. The following section discusses each part. The pseudo-code for the algorithm is presented in Algorithm 2.

Source host selection for VM migration: The second part of the energy saving strategy is VM consolidation on some hosts through VM migrations. A source host which has minimum VM running is chosen for VM migration. A host with minimum VMs has the maximum possibility to switch off. Besides, there is a need to check whether a host is operating in a safe band of workload. Every host has an upper and a lower threshold of load it can carry. A host operating below the lower threshold results in wastage of power on underutilized resources. If a host crosses the upper threshold, it affects the VMs performance and hampers the SLA. So each host adopts MM 40-80% policy [46] to strike a balance between energy consumption and SLA violation. If the upper or lower threshold load limit is violated, the VM migration from that source host takes place.

Destination host selection for VM migration: A host with the maximum number of cores or VCPUs is selected as a destination for the VM migration. If a host with the maximum number of cores is unable to accommodate the VM, the host with the next best number of cores that can accommodate the VM is chosen.

VM Migration: When a host is operating in lower migration region, the possibility of migrating all of the running VMs on that host is assessed so that it can be switched to a low power mode. Before proceeding with the VM migration, each VM completion time is checked from the source host. If a VM execution is near its finishing time, the algorithm allows the completion of its finishing time to avoid migration overhead. A VM migrated from a host operating in upper migration region brings the host in safe mode. Figure 3.2 illustrates the migration and safe operating mode of a host.

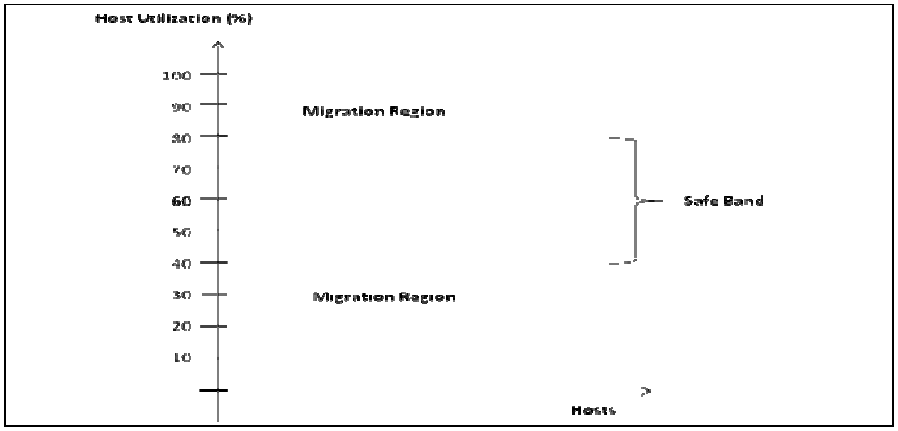


Figure 3.2: VM Migration band for a host

Algorithm 2: Energy efficient VM Migration

Input: PM list **Output:** VM migration list,

1. Arrange all $pm \in PM_{active}$ in ascending order of VCPUs
2. for all $pm_i \in PM_{active}$ do
3. if ($pm_{i.inactivitytime} > pm_{i.therasoldtime}$) then
4. for all $vm_j \in pm_i$ do
5. if($vm_{j.remainingcompletiontime} < vm_{j.migrationtime}$)then
6. *No vm_j migration from pm_i as if vm_j near to completion*
7. else
8. *vm_j are migrated to PM_{active} which has highest rank and can accommodate a new migrated vm_j*
9. end if
10. end for
11. If ($pm_i.allocatedVMs$ = NULL) then
12. $PM_{inactive} \leftarrow pm_i$
13. end if
14. else
15. *pm_i remains idle at power on mode*
16. end if
17. end for

3.4 Summary

This chapter presents the design of the resource scheduling algorithm, ECRS, as an outcome of the research work. It focuses on VM placement and optimization of the system workload on the physical machines with the energy conservation perspective. Initially, a formalization of energy-efficient resources scheduling algorithm has been discussed. Afterwards, the detailed discussion of the various system models of the energy efficient cloud service framework has been presented. At last, the algorithm, its mechanism, and the pseudo-code have been explained. The next chapter provides the design and detailed discussion of the proposed Green SLA aware cloud resource reservation algorithm.

Chapter 4

Proposed Green SLA aware Cloud Resource Reservation (GSLACRR) Algorithm

---

The previous chapter presented a set of algorithms for energy efficient resource scheduling at the data center level. Further, the various models have been introduced and discussed in detail to achieve the goal of energy efficiency and performance of the cloud environment.

This chapter focuses on energy reduction at the SLA level. Cloud resources are provisioned with the proposed algorithm which inclines the users towards cloud eco-system services. It attracts more number of users and also offers profits to both the users as well as the cloud service provider. This chapter includes a Green Service Level Agreement (GSLA) based resource management, GSLA template and negotiation strategies that take into account time, power consumption and trade-off between QoS parameters. The chapter concludes with the working, the pseudo-code and benefits of the proposed GSLA algorithm, which helps in accomplishing third objective of this research work.

4.1 Introduction

Over the last few years, extensive research has been done in the area of Service Level Agreement (SLA) for cloud computing. SLA is a formal contract between the user and the cloud service provider regarding the service quality expectations [128]. The fundamental issue is the management of SLAs and trading off multiple QoS parameters by tilting users towards green computing. Involving users in sustainable computing is a challenging problem because of the dynamic nature of user requests as well as the heterogeneous nature of resources.

Many Green SLA mechanisms have been proposed in the literature of cloud environments. However, early research on Green SLA was focused on energy contracts [120, 122] and measuring the eco-efficiency of green IT services [82]. In the commercial computing environment, there are other crucial QoS parameters those are considered such as reliability, trust, risk, security and cost [121]. In contrast to the discussed studies, this work focuses on two areas; firstly, the GSLA based negotiation that inclines user orientation towards sustainable computing, and secondly, the integration of the GSLA approaches with energy-efficient scheduling at the data center level. Thus, it increases the resources acceptance and also minimizes the energy consumption.

The subsequent sections comprise the following:

- GSLA-based resource management
- An Energy Based Resource Provisioning Policy
- GSLA aware Cloud Resource Reservation algorithm

4.2 Proposed Green Service Level Agreement (GSLA) based Resource Management

The functionality of GSLA based resource management is shown in Figure 4.1. A user can submit their request for cloud services through the web portal. The GSLA management and resource allocation layer is responsible for GSLA negotiation and resources allocation. This layer interacts with the hypervisor and is responsible for VM management and its deployments. A physical cloud resource lies at the foundation of the cloud model.

The GSLA management and resource allocation layer negotiates with the user about the requested resource services and allocates resources to them. It consists of several components: Admission Control, Negotiation/Renegotiation, Accounting, Final Agreement, Load Prediction, Resource Monitoring, Scheduling and Dispatching. The Admission Control interprets the users request and determines the acceptance/rejection of the request. The Negotiation/Renegotiation module negotiates the resources and the eco-system of the offered services as discussed in section 4.2.1, with the user. The user's account and its credits are managed by the Accounting component. Once the negotiation process is over, a final agreement copy is produced to the user on his/her consent. Load prediction determines the number of incoming requests and resources are estimated based on the present request status plus a margin of the growth, as discussed in the prediction model in chapter 3. The resource status is assessed by Resource Monitoring in terms of the number of running VMs, pending VMs and available resources. The Scheduling mechanism decides how to map the VM request on to the resources with energy efficiency, using algorithm as discussed in chapter 3. Dispatching starts the requested services on the allocated resources.

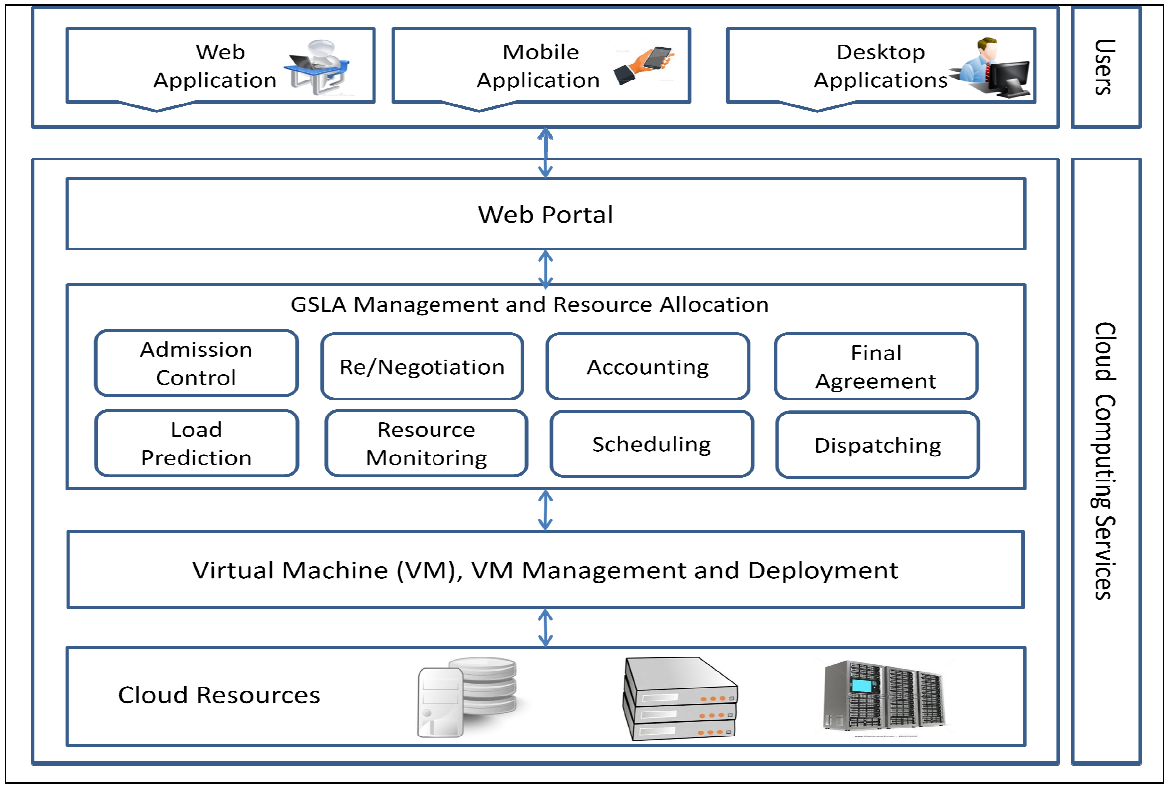
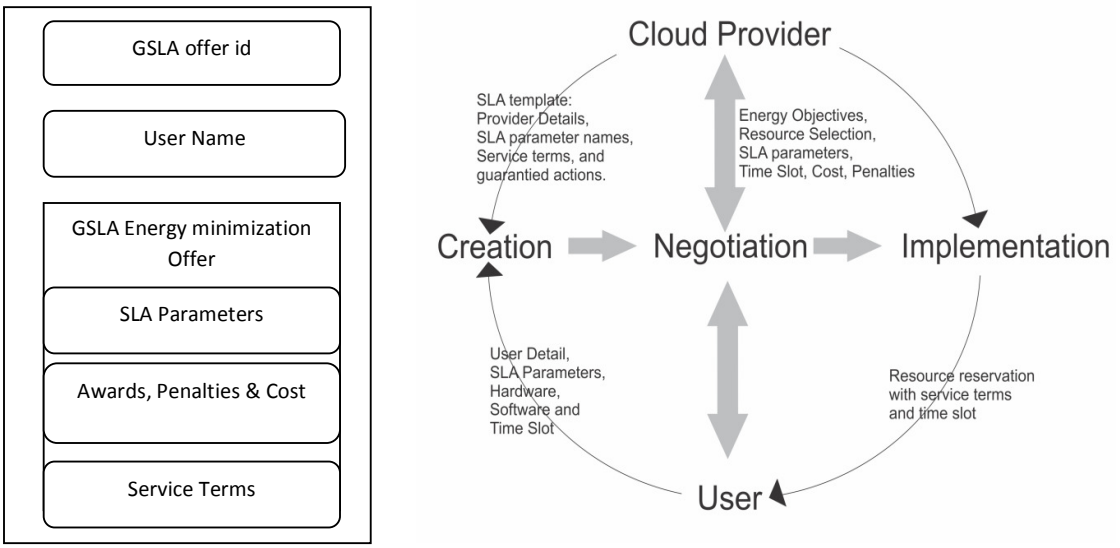


Figure 4.1: Proposed GSLA-based resource management



a) GSLA Template

b) GSLA Workflow

Figure 4.2 Proposed Green Service Level Agreement

4.2.1 Energy Based Resource Provisioning Policy (EBRPP): A Negotiation Approach

EBRPP has been designed on the basis of the requested resources, time slot and energy consumption parameters. When a user submits a request, the resource manager checks the availability of resources, if available; a further negotiation takes place on the basis of the SLA parametric values and the energy minimization policy, as discussed below. In case of non-availability of resources, a possible time slot for resource availability is negotiated. The main objectives of EBRPP are to provide resources as well as to minimize energy consumption. A negotiation comprises the following points:

i. Resource Reservation: As the resource reservation request comes from the user, resources are reserved for the demanded time slot based on the availability of resources. If the scheduler cannot make the reservation for the requested time slot, users will have two options; (a) a time slot closest to the user's time slot will be allocated to him, or (b) he is granted a time slot for the weekend. The objective of the negotiation is to aggregate all the reservations into one part of the slot from the big free time slot and switch off resources in the other part of the time slot [129].

ii. Credits: There is a credits system. Credits are added to the corresponding user's account who agrees to the resource usage time slot and performance modifications through negotiation. If a user agrees to work on during a specified time slot, the user will get 1% credits in his/her account.

iii. VM running mode: Generally specified resources for VM are overestimated, based on the assumption of peak workload. Moreover, some tasks use resources and after a while resources remain idle and still consume power. So there is a negotiation between cloud service provider and the user about the sleeping time of VM when it remains idle. When there is no interaction with the VM for a specified number of minutes, VM will be automatically switched to sleep mode. In the same way, it will take at least some amount of time to restore when user again resumes the VM.

iv. Power budget policy: The cloud provider offers a power budget policy specifying the maximum usage of power by the requesting user. When the user used resources reach a specific power limit (like 80%), a warning message will be generated for the user so that s/he can manage her/his work accordingly. This is optional, as most of the users are not concerned with the specification of the power budget in advance.

Apart from the above mentioned points, some of the standard performance parameters included are:

a) Availability of data: The requested data should be available 99% of the times.

b) Uptime of Resources: Specifies the uptime of the VM with special provision of point 3 from the above mentioned negotiation points.

c) Disaster Recovery: In the event of an untoward incident, the mean time for the recovery of the resource services.

d) Elasticity: It specifies the size to which a resource can grow.

e) Cost: It specifies the cost of the offered services with respect to time and resource usage.

Additionally, other SLA parameters like security, privacy, location of data, portability of data, legal issues, dispute process, exit strategies etc., can be included It all depends upon the

criticality of data and the applications. As a private cloud is deployed inside the premises of an organization, these parameters are not considered in the proposed framework.

4.2.2GSLA Workflow

The GSLA workflow comprises of the sequence of systematic activities between a user and the cloud administrator. The GSLA template and its workflow are shown in Figure 4.2 (a) and Figure 4.2 (b), respectively. A GSLA template is an extended version of the traditional SLA with the inclusion of an energy saving offer. The GSLA workflow consists of following steps:

Creation: GSLA creation takes input of the user's details, metric values for every SLA parameter, hardware, software, and time slot requirement.

Negotiation: A negotiation offer deals with various time slots, cost, awards, penalties, service terms, energy saving incentives, and alternative hardware configuration to users with requisite software.

Implementation: As per negotiation, either deny the request or reserve the resources and time slot with a user id.

4.3 Proposed GSLA aware Cloud Resource Reservation (GSLACRR) Algorithm

In this section the GSLACRR algorithm, that analyses whether a new VM request can be accepted or rejected based on the resource availability, has been presented. The input of the algorithm is the new VM request by the user and the output is the admission control with acceptance/rejection decision. As the new VM request comes from a high priority user (Teacher), the availability of resources is checked for the requested VM's execution time. On availability of resources, the request is handled by the system request handler and the user is sent an acceptance notification. On unavailability of resources, a rejection status is conveyed. The VM request from low priority users (students) is (re)negotiated based on the QoS parameters and the eco-system services, as stated in the aforementioned section 4.2.1. On successful negotiation, it is passed to the system request handler and an acceptance notification is sent to the user. If the negotiation is unsuccessful, a denial message is communicated to the

user. The acronyms used in the algorithm are listed in Table 4.1. The pseudo-code for the GSLACRR algorithm is presented in Algorithm 3.

Table 4.1: Acronyms used in the GSLACRR algorithm

symbol	meaning
UR_{HP}	User Request from high priority user
UR_{LP}	User Request from low priority user
pm_j	Physical Machine j
vm_i	Virtual Machine i
RH_{system}	System request handler

Algorithm 3: GSLACRR

```
1.  Input: New VM Request
2.  Output:  Admission Control
3.  for vm_i∈ ( UR_HP∪UR_LP) do
4.      if (vm_i∈ UR_HP ) then
5.       for all pm_j∈ PM  do
6.            pm_j.Status()
7.            If (pm_jhas enough resources for vm_i) then
8.                 RH_system← vm_i
9.                 SEND (accepted status to a user)
10.           else
11.                SEND(rejected status to a user)
12.           end if
13.       end for
14.     end if
15.    if (vm_i∈ UR_LP ) then
16.        EBRPP_NEGOTIATION ( UserID, time_slot, cost, awards, penalties,
    energy_saving_incentives,  service_terms)
17.        if (negot_res = null) then                  // Negotiation unsuccessful with a user
18.           round=0
19.           while (negot_offer_last) do            //  Rounds of alternate negotiation offer
20.                ALTERNATE_OFFER(UserID, time_slot, cost, awards, penalties,
    energy_saving_incentives,  service terms)
21.                round = round +1
22.           end while
23.           if (negot_res = true) then             // Negotiation successful with a student
24.                RH_system← vm_i
```

```
25.                 Energy efficient resource management (Algorithm 1 and Algorithm 2) to fulfil the
    contract
26.                 SEND (accepted status to a user with agreement details)
27.                 MONITOR( agreement )
28.                 EVALUATION_AGREEMENT(UserID)
29.                 if (agreement_violation = true) then
30.                     UPDATE_GSLA_REPOSITORY (UserID, credits)
31.                 end if
32.             else
33.                 SEND(Rejected status to a student)
34.             end if
35.         end if
36.     end if
37. end for
```

4.4 Advantage of the Proposed (GSLACRR) Algorithm

The proposed GSLA aware Cloud Resource Reservation (GSLACRR) provides several advantages over traditional SLAs for cloud computing. Some of them are list below:

Energy Conservation: The proposed algorithm inclines the users to strike a balance between energy conservation and performance. Through the negotiation and reservation mechanism, all the reservations are aggregated into one part of the slot from the big free time slot and resources are switched off in other parts of the free time slot [129]. The aggregated reservations increase the length of resource idleness. Hence, instead of the PMs are powered on all the time for partial workload, these are powered on in one part of the slot from the big free time slot and remain in power sleep mode in other parts of the time slot.

Usage of more resources: If the user agrees with the resource usage time slot and performance modifications, in lieu of it the user has an incentive to use more resources, with corresponding credits being also added to his/her account.

Increase lifetime of physical machines: A large number of cores PMs are always fully loaded or near-fully loaded, hence activated all the time. This continuous overusing reduces the reliability and life time of PMs [46] which in turn increases the chances of failure. This situation is handled by accommodating VM requests by low ranked PMs when the load is less on weekends or on off working days.

Increase acceptance rate: In traditional SLAs, on a user's request if resources are available for the requested time slot, those are allocated otherwise a rejection message is conveyed to the user. No efforts are put up to pursue the user for other available time slots. For experimental and batch kind jobs users can adjust resource availability time slots. Through negotiation users agree to various time slots to get incentive in terms of the privilege to use more resources or monetary benefits. It ultimately enhances the user acceptance rate.

Reductions of CO2 footprint: The carbon footprint is greatly influenced by the energy sources used [77]. Decreasing energy/power consumption through user's energy-awareness has a great impact on the environment. Reduction in the electricity demand of clouds contributes to reducing the carbon footprint in the environment and a greater sustainability to the ICT industry.

Cost savings: The proposed Energy Based Resource Provisioning Policy has immense potential as it offers significant cost savings at the user level and power saving cost at the data center level. There is still room for improvement on cost saving by working on the cost model and increasing the user's energy-awareness.

4.5 Summary

In this chapter, Green SLA based resource management, an Energy Based Resource Provisioning Policy (EBRPP), and the proposed GSLA aware Cloud Resource Reservation (GSLACRR) algorithm has been presented for a IaaS cloud. The aim is to minimize the energy consumption of a data center. Green SLA devises the provision of resources through negotiation with the users. The proposed GSLACRR algorithm leverages the free time slots and free weekend days, and adjusts the users request in these slots through negotiation. The next chapter provides the design and implementation of the proposed algorithms discussed in the previous and this chapter.

Chapter 5

Design and Implementation of Proposed Algorithms

The previous chapters have presented and discussed in detail, energy efficient algorithms for cloud computing. To demonstrate the working and usefulness of key algorithms, a cloud framework, named ACA-Cloud, has been designed and implemented. This chapter presents the architecture and implementation of the framework that includes the algorithms proposed in previous chapters.

This chapter starts with the discussion of the system architecture and the working model of each component. To design the structural components and their behaviour, various Unified Modeling Language (UML) diagrams have been prepared and discussed. It helps to analyse the requirements and visualize the design, and further validate the architectural design of the framework. Finally, the chapter outlines the details of implementation and the used technologies.

5.1 System Architecture

There are four high-level components in this hierarchical system. Figure 5.1 shows the different modules of the proposed system and their interactions. The next section explains the functionality of each component.

5.1.1 Interface Services

The Interface Service facilitates the users to request, reserve, negotiate, and monitor the resources. A user can access the portal as a cloud user or a cloud admin. As a cloud user, the user can register, request, provision and negotiate the resources. A cloud admin can authenticate new users, manage user accounts, monitor and get access to cloud resources. There is a Green SLA negotiation with the resource requesting user. After negotiation, a request can be denied or resources can be reserved within a time slot. Green SLA and EBRPP have already been discussed in chapter 4 and further, a negotiation process among the different entities is presented in sequence diagrams in Figure 5.10 and Figure 5.11.

5.1.2 Account Manager

The Account Manager provides authentication modules to handle different account types and authentication. Modules in this package handle the account validating credentials and store the account information in the user database. It allows to performing a number of user operations like creating, activating, deleting, and updating user accounts.

5.1.3 Cloud Cluster Controller

The Cloud Cluster Controller (CCC) is responsible for monitoring the status of all the physical machines/hosts and making appropriate resource management decisions in response to the present workload and incoming users requests. The functionality of various system components are explained below.

(a) VM Handler

The VM Handler (VMH) is invoked when a request for a new VM is received. VMH finds the destination host w.r.t energy and performance efficient ranking of hosts, and the accommodating capacity of the new VM request.

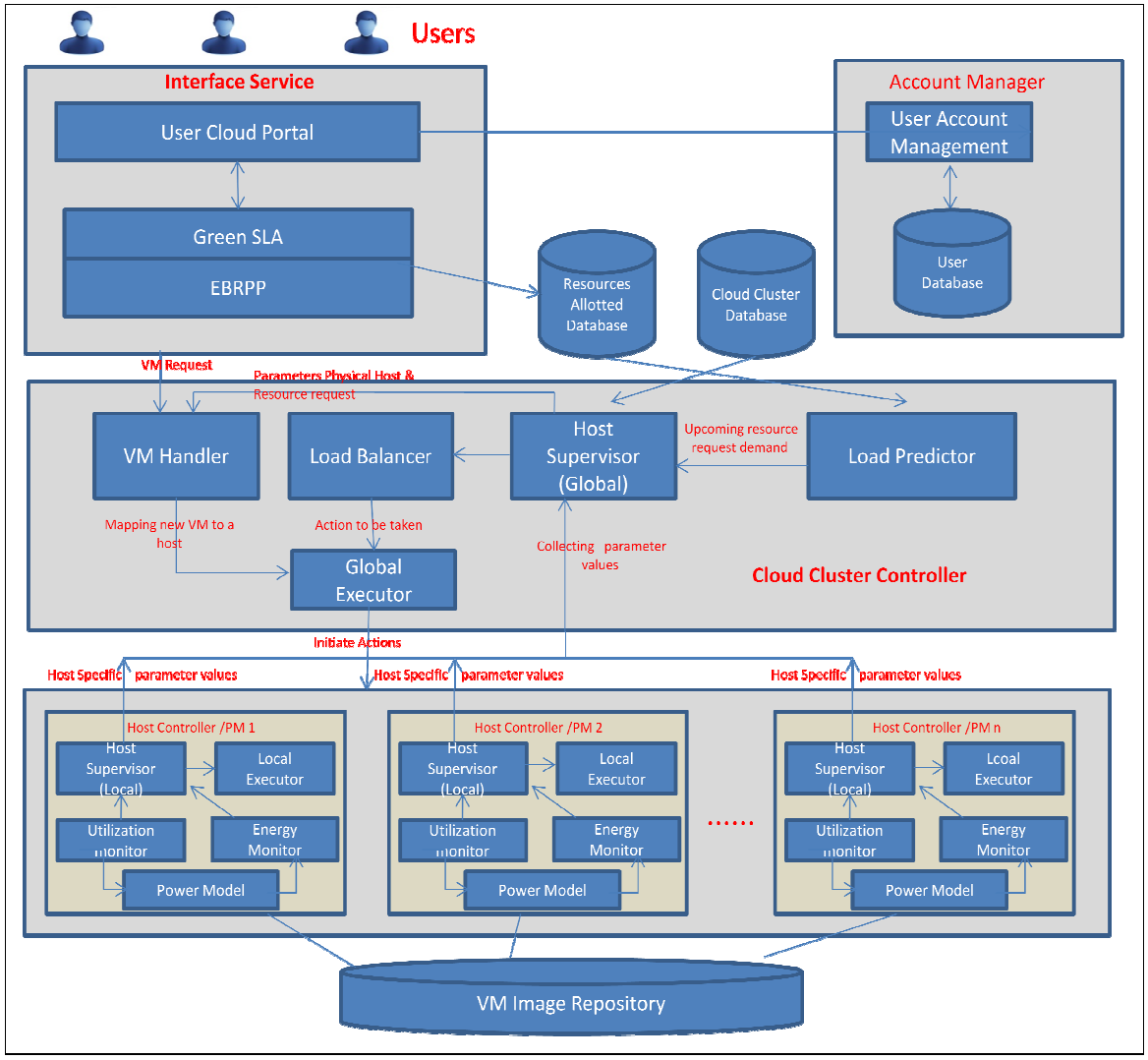


Figure 5.1: Architecture of proposed ACA-Cloud framework

The resource utilization of a new VM in general is assumed to be equivalent to the average resource utilization for small and medium VMs. High utilization is assumed for large VMs as these are generally used by research scholars for their scientific experiments. VMH gets the load status of present hosts, finishing time of currently running VMs, and estimates the forthcoming VM requests from Host Supervisor Global. Then according to the VM allocation policy, a new VM request is placed to $Host_i$ based upon energy and performance of host ranking, its accommodating capability, and the finishing time of other running VMs on destination host. VMH sends information of a new VM and destination host to Global Executor.

(b) Host Supervisor Global

The Host Supervisor Global (HSG) is an overseer module which periodically collects data from various hosts and the Load predictor. HSG analyses the current state of the physical hosts, their power consumption, and number of running VMs on each host, and gets forthcoming resource demands from the Load predictor. Besides this, it retrieves and updates each host information into the cloud cluster database like CPU mode, number of CPU(s), CPU frequency, CPU socket(s), Core(s) per socket, Thread(s) per core, NUMA cell(s), Memory size, available/allocated CPUs, IP, MAC address, number of running VMs etc. It further communicates information to the Load balancer and VM Handler.

(c) Load Balancer

The Load balancer (LB) is an intelligent module which assesses the present load plus a margin of the growth and takes decisions of load distribution so that there is optimal energy consumption. It estimates the number of resources needed to reserve and maintain a set of active/inactive hosts to cater to present requests and a margin of the growth, as discussed in the prediction model in chapter 3.

This module decides the source hosts and destination hosts for VM migration and performs VM migrations. It gathers various parameters from HSG, makes the decision and issues necessary commands to Global Executor to shift the load from source host to destination host and lower the power state of source machines. Class diagram and activity diagram of this module are explained in sections 5.2.2 and 5.2.3, respectively. Besides estimating the number resources for any time, there is a need to check whether a host is operating in the safe workload range. Every host has an upper and a lower threshold load that it can carry. If this limit is crossed then it affects VM's performance and hampers the SLA. So each host adopts MM 40-80% policy [46] to strike balance between energy consumption and SLA violation. If the upper or lower threshold load limit is violated then LB makes the decision to shift and balance the load.

(d) Global Executor

The Global Executor (GE) is connected directly with all the hosts in the cloud. This module is responsible for performing all the actions communicated by the VM Handler and the Load Balancer. Figure 5.2 general protocol used by VMH and LB while communicates with GE.

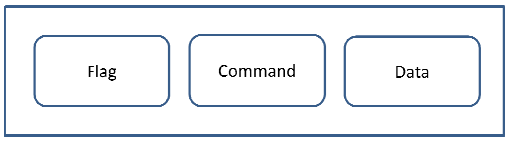


Figure 5.2: Communication protocol VMH-GE and LB-GE

Flag: The flag indicates the successful/failure execution of a GE.

Command: The command indicates the action (start, suspend, resume, save, restore, shutdown, reboot, destroy, migration VMs or pm-suspend, pm-power off, pm-hibernate, Wake On LAN etc.) needed to be performed.

Data: Data means source host, destination host, VM id, virtual networks etc.

5.1.4 Host Controller

Host Controller (HC) provides actual resources to a user's VM and communicates with the cloud cluster controller. The main components of HC are servers, workstations, storage, networks, etc. The functionality of each component in a host controller is as follows.

(a) Host Supervisor Local

The Host Supervisor Local (HSL) resides on each host to gather parameters at host level. It collects utilization, power consumption, information of a host, number of running VMs and its other information like VM Names, operating system type, state, CPUs, CPU time, maximum memory, used memory, security model, security DOI, security label, number of cores occupied by each VM, VM state, priority, VM's execution time etc. HSL provides information periodically to HS Global. It also gives necessary instructions to Local Executor to manage VMs running on a host and performs actions on local host such as suspend, power-off etc.

(b) Local Executor

The Local Executor (LE) is responsible for carrying out actions as communicated by HSL such as connect Hypervisor, start VM, suspend VM, resume VM, save VM state, shutdown VM, reboot VM, destroy VM, migration VMs, get virtual network list, get dom id, host suspend, host power off etc.

(c) Utilization Monitor

The Utilization Monitor is a periodically invoked module by HSL. It provides information of the percentage of host utilization.

(d) Energy Monitor

The Energy Monitor provides the power consumption information of a host. The energy consumption of servers depends on its power consumption function over a period of time. Therefore, the total energy consumption E of a data center is defined in equation 4, as discussed in chapter 3 as part of the energy consumption model.

5.2 Design Details of ACA-Cloud

The first step of designing is to assess the requirements, identify the boundaries and interactions among the system components and users. To design the structural components and their behaviour, Unified Modeling Language (UML) [130] has been used. UML is a general-purpose development modeling language that provides a standard way to develop an abstract model of a system. It helps to visualize and explore the system components from different prospective with varying degrees of abstraction. In this section, the various use case diagrams, class diagrams, activity diagrams, sequence diagrams and state diagram are shown to provide the finer details of the system.

5.2.1 Use Case Diagrams

Use Case diagrams help to analyse the functional requirements of the system. In Use Case diagrams, there are actors and ovals which represent entity and use-case, respectively. Figure 5.3 shows the Use Case diagram for user authentication and management. In the first use case, user's registration, login credentials and administrator management have been demonstrated. Once the user's credentials are verified, the user can submit resource requests and monitor the status of provisioned and pending resources requests as shown in Figure 5.4. The administrator can monitor the status of the cloud cluster and forthcoming VM requests.

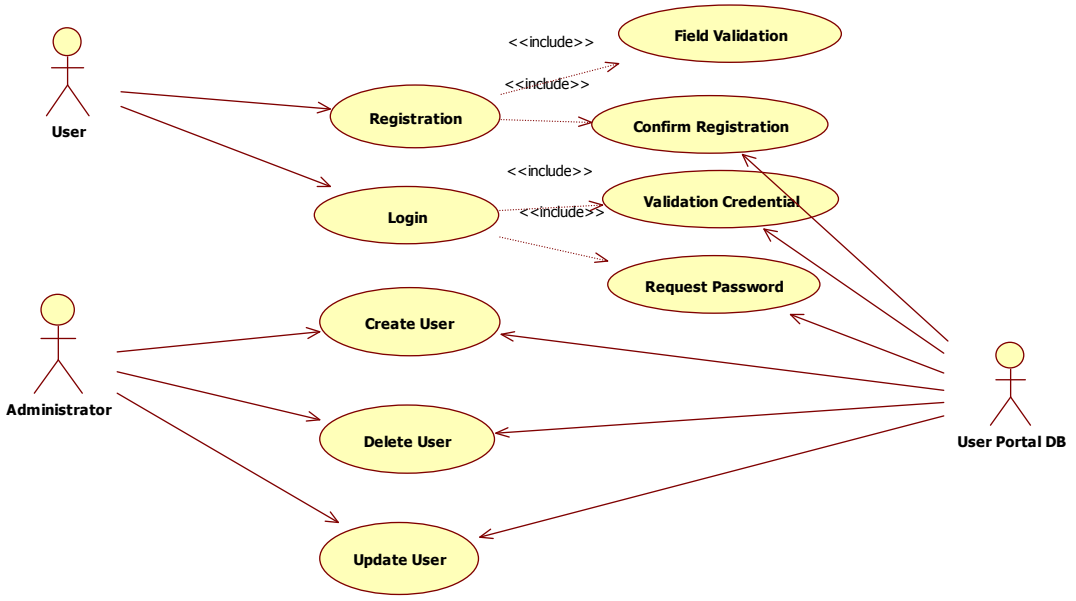


Figure 5.3: Use Case Diagram for user authentication and management

Figure 5.4: Use Case diagram for resource provision

5.2.2 Class Diagram

A Class diagram helps to analyse the structural requirements of the system. Figure 5.5 shows the structure of the system by various system classes and the relationship along with their cardinality. Figures 5.6 and 5.7 present two classes with their attributes and methods. The main set of classes details are covered in Appendix A.

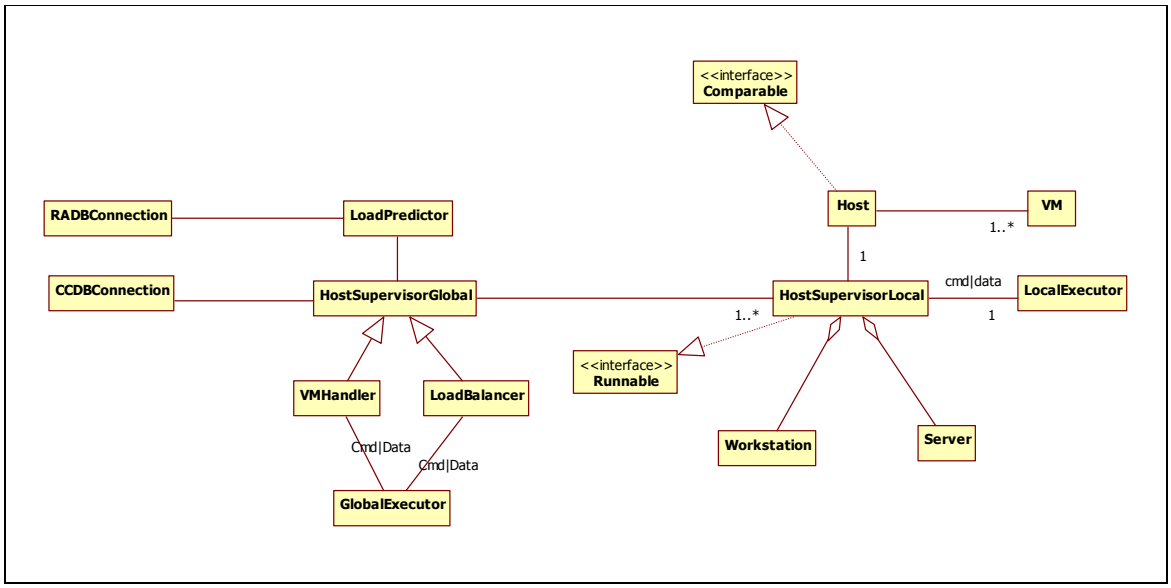


Figure 5.5: Class Diagram for ACA-Cloud

GlobalExecutor

#cmdExecStatus: int
#cmd: String

+executeCmd(cmd: String): int
+vmRunningOnHost(hostname: String): int
+getFileContents(fname: String): String
+connectHypervisor(hostname: String): boolean
+startVMOnRemoteHost(cmd: String): boolean
+suspendVMOnRemoteHost(cmd: String): boolean
+resumeVMOnRemoteHost(cmd: String): boolean
+saveVMStateOnRemoteHost(cmd: String): boolean
+restoreVMOnRemoteHost(cmd: String, restoreVMFileName: String): boolean
+shutdownVMOnRemoteHost(cmd: String): boolean
+rebootVMOnRemoteHost(cmd: String): boolean
+destroyVMOnRemoteHost(cmd: String): boolean
+migrationVM(cmd: String): boolean
+getVMIDOnRemoteHost(cmd: String): int
+getVirtualNetworkList(cmd: String): String
+hostSuspend(cmd: String): boolean
+hostPoweroff(cmd: String): boolean
+hostHibernate(cmd: String): boolean
+hostWoL(cmd: String): boolean

Figure 5.6: GlobalExecutor class of ACA-Cloud

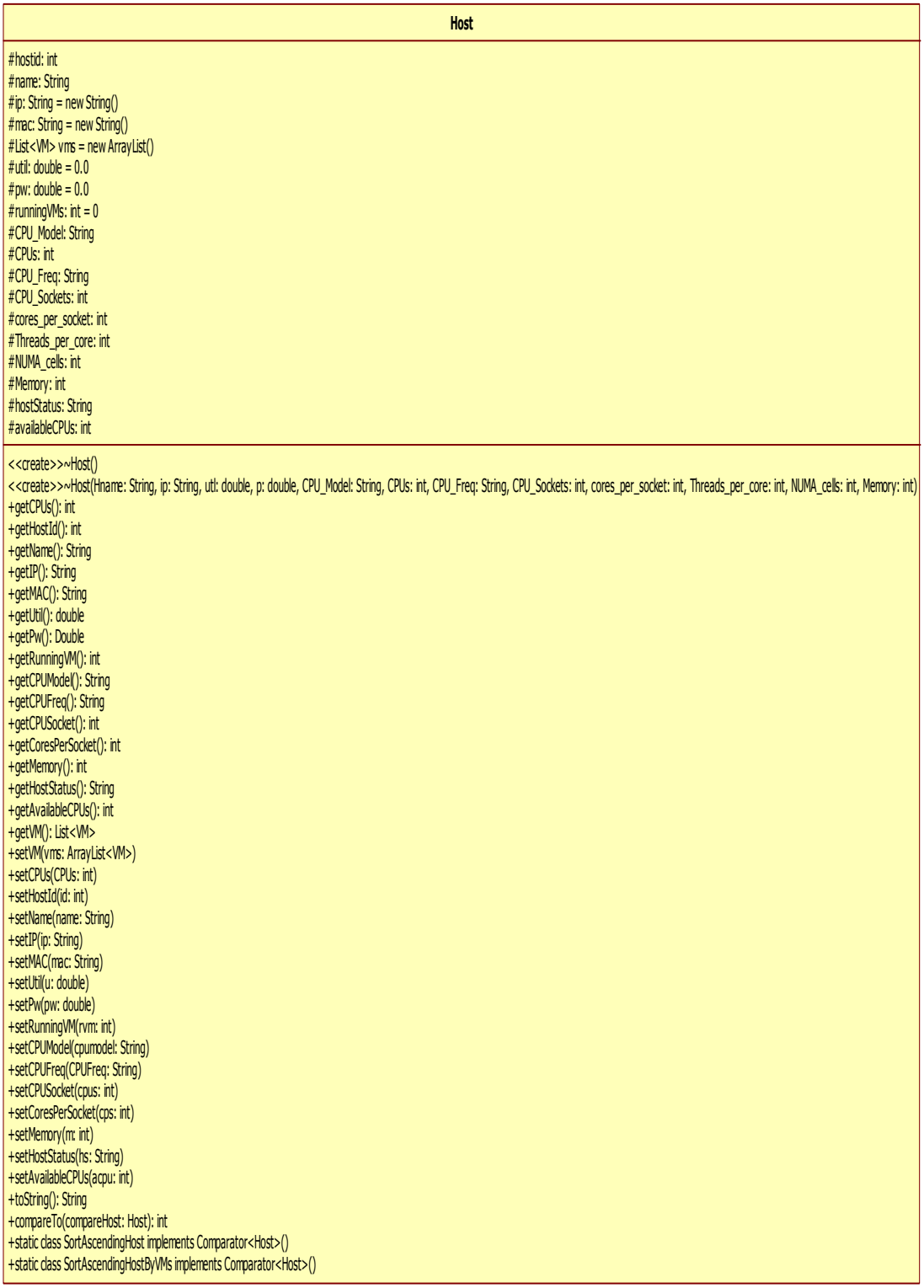


Figure 5.7: Host Class of ACA-Cloud

5.2.3 Activity Diagram

The Activity diagram is one of the useful graphical representations in UML to explain the dynamic nature of the system. It shows the sequence of activity along with conditions of flow, iteration, and concurrency.

5.2.3.1 ECRS Algorithm's VMHandler for Initial VM Placement

Figure 5.8 shows the Activity diagram of VM Handler. It shows the steps to be followed to map a new VM request to a host.

5.2.3.2 ECRS Algorithm's Load Balancer for Optimizing Current Workload

Figure 5.9 shows the Activity diagram of the Load Balancer. The LoadBalancer assesses the status of a cloud cluster and migrates the load among the active hosts when a host is under loaded or overloaded. It also handles the overhead of VM migration.

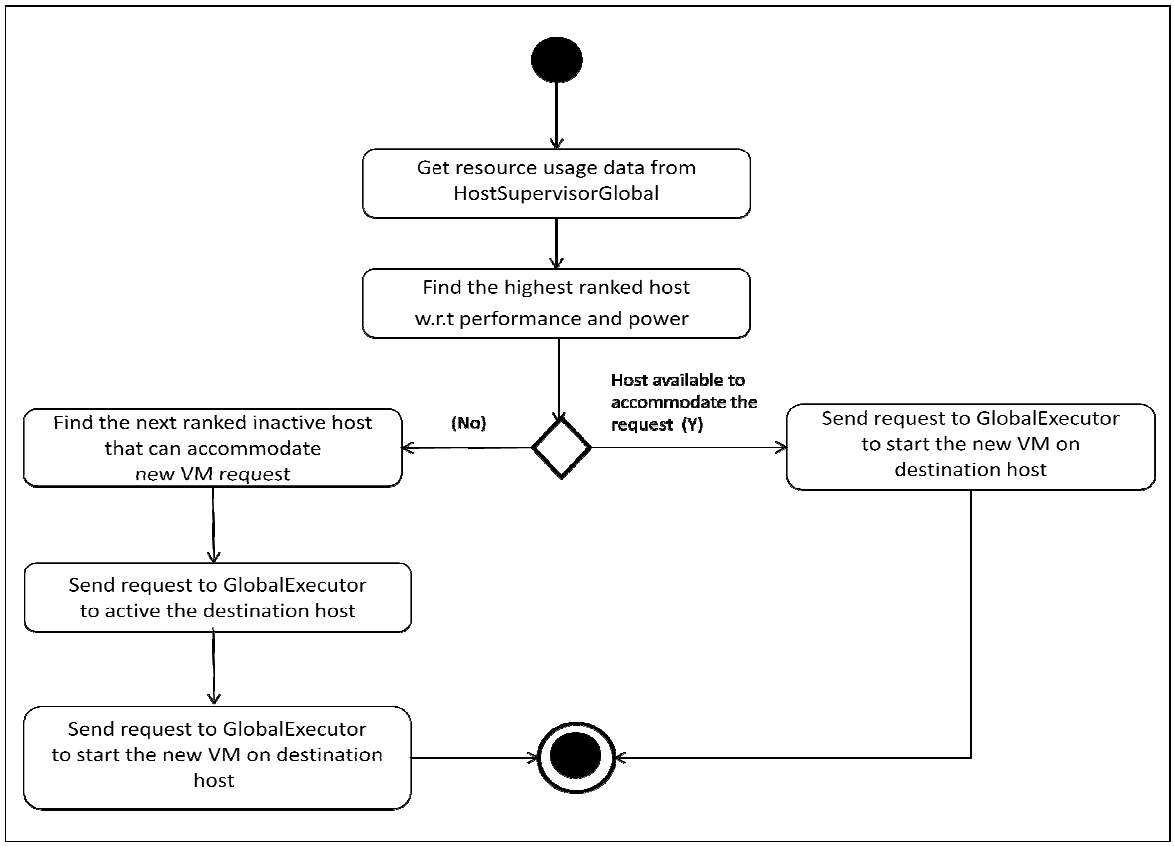


Figure 5.8: Activity diagram of VMHandler module of ECRS Algorithm

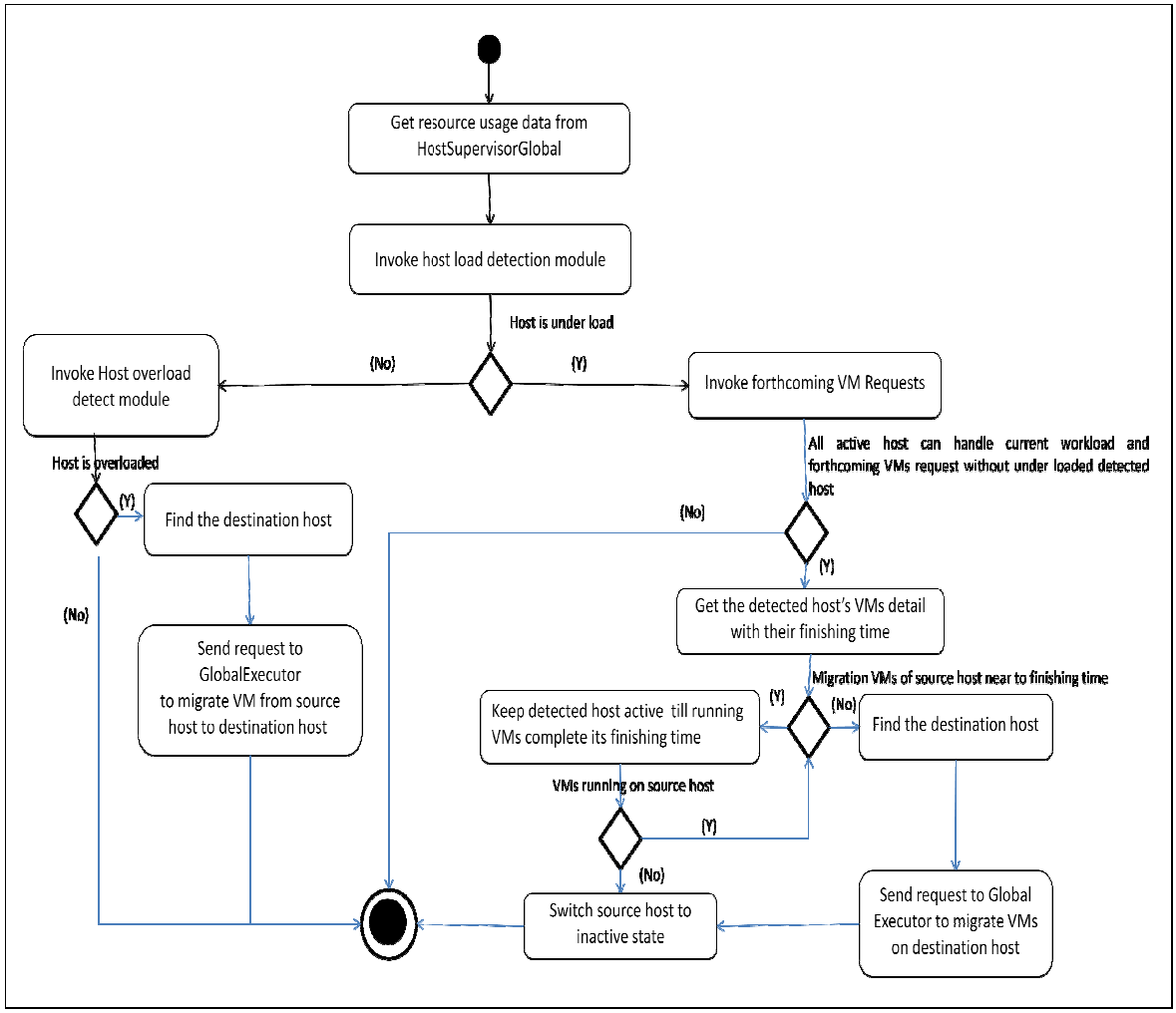


Figure 5.9: Activity diagram of LoadBalancer module of ECRS Algorithm

5.2.4 Sequence Diagram

Sequence diagrams are interaction diagrams which depict the sequence of messages exchanged between a numbers of entities. It is one of the useful design tools as it provides the dynamic behaviour of the system.

5.2.4.1 GSLACRR Successful User Negotiation

Figure 5.10 shows the sequence of events performed during a successful negotiation process between an authenticated user and other involved entities. These events are performed when resources are available as per the user's requirements for the desired time slot.

5.2.4.2 GSLACRR Renegotiation for User's Requirement

The sequence diagram (Figure 5.11) depicts the view of resources usage negotiations with cost benefits on non-availability of resources for the user's desired time slot, as explained in chapter 4.

5.2.5 State Diagram

State diagrams define the diverse states of an object during its lifetime. Figure 5.12 illustrates the different states that a user can experience during its lifetime. When a user submits a new VM request, it enters to the *Initiate* state. The request enters to the *Accepted* state on successful negotiation of usage resource and its execution time. If the resources are not available immediately, then the request enters the *Reservation* state. When the resource needs to initiate for the VM request or resources are available for the immediate VM request, the request enters to the *Provisioning* state. From the *Reservation* and the *Provisioning* states, it enters to the *Running* state when the user starts to use the VM services. The request enters the *Rejected* state when the user's negotiation is unsuccessful. Finished state is reached on three different occasions: (i) completion of VMs finishing time (ii) rejected state (iii) system failure.

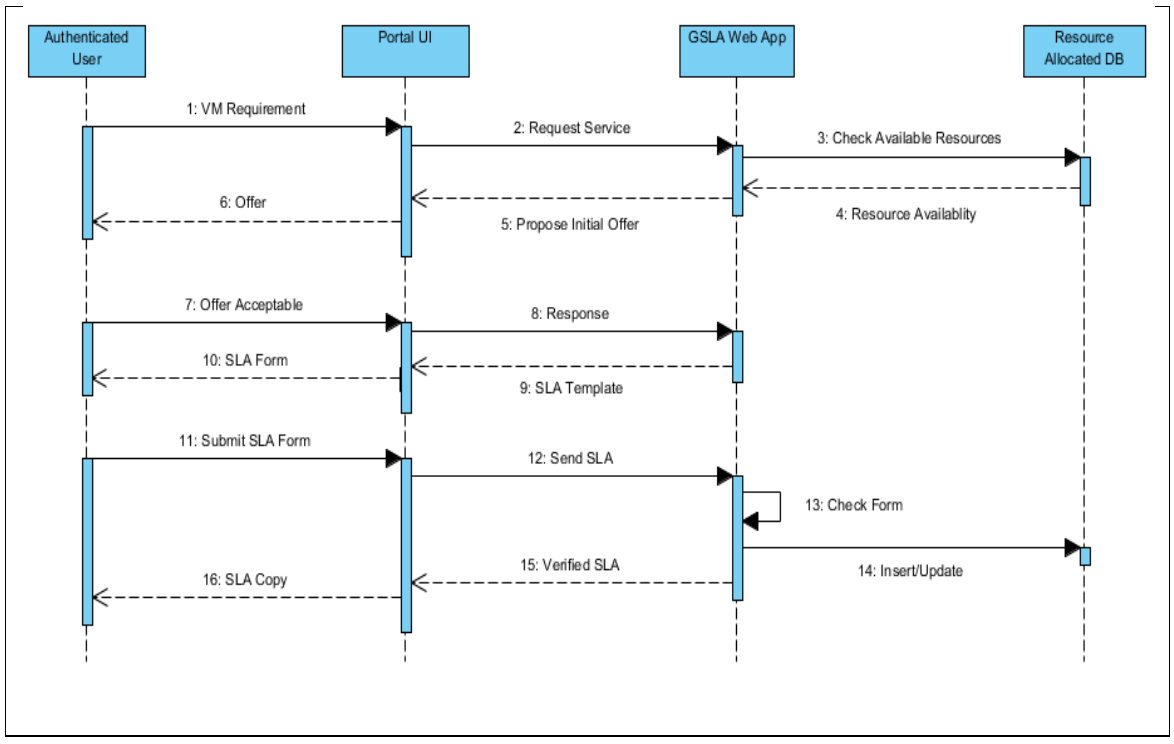


Figure 5.10: Sequence diagram of successful negotiation process of GSLACRR

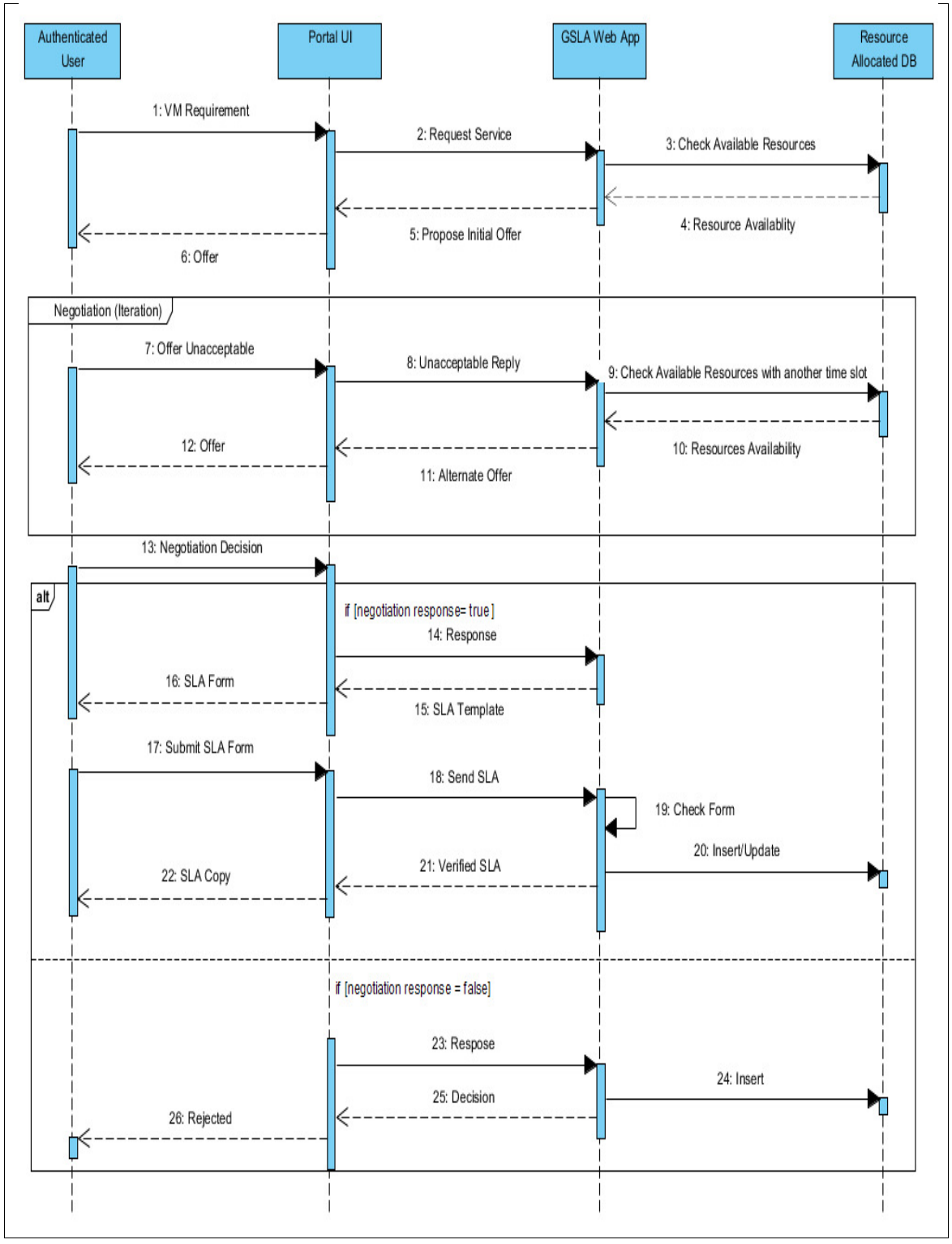


Figure 5.11: Sequence diagram of GSLACRR renegotiation for user's requirement

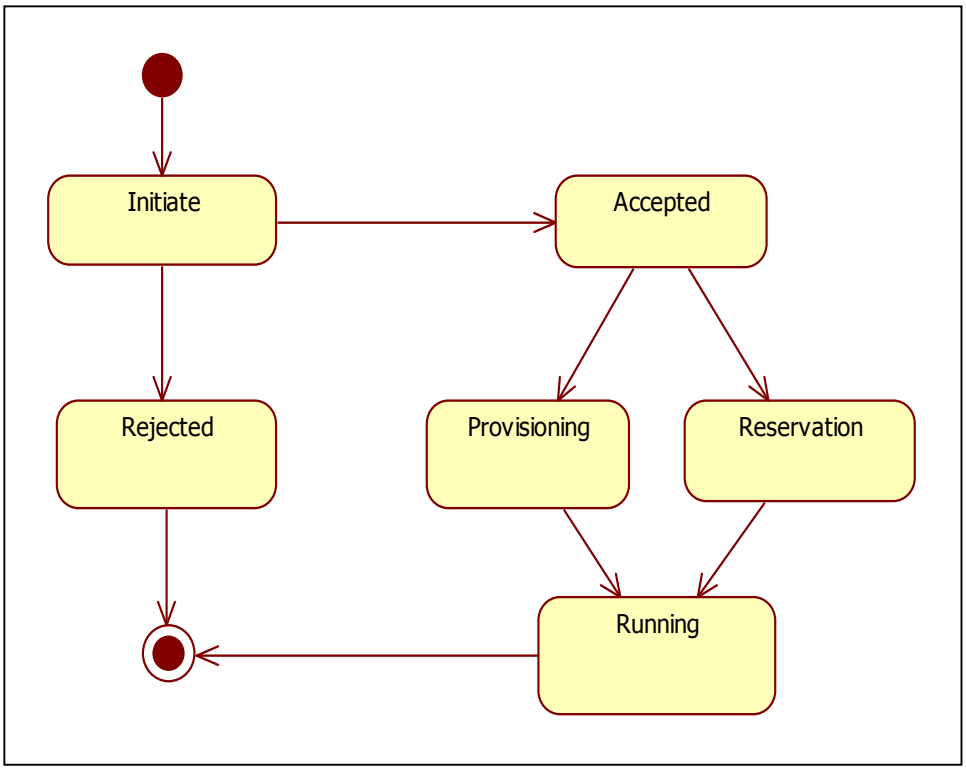


Figure 5.12: State diagram of user resource requests

5.3 Implementation Details

The ACA-Cloud has been designed using open source technologies namely Ubuntu OS [131], Kernel-based Virtual Machine (KVM) [132], Java [133], PHP [134] and MySQL [135]. To provision IaaS, "ACA-Cloud" has used Operating System-based Virtualization. Although there is a concern about the overhead of host operating system, there are a number of issues related to hardware-based virtualization. One of the prominent issues is the compatibility of the hardware device drivers with hardware-based virtualization which is readily available in the host operating system. Hardware device drivers must be supported by the hypervisor as the virtualization layer directly communicates with the host hardware in hardware-based virtualization. Secondly, a wide range of host management and administration based advanced features are not available in hardware-based virtualization, though these are commonly available in the host OS in Operating System-based Virtualization. Cloud setup uses KVM as a Virtual Machine Manager and an open source Libvirt API [136]. Libvert has a set of APIs to enumerate, monitor and manage virtual machines, virtual networks and storage. Livert has been chosen as it is compatible with various hypervisors such as KVM [137], Xen [108], and VMWare [138]. Thus, with minimum efforts the underlying hypervisor can be changed [139].

The "ACA-Cloud" is implemented in Java. Java is the chosen programming language asit is an incredibly entrenched technology. It has a number of advantages, such as; it is platform independent, fast, robust, secure and reliable. Moreover, powerful development tools such as Eclipse [140], Netbeans [141] for Java programming are freely available. Netbeans [141] has been used as a Software Development Platform. A user can request and access the resources using web based interface developed using PHP. The choice of PHP as a web based programming language has been made because it has excellent documentation and community support. Table 5.1 gives the details of implementation technologies.

Table 5.1: Implementation technology used by the ACA-Cloud

Datacentre	Heterogeneous
Platform	KVM 0.14.1
Virtualisation API	Libvirt 0.10.2.1
Host Operating system	Ubuntu 12.04
Network File System (NFS) Server	NFSv4
Technology	Java 1.7, PHP 5.5.12
Software Development Platform:	Netbeans 7.1
Webserver:	Apache Tomcat 7.0.22.0
Database:	MySQL 5.6.8

5.3.1 Experimental Setup

For experiment, a heterogeneous virtual environment has been setup that consists of eight servers with data as given in Table 5.2. All servers are connected with 100 Mbps Ethernet and 8 TB Networked Attached Storage (NAS) storage. Hosts run VM and a common storage provides seamless storage to allow live migration [142] of VMs. NFS server and client [143] has been configured to create NAS. The Dell Core 2 Duo machine was chosen as the cloud cluster controller and the rest of machines were selected as compute hosts.

Table 5.2: Experimental setup [144-147]

Server type	Dell power edge r710 server	Dell Power Edge 2900 server	Dell i5 2.5 GHZ	Dell Core 2 Duo E7500
Number	3	1	2	2
Model	Xeon E5520	Xeon®5400	i5 2.5 GHz	Core 2 Duo E7500
Cores	2*4	2*4	4	2
Memory	16 GB	8 GB	4 GB	2
Disk	1350 GB	438GB	320GB	120GB
P_{idle} (W)	128.7	40	39.8	39.3
P_{imax} (W	247	80	106.8	78.9

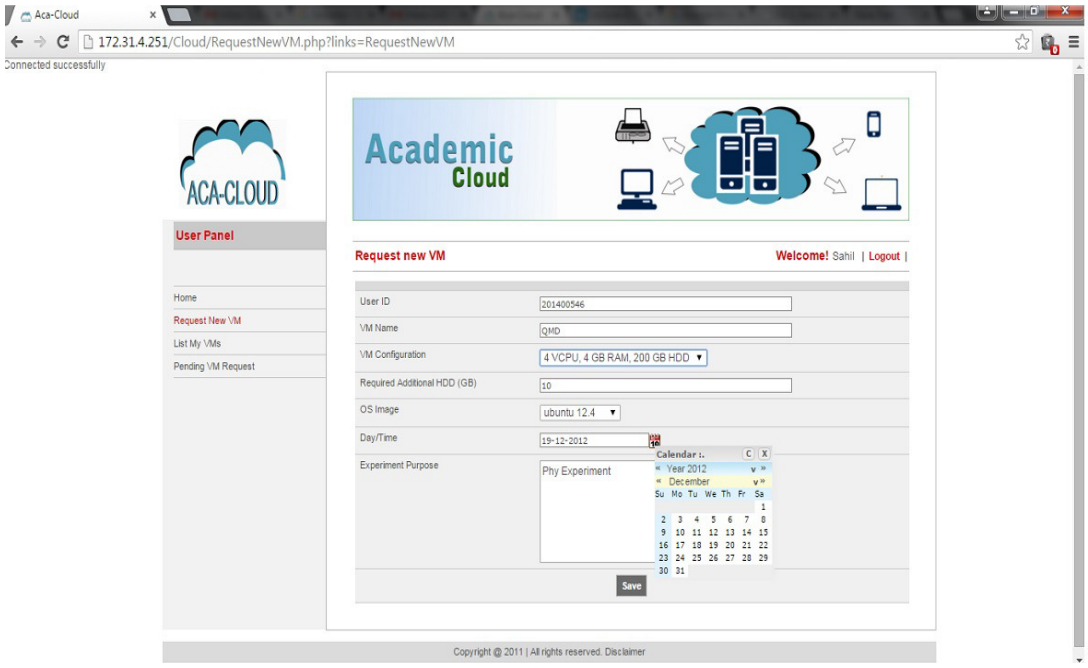


Figure 5.13: Authenticated user resource request screen

5.3.2 User Interface

The user can login into the system and submit the VM request via a web based user interface**.** Figures 5.13, 5.14 and 5.15 show screen shots of an authenticated user's VM request, GSLA negotiations, and the admin panel respectively. The web interface allows the admin to see the machine status, configuration, perform actions like start, power-off, and watch the VM status on each machine etc. Screen shots of the development environment for the ACA-Cloud project have been provided in Figures 6.17 and 6.18.

Figure 5.14: GSLA negotiation screen

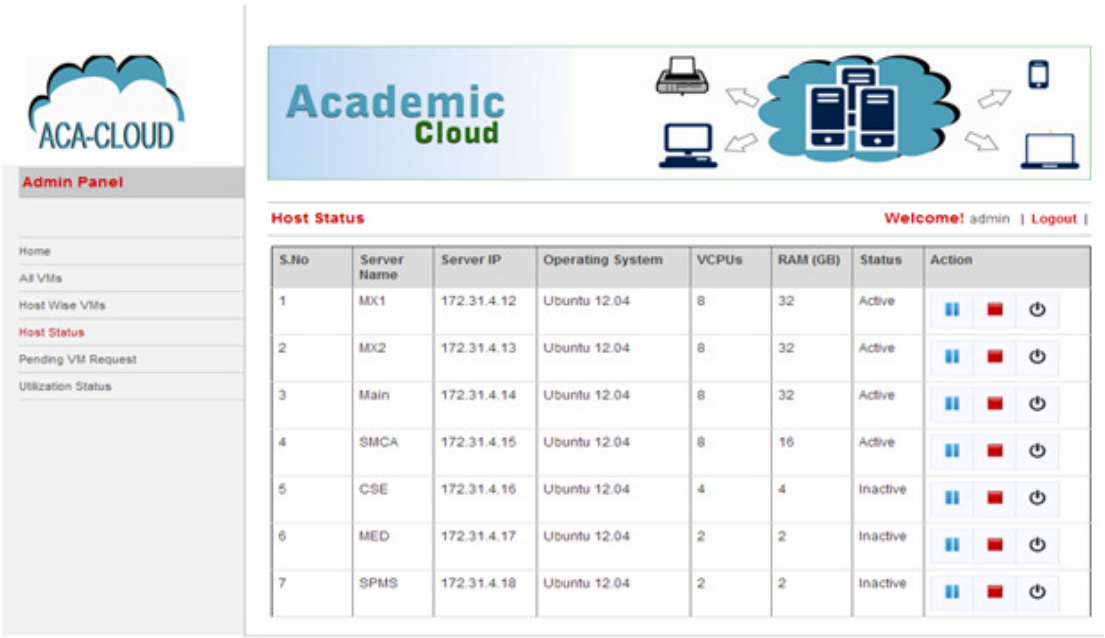


Figure 5.15: ACA-Cloud administration panel screen

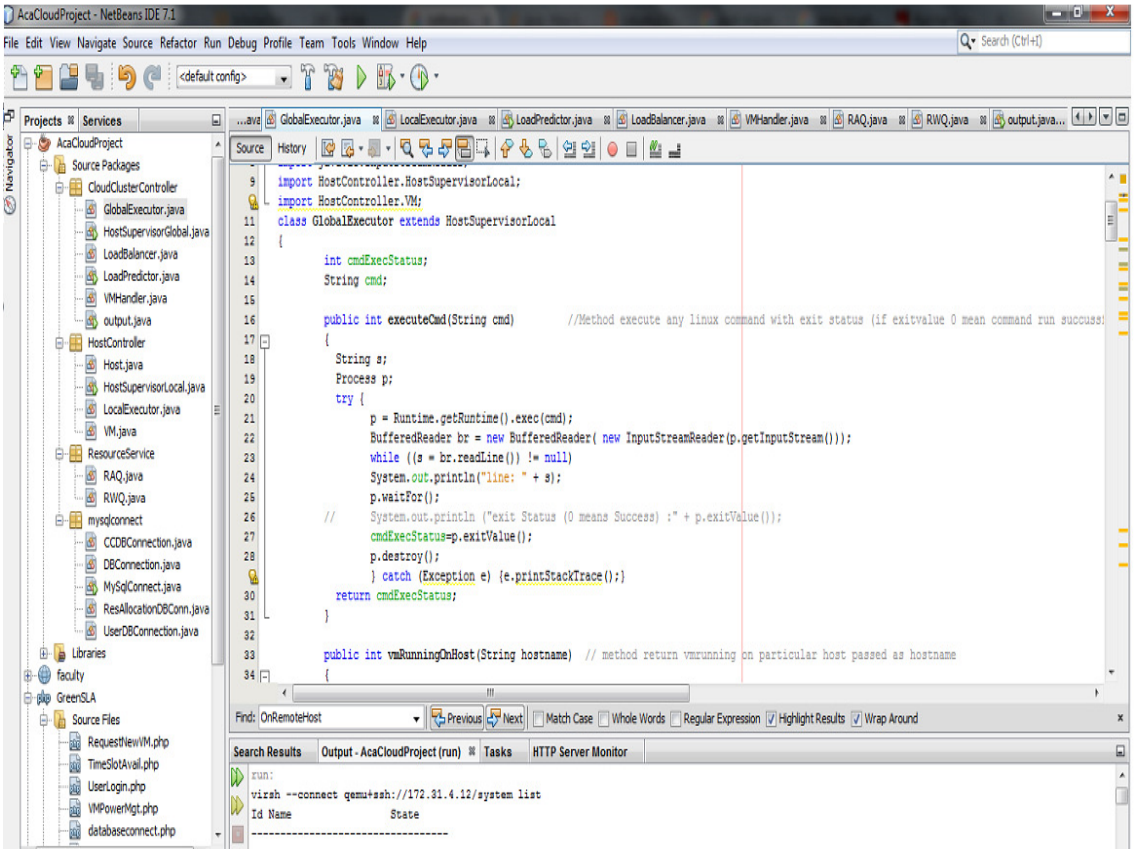


Figure 5.16: Development environment and output generated by GlobalExecutor for ACA-Cloud

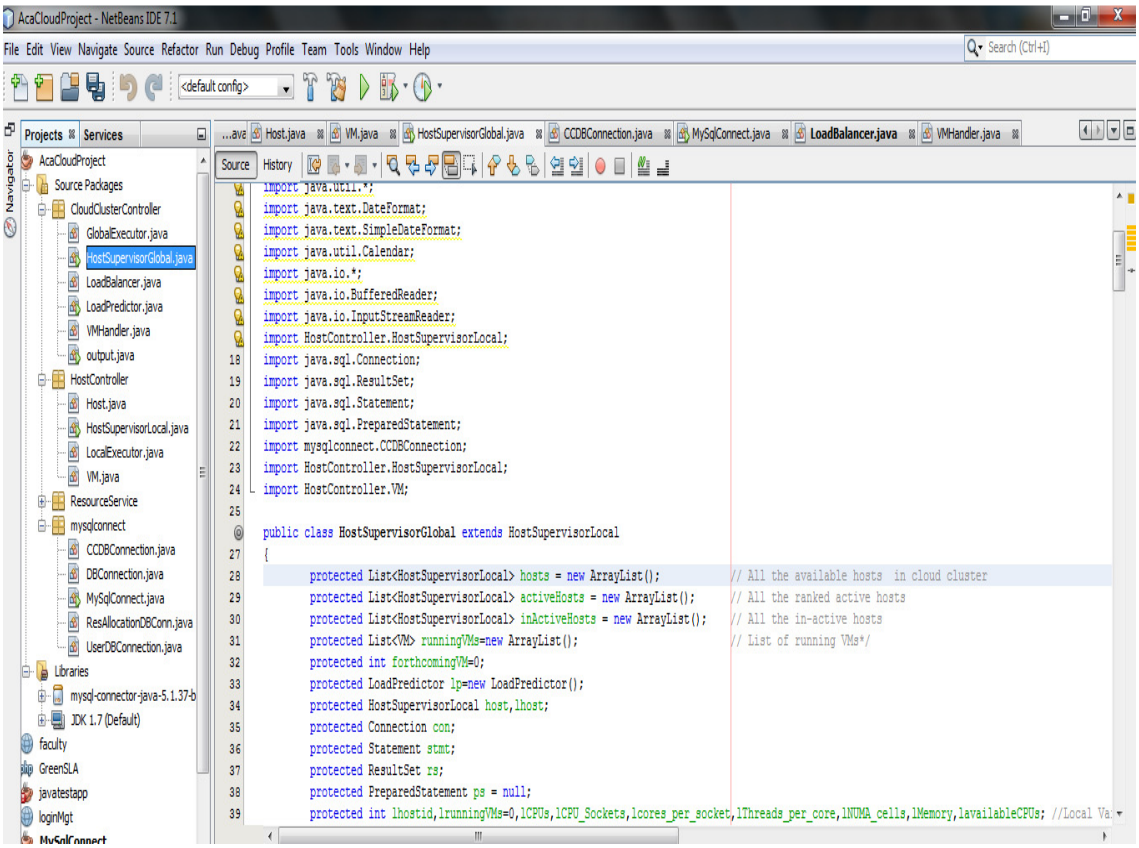


Figure 5.17: Snapshot of HostSupervisorGlobalfor ACA-Cloud

5.4 Summary

This chapter describes the design and implementation of the ACA-Cloud framework. The detailed requirements and design have been analyzed using UML. The implementation details of the experimental testbed that implements the proposed algorithms are provided. The main objective of the implementation of the ACA-Cloud framework is to demonstrate the working of the proposed algorithms. The next chapter discusses the experimental results and validation of the proposed algorithms.

Chapter 6

Test Results and Analysis

The previous chapters discuss energy efficient management of cloud resources and running services on it. Using the framework, named ACA-Cloud, cloud administrators can foresee the status of physical and virtual resources, and predict the run-time status. With this assessment, it offers users request execution with alternative time slots and thus accommodate more user requests. The setup of the physical testbed and the designed framework is discussed in the previous chapter 5.

This chapter presents the validation of the proposed algorithms. The experimental results help us to reach on a conclusion on the effectiveness of the proposed approach. This chapter evaluates the performance of the algorithms proposed in chapter 3 and chapter 4. Several experiments were performed on the experimental testbed. This chapter discusses the performance evaluation metrics and workload characteristics. The results have been discussed considering various perspectives such as the effects of resource heterogeneity, the effects of number of different type of requests etc. Finally, a statistical analysis of the output has been performed in order to assess the significance of differences between the proposed and existing algorithms. The result show that the proposed algorithms performs better in comparison to the existing algorithms with respect to energy efficiency and also handle more number of user VM requests.

6.1 Performance Evaluation Criteria

A performance evaluation criterion defines the effectiveness of algorithms while taking into account the relevant characteristics of the algorithms [148]. One of the objectives is to make the system more energy efficient. Energy efficiency can be defined as the ratio of the useful output to the energy consumed to do it [149]. However, an IaaS is not interested in knowing what kind of useful output is produced by running the applications; it concerns only with the resource utilization incurred by the system [127]. At the cloud infrastructure level, the useful output is a number 'N' of VMs hosted by a server with the desired QoS. So the objectives are to minimize the energy consumption and to host the maximum number of VMs. So these two objectives become metrics for performance evaluation and comparison. The next section discusses these performance evaluation metrics and the workload traces.

6.1.1 Performance Metrics

As energy efficiency is one of the important concerns for a cloud system designer and administrator, it is considered as one of the performance metrics to be monitored. Users' requests should not be ignored to make the system more energy efficient. So, the number of user requests handled by the proposed framework becomes another important performance metric. In this service framework, energy efficiency and the number of handled VM requests are measured using two metrics, namely Mean Energy Consumption (MEC) and Accepted VM (AVM). In order to aptly compare the efficiency of the algorithms, MEC and AVM have been used to evaluate the performances of the proposed and the existing algorithms. The first metric calculates the mean energy consumption of all computational resources. As described in the energy modelin chapter 3, MEC is measured in kWh. The second metric measures the number of accepted VM requests out of the total given requests for all schedule intervals.

6.1.2 Workload Data

For experimental results, workload trace logs have been generated based on the software running inside the VMs used by teacher, students, and research scholars on a private cloud. There are three standard VMs allocated to users. Table 6.1 describes characteristics of the deployed VMs. Small VMs are allocated to users who run simple programs of C, C++, Java, Oracle etc. Netbeans, Eclipse, IIS web server, SPSS etc. software run on medium sized VMs. On large Fedora VMs, physics experiments have been executed like as physics scholars use

Statistical Multi fragmentation Model (SMM), Quantum Molecular Dynamics (QMD), Isospin dependent Quantum Molecular Dynamics (IQMD), Boltzmann-Uehling-Uhlenbeck (BUU). Power consumption readings were taken after every minute.

Table 6.1: VMs deployed

INSTANCE (VM)	CORES	RAM	DISK SIZES
small	1	1 GB	40 GB
medium	2	2 GB	100 GB
large	4	4 GB	200 GB

6.2 Performance Evaluation of Proposed ECRS Algorithm

To validate the proposed algorithm in chapter 3, several experiments were conducted with different number of VM requests, VM types, which were executed according to different schedule patterns. The total power consumption of all the nodes was basically dependent on the number of VMs running, VMs size, duration, and the numbers of nodes in kept in sleeping state by the algorithm. For performance evaluation, the ECRS algorithm was compared to the existing green unbalanced scheduling algorithm [103] with respect to MEC and AVM metrics. For statistical correctness, seventy four different experiments were conducted.

6.2.1 Test Case 1: Low Heterogeneity Performance Evaluation

In this test case, four resources, having number of VCPUs between 2 to 4, have been taken as given in Table 5.2. In the first test sequence, a small number of requests were taken (14 to 26 VMs). To make the evaluation closer to the real-world scenario, the proposed algorithm response has been evaluated with different heterogeneity of jobs and resources. In this test sequence, total accepted VM requests and average power consumption have been evaluated in four scenarios with incoming VM requests (input) in the proportion of (i) Scenario 1 (S1): equal ratio of all VM types (small, medium, and large) (ii) Scenario 2 (S2): a large number of small VM types with proportion of 80:10:10 with small, medium, and large VM respectively (iii) Scenario 3 (S3): a large number of medium VM types with proportion of 10:80:10 with

small, medium, and large VM respectively and (iv) Scenario 4 (S4): a large number of large sized VM types with proportion of 10:10:80 with small, medium, and large VM respectively. In all the scenarios, same numbers of VM requests are sent. Figures 6.1 and 6.2 for ECRS and Green Unbalanced scheduling approaches are plotted with respect to MEC and AVM metrics.

While taking the scenario (S1 and S2), wherein the total number of VM requests are less and the accepted VMs are equal in both the algorithm (S1, from 14 to 23 in Figure 6.1), ECRS consumes comparatively lesser power (S1,S2, from 14 to 23 in Figure 6.2). The ECRS algorithm also performs better with respect to MEC on receiving a large number of medium VM type requests such as in (S3 from 14 to 19 in Figure 6.2); wherein the accepted VMs are equal as shown in Figure 6.1 As mentioned in the algorithm, servers are ranked based on the greater order of VCPUs. Most of the VM requests are catered by high power efficient and large core servers. The remaining servers stay in the low power mode. As the number of VM requests are increased, the ECRS algorithm accepts more requests (S3 from 20 to 25 in Figure 6.1) because best fit mapping. However, it consumes more power in these cases because it caters more number of requests and all resources are active/power-on with greater utilization level.

On receiving a large number of VM requests (Scenario S4), the ECRS algorithm accepts more VM requests in comparison to the Green Unbalanced algorithm as the large sized VM requests are handled by high power efficient and large core servers. The medium and small VM requests are allocated to small core servers. With this implementation, the ECRS algorithm is able to accommodate more VM requests. Now, in the Green Unbalanced algorithm, there is no ranking of servers and no specified order of VMs. So the medium and small VM requests are allocated to the large core servers leading to their unavailability when a large VM request arrives.

There is a trade off between accepted VM requests and power consumption. In scenarios, S3 and S4 (Figure 6.1 and 6.2), the graph depicts less VM request acceptance and more power consumption in comparison to the scenarios S1 and S2 in both the algorithms because to cater large number of medium and large VM type requests, it needs to activate all the PMs with all the VCPUs.

6.2.2 Test Case 2: High Heterogeneity Performance Evaluation

In this test case, the performance and power consumption of cloud applications for high resource heterogeneity with respect to MEC and AVM metrics have been evaluated. Each resource has between 2 to 8 VCPUs as given in Table 5.2, and number of VM requests counts up to 325.

Figures 6.3 and 6.4 show the accepted VM request and power consumption comparison of ECRS and Green Unbalanced algorithms, with high machine heterogeneity. It can be seen from the Figure 6.4 that the power consumption of ECRS is lesser than Green Unbalanced, when the numbers of VM requests are less (from schedule 1 to 14). There is a scenario when there are a large number of heterogeneous VM requests (such as schedule 12, 25 etc); in this case, all the servers in the data centre were found active to accommodate VM requests and there was no significant difference in the power consumption between the two algorithms. However, ECRS accepts more VMs as the VMs are sorted in a descending order with regard to their sizes and the servers are ranked with respect to the number of VCPUs. Servers having greater number of VCPUs are allocated VMs first. The ECRS algorithm places large sized VM requests on high power efficient and large core machines. Small core machines accommodate medium and small VM requests. Thus, the ECRS can accommodate or accept more VM requests as compared to Green Unbalanced algorithm.

From the experiment results, it can be concluded that ECRS performs better as compared to Green Unbalanced approach with respect to MEC when there are a small number of VM requests in proportion to available infrastructure. It performs better with respect to AVM when there is a large combination of large and medium sized VM requests. The results reveal that the proposed ECRS algorithm outperformed by 5% and 1.5% for power saving and AVM, respectively.

6.2.3 Statistical Analysis of Results

To get statistically significant differences between the existing and the proposed algorithms, paired samples t-test has been conducted. In this test, the mean power consumption of the Green Unbalanced algorithm has been compared with that of the ECRS algorithm. The null hypothesis of the mean power consumptions of the two algorithms is the same. Results reported (Table 6.2) have been found significant at 0.05 levels.

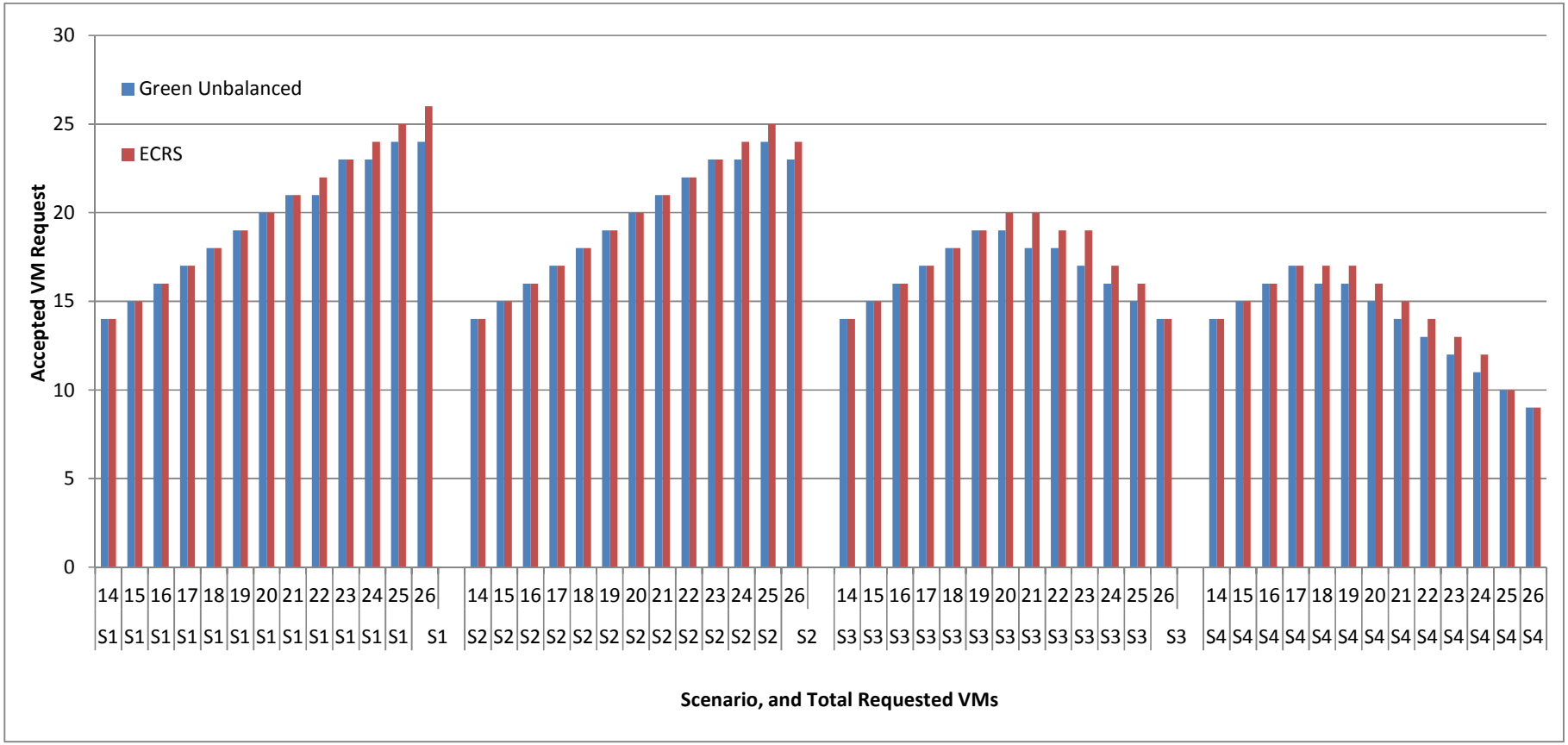


Figure 6.1: Accepted VM requests with low resource heterogeneity

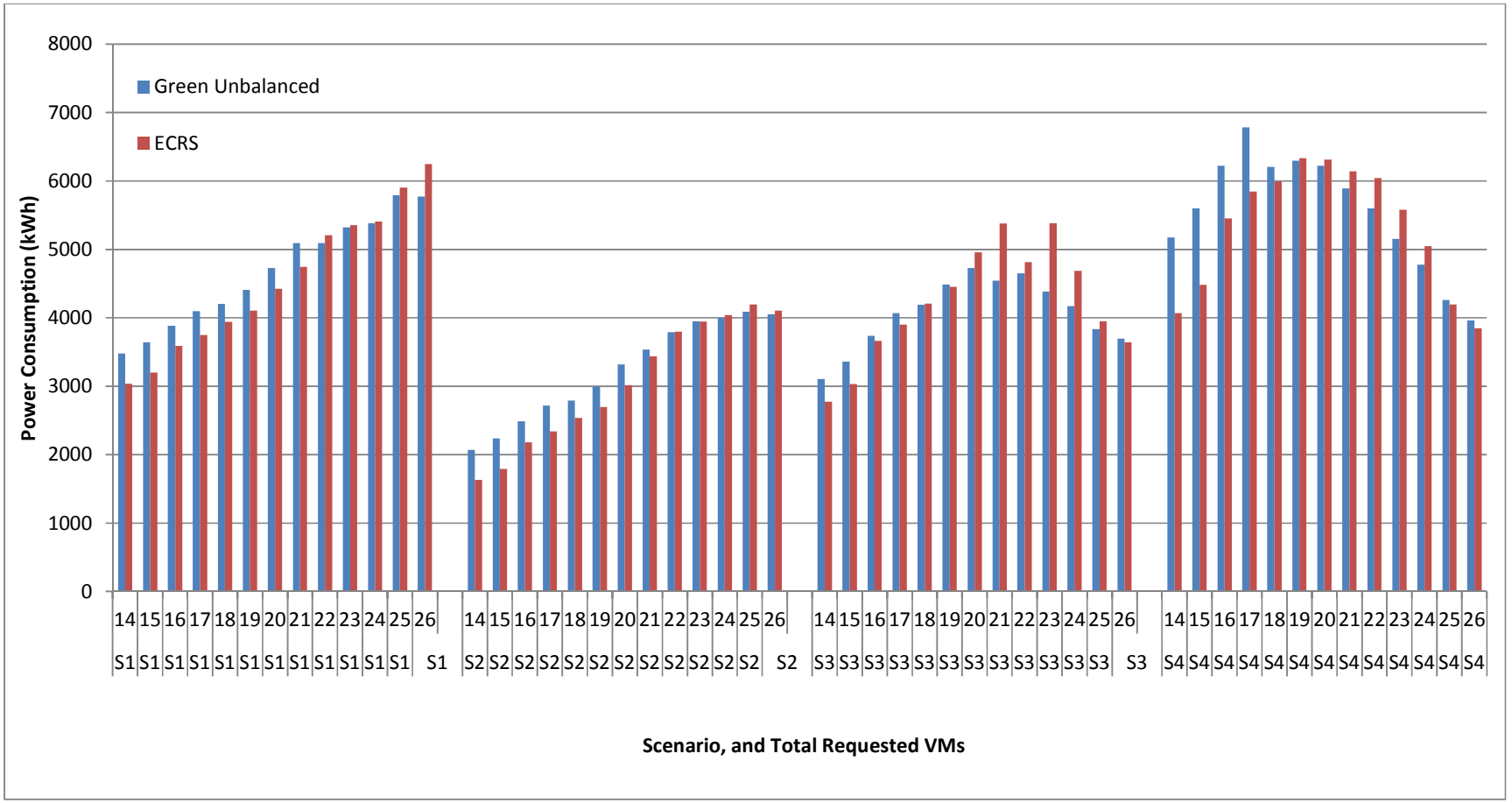


Figure 6.2: Average Power consumption of a centre with low resource heterogeneity

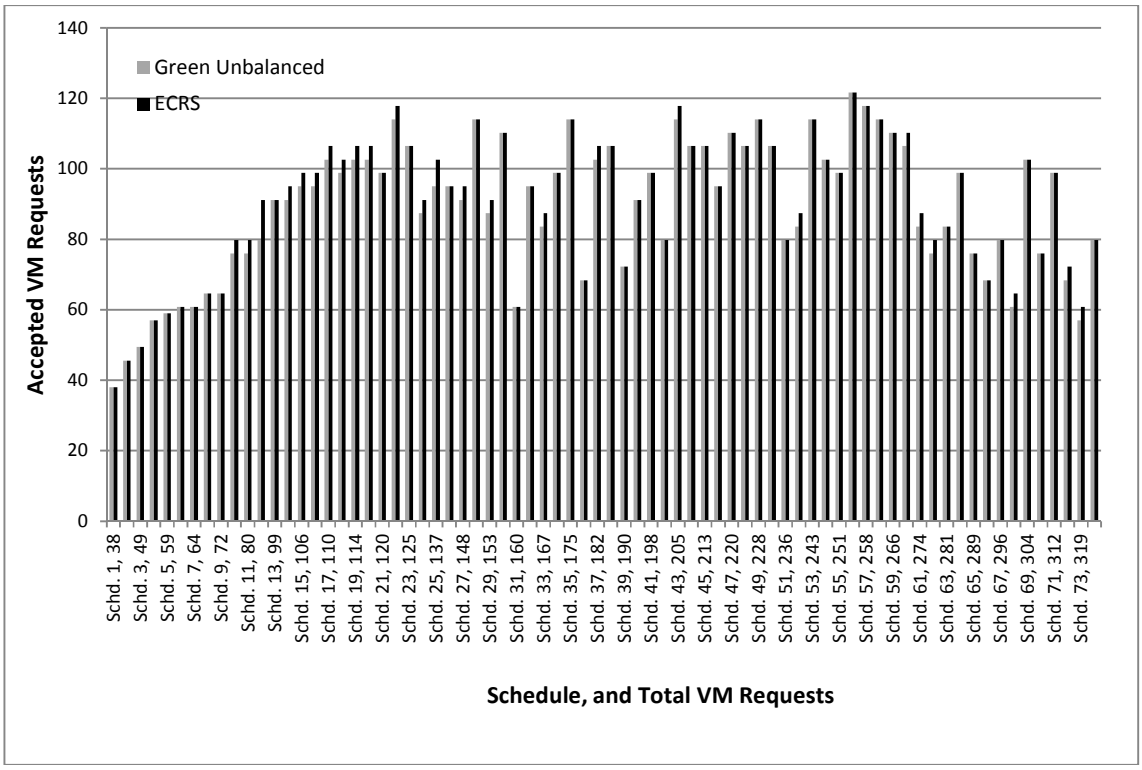


Figure 6.3: Accepted VMs requests with high resource heterogeneity

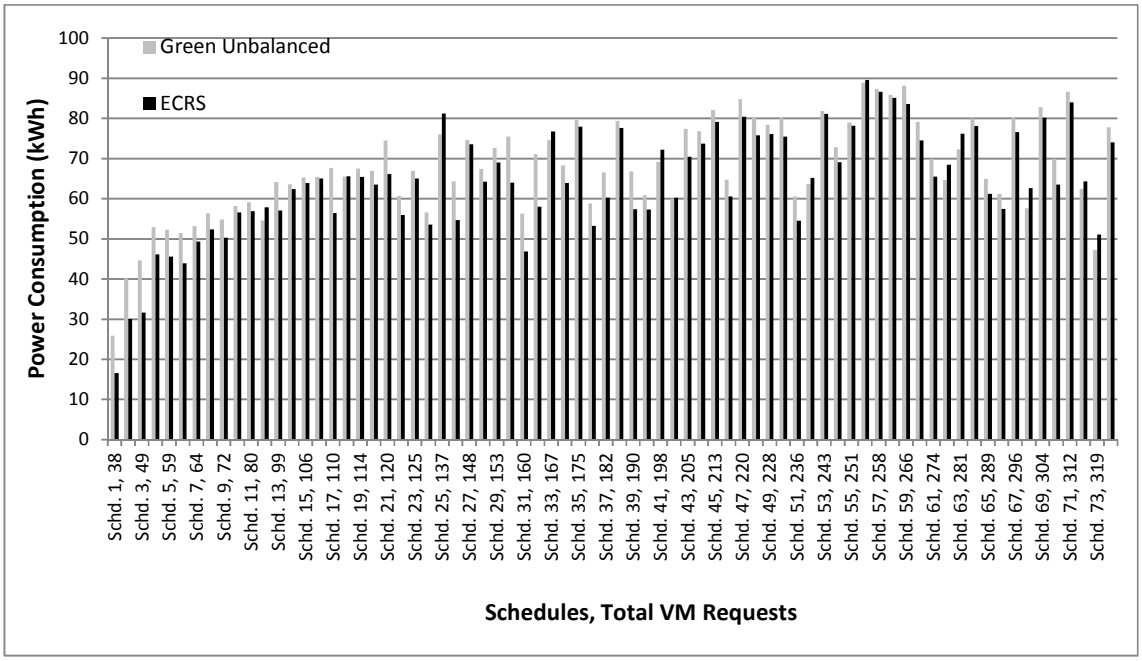


Figure 6.4: Average power consumption of a centre with high resource heterogeneity

Table 6.2: t-test: paired two sample for means

	Green Unbalanced	ECRS
Mean	67.84284811	64.42735865
Variance	145.5018268	184.0072481
Observations	74	74
Pearson Correlation	0.955700788	
Hypothesized Mean Difference	0	
df	73	
t Stat	7.177987069	
P(T<=t) one-tail	2.43297E-10	
t Critical one-tail	1.665996224	
P(T<=t) two-tail	4.86595E-10	
t Critical two-tail	1.992997097	

T-test results in Table 6.2 indicate the competent power saving capability of the ECRS algorithm in comparison to the Green Unbalanced algorithm. Therefore, it can be concluded that the proposed algorithm is significantly different from the existing algorithm with regard to the MEC metric. The AVM metric has been calculated on the basis of the collected experimental resultant data. It has been found that total number of accepted VMs of the proposed algorithm is greater than those of existing algorithm.

6.3 Performance Evaluation of Proposed GSLACRR Algorithm

This section evaluates the performance of the proposed GSLACRR algorithm as presented in chapter 4. The GSLACRR algorithm is compared with the non GSLA aware algorithm. In a non GSLA aware algorithm, there is a simple negotiation between the user and the cloud provider. Resources are allocated on a user's request if they are available for the requested time slot; else a rejection message is conveyed to the user. Also, in case of a non GSLA aware algorithm, there is no specified ranking of servers or order of VMs. In order to compare the efficiency of the proposed GSLACRR algorithm, two performance metrics have been used (described in Section 6.1.1). To evaluate the algorithms, the experimental setup as given in Table 5.2 has been used, except for one Dell i5 2.5 GHz machine.

6.3.1 Analysis of Results

Figure 6.5, presents a comparison chart of the GSLACRR and the non GSLA aware algorithms for a number of accepted VMs requests against the total number of the VM requests submitted. Figure 6.7, depicts the average power consumption of all the PMs against the total number of the VM requests submitted.

It has been found during the analysis that the proposed GSLACRR algorithm performs much better in terms of power consumption up to schd.22 (Figure. 6.7) when the number of VM requests are equal, as shown in Figure 6.5. After schd.22 in Figure 6.7, the GSLACRR algorithm consumes more power but it also accepts more requests for the corresponding schedules as shown in Figure 6.5. The GSLACRR aware algorithm performs better with respect to MEC because of two reasons. Firstly, as mentioned above, on receiving a less number of VM requests or small VM types, most of VM requests are accommodated by the high power efficient and large core servers. The remaining servers stay in low power mode. Secondly, through user negotiation all the VM request reservations are aggregated into one part of the slot from big free time slot and resources are switched off in other part of the time slot. In non GSLA aware algorithm, there is no provision to pursue the user for another time slot in case resources are not free during resource demanding time slot. The proposed GSLACRR algorithm conserves 1.5% more power as compared to the non GSLA aware algorithm.

Figure 6.6 presents the results of the rejected VM requests out of the total requested VMs. From the results, it is derived that there are lesser number of rejected VMs in case of the GSLACRR algorithm in comparison to the non GSLA algorithm. It is difficult for a cloud provider to allocate resources to the requested VMs all the time due to limited resource availability, but through the GSLACRR algorithm users can opt for an alternate time slot with their cost/credit benefits. It is a win-win situation for the service provider as well as the users. Altogether, it minimizes the rejection rate of service providers. Overall, the GSLACRR algorithm improves 1% of acceptance rate as compared to the non GSLA aware algorithm.

As shown in Figure 6.8, the GSLACRR algorithm accommodates equal number of VMs (Figure 6.5) with lesser number of PMs. As the number of VMs requests increased from 65, both algorithms used all of the PMs. This observation indicates that the proposed algorithm consolidates more VMs onto fewer PMs.

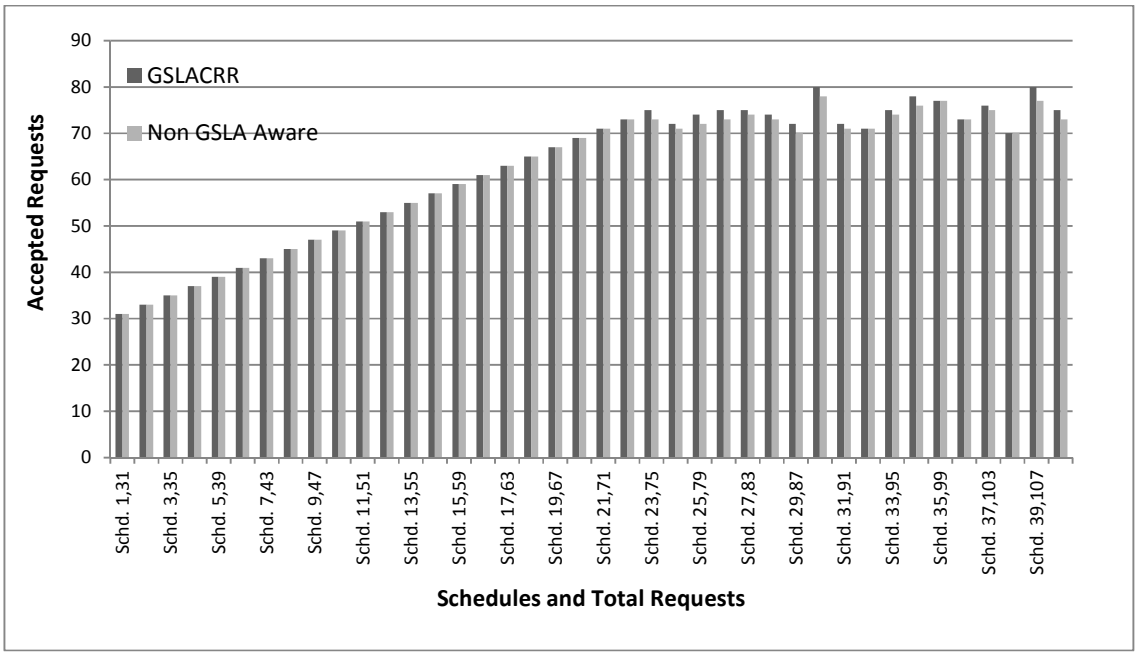


Figure 6.5: Accepted VM requests

6.3.2 Effect of the number of VM Requests

Experiments have also been observed with respect to different types of workload. Figure 6.9 compares the picture of accepted VMs of the two algorithms with reference to small, medium, and large VMs. Figure 6.9 (a) shows that on receiving lesser number of VM requests (up to 73 requests), the number of accepted small, medium and large VMs are equal in case of both the algorithms. As the number of requests increased (75 requests and above, Figure 6.9 (b)), the acceptance rate of the non GSLA aware algorithm decreased in comparison to the GSLACRR algorithm. In some cases (like 93, 105 requests, Figure 6.9 (b)), both algorithms accept equivalent number of small, medium, and large VMs requests. This is because of large number of small VM requests and the aggregate of the requests (small + medium + large) require lesser number of processing cores/less number of PMs (Figure 6.8). Figure 6.9 shows that the proposed approach GSLACRR consolidates 1.12%, 0.5%, 1.2% more for small, medium, and large VMs, respectively.

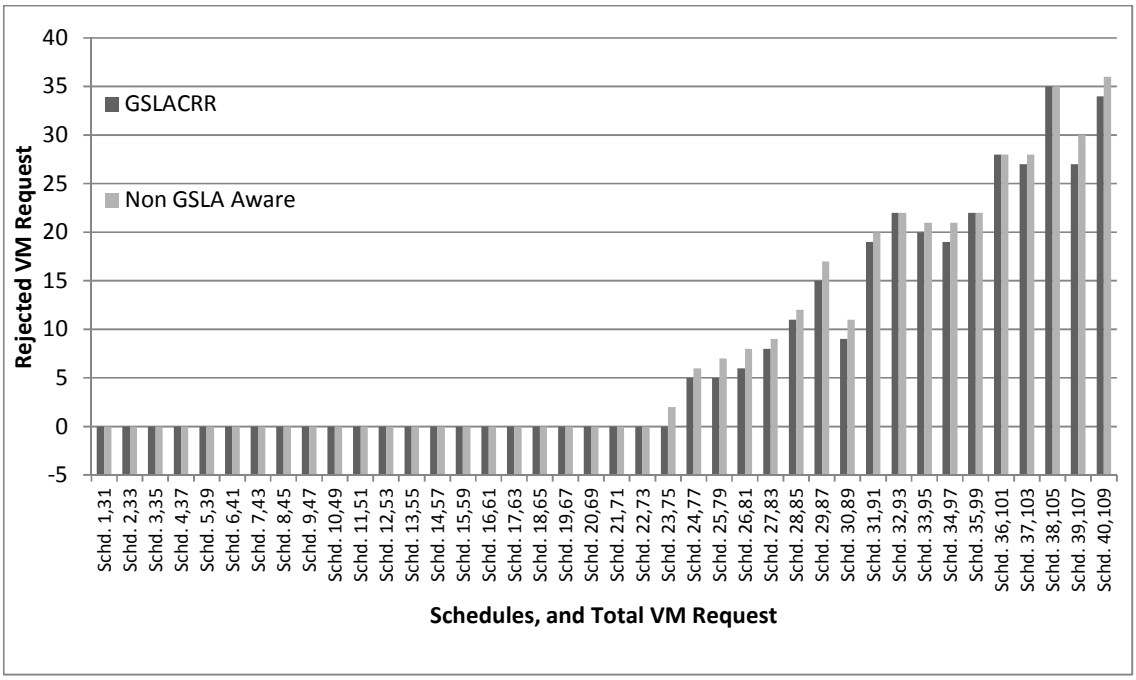


Figure 6.6: Rejected VM requests

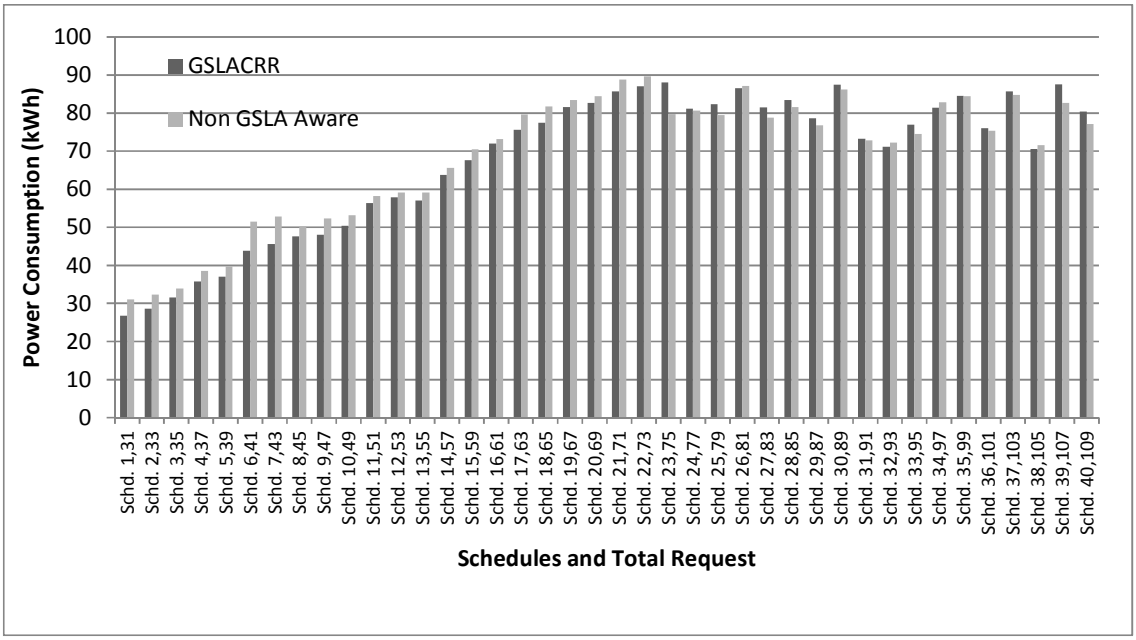


Figure 6.7: Average power consumption of a center

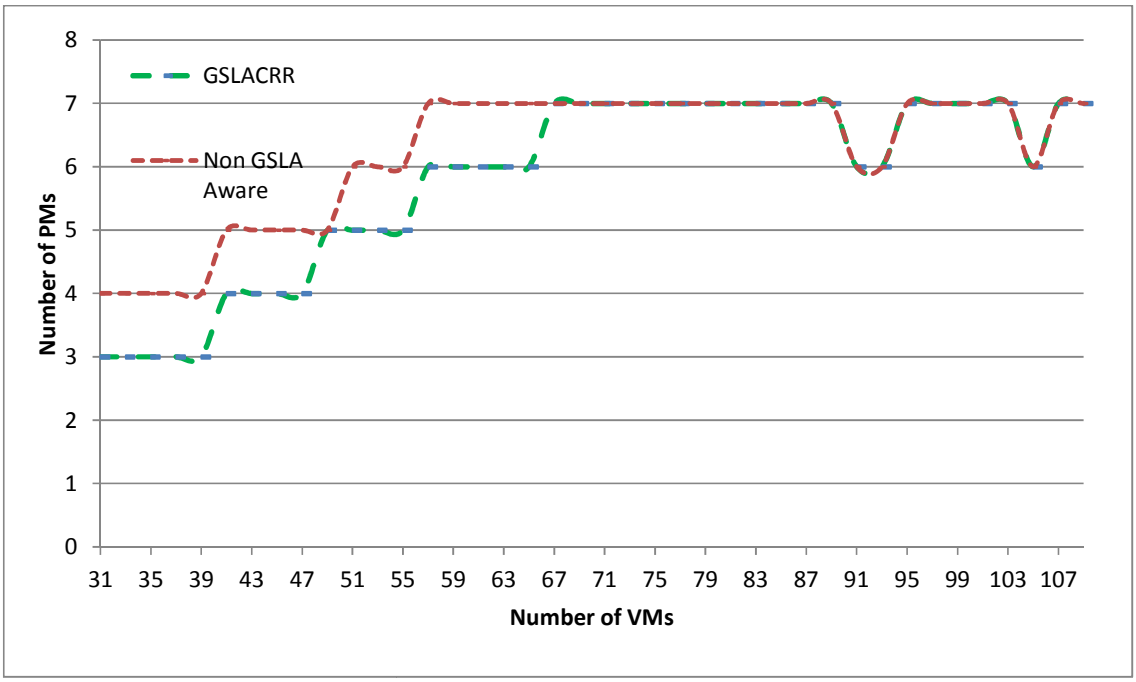
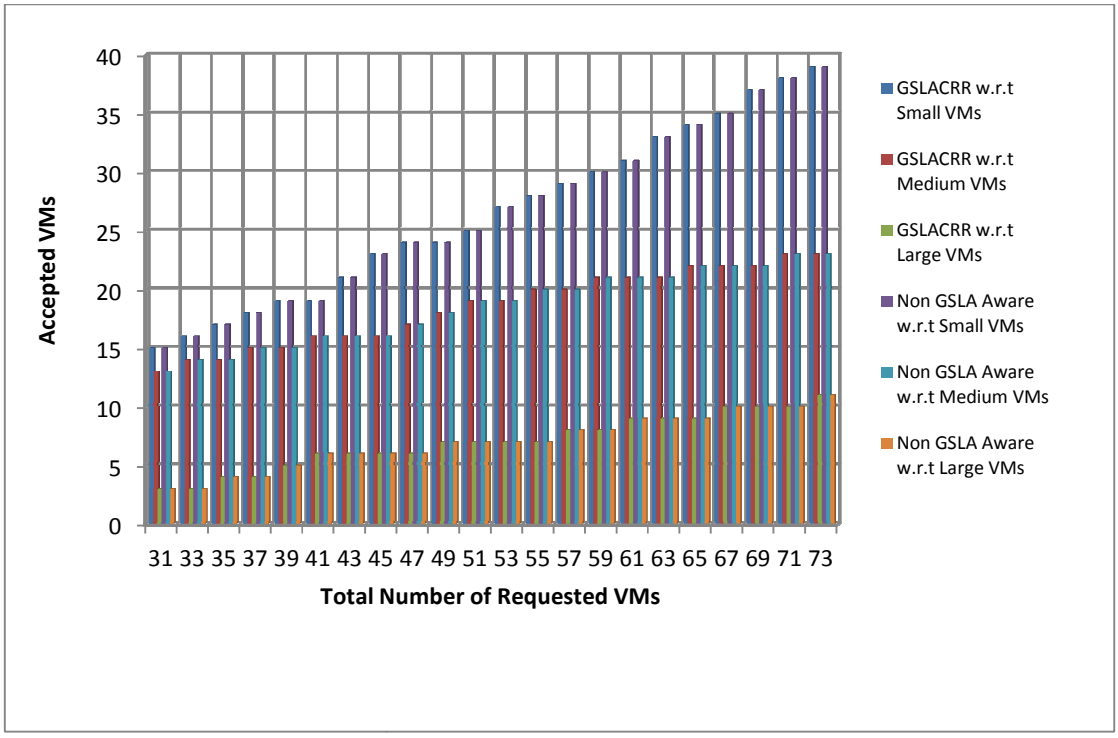
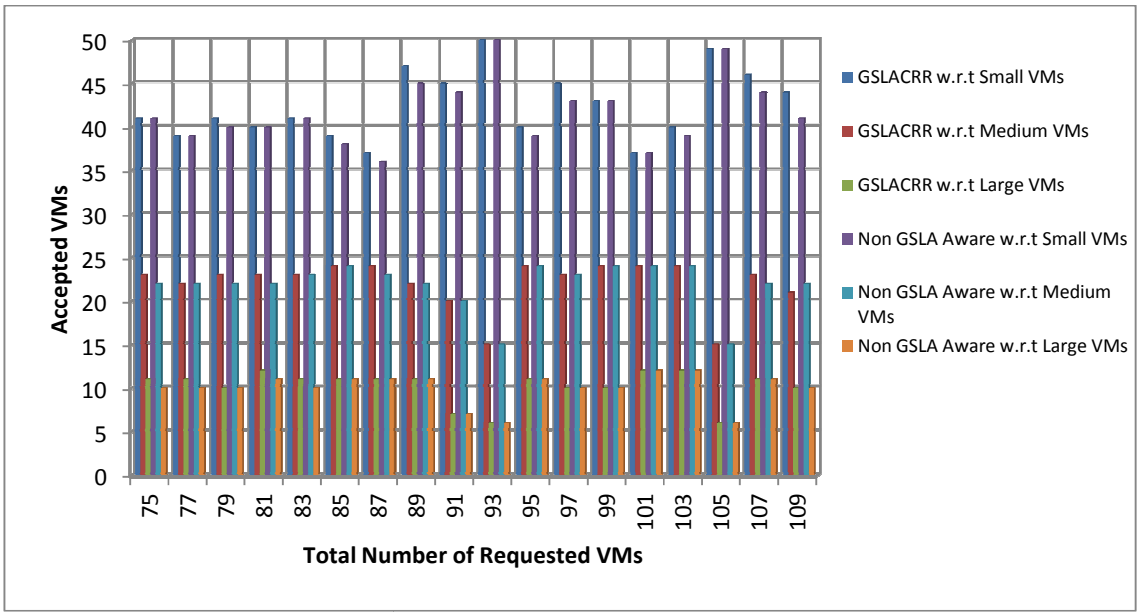


Figure 6.8: Number of PMs used with respect to VMs



(a) Test 1 with low number of VMs request



(b) Test 2 with high number of VMs request

Figure 6.9: The experimental results with different types of workload

6.3.3 Statistical Analysis of Results

For statistical validation, a t-test has been conducted where the MEC of GSLACRR algorithm is compared with the MEC of another non GSLA aware algorithm. The null hypothesis is that the MECs of the two algorithms are the same. The comparison results of the MEC of the GSLACRR algorithm with the non GSLA aware algorithm are reported in Table 6.3. The results are found significant at 0.05 levels. The comparison results of GSLACRR algorithm with the non GSLA aware for the AVM metric are reported in Table 6.4. The t-test results in Tables 6.3 and 6.4 clearly indicate that the proposed algorithm is significantly different from the non GSLA aware algorithm with respect to the MEC and AVM metrics.

6.4 Effects of the Number of Resources

There is a trade off accepted VM requests and energy conservation. Experimental results show that the proposed ECRS algorithm performs better in both scenarios in comparison to the Green Unbalanced approach. When the numbers of VM requests are less in proportion to available infrastructure, the proposed algorithm saves power by consolidating VMs on to fewer physical machines in addition taking care of the VM migration overhead.

Table 6.3 : t-test: paired two sample for means with respect to the MEC metric

	GSLACRR	Non GSLA Aware
Mean	67.91966834	68.95377817
Variance	356.854887	293.9093177
Observations	40	40
Pearson Correlation	0.989967185	
Hypothesized Mean Difference	0	
df	39	
t Stat	-2.116412915	
P(T<=t) one-tail	0.020372278	
t Critical one-tail	1.684875122	
P(T<=t) two-tail	0.040744555	
t Critical two-tail	2.022690901	

Table 6.4: t-test: paired two sample for means with respect to the AVM metric

	GSLACRR	Non GSLA Aware
Mean	62.2	61.675
Variance	224.7282051	209.5224
Observations	40	40
Pearson Correlation	0.998854437	
Hypothesized Mean Difference	0	
df	39	
t Stat	4.16202092	
P(T<=t) one-tail	8.40961E-05	
t Critical one-tail	1.684875122	
P(T<=t) two-tail	0.000168192	
t Critical two-tail	2.022690901	

As the number of user requests increased, the existing algorithms reject new VM requests beyond the available infrastructure capacity. The proposed algorithms address the increased VMs requests tactically. The results reveal that the proposed algorithms perform better in terms of acceptance of VM requests as well as power consumption. The proposed approach provides a significant edge at the data center level. Let us take the example of a micro data center with 1000 servers, each server drawing 200 watts of power (a reasonable ballpark guess, as it depends upon the server class, virtualization, server usage level, type of application running etc.); with the Green Unbalanced approach, a data center with 1000

servers ultimately consumes 200 KW. By using the ECRS algorithm, the energy consumption can be reduced by 10 KW or even more as a result of less power consumption by the cooling infrastructure involved in cooling the servers. Figure 6.10 shows the effect on power consumption in case of data centers having servers varying from 10 to 100000.

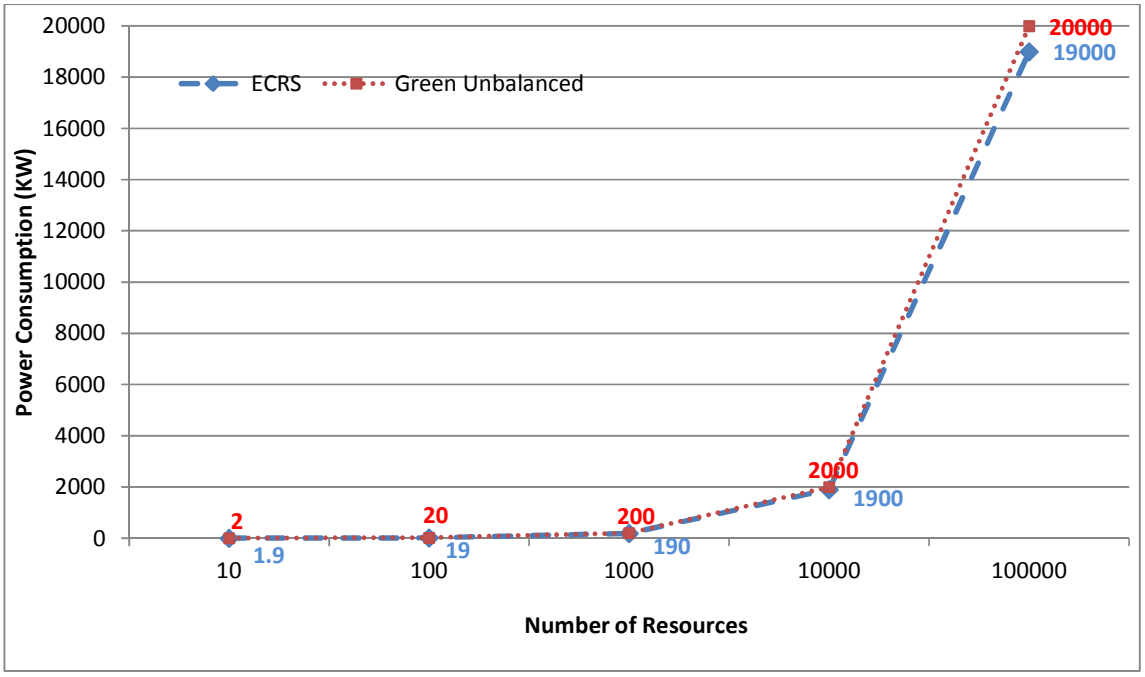


Figure 6.10: Effect of number of resources/servers on the power consumption in a data center

6.5 Summary

In chapter 6, the proposed algorithms have been validated and compared with existing ones. Several experiments were conducted with different number of VM requests, and VM types, and executed according to the different schedule patterns. Experimental results have been collected from the implementation and explained in chapter 5. To get statistically significant differences of the existing and proposed algorithms, paired samples *t*-test has been conducted. It has been established from the experiments that the proposed approach performs better in terms of energy consumption and the number of user VM requests. It also paves the way for future Clouds.

Chapter 7

Conclusion and Future Scope

This chapter gives the concluding remarks on the work carried out in this thesis. It highlights the main contributions of this research work. Due to the heterogeneous nature of the resource requests, resource management remains a major challenge for cloud computing. A number of issues have been identified and addressed. This thesis focuses on reduction in the energy consumption both at the data center level as well as at the SLA level. A number of other issues such as host reliability, user acceptance, resource prediction, and resource scheduling have been addressed. An energy efficient resource scheduling algorithms have been designed, implemented and tested in this thesis. To demonstrate the applicability and validation of the algorithms, an energy efficient cloud service framework, named ACA-Cloud, has been designed and developed. This chapter starts with explaining the outcome of each chapter and then discusses contribution the thesis in bringing energy efficiency to the cloud environment. In the end, the future scope of the work has been discussed.

7.1 Conclusion

The thesis, "Energy Efficient Resource Scheduling Algorithms for Cloud Computing" addresses the energy efficiency challenges of cloud computing. It focuses on the issue of the energy efficient usage of data center resources.

Green computing has emerged as an important discipline of cloud computing. The introductory chapter gives an overview of cloud computing, its evolution, benefits, and the types of cloud. It explains further the potential threat of large energy consumption (power and heat) and emission of carbon footprints into the environment as a result of the increasing expansion of data centers, worldwide. Further, it highlights the current energy efficient cloud computing mechanisms and technologies available to address these challenges. Literatures Review, chapter 2, focuses on the existing work on energy efficient cloud computing. It mainly highlights the prior work of energy efficient resource scheduling and the SLA based energy aware scheduling, used for cloud environment. After an extensive review, it becomes discernible that the main challenge of energy efficiency from the context of many organisations such as public sector units, government sectors, universities/ institutes etc. has not been addressed yet. To address this challenge, the research problem of energy efficient resource scheduling on the case study of the real problems faced by the universities has been formulated.

To address the issues identified in chapter 2, an energy efficient resources scheduling algorithm for the cloud environment, ECRS, has been proposed in chapter 3. It focuses on energy efficient VM placement, migration and consolidation among the servers. To demonstrate the working and usefulness of the algorithm, a cloud framework, named ACA-Cloud, has been designed and implemented. Further, a number of system models for the proposed solution have been presented and discussed in detail.

It is a challenge to involve users in sustainable computing while satisfying the cloud service provider as well as the users by minimizing the cost and energy consumption, and maximizing the resources utilization. To address these issues, a GSLACRR algorithm has been proposed in chapter 4. The proposed algorithm reduces the energy consumption of a data center through user negotiated energy saving strategies. Energy consumption is minimized through rounds of negotiation and inclining the user towards eco-system cloud services.

Chapter 5 provides the design and implementation details of the energy efficient cloud framework, named ACA-Cloud, which express the working and usefulness of the proposed algorithms. To analyse the complete requirements, design, and implementation, UML diagrams have been prepared. They describe the various components, architectural design, and development of the proposed cloud service framework. The experimental results of the proposed algorithms have been shown and compared with existing algorithms in chapter 6. The analyses of the evaluated results show that the proposed algorithms reduce overall energy consumption and increase the user acceptability in all scenarios. A statistical test has also been conducted to get the significant differences between the proposed and the existing algorithms. The thesis contributes as follows:

i. The thesis presents detailed literature review of the work done in the area of energy efficient cloud computing.

ii. The thesis identifies the potential of energy conservation, cost benefits, and efficient utilization of resources of IT infrastructure for many organisations such as public sector units, government sectors, universities/institutes etc. It presents the development and implementation of the energy efficient cloud service framework, named ACA-Cloud. The ACA-Cloud supports

    a) GSLA-based VM admission control

    b) User negotiated eco-system cloud services

    c) Advance resource reservation

    d) Energy efficient resource scheduling

    e) Resources prediction to meet current workload and a margin of the growth

iii. An energy efficient resource scheduling algorithm, named ECRS, has been designed that addresses the issues of performance and energy minimization.

iv. Design an SLA based VM admission control and scheduling algorithm that inclines the users towards sustainable computing. Its objective is to offer cloud resource services in an energy efficient manner with cost benefits to the users, which are not covered in traditional SLAs.

v. To validate the proposed algorithms, the algorithms have been implemented on heterogeneous testbed. The experimental results have been performed and analysed with different test cases. The results show that the proposed ECRS algorithm performs better in comparison to Green Unbalanced approach with respect to energy efficiency when there are a small number of user VM requests arrive. It also performs better by accepting more number of VM requests when there are a large combination of large and medium sized VM requests arrive. The results reveal that the proposed ECRS algorithm outperformed by 5% and 1.5% for power saving and accepting VM requests, respectively.

vi. Another proposed GSLACRR algorithm also performs much better in terms of power consumption and handle more number of user VM requests. The proposed approach devises the provision of resources through negotiation with the users. The proposed GSLACRR algorithm leverages the free time slots and free weekend days, and pursues the users to use resources in these time slots with cost and resource benefits. This is beneficial in particular for experimental and batch kind of applications.

7.2 Future Scope

This thesis contributes substantially in energy efficient resource scheduling and inclining users toward sustainable computing through Green SLA, there are open research challenges that can be further extended in the future for the advancement in this area:

i. Data centers can be made more environmentally sustainable by powering them through renewable energy sources such as sunlight, wind, rain, tides, waves, and geothermal heat. It will place caps on the carbon emissions. In the future, this can be extended by incorporating renewable energy percentage into users' contracts in lieu of which users will have an incentive to use more resources if they are powered by sustainable energy sources.

ii. The proposed solution accounts for scenarios where in all the machines are on the same rack and there is no communication between the VMs. In a large private cloud, there are many racks of servers, and in case of scientific and engineering applications, the VMs also communicate with each other. Communication

between different VMs at distant locations becomes a data transfer overhead and also consumes significant amounts of energy due to the involvement of network devices, which ultimately impacts the performance. Thus, another potential improvement can be introduced into VM migrations so as to place VMs in closely located machines based upon traffic volume, data transfer cost, and energy consumption parameters.

iii. A significant part of the power consumed by servers' converts to heat. Additionally, high density rack servers (1U) lead to high heat dissipation. An increase of the local ambient temperature inside a computing node reduces the reliability and life time of the machines [46] which in turn increases the chances of failure. To minimize the cooling operational cost and to enhance the reliability of the machines, there is need to continually monitor the thermal state of the machines and the VM migration from a machine when it is heated above its permissible limit. Therefore, this work can be further investigated to reduce the cooling cost, VM reallocation overhead, and performance degradation.

iv. In the current framework, there are three types of users: students/research scholars, teachers and administrators. In the future, more types of users from different domains can be included.

v. The 'ACA-Cloud' framework offers IaaS to end users. On the top of IaaS, PaaS can be provided. An AppScale[150] as PaaS can be used. It is an open source cloud platform that allows the developers to quickly and efficiently build web and mobile apps. It permits designing and developing applications using any programming language like Go [151], Java [133], PHP [134], Python [152]. Further, it can be deployed on any cloud platform such as Amazon EC2 [12], CloudStack [153], DigitalOcean [154], Eucalyptus [155], Google Compute Engine [16], Kernel-based Virtual Machine (KVM) [137], Microsoft Azure [21], OpenStack [156], RackSpace [17], SoftLayer [157], Xen [108] etc. It also separates the application logic from its service ecosystem to let developers and cloud administrators have exceptional control over application deployment, data storage, backup and migration.

vi. Various pricing policies can be integrated to outsource the cloud resources.

References

[1] R. Buyya, C.S. Yeo and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering it Services as Computing Utilities," Proc. IEEE Int',l Conf. High Performance Computing and Communications (HPCC ',08), pp. 5-13, Sept. 2008. doi: 10.1109/HPCC.2008.172.

[2] R. Buyya, C. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599-616, 2009.

[3] A. Fox., R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin and I. Stoica. "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, 28, p.13, 2009.

[4] R. Buyya, C. Vecchiola and S.T. Selvi, "Mastering cloud computing: foundations and applications programming." Newnes. 2013.

[5] M. Ebbers, W. O'Brien and B. Ogden, "Introduction to the new mainframe." [Armonk, N.Y.]: IBM, Redbooks, 2012.

[6] B.M. Leiner, M. Barry et al. "A brief history of the Internet." ACM SIGCOMM Computer Communication Review 39, no. 5 pp. 22-31, 2009.

[7] Internetsociety.org, "Brief History of the Internet - Internet Timeline | Internet Society," 2016. [Online]. Available: http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet. [Accessed: 12- Jan- 2016].

[8] I. Foster and C. Kesselman, "The Grid 2: Blueprint for a new computing infrastructure" Elsevier, 2003.

[9] Gridcomputing.com, "Grid Computing Info Centre (GRID Infoware)," 2016. [Online]. Available: http://www.gridcomputing.com/. [Accessed: 12- Jan- 2016].

[10] R. Buyya, J. Broberg and A. Goscinski, "Cloud computing: principles and paradigms," vol. 87, John Wiley & Sons, 2010.

[11] GlobalDots - Available: http://www.globaldots.com. [Accessed: 12- Jan- 2016].

[12] Amazon Web Services, Inc., "Elastic Compute Cloud (EC2) Cloud Server & Hosting – AWS," 2016. [Online]. Available: https://aws.amazon.com/ec2/. [Accessed: 12- Jan- 2016].

[13] Amazon Web Services, Inc., "What Is Amazon S3? - Amazon Simple Storage Service," Docs.aws.amazon.com, 2016. [Online]. Available: http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html.[Accessed: 12- Jan- 2016].

[14] Azure.microsoft.com, "What is Azure? - IaaS + PaaS | Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-in/overview/what-is-azure/. [Accessed: 12- Jan- 2016].

[15] Ibm.com, "IBM - Cloud Infrastructure & Cloud Management," 2016. [Online]. Available: https://www.ibm.com/cloud-computing/infrastructure. [Accessed: 12- Jan- 2016].

[16] Google Developers, "Compute Engine - IaaS," 2016. [Online]. Available: https://cloud.google.com/compute/. [Accessed: 12- Jan- 2016].

[17] Rackspace.com, "Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS | Knowledge Center | Rackspace Hosting," 2016. [Online]. Available: https://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas. [Accessed: 12- Jan- 2016].

[18] Amazon Web Services, Inc., "AWS Toolkit for Eclipse Documentation," 2016. [Online]. Available: https://aws.amazon.com/documentation/awstoolkiteclipse/. [Accessed: 12- Jan- 2016].

[19] Aws.amazon.com, "Release Notes: Amazon Web Services," 2016. [Online]. Available: https://aws.amazon.com/releasenotes/AWS-CloudFormation. [Accessed: 12- Jan- 2016].

[20] Google Developers, "App Engine - Platform as a Service," 2016. [Online]. Available: https://cloud.google.com/appengine/. [Accessed: 12- Jan- 2016].

[21] Azure.microsoft.com, "Microsoft Azure: Cloud Computing Platform & Services," 2016. [Online]. Available: http://azure.microsoft.com/. [Accessed: 12- Jan- 2016].

[22] Salesforce.com, "What is Platform as a Service (PaaS)," 2016. [Online]. Available: https://www.salesforce.com/paas/overview/. [Accessed: 12- Jan- 2016].

[23] Google.co.in, "Google Docs - create and edit documents online, for free.," 2016. [Online]. Available: https://www.google.co.in/docs/about/. [Accessed: 12- Jan- 2016].

[24] Microsoft.com, "Accessibility in Microsoft Office 365," 2016. [Online]. Available: https://www.microsoft.com/enable/products/office365/. [Accessed: 12- Jan- 2016].

[25] Dropbox, "Dropbox," 2016. [Online]. Available: https://www.dropbox.com/. [Accessed: 12- Jan- 2016].

[26] Salesforce.com, "Cloud Computing, Customer Relationship Management, (CRM) - Salesforce.com India," 2016. [Online]. Available: http://www.salesforce.com/in/. [Accessed: 12- Jan- 2016].

[27] Blue Jeans Network, "Blue Jeans Network," 2016. [Online]. Available: http://bluejeans.com/. [Accessed: 12- Jan- 2016].

[28] OnSIP Support, "Vidtel Multi-Party Video Conferencing," 2016. [Online]. Available: https://support.onsip.com/hc/en-us/articles/203672484-Vidtel-Multi-Party-Video-Conferencing. [Accessed: 12- Jan- 2016].

[29] Linthicum, S. David "Cloud computing and SOA convergence in your enterprise: a step-by-step guide." Pearson Education, 2009.

[30] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Information Technology Laboratory, Technical Report 53, no. 6.2009.

[31] B. Sosinsky," Cloud computing bible," John Wiley & Sons; 2010

[32] Salesforce UK Blog, "Why Move To The Cloud ," 2016. [Online]. Available: http://www.salesforce.com/uk/socialsuccess/cloud-computing/why-move-to-cloud-10-benefits-cloud-computing.jsp. [Accessed: 12- Jan- 2016].

[33] Carl Wiese, "The Return on Collaboration: Assessing the Value of Today's Collaboration Solutions," CISCO white paper, 2010

[34] Emersonnetworkpower.com, "State of the Data Center 2011," 2016. [Online]. Available: http://www.emersonnetworkpower.com/en-US/Solutions/infographics/Pages/2011DataCenterState.aspx. [Accessed: 12- Jan- 2016].

[35] Gartner.com, "Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions," 2016. [Online]. Available: http://www.gartner.com/it/page.jsp?id=503867. [Accessed: 12- Jan- 2016].

[36] M .Webb et.al.,"Smart2020:Enabling the low carbon economy in the information age," TheClimateGroup, London, 2008

[37] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing," Computer Journal, vol. 53, no. 7, pp. 1045–1051, 2010.

[38] S. Rivoire, M. A. Shah, P. Ranganathan and C. Kozyrakis, "A balanced energy-efficiency benchmark," in Proc. of the 2007 ACM SIGMOD International Conference on Management of Data, ACM, NY, USA, pp. 365-376, 2007.

[39] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," Computer, vol. 40, no. 12, pp. 33–37, 2007.

[40] J. Kaplan, W. Forrest and N. Kindler, "Revolutionizing Data Center Energy Efficiency," Technical report, McKinsey & Company, 2008.

[41] Ministry of Economy, Trade and Industry, Government of Japan. "Establishment of the Japan Data Center". Press Release, 2008.

[42] Federal Data Center Consolidation Initiative (FDCCI) "Data Center Consolidation and Optimization," USA, 2010, Available online: https://cio.gov/drivingvalue/data-center-consolidation/

[43] Thegreengrid.org, "The Green Grid › About The Green Grid," 2016. [Online]. Available: http://www.thegreengrid.org/about-the-green-grid.aspx. [Accessed: 12-Jan- 2016].

[44] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges, " in Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.

[45] A. J. Younge, G. Laszewski, L. Wang, S. Lopez-Alarcon and W. Carithers "Efficient Resource Management for Cloud Computing Environments," In Proc. of the IEEE International Green Computing Conference (IGCC), Chicago, IL, pp. 357-364 , 2010

[46] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer System, vol. 28, no. 5, pp. 755–768, 2012.

[47] M.T. Chaudhry, T.C. Ling, A. Manzoor, S.A. Hussain, and J. Kim, "Thermal-Aware Scheduling in Green Data Centers". ACM Computing Surveys (CSUR), vol. 47, no. 3, pp.39, 2015

[48] R. Sharma, C. Bash, C. Patel, R. Friedrich and J. Chase, "Balance of Power: Dynamic Thermal Management for Internet Data Centers," IEEE Internet Computing, vol. 9, no. 1, pp. 42-49, 2005.

[49] J. D. Moore, J. S. Chase, P. Ranganathan & R.K. Sharma "Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers," In USENIX annual technical conference, General Track, pp. 61-75, 2005.

[50] A. Berl, H. de Meer, "An Energy-Efficient Distributed Office Environment," in First IEEE International Conference on Emerging Network Intelligence, pp.117-122, 2009, doi: 10.1109/EMERGING.2009.13

[51] A. Berl, H. Hlavacs, R. Weidlich, M. Schrank, and H. de Meer, "Network Virtualization," in Future Home Environments. Proc. Int. Workshop on Distributed Systems: Operations and Management (DSOM09), Italy, Lecture Notes in Computer Science. Springer, 2009

[52] L. Minas and B. Ellison, "Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers," Intel Press, pp. 27-44, 2009

[53] K. Francis and P. Richardson "Green Maturity Model for Virtualization' in Architecture Journal, Microsoft, vol. 18, pp. 9-15.

[54] D. Yuan, Y. Yanga, X. Liua and J. Chena , "A data placement strategy in scientific cloud workflows," Future Generation Computer Systems ,vol. 26, no. 8, pp. 1200-1214, 2010

[55] C. Liu, X. Qin, S. Kulkarni, C. Wang, S. Li, A. Manzanares, and S. Baskiyar, "Distributed Energy-Efficient Scheduling for Data-Intensive Applications with Deadline Constraints on Data Grids," in Proc. IEEE International Performance Computing and Communications Conference (IPCCC), pp. 26 - 33, 2008. DOI: 10.1109/PCCC.2008.4745123

[56] T. Mastelic, A. Oleksia, H. Claussen, I. Brandic., J. M. Pierson, & A. V. Vasilako. "Cloud Computing: Survey on Energy Efficiency," ACM Computing Surveys (CSUR) 47, no. 2, pp. 33. 2015.

[57] S. Zeadally, S. Khan and N. Chilamkurti, "Energy-efficient networking: past, present, and future," J Supercomput, vol. 62, no. 3, pp. 1093-1118, 2012.

[58] Jing Nie; Zheng Zhou, "An energy based power-aware routing protocol in ad hoc networks," IEEE International Symposium on Communications and Information Technology, vol.1, pp.280-285, 2004, doi: 10.1109/ISCIT.2004.1412854.

[59] S. Mahfoudh, P. Minet "Survey of Energy-Efficient strategies in wireless ad hoc and sensor networks," In Proceedings of 7th international conference on networking (ICN'08), Cancun, Mexico, 2008.

[60] C.E. Jones,M. Sivalingam, P. Aggrawal, J. Chen, "A survey of techniques for Energy-Efficient network protocols for wireless networks," Wireless Network, vol. 7, no. 4, pp.343–358, 2001.

[61] I.F. Khan, M.Y. Javed, "A Survey on Routing Protocols and Challenge of Holes in Wireless Sensor Networks," International Conference on Advanced Computer Theory and Engineering, 2008. ICACTE '08., pp.161-165,Dec.2008. doi: 10.1109/ICACTE.2008.146.

[62] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," Ad Hoc Networks, vol. 3, no. 3, pp. 325-349, 2005.

[63] A. Sankar, Liu Zhen, "Maximum lifetime routing in wireless ad-hoc networks," IEEE Computer and Communications Societies Twenty-third Annual Joint Conference on INFOCOM 2004, vol. 2, pp. 1089-1097, 2004. doi: 10.1109/INFCOM.2004.1356995.

[64] N. Vasić, and D. Kostić, "Energy-aware traffic engineering," In ACM Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, pp. 169-178. 2010.

[65] N. Vasić et al. "Identifying and using energy-critical paths, " in ACM Proceedings of the Seventh Conference on emerging Networking Experiments and Technologies. pp. 18, 2011.

[66] Z. Mingui, Y. Cheng Yi, L. Bin, Z. Beichuan, "GreenTE: Power-aware traffic engineering," in 18th IEEE International Conference on Network Protocols (ICNP), 2010, pp.21-30, 2010, doi: 10.1109/ICNP.2010.5762751

[67] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01). pp. 1548–1557, 2001.

[68] N. Laoutaris, M. Sirivianos, X. Yang and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, p. 74-85, 2011.

[69] N. Rizvandi, J. Taheri and A. Zomaya, "Some observations on optimal frequency selection in DVFS-based energy consumption minimization," Journal of Parallel and Distributed Computing, vol. 71, no. 8, pp. 1154-1164, 2011.

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

[71] K. H. Kim, A. Beloglazov, & R. Buyya "Power-aware provisioning of cloud resources for real-time services," In Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science, pp. 1, 2009.

[72] E. Kan, W. Chan and T. Tse, "EClass: An execution classification approach to improving the energy-efficiency of software via machine learning," Journal of Systems and Software, vol. 85, no. 4, pp. 960-973, 2012.

[73] M. Guzek et al. "Impact of voltage levels number for energy-aware bi-objective dag scheduling for multi-processors systems," Advances in Information Technology. Springer Berlin Heidelberg, pp. 70-80. 2012.

[74] G. Laszewski, L. Wang, A. J. Younge, and X. He, "Power-aware scheduling of virtual machines in DVFS-enabled clusters," in IEEE International Conference on Cluster Computing and Workshops, pp.1-10, 2009.

[75] O. VanGeet "Best practices guide for energy-efficient data center design," EERE Publication and Product Library, No. DOE/GO-102010-2956. 2010

[76] Google.com, "Renewable energy – Data Centers – Google," 2016. [Online]. Available: http://www.google.com/about/datacenters/renewable/. [Accessed: 13-Jan- 2016].

[77] E. Oró, V. Depoorter, A. Garcia and J. Salom, "Energy efficiency and renewable energy integration in data centres. Strategies and modelling review," Renewable and Sustainable Energy Reviews, vol. 42, pp. 429-445, 2015.

[78] N. Deng, C. Stewart, & J. Li "Concentrating renewable energy in grid-tied datacenters" In IEEE Proceeding of the International Symposium on Sustainable Systems and Technology (ISSST) pp. 1-6, 2011

[79] Google.co.in, "Efficiency: How we do it – Data Centers – Google," 2016. [Online]. Available: http://www.google.co.in/about/datacenters/efficiency/internal/index.html#water-and-cooling. [Accessed: 13- Jan- 2016].

[80] S. Garg, C. Yeo, A. Anandasivam and R. Buyya, "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers," Journal of Parallel and Distributed Computing, vol. 71, no. 6, pp. 732-749, 2011.

[81] G. Copil et al. "Cloud SLA negotiation for energy saving—A particle swarm optimization approach, " In IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 289-296, 2012.

[82] G. von Laszewski, & L. Wang, "GreenIT service level agreements ," In Grids and Service-Oriented Architectures for Service Level Agreements Springer US, pp. 77-88, 2010.

[83] S. S. Manvi and G. Krishna Shyam, “Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey,” J. Network and Computer Application, vol. 41, no. 1, pp. 424–440, 2014.

[84] Kavli Foundation and the Institute for Energy Efficiency, "Scalable, Energy-Efficient Data Centers and Clouds," Santa Barbara, 2016.

[85] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," In Proceedings of the 2008 conference on Power aware computing and systems, vol. 10. 2008.

[86] A. Verma, G. Dasgupta, T.K. Nayak, P. De and R. Kothari, "Server workload analysis for power minimization using consolidation" In Proceedings of USENIX Annual technical conference, USENIX Association, pp. 28-28, 2009.

[87] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-efficient virtual machine provision algorithms for cloud systems," in Proc. of the IEEE UCC, pp. 81-88, 2011.

[88] D. M. Quan, F. Mezza, D. Sannenli, and R. Giafreda, "T-Alloc: A practical energy efficient resource allocation algorithm for traditional data centers," Future Generation Computer System, vol. 28, no. 5, pp. 791–800, 2012.

[89] A. Beloglazov & R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, vol 24, no.13, pp.1397-1420, 2012.

[90] P. Merz and B. Freisleben, "Greedy and local search heuristics for unconstrained binary quadratic programming, " Journal of Heuristics, vol. 8, no. 2, pp.197-213, 2002.

[91] T. Setzer, A. Stage,"Decision support for virtual machine reassignments in enterprise data centers, " IEEE/IFIP Network Operations and Management Symposium Workshops, Osaka, Japan, pp. 88-94, , 2010.

[92] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," J. Supercomputing, vol. 60, no. 2, pp. 268–280, 2012.

[93] C. H. Hsu, K. D. Slagter, S. C. Chen, and Y. C. Chung, "Optimizing energy consumption with task consolidation in clouds," Information Sciences, vol. 258, pp. 452–462, 2014.

[94] F. Dressler and O. Akan, "A survey on bio-inspired networking," Computer Networks, vol. 54, no. 6, pp. 881-900, 2010.

[95] M. Meisel, V. Pappas and L. Zhang, "A taxonomy of biologically inspired research in computer networking," Computer Networks, vol. 54, no. 6, pp. 901-916, 2010.

[96] E. Feller, L. Rilling, and C. Morin, “Energy-aware ant colony based workload placement in clouds,” in Proc. of the 12th IEEE/ACM Int. Conf. Grid Comput. Grid 2011, pp. 26–33, 2011.

[97] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29–41, 1996.

[98] M. Dorigo and L. Gambardella, "Ant colonies for the travelling salesman problem," Biosystems, vol. 43, no. 2, pp. 73-81, 1997.

[99] N. J. Kansal & I. Chana, "Artificial bee colony based energy aware resource utilization technique for cloud computing," Concurrency and Computation: Practice and Experience, vol. 27, no. 5, pp. 1207-1225, 2015.

[100] I. Takouna, W. Dawoud, & C. Meinel, "Energy efficient scheduling of HPC-jobs on virtualize clusters using host and VM dynamic configuration, "ACM SIGOPS Operating Systems Review, vol. 46, no. 2, pp. 19-27, 2012.

[101] R. N. Calheiros and R. Buyya, “Energy-Efficient Scheduling of Urgent Bag-of-Tasks Applications in Clouds through DVFS,” in Proc. of the IEEE 6th Int. Conf. Cloud Computing Technology and Science, pp. 342–349, 2014.

[102] L. Lefèvre and A. C. Orgerie, "Towards Energy Aware Reservation Infrastructure for Large-Scale Experimental Distributed Systems," Parallel Processing Letters, vol. 21, no. 02, pp. 419-433, 2009.

[103] L. Lefèvre and A. C. Orgerie, “Designing and evaluating an energy efficient Cloud,” J. Supercomputing, vol. 51, no. 3, pp. 352–373, 2010.

[104] A. C. Orgerie, L. Lefèvre, M. D De Assuncao, "Energy aware clouds". In Grids, Clouds and Virtualization, pp. 143-166. Springer London, 2011.

[105] A. C. Orgerie, L. Lefèvre, "Energy-Efficient Reservation Infrastructure for Grids, Clouds, and Networks," Energy-Efficient Distributed Computing Systems, pp.133-161, 2012.

[106] Z. Xiao, W. Song, & Q. Chen "Dynamic resource allocation using virtual machines for cloud computing environment" , IEEE Transactions on Parallel and Distributed Systems, Vol. 24, no. 6, 1107-1117, 2013

[107] N. Kim, J. Cho, and E. Seo, “Energy-credit scheduler: An energy-aware virtual machine scheduler for cloud systems,” Future. Generation Computer System, vol. 32, no. 1, pp. 128–137, 2014.

[108] Xenproject.org, " Xen Open Source Hypervisor ," 2016. [Online]. Available: http://www.xenproject.org/. [Accessed: 13- Jan- 2016].

[109] Y. Ding, X. Qin, L. Liu, and T. Wang, “Energy efficient scheduling of virtual machines in cloud with deadline constraint,” Future Generation Computer Systems. vol. 50, pp. 62–74, 2015

[110] Y. Gao, H. Guan, Z. Qi, T. Song, F. Huan and L. Liu, "Service level agreement based energy-efficient resource management in cloud data centers," Computers & Electrical Engineering, vol. 40, no. 5, pp. 1621-1633, 2014.

[111] C. Xuedi, Li Kenli , L. Chubo Liu, Li Keqin, "SLA-based energy aware scheduling of precedence-constrained applications on DVFS-enabled clusters," in 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), pp.336-343, 2014.

[112] Y. Gao, H. Guan, Z. Qi, B. Wang and L. Liu, "Quality of service aware power management for virtualized data centers," Journal of Systems Architecture, vol. 59, no. 4, pp. 245-259, 2013.

[113] R. Basmadjian et al., “Fit4green-energy aware ict optimization policies,” in Proceedings of the COST Action IC0804 on Energy Efficiency in Large Scale Distributed Systems-1st Year , pp. 1–5, 2010.

[114] D. Borgetto, M. Maurer, G. Da-Costa, J.M. Pierson & I. Brandic, "Energy-efficient and SLA-aware management of IaaS clouds, " In ACM Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, pp. Article No. 25, 2012.

[115] Avtech.com, "AVTECH - Recommended Data Center Temperature & Humidity," 2016. [Online]. Available: http://www.avtech.com/About/Articles/AVT/NA/All/-/DD-NN-AN-TN/Recommended_Computer_Room_Temperature_Humidity.htm. [Accessed: 13- Jan- 2016].

[116] C. Belady, A. Rawson,J. Pfleuger and T. Cader, "Green grid data center power efficiency metrics: PUE and DCIE," Technical report, Green Grid, 2008.

[117] D. Kliazovich, P. Bouvry and S. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers," Journal of Supercomputing, vol. 62, no. 3, pp. 1263-1283, 2010.

[118] The Green Grid "A Framework for Data Center Energy Productivity, " White Paper #13 [Online] Available: http://www.thegreengrid.org/~/media/WhitePapers/WhitePaper13FrameworkforDataCenterEnergyProductivity5908.ashx?lang=en [Accessed: 13- Jan- 2016]

[119] D. Greenhill, "SWaP (Space, Watts and Performance) Metric, "Technical Report, Sun microsystems, [Online] Available: https://www.energystar.gov/ia/products/downloads/Greenhill_Pres.pdf [Accessed: 13- Jan- 2016]

[120] C. Bunse, S. Klingert, and T. Schulze, "Greenslas: Supporting energy-efficiency through contracts," In Energy Efficient Data Centers, Springer Berlin Heidelberg, pp. 54-68

[121] H. Rasheed, A. Rumpl, , O. Wäldrich, & W. Ziegler, "A standards-based approach for negotiating service QoS with cloud infrastructure providers," In eChallenges Conference. 2012

[122] M. Haque, K. Le, and Í. Goiri, "Providing Green SLAs in High Performance Computing Clouds," In International Green Computing Conference (IGCC), pp. 1-11, 2013

[123] C. Dupont, F. Hermenier, T. Schulze, R. Basmadjian, A. Somov and G. Giuliani, "Plug4Green: A flexible energy-aware VM manager to fit data centre particularities," Ad Hoc Networks, vol. 25, pp. 505-519, 2015.

[124] Y. Ajiro, A. Tanaka, "Improving packing algorithms for server consolidation," in Proceedings of the International Conference for the Computer Measurement Group (CMG), pp. 399–406.

[125] T. Ma, Y. Chu, L. Zhao and O. Ankhbayar, "Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm," IETE Technical Review, vol. 31, no. 1, pp. 4-16, 2014.

[126] S. Greenberg, E. Mills, B. Tschudi, and P. Rumsey, "Best practices for data centers: Results from benchmarking 22 data centers," In Proceedings of the 2006 ACEEE Summer Study on Energy Efficiency in Buildings, 2006.

[127] G. Katsaros, J. Subirats, J. Fitó, J. Guitart, P. Gilet and D. Espling, "A service framework for energy-aware monitoring and VM management in Clouds," Future Generation Computer Systems, vol. 29, no. 8, pp. 2077-2091, 2013.

[128] M. Buco, R. Chang, L. Luan, C. Ward, J. Wolf and P. Yu, "Utility computing SLA management based upon business objectives," IBM Syst. J., vol. 43, no. 1, pp. 159-178, 2004.

[129] A-C Orgerie, "An Energy-Efficient Reservation Framework for Large-Scale Distributed Systems." PhD thesis, Ecole Normale Supérieure de Lyon – France, 2011.

[130] Uml.org, "Unified Modeling Language (UML)," 2016. [Online]. Available: http://www.uml.org/. [Accessed: 13- Jan- 2016].

[131] Ubuntu.com, "About Ubuntu," 2016. [Online]. Available: http://www.ubuntu.com/about/about-ubuntu/our-philosophy. [Accessed: 13- Jan- 2016].

[132] Openvirtualizationalliance.org, "What is KVM | Open Virtualization Alliance," 2016. [Online]. Available: https://openvirtualizationalliance.org/what-kvm. [Accessed: 13- Jan- 2016].

[133] Oracle.com, "Java SE Community - Open-Source JDK," 2016. [Online]. Available: http://www.oracle.com/technetwork/articles/javase/opensourcejdk-jsp-136417.html. [Accessed: 13- Jan- 2016].

[134] Secure.php.net, "PHP: License Information," 2016. [Online]. Available: https://secure.php.net/license/. [Accessed: 13- Jan- 2016].

[135] Mysql.com, "MySQL," 2016. [Online]. Available: https://www.mysql.com/. [Accessed: 13- Jan- 2016].

[136] Libvirt.org, "libvirt: The virtualization API," 2016. [Online]. Available: http://libvirt.org/ [Accessed: 13- Jan- 2016].

[137] Linux-kvm.org, "KVM," 2016. [Online]. Available: http://www.linux-kvm.org /page/ Main_Page. [Accessed: 13- Jan- 2016].

[138] Vmware.com, "VMware Virtualization for Desktop & Server, Application, Public & Hybrid Clouds | VMware India," 2016. [Online]. Available: http://www.vmware.com/in. [Accessed: 13- Jan- 2016].

[139] Libvirt.org, "libvirt: Applications using libvirt," 2016. [Online]. Available: https://libvirt.org/apps.html. [Accessed: 13- Jan- 2016].

[140] Eclipse.org, 2016. [Online]. Available: https://eclipse.org/org/. [Accessed: 13- Jan- 2016].

[141] Netbeans.org, "Open Source," 2016. [Online]. Available: https://netbeans.org /about /os/. [Accessed: 13- Jan- 2016].

[142] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, "Live migration of virtual machines," In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, USENIX Association, vol. 2, pp. 273-286. 2005.

[143] Help.ubuntu.com, "SettingUpNFSHowTo - Community Help Wiki," 2016. [Online]. Available: https://help.ubuntu.com/community/SettingUpNFSHowTo. [Accessed: 13- Jan- 2016].

[144] Power consumption of Dell PowerEdge r710 server 2014. [Online]. Available: http://www.dell.com/downloads/global/products/pedge/en/dell_poweredge_r710_2p_e5620_870w_energy_star_data_sheet.pdf [Accessed: 02- Feb- 2016]

[145] Power consumption of Dell PowerEdge 2900 server 2014. [Online]. Available: http://www.intel.com/content/dam/www/public/us/en/documents/datasheets/quad-core-xeon-5400-datasheet.pdf [Accessed: 02- Feb- 2016]

[146] I. Gavrichenkov, CPU Benchmark 2014. [Online]. Available: http://www.xbitlabs.com/articles/cpu/display/cpu-benchmark-mainstream_10.html [Accessed: 02- Feb- 2016]

[147] I. Gavrichenkov (2014) Power consumption of IntelCore i5 processor. [Online]. Available: http://www.xbitlabs.com/articles/cpu/display/core-i5-2500-2400-2300_10.html. [Accessed: 02- Feb- 2016]

[148] Beloglazov, "Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing," PhD thesis, Department of Computing and Information Systems, The University of Melbourne, Australia, 2013.

[149] M.G Patterson, "What is energy efficiency: Concepts, indicators and methodological issues," Energy policy, vol. 24, no. 5, pp.377-390. 1996

[150] Appscale.com, "AppScale: Freedom for Your Applications," 2014. [Online]. Available: http://www.appscale.com/. [Accessed: 13- Jan- 2016].

[151] Golang.org, "The Go Programming Language," 2016. [Online]. Available: https://golang.org/. [Accessed: 13- Jan- 2016].

[152] Python.org, "Welcome to Python," 2015. [Online]. Available: https://www.python.org/. [Accessed: 13- Jan- 2016].

[153] Apache Cloudstack, "Apache Cloudstack," 2016. [Online]. Available: https://cloudstack.apache.org/. [Accessed: 13- Jan- 2016].

[154] Digitalocean.com, "Simple Cloud Infrastructure for Developers | DigitalOcean," 2016. [Online]. Available: https://www.digitalocean.com/. [Accessed: 13- Jan- 2016].

[155] www8.hp.com, "HPE Helion Eucalyptus | Hewlett Packard Enterprise," 2016. [Online]. Available: http://www8.hp.com/us/en/cloud/helion-eucalyptus-overview.html. [Accessed: 13- Jan- 2016].

[156] Openstack.org, "Home » OpenStack Open Source Cloud Computing Software," 2016. [Online]. Available: https://www.openstack.org/. [Accessed: 13- Jan- 2016]

[157] Softlayer.com, "SoftLayer | Cloud Servers, Storage, Big Data, & More IAAS Solutions," 2016. [Online]. Available: http://www.softlayer.com/. [Accessed: 13- Jan- 2016].

List of Publications

International Journal:

P1. S. Goyal, S. Bawa and B. Singh, “Energy optimized resource scheduling algorithm for private cloud computing,” *International Journal of Ad Hoc and Ubiquitous Computing,* Inderscience. (Accepted).

P2. S. Goyal, S. Bawa and B. Singh, “Green Service Level Agreement (GSLA) framework for Cloud Computing," *Journal of Computing*, Springer. DOI: 10.1007/s00607-015-0481-6.

P3. S. Goyal, S. Bawa and B. Singh, "Energy-Efficient Resource Scheduling in Cloud Computing: Policies, Algorithms and Research Challenges," *IEEE Computer Journal, SCI.* (Communicated).

International Conferences:

P4. S. Goyal, S. Bawa and B. Singh, “Experimental Comparison of Three Scheduling Algorithms for Energy Efficiency in Cloud Computing,” In IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2014 (pp. 1-6).

Appendix A: Implementation Details

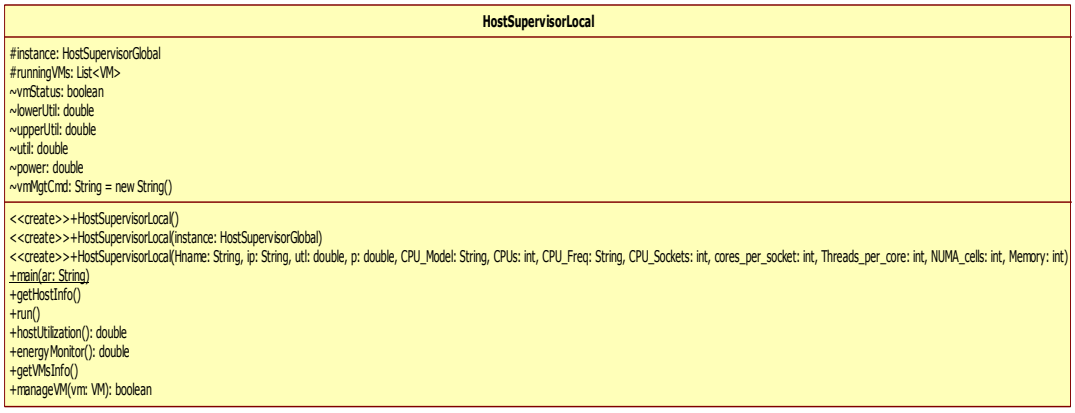


Figure A1: HostSupervisorLocal class of ACA-Cloud

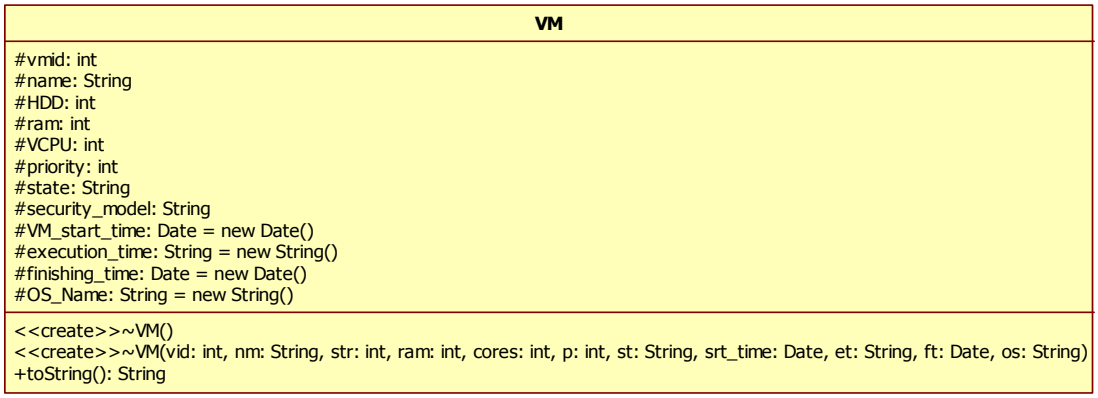


Figure A2: VM class of ACA-Cloud

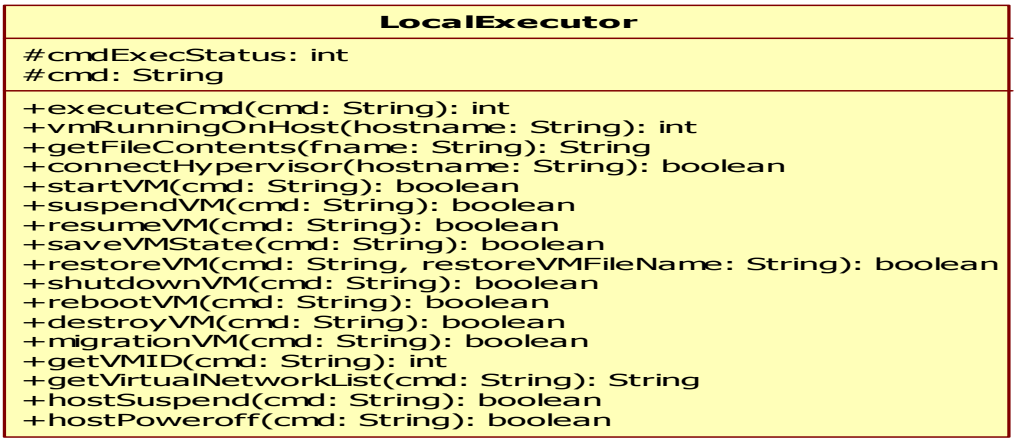


Figure A3: LocalExecutor class of ACA-Cloud

Figure A4: HostSupervisorGlobal class of ACA-Cloud

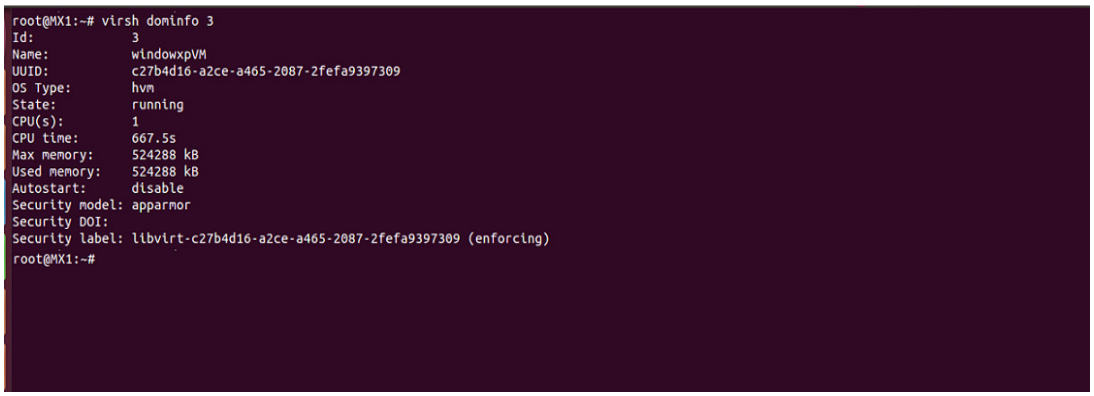


Figure A5: Screen shot of VM information

Figure A6: Screen shot of one of the host

Figure A7: Screen shot of list of VMs on a host