International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

# An Optimized Task Submission and VM Placement method to Reduce Energy Consumption in Green Cloud Computing

**Divyesh Hasmukhbhai Joshi[1], Dr. Jaykumar Shantilal Patel[2]**

**Submitted**:11/03/2024 **Revised**: 26/04/2024 **Accepted**: 03/05/2024

**Abstract:** In the context of cloud computing, energy consumption is a significant concern, particularly in green computing, which seeks to minimize environmental impact while maintaining high computational efficiency. This paper proposes an optimized task submission and virtual machine (VM) placement method aimed at reducing energy consumption in cloud data centers. The proposed approach integrates task scheduling with VM allocation to ensure that computational resources are used efficiently. By considering factors such as task priorities, resource requirements, and VM energy profiles, the method minimizes idle times and balances workload distribution across physical hosts. Simulation results demonstrate that the proposed technique significantly reduces energy consumption compared to traditional task submission and VM placement strategies. The approach ensures optimal resource utilization while meeting quality-of-service (QoS) requirements, offering a sustainable solution for green cloud computing environments.

**Keywords:** *IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), VM (Virtual Machine), Hypervisor*

## 1. Introduction

Cloud computing has revolutionized the way businesses and individuals access computing resources. However, the rapid expansion of cloud data centers has led to an increase in energy consumption, which has significant environmental and financial implications. As a result, the need for energy-efficient solutions in cloud computing has become paramount, giving rise to the concept of Green Cloud Computing.

Green Cloud Computing aims to minimize energy usage and carbon footprints while maintaining or enhancing the performance of cloud-based services. Energy consumption in cloud data centers arises from both the computational workloads processed by virtual machines (VMs) and the infrastructure that supports them, including servers, storage, cooling systems, and networking components. Therefore, optimizing cloud operations is crucial to reducing their environmental impact. One of the most effective approaches to achieve energy savings in cloud computing is through the optimization of task submission and VM placement. These two factors play a significant role in determining how efficiently resources are utilized, directly impacting the energy consumption of a cloud environment.

The integration of optimized task submission and VM placement methods holds the key to reducing energy consumption in green cloud computing. By focusing on energy-efficient task scheduling, load balancing, and intelligent VM allocation, cloud providers can achieve significant reductions in power consumption while maintaining high levels of service performance.

*[1] Research Scholar ,Gujarat Technological University,Ahmedabad,Gujarat-382350,India*
*divyeshjoshi89@gmail.com*
*[2] Professor,Chaudhari Technical Institute,Gandhinagar,Gujarat-382007,India*
*jay_sp_mca@yahoo.co.in*

### 1.1 Optimized Task Submission

Task submission refers to how computing tasks (or jobs) are submitted to the cloud infrastructure for execution. Traditional cloud systems often submit tasks to servers or VMs without considering factors such as load balancing or energy efficiency. However, a more optimized task submission strategy can significantly reduce energy consumption.

Optimizing task submission in cloud computing is a key approach to achieving energy savings, as it directly impacts how computational resources are allocated, scheduled, and utilized. Efficient task submission ensures that energy consumption is minimized while maintaining performance and meeting the users' needs. Below are some strategies and approaches to achieve energy savings through optimization of task submission:

**1.1.1 Task Consolidation (Batching)**:Instead of submitting tasks individually, tasks can be grouped together and submitted in batches. By executing multiple tasks simultaneously, cloud providers can better utilize their resources, reduce idle time, and optimize power consumption. It can be beneficial to Reduces overhead from frequent task switching and improves resource utilization, leading to energy savings.

**1.1.2. Dynamic Task Scheduling** : Dynamic scheduling ensures that tasks are assigned to the most energy-efficient servers based on factors such as the current workload, server power consumption, and available capacity. Load balancing algorithms can distribute tasks more evenly, reducing energy consumption on heavily loaded servers. Various technique use for task scheduling like: Scheduling algorithms like Energy-Aware Scheduling, Demand-Aware Scheduling, and Load Balancing can be employed to ensure that tasks are assigned in a way that minimizes energy consumption.

**1.1.3. Green Scheduling Policies**: Green scheduling focuses on minimizing the energy usage of the cloud data center while fulfilling service-level agreements (SLAs). These policies can include power-aware scheduling strategies, where tasks are routed to servers that are energy-efficient or to servers operating during off-peak hours.Most coomoly used technique are DVFS (Dynamic Voltage and Frequency Scaling) can be used for energy-aware task scheduling, allowing cloud servers to scale down power consumption without impacting performance.

**1.1.4. Resource Pooling and Virtualization**: Cloud computing heavily relies on virtualization to run multiple virtual machines (VMs) on a single physical machine. By optimizing the number of VMs and consolidating tasks across fewer physical machines, energy consumption can be reduced.it will be used for reduces the need for excessive physical machines, resulting in better power efficiency by fully utilizing available hardware.

**1.1.5. Energy-Aware Resource Allocation**: Instead of randomly allocating tasks, energy-aware resource allocation ensures that tasks are submitted to resources that are both capable of handling the workload and optimized for energy use. This involves real-time analysis of server loads and power consumption patterns to make informed decisions on where to place tasks. Various technique used for energy aware resource allocation likeEnergy-Aware Virtual Machine Placement and Energy-Aware Resource

Management optimize which resources are used based on the available energy capacity.

**1.1.6. Energy Efficiency Through Task Offloading**: Offloading computation-heavy tasks from local devices to cloud servers or edge servers can reduce energy consumption on the client-side. Efficient task offloading can also reduce the load on data centers and balance energy consumption across different locations. Various technique used for energy Efficiency like Edge computing can be integrated to offload computation to the cloud or edge nodes based on energy consumption and task priority.

**1.1.7. Autonomic Computing**: Autonomic computing refers to the self-management capabilities of systems, where cloud infrastructures automatically adjust task submission strategies based on real-time resource availability and energy consumption. This can include scaling resources up or down based on load and implementing power-saving strategies. Various technique used for Autonomic computing like Self-healing, self-optimizing, and self-configuring systems can dynamically adjust task submissions and resource usage to optimize for energy efficiency.

**1.1.8. Task Prioritization Based on Energy Impact**: Prioritizing tasks based on their energy impact helps ensure that more energy-intensive tasks are given priority and optimized for execution on resources that minimize power consumption, such as those using renewable energy or those that are less energy-hungry. Various technique used for Task Prioritization base on energy impact Energy-aware task prioritization algorithms can be used to decide the most efficient sequence and allocation of tasks.

**1.1.9. Predictive Task Scheduling**: Predictive scheduling involves analyzing historical data to predict task execution times and resource requirements, allowing tasks to be submitted at optimal times when energy costs (and load) are lower.

**1.1.10. Green Data Center Design**: Beyond task submission, the design of the data center itself can influence energy consumption. Cloud providers can optimize task submission by ensuring tasks are routed to energy-efficient data centers, using renewable energy sources, or employing energy-efficient hardware.It can be helpful for Integrating renewable energy sources and efficient cooling techniques (e.g., liquid cooling) within the data center can reduce the overall energy consumption of task submissions.

**1.1.11. Task Migration and Dynamic Scaling**: Migrating tasks between different servers or cloud environments based on their energy efficiency can lead to optimized resource usage and power consumption .Many Techniques like VM migration or elastic scaling can adjust the number of active resources based on the task demand and energy availability.

**1.1.12. Power-Aware Parallel Computing:** For computationally intensive tasks, parallelizing them and running them on multiple low-power machines, rather than a few high-power machines, can save energy.many technique used like Parallel computing models can balance the load across multiple machines to reduce the overall energy consumption.

By integrating these strategies into task submission processes, cloud providers and users can minimize energy consumption while maintaining high performance, thereby achieving sustainable and efficient cloud computing operations.

### 1.2 Optimized VM Placement

VM placement refers to the process of allocating virtual machines to physical servers within a cloud data center. Inefficient placement can lead to idle resources, overloading of specific servers, and increased energy consumption due to unnecessary cooling and power requirements.Achieving energy savings in cloud computing through Virtual Machine (VM) placement involves strategically assigning VMs to physical servers (hosts) in a way that minimizes energy consumption while maintaining performance and meeting service-level agreements (SLAs). VM placement is crucial because it directly affects how efficiently physical resources (such as CPU, memory, and storage) are utilized. Below are some key approaches to achieving energy savings through VM placement methods:

**1.2.1.VM Consolidation**: VM consolidation involves grouping multiple VMs onto fewer physical machines, thereby reducing the number of active servers. This reduces the overall energy consumption since idle or underutilized servers can be powered down or put into a low-energy state. Various approach used for VM Consolidation like by allocating multiple VMs with lower resource requirements to fewer physical machines, cloud providers can free up resources for energy-saving techniques (such as dynamic scaling) and minimize the number of servers that need to be powered on. It can be helpful for Reduced idle time and underutilization of resources lead to significant energy savings.

**1.2.2. Dynamic Resource Allocation and Scaling (Elasticity):** Dynamic VM placement adjusts the number of active servers based on current demand. When the demand is low, some physical servers can be powered off, and VMs can be dynamically migrated to active servers with more capacity. Various Approach used like Elasticity in cloud environments means that the number of active servers scales up or down based on workload. By monitoring real-time workload metrics, cloud platforms can place VMs on fewer physical machines during low-demand periods, ensuring the least amount of power consumption. Many Techniques used like Auto-scaling, VM migration, and load balancing can be employed to ensure resources are dynamically reallocated, reducing overall energy consumption.

**1.2.3. Energy-Aware VM Placement Algorithms**: Energy-aware algorithms take into account the energy consumption of physical servers while making VM placement decisions. These algorithms aim to minimize the energy usage of cloud data centers while meeting the required performance metrics. Various Techniques used like Energy-Efficient Load Balancing which is used for Distribute VMs across physical servers in a way that reduces energy consumption while ensuring no server is overburdened. and Energy-Aware VM Placement These algorithms consider the energy profile of each host (i.e., how much power it consumes based on its load) and aim to place VMs in a way that minimizes overall energy consumption. It will be Helpful for balance server load effectively while minimizing power consumption.

**1.2.4. VM Migration**: VM migration involves moving VMs from one physical server to another, often to consolidate workloads or improve resource utilization. By migrating VMs away from underutilized servers, resources can be freed up, and servers can be powered down, reducing energy consumption. Many Techniques used like Live VM migration allows VMs to be moved with minimal disruption to their execution. VM migration can be done when certain thresholds (e.g., CPU usage, memory usage) are reached on the physical host, thus preventing over-provisioning and underutilization.It will be Helpful for VM migration can ensure that fewer physical servers are required, thereby cutting down on the energy used to keep unnecessary servers running.

**1.2.5. Dynamic Voltage and Frequency Scaling (DVFS)**: DVFS allows cloud servers to dynamically adjust their operating voltage and frequency based on the current load. When a server is lightly loaded, its power consumption can be reduced by lowering the frequency and voltage, which saves energy. Various Approach used like Energy-aware VM placement algorithms can ensure that VMs are placed on servers that are capable of scaling down their energy consumption via DVFS. This minimizes energy consumption without sacrificing performance. It will be Helpful for Reduces power usage while maintaining the desired performance levels.

**1.2.6. Server Power States (Sleep Mode and Hibernation):** Servers can be placed in low-power states when they are not in use. VM placement strategies that consolidate workloads onto fewer servers can allow other servers to enter power-saving states such as sleep or hibernation. Various Approach used like by ensuring that only the required number of physical machines are kept running, the rest can be switched off or placed in low-power states, such as sleep or idle mode, reducing energy usage significantly. It will be Helpful for Powers down unused servers, leading to significant energy savings.

**1.2.7. Using Renewable Energy Sources**: Cloud providers can prioritize placing VMs on servers located in data centers powered by renewable energy sources (e.g., solar, wind). VM placement decisions can take into account the environmental energy mix of different data centers to optimize energy usage.various approach like Data centers with a higher proportion of renewable energy (compared to those powered by non-renewable sources) could be prioritized for VM placement, making the cloud operation greener and more energy-efficient. It will be Helpful for to Reduces the carbon footprint of the data center while promoting sustainability in cloud computing.

**1.2.8. Thermal-aware VM Placement**: The temperature of a server directly affects its power consumption, as cooling requirements increase with higher temperatures. Thermal-aware VM placement algorithms aim to optimize the placement of VMs in a way that minimizes the need for additional cooling, thus saving energy. various Approach like by distributing VMs to prevent localized hotspots, thermal-aware placement algorithms ensure that cooling systems are not overburdened, reducing the power required for cooling. It will be Helpful for Reduces the overall power required for cooling, which can account for a significant portion of a data center's energy consumption.

**1.2.9. Proactive Load Forecasting and Scheduling**: Forecasting future workloads based on historical patterns allows cloud providers to place VMs proactively to prevent resource over-provisioning or under-utilization, thus optimizing energy consumption. Various Approach used like Machine learning algorithms can predict periods of high or low demand and place VMs in advance, ensuring that resources are utilized efficiently without the need for last-minute scaling, which could be energy-inefficient. It will be Helpful for Ensures energy-efficient VM placement by anticipating load requirements ahead of time, minimizing wasted resources.

**1.2.10.Virtual Machine Placement with Quality-of-Service (QoS) Considerations:** VM placement algorithms can also take QoS parameters (e.g., response time, throughput) into account. These algorithms aim to balance energy efficiency with the required performance guarantees for each VM. Various Approach used like By optimizing VM placement based on both energy and QoS metrics, cloud providers can ensure that energy savings do not come at the expense of performance degradation. It will be Helpful for Maintains high-quality service while reducing energy consumption.

**1.2.11.Multi-Objective Optimization**: Multi-objective optimization techniques aim to balance various conflicting goals, such as minimizing energy consumption, maximizing resource utilization, and ensuring service quality. Many Approach used like Algorithms like genetic algorithms or particle swarm optimization can be used to simultaneously optimize for energy efficiency and other operational constraints, ensuring a holistic approach to VM placement. It will be Helpful for Achieves an optimal trade-off between energy savings and system performance.

By employing these VM placement methods, cloud service providers can significantly reduce the energy consumption of their data centers, contributing to both cost savings and environmental sustainability. The key to energy-efficient VM placement is ensuring that workloads are distributed optimally to reduce server idle time, improve resource utilization, and minimize the number of active servers.

## 1.3 Need for Energy Efficiency in green cloud computing

Industrial applications are increasingly being hosted in cloud data centers. It's critical to lower data center electricity and cooling operating expenses. The fundamental issue is that the tremendous improvement in computer capability has coincided with an even bigger increase in power consumption [1].

The necessity to reduce energy consumption in Cloud data centers is the driving force behind this research. Electricity is used extensively by IT companies, space centers, research centers, corporations, and web applications. As a result of climate change and global warming, energy efficiency has become the most important design criteria for computing systems.

The massive rise in computer performance has been accompanied by an even bigger increasein power consumption. Powering and cooling data centers at a lower cost is the main objective of cloud computing.Improving reliability in cloud computing is also one of the challenges. As a rule of thumb,the failure rate of a system doubles for every 10°C increase in temperature. The accuracy ofthe results was influenced by the computing environment [2].An optimized task submission and VM placement method can lead to a variety of energy efficiency benefits, including:

**1.3.1 Reduced Power Consumption**: By intelligently submitting tasks and allocating VMs, energy wastage is minimized, leading to overall lower energy consumption.

**1.3.2 Improved Resource Utilization**: Better allocation of tasks and VMs ensures that hardware resources are used more effectively, reducing the need for additional resources.

**1.3.3Lower Carbon Footprint**: By reducing energy consumption, green cloud computing methods help reduce the carbon footprint of cloud services, supporting sustainability goals.

## 1.4 Load Balancing in Cloud Computing

Load balancing is a mechanism for assigning or reassigning jobs on various areas of the system to better utilize resources and reduce system response time and processing time [3].

To achieve optimal performance, more effective load balancing solutions must be developed as the number of tasks required increases. The goal is to improve performance by balancing load among available resources or underlying hardware to achieve optimal resource usage, high throughput, and minimize overloading. We can reduce reaction time and increase throughput by using an innovative load balancing strategy that makes use of network and other resources. Load balancing is an important part of cloud architecture that distributes and allocates load to available resources to enhance performance. A few unique load balancing techniques for cloud computing are well-known for their overall system performance gains [4].

**1.4.1 Round Robin Algorithm**: In a Round Robin technique, the Round Robin algorithm divides all existing processes among all processors and assigns each one to a processor. The task is distributed evenly across the available processors using this approach. The Round Robin scheduling technique successfully employs the time slice paradigm. In this method, total time is divided into many partitions, and each participating node is assigned a specified time interval. Each node will be required to complete its task or activity within the time frame given. If the task is finished within the specified time limit, the user will not have to wait; otherwise, the user will have to wait for the next round.

**1.4.2Equally Spread Current Execution (ESCE) Algorithm** :The load balancer closely monitors the load on all VMs and divides it evenly across them using the equally spread current execution (ESCE) or active monitoring technique. The system load balancer keeps track of the VMs' index table and the quantity of provisions assigned to each one. When a new job comes, the data center controller requests a new VM from the load balancer. The load balancer locates the index table for the VM with the least amount of load. When the load balancer finds the VM with the lowest load, it informs the data center controller with the VM id.

**1.4.3 Throttled Algorithm**: Each VM in the Throttled algorithm handles just one job at a time; a new job can be processed only after the current one has been successfully completed. The load balancer entity maintains an index table of all the virtual machines (VMs) and their current states (available or busy). The load balancer scans the index table for the available VM as soon as the data center managers request a VM allocation. The VMID is returned to the data center controller if the load balancer detects an available VM at that moment. The load balancer will just return null if this is not the case. The request is queued until a VM becomes available if the data center controller receives null from the load balancer.

**1.4.4 Least Frequently Used (LFU) Algorithm** :When a task is submitted to a specific data center for execution using the least frequently used (LFU) algorithm, the execution option is made based on the VM to which the tasks have been assigned the fewest number of times. As a result of this approach, VMs are kept active for the bulk of the time to process tasks. Any present strategy to increase server load in a cloud environment causes some Quality of Service (QoS) difficulties and degrades overall performance. To maximize resource utilization, our suggested system model relies on server response time and data center processing time.

## 2. Literature Review

**2.1 Energy Efficiency in Cloud Computing**: Cloud computing is best-suited for its potential benefits, and it delivers computer resources such as CPU (Processing Power), memory, storage, networking, and other services as a service [5]. According to the NIST definition, cloud computing is an on-demand and convenient paradigm for computing resources with speedy provisioning and minimal management or service provider involvement. Amazon Web Services (AWS) [6], Google Cloud Platform (GCP) [7], Microsoft Azure [8], IBM Cloud [9], and Alibaba Cloud [10] are among the leading cloud service providers, all of which are doing well and investing extensively to increase their capabilities in many industries.

Green Cloud Computing is a concept that works with cloud resources that can manage energy. It's a key solution for all scalability-sensitive dynamic resource demands. The growing number of internet users necessitates the establishment of data centers in various zones and geographical regions around the world. According to a report submitted to the US Department of Energy in 2014 [11], data centers in the US consume 1.8 percent of total energy. Data centers emit a large amount of $CO_2$ [12], which is a major problem for the environment and could result in global warming. Because of this serious problem, reducing energy use should be a primary goal. The solution to this dilemma will contribute to environmental preservation while also supporting leading firms in reaching significant growth.

Green IT (GIT) is an idea of a green infrastructure program aiming at increasing efficiency and productivity by using and growing resources in companies and society in a sustainable manner. Information Technology (IT) organizations have used GIT to accomplish successful operations with little carbon waste. However, the short lifespan of IT goods, as well as some manufacturing and disposal methods, have sparked widespread concern about negative consequences, such as higher energy use by businesses. Furthermore, the carbon emissions from IT products and systems outnumber those from the aircraft industry. Green computing is a solution to environmental issues [13].

**2.2 Literature Survey**:

T. Mastelic et.al. in [14], conducted a complete energy efficiency analysis of infrastructure supporting the cloud computing paradigm. First, they established a method for measuring the energy efficiency of the most essential data center domains, such as server and network equipment, as well as cloud management systems and appliances, which are software-based appliances used by end-users. Second, they used this method to examine accessible scientific and industry literature on cutting-edge data center operations and equipment. Finally, they outlined current research difficulties and prospective research directions. As shown in Fig 1, the authors have presented energy flow in the system and how energy is entered in the system, energy consumed, energy used, and finally the task is carried out as output.

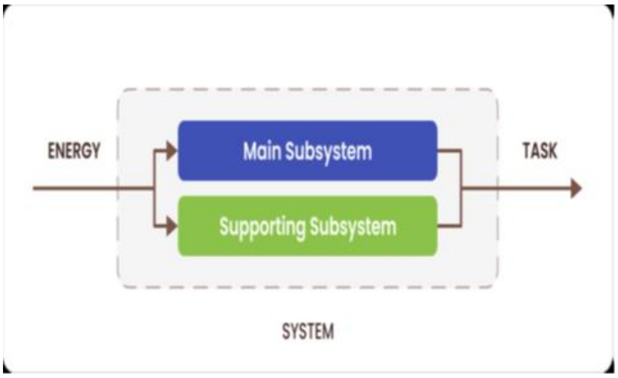


**Fig 1:** Energy flow in system

In [15], M. Pantazoglou et.al. described a decentralized method for managing scalable and energy-efficient virtual machine (VM) instances provisioned by big enterprise clouds. The data center's computation resources are successfully grouped into a hypercube structure in our approach. As resources are added or deleted in response to changes in the number of provided VM instances, the hypercube scales up and down effortlessly. Each compute node functions independently of any central components and handles

its own workload using a set of distributed load balancing rules and algorithms. On the one hand, underutilized nodes try to offload their workload to their hypercube neighbours by turning it off.

X. Xu et.al. in [16], presented EnReal, an Energy-aware Resource Allocation approach. Essentially, the authors used dynamic virtual machine deployment to execute scientific workflows. In particular, an energy consumption model for cloud computing applications is

provided, and a related energy-aware resource allocation algorithm for virtual machine scheduling to complete scientific workflow executions is proposed. The proposed strategy is both successful and efficient, according to the results of the experiments.

In [17], V. Cima et. al. presented variants of live migration as described in Fig. 2. There are three types of live migration. E.g. Pre-copy, Post-copy, and Hybrid live migration.

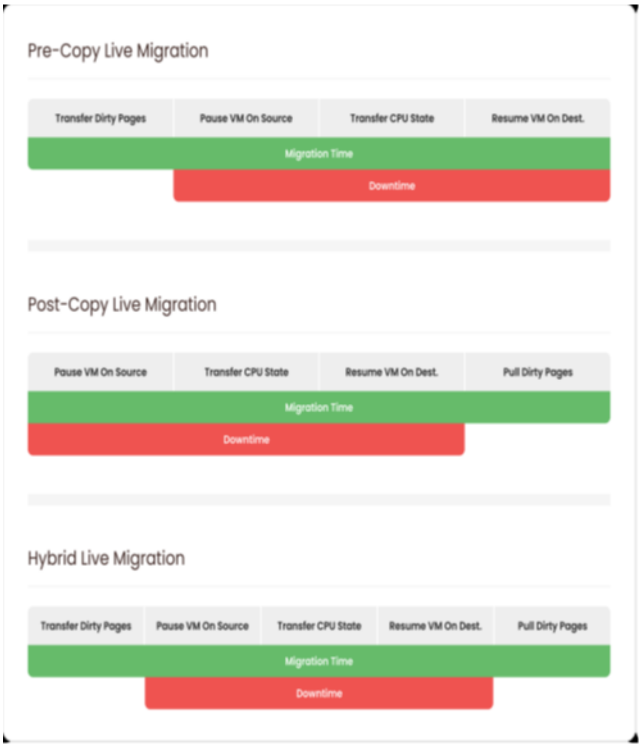


**Fig 2:** Pre-copy, Post-copy and Hybrid Live Migration Mechanism

It gives a highlight of how the energy efficiency concept can be added to OpenStack. The main goal is to create OpenStack resource management that is energy efficient. There are two steps to it. In the first step, it is examined how cloud resources are being used in terms of resources and energy. In the second step, there is an approach for static migration based on manual resource consumption monitoring.

N. Akhter et. al. in [18], described different reviews and open challenges for energy aware resource allocation strategies of cloud data centers. The authors have presented an exhaustive collective summary for energy aware resource allocation methods and virtual machine identification strategies. Fig.3 represents energy consumption analysis model for the cloud environment. The data collection engine retrieves users' requests from cloud data centers and sends them to the analysis engine. It also gets cloud resource details and is attached with a policy for the energy consumption model. Simultaneously analysis engine is fed with task description and system configuration details. As an output analysis engine generates analysis results.

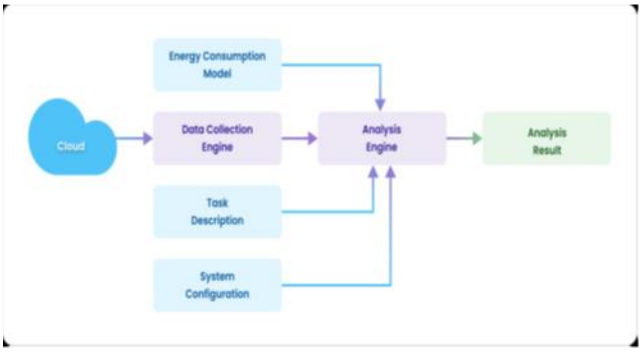


**Fig 3:** Energy Consumption Analysis Model

In [19], Anton Beloglazov et al. have suggested an efficient resource management policy for virtualized cloud data centers. The goal is to continuously condense VMs via live migration and turn off idle nodes to save power while maintaining the appropriate Quality of Service. The authors also offered evaluation results that show that dynamic VM reallocation saves a significant amount of energy, justifying the suggested policy's ongoing development. Due to the trust elements of cloud computing, various areas are explored and proposed to work on energy efficient methodologies in the cloud environment.

David Aikema et al. [20] proposed a VM migration strategy with two approaches: in the first, authors calculated migration time and associated energy consumption, whereas in the second, they directly migrated all workload to accessible other VMs based on threshold values. The authors established a correlation with many scenarios, concluding that determining migration time and needed energy utilization computation is really beneficial. The authors presented the findings of an experimental investigation of the power and energy consumption of various workloads throughout the migration process. The authors concluded about the sorts of workloads that are suitable for migration and how each can be converted successfully.

Minu Bala and Devanand in [21] focused on CO2 emissions and VM migration based on CPU utilization parameters in one of the strategies for VM migration. They modified the upper and lower thresholds of the CPU limit, based on which VMs are migrated to different data centers on different hosts. In order to reduce power consumption, various green computing solutions have been investigated, both at the hardware and software levels. The experimental study shows a simulation environment that captures the energy consumption of computing and communicating devices in the cloud. It also demonstrates how a variety of green computing solutions outperform standard computing approaches based on various data center designs.

In a cloud computing context, K. Zhang et al. [22] considered virtual machine (VM) energy saving solutions on an overloaded host. They researched the energy affecting factors and developed energy efficient VM selection strategies based on greedy algorithms and the dynamic programming process during a VM migration. A range of green computing solutions have been researched at both the hardware and software levels in order to reduce power usage. The lab experiment exhibits a simulated environment that depicts the energy usage of cloud computing and connectivity devices. It also shows how a variety of green computing solutions outperform standard computing approaches based on various data center layouts.

To optimize energy consumption for OpenNebula-based Cloud, Yacine Kessaci et al. [23] suggested an EMLS-ONC method. The authors created the EMLS-ONC (Energy-aware Multi start Local Search algorithm for an OpenNebula-based Cloud), which minimizes the energy consumption of a geographically scattered OpenNebula-managed cloud computing infrastructure. Our EMLS-ONC scheduler's output is compared to OpenNebula's default scheduler's output. The two approaches were tested using a variety of (VM) arrival scenarios and hardware architectures. The results show that EMLS-ONC significantly beats the previous OpenNebula scheduler in terms of energy consumption. EMLS-ONC has also been shown to help schedule more applications.

A User-Priority Guided Min-Min Scheduling algorithm has been developed by Huankai Chen et al. [24]. Two distinct strategies have been offered by the writers. Priority Aware Load Balance Improved Min-Min (LBIMM) and Load Balance Improved Min-Min (LBIMM) (PA-LBIMM). When compared to the Min-Min algorithm, the results of LBIMM and PA-LBIMM are far superior in terms of job completion time, resource load balancing, and overall system performance. This new method improves system performance by more than 20% in terms of resource consumption and user services.

Veerawali Behal et al. [25] suggested a Load Balancing method for Heterogeneous Environments. The authors present a throttled strategy for dealing with optimal reaction time. Using broker policy, this strategy improves performance in a distributed cloud system.

Yunliang Chen et al. [26] suggested a two-tier VM placement architecture based on a feasibility-driven stochastic VM placement (FDSP) method to reduce VM placement energy usage. When compared to conventional algorithms, the suggested algorithm consumes 15.3 percent less energy and costs 15.7 percent less than other VM placement rules.

To determine the best cost for VM placement in cloud-based data centers, Sourav Kanti

Addya and Anurag Satpathy [27] suggested a Game theoretic technique. To get the precise value for capital expenditure, the authors analyzed n-users in a cooperative gaming context. Integer linear programming is offered as an energy-efficient placement policy, with Microsoft Azure and Amazon EC2 as cloud service providers. Finally, the authors' proposed game theme method is more effective than first fit decreasing (FFD) and improved first fit decreasing (EFFD) in terms of utilizing less energy (EFFD).

Khosravi A., et.al. in [28], proposed a unique VM placement method that takes into account remote data centers with varying carbon footprint rates and PUEs to improve environmental sustainability. In comparison to existing competitive algorithms, simulation results show that the suggested algorithm reduces CO2 emissions and power usage while maintaining the same level of service quality.

Kansal, A., Zhao, F., et.al. in [29], proposed a metering feature for VM power capping, an approach for lowering data center power provisioning costs. Experiments are run on server traces from tens of thousands of production servers that host real-world Microsoft programs like Windows Live Messenger. The findings reveal that not only does VM power metering allow virtualized data centers to save the same amount of money that non-virtualized data centers did with physical server power capping, but it also allows virtualization to save even more money on provisioning expenses.

### 2.3 Comparative Analysis

| Sr No | Title of | Parameter Used | Advantage |
|---|---|---|---|
| 1 | Cloud Computing: Survey on Energy Efficiency [14] | Network<br>Server<br>CMS<br>Appliance | By considering various domain studies like Network, Server, CMS, and Appliance, easy to identify energy consumption for each. Detect energy consumption and energy loss for each domain |
| 2 | Decentralized and Energy Efficient Workload Management in Enterprise Clouds [15] | CPU<br>Network | Based on current power consumption, there are five VM states. E.g. Switched off, Idle, Underutilized, Ok, Overutilized, which makes power utilization regularized |
| 3 | EnReal: An Energy-Aware Resource Allocation Method for Scientific Workflow Executions in Cloud Environment [16] | CPU<br>Memory<br>Storage | Calculate energy consumption based on<br>• application executions<br>• dynamic operations<br>Allocation of resources to run a present application, Dynamic operations planning, With proper resource allocation, power consumption can be reduced. |
| 4 | Energy Efficient Allocation of Virtual Machines in Cloud Data Centers [19] | CPU<br>Memory<br>Storage | VMs that use live migration and turn down inactive nodes to save power while maintaining the requisite Quality of Service |
| 5 | Green Cloud VM Migration: Power Use Analysis [20] | CPU<br>Storage<br>Network | The many types of workloads that can be transferred and how they can be migrated most efficiently |

| | | | |
|---|---|---|---|
| 6 | Performance evaluation of cloud data centers using various green computing tactics [21] | ICT Hardware/ Software, Network | The effectiveness of various green computing strategies in comparison to traditional computing approaches employing various data center layouts. |
| 7 | Virtual Machine Migration in an Over-Committed Cloud [22] | ICT Hardware/ Software Network | Placement and migration of virtual machines, with host use balanced across all time epochs |
| 8 | An Energy-aware Multi-start Local Search Heuristic for Scheduling VMs on the OpenNebula Cloud Distribution [23] | ICT Software CPU | In terms of energy usage, EMLS-ONC outperforms the old OpenNebula scheduler by a large margin. |
| 9 | User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing [24] | ICT Software CPU | PA-LBIMM was created so that users' demands may be met on a need-to-know basis. |

Table 1: Comparative Analysis of various techniques

## 3. Problem Formulation

### 3.1 Problem Statement:

Optimizing task submission and VM placement are crucial in reducing energy consumption in cloud data centers. Inefficient task allocation and VM placement can lead to unnecessary resource wastage, increased power consumption, and underutilized hardware.

The majority of the work has focused on reducing energy usage in Green Cloud Computing by scheduling, job placement, or employing particle swarm optimization (PSO) techniques. To execute the allocated workload, several parameters such as CPU, memory, bandwidth, storage, GPU, and others require energy.

The majority of research, on the other hand, has employed one or a combination of the parameters, and the remaining parameters may be neglected when calculating the system's actual capacity. The distribution of virtual machines can be divided into two challenges in order to reduce energy consumption and improve system performance :Installation of a new virtual machine (VM) on the host and Allocation of present virtual machines (VMs) optimization This paper proposes a novel approach to optimize task submission and VM placement to minimize energy consumption in green cloud environments. It is critical to investigate the power flow in typical cloud data centers and understand how power is distributed in order to increase energy efficiency in the Cloud. Cooling equipment power consumption is significant, but it is proportional to IT power usage. The goal of energy efficient Cloud data center resource allocation is to detect and assign resources to each incoming user request in such a way that the user's requirements are met, the least number of resources is consumed, and cloud data center energy efficiency is maximized.

The objective of this work is to design and develop models and algorithms for energy efficient resource allocation in cloud data centers. The major purpose of this research is to design, develop, and test resource allocation optimization methods for standard cloud computing architectures, which are commonly used to manage clouds.

### 3.2 Research Methodology

Following are the proposed steps to minimize power consumption:

• Select Overloaded Physical Machine (Host)

• Migration using VM Selection

• VM Placement using Physical Machine (Host) Selection

System architecture with a Load-Aware VM placement mechanism has been proposed to

manage cloud resources. This system mechanism helps to optimize resource allocation,

improves system performance, and reduces energy consumption.

Task submission and VM placement have been proposed for the research hypothesis.The proposed methods help to optimize system performance and to reduce energy consumption. Three main parameters memory, mips, and bandwidth have been used for the proposed methods.

The ultimate aim of considering these parameters is to minimize energy consumption in an efficient way. Memory, mips, and bandwidth utilization need to be calculated to get the Total

Utilization Ratio (TUR) value which will be required to compute total power consumption.

Memory or ram is required to execute any process or task. It is one of the key factors that consume power of the resources or systems which are in use. Memory utilization is the ratio of memory used by all the virtual machines which are running over a particular physical machine. The mathematical representation of memory utilization is as per Equation (1).

$ramUtilization =$ …………...(1)

CPU or mips helps to process instructions or data. As the name suggests mips executes millions of instructions in parallel that requires a significant amount of power. CPU utilization is the ratio of mips used by all the virtual machines which are running over a particular physical machine. The mathematical representation of mips utilization is as per Equation (2).

$mipsUtilization =$ ……. (2)

Network or bandwidth is an important parameter as it is a communication channel of the cloud resources for data or instructions exchange. Considering a significant CPU and memory capacity compared to bandwidth may lead to performance degradation. Not offering proper bandwidth may also increase execution time for the resources being used, hence increasing power consumption. Bandwidth utilization is the ratio of bandwidth used by all the virtual machines which are running over a particular physical machine. Mathematical representation of bandwidth utilization is as mentioned in Equation (3).

$bwUtilization =$ …… …..
(3)

Total Utilization Ratio (TUR) is calculated as an average utilization of ram, mips, and

bandwidth according to Equation (1), (2), and (3).

Total Utilization Ratio=((RamUtilization+MipsUtilization+BwUtilization))/3 ..(4)

Where,

ramUtilization = Utilization of ram (Memory)

mipsUtilization = Utilization of mips (CPU)

bwUtilization = Utilization of bandwidth (Network)

Because the CPU is the primary consumer of dynamic power and its power is mostly influenced by its power state, a utilization-based power model was deemed a proxy for predicting the host total power (active or sleeping). This assumption will fail for workloads that aren't CPU heavy. The utilization-based power model is defined as follows:

$Pc = Pmin + (Pmax - Pmin) + Total\ Utilization$ Ratio ……………………………. ………………………………… (5)

where,

Pmin is minimum power consumed at idle state

Pmax is the maximum power consumed at peak load

Utilization is between 0 and 1.

$Pc = Pmin + (Pmax - Pmin)$ +((RamUtilization+MipsUtilization+BwUtilization))/3 …(6)

As mentioned in Equation (6), calculated power consumption will be used for the proposed method result parameter calculations. Here we described the calculation of power consumption by considering ram utilization, mips utilization, and bw utilization together. Also, maximum power consumed at peak load (Pmax) and minimum power consumed at idle state (Pmin) are also considered for power consumption calculation.

## 4. Proposed Methodology

Most of the work has been carried out with different strategies like load balancing, resource allocation, virtual machines migration, etc. Also, the existing approaches majorly rely on mips or CPU utilization.

### 4.1 Task submission and VM Placement

Task submission and VM placement method uses parameters like task length, mips, no. of PE (Processing Elements/cores), no. of tasks, and no. of VMs. The process of task submission and VM placement can be furnished with the following steps(Fig 4):

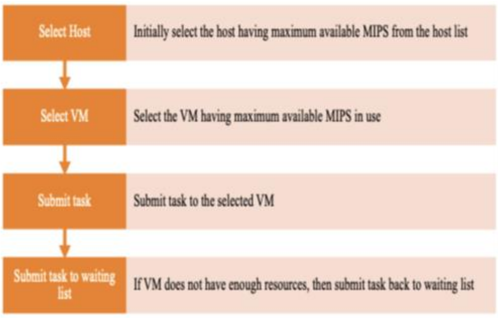


**Fig 4** :Task Submission and VM Placement Steps

1. Select Host:In the first step, the host is identified from the host list on the basis of the maximum available mips.

2. Select VM:In the second step, VM is identified on the basis of the maximum available mips.

3. Submit Task:In the third step, the task is submitted to identified VM in step 2.

4. Submit Task to the waiting list:In the last step, the submitted task is returned and then placed into the queue (waiting list) in case of VM does not have enough resources.

**Algorithm 4.1: Task submission and VM Placement**

```
Input: Vm_list, Task_list, Host_list
Output: VMid(j) to submit task ti
Hostid = null
VMid = null
//Find Host from host k having maximum available mips
For each ti task from Task_list
    Hostavailable_mips = 0
    for each Host k from Host_list
        if (Hostavailable_mips < Host_mipsk)
            Hostavailable_mips = Host_mipsk
            Hostid = Hostk.Hostid
        End if
     End for
End for
//Find VM from host k having maximum available mips
For each vm from host having id = Hostid
   VMavailable_mips = 0
   for each vm j from Vm_list
       if (VMavailable_mips < vm_mipsj)
           VMavailable_mips = vm_mipsj
            VMid = VMk.VMid
       End if
   End for
  End for
If VMid is not null and resources are available in VMid then
    submit task to VM having id = VMid
else
     submit task to Task_list
```

### 4.2 Host Selection Strategy

There may be more than one available host which can accept the task. Thus, the task must find the most suitable host based on priority information. It should consider the available host with minimum load.

### 4.3 VM Selection Strategy

From the selected host, the VM Selection strategy identifies a suitable virtual machine from the available list of virtual machines. A virtual machine is preferred which can execute and finish tasks in the minimum possible processing time.

## 5. Implementation Tool and Experimental Setup

At the initial stage, implementation of the VM relocation method has been carried out on a virtual cloud environment known as CloudSim [30]. It provides simulation for cloud computing infrastructure and services. A step ahead, OpenStack has been explored and hosted over a dedicated machine with Mac OS. Proposed methods have been implemented on OpenStack for the experimentation of energy consumption as well as a number of virtual machine migration with a workload of 500 to 3500 cloudlets. Moving forward, OpenStack has been hosted over the scalable infrastructure of AWS cloud to sustain a higher workload. Load-Aware VM placement has been implemented on OpenStack with AWS environment to determine various parameters like energy consumption, virtual machine migration, and mean time before VM migration.

### 5.1 Amazon Web Services (AWS) – Compute Service

Amazon Elastic Computational Cloud (Amazon EC2) delivers the largest and deepest compute platform available, with over 500 instances and a choice of the latest processor, storage, networking, operating system, and purchase model. We were the first major cloud service to support Intel, AMD, and Arm CPUs, as well as the first to offer on-demand EC2 Mac instances with 400 Gbps Ethernet networking. In the cloud, we have the lowest cost per inference instance and the best pricing performance for machine learning training. More SAP, high-performance computing (HPC), machine learning (ML), and Windows programs are hosted on AWS than on any other cloud.

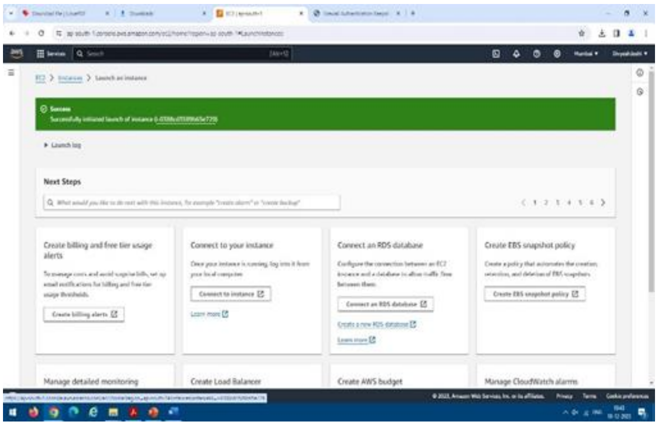

**Fig 5** : EC 2 Instance Launch Status

Once the EC2 instance is in running mode(Fig 5), it has to pass health checks. AWS performs two health checks to make sure that the EC2 instance is working fine.

### 5.2 OpenStack

OpenStack [30] is a cloud operating system that uses APIs and standard authentication techniques to manage and deploy large pools of compute, storage, and networking resources across a data center. A dashboard is also available(Fig 5), allowing administrators to keep control while empowering users to provision resources using a web interface. OpenStack is divided into services, allowing you to plug and play components to meet your specific requirements. The OpenStack map provides an "at a glance" perspective of the opensource environment, allowing you to see where those services fit in and how they can collaborate.

OpenStack launched in 2010 as a collaborative initiative of Rackspace Hosting and NASA, and it manages enormous pools of computing, storage, and networking resources throughout a data center. More than 500 firms have joined the project, which is now overseen by the OpenStack Foundation, a non-profit corporation founded in September 2012 to promote OpenStack software and its community. Some of the OpenStack Services are compute, Hardware lifecycle, Storage, and Networking. It also provides a Ceilometer kind of tool for Monitoring. OpenStack is committed to a transparent design and development methodology. A six month release cycle is followed by the community, which includes frequent development milestones. The community gathers for the Forum during each release cycle to gather user requirements, after which developers gather for the Project Teams Gathering (PTG) to begin development work and cross project collaboration.

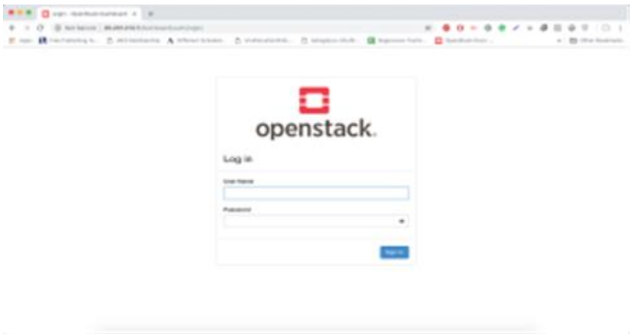


**Fig 6:** Openstack Authentication

#### 5.2.1 Components of OpenStack

OpenStack is a cloud environment that can be hosted anywhere with different operating systems. OpenStack has a wide range of component resources that are used to create a cloud environment as per the need. Nova is used as compute service, Glance is used for image service, Swift is used for object storage, Ceilometer is used for telemetry, etc.

##### 5.2.1.1 KiloWatt API (KWAPI)

KiloWatt API (KWAPI) [31] is an energy monitoring software framework that can manage OpenStack clouds. The KWAPI framework, which has a scalable, extendable, and fully integrated design, supports numerous wattmeter devices, multiple measurement formats, and reduces communication overhead (Fig 7 ) represent KiloWatt API (KWAPI) Framework.

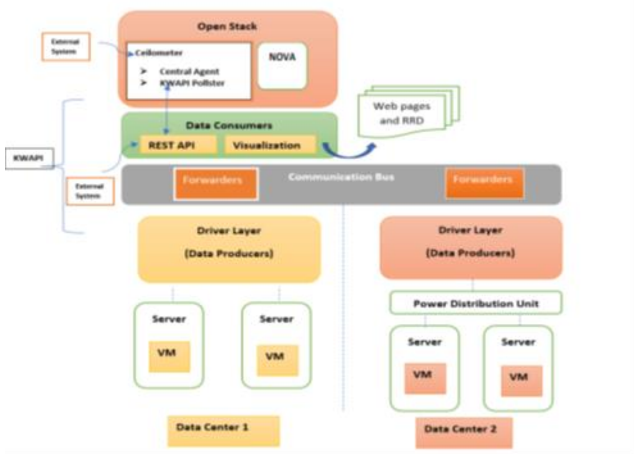


**Fig 7:** KiloWatt API (KWAPI) Framework

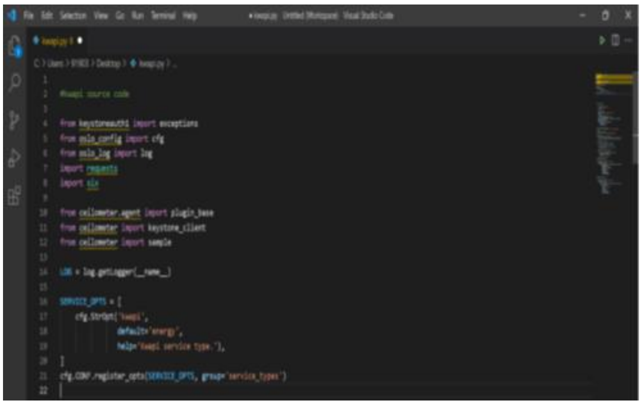

**Fig 8:** KWAPI Library and module

KWAPI implemented (Fig 8) over Python programming language

##### 5.2.1.2 Ceilometer

The Ceilometer project [32] is a data collection service that allows users to normalize and transform data across all current OpenStack core components, with plans to support future OpenStack components in the near future. The Telemetry project includes Ceilometer as a component (Fig 9). Its data can be used across all OpenStack core components to provide client billing, resource tracking, and warning capabilities.

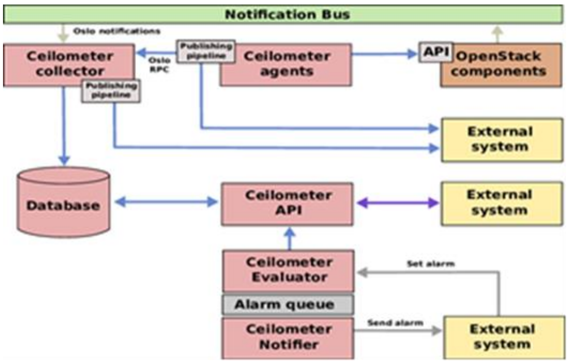


**Fig 9**: Ceilometer structure

### 5.3 Result and Discussion

The overall response time and data center processing time are calculated using the task submission and VM placement procedure. Round Robin, Equally Spreaded Current Execution (ESCE), Throttled, Least Frequently Used (LFU), and suggested Load Balancing Policy are all calculated and compared. Proximity-based routing, Optimized response time, and Dynamically reconfigured approaches were used in all of these comparisons.

1. Service Proximity-based Routing

This routing policy identifies and picks the data center with the lowest network latency, which is the data center closest to the user. When two or more nearby data centers are available, a random selection should be made. During its operation, this routing policy does not take into account load or cost.

2. Performance Optimized Routing

The performance of all data centers is monitored by the service broker in a performance optimized routing strategy, and traffic is routed to the data centers depending on the best response time. This routing policy does not take into account the cost of its operation.

3. Dynamically Reconfigurable Routing

This routing strategy is an extension of service proximity-based routing, in which the routing logic stays the same but the service broker is given the additional job of scaling the application deployment to the load. This routing scheme has the potential to raise or decrease the number of virtual machines created in existing data centers. To reduce energy consumption and VM migration while increasing mean time, load-aware VM placement and overloaded / underloaded host selection methods are applied. All of these parameters are compared to existing approaches such as ThrMmt [33], ThrRs [34], MadMu [35], and MadRs [36].

**1. ThrMmt:**ThrMmt is the Threshold Minimum Migration Time method. ThrMmt is a combination of Threshold VM allocation policy and Minimum Migration Time VM selection policy.

**2. ThrRs**:ThrRs is the Threshold Random Selection method. ThrRs method uses Static Threshold (THR) VM allocation policy and Random Selection (RS) VM selection policy.

**3. MadMu**:MadMu is the Median Absolute Deviation Minimum Utilization method. MadMu comprises of Median Absolute Deviation (MAD) VM allocation policy and Minimum Utilization (MU) VM selection policy.

**4. MadRs**: MadRs is the Median Absolute Deviation Random Selection method. MadRs method consists of Median Absolute Deviation (MAD) VM allocation policy and Random Selection (RS) VM selection policy.

#### 5.3.1 Result Discussion of Task submission and VM placement method

We installed and configured OpenStack on the top of Windows OS to perform task submission and the VM placement method. To begin, we considered ten hosts in the initial experiment.

A total of 50 virtual machines with a capacity of 5 processing components were installed. We fed 500 tasks with a total duration of 5000 jobs into the input workload. The above-mentioned experimental setup yielded numbers for overall response time and data center processing time. Three approaches, Service Proximity-based Routing,Performance Optimized Routing, and Dynamically Reconfigurable Routing, were used to calculate overall response time and data center processing time.

The proposed load balancing strategy is used in conjunction with existing approaches such as Round Robin, Equally Spreaded Current Execution, Throttled, and Least Frequency Used to analyze the results. Tables 2 and 3 and Fig 10 and 11 show the results for overall response time and data center processing time, respectively.

| | **Proximity based Routing (ms)** | **Optimized Response Time (ms)** | **Dynamically Reconfigured (ms)** |
|---|---|---|---|
| Round Robin | 158.48 | 152.63 | 158.49 |
| Equally Spreaded Current Execution (ESCE) | 152.18 | 153.50 | 157.24 |
| Throttled | 152.09 | 152.79 | 156.96 |

| | | | |
|---|---|---|---|
| Least Frequently Used (LFU) | 151.58 | 151.69 | 155.89 |
| Proposed Load Balancing Policy | 150.38 | 151.18 | 155.6 |

**Table 2**:Comparative Analysis of overall response time

| | Proximitybased Routing (ms) | Optimized Response Time (ms) | Dynamically Reconfigured (ms) |
|---|---|---|---|
| Round Robin | 0.98 | 1.46 | 7.09 |
| Equally Spreaded Current Execution (ESCE) | 0.61 | 2.29 | 5.79 |
| Throttled | 0.48 | 1.49 | 5.46 |
| Least Frequently Used (LFU) | 0.49 | 1.25 | 5.39 |
| Proposed Load Balancing Policy | 0.39 | 1.05 | 5.26 |

**Table 3:** Comparative Analysis of data center Processing time.

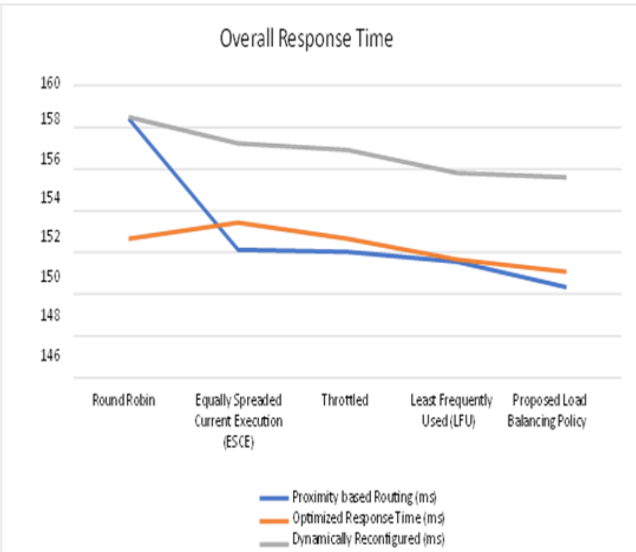


**Fig 10:** Comparison of Overall Response Time

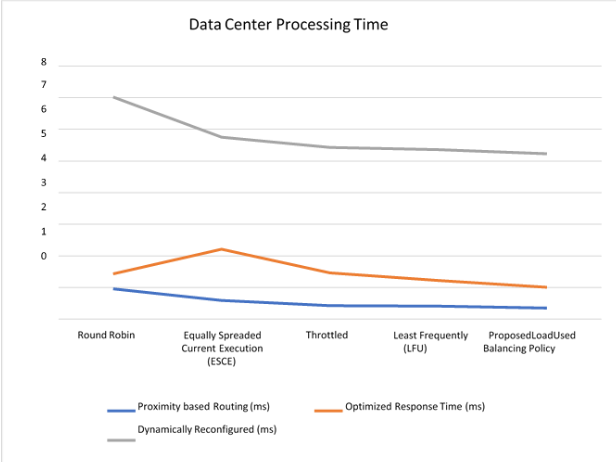


**Fig 11:** Comparison of Data center Processing Time

## 6 . Conclusion and Future work

There is a necessity to build a huge number of cloud data centers because of the drastic increase in internet users that in turn leads to avail high demand for computing capacity. A huge amount of energy is required to operate and manage data centers. In all existing traditional approaches mostly either mips or memory has been considered to improve system performance and to reduce energy consumption. Task submission and VM placement method considerably improve performance on overall response time and data center processing time compared to traditional Round Robin and other scheduling algorithms. The proposed approach has 4% less VM migration and 3% less energy consumption compared to the MadMu method, while it has 2.28% improvements for Mean time compared to the ThrMmt method. Task queues are managed by proposed algorithms that confirm better Quality of Service (QoS) and SLA management.

## References


[1] S. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Energy-Efficient Scheduling of HPC Applications in Cloud Computing Environments," Mar. 2009.

[2] Nureddine, S. Islam, and R. Bashroush, "Jolinar: analysing the energy footprint of software applications (demo)," Mar. 2016, pp. 445–448, doi: 10.1145/2931037.2948706.

[3] J. von Kistowski, H. Block, J. Beckett, C. Spradling, K.-D. Lange, and S. Kounev, "Variations in CPU Power Consumption," Mar. 2016, doi: 10.1145/2851553.2851567.

[4] T.-D. Le, D. Lo, C. Goues, and L. Grunske, "A learning-to-rank based fault localization approach using likely invariants," Mar. 2016, pp. 177–188, doi: 10.1145/2931037.2931049.

[5] R. Trobec, M. Depolli, K. Skala, and T. Lipić, "Energy efficiency in large-scale distributed computing

systems," in 2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2013, pp. 253–257.

[6] "AWS Services." https://aws.amazon.com/.

[7] "Google Cloud Platform." https://cloud.google.com/gcp/.

[8] "Microsoft Azure," [Online]. Available: https://azure.microsoft.com/.

[9] "IBM Cloud." https://www.ibm.com/in-en/cloud .

[10] "Alibaba Cloud." https://in.alibabacloud.com/ Shehabi et al., "United states data center energy usage report," 2016

[11] D. C. Ketankumar, G. Verma, and K. Chandrasekaran, "A Green Mechanism Design Approach to Automate Resource Procurement in Cloud," Procedia Comput. Sci., vol. 54, pp. 108–117, 2015, doi: https://doi.org/10.1016/j.procs.2015.06.013.

[12] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," Futur. Gener. Comput. Syst., vol. 37, pp. 141–147, 2014, doi: https://doi.org/10.1016/j.future.2013.06.009.

[13] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V Vasilakos, "Cloud Computing: Survey on Energy Efficiency," ACM Comput. Surv., vol. 47, no. 2, Dec. 2014, doi: 10.1145/2656204.

[14] M. Pantazoglou, G. Tzortzakis, and A. Delis, "Decentralized and Energy-Efficient Workload Management in Enterprise Clouds," IEEE Trans. Cloud Comput., vol. 4, no. 2, pp. 196–209, 2016, doi: 10.1109/TCC.2015.2464817.

[15] X. Xu, W. Dou, X. Zhang, and J. Chen, "EnReal: An Energy-Aware Resource Allocation Method for Scientific Workflow Executions in Cloud Environment," IEEE Trans. Cloud Comput., vol. 4, p. 1, 2015, doi: 10.1109/TCC.2015.2453966.

[16] V. Cima, B. Grazioli, S. Murphy, and T. M. Bohnert, "Adding energy efficiency to Openstack," in 2015 Sustainable Internet and ICT for Sustainability (SustainIT), 2015, pp. 1–8, doi: 10.1109/SustainIT.2015.7101358.

[17] N. Akhter and M. Othman, "Energy Aware Resource Allocation of Cloud Data Center: Review and Open Issues," Cluster Comput., vol. 19, no. 3, pp. 1163–1182, Sep. 2016, doi: 10.1007/s10586-016-0579-4.

[18] Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," CCGrid 2010 - 10th IEEE/ACM Int. Conf. Clust. Cloud, Grid Comput., pp. 577–578, 2010, doi: 10.1109/ccgrid.2010.45.

[19] D. Aikema, A. Mirtchovski, C. Kiddle, and R. Simmonds, "Green Cloud VM Migration: Power Use Analysis," 2012, pp. 1–6, doi: 10.1109/IGCC.2012.6322249

[20] M. Bala and Devanand, "Performance evaluation of cloud datacenters using various green computing tactics," in 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 956–961.

[21] X. Zhang, Z.-Y. Shae, S. Zheng, and H. Jamjoom, "Virtual machine migration in an over-committed cloud," in 2012 IEEE Network Operations and Management Symposium, 2012, pp. 196–203, doi: 10.1109/NOMS.2012.6211899.

[22] Y. Kessaci, N. Melab, and E. G. Talbi, "An energy-aware multi-start local search heuristic for scheduling VMs on the OpenNebula cloud distribution," Proc. 2012 Int. Conf. High Perform. Comput. Simulation, HPCS 2012, pp. 112–118, 2012, doi: 10.1109/HPCSim.2012.6266899.

[23] H. Chen, F. Z. Wang, N. Helian, and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," 2013 Natl. Conf. Parallel Comput. Technol., pp. 1–8, 2013.

[24] V. Behal and A. Kumar, "Comparative Study of Load Balancing Algorithms in Cloud Environment using Cloud Analyst," Int. J. Comput. Appl., vol. 97, pp. 36–40, 2014, doi: 10.5120/16974-6974.

[25] Y. Chen et al., "Stochastic scheduling for variation-aware virtual machine placement in a cloud computing CPS," Futur. Gener. Comput. Syst., vol. 105, p. 779, Apr. 2020, doi: 10.1016/j.future.2017.09.024.

[26] S. K. Addya, A. Satpathy, B. C. Ghosh, S. Chakraborty, S. K. Ghosh, and S. K. Das, "CoMCLOUD: Virtual Machine Coalition for Multi-Tier Applications over Multi- Cloud Environments," IEEE Trans. Cloud Comput., vol. PP, no. AUGUST, p. 1, 2021, doi: 10.1109/TCC.2021.3122445.

[27] Khosravi, S. K. Garg, and R. Buyya, "Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers," in Euro-Par 2013 Parallel Processing, 2013, pp. 317–328.

[28] Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual Machine Power Metering and Provisioning," in Proceedings of the 1st ACM Symposium on Cloud Computing, 2010, pp. 39–50, doi: 10.1145/1807128.1807136.

[29] Golden and C. Scheffy, "Virtualization for Dummies,"

Sun and AM., J. Bingham and R. Mengle, Eds. USA: USA: Wiley Publishing, Inc, 2008, pp. 6–12.

[30] L. Wang et al., “Cloud Computing: a Perspective Study,” New Gener. Comput., vol. 28, no. 2, pp. 137–146, 2010, doi: 10.1007/s00354-008-0081-5.

[31] Vecchiola, S. Pandey, and R. Buyya, “High-Performance Cloud Computing: A View of Scientific Applications.”

[32] [33]R. Buyya, A. Beloglazov, and J. Abawajy, “Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges,” PDPTA, Mar. 2010.

[33] “Implementing Energy Efficient Data Centers Revision 1.”

[34] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility,” Futur. Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, Jun. 2009, doi: 10.1016/j.future.2008.12.001.

[35] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, “A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters,” Futur. Gener. Comput. Syst., vol. 37, pp. 141–147, 2014, doi: https://doi.org/10.1016/j.future.2013.06.009