Energy Efficiency & Consumption in Data Centre by Dynamic Resource Allocation Technique for Green Cloud Computing

MSc Research Project
MSc in Cloud Computing
(MSCCLOUD1_JAN21)

Tushar Sonawane
X20195753

School of Computing
National College of Ireland

Supervisor: Punit Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Tushar Pramod Sonawane.

Student ID: X20195753.

Programme: MSc Cloud Computing. **Year:** Jan 2022 – 23

Module: MSc Research Project.

Supervisor: Punit Gupta.

Submission Due Date: 1/ 2 / 2023

Project Title: Energy Efficiency & Consumption in Data Centre by Dynamic Resource Allocation Technique for Green Cloud Computing

Word Count: 6468 **Page Count:** 20.

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: 31 Jan 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Energy Efficiency & Consumption in Data Centre by Dynamic Resource Allocation Technique for Green Cloud Computing

Tushar Pramod Sonawane
X20195753

Abstract

Data Centre is a designated area where majority of the computer system, the accompanying storages, and other network equipment is placed in a number of racks. Depending on the demands, these racks are often removable and easily serviceable. Since the number of users is now growing exponentially, more processing power must be made available to them. Nowadays, since so many people rely on cloud computing and storage, enormous data centers with lots of equipment are needed. However, because maintaining all of this equipment is not always simple, it is important to make the most of these resources. In the research we've suggested, we're concentrating on reducing energy use while assigning the same amount of computing power. In this manner, the ideal number of resources on the data center are maintained operationally, while all other resources are either shut down or placed in standby. In order to reduce the carbon footprint of these ever-growing data centers, we are recommending dynamic resource allocation technique in order to implement an energy-efficient device reallocation. Green Cloud Computing benefits from energy-efficient data centers because they provide quicker data services with less energy use.

1 Introduction

A data center is a dedicated area where several computing servers and all the other equipment, including routers, memory, power supply, and other things are located. Even if there are several data centers established worldwide, resource distribution is typically sequential and dependent on the installation date. The infrastructure of the data center is not disturbed when new hardware is connected. Although easy to use, this strategy is ineffective. Cloud applications and storage are pushing an increasing amount of data onto data centers, which has a significant carbon impact. The data center is increasing every day and now uses around 3% of the energy generated. There is a lack of awareness of energy-efficient distribution in the current methods for placing data resources in the data center. We provide a method for effectively utilizing all of the data center's resources while continuously attempting to lower the amount of power the facility uses. Due to the dynamic resource allocation, since our data is time-dependent, it will alter the timestamp when changed. Our research intends to create Green Cloud Computing by allocating data center resources to a specific real-time dynamic operation. The resources of data is taken into account include computer system, virtual machines (VM's), display units (monitors), SSD or HDD storage, VM allocators, user interfaces, and the resource analyser. This project's overarching goal is to cut down on the resources used by each individual application operating on the data center's cloud. Additionally, We're interested in learning whether using the least resources is typically the most energy-efficient option or whether there is another approach that is more successful than using the least data center resources.

Assigning data to resources is the best way to make a data center work with quicker speeds, that might or might not lead to greater efficiency. Consider two data resources, one made up of sluggish optical disks that use 1W/hr and the other of an SSD that uses 2 Watt per hour. If an application runs for 4 hours on a sluggish optical drive, it will use a total of 4 Watts of power; but, if the same application runs for 30 minutes on an SSD, it will use a total of 1W/hr. Because of this, it is not always obvious which resources will result in the data center's low electricity consumption. In order to optimize, it is necessary to ascertain the speed of operations and perform certain anticipatory upcoming tasks. Data collecting requires knowledge of the data center's location because its surroundings have an impact on a number of variables, primarily the environmental temperature, which accounts for the majority of energy use. The installation of data centers in cold, controlled environments is advised at the moment, the bulk of data centers are situated close to the arctic ocean, and some situated in the sea. Normally, data center are cleaned right before setup to ensure that the performance won't be impacted by ambient air or other airborne debris. In some circumstances, a positive pressure room is constructed or an inert gas is requested in the data center so that people can enter and exit with the proper personal safety equipment. Storage devices, computer hardware, communication devices, connectors, power backup, cooling equipment, and racks to house this device are the usual components of a data center. The vast bulk of this data center's software is written in high level or low level programming languages such as C, Java, or Python and is supported by a Linux backend. Our research aims to first comprehend the performance and power consumption of each component in the data center. The Energy Efficient Data Center Resources Allocation Algorithm conceptual diagram is shown in Figure 1, with a variety of data center resources, including hard drives, networking devices, power devices, computing machines, racks, etc. The entire amount of resources for each type of activity is calculated, and following EEDCRA, the rest are placed on Standby, and just the resources that are absolutely necessary are kept on.

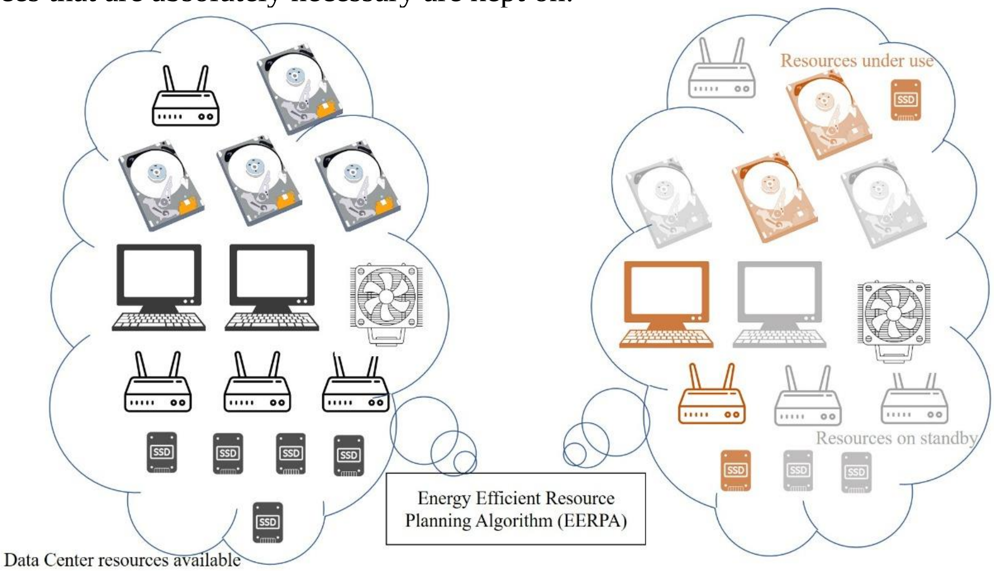

Figure 1: Energy Efficient Resource Allocation Algorithm

We may simply assign workloads utilizing the fewest resources, as shown in the orange hue, which we have designated as resources in use, while other resources in grey are on standby, as seen in the above Figure 1, which shows data center resources that are vacant. As the silicon production and technology industries have expanded, the processing center spaces have begun to get smaller, but at the same time, the number of people using computers has been fast rising. Calculations were initially being done on separate computers by each user, but as the telecom business increased, so did consumer demand for in-home computing (cloud storage). Datacenters are the autonomous, dedicated locations for all cloud computing platforms, together with the corresponding fiber optic and power management networks. Typically, one or more administrators are assigned to the data center. Usually, the data center has one or more administrators assigned to it. For the past 20 years, cloud computing and storage services have been acknowledged as such.

1.1 Background

As the IT sector grows every day, the components of green cloud technology of energy consumption, cloud application deployment, power distribution systems, and power techniques work together. Utilizing resources as effectively as possible while developing, producing, and deploying a system is the core goal of green computing. The strategy should be constructed in a way that leaves a very small carbon footprint while having no effect on the data center's performance. Up to 80% of the energy was used by Azure, AWS, Google Cloud Platform, and Oracle Datacenters in 2022, and this percentage is predicted to increase by at least 32% annually. If its data center does not perform resource allocation. In order to save a significant amount of energy, it is crucial that all the resources in the data center are utilized effectively.

People have ceased purchasing a large number of storage devices and have instead begun investing more in cloud services as we have all migrated to cloud storage, where substantial investments are being made to give the user more and more memory. In order to function, all cloud services must have a data center that is operational around-the-clock, continuous mobile antenna transmission, and wireless transition reception via huge optical fibers. It is clear that as cloud computing expands, the earth will feel the effects more strongly due to its carbon footprint. In 2012, the idea of "green computing" was introduced. When manufacturers spent a lot of money on creating environmentally friendly manufacturing processes and getting rid of hazardous computing equipment, one of the biggest challenges facing the IT industry is how to run the world's datacenters, which account for 3% of all energy consumption and between 1.8 and 3.9% of greenhouse gas emissions globally. The benefits of the cloud cannot be negated, so it is impossible to stop using cloud services, shut down data centers, and live swap all the resources while maintaining data access on backup devices. When a data center is established, servers typically have a lifespan of 10 years, which is a significant amount of time for any electronic device. By using efficient resource allocation, one may concentrate on computing.

1.2 Research Question (RQ)

Since data center include incredibly minute details about a cloud operation, many of the resources inside a data center are kept operational in a redundant manner. It is crucial to offer a method that is energy-efficient and can accurately determine how many resources are needed in the data center. Our goal is to determine how much energy is saved by using our proposed technique as opposed to the conventional approach by developing a resource-optimized datacenter that can hot swap the necessary computing resources as needed.

1.3 Document Structure:

Table 1: Document Structure

Section No	Section Name	Content
1	Introduction	Energy Efficiency & Consumption in Data Centers via Dynamic Resource Allocation Technique for Green Cloud Computing is introduced. It also gives a basic overview of the necessity of reducing carbon footprints. Finally, the paper justifies how the advancement in technology and the study proposal might further improve the total energy efficiency and consumption in data centers. It also covers the present systems in place and how they have been underutilized.
2	Literature Review	Consist of the whole body of prior research in this area is covered. Additionally, it provides a comparative analysis of the works of literature on resource allocation and green cloud computing.
3	Research Methodology and Specification	The strategy used to accomplish the objectives specified in Section 1 is described in Section 3 in detail. It begins by explaining the system's block diagram. It also describes the procedures that is been used in application development. It also describes the techniques utilized to leverage green cloud computing to achieve energy efficiency and consumption. Additionally, it shows a expected results of Energy Consumption.
4	Conclusion	The research project comes to a conclusion. It gives a summary of all the activities carried out during the research and the results expected from each one.

2 Related Work

During the process of doing literature research, we learned about a variety of strategies used in Green Cloud Computing for Energy Consumption and Efficiency in Data centers using Dynamic Resource Allocation. By using virtual machines and dynamic resource allocation while concentrating on the many methods and techniques developed by researchers, we sought to provide the optimum allocation strategy.

2.1 Virtual Machines for Green Cloud Computing

Singha et al. (2022) had a review of different green cloud computing strategies. Additionally, they focused on issues like high availability, data center monitoring, part reusability, and other related issues. Despite the fact that their study was merely a review of previous work, controlled scripts that merged the ideas of energy conservation and cloud computing were used for the first time.

Sharma et al. (2022) offered a concept for green cloud computing for visual machine mitigation. In their article, they outlined different Virtual Machine mitigation strategies. They also gave a brief overview of the several mitigation techniques used by scientists. Even though they only recently added the poll on various virtual mitigating strategies. Based on their energy usage, resource usage, numbers of virtual servers, outage issues, and differences with respect to active equipment and violations, they compared these papers. 93% of the literature review for green cloud computing, according to their report, focuses on energy usage.

2.2 Migration and Allocation of Resources.

Dhaya et al. (2022) presented energy-efficient resource migration and allocation. For implementation purposes, they have shown the output on a private cloud data center. Bandwidth, system integrability, and Quality of service were considered when making decisions in all 4 different deployment sessions where the entire schedule time range was measured. The outcomes for all three algorithms were reported after three methods were compared. The key challenge was that even though their study was primarily concerned with scheduling algorithm in a private data center, the output runs were inconsistent and varied despite the fact that they adapted the result with four scenarios. It was therefore unclear why the suggested algorithm needs to be used consistently for green cloud computing.

Jeba et al. (2019) in their research proposal, they used a GCC technique to divide a single virtual machine into two independent stages, hence reducing the entrance consumption for the cloud data centers. Using an activator engine, the second step of mitigation was launched after static data regarding the quantity of hosts, services, and consumption on the average throughput was pre-processed. In order to consolidate servers, they have employed virtualization. Initially, resources are gathered, and after this information is understood, virtual machine sizing is managed, following the initial VM sizing placement decision, the resource sharing techniques are shared with the server's location and newly activated hotspots, and then the resizing of virtual machines is carried out. Following resizing, the VM is eventually remapped to a physical machine (PM) in accordance with the placement strategy. The server consolidation is done by VM migration. They have used a total of 10 2.7GHz dual-core Windows PCs for this purpose. The energy usage was measured in KW/hr and the number of servers ranged from 3 to 10. They were able to surpass the 14.44 KW/hr of BFV VMCL in the literature by producing up to 10.5 KW of energy.

Theign et al. (2020) proposed when calculating the effectiveness of power usage, the efficiency of the data infrastructure, and the percentage of host VM utilization over a period of more than 7 days, the energy-efficient resource allocation in the cloud datacenter utilized up to 200 virtual machines. For resource allocation that was energy-efficient, they used a level of service manager, an energy-efficient data center analyser, and a managed hosting resource monitor.

2.3 Multicore Allocation to Virtualizations.

Sukur et al. (2020) proposed the idea of cloud computing Hyper-V allocation of resources for a distributed system of computers has been applied. For resource allocation that was energy-efficient, they used a level of service manager, an energy-efficient data center analyzer, and a managed hosting resource monitor. Their strategy centered on balancing the workload while actively meeting the needs of the client. They have demonstrated how to leverage programming languages like Java and MATLAB along with simulators like CLOUD SIM and GMETAL to achieve the energy effect configuration. as per the review of the lecture, in various distributed systems. In their paper, they provide an overview of 16 different systems that operated between 2014 and 2020 and used various resource allocation strategies. even when their work received positive reviews, In our research study, we look at the statistics-based findings to see how much greater energy efficiency our method achieves. They provided no quantitative data to support the facts for the earlier resource allocation methods.

Yuvan et al. (2018) surprise the cross-talks on multicore fibers, a resource allocation plan was developed after extensive investigation. They evaluated a data center with seven to sixty-one cores utilizing eight different types of MCF (multi core fiber) and 3 distinct multiplexing systems with two different traffic scenarios. Researchers were able to increase overall capacity and increase connection area efficiency by up to 33%. The only resource they used to optimize in terms of cores and how transmission distance affected the transmitted power were multicore fibers. Although the strategy of selecting the optimal core to transmit the signal is intriguing, the total execution of such a device in practice is exceedingly challenging results were supposed to be based solely on two unreliable scenarios results were supposed to be based solely on two unreliable scenarios. We would further investigate the impact of multiple core fiber (MCF) on the data center's back hole in accordance with our research proposal. The smallest number first technique will be investigated for other data center devices even if optimizing the number of fiber cores is outside the purview of the proposed research.

Wang et al. (2018) has researched for the best resource distribution in the data center, and his group choose to employ virtualized network function. They tested the system using a three-tier virtual network design, introduced some restrictions, and needed eight regions with five tiers of resource distribution. Their first-tier edge nodes had a latency of 0.1 ms and were located within 20 km. Their intermediary transmission had a latency of 2.5 milliseconds and was within 500 kilometers. Their previous cloud design had 1.2 millisecond response times and could reach distances of up to 250 kilometers. Customers were able to achieve latency ranges between 1 and 5 milisecods, which is respectable, but the total cost could not be reduced under 4 for the 6 ms delay limit. It was predicted that their intermediate fog network would have a distance range of up to 500 km, which is far too distant for any device to communicate without making several hops. The data center itself is the third tire of the cloud, hence we refrained from using the 3-tier architecture. In our proposed method, we will only look into their visualization-based heuristic model for resource allocation.

Zhao et al. (2019) reported a highly heterogeneous virtual machines analyzer created specifically for 5G Edge Data Center was employed. The number of servers was decreased from 78 to 72, which represents an improvement of about 8%. They used slicing as the basis for their VM allocation technique and made the premise that each original VM might generate an equal virtual machine with a constant amount of computing power. Additionally, we would like to enhance resource allocation overall by at least 8% over their efforts. They only concentrated on one kind of resource, whereas our method implies that every single component of the virtual machine can be temporarily detached.

2.4 Optical Switching and dynamic traffic prediction.

Balanici et al. (2019) created a resource allocation plan for a data center using traffic prediction utilizing optical switching. As the amount of epochs rises, their strategy focuses on anticipating the traffic and lowering the MSE mean error function. They utilized 106 data points and performed timing- and prediction-phased resource allocation. They obtained testing coefficient correlations of 0.9231 and learning coefficients of 0.9998. Their entire methodology involved anticipating high traffic data and on-demand cloud deployment. Despite the fact that their strategy was solely focused on traffic and that cloud deployment was the only overall resource allocation, If the forecast is off, it might not succeed.

Yi et al. (2020) concentrated on using the virtualized data center with network function virtualization, recommended implementing the centralized control physical resources, and suggested an unique resource allocation method based on software-defined networking (SDN). They set the server apart from other resources as well. In contrast to the state-of-the-art technology, the overall cost of mapping increased by 71% while the total cost ranged from 25% to 10%. However, they still need to be improved upon as just 31% of people accepted them. Their concept of software-defined networking is quite intriguing and can be investigated in the study we've suggested. Overall, their dismal outcomes show that there is still much to be done in this area before any durable solutions are implemented.

Ekici et al. (2019) built a data center resource allocation system that is hardware-accelerated and based on power usage. They used the MATLAB simulator to apply the suggested model and plot time versus the number of real machine turns on and time versus power usage for performance study. Additionally, they showed that, in contrast to their algorithm, resource allocation is influenced by actual computers, time complexity, and total power used. With a resource allocation, we'd like to investigate their computing resources and associated resource logic paradigm based on vector programming for integers.

Ouhame et al. (2021) proposed the use of a convolution neural network and long short-term memory (LSTM), developed a resource-efficient resource utilization system that allowed for improvements in accuracy and error performance of 3.8% to 0.9% and 7 to 8.5%, respectively. Their forecasting data is fed into a convolution neural network, which creates an LSTM input with a single function. Similar to how we do our task, we also employ the historical data record to analyze the amount of energy used for each active action. The LSTM decides whether or not to use the resource based on historical data. To establish whether the indicated resource must be used or whether it may be put on standby, we also want to look into CNN and LSTM networks as part of our proposed study approach.

2.5 Efficient energy aware allocation of resources.

Akhter et al. (2016) has been suggested to do a complete summer study on using cloud-based data centers to allocate resources in an energy-conscious way. They looked into a variety of tactics, concentrating on resource sharing, on-demand services, and customer billing while also looking into emergency personnel, virtual machine monitoring, service-level agreements, and customer broker systems. The detailed flow diagram they provided divides the entire resource allocation process into energy monitoring, heat dissipation, and dedicated physical and machine level resource allocation. They have outlined the requirements for resource allocation, including task descriptions, system energy consumption models, and data collection tools for user inputs. They suggested utilizing virtual machines.

Huang et al. (2021) A technique that permits data center virtual machine instances in accordance with user needs was developed. This technique makes it possible to schedule virtual machines while lowering the real number of PM needed. The VM was tested from 100 to 5000, and the CPU usage was at least 10% higher. They could utilize 192 GB RAM and 56 CPU cores running at a maximum power 347 Watts. Also 8 CPU and 16 GB RAM, they used the least amount of electricity (16.3 Watt). Around 88 physical machines (PM) were needed in all, with 10 being the absolute minimum. The lowest amount of energy they could use per hour was 28.4 KWatt. Even though they had utilized a number of VM for effective usage, we believed their PM count to be higher than average for the amount of energy consumed. We would attempt to install a comparable amount of VMs in our research proposal, but with less PM(Physical Machine). They haven't given any thought to the cooling system, power supplies (SMPS), or hard drives, which may have saved a significant amount of energy. By addressing this research gap and utilizing all of the data center's resources, as opposed to just one, we hope to fill it.

2.6 Literature Review Table.

Each approach used in the literature review is distinct and has pros and cons for handling dynamic resource allocation in green cloud computing. With a focus on the literature we looked at, we created Table 2 to identify the relevant techniques and algorithms in the order that they were first proposed. Each algorithm and technique addressed in our work is briefly summarized below, along with a description of its key development principle, benefits, and drawbacks.

Author	Year	Method	Energy saving (%)	Experiment conducted	Number of VM's	Run Time (hours)	Interprocess Operations	Acceptance Rate %
Jeba et al.	2018	GCC Algorithm	30	--	--	--	--	--
Dhaya et al.	2021	Scheduling time span	--	4	--	--	--	--
Thein et al.	2018	Reinforcement learning	--	--	200	168	--	--
Shukur et al.	2020	Client Requirement based Scheduling	--	16	--	--	--	--
Yuvan et al.	2018	Space division multiplexing	33	3	--	--	--	--
Wang et al.	2018	Virtualized Network Function	--	4	--	--	1.5 milliseconds	--
Zhao et al.	2020	Slicing	8	--	78	--	--	--
Balanici et al.	2022	Traffic Prediction based optical switching	--	--	--	--	0.9431 milliseconds	--
Yi et al.	2020	SDN	71	--	--	--	--	31
Ouhame et al.	2021	LSTM CNN	8.5	--	--	--	--	10.9
Akhter et al.	2018	VM Scheduling	--	--	5000	--	--	10

Table 2: Reviewed Algorithm & Method.

3 Research Methodology

As cloud technologies advance, data centers are coming under increasing pressure to address concerns with energy usage, load distribution, performance, and application deployment. Despite the fact that many cloud service providers claim to have decreased total carbon footprints, they never address data center energy use. Infrastructure and its use are a concern as well because they are crucial to data centers. The best utilization of virtual machines for green cloud computing has been demonstrated by researchers including (Sharma et al.) & (Singha et al). Energy-efficient resource allocation was recommended by (Dhaya et al.) and (Jeba et al.) as a technique to support green cloud computing in their work.

In order to develop our unique Energy Efficient Algorithm, we tested a conventional genetic algorithm in combination with a grasshopper algorithm. The data center is where all of the information is gathered. At first, parent 1 and parent 2 were assigned at random, and the solution and associated results were taken into consideration as the cities of parent 1 and parent 2, respectively.

3.1 Block Diagram

The new solution is computed after mutation, which is a little updating of the resources allotted, and each of these computations is referred to as an event in the tournament with a certain city. Once the best solution has been found, the child moves on to the next city and repeats the evaluation. Once all of the cities in the tournament have been evaluated, the fitness of each city is computed. based on the mutation the child underwent during the cross-over. The final solution is then determined to be the value with the best fit.

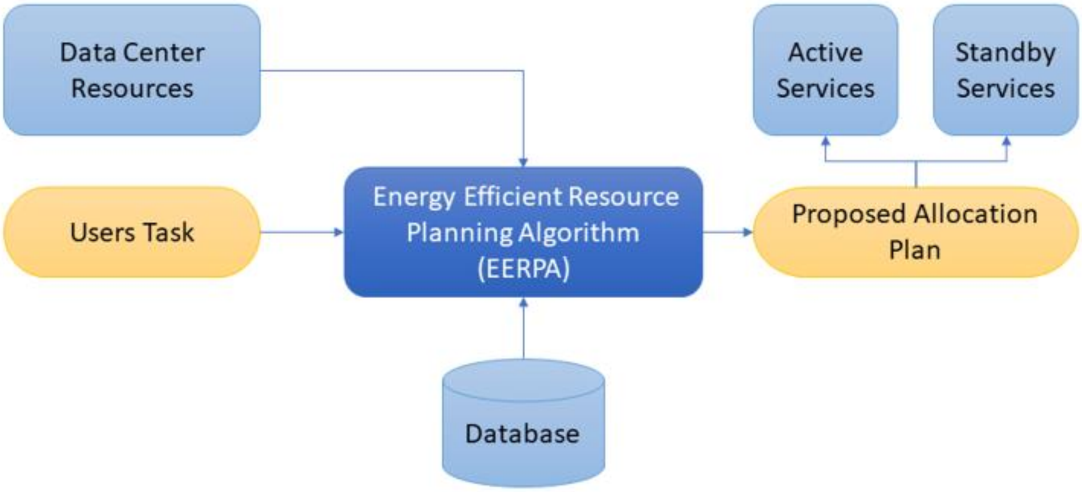

Figure 2: Energy Efficient Resource Planning Algorithm (EERPA)

The workflow diagram shown in Figure 2 shows how the data center resources, the user's current task, database information about earlier actions and their related energy requirements, and all of these are delivered to EERPA. The EERPA algorithm typically determines the ideal number of resources and presents the user with a recommended allocation plan. The data center administrator may then set the devices to either their active stage or standby state in accordance with the proposed allocation plan.

All low value tasks are combined with a time-critical work to establish an ideal seed point. Availability of resources or first come, first served will better understand the importance if there are multiple time-sensitive jobs. The information is used by data center resources like hard disks, network devices, etc. for EEPRA. Application receives a user task as input, and then generates data and resource planning based on the database. A request for active and stand-by services is issued to the data center administrator in accordance with the allocation plan.

3.2 Workflow

Figure 3 presents the workflow for assigning resources in a data center with a focus on energy efficiency. A GUI is utilized to gather information about user tasks, hardware and software resources, and administrator actions. Upon receipt, the data is directed to a hardware resource allocation algorithm aimed at maximizing energy efficiency regardless of the task. The algorithm is tested using historical data from a dataset after improvements in energy efficiency have been made. The optimization process will continue until every component in the data center is set to Active or Standby.

A Genetic algorithm and Grasshopper optimizer were combined during optimization to update and replace results with better ones. The best solution is preserved through Elitism, even after each grasshopper jump. The program will terminate when all tasks are completed, or else it will continue running. To ensure the best outcomes, the workflow must be followed as planned. It is critical for software resources information to be inputted first so that hardware allocation can be synchronized with user tasks. The database inputs are important for training datasets and the energy compute module monitors energy consumption to determine task completion before reassigning to users and continuing the flow.

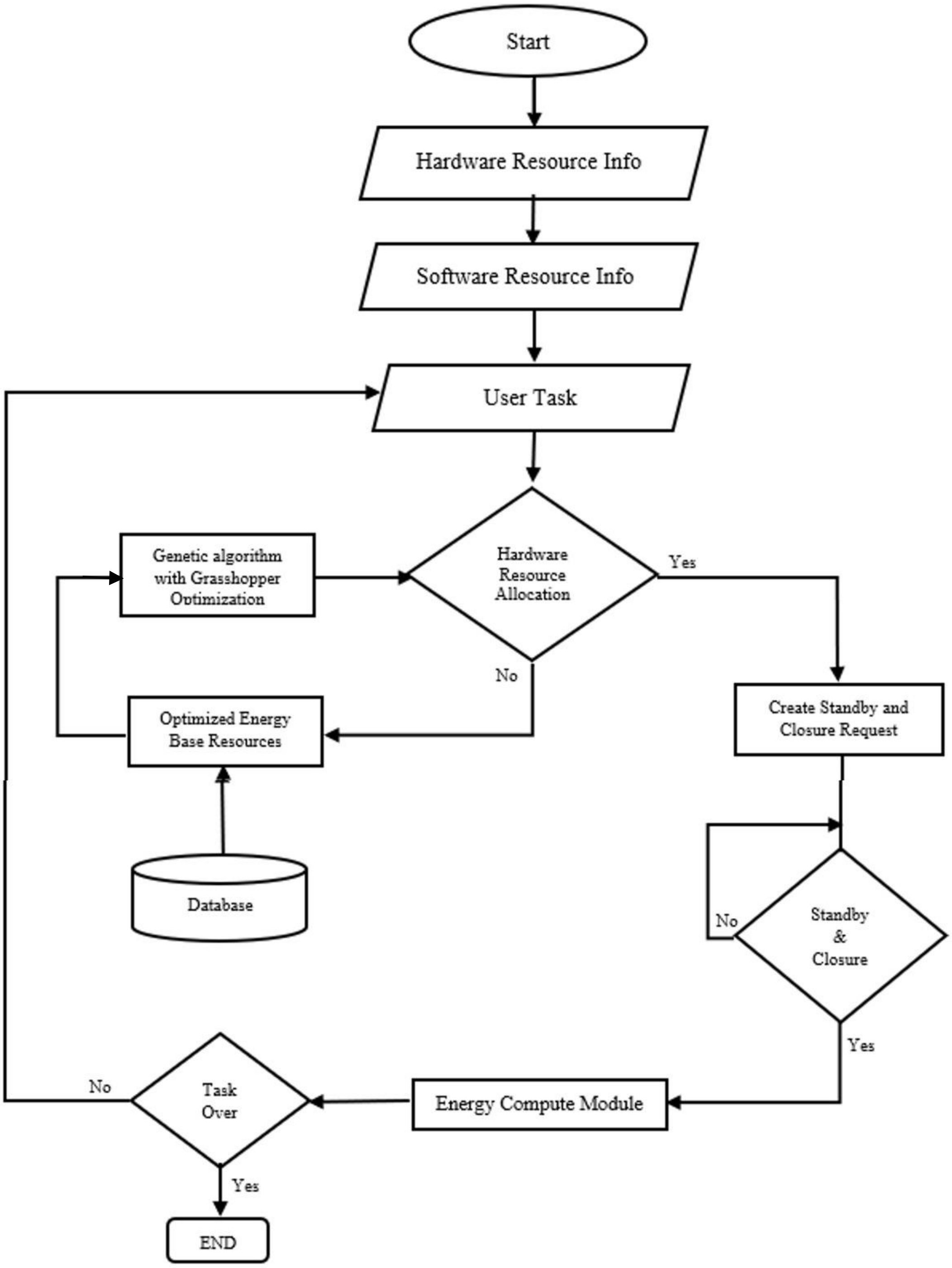

Figure 3: Proposed workflow of Energy Efficient Resource Allocation Algorithm

3.3 Proposed Algorithms

The suggested algorithm leverages the database's current input to determine which tasks to assign the information, based on the database's history. The majority of the present resource allocation approaches described in the literature have concentrated on tried-and-true methods that excel with static data nevertheless, in our suggested mechanism, we will gather user tasks, on the basis of which resource estimation will be based. It is crucial to discover each task separately and run them to ensure that performance is unaffected in order to have an optimal resource allocation that should not compromise performance. The distinct runs come to an end if the performance begins to suffer.

Algorithm 1 Energy Efficient Data Center Resources Allocation Algorithm (EEDCRA)

Require: $Hardware\ information$ (CPU, HDD, N/W devices, Power devices, Cooling System)
Require: $Software\ information$ (VMs, Schedulers, Watchdog timers, Energy monitor)
Require: $Current\ USER\ Task$
  $CPU_i \leftarrow CPUsinformation$
Ensure: $HDD_j \leftarrow Harddisks\ information$
  $NW_k \leftarrow Networking\ Devices\ information$
  $PD \leftarrow Power\ Devices\ information$
  $CL \leftarrow Cooling\ system\ information$
  $Sft \leftarrow Software\ information$
  $UT_n \leftarrow USER\ Tasks$
  $N \leftarrow Total\ Number\ of\ Task$
  $ET \leftarrow Total\ Energy$
  $OTE \leftarrow Optimized\ Total\ Energy$
  while $n > N$ **do**
    if $ET > OTE$ **then**
      if Last tour city Current city **then**
        $parent_1(m) \leftarrow random(Allotment)$
        $parent_2(m) \leftarrow random(Allotment)$
        $Child \leftarrow MutationParent_1, Parent2$
        $Parent_1, Parent_2 \leftarrow newcities$
        $Newchild \leftarrow increamentcurrentcity$
      end if

Each unique run is allocated one resource, and asper the specifications, the unidentified task, for which the optimal number of resources is unknown, must be completed using all variations and combinations of virtually simulated resources. The administrator will have complete authority to alter the suggested resource allocation plan for resource management. For a better understanding of the method and its step-by-step explanation, the application's proposed algorithm is detailed. Each task must be successfully completed by following every step in the algorithm.

Inputs: Information about hardware, including details about the CPU, hard drive, network devices, power devices, etc. Software information includes schedulers, temporal watchdog clocks, and energy measurement software, as well as virtual machines' allocation in physical computers. The number of user tasks and their details are listed in the next several phases.

- Step 1: CPU information is allocated to the variables *(CPU_i)*.
- Step 2: Allocate the hard disk information to separate variables for each hard disk *(HDD_j)*.
- Step 3: Assign the networking device information's to separate variables *(NW_k)*.
- Step 4: Assign all the power device, attached information to variables in PD power devices where PD is the array of double size.

- Step 5: Assign all the cooling information such as fan speed fan type material in cooling etc. to *(CL)* variable *(CL)* is a two-dimensional array of type double.
- Step 6: Collect the input user task and assign number to each task resulting into a *1xN* type array.
- Step 7: Calculate total energy based on the values in the different variables reported in *Step 1 to 6*.
- Step 8: Compute the Optimized Total Energy *(OTE)* required.
- Step 9: Initiate maximum number of CPUs for faster operations so that the task is completed in minimal time. Here a trade of between number of CPUs and Time required for execution is very required. It is recommended to reduce the number of CPUs one at a time.
- Step 10: Change the VM & Host size in constant file and check the results.
- Step 11: Again, check if parent1 is in same city of tournament or in the different city that of child.
- Step 12: During each tournament the candidates was allocated to random tour just to ensure that there is zero biasing while allocating the data.
- Step 13: The tournament is selected in random manner and fitness tournament is selected.
- Step 14: **Selection:** Select two parent chromosomes from the population based on their fitness. This can be done using a selection operator such as Roulette Wheel Selection.
- Step 15: **Mutation:** Apply mutation to the selected parent chromosomes to generate two offspring chromosomes. Mutation can be performed using a mutation operator such as Random Resetting.
- Step 16 **Crossover:** Apply crossover to the two offspring chromosomes to generate two child chromosomes. Crossover can be performed using a crossover operator such as Single-Point Crossover.
- Step 17 **Update Population:** Replace the worst chromosomes in the population with the child chromosomes.
- Step 18 **Grasshopper Optimization:** Perform the Grasshopper Optimization on the child chromosomes by applying the Lévy flight-based movement on each dimension of the chromosome.
- Step 19 Repeat steps 15 to 18 until the termination criteria are met.

Grasshopper Optimization (GO) is a metaheuristic optimization algorithm that is inspired by the foraging behavior of grasshoppers. The algorithm is used to solve optimization problems by finding the optimal solution among a set of candidate solutions. GO works by representing candidate solutions as "grasshoppers" that move randomly in the solution space. The movement of each grasshopper is influenced by the quality of the solution at its current location, as well as the proximity of other grasshoppers. Over time, grasshoppers that encounter high-quality solutions tend to aggregate in that area, while those at lower-quality solutions move elsewhere. The algorithm repeatedly updates the position of each grasshopper, until a stopping criterion is met, such as a maximum number of iterations or a satisfactory solution is found. The final result is the best solution discovered by the algorithm.

GO has been applied to various optimization problems, including function optimization, feature selection, and scheduling problems. The algorithm has been shown to be effective and efficient in finding optimal solutions compared to other optimization algorithms.

The proposed solution utilizes pre-defined packages provided by CloudSim to simulate the resource allocation in a data center environment. The program creates multiple random solutions, called tours, and then improves them using genetic algorithms and operations such as cross overs and mutations. These solutions are then modified according to the grasshopper algorithm, including variations in mutation rate and implementing the concept of "elitism" (preserving the best solution) to achieve the optimal solution.

3.4 Data Flow Sequence Diagram

Figure 4 shows a flow sequence for allocating resources in a data center in an energy-efficient manner. The customer job, along with details about the software and hardware is given into the energy consumption database's resource allocation manager and algorithm.

One of the software processors is then advised of the conclusion of the closure after the data has undergone in-depth analysis. A standby allocation message is delivered to the hardware controller for different hardware configurations if the closure has been successfully completed, and a verification signal is sent back to the resource allocation management for successful closures. This data is then saved in the energy usage database based on the appropriate information from the hardware management to the resource management. The energy compute monitoring method is then supplied the actual software and hardware needed to conduct the user-defined task. The energy compute module is also given a resource allocation plan so that the efficiency with and without may be determined. The output is then provided together with the final resource allocation strategy and the overall percentage of energy saved.

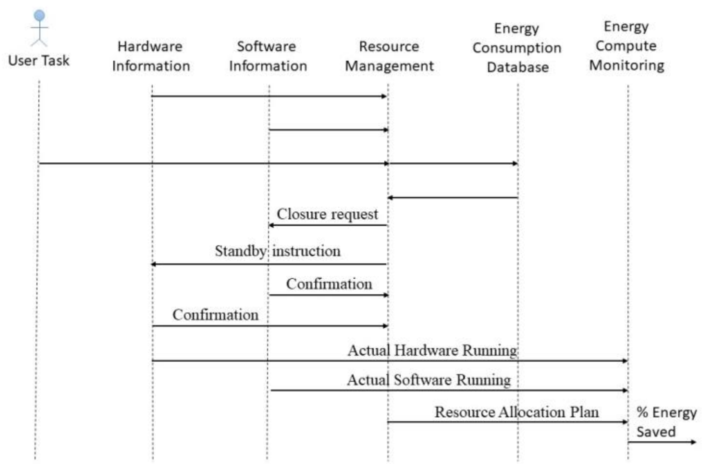

Figure 4: Proposed sequence for Energy Efficient Resource Allocation.

4 Design Specification

The study describes the specifications that were utilized to create the project that was suggested. The system's specification is given in Section 4.1. Section 4.2 describes the system's architecture, while Section 4.3 describes the system’s execution of the system's singular execution.

4.1 System Specification:

A new swarm intelligence system called the Grasshopper Optimization Algorithm (GOA) was inspired by the feeding and swarming activity of grasshoppers in the wild. The GOA method has proven its worth in the literature by being effectively used to tackle a variety of optimization problems in a number of fields.

Pseudocode for Grasshopper Optimization (GO) algorithm:

- *Initialize the grasshopper population (i.e., candidate solutions)*
- *Evaluate the objective function for each grasshopper*
- *Sort the grasshoppers based on their objective function values*
- *Repeat the following steps until a stopping criterion is met:*
- *Update the position of each grasshopper using the following formula:*
- *New_position = Current_position + a * random_number * (Best_position - Current_position) + b * random_number * (Global_best_position - Current_position)*
- *where "a" and "b" are weight parameters, "random_number" is a random number generated in the range [0,1], "Best_position" is the best solution found so far by the grasshopper, and "Global_best_position" is the best solution found so far by all grasshoppers.*
- *Evaluate the objective function for each grasshopper at the new position*
- *Update the best solution found so far by each grasshopper*
- *Update the global best solution found so far by all grasshoppers*
- *Return the global best solution found as the final result*

The GO algorithm starts by initializing the grasshopper population with a set of candidate solutions. Then, it evaluates the objective function for each grasshopper to obtain its fitness value. The grasshoppers are then sorted based on their fitness values, with the best solutions at the top. In each iteration, the position of each grasshopper is updated based on the formula mentioned in the pseudocode. The new position is a combination of the current position, the best position found so far by the grasshopper, and the global best position found so far by all grasshoppers. The objective function is then evaluated for each grasshopper at the new position, and the best solutions found so far are updated.

The process repeats until a stopping criterion is met, such as a maximum number of iterations or a satisfactory solution is found. The final result is the global best solution found by the algorithm.

4.2 System Architecture:

An open-source framework called CloudSim is used to model cloud computing services and infrastructure. It is created by the CLOUDS Labs and is totally written in Java. In order to replicate tests and outcomes, it is used for simulating and modelling a cloud computing environment as a way to assess a hypothesis before software development.

We are working on the VM Service Layers and the part include the VM Management and allocation of VM Machines as highlighted in Figure 5 below, which are best fitted to our research project. We have considered layered architecture of CloudSim for better understating. There are varios parameters of CloudSim which are listed below.

- **CloudSim Simulation Engine:** Offers interfaces for the control of virtualized datacentre resources including bandwidth, memory, and VMs.

- **CloudSim Layer:** The formation and operation of fundamental entities like VMs, Cloudlets, Hosts, etc. are managed by the CloudSim layer. Along with resource supply, execution, and administration, it also takes care of network-related execution.
- **Datacenter:** Modelling the Datacenter, the core hardware component of any cloud system. This class offers ways to establish the allocation rules for the VMs, the Datacenter's functional needs, and other things.
- **Cloudlet:** Any task that is executed on a virtual machine, such as a processing task, memory access task, file updating task, etc., is represented by a cloudlet class. It has methods that are identical to those in the VM class and maintains parameters describing a task's length, size, mi (million instructions), as well as its execution time, status, cost, and history.

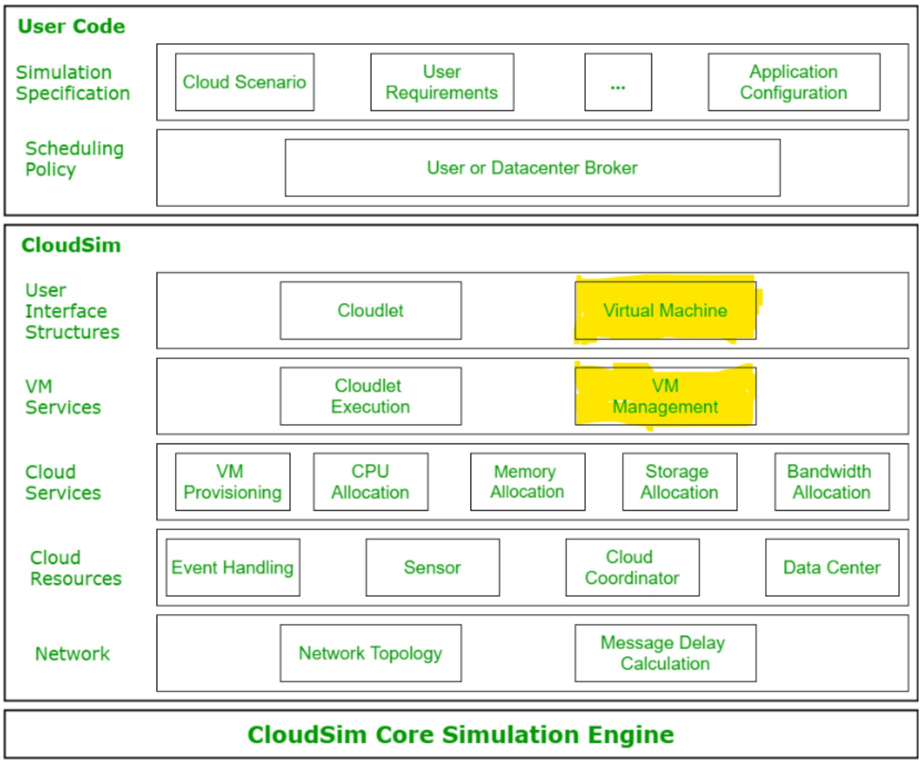

Figure 5: CloudSim Layered Architecture.

- **Datacenter Broker:** A service that represents the user or consumer. It is in charge of how VMs work, including how they are created, managed, destroyed, and how cloudlets are sent to them.

4.3 System Execution:

The **Configuration Manual** provides an illustration of the whole step-by-step process. The manual also includes the process flow and instructions on how to set up the virtual datacenter with the VMs, and Hosts. It also includes information on how we developed the algorithm and the results for energy consumptions.

5 Implementation

In order to model datacentre resource allocation and obtain results for energy consumption and efficiency, we used CloudSim. We suggested a modified genetic algorithm with a 15% maturation rate for resource allocation. The tournament size is set to 5 for the imitated solutions. ELITISM is initially set to 0 for fitness, and once fitness is attained, it is updated with the appropriate population size.

In our scenario, we consider two parents and one child to be a single computation and the two parameters determine how many parents and child nodes need to be computed. After each repetition, the population size matures to produce new genetic material. The crossovers are applied to a set of parents to produce the offspring. The allocation of parents and child is computed at random according to the positions of the parents in the iteration. The child moves to the next city if the estimated population is higher, signalling a new allocation of resources at the data center. The child stays in the city if the error is inside the margin, which means the initial random distribution of resources is carried over to the following iteration. If the child is coming from parent 1 city and there is a change of city, parent to the city is preferred.

while $n > N$ **do**
  if $ET > OTE$ **then**
    if Last tour city Current city **then**
      $parent_1(m) \leftarrow random(Allotment)$
      $parent_2(m) \leftarrow random(Allotment)$
      $Child \leftarrow MutationParent_1, Parent2$
      $Parent_1, Parent_2 \leftarrow newcities$
      $Newchild \leftarrow increamentcurrentcity$
    end if

Figure 6: Logic behind selection process of the algorithm.

If both parents are from the same city, a new city is added to the child's list of possible destinations (random resource allocation). If the kid is not given the new city, then the parent of the city is assumed to be the final city in this situation. The child is believed to be the allocation in the upcoming stage that gains effective power. All of the tour cities are updated as they change. The VM machines' function is to gather available resources. In our scenario, the final power can be studied for each VM. The number of initial populations that were set determines the tournament population. The Fit function is used to show how far the desired power has been attained.

To guarantee that there is no biasing in the data allocation throughout each tournament, the candidates were assigned to random tours. The tournament with the best chances of winning is chosen at random. The iteration ends when the power level stays constant, and the winner of the fittest tournament is taken into account for the final allocation of resources. The Configuration Manual additionally includes information on detailed implementation.

6 Evaluation

We implemented our method as part of the project's study and implementation, carried out experiments with hosts and virtual machines of various sizes, logged our test cases with workloads of various sizes, and plotted the graph. Our main attention is on the criteria stated below that we believe will assist us answer the research topic.

- **Number of Hosts:** A host is a piece of hardware that can provide a network's accessibility via a graphical interface, software applications, access point, embedded devices, or other ways.
- **Number of VM's:** A virtual machine (VM) is a virtual environment that operates as a virtual computer system with its own CPU, ram, hostname, and memory on a real hardware system (located off- or on-premises).
- **Simulation Time:** Virtual time is a positive real number that begins at zero. To clearly separate it from real (wall-clock) time, this virtual time is referred to as simulation time.
- **Energy Consumption:** Total energy used for and execution of task is referred to as energy consumption.
- **VM's Migration:** the migration of virtual servers (VMs) between resources, such as moving them from one data storage to another or moving them from one physical host to another.

6.1 Experiment 1: Dynamic Voltage & Frequency Scaling Vs Energy Efficient Data Center Resource Allocation

We are running **three iterations** of this experiment in order to calculate the average energy consumed to construct our method. This experiment was completed by starting a simulated environment and creating a task model for the data center device processing. Next, before assigning tasks, our algorithm is put to use to validate the specified parameters (such as the number of Hosts and VM's). The test's outcomes are displayed below (Figure 6). The results show that more energy is generally spent when doing tasks in DVFS Algorithm as the energy used when the simulator's default algorithm is used. A comparison of energy consumption was illustrated. The graph displays the suggested algorithm's efficiency level.

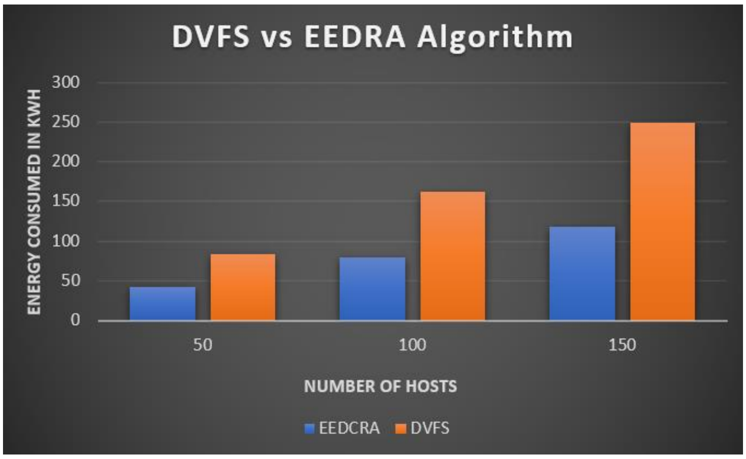

Figure 7: Energy Efficient Data Center Resource Allocation (EEDRA) vs DVFS

6.2 Experiment 2: Inter Quartile Range (IQR) Minimum Migration Time (MMT) vs Energy Efficient Data Center Resource Allocation.

In this trial, we contrast the time required to finish the task using the proposed method with the time required to do so using the simulator's standard technique. We found that our method outperforms and yields better results than the standard methods. As can be seen from the graph (Figure 7), our suggested technique results in a completion time decrease of over 31%.

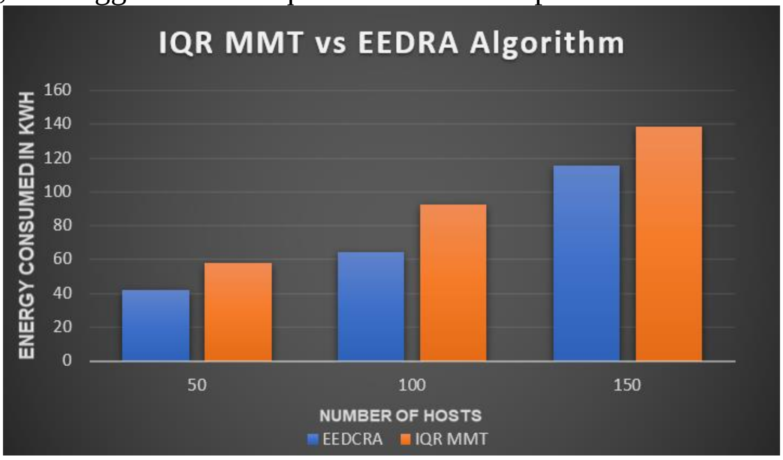

Figure 8: Energy Efficient Data Center Resource Allocation (EEDRA) vs IQR MMT

6.3 Discussion

A wide range of algorithms have been proposed by various writers in the subject of green cloud computing. These algorithms aid in reducing response times and achieving low latency, both of which are crucial for energy efficiency. We suggest a method that assesses a data center's varying energy capacity before load scheduling in order to decrease energy consumption, dynamic allocation of resources, and carbon footprints. In order to increase energy efficiency and consumption in data centers and lower carbon footprints, an unique energy-efficient datacenter allocation of resources algorithm has been employed in this work for green cloud computing. The suggested approach tries to reduce energy use while also aiding in load distribution.

After comparing two alternative pre-existing algorithms, we discovered that our algorithm is optimizing and using the energy necessary for our project as we examined the various test cases. Because we created this project using CloudSim using JAVA, we discovered that Generic Algorithms can be implemented using Python and CloudSim's libraries, therefore we can simply switch between the algorithms using Python.

7 Conclusion and Future Work

Since there is a growing need for cloud datacenters, it is essential to maximize resource utilization throughout the extensive life of the specific data center. Since doing so would raise the carbon footprint, it is not acceptable to abandon any of the pricy datacenter resources. We learnt from the lecture review that many of the current resource allocation techniques concentrate on just single resource at a point, either a hardware resource or a software resource. Hibernating the hardware and releasing the software from memory together will ultimately result in the datacenter having faster programming and more efficient hardware.

In terms of limitations, the method was regularly evaluated using three to four alternative algorithms and a simulation tool. This research project may be tested in a real-world setting. Future work for this project will include more refinements and a performance test of the suggested approach under various conditions. Based on past data and analysis, we may utilize deep learning to automatically select this priority designation of virtual machines in future work. In addition, calculations of energy consumption demands can aid in making decisions and lead to a more system which combines for energy efficiency and consumption in datacentres.

References

Akhter, N., & Othman, M. (2016). Energy aware resource allocation of cloud data center: review and open issues. *Cluster computing*, *19*(3), 1163-1182.

Balanici, M., & Pachnicke, S. (2019, March). Machine learning-based traffic prediction for optical switching resource allocation in hybrid intra-data center networks. In *2019 Optical Fiber Communications Conference and Exhibition (OFC)* (pp. 1-3). IEEE.

Luo, Y., Cao, X., Chen, J., Gu, J., Yu, H., Sun, J., & Zou, J. (2022). Platelet-Derived Growth Factor-Functionalized Scaffolds for the Recruitment of Synovial Mesenchymal Stem Cells for Osteochondral Repair. *Stem Cells International*, *2022*.

Ekici, N. U., Schmidt, K. W., Yazar, A., & Schmidt, E. G. (2019, July). Resource allocation for minimized power consumption in hardware accelerated clouds. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-8). IEEE.

Huang, Y., Xu, H., Gao, H., Ma, X., & Hussain, W. (2021). SSUR: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center. *IEEE Transactions on Green Communications and Networking*, *5*(2), 670-681.

Jeba, J. A., Roy, S., Rashid, M. O., Atik, S. T., & Whaiduzzaman, M. (2021). Towards green cloud computing an algorithmic approach for energy minimization in cloud data centers. In *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing* (pp. 846-872). IGI Global.

Ouhame, S., Hadi, Y., & Ullah, A. (2021). An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. *Neural Computing and Applications*, *33*(16), 10043-10055.

Singha, A., Sarkar, S. J., Nayak, S., & Patgiri, R. (2022, January). Green Cloud Computing-To Build A Sustainable Tomorrow. In *2022 International Conference for Advancement in Technology (ICONAT)* (pp. 1-6). IEEE.

Sharma, R., Bala, A., & Singh, A. (2022, April). Virtual Machine Migration for Green Cloud Computing. In *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* (pp. 1-7). IEEE.

Shukur, H., Zeebaree, S., Zebari, R., Zeebaree, D., Ahmed, O., & Salih, A. (2020). Cloud computing virtualization of resources allocation for distributed systems. *Journal of Applied Science and Technology Trends*, *1*(3), 98-105.

Thein, T., Myo, M. M., Parvin, S., & Gawanmeh, A. (2020). Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. Journal of King Saud University – Computer and Information Sciences, 32(10), 1127–1139.

Wang, C. C., Lin, Y. D., Wu, J. J., Lin, P. C., & Hwang, R. H. (2018). Toward optimal resource allocation of virtualized network functions for hierarchical datacenters. *IEEE Transactions on Network and Service Management*, *15*(4), 1532-1544.

Yuan, H., Furdek, M., Muhammad, A., Saljoghei, A., Wosinska, L., & Zervas, G. (2018). Space-division multiplexing in data center networks: on multi-core fiber solutions and crosstalk-suppressed resource allocation. *Journal of optical communications and networking*, *10*(4), 272-288.

Yi, B., Wang, X., Huang, M., & Zhao, Y. (2020). Novel resource allocation mechanism for SDN-based data center networks. *Journal of Network and Computer Applications*, *155*, 102554.

Zhao, C., Wang, T., & Yang, A. (2019). A heterogeneous virtual machines resource allocation scheme in slices architecture of 5G edge datacenter. *Comput. Mater. Contin*, *61*, 423-437.