NOVEL ENERGY EFFICIENT VIRTUAL MACHINE CONSOLIDATION FOR ENHANCING QUALITY OF SERVICE IN CLOUD COMPUTING ENVIRONMENT

A Thesis Submitted to the

BHARATHIDASAN UNIVERSITY

TIRUCHIRAPPALLI-620 024

In Partial fulfillment of the degree in

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

Submitted by

Mrs. P. NITHYA

(Ref.No. 07873/Ph.D.K10/Computer Science/April 2017/PT)

Under the Guidance of

Dr. L. JAYASIMMAN

Assistant Professor & Research Supervisor

PG AND RESEARCH DEPARTMENT OF COMPUTER SCIENCE

BISHOP HEBER COLLEGE (AUTONOMOUS)

(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI - 620 017

MAY 2022

Dr. L. JAYASIMMAN
Assistant Professor and Research Supervisor
PG and Research Department of Computer Science
Bishop Heber College (Autonomous)
Tiruchirappalli - 620 017, India

CERTIFICATE

This is to certify that the thesis entitled "**NOVEL ENERGY EFFICIENT VIRTUAL MACHINE CONSOLIDATION FOR ENHANCING QUALITY OF SERVICE IN CLOUD COMPUTING ENVIRONMENT**" submitted by **P. NITHYA (Ref.No. 07873/Ph.D.K10/Computer Science/April 2017/PT)** in PG and Research Department of Computer Science, Bishop Heber College (Autonomous), Tiruchirappalli-620 017, for the award of **Doctor of Philosophy in Computer Science** is a record of original work carried out by her under my supervision and guidance. The results embodied in the thesis have not been submitted to any other university or institute for the award of any degree or diploma.

Station: Tiruchirappalli **(Dr. L. JAYASIMMAN)**

Date: **Research Supervisor and Guide**

Curiginal

Document Information

Analyzed document	Ref .No. 07873_P Nithya.pdf (D137326342)
Submitted	2022-05-20T13:23:00.0000000
Submitted by	Srinivasa ragavan S
Submitter email	bdulib@gmail.com
Similarity	1%
Analysis address	bdulib.bdu@analysis.urkund.com

Sources included in the report

W	URL: https://eprints.kingston.ac.uk/id/eprint/43009/1/Liyanage-S.pdf Fetched: 2021-07-14T10:52:35.9170000	4
W	URL: https://www.researchgate.net/publication/341062587_PAPSO_A_Power-Aware_VM_Placement_Technique_Based_on_Particle_Swarm_Optimization Fetched: 2021-05-17T14:19:34.3700000	6
W	URL: https://www.hindawi.com/journals/mpe/2020/7848232/ Fetched: 2020-12-08T17:52:10.2800000	1
W	URL: https://dyuthi.cusat.ac.in/jspui/bitstream/purl/5569/1/Dyuthi%20T-2611.pdf Fetched: 2021-08-07T10:57:25.0900000	1
W	URL: https://www.researchsquare.com/article/rs-1645107/latest.pdf Fetched: 2022-05-20T13:31:50.9000000	1
W	URL: https://research.tees.ac.uk/files/8481231/VM_FGCS_manuscript.pdf Fetched: 2022-01-15T07:33:25.3600000	2
W	URL: https://www.researchgate.net/publication/323646073_Energy-aware_Dynamic_Virtual_Machine_Consolidation_for_Cloud_Datacenters Fetched: 2020-06-12T03:44:17.4170000	1
W	URL: https://www.mdpi.com/1999-5903/10/6/52/pdf-vor Fetched: 2020-04-28T10:00:33.6870000	2
W	URL: https://core.ac.uk/download/pdf/293867727.pdf Fetched: 2020-07-18T06:19:38.3470000	1

P. NITHYA
Research Scholar
PG and Research Department of Computer Science
Bishop Heber College (Autonomous)
Tiruchirappalli - 620 017, India

DECLARATION

I hereby declare that the thesis titled **"NOVEL ENERGY EFFICIENT VIRTUAL MACHINE CONSOLIDATION FOR ENHANCING QUALITY OF SERVICE IN CLOUD COMPUTING ENVIRONMENT"** is a research work done by me under the supervision and guidance of **Dr. L. JAYASIMMAN**, **Assistant Professor, PG and Research Department of Computer Science**, **Bishop Heber College (Autonomous)**, Tiruchirappalli-620 017, India. The thesis of any part thereof has not formed the basis for the award of any Degree, Diploma, Fellowship, or any other similar titles.

Station: Tiruchirappalli **P. NITHYA**

Date: **Research Scholar**

ACKNOWLEDGEMENT

Completion of Doctoral Dissertation had been made possible with the support of several people. I do not have enough words to describe the blessings showered on me by God Almighty in my toughest times while pursuing my research.

I would like to take this opportunity to express my immense gratitude to all those persons who have given their invaluable support and assistance.

It is indeed a great privilege to express my utmost gratitude to my Research Supervisor **Dr. L. JAYASIMMAN*, Assistant Professor, Department of Computer Applications, Bishop Heber College (Autonomous)***, Tiruchirappalli for his sincere and selfless support, prompt and useful advice throughout this research work. I feel happy to place my gratitude to his worthy counsel suggestions, guidance and moral support given to me at every stage of this work. Undoubtedly without his valuable guidance and suggestions, this work would not have seen the light of the day. I am thankful to almighty for providing such a great personal as my Research Guide.

I extend my sincere thanks to the members of doctoral Committee **Dr. R. SUGUMAR**, ***Professor & Research Director, Christu Raj College, Panjappur, Tiruchirappalli*** and **Dr. J. JOHN RAYBIN JOSE**, ***Associate Professor & Head, Dept. of Information Technology, Bishop Heber College (Autonomous)***, Tiruchirappalli for their guidance, timely suggestions, comments, incredible support, keen interest, counsel, boundless help and guidance throughout the research period.

I wish to take this opportunity to express my deep sense of gratitude to the **Rt. Rev. Dr. D. CHANDRASEKARAN*, Bishop and Chairman*** for providing me all the facilities and assistance to undertake my study. I am extremely thankful to **Dr. D. PAUL DHAYABARAN*, Principal, Bishop Heber College (Autonomous)***, Tiruchirappalli and all the members of the college management for providing me all the facilities to done by research work.

I would like to thank our Research Head **Dr. JGR. SATHYASEELAN**, ***Associate Professor and Head of Computer Science, Bishop Heber College (Autonomous)***, Tiruchirappalli for his valuable support and guidance during my course of work.

I wish to take this opportunity to express my sincere thanks to head of the department **Dr. R. THAMARAISELVI**, ***Associate Professor and Head of Computer Applications, Bishop Heber College (Autonomous)***, Tiruchirappalli for her continuous encouragement.

I extend my sincere thanks to **Mr. K. SRIDHARAN, Dr. M. PITCHAIMANI** ***Principal, Srimad Andavan Arts and Science College, Tiruchirappalli,*** and **Mr. KARTHICK DAKSHINAMOORTHY** for their valuable support and guidance during my course of work.

I am very grateful to his Holiness **Srimad Andavan Swamigal**, Founder, Srimad Andavan Arts and Science College (Autonomous), I extend my heartful thanks to the Secretary, Director, Principal, Vice Principal, Controller of Examinations, Deans and Staff Members of Srimad Andavan College (Autonomous) and Bishop Heber College (Autonomous), Tiruchirappalli.

I also thank all my friends, Fellow Research Scholars, Teaching and Non-Teaching Staff Members for their assistance and valuable support.

Last, but definitely not the least, I am greatly indebted to my family. It was my husband's and parents' unconditional love, care and tolerance which made hardship of writing thesis worthwhile. Finally, words are not sufficient to express gratitude my cherished family members and my children for supporting me. Without their support, I do not think that I could overcome the difficulties during these years.

I extend my thanks to one and all who have contributed even in a small way in the completion of my Research work.

P. NITHYA

CONTENTS

ABSTRACT

Now a days, cloud computing becomes very famous and popular among the users because of its various features because the computing resources are taking on rent when needed. So, cloud computing is based on on-demand method or pay-as-you-go model. The environment of cloud computing provides an illusion of infinite computing resources to cloud users. These computing resources are increased or decreased according to the demands of user on cloud. So, in order to achieve efficient utilization proper task scheduling is a prerequisite. Cloud computing develops as another computing worldview which means to give solid, modified and QoS (Quality of Service) ensured computing dynamic situations for end-clients.

Cloud computing brings down cost by avoiding the capital consumption by the company in leasing the physical infrastructure from an outsider supplier. The remote accessibility enables us to access the cloud services from anywhere at any time. To gain the maximum level of the above-referenced advantages, the services offered as far as resources ought to be allocated optimally to the applications running in the cloud. Virtualization allows one user to create several virtual machines (VMs) on a single physical hardware.

Virtualization technologies allow improving resource utilization, technically termed VM Consolidation (or Server Consolidation or Workload Consolidation) is an essential part of resource management of virtualized data centers. Through this research work, a novel optimization-based task scheduling algorithm, optimized based Host detection technique for VM consolidation and an Adaptive Virtual Machine Selection

and Migration Approach for VM consolidation in Cloud Computing has to be proposed for enhancing the Quality of Service (QoS) over cloud environment.

In the first contribution, an efficient task scheduling method is proposed using optimization algorithms. Cultural Algorithm is an evolutionary computation algorithm, which is used to schedule the task in the cloud environment, the belief space in this algorithm is used to store the best optimal solution, which further improve the efficiency of the task scheduling. Particle Swarm Optimization (PSO) is used in this research work is to translate the CA solutions to the PSO population. The best particle value is encoded with CA's chromosomes in the crossover operation of CA. This proposed work reduces the average execution time, response time and makespan, whereas the resource utilization is increased.

The use of cloud computing data centers is growing rapidly to meet the tremendous increase in demand for high-performance computing (HPC), storage and networking resources for business and scientific applications. Virtual machine (VM) consolidation involves the live migration of VMs to run on fewer physical servers, and thus allowing more servers to be switched off or run on low-power mode, as to improve the energy consumption efficiency, operating cost and CO2 emission. A crucial step in VM consolidation is host overload detection, which attempts to predict whether or not a physical server will be oversubscribed with VMs.

In the second contribution, An Optimization technique called Artificial Bee Colony Optimization algorithm and Artificial Neural Network is utilized to find the type of host in the cloud environment. An optimization-based Host Detection methodology is proposed to assess the host based on CPU, Bandwidth, and RAM utilizations in order to find overloaded and underloaded hosts. The Artificial Bee Colony Optimization approach is used to update the weights of the Neural Network for Host classification, and it uses power, CPU, and memory thresholds to prevent underloading and overloading. The performance of the proposed Host detection approach is evaluated with the similar host detection techniques with various evaluation parameters. The proposed host detection approach reduces the energy consumption and SLA violation with varying number of tasks, VMs and Hosts.

Dynamic consolidation of Virtual Machines (VMs), using live migration of the VMs and switching idle servers to sleep mode or shutdown, optimizes the energy consumption. The normal host detected from the contribution – 2 is considered for allocating the VM, VM migration for VM consolidation framework. In this contribution, an adaptive VMs migration selecting method and heuristic algorithm for dynamic consolidation of VMs to be proposed based on the analysis of the historical data. This proposed Adaptive VM consolidation strategy includes the coding, fitness function, selection, crossover, mutation and greedy selection.

.

LIST OF ABBREVIATIONS

Term	Abbreviation
A2SC	Agent based Automated Service Composition
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ACS	Ant Colony System
ANN	Artificial Neural Network
API	Application Programming Interface
BBO	Biogeography Based Optimization
BCAVMP	Balance-based Cultural Algorithm for Virtual Machine Placement
BFD	Best Fit Decreasing
CA	Cultural Algorithm
CC	Cloud Computing
CLS	Cloud List Scheduling
CPU	Central Processing Unit
CSP	Cloud Service Providers
CS-TS	Cultural Swarm based Task Scheduling
DCBB	Divide and Conquer Branch and Bound
DCMMT	Dynamic Consolidation with Minimization of Migration Thrashing
DE	Differential Evolution
DHCI	Dynamic Hadoop Cluster on IaaS
DHSJF	Dynamic Heterogenous Shortest Job First

DPQ	Dynamic Priority-Queue
DRFA	Dominant Resource First Allocation
DVMC	GC-based Dynamic Virtual Machine Consolidation
EALB	Energy Aware Load Balancing
EC2	Elastic Compute Cloud
EES	Energy-Efficient Strategy
EFDTS	Eenrgy Aware Fault Tolerant Dynamic Task Scheduling
EP	Evolutionary Programming
EQ-VMC	Energy Efficient- Quality Aware Virtual Machine Consolidation
ETSA	Energy Efficient Task Scheduling Algorithm
FCFS	First Come First Serve
GA	Genetic Algorithm
GC	Gossip Contracts
GEP	Genetic Expression Programming
GWO	Grey Wolves Optimization
IaaS	Infrastructure as a Service
ICT	Information and Communication Technology
ILP	Integer Linear Programming
IoT	Internet of Things
IT	Information Technology
LB	Load balancing
LPTF	Largest Processing Time First
LR	Local Regression

MAD	Median Absolute Deviation
MCC	Mobile Cloud Computing
MCC	Minimum Completion Cloud
MCDM	Multi Criteria Decision Making
MCI	Machine Condition Index
MIPS	Millions of Instruction per Second
ML	Machine Learning
MLPQ	Minimum-Level Priority Queue
MOVMrB	Multi-Objective VMreBalance solution
MPSO	Multi Particle Swarm Optimization
MV-FRS	Moving Average-based Fuzzy Resource Scheduling
NIST	National Institute of Standards and Technology
OML	Optimized Migration Layer
OS	Operating System
OVAF	Optimal Virtual Machine Allocation Framework
PaaS	Platform as a Service
PM	Physical Machine
PSO	Particle Swarm Optimization
PSOSA	Particle Swarm Optimization with Simulated Annealing
QoS	Quality of Service
RAM	Random Access Memory
RBD	Rados Block Device
SA	Simulated Annealing

SaaS	Software as a Service
SJF	Shortest Job First
SLA	Service Level Agreement
SVM	Support Vector Machine
TCO	Total Cost of Ownership
TLBO	Teaching-Learning-Based Optimization
TMMO	Threshold Multi-Objective Memetic Optimization
T-MMORRS	Threshold Based Multi-Objective Memetic Optimized Round Robin Scheduling
TS	Task Scheduling
VM	Virtual Machine
VMM	Virtual Machine Monitor
VMP	Virtual Machine Placement
WMMORRS	Weighted Multi-Objective Memetic Optimized Round Robin Scheduling

LIST OF FIGURES

LIST OF TABLES

PAPER PUBLICATIONS

INTERNATIONAL JOURAL PUBLICATIONS

[1] **P. Nithya, Dr. L. Jayasimman**, "**An Adaptive Virtual Machine Selection and Migration Approach for VM Consolidation in Cloud Computing**", ***International Journal of Electrical Engineering and Technology (IJEET)***, Volume 11, Issue 6, August 2020, pp. 145-154.

[2] **P. Nithya, Dr. L. Jayasimman**, "**Host Detection Techniques for Efficient Virtual Machine Consolidation in Cloud Computing**", ***International Journal of Advanced Research in Engineering and Technology (IJARET)***, Volume 11, Issue 9, September 2020, pp. 1245-1254.

[3] **P. Nithya, Dr. L. Jayasimman**, "**Cultural Swarm based Task Scheduling (CSTS) Technique for Cloud Computing Environment**", ***International Journal of Advanced Science and Technology***, Vol. 29, No. 9s, (2020), pp. 5025-5034.

[4] **P. Nithya, Dr. L. Jayasimman**, "**An Investigation on Performance Analysis of Host Detection Techniques for Efficient Virtual Machine Consolidation in Cloud Computing**", ***Journal of Xidian University***, Vol. 14, No. 8, (2020), pp. 63-71.

[5] **P. Nithya, Dr. L. Jayasimman**, "**Optimization Based Host Detection Technique for VM Consolidation in Cloud Computing**", ***Design Engineering***, Vol. 2021, No. 079, (2021), pp. 7083-7098.

CHAPTER – 1

INTRODUCTION

1.1 OVERVIEW

Cloud computing includes parallel computing, grid computing, and distributed computing, all of which will be provided as a service. Cloud Computing (CC) is a cutting-edge technology. CC not only helps Information Technology (IT) focus on resource utilisation projects, but it also reduces IT capital expenditure and data centre space because it provides resources to clients as a service rather than a product or software. As a result, IT is shifting to CC. It offers resources to Internet consumers by hiding the platform with other facts utilising virtualization theory. Cloud service providers (CSP) provide computational and storage resources at reduced prices. Organizations are changing to advanced business models in this dynamic IT world, where Internet-of-things (IoT) and Big Data knowledge must be applied to survive. CC has gotten a lot of attention because of its metamorphosis. Concerns about its dependability and trust, mainly on information security, as well as the several other issues encountered by CC [1][2], are also present.

The core concept of CC is that computing takes place "in the cloud," which is a metaphor for the Internet or a network. It entails software access, data storage, and related services in the cloud. The CC model allows network users to make appropriate requests to a pool of significant computing resources (e.g., networks, applications, services, storage, and servers) that are quickly provisioned and approved with minimal administrative effort or CSP intervention.

Because it is still in the exploratory stage, the CC is defined from various angles. The National Institute of Standards and Technology (NIST) has presented a fundamental idea of CC that is widely accepted [3]. The US National Institute of Standards and Technology (NIST) has developed working explanations for the universally accepted CC principles. They define CC as a model that allows several customers to share many computing resources as services. Clients can easily update or adjust their service demands in this approach at no cost.

The NIST working definition summarizes CC as:

The NIST definition is the most extensive and distinct explanation of CC, and it is widely cited in US regime initiatives and papers. According to this description, CC consumes five critical characteristic models, four deployment models, and three service models. The following are essential characteristics:

- **On-demand self-service:** whenever computational resources may be integrated and used without human contact with CSP Storage, processing power, virtual machines, and other computational resources are examples.
- **Broad network access:** Computational resources can be accessed via the network using a variety of devices, including mobile phones and laptops.
- **Resource pooling:** For multi-user sharing, CSP groups their resources. Multiple tenancies are when a physical server hosts multiple virtual machines for distinct users.
- **Rapid elasticity:** Users quickly obtain more resources from the cloud. They are reduced in size by releasing resources that are no longer required.
- **Measured service:** The utilisation of resources is calculated using appropriate measures such as storage usage, bandwidth usage, CPU hours, and so on.

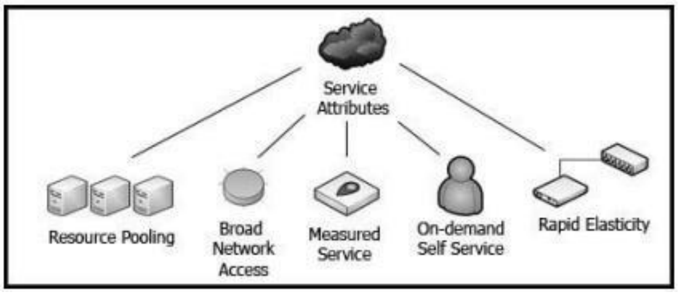

Figure 1.1: Cloud Computing Service Attributes

1.2 CLOUD DEPLOYMENT MODELS

Providers typically employ three types of service models to supply cloud services: software, infrastructure, and platform, with four types of deployment models: hybrid, private, community cloud, and public cloud. [4][5].

1.2.1 Public Cloud

A cloud that the general public can use. Because of its openness, this cloud is not the most secure. The public cloud is the most common type of contemporary CC deployment architecture; it requires significant investment and is typically held by huge organisations such as Google (AppEngine), Microsoft (Azure), or Amazon (Amazon Web Services) (EC2, S3).

1.2.2 Private Cloud

One corporation, organisation, or one of its clients uses a cloud exclusively. The cloud can be managed independently or by others. Private clouds provide better security at a higher expense. Because all data is stored on private servers, private clouds provide the highest level of security to their organisations. While the organisation bears the initial setup costs, the security benefit it delivers is well worth it. In contrast to the public cloud, where boundless resources are utilised and a high level of scalability is achieved, scalability is dependent on current resources and is insufficient.

1.2.3 Community Cloud

It is a cloud in which cloud services are shared by numerous enterprises. It aids a specific community of unique observations (e.g. security, mission, considerations, etc.). It is carried out by the organisation or a third party, either internal or external to the company. This cloud, like the private cloud, provides secure infrastructure for the enterprises participating.

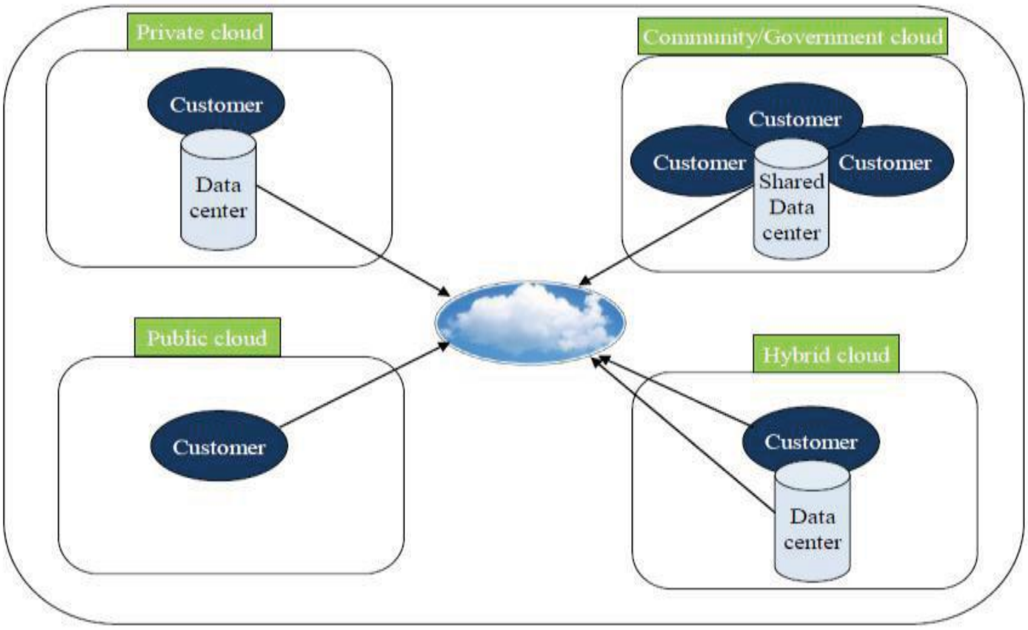

Figure 1.2: Cloud Deployment Models

1.2.4 Hybrid Cloud

A model that consists of two or more clouds in a certain configuration (public, private, community). These clouds are distinct entities that are bound by certain technology or standards, allowing for application (e.g., load balancing during cloud bursts) and data portability. Hybrid clouds combine the advantages of both private and public clouds. Scalability is no longer limited to private resources; instead, it can be expanded to public cloud resources as needed. Similarly, data security concerns in hybrid cloud can be readily addressed by storing critical and secret data on private servers.

1.3 CLOUD COMPUTING ACTORS

Providers, users, and brokers are the three primary participants in the Cloud environment, each with their own functions and relationships [6].

- **Cloud Provider:** The CSP is responsible for maintaining the Cloud infrastructure on which Cloud services are built. This actor is in charge of managing and controlling cloud resources as well as responding to user queries.
- **Cloud User:** A Cloud user is a company or individual who uses Cloud services.
- **Cloud Broker:** The Broker acts as a link between Cloud consumers and CSPs. It is in charge of distributing received requests among the various providers based on the needs of the users. A Cloud broker, to use a simplistic comparison, is like a travel agency that operates as a mediator between clients and CSPs.

1.4 IMPACT OF CLOUD COMPUTING

With its disadvantages and advantages, CC has the potential to influence the IT industry in a variety of ways. The two main advantages that CC provides to businesses are discussed. First, CC allows organisations to use computer resources that they do not own or operate, lowering computing costs. Next, CC offers more prebuilt components so that businesses don't have to start from scratch. These two aspects will aid cloud computing in obtaining numerous opportunities for practise in enterprise IT. According to further research, CC will introduce a slew of new features to enterprise IT. First, CC provides a framework that allows solutions from various cloud resources to be equalised. Furthermore, CC has the potential to expand company IT activities and broaden bandwidth commoditization, allowing businesses to use CC as if it were a local network. Finally, CC can favourably benefit company IT because it may always search out the most innovative or up-to-date cloud resource providers [7][8].

From an application standpoint, CC has designed or developed a plethora of creative applications. Parallel batch processing, corporate analytics, mobile interactive applications, the accumulation of compute-intensive desktop programmes, and some applications for specific regions are among them. From another aspect, many economic analysts anticipate that cloud computing will invoke a tremendous influence on the computing sector. Companies that have already made significant investments in software infrastructure and datacenters may decide to become CC providers with the goals of increasing profits, improving the efficiency of existing investments, defending a franchise, growing reputations, strengthening customer relationships, and appealing a platform.

1.5 CLOUD SERVICE MODELS

Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS) are the three types of traditional Cloud service models (IaaS) [9][10].

1.5.1 Infrastructure as a Service (IaaS)

IaaS is the most cost-effective way to provide cloud services. It refers to the provisioning and delivery of VMs, networks, physical servers, and storage resources. Instead of investing in their own infrastructure, businesses can rent resources and use them on demand rather than having their own. Users who utilise IaaS have direct access to the bottom of the stack and are capable of creating their own application environments from start. Amazon EC2 is an example of a popular IaaS Cloud.

1.5.2 Platform as a Service (PaaS)

In comparison to IaaS, PaaS is a more sophisticated and higher-level service. It provides application development environments and software platforms that let users to manage, deploy, and build Cloud applications without having to worry about the technology. Google App Engine and Microsoft Azure Services are the most popular cloud systems.

1.5.3 Software as a Service (SaaS)

The highest level of the Cloud service model is SaaS. As a result, whole apps are made available to consumers via the internet. The infrastructure is managed by SaaS providers, and they have complete control over the application software. Users just approach their requests as if they were hosted locally, with no need to know anything about the Cloud or the technology's specifics. Social media platforms, email, and project management systems are examples of SaaS applications, with Google Apps and Google Documents being the most popular.

1.6 OVERVIEW OF CLOUD VIRTUALIZATION

CC's adoration revolves around the concept of virtualisation. This technique allows multiple operating system (OS) instances to run on a single physical machine (PM), allowing hardware to be separated from a single OS. A virtual machine monitor (VMM), often known as the hypervisor [11][12], oversees each guest's 'operating system.' Virtualization allows an operator to manage memory, CPU, storage, and other resources for a guest OS, ensuring that these resources are effectively shared among the guest OS.

It is possible to move an OS from one PM to another since the guest OS is not limited to the hardware. VM migration is another name for this technology. A cloud datacenter's support can be equated to VM migration. When a guest OS or VM begins to consume the most resources during a peak period, it can be moved to a PM with lower demands.

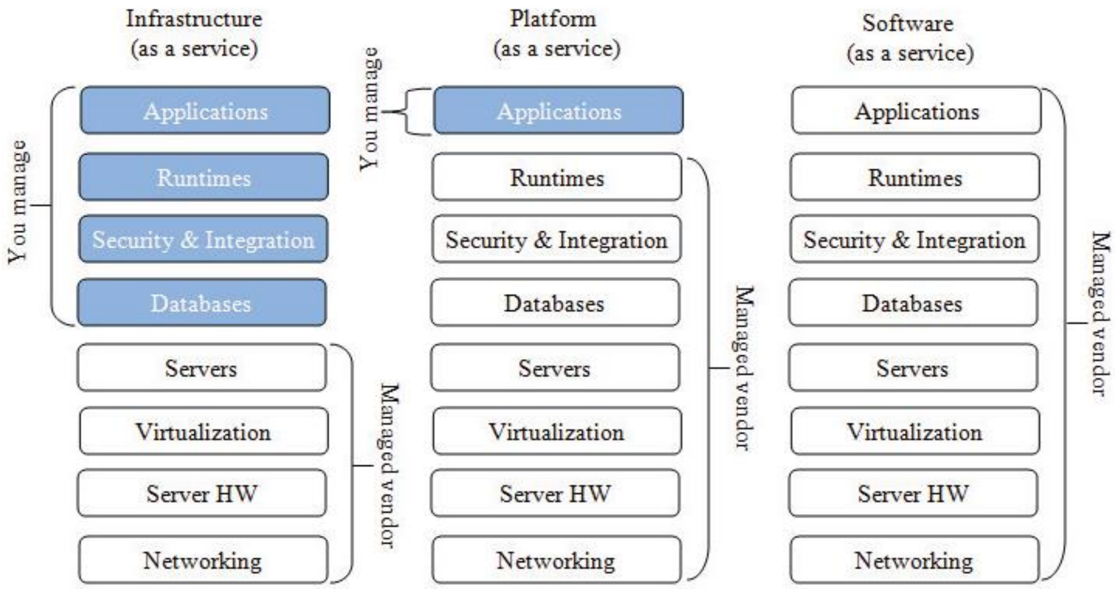

Figure 1.3: Cloud Computing Service Models

Fault tolerance, system performance, and manageability are all goals of VM Migration. It exhibits the load-balancing aspect of a datacenter by migrating VMs from an overburdened server to an underburdened server or PM. It helps a datacenter save energy by allowing underutilised PMs to stop operating by shifting their VMs to other PMs with better utilisation. The key secret of a VM migration is that the application and its associated processes are unaware of the VM migration's progress, hence imparting the sense of transparency.

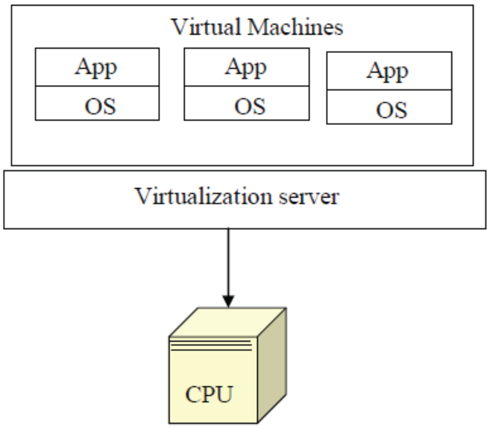

Figure 1.4: Cloud Virtualization Structure

A supplier in CC does not directly deliver real resources to its clients. It is more expensive to dedicate one computer to a single client than to provide it to a large number of clients. The supplier employs virtualization, a computer architecture technology, to provide the same services to numerous clients while only using one computer. A virtual machine's goal is to improve resource sharing among many users while also improving the computer's performance in terms of resource utility and application flexibility. Hardware resources like as CPUs, I/O devices, memory, and so on, as well as software resources such as software libraries and the operating system, are virtualized in many functional layers that are reliant on the applications.

Virtualization's main goal is to separate the hardware and software layers in order to improve system efficiency. Virtualization techniques are also used to improve the efficiency of computational engines, storage, and networks. Virtualization allows any computer platform to be developed or fixed in another host computer, even if the processors and operating systems are different.

Consumers often utilise a standard computer with a host operating system that is optimised for the hardware. With virtualization, a traditional computer can offer different user applications that each require their own operating system (known as guest OSs) that run on the same hardware as the host OS. Virtualization layers are available in many host operating systems to enable this model of operation. VMM, or hypervisor, is the name of this layer.

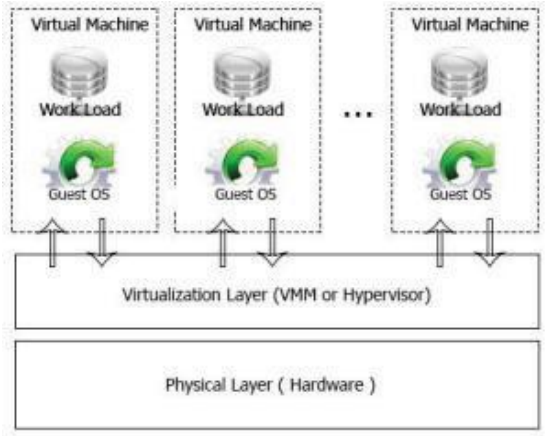

Figure 1.5: Layered Virtualization Technology

1.7 ARCHITECTURE OF VIRTUALIZATION

Portioning, isolation, and encapsulation are the three primary aspects of virtualization architectures.

Every VM is isolated from other VMs as well as its physical host system. The main advantage is that if one VM crashes, it does not affect other VMs or the host. Data in one VM is not visible on other VMs because they are virtual machines. Data in isolated virtual machines (VMs) cannot be shared or accessed by another computer or user. Virtualization employs an encapsulation technique that provides complete control over system resources. The hypervisor provides a background for the programmes to generate and operate a VM, which is primarily similar to the unique machine environment [13][14].

There are five diversified levels of virtualization:

- Application Level

- Library Level
- OS Level
- Hardware abstraction layer level and in addition
- Instruction set architecture level

Every stage of implementation serves a different purpose. Virtualization at the application level is used to separate an application installation from the client machine that uses it. Virtualization at the library level is used to hide the OS-related nitty-gritty details from ordinary programmers. OS level virtualization is a server virtualization procedure in which the OS kernel allows many remote user space instances. The hardware abstraction layer level is used to create a unique architecture that utilises the host platforms, such as an emulator, simulator, and so on. The instruction set architecture level is all about instruction set imitation, which is a software-only method of considering instructions.

CC's extremely flexible IaaS approach is increasingly gaining traction among businesses as a way to lower their Total Cost of Ownership (TCO) by better utilising resources and money. Amazon's Elastic Compute Cloud (EC2) is now forcing firms to switch to green data centres, requiring the use of passively running computers. This necessitates adequate hardware solutions, such as Virtualization, which is the CC technology's backbone. It promotes collaboration by dividing computed resources. The VMM, a smaller software programme, manages and controls all VM-related operations. Load balancing (LB) is aided by live VM migration during amalgamation.

The importance of virtual machines is that they provide a notion of hardware that allows services to run on heterogeneous hardware. Furthermore, they enable isolation between processes running in distinct VMs, which serves as the foundation for security attributes shared by services belonging to different clients of the same infrastructure provider. Finally, the fact that several VMs are executed within the same PM allows for consolidation strategies to boost PM utilisation rates [15].

1.8 PROBLEM STATEMENT

The information parameters to resource allocation strategy and the way of resource allocation vary based on the services, infrastructure and the nature of applications which demand resources. Coming up next are the issues considered the resource allocation in the cloud condition.

- **Policy:** Since centralized client and resource management lacks in scalable management of clients, resources and organization level security policy.
- **Execution time:** estimating the execution time for a vocation is a hard task for an error and client.
- **Virtual Machine:** The framework made out of a virtual network of virtual machines capable of live migration across physical infrastructure of multidomain.

1.9 OBJECTIVE OF THE RESEARCH WORK

The following are the objectives of the research work:

- To reduce the energy consumption for the VM allocation, selection and Migration.
- To reduce the Service Level Agreement (SLA) violations to improve the Quality of Service (QoS).

- To reduce the performance degradation due to migration (PDM) for improving the QoS.

1.10 ORGANIZATION OF THE THESIS

The thesis will consist of the chapters as follows:

Chapter 1 gives an introduction where the theoretical background of the given topic is discussed.

Chapter 2 presents the review of the literature of the previous research findings, related to the topic chosen and also provides the foundation to proceed in the research work.

Chapter 3 depicts the proposed Cultural Swarm based Task Scheduling (CS-TS) Technique for improving Quality of Service in Cloud Computing environment.

Chapter 4 presents the proposed Optimization based Host Detection Technique for VM Consolidation in Cloud Computing.

Chapter 5 presents the proposed An Adaptive Virtual Machine Selection and Migration Approach for VM consolidation in Cloud Computing.

Chapter 6 describes the results and discussions of the proposed works. Then finally, the comparison of the proposed work with an existing technique.

Chapter 7 presents the summary of the previous chapters, conclusion, and highlights the research contributions and further extensions of the research.

CHAPTER – 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 INTRODUCTION

Cloud computing is evolving as a new computing paradigm that aims to provide end-users with stable, modified, and QoS (Quality of Service)-assured computing dynamic situations [16]. Cloud computing is the combination of grid computing, parallel processing, and distributed processing. The fundamental rule of cloud computing is that client data is stored in the web's data centre rather than locally. The cloud computing service providers could monitor and maintain the activity of these data centres. Clients can access their stored data at any time by using the Application Programming Interface (API) provided by cloud providers through any terminal equipment connected to the internet.

Not only are storage services offered, but also software and hardware services to the general public and business markets. Service providers might provide everything from software or platforms to infrastructure and resources. SaaS, PaaS, and IaaS are all terms for similar services [17]. Cloud computing has a number of advantages, the most basic of which are decreased prices, resource re-provisioning, and remote access. Cloud computing saves money by avoiding the company's capital investment in leasing physical infrastructure from a third-party provider. We can quickly obtain extra resources from cloud suppliers when we need to expand our firm because of the adaptive nature of cloud computing. We can access cloud services from anywhere at any time thanks to remote accessibility. To get the most out of the above-mentioned benefits, the services provided in terms of resources should be allocated best to cloud-based apps.

2.2 BACKGROUND STUDY ON TASK SCHEDULING

The term "cloud computing" refers to the use of a platform, software, or computing as a service. It's a utility billing technique in which the customer just pays for what they use and isn't needed to have the necessary infrastructure. Virtual machines (VMs) are used to spread compute resources. Virtualization allows you to run at least two operating systems on a single computer or embedded device. As a result, it aids in the efficient use of operational resources and the development of an effective system. The cloud environment gives huge potential and capabilities, and scheduling algorithms are required to utilise them effectively and properly. The cloud resource manager is responsible for assigning tasks to cloud resources. For cloud environments, there are several scheduling algorithms [18][65].

Cloud scheduling algorithms' primary goal is to reduce overall task completion time by identifying the most appropriate resources to allocate to tasks. However, reducing the overall completion time of tasks does not always imply a reduction in the execution time of each individual task. As a result, the task scheduling algorithm aims to efficiently schedule jobs in order to reduce turnaround time and maximise resource usage.

Application scalability is a major benefit of cloud computing. Scalability of Cloud Resources allows for immediate resource provisioning to meet application needs. Tasks are scheduled on a regular basis based on user requirements. Novel scheduling methods must be presented to overcome the challenges given by network properties between resources and users. To achieve better and more effective job scheduling, various scheduling methodologies are integrated. In the past, grids used scheduling algorithms. Implementing scheduling algorithms in the cloud has a precondition. Grid performance has been lowered.

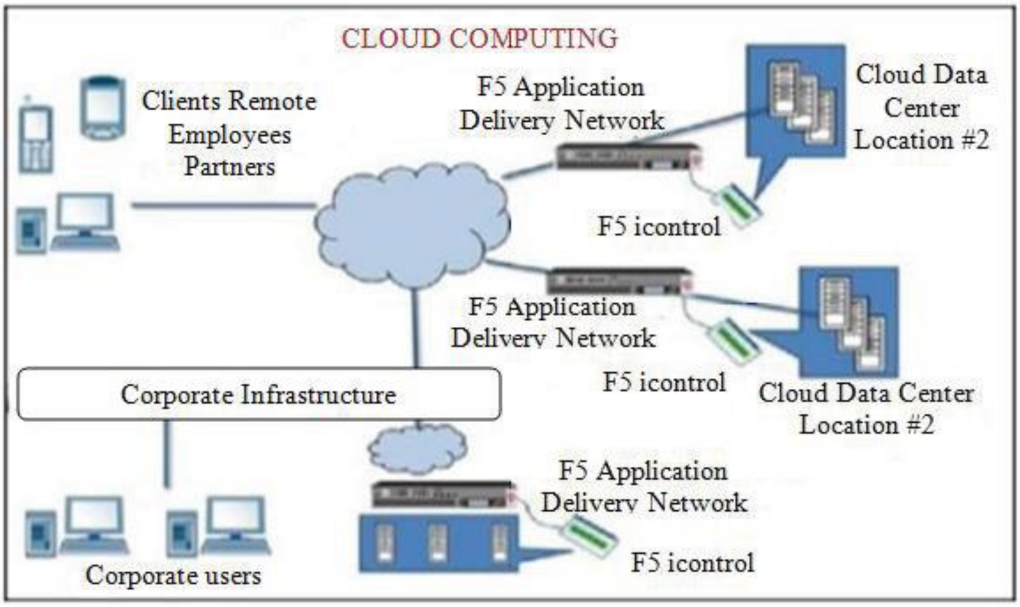

Figure 2.1: Scheduling in Cloud Computing Environment

As shown in Figure 2.1, a Scheduling procedure in the CC circumstance contains a few auxiliary components. This paradigm is provided for the CC system's use of virtualization. The number of VMs in the cloud framework that processes the submitted tasks provides all calculating tools such as operating system, software, and so on.

- **Computing entity:** They are assigned based on processing capacity, which is measured in the number of instructions it can handle in a given amount of time.
- **Job scheduler** In the CC condition, it is a critical component of the scheduling technique. It determines the activity sequence for each of the jobs in the queue to be executed.
- **Job waiting queue:** It's the queue where jobs are waiting to be assigned to a proper machine for processing.
- **Job arriving process:** It is the method by which jobs are placed in the scheduling system.

2.2.1 Task Scheduling

Many resources, like as processors, memory, storage, and multiple programmes, are used in the CC with the end purpose of providing services. The resources are designed and acquired according to the requirements. The CC prototype lowered hardware and software acquisition costs while also increasing service quality. The concept of virtualization is used to share cloud resources among end-clients [25].

Under remote running situations, virtualization allows for optimal utilisation of physical resources and energy. In the cloud datacenter, a virtual machine serves as an important component of software stacks.

In the field of computer science, task scheduling is a point of reference. Scheduling can help with QoS by allocating resources among tasks supplied by customers at a specific time. The goal of task scheduling is to ensure that a task is assigned and completed on the appropriate resource and at the appropriate time. Process scheduling, which refers to thread management in an operating system, and energy management via the task scheduler in CC, are two exciting study topics. In recent years, people have been drawn to CC because of its dependability, scalability, cost-cutting, and information-sharing qualities that can be accessed at any time and from any location. All things considered, CC is currently the main emphasis of everyone, as opposed to other developments with the final objective of research and utility [26].

A process scheduler's goal in CC is to ensure that all resources are used correctly among the jobs provided by various users. There are several requests for a large number of tasks to the cloud framework, making it difficult for the cloud framework to allocate all resources to all assignments. This must be completed while keeping in mind that the QoS must not be compromised. Effective job scheduling is a difficult requirement for ensuring QoS to consumers. Users, predictably, will be hesitant to pay for services that do not meet their expectations. As a result, scheduling is crucial in CC culture.

The fundamental point of the CC framework is to make optimal use of all available resources. The scheduling algorithms play a key role in achieving this. As a result, users' requests for jobs should be lawfully planned using task scheduling algorithms. A scheduling algorithm's main goals are to reduce execution time, increase resource utilisation, and achieve LB. The primary rationale for task scheduling in any computing framework is to distribute assignments in an acceptable sequence under issue-specific imperatives. The great performance of the CC environment is prompted by productive resource scheduling [30][31]. The majority of extant scheduling algorithms prioritise various factors such as cost reduction, makepan minimization, scheduling rate optimization, resource use, and more.

2.2.2 Types of Task Scheduling

The different types of task scheduling in cloud computing environment are:

- ***User level scheduling:*** Market-based and auction-based scheduling are two types of user-level scheduling. In user level scheduling, methods such as First In First Out (FIFO), priority-based, and non-pre-emptive scheduling are utilised.
- ***Cloud Service Scheduling:*** Cloud service scheduling is divided into two categories: user and system. It primarily evaluates the service in terms of difficulties between the supplier and the user at the user level. Scheduling and resource management are handled at the system level. Fault tolerance, reliability, resource sharing, and QoS parameters are all taken into account in addition to real-time satisfaction.

- ***Static and Dynamic Scheduling:*** Static scheduling allows for the pre-fetching of essential data and the pipelining of task execution phases. The overhead of static scheduling is minimal. In the case of dynamic scheduling, the work components or task information is unknown beforehand. As a result, the task's execution time may be unknown, and task allocation occurs only as the application runs.
- ***Heuristics Scheduling:*** In cloud environment, heuristic based scheduling can be done to optimise results. Heuristic approaches can produce more accurate outcomes.
- ***Workflow Scheduling:*** For the management of workflow execution, workflow scheduling is done. Real Time Scheduling: Real Time Scheduling in cloud environment is done to increase the throughput and to decrease the average response time instead of meeting deadline.

2.3 LITERATURE REVIEW ON TASK SCHEDULING

Shah-Mansouri, Hamed, Vincent WS Wong, and Robert Schober [16] studied the following problems in Mobile Cloud Computing (MCC) systems: (i) which tasks should be offloaded to cloud servers, (ii) and what is the optimal price of cloud services. The authors further proposed an algorithm using convexification and primal dual methods to mitigate the non-convexity. Through numerical studies, the authors investigated the mobile users' behavior and the Cloud Service Providers (CSP's) pricing strategy.

Kong, Weiwei, Yang Lei, and Jing Ma [17] a novel adaptive VM resource scheduling algorithm based on auction mechanism is presented to overcome the problem of virtual machine (VM) scheduling in the cloud computing environment by considering multiple factors including network bandwidth and auction deadline.

Ma, Juntao, et al [18] proposed a novel dynamic task scheduling algorithm based on improved genetic algorithm. On the basis of the genetic algorithm, the proposed algorithm gives full consideration to the dynamic characteristics of the cloud computing environment. The CloudSim simulation platform is selected for simulation; experimental results show that the proposed algorithm can effectively improve the throughput of cloud computing systems, and can significantly reduce the execution time of task scheduling.

Verma, Manish, et al [19] presented a dynamic resource demand prediction and allocation framework in multi-tenant service clouds. The novel contribution of the proposed framework is that it classifies the service tenants as per whether their resource requirements would increase or not; based on this classification, our framework prioritizes prediction for those service tenants in which resource demand would increase, thereby minimizing the time needed for prediction.

Bala, Anju, and Inderveer Chana [20] priority-based task scheduling approach has been proposed that prioritizes the workflow tasks based on the length of the instructions. The proposed scheduling approach prioritize the tasks of cloud applications according to the limits set by six sigma control charts based on dynamic threshold values.

Yang, Xianda, et al [21] proposed a multi-resource virtual machine allocation algorithm named Dominant Resource First Allocation (DRFA). This paper aimed to maximize the resource utilization in heterogeneous cloud computing environment. By computing the dominant resource under multiple resource dimensions, our proposed algorithm DRFA can make full advantage of the heterogeneous physical resources.

Saxena, Deepika, R. K. Chauhan, and Ramesh Kait [22] The very objective of this paper is to dynamically optimize task scheduling at system level as well as user level.

This paper relates benefit-fairness algorithm based on weighted-fair Queuing model which is much more efficient than simple priority queuing. In proposed algorithm, the authors have classified and grouped all tasks as deadline based and minimum cost-based constraints and after dynamic optimization, priority of fairness is applied. Here different priority queue (high, mid, low) are implemented in round-robin fashion as per weights assign to them.

Tao, Dan, et al [23] In this paper, we firstly propose a new Dynamic Hadoop Cluster on IaaS (DHCI) architecture, which includes four key modules: monitoring module, scheduling module, virtual machine management module and virtual machine migration module. Secondly, the authors presented a load feedback-based resource scheduling scheme. Thirdly, the authors reused the method of VM migration and propose a dynamic migration-based data locality scheme.

Djebbar, Esma Insaf, and Ghalem Belalem [24] proposed two strategies for task scheduling and resource allocation for high data in Cloud computing. The main objective is to improve data management in virtual machine in Cloud computing and optimize the total execution time of all tasks.

Pop, Florin, and Maria Potop-Butucaru [25] this paper presented advances in virtual machine assignment and placement, multi-objective and multi-constraints job scheduling, resource management in federated Clouds and in heterogeneous environments, dynamic topology for data distribution, workflow performance improvement, energy efficiency techniques and assurance of Service Level Agreements.

Wang, Xiaoli, Yuping Wang, and Yue Cui [26] propose a new energy aware multi-job scheduling model based on MapReduce in this paper. In the proposed model, first, the variation of energy consumption with the performance of servers is taken into account; second, since network bandwidth is a relatively limited resource in cloud computing, 100%data locality is guaranteed; last but not least, considering that task-scheduling strategies depend directly on data placement policies, the authors formulate the problem as an integer bi-level programming model. It is worth noticing that there are usually tens of thousands of tasks to be scheduled in the cloud, so this is a large-scale optimization problem. In order to solve it efficiently, a local search operator is specifically designed, based on which, a bi-level genetic algorithm is proposed in this paper.

Mousavi, Seyedmajid, Amir Mosavi, and Annamária R. Varkonyi-Koczy [27] this paper proposed a hybrid load balancing algorithm with combination of Teaching-Learning-Based Optimization (TLBO) and Grey Wolves Optimization algorithms (GWO), which can well contribute in maximizing the throughput using well balanced load across virtual machines and overcome the problem of trap into local optimum. The hybrid algorithm is benchmarked on eleven test functions and a comparative study is conducted to verify the results with particle swarm optimization (PSO), Biogeography-based optimization (BBO), and GWO.

Singh, Aarti, Dimple Juneja, and Manisha Malhotra [28] this research proposes a new Agent based Automated Service Composition (A2SC) algorithm comprising of request processing and automated service composition phases and is not only responsible for searching comprehensive services but also considers reducing the cost of virtual machines which are consumed by on-demand services only.

Sarkhel, Preeta, Himansu Das, and Lalit K. Vashishtha [29] proposed three different task-scheduling algorithms such as Minimum-Level Priority Queue (MLPQ), MIN-Median, Mean-MIN-MAX which aims to minimize the makespan with maximum utilization of cloud. The results of the proposed algorithms are also compared with some existing algorithms such as Cloud List Scheduling (CLS) and Minimum Completion Cloud (MCC) Scheduling.

Mohanty, Subhadarshini, et al [30] In this paper, a load balancing algorithm using Multi Particle Swarm Optimization (MPSO) has been developed by utilizing the benefits of particle swarm optimization (PSO) algorithm. Proposed approach aims to minimize the task overhead and maximize the resource utilization in a homogenous cloud environment. Performance comparisons are made with Genetic Algorithm (GA), Multi GA, PSO and other popular algorithms on different measures like makespan calculation and resource utilization.

Priya, V., and C. Nelson Kennedy Babu [31] proposed a method called, Moving Average-based Fuzzy Resource Scheduling (MV-FRS) for virtualized cloud environment to optimize the scheduling of resources through virtual machines. Initially, the MV-FRS method starts by predicting the resource (i.e. bandwidth, memory and processing cycle) requirements. Then a measure of relationships between availability of resources and the requirements of resources are made. Finally, a fuzzy control theory is designed to accomplish system accessibility between user cloud requirements and cloud users' resources availability.

Suri, P. K., and Sunita Rani [32] The proposed scheduling model is constructed for cloud applications in multi cloud environment and implemented in three phases (minimization, grouping & ranking and execution) and considered average waiting time, average turnaround time, completion time and makespan as performance parameters. In this scheduling model, execution time of tasks in cloud applications is generated through normal distribution and exponential distribution. Ranking of tasks is based upon shortest job first strategy (SJF) and results are compared with other ranking method based upon first come first serve (FCFS) and largest processing time first (LPTF).

Seth, Sonam, and Nipur Singh [33] proposed resource allocation model on the basis of dynamic parameters. The proposed method, dynamic threshold-based dynamic resource allocation can optimize the resource utilization and time.

Singh, Poonam, Maitreyee Dutta, and Naveen Aggarwal [34] This paper accomplished a review of using meta-heuristics techniques for scheduling tasks in cloud computing. We presented the taxonomy and comparative review on these algorithms. Methodical analysis of task scheduling in cloud and grid computing is presented based on swarm intelligence and bio-inspired techniques. This work will enable the readers to decide suitable approach for suggesting better schemes for scheduling user's application.

Alla, Hicham Ben, Said Ben Alla, and Abdellah Ezzati [35] proposed a new Dynamic Priority-Queue (DPQ) approach based on a hybrid multi-criteria decision making (MCDM) namely ELECTRE III and Differential Evolution (DE). Furthermore, to schedule the tasks, the authors introduced a hybrid meta-heuristic algorithm based on Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The proposed DEELDPQ-SAPSO approach has been validated through the CloudSim simulator.

Latiff, Muhammad Shafie Abd, Syed Hamid Hussain Madni, and Mohammed Abdullahi [36] proposed a dynamic clustering league championship algorithm (DCLCA) scheduling technique for fault tolerance awareness to address cloud task execution which would reflect on the current available resources and reduce the untimely failure of autonomous tasks.

Seth, Sonam, and Nipur Singh [37] The problem of resource utilization in heterogeneous computing system has been studied with variations. Scheduling of independent, noncommunicating, variable length tasks in the concern of CPU utilization, low energy consumption, and makespan using dynamic heterogeneous shortest job first (DHSJF) model is discussed in this paper. Tasks are scheduled in such a manner to minimize the actual CPU time and overall system execution time or makespan.

Juarez, Fredy, Jorge Ejarque, and Rosa M. Badia [38] proposed a real-time dynamic scheduling system to execute efficiently task-based applications on distributed computing platforms in order to minimize the energy consumption. Scheduling tasks on multiprocessors is a well know NP-hard problem and optimal solution of these problems is not feasible, we present a polynomial-time algorithm that combines a set of heuristic rules and a resource allocation technique in order to get good solutions on an affordable time scale.

Arya, K. S., P. V. Divya, and KR Remesh Babu [39] In this paper dynamic method for task allocation and resource allocation is introduced to reduce virtual machine migrations and execution time. The proposed algorithm is simulated and results are compared with the existing algorithm.

Agarwal, Mohit, and Gur Mauj Saran Srivastava [40] proposed the cuckoo search-based task scheduling approach which helps in distributing the tasks efficiently among the available VM's and also keeps the overall response time QoS minimum. This algorithm assigns the tasks among the virtual machines on the basis of their processing power, i.e., million instructions per seconds (MIPS) and length of the tasks.

Moazemi, Setareh, and Mehdi Javanmard [41] this paper proposed a virtual energy virtualization migration technique that emits live VMs from an active node to another active node. The proposed technique uses the biography-inspired worn-out optimization technique to find the best node for over-migrating VMs to achieve energy efficiency in cloud data centers. This optimizes energy efficiency through the optimal migration of VMs, thereby improving the level of resource utilization.

Alla, Hicham Ben, et al [42] proposed two hybrid metaheuristic algorithms, the first one using Fuzzy Logic with Particle Swarm Optimization algorithm (TSDQ-FLPSO), the second one using Simulated Annealing with Particle Swarm Optimization algorithm (TSDQ-SAPSO).

Xu, Xiaolong, et al [43] a dynamic resource allocation method, named DRAM, for load balancing in fog environment is proposed in this paper. Technically, a system framework for fog computing and the load-balance analysis for various types of computing nodes are presented first. Then, a corresponding resource allocation method in the fog environment is designed through static resource allocation and dynamic service migration to achieve the load balance for the fog computing systems.

Fatima, Aisha, et al [44] In this article, a resource allocation model is presented in order to optimize the resources in residential buildings. The whole world is categorized into six regions depending on its continents. The fog helps cloud computing connectivity on the edge network. It also saves data temporarily and sends to the cloud for permanent storage. Each continent has one fog which deals with three clusters having 100 buildings.

Microgrids (MGs) are used for the effective electricity distribution among the consumers. Particle Swarm Optimization with Simulated Annealing (PSOSA) is used for load balancing of Virtual Machines (VMs) using multiple service broker policies.

Toosi, Adel Nadjaran, Richard O. Sinnott, and Rajkumar Buyya [45] proposed a new resource provisioning algorithm to support the deadline requirements of data-intensive applications in hybrid cloud environments. To evaluate the proposed algorithm, implement it in Aneka, a platform for developing scalable applications on the Cloud.

2.4 BACKGROUND STUDY ON VIRTUAL MACHINE CONSOLIDATION

Cloud computing has emerged as the next major revolution in computer networks and Web provisioning in recent years. As a result of the increased demand, many vendors, including Amazon and IBM, have begun designing, creating, and implementing Cloud technologies to maximise the use of their own data centres, and some open-source solutions, such as Eucalyptus and OpenStack, are also in the works. Virtualization strategies are used in cloud architectures to provision multiple VMs on the same physical host in order to efficiently leverage available resources, such as consolidating VMs into the smallest number of physical servers possible to minimise runtime power consumption. In order to prevent performance degradation and SLA breaches, VM consolidation must carefully consider the aggregated resource usage of co-located VMs [47][49].

In recent years, cloud computing architectures have gotten a lot of attention, and some vendors are looking at them as a viable option for optimising the use of their own infrastructures [52]. Indeed, these systems have pools of virtualized computing resources that are charged on a pay-per-use basis, lowering the initial investment and maintenance costs dramatically. As a result, the Cloud's key strength is in supplying computing tools to third-party service providers who don't want to develop or manage their own IT infrastructure.

At this time, however, many management problems, such as Cloud architectures for large-scale data centres that require substantial financial investments on the part of the Cloud owner, require further investigation. Above all, the energy consumed by IT resources (servers and network elements) as well as cooling systems adds a major cost to the equation. As a result, cloud providers are looking for efficient techniques that dynamically reconfigure IT infrastructure to reduce total data centre power consumption, which is also fueled by emerging Green Computing research [60]. To achieve this, diligent application of virtualization techniques aims to minimise power consumption by consolidating the execution of multiple Virtual Machines (VMs) on the same physical host, a process known as VM consolidation. The key concept is to run the virtual machines on as few physical servers as possible in order to concentrate the workload and keep the number of physical servers turned on to a minimum. For example, if two physical servers each run one VM and neither is using its full computing power, assigning both VMs to the same physical server will save money by turning one server off. It goes without saying that in large-scale data centres, where many physical servers can be operated to save more and more resources, power savings will become increasingly efficient.

Since it leads to an optimum utilisation of available resources while preventing extreme performance degradation due to resource usage of co-located VMs, VM consolidation poses a number of management concerns. On the one hand, several works have already addressed VM consolidation by considering local physical node resource constraints, namely CPU and memory sharing, and several algorithms have been proposed to solve the VM consolidation problem with various objectives, such as increasing load balancing among servers, minimising truncation time, and so on. In a more realistic sense, however, deploying such studies and findings in real-world Cloud environments poses new challenges, necessitating new in-field studies to validate their usability and core assumptions [66]. For example, only a few seminal studies of real power savings achieved by VM consolidation for various types of services (CPU and network intensive) are now available, and in-depth studies of networking issues caused by heavy communications between different VMs consolidated over the same physical server are also still lacking, particularly for new Cloud platforms like the vSphere.

2.4.1 Literature Review on VM consolidation

Ashraf, Adnan, and Ivan Porres [46] For virtual machine (VM) consolidation in cloud data centres, a novel multi-objective ant colony system algorithm was presented. The proposed algorithm creates VM migration plans, which are then used to reduce physicalmachine (PM) over-provisioning by consolidating VMs on under-utilized PMs. It optimises two targets that are ranked in order of importance. The proposed algorithm's first and most important goal is to maximise the number of published PMs. Furthermore, since VM migration is a resource-intensive process, it tries to keep the number of VMmigrations to a minimum. In a series of tests, the proposed algorithm is empirically tested.

Yousefipour, Amin, Amir Masoud Rahmani, and Mohsen Jahanshahi [47] presented a statistical model for reducing power usage and costs in cloud data centres by implementing successful VM consolidation. Following that, we propose a genetic algorithm–based meta-heuristic algorithm for resolving the problem, namely, energy and cost-aware VM consolidation.

Chol, Kim Ryo, and Kim Sun Hui [48] presented a taxonomy for dividing server resources into categories. In cloud computing environments based on OpenStack, a virtual machine migration scheme is investigated for server load balancing and power reduction.

Callau-Zori, Mar, et al [49] presented an energy model for VM management operations such as VM placement, VM initialization, and VM migration that provides energy estimation.

Shaw, Subhadra Bose, Anil Kumar Singh, and Shailesh Tripathi [50] The definition of chance has been used to make virtual machine (VM) migration decisions from over-utilized and under-utilized nodes. A novel method for selecting the destination server where a migrated VM would be put has also been suggested. The current usage of the Central Processing Unit (CPU), memory, and network bandwidth is used to develop this approach. The proposed scheme strikes a balance between energy consumption and improved efficiency.

Ding, Weichao, et al [51] Since multiple components need the backing of back-end storage, the storage system's high reliability and scalability is crucial for the OpenStack cloud platform system. In this paper, the RADOS BLOCK DEVICE (RBD), a block storage device for Ceph (one of the most common distributed file systems), is used as the back-end storage for the OpenStack components glance, nova, and cinder. In addition, RBD-OpenStack, a unified storage cloud platform, has been developed. Many advantages have been demonstrated by studying the virtual machine operation process under the RBDOpenStack platform, such as the efficiency of the virtual machine service, and so on.

Zhao, Changming, and Hao Yang [52] A novel online virtual machine placement algorithm was suggested. First, based on historical static data, the law of large numbers, and a given Service Level Agreement, it developed a virtual machine equivalent peak resource utilisation model for one-dimension resource constrained conditions (SLA). Second, using a SLA constrained and virtual machine random exchange strategy, it is able to simplify and transform the placement mechanism to a better bin packing algorithm, according to the equivalent model. Tertiary constructs a placement arrangement algorithm for multi-dimension resource situation based on polling technique and statistic theory.

Jangiti, Saikishor, and Shankar Sriram VS [53] For reserved requests, a migration-based VM consolidation method was proposed. In the simulations, the Real Dataset EC2 was used. During a reservation transition time, the proposed BBPMM demonstrated the elasticity of its ability to change running PMs, reducing 38 percent of running PMs.

Masdari, Mohammad, et al [54] A detailed survey and taxonomy of bio-inspired VMP schemes was presented. To that end, we first go over the basic principles of the VMP and then go over the different goals and considerations that can be considered during the process. Then, based on their implemented optimization algorithms, we create a taxonomy of VMP schemes and compare their employed factors in the VMP phase, as well as simulator environments and metrics used in the verification of the investigated VMP frameworks.

Mohammadhosseini, Mahdieh, Abolfazl Toroghi Haghighat, and Ebrahim Mahdipour [55] The authors proposed a VM placement approach based on the cultural algorithm with the aim of minimising energy consumption in cloud data centres. A new fitness function is implemented to test VM allocation solutions in the proposed algorithm called balance-based cultural algorithm for virtual machine placement (BCAVMP). This function considers balanced resource allocation by using the number of balance vector lengths for each VM placement. Furthermore, by incorporating the amount of energy consumed into the exercise feature, solutions with lower energy consumption are intended.

Shang, Fengjun, Xiong Xiong, and Luzhong Li [56] has conducted extensive research on popular open source cloud computing systems, such as OpenStack, and improved the existing resource scheduling system. The works are divided into two sections: a virtual machine initial placement mechanism based on resource output perception and a virtual machine dynamic migration mechanism based on load perception.

Akintoye, Samson Busuyi, and Antoine Bagula [57] The task allocation and virtual machine placement problems were developed and presented in a single cloud/fog computing setting, and solutions for the task allocation and virtual machine placement problem models were proposed as a task allocation algorithmic solution and a Genetic Algorithm Based Virtual Machine Placement.

Qiao, Lei, et al [58] based on historical evidence, used Genetic Expression Programming (GEP) to derive a regression model for migrate consideration. The authors then performed dynamic VM migrations from overloaded to underloaded hosts, taking into account the VM migration considerations obtained from this regression model.

López, Jorge, Natalia Kushik, and Djamal Zeghlache [59] Dedicated to determining the consistency of virtual machine (VM) placement in cloud infrastructures, i.e., selecting the best hosts for a given set of VMs. The authors concentrated on test generation and monitoring techniques for comparing a given implementation's placement outcome with an optimal solution for a set of parameters. The authors demonstrated how Integer Linear Programming problems can be formulated and used to generate test suites and optimal solutions for determining the quality of VM placement implementations; the quality is measured as the distance from an optimal placement for a given criterion (or a set of criteria). The strategy proposed is generic, with a focus on resource use, energy usage, and resource over-commitment costs.

López-Pires, Fabio, et al [60] Focused on Virtual Machine Placement (VMP) issues as a viable solution to the challenges listed. The results of an experimental evaluation of 36 VMP optimization algorithms for reducing power consumption are discussed. Algorithms were tested using 400 experimental scenarios and an average objective function cost as the evaluation criteria, with the variance of four distinct dynamic parameters.

Zhao, Yan, et al [61] proposed a heterogeneous and multidimensional clairvoyant dynamic bin-packing model, in which the scheduler would perform more efficient VMP processes using additional information on virtual machine arrival time and length to minimise datacenter size and thereby lower the upfront cost of private clouds. A novel branch-and-bound algorithm with a divide-and-conquer strategy (DCBB) is also proposed to address the derived problem effectively and efficiently. One state-of-the-art VMP method and many classic VMP methods are also updated to conform to the proposed model in order to observe and compare their output with our proposed algorithm.

Tarahomi, Mehran, and Mohammad Izadi [62] To minimise power consumption, an online cloud resource management system with live virtual machine (VM) migration was proposed. A prediction-based and power-aware virtual machine allocation algorithm is proposed to accomplish this. The authors have proposed a three-tier architecture for resource management in cloud data centres that is energy efficient.

Li, Jingmei, et al [63] proposed a dynamic virtual machine scheduling strategy based on an improved genetic algorithm that completely considers the host's CPU and memory usage and obtains the smallest dynamic migration overhead virtual machine scheduling scheme when the above resource utilisation is relatively balanced.

Liaqat, Misbah, et al [64] Virtualization allows many virtual machines (VMs) to share underlying resources on a single physical server, allowing for more efficient resource management. The decision on "what" and "where" to position workloads, on the other hand, has a huge effect on the output of hosted workloads. Current cloud schedulers co-locate workloads using a single resource (RAM), resulting in SLA violations due to non-optimal VM location. To address this issue, the current research has updated the nova scheduler to propose a multi-resource dependent VM placement approach that improves application performance in terms of CPU utilisation and execution time.

Kandoussi, El Mehdi, et al [65] Using a continuous Markov Chain, we modelled migration in a cloud setting. The authors then looked at the possibility of a VM being infected based on the parameters of the destination server.

Ismaeel, Salam, Ali Miri, and Ayman Al-Khazraji [66] proposed a real-time method for optimising the VM consolidation process on the basis of energy consumption In large scale heterogeneous data centres, it introduces a new index called System Condition Index (MCI) of a server to calculate the degree to which physical machine

(PM) is suitable to manage the new (consolidated) VM. This index describes a collection of weighted components used in every cloud data centre, including the network, physical machines, storage, power system, and facilities.

Yang, Chao, et al [67] Cross-VM side-channel attacks in the cloud may result from the co-residency of virtual machines (VMs) from different tenants on the same physical platform. While most existing countermeasures fail to achieve real-time or immediate implementation due to the need for virtualization structure modification, the authors used dynamic migration, a built-in feature of the cloud platform, as a general defence against such threats. To accomplish this, the authors first developed a unified realistic information leakage model that depicts the factors that affect side channels and explains how they impact the damage caused by side-channel attacks. Since migration is used to reduce the length of time spent in co-residency, the authors see this protection as an optimization challenge. They set up an Integer Linear Programming (ILP) to determine the best migration strategy, which is intractable due to the high computational complexity. As a result, the authors used a baseline genetic algorithm to estimate the ILP, which was then improved for optimality and scalability.

Li, Rui, et al [68] The current research on load rebalancing were summarised and analysed. We then propose a Pareto-based Multi-Objective VMreBalance solution (MOVMrB) to the VMs to be reBalanced (VMrB) problem, which aims to simultaneously minimise the disequilibrium of both inter-HM and intra-HM loads. It's one of the first solutions to use inter-HM and intra-HM loads in combination with a multi-objective optimization approach to solve the virtual machine rebalancing problem. We consider migration costs in our work and suggest a hybrid VM live migration algorithm that reduces the I/O complexity of VMrB processing significantly.

Talebian, Hamid, et al [69] presented a detailed analysis of the virtual machine placement problem, as well as a summary of various approaches to solving the problem. The aim of this paper is to highlight current virtual machine placement techniques' challenges and issues. Furthermore, the authors presented a virtual machine placement taxonomy based on a variety of factors including methodology, number of objectives, activity mode, problem objectives, resource demand type, and cloud number. The most up-to-date VM Single-objective and multi-objective placement strategies are categorised, with a range of notable works evaluated in each category. Finally, some unresolved problems and potential developments are addressed, serving as a foundation for future research in this field.

Zhao, Hui, et al [70] proposed a virtual machine (VM) performance maximisation and physical machine (PM) load balancing Virtual Machine Placement system (PLVMP) in the cloud, which aims to optimise VM performance and balance PM workload from the perspectives of both users and cloud providers. First, the authors investigated the relationship between PM workload and VM performance in order to develop a new and improved VM performance model that can more reliably predict VM performance and assist the next VMP. Second, we formulate the VMP as an optimization problem that attempts to optimise VM output for users while also balancing workload among PMs for cloud providers. Third, the authors proposed a greedy-based algorithm for efficiently solving the VMP problem. On the CloudSim platform and a real OpenStack platform, the authors compare PLVMP to other VMP methods.

Yahaya, Rahimatu Hayatu, Faruku Umar Ambursa, and Bashir Galadanci [71] aimed to suggest a new triggering method for a better two-phase optimization scheme for the VMP problem. The new triggering method is based on a Damped Trend Exponential

Smoothing prediction approach. The proposed method tests and determines when the VMPr process should be activated for re-calculation and reconfiguration of the placement.

Kumari, K. Aruna, J. K. R. Sastry, and K. Rajasekhara Rao [72] presented an updated layered architecture that involves the inclusion of a new layer (Optimised Migration Layer) that is responsible for optimising the migration process, with a focus on reducing the time delay caused by the migration. The additional layer employs the principle of containerization to optimise and reduce the time spent waiting for the Migration to complete. In the OML layer, an effective placement algorithm has been introduced. Validation, company, transition, and the deployment process are all components of OML. By making changes to the configuration files, the component that is responsible for the Migration is quarantined. Validation, Market Phase, and Transformation Phase are the four phases that make up OML.

Pyati, Mahantesh, D. G. Narayan, and Shivaraj Kengond [73] proposed a new framework for dynamically consolidating virtual machines in the OpenStack cloud. Overload detection of a host is carried out using the SVM Classification model, which monitors CPU usage, RAM utilisation, and the number of instances of each host over time.

Sutar, Sandeep G., Pallavi J. Mali, and Amruta Y. More [74] The proposed framework aims to optimise the use of computing resources (both physical and virtual) while reducing resource energy consumption. The proposed framework implements a live VM (virtual machine) migration strategy that is both dynamic and energy efficient. This system saves resources by reducing power waste by placing idle physical devices into sleep mode. The authors suggested a framework that would be made up of seven modules.

(1) The energy usage of resources is analysed by the resource controller. (2) The capacity provider distributes the physical machines' maximum and minimum capacity. (3) The task allocator decides which servers are overburdened. (4) The optimizer uses an ant colony optimization algorithm to analyse the load on the physical system. (5) The Local Migration Agent measures the load of the VMs to be migrated and chooses the best physical server for them. (6) Migration Orchestrator migrates the VM when taking into account the load. (7) When a physical system is idle, the Energy Manager switches it to sleep mode (PM).

Braiki, Khaoula, and Habib Youssef [75] The VM reallocation problem was tackled using a multi-objective best-fit-decreasing (BFD) approach. Power costs and resource usage are taken into account in the proposed multi-objective formulation. The authors used fuzzy algebra's expressive power to incorporate both goals into a single function.

Liu, Xialin, et al [76] dynamic consolidation with minimization of migration thrashing (DCMMT) is a proposed solution that prioritises VMs with high availability, dramatically reduces migration thrashing, and the number of migrations to ensure service-level agreement (SLA) by keeping VMs likely to suffer from migration thrashing on the same physical servers rather than migrating.

Tarafdar, Anurina, et al [77] proposed a strategy for VM consolidation that is both energy efficient and QoS aware. The authors used a Markov chain-based prediction method to classify the data center's over- and under-utilized hosts. The authors also suggested an effective VM selection and placement strategy based on a linear weighted sum approach to move VMs from over-utilized to under-utilized hosts while taking energy and QoS into account.

Abdullaha, Muhammad, et al [78] In different sizes of datacenters, different VM placement methods were explored to minimise bandwidth consumption. Five separate VM placement algorithms were introduced and tested by the authors.

Monshizadeh Naeen, Hossein, Esmaeil Zeinali, and Abolfazl Toroghi Haghighat [79] used Markov Chain models to suggest a novel VM consolidation method that can be used to specifically set a desired degree of QoS constraint in a data centre to ensure QoS goals while maximising device utilisation. An energy-efficient and QoS-aware best fit decreasing algorithm for VM placement is proposed for this reason, which takes the QoS objective into account when deciding the position of a migrating VM. To deal with the nonstationary nature of real workload data, this algorithm used an online transition matrix estimator tool. New policies for identifying overloaded and underloaded hosts were also suggested by the authors.

Yang, QiangQiang, et al [80] An energy-aware and load-balancing based dynamic migration strategy (EALB) is proposed to address the existing energy consumption problems of cloud data centres and the running efficiency of virtual machines. This technique chooses the host to migrate, the virtual machine to migrate, and the target host, then completes the virtual machine migration, reducing the number of active servers. The load rate of the hosts was predicted using the grey prediction algorithm in this strategy. After the migration is over, each host's load can be balanced.

Nair, Susmita JA, and TR Gopalakrishnan Nair [81] proposed a novel method for determining the best server for VM placement. The Optimal VM Allocation Framework (OVAF) is a system in which the source hosts ask the destination for their available slots. The utilisation factor is determined based on the response from the available servers, and the appropriate destination for VM placement is chosen.

CHAPTER – 3

CULTURAL SWARM BASED TASK SCHEDULING (CS-TS) TECHNIQUE

3.1 OVERVIEW

Cloud computing provides dynamic resource allocation on demand, a quality that sets it apart by delivering high performance, scalability, cost efficiency, and low maintenance, making it an excellent alternative. Task scheduling becomes a critical aspect in improving performance for dynamic resource allocation, which is critical in the cloud environment to improve performance while lowering costs. Issue scheduling and virtual resource allocation in such a widely distributed dynamic context is a difficult task. The task scheduling algorithm's major consideration is achieving proficiency and providing fairness in task execution. The goal of this research is to look at how to deal with dynamic resource allocation in the cloud. In NP-hard problems, optimization methods are commonly used. This chpater proposes an optimization-based task scheduling technique. User tasks are used in the queue manager in this suggested Cultural Swarm based Task Scheduling (CS-TS) technique. The queue manager sends the user task to the CS-TS technique for resource allocation. The Cultural Algorithm and Particle Swarm Optimization approaches are combined in this proposed CS-TS.

3.2 INTRODUCTION

Cloud computing is evolving as a new computing paradigm that aims to provide end-users with stable, modified, and QoS (Quality of Service)-assured computing dynamic scenarios [82]. Cloud computing is the combination of grid computing, parallel processing, and distributed processing. The fundamental rule of cloud computing is that client data is stored in the web's data centre rather than locally. The cloud computing service providers could monitor and maintain the activity of these data centres. Clients can access their stored data at any time by using the Application Programming Interface (API) provided by cloud providers through any terminal equipment connected to the internet.

Not only are storage services offered, but also software and hardware services to the general public and business markets. Service providers might provide everything from software or platforms to infrastructure and resources. Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) are all terms for similar services [83]. Cloud computing has a number of advantages, the most basic of which are decreased prices, resource re-provisioning, and remote access. Cloud computing saves money by avoiding the company's capital investment in leasing physical infrastructure from a third-party provider. We can quickly obtain extra resources from cloud suppliers when we need to expand our firm because of the adaptive nature of cloud computing. We can access cloud services from anywhere at any time thanks to remote accessibility. To get the most out of the above-mentioned benefits, the services provided in terms of resources should be allocated best to cloud-based apps.

3.3 RELATED WORKS

Xavier, VM Arul, and S. Annadurai [84] Modeling the swarm intelligence of social spiders using chaotic inertia weight based random selection, we focused on lowering overall makespan with appropriate load balancing. The suggested technique avoids local convergence and investigates global intelligent searching in order to locate the best optimised virtual machine for the user job among a set of virtual machines with the shortest makepan and the most balanced resource consumption.

Marahatta, Avinab, et al [85] The energy-aware fault-tolerant dynamic scheduling system (EFDTS) is a dynamic task assignment and scheduling method that uses a fault-tolerant technique to coordinate resource utilisation and energy consumption. A task classification approach is established in the task assignment scheme to partition incoming tasks into several classes and then assign them to the most appropriate virtual machines based on their classifications to reduce mean response time while considering energy consumption. To reduce the job rejection ratio caused by machine failure and delay, replication is employed for fault tolerance. To improve resource usage and energy efficiency, an elastic resource provisioning system is proposed in the context of fault-tolerance. In addition, a migration policy is being designed that can increase resource usage and energy efficiency at the same time.

Tang, Hengliang, et al [86] The dynamic resource allocation approach for cloud-edge environments has been suggested. The resource scheduling algorithm and the resource matching algorithm make up the dynamic resource allocation algorithm. According to the stored penalty of scheduling contents, the value of scheduling contents, and the transmission cost of scheduling contents, a resource scheduling problem can be obtained using the resource scheduling algorithm. After that, the tabu search method is used to find the best resource scheduling option.

Alkhalaileh, Mohammad, et al [87] proposed a dynamic resource allocation model for scheduling data-intensive applications on a hybrid mobile cloud computing system made up of mobile devices, cloudlets, and public cloud (hybrid-MCC). The allocation method is based on a system model that takes into account various application structure, data size, and network configuration criteria.

Nayak, Biswajit, Sanjay Kumar Padhi, and Prasant Kumar Pattnaik [88] When you know what chores are scheduled or when you don't know what tasks are scheduled, mapping is a viable option. If it is known, all that is left is to determine the route so that it can be accurately mapped; otherwise, it must take into account a variety of factors. When tasks are scheduled, the proposed paper focuses on how to choose an algorithm for mapping.

Kaur, Simranjit, et al [89] Workflow scheduling is the process of mapping complex tasks to cloud resources while considering various QoS criteria. As the research of cloud computing continues, it has become more difficult to determine the best scheduling system for the execution of workflows according to user demands. This report also outlined potential research difficulties in the field of workflow scheduling as an optimization issue, with the goal of encouraging more research in this area.

Nayak, Suvendu Chandan, et al. [90] To plan deadline-based jobs, a novel backfilling-based task scheduling algorithm was proposed. Without a decision maker, the proposed method schedules the amount of jobs. To boost scheduling performance, an additional queue and the system's current time are added. In terms of the number of leases scheduled and resource consumption, it performs admirably.

Selvakumar, A., and G. Gunasekaran [91] The fundamental goal of load balancing in cloud computing is to meet the needs of customers by distributing the load evenly among all servers in the cloud to improve resource utilisation, boost throughput, provide good response time, and reduce energy usage. The authors presented a unique load-balancing strategy based on enhanced ant colony optimization to optimise resource allocation and ensure service quality.

Prassanna, J., and Neelanarayanan Venkataraman [92] proposed the Threshold Based Multi-Objective Memetic Optimized Round Robin Scheduling (T-MMORRS) Technique. User queries are sent to the cloud server first. After that, the T-MMORRS Technique uses a burst detector to evaluate whether the workload is normal or bursty. T-MMORRS Technique applies two separate load balancing algorithms for optimally scheduling user tasks based on the burst detector result. In a normal workload environment, the T-MMORRS Technique uses Threshold Multi-Objective Memetic Optimization (TMMO), and in a burstiness workload situation, it uses Weighted Multi-Objective Memetic Optimized Round Robin Scheduling (WMMORRS).

Panda, Sanjaya K., and Prasanta K. Jana [93] To overcome the drawbacks of work consolidation and scheduling, researchers proposed an energy-efficient task scheduling algorithm (ETSA). To make a scheduling choice, the suggested algorithm ETSA considers the completion time and total resource use of a job, as well as a normalisation technique.

3.4 CULTURAL ALGORITHM

Tools for storing, signalling, and transferring knowledge from one generation to the next are constrained by traditional evolutionary calculation procedures. Reynolds introduced the Cultural Algorithm (CA) in 1994 [85]. The CA is a dual inheritance system that describes human cultural evolution at both the macro-evolutionary level, which occurs within the belief space, and the micro-evolutionary level, which occurs within the population space. CA is made up of two parts: a social population and a belief space. The acceptance function selects individuals from the population space to develop problem-solving knowledge that is stored in the belief space. The belief space stores and manipulates the knowledge that individuals in the population space have gained from their experiences. By using the influence function, this knowledge can govern the evolution of the population component. As a result, CA can be used to describe self-adaptation in an EC system by providing an explicit method for global knowledge and an usable framework within which to study it. Evolutionary Programming will be used to implement the cultural algorithm at the population level (EP). The population's gained global knowledge will be described in terms of both normative and situational knowledge. The belief space and the population space are both initialised first in this technique. The algorithm will then repeat the process for each generation until it reaches a termination condition. The performance function is used to assess individuals. The acceptance and influence functions interact between the two layers of the Cultural Algorithm. The acceptance function chooses which members of the current population will have an impact on the belief space. The update function applies the generalised experiences of the selected individuals to alter the present beliefs in the belief space. The new beliefs can then be utilised to direct and affect the next generation's evolutionary process. Three components make up the cultural algorithms discussed above. The first component is the population, which contains the social population to be generated as well as the methods for evaluating, reproducing, and modifying it. Second, there is a belief space, which represents the population's bias gained during the problem-solving process. The communications protocol is the third component, and it is utilised to determine the interaction between the population and their views. Cultural algorithm is a detailed examination of the original evolution theory's superiority based on social (cultural) evolution theory in the social sciences.

CA simulates the long-term evolution of a traditional module in an evolutionary computational system. Culture assists people in adjusting to their surroundings. Culture can be seen in human communities as a generalisation, medium for encoding, and storage of knowledge that is possibly manageable by all members of the society. It's a good idea to keep track of their problem-solving activities. It is often assumed that there is a way to epitomise, gain, and purpose concerning knowledge in cultural material when discussing human civilization. CA is a type of computer model derived from cultural evolutionary approaches' representations. The primary idea is to include knowledge modules into traditional evolutionary computational systems.

The two stages of evolution in the Cultural Algorithm (CA) methodologies are the belief space level and the population space level. An obvious communication protocol connecting the two spaces is composed of an acceptance function and an influence function, which are represented as accept () and influence () respectively. The acceptance function is used to assemble the understanding of specific personalities from the population; an update function, represented here as an update (), can then improve the belief space; and finally, the influence function can brand the practise of problem-solving knowledge in the belief space to monitor the evolution of the population component. Individuals are initially rated in the population space by a performance function obj, much as they are in traditional evolutionary population models (). A generation function is used to create new individuals (). The population is then chosen for the next generation using the selection function select ().

The pseudo-code for the Cultural Algorithm has follows:

Step 1: t=0;

Step 2: Initialize population P^t

Step 3: Initialize belief space B^t

Step 4: Do {

Step 5: Evaluate the performance scores of P^t

Step 6: B^{t+1} = update (B^t, accept (P^t));

Step 7: P^t = generate (P^t, influences (B^{t+1}));

Step 8: t = t+1

Step 9: P^t = select (P^{t-1}, P^t);

Step 10: } while (termination condition).

3.5 PARTICLE SWARM OPTIMIZATION (PSO)

It's a population-based stochastic optimization technique inspired by bird flocking's social behaviour. Eberhart and Kennedy proposed PSO in 1995 [96]. It's a metaheuristic because it can search through a search space with no or few prior assumptions about the problem and find the best solution. The candidate solutions, referred to as particles in the approach, fly around in a multi-dimensional search space, competing and cooperating to discover an optimal or sub-optimal answer. PSO, like GA, starts with a bunch of random particles and then searches for optima by moving candidate solutions around the search space. Each particle is represented by $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, where D is the number of features in the dataset. As a result, each particle has a D-dimensional velocity of $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. Each particle is updated with three values in each iteration: (1) prior velocity, which represents the trend of particle flow over the search space; (2) pbest, which represents the particles' best fitness values up to the current iteration; and (3) gbest, which represents the entire generation's best fitness value up to the current iteration. The following equations are used to update the particle's position and velocity:

$$v_{id}^{k+1} = w * v_{id}^{k} + c1 * r1 * \left(p_{id} - x_{id}^{k}\right) + c2 * r2 * \left(p_{gd} - x_{id}^{t}\right) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \quad (2)$$

The kth iteration is represented by k, and the d^{th} feature in the vector is represented by d. w is the inertia factor, which gives the influence of prior velocity a weight. The acceleration constants are $c1$ and $c2$. $r1$ and $r2$ are two random numbers between 0 and 1. In pbest and gbest, pgd and gid denote the condition of the d^{th} feature.

3.6 PROPOSED CULTURAL SWARM BASED TASK SCHEDULING TECHNIQUE

Cultural Algorithms (CA) is an advanced method evolved from nature's cultural growth method, according to Reynolds [94][95]. It consists of trust and population spaces, as well as a system of communication links between these galaxies, all of which are used to control the prominence of the pooled information and its brand. A CA is an evolving automated classification that is based on knowledge. Its core data is the integration of knowledge tools into predictable sophisticated computational systems. Its replicas are divided into two stages of improvement: belief space and population space. A firm communication technique consists of an acceptance function and an influence function, denoted as Accept () and Influence (), respectively, connects the two space levels.

Particle Swarm Optimization [96] is an energy conservation optimization technique. We chose PSO as the optimization method since it offers superior optimization capabilities and is simple to utilise. PSO was used in the proposed model, which is referred further, with the purpose of making path discovery easier and smarter when discovering the position and energy of VMs on any host.

To initialise the individuals, the suggested CS-TS technique uses the CA algorithm. The tasks and resources are encoded indirectly in chromosomes. Gene value is used to represent resources. The fitness function is supplied by equation 3, which analyses the chromosomes in the population and identifies fitting individuals from it. The fitness function is written like this:

$$Fitness = \min \sum_{ir} \frac{1}{CT_{ir}} + \max \sum_{ir} (AV_{ir} + SC_{ir}) \quad (3)$$

Ranking algorithms are used in selection operations to select the best persons for further operations. Using the following function, the CA solution is translated into the PSO's initial population:

$$T_i^S(0) = \alpha . CA_n$$

α is a constant, and CA_n is a CA solution. PSO's first solution is $T_i^S(0)$. PSO uses the PSO's random initialization process to initialise the particles. PSO compares all tasks to pbest and gbest values, which are updated when the pbest and gbest values are higher. Particles adjust their position and velocity based on the best values at each iteration using the equation:

$$V(t) = \omega v_1(t-1) + k_1 n_1\big(x_1 \Lambda (t-1) - x_1(t-1) + k_2 n_2\big(x^*(t-1)\big) - x_1(t-1)\big) \quad (4)$$

Where $x_1(t)$ is the position vector, I is the particle index, $v_1(t)$ is the velocity vector, $x_1 \Lambda (t-1)$ is the position particle, x(t-1) is the position vector of the best particle, $n_1$and $n_2$are random values between 0 and 1, k_1 *and* k_2 are positive constants, and is the inertia factor.

For crossover, the best particle value is encoded as CA's chromosomes. This chromosome is crossed with a chromosome chosen at random from the selection output of the CA. A new chromosome is created and flipped. To maintain genetic diversity, mutation is given to the offspring by swapping 1s to 0s and 0s to 1s generated during the crossover process.

Step by Step Procedure CS-TS Technique

Step 1: Begin

Step 2: Initialize the parameter setting

Step 2.1: Setting the parameter

Step 2.2: Generate an initial population

Step 3: Calculate the fitness function

Step 3.1: Candidate solution i, i=1,2, 3,…,n

Step 3.2: $fitness_i = f(CA)$

Step 4: Perform selection operation

Step 4.1: for each candidate solution i, i=1,2, 3,…,n

$$p_i = q(1-q)^{rank-1}$$

Step 5: $S_{pso} \leftarrow 0$

Step 5.1: Generating the initial particles

Step 5.1.1: Parameters are obtained from CA method (p_i)

Step 5.1.2: Initialize the PSO particle

Step 5.2: While $S_{pso} \leq p_i$ do

Step 5.2.1: Compute the objective function

Step 5.2.2: If fitness $(CT_k \ \leq pbest_k)$ then $pbest_k \ \leftarrow CT_k$

Step 5.2.3: Endif

Step 5.2.4: If fitness $(CT_k \ \leq gbest_k)$ then $gbest_k \ \leftarrow CT_k$

Step 5.2.5: End

Step 5.2.6: Update the particle velocity and position.

Step 5.2.7: End while

Step 6: Update the knowledge space and belief space with existing candidate solution.

Step 7: For each candidate solution i, $i = 1,2, \dots, A_{CA}$ do if $rand() \leq gbest$ then

Step 8: Perform crossover operation

Step 9: Conduct mutation process

Step 10: End if

Step 11: End for

Step 12: End for

Step 13: Performing Allocation $A_{CA} \ \leftarrow A_{CA} + 1$

Step 14: End

3.7 RESULT AND DISCUSSION

This section illustrates the experimental setup and the results obtained with the proposed Cultural Swarm based Task Scheduling (CS-TS) technique. It is implemented using CloudSim. The proposed CS-TS techniques computes the resource utilization by calculating the execution time, response time, makespan:

- ***Execution time:*** It is the of total time taken for scheduling total cloudlets in VMs.
- ***Response time:*** It is the quantity of time taken between submission of asking and the initial response that's created.
- ***Makespan:*** It is the overall completion time of task on particular Virtual Machine.
- ***Resource Utilization:*** It is one of the most important parameters which have to be measured for the load levelling strategy.

$$Resoure\ utilization = \frac{VM\ Demand}{Range\ of\ Tasks}$$

Table 3.1 depicts the cloud simulator parameters setting used for proposed CS-TS technique.

Table 3.1: Cloud simulator parameter setting for Proposed CS-TS technique

Parameter	Value
Bandwidth	1000
Virtual Machine (VM) Count	10-210
Number of tasks	200-1400
Task Length	1000-15000
Host number	5
Virtual Machine Memory	512–2048
Host Memory (MB)	2048–10,240

Table 3.2 gives the average execution time (in Seconds) of the Proposed CS-TS technique with varying number of tasks. Figure 3.1 depicts the graphical representation

of the average execution time (in Seconds) of the Proposed CS-TS technique with varying number of tasks. From the table 3.2 and figure 3.1, it is shown that the proposed CS-TS technique consuming increasing average execution time for varying number of tasks.

Table 3.2: Average Execution time (in Seconds) of the Proposed CS-TS technique

Number of Tasks	Average Execution time (in Seconds) by Proposed CS-TS technique
200	28
400	96
600	165
800	250
1000	400
1200	605
1400	800

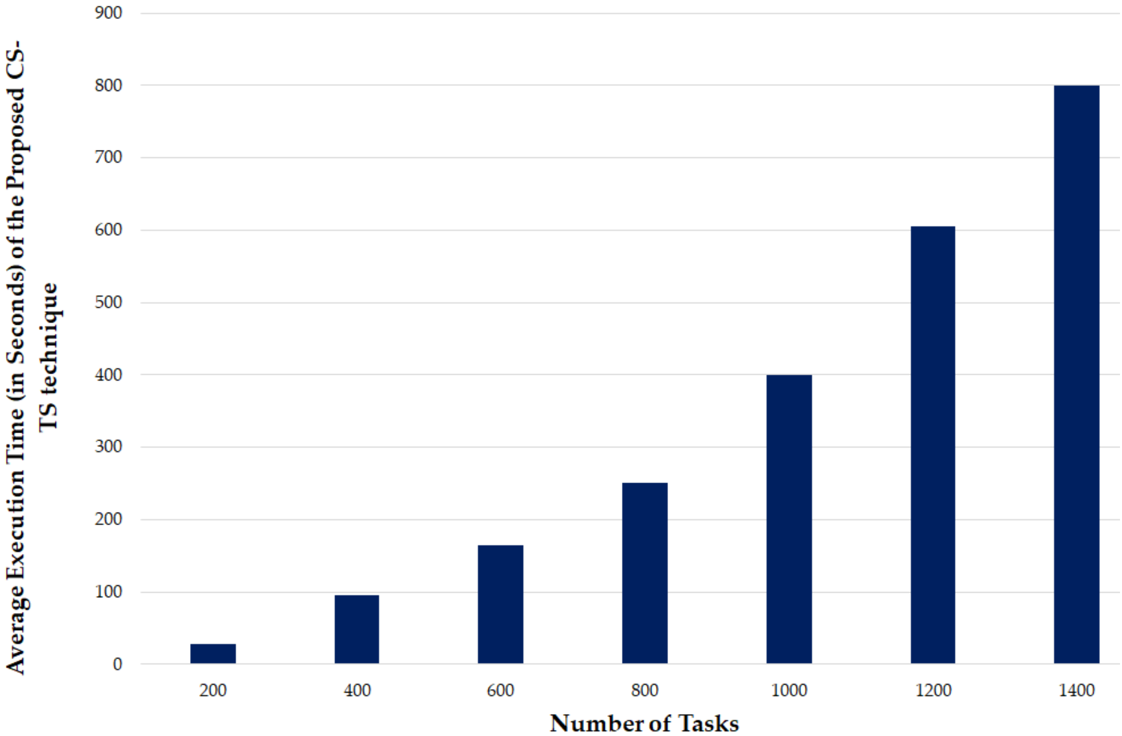

Figure 3.1: Graphical representation of the Average Execution time (in Seconds) of the Proposed CS-TS technique

Table 3.3 gives the Utilization Rate (in %) of the Proposed CS-TS technique with varying number of tasks. Figure 3.2 depicts the graphical representation of the Utilization Rate (in %) of the Proposed CS-TS technique with varying number of tasks. From the table 3.3 and figure 3.2, it is shown that the proposed CS-TS technique increases the utilization rate by increasing the number of tasks.

Table 3.3: Utilization Rate (in %) of the Proposed CS-TS technique

Number of Tasks	Utilization Rate (in %) by Proposed CS-TS technique
200	12
400	28
600	40
800	52
1000	68
1200	80
1400	92

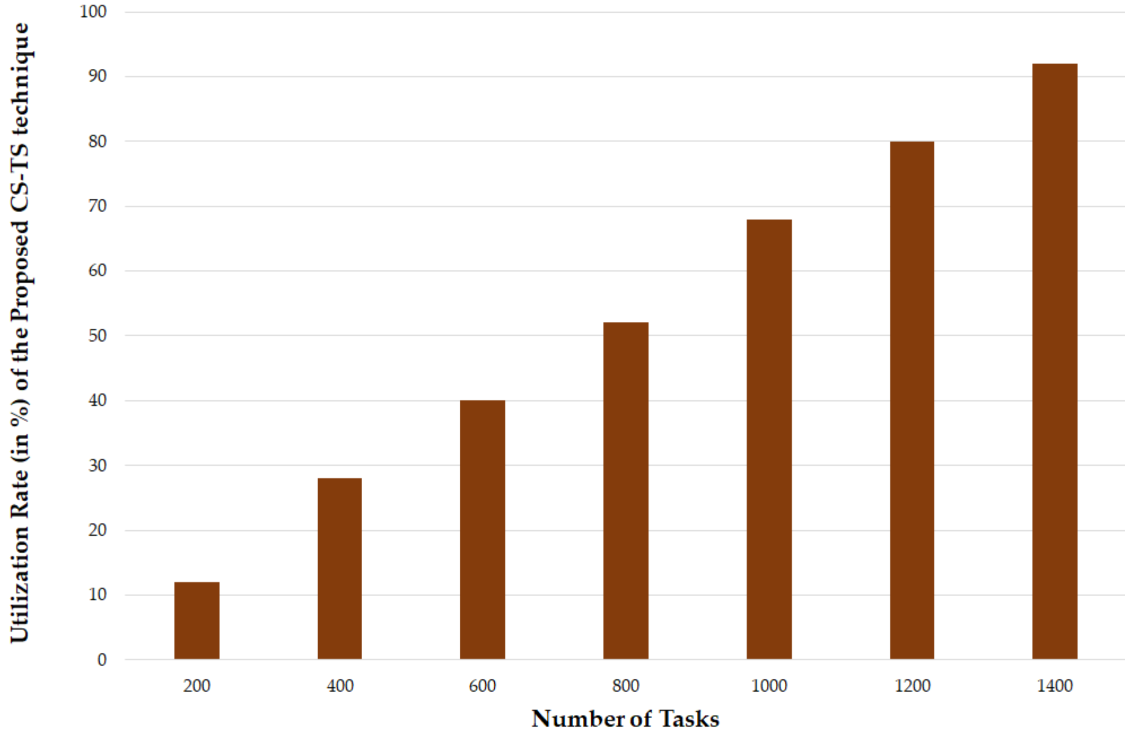

Figure 3.2: Graphical representation of the Utilization Rate (in %) of the Proposed CS-TS technique

Table 3.4 gives the average makespan (in Seconds) of the Proposed CS-TS technique with varying number of tasks. Figure 3.3 depicts the graphical representation of the average makespan (in Seconds) of the Proposed CS-TS technique with varying number of tasks. From the table 3.4 and figure 3.3, it is shown that the proposed CS-TS technique consumes increasing makespan for varying number of tasks.

Table 3.4: Average Makespan (in Seconds) of the Proposed CS-TS technique

Number of Tasks	Average Makespan (in Seconds) by Proposed CS-TS technique
200	0.45
400	0.55
600	0.75
800	1.00
1000	1.05
1200	1.15
1400	1.45

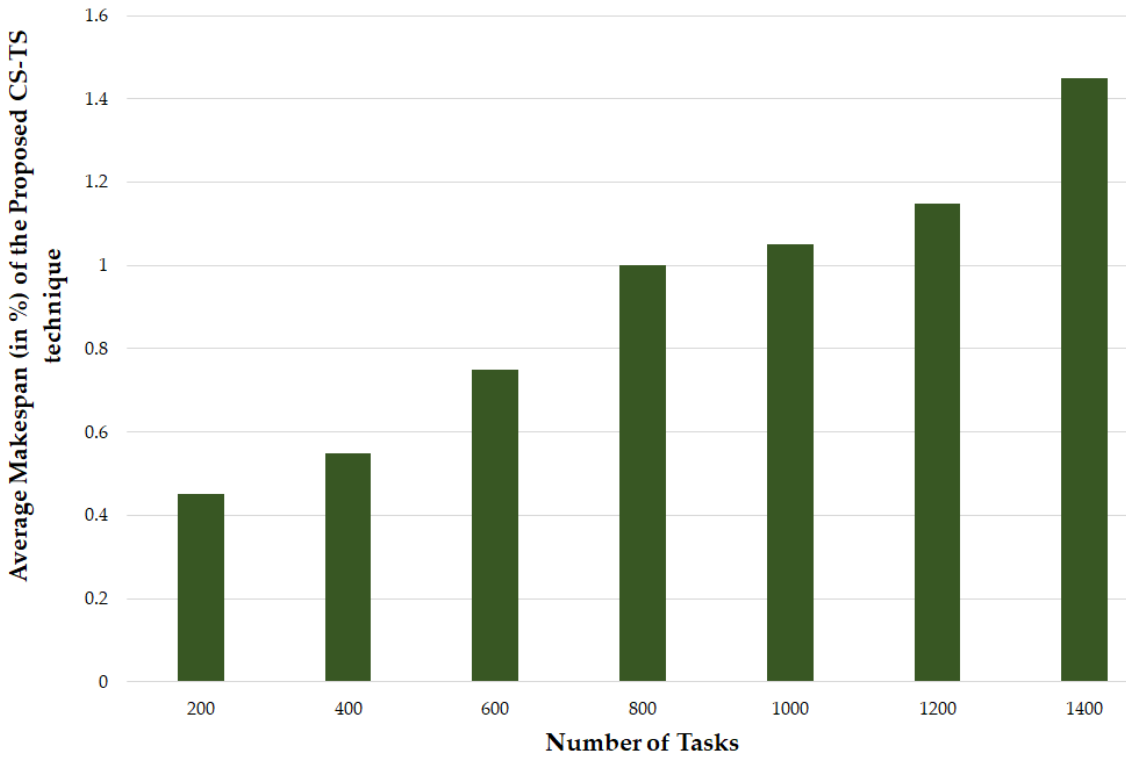

Figure 3.3: Graphical representation of the Average Makespan (in Seconds) of the Proposed CS-TS technique

Table 3.5 gives the average response time (in Seconds) of the Proposed CS-TS technique with varying number of tasks. Figure 3.4 depicts the graphical representation of the average response (in Seconds) of the Proposed CS-TS technique with varying number of tasks. From the table 3.5 and figure 3.4, it is shown that the proposed CS-TS technique consumes increasing the responses time with varying number of tasks.

Table 3.5: Average Response Time (in Seconds) of the Proposed CS-TS technique

Number of Tasks	Average Response Time (in Seconds) by Proposed CS-TS technique
200	30
400	95
600	120
800	250
1000	400
1200	600
1400	785

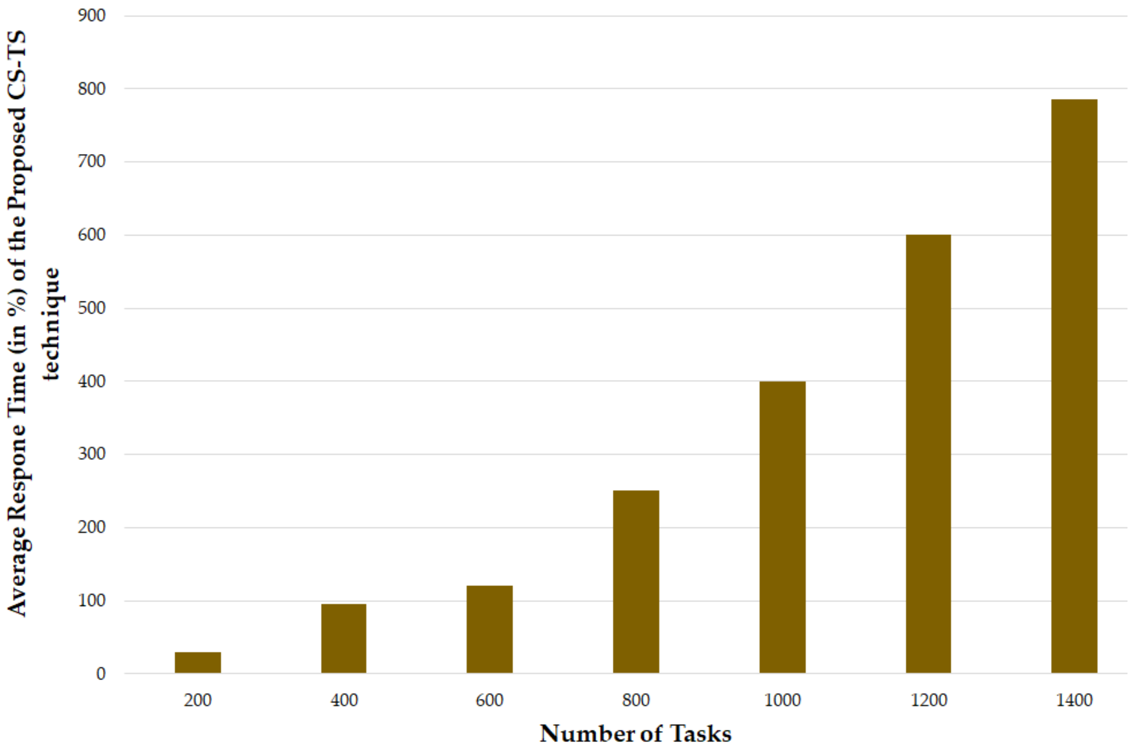

Figure 3.4: Graphical representation of the Average Response Time (in Seconds) of the Proposed CS-TS technique

3.8 SUMMARY

In this chapter, a new task scheduling is proposed using optimization techniques called Cultural Algorithm (CA) and Particle Swarm Optimization (PSO). The average execution time, average response time, average makespan and utilization rate are computed using CS-TS technique with varying number of tasks. From the results obtained, the average execution time, response time, makespan and utilization rate are increasing by increasing the number of tasks.

CHAPTER – 4

OPTIMIZATION BASED HOST DETECTION TECHNIQUE FOR VM CONSOLIDATION

4.1 OVERVIEW

Because of its scalable and dynamic features, cloud computing has gotten a lot of attention from the research community and IT management. Cloud computing allows businesses to outsource their IT infrastructure by providing on-demand access to a shared pool of computing resources. Cloud providers are constructing data centres to meet the ever-increasing demands of cloud consumers. As a result, these cloud data centres use a lot of energy and have the potential to squander a lot of it. Consolidation of Virtual Machines (VMs) aids in resource optimization and, as a result, lowers energy consumption in a cloud data centre. When it comes to VM consolidation, VM placement is crucial. Cloud computing is a new paradigm for provisioning virtual resources on a pay-as-you-go basis. When users' work needs are received, they are mapped to virtual resources operating on datacenter hosts. It is necessary to detect overloaded hosts in order to achieve workload consolidation. Overloaded host detection is carried out for workload balancing, creating a list of overloaded hosts that will be useful when placing VMs (by not putting a VM on an already overloaded host) to avoid Service Level Agreement (SLA) violations, and when checking the underloaded host, the overloaded hosts are omitted to save computational cost. An optimization-based Host Detection methodology is proposed in this research study to analyse the host based on CPU, Bandwidth, and RAM utilizations in order to find overloaded and underloaded hosts.

4.2 INTRODUCTION

Cloud computing is a novel computing model that allows users to rent resources over the Internet on a pay-as-you-go basis. As a result, service providers who do not want to invest in infrastructure could just rent resources from infrastructure providers and pay for what they use. Aside from its enormous economic impact, cloud technology today has a strong chance of becoming a cornerstone of a new generation of sustainable and energy-efficient ICT [97] [98]. Virtualization technology in cloud data centres, on the other hand, plays an important role in reducing energy consumption by allowing for the use of fewer physical servers with much higher per-server utilisation, but it also introduces new management challenges because a large pool of virtual machines must be provisioned and managed [99].

In reality, in a cloud computing context, energy efficient resource management means assigning dynamically physical resources to virtual machines in a way that reduces data centre energy usage while maintaining SLA-based service quality. As a result, researchers in this discipline have been working to develop an efficient and optimal management system that can meet these requirements. A management system that uses dynamic VM consolidation as a dynamic control process to optimise energy efficiency in a cloud data centre [100] [101]. The challenge of dynamic VM consolidation can be divided into four sub-problems in general. (1) determining when a host is overloaded (host overloading detection), in which case live migration is required to migrate one or more VMs from the overloaded host; (2) determining when a host is under-loaded (host under-loading detection), in which case the host is ready to go to sleep mode, in which case all VMs must migrate from it. (3) deciding which VMs must be chosen to migrate from an overburdened host (VM selection) and (4) determining which hosts must be chosen to host migrated VMs (VM placement).

4.3 RELATED WORKS

Mc Donnell, Nicola, Enda Howley, and Jim Duggan [102] Gossip Contracts (GC), a new multi-agent framework for developing decentralised co-operation techniques, was proposed. The Gossip and Contract Net protocols influenced GC. The authors suggested a GC-based Dynamic Virtual Machine Consolidation (DVMC) technique and compared it to two popular strategies: Sercon, which is a centralised method, and eco Cloud, which is a distributed technique, using GC.

Yadav, Rahul, et al [103] suggested adaptive heuristic techniques for overloaded host identification and VM selection from overloaded hosts, including least medial square regression and minimal utilisation prediction. These heuristic techniques reduce CDC energy consumption while using the least amount of SLA. Unlike previous techniques, the proposed VM selection technique takes into account the types of applications running and their CPU use at various time intervals across the VMs.

Ding, Weichao, et al [104] The framework proposed consists of four stages: (1) host overload detection based on residual available computing capacity; (2) selection of appropriate VMs for migration from overloaded hosts based on minimum data transfer; (3) host underload detection based on multi-criteria Z-score approach; and (4) allocating the VMs selected for migration from overloaded and underloaded hosts based on a multi-criteria Z-score approach.

Xiao, Hui, Zhigang Hu, and Keqin Li [105] based on multiple thresholds and an Ant Colony System (ACS), developed a new multi-objective VM consolidation strategy. The suggested methodology uses two CPU usage criteria to determine the host load status; VM consolidation occurs when the host is overburdened or underloaded. During consolidation, the technique uses ACS to choose migrating VMs and destination hosts at the same time, with different selection strategies depending on the host load status.

Aslam, Anjum Mohd, and Mala Kalra [106] proposed an Artificial Neural Network (ANN) based VM selection method. It trains a feed forward neural network to select a VM from an overloaded host using a back propagation learning approach. As a result, it improves the performance of the selection method and optimises the problem of VM selection by learning training dataset.

Hsieh, Sun-Yuan, et al [107] Through host overload detection (UP-POD) and host underload detection (UP-PUD), the proposed VM consolidation methodology evaluates current and future resource use. A Gray-Markov-based model is used to properly anticipate future resource use.

Li, Lianpeng, et al [108] provided a Host Overloading/Underloading Detection algorithm and a novel VM placement method for SLA aware and energy-efficient consolidation of virtual machines in cloud data centres, both based on the suggested Robust Simple Linear Regression prediction model. The proposed approaches, unlike native linear regression, revise the prediction and squint towards over-prediction by including the mistake in the forecast.

Li, Zhihua, et al [109] In order to establish the best mapping between VMs and PMs in the search space, a combined optimization model for VM placement was designed. An enhanced heuristic evolutionary algorithm is used to solve the optimization model, ensuring the globally optimal results, namely the best VM placement scheme. The Energy Efficient- Quality Aware Virtual Machine Consolidation (EQ-VMC) technique presented by the authors combines sub-algorithms for host overloading detection, VM selection, and under-loaded host detection for VM consolidation.

Moghaddam, Seyedhamid Mashhadi, et al [110] Different fine-tuned Machine Learning (ML) prediction models for specific VMs were constructed to anticipate the ideal timing to start host migrations. When selecting VMs to migrate in the second step, lexicographically evaluate migration time and host CPU use. Finally, to pick a destination host for the VMs being migrated, a novel approach based on the Best Fit Decreasing (BFD) algorithm was devised.

4.4 PROPOSED OPTIMIZATION TECHNIQUES BASED HOST DETECTION TECHNIQUE FOR CLOUD COMPUTING

4.4.1 Artificial Bee Colony Optimization

The artificial bee colony algorithm is a population-based optimization method based on bee intelligence, which is divided into two types: foraging and reproduction (mating) behaviour [111]. Artificial bees scour the search space for viable ideas in a fair amount of time. In the subject of work scheduling, foraging behaviour is the most commonly utilised method. The employed bees are responsible for allocating tasks on a resource and sharing their information about food sources with onlooker bees; onlooker bees are responsible for calculating the fitness in order to use it for further management of the entire assignment process; scout bees look for new search sites [112].

All bees go through a ruthless selection process. If the new amount of nectar is more than the old, the bees remember the new position; otherwise, they remember the old one. When the maximum cycle number is reached, the process usually finishes.

This method is inspired by the foraging behavior of honey bees. In ABC model, there are three kinds of honey bee to search food sources, which include scout bees search for food source randomly, employed bees search around the food source and share food information to the onlooker bees, and onlooker bees calculate the fitness and select the best food source. In the nature, bees can extend themselves over long distances in multiple directions. After scout bees find the food source and return to the hive, they compare the quality of food source and go to the dance floor to perform a dance known as "waggle dance". The waggle dance is the communication of bees to shares the information about direction of the food source, distance from the hive, and the nectar amount of the food source. While sharing information, bees evaluate the nectar quality and energy waste. After sharing information on the dance floor, onlooker bees select the best food source and then scout bees will return to the food source to bring nectar back to the hive.

Begin

Step 1: Initialize the population of the scout bees, generate randomly scout bees into the food sources and calculate the fitness values.

Step 2: Repeat

Step 3: Each of the employ bees search around the food sources and update the new fitness, if the new fitness is better than the old values.

Step 4: Select employed bees and recruit onlooker bees to search around the food sources and calculate on their fitness value.

Step 5: Choose the onlooker bees with have the best fitness value.

Step 6: Send scout bees into the food sources to discover new food sources.

Step 7: Until (Stopping criterion is not met)

Step 8: End

4.4.2 Artificial Neural Network

Artificial Neural Networks [113] are distributed and parallel processing systems made up of a large number of simple and highly coupled processors. The MLP design is now the most widely used artificial neural network design. A three-layer multilayer feed forward network is shown in Figure 6.1. Unidirectional branches connect all neurons in a layer to all neurons in subsequent layers, which is a typical property of this type of neural network architecture. That is, the branches and ties can only send data in one way, namely "forward." A learning law can be used to change the weights associated with the branches.

In the training of feed forward neural networks, the back propagation technique is often utilised. When the back propagation algorithm is used to train the network, it creates a non-linear mapping between the input and output variables. As a result, the back-propagation algorithm can modify the network's weights to reflect the non-linear relationship given the input/output pairings. After training, networks with fixed weights will offer the output for the given input.

For network training, the basic back propagation technique is based on minimising an energy function that reflects the instantaneous error. To put it another way, we wish to reduce the size of a function called

$$E(m) = \frac{1}{2}\sum_{q=1}^{q}\left[d_q - y_q\right]^2$$

Where d_q is the ideal network output for the qth input pattern, and y_q is the actual output of the neural network. Each eight is modified in accordance with the following rule:

$$\triangle w_{ij} = -k \frac{dE}{dw_{ij}}$$

w_{ij}denotes the weights of the relation between neurons j and i where k signifies a proportionality constant, E represents the error function, and E denotes the error function. Repeat the weight adjustment technique until the difference between the node output and the actual output is within a reasonable range.

4.4.3 Proposed Host Detection Approach using ANN and ABC optimization technique

Artificial Neural Network is used in this optimization technique-based Host Detection. When it comes to VM consolidation, VM placement is crucial. Cloud computing is a relatively new paradigm in which virtual resources are provisioned on a pay-per-use basis. When a user's work requirement is received, it is mapped onto virtual resources operating on datacenter hosts. It is necessary to detect overloaded hosts in order to achieve workload consolidation. Overloaded host detection is carried out for workload balancing, creating a list of overloaded hosts that will be useful when placing VMs (by not putting a VM on an already overloaded host) to avoid Service Level Agreement (SLA) violations, and when checking the underloaded host, the overloaded hosts are omitted to save computational cost. An optimization-based Host Detection methodology is proposed in this contribution to assess the host based on CPU, Bandwidth, and RAM utilizations in order to find overloaded and underloaded hosts. The Artificial Bee Colony Optimization approach is used to update the weights of the Neural Network for Host classification, and it uses power, CPU, and memory thresholds to prevent underloading and overloading.

When the consumption of the power, CPU, and Memory of the hosts is more than the specified threshold in this suggested Optimization based Host Detection methodology, a neuron in the NN is sent to the overloaded state with regard to potential time. The hosts are underloaded if the use of the Power, CPU, and memory is less than the prescribed threshold. If it's the same, the host is good to go for processing. The Energy Consumed by the Nodes during a specific time is used to compute the thresholds. As a result, overall energy consumption equals the sum of each node's energy consumption multiplied by the total number of nodes.

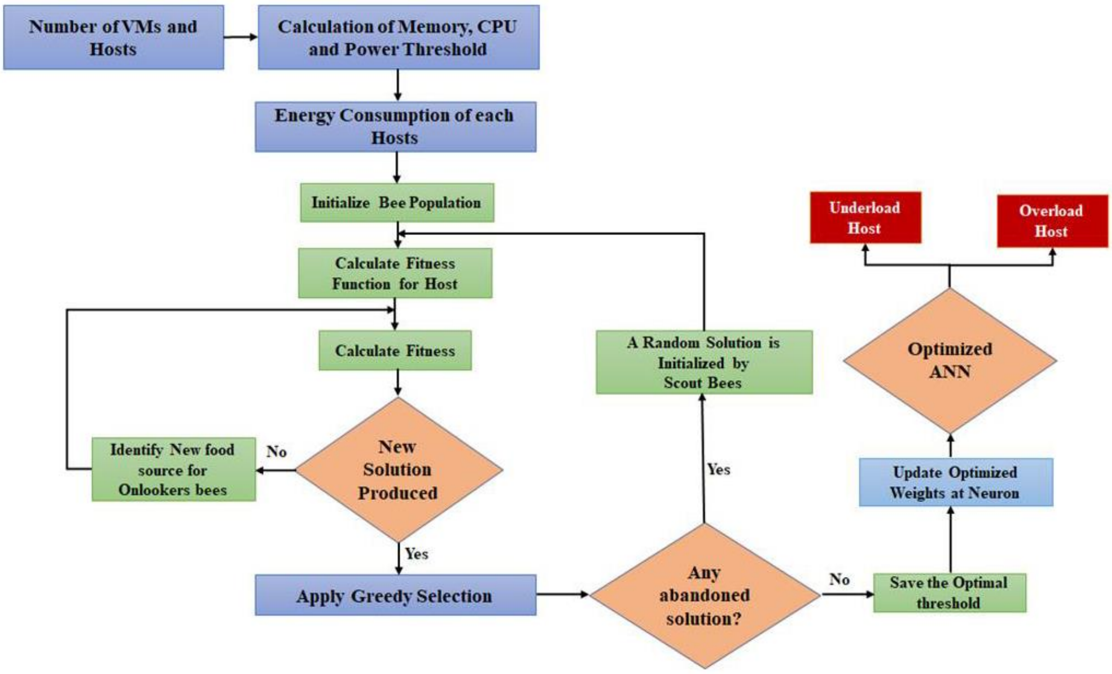

Figure 4.1: Flowchart of Proposed Optimization based Host Detection Technique

Algorithm: Proposed Optimization based Host Detection Technique

Input: Number of VMs and Hosts

Output: Host Classification (Underload or Overload)

Step 1: Calculation of host CPU and memory usage based on the number of VMs on the node and the number of jobs allocated to the VMs.

Step 2: The Hosts' power calculations are based on CPU and memory use. With a minimum and maximum power usage, the power consumption is normalized using the ABC technique.

Step 3: The power consumption of the node during the units of time divided by the energy consumption of the given host.

Step 4: Establishment of an Artificial Bee Colony Optimization for host categorization.

> ***Step 4.1:*** Create a population of solutions that are generated at random by scout bees.
>
> ***Step 4.2:*** Using the default function, assess the current population.
>
> ***Step 4.3:*** Begin training the Neural Network (NN) by formulating each answer using the model's input layer neurons, hidden layer neurons, and output layer neurons.
>
> ***Step 4.4:*** The food position in the total population is randomly distributed during the training phase, as the weight is randomly divided among neurons.
>
> ***Step 4.5:*** Evaluate the population's fitness function.
>
> ***Step 4.6:*** Calculate the new solution using the onlooker bees, and assess the new source's suitability.
>
> ***Step 4.7:*** Begin the ruthless selection process.
>
> ***Step 4.8:*** Calculate the solution's probability value and normalize it into the interval.
>
> ***Step 4.9:*** Onlooker bees use probability to come up with fresh solutions.
>
> ***Step 4.10:*** Determine the new fitness level.
>
> ***Step 4.11:*** Re-use the greedy selection method.

Step 4.12: If the probability of a solution is not approved, the scout bee will forsake the food source. The generator generates a new random value.

Step 4.13: Based on fitness, save the best solution and delete the prior number.

Step 5: Update the NN weights with the ABC's ideal best solution.

Step 6: ANN is used to classify hosts.

4.5 RESULT AND DISCUSSION

4.5.1 Simulation Setup

The data-center used in this work is considered from [21] which is also included in CloudSim. The data-center has 800 hosts from two server models (400 hosts from each server type) and four types of VMs. The CPU capacity of the VM instances is given in millions of instructions per second (MIPS). Table 4.1 gives the two types of host characteristic used in this experiment. Table 4.2 gives the VM types used in this research work. The Energy Consumption (kWh), SLA violation and Classification Accuracy is considered as the performance metrics.

Table 4.1: Two type of host characteristics used in this work

Type	Number of Host	Storage	Number of Cores	RAM	Bandwidth	MIPS
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	1860
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	2660

Table 4.2: Used VM types characteristics in this work

Type of VM	Number of Cores	RAM	MIPS	Storage
VM1	1	613	500	2.5
VM2	1	1740	1000	2.5
VM3	1	1740	2000	2.5
VM4	1	2500	2500	2.5

The computation of the Energy Consumption (kWh) and Service Level Agreement (SLA) violations (in %) with varying number of hosts and number of Virtual Machines (VMs) by the proposed Optimization based Neural Network (O-NN).

4.5.2 Result obtained at Number of Hosts = 400 and Number of VMs = 100

Table 4.3a gives the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100. Figure 4.2a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100. From the table 4.3a, and figure 4.2a, it is shown that the proposed ONN based host detection method consumes less energy than the ABC based Host detection method.

Table 4.3a: Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100

Number of Tasks	Energy Consumption (in kWh) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	7.41	11.02
200	8.32	12.5
300	9.31	13.7
400	9.97	14.3
500	10.12	15.4
600	10.89	16.3
700	11.23	17.5
800	11.82	18.6
900	12.56	19.1

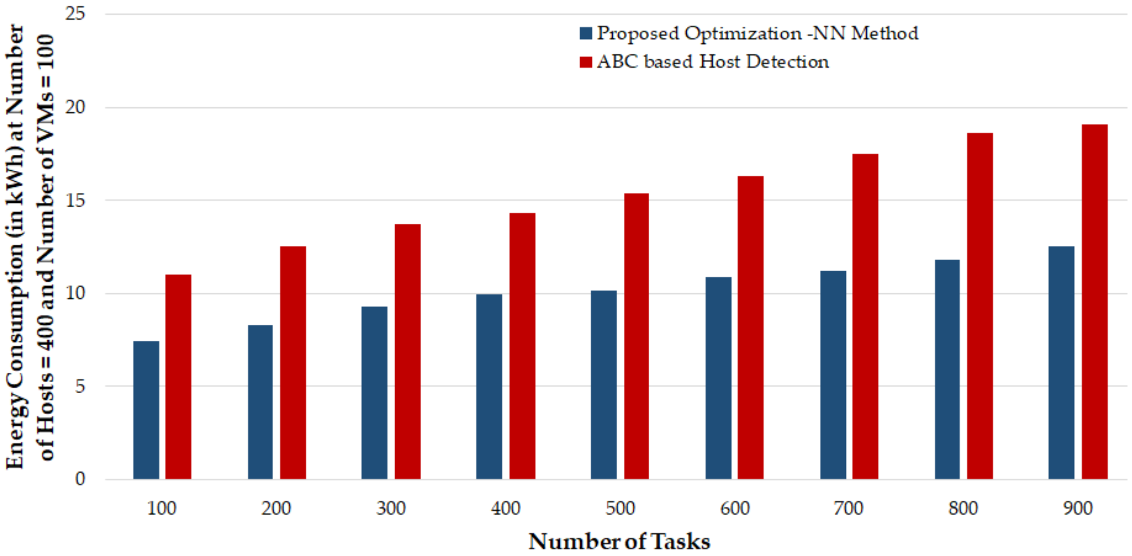

Figure 4.2a: Graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100

Table 4.3b gives the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100. Figure 4.2b depicts the graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100. From the table 4.3b, and figure 4.2b, it is shown that the proposed ONN based host detection method reduces the SLA violation than the ABC based Host detection method.

Table 4.3b: SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100

Number of Tasks	SLA Violation (in %) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	0.045	0.865
200	0.387	1.103
300	0.638	2.232
400	0.812	3.275
500	1.149	4.357
600	1.473	5.531
700	1.985	6.642
800	2.347	7.653
900	2.817	8.4751

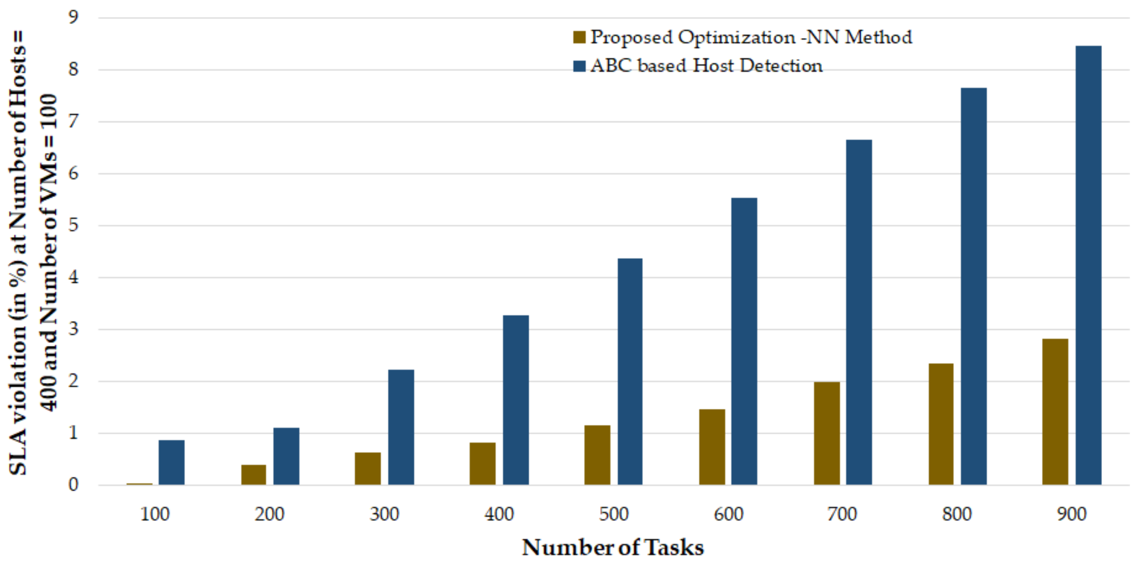

Figure 4.2b: Graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 100

4.5.3 Result obtained at Number of Hosts = 400 and Number of VMs = 200

Table 4.4a gives the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200. Figure 4.3a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200. From the table 4.4a, and figure 4.3a, it is shown that the proposed ONN based host detection method consumes less energy than the ABC based Host detection method.

Table 4.4a: Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200

Number of Tasks	Energy Consumption (in kWh) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	7.96	13.4
200	8.42	14.9
300	9.14	15.5
400	9.85	16.4
500	10.19	17.1
600	10.89	18.3
700	11.27	19.7
800	11.96	20.2
900	12.54	20.9

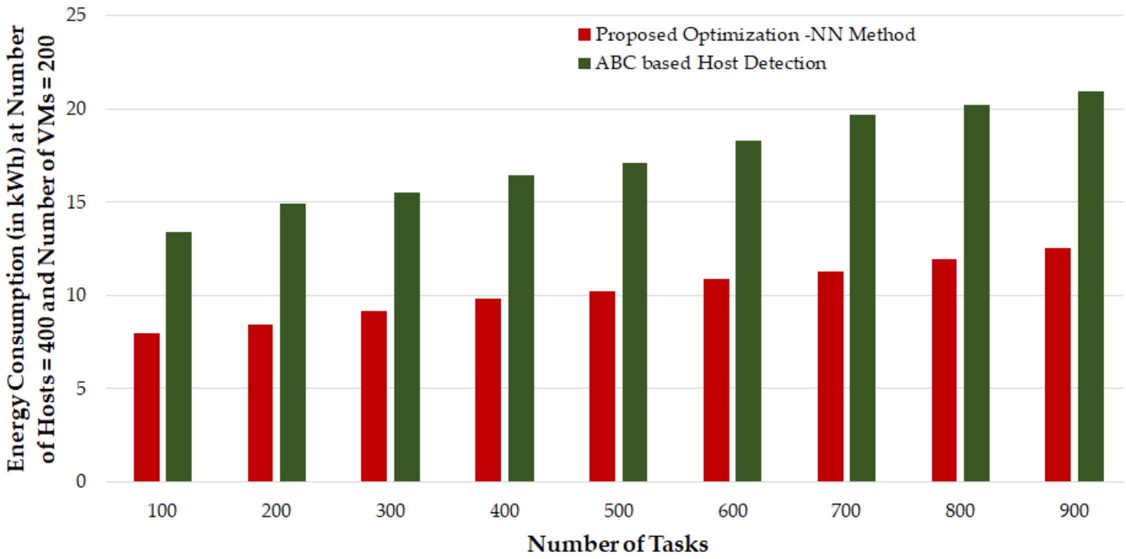

Figure 4.3a: Graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200

Table 4.4b gives the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200. Figure 4.3b depicts the graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200. From the table 4.4b, and figure 4.3b, it is shown that the proposed ONN based host detection method reduces the SLA violation than the ABC based Host detection method.

Table 4.4b: SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200

Number of Tasks	SLA Violation (in %) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	0.452	0.974
200	0.976	1.814
300	1.327	2.411
400	2.123	3.583
500	3.268	4.549
600	4.582	5.743
700	5.874	6.831
800	6.436	7.845
900	6.926	8.687

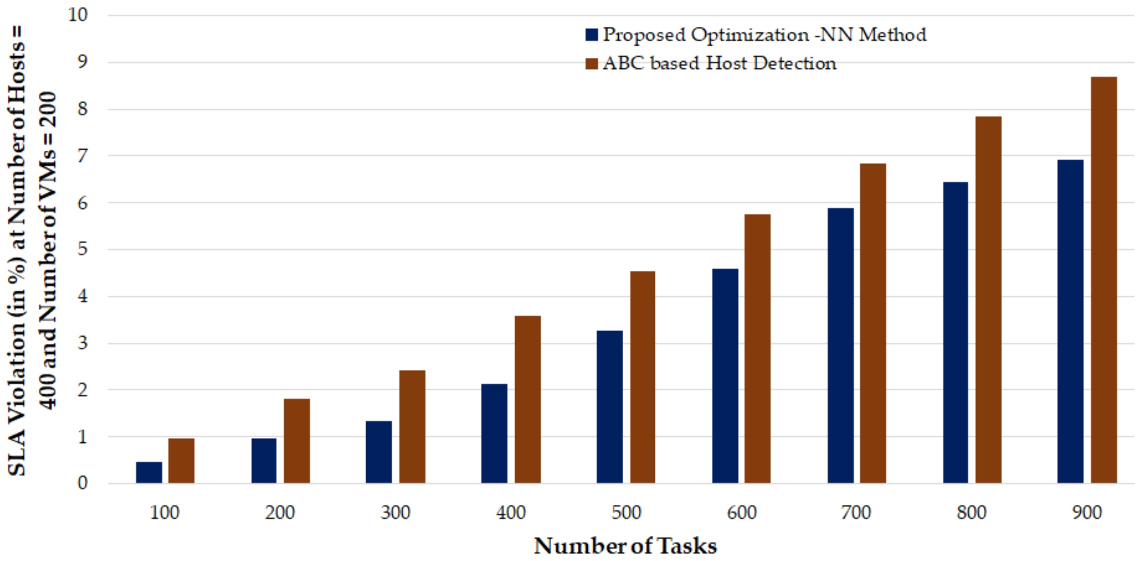

Figure 4.3b: Graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 400 and number of VMs = 200

4.5.4 Result obtained at Number of Hosts = 800 and Number of VMs = 100

Table 4.5a gives the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100. Figure 4.4a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100. From the table 4.5a, and figure 4.4a, it is shown that the proposed ONN based host detection method consumes less energy than the ABC based Host detection method.

Table 4.5a: Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100

Number of Tasks	Energy Consumption (in kWh) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	14.3	16.8
200	15.7	17.2
300	16.2	18.6
400	17.4	19.9
500	18.5	20.4
600	18.9	21.3
700	19.7	22.5
800	20.6	23.7
900	21.2	24.8

\

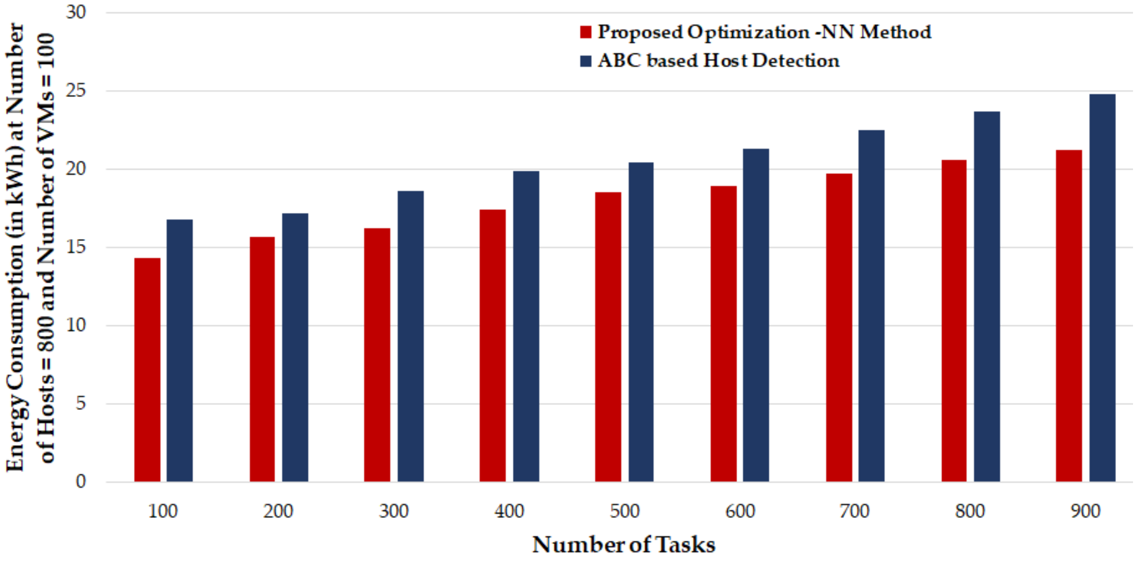

Figure 4.4a: Graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100

Table 4.5b gives the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100. Figure 4.4b depicts the graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100. From the table 4.5b, and figure 4.4b, it is shown that the proposed ONN based host detection method reduces the SLA violation than the ABC based Host detection method.

Table 4.5b: SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100

Number of Tasks	SLA Violation (in %) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	1.371	1.863
200	2.764	2.925
300	3.436	3.532
400	4.315	4.772
500	5.457	5.738
600	6.761	6.931
700	7.663	7.622
800	8.624	8.934
900	9.714	9.876

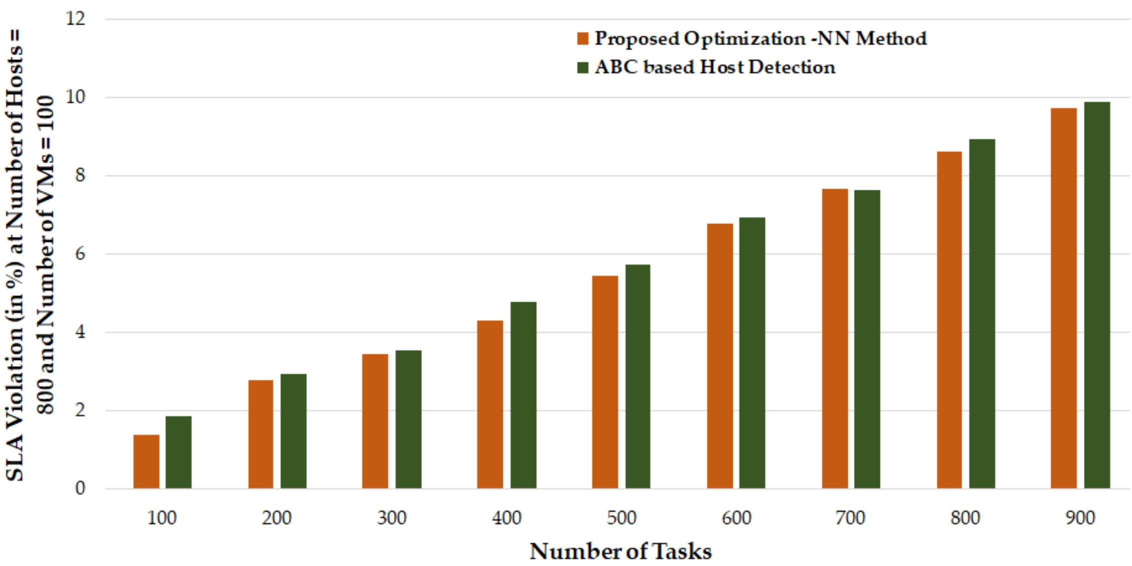

Figure 4.4b: Graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 100

4.5.5 Result obtained at Number of Hosts = 800 and Number of VMs = 200

Table 4.6a gives the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200. Figure 4.5a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200. From the table 4.6a, and figure 4.5a, it is shown that the proposed ONN based host detection method consumes less energy than the ABC based Host detection method.

Table 4.6a: Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200

Number of Tasks	Energy Consumption (in kWh) of the Host Detection Methods	
	Proposed Optimization -NN Method	ABC based Host Detection
100	16.3	19.6
200	17.9	20.3
300	18.5	21.7
400	19.2	22.6
500	20.4	23.5
600	21.6	24.4
700	22.25	25.9
800	22.9	26.7
900	23.6	27.9

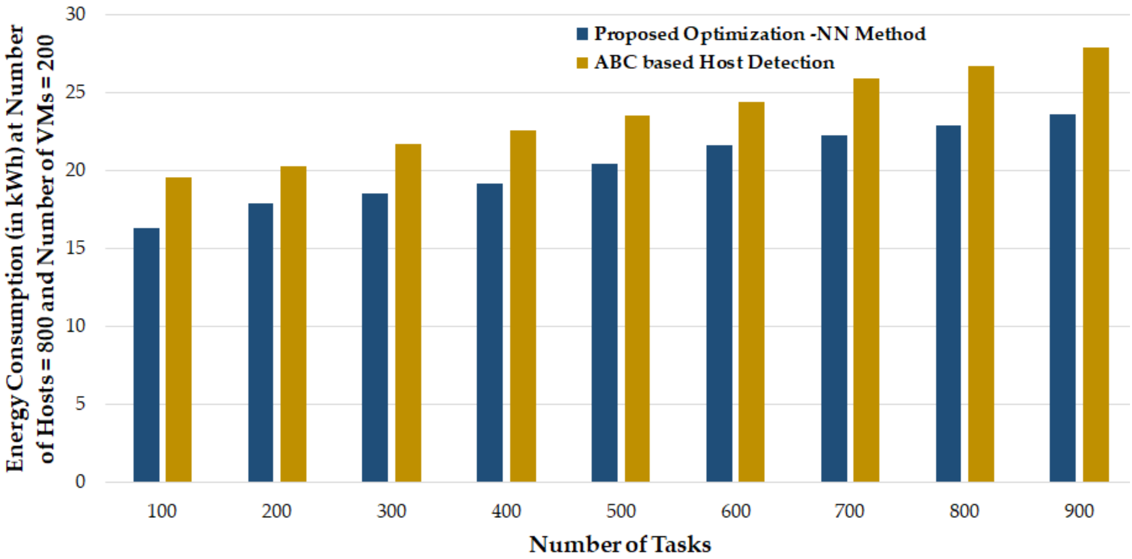

Figure 4.5a: Graphical representation of the Energy Consumption (in kWh) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200

Table 4.6b gives the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200. Figure 4.5b depicts the graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200. From the table 4.6b, and figure 4.5b, it is shown that the proposed ONN based host detection method reduces the SLA violation than the ABC based Host detection method.

Table 4.6b: SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200

Number of Tasks	SLA Violation (in %) of the Proposed Host Detection Method	
	Proposed Optimization -NN Method	ABC based Host Detection
100	1.952	2.641
200	3.542	3.713
300	4.614	4.313
400	5.537	5.551
500	6.679	6.916
600	7.943	7.852
700	8.772	8.813
800	9.916	9.726
900	10.626	10.964

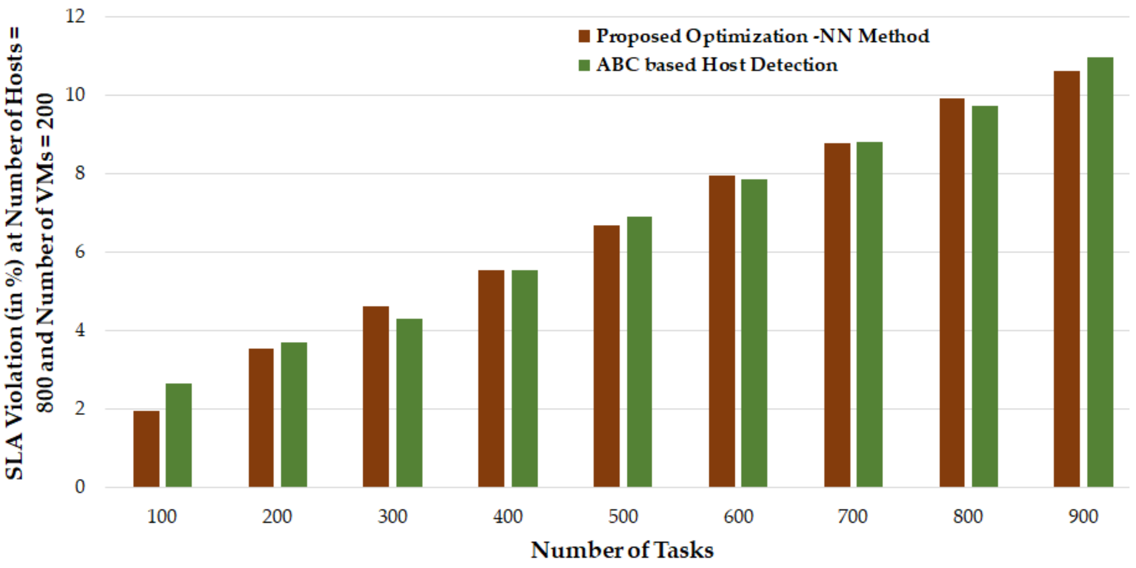

Figure 4.5b: Graphical representation of the SLA violation (in %) of the Proposed Optimization based NN, ABC based Host detection method at number of hosts = 800 and number of VMs = 200

4.6 SUMMARY

In this chapter, the optimization and Neural Network based host detection method is proposed using Artificial Bee Colony Optimization. The performance of the proposed host detection is evaluated using Energy consumption and SLA violation with varying number of tasks, hosts and VMs. From the results, it is shown that the proposed host detection consumes less energy as well reduced the SLA violation than ABC based host detection method.

CHAPTER – 5

AN ADAPTIVE VIRTUAL MACHINE SELECTION AND MIGRATION APPROACH FOR VM CONSOLIDATION

5.1 OVERVIEW

With its tremendous advantages, cloud-based computing often has a detrimental impact on the environment. Release of greenhouse gas and electricity from cloud data centres, which requires significant attention, is the most important issue. A popular approach for energy-efficient and optimum use of resources is virtual machine Consolidation (VM). Consolidation of virtual machine (VM) is one of the main mechanisms for creating a complex cloud resource management system which is energy efficient. The assumption is that migrating VMs into less Physical Machines will accomplish both optimization goals and maximise the usage of cloud server while also reducing Cloud data centre energy consumption. The best fit decrease approach (BFD) for VM placement is used in this proposed strategy. For VM migration an increased Greedy selection is suggested, based on the optimization algorithm Genetic Algorithm.

5.2 INTRODUCTION

Cloud computing provides shared access to services on request without much contact between service providers. Cloud service providers provide their customers with infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [114][115]. These services support all web users directly / indirectly. The development of cloud data centres has a remarkable impact on the IT industry. These data centres, on the other hand, need massive power and high carbon footprints to run.

Infrastructure as a service (IaaS) has in recent years been very common in the field of cloud computing. Popular cloud providers including Rackspace and Amazon EC2 offer such virtual services over the internet to customers in various data centres. Because of the immense demand for cloud computing, there have been thousands of large cloud computing centres, leading to high power consumption [116]. The key cause of this high energy usage is the inefficient use of these cloud resources, not the power consumption of vast amounts of hardware. Therefore, it is very important from an energy management perspective to try to establish adaptive strategies to dynamically change each source in cloud data centres.

The use of virtualization technology [115], improved efficiency of resource use through the sharing of a PM between multiple VMs, is one of the solutions to improve energy efficiency. VMs can be migrated from one machine to another through live migration [117] from the virtual machine to dynamically managed resource, while the services provided by that machine are kept available. This technique allows the playing of VMs with various energy efficiency purposes [118] or load balance [119] to be configured dynamically at all times, depending on resource use.

VM consolidation [120], which aims to transfer VM into a lower number of PMs to maximise the usage of the resources of cloud data centres, is a key mechanism for increasing energy efficiency in the cloud data centres. In some cases, however, too many VMs can be bundled in one PM; because VMs on one PM often have similar physical resources, they can lead to poor quality of service. Thus, consolidation strategies for the VM should ensure comprehensive reliability of the QoS that is frequently specified through service level agreements [121] and seek to change VM 's placement dynamically. In addition, each phase of the VM consolidation should take into account power consumption [122] and VM migration costs [123].

5.3 RELATED WORKS

Ruan, Xiaojun, et al [124] Proposed a 'PPRGear' strategy based on the use-level sampling with distinctive PPRs measured as the number of server-side Java activities performed during a certain time span separated by the average active power consumption over the same period. The strategy is based on the sampling of the user levels. We also present a system for the allocation and relocation of virtual machines that leverages PPR for different host types.

Soltanshahi, Minoo, Reza Asemi, and Nazi Shafiei [125] In cloud data centres, the Krill Herd algorithm, which is the newest collective intelligence algorithm recently implemented, has been proposed to assign virtual machines in physical hosts.

Mc Donnell, Nicola, Enda Howley, and Jim Duggan [126] The new multi-agent system for the development of decentralised collaboration strategies is being put forward by Gossip Contracts (GC). GC is based on the Contract Net Protocols and Gossip's inspiration. Using GC, the authors developed a DVMC strategy based on the GC and compared it to two other common strategies: Sercon and the distributed strategy of ecoCloud. One comparison experiment uses Google cluster-use traces; a dataset from a Google data centre with real-time use data. The GC-based approach works better against SLA infringements and resembles or outperforms other energy consumption strategies.

Zhang, Peiyun, MengChu Zhou, and Xuelei Wang [127] Developed an optimised VM assignment model and developed an improved DD (IDE) approach in the sense of an array of user tasks for solving this optimization problem. The authors have compared the proposed method with various methods, including round-robin (RR), min-min and differential growth.

Hsieh, Sun-Yuan, et al [128] The solution proposed to consolidate the VM takes into account current and potential resource use, by UP-POD and UP-PUD. A Gray-Markov model is used to forecast the potential usage of resources accurately.

Liu, Xialin, et al [129] proposed a Dynamic Consolidation with Minimization of Migrating Thrashing (DCMMT) strategy that priority high capacity VMs, considerably reducing migration thrashing and the number of migrations in order to ensure service-level agreement (SLA) because VMs are likely to be migrated instead of migrating to migration.

Jin, Shunfu, Xiuchen Qie, and Shanshan Hao [130] A two-dimensional Markov chain has been developed to assess mathematically energy saving and response efficiency. Numerical analysis and simulation experiments demonstrate that our proposed strategy to assign VM can efficiently reduce energy consumption and guarantee reaction efficiency.

Saadi, Youssef, and Said El Kafhali [131] proposed an Energy-Efficient Strategy (EES) to consolidate cloud virtual machinery with a goal of reducing energy consumption and fulfilling more high throughput tasks. The performance-to-power ratio is used in our proposal to set the highest threshold for overload detection. Moreover, EES takes into account the total workload utilisation of the data centre to determine lower levels that can reduce the number of virtual machine migrations.

Qie, Xiuchen, Shunfu Jin, and Wuyi Yue [132] Proposed an asynchronous multi-sleep and adaptive work migration solution for energy-efficient virtual machine (VM). The virtual-cluster VMs are split into two modules, Module I and Module II. Module I VMs are always awake, and Module II VMs are autonomous if possible. VMs in module I are always wakeful. Therefore, a queuing model is developed to capture the working theory of the proposed strategy with a partial, asynchronous multiple vacation. Using the matrix-geometric solution approach, performance measurements are mathematically derived for the average response time and energy saving rate.

5.4 GENETIC ALGORITHM

Genetic algorithms [133] employ metaphor from biology and genetics to iteratively evolve a population of initial individuals to a population of high-quality individuals, where each individual represents a solution of the problem to be solved and is compose of a fixed number of genes. The number of possible values of each gene is called the cardinality of the gene. Each individual is called as chromosome. The set of chromosomes forms population.

Functioning of genetic algorithm starts with randomly generated population of individuals. By various generations these population evolved and individuals' quality gets improved. In every generation, three basic operators of genetic algorithm i.e. selection, crossover and mutation are applied to each individual. Crossover means exchanging the genes between two chromosomes while mutation means random changing of a value of a randomly chosen gene of a chromosome. These individuals are representation of the problem required to be solved. Different positions of each individual can be encoded as bits, characters and numbers [134].

Here, the numbers of best-fit individuals are selected. For this user defined fitness function is used. Fitness function is used to measure quality of each chromosome. Remaining individuals are paired and by process of crossover, new offspring is produced by partially exchanging their genes. When genetic algorithm is used for problem solving, three factors will have impact on the effectiveness of the algorithm, they are [133]:

a. The selection of fitness function
b. The representation of individuals and
c. The values of the genetic parameters.

5.5 OPTIMIZATION BASED VIRTUAL MACHINE CONSOLIDATION

Dynamic consolidation of Virtual Machines (VMs), using live migration of the VMs and switching idle servers to sleep mode or shutdown, optimizes the energy consumption. The normal host detected through host detection technique is considered for allocating the VM, VM migration for VM consolidation framework. In this contribution, an adaptive VMs migration selecting method and heuristic algorithm for dynamic consolidation of VMs to be proposed based on the analysis of the historical data.

This proposed Adaptive VM consolidation strategy includes the coding, fitness function, selection, crossover, mutation and greedy selection. The VM placement is carried out with a bin packing problem with variable bin sizes, where the bin sizes are represent by the physical nodes; items are the VMs that have to be allocated; CPU capabilities, and power consumption by the nodes. For VM placement, Best Fit Decreasing (BFD) method is utilized in this proposed strategy. For VM migration, an enhanced Optimization algorithm called Greedy Selection based Genetic algorithm is proposed. The following procedures gives the VM placement strategy using Best Fit Decreasing (BFD) algorithm. The host list for the VM placement is taken from the host detection method.

Input: Host List (hl) and Virtual Machine List (vml)
Output: Allocation of Virtual Machines (VMs)
Step 1: for each vm in vml do
    Step 1.1: $minPower \leftarrow Max$
    Step 1.2: $allocatedHost \leftarrow NULL$
    Step 1.3: foreach host in hl do
        Step 1.3.1: if host has enough resource for vm then
        Step 1.3.2: $power \leftarrow estimatePower(host, vm)$
        Step 1.3.3: if $power < minPower$ then
            $allocatedHost \leftarrow host$
            $minPower \leftarrow power$
    Step 1.4: if $allocatedHost \neq NULL$ then
        Step 1.4.1: allocate vm to allocateHost
Step 2: return allocation

In this proposed Adaptive approach, an enhanced Genetic algorithm is proposed for VM Migration. In this proposed Adaptive VM consolidation approach, a Greedy Search algorithm is used to find the optimized VM with the help of host detection techniques. In this proposed VM consolidation strategy, a Greedy Selection algorithm is utilized in the children coding after each mutation and crossover operator. In the proposed VM migration approach, the dynamic threshold methodology is carried out. The upper threshold for the VM migration is obtained by using proposed. For this approach, the host list and VM selected list is given as the input.

- Randomly generate several codes to form a population.
- Using binary code to coding all tasks.

- Calculate the fitness value of all coding in the population.
- Select several pairs of codes to do the crossover operation to form new population (the total number of population remains unchanged);
- Perform mutation operation through selection operator again.
- Make greedy choices for all codes of the newly formed population. Form a whole generation of new population.
- Repeat the process until a stable optimal individual fitness is found, that is, the optimal threshold.

Input: Host list (hl), Virtual Machine List (vml)

Output: VM migration list

Step 1: vml.sortDecreasingUtilization()

Step 2: for each h in hl do

Step 2.1: $hutil \leftarrow h.util()$

Step 2.1.1: *Initialize the population for GA.*

Step 2.1.2: *Binary coding of the tasks assigned in the VMs are carried in this step.*

Step 2.1.3: *The fitness value of each individual is computed.*

Step 2.1.4: *Roulette Wheel method is used for the selection of the individual based on the probability, which is reflected by the fitness value of the population.*

Step 2.1.5: *Two-point crossover is calculated with the two individuals.*

Step 2.1.6: *Mutation operator is applied to improve the local search capability of GA algorithm and to maintain the diversity of the population.*

Step 2.1.7: *The greedy selection in this study occurred in the children coding after each mutation and crossover operator.*

Step 2.2: $bestFitUtil \leftarrow MAX$

Step 2.3: while $hutil > h.Th_{upper}()$ do

Step 2.3.1: foreach vm in vml do

If $vm.util() > hutil - h.Th_{upper}()$ then

$t \leftarrow vm.util() - hutil + h.Th_{upper}()$

If $t < bestFitUtil$ then

$bestFitUtil \leftarrow t$

$bestFitVm \leftarrow vm$

Else

If $bestFitUtil = MAX$ then

$bestFitVm \leftarrow vm$

Break

Step 2.3.2: $hutil \leftarrow hutil - bestFitVm.util()$

Step 2.3.3: migration.List.add(h.getvml())

Step 2.3.4: vmList.remove(h.getvml())

Step 3.1: if $hutil < Th_{low}()$ then

Step 3.1.1: migrationList.add(h.getvml())

Step 3.1.2: vml.remove(h.getvml())

Step 4: return migrationlist

5.6 RESULT AND DISCUSSION

5.6.1 Simulation Setup

In the research work, the data-center used in this work is considered which is also included in CloudSim. The data-center has 800 hosts from two server models (400 hosts from each server type) and four types of VMs. The CPU capacity of the VM instances is given in millions of instructions per second (MIPS). The Table 4.1 and Table 4.2 given in the Chapter – 4. The metrics used in the chapter 4 is considered in this contribution. Energy Consumption (in kWh) and SLA Violation (in %) with varying number of tasks, VMs and Hosts. The performance of the Proposed VM consolidation strategy and Proposed O-NN based host detection method. This proposed VM consolidation strategy will take the optimal host for the VM consolidation.

5.6.2 Result obtained at Number of Hosts = 400 and Number of VMs = 100

Table 5.1a gives the Energy Consumption (in kWh) of the Proposed VM consolidation strategy and Proposed Optimization-Neural Network based Host detection method at number of hosts = 400 and number of VMs = 100. Figure 5.1a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed VM consolidation strategy and Proposed O-NN method at number of hosts = 400 and number of VMs = 100. From the table 5.1a, and figure 5.2a, it is shown that the proposed VM consolidation strategy consumes less energy than the proposed O-NN based Host detection method.

Table 5.1a: Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100

Number of Tasks	Energy Consumption (in kWh)	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	3.94	7.41
200	4.48	8.32
300	4.96	9.31
400	5.51	9.97
500	5.95	10.12
600	6.38	10.89
700	6.99	11.23
800	7.28	11.82
900	7.84	12.56

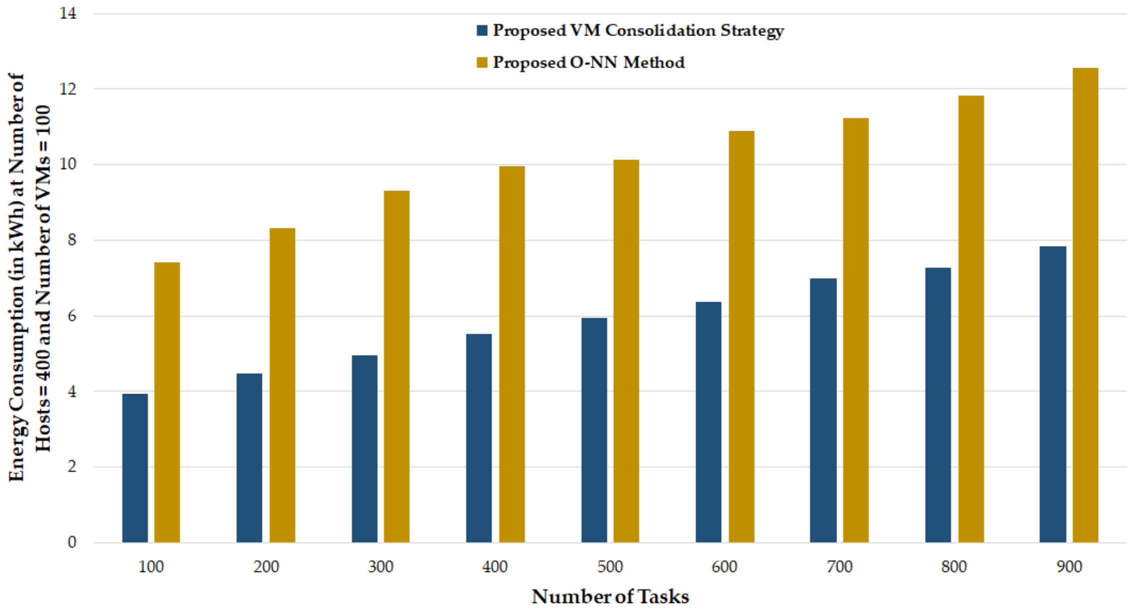

Figure 5.1a: Graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100

Table 5.1b gives the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100. Figure 5.1b depicts the graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100. From the table 5.1b, and figure 5.1b, it is shown that the proposed VM consolidation Strategy reduces SLA violation than the proposed O-NN based Host detection method.

Table 5.1b: SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100

Number of Tasks	SLA Violation (in %)	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	0.002	0.045
200	0.165	0.387
300	0.316	0.638
400	0.498	0.812
500	0.501	1.149
600	0.563	1.473
700	0.599	1.985
800	0.621	2.347
900	0.659	2.817

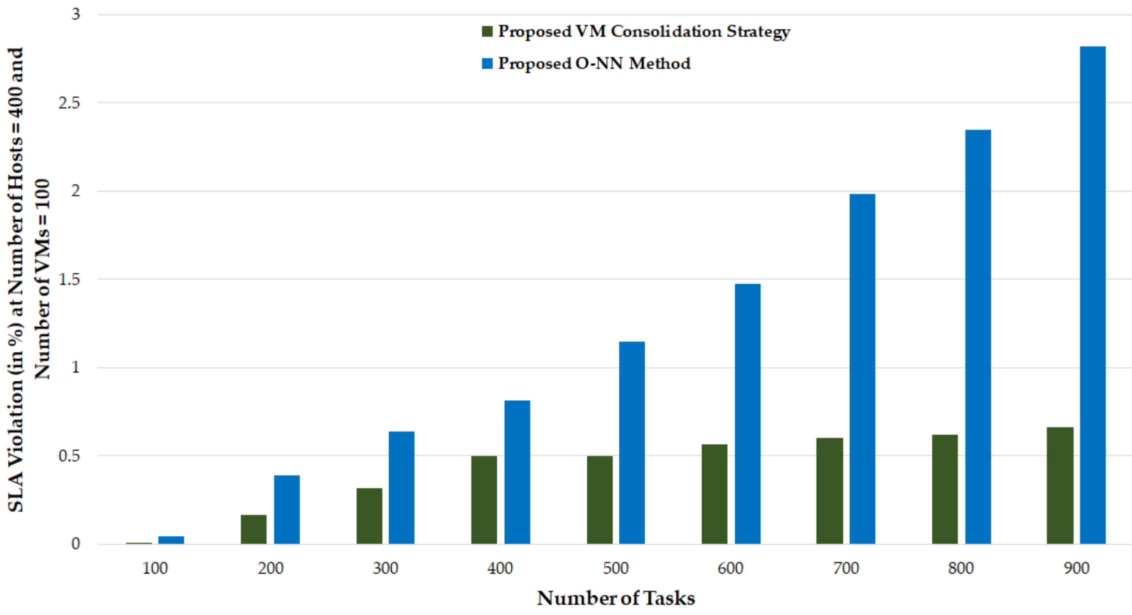

Figure 5.1b: Graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 100

5.6.3 Result obtained at Number of Hosts = 400 and Number of VMs = 200

Table 5.2a gives the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200. Figure 5.2a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200. From the table 5.2a, and figure 5.2a, it is shown that the proposed VM consolidation strategy consumes less energy than the proposed O-NN based Host detection method.

Table 5.2a: Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200

Number of Tasks	Energy Consumption (in kWh)	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	4.54	7.96
200	4.98	8.42
300	5.46	9.14
400	5.93	9.85
500	6.37	10.19
600	6.88	10.89
700	7.43	11.27
800	7.96	11.96
900	8.54	12.54

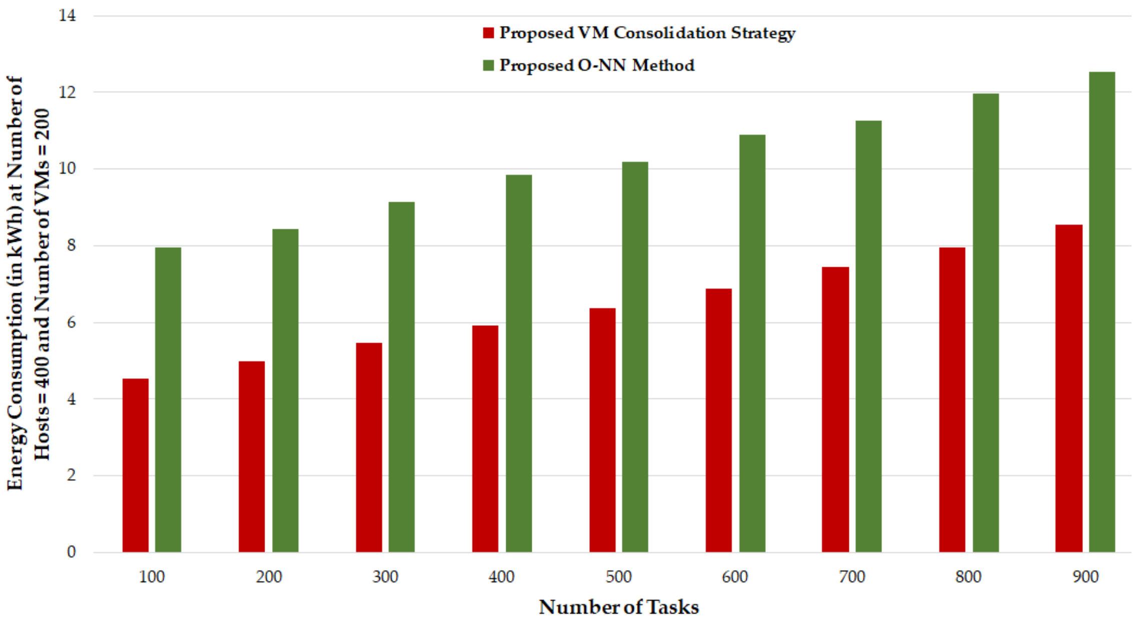

Figure 5.2a: Graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200

Table 5.2b gives the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200. Figure 5.2b depicts the graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200. From the table 5.2b, and figure 5.2b, it is shown that the proposed VM consolidation Strategy reduces SLA violation than the proposed O-NN based Host detection method.

Table 5.2b: SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200

Number of Tasks	SLA Violation (in %)	
	Proposed VM Consolidation Strategy	**Proposed Optimization -NN Method**
100	0.045	0.452
200	0.192	0.976
300	0.424	1.327
400	0.506	2.123
500	0.613	3.268
600	0.741	4.582
700	0.871	5.874
800	0.913	6.436
900	1.148	6.926

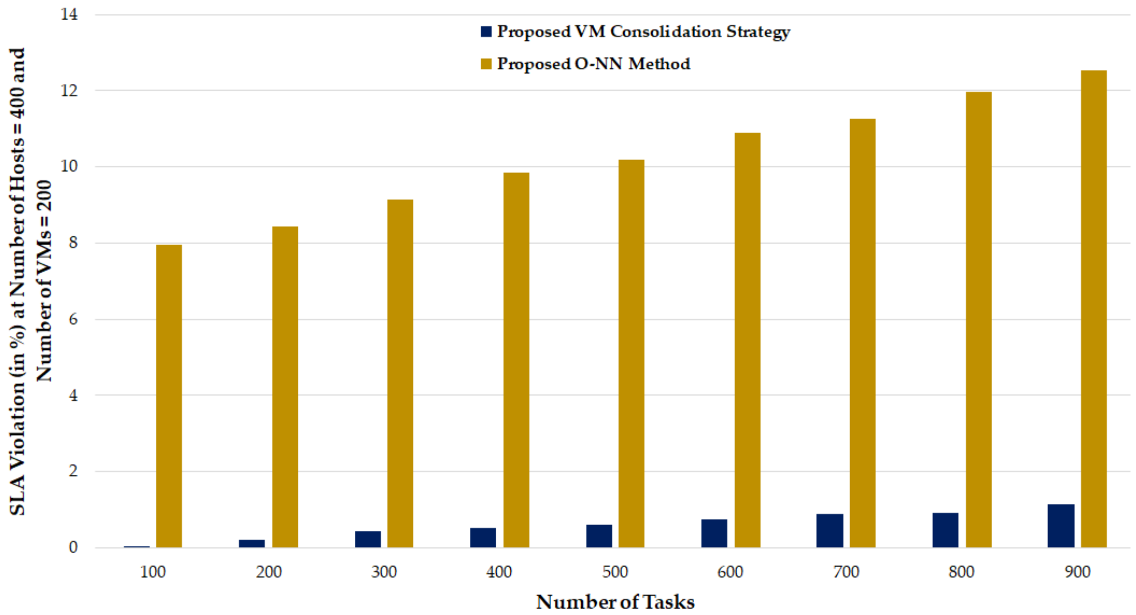

Figure 5.2b: Graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 400 and number of VMs = 200

5.6.4 Result obtained at Number of Hosts = 800 and Number of VMs = 100

Table 5.3a gives the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100. Figure 5.3a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100. From the table 5.3a, and figure 5.3a, it is shown that the proposed VM consolidation strategy consumes less energy than the proposed O-NN based Host detection method.

Table 5.3a: Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100

Number of Tasks	Energy Consumption (in kWh)	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	6.36	14.3
200	7.16	15.7
300	8.24	16.2
400	9.71	17.4
500	10.19	18.5
600	11.60	18.9
700	12.41	19.7
800	13.84	20.6
900	14.32	21.2

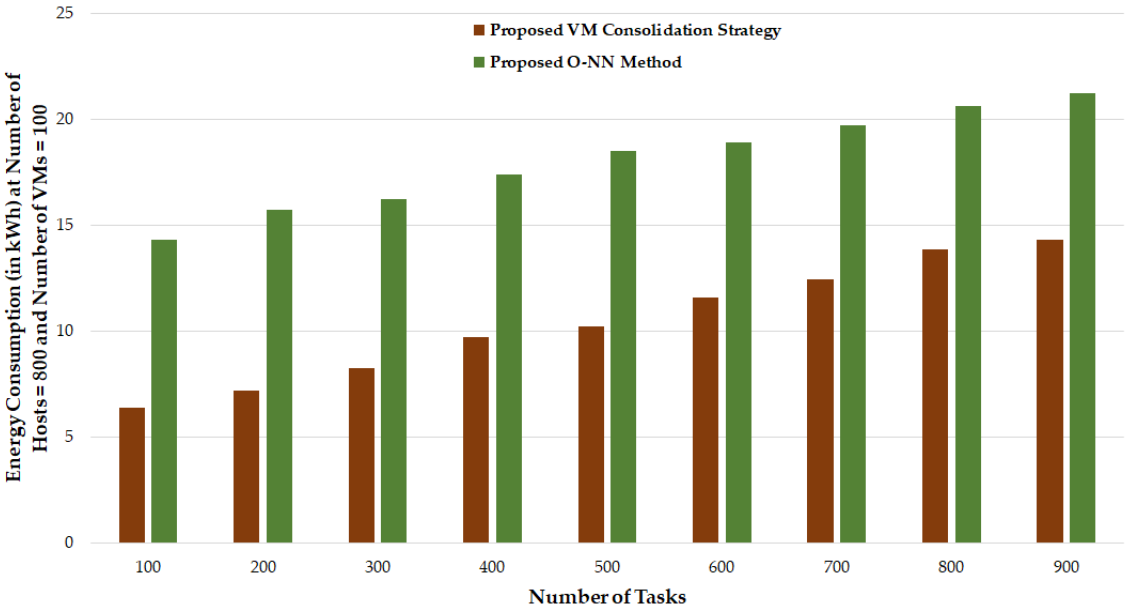

Figure 5.3a: Graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100

Table 5.3b gives the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100. Figure 5.3b depicts the graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100. From the table 5.3b, and figure 5.3b, it is shown that the proposed VM consolidation Strategy reduces SLA violation than the proposed O-NN based Host detection method.

Table 5.3b: SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100

Number of Tasks	SLA Violation (in %)	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	0.452	1.371
200	0.631	2.764
300	0.827	3.436
400	1.078	4.315
500	1.392	5.457
600	1.796	6.761
700	2.143	7.663
800	2.635	8.624
900	2.968	9.714

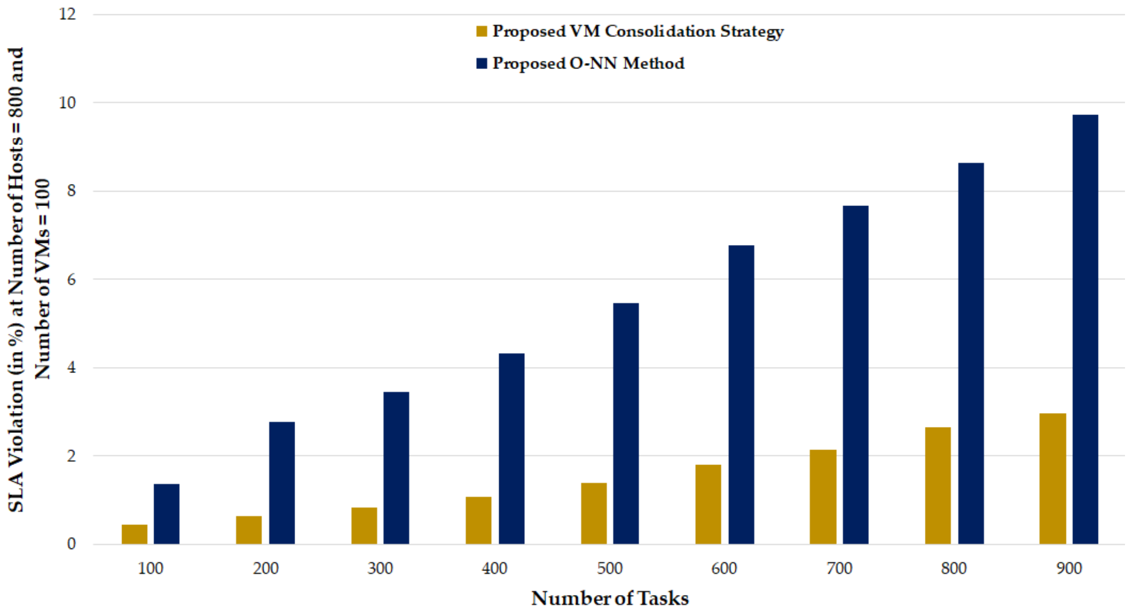

Figure 5.3b: Graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 100

5.6.5 Result obtained at Number of Hosts = 800 and Number of VMs = 200

Table 5.4a gives the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200. Figure 5.4a depicts the graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200. From the table 5.4a, and figure 5.4a, it is shown that the proposed VM consolidation strategy consumes less energy than the proposed O-NN based Host detection method.

Table 5.4a: Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200

Number of Tasks	Energy Consumption (in kWh) of the Host Detection Methods	
	Proposed VM Consolidation Strategy	Proposed Optimization -NN Method
100	7.44	16.3
200	8.24	17.9
300	9.12	18.5
400	10.52	19.2
500	11.27	20.4
600	12.42	21.6
700	13.22	22.25
800	14.62	22.9
900	15.14	23.6

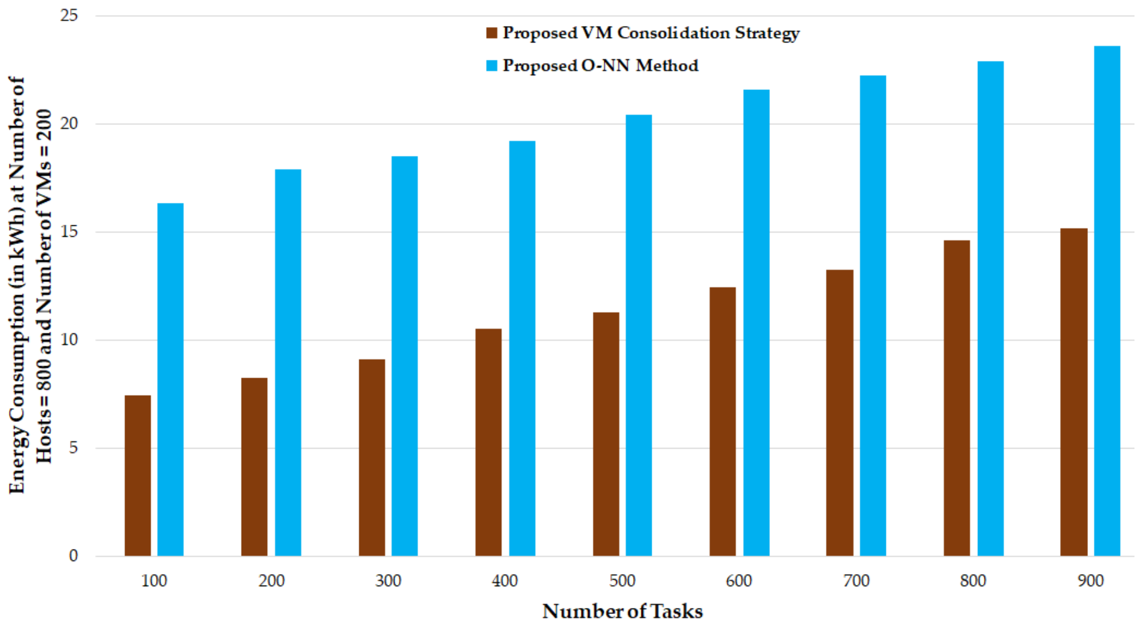

Figure 5.4a: Graphical representation of the Energy Consumption (in kWh) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200

Table 5.4b gives the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200. Figure 5.4b depicts the graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200. From the table 5.4b, and figure 5.4b, it is shown that the proposed VM consolidation strategy reduces SLA violation than the proposed O-NN based Host detection method.

Table 5.4b: SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200

Number of Tasks	SLA Violation (in %)	
	Proposed VM Consolidation Strategy	**Proposed Optimization -NN Method**
100	0.671	1.952
200	0.813	3.542
300	1.245	4.614
400	1.866	5.537
500	2.404	6.679
600	2.914	7.943
700	3.331	8.772
800	3.813	9.916
900	4.146	10.626

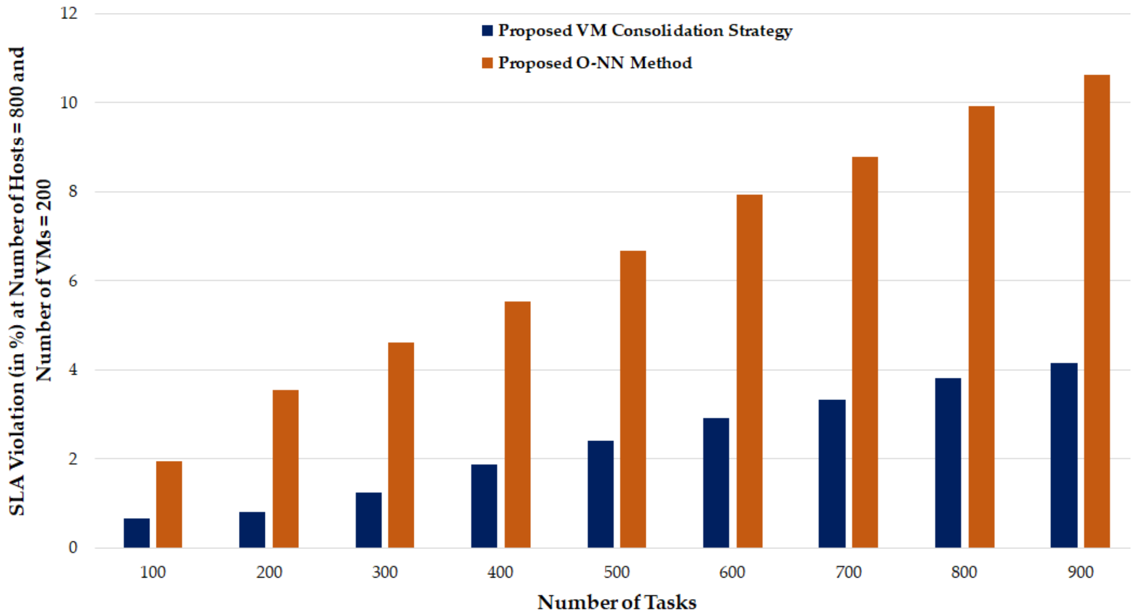

Figure 5.4b: Graphical representation of the SLA violation (in %) of the Proposed VM Consolidation Strategy and Proposed O-NN based Host Detection Method at number of hosts = 800 and number of VMs = 200

5.7 SUMMARY

In this chapter, VM consolidation strategy is proposed with Proposed Optimization -NN based host detection method. The placement of VM is done using Best Fit Decreasing (BFD) approach and an enhanced Genetic algorithm is proposed for VM Migration. The combination of these three approaches is named as adaptive VM consolidation strategy. The performance of the proposed VM consolidation strategy is evaluation with proposed O-NN based host detection using Energy consumption and SLA violation. From the results, it is shown that the Proposed VM consolidation strategy consumes less energy as well as it also reduces the SLA violation.

CHAPTER – 6

PERFORMANCE ANALYSIS

6.1 OVERVIEW

In this chapter, the performance of the Proposed Cultural Swarm based Task Scheduling method, Proposed Optimization – Neural Network based Host Detection method and Proposed VM Consolidation strategy is evaluated with their existing techniques for various metrics.

6.2 PERFORMANCE ANALYSIS OF THE PROPOSED CULTURAL SWARM BASED TASK SCHEDULING (CS-TS) METHOD

In this section, the proposed CS-TS method is evaluated with the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Cultural Algorithm (CA) based task scheduling methods using the metrics defined in the chapter 3.

Table 6.1 depicts the average execution time (S) by the existing optimization technique-based task scheduling strategies and proposed CS-TS technique. The average execution time is calculated with the number of tasks considered. Figure 6.1 depicts the graphical representation of the average execution time (in seconds) by the proposed CS-TS technique and existing GA, PSO, CA based Task scheduling techniques against number of tasks. From the table 6.1 and figure 6.1, it is clear that the proposed CS-TS technique done the resource allocation for the large number of tasks with less execution time than the existing techniques.

Table 6.1: Average execution time of the proposed CS-TS technique and existing GA, PSO and CA based task scheduling techniques against number of tasks

Number of tasks	Average Execution time in (s)			
	GA	PSO	CA	Proposed CS-TS
200	100	80	75	28
400	180	165	145	96
600	220	210	200	165
800	405	395	375	250
1000	600	580	575	400
1200	895	815	795	605
1400	1200	1075	900	800

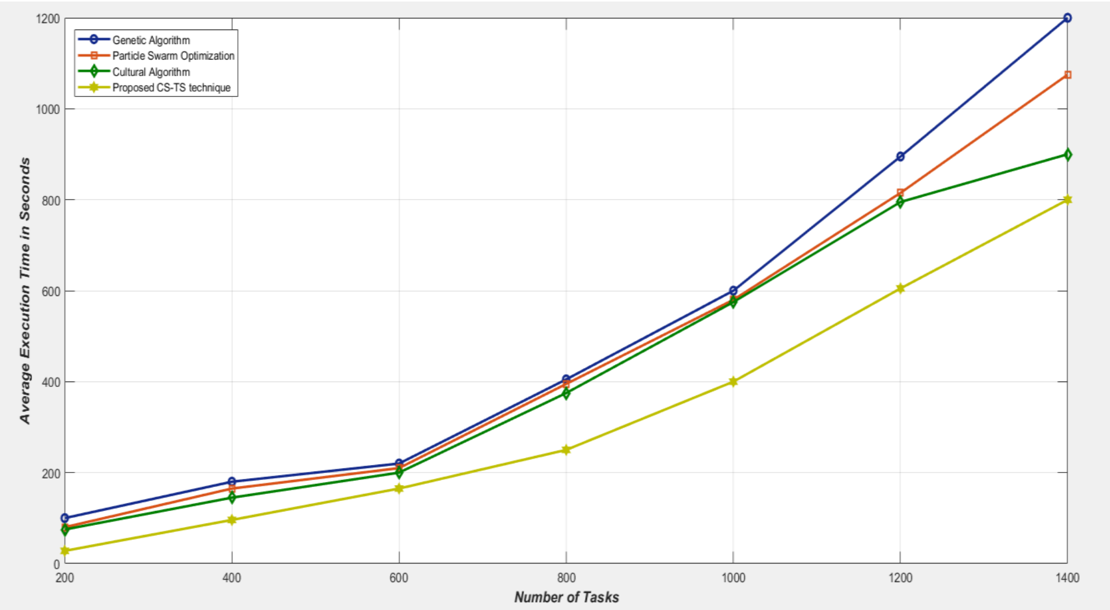

Figure 6.1: Graphical Representation of the Average Execution time in seconds by the Proposed CS-TS techniques and GA, PSO, CA based Task scheduling techniques against number of tasks

Table 6.2 depicts the utilization rate in % by the existing optimization technique-based task scheduling strategies and proposed CS-TS technique. The utilization rate is calculated with the number of tasks considered. This utilization rate indicates that the task scheduling technique selects the suitable VM for improving the balance of the system. Figure 6.2 depicts the graphical representation of the utilization rate (in %) by the proposed CS-TS technique and existing GA, PSO, CA based Task scheduling techniques against number of tasks. From the table 6.2 and figure 6.2, it is clear that the proposed CS-TS technique resulted in accurate VM allocation with load balancing of the system for the large number of tasks with maximum utilization rate than the existing task scheduling techniques.

Table 6.2: Utilization Rate of the proposed CS-TS technique and existing GA, PSO and CA based task scheduling techniques against number of tasks

Number of tasks	Utilization rate in (%)			
	GA	PSO	CA	Proposed CS-TS
200	8	11	10	12
400	10	20	18	28
600	21	25	30	40
800	31	35	36	52
1000	40	42	48	68
1200	48	52	59	80
1400	55	58	63	92

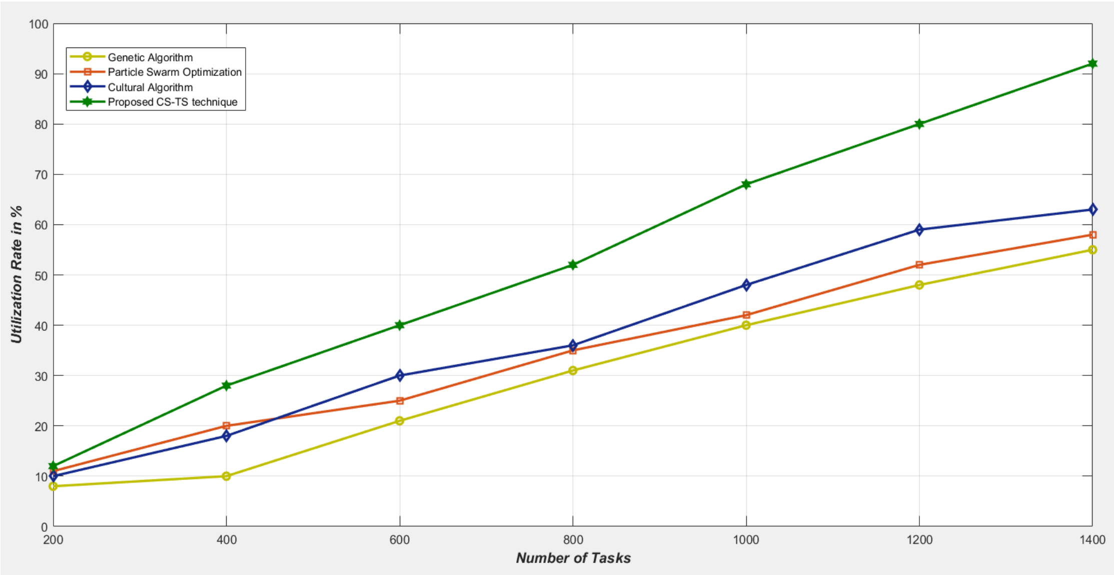

Figure 6.2: Graphical Representation of the utilization rate in % by the Proposed CS-TS techniques and GA, PSO, CA based Task scheduling techniques against number of tasks

Table 6.3 depicts the average makespan (s) by the existing optimization technique-based task scheduling strategies and proposed CS-TS technique. The average makespan is calculated with the number of tasks considered. Figure 6.3 depicts the graphical representation of the average makespan (in seconds) by the proposed CS-TS technique and existing GA, PSO, CA based Task scheduling techniques against number of tasks. From the table 6.3 and figure 6.3, it is clear that the proposed CS-TS technique done the resource allocation for the large number of tasks with less makespan than the existing techniques.

Table 6.3: Average makespan (in seconds) of the proposed CS-TS technique and existing GA, PSO and CA based task scheduling techniques against number of tasks

Number of tasks	Average Makespan in Seconds			
	GA	**PSO**	**CA**	**Proposed CS-TS**
200	0.8	0.9	0.6	0.45
400	1.0	0.8	0.7	0.55
600	1.38	1.08	1.00	0.75
800	1.65	1.45	1.50	1.00
1000	1.75	1.68	1.45	1.05
1200	1.95	1.70	1.60	1.15
1400	2.05	1.95	1.74	1.45

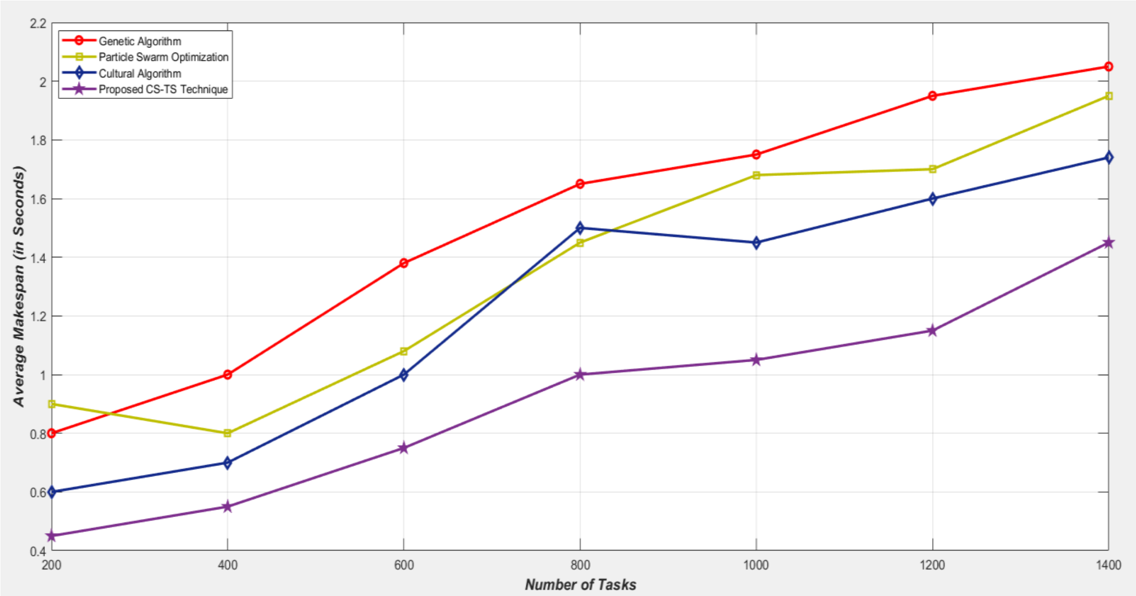

Figure 6.3: Graphical Representation of the Average Makespan in seconds by the Proposed CS-TS techniques and GA, PSO, CA based Task scheduling techniques against number of tasks

Table 6.4 depicts the average response time (S) by the existing optimization technique-based task scheduling strategies and proposed CS-TS technique. The average execution time is calculated with the number of tasks considered. Figure 6.4 depicts the graphical representation of the average response time (in seconds) by the proposed CS-TS technique and existing GA, PSO, CA based Task scheduling techniques against number of tasks. From the table 6.4 and figure 6.4, it is clear that the proposed CS-TS technique done the resource allocation for the large number of tasks with less response time than the existing techniques.

Table 6.4: Average Response Time (in seconds) of the proposed CS-TS technique and existing GA, PSO and CA based task scheduling techniques against number of tasks

Number of tasks	Average Response in Seconds			
	GA	PSO	CA	Proposed CS-TS
200	75	70	68	30
400	175	165	160	95
600	210	205	200	120
800	420	380	365	250
1000	610	600	570	400
1200	900	800	740	600
1400	1200	1050	900	785

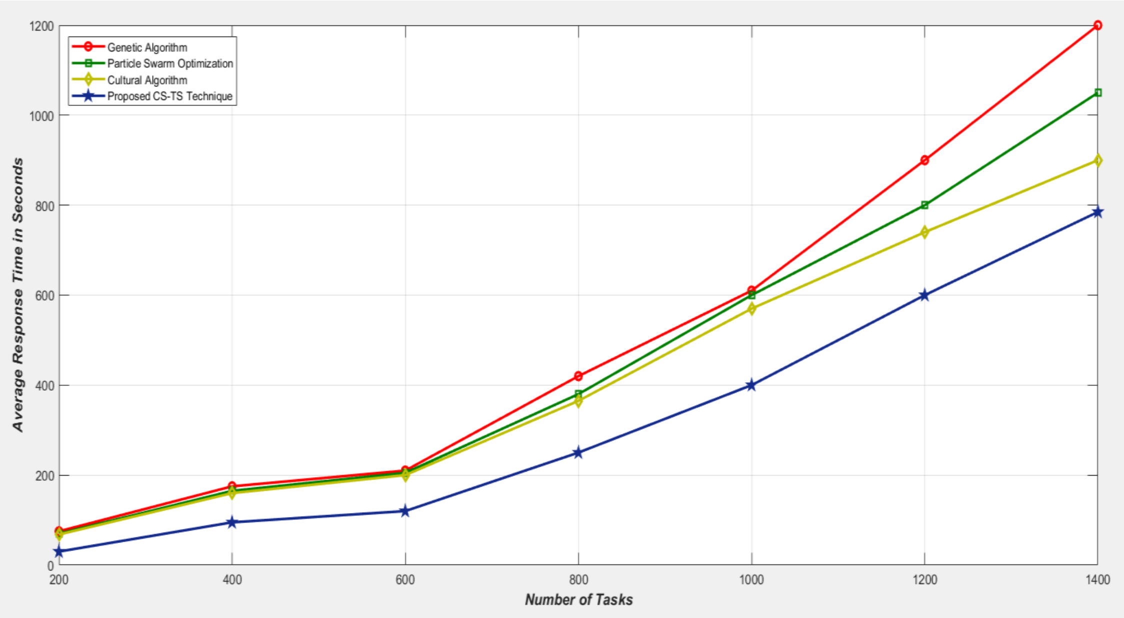

Figure 6.4: Graphical Representation of the Average Response time in seconds by the Proposed CS-TS techniques and GA, PSO, CA based Task scheduling techniques against number of tasks

6.3 PERFORMANCE ANALYSIS OF THE PROPOSED OPTIMIZATION BASED NEURAL NETWORK BASED HOST DETECTION METHOD

In this section, the performance of the proposed Optimization – Neural Network (O-NN) based host detection method is evaluated with the existing host detection techniques like Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Local Regression (LR) and Median Absolute Deviation (MAD) using the evaluation metrics defined in the chapter 4.5.1 for varying number of tasks, hosts and VMs.

6.3.1 Result obtained by the Host Detection techniques at Number of Hosts=400 and Number of VMs=100

Table 6.5 depicts the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100. Figure 6.5 gives the graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100. From the table 6.5 and figure 6.5, it is clear that the proposed Optimization-NN based detection method consumes less energy when it is compared with existing techniques.

Table 6.5: Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	7.41	11.02	13.5	14.7	15.6
200	8.32	12.5	15.3	16.8	17.5
300	9.31	13.7	16.9	17.6	18.1
400	9.97	14.3	17.8	18.2	19.2
500	10.12	15.4	18.4	19.3	20.3
600	10.89	16.3	19.6	20.4	21.5
700	11.23	17.5	20.7	21.2	22.3
800	11.82	18.6	21.2	22.6	23.4
900	12.56	19.1	22.7	23.9	24.7

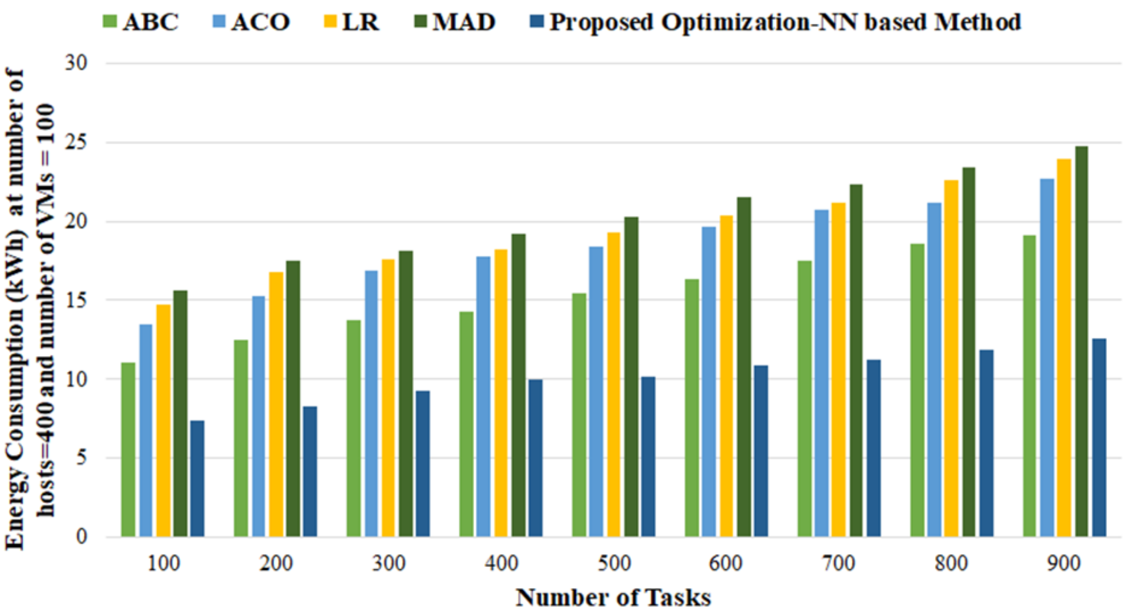

Figure 6.5: Graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100

Table 6.6 depicts the Service Level Agreement (SLA) Violations (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100. Figure 6.6 gives the graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100. From the table 6.6 and figure 6.6, it is clear that the proposed Optimization-NN based detection method reduced the SLA violation when it is compared with existing techniques.

Table 6.6: SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100

Number of Tasks	SLA Violation (in %) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	0.045	0.865	0.955	1.082	1.956
200	0.387	1.103	1.312	1.893	2.523
300	0.638	2.232	2.567	2.963	3.478
400	0.812	3.275	3.825	4.012	4.961
500	1.149	4.357	4.952	5.374	6.017
600	1.473	5.531	5.745	6.245	6.984
700	1.985	6.642	6.923	7.632	7.978
800	2.347	7.653	7.954	8.142	8.932
900	2.817	8.4751	8.863	9.687	10.451

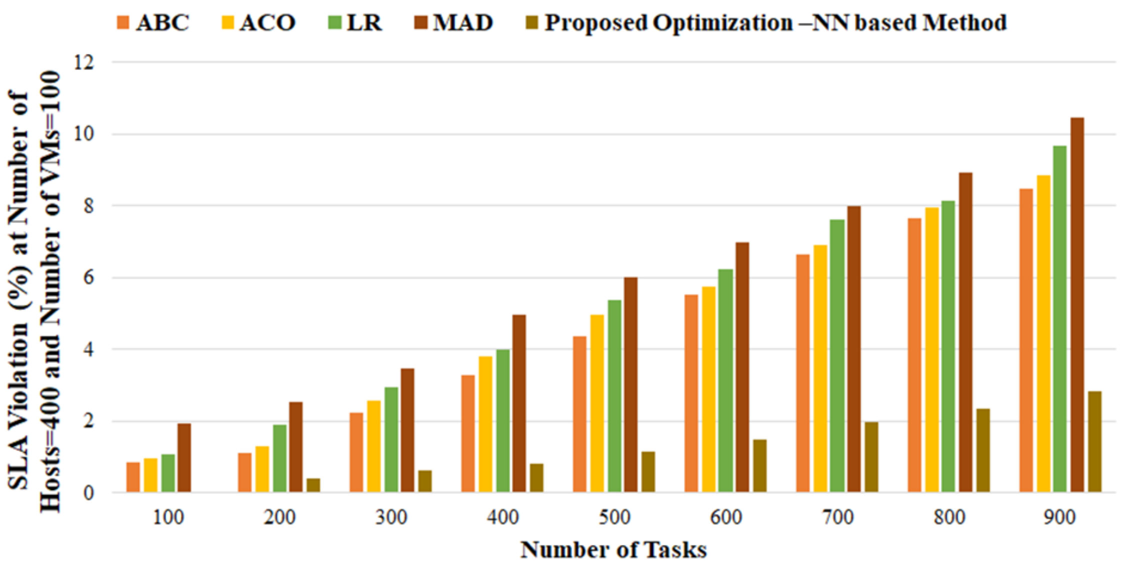

Figure 6.6: Graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=400 and number of VMs = 100

6.3.2 Result obtained by the Host Detection techniques at Number of Hosts=400 and Number of VMs=200

Table 6.7 depicts the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200. Figure 6.7 gives the graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200. From the table 6.7 and figure 6.7, it is clear that the proposed Optimization-NN based detection method consumes less energy when it is compared with existing techniques.

Table 6.7: Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	7.96	13.4	15.8	15.9	16.4
200	8.42	14.9	16.7	16.8	17.5
300	9.14	15.5	17.3	17.9	18.2
400	9.85	16.4	18.2	18.6	19.1
500	10.19	17.1	19.6	20.1	20.8
600	10.89	18.3	20.4	21.4	21.9
700	11.27	19.7	21.3	22.7	23.6
800	11.96	20.2	22.5	23.4	24.8
900	12.54	20.9	23.9	24.7	25.4

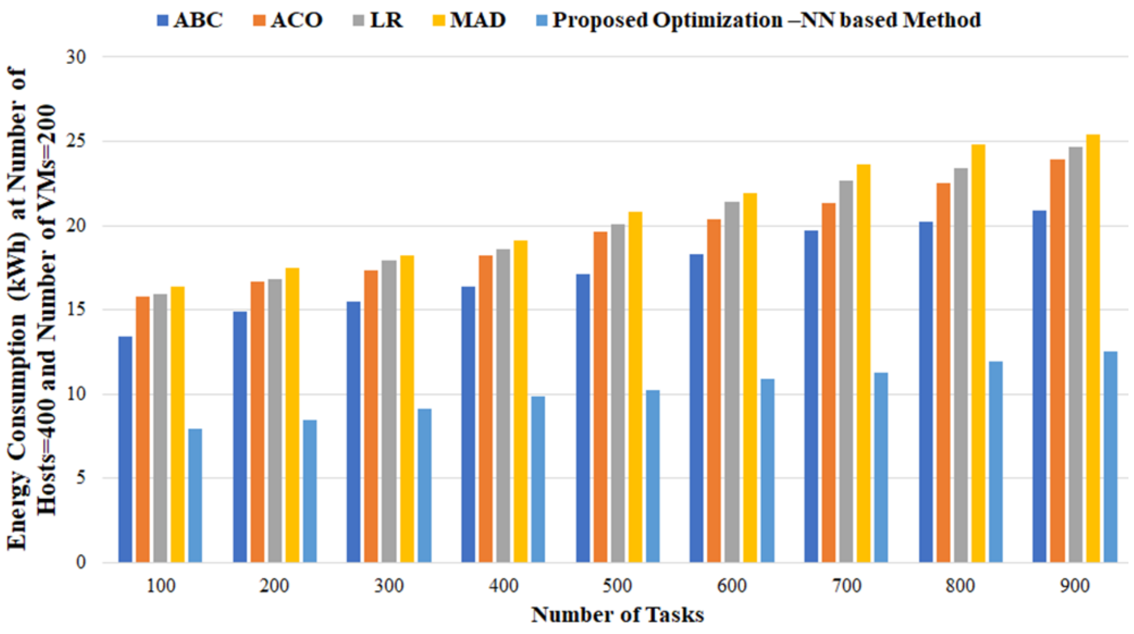

Figure 6.7: Graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=400 and number of VMs = 200

Table 6.8 depicts the Service Level Agreement (SLA) Violations (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200. Figure 6.8 gives the graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200. From the table 6.8 and figure 6.8, it is clear that the proposed Optimization-NN based detection method reduced the SLA violation when it is compared with existing techniques.

Table 6.8: SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200

Number of Tasks	SLA Violation (in %) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	0.452	0.974	2.844	3.171	3.956
200	0.976	1.814	3.221	5.782	4.914
300	1.327	2.411	4.458	6.852	6.587
400	2.123	3.583	5.734	7.101	7.852
500	3.268	4.549	6.841	8.283	8.726
600	4.582	5.743	7.834	8.134	9.173
700	5.874	6.831	8.732	9.513	10.167
800	6.436	7.845	9.843	10.231	10.891
900	6.926	8.687	10.952	11.776	12.020

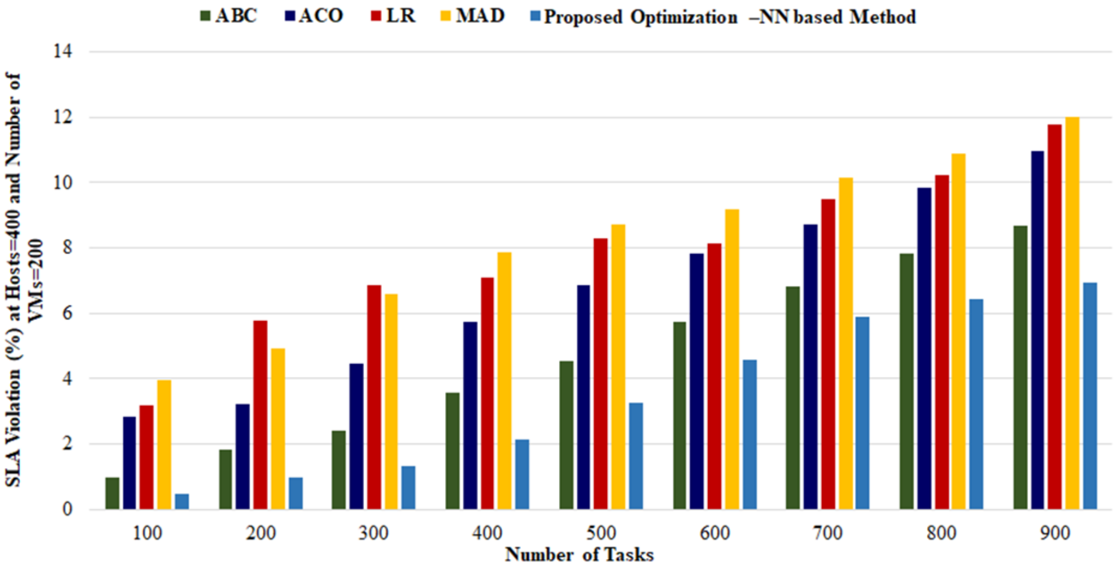

Figure 6.8: Graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=400 and number of VMs = 200

6.3.3 Result obtained by the Host Detection techniques at Number of Hosts=800 and Number of VMs=100

Table 6.9 depicts the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100. Figure 6.9 gives the graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100. From the table 6.9 and figure 6.9, it is clear that the proposed Optimization-NN based detection method consumes less energy when it is compared with existing techniques.

Table 6.9: Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	14.3	16.8	21.5	22.3	23.4
200	15.7	17.2	22.7	23.4	24.6
300	16.2	18.6	23.4	24.6	25.7
400	17.4	19.9	24.9	25.8	26.6
500	18.5	20.4	25.6	26.9	27.8
600	18.9	21.3	26.7	27.1	28.3
700	19.7	22.5	27.1	28.4	29.5
800	20.6	23.7	28.4	29.3	30.2
900	21.2	24.8	29.6	30.7	31.3

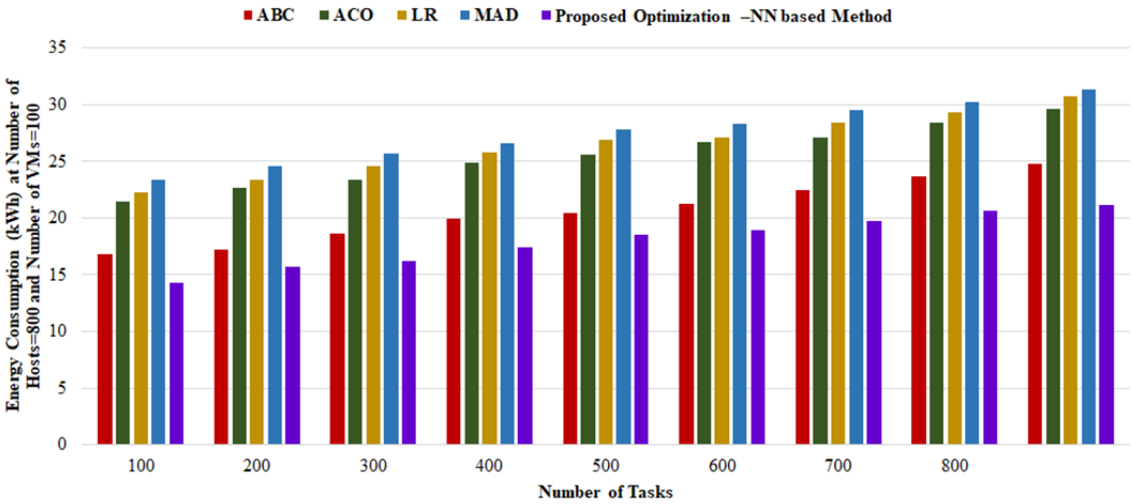

Figure 6.9: Graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=800 and number of VMs = 100

Table 6.10 depicts the SLA violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100. Figure 6.10 gives the graphical representation of the SLA violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100. From the table 6.10 and figure 6.10, it is clear that the proposed Optimization-NN based detection method reduced the SLA violation when it is compared with existing techniques.

Table 6.10: SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100

Number of Tasks	SLA Violation (in %) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	1.371	1.863	3.632	4.352	5.174
200	2.764	2.925	4.412	5.971	6.826
300	3.436	3.532	6.647	7.641	7.409
400	4.315	4.772	7.523	8.322	8.643
500	5.457	5.738	8.632	9.651	9.835
600	6.761	6.931	9.622	10.253	10.391
700	7.663	7.622	10.521	11.425	11.349
800	8.624	8.934	11.932	12.422	12.672
900	9.714	9.876	12.841	13.664	13.313

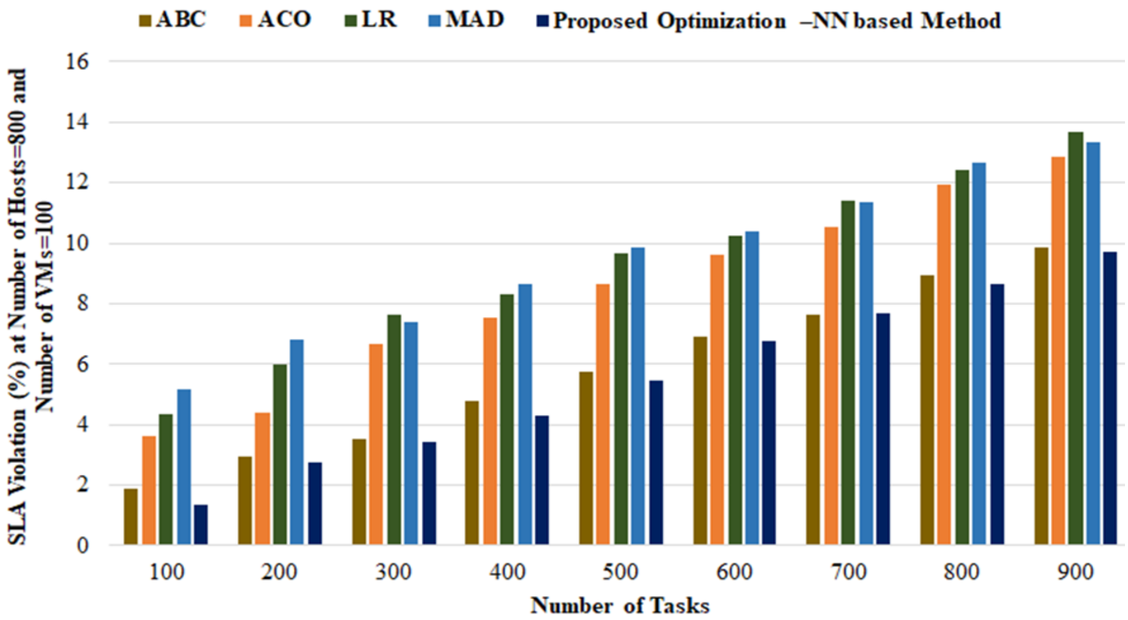

Figure 6.10: Graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=800 and number of VMs = 100

6.3.4 Result obtained by the Host Detection techniques at Number of Hosts=800 and Number of VMs=200

Table 6.11 depicts the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200. Figure 6.11 gives the graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200. From the table 6.11 and figure 6.11, it is clear that the proposed Optimization-NN based detection method consumes less energy when it is compared with existing techniques.

Table 6.11: Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	16.3	19.6	28.5	29.6	30.3
200	17.9	20.3	29.4	30.4	31.8
300	18.5	21.7	30.6	31.7	32.5
400	19.2	22.6	31.9	32.4	33.7
500	20.4	23.5	32.4	33.8	34.5
600	21.6	24.4	33.7	34.1	35.9
700	22.25	25.9	34.8	35.5	36.4
800	22.9	26.7	35.6	36.1	37.3
900	23.6	27.9	36.8	37.9	38.2

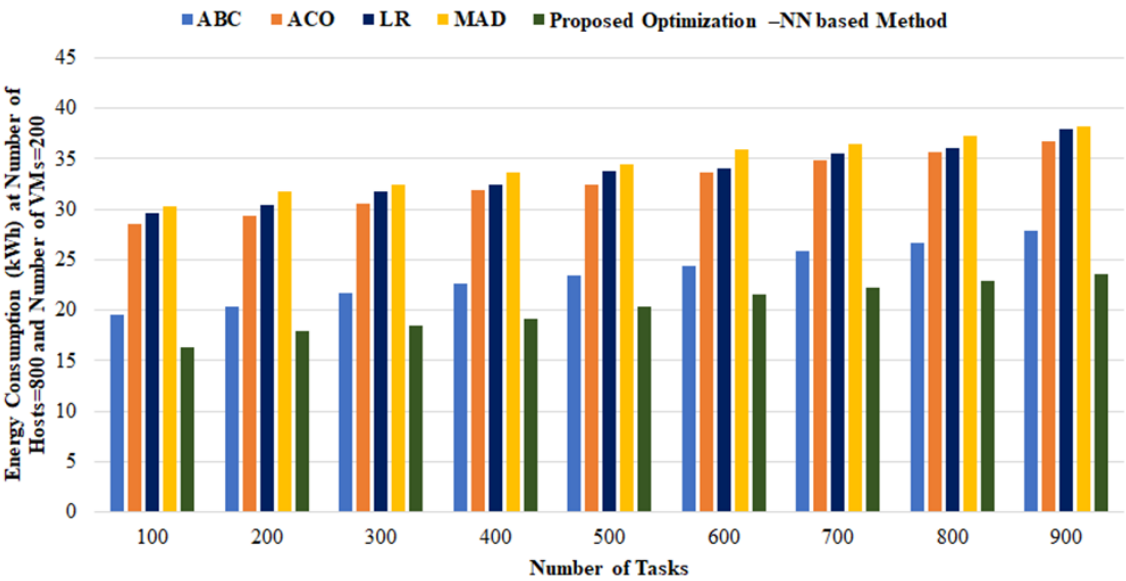

Figure 6.11: Graphical representation of the Energy consumption (kWh) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=800 and number of VMs = 200

Table 6.12 depicts the SLA violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200. Figure 6.12 gives the graphical representation of the SLA violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200. From the table 6.12 and figure 6.12, it is clear that the proposed Optimization-NN based detection method reduced the SLA violation when it is compared with existing techniques.

Table 6.12: SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200

Number of Tasks	SLA Violation (in %) by Host Detection techniques				
	Proposed Optimization-NN	ABC	ACO	LR	MAD
100	1.952	2.641	5.414	5.531	6.253
200	3.542	3.713	6.624	6.862	7.614
300	4.614	4.313	7.735	7.932	8.618
400	5.537	5.551	8.715	8.946	9.825
500	6.679	6.916	8.814	9.873	10.617
600	7.943	7.852	9.815	10.972	11.272
700	8.772	8.813	10.862	11.814	12.737
800	9.916	9.726	12.713	12.947	13.481
900	10.626	10.964	13.662	14.132	14.424

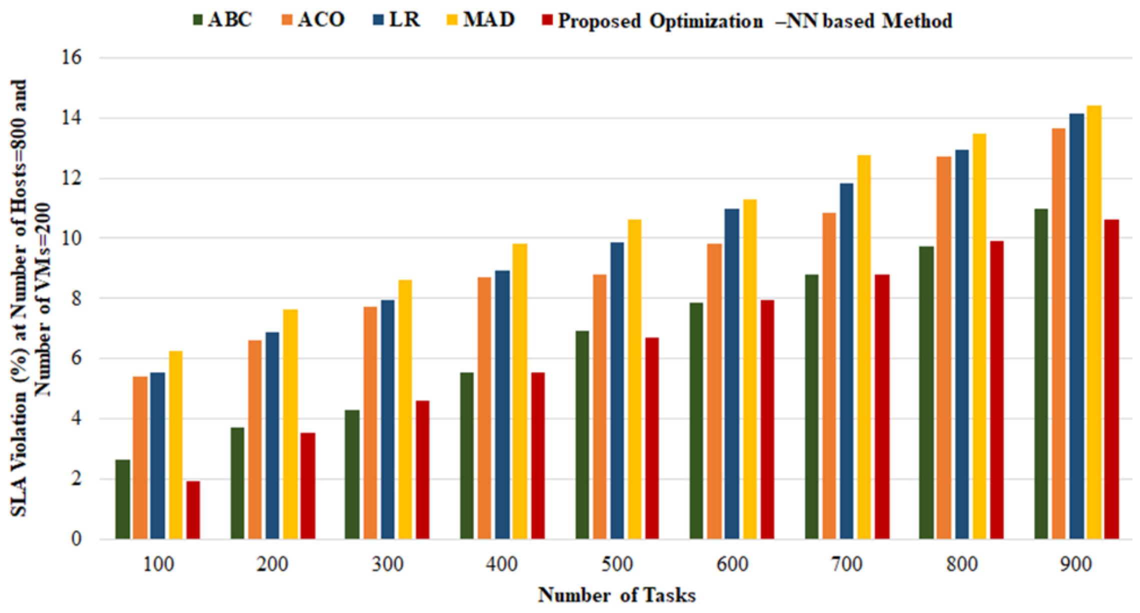

Figure 6.12: Graphical representation of the SLA Violation (in %) of the Proposed Optimization-NN based Method, LR, MAD, ABC, and ACO at number of hosts=800 and number of VMs = 200

6.4 PERFORMANCE ANALYSIS OF THE PROPOSED VM CONSOLIDATION STRATEGY

The performance of the Proposed VM consolidation strategy is evaluated with the existing techniques like Cultural Algorithm (CA), Artificial Bee Colony Optimization (ABC).

Table 6.13 depicts the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Cultural Algorithm (CA), Artificial Bee Colony (ABC) optimization and Proposed Optimization – NN based host detection method at number of hosts=400 and number of VMs = 100. Figure 613 depicts the graphical representation of the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, CA, ABC optimization and Proposed O–NN based host detection method at number of hosts=400 and number of VMs = 100. From the table 6.13 and figure 6.13, it is clear that the proposed VM consolidation strategy consumes less energy when increasing the number of tasks.

Table 6.13: Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh)			
	Proposed VM consolidation Strategy	**Proposed O-NN**	**CA**	ABC
100	3.94	7.41	9.61	11.02
200	4.48	8.32	10.32	12.5
300	4.96	9.31	11.12	13.7
400	5.51	9.97	11.94	14.3
500	5.95	10.12	12.32	15.4
600	6.38	10.89	13.87	16.3
700	6.99	11.23	14.31	17.5
800	7.28	11.82	15.52	18.6
900	7.84	12.56	16.55	19.1

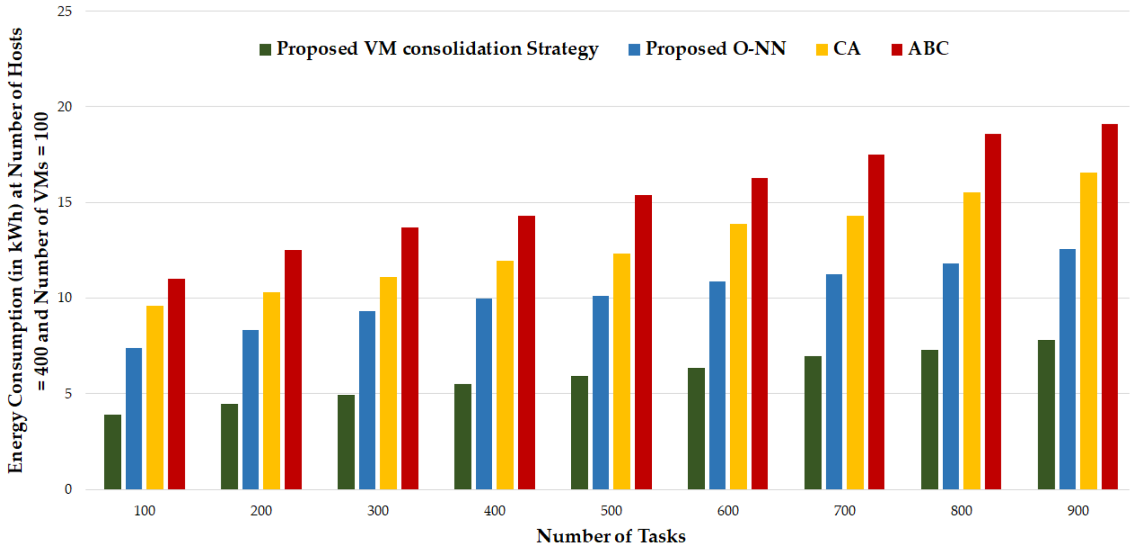

Figure 6.13: Graphical representation of the Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 100

Table 6.14 depicts the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 400 and number of VMs = 100. Figure 6.14 presents the graphical representation of the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 400 and number of VMs = 100. From the table 6.14 and figure 6.14, it is clear that the proposed VM consolidation strategy gives less SLA violations when it is compared with proposed Optimization-NN based Method, ABC and CA.

Table 6.14: SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 100

Number of Tasks	SLA Violation (in %)			
	Proposed VM consolidation Strategy	Proposed O-NN	CA	ABC
100	0.002	0.045	0.234	0.865
200	0.165	0.387	0.496	1.103
300	0.316	0.638	0.812	2.232
400	0.498	0.812	1.624	3.275
500	0.501	1.149	2.238	4.357
600	0.563	1.473	2.781	5.531
700	0.599	1.985	3.794	6.642
800	0.621	2.347	4.236	7.653
900	0.659	2.817	5.626	8.4751

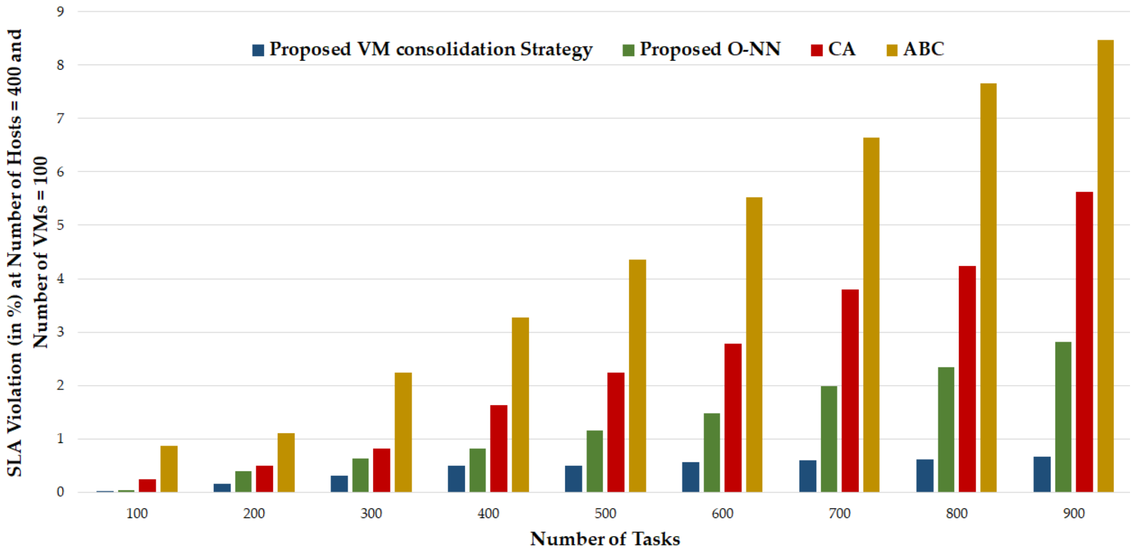

Figure 6.14: Graphical representation of the SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 100

Table 6.15 gives the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, and ABC at number of hosts=400 and number of VMs = 200. Figure 6.15 depicts the graphical representation of the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, ABC at number of hosts=400 and number of VMs = 200. From the table 6.15 and figure 6.16, it is clear that the proposed VM consolidation strategy consumes less energy when it is compared with proposed Optimization-NN based Method, CA and ABC.

Table 6.15: Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh)			
	Proposed VM consolidation Strategy	**Proposed O-NN**	**CA**	ABC
100	4.54	7.96	8.85	13.4
200	4.98	8.42	9.51	14.9
300	5.46	9.14	10.23	15.5
400	5.93	9.85	11.94	16.4
500	6.37	10.19	12.28	17.1
600	6.88	10.89	13.97	18.3
700	7.43	11.27	14.36	19.7
800	7.96	11.96	15.85	20.2
900	8.54	12.54	16.32	20.9

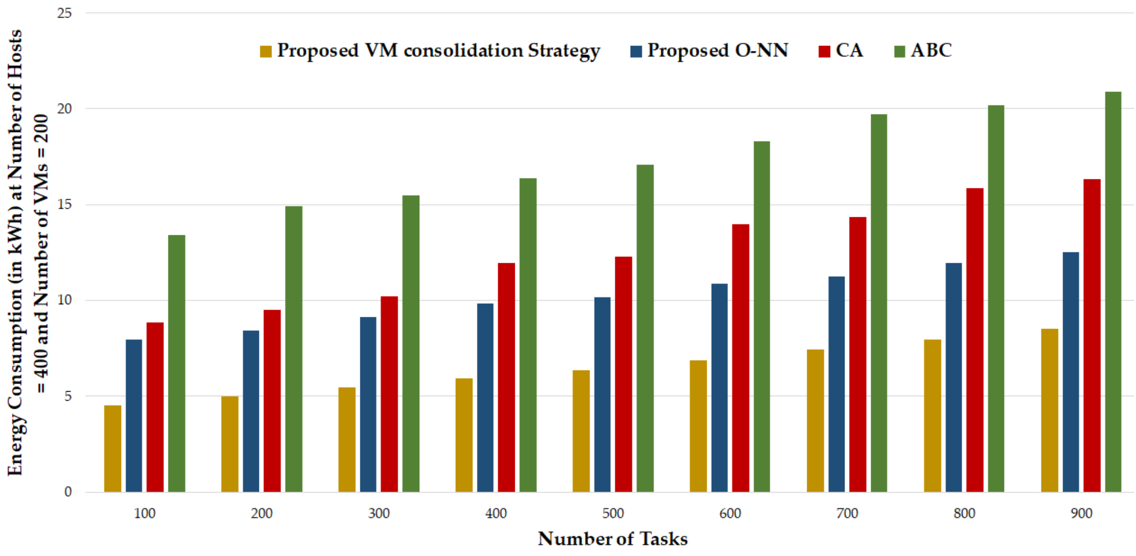

Figure 6.15: Graphical representation of the Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 200

Table 6.16 depicts the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 400 and number of VMs = 200. Figure 6.16 presents the graphical representation of the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 400 and number of VMs = 200. From the table 6.16 and figure 6.16, it is clear that the proposed VM consolidation strategy gives less SLA violations when it is compared with proposed Optimization-NN based Method, ABC and CA.

Table 6.16: SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 200

Number of Tasks	SLA Violation (in %)			
	Proposed VM consolidation Strategy	Proposed O-NN	CA	ABC
100	0.002	0.045	0.234	0.865
200	0.165	0.387	0.496	1.103
300	0.316	0.638	0.812	2.232
400	0.498	0.812	1.624	3.275
500	0.501	1.149	2.238	4.357
600	0.563	1.473	2.781	5.531
700	0.599	1.985	3.794	6.642
800	0.621	2.347	4.236	7.653
900	0.659	2.817	5.626	8.4751

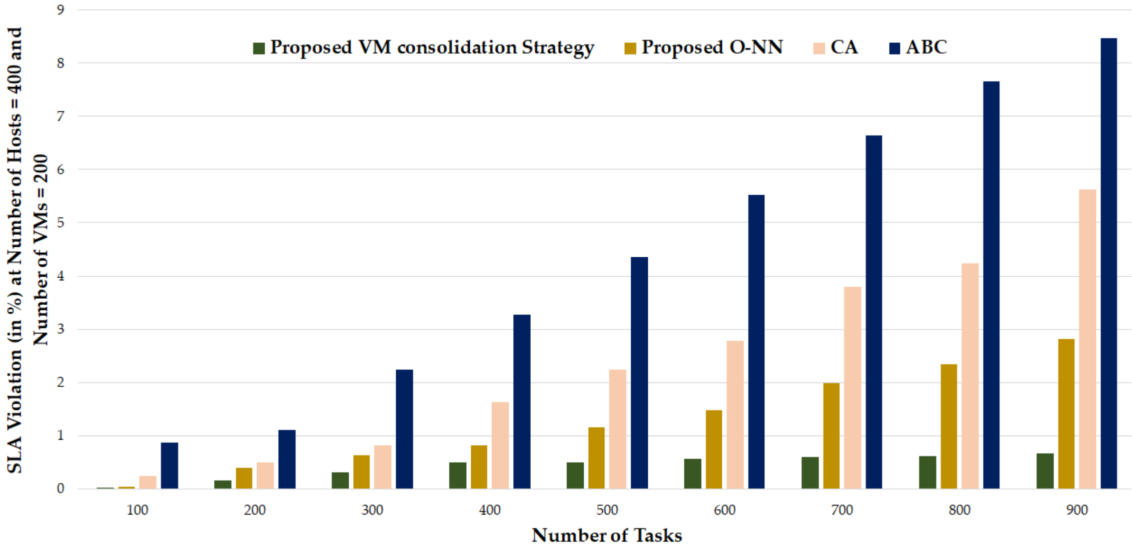

Figure 6.16: Graphical representation of the SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=400 and number of VMs = 200

Table 6.17 gives the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, and ABC at number of hosts=800 and number of VMs = 100. Figure 6.17 depicts the graphical representation of the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, ABC at number of hosts=800 and number of VMs = 100. From the table 6.17 and figure 6.17, it is clear that the proposed VM consolidation strategy consumes less energy when it is compared with proposed Optimization-NN based Method, CA and ABC.

Table 6.17: Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh)			
	Proposed VM consolidation Strategy	**Proposed O-NN**	**CA**	ABC
100	6.36	14.3	15.22	16.8
200	7.16	15.7	16.61	17.2
300	8.24	16.2	17.13	18.6
400	9.71	17.4	18.32	19.9
500	10.19	18.5	19.42	20.4
600	11.60	18.9	20.83	21.3
700	12.41	19.7	21.62	22.5
800	13.84	20.6	22.54	23.7
900	14.32	21.2	23.19	24.8

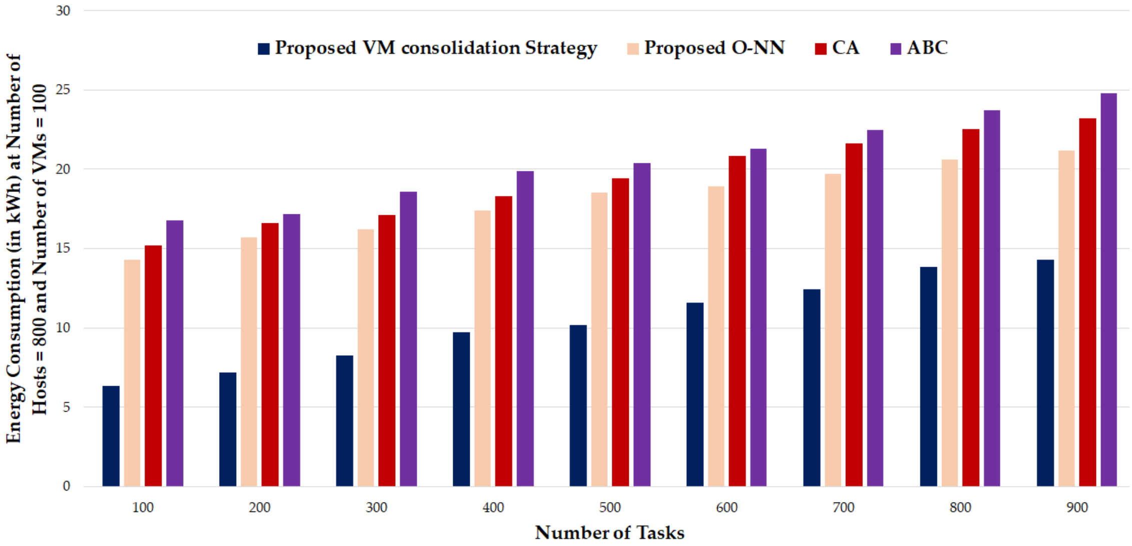

Figure 6.17: Graphical representation of the Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 100

Table 6.18 depicts the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 800 and number of VMs = 100. Figure 6.18 presents the graphical representation of the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 800 and number of VMs = 100. From the table 6.18 and figure 6.18, it is clear that the proposed VM consolidation strategy gives less SLA violations when it is compared with proposed Optimization-NN based Method, ABC and CA.

Table 6.18: SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 100

Number of Tasks	SLA Violation (in %)			
	Proposed VM consolidation Strategy	Proposed O-NN	CA	ABC
100	0.452	1.371	1.452	1.863
200	0.631	2.764	2.753	2.925
300	0.827	3.436	3.513	3.532
400	1.078	4.315	4.424	4.772
500	1.392	5.457	5.568	5.738
600	1.796	6.761	6.852	6.931
700	2.143	7.663	7.619	7.622
800	2.635	8.624	8.735	8.934
900	2.968	9.714	9.753	9.876

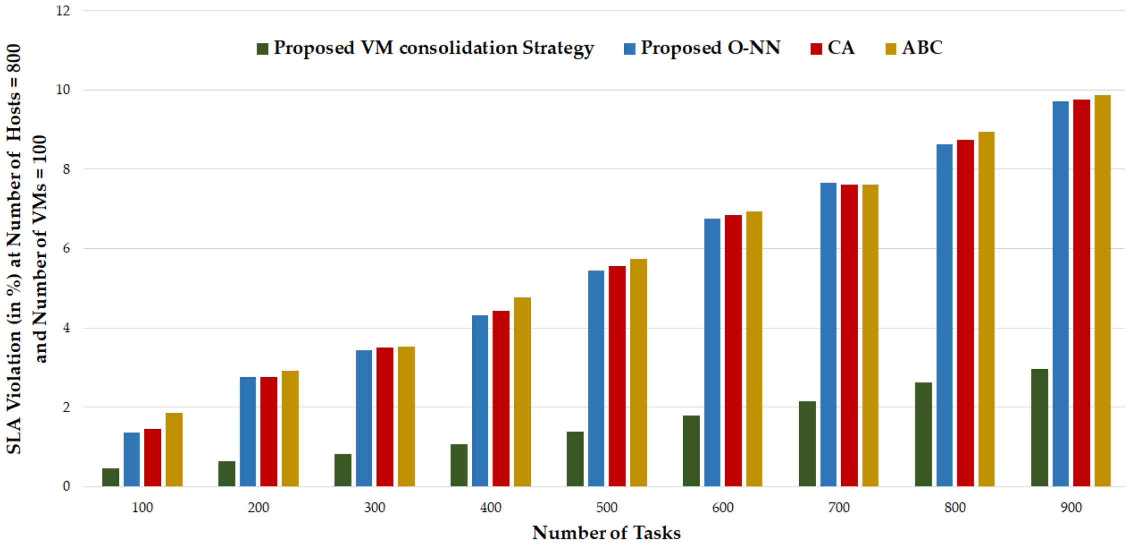

Figure 6.18: Graphical representation of the SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 100

Table 6.19 gives the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, and ABC at number of hosts=800 and number of VMs = 200. Figure 6.19 depicts the graphical representation of the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Proposed O-NN, CA, ABC at number of hosts=800 and number of VMs = 200. From the table 6.19 and figure 6.19, it is clear that the proposed VM consolidation strategy consumes less energy when it is compared with proposed Optimization-NN based Method, CA and ABC.

Table 6.19: Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh)			
	Proposed VM consolidation Strategy	**Proposed O-NN**	**CA**	ABC
100	7.44	16.3	17.2	19.6
200	8.24	17.9	18.8	20.3
300	9.12	18.5	19.43	21.7
400	10.52	19.2	20.19	22.6
500	11.27	20.4	21.33	23.5
600	12.42	21.6	22.53	24.4
700	13.22	22.25	23.34	25.9
800	14.62	22.9	24.82	26.7
900	15.14	23.6	25.55	27.9

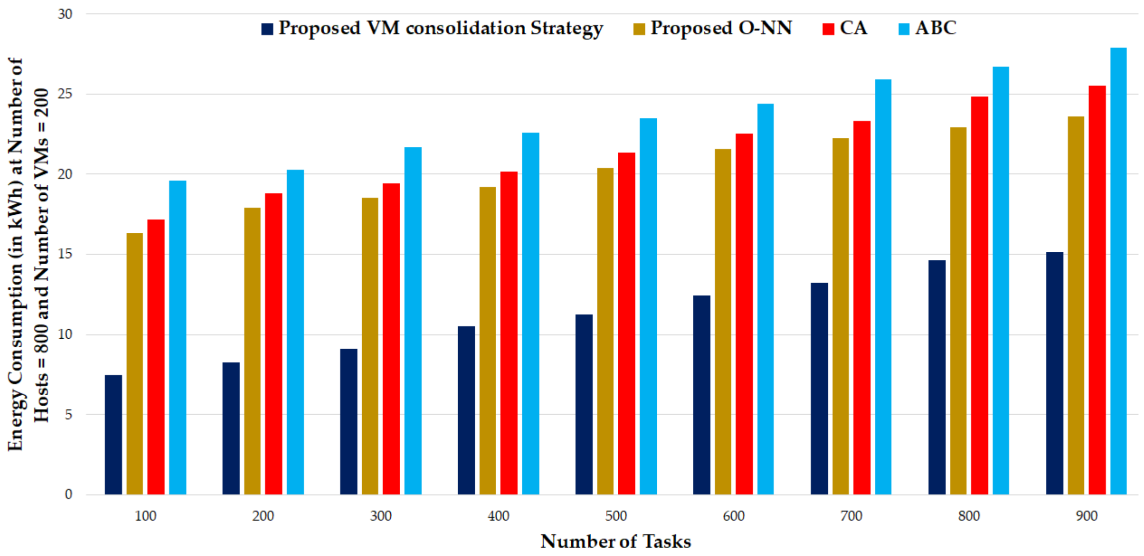

Figure 6.19: Graphical representation of the Energy Consumption (in kWh) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 200

Table 6.20 depicts the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 800 and number of VMs = 200. Figure 6.20 presents the graphical representation of the SLA violation (in %) obtained by the Proposed VM Consolidation Strategy, Proposed O-NN, CA and ABC at the number of hosts = 800 and number of VMs = 200. From the table 6.20 and figure 6.20, it is clear that the proposed VM consolidation strategy gives less SLA violations when it is compared with proposed Optimization-NN based Method, ABC and CA.

Table 6.20: SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 200

Number of Tasks	SLA Violation (in %)			
	Proposed VM consolidation Strategy	**Proposed O-NN**	**CA**	ABC
100	0.671	1.952	2.241	2.641
200	0.813	3.542	3.651	3.713
300	1.245	4.614	4.723	4.313
400	1.866	5.537	5.628	5.551
500	2.404	6.679	6.782	6.916
600	2.914	7.943	7.623	7.852
700	3.331	8.772	8.582	8.813
800	3.813	9.916	9.524	9.726
900	4.146	10.626	10.215	10.964

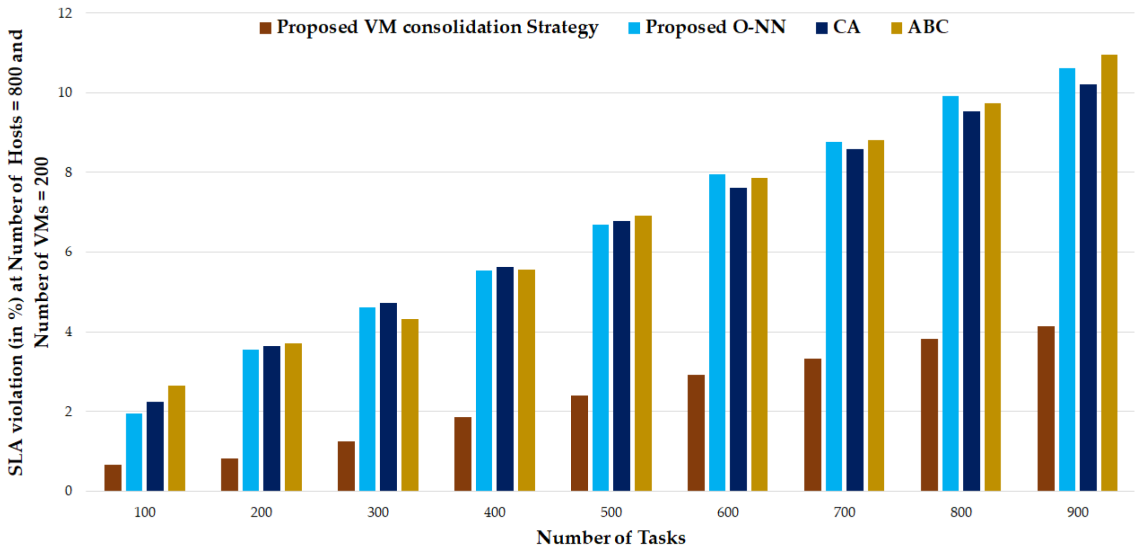

Figure 6.20: Graphical representation of the SLA Violation (in %) obtained by the Proposed VM consolidation strategy, Proposed O-NN, CA and ABC at number of hosts=800 and number of VMs = 200

6.5 SUMMARY

In this chapter, the performance of the proposed methodologies are evaluated with their existing techniques using various evaluation metrics. From the results given in the chapter, it is shown that the proposed methodologies performs better than the existing techniques.

CHAPTER – 7

CONCLUSION

The unusual trend of an expanding number of clients using public cloud computing services drives cloud service providers to optimise their resource utilisation and administration to the maximum. This includes managing cloud users' virtual machines (VMs) that run on one or more of the cloud datacenters' thousands of hosting servers or physical machines (PMs). The major concerns of cloud service providers are two main problems that have a significant impact on their infrastructure usage and utilisation: where to initially place the VMs and where to relocate them if we need to move them. VM migration, in conjunction with the VM consolidation technique, will aid in preventing physical servers from overloading or reducing the number of active physical servers for better resource efficiency and energy savings. Effectively recognising overloaded servers will aid in increasing cloud system performance and lowering total operational costs, allowing cloud providers to compete in the market.

Users may access hundreds of thousands of virtualized resources in cloud computing, making it impractical for everyone to distribute each work manually. Cloud computing has shifted the job scheduling complexity to the virtual machine layer as a result of commercialization and virtualization. As a result, scheduling is critical in cloud computing for efficiently and effectively assigning resources to each activity. The continuous change of VM availability in the cloud platform makes meeting the task QoS requirement difficult.

This research work proposed a task scheduling algorithm based on VM availability awareness to solve the matching problem between available task processing capacities of VMs and task QoS requirement and realised workload balancing of servers in the cloud platform to improve task scheduling capacities of VMs in the cloud platform and satisfy task QoS requirement.

Through the first contribution, a hybrid Cultural Swarm based Task Scheduling method is proposed, since Task Scheduling plays a major issue in the cloud computing environment. Evolutionary algorithms like Particle Swarm Optimization and Cultural Algorithm are combined to increase the utilization rate and reduce the average response time, execution time and makespan against number of tasks allocation for the VM.

In the second contribution, the Artificial Bee Colony Optimization approach was used to improve the weight of the Artificial Neural Network for host classification. By multiplying the number of VMs and hosts, ABC calculated the memory, CPU, power threshold, and energy usage. Based on the findings, it is obvious that the proposed Optimization-based Host identification methodology used less energy and reduced SLA violations as the number of hosts and virtual machines increased. The proposed method is compared to other optimization approaches such as Ant Colony Optimization, ABC, Local Regression, and Median Absolute Deviation. From the results, it is clear that the proposed Optimization based host detection performs better in terms of energy consumption and SLA violation than the other techniques.

In the third contribution, proposed VM consolidation strategy is proposed with BFD algorithm for VM placement and Enhanced Genetic Algorithm based VM migration. This proposed strategy computes the type of host using second contribution, which reduces the energy consumption and SLA violation with varying number of tasks, VMs, and hosts.

REFERENCES

[1] Dillon, Tharam, Chen Wu, and Elizabeth Chang. "Cloud computing: issues and challenges." *2010 24th IEEE international conference on advanced information networking and applications*. Ieee, 2010.

[2] Armbrust, Michael, et al. "A view of cloud computing." *Communications of the ACM* 53.4 (2010): 50-58.

[3] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).

[4] Savu, Laura. "Cloud computing: Deployment models, delivery models, risks and research challenges." *2011 International Conference on Computer and Management (CAMAN)*. IEEE, 2011.

[5] Zhao, Gansen, et al. "Deployment models: Towards eliminating security concerns from cloud computing." *2010 International Conference on High Performance Computing & Simulation*. IEEE, 2010.

[6] Leimeister, Stefanie, et al. "The business perspective of cloud computing: actors, roles and value networks." (2010).

[7] Huang, C., and C. Hsieh. "Sociology view on cloud computing value: Actor network theory perspective." *Proceedings 1st Conference on Cloud Computing GRIDs and Virtualization, Lisbon, Portugal*. 2010.

[8] Morgan, Lorraine, and Kieran Conboy. "Key factors impacting cloud computing adoption." *Computer* 46.10 (2013): 97-99.

[9] Gorelik, Eugene. *Cloud computing models*. Diss. Massachusetts Institute of Technology, 2013.

[10] Chang, Victor, et al. "A categorisation of cloud computing business models." *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE, 2010.

[11] Manohar, Nivedita. "A survey of virtualization techniques in cloud computing." *Proceedings of international conference on vlsi, communication, advanced devices, signals & systems and networking (vcasan-2013)*. Springer, India, 2013.

[12] Younge, Andrew J., and Geoffrey C. Fox. "Advanced virtualization techniques for high performance cloud cyberinfrastructure." *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE, 2014.

[13] Malhotra, Lakshay, Devyani Agarwal, and Arunima Jaiswal. "Virtualization in cloud computing." *J. Inform. Tech. Softw. Eng* 4.2 (2014): 1-3.

[14] Kumar, Rakesh, and Shilpi Charu. "An importance of using virtualization technology in cloud computing." *Global Journal of Computers & Technology* 1.2 (2015).

[15] Jain, Nancy, and Sakshi Choudhary. "Overview of virtualization in cloud computing." *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016.

[16] Shah-Mansouri, Hamed, Vincent WS Wong, and Robert Schober. "Joint optimal pricing and task scheduling in mobile cloud computing systems." *IEEE Transactions on Wireless Communications* 16.8 (2017): 5218-5232.

[17] Kong, Weiwei, Yang Lei, and Jing Ma. "Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism." *Optik* 127.12 (2016): 5099-5104.

[18] Ma, Juntao, et al. "A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing." *Wireless Communications, Networking and Applications*. Springer, New Delhi, 2016. 829-835.

[19] Verma, Manish, et al. "Dynamic resource demand prediction and allocation in multi- tenant service clouds." *Concurrency and Computation: Practice and Experience* 28.17 (2016): 4429-4442.

[20] Bala, Anju, and Inderveer Chana. "Multilevel priority-based task scheduling algorithm for workflows in cloud computing environment." *Proceedings of International Conference on ICT for Sustainable Development*. Springer, Singapore, 2016.

[21] Yang, Xianda, et al. "Multi-resource allocation for virtual machine placement in video surveillance cloud." *International Conference on Human Centered Computing*. Springer, Cham, 2016.

[22] Saxena, Deepika, R. K. Chauhan, and Ramesh Kait. "Dynamic fair priority optimization task scheduling algorithm in cloud computing: concepts and implementations." *International Journal of Computer Network and Information Security* 8.2 (2016): 41.

[23] Tao, Dan, et al. "Resource Scheduling and Data Locality for Virtualized Hadoop on IaaS Cloud Platform." *International Conference on Big Data Computing and Communications*. Springer, Cham, 2016.

[24] Djebbar, Esma Insaf, and Ghalem Belalem. "Tasks scheduling and resource allocation for high data management in scientific cloud computing environment." *International Conference on Mobile, Secure, and Programmable Networking*. Springer, Cham, 2016.

[25] Pop, Florin, and Maria Potop-Butucaru. "ARMCO: Advanced topics in resource management for ubiquitous cloud computing: An adaptive approach." (2016): 79-81.

[26] Wang, Xiaoli, Yuping Wang, and Yue Cui. "An energy-aware bi-level optimization model for multi-job scheduling problems under cloud computing." *Soft Computing* 20.1 (2016): 303-317.

[27] Mousavi, Seyedmajid, Amir Mosavi, and Annamária R. Varkonyi-Koczy. "A load balancing algorithm for resource allocation in cloud computing." *International Conference on Global Research and Education*. Springer, Cham, 2017.

[28] Singh, Aarti, Dimple Juneja, and Manisha Malhotra. "A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing." *Journal of King Saud University-Computer and Information Sciences* 29.1 (2017): 19-28.

[29] Sarkhel, Preeta, Himansu Das, and Lalit K. Vashishtha. "Task-scheduling algorithms in cloud environment." *Computational Intelligence in Data Mining*. Springer, Singapore, 2017. 553-562.

[30] Mohanty, Subhadarshini, et al. "MPSO: A Novel Meta-Heuristics for Load Balancing in Cloud Computing." *International Journal of Applied Evolutionary Computation (IJAEC)* 8.1 (2017): 1-25.

[31] Priya, V., and C. Nelson Kennedy Babu. "Moving average fuzzy resource scheduling for virtualized cloud data services." *Computer Standards & Interfaces* 50 (2017): 251- 257.

[32] Suri, P. K., and Sunita Rani. "Design of task scheduling model for cloud applications in multi cloud environment." *International Conference on Information, Communication and Computing Technology*. Springer, Singapore, 2017.

[33] Seth, Sonam, and Nipur Singh. "Dynamic threshold-based dynamic resource allocation using multiple vm migration for cloud computing systems." *International Conference on Information, Communication and Computing Technology*. Springer, Singapore, 2017.

[34] Singh, Poonam, Maitreyee Dutta, and Naveen Aggarwal. "A review of task scheduling based on meta-heuristics approach in cloud computing." *Knowledge and Information Systems* 52.1 (2017): 1-51.

[35] Alla, Hicham Ben, Said Ben Alla, and Abdellah Ezzati. "A Priority Based Task Scheduling in Cloud Computing Using a Hybrid MCDM Model." *International Symposium on Ubiquitous Networking*. Springer, Cham, 2017.

[36] Latiff, Muhammad Shafie Abd, Syed Hamid Hussain Madni, and Mohammed Abdullahi. "Fault tolerance aware scheduling technique for cloud computing

environment using dynamic clustering algorithm." *Neural Computing and Applications* 29.1 (2018): 279-293.

[37] Seth, Sonam, and Nipur Singh. "Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems." *International Journal of Information Technology* (2018): 1-5.

[38] Juarez, Fredy, Jorge Ejarque, and Rosa M. Badia. "Dynamic energy-aware scheduling for parallel task-based application in cloud computing." *Future Generation Computer Systems* 78 (2018): 257-271.

[39] Arya, K. S., P. V. Divya, and KR Remesh Babu. "Dynamic Resource Management Through Task Migration in Cloud." *International Conference on Intelligent Data Communication Technologies and Internet of Things*. Springer, Cham, 2018.

[40] Agarwal, Mohit, and Gur Mauj Saran Srivastava. "A cuckoo search algorithm-based task scheduling in cloud computing." *Advances in Computer and Computational Sciences*. Springer, Singapore, 2018. 293-299.

[41] Moazemi, Setareh, and Mehdi Javanmard. "A New Method of Scheduling Tasks in Cloud Computing." *Revista Publicando* 5.16 (2) (2018): 227-245.

[42] Alla, Hicham Ben, et al. "A novel task scheduling approach based on dynamic queues and hybrid meta-heuristic algorithms for cloud computing environment." *Cluster Computing* 21.4 (2018): 1797-1820.

[43] Xu, Xiaolong, et al. "Dynamic resource allocation for load balancing in fog environment." *Wireless Communications and Mobile Computing* 2018 (2018).

[44] Fatima, Aisha, et al. "Efficient resource allocation model for residential buildings in smart grid using fog and cloud computing." *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. Springer, Cham, 2018.

[45] Toosi, Adel Nadjaran, Richard O. Sinnott, and Rajkumar Buyya. "Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka." *Future Generation Computer Systems* 79 (2018): 765-775.

[46] Ashraf, Adnan, and Ivan Porres. "Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system." *International Journal of Parallel, Emergent and Distributed Systems* 33.1 (2018): 103-120.

[47] Yousefipour, Amin, Amir Masoud Rahmani, and Mohsen Jahanshahi. "Energy and cost-aware virtual machine consolidation in cloud computing." *Software: Practice and Experience* 48.10 (2018): 1758-1774.

[48] Chol, Kim Ryo, and Kim Sun Hui. "Virtual Machine Migration for Dynamic Server Resource Management in Cloud Computing Environments Based on OpenStack." (2018).

[49] Callau-Zori, Mar, et al. "An experiment-driven energy consumption model for virtual machine management systems." *Sustainable Computing: Informatics and Systems* 18 (2018): 163-174.

[50] Shaw, Subhadra Bose, Anil Kumar Singh, and Shailesh Tripathi. "Optimum Utilization of Resources Through Restricted Virtual Machine Migration and Efficient VM Placement in Cloud Data Center." *International Journal of Distributed Systems and Technologies (IJDST)* 9.4 (2018): 1-19.

[51] Ding, Weichao, et al. "Construction and performance analysis of unified storage cloud platform based on openstack with ceph rbd." *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 2018.

[52] Zhao, Changming, and Hao Yang. "The Virtual Machine Placement Algorithm Based on Equivalent Resource Model in the Offloading Data Center of Mobile Cloud Environment." *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*. IEEE, 2018.

[53] Jangiti, Saikishor, and Shankar Sriram VS. "Bulk-bin-packing based migration management of reserved virtual machine requests for green cloud computing." *EAI Endorsed Transactions on Energy Web* 6.24 (2019).

[54] Masdari, Mohammad, et al. "Bio-inspired virtual machine placement schemes in cloud computing environment: taxonomy, review, and future research directions." *Cluster Computing* (2019): 1-31.

[55] Mohammadhosseini, Mahdieh, Abolfazl Toroghi Haghighat, and Ebrahim Mahdipour. "An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm." *The Journal of Supercomputing* 75.10 (2019): 6904-6933.

[56] Shang, Fengjun, Xiong Xiong, and Luzhong Li. "Load-aware virtual machine placement and dynamic migration mechanism for OpenStack." *Journal of Computational Methods in Sciences and Engineering* 19.3 (2019): 719-749.

[57] Akintoye, Samson Busuyi, and Antoine Bagula. "Improving quality-of-service in cloud/fog computing through efficient resource allocation." *Sensors* 19.6 (2019): 1267.

[58] Qiao, Lei, et al. "Genetic Expression Programming Based Dynamic Virtual Machine Consolidation in Cloud Computing." *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, 2019.

[59] López, Jorge, Natalia Kushik, and Djamal Zeghlache. "Virtual machine placement quality estimation in cloud infrastructures using integer linear programming." *Software Quality Journal* 27.2 (2019): 731-755.

[60] López-Pires, Fabio, et al. "Evaluation of Two-Phase Virtual Machine Placement Algorithms for Green Cloud Datacenters." *2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W)*. IEEE, 2019.

[61] Zhao, Yan, et al. "Reducing the upfront cost of private clouds with clairvoyant virtual machine placement." *The Journal of Supercomputing* 75.1 (2019): 340-369.

[62] Tarahomi, Mehran, and Mohammad Izadi. "A prediction-based and power-aware virtual machine allocation algorithm in three-tier cloud data centers." *International Journal of Communication Systems* 32.3 (2019): e3870.

[63] Li, Jingmei, et al. "Research on Dynamic Virtual Machine Scheduling Strategy Based on Improved Genetic Algorithm." *Journal of Physics: Conference Series*. Vol. 1168. No. 5. IOP Publishing, 2019.

[64] Liaqat, Misbah, et al. "Characterizing dynamic load balancing in cloud environments using virtual machine deployment models." *IEEE Access* 7 (2019): 145767-145776.

[65] Kandoussi, El Mehdi, et al. "Modeling Virtual Machine Migration as a Security Mechanism by using Continuous-Time Markov Chain Model." *2019 4th World Conference on Complex Systems (WCCS)*. IEEE, 2019.

[66] Ismaeel, Salam, Ali Miri, and Ayman Al-Khazraji. "A novel host readiness factor for energy-efficient VM consolidation in cloud data centers." *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*. IEEE, 2019.

[67] Yang, Chao, et al. "An effective and scalable VM migration strategy to mitigate cross-VM side-channel attacks in cloud." *China Communications* 16.4 (2019): 151-171.

[68] Li, Rui, et al. "Multi-objective optimization for rebalancing virtual machine placement." *Future Generation Computer Systems* 105 (2020): 824-842.

[69] Talebian, Hamid, et al. "Optimizing virtual machine placement in IaaS data centers: taxonomy, review and open issues." *Cluster Computing* 23.2 (2020): 837-878.

[70] Zhao, Hui, et al. "Vm performance maximization and pm load balancing virtual machine placement in cloud." *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020.

[71] Yahaya, Rahimatu Hayatu, Faruku Umar Ambursa, and Bashir Galadanci. "A Two Phase Optimization Scheme with Efficient Prediction-based Triggering for Virtual Machine Placement in Cloud Datacenters." *Ilorin Journal of Computer Science and Information Technology* 3.1 (2020): 40-51.

[72] Kumari, K. Aruna, J. K. R. Sastry, and K. Rajasekhara Rao. "An Enhanced Cloud Computing Architecture focussed at Optimising VM Migration through an efficient Placement Algorithm." *International Journal* 8.6 (2020).

[73] Pyati, Mahantesh, D. G. Narayan, and Shivaraj Kengond. "Energy-efficient and Dynamic Consolidation of Virtual Machines in OpenStack-based Private Cloud." *Procedia Computer Science* 171 (2020): 2343-2352.

[74] Sutar, Sandeep G., Pallavi J. Mali, and Amruta Y. More. "Resource utilization enhancemnet through live virtual machine migration in cloud using ant colony optimization algorithm." *International Journal of Speech Technology* 23.1 (2020): 79-85.

[75] Braiki, Khaoula, and Habib Youssef. "Fuzzy-logic-based multi-objective best-fit-decreasing virtual machine reallocation." *The Journal of Supercomputing* 76.1 (2020): 427-454.

[76] Liu, Xialin, et al. "Virtual Machine Consolidation with Minimization of Migration Thrashing for Cloud Data Centers." *Mathematical Problems in Engineering* 2020 (2020).

[77] Tarafdar, Anurina, et al. "Energy and quality of service-aware virtual machine consolidation in a cloud data center." *The Journal of Supercomputing* (2020): 1-32.

[78] Abdullaha, Muhammad, et al. "Application Centric Virtual Machine Placements to Minimize Bandwidth Utilization in Datacenters." *INTELLIGENT AUTOMATION AND SOFT COMPUTING* 26.1 (2020): 13-25.

[79] Monshizadeh Naeen, Hossein, Esmaeil Zeinali, and Abolfazl Toroghi Haghighat. "Adaptive Markov-based approach for dynamic virtual machine consolidation in cloud data centers with quality-of-service constraints." *Software: Practice and Experience* 50.2 (2020): 161-183.

[80] Yang, QiangQiang, et al. "Energy-aware and load balancing based dynamic migration strategy for virtual machine." *2020 4th International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*. IEEE, 2020.

[81] Nair, Susmita JA, and TR Gopalakrishnan Nair. "VM placement with effective energy management in cloud using optimal VM allocation framework (OVAF)." *Indonesian Journal of Electrical Engineering and Computer Science* 18.3 (2020): 1531-1538.

[82] Shah-Mansouri, Hamed, Vincent WS Wong, and Robert Schober. "Joint optimal pricing and task scheduling in mobile cloud computing systems." *IEEE Transactions on Wireless Communications* 16.8 (2017): 5218-5232.

[83] Kong, Weiwei, Yang Lei, and Jing Ma. "Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism." *Optik* 127.12 (2016): 5099-5104.

[84] Xavier, VM Arul, and S. Annadurai. "Chaotic social spider algorithm for load balance aware task scheduling in cloud computing." *Cluster Computing* 22.1 (2019): 287-297.

[85] Marahatta, Avinab, et al. "Energy-aware fault-tolerant dynamic task scheduling scheme for virtualized cloud data centers." *Mobile Networks and Applications* 24.3 (2019): 1063-1077.

[86] Tang, Hengliang, et al. "Dynamic resource allocation strategy for latency-critical and computation-intensive applications in cloud–edge environment." *Computer Communications* 134 (2019): 70-82.

[87] Alkhalaileh, Mohammad, et al. "Dynamic Resource Allocation in Hybrid Mobile Cloud Computing for Data-Intensive Applications." *International Conference on Green, Pervasive, and Cloud Computing*. Springer, Cham, 2019.

[88] Nayak, Biswajit, Sanjay Kumar Padhi, and Prasant Kumar Pattnaik. "Static Task Scheduling Heuristic Approach in Cloud Computing Environment." *Information Systems Design and Intelligent Applications*. Springer, Singapore, 2019. 473-480.

[89] Kaur, Simranjit, et al. "Quality of Service (QoS) Aware Workflow Scheduling (WFS) in Cloud Computing: A Systematic Review." *Arabian Journal for Science and Engineering* 44.4 (2019): 2867-2897.

[90] Nayak, Suvendu Chandan, et al. "Dynamic Backfilling Algorithm to Increase Resource Utilization in Cloud Computing." *International Journal of Information Technology and Web Engineering (IJITWE)* 14.1 (2019): 1-26.

[91] Selvakumar, A., and G. Gunasekaran. "A Novel Approach of Load Balancing and Task Scheduling Using Ant Colony Optimization Algorithm." *International Journal of Software Innovation (IJSI)* 7.2 (2019): 9-20.

[92] Prassanna, J., and Neelanarayanan Venkataraman. "Threshold based multi-objective memetic optimized round Robin scheduling for resource efficient load balancing in cloud." *Mobile Networks and Applications* 24.4 (2019): 1214-1225.

[93] Panda, Sanjaya K., and Prasanta K. Jana. "An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems." *Cluster Computing* 22.2 (2019): 509-527.

[94] Dixit, Abhishek, et al. "CA-DE: Hybrid Algorithm Based on Cultural Algorithm and DE." *Machine Intelligence and Signal Analysis*. Springer, Singapore, 2019. 185-196.

[95] Jafari, Malihe, Eysa Salajegheh, and Javad Salajegheh. "An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses." *Engineering with Computers* 35.3 (2019): 781-801.

[96] Pan, Xiuqin, et al. "Hybrid particle swarm optimization with simulated annealing." *Multimedia Tools and Applications* 78.21 (2019): 29921-29936

[97] Venters, Will, and Edgar A. Whitley. "A critical review of cloud computing: researching desires and realities." *Journal of Information Technology* 27.3 (2012): 179-197.

[98] Lele, Ajey. "Cloud computing." *Disruptive technologies for the militaries and security*. Springer, Singapore, 2019. 167-185.

[99] Carroll, Mariana, Alta Van Der Merwe, and Paula Kotze. "Secure cloud computing: Benefits, risks and controls." *2011 Information Security for South Africa*. IEEE, 2011.

[100] Fard, Seyed Yahya Zahedi, Mohamad Reza Ahmadi, and Sahar Adabi. "A dynamic VM consolidation technique for QoS and energy consumption in cloud environment." *The Journal of Supercomputing* 73.10 (2017): 4347-4368.

[101] Cao, Zhibo, and Shoubin Dong. "Dynamic VM consolidation for energy-aware and SLA violation reduction in cloud computing." *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*. IEEE, 2012.

[102] Mc Donnell, Nicola, Enda Howley, and Jim Duggan. "Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing." *Future Generation Computer Systems* 108 (2020): 288-301.

[103] Yadav, Rahul, et al. "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center." *Wireless Networks* 26.3 (2020): 1905-1919.

[104] Ding, Weichao, et al. "Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers." *Future Generation Computer Systems* (2020).

[105] Xiao, Hui, Zhigang Hu, and Keqin Li. "Multi-objective vm consolidation based on thresholds and ant colony system in cloud computing." *IEEE Access* 7 (2019): 53441- 53453.

[106] Aslam, Anjum Mohd, and Mala Kalra. "Using Artificial Neural Network for VM Consolidation Approach to Enhance Energy Efficiency in Green Cloud." *Advances in Data and Information Sciences*. Springer, Singapore, 2019. 139-154.

[107] Hsieh, Sun-Yuan, et al. "Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers." *Journal of Parallel and Distributed Computing* 139 (2020): 99-109.

[108] Li, Lianpeng, et al. "SLA-aware and energy-efficient VM consolidation in cloud data centers using robust linear regression prediction model." *IEEE Access* 7 (2019): 9490- 9500.

[109] Li, Zhihua, et al. "Energy-efficient and quality-aware VM consolidation method." *Future Generation Computer Systems* 102 (2020): 789-809.

[110] Moghaddam, SeyedhamidMashhadi, et al. "Embedding individualized machine learning prediction models for energy efficient VM consolidation within Cloud data centers." *Future Generation Computer Systems* 106 (2020): 221-233.

[111] Kansal, Nidhi Jain, and Inderveer Chana. "Artificial bee colony based energy-aware resource utilization technique for cloud computing." *Concurrency and Computation: Practice and Experience* 27.5 (2015): 1207-1225.

[112] Kruekaew, Boonhatai, and Warangkhana Kimpan. "Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing." *International Journal of Computational Intelligence Systems* 13.1 (2020): 496-510.

[113] Witanto, Joseph Nathanael, Hyotaek Lim, and Mohammed Atiquzzaman. "Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management." *Future generation computer systems* 87 (2018): 35-42.

[114] Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." *Journal of network and computer applications* 34.1 (2011): 1-11.

[115] Gorelik, Eugene. *Cloud computing models*. Diss. Massachusetts Institute of Technology, 2013.

[116] Lin, Ching-Chi, Pangfeng Liu, and Jan-Jan Wu. "Energy-aware virtual machine dynamic provision and scheduling for cloud computing." *2011 IEEE 4th International Conference on Cloud Computing*. IEEE, 2011.

[117] Huang, Qiang, et al. "Power consumption of virtual machine live migration in clouds." *2011 Third International Conference on Communications and Mobile Computing*. IEEE, 2011.

[118] Beloglazov, Anton, and Rajkumar Buyya. "Energy efficient allocation of virtual machines in cloud data centers." *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE, 2010.

[119] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." *2009 Fifth International Joint Conference on INC, IMS and IDC*. IEEE, 2009.

[120] Cao, Zhibo, and Shoubin Dong. "An energy-aware heuristic framework for virtual machine consolidation in Cloud computing." *The Journal of Supercomputing* 69.1 (2014): 429-451.

[121] Gao, Yongqiang, et al. "Service level agreement based energy-efficient resource management in cloud data centers." *Computers & Electrical Engineering* 40.5 (2014): 1621-1633.

[122] Yousefipour, Amin, Amir Masoud Rahmani, and Mohsen Jahanshahi. "Energy and cost-aware virtual machine consolidation in cloud computing." *Software: Practice and Experience* 48.10 (2018): 1758-1774.

[123] Wang, Hui, and Huaglory Tianfield. "Energy-aware dynamic virtual machine consolidation for cloud datacenters." *IEEE Access* 6 (2018): 15259-15273.

[124] Ruan, Xiaojun, et al. "Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds." *Future Generation Computer Systems* 100 (2019): 380-394.

[125] Soltanshahi, Minoo, Reza Asemi, and Nazi Shafiei. "Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers." *Heliyon* 5.7 (2019): e02066.

[126] Mc Donnell, Nicola, Enda Howley, and Jim Duggan. "Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing." *Future Generation Computer Systems* 108 (2020): 288-301.

[127] Zhang, Peiyun, MengChu Zhou, and Xuelei Wang. "An Intelligent Optimization Method for Optimal Virtual Machine Allocation in Cloud Data Centers." *IEEE Transactions on Automation Science and Engineering* (2020).

[128] Hsieh, Sun-Yuan, et al. "Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers." *Journal of Parallel and Distributed Computing* 139 (2020): 99-109.

[129] Liu, Xialin, et al. "Virtual Machine Consolidation with Minimization of Migration Thrashing for Cloud Data Centers." *Mathematical Problems in Engineering* 2020 (2020).

[130] Jin, Shunfu, Xiuchen Qie, and Shanshan Hao. "Virtual machine allocation strategy in energy-efficient cloud data centres." *International Journal of Communication Networks and Distributed Systems* 22.2 (2019): 181-195.

[131] Saadi, Youssef, and Said El Kafhali. "Energy-efficient strategy for virtual machine consolidation in cloud environment." *Soft Computing* (2020): 1-15.

[132] Qie, Xiuchen, Shunfu Jin, and Wuyi Yue. "An energy-efficient strategy for virtual machine allocation over cloud data centers." *Journal of Network and Systems Management* 27.4 (2019): 860-882.

[133] Song, Fei, et al. "An optimization-based scheme for efficient virtual machine placement." *International Journal of Parallel Programming* 42.5 (2014): 853-872.

[134] Marotta, Antonio, et al. "A fast robust optimization-based heuristic for the deployment of green virtual network functions." *Journal of Network and Computer Applications* 95 (2017): 42-53.

International Journal of Electrical Engineering and Technology (IJEET)
Volume 11, Issue 6, August 2020, pp. 145-154, Article ID: IJEET_11_06_014
Available online at https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=6
ISSN Print: 0976-6545 and ISSN Online: 0976-6553
DOI: https://doi.org/10.34218/IJEET.11.6.2020.014

© **IAEME** Publication **Scopus** Indexed

AN ADAPTIVE VIRTUAL MACHINE SELECTION AND MIGRATION APPROACH FOR VM CONSOLIDATION IN CLOUD COMPUTING

P. Nithya
Research Scholar, Department of Computer Science, Bishop Heber College,
(Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India

Dr. L. Jayasimman
Assistant Professor, Department of Computer Application, Bishop Heber College,
(Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India

ABSTRACT

With its tremendous advantages, cloud-based computing often has a detrimental impact on the environment. Release of greenhouse gas and electricity from cloud data centres, which requires significant attention, is the most important issue. A popular approach for energy-efficient and optimum use of resources is virtual machine Consolidation (VM). Consolidation of virtual machine (VM) is one of the main mechanisms for creating a complex cloud resource management system which is energy efficient. The assumption is that migrating VMs into less Physical Machines will accomplish both optimization goals and maximise the usage of cloud server while also reducing Cloud data centre energy consumption. The best fit decrease approach (BFD) for VM placement is used in this proposed strategy. For VM migration an increased Greedy selection is suggested, based on the optimization algorithm Genetic Algorithm.

Key words: Cloud Computing, Virtual Machine Consolidation, Energy Efficient, Optimization, Greedy Selection, Genetic Algorithm, VM migration

Cite this Article: P. Nithya and L. Jayasimman, An Adaptive Virtual Machine Selection and Migration Approach for VM Consolidation in Cloud Computing, *International Journal of Electrical Engineering and Technology (IJEET),* 11(6), 2020, pp. 145-154.
https://iaeme.com/Home/issue/IJEET?Volume=11&Issue=6

1. INTRODUCTION

Cloud computing provides shared access to services on request without much contact between service providers. Cloud service providers provide their customers with infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [1][2]. These services support all web users directly / indirectly. The development of cloud data centres has a remarkable impact on the IT industry. These data centres, on the other hand, need massive power and high carbon footprints to run.

Infrastructure as a service (IaaS) has in recent years been very common in the field of cloud computing. Popular cloud providers including Rackspace and Amazon EC2 offer such virtual services over the internet to customers in various data centres. Because of the immense demand for cloud computing, there have been thousands of large cloud computing centres, leading to high power consumption [3]. The key cause of this high energy usage is the inefficient use of these cloud resources, not the power consumption of vast amounts of hardware. Therefore, it is very important from an energy management perspective to try to establish adaptive strategies to dynamically change each source in cloud data centres.

The use of virtualization technology [2], improved efficiency of resource use through the sharing of a physical machine (PM) between multiple digital machines (VMs), is one of the solutions to improve energy efficiency. VMs can be migrated from one machine to another through live migration [4] from the virtual machine to dynamically managed resource, while the services provided by that machine are kept available. This technique allows the playing of VMs with various energy efficiency purposes [5] or load balance [6] to be configured dynamically at all times, depending on resource use.

VM consolidation [7], which aims to transfer VM into a lower number of PMs to maximise the usage of the resources of cloud data centres, is a key mechanism for increasing energy efficiency in the cloud data centres. In some cases, however, too many VMs can be bundled in one PM; because VMs on one PM often have similar physical resources, they can lead to poor quality of service. Thus, consolidation strategies for the VM should ensure comprehensive reliability of the QoS that is frequently specified through service level agreements [8] and seek to change VM 's placement dynamically. In addition, each phase of the VM consolidation should take into account power consumption [9] and VM migration costs [10].

2. RELATED WORKS

Ruan, Xiaojun, et al [11] Proposed a 'PPRGear' strategy based on the use-level sampling with distinctive PPRs measured as the number of server-side Java activities performed during a certain time span separated by the average active power consumption over the same period. The strategy is based on the sampling of the user levels. We also present a system for the allocation and relocation of virtual machines that leverages PPR for different host types.

Soltanshahi, Minoo, Reza Asemi, and Nazi Shafiei [12] In cloud data centres, the Krill Herd algorithm, which is the newest collective intelligence algorithm recently implemented, has been proposed to assign virtual machines in physical hosts.

Mc Donnell, Nicola, Enda Howley, and Jim Duggan [13] The new multi-agent system for the development of decentralised collaboration strategies is being put forward by Gossip Contracts (GC). GC is based on the Contract Net Protocols and Gossip's inspiration. Using GC, we developed a DVMC strategy based on the GC and compared it to two other common strategies: Sercon and the distributed strategy of ecoCloud. One comparison experiment uses Google cluster-use traces; a dataset from a Google data centre with real-time use data. The GC-based approach works better against SLA infringements and resembles or outperforms other energy consumption strategies.

Zhang, Peiyun, MengChu Zhou, and Xuelei Wang [14] Developed an optimised VM assignment model and developed an improved DD (IDE) approach in the sense of an array of user tasks for solving this optimization problem. The authors have compared the proposed method with various methods, including round-robin (RR), min-min and differential growth.

Hsieh, Sun-Yuan, et al [15] The solution proposed to consolidate the VM takes into account current and potential resource use, by UP-POD and UP-PUD. A Gray-Markov model is used to forecast the potential usage of resources accurately.

Liu, Xialin, et al [16] proposed a Dynamic Consolidation with Minimization of Migrating Thrashing (DCMMT) strategy that priority high capacity VMs, considerably reducing migration thrashing and the number of migrations in order to ensure service-level agreement (SLA) because VMs are likely to be migrated instead of migrating to migration.

Jin, Shunfu, Xiuchen Qie, and Shanshan Hao [17] A two-dimensional Markov chain has been developed to assess mathematically energy saving and response efficiency. Numerical analysis and simulation experiments demonstrate that our proposed strategy to assign VM can efficiently reduce energy consumption and guarantee reaction efficiency.

Saadi, Youssef, and Said El Kafhali [18] proposed an Energy-Efficient Strategy (EES) to consolidate cloud virtual machinery with a goal of reducing energy consumption and fulfilling more high throughput tasks. The performance-to-power ratio is used in our proposal to set the highest threshold for overload detection. Moreover, EES takes into account the total workload utilisation of the data centre to determine lower levels that can reduce the number of virtual machine migrations.

Qie, Xiuchen, Shunfu Jin, and Wuyi Yue [19] Proposed an asynchronous multi-sleep and adaptive work migration solution for energy-efficient virtual machine (VM). The virtual-cluster VMs are split into two modules, Module I and Module II. Module I VMs are always awake, and Module II VMs are autonomous if possible. VMs in module I are always wakeful. Therefore, a queuing model is developed to capture the working theory of the proposed strategy with a partial, asynchronous multiple vacation. Using the matrix-geometric solution approach, performance measurements are mathematically derived for the average response time and energy saving rate.

3. GENETIC ALGORITHM OPTIMIZATION

Genetic algorithms [20] employ metaphor from biology and genetics to iteratively evolve a population of initial individuals to a population of high-quality individuals, where each individual represents a solution of the problem to be solved and is compose of a fixed number of genes. The number of possible values of each gene is called the cardinality of the gene. Each individual is called as chromosome. The set of chromosomes forms population.

Functioning of genetic algorithm starts with randomly generated population of individuals. By various generations these population evolved and individuals' quality gets improved. In every generation, three basic operators of genetic algorithm i.e. selection, crossover and mutation are applied to each individual. Crossover means exchanging the genes between two chromosomes while mutation means random changing of a value of a randomly chosen gene of a chromosome. These individuals are representation of the problem required to be solved. Different positions of each individual can be encoded as bits, characters and numbers [21].

Here, the numbers of best-fit individuals are selected. For this user defined fitness function is used. Fitness function is used to measure quality of each chromosome. Remaining individuals are paired and by process of crossover, new offspring is produced by partially exchanging their genes. When genetic algorithm is used for problem solving, three factors will have impact on the effectiveness of the algorithm, they are [20]:

a. The selection of fitness function
b. The representation of individuals and
c. The values of the genetic parameters

4. OPTIMIZATION BASED VIRTUAL MACHINE CONSOLIDATION

Dynamic consolidation of Virtual Machines (VMs), using live migration of the VMs and switching idle servers to sleep mode or shutdown, optimizes the energy consumption. The normal host detected through host detection technique is considered for allocating the VM, VM migration for VM consolidation framework. In this contribution, an adaptive VMs migration selecting method and heuristic algorithm for dynamic consolidation of VMs to be proposed based on the analysis of the historical data.

This proposed Adaptive VM consolidation strategy includes the coding, fitness function, selection, crossover, mutation and greedy selection. The VM placement is carried out with a bin packing problem with variable bin sizes, where the bin sizes are represent by the physical nodes; items are the VMs that have to be allocated; CPU capabilities, and power consumption by the nodes. For VM placement, Best Fit Decreasing (BFD) method is utilized in this proposed strategy. For VM migration, an enhanced Optimization algorithm called Greedy Selection based Genetic algorithm is proposed. The following procedures gives the VM placement strategy using Best Fit Decreasing (BFD) algorithm. The host list for the VM placement is taken from the host detection method.

Input: Host List (hl) and Virtual Machine List (vml)
Output: Allocation of Virtual Machines (VMs)
Step 1: for each vm in vml do
    Step 1.1: $minPower \leftarrow Max$
    Step 1.2: $allocatedHost \leftarrow NULL$
    Step 1.3: foreach host in hl do
        Step 1.3.1: if host has enough resource for vm then
        Step 1.3.2: $power \leftarrow estimatePower(host, vm)$
        Step 1.3.3: if $power < minPower$ then
            $allocatedHost \leftarrow host$
            $minPower \leftarrow power$
    Step 1.4: if $allocatedHost \neq NULL$ then
        Step 1.4.1: allocate vm to allocateHost
Step 2: return allocation

In this proposed Adaptive approach, an enhanced Genetic algorithm is proposed for VM Migration. In this proposed Adaptive VM consolidation approach, a Greedy Search algorithm is used to find the optimized VM with the help of host detection techniques. In this proposed VM consolidation strategy, a Greedy Selection algorithm is utilized in the children coding after each mutation and crossover operator. In the proposed VM migration approach, the dynamic threshold methodology is carried out. The upper threshold for the VM migration is obtained by using proposed. For this approach, the host list and VM selected list is given as the input.

- Randomly generate several codes to form a population.
- Using binary code to coding all tasks.
- Calculate the fitness value of all coding in the population.

- Select several pairs of codes to do the crossover operation to form new population (the total number of population remains unchanged);
- Perform mutation operation through selection operator again.
- Make greedy choices for all codes of the newly formed population. Form a whole generation of new population.
- Repeat the process until a stable optimal individual fitness is found, that is, the optimal threshold.

Input: Host list (hl), Virtual Machine List (vml)
Output: VM migration list
Step 1: vml.sortDecreasingUtilization()
Step 2: for each h in hl do
    Step 2.1: $hutil \leftarrow h.util()$
        Step 2.1.1: *Initialize the population for GA.*
        Step 2.1.2: *Binary coding of the tasks assigned in the VMs are carried in this step.*
        Step 2.1.3: *The fitness value of each individual is computed.*
        Step 2.1.4: *Roulette Wheel method is used for the selection of the individual based on the probability, which is reflected by the fitness value of the population.*
        Step 2.1.5: *Two point crossover is calculated with the two individuals.*
        Step 2.1.6: *Mutation operator is applied to improve the local search capability of GA algorithm and to maintain the diversity of the population.*
        Step 2.1.7: *The greedy selection in this study occurred in the children coding after each mutation and crossover operator.*
    Step 2.2: $bestFitUtil \leftarrow MAX$
    Step 2.3: while $hutil > h.Th_{upper}()$ do
        Step 2.3.1: foreach vm in vml do
            If $vm.util() > hutil - h.Th_{upper}()$ then
                $t \leftarrow vm.util() - hutil + h.Th_{upper}()$
                If $t < bestFitUtil$ then
                    $bestFitUtil \leftarrow t$
                    $bestFitVm \leftarrow vm$
            Else
                If $bestFitUtil = MAX$ then
                    $bestFitVm \leftarrow vm$
                Break
        Step 2.3.2: $hutil \leftarrow hutil - bestFitVm.util()$
        Step 2.3.3: migration.List.add(h.getvml())
        Step 2.3.4: vmList.remove(h.getvml())
    Step 3.1: if $hutil < Th_{low}()$ then
        Step 3.1.1: migrationList.add(h.getvml())
        Step 3.1.2: vml.remove(h.getvml())
Step 4: return migrationlist

5. RESULT AND DISCUSSION

In the research work, The data-center used in this work is considered which is also included in CloudSim.

The data-center has 800 hosts from two server models (400 hosts from each server type) and four types of VMs. The CPU capacity of the VM instances is given in millions of instructions per second (MIPS). The following table 1a and table 1b gives the two types of host characteristic and the VM types used in this experiment. The Energy Consumption (kWh), SLA violations, Performance Degradation due to Migration (PDM) are considered as the performance metrics.

Table 1a Two types of Host characteristics

Type	Number of Host	Storage	Number of Cores	RAM	Bandwidth	MIPS
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	1860
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	2660

Table 1b Used VM types characteristics

Type of VM	Number of VMs	RAM	MIPS	Storage
VM1	1	613	500	2.5
VM2	1	1740	1000	2.5
VM3	1	1740	2000	2.5
VM4	1	2500	2500	2.5

Table 2a depicts the Energy consumption (kWh) of the Proposed VM Consolidation Strategy, Cultural Algorithm (CA), Artificial Bee Colony (ABC) optimization at number of hosts=400 and number of VMs = 100. From the table 2a, it is clear that the proposed VM consolidation strategy consumes less energy when increasing the number of tasks.

Table 2a Energy consumption (kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh)		
	Proposed VM consolidation Strategy	CA	ABC
100	3.94	9.61	11.02
200	4.48	10.32	12.5
300	4.96	11.12	13.7
400	5.51	11.94	14.3
500	5.95	12.32	15.4
600	6.38	13.87	16.3
700	6.99	14.31	17.5
800	7.28	15.52	18.6
900	7.84	16.55	19.1

Table 2b gives the SLA violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 100. From the table 2b, it is clear that the proposed VM consolidation strategy consumes less energy as well as it reduces the SLA violations than the existing optimization techniques.

Table 2b Service Level Agreement (SLA) Violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 100

Number of Tasks	SLA Violation (%)		
	Proposed VM consolidation Strategy	CA	ABC
100	0.002	0.234	0.865
200	0.165	0.496	1.103
300	0.316	0.812	2.232
400	0.498	1.624	3.275
500	0.501	2.238	4.357
600	0.563	2.781	5.531
700	0.599	3.794	6.642
800	0.621	4.236	7.653
900	0.659	5.626	8.4751

Table 3a depicts the energy consumption (in kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 200. Table 3b gives the SLA violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 200. From the table 3a and table 3b, it is clear that the proposed VM consolidation strategy consumes less energy when increasing the number of tasks, and it reduces the SLA violations also.

Table 3a Energy consumption (kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh)		
	Proposed VM consolidation Strategy	CA	ABC
100	4.54	8.85	13.4
200	4.98	9.51	14.9
300	5.46	10.23	15.5
400	5.93	11.94	16.4
500	6.37	12.28	17.1
600	6.88	13.97	18.3
700	7.43	14.36	19.7
800	7.96	15.85	20.2
900	8.54	16.32	20.9

Table 3b Service Level Agreement (SLA) Violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=400 and number of VMs = 200

Number of Tasks	SLA Violation (%)		
	Proposed VM consolidation Strategy	CA	ABC
100	0.045	0.561	0.974
200	0.192	1.265	1.814
300	0.424	1.818	2.411
400	0.506	2.314	3.583
500	0.613	3.179	4.549
600	0.741	4.671	5.743
700	0.871	5.963	6.831
800	0.913	6.524	7.845
900	1.148	7.815	8.687

Table 4a depicts the energy consumption (in kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 100. Table 4b gives the SLA violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 100. From the table 4a and table 4b, it is clear that the proposed VM consolidation strategy consumes less energy when increasing the number of tasks, and it reduces the SLA violations also.

Table 4a Energy consumption (kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh)		
	Proposed VM consolidation Strategy	CA	ABC
100	6.36	15.22	16.8
200	7.16	16.61	17.2
300	8.24	17.13	18.6
400	9.71	18.32	19.9
500	10.19	19.42	20.4
600	11.60	20.83	21.3
700	12.41	21.62	22.5
800	13.84	22.54	23.7
900	14.32	23.19	24.8

Table 4b Service Level Agreement (SLA) Violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 100

Number of Tasks	SLA Violation (%)		
	Proposed VM consolidation Strategy	CA	ABC
100	0.452	1.452	1.863
200	0.631	2.753	2.925
300	0.827	3.513	3.532
400	1.078	4.424	4.772
500	1.392	5.568	5.738
600	1.796	6.852	6.931
700	2.143	7.619	7.622
800	2.635	8.735	8.934
900	2.968	9.753	9.876

Table 5a depicts the energy consumption (in kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 200. Table 5b gives the SLA violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 200. From the table 5a and table 5b, it is clear that the proposed VM consolidation strategy consumes less energy when increasing the number of tasks, and it reduces the SLA violations also.

Table 5a Energy consumption (kWh) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh)		
	Proposed VM consolidation Strategy	CA	ABC
100	7.44	17.2	19.6
200	8.24	18.8	20.3
300	9.12	19.43	21.7
400	10.52	20.19	22.6
500	11.27	21.33	23.5
600	12.42	22.53	24.4
700	13.22	23.34	25.9
800	14.62	24.82	26.7
900	15.14	25.55	27.9

Table 5b Service Level Agreement (SLA) Violations (in %) of the Proposed VM Consolidation Strategy, CA and ABC at number of hosts=800 and number of VMs = 200

Number of Tasks	SLA Violation (%)		
	Proposed VM consolidation Strategy	CA	ABC
100	0.671	2.241	2.641
200	0.813	3.651	3.713
300	1.245	4.723	4.313
400	1.866	5.628	5.551
500	2.404	6.782	6.916
600	2.914	7.623	7.852
700	3.331	8.582	8.813
800	3.813	9.524	9.726
900	4.146	10.215	10.964

6. CONCLUSION

In cloud computing, users may utilize hundreds of thousands of virtualized resources and it is impossible for everyone to allocate each task manually. Due to commercialization and virtualization, cloud computing left the task scheduling complexity to virtual machine layer by utilizing resources virtually. Hence to assign the resources to each task efficiently and effectively, scheduling plays an important role in cloud computing. In the cloud platform, dynamic change of VM availabilities makes satisfying the task QoS requirement difficult. To improve task scheduling capacities of VMs in the cloud platform and satisfy task QoS requirement, a task scheduling algorithm based on VM availability awareness was proposed in this study to solve the matching problem between available task processing capacities of VMs and task QoS requirement and realized workload balancing of servers in the cloud platform

REFERENCES

[1] Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." *Journal of network and computer applications* 34.1 (2011): 1-11.

[2] Gorelik, Eugene. *Cloud computing models*. Diss. Massachusetts Institute of Technology, 2013.

[3] Lin, Ching-Chi, Pangfeng Liu, and Jan-Jan Wu. "Energy-aware virtual machine dynamic provision and scheduling for cloud computing." *2011 IEEE 4th International Conference on Cloud Computing*. IEEE, 2011.

[4] Huang, Qiang, et al. "Power consumption of virtual machine live migration in clouds." *2011 Third International Conference on Communications and Mobile Computing*. IEEE, 2011.

[5] Beloglazov, Anton, and Rajkumar Buyya. "Energy efficient allocation of virtual machines in cloud data centers." *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE, 2010.

[6] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." *2009 Fifth International Joint Conference on INC, IMS and IDC*. IEEE, 2009.

[7] Cao, Zhibo, and Shoubin Dong. "An energy-aware heuristic framework for virtual machine consolidation in Cloud computing." *The Journal of Supercomputing* 69.1 (2014): 429-451.

[8] Gao, Yongqiang, et al. "Service level agreement based energy-efficient resource management in cloud data centers." *Computers & Electrical Engineering* 40.5 (2014): 1621-1633.

[9] Cao, Zhibo, and Shoubin Dong. "An energy-aware heuristic framework for virtual machine consolidation in Cloud computing." *The Journal of Supercomputing* 69.1 (2014): 429-451.

[10] Wang, Hui, and Huaglory Tianfield. "Energy-aware dynamic virtual machine consolidation for cloud datacenters." *IEEE Access* 6 (2018): 15259-15273.

[11] Ruan, Xiaojun, et al. "Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds." *Future Generation Computer Systems* 100 (2019): 380-394.

[12] Soltanshahi, Minoo, Reza Asemi, and Nazi Shafiei. "Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers." *Heliyon* 5.7 (2019): e02066.

[13] Mc Donnell, Nicola, Enda Howley, and Jim Duggan. "Dynamic virtual machine consolidation using a multi-agent system to optimise energy efficiency in cloud computing." *Future Generation Computer Systems* 108 (2020): 288-301.

[14] Zhang, Peiyun, MengChu Zhou, and Xuelei Wang. "An Intelligent Optimization Method for Optimal Virtual Machine Allocation in Cloud Data Centers." *IEEE Transactions on Automation Science and Engineering* (2020).

[15] Hsieh, Sun-Yuan, et al. "Utilization-prediction-aware virtual machine consolidation approach for energy-efficient cloud data centers." *Journal of Parallel and Distributed Computing* 139 (2020): 99-109.

[16] Liu, Xialin, et al. "Virtual Machine Consolidation with Minimization of Migration Thrashing for Cloud Data Centers." *Mathematical Problems in Engineering* 2020 (2020).

[17] Jin, Shunfu, Xiuchen Qie, and Shanshan Hao. "Virtual machine allocation strategy in energy-efficient cloud data centres." *International Journal of Communication Networks and Distributed Systems* 22.2 (2019): 181-195.

[18] Saadi, Youssef, and Said El Kafhali. "Energy-efficient strategy for virtual machine consolidation in cloud environment." *Soft Computing* (2020): 1-15.

[19] Qie, Xiuchen, Shunfu Jin, and Wuyi Yue. "An energy-efficient strategy for virtual machine allocation over cloud data centers." *Journal of Network and Systems Management* 27.4 (2019): 860-882.

[20] Song, Fei, et al. "An optimization-based scheme for efficient virtual machine placement." *International Journal of Parallel Programming* 42.5 (2014): 853-872.

[21] Marotta, Antonio, et al. "A fast robust optimization-based heuristic for the deployment of green virtual network functions." *Journal of Network and Computer Applications* 95 (2017): 42-53.

International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 11, Issue 9, September 2020, pp. 1245-1254 Article ID: IJARET_11_09_123
Available online at https://iaeme.com/Home/issue/IJARET?Volume=11&Issue=9
ISSN Print: 0976-6480 and ISSN Online: 0976-6499
DOI: 10.34218/IJARET.11.9.2020.123

© **IAEME** Publication **Scopus** Indexed

HOST DETECTION TECHNIQUES FOR EFFICIENT VIRTUAL MACHINE CONSOLIDATION IN CLOUD COMPUTING

P. Nithya
Research Scholar, Department of Computer Science, Bishop Heber College, (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India

Dr. L. Jayasimman
Assistant Professor, Department of Computer Application, Bishop Heber College, (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India

ABSTRACT

Cloud computing data centers are growing rapidly in both number and capacity to meet the increasing demands for highly-responsive computing and massive storage. Such data centers consume enormous amounts of electrical energy resulting in high operating costs and carbon dioxide emissions. The reason for this extremely high energy consumption is not just the quantity of computing resources and the power inefficiency of hardware, but rather lies in the inefficient usage of these resources. VM consolidation involves live migration of VMs hence the capability of transferring a VM between physical servers with a close to zero down time. It is an effective way to improve the utilization of resources and increase energy efficiency in cloud data centers. VM consolidation consists of host overload/underload detection, VM selection and VM placement. Most of the current VM consolidation approaches apply either heuristic based techniques, such as static utilization thresholds, decision-making based on statistical analysis of historical data; or simply periodic adaptation of the VM allocation. Most of those algorithms rely on CPU utilization only for host overload detection. In this paper, optimization techniques-based, Median Absolute Derivation, Local Regression techniques for Host overload and underload detection. The performance analysis of those techniques can be compared with SLA, Energy Consumption by varying number of hosts and tasks.

Key words: Cloud Computing, Host Detection, Overload/Underload, Virtual Machine, Optimization Techniques, Local Regression, Median Absolute Derivation

Cite this Article: P. Nithya and L. Jayasimman, Host Detection Techniques for Efficient Virtual Machine Consolidation in Cloud Computing, *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 11(9), 2020, pp. 1245-1254.
https://iaeme.com/Home/issue/IJARET?Volume=11&Issue=9

1. INTRODUCTION

Cloud computing can be considered as a computing paradigm with many exciting features like on-demand computing resources, elastic scaling, elimination of up-front capital and operational expenses, and establishing a pay-as-you-go business model for computing and information technology services [1]. Cloud computing data centers consume enormous amounts of electrical energy resulting in high operating costs and carbon dioxide emissions. The reason for high energy consumption is not just the quantity of computing resources and the power inefficiency of hardware, but rather lies in the inefficient usage of these resources. One way to address the energy inefficiency is to leverage the capabilities of the virtualization technology. The reduction in energy consumption can be achieved by switching idle nodes to low power modes (i.e., sleep, hibernation), thus eliminating the idle power consumption. Moreover, by using live migration the VMs can be dynamically consolidated to the minimal number of physical nodes according to their current resource requirements [2].

2. BACKGROUND STUDY ON VM CONSOLIDATION

All Dynamic Virtual Machine (VM) consolidation is a promising approach for reducing energy consumption by dynamically adjusting the number of active machines to match resource demands [3]. To address this problem, most of the current approaches apply Regression based algorithms that is based on estimation of future CPU utilization. The limitation of these approaches is that they lead to sub-optimal results and do not allow the administrator to explicitly set a QoS goal. The static utilization threshold is a simple method since it is based on a fixed CPU utilization threshold but it is unsuitable for dynamic environment [4]. Adaptive utilization-based algorithms are suitable for dynamic environment but give poor prediction of host overloading. Regression based algorithms give better predictions of host overloading since they are based on estimation of future CPU utilization but they are complex. Once a host overload is detected, the next step is to select VMs to offload the host to avoid performance degradation. Once a host overload is detected, the next step is to select VMs to offload the host to avoid performance degradation.

Cloud computing infrastructure providers different types of resources to be available on demand. The process of allocating these resources to a set of VMs is called resource management. Resource management aims to meet the objectives of both of the cloud provider and the customer. Some of the big challenges faced by the resource management process are the growing need for resources during running time, the heterogeneity in the needed resources, and the interdependencies between these resources. Infrastructure scaling, virtualization, and VM migration are the main technologies that enable the resource management process [5][6]. Infrastructure scaling is the ability to increase or decrease the VM's allocated resources at run time as needed. Virtualization technology is the process of creating a virtual image of the physical resources of a host or a set of hosts to be used as a single machine by a user or multiple users. There are many benefits that can be obtained by using virtualization, including the ability to run multiple operating systems at the same time on the same host, which increases the utilization of resources and increases the number of customers who can use these resources [7]. The virtualization model consists of several layers. The layer that provides virtualization is called the hypervisor or the virtual machine monitor (VMM) layer. The hypervisor is a software layer that exists in a physical machine between the hardware and the guest operating systems of VMs running on the physical machine. It emulates the underlying hardware layer for each VM. The hypervisor is also responsible for creating, running and managing the execution of the guest VMs [8]. By using the virtualization, a property of direct communication between the VM's operating system and the hardware no longer exists and the hypervisor becomes the only way through which the VM's operating system can access the hardware resources [9]. There are many benefits that can be obtained when using virtualization. The main benefit is the ability to run multiple operating systems at the same time on the same physical machine which will increase the utilization of resources and increase the number of customers who use these resources Virtualization usually supports a migration mechanism of VMs to increase resource utilization. The migration is a process of copying the memory pages and the state of a VM from a source physical machine to a destination physical machine [10]. The VM then can continue executing from the its saved state. The essence of the migration policy is to migrate the VMs from overloaded hosts to the underutilized ones. There are two main steps to perform the migration process. The first one is to determine which host is overloaded or underloaded. The second step is to choose which VMs running on this host that need to be migrated. Usually, a host's overloaded/underloaded/normal status is determined based on its resource utilization level. Let us take CPU utilization for example. Upper and lower utilization thresholds are determining for each host and the CPU utilization of the host should stay between these thresholds. If the CPU utilization drops below the lower threshold, all VMs should be migrated and the host directly go to sleep or power saving mode in order to save power consumption. On the other hand, some of the VMs should be migrated if the CPU utilization increases beyond the upper threshold in order to prevent SLA violations [11] [12]. For underloaded detection, a specific utilization threshold, such as 10% can be used to determine that a host is underloaded. Many techniques are available in the literature to predict the host overloaded state and to select a specific VM to be migrated if needed.

3. HOST DETECTION TECHNIQUES

3.1. Local Regression Technique

The regression model [13][14] is used to find out the best trend line for a set of observed data. The trend line is calculated by equation (1), and it is fitted to the observed data using the least-squares method shown in equation (2). In other words, the least-squares method finds the optimal value for the slope of the line (b1) and the intercept (b0) by minimizing the residual sum of squares (RSS).

$$y = b_0 + b_1 * x \qquad (1)$$

$$RSS = \sum_{i=1}^{m} (y_i - b_1 * x_i - b_0)^2 \qquad (2)$$

The slope (b_1) and the intercept b_0 are derived by taking partial derivatives of RSS with respect to b_1 and b_0, and they are defined by using equations (3) and (4) respectively.

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad (3)$$

$$b_0 = \bar{y} - b_1 * \bar{x} \qquad (4)$$

Where n is the length of the observation history of host utilization, $\bar{x}\ and\ \bar{y}$ are the names of the x_i and y_i observations variables respectively.

3.2. Median Absolute Derivation Technique

Median Absolute Deviation (MAD) [15] [16] is a statistical model that is used for detecting overloaded host by determining an upper and lower utilization threshold for all VMs inside the host. This model calculates the utilization prediction for a host. The host is considered overloaded if the predicted utilization value exceeds the upper threshold. In this case, some VMs should be migrated to another host in order to reduce the utilization to prevent a possible SLA violation. If the predicted utilization of the host is lower than the lower utilization threshold, all VMs running on the host should be migrated and the host is switched to the sleep mode in order to reduce the number of active hosts which leads to less energy consumption.

MAD gets the median of the absolute derivations from the median of a univariate data set $X_1, X_2, \dots, X_n$ as shown in the following equation (5):

$$MAD = median\left(\left|X_i - median_j\left(X_j\right)\right|\right) \quad (5)$$

While the upper utilization threshold (T_u) is computed according to the equation (6), where s is called the safety parameter which controls how aggressively the system consolidates VMs. Lower s values produce larger threshold which causes lower energy consumption and higher SLA violations as a result of the consolidation process.

$$T_u = 1 - s.MAD \quad (6)$$

3.3. Artificial Bee Colony Optimization

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga [17] [18]. It is the method to find the appropriate value. This method is inspired by the foraging behavior of honey bees. In ABC model, there are three kinds of honey bee to search food sources, which include scout bees search for food source randomly, employed bees search around the food source and share food information to the onlooker bees, and onlooker bees calculate the fitness and select the best food source. In the nature, bees can extend themselves over long distances in multiple directions. After scout bees find the food source and return to the hive, they compare the quality of food source and go to the dance floor to perform a dance known as “waggle dance”. The waggle dance is the communication of bees to shares the information about direction of the food source, distance from the hive, and the nectar amount of the food source. While sharing information, bees evaluate the nectar quality and energy waste. After sharing information on the dance floor, onlooker bees select the best food source and then scout bees will return to the food source to bring nectar back to the hive.

Begin

Step 1: Initialize the population of the scout bees, generate randomly scout bees into the food sources and calculate the fitness values.

Step 2: Repeat

Step 3: Each of the employ bees search around the food sources and update the new fitness, if the new fitness is better than the old values.

Step 4: Select employed bees and recruit onlooker bees to search around the food sources and calculate on their fitness value.

Step 5: Choose the onlooker bees with have the best fitness value.

Step 6: Send scout bees into the food sources to discover new food sources.

Step 7: Until (Stopping criterion is not met)

Step 8: End

3.4. Ant Colony Optimization

The ant colony optimization [19] [20] is a heuristic bionic algorithm proposed by Italian scholar M.Dorigo, which is mainly used to solve the optimal solution of combinatorial optimization problem. And cloud computing task scheduling problem has been proved as NP-Hard problem, so the ant colony algorithm used in cloud computing task scheduling can greatly improve the scheduling efficiency. In this paper, based on the study of the standard ant colony algorithm, it is applied to the cloud computing task scheduling to re-establish the model. In the algorithm, each ant looks for the appropriate virtual machine for the corresponding task. The ants finally find an optimal matching scheme according to the optimized target. The ants find the optimal solution in parallel, and communicate with each other through information. We adjust the pheromone on the task with the virtual machine path, which affects the selection of the ant for the other ant and the next iteration for the ant.

Step 1: Initialize the pheromone. Set the maximum number of iterations, the pheromone energetic factor, the expected heuristic factor, the volatile factor, and the number of ants.

Step 2: Place all ants at starting VMs randomly.

Step 3: Each ant calculates the probability of the current task selected on each virtual machine in the set of optional virtual machines. And then the ant chooses the matching *VM* for the current task according to the roulette method. And then add the selected *VM* to the taboo table.

Step 4: When an ant completes a solution, update the pheromone on the matching scheme path found by the ant. Compare with the previous optimal solution and update the optimal solution.

Step 5: If all the ants end their tour, calculate the global optimal solution and update the pheromone on the optimal solution path. otherwise, repeat Step 3.

Step 6: If the current number of iterations is less than the maximum number of iterations, Clear taboo table and return to Step 2. Otherwise, end the iteration and output the best solution.

4. RESULT AND DISCUSSION

4.1. Simulation Setup

The data-center used in this work is considered from [21] which is also included in CloudSim. The data-center has 800 hosts from two server models (400 hosts from each server type) and four types of VMs. The CPU capacity of the VM instances is given in millions of instructions per second (MIPS). Table 1 gives the two types of host characteristic used in this experiment. Table 2 gives the VM types used in this research work. The Energy Consumption (kWh) is considered as the performance metrics.

Table 1 Two type of host characteristics used in this work

Type	Number of Host	Storage	Number of Cores	RAM	Bandwidth	MIPS
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	1860
HP ProLiant ML 110G4	400	1GB	2	4096	1GB	2660

Table 2 Used VM types characteristics in this work

Type of VM	Number of Cores	RAM	MIPS	Storage
VM1	1	613	500	2.5
VM2	1	1740	1000	2.5
VM3	1	1740	2000	2.5
VM4	1	2500	2500	2.5

Table 3 gives the energy consumption (kWh) by the Local Regression (LR), Median Absolute Derivation (MAD), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) with varying number of tasks for number of hosts=400 and number of VMs=100. Figure 1 depicts the graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=400 and number of VMs =100. From the table 3 and figure 1, it is clear that the ABC based host detection takes less energy when compared with other host detection techniques like LR, ACO and MAD at number of hosts=400 and number of VMs = 100.

Table 3 Energy consumption (kWh) of the LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh) by Host detection techniques			
	ABC	**ACO**	**LR**	**MAD**
100	11.02	13.5	14.7	15.6
200	12.5	15.3	16.8	17.5
300	13.7	16.9	17.6	18.1
400	14.3	17.8	18.2	19.2
500	15.4	18.4	19.3	20.3
600	16.3	19.6	20.4	21.5
700	17.5	20.7	21.2	22.3
800	18.6	21.2	22.6	23.4
900	19.1	22.7	23.9	24.7

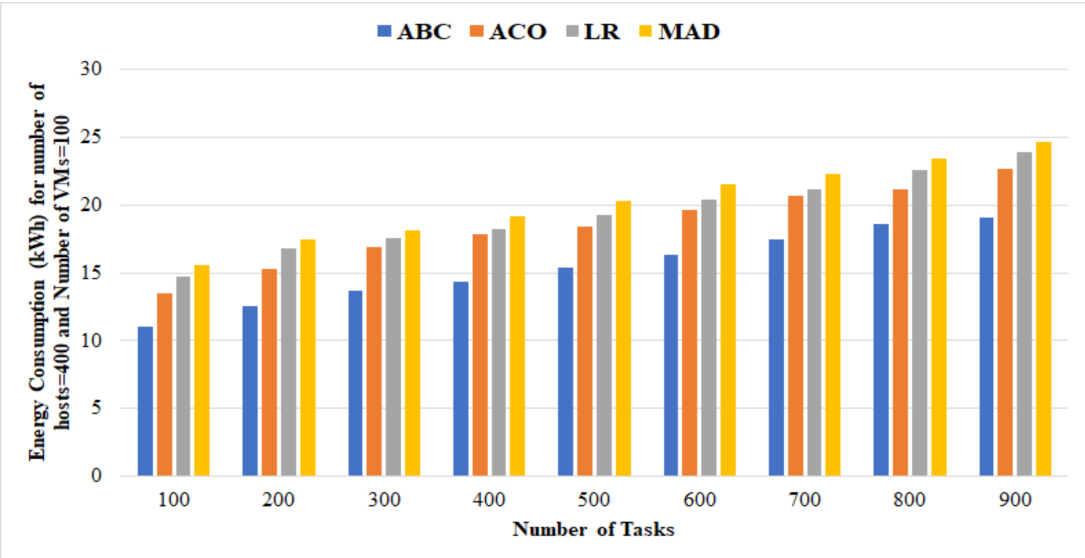

Figure 1 Graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=400 and number of VMs =100

Table 4 gives the energy consumption (kWh) by the Local Regression (LR), Median Absolute Derivation (MAD), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) with varying number of tasks for number of hosts=400 and number of VMs=200. Figure 2 depicts the graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=400 and number of VMs =200. From the table 4 and figure 2, it is clear that the ABC based host detection takes less energy when compared with other host detection techniques like LR, ACO and MAD at number of hosts=400 and number of VMs = 200.

Table 4 Energy consumption (kWh) of the LR, MAD, ABC, ACO at number of hosts=400 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh) by Host detection techniques			
	ABC	ACO	LR	MAD
100	13.4	15.8	15.9	16.4
200	14.9	16.7	16.8	17.5
300	15.5	17.3	17.9	18.2
400	16.4	18.2	18.6	19.1
500	17.1	19.6	20.1	20.8
600	18.3	20.4	21.4	21.9
700	19.7	21.3	22.7	23.6
800	20.2	22.5	23.4	24.8
900	20.9	23.9	24.7	25.4

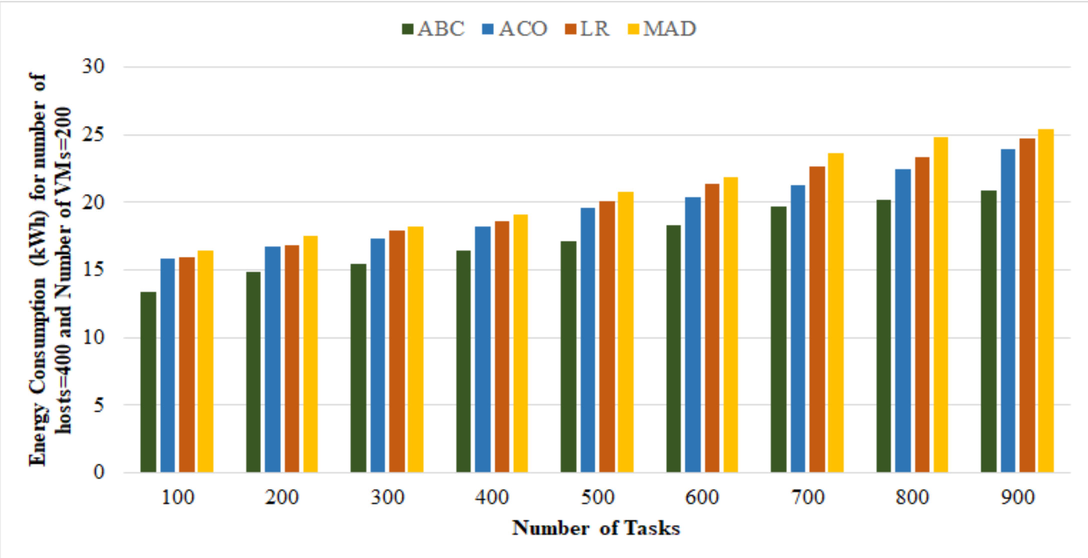

Figure 2 Graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=400 and number of VMs =200

Table 5 gives the energy consumption (kWh) by the Local Regression (LR), Median Absolute Derivation (MAD), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) with varying number of tasks for number of hosts=800 and number of VMs=100. Figure 3 depicts the graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=800 and number of VMs =100. From the table 5 and figure 3, it is clear that the ABC based host detection takes less energy when compared with other host detection techniques like LR, ACO and MAD at number of hosts=800 and number of VMs = 100.

Table 5 Energy consumption (kWh) of the LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 100

Number of Tasks	Energy Consumption (kWh) by Host detection techniques			
	ABC	ACO	LR	MAD
100	16.8	21.5	22.3	23.4
200	17.2	22.7	23.4	24.6
300	18.6	23.4	24.6	25.7
400	19.9	24.9	25.8	26.6
500	20.4	25.6	26.9	27.8
600	21.3	26.7	27.1	28.3
700	22.5	27.1	28.4	29.5
800	23.7	28.4	29.3	30.2
900	24.8	29.6	30.7	31.3

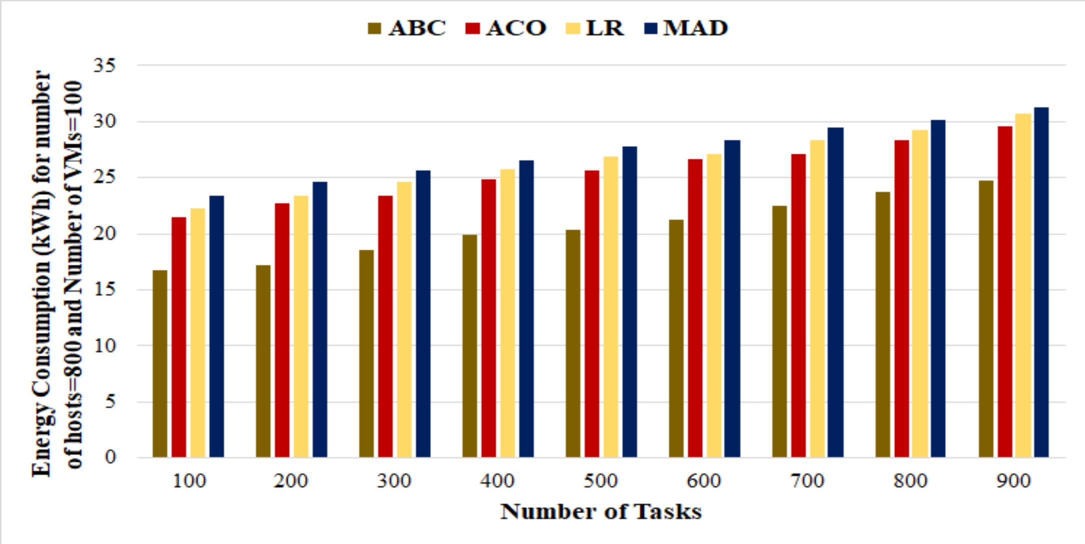

Figure 3 Graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=800 and number of VMs =100

Table 6 Energy consumption (kWh) of the LR, MAD, ABC, ACO at number of hosts=800 and number of VMs = 200

Number of Tasks	Energy Consumption (kWh) by Host detection techniques			
	ABC	ACO	LR	MAD
100	19.6	28.5	29.6	30.3
200	20.3	29.4	30.4	31.8
300	21.7	30.6	31.7	32.5
400	22.6	31.9	32.4	33.7
500	23.5	32.4	33.8	34.5
600	24.4	33.7	34.1	35.9
700	25.9	34.8	35.5	36.4
800	26.7	35.6	36.1	37.3
900	27.9	36.8	37.9	38.2

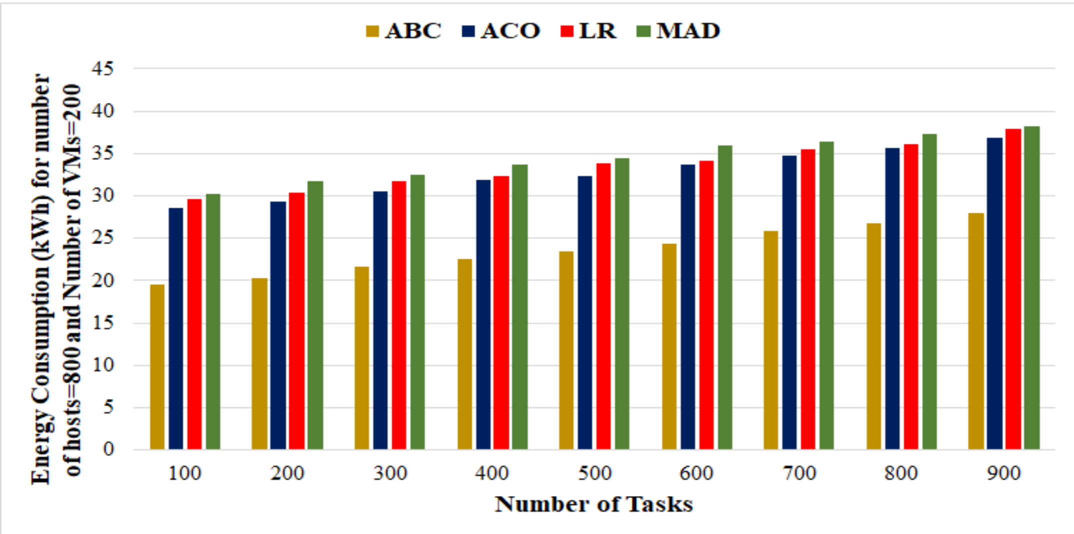

Figure 4 Graphical representation of the energy consumption (kWh) of ABC, ACO, LR and MAD number of hosts=800 and number of VMs =200

Table 6 gives the energy consumption (kWh) by the Local Regression (LR), Median Absolute Derivation (MAD), Artificial Bee Colony (ABC) and Ant Colony Optimization

(ACO) with varying number of tasks for number of hosts=800 and number of VMs=200. From the table 6, it is clear that the ABC based host detection takes less energy when compared with other host detection techniques like LR, ACO and MAD.

5. CONCLUSION

The unprecedented trend of using public cloud computing services by increasing number of customers motivates cloud services providers to optimize their resources usage and management to the limit. This is including managing cloud user's virtual machines (VM) running on one or more of the thousands of hosting servers or physical machines (PMs) of the cloud datacenters. The cloud service providers are mainly concerned on answering the two main questions that dramatically impact their infrastructure usage and utilization; Where to initially place the VMs and where to move them in case we need to move them. Along with the VM consolidation technique, VMs migration will help in protecting the physical servers from being overloaded or reduce the number of active physical servers for better resources utilization and energy saving. Efficiently detecting overloaded servers will help in improving the cloud system performance and reduce the total operational costs which will provide competitiveness for the cloud provider in the market. In this paper, optimization techniques based, LR and MAD host detection are applied and compared the efficiency of the host detection techniques are compared with Energy consumed. From the result obtained, it is clear that the ABC technique-based host detection technique takes less consumption of energy with number of tasks. The energy consumption by the host detection techniques are compared with number of hosts and number of VMs.

REFERENCES

[1] Furht, Borko. "Cloud computing fundamentals." *Handbook of cloud computing*. Springer, Boston, MA, 2010. 3-19.

[2] Hameed, Abdul, et al. "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems." *Computing* 98.7 (2016): 751-774.

[3] Hatzopoulos, Dimitris, et al. "Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources." *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013.

[4] Ezugwu, Absalom E., Seyed M. Buhari, and Sahalu B. Junaidu. "Virtual machine allocation in cloud computing environment." *International Journal of Cloud Applications and Computing (IJCAC)* 3.2 (2013): 47-60.

[5] Younge, Andrew J., et al. "Efficient resource management for cloud computing environments." *International Conference on Green Computing*. IEEE, 2010.

[6] Li, Qiang, et al. "Adaptive management of virtualized resources in cloud computing using feedback control." *2009 First International Conference on Information Science and Engineering*. IEEE, 2009.

[7] Van, Hien Nguyen, Frederic Dang Tran, and Jean-Marc Menaud. "SLA-aware virtual resource management for cloud infrastructures." *2009 Ninth IEEE International Conference on Computer and Information Technology*. Vol. 1. IEEE, 2009.

[8] Khan, Md Anit, et al. "Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: a review." *Sustainable cloud and energy services*. Springer, Cham, 2018. 135-165.

[9] Mishra, Mayank, et al. "Dynamic resource management using virtual machine migrations." *IEEE Communications Magazine* 50.9 (2012): 34-40.

[10] Ahmad, Raja Wasim, et al. "A survey on virtual machine migration and server consolidation frameworks for cloud data centers." *Journal of network and computer applications* 52 (2015): 11-25.

[11] Van, Hien Nguyen, Frederic Dang Tran, and Jean-Marc Menaud. "SLA-aware virtual resource management for cloud infrastructures." *2009 Ninth IEEE International Conference on Computer and Information Technology*. Vol. 1. IEEE, 2009.

[12] Van, Hien Nguyen, Frederic Dang Tran, and Jean-Marc Menaud. "SLA-aware virtual resource management for cloud infrastructures." *2009 Ninth IEEE International Conference on Computer and Information Technology*. Vol. 1. IEEE, 2009.

[13] Beloglazov, Anton, and Rajkumar Buyya. "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints." *IEEE transactions on parallel and distributed systems* 24.7 (2012): 1366-1379.

[14] Jararweh, Yaser, et al. "Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation." *Sustainable Computing: Informatics and Systems* 19 (2018): 262-274.

[15] Ghafari, Seyed Mohssen, et al. "Bee-MMT: A load balancing method for power consumption management in cloud computing." *2013 Sixth International Conference on Contemporary Computing (IC3)*. IEEE, 2013.

[16] Moura, Bruno, et al. "Type-2 Fuzzy Logic Approach for Overloaded Hosts in Consolidation of Virtual Machines in Cloud Computing." *2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*. Atlantis Press, 2019.

[17] Pan, Jeng-Shyang, et al. "Interaction artificial bee colony based load balance method in cloud computing." *Genetic and Evolutionary Computing*. Springer, Cham, 2015. 49-57.

[18] Benali, Asmae, Bouchra El Asri, and Houda Kriouile. "A pareto-based Artificial Bee Colony and product line for optimizing scheduling of VM on cloud computing." *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. IEEE, 2015.

[19] Moon, YoungJu, et al. "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments." *Human-centric Computing and Information Sciences* 7.1 (2017): 28.

[20] Tawfeek, Medhat A., et al. "Cloud task scheduling based on ant colony optimization." *2013 8th international conference on computer engineering & systems (ICCES)*. IEEE, 2013.

[21] Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. ConcurrComputPract Exp 24(13):1397–1420.