Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 26, Issues 3, pp. 1165–1179, DOI 10.12694/scpe.v26i3.4331

ENERGY EFFICIENCY TASK RE-SCHEDULING IN VIRTUALIZED CLOUD COMPUTING

P.HANUMANTHA RAO[*], P.S. RAJAKUMAR[†], AND S. GEETHA[‡]

Abstract. Reduced energy consumption is an important goal for virtualized cloud computing systems since it has the potential to improve system efficiency, save operating costs, and lessen environmental impact. These objectives can be achieved by using an energy-efficient approach to job scheduling. The huge challenge lies in coordinating user demands with available cloud resources in a way that maximizes performance while reducing energy usage, all within the time frame that the user specifies. This article suggests a novel method called Energy Efficient Task Re-scheduling (EETRS) for a heterogeneous virtualized cloud environment as a solution to the problem of energy usage. The first step of the suggested approach assigns jobs strictly according to due dates, ignoring energy consumption. Task reassignment scheduling determines the optimal execution location within the deadline constraints while minimizing energy consumption in the second stage of the proposed method, which speeds up execution and meets deadlines. According to the simulation results, the suggested technique helps to significantly reduce energy use and boost performance by 5% while satisfying deadline constraints, in comparison to the current energy-efficient scheduling methods of EPETS, AMTS, and EPAGA. The proposed method outperforms the existing one with less than 1% total execution time, a reduction of 14% in total execution cost, a 3% decrease in energy consumption, and a 3% reduction in average resource utilization.

Key words: Cloud Computing, Entergy Consumption, Task Re-scheduling, Energy minimization, Performance enhancement

1. Introduction. Computing entered a new age with the advent of cloud computing, as a result of technological advancements that integrated storage, processing power, and networks. The term ”cloud computing” refers to a new face of shared computing that enables users to gain access to shared computer resources whenever they need them through an internet connection. Providers of cloud computing services that host several applications have several responsibilities, including adhering to service-level agreements (SLAs), ensuring reliable and secure data management, meeting task deadlines, and achieving low access latencies. Commercial goals of cloud providers may conflict with energy-efficient, cost-effective hardware designs and capacity planning strategies used by back-end data centers.

Data center energy management is intricate because it requires real-time evaluation of dynamic factors such as traffic conditions, inter-process communication, workload and resource allocation, and cooling plans. With the market for cloud pricing becoming more intensely competitive, cloud companies are under increasing pressure to find ways to power down their data centers' backend [1]. The typical data center workload is around 30% and does not require full computer resources to be used [2]. Consequently, it is possible to match the workload demands of the data center while saving energy by turning off certain idle equipment. But data replication, client SLAs, performance, and latency issues, and data center traffic patterns [3] must all be carefully considered when data center resource scheduling is to be performed.

Critical and energy-intensive, data centers deliver Internet-based services on a massive scale. To reduce excessive energy use in data centers, power utilization models are crucial for developing and improving energy-efficient operations. The rapid expansion of distributed cloud computing network services has led to an explosion of data sizes across various industries, encompassing signal processing, bio-informatics, IoT, and scientific computing. Cloud computing applications make use of the thousands of powerful servers hosted in cloud data centers to execute millions of jobs. Virtualization is one of the greatest advantages for consumers who can

*Dept. of CSE, Dr.M.G.R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu, India (hanumanthraovbit@gmail.com)

†Dept. of CSE ,Dr.M.G.R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu, India (Rajakumar.subramanian@drmgrdu.ac.in)

‡Dept. of CSE, Dr.M.G.R. Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu, India (somangeetha@drmgrdu.ac.in)

take advantage of the many services offered by the cloud through virtual machines (VMs). A lot of power is typically used by these virtual machines. The cost of electricity goes up and the environment suffers as a result of this kind of energy usage [4]. Despite technological efforts to reduce energy use, the data center is predicted to discharge 62 million tonnes of carbon dioxide (CO2) into the environment [5]. Low utilization of computing resources and inefficient job scheduling are primary causes of enormous energy consumption [6]. In addition, the duties included in each application are diverse and might range in size. So, it's critical to manage energy consumption in cloud data centers and make them more energy efficient when scheduling. Energy efficiency in data centers is becoming increasingly important and complex in recent years. To keep data accessible at all times, the data center's various components must work together to reduce energy consumption and downtime. All information technology (IT) infrastructures are based on the technical infrastructure, which includes power supply, technical cooling, and technical security.

Presently, the vast majority of studies just allocate tasks to Virtual Machines (VMs) without taking into account the fact that different jobs have variable resource requirements [7]. A tremendous increase in energy use and a loss of valuable resources ensued. Regardless, getting optimal energy consumption and scheduling all workloads to appropriate servers is a huge concern. Many researchers in the academic community have worked on these problems to lower the power consumption of cloud data centers. Specifically, the energy-efficient scheduling issue in a wide variety of devices was investigated in research [8], [9], [10], and [11]. Scheduling all tasks according to their quality of service criteria while minimizing power usage was the main objective. Evidence from studies [12], [13], and [14] shows that hybrid data centers can save energy by combining powerful and less powerful servers. At now, most data centers use a variety of servers with varying degrees of computational power and power consumption features [15]. This means that various virtual machines (VMs) will have different execution durations and energy usage when processing the same activity. If the work is scheduled to the correct server form, it can help improve system performance and save energy usage.

The processing time of the task is practically related to the energy consumption problem. There are several obstacles to overcome in order to determine how long each task should take to complete in response to the following study questions.

1. Assigning tasks to the correct machine at the right time is difficult to reduce energy usage. Energy usage and resource waste are both exacerbated by the host's tough assignment of tasks.
2. Assigning all tasks to the same virtual machine, or one that is slower, would cause processing times to increase. Consequently, the scheduler will make the execution time increase, which will disrupt the deadlines of jobs. If all tasks are sent to a faster virtual machine, processing times will be reduced, but more energy will be consumed.
3. Earliest finish times (EFT), Processing times, latest completion times (LFT), and a deadline need to be defined in order to allocate jobs to virtual machines. There would be different slack times due to different allocations. Assigning work to slower machines reduces slack time and energy usage, but there is an enormous number of combinations of possible task allocation possibilities. Thus, evaluating all the combinations is extremely difficult, and perhaps impossible.

This research took into account several different tasks that each had their quality of service criteria, such as deadline, workload size, and execution priority. Using a diverse set of resources to choose a virtual machine that both satisfies a task's quality of service needs and uses the least amount of energy is thus becoming an increasingly difficult scheduling challenge. Given the aforementioned rationale, this article proposes a method, Energy Efficient Task Re-scheduling (EETRS), for scheduling tasks in a heterogeneous virtualized cloud that minimizes both energy consumption and enhances performance. The goal is to meet task objectives without sacrificing service quality while minimizing energy consumption during scheduling. The paper's outline is as follows: A brief synopsis of relevant studies is given in the second section. Section 3 describes the algorithm and workings of the proposed approach EETRS. Section 4 presents the suggested method's experimental settings, results, and a thorough performance analysis. Section 5 outlines conclusions and future steps.

2. Literature Review. Recent years have seen virtualization technologies rise to the forefront of computer system architecture once again.

Secure computing, transparent migration, and consolidation of servers are just a few examples of the new capabilities that may be added to a system through virtual machines (VMs), which also allow developers to keep existing operating systems and applications compatible. Modern virtualized environments need all virtual machines (VMs) to share the same core to follow the hypervisor-controlled power management strategy. Various constraints apply to certain configurations. There is little opportunity for users to personalize the power management parameters for individual virtual machines. As a second point, it often impacts the energy efficiency of some or all virtual machines, especially when those VMs have different energy regulating plans that they require.

To address the challenges mentioned above, Li et al. [16] proposed a technique of power control that is specific to each virtual machine (VM). This method would allow each VM's guest operating system to use its preferred energy management strategy while simultaneously preventing similar VMs from competing with one another. Virtual performance (VIP) improves the timing of CPU-intensive applications by 32% and reduces power consumption by 27% in contrast to the Xen hypervisor's default on-demand governor, respectively, all without violating the service level agreement (SLA) of latency-sensitive implementations. The individual strategy of energy management is not possible in practice. In addition to optimizing energy efficiency by analysing the VM scheduling mechanism along with the virtualization paradigm of Input and output, Lee et al. [17] introduced a new offset mechanism to conquer fast input and output performance while power-fairness credit sequencing strategy. Also, virtual machine resource calibration was presented by Sheikh et al. [18]. They developed a method that uses power monitoring services and controlled feedback architecture to lower virtual servers' energy consumption. The above methods were inefficient both in minimizing the energy and enhancing the performance successfully.

Architectures typically provide several methods and processes for how to distribute and organize work across many resources using methods, including virtual machine placement, migration, consolidation, scheduling that minimizes energy use, and virtualization. A virtual machine placement problem was addressed by the authors in [19] through the use of an online meta-heuristic method that was dependent on the Ant Colony System. An objective function is used by the algorithm to find an approximation of the ideal solution. They were able to achieve better power usage without sacrificing the performance needed for a cloud data center. This method was less efficient, compared to other existing solutions. In their study, Arianyan et al. [20] implemented a method that helps improve resource utilization and performance while decreasing energy usage in cloud data centers. Energy usage, live migration, and SLA violations formed the basis of their new and effective resource management system. They broke the whole issue down into two smaller ones. After identifying the overworked server, it uses a multi-criteria selection decision approach to choose which virtual machine (VM) to move. However, this method introduces a lot of time complexity. On the other hand, a great deal of work has been done to determine how energy consumption relates to performance by the authors of [21]. In an effort to strike a better balance between power consumption and performance, they rethought the VM integration metric with energy efficiency. They established an SLA conflict algorithm inside this framework that takes into account minimal power consumption, maximum utilization, and SLA conflict as ways to identify when a server is overloaded. This effort aims to minimize energy usage without sacrificing service quality. This method incurs in overhead of large complex computations.

An analysis of the energy utilization of cloud computing was given in [22]. Public and private clouds were also taken into account in the study, along with the energy requirements of data computing, storage, and switching. They proved that a large amount of cloud computing's energy demand could be attributable to power consumption during transit and switching. Cloud computing (CC) is seen by their proposed approach as a counterpart to a traditional supply chain and logistics problem that accounts for the cost or power consumption of processing, storing, and transferring physical items. In addition, Yang Qin et al. [23] use an ILP model based on task profile information to optimize energy consumption for real-time activities using the intra-task DVFS scheduling strategy, assuming zero transfer overhead. Obtaining the smallest average energy by determining the optimal execution frequency of each generic block is the primary task of this ILP model. An extension to the ILP formula finds the optimal execution frequency and places it in the program to insert the conversion instructions; this helps with the DVFS conversion overhead. This method fails to address the issue of resource idleness for a longer period.

An indicator of energy awareness is the growth of studies focusing on efficient task scheduling in response to rising computer system energy consumption. [24], [25] discuss research on DVFS-based energy-efficient task scheduling. Meeting quality of service requirements while optimizing energy consumption is the objective of these task schedules. Assigning CPUs and scaling frequencies are the two typical stages. Assigning tasks to the processor in such a way that they are all complete by the due date while using the least amount of energy possible is the goal. Quality of service requirements for real-time tasks are expressed in [26] using a general model. They improve energy-aware task allocation and the ability to relocate tasks during runtime. Upon a task's completion during runtime, the remaining tasks could not execute to their full potential. To cut down on power usage, they suggested a method for local task relocation and tweaked the frequencies of the related CPUs. This method fails to take into account the time frames of the tasks assigned.

At the OS, hardware, virtualization, and data center levels, the authors of [27] mapped out a taxonomy of energy-efficient computer system design, and they also discussed the causes and problems of excessive power/energy utilization. They looked into several methods of limiting power usage from the operating system level using DVFS and other algorithms and strategies for power savings, and they sorted them all out. However, they failed to address the issue with optimal solutions. An Energy-Aware Task Scheduling Algorithm (ETSA) was suggested by Rohith et al. in [28] to tackle the issues that are directly tied to task consolidation and scheduling. After going through a normalization method, the suggested ETSA algorithm takes into account the task's overall resource utilization, and completion time, and uses that information to determine the best time to run the program. It fails to minimize the energy consumption to the maximum extent. Furthermore, the authors in [29] brought attention to the research challenges associated with the conflicting demands of improving the QoS provided by cloud services while simultaneously cutting down on the power usage of data center assets. In order to combine data center capabilities while lowering the influence on the quality of service objectives, they tackled the idea of designing an energy-efficient data center controller. For the purpose of conducting energy-efficient operations, they looked at methods of managing and coordinating data center resources. In addition, they suggested resource controller cooperation and proposed the idea of a central controller. Various mechanisms and frameworks for designing energy-effective data centers were discussed by the authors in [30]. Operating systems, virtual machines, and software applications were the subjects of their investigation into various power models.

An adaptive task-scheduling technique was proposed by Yao Sheng et al. [31]. An evolutionary algorithm was suggested (E-PAGA) to accomplish adaptive regulations for various energy and performance needs in cloud tasks after they modelled the energy of virtual machines for scheduling work in the cloud. In order to choose the next generation using distinct energy and performance measurements, they developed two distinct fitness criteria. They suggested customizing the cloud work to each user's needs by adaptively adjusting the target performance and energy. In practice, this method fails to uphold the deadlines of the tasks scheduled. In [32], the authors address scheduling issues with multiple purposes by using the improved PSO algorithm (AMTS). They begin by formulating the scheduling problem. Optimal resource utilization, average cost, average power consumption, and task completion time are achieved by advancing the task scheduling policy. The acceptance of the adaptive acceleration coefficient is contingent upon the preservation of particle diversity. This method fails to address the problem of achieving timelines successfully for all the assigned tasks.

Authors in [33] proposed Energy and Performance-Efficient Task Scheduling [EPETS] to solve the energy usage problem with deadline restrictions in heterogeneous virtualized cloud computing. The suggested method aims to provide good performance while reducing total energy usage within the given time limitations. In order to achieve the deadline with minimal energy consumption, the task reassignment algorithm assigns work to a machine with a medium or low processing speed with relative ease. However, these method incurs huge overhead and employs complex calculations. All the above methods discussed lack in addressing the energy consumption problem while maintaining the performance along with the timelines of all the tasks in a heterogeneous cloud environment.

This research article presents a novel proposed method called Energy Efficient Task Re-scheduling (EETRS) for processing jobs on heterogeneous computing virtual machines using various components promptly. A variety of virtual computers build the system, and the scheduler is responsible for allocating work according to the system's quality of service standards. In the simulation stage, experiments were conducted on the real test-bed traces to evaluate the efficiency and efficacy of the suggested scheme.

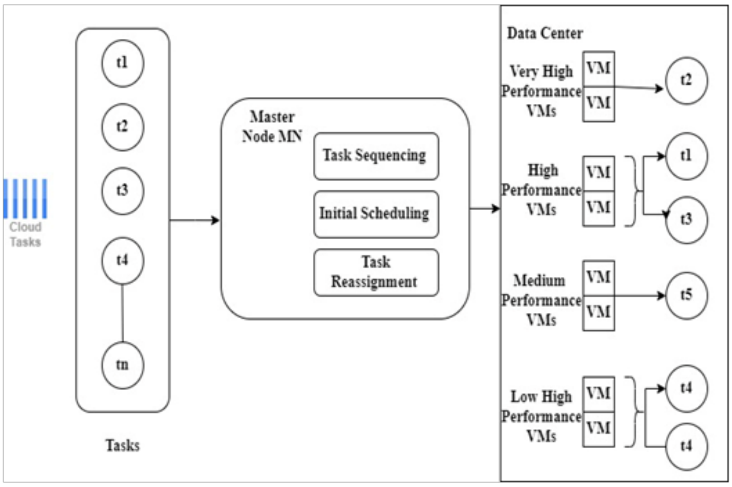

Fig. 3.1: Architectural framework of the proposed system

3. Proposed Method. The energy-efficient task scheduling problem on heterogeneous virtual machines is addressed in this article by proposing a method called Energy Efficient Task Re-scheduling (EETRS), which employs several strategies. Here are a few ways in which the EETRS algorithm differs from the aforementioned methods. The proposed method comprises the following steps:

1. An equitable trade-off between energy efficiency and task scheduling can be achieved by using the energy-efficient task priority framework, which is proposed in this paper.
2. The task finish time calculation is taken into account, which aids in enhancing system performance, decreasing energy usage, and preventing deadline breaches.
3. There are two parts to EETRS. The first step is to plan out all of the tasks that are needed to meet the SLA. The second step involves redistributing tasks to make the energy stability of the resources better

3.1. Proposed Architecture. Fig.3.1 shows the different components of the architecture that this study devises. Three components make up the suggested architecture. In the initial stage, users submit their independent tasks to the framework. The master node (MN) oversees multiple components in the management layer to ensure a successful execution in the second step of the proposed method. There are three main components: job reassignment, task sequencing, and initial scheduling. The research also takes into account the public data center single cloud, which includes four different kinds of heterogeneous virtual machines (VMs): small, medium, large, and very large. If you send tasks to the MN, the scheduling procedure will begin as follows and the proposed method will accomplish the following phases

1. Task sequencing is used to ease scheduling processes. These rules are applied to both submitted tasks and tasks that have come statically in the system.
2. To cut down on execution time and make deadlines, the scheduler initially tries to assign maximal and limited finish time tasks to the quicker machines.
3. To decrease energy usage without compromising deadlines, the faster machines should have their work redistributed to the slower ones

3.2. Task and Resource Features. A wide variety of independent tasks are now present in many large applications and businesses (e.g., manufacturing, disease diagnostics, etc.). Each task completes its own set of instructions. Different tuples of requirement quality (such as priority, workload, and due deadline) are stored for each job. However, there is a stringent standard for processing all tasks due to their characteristics. Due to this non-trivial challenge, Quality of Service (QoS) aware scheduling for distinct jobs in the cloud system has been taken into consideration. Under the QoS constraints, all jobs necessitate resource-intensive services and are computationally intensive. Amazon, Google, Alibaba, and Azure are just a few of the several public cloud providers that provide a range of service quality to their respective clientele. Virtual machines are used to access the services. Anyone may access these services anytime, anywhere because of the Internet. Computer systems and their ancillary components, including storage and telecommunications networks, are stored in data centers, which can be physical buildings, specific areas within larger buildings, or even clusters of buildings. A wide variety of services offered by the aforementioned suppliers allows the data center to host applications that require a lot of processing power and other resources. To efficiently conduct those mentioned compute-intensive operations, the data center can incorporate a variety of virtual machines into the system, such as a mix of vendors. Scheduling compute-intensive tasks in the cloud while considering their energy consumption and quality of service requirements is a complex and challenging issue. This research takes into account several virtual machines (e.g., Amazon) to manage the scheduling problem of many jobs. When compared to other sellers, Amazon's low-priced services are the key draw for customers.

3.3. Assumptions. During energy-efficient scheduling, this study takes into account the following assumptions about the problem. All jobs are distinct, with their workloads and due dates of personal ones.

1. With this setup, there is no need for communication between nodes because all tasks are scheduled on the same data center. Once tasks are assigned to virtual machines, they can be reassigned to other machines without affecting deadlines.
2. Since all machines share the same data center, only the power consumption of each machine is taken into account, which helps to reduce energy usage.
3. Task failure scenario was not considered in this study because it was assumed that all virtual machines could scale up and down. itemOn the other hand, the deadline requirement is considered when tasks are rejected.

3.4. Methodology. The process of the proposed method EETRS is detailed below:

1. The execution time of the task is calculated based on the size of the task and the computing power of the virtual machine.
2. The finish time of the virtual machine with current jobs is determined next.
3. The finish time of the task under consideration on a particular virtual machine is determined based on the finish time of the VM, as the third step.
4. Finally finish time of the task is compared with the deadline to assign it to that VM.
5. Then energy consumption of the VM is computed using its computing power and execution time taken by that task.
6. Finish time vs Deadline and Energy consumption of the VM is taken into consideration for reassigning the tasks for energy-efficient task scheduling.

To satisfy user needs, this study aims for a virtualized cloud with a collection of virtual machines ($V =$ v1, v2,..., vM). Different computing speeds along with powers create heterogeneous virtual machines, represented by Cj(j = 1,..., M) and Pj, respectively. The CPU power performance of VM vj is measured by MIPS (Million instructions per Second), as given in Equ.3.1

$$PW_j = \frac{C_j}{P_j} \tag{3.1}$$

where C_j is the Computing Speed and P_j is the Power

The suggested strategy EETRS aims to plan tasks to reduce cloud energy usage and meet data center VM deadlines. All jobs can be run on either faster or slower VMs. The tasks are delivered in a set of T= t1,t2,t3,...,tN and arrive simultaneously in a cloud system.

Each job ti contains parameters ti=si, di, where si(i = 1,...., N) and di represent task size and deadline. Initially, to preserve cloud resources, tasks are assigned to slower VMs to meet deadlines and reduce energy. The energy consumption of a task ti is determined by its power Pj and execution time EXTi, taking into account VM CPU processing power variations.

Let EXTfm i,j and EXTsm i,j represent task ti execution time on VM vj, where fm represents "faster machine" and sm represents "slower machine". The execution time of task ti is calculated as stated in Equ.3.2.

Execution time of the task on a virtual machine j

$$EXTi = \frac{S_i}{C_j} \tag{3.2}$$

where Si is te Size of the Task ti, Cj is the Computing Speed of the Virtual Machine j, Finish time FTj, 0 is initialized to 0 for each VM vj.

The execution time of the current task tk as mentioned in Equ.3.3.

$$\sum_{k=1}^{N} x_j, kEXTk \tag{3.3}$$

The current task $t - k$ and the finish time of the preceding job $t_k - 1$ determine the finish time FTj,k of the virtual machine vj when task tk is being executed.

The finish time of the Virtual Machine is given in Equ.3.4.

Finish Time of the VM_j

$$FT_{j,k} = T_{jk} - 1 + \sum_{k=1}^{N} xj, kEXTk \tag{3.4}$$

The task's completion time on VM vj is described using FTfm j,k and FTsm j,k, respectively.

The completion finish time of task ti is expressed using Ffm I (for faster machine) and Fsm i (for slower machine). Starting with each virtual machine (VM),

The finish time of task ti on VM vj can be computed as follows using Equ.3.5. Finish time of the task ti

$$Fi = \sum_{j=1}^{M} x_{i,j} * FT_{j,k} \tag{3.5}$$

where $FT_{j,k}$ = Finish time of the virtual machine.

Finish time Fi of the Task Ti completion time should be less than or equal to the deadline di, as stated in Equ.3.6.

$$Fi \leq di \tag{3.6}$$

3.5. Energy consumption model. The central processing unit (CPU), random access memory (RAM), disc storage, and network interface (NIC) are the primary determinants of data center computing server energy usage. Furthermore, there are two types of energy consumption: static and dynamic [34]. A computer's central processing unit (CPU) and other dynamic components account for the bulk of its static energy usage. Thus, we focus our attention on dynamic energy use when we construct models of energy consumption.

The power Pj of the virtual machine (VM) vj and the execution time (EXTi) of the process determine the energy consumption (Ei,j) that the task ti consumes while running on the VM vj as shown in Equ.3.6.

Energy consumption of VM:

$$E_{i,j} = Pj * EXTi \tag{3.7}$$

where, Pj is the power of VM, EXTi is the Execution Time of task ti.

When the faster virtual machine (VM) vj is used to perform task ti, the energy consumption of the task is shown as in Equ.3.7.

Energy Consumption on a faster machine:

$$Efm_{i,j} = Pj * ETi \tag{3.8}$$

By contrast, when the slower VM is used, the energy consumption is shown as in Equ.3.8.

$$Esm_{i,j} = Pj * ETi \tag{3.9}$$

The goal is to reduce cloud resource energy usage by completing all jobs as calculated below in Equ.3.9.

$$\min Etotal = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} * Pj * ETi \tag{3.10}$$

where $x_{i,j}$ is the Task sequenced, Pj is the power of VM, EXTi is the Execution Time of task ti.

The proposed system EETRS schedules tasks efficiently using the above methodology. This approach reduces cloud resource energy usage and boosts system efficiency while fulfilling deadlines. Slower machines can use less energy than faster ones. However, slower equipment can dramatically impair system efficiency and deadlines. Assigning all jobs to a slower machine deadline increases violations and system performance.

This proposed EETRS technique optimizes the performance by reassigning the tasks dynamically by considering the deadlines and energy usage.

3.6. Proposed Algorithm. The section provides an overview of the proposed system and how it addresses the problem of scheduling tasks in virtual machines in an energy-efficient manner across various environments. Energy Efficient Task Re-Scheduling, or EETRS, is an algorithm with two steps. First schedule to finish all jobs by their due date. This allows for a decrease in the total amount of time it takes to execute jobs. But the faster machines meant that the original plan used a lot of energy, even though it could attain fair efficiency. Second, the proposed novel task reassignment mechanism addresses this gap and lowers minEtotal's energy consumption. Figure 3.2 is a flow diagram depicting the entire EETRS job scheduling algorithm. The algorithm is described below (Algorithm 1).

Consider:

1. FiFM = Finish time of the task on a faster VM
2. FiSM = Finish time of the task on a slower VM
3. Ei FM = Energy consumption on faster VM
4. Ei SM = Energy consumption on slower VM
5. di = deadline of the task.

Then,

If FiFM <= di and Ei FM <= Ei SM

Then allocate the Task ti to a faster virtual machine

If FiFM <=di and Ei FM > Ei SM and FiSM <=di

Then allocate the Task ti to a slower virtual machine

If FiSM <=di and Ei FM > Ei SM

Then allocate the Task ti to a slower virtual machine

Else (in all other cases)

allocate the Task ti to a slower virtual machine.

4. Simulation and results. Here, the performance of the proposed Energy Efficient Task Re-Scheduling (EETRS) algorithm is shown, and how well it works with a variety of randomly generated task counts. EETRS is compared quantitatively with three other methods that are already in use: EPETS [33], AMTS [32], and E-PAGA [31].

4.1. Simulation Settings. Four distinct virtual machines (VMs) built on Amazon Elastic Compute Cloud (EC2) are taken into consideration. Table.4.1 shows the different virtual machines (VMs), their configurations, processing speeds, and power consumption. Although the processing rates and powers vary, the number of cores in each virtual machine is set to 1. Python is used to implement all parameters and algorithms. The system running on Windows Server has 8 GB of RAM and an Intel (R) Core (TM) i7-9750H CPU with 2.60 GHz.

4.2. Experimental Findings.

4.2.1. Task Sequinning. First, all tasks are categorized according to the FCFS and EDF methods for sequencing, which stand for First Come, First Serve, and Early Deadliest First, respectively. Task sequencing is a prerequisite to scheduling tasks on a network of diverse virtual machines.

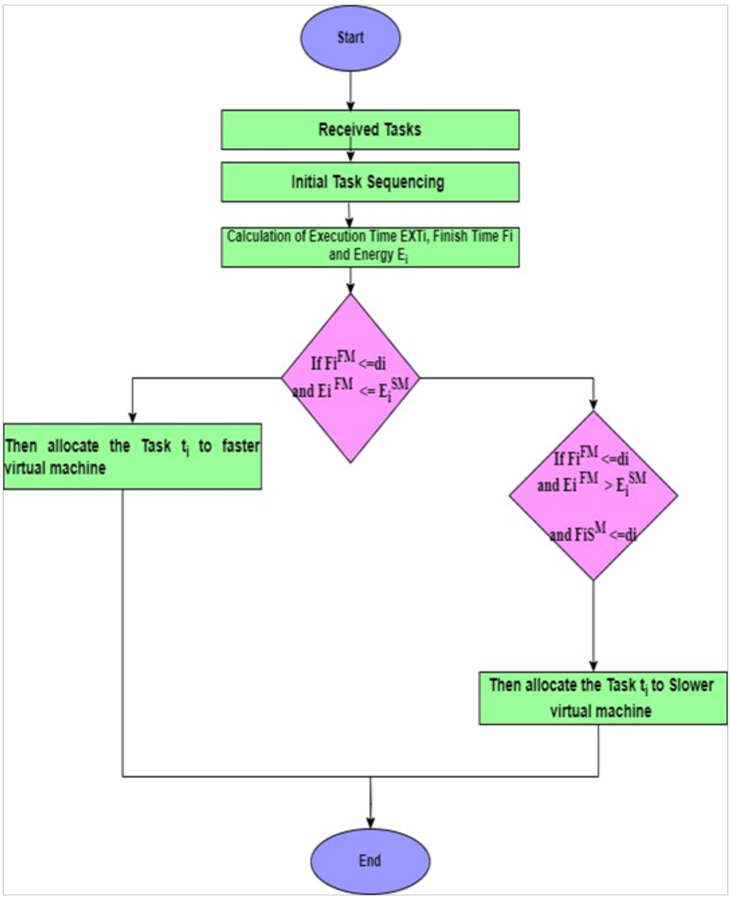

Fig. 3.2: Flowchart of the Proposed methodology EETRS.

Table 4.1: VMs Specification

VM	Small VM1	Medium VM1	Large VM
Core	1	1	1
MIPS/Core	200	400	600
Power/Core	50W	100W	200W

Table 4.2: Initial Scheduling

Task	VM1 (Slower)	VM2 (Slower)	VM3 (Medium)	VM4 (Faster)
T1	yes			
T2		yes		

4.2.2. Initial Scheduling. The optimal energy efficiency of the cloud resources is not taken into consideration when all tasks are planned on separate machines in the initial stage, depending on the task series. The goal of this stage is to organize everything in such a way that the total execution time can be lowered while still achieving all of the deadlines. Initial scheduling is tabulated in Table.4.2.

Algorithm 1 Task Scheduling

Input: Ti (Task), Si (Task Size), Ci (Computation Power), EXTi (Execution Time), FTfm (Finish Time-Faster Machine), FTsm (Finish Time-Slower Machine), EiSM (Energy-Slower Machine), EiFM (Energy-Faster Machine), di (Deadline).
Output: Ti (Task Allocation to the Virtual Machine)
Begin

1. Task Sequencing and Initial Scheduling
2. For each Ti ϵ Qt do
3. Execution Time, EXTi $\longleftarrow$ Si/Cj
   Finish Time of VM, FTj,k $\longleftarrow T_{j,k} - 1 + \sum_{k=1}^{N} xj, kEXTk$

   Finish Time of the Task Ti: $Fi \longleftarrow \sum_{j=1}^{M} x_{i,j} * FTj, k$
4. Energy Consumption, $E_{i,j} \longleftarrow Pj * EXTi$
5. Task Reassignment
6. If FiFM <= di and Ei FM <= Ei SM
   Then allocate the Task ti to a faster virtual machine
7. Else If FiFM <= di and Ei FM > Ei SM and FiSM <= di
   Then allocate the Task ti to a slower virtual machine
8. Else If FiSM <= di and Ei FM > Ei SM
   Then allocate the Task ti to a slower virtual machine
9. Else (in all other cases)
   Allocate the Task ti to a slower virtual machine
10. end if
11. End

Table 4.3: Task Re-Scheduling

Task	Execution Time	Deadline	Finish Time		Energy Consumption		Task Reassignment
T1	3	5	VM1S	6	VM1	150	T1 is assigned to VM2, due to less energy consumption and meeting deadline comparatively
	3	5	VM2S	5	VM2	150	
	2	5	VM3M	5	VM3	200	
	1	5	VM4F	4	VM4	200	
T2	4	6	VM1S	8	VM1	200	T2 is assigned to VM3, due to less energy consumption and meeting deadline comparatively.
	5	6	VM2S	8	VM2	250	
	2	6	VM3M	5	VM3	200	
	3	6	VM4F	4	VM4	600	

4.2.3. Task Re-Scheduling. The execution time, finish time and energy consumption are computed for the tasks sequenced and virtual machines available using the aforementioned equations. The stats are tabulated below in Table 4.3. Based on the calculations, tasks are reassigned to achieve energy efficiency while meeting the deadlines.

Task T1 is re-assigned to the slower virtual machine VM2 as execution time and deadline are within reachable limits and the energy consumption is less compared to other virtual machines. Task 2 is reassigned to Medium virtual machine VM3 due to less energy consumption and meeting the deadline compared to other virtual machines.

4.3. Performance Analysis. EETRS is compared to EPETS [33], AMTS [32], and E-PAGA [31] in terms of execution time, cost, energy usage, and resource utilization. The suggested method outperforms existing methods in all performance parameters, as shown below.

Table 4.4: Total Execution time compared with No. of Tasks

No. of Tasks	Proposed EETRS	EPETS	AMTS	EPAGA
10	616	620	812	1078
20	982	996	1134	1256
40	1274	1292	1354	1567
60	1656	1663	1838	2085
80	1864	1872	2092	2234
100	2005	2028	2384	2476

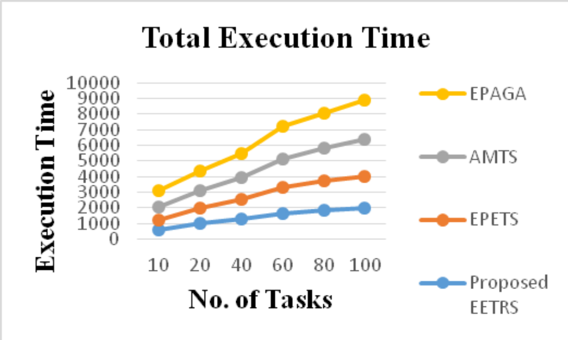

Fig. 4.1: Total Execution time vs No. of Tasks.

Table 4.5: Total Execution cost compared with No. of Tasks

No. of Tasks	Proposed EETRS	EPETS	AMTS	EPAGA
10	90012	90036	93245	97425
20	96054	96114	98423	102243
40	101224	102315	103245	107452
60	109018	110031	120234	123314
80	125423	128535	138354	145564
100	128434	133537	143637	156672

4.3.1. Total Execution Time. As seen in Fig.4.1 execution time was plotted versus job count. It is evident that EETRS takes equivalent execution times as other methods EPETS, AMTS, and EPAGA lag in this field a bit. Table.4.4 shows the total execution time of all the algorithms EETR, EPETS, AMTS, and EPAGA. From the graph in Fig.4.1, and Table.4.4, it is clear that the proposed EETRS shows less execution compared to EPETS, AMTS, and EPAGA algorithms. The proposed EETRS shows 1 % less execution time than EPETS over an average no of jobs.

4.3.2. Total Execution Cost. Fig.4.1 shows an association between job number and total execution cost. This graph shows that EETRS has a lower total cost than EPETS, AMTS, and EPAGA. Table.4.5 shows the total execution costs for the proposed EETRS and other existing algorithms. The stats show that EETRS exhibits a lower cost of 14% when compared to other algorithms over an average no of jobs.

4.3.3. Energy Consumption. The suggested EETRS algorithm has much lower energy usage compared to the EPAGA, EPETS, and AMTS algorithms. Fig.4.2 and Table.4.6 present the energy consumption comparison of proposed and existing algorithms. The benchmark algorithms generate massive energy consumption preparation, as shown by this figure and table. The rationale for this is that approximate task execution position generation does not yield optimal results when VM performance is reasonably high. The suggested approach

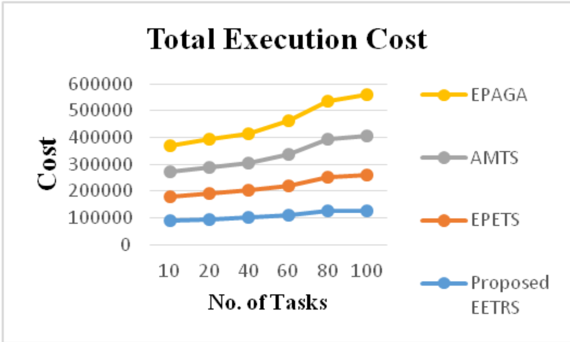

Fig. 4.2: Total Execution cost vs No. of Tasks

Table 4.6: Energy consumption compared with No. of Tasks

No. of Tasks	Proposed EETRS	EPETS	AMTS	EPAGA
10	4	7	9	12
20	6	9	12	17
40	8	13	16	22
60	12	17	21	27
80	14	21	24	32
100	17	24	27	38

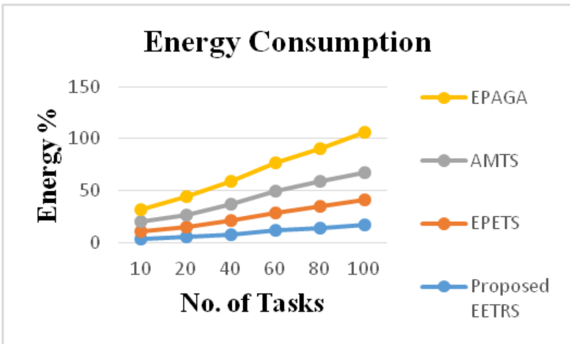

Fig. 4.3: Energy consumption vs No. of Tasks.

EETRS optimizes the use of slower computer resources, uses energy-efficient job sequencing, and reassigns tasks to machines with more spare time, resulting in higher energy efficiency than the existing algorithms. In addition, free machines can sit idle, significantly reducing energy consumption, when jobs on faster machines are transferred to slower machines. Because of such, the energy consumption outcome. The proposed EETRS shows a less energy consumption of 3% compared to other existing methods.

4.3.4. Average Resource Utilization. When it comes to scheduling tasks, the cloud data center's resource utilization is important. The utilization is contingent upon the workload execution that various users request. Equ.4.10 measures the average utilization of resources and performances.

$$AvgUtilization = \frac{\sum_{j=1,i=1}^{M} Si/Cj}{M} \tag{4.1}$$

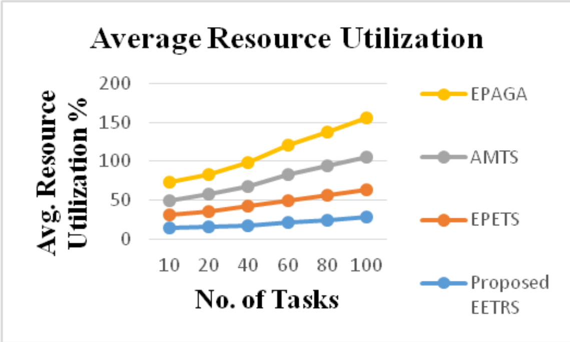

Fig. 4.4: Average resource utilization vs No. of Tasks

Table 4.7: Average resource utilization compared with No. of Tasks

No. of Tasks	Proposed EETRS	EPETS	AMTS	EPAGA
10	14	17	19	23
20	16	20	22	25
40	18	24	26	31
60	22	28	33	38
80	25	31	38	43
100	28	35	43	49

where Si is the Size of the task, Cj is the Computing power of VM.

When it comes to scheduling and accomplishments, the EETRS system improves resource consumption (e.g., CPU usage accurately without wastage). On the other hand, these resource and efficiency savings from system utilization in a network of heterogeneous virtual machines have not been considered in previous research. The multiple reasons why EETRS makes efficient use of its resources without wasting are listed below. (i) In the first stage, all tasks are planned to meet the needs of the SLA. (ii) In the second stage, tasks are redistributed to make the resource more energy stable. (iii) Deadline times save resources by task sequencing, thus no pool of resources is lacking. Thus, the suggested approach EETRS improves the system's overall reliability and maintains its full performance in the long run. Figure 4.4 and Table 4.7 show the average resource utilization of all the algorithms including proposed EETRS and existing under consideration. The proposed EETRs exhibit effective resource utilization with 3% lower when compared with other algorithms.

5. Conclusion. This article covers the Energy Efficient Task Re-Scheduling problem with deadline constraints in heterogeneous virtualized cloud computing. The proposed strategy EETRS performed better than all existing schemes, according to the simulation results. In order to meet the deadline, the proposed solution aims to achieve good performance while reducing the total usage of energy and improving the overall data center performance by lowering the execution time. The purpose of this study is to present an EETRS heuristic method that combines initial scheduling with task reassignment scheduling. Before any scheduling can take place, tasks must be sequenced, primary assignments made, and execution slots distributed. This method initially tried to assign the fastest machines maximum and lowest slack time tasks in the first scheduling without considering energy optimization. In the second stage, to reduce power consumption and still make the deadline, the task reassignment algorithm moves jobs from a fast computer to a medium or slower one. The suggested EETRS algorithm outperformed the existing methods in terms of total execution time, total execution cost, energy consumption, and resource utilization according to the simulation findings. In subsequent research, quantum-inspired methods can be included in a heterogeneous virtualized cloud environment to refine the proposed EETRS algorithm for scheduling tasks with minimal energy consumption.

REFERENCES

[1] Dayarathna, Miyuru, Yonggang Wen, and Rui Fan. "Data center energy consumption modeling: A survey." IEEE Communications surveys & tutorials 18.1 (2015): 732-794.

[2] Kliazovich, Dzmitry, et al. "Energy consumption optimization in cloud data centers." Cloud services, networking, and management (2015): 191-215.

[3] Zafar, Saima, Shafique Ahmad Chaudhry, and Sara Kiran. "Adaptive trimtree: Green data center networks through resource consolidation, selective connectedness and energy proportional computing." Energies 9.10 (2016): 797.

[4] Wu, Yu, et al. "Green data center placement in optical cloud networks." IEEE Transactions on Green Communications and Networking 1.3 (2017): 347-357.

[5] Ghafari, R., F. Hassani Kabutarkhani, and Najme Mansouri. "Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review." Cluster Computing 25.2 (2022): 1035-1093.

[6] Zhu, Xia, Mehboob Hussain, and Xiaoping Li. "Energy-efficient independent task scheduling in cloud computing." Human Centered Computing: 4th International Conference, HCC 2018, Mérida, Mexico, December, 5–7, 2018, Revised Selected Papers 4. Springer International Publishing, 2019.

[7] Chen, Huangke, et al. "EONS: minimizing energy consumption for executing real-time workflows in virtualized cloud data centers." 2016 45th International Conference on Parallel Processing Workshops (ICPPW). IEEE, 2016.

[8] Jena, R. K. "Energy efficient task scheduling in cloud environment." Energy Procedia 141 (2017): 222-227.

[9] Medara, Rambabu, and Ravi Shankar Singh. "Energy efficient and reliability aware workflow task scheduling in cloud environment." Wireless Personal Communications 119.2 (2021): 1301-1320.

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

[11] Kak, Sanna Mehraj, Parul Agarwal, and M. Afshar Alam. "Task scheduling techniques for energy efficiency in the cloud." EAI Endorsed Transactions on Energy Web 9.39 (2022): e6-e6.

[12] Tang, Chaogang, et al. "Energy-aware task scheduling in mobile cloud computing." Distributed and Parallel Databases 36 (2018): 529-553.

[13] BEN ALLA, Said, et al. "An efficient energy-aware tasks scheduling with deadline-constrained in cloud computing." Computers 8.2 (2019): 46.

[14] Walia, Navpreet Kaur, et al. "An energy-efficient hybrid scheduling algorithm for task scheduling in the cloud computing environments." IEEE Access 9 (2021): 117325-117337.

[15] Yang, Jiachen, et al. "A task scheduling algorithm considering game theory designed for energy management in cloud computing." Future Generation computer systems 105 (2020): 985-992.

[16] Li, Xiaoping, et al. "Energy-aware cloud workflow applications scheduling with geo-distributed data." IEEE Transactions on Services Computing 15.2 (2020): 891-903.

[17] Lee, Heecheon, et al. "Energy-efficient design of a novel double annular separation column using pinch pressure." Industrial & Engineering Chemistry Research 59.32 (2020): 14398-14409.

[18] Sheikh, Hafiz Fahad, Ishfaq Ahmad, and Sheheryar Ali Arshad. "Performance, energy, and temperature enabled task scheduling using evolutionary techniques." Sustainable Computing: Informatics and Systems 22 (2019): 272-286.

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

[20] Aryania, Azra, Hadi S. Aghdasi, and Leyli Mohammad Khanli. "Energy-aware virtual machine consolidation algorithm based on ant colony system." Journal of Grid Computing 16 (2018): 477-491.

[21] Zhang, Qing, et al. "Energy-aware scheduling in edge computing based on energy internet." IEEE access 8 (2020): 229052-229065.

[22] Shu, Zhaogang, et al. "Cloud-integrated cyber-physical systems for complex industrial applications." Mobile Networks and Applications 21 (2016): 865-878.

[23] Qin, Yang, et al. "Energy-efficient intra-task DVFS scheduling using linear programming formulation." Ieee Access 7 (2019): 30536-30547.

[24] Wang, Songyun, et al. "A DVFS based energy-efficient tasks scheduling in a data center." Ieee Access 5 (2017): 13090-13102.

[25] Yu, Ke, et al. "An improved DVFS algorithm for energy-efficient real-time task scheduling." 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2020.

[26] Ahn, Gilseung, and Sun Hur. "Multiobjective Real-Time Scheduling of Tasks in Cloud Manufacturing with Genetic Algorithm." Mathematical Problems in Engineering 2021.1 (2021): 1305849.

[27] Tian, Wenhong, et al. "On minimizing total energy consumption in the scheduling of virtual machine reservations." Journal of Network and Computer Applications 113 (2018): 64-74.

[28] Zambre, Rohit, et al. "Breaking band: A breakdown of high-performance communication." Proceedings of the 48th International Conference on Parallel Processing. 2019.

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

[30] Ahmed, Kazi Main Uddin, Math HJ Bollen, and Manuel Alvarez. "A review of data centers energy consumption and reliability modeling." IEEE access 9 (2021): 152536-152563.

[31] Shen, Yao, et al. "Adaptive task scheduling strategy in cloud: when energy consumption meets performance guarantee." World Wide Web 20 (2017): 155-173.

[32] Mubeen, Aroosa, et al. "Alts: An adaptive load balanced task scheduling approach for cloud computing." Processes 9.9 (2021):

1514.
[33] Hussain, Mehboob, et al. "Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing." Sustainable Computing: Informatics and Systems 30 (2021): 100517.
[34] Katal, Avita, Susheela Dahiya, and Tanupriya Choudhury. "Energy efficiency in cloud computing data centers: a survey on software technologies." Cluster Computing 26.3 (2023): 1845-1875.

Edited by: Dhilip Kumar V
Special issue on: Unleashing the power of Edge AI for Scalable Image and Video Processing
Received: Jul 22, 2024
Accepted: Sep 26, 2024