



ARTICLE

A Novel Energy and Communication Aware Scheduling on Green Cloud Computing

Laila Almutairi¹ and Shabnam Mohamed Aslam^{2,*}

¹Department of Computer Engineering, Computer Science and Information Technology College, Majmaah University, Al Majmaah, 11952, Saudi Arabia

²Department of Information Technology, Computer Science and Information Technology College, Majmaah University, Al Majmaah, 11952, Saudi Arabia

*Corresponding Author: Shabnam Mohamed Aslam. Email: s.aslam@mu.edu.sa

Received: 11 March 2023 Accepted: 27 September 2023 Published: 26 December 2023

ABSTRACT

The rapid growth of service-oriented and cloud computing has created large-scale data centres worldwide. Modern data centres' operating costs mostly come from back-end cloud infrastructure and energy consumption. In cloud computing, extensive communication resources are required. Moreover, cloud applications require more bandwidth to transfer large amounts of data to satisfy end-user requirements. It is also essential that no communication source can cause congestion or bag loss owing to unnecessary switching buffers. This paper proposes a novel Energy and Communication (EC) aware scheduling (EC-scheduler) algorithm for green cloud computing, which optimizes data centre energy consumption and traffic load. The primary goal of the proposed EC-scheduler is to assign user applications to cloud data centre resources with minimal utilization of data centres. We first introduce a Multi-Objective Leader Salp Swarm (MLSS) algorithm for task sorting, which ensures traffic load balancing, and then an Emotional Artificial Neural Network (EANN) for efficient resource allocation. EC-scheduler schedules cloud user requirements to the cloud server by optimizing both energy and communication delay, which supports the lower emission of carbon dioxide by the cloud server system, enabling a green, unalloyed environment. We tested the proposed plan and existing cloud scheduling methods using the GreenCloud simulator to analyze the efficiency of optimizing data centre energy and other scheduler metrics. The EC-scheduler parameters Power Usage Effectiveness (PUE), Data Centre Energy Productivity (DCEP), Throughput, Average Execution Time (AET), Energy Consumption, and Makespan showed up to 26.738%, 37.59%, 50%, 4.34%, 34.2%, and 33.54% higher efficiency, respectively, than existing state of the art schedulers concerning number of user applications and number of user requests.

KEYWORDS

EC-scheduler; green cloud; energy efficiency; task scheduling; task sorting; resource allocation

1 Introduction

The terminology Green Cloud Computing has evolved through Parallel-Computing, Grid-Computing, and Utility-Computing technologies. The conventional technologies of resource sharing provide the basis for Green Cloud Computing's emergence in the current era.



1.1 Parallel Computing

Traditionally, the software has been written for serial computation; that is, for programs to be run on a single computer with a single Central Processing Unit (CPU), and a problem is broken down into a discrete series of instructions, and instructions are executed one after another. Only one instruction may be performed at any moment. Later, there was a need for complex computing systems to solve complex problems using mass data volumes. One solution to this problem is Parallel Computing [1]. Parallel Computing involves using multiple computing resources to solve a computational problem in which programs are run using multiple CPUs. A problem can be divided into discrete parts that can be solved simultaneously. Each piece is further broken down into a series of instructions, where each function executes simultaneously on different CPUs. Traditionally, parallel computing has been considered to be “the high end of computing” and has been motivated by numerical simulations of complex systems and “Grand Challenge Problems”, such as weather and climate, chemical and nuclear reactions, human genome, geology, seismic activity mechanical devices, prosthetics, and spacecraft electronic circuit manufacturing. Parallel computer architectures will become increasingly hybrid, combining hardware multithreading, many cores, SIMD unit accelerators, and on-chip communication systems, which require the programmer and compiler to adopt parallelism, orchestrate computations, and manage data locality at several levels to achieve reasonable performance.

1.2 Grid Computing

In early 2000, the process of computing became pervasive, and individual users (or client applications) gained access to computing resources (processors, storage, data, applications, etc.) as needed, with little or no knowledge of where those resources are located or the underlying technologies, hardware, and operating system. This paradigm is known as Grid Computing. If we focus on distributed computing solutions, we can consider one definition of grid computing as distributed computing across virtualized resources [2]. The goal is to create the illusion of a simple yet large and powerful virtual computer from a collection of connected (and possibly heterogeneous) systems sharing various combinations of resources. Grid computing provides an architecture for creating a virtual supercomputer comprising distributed computer nodes. Most grid computing projects have no time dependency, and large projects are typically deployed across many countries and continents. In many cases, a grid computing system leverages a node’s idle resources to perform grid-related tasks, known as cycle-scavenging or CPU-scavenging.

1.3 Utility Computing

With the growing demand for computing resources and network capacity, providing scalable and reliable computing services on the Internet has become challenging. Recently, more attention has been paid to the Utility Computing concept, which aims to provide computing as a utility service, similar to water and electricity. Utility Computing offers online computation or storage as a commercial metered service, computing on demand, or cloud computing [2]. It creates a “virtual supercomputer” by using spare computing resources within an organization. Utility computing differs from cloud computing because it relates to the application infrastructure resource business model. This could be either hardware or software delivered, whereas cloud computing relates to how it is designed, deployed, built, and runs applications that operate in a virtual environment.

1.4 Green Computing

Green cloud Computing refers to an environmentally friendly Cloud Computing system that minimizes energy consumption and reduces carbon emissions from the computing system. The green characteristics of Information and Communication Technology (ICT) products and services can be observed in sustainability-related concepts, including green ICTs, ecological informatics, environmental informatics, sustainable computing, and green computing. ICTs have been studied throughout their lifecycle to promote green and sustainable development. This can substantially contribute to improving the existing state of the environment by mitigating adverse effects that have become more severe in recent decades. Producers are under intense pressure to comply with environmental standards and offer products and services that have the least detrimental impact on the ecosystem.

1.5 Migration to Sustainable Green Cloud Computing

Green cloud computing involves designing, producing, and using digital spaces that primarily minimize unfavourable environmental impacts. It involves finding and producing energy-saving digital methods to minimize carbon emissions to the ecosystem [3,4]. It saves energy and reduces the enterprise costs required for operations. Cloud storage benefits are addressed by users using green cloud computing while simultaneously diminishing unfavourable climatic impacts, thereby affecting human well-being. Green cloud computing is used for many purposes, such as allocating resources and improving communication protocol performance [5]. Cloud computing information is expected to increase with the rapid development of cloud computing. This has become an unavoidable trend in developing data centres for green cloud computing [6]. Green cloud data centres have become progressively more significant because they must manage an ever-increasing number of cloud platform administrators. Different operations run simultaneously in a green cloud, necessitating high-scope framework resources that commonly incorporate many servers and cooling offices [7]. To achieve high energy proficiency, every application is generally sent to various green cloud data centres situated in various locations. Each green cloud data centre generally requires several megawatts of force network energy and environmentally friendly power for cooling and executing different tasks. In addition, because the energy cost of green cloud data centres is increasing, it is vital to improve the number of servers in the green cloud at its ongoing speed [8]. With the increasing size of green cloud data centres, energy consumption is increasing. For collaborative computing, cloud computing is a supercomputer model that uses high-bandwidth networks, large-scale storage systems, large data centres, and various distributed computing resources. Consequently, effective management is required for many servers in data centres [9]. The monitoring objective for energy utilization in cloud data centres is to use green cloud computing, which is a novel computing model. Green cloud computing administrators have paid significant attention to the utilization productivity of energy, providing step-by-step directions for reducing emissions of carbon by-products, thereby saving money [10]. In any case, a decrease in energy utilization might create more time for the service's response, which affects service performance; therefore, striking a balance between energy usage and performance is critical. Utility, grid, and parallel computing are some of the stages of cloud computing [11,12]. In green cloud computing, energy efficiency has become a paradox owing to the rapid development of data centres, while it may result in decreased performance and delays in service response in terms of performance and energy. A group of green communications, cloud computing and innovation, and communications aims to increase data centre computability while reducing carbon dioxide emissions [13]. The green cloud management system is the main resource to balance the underlying scheduling resources and infrastructure. Resource scheduling is under investigation, and no industry standard has yet been established in green cloud computing [14,15]. For cloud computing applications, more communication

resources are required. Cloud applications require more bandwidth to transfer large amounts of data and satisfy end-user requirements. There must be no communication source that causes congestion or loss owing to unnecessary switching buffers. Green computing is based on reducing energy consumption using optimal algorithms [16]. Data centres must effectively manage their resources to reach this goal in a green cloud. Using optimal scheduling algorithms [17] in green cloud computing to assign tasks to specific resources reduces processing time and energy usage. Tasks arrive regularly, and it is impossible to perform all of them with the restricted resources available in green cloud data centres. The admissions controller mechanism in green cloud data centres is often structured to reject specific jobs and avoid overpopulation. There has been no evidence of a link between job rejection by applicants and green cloud data centre invoices. An approach known as Time-Aware Task Scheduling (TATS) [18] considers temporal variance, and the tasks admitted are scheduled to run green cloud data centres while staying within their delay restrictions. The Spatial Task Scheduling and Resource Optimization (STSRO) technique [19] reduced the overall service cost by effectively scheduling all incoming activities from diverse products to satisfy task delay-bound limitations. Green energy-gathering Distributed Energy Resources (DER) [20] can aid in alleviating energy poverty and pursuing high network energy efficiency. The energy and non-service level agreement aware algorithm (EANSA) [21,22] uses the environment to target energy reduction but not explicitly the workload, power consumption model, and experimental setup. The high energy utilization of cloud data centres has become an important topic in the ICT world [23]. The Amazon Cloud computing platform Amazon Elastic Cloud (EC2) with a VANET simulator explores the performance efficiency of cloud solutions [24]. Computer manufacturing companies, such as Microsoft, Dell, and Hewlett-Packard, contribute to Green computing by manufacturing environmentally friendly computer hardware, such as energy-efficient processors designed with power-saving algorithms [25]. IBM investigated the elements of sustainable ICT, discussed its evolution as a service, and offered criteria to increase its alignment with corporate sustainability strategies. Soft computing techniques solve various task-scheduling problems in cloud computing environments. Different algorithms, such as the genetic algorithm, particle swarm optimization, ant colony optimization, and artificial bee colony, are suitable for efficiently scheduling tasks to resources. We propose a meta-heuristic bio-inspired approach called EC-scheduler to schedule tasks to optimize resource utilization in a Green cloud environment.

1.6 Research Objective

The objective of our work is green cloud computing and energy- and communication-aware scheduling (EC-scheduler), which optimizes data centre energy consumption and traffic load. The primary objectives of the proposed EC-scheduler are as follows:

1. A Multi-Objective Leader Salp Swarm (MLSS) algorithm is used for task sorting to balance traffic load.
2. An Emotional Artificial Neural Network (EANN) is utilized for efficient resource allocation based on cloud user requirements, which jointly optimizes energy- and communication-related delay and packet losses to ensure Quality of Service (QoS).
3. EC-scheduler is implemented in the GreenCloud simulator, and the results demonstrate that it enhances QoS performance.

The remainder of this paper is organized as follows. [Section 2](#) presents a literature survey, and [Section 3](#) concerns the proposed EC-scheduler and system model, task sorting, and resource allocation algorithms using mathematical models. [Section 4](#) describes the EC-scheduler's performance evaluation and comparative analysis with existing energy-centric schedulers. Finally, [Section 5](#) concludes the paper.

2 Literature Survey

This section examines recent literature on green cloud computing and energy saving from various perspectives. Table 1 summarizes the existing literature in several areas. As with existing research, in the Clonal Selection Resource Scheduling Algorithm (CSRSA) [26], resource-aware scheduling is performed based on the clonal selection principle and a load-balancing method and is statistically proven to improve performance. This fact ensures that research on the scheduling and optimal allocation of resource nodes in data centres can minimize cloud platform maintenance and operating costs and that heat generation and energy consumption are practically and theoretically essential for green cloud computing. According to previous research, the CSRSA significantly reduces energy consumption in green cloud computing, and its exploitation and exploration capabilities are balanced and improved. A time and energy-aware algorithm was proposed for task scheduling in a diverse context. The Energy Trade-Off Multi-Resource Cloud Task Scheduling Algorithm (ETMCTSA) [27] recognizes the importance of developing a technique that constantly alters them depending on the latest workload conditions, but also improved trade-offs need not just tune static α . The probability parameters of the algorithm can be adjusted by users to regulate and manage the energy consumption and performance of the cloud system. A Spatiotemporal Task Scheduling Algorithm (STTS) [28] was used to schedule incoming tasks efficiently to fulfil delays and reduce energy consumption. Temporal and spatial differences in distributed green data centres were thoroughly studied using STTS. Nonlinear restricted optimization issues were solved, such as energy cost minimization problems. While meeting all task delay bound criteria precisely, STTS achieves lower energy costs and higher Throughput than several other task scheduling systems. By task scheduling various applications intelligently to fulfil their response time restrictions, the Profit-Sensitive Spatial Scheduling algorithm (PS3) [29] was proposed to increase the total profit of a distributed green data centre provider. This scheduling approach effectively utilizes those, as mentioned above, variable spatial diversity. PS3 solves the profit maximization problem for a distributed green data centre provider as a restricted nonlinear program and achieves higher Throughput and total profit than two common approaches of task scheduling according to real-life trace-driven simulation trials. The Grey Wolf Optimisation Algorithm (GWO) [30] was used to solve the issue of workflow scheduling in green cloud computing data centres, where its goal is to reduce the cost, time, and power consumption for executions.

Table 1: Summary of research gaps

Ref.	Scheduling scheme	Target	Remarks
[26]	CSRSA	Energy consumption, load balancing	Energy reduction may result in a service delay.
[27]	ETMCTSA	Time and energy-aware	Certainly causes synchronization and task subdivision problems
[28]	STATS	Delay and energy consumption	Highly challenging to minimize
[29]	PS3	Higher total profit and throughput	Desirable explored segments of the search space
[30]	IGWO	Power consumption, cost, and makespan	The task scheduling is acceptable against reliability
[31]	GHSA_di	Energy consumption and scalability	Real Pareto front problem due to huge workload

(Continued)

Table 1 (continued)

Ref.	Scheduling scheme	Target	Remarks
[32]	FSTS	Lower energy cost	Huge makespan, communication costs
[33]	JVCMMD	Energy consumption, container migrations and SLA violations	Scheduling against reliability and stability
[34]	Lion optimization (LO.)	Energy efficiency	Uncertainty propagation towards task execution and data transfer time
[35]	NLR prediction model	SLA violation	Reduction of energy consumption deteriorates task performance

The algorithm was tested and found to reduce the cost, energy, and runtime in a simulation. To emphasize the benefits of hardware energy regulation concepts, a co-evolutionary dynamics equation was used in the Heuristic Scheduling Algorithm (GHSA) di algorithm [31]. The GHSA di algorithm has been used for its apparent scalability, energy savings, and overall performance for data- and computationally-intensive cases. Hardware energy regulation principles are emphasized and exploited in the co-evolutionary dynamics equation. Three-dimensional biomimetic encoding and decoding considering individuals and their corresponding evolutionary mechanism for scheduling and creative hierarchical parallelization are suited to schedule servers' super-hybrid systems. To reduce the data centre provider's energy costs by allocating heterogeneous application tasks optimally across many data centres, to stay under response time constraints of the studies strictly, and to specify the quantity of running speed and power on servers of every server in data centres, a fine-grained resource Provisioning and Task Scheduling (FSTS) algorithm [32] was utilized. By comparing various up-to-date scheduling approaches, real-world data-driven trials show that FSTS saves energy, thereby ensuring the highest Throughput.

To solve the issue of green resource management in container-based cloud data centres, parameters such as energy usage, quantities of container migrations, Virtual Machine (VM) and Service Level Agreement (SLA) violations are considered. The eight subproblems that constitute the container consolidation problem and joint VM are solved using a Joint Virtual Machine and Container Migration (JVCMMD) algorithm [33] for deciding VM transfer. To show that their solutions have a significant migration in reducing energy consumption, the number of VMs migrating to cloud data centres and SLA violations, the CloudSim simulator was used to confirm the applicability of their policies. The green and cloud manager layers [34] provided an approach for adequate resource availability to users with an uncompromised QoS. From each accessible resource, the cloud manager layer is accountable for choosing suitable resources. The best one is selected by the green manager layer. Due to its optimal resource selection, the standard service response time diminishes with decreased power utilization. The managing layers consider the distance between the cloud server and service requester, assigning the queue length, optimum resource, and present workload, thus further developing the QoS. Using the New Linear Regression (NLR) and Modified Power-Aware Best-Fit Decreasing (MPABFD) algorithms to detect under- and overloaded hosts resulted in good performance. The NLR [35] prediction model significantly outperformed the eminent expectation models, as indicated by outcomes and execution examinations. The NLR forecast model reduces energy use and SLA violations through CoT utilization to establish a sustainable and intelligent

climate for smart cities. Our proposed Energy-and-Communication-aware-scheduling system aims to bridge the research gaps of existing systems, such as energy reduction without service delay [26], synchronization and task subdivision problems due to the time and energy-aware scheduling [27], delay and energy consumption while scheduling [28], desirable explored segments of the search space [29], reliability of the system while achieving power consumption, cost, and makespan during scheduling [30], real Pareto front problem due to huge workload, Huge makespan, communication costs [31], high communication costs [32], scheduling against reliability and stability [33], uncertainty propagation towards task execution and data transfer time [34], and reduction of energy consumption deteriorating task performance [36]. Dynamic resource provisioning is a critical challenge due to the varying task resource requirements in green cloud computing. An abnormal workload causes resource scarcity, waste, and erratic resource and task allocation, all affecting task scheduling and contributing to SLA violations. There is inefficient and environmentally hazardous use of cloud resources. Recognizing the seriousness of this situation, several scholars have contributed to promoting green cloud computing by using various methods. Green cloud computing is implemented to improve the utilization of calculating assets to decrease energy usage and the ecological consequences of their use. As a result, the importance of green computing has increased to minimize data centres' harmful effects, energy, CO₂ emissions, and water and power consumption, which are hazardous to the environment. [Table 1](#) presents the significant characteristics of existing scheduling systems.

The research gaps are addressed by the following research objectives:

1. The proposed EC-scheduler was devised to optimize energy- and communication-aware scheduling.
2. To generate the required power consumption using our EC-scheduler, suitable for a green cloud computing environment.
3. To migrate virtual machines between servers in green cloud data centres while adhering to power consumption and service time constraints.
4. To achieve effective task scheduling through optimal task sorting and efficient resource allocation.

3 Proposed System Model

The working process of our proposed EC-scheduler for green cloud computing is jointly optimizing the energy consumption and communication traffic load. [Fig. 1](#) depicts the system model of the proposed EC-scheduler.

Customer service requests and product performance for every service are delivered to the data centre. A task that can be handled by a virtual machine in a data centre is referred to as a service request. Customers who have varying computing service resources and response times make service requests. Each data centre hosts Virtual machines on several real servers or computers. Assume $SD = \text{Server1, Server2, } \dots, \text{ Server}n$ is the server set and $VD = V1, V2, \dots, VM$ is the virtual machine (VM) set installed on the servers. The sorted groupings of requests are delivered to the tasks (scheduling units). Each request represents a task that a data centre VM may be able to perform. The central database unit stores the operational and structural data for all physical servers and VMs of the data centre. The memory capacity, computation speed, current utilization percentage, failure rate, power consumption rate, availability, etc., should all be included in this unit for each physical resource or VM. The resource allocation unit is in charge of computing underutilized servers that must be hibernated or slept, and over-utilized servers will be in charge of receiving requests and migrating the VMs with their requests. To perform this action, the resource allocation unit queries the data centre's central

database for the current server utilization. The server monitor unit provides the server information to the central database. The main task of the server monitor unit is to monitor the servers and submit periodic reports to the central database regarding their current state.

The working process of the EC-scheduler is as follows:

- Task sorting using the MLSS algorithm
- Resource allocation using EANN

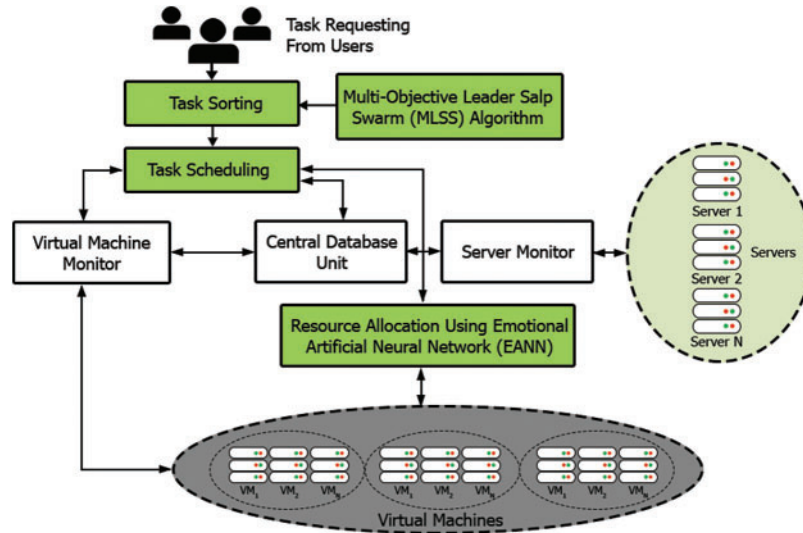


Figure 1: Proposed system model

3.1 Task Sorting

Salp swarms are transparent jellyfish-like organisms found in the sea. The evolutionary meta-heuristic Salp Swarm Algorithm (SSA), which has a Salp predation technique, has chain-like behaviour called group chain. The SSA uses the chain behaviour to obtain the best solution. There are two kinds of Salp swarms, where one is “leading” and the other is “following”. The leader is the Salp at the top of the chain, and the others are followers. To keep the chain flexible, the leader at the chain front helps followers search for food, and all food signals are sent out by the followers of the most recent Salp. Each Salp site in this study was programmed to look for food in an nD-dimensional search space, where D signifies the search dimension and n denotes the population size. In the MLSS algorithm, the j-th Salp position $y_{D(j=1,\dots,n),(D=1,\dots,d)}^j$ and D-th dimension are given by Eq. (1).

$$y_D^j = \begin{bmatrix} y_1^1 & y_2^1 & \cdot & y_d^1 \\ y_1^2 & y_2^2 & \cdot & y_d^2 \\ \cdot & \cdot & \cdot & \cdot \\ y_1^n & y_2^n & \cdot & y_d^n \end{bmatrix} \quad (1)$$

In the D-dimensional search space, the leader position $y_{D(D=1,2,\dots,d)}^j$ is assigned. The best answer in terms of functionality based on food supply is likewise the leader position, and it is configured to obtain food in the area of searching, which in turn is sought and followed by the chain of the Salp swarm. According to the location of the food supply, the leader adjusts its position according to Eq. (2).

$$y_D^1 = \begin{cases} f_D + C_1((ua_D - la_D)C_2 + la_D), & C_3 \geq P, \\ f_D - C_1((ua_D - la_D)C_2 + la_D), & C_3 < P, \end{cases} \tag{2}$$

where f_D represents the food position. y_D^1 represents the leader's position. The D-th dimension search space represents la_D the lower bound and ua_D the upper bound. The parameters C_3 , C_2 , P and P uniformly generate random numbers in the range $[0, 1]$. The expression coefficient is represented by the parameter, which can be written as

$$C_1 = 2E^{-(4s/S)} \tag{3}$$

where E denotes the natural base. s represents the current number of iterations, and S represents the maximum iteration number. During each seeking phase, each follower tracks the leader's position by following the other followers. Examples of possible follower positions are as follows:

$$y_D^j = \frac{1}{2} (y_D^j + y_D^{j-1}) \tag{4}$$

In the D-th dimension, $i \geq 2y_D^j y_D^{j-1}$ denotes the j-th follower's and neighbouring positions, respectively. The Perturbation Weight Salp Swarm Algorithm (PWSSA) is proposed in this paper to add a variable perturbation weight mechanism to the basic SSA to improve the search strategy and eliminate ignorance in the process. The distance between the population and the best solutions is changed by the technique of perturbation weight, and using asymptotic circular searching, the search domain is controlled, resulting in a superior leader searching strategy that is faster balanced. With more iterations, the positions of the followers will improve, and the adjusted position will alter more. The perturbation weight mechanism improves SSA's ability to find the best solution. Eqs. (5) and (6) show the updates of factors C_1 and C_2 .

$$C_{1new} = u_1 \cdot \left(1 - \frac{s}{S}\right) \tag{5}$$

$$C_{2new} = u_2 \cdot \left(1 - \frac{s}{S}\right) \tag{6}$$

where u_2 and u_1 fulfil the usual distribution standards.

$$u_2 \sim \text{num} [0, 1]$$

$$u_1 \sim \text{num} [0, 1]$$

The features of standard normal fractions have consistent variability in symmetry and concentration. When updating the new position for the leader using the equation, the follower position becomes like Eq. (7):

$$y_D^1 = \begin{cases} f_D + C_{1new}((f_d - y_D^1)C_{2new} + la_D), & C_3 \geq P, \\ f_D - C_{1new}((f_d - y_D^1)C_{2new} + la_D), & C_3 < P, \end{cases} \tag{7}$$

The multidirectional cross-searching technique was introduced to the basic SSA to increase the diversity of follower placements:

$$y_D^j = W_1 [(W_2 \cdot f_d - y_D^j) + (W_3 \cdot f_d - y_D^{j-1})] \tag{8}$$

where random parameters W_1 , W_2 , and W_3 are represented over the range of $[-1, 1]$.

Algorithm 1 presents the pseudocode of task sorting using the MLSS algorithm.

Algorithm 1: Task sorting using MLSS algorithm

Input: fitness function (f), D -dimension, parameters

Output: f^*

Step 1: Initialize the population randomly;

Step 2: The leader position is denoted by

$$y_D^1 = \begin{cases} f_D + C_1((ua_D - la_D)C_2 + la_D), & C_3 \geq P, \\ f_D - C_1((ua_D - la_D)C_2 + la_D), & C_3 < P, \end{cases}$$

Step 3: If $i=0$ do

Step 4: Coefficient represented as $C_1 = 2E^{-(4s/S)}$

Step 5: Each follower position is defined as $y_D^j = \frac{1}{2}(y_D^j + y_D^{j-1})$

Step 6: New factors $c1$ and $c2$ updated

$$C_{1new} = u_1 \cdot \left(1 - \frac{S}{S}\right)$$

$$C_{2new} = u_2 \cdot \left(1 - \frac{S}{S}\right)$$

Step 7: End if

Step 8: The new leader position is updated as

$$y_D^1 = \begin{cases} f_D + C_{1new}((f_d - y_D^1)C_{2new} + la_D), & C_3 \geq P, \\ f_D - C_{1new}((f_d - y_D^1)C_{2new} + la_D), & C_3 < P, \end{cases}$$

Step 9: Find the best solution

3.2 Resource Allocation

It is recommended that tasks should be allocated to resources in a queue fashion using grasshopper sorting as well as heuristic algorithms, thereby choosing the most suitable ways to perform every job depending on the significant factor considered by the service provider or end-user to obtain optimal resources in a timely and cost-effective manner. The process of resource allocation is performed by EANN for efficient resource allocation based on user requirements, which jointly optimizes both energy consumption- and communication-related delay and packet losses for QoS.

Despite the capacity of ANNs to model decisions, certain things may need improvement if the supplied time series for ANN training needs to be more sampled with seasonal changes. Underestimation and overtraining of peak values are the flaws. To overcome these problems, several data preprocessing techniques have been proposed. In most areas of hydrology, multiresolution analysis capacity with wavelet-based data processing techniques is linked to ANNs to improve modelling efficiency. However, emotions interact dynamically in AI systems, and others follow suit by building EANN models that include artificial emotions in the ANN. From a biological standpoint, an animal's mood and emotion, resulting from hormone gland activity, can influence its neurophysiological reaction, sometimes by delivering various behaviours for the same job from different perspectives. In an EANN, a feedback loop exists between the neurological and hormonal systems, which enhances the training capabilities of the network. The explicit equation for determining the EANN output value is derived as

$$\hat{x}_i = F_i \left[\sum_{h=1}^M W_{ih} \times F_g \left(\sum_{k=1}^N W_{ph} y_k + W_{ha} \right) + W_{ia} \right] \quad (9)$$

where the input, hidden, and output layers of neurons and bias are represented by p , h , I , and a , respectively. W denotes the weight applied to the neuron, F_k means the activation function for the

output, F_h represents the hidden layers, N denotes the number of hidden and input neurons, M denotes the input layer variable, and x denotes the computed output neuron values.

The EANN model is a more advanced version of the traditional ANN as it involves a sentimental system that creates artificial hormones to affect the function of every neuron in a feedback mechanism, with hormonal variables being influenced more by the inputs and outputs of neurons.

When the Feed Forward Neural Network (FFNN) and EANN are compared, it can be shown that, unlike the FFNN, an EANN neuron may reversibly receive information through inputs and outputs and provide hormones. These hormones are set up as dynamic coefficient characteristic patterns of input (and target) samples and then tweaked over time. Throughout the training process, they may alter all components of the neuron. The outputs of the p^{th} neuron in an EANN with three hormone glands are expressed as follows:

$$x_i = \lambda_p + \sum_h \sigma_{ph} G_h \times F \left(\sum_p \beta_k + \sum_h \zeta_{k,h} G_h + \left(\sum_h \chi_{p,h} G_h + \alpha_k \right) + \left(\sum_h \phi_{p,i,K} G_h \right) Y_{p,i} \theta_{p,i} + \theta_{p,i} \right) \quad (10)$$

where the EANN's total hormone value is calculated as

$$G_h = \sum G_{p,h} \quad (h = a, b, c) \quad (11)$$

$$G_{p,h} = \overset{p}{glandity}_{p,h} \times X_p \quad (12)$$

Each gland's hormone level is used as a calibration parameter. Various schemes have been employed to initialize the value of each hormone (G_h) based on the input pattern, such as the mean of each sample's input vector (input parameter values). In every period of the EmBP training phase, the output neuron (Δ) containing the error signal is communicated back to change the conventional weights of the hidden layer (W_{ih}) and bias (W_{ia}) as required.

$$W_{ih}(New) = W_{ih}(Old) + \eta \cdot \Delta \cdot XG_h + \alpha W_{ih}(Old) \quad (13)$$

$$W_{ia}(New) = W_{ia}(Old) + \eta \cdot \Delta + \alpha \cdot [\delta W_{ia}(Old)] \quad (14)$$

where the result of the h^{th} hidden neuron is YG_h , the last weight is $\alpha W_{ih}(Old)$, and the bias-value alternation is $\alpha W_{ia}(Old)$.

In addition, the emotional weight (W_{im}) has been changed to

$$W_{im}(New) = W_{im}(Old) + \mu \cdot \Delta \cdot X_{avg} + K \cdot [\delta W_{im}(Old)] \quad (15)$$

where Y_{avg} is the emotional weight's prior variation and $\alpha W_{im}(Old)$ provides the network subjected to the mean value of the input patterns at every period. Anxiety and confidence factors are also identified.

$$\mu = X_{avg} + \Delta \quad (16)$$

$$K = \mu_0 - \mu \quad (17)$$

where μ_0 represents the level of concern upon completion of the first repetition. The weights and bias from the hidden layer to the input layer are adjusted likewise. It is important to note that networks are typically enforced with normalized data. The following efficiency criteria were used to evaluate the model performance in this study:

$$dc = 1 - \frac{\sum_{p=1}^n (Y_p - \hat{x}_p)^2}{\sum_{p=1}^n (Y_p - \bar{Y}_p)^2} \quad (18)$$

$$rmse = \sqrt{\frac{1}{n} \sum_{p=1}^n (Y_p - \hat{x}_p)^2} \quad (19)$$

where Y_p , \bar{Y}_p , n , and \hat{x}_p are the observed data, number of observations, mean of the observed data, and computed values, respectively. The variable dc denotes the determination coefficient, and $rmse$ indicates the root-mean-square error. Because extreme values are essential in rainfall-runoff modelling, Eq. (20) represents the model performance test to recognize the maximum values of the runoff time series:

$$dc_{peak} = 1 - \frac{\sum_{p=1}^{n_q} (Y_{qc_p} - Y_{qo_p})^2}{\sum_{p=1}^{n_q} (Y_{qo_p} - \bar{Y}_{qo})^2} \quad (20)$$

Algorithm 2 presents the pseudocode of resource allocation using the EANN algorithm.

Algorithm 2: Resource allocation using EANN

Input: Task requests, sorted requests, control metrics

Output: Allocate resources ($X(\hat{x}_p)$)

Step 1: Initialize the best population

Step 2: Output value of FFNN

$$\hat{x}_i = F_i \left[\sum_{h=1}^M W_{ih} \times F_h \left(\sum_{p=1}^N W_{ph} Y_p + W_{ha} \right) + W_{ia} \right]$$

Step 3: While doing $G_h = \sum_p G_{p,h}$

Step 4: Compute overall hormone value

Step 5: Emotional weight (W_{im}) is updated as

$$W_{im}(\text{New}) = W_{im}(\text{Old}) + \mu \cdot \Delta \cdot X_{avg} + K \cdot [\delta W_{im}(\text{Old})]$$

Step 6: Anxiety and confidence factors $\mu = X_{avg} + \Delta$

Step 7: Generate a new solution by flying randomly

Step 8: Compute Heidke Skill Score

$$hss = \frac{\frac{1}{n} \sum_{p=1}^a X(\hat{x}_p, Y_p) - \frac{1}{n^2} \sum_{p=1}^a X(Y_p) \times X(\hat{x}_p)}{1 - \frac{1}{n^2} \sum_{p=1}^a X(Y_p) \times X(\hat{x}_p)}$$

Step 9: End

The Multicategory Heidke Skill Score (HSS) for evaluating and comparing the forecasting model performance in multiple flow categories, such as low- and high-flow regimes, is given by

$$HSS = \frac{\frac{1}{n} \sum_{p=1}^a X(\hat{x}_p, Y_p) - \frac{1}{n^2} \sum_{p=1}^a X(\hat{x}_p) \times X(Y_p)}{1 - \frac{1}{n^2} \sum_{p=1}^a X(\hat{x}_p) \times X(Y_p)} \quad (21)$$

Both forecasts and observations were included in the HSS computation. The dataset (37) intervals are divided into groups. The number of estimates in Category K with the total number of predictions and observations in Category J are counted and used in Eq. (21). HSS calculates the percentage of correct projections that would be accurate based on pure randomness after eliminating those forecasts. Algorithm 2 presents the pseudocode of the resource allocation process using EANN.

4 Results and Discussion

In this section, the performance of the proposed EC-scheduler for green cloud computing is evaluated and validated. The EC-scheduler was tested using the GreenCloud simulator and compared with state-of-the-art scheduling techniques, such as energy-aware scheduler (E-aware), Energy Trade-off Multi-Resource Cloud Task Scheduling Algorithm (ETMCTSA), Proactive and Reactive Scheduling (PRS), time-critical (TC), Time NonCritical (TNC), Enhanced Conscious Task Consolidation (ECTC), Energy-Efficient Hybrid (EEH), Best Heuristic Scheduling (BHS), and Multi-Heuristic Resource Allocation algorithm (MHRA). The experiments were carried out in different cases, such as i. testing scheduling algorithm's parameters against the number of user applications, ii. testing scheduling parameters against the number of user requests, and iii. comparison of essential metrics of EC-scheduler with those of state-of-the-art schedulers.

4.1 Performance Measures

Different parameters, such as Throughput (TP), Data Centre Energy Productivity (DCEP), Power Usage Effectiveness (PUE), Average Execution Time (AET), Energy Consumption, and Makespan, were used to validate the performance of our proposed EC-scheduler. The simulation parameters were as follows: PUE is a metric for determining the effectiveness of a data centre's power consumption. Eq. (22) defines PUE. Eq. (23) defines DCEP.

$$PUE = \frac{DC_{TP}}{IT - DC_{TP}} \quad (22)$$

$$DCEP = \frac{DCW_t}{DCE_t} \quad (23)$$

The variables represent the total work done as W_t in the data centre DC during time t . The quantity of electrical energy consumed during this period is E_t . Customers care about the average execution time (AET). Customers wish to handle requests within the shortest possible time. The following formula was used to calculate the AET:

$$AET = \frac{\sum_{i=1}^N \frac{L_{T_i}}{S_{speed}}}{n} \quad (24)$$

where L_T is the request length, S_{speed} is the switch speed, and n is the number of requests. TP refers to the number of requests used for a data centre at a particular time as follows:

$$TP = \frac{DCQ_t}{t} \quad (25)$$

where Q_t denotes the request amount for data centre DC at time t .

4.2 Comparative Analysis

In this subsection, the proposed EC-scheduler is evaluated and compared with existing schedulers using different scenarios, such as the impact of user applications, user requests, and vital factors.

4.2.1 User Application Impacts

The authors performed this test to evaluate the EC-scheduler performance concerning the number of user applications and compared it with state-of-the-art scheduler performances through metrics. In this test scenario, between 100 and 1000 user applications are generated consistently, with user time requirements ranging from 10 to 1000 h. In terms of the impact of user applications, [Table 2](#) presents the metrics of our proposed EC-scheduler and existing state-of-the-art schedulers TC, TNC, and E-aware [37]. The parameters evaluated were PUE, DCET, AET, and TP for different numbers of user application executions. When the number of applications increased, the PUE value for the existing schedulers increased. Compared to the TC, TNC, and E-aware schedulers, the proposed EC-scheduler's PUE, DCEP, and AET were reduced. Thus, execution time is reduced in our proposed system. According to the TP of EC-scheduler, it returns high values compared to existing schedulers for different numbers of user applications varying from 200 to 1000.

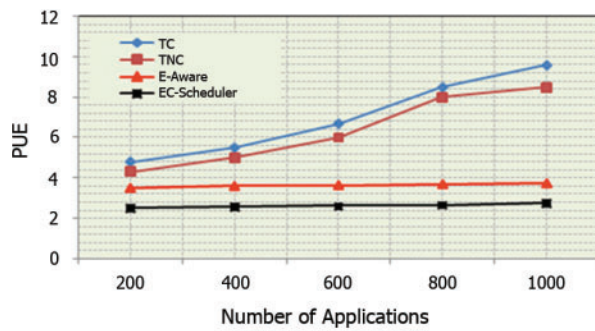
Table 2: Comparative analysis concerning user applications

No. of applications	PUE				DCEP			
	TC	TNC	E-aware	EC-scheduler	TC	TNC	E-aware	EC-scheduler
200	4.8	4.3	3.5	2.5	48	43	35	29
400	5.5	5	3.6	2.55	49	44	36	31
600	6.7	6	3.65	2.6	52	46	37	32
800	8.5	8	3.7	2.65	54	48	38	34
1000	9.6	8.5	3.75	2.75	55	49	39	35

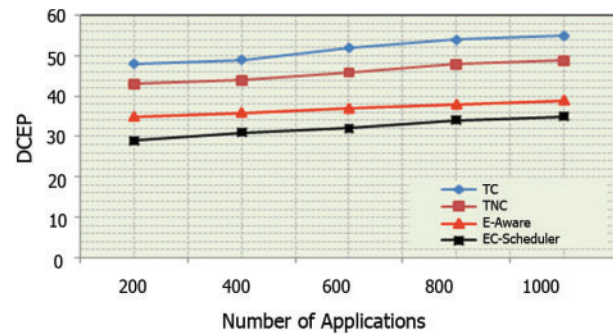
	AET (mins)				TP (Requests/s)			
	TC	TNC	E-aware	EC-scheduler	TC	TNC	E-aware	EC-scheduler
200	1.5	1.2	0.8	0.4	5	6	7	8
400	2.2	1.5	1.2	0.8	4	5	6	7
600	2.6	2.2	1.5	1.2	3	4	5	6
800	2.9	2.6	2.2	1.5	2	3	4	5
1000	3.3	2.9	2.6	2.2	1	2	3	4

[Fig. 2a](#) shows that the PUE values of our proposed EC-scheduler are 62.821%, 58.962%, and 28.297% more efficient than the existing TC, TNC, and E-aware schedulers, respectively, as shown by the black curve, which is lower than the other coloured curves; hence, the PUE consumption is low. [Fig. 2b](#) shows the DCEP values of the proposed and existing state-of-the-art schedulers. This implies that as the number of applications increases, the DCEP value increases for all schedulers (indicated by the coloured curves). The DCEP value of our proposed EC-scheduler (black curve) is 37.597%, 30.000%, and 12.973% more efficient than the existing TC, TNC, and E-aware schedulers, respectively, indicating that the energy consumption of the data centre is lower than that using the other

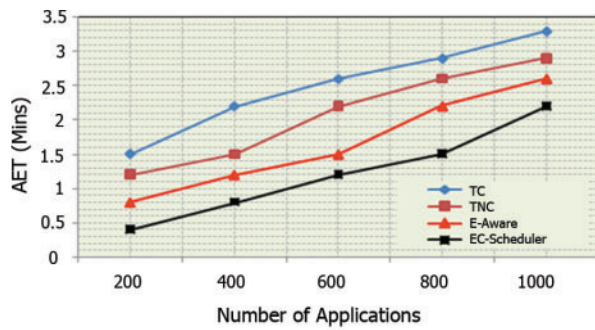
schedulers. Fig. 2c shows the AET values of our proposed and existing state-of-the-art schedulers. When the number of applications grows, the value of AET increases for all schedulers. The black curve represents the EC-scheduler, which is lower than the coloured curves; hence, the EC-scheduler requires less execution time. The AET value of our proposed EC-scheduler is 51.2%, 41.346%, and 26.506% more efficient than those of the existing TC, TNC, and E-aware schedulers, respectively. Fig. 2d shows the Throughput of our proposed and existing state-of-the-art schedulers, and the black curve, which is above the other coloured curves, indicates that EC-scheduler's Throughput is higher than that of all other schedulers. This shows that as the number of applications increased, the Throughput for both schedulers decreased. However, when employing the TC, TNC, and E-aware schedulers, the optimization rate for the EC-scheduler was faster. The throughput value of our proposed EC-scheduler is 50%, 33.333%, and 16.667% more efficient than those of the existing TC, TNC, and E-aware schedulers, respectively.



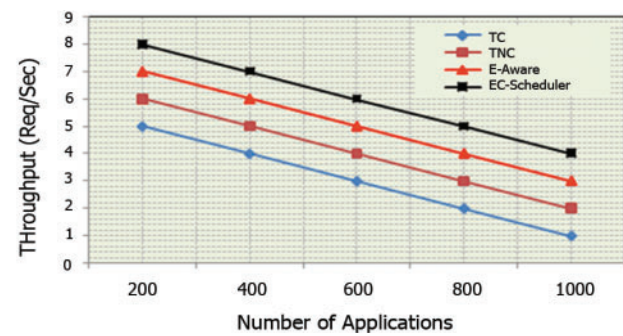
(a) Comparative analysis of PUE with user applications



(b) Comparative analysis of DCEP with user applications



(c) Comparative analysis of AET with user applications



(d) Comparative analysis of Throughput with user applications

Figure 2: EC-scheduler comparative analysis

4.2.2 Impact of Requests

The authors performed this test to evaluate the EC-scheduler performance for the number of user requests and compared it with state-of-the-art scheduler performances through metrics. The number of user requests in this test scenario ranges from 1000 to 5000, with the request times ranging from 10 to 1000 h. Table 3 presents a comparative analysis of the proposed EC-scheduler and existing state-of-the-art PRS, ECTC, ETMCTSA, and EEH schedulers [38] with user request impacts.

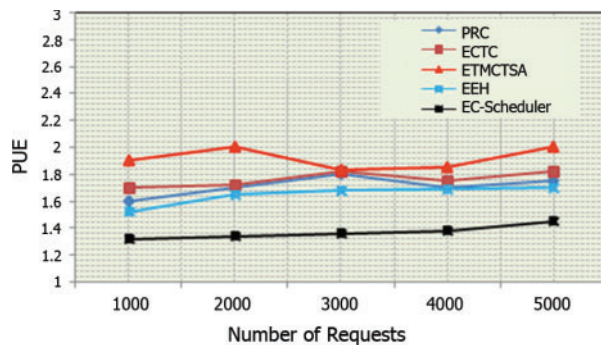
Table 3: Comparative analysis concerning user requests

No. of applications	PUE				
	PRS	ECTC	ETMCTSA	EEH	EC-scheduler
1000	1.6	1.7	1.9	1.52	1.32
–2000	1.7	1.72	2	1.65	1.34
3000	1.8	1.82	1.7	1.68	1.36
4000	1.7	1.75	1.75	1.69	1.38
5000	1.75	1.82	2	1.7	1.45
	DCEP				
	PRS	ECTC	ETMCTSA	EEH	EC-scheduler
1000	310	300	250	330	360
2000	300	300	240	310	345
3000	280	275	230	280	325
4000	270	260	220	270	310
5000	260	240	200	260	295
	AET (mins)				
	PRS	ECTC	ETMCTSA	EEH	EC-scheduler
1000	0.3	0.35	0.5	0.3	0.1
2000	0.6	0.65	0.8	0.6	0.15
3000	0.95	1.25	1.35	0.8	0.2
4000	1.3	1.35	1.75	1.2	0.25
5000	1.48	1.5	2	1.5	0.3
	TP (Requests/s)				
	PRS	ECTC	ETMCTSA	EEH	EC-scheduler
1000	5.5	3.8	3.7	6.5	10
2000	8	6.5	5	10	13.5
3000	11	6.8	6.5	13.2	14.56
4000	10	9	6.8	13.5	15.89
5000	9	9.3	7.5	14	16.23

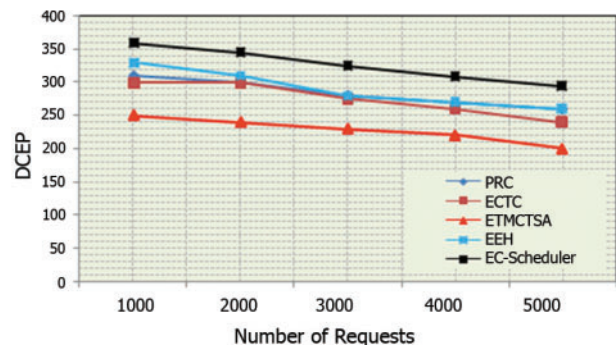
The PUE value for the EC-scheduler is significantly less than that of the existing PRS, ECTC, ETMCTSA, and EEH; hence, the EC-scheduler utilizes much less server energy. The value of DCEP of the EC-scheduler is more than that of PRS, ECTC, ETMCTSA, and EEH, which implies that the EC-scheduler maximizes Data Centre Energy production. AET for EC-scheduler shows lower values than PRS, ECTC, ETMCTSA, and EEH, meaning that EC-scheduler takes minimal execution time. The Throughput of the EC-scheduler is also high compared to PRS, ECTC, ETMCTSA, and EEH,

which implies that the number of instructions executed by our scheduler is more than that of others for the user requests ranging from 1000 to 5000.

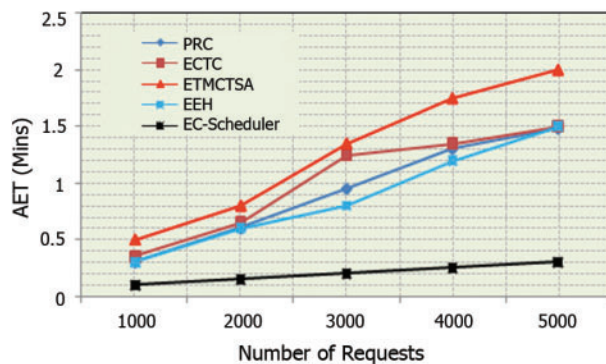
Fig. 3a shows the PUE values of the proposed and existing state-of-the-art schedulers. It shows that when the number of user requests increases, the value of PUE for all schedulers increases. The increase in the rate for the proposed EC-scheduler is more significant than those of the PRS, ECTC, ETMCTSA, and EEH schedulers. The PUE value of the proposed EC-scheduler is 19.883%, 22.247%, 26.738%, and 16.869% more efficient than those of the existing PRS, ECTC, ETMCTSA, and EEH schedulers, respectively. The DCEP of the proposed scheduler and existing state-of-the-art schedulers is shown in Fig. 3b. The figure shows that the DCEP value increases for all schedulers when the number of user requests grows. The increase in the rate for the proposed EC-scheduler is more significant than that for the PRS, ECTC, ETMCTSA, and EEH schedulers. The DCEP value of the proposed EC-scheduler is 13.150%, 15.902%, 30.275%, and 11.315% more efficient than those of the existing PRS, ECTC, ETMCTSA, and EEH schedulers, respectively.



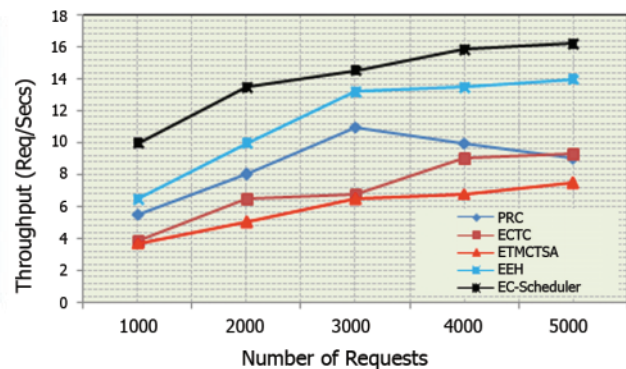
(a) Comparative analysis of PUE with user requests



(b) Comparative analysis of DCEP with user requests



(c) Comparative analysis of AET with user requests



(d) Comparative analysis of Throughput with user requests

Figure 3: EC-scheduler comparative analysis

Fig. 3c shows the average execution time of the proposed and existing state-of-the-art schedulers as the user request number increases and the average execution time for all schedulers increases. The growth rate for the proposed EC-scheduler is more significant than those of the PRS, ECTC, ETMCTSA, and EEH schedulers. The average execution time of our proposed EC-scheduler is

78.402%, 80.392%, 84.375%, and 77.273% more efficient than those of the existing PRS, ECTC, ETMCTSA, and EEH schedulers, respectively. Fig. 3d shows the Throughput of the proposed and existing state-of-the-art schedulers. As the number of user requests increases, the Throughput of all schedulers increases. The increase in the rate for the proposed EC-scheduler is more prominent than those of the PRS, ECTC, ETMCTSA, and EEH schedulers. Our proposed EC-scheduler's Throughput is 38.017%, 49.558%, 57.965%, and 18.495% more efficient than those of the existing PRS, ECTC, ETMCTSA, and EEH schedulers, respectively.

4.2.3 Comparative Analysis for Important Metrics

In this section, we increased the significance factor (α) from 0.0 to 1.0 in increments of 0.1 for 1000 tasks. According to the results shown in Fig. 4a, performing this work is more appropriate than other approaches and produces more optimal outcomes. The graph shows that our EC-scheduler is 34.2% and 19.73% more energy efficient than the current state-of-the-art BHS and MHRA schedulers, respectively.

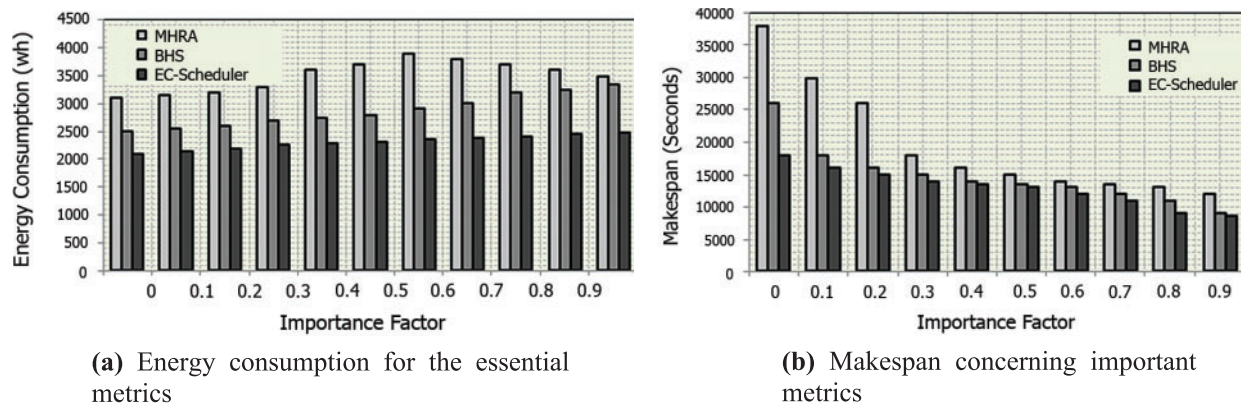


Figure 4: Energy consumption and makespan performance by EC-scheduler

The time spans of the proposed and existing scheduling strategies are shown in Fig. 4b. From the figure, the average energy utilization of our proposed EC-scheduler clearly shows 33.549% and 12.0378% improved efficiency compared with the existing state-of-the-art BHS and MHRA schedulers, respectively.

5 Conclusion

For green cloud computing, we proposed the EC-scheduler, which optimizes data centre utilization of energy and traffic load. In EC-scheduler, an MLSS algorithm is used for task sorting, which ensures traffic load balancing. Then, an EANN is utilized for resource allocation for cloud user requirements, which jointly optimizes energy consumption and communication costs. Our proposed EC-scheduler was implemented with the GreenCloud simulator, and the simulation outcome proved that it is efficient in terms of increased TP, DCEP, PUE, and AET and less decreased energy consumption compared to other scheduling techniques that focus on the path of green environment sustainability, although user load increases in cloud computing systems.

Acknowledgement: Dr. Laila Almutairi would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project Number R-2023-652.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design, analysis and interpretation of results are made by L. Almutairi; Draft manuscript preparation, Algorithm implementations are made by S. M. Aslam. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We used the Google cluster-usage trace dataset to generate synthetic data.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] C. Kessler and J. Keller, "Models for parallel computing: Review and perspectives. Mitteilungs-gesellschaft für informatik eV," *Parallel-Algorithmen und Rechnerstrukturen*, vol. 24, pp. 13–29, 2007.
- [2] B. Jacob, M. Brown, K. Fukui and N. Trivedi, "Introduction to grid computing," in *IBM Redbooks*, 1st ed., US: IBM Corp., pp. 3–6, 2005.
- [3] R. P. Franca., Y. Iano, A. C. B. Monteiro and R. Arthur, "Better transmission of information focused on green computing through data transmission channels in cloud environments with Rayleigh fading," in *Green Computing in Smart Cities: Simulation and Techniques*, 1st ed., pp. 71–93, Nature Switzerland AG: Springer, 2021.
- [4] X. Deng, D. Wu, J. Shen and J. He, "Eco-aware online power management and load scheduling for green cloud datacenters," *IEEE Systems Journal*, vol. 10, no. 1, pp. 78–87, 2014.
- [5] D. Cheng, J. Rao, C. Jiang and X. Zhou, "Elastic power-aware resource provisioning of heterogeneous workloads in self-sustainable data centres," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 508–521, 2015.
- [6] Z. Zhou, F. Liu, R. Zou, J. Liu, H. Xu *et al.*, "Carbon-aware online control of geo-distributed cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2506–2519, 2015.
- [7] Z. Asad, M. A. R. Chaudhry and D. Malone, "Greener data exchange in the cloud: A coding-based optimization for big data processing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1360–1377, 2016.
- [8] P. Cao, W. Liu, J. S. Thompson, C. Yang and E. A. Jorswieck, "Semidynamic green resource management in downlink heterogeneous networks by group sparse power control," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1250–1266, 2016.
- [9] Y. Shi, J. Cheng, J. Zhang, B. Bai, W. Chen *et al.*, "Smoothed L_p -minimization for green cloud-RAN with user admission control," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 1022–1036, 2016.
- [10] Z. Zhou, F. Liu and Z. Li, "Bilateral electricity trade between smart grids and green datacenters: Pricing models and performance evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3993–4007, 2016.
- [11] K. Guo, M. Sheng, J. Tang, T. Q. S. Quek and Z. Qiu, "Exploiting hybrid clustering and computation provisioning for green C-RAN," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 4063–4076, 2016.
- [12] Y. Yang, X. Chang, J. Liu and L. Li, "Towards robust green virtual cloud data centre provisioning," *IEEE Transactions on Cloud Computing*, vol. 5, no. 2, pp. 168–181, 2015.
- [13] Q. Fan, N. Ansari and X. Sun, "Energy driven avatar migration in green cloudlet networks," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1601–1604, 2017.
- [14] Y. Wu, M. Tornatore, S. Ferdousi and B. Mukherjee, "Green data centre placement in optical cloud networks," *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 3, pp. 347–357, 2017.

- [15] G. Portaluri, D. Adami, A. Gabbrielli, S. Giordano and M. Pagano, "Power consumption-aware virtual machine placement in the cloud data centre," *IEEE Transactions on Green Communications and Networking*, vol. 1, no. 4, pp. 541–550, 2017.
- [16] I. F. Siddiqui, S. U. J. Lee, A. Abbas and A. K. Bashir, "Optimizing lifespan and energy consumption by smart meters in green-cloud-based smart grids," *IEEE Access*, vol. 5, pp. 20934–20945, 2017.
- [17] Qiu M., Ming Z., Li J., Gai K. and Zong Z., "Phase-change memory optimization for the green cloud with a genetic algorithm," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3528–3540, 2015.
- [18] H. Yuan, J. Bi, M. Zhou and A. C. Ammari, "Time-aware multi-application task scheduling with guaranteed delay constraints in the green data centre," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1138–1151, 2017.
- [19] H. Yuan, J. Bi and M. Zhou, "Spatial task scheduling for cost minimization in distributed green cloud data centres," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 729–740, 2018.
- [20] D. Zeng, J. Zhang, L. Gu, S. Guo and J. Luo, "Energy-efficient coordinated multipoint scheduling in green cloud radio access network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9922–9930, 2018.
- [21] L. Ismail and H. Materwala, "Energy-aware VM placement and task scheduling in cloud-IoT computing: Classification and performance evaluation," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5166–5176, 2018.
- [22] W. Chen, D. Wang and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726–738, 2018.
- [23] M. A. Iqbal, M. Aleem, M. Ibrahim, S. Anwar and M. A. Islam, "Amazon cloud computing platform EC2 and VANET simulations," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 30, no. 3, pp. 127–36, 2019.
- [24] P. Kurp, "Green computing," *Communications of the ACM*, vol. 51, no. 10, pp. 11–13, 2008.
- [25] R. Harmon, H. Demirkan, N. Auseklis and M. Reinoso, "From green computing to sustainable IT: Developing a sustainable service orientation," in *Proc. of 2010 43rd Hawaii Int. Conf. on System Sciences*, Honolulu, HI, USA, IEEE, pp. 1–10, 2010.
- [26] Lu Y. and Sun N., "An effective task scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment," *Cluster Computing*, vol. 22, no. 1, pp. 513–520, 2019.
- [27] L. Mao, Y. Li, G. Peng, X. Xu and W. Lin, "A multi-resource task scheduling algorithm for energy-performance trade-offs in green clouds," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 233–241, 2018.
- [28] H. Yuan, J. Bi and M. Zhou, "Spatiotemporal task scheduling for heterogeneous delay-tolerant applications in distributed green data centers," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1686–1697, 2019.
- [29] H. Yuan, J. Bi and M. Zhou, "Profit-sensitive spatial scheduling of multi-application tasks in distributed green clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1097–1106, 2019.
- [30] A. Mohammadzadeh, M. Masdari, F. S. Gharehchopogh and A. Jafarian, "Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing," *Evolutionary Intelligence*, vol. 14, no. 4, pp. 1997–2025, 2021.
- [31] S. Li, H. Liu, B. Gong and J. Wang, "An algorithm incarnating deep integration of hardware-software energy regulation principles for heterogeneous green scheduling," *IEEE Access*, vol. 8, pp. 111494–111503, 2020.
- [32] H. Yuan, M. Zhou, Q. Liu and A. Abusorrah, "Fine-grained resource provisioning and task scheduling for heterogeneous applications in distributed green clouds," *EEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1380–1393, 2020.

- [33] N. Gholipour, E. Arianyan and R. Buyya, "A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centres," *Simulation Modelling Practice and Theory*, vol. 104, pp. 102127, 2020.
- [34] P. Geetha and C. R. Robin, "Power conserving resource allocation scheme with improved QoS to promote green cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7153–7164, 2021.
- [35] N. K. Biswas, S. Banerjee, U. Biswas and U. Ghosh, "An approach towards the development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing," *Sustainable Energy Technologies and Assessments*, vol. 45, pp. 101087, 2021.
- [36] Z. Peng, B. Barzegar, M. Yarahmadi, H. Motameni and P. Pirouzmand, "Energy-aware scheduling of workflow using a heuristic method on green cloud," *Scientific Programming*, vol. 2020, pp. 8898059, 2020.
- [37] M. Amoon, "A green energy-efficient scheduler for cloud data centres," *Cluster Computing*, vol. 22, no. 2, pp. 3247–3259, 2019.
- [38] A. Alarifi, K. Dubey, M. Amoon, T. Altameem, F. E. Abd El-Samie *et al.*, "Energy-efficient hybrid framework for green cloud computing," *IEEE Access*, vol. 8, pp. 115356–115369, 2020.