An Energy Efficient Resource Scheduling Approach for Cloud Data Centers

Thesis submitted in partial fulfilment of the requirements for the award of the degree of

Master of Engineering
in
Software Engineering

Submitted By
Amanpreet Kaur
(Roll No: 801631002)

Under the supervision of

(Dr. V. P. Singh)
Associate Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology

(Dr. Sukhpal Singh Gill)
Research Fellow
University of Melbourne

COMPUTER SCIENCE AND ENGINEERING
DEPARTMENT THAPAR INSTITUTE OF ENGINEERING
AND TECHNOLOGY PATIALA, PUNJAB, INDIA

June 2018

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*An Energy Efficient Resource Scheduling Approach for Cloud Data Centers*", in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. V. P. Singh and Dr. Sukhpal Singh Gill* and refers other researcher's work which are duly listed in the reference section.

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

Amanpreet

(Amanpreet Kaur)

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

(Dr. V.P. Singh)
Associate Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology

(Dr. Sukhpal Singh Gill)
Research Fellow
University of Melbourne

Acknowledgement

I would like to thank God for blessing me with all the strength and resources required to complete this task. I would like to express my deepest gratitude to Dr. V.P Singh and Dr. Sukhpal Singh Gill for guiding me through the whole process and providing me with your knowledge and experience.

I am also heartily thankful to Dr. Maninder Singh, Professor and Head, Computer Science and Engineering Department for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection.

Last but not the least, I would like to thank my family for their wonderful support and encouragement without which none of this would have been possible.

Amanpreet

Amanpreet Kaur

Abstract

Cloud Computing provides the mechanism of delivering application as services as well as the resources in data centers that provide those services. Cloud Computing has revolutionized the Information and Communication Technology (ICT) industry. These services are flexible in terms of their usage i.e., pay-as-you-use. The mapping of best resources remains a complex job in cloud environment due to heterogeneity of various resources. Scheduling the best resource–workload efficiently agreeing to user service requests is an energy optimization issue. The foremost objective of the Resource scheduler is to schedule the resources effectively and with maximum resource utilization. Resources dispersion, heterogeneity, uncertainty is a big issue for resource scheduling techniques in Cloud environment.

Current cloud computing framework hosts millions of physical servers that generate lot of heat requiring cooling units in turn to eliminate the effect of heat. Thus, overall energy consumption of the data center increases tremendously servers as well as cooling units. However existing resource scheduling techniques works mostly for virtual cloud environment. In this thesis, an energy efficient approach has been presented which schedules heterogeneous resources on the physical machines and executes cloud workloads on corresponding resources. The proposed approach improves the energy and resource utilization along with reducing the Service Level Agreement (SLA) violation. In CloudSim toolkit, we are executing proposed technique, and experimental results show that the proposed technique has better energy utilization as compared to existing resource scheduling approaches.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Notation	Definition
VM	Virtual Machine
PM	Physical Machine
PE	Processing Element
SLA	Service Level Agreement
QoS	Quality of Service
FoS	Focus of Study
CRAC	Computer Room Air Conditioning
DVFS	Dynamic Voltage Frequency Scaling
MAUD	Minimize Average Utilization Difference
MIPS	Million Instructions per Second
EEVM	Energy Efficient Virtual Machine

Chapter 1
Introduction

This chapter gives an overview of Cloud Computing, Cloud Computing advancements, deployment models architecture and elements of Cloud Computing. It briefly presents the research motivation for Cloud Computing and presents primary contributions of this research. In the last, the structure of the rest of the thesis is provided.

1.1 Cloud Computing Concepts

Cloud computing is a model that permits the provisioning of flexible resources (infrastructure/platform/software) on request. These resources are provisioned as services to the users according to their usage and are charged accordingly. Cloud is developing as an expansive field of research with the goal of investigating the immense measure of data and client solicitations to extract information. Cloud Computing's versatile nature of dynamic resource provisioning, low-latency, economical and on-demand elasticity has developed cloud computing into applicable framework for big data handling [1]. Cloud environment consists of lakhs of physical servers and devices geographically distributed across number of server farms. Cloud Computing allows on-demand service access to a common group of configurable. Cloud computing framework is supported by the idea of virtualization that utilizes virtual resources. These are software implementation of physical machine called Virtual Machines (VMs). Figure 1.1 shows the conceptual view of "cloud computing environment".

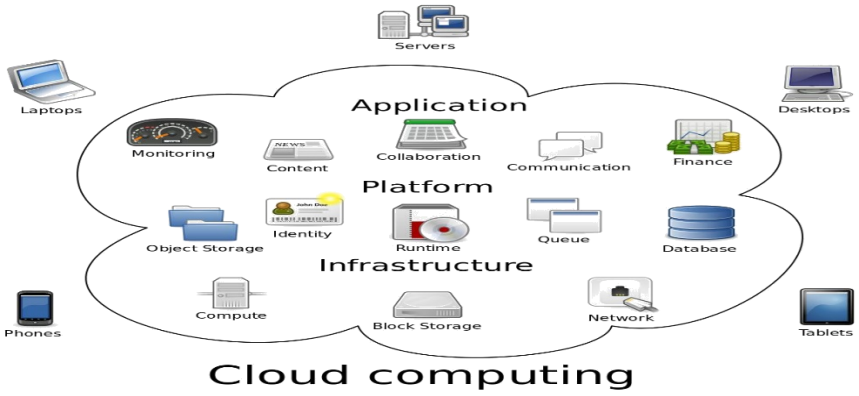


Figure 1.1 Conceptual View of Cloud Computing [2]

1.2 Cloud Computing Evolution

Cloud computing indeed advanced out of Grid Computing. The evolution has been through a number of levels which include Grid and Utility Computing, Software as a Service. Grid computing uses parallel computing to solve large problems. Utility computing was provisioning of metered services. SaaS was network based subscription to services. Emergence of Cloud computing is shown in Figure 1.2. Thus, Cloud computing results from the Grid Computing that aims to delivers storage and compute resources and services over the internet using hardware more proficiently [3].

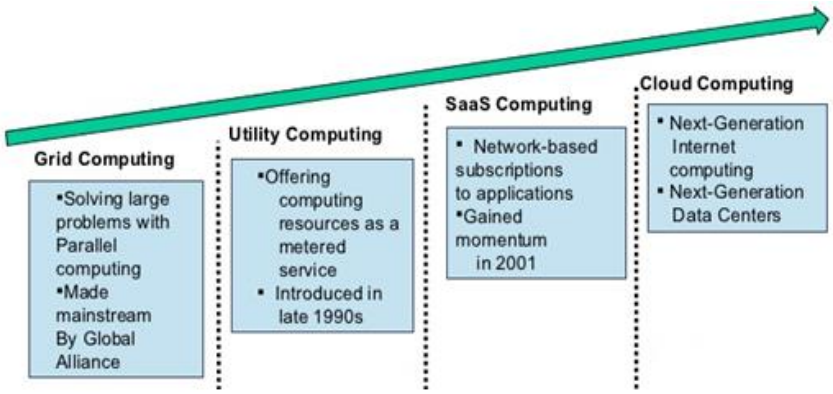


Figure 1.2 Emergence of Cloud Computing [4]

1.3 Cloud Computing Architecture

Cloud Computing is an on request model i.e. "pay as you utilize" this saying signifies the fundamental notion of "cloud computing"; client's simply pay for the part or measure of assets they utilize. Clients can increment or decline the measure of cloud services as per their necessity. Cloud Computing is a plan for mass scale distributed computing that uses different prevailing strategies virtualization, grid computing. In "cloud computing environment" the cloud service providers delivers the cloud consumers several types of services & cloud consumers use required services and are charged accordingly.

NIST (National Institute of Standards and Technology) [5] established the idea of Cloud Computing design by proposing five important features, three cloud services models and four cloud deployment.

1.3.1 Important Features of Cloud Computing

There are following important features of cloud computing [5]:

- *Virtualization Support*: Virtualized assets could be estimated and resized with certain versatility. These qualities make equipment virtualization, the ideal innovation to make a virtual framework. Likewise, by joining all open stockpiling systems in an information center, it permits making virtual circles free from apparatus and area.
- *High Availability and Data Recovery*: A few virtual infrastructure chiefs complete this by outfitting a failover framework, which recognizes failure of both physical and virtual servers and restarts virtual machines on physical servers. This style of high openness secures from host, yet not virtual machine.
- *On-demand Resource Provisioning*: Empowers the client to effectively access services from the cloud without connecting with human
- *Rapid Elasticity*: Services are provisioned rapidly and elastically.
- *Metered Service*: Services are valued timely, permitting clients to discharge and when resources are not required then they are not supposed to pay. Cloud computing permits the client to ask for and utilize just the vital measure.

1.3.2 Cloud Service Models

There are three Cloud Services Models: software, platform or infrastructure as a service. Frequently known as “SPI model” [5].

- Software as Service (SaaS) - An ability given to consumer who can utilize the supplier's requests executing on the cloud. SaaS or application Cloud is one sort of Cloud administrations, where programming usefulness is conveyed as a service. SaaS gives advantages to benefit customers. SaaS should save a nearly more prominent level of its quality than traditional framework. The most renowned vendors and useful examples of SaaS are Gmail or Salesforce [6].

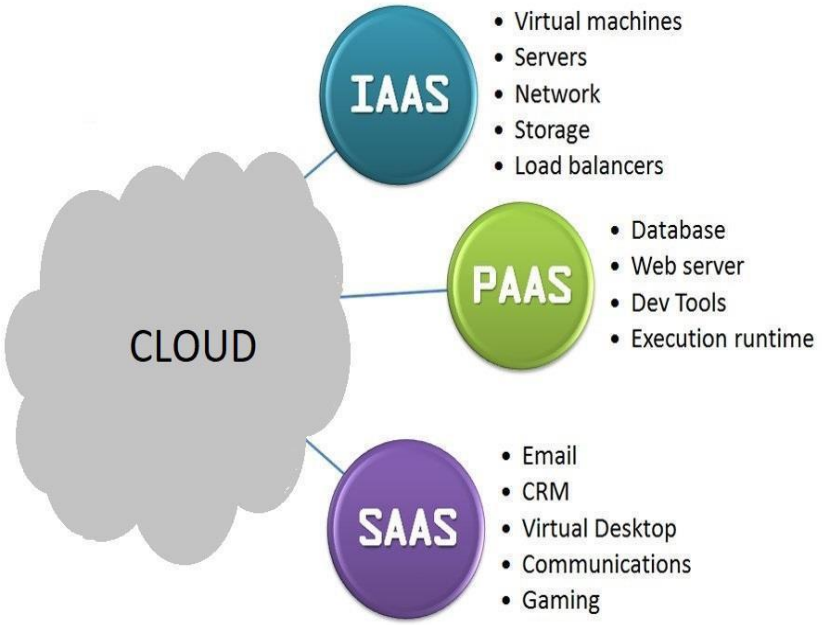


Figure 1.3 Service Models of Cloud

- Cloud Platform as Service (PaaS) – According to this service model, the customer can use the programming languages or tools delivered by service provider, in the "cloud environment" to create applications. PaaS enables the deployment and scalability of user application trivial and charges incremental and practicably foreseeable –a user can deploy applications on PaaS. Google and Microsoft is PaaS provider [6]. User can execute their application with little modification and probably execute already existed applications with efforts.
- Infrastructure as Service (IaaS) - This is an ability given to the client by which, a client can install and execute the software (i.e. operating systems, applications). Consumer can access processing, storage, networks and other necessary computing resources. Through IaaS the resources are distributed to the Cloud consumers such as servers, storage, and associated tools essential over the Internet, permitting enterprises to develop an application environment from scratch based on requirement is very easy and inexpensive. Billing is based on the usage of service and can get complicated with tiered on-demand valuing. Amazon is an IaaS provider.

1.3.3 Cloud Deployment Models

Depending on the cloud consumer requirement and availability Cloud computing can be run in four different models as shown in Figure 1.4.

- Public Cloud - The cloud computing services accessible to the overall population. This is immaculate cloud computing.
- Private Cloud - The cloud deployment model solely available to a particular organization.
- Community Cloud - Several organizations come and share this infrastructure a particular community with common issues.
- Hybrid Cloud – It consists of more than two deployment models i.e. private, community or public.

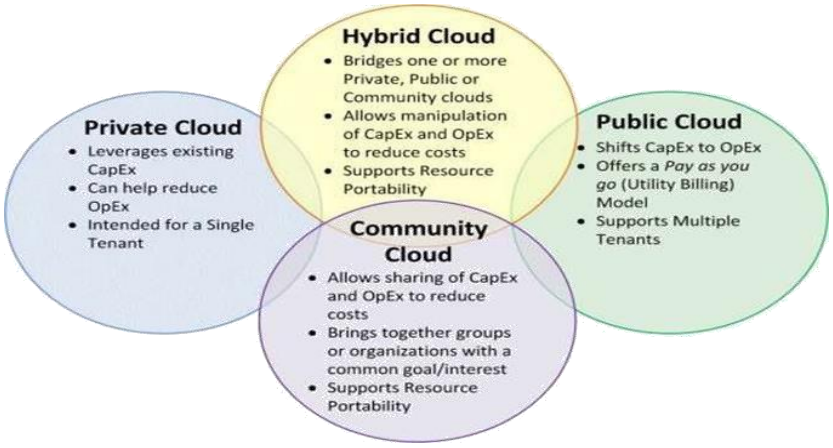


Figure 1.4 Cloud Deployment Models [8]

1.4 Open Challenges

Generally, there are a few complex issues advancing in the distributed computing condition in which note-worthy commitments can be made given that appropriate consideration is paid to them. Many research issues are yet to be resolved in cloud computing which are as follows: There are following open challenges [9-12] in various different aspects of holistic management:

- *Server Consolidation*: There is significant in-crease in the utilization of used servers, de-grading their performance as whole workload is concentrated on these servers. It may de-grade the response time and maximize the transition costs.
- *Dynamic Voltage Frequency Scaling (DVFS)*: The confinement of DVFS, is that a diminishment in frequency likewise lessens the performance of the circuit which consequently, affect the system performance. Thus, DVFS need to be used wisely, to maintain the performance.
- *Thermal Aware*: Monitoring the accurate inlet temperature of the servers, ambient temperature continuously is a tedious job, hence thermal aware scheduling need appropriate mechanisms for determining the temperature.
- *Workload Aware*: Predicting the nature of the workload according the history is quite cumbersome.

1.5 Research Motivation

Reducing the energy usage and corresponding environmental impact resource scheduling must be done efficiently and effectively by implementing efficient and effective policies. Advance work has been done to build energy saving servers and network infrastructure. Through data centers up to 20% of energy saving can be done in addition they economize up to 30% on cooling [2]. Traditional resource allocation policies are not efficient due to heterogeneity and uncertainty of resources. A typical cloud model utilizes an important energy, which rises the Carbon Dioxide (CO2) level ultimately. The energy pitfall is a huge task, and so, cloud suppliers should attempt to limit its energy utilization as could reasonably be expected while fulfilling the consumer's need and assuring Quality of Service (QoS). Thus cloud providers focus on developing energy-efficient approaches and policies [13]. The cloud computing is stated as a fundamentally energy efficient policy owing to the ascendable assessment of its resources and multitenant capability. Inadequate resource scheduling techniques in cloud lead to the wastage of resource utilization and energy [14-16]. The problem of allocating user workloads to a set of virtual machines and allocating virtual machines on different server hubs adhering to the terms of service as cited in Service Level Agreements (SLAs) and sustaining the QoS is stated as the service provisioning issue [17, 18]. In this thesis, considering the thermal characteristics of the host

focusing on the issue of allocating VMs to hosts in the server farms and assigning workload to the appropriate resources considering utilization characteristics. The VMs are clustered according to their utilization, resource usage, and memory and bandwidth usage. The anticipated scheduling policy reduces the energy of the physical machine, resource utilization with the aid of proficient distribution strategies. The entire incoming load of the server farm is involved with several VMs for execution. The aggregate workload of the server farm is the finite number of jobs where each job assigned to a few VMs for execution which in turn are hosted by physical machines. Therefore, a cloud scheduling framework has been presented which take cares of energy consumption and resource utilization by implementing policy at VM level and cloudlet level. An experimental assessment to approve planned arrangement using the CloudSim toolkit as simulation environment.

1.6 Thesis Outline

The rest of chapters in this thesis is structured as:

Chapter 2 – This chapter provides related work done in Cloud Computing to resolve various concerns mostly associated with current work focusing on energy issues.

Chapter 3 – This chapter discusses the Problem Statement and Objectives and Commitments.

Chapter 4 – This chapter describes proposed scheduling algorithm for the energy efficiency difficulty and demonstrates our projected effort to enhance several factors in an uncertain computing environment.

Chapter 5 – This chapter describes the Tools for Setting Cloud Environment, Scheduling approach Execution, and Implementation of Proposed Approach, analyses experimental outcomes and demonstrates the efficiency of algorithm paralleled to prevailing typical algorithms.

Chapter 6 – This chapter summarizes the Conclusions drawn in the thesis along with Thesis Contribution and Future Research Directions.

Chapter 2
Literature Survey

This chapter provides related work done in the area of various concerns in the cloud with main focus on energy efficiency in the cloud.

2.1 Background

In general, energy efficiency strategies can be employed into following major areas: i) Servers ii) Storage iii) Memory iv) Network v) Cooling. Servers are the significant consumers of power, approaches for energy conserving for physical hubs comprise Dynamic Voltage Frequency scaling (DVFS), Server Consolidation, and Virtualization. Another significant energy consumer is networking infrastructure strategy for saving energy is turning off system components or placing them into rest mode, another solution is assigning the virtual network requests to a few physical network devices in case of low traffic. Cooling is other major aspect for achieving energy efficiency one of the approach for cooling is raised floor to change over warm air to cool air by expelling warmth to the outside.

2.1.1. Current Status of Cloud Computing

The evolution of energy efficiency describes the advancement in existing strategies, new techniques are built in cloud computing to minimize the energy. This section explores the evolution of work done in energy over the years based on the parameters Quality of Service (QoS) and Focus of Study (FoS) as shown in Figure 2.1. Many Energy Efficient Algorithms (EEAs) improve the cloud environment along with improving the utilization, responsiveness, performance and other QoS important parameters.

In 2018, Ibrahim H. et al. [19], built up an Integer Linear Programming (ILP) model that lessens the energy usage of data center. It focuses on dynamic workload scheduling technique. Energy efficiency and also the near optimal scheduling decisions are achieved by implementing an adaptive genetic algorithm. The algorithm finds the timetable for the

underlying arrangement of tasks as received. Before introducing the new arrangement of tasks, the calculations experience the rundown of received undertakings and builds the list which depends on asked for and accessible capacities of assets equipped for executing each assignment.

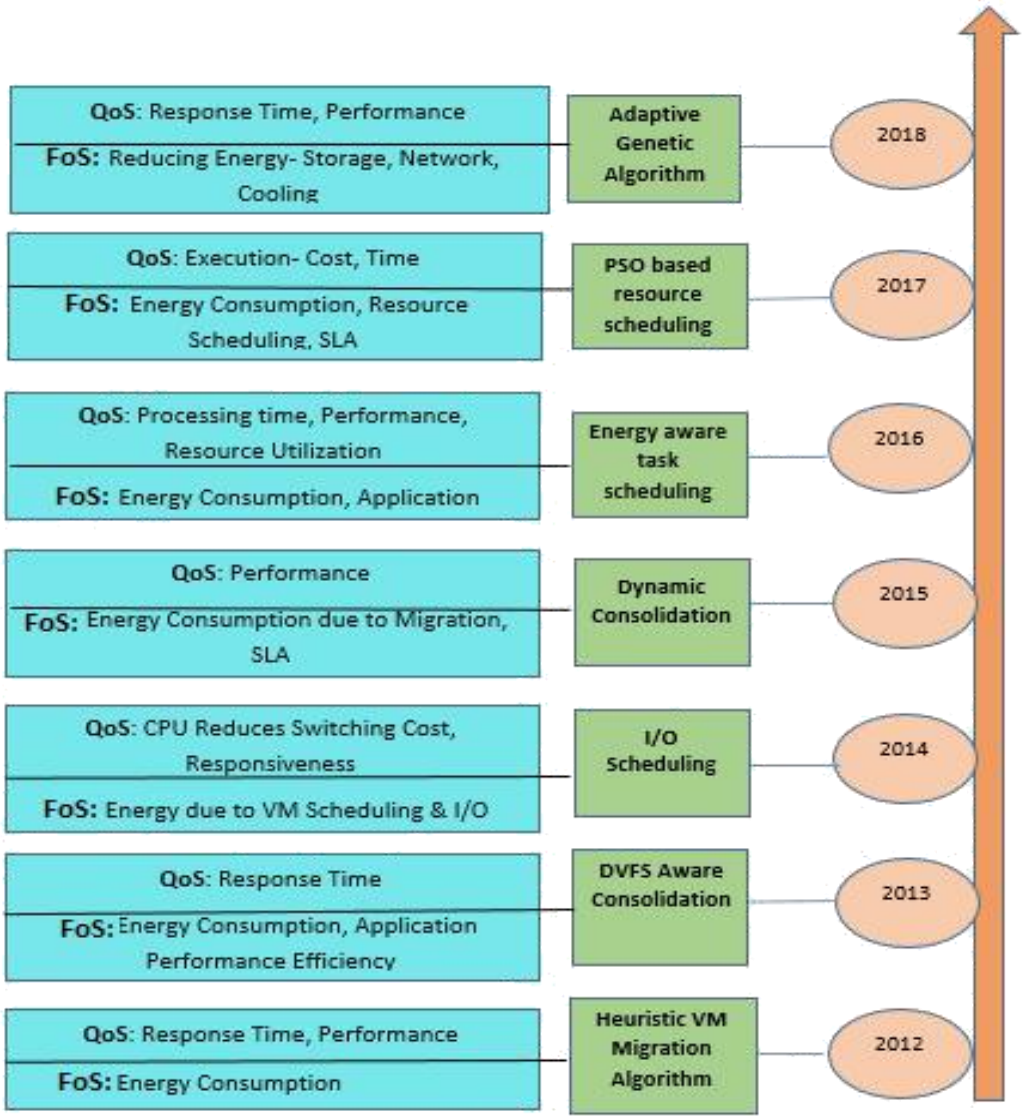


Figure 2.1 Evolution of Energy Efficiency

In 2017, Sukhpal et al. [20] proposed Particle Swarm Optimization (PSO) based resource provisioning and scheduling technique that aims to decrease energy usage and resource wastage along with execution cost, time and SLA as other parameters.

In 2016, Leila Ismail et al. [21] derived an energy aware task scheduling strategy that considers the power consumption of the Cloud for energy-efficient resource usage and increases the application efficiency.

In 2015, Shaw et al. [22] utilized the proactive and reactive hotspot detection practice to decrease virtual machine migration as a result lessens the energy usage in cloud environment. The concept is migration is performed after analysing that migration is required or not in case of hotspot detection. After taking the decision the VM will be shifted to a new host using a novel approach based on predicting the future load on the respective load. It performs when and where will VM will be migrated.

In 2014, Xiao et al. [23] explored the VM scheduling policy in order to compensate the energy damages triggered by I/O virtualization.

In 2013, Gao et al. [24] proposed a scheme of dynamic resource manager that took advantage of server consolidation and dynamic voltage frequency scaling. The energy efficient resource management framework where incoming workloads are submitted to its corresponding application manager through dispatcher module. These work-loads are than allocated in round robin fashion to their virtual machine. In 2012, R. Karthikeyan [25] discovered an energy efficient VM migration algorithm by using heuristic strategy.

2.2 Areas to Explore: Opportunities

Conserving energy in cloud computing particularly in data centers is a critical issue for the researchers. In this work, we aim to explore the various strategies for energy optimization in cloud data center. There are number of ways by which energy consumption by data centers in cloud can be lowered some of the major techniques to save the energy consumption can be classified as shown in Figure 2.2.

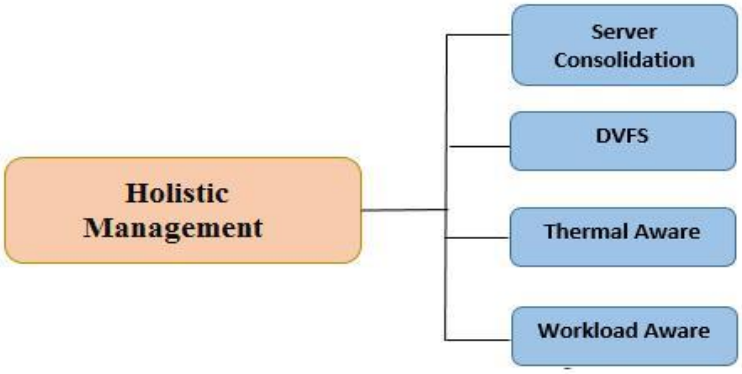


Figure 2.2 The Components of Holistic Management

The techniques can be further classified based on other parameters. Most of the proposed scheduling algorithms aim to decrease the average energy consumption in the cloud center, other scheduling mechanisms target on reducing the high temperature of physical hosts while a few techniques are designed with a goal to reduce peak power consumption.

2.2.1 Server Consolidation

Aggregating the workload on fewer physical machines while turning off the rest of server machines. It is one of the energy efficient approach for achieving energy efficiency via virtualization or migration. To achieve energy efficiency low stacked Physical Machines are virtualized and run on couple of physical machines. Consolidation was done statically before, where low loaded virtual machines were manually migrated to one physical server. Dynamic consolidation permits to adjust the quantity of physical servers as per existing workload. It allows periodic reallocation of virtual machines to under loaded or normal hosts. It implicates the discovery of overloaded and under loaded hosts in the data center, which virtual machine to be migrated when to be migrated and where (physical machine) to be migrated [26].

Generally, there are two ways in which migration can be performed: regular migration and live migration. The main strategy includes moving a virtual machine from one host to other by delaying the initially utilized server, and continuing it on the target server while duplicating its memory substance from original server. The second technique plays out a similar usefulness yet without stopping the server [27].

Dongyan Deng et al. [28] presented an energy efficient-oriented framework based on virtual machine framework. They introduced a VM placement policy called Minimize Average Utilization Difference (MAUD) that take host list and VM migration list as input, algorithm considers the load balancing problem. The algorithm uses the difference between host utilization after accepting VM and the average utilization of data center to optimize the energy.

Chun et al. [29] propose a hybrid server farm plan that utilizes heterogeneous stages to spare power.

2.2.2 Energy Efficiency

Dynamic Voltage Frequency Scaling Scheduling (DVFS) is an energy optimization dynamic technique for managing the power. DVFS is mainly done to lower the power

consumption. DVFS is basically the adjustment of power and frequency settings of the computing devices in order to optimize the resource allotment for tasks and if resources are not required then maximize the power savings. Due to reduction in clock frequency of the processors less voltage is supplied. DVFS technique is used for virtual machines hosted by physical machines along with the algorithm or scheduling mechanism to reduce the energy. DVFS technique manages the power expenditure of processors like multicore, DRAM memory bank and other elements. DVFS system and workload planning can be joined in two ways: (1) workload scheduling, and (2) slack recovery. In the schedule generation, tasks graph is (re)scheduled on DVFS-empowered servers in a function that includes both energy saving and make span to meet both the requirements in the meantime. In slack reclamation, which fills in as post preparing method on the yield of planning calculations, DVFS procedure is utilized to limit the vitality utilization of undertakings in a timetable created by a different scheduler [21-22].

Patricia et al. [30] explored the dynamic voltage frequency adjustment DVFS policy that takes into account the trade-off between the energy expenditure and production and the novel consolidation algorithm which is frequency aware while allocating cloud workload. The algorithm helps in boosting up the consolidation and decreases the count of active hosts.

Zhuo et al. [31] addressed the problem of energy consumption by proposing a DVFS enabled heuristic scheduling algorithm. First calculates the initial order of tasks and obtain the make span and deadline constrains using heft algorithm. From energy utilization obtain and merges the inefficient processors by reclaiming the slack time and redistributes tasks on it. Sharma [32] presented an adaptive algorithm that controls the recurrence and voltage levels to keep the momentary use of servers limited utilizing feedback loop. The algorithm is implemented inside the Linux kernel for DVFS enabled processors, algorithm also adheres to the SLA.

2.2.3 Thermal Aware Scheduling

Allocation of workloads according to the temperature of physical machines in sequence to optimize the energy expenditure as a way for reducing the cooling cost and the average temperature of the server. While scheduling the work-load the operating system decides on which server the workload will be executed according to the temperature history of the physical

machine. Thermal aware scheduling aims to avoid the creation of hotspots, performance degradation and reliability. To determine the temperature over time for server's various techniques have been proposed. Thermal-aware monitoring and profiling is one of the technique for performing thermal aware scheduling. Thermal-aware monitoring includes frameworks to save and survey the heat dissipated from server farms. Thermal profiling is storing the record of the attributes of heat dissipated from physical machines, microchips, and computational task. Thermal Aware Scheduling is done in 3 ways: i) Reactive Approach, ii) Proactive Approach and iii) Mixed Approach.

In *reactive approach*, the scheduling of the workload is done after thermal anomaly has occurred while in proactive scheduling is done before occurrence of any thermal anomaly. For minimizing the warmth distribution the thermal-aware scheduler utilizes the thermal profiles and estimations to assign the task through the server farm [l]. Ying-Jun Chen [33] authors proposed a thermal aware virtual machine migration manager which transfers load from overheated physical machines to normal ones, by determining the temperature and resource utilization of the physical machine. Uses the proactive approach to save power; that employs heat distribution and migration time as a measure for VM selection policy and load balancing as VM allocation.

Moore et al. [34] developed two temperature aware workload placement policies: Zone based discretization and minimize-heat-recirculation. The chief policy utilizes the data regarding hot spots and cold spots of steady state in the data center for. Second policy minimize the total amount of heat that recirculates before returning to the CRAC units and maximizes the potential utilization of each server. Yousri et al. [35] implemented the thermal aware scheduler that maps the virtual machine request to a physical machine with respect to the temperature of the host. Uses the thermal and power model for migrating the virtual machines according to temperature and utilization of the servers.

2.2.4 Workload Aware Scheduling

Present day server farms commonly have an expansive amount of servers and thus, the choice regarding allocating the workload on particular servers influences the heat dissemination and power-utilization. Incoming user requests are scheduled on the basis of the nature of their requests i.e., computational workloads resource requirement differ from

workload that requires storage. In-appropriate arrangement results in incredible expand in the temperature of the data center which will additionally build the warmth dispersal of the physical machines and furthermore increment the cooling necessities. Thus, work-load-scheduling strategies have been proposed which put the workloads on accessible physical machines with the objective of power saving, lessening the temperature and the cooling necessities.

Ehsan Pakbaznia [36] employed a short-term workload estimating method to forecast the incoming task to decide on the number of on servers and placement of workloads while concurrently regulating the inlet cold air temperature. Achieves the power savings by performing dynamic resource provisioning. R.K.Jena [37] paper centers to optimize energy and time using workload planning utilizing clonal section algorithm. The clonal algorithm is and adaptive based on clonal section theory as the new request of resources arrives, Clonal Section Algorithm is executed by the system to adjust the placement of resources. The algorithm optimally schedules user tasks to data centers randomly and each user task is assigned to the processing element of each allocated data center**.**

2.3 Holistic Management Aspects: A Comparison

Based on the above discussed literature, Table 2.1 presents the comparisons of different scheduling techniques using different criteria such as year, algorithm, environment, scope, technology. The taxonomy shown above aims to emphasize the need of the efficient technique which would help in achieving the proficiency of data centers on energy grounds. In the future, the above-mentioned techniques will be applied in a synergistic way to provide much energy savings in a holistic way. The challenge is to develop an energy efficient VM scheduling technique for efficient execution of cloud workload.

Table 2.1 Comparison of Different Approaches

Year	Scheduling Technique	Algorithm	Environment	Scope	Technology
2018	Workload aware scheduling [19]	Adaptive Genetic Algorithm	Dynamic	Server, Storage, Network, Cooling	Single cloud data center
2017	Energy aware scheduling [21]	Scheduler	Homogenous	Server	CloudSim
2016	Server Consolidation [28]	Underload decision algorithm	Dynamic	Server	CloudSim
2016	Workload aware scheduling [11]	Genetic Algorithm	Dynamic	Server, Storage	Single cloud data center
2015	DVFS aware scheduling [30]	Dynamic consolidation algorithm	Dynamic	Severs	CloudSim
2014	Energy aware scheduling [23]	Scheduler-RESCUE	Heterogeneous	Server	Private Cloud
2013	DVFS + Server Consolidation [15]	Dynamic Resource Management	Heterogeneous	Server, Network	Own testbed
2012	Thermal aware scheduling [14]	Task Scheduling algorithm	Heterogeneous	Server, Cooling	Real Data Center Environment

2.4 Discussion

In this research, we explored the issues in cloud computing environment more specifically pertaining to energy related. Analysed various algorithms employed using energy-efficient strategies in cloud data centers. Mostly research proposals are concentrated on energy conserving methodologies for servers. For the sustainability of cloud computing reducing the power consumption has become an important issue due to rise in power cost and rise in carbon emission. Researchers have applied various mechanisms to achieve the energy efficiency while maintaining the SLA violations. This research effort presents major energy efficient approaches in the cloud.

Chapter 3
Problem Statement

Based on literature survey, the problem of scheduling policy of Cloud workloads and Virtual machines has been identified. This chapter presents the problem statement and objectives and commitments of this research work.

3.1 Problem Analysis

Current cloud computing framework supports/hosts millions of physical servers that emit lot of heat requiring cooling units in turn to eliminate the effect of heat. Thus overall energy consumption of the data center increases tremendously servers as well as cooling units. However, existing scheduling techniques works mostly on Virtual Machine (VM) allocation policies. Requiring energy efficient strategies for scheduling heterogeneous resources to the physical machines and cloud workload to corresponding resources. Energy efficient technique has been presented that for energy proficiency in cloud data center, VM placement is critical. When workload of a physical hub is continuously increased, then, at some point of time, the temperature of this machine may exceed its maximum working temperature, that is, threshold temperature. If a machine runs beyond this maximum temperature, then the danger of hardware or software break-down increases and machine needs substantial amount of cooling. Excessive heating can even completely destroy the essential hardware circuitry of the Physical Machine. So, it is necessary to proactively deal with the rising temperature of the servers and there must be resourceful and proficient policy of allocation and migration of VMs which considers the thermal characteristics of the Physical Machines before making scheduling decisions, so that maximum threshold temperature can be avoided.

3.2 Objectives and Commitments

The key aim is to analyses the cloud workloads, virtual machines based on their utilization and physical machines based on their thermal behaviour. This helps in efficient allocation of VM(s) to hosts and mapping of cloud workloads to virtual machines. The following are primary objectives of this work:

1. To study existing energy efficient VM scheduling techniques.
2. To propose a novel energy efficient VM scheduling technique for optimization of energy consumption during workload execution.
3. To validate the proposed approach in cloud environment using CloudSim toolkit.

Chapter 4

Proposed Energy Efficient VM Scheduling Approach

This chapter presents a scheduling approach for a Cloud environment. Framework comprises of Workload scheduling and VM allocation. Proposed approach is based on two resource scheduling policies Thermal Aware based Scheduling policy and Utilization based Scheduling policy that minimizes the energy consumption.

4.1 Preliminaries

This section describes the basic resource required for the scheduling approach.

4.1.1 Virtual Machines

A VM is a software application of a machine (i.e., a computer) that performs programs like a Physical Machine. A VM is hosted by a PM not bounded to a hardware machine, i.e., a software environment.

Each VM can have an alternate OS, and a Virtual Machine Monitor (VMM) is utilized to control and deal with the VMs on a solitary physical hub. A VMM is frequently alluded to as a hypervisor.

4.1.2 Cloud Workloads

Cloud workload is generalization of work of that instance or set of instances going to perform. For Example: Running a web services [38].

Figure 4.1 illustrates the basic scheduling design for the “cloud environment”. Area for work is to propose scheduling strategy for Virtual Machines and Cloud Workloads based on thermal model and utilization model respectively.

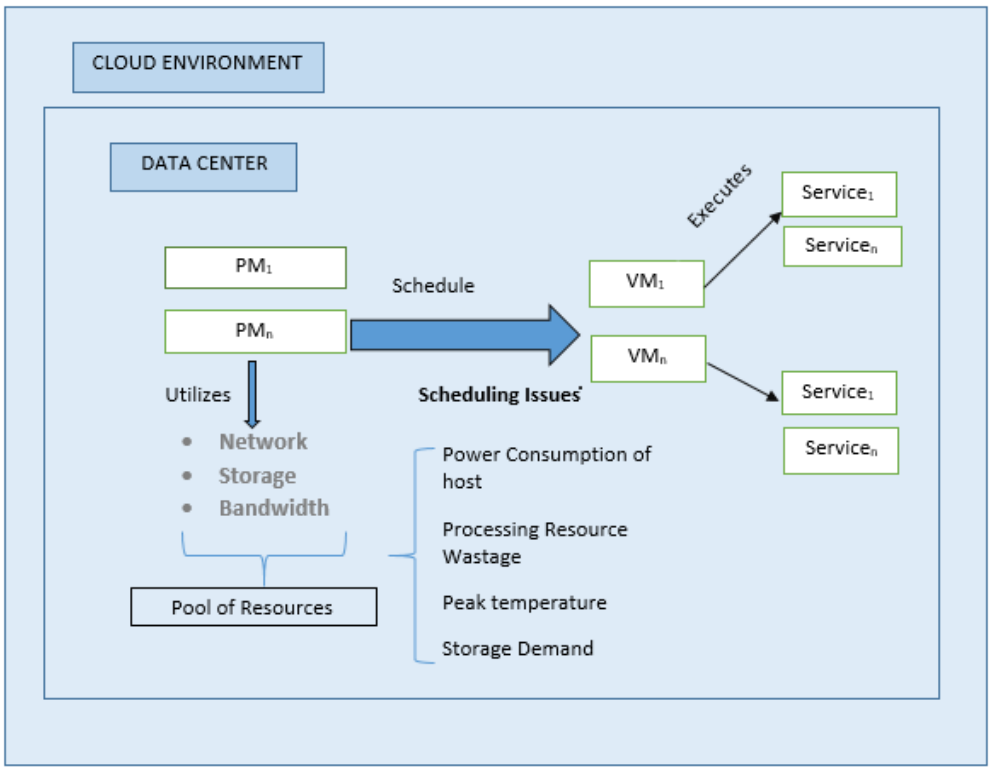


Figure 4.1 Cloud Environment Scheduling Architecture

4.2 Existing Approach

Several techniques have been proposed for the scheduling of virtual machine in the host based on utilization history of the host.If the host is being underutilized it is better to turn it off to save power consumed by it. But one should avoid throttling of host because it can cause SLA violation and often lead to single point failure and also degrade performance. Various algorithms are proposed in the literature such as Inter Quartile Range Maximum Correlation [39] [44] [45]. This algorithm selects hosts whose utilization history was between down and up threshold. For VM selection: The Maximum Correlation Policy (MC). According to this probability of overloading of a PM is directly proportional to the correlation among the VM's utilization or applications of that PM. So, according to this policy that VM is selected for migration which has the maximum correlation of CPU utilization with the sum of other VM s on that host/ PM

Existing algorithms do not consider the thermal characteristics of host. This problem of VM allocation policy is taken into account in this thesis. Dynamic VM migration consists of following steps:

a) Deciding when to migrate a VM?
b) Choosing which VM to migrate?
c) Choosing a destination where to migrate the selected VM?

4.3 Proposed System Design

Proposed design for scheduling is based on: 1) Utilization Model 2) Thermal Model. In utilization model, cloud workloads are assigned to VMs based on their resource utilization, while thermal model considers the thermal characteristics of host machine and accordingly VMs are scheduled on PMs.

4.3.1 Utilization Model

The jobs devised originated by cloud consumers over the Internet is stated to as cloud workload or service demand. Consumers demand facilities from the "Cloud Service Supplier (CSP)". These demanded services known as "cloud workloads" are submitted to the workload queue of the cloud system.

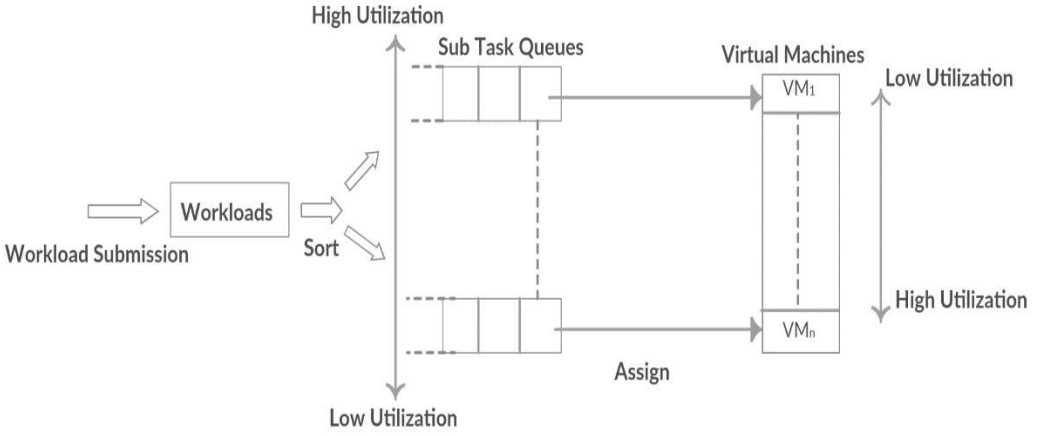


Figure 4.2 Workload Classification and Assignment

The task manager sorts the cloud workload in ascending order based on resource utilization and placed in the queue to assign workload to virtual machines. These Virtual machines are also sorted in terms of their utilization, memory consumption and bandwidth usage in descending order i.e. opposite order in which workloads are sorted and placed in the queue. The cloud workloads present in the sub task queues are submitted to the data center broker. Workload classification and assignment are shown in Figure 4.2. Thus cloud workloads are mapped to the virtual machines based on proposed policy.

4.3.2 Thermal Model

The idea of algorithm is to design a scheduling policy for virtual machines based on the CPUs temperature characteristics. Thus a thermal model is needed that describes the changes of this parameter when applications (here virtual machines) are running.

Thermal aware scheduling considers current temperature and maximum working temperature, that is, threshold temperature of every machine, before making scheduling decisions.

Let maximum threshold temperature of a server machine be *Tover* and let current temperature of a server machine be *Tcu. Tover* is the temperature beyond which a machine is overheated. *Tcu* is the temperature of host on which the machine is currently running. The heuristic chosen for VM scheduling is the difference between threshold and present temperature, as formulated in equation 1:

$$\Delta Tvi = Tover - Tcu \quad (1)$$

The current temperature *Tcu* is calculated based on the bandwidth of host, mips of host and memory utilization of host.

4.4. Proactive Thermal Management

Virtual machines are separated into various classes in view of the temperature attributes and they at that point are dispensed to have as indicated by the temperature of the host. The virtual machine movement component is directed to guarantee the unwavering quality of the framework when a host achieves threshold temperature.

The past thermal administration methodologies may back off or close down the physical machines when the temperature achieves a basic esteem, yet did not decrease the

recurrence of warm hotspots. It anticipates the effect of hotspots before they happen, rather than responding after warm hotspots and anomalies happen on the framework. In this theory, a proactive hotspot strategy is introduced in view of the temperature attributes of virtual machine and temperature of host at during execution to limit the hotspots.

4.4.1 Virtual Machine Classification and Scheduling

VM's are categorized into three classes in accordance with the thermal features: hot, warm, and cold. A 'cold' VM symbolizes that a VM may decrease the temperature of a host if its temperature is greater than θ_{vl}. A 'hot' VM states that a VM may arise the temperature of a host if its temperature greater than θ_{vh}. A 'warm' VM signifies the VM with lower temperature deviation.

Thermal scheduler allocates VMs to the Physical Machine, whose temperature is farthest away from its maximum threshold temperature. Scheduler also manages a waiting queue. It is the queue through which demand for new Virtual Machine is fulfilled. Each new request for Virtual Machine is added at the end of waiting queue. The scheduler will remove that VM request from waiting queue; then further queues the VMs into sub-queues based on the temperature variation of host due to VM i.e., value of ΔTvi. If temperature variation of host is greater than the high temperature threshold, then queues in hot queue; if temperature variation of host is less than the low temperature threshold, then queues to cold queue else to warm queue.

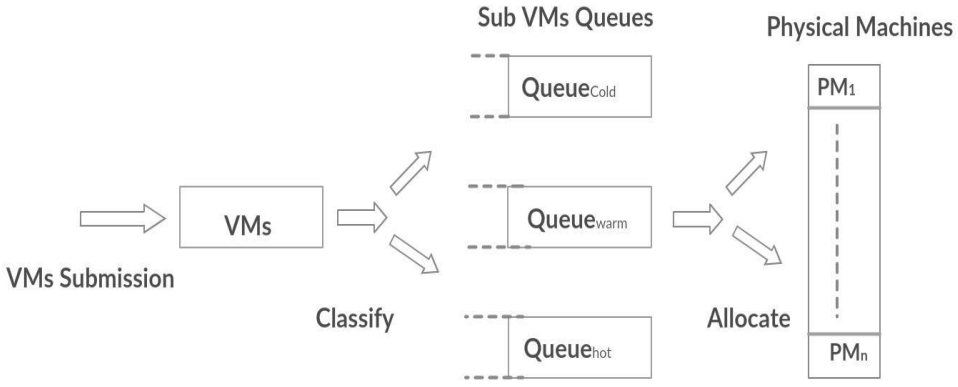


Figure 4.3 Process of Virtual Machine Classification and Allocation

For VM allocation, comparison is done if the current temperature of a host is more than high temperature threshold of host, a cold VM from cold queues will be allocated. Subsequently host temperature lowers to the ordinary state, a VM will be dispensed to execute. When the temperature of a host is greater than θch and VM from cold queue is allocated to the host, the temperature then becomes unmanageable. When the temperature of a host is less than θcl, a hot VM will be selected to execute. For cases that the temperature of a host is between θch and θcl, the VM from warm queue will be executed.

4.5 Proposed Algorithm

This section discusses scheduling mechanism described above by our proposed algorithm.

4.5.1 Proposed Thermal Aware Algorithm

Thermal Aware Algorithm [Algorithm 1] is implemented contains basically two phases: 1) Sorting and 2) resource scheduling. Thermal Aware Algorithm sorts the Virtual Machines based on their execution effect on physical machines. Thermal Aware Algorithm executes hot VM on cold nodes and cold VM on hot nodes and tries to reduce energy consumption.

The below mentioned algorithm is described step by step.

Algorithm 1 [Thermal Aware Algorithm.]

i. Initialize the variable host as available at time stamp to 0 i.e. host is available at beginning.
ii. Calculate the temperature variation, before and after allocating the VM until queue is not null.
iii. VMs in sub-queues according to the temperature variation value of host due to VM_i.
iv. Dequeues VMs, from sub-queues according to the host state and allocates VM to PM.
v. Execute the allocated VM on PM.

Symbols used: The symbols used in the [Algorithm 1] mentioned below are defined as following:

ΔTvi: Temperature variation of host due to VMi.

Tover : Temperature of overheated host Tnormal :Temperature of normal host

Tdanger : Temperature of overheating host

θvh : The low temperature threshold of ΔTvi

θvl : The high temperature threshold of ΔTvi

θch : Low temperature threshold of host // normal host temperature

θcl : High temperature threshold of host //overheating host temperature

For specifying different type of Virtual Machines according to temperature variation of the host machine;

θvh : Tover – Tdanger

θvl : ½{Tnormal - Tdanger}

1. Tover = 79°C. // Overheated host temperature
2. Tnormal = 44°C. // normal host temperature
3. Tdanger = 70°C. //overheating host temperature

Algorithm 1: Thermal Aware Energy Efficient Algorithm

1. **Input:** Number of VMs and number of available PMs
2. **Output:** Scheduling of VMs o the PMs
3. **Start**
4. Initialize all host list (Number of PMs)
5. Initialize all VMs list (Number of Resources)
6. do
7. {
8. Dequeue VM from queue
9. Switch ΔTvi
10. if (ΔTvi > θ vh)
11. do Enqueue the VM to QH
12. else if (ΔTvi < θ vl)
13. do Enqueue the VM to QC
14. else (θ vh > ΔTvi > θvl)
15. do Enqueue the VM to QW
16. }
17. while vmlist!= NULL
18. if (Tover > θ ch and QC!= NULL) then dequeue(QC);
19. else if (QW!=NULL) then dequeue(QW);
20. else
21. dequeue(QH);
22. if (Tover < θ ch & QH!=NULL) dequeue(QH);
23. else if (QW!=NULL) then dequeue(QW);
24. else dequeue(QC);
25. Allocate VM to PM.

4.5.2 Proposed Utilization Based Algorithm

Implemented utilization based algorithm which i) sorts the cloud workloads and ii) map to VMs. Based on the utilization of cloud workloads, workloads are organized in increasing order of utilization, and VMs are in decreasing order, then schedule the tasks accordingly, as tasks which lowers the utilization i.e. light tasks are executed on VM with high utilization and vice-versa.

The below mentioned algorithm is described step by step.

Algorithm 2 [Utilization Aware Algorithm.]

i. Initialize the variable host as available at time stamp to 0 i.e. host is available at beginning.

ii. Sort workloads and VMs based on utilization.

iii. Map workload to VMs.

Symbols used: The symbols used in the [Algorithm 2] mentioned below are defined as following:

T_i $\Delta Util$: Variation in utilization while executing task$_i$**.**

VM_j $Util$: Utilization of VM j.

ALGORITHM 2: Utilization based Algorithm
1. **Input:** Number of workloads and number of available resources 2. **Output:** Mapping of each workload to the resource 3. **Start** 4. Initialize all resource list (Number of Resources) 5. Initialize all workload list (Number of Jobs) 6. For each task i=1 to I do 7. sort task with increased Ui(T_i) 8. For each VM j=1 to J do 9. sort VM with decreased VMj Util (Vj) 10. Schedule the task on VM

Chapter 5

Implementation and Experimental Results

This chapter focuses on the implementation of the proposed framework followed by framework execution. The results of the proposed scheduling policies is validated in CloudSim 3.0 toolkit using NetBeans as a platform. The implementation of the proposed scheduling policy is described in this chapter. To evaluate the performance, proposed scheduling policies compares with existing policies.

5.1 Tools for Setting Cloud Environment

A brief introduction of tools which are used in designing and implementation of proposed framework is given below.

5.1.1 CloudSim Toolkit

CloudSim is an extensible simulation toolkit that facilitates modeling and simulation of Cloud computing structures and request provisioning environments. The CloudSim toolkit gives both structure and performance modeling of Cloud system modules such as server farms, virtual machines (VMs) and resource provisioning strategies. CloudSim supports modeling and simulation of cloud computing data centers, virtualized hosts, allocation strategies, network topologies and message passing applications. Moreover, the framework can be integrated with Eclipse and NetBeans, allowing you to code in Java. It implements nonexclusive application provisioning methods that can be reached out effortlessly and restricted exertion. Presently, it bears system and behavioural modelling of Cloud environment components consisting of Virtual Machines, and asset delivering strategies. Also simulates data center environment situations comprising of both single and inter-networked Clouds (organization of Clouds). Additionally, it uncovered custom interfaces for actualizing approaches and allocating strategies for designation of VMs under between organized “Cloud computing” situations.

CloudSim offers the accompanying novel highlights [40]:

• Assist modelling and simulation of vast scale “Cloud computing environment”, including server farms, on a solitary physical computing hub.
• An independent stage for displaying Clouds, benefit intermediaries, provisioning, and portion strategies.
• Aid simulation of system associations amid the simulated framework components.
• Capability for simulation of combined Cloud condition that internetworks assets from both private and open spaces, an element basic for look into contemplates identified with Cloud-Bursts and programmed application scaling.
• Accessibility of a virtualization device that guides in the formation and administration of numerous, autonomous, and co-facilitated virtualized services on a server farm hub.
• Switching capability between space-shared and time-shared designation of physical hubs to virtualized services.

These convincing highlights of CloudSim would accelerate the improvement of new request provisioning calculations for Cloud Computing. Figure 5.1 demonstrates the manifold plan of the CloudSim programming system and its building parts. The CloudSim simulation level offers help for demonstrating and simulation of virtualized Cloud-based server farm conditions including devoted administration interfaces for VMs, memory, and data transfer capacity. The key issues, for example, provisioning of hosts to VMs, overseeing application execution, and checking dynamic framework state, are taken care of by this layer. A Cloud supplier, who needs to think about the proficiency of various approaches in distributing its hosts to VMs (VM provisioning), would need to execute his techniques at this layer. Such usage should be possible by automatically broadening the center VM provisioning usefulness. Cloud application designer can play out the accompanying exercises: (i) create a blend of workload ask for dispersions, application arrangements; (ii) demonstrate Cloud accessibility situations and perform hearty tests in light of the custom setups (iii) actualize custom application provisioning systems for Clouds.

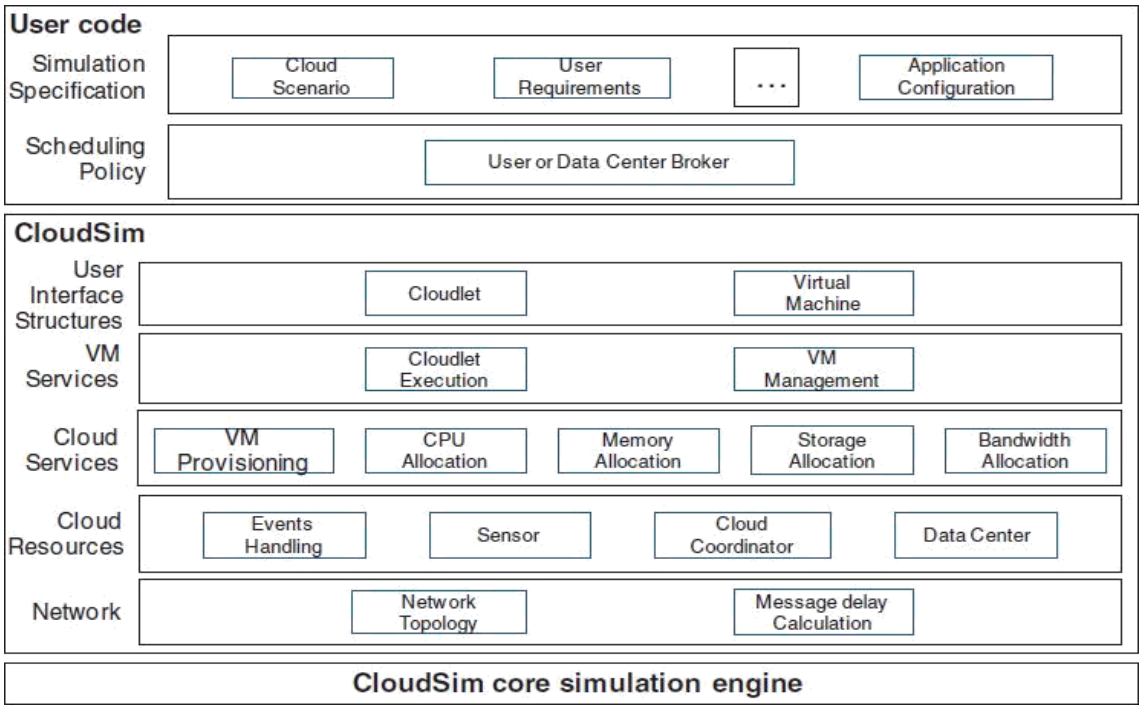


Figure 5.1 CloudSim Architecture [41]

CloudSim objective is to give a summed up and extensible simulation structure that empowers demonstrating, simulation, and experimentation of rising Cloud Computing frameworks and application administrations. Enabling its clients to center around particular framework arrangement problems that they need to discover, without getting worried about the low level facts of interest identified with Cloud-based foundations and administrations.

5.1.2 NetBeans IDE 8.2

NetBeans IDE is a free and open source incorporated advancement condition for application improvement on Windows, Mac, Linux, and Solaris working frameworks. The IDE rearranges the advancement of web, venture, work area, and versatile applications that utilization the Java and HTML5 stages. The IDE additionally offers bolster for the advancement of PHP and C/C++ applications [42].

5.2 Implementation of Proposed Approach

The framework executes as follows:

1. Initialize cloudsim package (before creating any entity)

2. Create data center(s)

{

Processing element (pe) list →host list → define data center characteristics → data center instance

}

3. Create data center broker (dcb) // user is represented by the dcb.
4. Create virtual machine // define procedure for task scheduling algorithm
5. Submit virtual machine to data center broker
6. Create cloudlet(s) / tasks / workload
7. Submit cloudlets to data center broker
8. Map: cloudlet → virtual machine

Figure 5.2 shows the proposed Utilization based approach for workload mapping.

```
for(Cloudlet cloudlet : getCloudletList())
{
tempList.add(cloudlet);
}
int totalcloudlets = tempList.size();
for ( int i=0; i< totalcloudlets; i++)
{
    Cloudlet smallestcloudlet = tempList.get(0);
    for(Cloudlet checkcloudlet : tempList)
    {
        if(smallestcloudlet.getUtilizationOfCpu(i)> checkcloudlet.getUtilizationOfCpu(i))
        {
          smallestcloudlet =  checkcloudlet;
        }

    }
    sortList.add(smallestcloudlet);
    tempList.remove(smallestcloudlet);

}
```

Figure 5.2 shows the proposed Utilization based approach for workload mapping.

9. Allocate: virtual machine→ host/physical machine.

Figure 5.3 shows the proposed Thermal based approach for VM selection policy and VM scheduling approach is shown in Figure 5.4.

```
double difference=10;
PowerHost tempHost =null ;
for (PowerHost host : this.<PowerHost> getHostList()) {
        if (excludedHosts.contains(host)) {
                continue;
        }
        if (host.isSuitableForVm(vm)) {
                currentTemperature = hotTemperatureOfServer();
                tempWithoutVM = calculateNewTempHS(host,vm);
                skewWithoutVM= calculateNewSkewHS(host,vm);
                tempDif = Math.abs(currentTemperature - tempWithoutVM);
                skewDif = Math.abs(currentSkewness - skewWithoutVM);
                if (currentTemperature ==0 && currentSkewness ==0){
                        tempHost=host;
                }
                if ((tempDif + skewDif) < difference){
                        difference = tempDif + skewDif;
                        tempHost = host;
                }
        }
}
allocatedHost = tempHost;
return allocatedHost;
```

Figure 5.3 VM Selection based on host temperature

```
double difference=0;
Vm vmToMigrate = null;
Vm tempVM =null ;
for (Vm vm : migratableVms)
{
        if (vm.isInMigration())
        {
                continue;
        }
        tempWithoutVM = calculateNewTemp(host,vm);
        skewWithoutVM= calculateNewSkew(host,vm);
        tempDif = currentTemperature - tempWithoutVM;
        skewDif = currentSkewness - skewWithoutVM;
        if (currentTemperature ==0 && currentSkewness ==0){
                tempVM=vm;
        }
        if ((tempDif + skewDif)> difference){
                difference = tempDif + skewDif;
                tempVM = vm;
        }
        }
vmToMigrate = tempVM;
```

Figure 5.4 VM Scheduling

10. Start simulation (Automated process handled through descreted event simulation engine)
11. Print results when simulation is over. Figure 5.5 shows the simulation output of proposed approach.

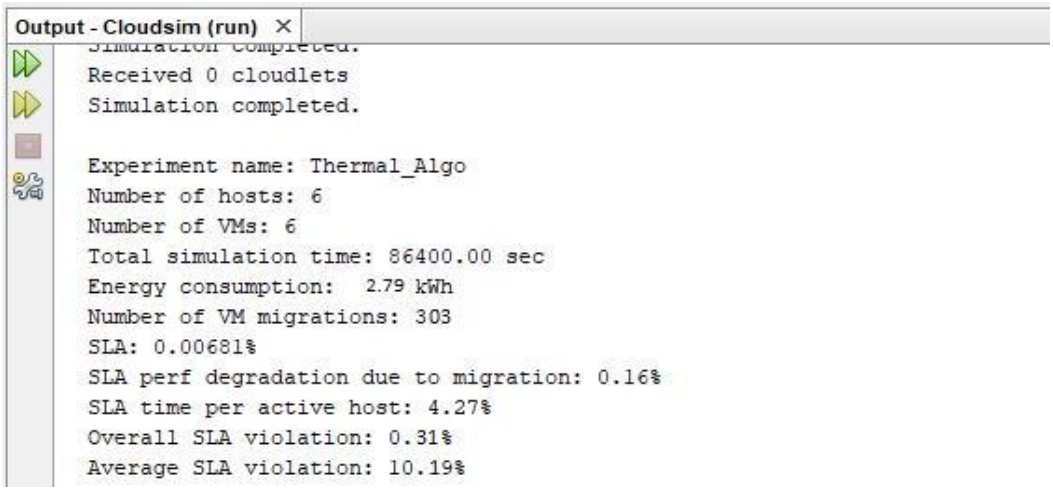


Figure 5.5 Simulation Output of Proposed Approach

5.3 Simulation Setup

The experiment is carried out in simulating environment CloudSim Toolkit [40]. The proposed algorithm Thermal aware based and utilization based is implemented in Java on NetBeans framework. Virtual Machines and Physical Machines. The proposed algorithm is executed ten times in simulation to avoid anomalies. Proposed algorithm is compared with existing algorithm IQRMC (Inter Quartile Range Maximum Correlation) [39] [44] [45]. The IQRMC scheduling technique considers only 2/3 of the host utilization and also increases the computation overhead. Table 5.1 shows the configuration and resource characteristics parameters defined in the simulating environment.

Table 5.1 Scheduling Parameters

Parameter	Value
Number of Resources	20-100
Number of Cloudlets	50
Size of Cloud Workload	1000-2000
Bandwidth	1000-3000 B/S
Number of PEs per Machine	1
File Size	300 MB
Cloud Workload Output Size	300 MB

Table 5.2 gives the assumed values for defining thermal constants of physical machine in the data center.

Table 5.2 Experimental Constants setup

	Thermal Parameter	Value	Unit
Thermal Constants	Overheated Temperature(Tover)	79	Celsius
	Normal Temperature (Tnormal)	44	Celsius
	Overheating Temperature (Tdanger)	70	Celsius

5.4 Experimental Results

Efficiency of proposed scheduling algorithms “Thermal Aware based” scheduling and “Utilization based” scheduling is compared against other algorithm with same functionality. Variation of three optimal QoS parameters (Energy Consumption, No. of VM migrations, SLA violation) is measured using proposed technique. The total energy consumption is given by the first performance metric of the data center by the proposed scheduling algorithm due to the execution of different number of resources. The second performance metric gives the

resource utilization due to execution of number of resources. And the third performance metrics gives percentage of SLA violations.

Test Case 1: Energy Consumption: Calculated the value of Energy Consumption in kWh with diverse number of resources. Figure 5.6 shows the linear relationship with number of resources. The proposed algorithm EEVM has 6.92% less energy consumption than the IQR.

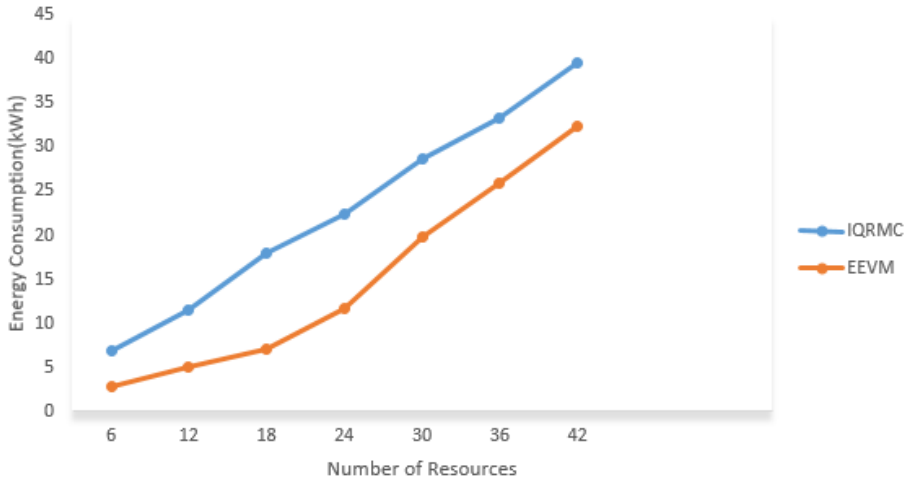


Figure 5.6 Effect of Number of Resources on Energy Consumption

Test Case 2: Number of VM Migrations: The main source of unnecessary virtual machine migration is sloppy allocation of virtual machine to the host. Our proposed algorithm improves it and reduces the number of virtual machine migration and thus reduces the cost associated with the migration process. Figure 5.7 shows the number of average VM migrations are reduced which saves 8.4% energy consumption.

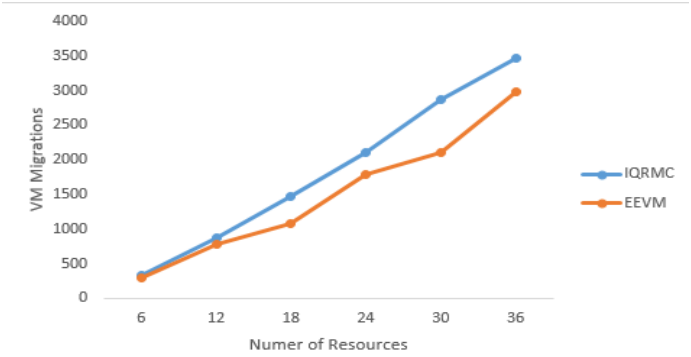


Figure 5.7 Number of VM Migrations vs. Number of Resources

Test Case 3: SLA violation Rate: When the virtual machine does not get the required amount of Millions Instructions per Second (MIPS), SLA violations occurs. It is defined as the product of Failure rate and weight of SLA [42]. We have used following formula to calculate SLA violation rate (Eq. 2). List of SLA = < m_1,m_2……..m_n >, where n is total number of SLAs [43].

$$Failure\ (m) = \begin{cases} \text{m is not violated, Failure (m) = 1} \\ \text{m is violated, Failure (m) = 0} \end{cases}$$

$$Failure\ Rate = \sum_{i=1}^{n} Failure\ (mi)/n$$

$$SLA\ Violation\ Rate = Failure\ Rate \times \sum_{i=1}^{n}(wi) \qquad (2)$$

Where wi is weight for every SLA. We have analysed the effect of change in number of resources on SLA violation rate. SLA violation rate is changed with different number of resources as shown in Fig. 5.8. Value of SLA violation rate is varied between 0 and 100%. SLA violation rate in EEVM is 9.5% lesser than IQR.

Figure 5.8 Overall SLA Violation vs. Number of Resources

Testing results shows that the proposed algorithm performs better for different number of resources taken in testing. The proposed algorithm EEVM consumes less energy as compare to IQR. The results also show that the number of VM migrations and overall SLA violations in case of proposed algorithm is much less than the other IQR algorithm. The simulation results are shown in table 5.3 below.

Table 5.3 Simulation Results

Number of Resources	**Energy Consumption (kWh)**		**Number of VM Migrations**		**Overall SLA Violations %**	
	IQR	**EEVM**	**IQR**	**EEVM**	**IQR**	**EEVM**
6	6.9	2.79	336	303	0.48	0.31
12	11.47	4.96	872	785	0.39	0.29
18	17.96	7.03	1476	1086	0.72	0.65
24	22.35	11.65	2109	1784	0.88	0.77
30	28.64	19.86	2882	2099	0.80	0.79
36	32.26	25.89	3465	2985	0.92	0.81

Chapter 6

Conclusions and Future Scope

This chapter summarizes the research work proposed in this thesis. It also discusses open research problems and outlines a number of future research directions.

6.1 Conclusions

Cloud Computing and its architecture has been discussed in this thesis. This thesis emphasize on the resource scheduling issues in cloud computing environment more specifically to energy efficiency. Various energy efficiency algorithms proposed till now have been analysed. This research effort presents energy efficient scheduling approach for cloud workloads and VMs while maintain the SLA violations. The experimental evaluation is done through CloudSim 3.0 simulator for validating the effectiveness of results. To conclude the gathered results shows that the proposed approach has better performance in terms of energy consumption, number of VM migrations and SLA violations.

6.2 Future Scope

The further future directions can be:

1. *High Energy Demand in Cooling Servers*: High power consumption lead to creation of hot spots and increase in server temperature. Thus requiring cooling methodology for cooling the data centres.
s
2. *Peak Temperature among Servers*: Temperature is another important parameter for both physical servers and virtualization solutions. Variance in the on chip temperature and the resultant occurrence of hot spots degrades the performance of processors, increases the energy consumption. Thermal management strategies are required to uniformly distribute the temperature.

3. *High Level of Power Consumption by the Servers*: In efficient or non-energy aware scheduling techniques lead to increase in power consumption among the servers which degrades the server's reliability, performance.

References

[1] D. Puthal, B.P.S. Sahoo, S. Mishra and S. Swain, "Cloud computing features, issues, and challenges: a big picture." In *Computational Intelligence and Networks (CINE), International Conference on*, pp. 116-123. IEEE, 2015.

[2] "Overview of Cloud Computing", [Online]. Available: https://en.wikipedia.org/wiki/Cloudcomputing. [Accessed 15 4 2018].

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared" *In Grid Computing Environments Workshop Ieee*, pp. 1-10, 2008.

[4] "Cloud Computing" [Online]. Available: https://www.slideshare.net/PallaviRai2/cloud-computing-11983526. [Accessed 17 4 2018].

[5] P. Mell and T. Grance, "The NIST definition of cloud computing", 2011.

[6] B. P. Rimal and E. Choi, "A taxonomy and survey of cloud computing systems," *in Fifth International Joint Conference on INC, IMS and IDC*, Seoul, Korea, 2009.

[7] "Different Types of Service Models in Cloud Computing", [Online]. Available: https://www.bluepiit.com/blog/different-types-of-cloud-computing-service-models/ [Accessed 27 4 2018]

[8] J.D. Jesus, "Cloud Deployment Models", IBM, [Online]. Available: http://www.ibm.com/developerworks/websphere/techjournal/1206_dejesus/1206_ dejesus.html [Accessed 3 5 2018].

[9] S.S. Gill and R. Buyya, "A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View" *preprint arXiv: 1712.02899,2017*.

[10] D. Kliazovich, P. Bouvry and S.U. Khan, "Green Cloud: a packet-level simulator of energy-aware cloud computing data centers", *The Journal of Supercomputing*, *62*(3), pp.1263-1283, 2012.

[11] S. Singh, and, I. Chana, "EARTH: Energy-aware autonomic resource scheduling in cloud computing." *Journal of Intelligent & Fuzzy Systems*, *30*(3), pp.1581-1600, 2016.

[12] S. Singh, and, I. Chana, M. Singh and R. Buyya, "SOCCER: self-optimization of energy-efficient cloud resources. Cluster Computing", 19(4), pp.1787-1800, 2016.

[13] S. Zeadally, S.U. Khan and, N. Chilamkurti, “Energy-efficient networking: past, present, and future”, *The Journal of Supercomputing*, *62*(3), pp.1093-1118, 2012.

[14] L. Wang, S.U. Khan and J. Dayal, “Thermal aware workload placement with task-temperature profiles in a data center" *The Journal of Supercomputing*, *61*(3), pp.780-803, 2012.

[15] L. Wang, and S.U. Khan “Review of performance metrics for green data centers: a taxonomy study” *The journal of supercomputing*, *63*(3), pp.639-656, 2013.

[16] P. Sarwesh, N.S.V. Shet and K. Chandrasekaran, “Effective Integration of Reliable Routing Mechanism and Energy Efficient Node Placement Technique for Low Power IoT Networks”, International *Journal of Grid and High Performance Computing (IJGHPC)*, *9*(4), pp.16-35, 2017.

[17] A. Hameed, A. Khoshkbarforoushha, R. Ranjan, P.P. Jayaraman, J. Kolodziej, P. Balaji, S. Zeadally, Q.M. Malluhi, N. Tziritas, A. Vishnu, and S.U. Khan, “A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems”, pp.751-774, 2016.

[18] J. Tao, J Kolodziej, R. Ranjan, Prakash Jayaraman, P. and R. Buyya, “A note on new trends in data-aware scheduling and resource provisioning in modern HPC systems”, *Future Generation Computer Systems*, *51*(C), pp.45-46, 2015.

[19] H. Ibrahim, Aburukba, R.O. and El-Fakih, K., “An Integer Linear Programming model and Adaptive Genetic Algorithm approach to minimize energy consumption of Cloud computing data centers”, *Computers & Electrical Engineering*, *67*, pp.551-565, 2018.

[20] S.S. Gill, R. Buyya, I. Chana, M Singh, and A. Abraham, “BULLET: particle swarm optimization based scheduling technique for provisioned cloud resources”, *Journal of Network and Systems Management*, *26*(2), pp.361-400, 2018.

[21] L. Ismail, and A. Fardoun “Eats: Energy-aware tasks scheduling in cloud computing systems”, *Procedia Computer Science*, *83*, pp.870-877, 2016.

[22] S.B. Shaw, and A.K. Singh, “Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center”, *Computers & Electrical Engineering*, *47*, pp.241-254, 2015.

[23] Xiao, P., Hu, Z., Liu, D., Zhang, X. and Qu, X., "Energy-efficiency enhanced virtual machine scheduling policy for mixed workloads in cloud environments" *Computers & Electrical Engineering*, *40*(5), pp.1650-1665, 2014.

[24] 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*, *40*(5), pp.1621-1633, 2014.

[25] R. Karthikeyan and P. Chitra, "Novel heuristics energy efficiency approach for cloud Data Center", In *Advanced Communication Control and Computing Technologies (ICACCCT), IEEE International Conference* pp. 202-207, 2012.

[26] C.H. Hsu, S.C. Chen, Lee, C.C., Chang, H.Y., Lai, K.C., Li, K.C. and Rong, C., "Energy-aware task consolidation technique for cloud computing", In *Cloud Computing Technology and Science, IEEE Third International Conference* pp. 115-121, 2011.

[27] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges", *Journal of grid computing*, *14*(2), pp.217-264, 2016.

[28] D. Deng, K. He and Y. Chen, "Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers" In *Cloud Computing and Intelligence Systems (CCIS), 2016 4th International Conference on* pp. 366-370, 2016.

[29] B.G. Chun, G. Iannaccone, R. Katz, G. Lee and L. Niccolini, "An energy case for hybrid datacenters", *ACM SIGOPS Operating Systems Review*, pp.76-80, 2010.

[30] P. Arroba, J.M. Moya, J.L. Ayala and R. Buyya, "DVFS-aware consolidation for energy-efficient clouds", In *Parallel Architecture and Compilation (PACT), 2015 International Conference on* pp. 494-495, 2015.

[31] Z. Zhou, J. Abawajy, M. Chowdhury, Z. Hu, K. Li,, H. Cheng, A.A. Alelaiwi and F. Li, "Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms", *Future Generation Computer Systems*, 2017.

[32] V. Sharma, A. Thomas, T. Abdelzaher, K Skadron and Z. Lu, 2003, "Power-aware QoS management in web servers", In *Real-Time Systems Symposium, RTSS 2003. 24th IEEE* pp. 63-72, 2013.

[33] Y.J. Chen, G.J. Horng, J.H. Li and S.T. Cheng, "Using Thermal-Aware VM Migration Mechanism for High-Availability Cloud Computing", *Wireless Personal Communications*, *97*(1), pp.1475-1502, 2017.

[34] J.D. Moore, J.S. Chase, P. Ranganathan and R.K. Sharma, "Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers" In *USENIX annual technical conference, General Track* pp. 61-75, 2015.

[35] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej and A. Streit, "Load and thermal-aware VM scheduling on the cloud", In *International Conference on Algorithms and Architectures for Parallel Processing* pp. 101-114, 2013.

[36] E. Pakbaznia, M. Ghasemazar and M. Pedram, "Temperature-aware dynamic resource provisioning in a power-optimized datacentre", In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010* pp. 124-129, 2010.

[37] R.K. Jena, "Energy Efficient Task Scheduling in Cloud Environment", *Energy Procedia*, pp.222-227, 2017.

[38] W. Cirne and F. Berman, 2001, "A comprehensive model of the supercomputer workload", In *Workload Characterization, IEEE International Workshop on* pp. 140-148, 2001.

[39] A. Beloglazov and 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*, pp.1397-1420, 2012.

[40] Calheiros, R.N. Ranjan, De Rose, C.A. and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services", *arXiv preprint arXiv: 0903.2525*, 2009.

[41] "NetBeans IDE 7.1.2", [Online]. Available: https://netbeans.org/community/news/ /shows/1556.html [Accessed 25 5 2018].

[42] S.S. Gill, I. Channa, M. Singh and R. Buyya, "STAR: SLA-aware autonomic management of cloud resources", *IEEE Transactions on Cloud Computing*, 2017.

[43] S.S. Gill, I. Channa, M. Singh and R. Buyya, "CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing", *Cluster Computing*, pp.1-39, 2017.

[44] P. Shukla and R. K. Pateriya. “IQR based Approach for Energy Efficient Dynamic VM Consolidation for Green Cloud Data Centers.” *International Journal of Computer Applications* 123 (9) 2015.

[45] C. Kaur. “An Energy Efficient Virtual Machine Allocation Policy in Cloud Environment.” Master’s Thesis, 2014.

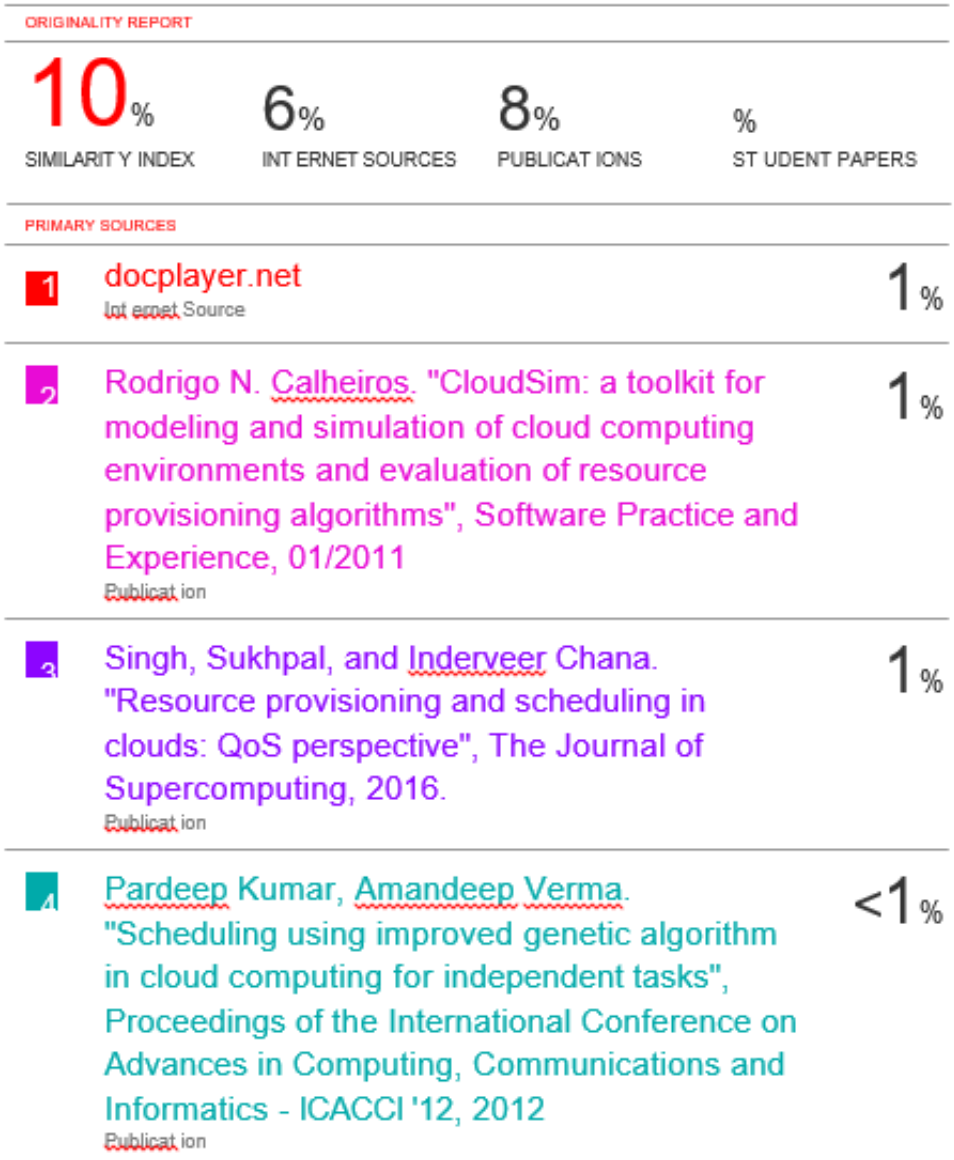
List of Publications

Communicated

[1] Amanpreet Kaur, V.P. Singh and S.S. Gill, "The Future of Cloud Computing: Opportunities, Challenges and Tends", in proceeding of IEEE International Conference on I-SMAC (IOT in Social, Mobile, Analytics and Cloud) 2018, organized by SCAD Institute of Technology, Chennai.

[2] A. Kaur, V.P. Singh and S.S. Gill, "Thermal Aware Resource Scheduling for Cloud Computing". [To be Communicated]

Plagiarism Report



ORIGINALITY REPORT

10%	6%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.net Internet Source	1%
2	Rodrigo N. Calheiros. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software Practice and Experience, 01/2011 Publication	1%
3	Singh, Sukhpal, and Inderveer Chana. "Resource provisioning and scheduling in clouds: QoS perspective", The Journal of Supercomputing, 2016. Publication	1%
4	Pardeep Kumar, Amandeep Verma. "Scheduling using improved genetic algorithm in cloud computing for independent tasks", Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12, 2012 Publication	<1%

5	Singh, Sukhpal, and Inderveer Chana. "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges", Journal of Grid Computing, 2016. Publication	<1%
6	www.ijeat.org Internet Source	<1%
7	www.ijesrt.com Internet Source	<1%
8	Lecture Notes in Computer Science, 2013. Publication	<1%
9	Carlos R. Senna, Luiz F. Bittencourt, Edmundo R. M. Madeira. "Performance evaluation of virtual machines in a service-oriented Grid testbed", 2010 International Conference on High Performance Computing & Simulation, 2010 Publication	<1%
10	repository.um.edu.my Internet Source	<1%
11	Khaled M. Attia, Mostafa A. El-Hosseini, Hesham A. Ali. "Dynamic power management techniques in multi-core architectures: A survey study", Ain Shams Engineering Journal, 2017 Publication	<1%

12	zenodo.org Internet Source	<1%
13	sameekhan.org Internet Source	<1%
14	ar.scribd.com Internet Source	<1%
15	espace.curtin.edu.au Internet Source	<1%
16	Abdulaziz Alarifi, Amr Tolba, Zafer Al-Makhadmeh, Wael Said. "A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks", The Journal of Supercomputing, 2018 Publication	<1%
17	Kansal, Nidhi Jain, and Inderveer Chana. "Artificial bee colony based energy-aware resource utilization technique for cloud computing : ABC BASED ENERGY-AWARE RESOURCE UTILIZATION TECHNIQUE FOR CC", Concurrency and Computation Practice and Experience, 2014. Publication	<1%
18	Aruzhan Kulseitova, Ang Tan Fong. "A survey of energy-efficient techniques in cloud data centers", International Conference on ICT for	<1%

	Smart Society, 2013 Publication	
19	Singh, Sukhpal, and Inderveer Chana. "QRSF: QoS-aware resource scheduling framework in cloud computing", The Journal of Supercomputing, 2015. Publication	<1%
20	www.politesi.polimi.it Internet Source	<1%
21	www.thinkmind.org Internet Source	<1%
22	"Progress in Advanced Computing and Intelligent Engineering", Springer Nature, 2018 Publication	<1%
23	www.ijcaonline.org Internet Source	<1%
24	eprints.soton.ac.uk Internet Source	<1%
25	cloudbus.org Internet Source	<1%
26	Berral, Josep Ll., Iñigo Goiri, Ramon Nou, Ferran Julià, Josep O. Fitó, Jordi Guitart, Ricard Gavaldá, and Jordi Torres. "Toward Energy-Aware Scheduling Using Machine Learning", Energy-Efficient Distributed	<1%