https://dx.doi.org/10.4314/ijs.v25i3.8

Ife Journal of Science vol. 25, no. 3 (2023)

AN EFFICIENT REDUCING MECHANISM FOR ENERGY CONSUMPTION IN DATA CENTER USING HYBRID CONSOLIDATION TECHNIQUES

Oyekanmi, E.O.[1,*] and Adegoke, O.M.[2]

[1]Department of Computer Science, Achievers University, Owo, Nigeria.

[2]Department of Computer Science, Joseph Ayo Babalola University, Ikeji-Arakeji, Nigeria.

[*]Corresponding Author's Email: e.oyekanmi@achievers.edu.ng

(Received: 19th may, 2023; Accepted: 24th October, 2023)

ABSTRACT

The rise in internet user demand is a major factor in the expansion of infrastructure and the upsurge in energy use in cloud, colocation, and some business data centres. The advent of 5G has compounded the situation, as it substantially gives room for many new types of digital services, resulting in a need for richer media formats and higher resolution content which consume a lot of energy when no scaling method is applied. Services such as email, data storage and retrieval and other cloud services also require a lot of high energy consumption which eventually result into carbon(IV) oxide (CO_2) emissions to the environment. This research therefore, focuses on lowering the energy usage of a data centre with heterogeneous power awareness either in an idle server state or high-performance state using a novel hybridized algorithm called "DyVoFesLoReMu", comprising Dynamic Voltage Frequency Scaling (Dvfs) and a modified Local Regression Minimum Utilization (LrMu). A real dataset (workload) obtained online from PlanetLab consisting of hosts and Virtual Machines (VM) was simulated on a data center in CloudSim 3.0.3. Tool kit with preset parameters consisting of VM Allocation Policy and VM Selection Policy was used. The tool kit was utilised to create cloud infrastructure and simulate the essential features of a cloud environment. The Cloudsim was installed on Eclipse Integrated Development Environment (IDE) 2019 version on Windows 10 operating system. The hybridized algorithm was compared with other five (5) existing energy reducing algorithms and it was found to be more efficient with a range of 41-90% reduction in energy usage from the ten days workload traces and in comparison with the existing algorithms used for the simulation.

Keywords: Energy Consumption, High performance, Data center, Cloud.

INTRODUCTION

There had been a tremendous increase in the usage of internet resources during the pandemic era (COVID-19). Numerous meetings and services are carried out using Whatsapp, Zoom, Facebook, and other social media platforms. Business, trainings, and conferences are handled virtually. All these social media activities are controlled and coordinated from a data center in a cloud environment, thus increasing technology drive and creating new business opportunities revolution for the sustenance of economic development and new technology drive. Consequently, revolution in information technology (IT) and advancement in data storage and retrieval are having a major negative impact on the data center when the cloud is loaded with more tasks on data storage and retrieval, email services, and other cloud services, thus leading to high energy consumption and CO_2 emission. Recent research has shown that energy consumption will likely increase in the future due to growing equipment stocks to support the Internet infrastructure and use of modern IT equipment (Spengler and Wilmsmeier, 2019). The main reason for this extreme high energy consumption is not just the use of computing resources in large-scale and the energy inefficiency of hardware, but rather lies in the inefficient usage of these resources (Yadav *et al.*, 2019).

World widely, datacenters are estimated to spend $27 billion annually on energy (Jay and Chuck, 2021),and to meet up with this increasing demand of cloud services, new datacenters may need to be constructed, else the capabilities or existing datacenters may be expanded. However, expanding datacenter capability is not always feasible due to economical and physical constraints such as fixed capacity of power generating facilities and limited achievable server density (Leverich, 2014). Building new data centres may be extremely expensive, greatly raising the total cost of ownership in terms of both the capital expense needed to build a new facility and the operational expense needed to manage the datacentre (Leverich, 2014).

Investigating and locating inefficiencies in order to enhance the capabilities of current facilities is an alternative to constructing new data centres or expanding existing ones.

Data growth trends will continue to increase as the world consumes more and more data especially now that pandemic is ravaging the world and there was a paradigm shift to the usage of cloud services. Survey conducted by Bashroush (2020) suggests that users of the internet, particularly those who use social media and cryptocurrencies run by datacenters, require large-scale data centers that need enormous amount of electric power to provide good quality of service (Yadav and Garg, 2019). According to US figures, data centres quadruple every five years and use around 3% of the world's electrical generation (Rallo, 2014; Shaw, 2007).

One of the ways to ensure effective cloud computing technology is to avoid wastage or loss of data stored in the cloud. This can only be done when power is conserved. Hence, a novel technique is presented to continue in the conservation of energy usage on datacenter as a way of reducing energy consumption by its IT infrastructure.

Energy conservation methods have been employed in data centres. These methods provide servers with fundamental energy management. The methods include putting servers in sleep mode, turning them on and off, using dynamic voltage/frequency scaling (DVFS) to change the power states of servers, and putting virtual migration strategy into practise. Despite all the methods for conserving energy in data centres, the intricate operations of cloud data centres necessitate an upgrade to a high efficiency management system so that servers can operate at their peak efficiency. The energized processing power of data centers during the distribution, processing and storage of data also necessitate the need for the establishment of an effective, efficient, reliable way that can withstand or prevent any loss, wastage or reduction in energy consumption during the idle or busy state of servers. Leveraging on the earlier discussed issues, in this research, a hybridized scheme comprising Dvfs and LrMu algorithms was designed and simulated to aid energy reduction in datacenter.

In order to increase energy efficiency and lower energy consumption of the facilities used in data centres, this research developed a novel strategy called *DyVoFesLoReMu*.

RELATED WORK

Many authors have contributed to knowledge in the area of reducing energy consumption and their works span through placement of servers and adjustment of power state of servers. It was suggested in (Abdelsamea *et al.*, 2017) to improve virtual machine consolidation using hybrid regression algorithms. This technique, called a Hybrid Local Regression Host Overload Detection algorithm (HLRHOD), was created to reduce energy usage while assuring a high level of adherence to Service Level Agreements (SLA). Instead of using just one utilisation parameter (CPU usage), three were used: CPU, Memory, and Bandwidth. VM behaviour is rarely a function of one variable; rather, it should be a function of several factors since the results from the combined factors are better than the results from the single-factor methods. The outcome was tested on a single PlanetLab workload with 800 hosts and 1033 virtual machines, however the percentage reduction was between 20 and 24. This study has motivated us to develop a hybrid method that can improve energy efficiency in the cloud.

Zahedi *et al.* (2017) accomplished energy reduction, Quality of Service and temperature balancing in the cloud data centre using a unique virtual machine consolidation technique. The authors used CloudSim tool to simulate the suggested technique. The outcome of the simulation certified that physical machine temperature, SLA, and migration technique can together control the energy consumption and QoS in a cloud data center.

As an alternative to Dynamic Voltage Frequency Scaling and VM consolidation, a new algorithm dubbed Brownout was also proposed by Buyya and Gill (2018). The combined brownout approach reduces energy consumption by selectively and dynamically deactivating application optional components, and it can also be used with self-contained microservices. The findings demonstrate that their method can reduce energy use by more than 20%, albeit there are trade-offs between energy savings and user discounts. It was discovered to be better than DVFS at preventing overloading. In comparison to the outcome of our established hybrid approach, the reduction in percentage is still modest.

Ahmad et al. (2018) proposed a solution for the energy saving problem by enabling dynamic voltage and frequency scaling technique for gaming data centers. The dynamic voltage and frequency scaling technique was compared against non-power aware and static threshold detection techniques. This helps service providers to meet the quality of service and quality of experience constraints by meeting service level agreements. Game traces were used as a workload for testing the technique. Selection of better techniques helped gaming servers to save energy cost and maintain a better quality of service for users placed globally. This work provides an opportunity to investigate which technique behaves better — dynamic, static or non-power aware. The results demonstrate that less energy is consumed by implementing a dynamic voltage and frequency approach in comparison with static threshold consolidation or non-power aware technique. However, the work failed to carried out enhancement for energy saving technique in Big Data (such as the one used in the experimentation of our study) and detailed analysis were not provided.

LrMu algorithm was analysed and compared with Non Power aware, Dvfs, ThRs, IqrMc,Madmmt algorithms (Nagpal *et al.*, 2018). But, the work was carried out using a random workload and less VM and host.

Yavari *et al.* (2019) also improve the consolidation problem of VMs by combining two novel techniques which are heuristic energy and temperature aware based virtual machine consolidation (HET-VC) and Firefly energy and temperature aware based virtual machine consolidation (FET-VC).

Zhou *et al.* (2021) addresses the issue of load variation during VM allocation and placement, an energy-efficient VM allocation and deployment technique based on an adaptive energy-aware architecture for internet of thing (IOT) applications was given and the experimental analysis was carried out using PlanetLab virtual machines running a real-world workload with various safety parameter ranges between 0.5 and 3.0.

A dynamic VM selection algorithm known as Minimum Size Utilisation (MSU) was devised by Yadav et al (2021) to choose the VMs from an overloaded host for VM consolidation. The goal of this study was to improve cloud data centres' energy efficiency. In comparison to the other baseline schemes, the suggested algorithm was able to reduce energy usage and SLA violation by an average of 23% and 27.5%, respectively.

The performance of energy consumption reduction using our suggested strategy was found to be more effective than the state-of the-art based on our findings.

SYSTEM MODEL ARCHITECTURE

One way to reduce energy consumption is by way of dynamically readjusting the VM. With this approach of consolidation, virtual machines are placed on all host as a form of initialization based on the requirement of the host. This is referred to as default placement of VMs. However, as the needs of the VMs evolve over time, readjustment (consolidation) of the VM happens dynamically. As a result of this dynamic shift, periodic provisioning of the underlying resources is required to adequately support the VMs while minimising energy consumption and Service Level Agreement (SLA) violations. A complementary approach to VM consolidation using DyVoFesLoReMU while finding suitable host for a VM is designed as shown in Figure 1. During checking of Overloaded Host, a Mean Size Utilization Scheme was also implemented towards reducing the energy consumption in a datacenter.

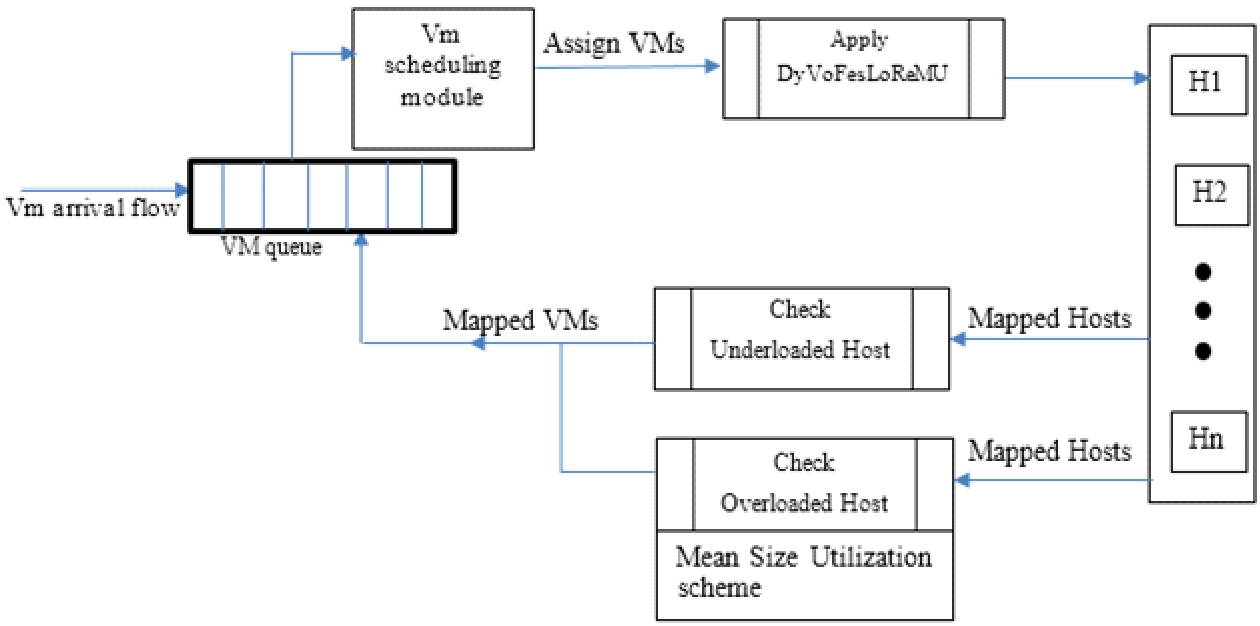

Figure 1: DyVoFesLoReMU Architecture.

There are 5 components to the design depicted in Figure 1. They are discussed as follows.

Initial VM Placement

The first step is to set up the virtual machines (VMs) on the hosts (physical machines), taking into account the resources that each machine needs. However, while the machines are operating, the demand could alter, necessitating the use of overloading or underloading detection mechanisms to condense them.

Detecting Overloaded Hosts

The hosts are all examined by the overload detection algorithm for overload. The VMs must be moved away from the hosts if any of them are overcrowded in order to lessen the CPU burden and avoid SLA violations.

Detecting Underloaded Hosts

The underload detection method evaluates each host for underload so that all of the VMs can be moved to other hosts and the hosts can be switched to a power-saving state.

Selecting the VMs for Migration from the Hosts

The combined migration map for the overloaded and underloaded hosts is returned by the VM selection algorithm and shows where to put the selected VMs for migration.

VM Placement

The VM placement is then completed in accordance with the migration map. The virtualization technology allows one to build many virtual machines (VM_1, VM_2,..., VM_n) on a real machine, hence reducing the amount of hardware required and increasing resource utilisation. This technology can be used to access the resources of cloud infrastructure.

Many algorithms had been used in power consumption reduction in order to consolidate the virtual machine by cloud computing data center as discussed in the related works. In this research, Median Absolute Deviation (MAD) algorithm was used as the host overload detection algorithm whose VM selection algorithm is Minimum Utilization. This scheme was used purposely to solve the problem of outliers. Some of the benchmark algorithms in the same family in CloudSim environment were compared and evaluated in this research. Energy consumption value, SLATAH (SLA Violation Time per Active Host), number of hosts shut down, and number of migrations are among the comparison metrics used.

Workload Characterization

Simulation and evaluation employed workload traces from a real system. Workload information was imported from the CoMon project's PlanetLab component (Tang *et al.*, 2007). In a random 10-day period from March to April 2011, the workload traces were gathered. You may find more information about these traces at https://github.com/beloglazov/planetlab-workload-traces. According to workload traces, the CPU utilization used during data collection was less than 50%, and during the simulation, the VM assignment had been random. The characteristics of the workload are shown in Table 1. More than 500 servers around the world's locations were used to gather data on CPU usage. The task involved thousands of VMs, and measurements of CPU utilisation showed that it took 5 minutes (Beloglazov and Buyya, 2012).

Table 1: CPU utilization of workload data (Beloglazov and Buyya, 2012).

Date	Number of VMs	Number of Host	Mean (SD) %	Quartile1 (%)	Median (%)	Quartile3 (%)
03/03/2011	1052	800	12.31(17.09)	2	6	15
06/03/2011	898	800	11.44(16.83)	2	5	13
09/03/2011	1061	800	10.70(15.57)	2	4	13
22/03/2011	1516	800	9.26(12.78)	2	5	12
25/03/2011	1078	800	10.56(14.14)	2	6	14
03/04/2011	1463	800	12.39(16.55)	2	6	17
09/04/2011	1358	800	11.12(15.09)	2	6	15
11/04/2011	1233	800	11.56(15.07)	2	6	16
12/04/2011	1054	800	11.54(15.15)	2	6	16
20/04/2011	1033	800	10.43(15.21)	2	4	12

RESEARCH MIGRATION POLICY SCHEME

In statistics, outliers are known to be extreme values in a data distribution which are likely to be erroneous. Finding a method to recognise and remove such data from the distribution is one technique to address this extreme situation. Since the number of host is eight hundred (800) while number of VM is varied with a range of 898 and 1516, there is possibility of having extremely high or extremely low values and non-normality during simulation. Median Absolute Deviation (MAD) is used because of its various advantages. It enhances robust measure for data which are not normally distributed, serves as an outlier-tuner technique which uses standard deviation from the mean and sometimes its uses deviation from the median, which is less susceptible to distortion caused by outlying values. In this research therefore,

$$\forall\, U_i |\; i\; \in I$$

Where I is a set of utilization history and U_i is the ith utilization value of VM measured in MIPS, then

$$\text{MAD} = \text{median}\,(|\,\text{median}(U_i) - U_i\,|) \qquad (1)$$

where $0 \leq I < |\text{utilization history}|$

MAD is the returned utilization mean in MIPS as shown in equation 1.

In CloudSim 3.0.3, a standard median deviation VM power policy allocation file named "PowerVmAllocationPolicyMigrationMedianAbsoluteDeviation" was used for migration of virtual machines during the simulation. However, in the overloading module of the file, a new scheme was implemented which is discussed in the next section.

VM Migration Scheme for Overloaded Host

One of the main causes of an increase in energy consumption is thought to be an overworked host. In order to solve this problem, the cost of migration and service level agreement violation are needed to be reduced by using a novel scheme called Mean Size Utilization scheme (MSU). This method was used to choose the VMs when the hosts were found to be overloaded, as stated in equation 2. The scheme reduces the idleness of host, maximize its capacity utilization and prevent it from being overloaded. The algorithm for MSU scheme is shown in Algorithm 1

$$MSU = \frac{\sum_{i=1}^{n} U(H_i) + 1}{|H| + 1} \qquad (2)$$

where $U(H_i)$ is the cpu utilization of host i and $|H|$ is the total number of all the host in the datacenter. One (1) was added to both the numerator and denominator to prevent null value and division error especially when the length of host is zero. Although this kind of scenario is actually not common.

Algorithm for MSU Migration Scheme

1: Input: vmTomigrateList
2: Output: Selected VM
3: vmTotalUtilization $\leftarrow$ 0
3: For each VM in vmTomigrateList do
4: vmTotalUtilization $\leftarrow$ vmTotalUtilization + vmUtilization(VM)
5: EndFor

6: $vm\_utilizationmetric \leftarrow \frac{\mathbf{vmTotalUtilization} + 1}{\mathbf{vmTomigrateList.size}\,() + 1}$

7: For each VM in vmTomigrateList do

8: *IF vm_utilizationmetric < vmUtilization (VM) then*
9: *vmToMigrate.add (VM)*
10: EndIf
11: EndFor
12: return *vmToMigrate*

DyVoFesLoReMU Algorithm

This algorithm combined VM migration scheme with host suitability. While finding suitable host, the utilization of the energy consumption of the host is calculated at the end of each time frame before starting another simulation. **DyVoFesLoReMU**, which is a combination of Dynamic Voltage/Frequency Scaling (DyVoFes) and Local Regression VM allocation policy with Mean Utilization VM selection policy (LoReMU) was implemented, such that at each time frame the utilization value equals 0%, Dynamic Voltage Frequency Scaling power aware policy is invoked and any host of such value is shutdown (turned off) to save the energy. LoRe VM allocation policy determines whether a host is overcrowded or not by using Local Regression (LR) to forecast host utilisation (load). The VM with the Mean (Average) CPU Utilisation (MU) is the one that is chosen for migration under the MU VM Selection Policy. If the required MIPS at the end of each time frame computation is higher than the total MIPS allotted to the host, the host is regarded as being under allocated for the VM. If a virtual machine's utilisation is average, migration is considered. MSU has been discussed in Algorithm 1 and implemented for VM migration in LoReMU policy file. The algorithm is shown in Algorithm 2 as follows:

DyVoFesLoReMU procedure

1: Input: HostList, VM
2: Output: Select Suitable Host
3: FOR each host in HostList DO
4: IF (getUtilizationOfCpuMips(host)!=0)
5: //Begin LrMu implementation
6: IF((getUtilizationOfCpuMips(host)> vmUtilization(*VM*)) andand (getUtilizationOfCpuMips(host)<=*vm_utilizationmetric*))
7: allocate (host, VM)
8: suitableHost.add(host)
9: ELSE
10: IF (getUtilizationOfCpuMips(host)< vmUtilization(*VM*))
11: Switchoff(host) // to conserve power and reduce energy consumption
12: ELSE IF(getUtilizationOfCpuMips(host) > *vm_utilizationmetric*))
13: FOR each vm in host DO
14: *vmToMigrate.add (vm)*
15: IF(getUtilizationOfCpuMips(host)< *vm_utilizationmetric*)
16: Break;
17: ENDIF
18: ENDFOR
19: ENDIF
20: ENDIF
21: //End of LrMu implementation
22: ELSE
23: Switchoff (host)//Applying DvFS
24: ENDIF
25: ENDFOR
26: RETURN *suitableHost*

Hardware and Software Configuration

The simulation for this study was conducted on a server with a 64-bit Core i3 operating system (OS) and four logical processors, which divide a server's computing capacity into separate halves to allow for parallel processing. Each logical processor has the capacity to carry out its own stream of instructions concurrently, and the OS can assign concurrent independent units of work to each logical processor as indicated in Table 2. As a framework for modelling and simulating cloud computing infrastructures and services with limited computing system capability, CloudSim is one of the software requirements. It may replicate cloud-based systems on any computer system with a dual-core processor, 2 GB RAM, and 1 GB storage using a Java Runtime Environment (JRE). The CloudSim toolkit 3.0.3 was installed and set up for efficient operation using Java Eclipse as the development environment, as indicated in Table 3. The Apache Commons Math library, which addresses the most prevalent practical issues not immediately addressed in the Java programming language or commons-lang, was imported into the Java Eclipse. For example, regression calculations performed in Local Regression Mean Utilisation are addressed by this library of lightweight, self-contained mathematics and statistics components. The Java Runtime Environment (JRE) was able to execute the CloudSim source since once it has been converted by the Java Development Kit (JDK).

Table 2: The configuration of hardware used for the simulation.

Operating System	Processor	Installed RAM	Hard Disk
Windows (10) 64-bit operating system	Intel (R) Core(TM) i3 Processor	4.00 GB	320GB

Table 3. The configuration of software used for the simulation.

CloudSim	Eclipse	Java Version	Apache
Cloud Sim Tool kit 3.0.3	Java Eclipse IDE (2019)	Java Development Kit (JDK 1.8)	Apache commons - maths3-3.6.1

RESULTS AND DISCUSSION

The simulation was done in two ways. The first aspect of the results was based on the efficiency of the proposed method on ten days workload traces from PlanetLab without adjusting the safety parameter. The second result was focused on the workload trace of 03/03/2011 labelled as "20110303" file. The safety parameter on this file was adjusted and the result of the proposed method was compared with other method in the same policy selection category.

Simulation result without adjusting safety parameter

The hybrid method was simulated within a total simulation in general based on the workload traces from PlanetLab for ten days. Table 4 shows energy reduction value in kilowatt per hour for both the proposed method and compared methods for ten days workload traces from PlanetLab. Figure 2 shows the bar chart of the comparison with DyVoFesLoReMU having lower energy consumption value compare to the other five methods.

Table 4: Energy Consumption result of cloud methods and **DyVoFesLoReMU**

Workload	Energy Consumption (KWh)					
	DyVoFesLoReMU	Dvfs	lrmu	thr	mad	iqrm
20110303	102.79	803.91	174.24	206.73	275.84	204.22
20110306	76.4	623.77	132.33	157.41	151.49	153.02
20110309	83.18	708.68	151.74	182.62	177.7	180.17
20110322	104.34	1014.21	188.31	220.64	214.02	219.49
20110325	89.09	785.49	162.39	189.41	183.73	188.44
20110403	133.84	1071.9	240.02	279.8	271.67	275.84
20110409	112.48	928.59	190.87	223.5	217.06	222.36
20110411	108.54	903.08	187.12	221.31	212.9	218.16
20110412	89.98	766.75	163.6	193.24	186.8	191.27
20110420	77.06	688.63	139.98	166.19	164.81	170.5

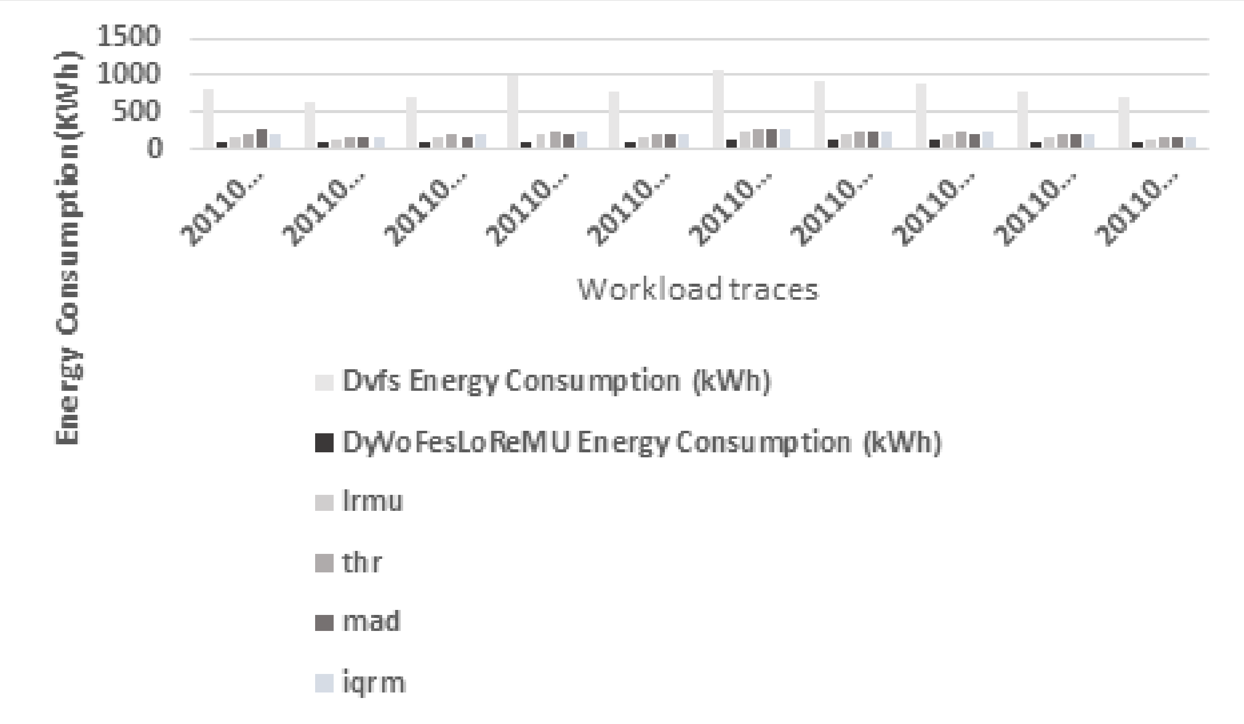

Figure 2: Energy Consumption for ten days workload traces.

Table 5 shows the number of VM migration that occurs at every instance of workload. Zero (no migration) was recorded for Dvfs, because it does not involve any virtual migration scheme. Figure 3 displays the workload trace graphed against the number of migrated virtual machines during the experiment.

Table 5: Simulation result of Number of Virtual machine Migration.

Workload	NUMBER OF VM MIGRATION					
	DyVoFesLoReMU	dvfs	lrmu	thrmu	madmu	iqrmu
20110303	2458	0	29555	30188	30051	29901
20110306	2166	0	21915	23729	23566	23256
20110309	2657	0	25876	27194	26536	27177
20110322	3423	0	32710	33552	32035	33084
20110325	2536	0	27426	27939	27485	27754
20110403	3180	0	40277	40430	39815	39910
20110409	3041	0	32546	33047	32292	32502
20110411	2925	0	31007	32703	31336	32064
20110412	2528	0	27882	28813	27150	28026
20110420	2483	0	24998	26250	25890	26511

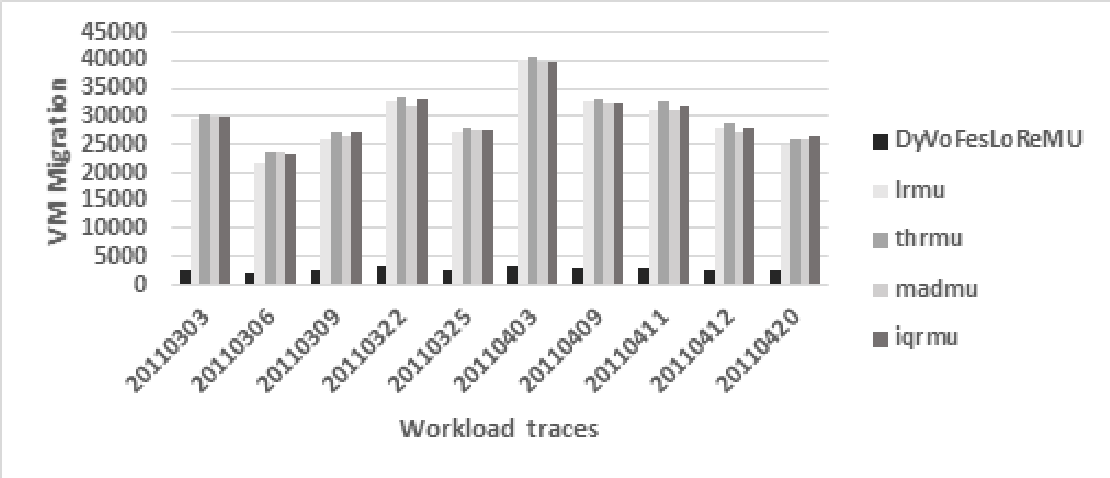

Figure 3: Graph of VM migration against workload traces.

Table 6 displays the number of hosts that were shut down during the ten-day workload traces for the proposed approach and the compared methods. Figure 4 shows the graphical representation of the workload trace against the number of host shut down during the simulation.

Table 6: Simulation result of number of host shutdown.

Workload	Number of Host Shutdown					
	DyVoFesLoReMU	dvfs	lrmu	thrmu	madmu	iqrmu
20110303	774	457	5525	6474	6353	6420
20110306	781	545	4305	5146	5062	5017
20110309	786	582	4899	5827	5640	5786
20110322	774	385	5744	6878	6592	6775
20110325	775	467	5090	5928	5847	5978
20110403	762	358	7237	8360	8186	8260
20110409	771	412	5892	6897	6675	6814
20110411	772	419	5728	6836	6603	6685
20110412	778	476	476	6125	5908	6079
20110420	781	519	4670	5482	5442	5539

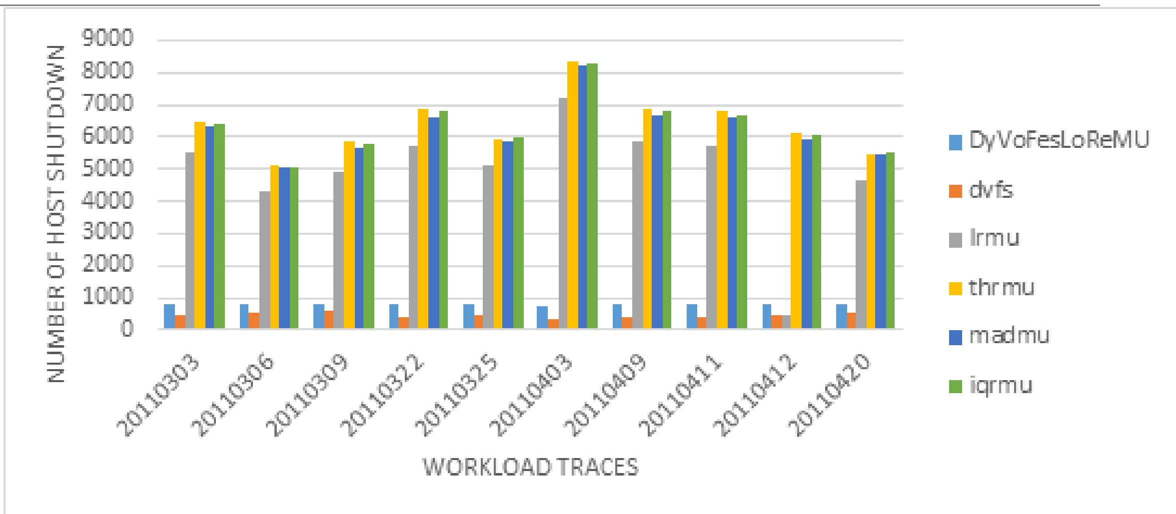

Figure 4: Graph of the number of host shut down against workload trace.

Table 7 shows the execution total mean time for both the proposed method for ten days workload traces from PlanetLab. The graph of the total mean time plotted against the number of VMS is shown in Figure 5. The equation from the graph shows that the performance of the method used is 82% efficient.

Table 7: Simulation result of total mean time using **DyVoFesLoReMU**.

Workload	VM	Execution total mean time(SD) (sec)
20110303	1052	0.14975 (0.404)
20110306	898	0.10039 (0.311)
20110309	1061	0.11941 (0.352)
20110322	1516	0.17758 (0.418)
20110325	1078	0.11207 (0.388)
20110403	1463	0.20926 (0.393)
20110409	1358	0.18378 (0.397)
20110411	1233	0.17152(0.348)
20110412	1054	0.12463 (0.332)
20110420	1033	0.11585 (0.296)

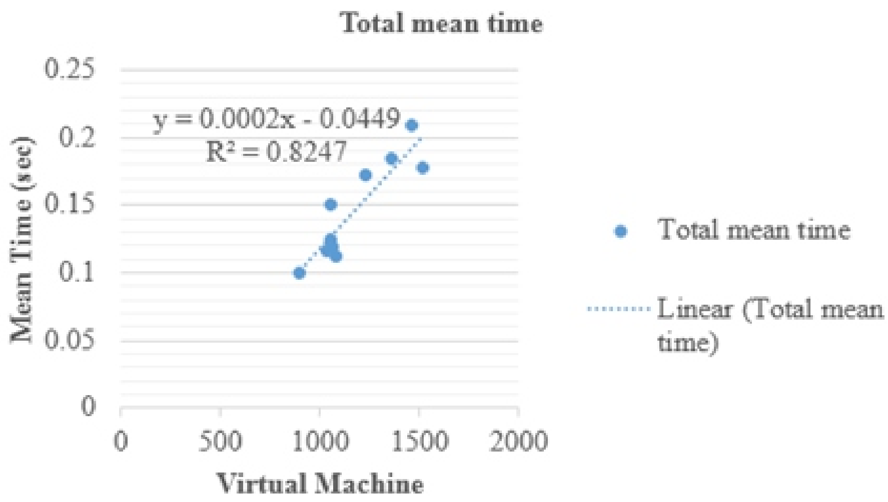

Figure 5: Graph of total mean time against number of VMS.

Simulation Result with Adjustment of Safety Parameter

In CloudSim 3.0.3, the standard policy used for VM allocation is median deviation VM power policy which is implemented in a file labelled as "PowerVmAllocationPolicyMigrationMedianAbsoluteDeviation". This file uses a VM selection policy to choose the VM with the lowest CPU Utilisation for migration. This is checked especially for overloaded host. However, we modified this file and a mean utilization was used following the MSU scheme earlier discussed in section III. The MSU scheme was induced on all the methods in Cloudsim 3.0.3 that uses median absolute deviation scheme with minimum utilization and the safety parameter of the methods were also adjusted. This section's results demonstrate the high effectiveness of the suggested induced MSU on all the approaches. The performance was measured by observing the percentage of reduction on average using the report of March 3, 2011 from PlanetLab which is labelled as “20110303”.

The energy consumption in kilowatt per hour is shown on Table 8 for four different methods that uses CPU minimum utilization for overloaded host. The normal value when minimum utilization was used for overloaded host was higher compared with when the scheme was changed to mean utilization scheme. In this result, safety parameter was not adjusted, and was allowed to run only under the adjusted scheme for CPU utilization. The result shows that with LRMU has better percentage reduction report compared with other methods with the same scheme.

Table 8: Energy consumption under CPU mean (induced) and minimum utilization.

	LRMU_1.2 (kw/h)	THRMU_0.8 (kw/h)	MAD_2.5 (kw/h)	IQRMU_1.5 (kw/h)
With MSU Induced	102.79	102.41	100.12	100.7
normal	174.24	206.73	275.84	204.22
Reduction rate (%)	**59**	**50**	**36**	**49**

The safety parameter was adjusted and simulated under the proposed CPU utilization scheme induced in all the related methods. Another simulation was run without the proposed scheme but yet with the variant safety parameter. The output was recorded on average scale as shown on Tables 9-12 and the graphs are represented on Figures 6-9.

Table 9: Energy consumption reduction under Threshold Random selection method

Method / Safety Coefficient	ThrMu	Induced MSU POLICY
0.5	226.02	133.21
1	190.04	88.49
1.5	95.28	63.56
2	93.33	52.8
2.5	91.57	43.91
3	91.57	36.78
AVERAGE	**131.30**	**69.79**
% Average Reduction	**46.84**	

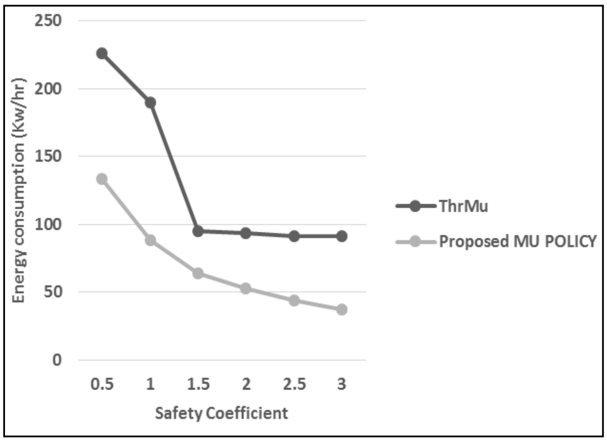

Figure 6: Proposed MU policy on ThrMU.

Table 10: Energy consumption reduction under Median Absolute Deviation selection method

Method / Safety Coefficient	MadMu	Induced MSU POLICY
0.5	193.1	91.55
1	195.98	91.9
1.5	196.88	94.44
2	196.41	97.59
2.5	200.4	100.12
3	201.6	102.24
AVERAGE	**197.40**	**96.31**
% Average Reduction	**51.21**	

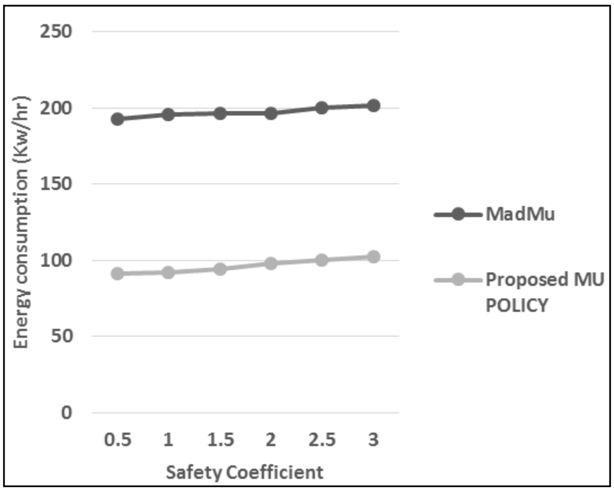

Figure 7: Proposed MU policy on MadMu.

Table 11: Energy consumption reduction under Inter Quartile Range selection method

Method / Safety Coefficient	IqrMu	Induced MSU POLICY
0.5	194.29	92.1
1	198.46	98.14
1.5	204.22	100.7
2	210.16	105.92
2.5	214.46	108.97
3	222.78	113.34
AVERAGE	**207.40**	**103.20**
% Average Reduction	**50.24**	

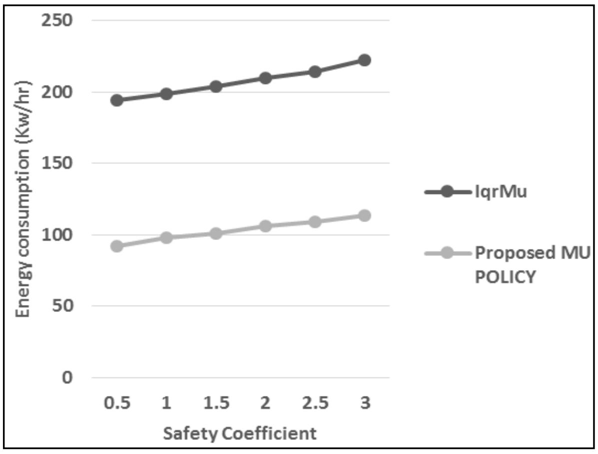

Figure 8: Proposed MU policy on IqrMU.

Table 12: Energy consumption reduction under local regression selection method

Method / Safety Coefficient	LrMu	Induced MSU POLICY
0.5	92.96	54.49
1	142.73	91.21
1.5	199.99	117.65
2	236.23	131.13
2.5	268.77	133.61
3	304.35	133.81
AVERAGE	**207.51**	**110.32**
% Average Reduction	**46.84**	

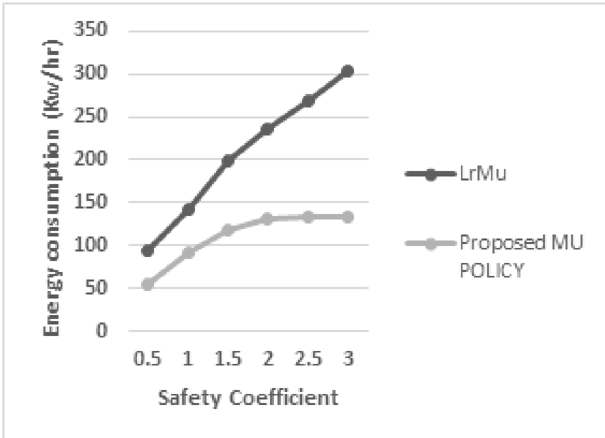

Figure 9: Proposed MU policy on LrMU.

The report shown on Tables 9-12 reflect that the implementation of mean size utilization of the CPU on Median Absolute Deviation selection method, has the highest reduction value on average, followed by Inter Quartile Range selection method with 51.21% and 50.24%, respectively. Both Threshold Random and Local Regression selection methods have same value of 46.84% reduction on average. The report of the SLA violation is shown in Table 13 and Figure 10.

Table 13: SLA Violation at default safety parameter.

	LRMU_1.2 (%)	**THRMU_0.8 (%)**	**MAD_2.5 (%)**	**IQRMU_1.5 (%)**
With MSU Induced	11.57	13.16	12.6	12.69
normal	9.71	10.09	10.28	10.17

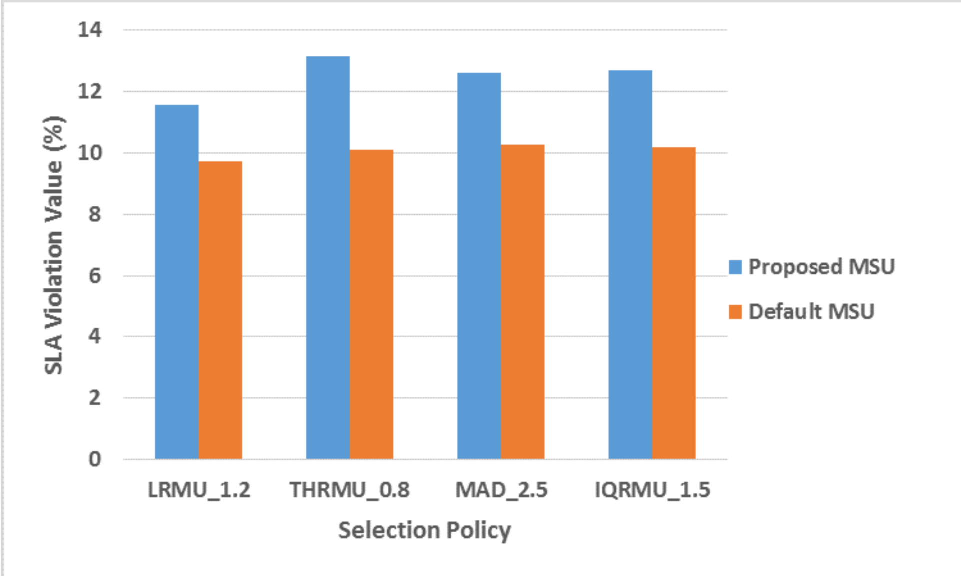

Figure 10: SLA Violation of Proposed MSU.

CONCLUSION

This paper has shown improvement in energy consumption reduction by applying hybrid of DvFs and LrMU methods called DyVoFesLoReMU in power Cloudsim. The results show that under default safety parameter, the proposed method achieved a range of 41-90% reduction in energy usage from the ten (10) days workload traces and in comparison with the existing algorithms. Also, it was established that the proposed mean size utilization CPU for overloaded host performed better than the default minimum size CPU utilization under varying safety parameter. Future work would examine improving on the reduction of SLA violation value which is also a metric for high performance in Cloudsim computation.

CONFLICT OF INTEREST:

Authors declare no conflicts of interest on this manuscript

AUTHOR'S CONTRIBUTION

O.E.O: Conceptualization, Methodology, Software. **O.E.O**: Data curation, Writing- Original draft preparation. **A.O.M**: Visualization, Investigation. **O.E.O**: Supervision. **O.E.O**: Software, Validation. **A.O.M**: Writing- Reviewing and Editing

REFERENCES

Abdelsamea, A., El-Moursy, A. A., Hemayed, E. E., and Eldeeb, H., 2017. Virtual machine consolidation enhancement using hybrid regression algorithms. Egyptian Informatics Journal, 18(3): 161-170. doi: 10.1016/j.eij.2016.12.002

Ahmad, B., McClean, S., Charles, D., and Parr, G., 2018. Analysis of energy saving technique in CloudSim using gaming workload. Cloud Computing, 2018, 143.

Bashroush, R., 2020. Data center energy use goes up and up and up. Uptime Inst.

Beloglazov, A., and Buyya, R., 2012. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience, 24(13): 1397-1420. doi: 10.1002/cpe.1867

Buyya, R., and Gill, S. S., 2018. Sustainable cloud computing: foundations and future directions. arXiv preprint arXiv:1805.01765. doi.org/10.48550/arXiv.1805.01765

Jay, K. and Chuck, R., 2021. Improving data center efficiency through Intel technologies and high ambient temperature operation. Retrieved September 2021, from https://www.intel.com/content/dam/doc/technology-brief/efficient-datacenter-high-ambient-temperature-operation-brief.pdf, 2021.

Leverich, J. B., 2014. Future scaling of datacenter power-efficiency. Stanford University.

Nagpal, S., Neeraj, D. and Surjeet D., 2018. Analysis of LrMu Power Algorithm in the Cloud Computing Environment using CloudSim Toolkit. IJRECE, 6(3): 1175-1177.

Rallo, A., 2014. Industry outlook: Data center energy efficiency. Retrieved September, 24, 2018.

Shaw, D., 2005. EPA's 2007 report on the environment: Science report (SAB review draft).

Spengler, T., and Wilmsmeier, G., 2019. Sustainable performance and benchmarking in container terminals—The energy dimension. In Green ports (pp. 125-154). doi: 10.1016/B978-0-12-814054-3.00007-4

Tang, L., Chen, Y., Li, F., Zhang, H., and Li, J., 2007. Empirical study on the evolution of planetlab. In Sixth International Conference on Networking (ICN'07) (pp. 64-64). doi:10.1109/ICN.2007.40

Yadav, A. K., Garg, M. L., and Ritika, 2019. The Issues of Energy Efficiency in Cloud Computing Based Data Centers. Bioscience Biotechnology Research Communications, 12(2). doi: 10.21786/bbrc/12.2/35

Yadav, R., Zhang, W., Li, K., Liu, C., and Laghari, A. A., 2021. Managing overloaded hosts for energy-efficiency in cloud data centers. Cluster Computing, 1-15. doi.:10.1007/s10586-020-03182-3

Yavari, M., Ghaffarpour Rahbar, A., and Fathi, M. H., 2019. Temperature and energy-aware consolidation algorithms in cloud computing. Journal of Cloud Computing, 8(1), 1-16. doi: 10.1186/s13677-019-0136-9

Zahedi Fard, S. Y., Ahmadi, M. R., and Adabi, S., 2017. A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. The Journal of Supercomputing, 73(10), 4347-4368. doi: 10.1007/s11227-017-2016-8

Zhou, Z., Shojafar, M., Alazab, M., Abawajy, J., and Li, F., 2021. AFED-EF: An energy-efficient VM allocation algorithm for IoT applications in a cloud data center. IEEE Transactions on Green Communications and Networking, 5(2), 658-669. doi.org/10.1109/TGCN.2021.3067309