Baltic J. Modern Computing, Vol. 12 (2024), No.3, 304-326
https://doi.org/10.22364/bjmc.2024.12.3.06

EnE-Rep: An Energy-Efficient Data Replication Strategy for Clouds

Mohammed ALGHOBIRI

Business Informatics Department
King Khalid University, Abha 62585 Saudi Arabia

maalghobiri@kku.edu.sa

ORCID 0000-0002-6414-739X

Abstract. The rapid rise in the popularity of cloud computing can be attributed to its inherent advantages. However, its expanding infrastructure leads to higher energy consumption and increased network latency. Virtual machine consolidation (VMC) and dynamic power management (DPM) are popular methods to improve energy efficiency. However, these energy-saving approaches are incompatible with data replication. Our approach in this study is called EnE-Rep, that categorizes cloud data center nodes based on workload and applies targeted strategies for each category. In addition, EnE-Rep leverages a robust collection of components including a load manager, energy monitor, and replicator for achieving energy-efficient data replication. Furthermore, intelligent placement decisions are made based on key factors like CPU utilization, server proximity, available bandwidth, and memory usage. Finally, CloudSim simulations validate the effectiveness of EnE-Rep, demonstrating significant reductions in energy consumption alongside improved performance metrics such as VM migration frequency, host shutdown rate, and data access time.

Keywords: Carbon Emission, Cloud Computing, Energy Efficiency, Data replication, Data center, Data-Intensive Computing.

1. Introduction

The rapid proliferation of cloud computing within the contemporary technological landscape can be attributed to its inherent advantages such as its ability to leverage a pool of shared resources that are readily scalable to meet user demands (Balakrishnan et al., 2017), Ruan et al., 2013). Cloud networks function by aggregating heterogeneous computing nodes from diverse locations. These nodes are then dynamically provisioned to users on an as-needed basis, offering a flexible and cost-effective solution. Service Level Agreements (SLAs) further govern the specifics of these cloud services, outlining performance guarantees and resource allocation. This 'on-demand scalability', a key feature of cloud computing, allows users to dynamically adjust resource utilization based on their evolving needs (Ding et al., 2015). Consequently, the scalability feature is further bolstered by the "pay-as-you-go" pricing model, a revolutionary aspect of cloud computing (Buyya, 2009). By eliminating the need for upfront hardware and software investments, cloud computing empowers businesses to streamline operations and dedicate resources to core competencies (Beloglazov et al., 2012). Furthermore, economies of scale achieved through shared infrastructure contribute to the significant cost-effectiveness that drives widespread adoption of cloud computing solutions.

According to a report by CISCO, a staggering 94% of the total workload was processed by cloud computing in 2021. This widespread adoption can be attributed primarily to the ability of cloud infrastructure to provide access from anywhere in the world. Consequently, by 2020, a significant portion, 67%, of enterprise infrastructure had shifted to the cloud. Furthermore, cloud computing spending has grown at a remarkable pace, outpacing overall IT spending by a factor of six between 2015 and 2020. Currently, more than half of all IT spending goes towards cloud computing solutions (Kappelman et al., 2022). However, while the dynamism and flexibility of the cloud have undoubtedly fueled its growth, these features also present challenges, particularly regarding resource management, scheduling, and energy consumption (Jennings and Stadler, 2015), Ksentini et al., 2014). By 2020, cloud data center (DC) energy consumption was projected to reach an alarming 140B KW/H annually, equivalent to the energy produced by approximately 50 power plants. The financial and environmental costs associated with this immense energy consumption are significant. Quantifying this impact, the annual financial cost has reached $13 billion whereas the environmental cost translates to 100 million metric tons of CO2 emissions (Mytton, 2020). This high energy consumption is further reflected in global data center usage, which accounts for an estimated 205-Terawatt hours of power consumption per year which represents a significant 1% of the world's total energy consumption (Bonzi, 2021). The environmental impact is further emphasized by the fact that in 2018, data centers were responsible for a substantial 900 billion kilograms of carbon emissions, releasing approximately 4.4 kilograms of CO2 every hour (Bonzi, 2021).

Beyond scalability, another key challenge for cloud computing is efficient resource management, which directly impacts cost-effectiveness. In United States alone, estimates suggest that there are nearly three million data centers (DCs) accommodating approximately 12 million servers (Jahangir et al., 2021). However, it is worth noting that up to 30% of these servers are deemed unnecessary, with many others being underutilized. Despite this redundancy, the collective power consumption of these servers amounts to a substantial 140 billion KW/h annually, contributing to an alarming 150 million metric tons of carbon emissions per year. Studies reveal that roughly 15-30% of data center equipment consumes energy while idle. In fact, server utilization rates typically hover between a meager 5% and 15%, even though they continue to draw full power (Jahangir et al., 2021). This underscores a critical inefficiency, indicating that server utilization in data centers falls significantly short of optimal levels. To address the challenge of rising energy consumption, cloud computing leverages techniques like dynamic power management (DPM) and virtual machine (VM) consolidation. VM consolidation involves migrating workloads from underutilized systems to others, allowing idle servers to be powered off. This approach has demonstrably reduced peak power consumption of servers during idle states – from 50% to 20% over the past decade (Pierson and Hlavacs, 2015).

In addition, the rapid growth of the internet presents a significant challenge to achieving the goal of green computing. This rapid expansion, characterized by an increase in the number of users, devices and data results in an unconventional source of increased energy consumption. According to a report by CISCO, global internet users are estimated to reach 5.3 billion by 2023, representing two-thirds of the world's population (Zhang et al., 2019). Furthermore, per capita device ownership is expected to reach 3.6 devices, with a total of 29.3 billion devices by 2023. Recent developments in the Internet of Things (IoT) are further accelerating the growth of the internet, with an estimated 14.7 billion smart devices, connected for communication, are expected to be operational by 2023 (Zhang et al., 2019). Similarly, the rapid growth in internet activity leads to sharp increase in data volume. For instance, approximately 300 million mobile applications have been downloaded by 2020 alone (Zhang et al., 2019). This exponential increase in data generation inflates the data volume that is projected to reach a staggering 150 zettabytes by 2024 (Kireev et al., 2019). On the other hand, in 2020, individual data generation reached an estimated 1.7 megabytes per second and 2.5 quintillion bytes per day. Underscoring the rapid growth, it is estimated that 90% of the world's data has been produced in just the last two years (Roser, 2022). These rising trends contribute to a growing data storage demand in the data centers (DCs), resultantly, storage alone accounts for 11% of total DC power consumption (Jahangir et al., 2021). Furthermore, the common practice of storing multiple copies of the same data within DCs significantly expands the data volume (Jahangir et al., 2021).

Therefore, the growing volume of big data and the challenge of data latency requires the use of well-established mechanisms such as data replication. In cloud environments, data replication plays an important role in achieving reliability and fault tolerance that ensures adherence to Service Level Agreements (SLAs). This process involves copying essential data closer to the client, minimizing the distance data must travel, and reducing latency. Data replication follows a three-phase process: staging, placing, and moving. However, a significant drawback of data replication is that once a specific node is activated, it cannot be deactivated, even when the node is idle. The key reason of the drawback lies in continuous operation stemming from the node's responsibility to maintain data availability causing a conflict with the conventional energy-saving techniques like VM consolidation and dynamic power management. Additionally, the increasing frequency of data replication results in a higher number of idle nodes hosting replicated data, leading to substantial energy wastage.

This study presents a novel approach that addresses the challenge of balancing data replication with energy efficiency in cloud computing. The proposed approach integrates two conflicting paradigms including energy efficiency and data replication. Energy efficiency requires shutting down underutilized nodes, whereas data replication aims to place replicated data on underloaded nodes for faster access (potentially saving time as compared to complex retrieval algorithms). Additionally, the study presents a mechanism that enables simultaneous operation of data replication and dynamic power management (DPM) including an intelligent data replication placement strategy. The placement strategy categorizes the nodes based on their current workloads and implements a tailored policy for each workload category. Based on CPU utilization, the workloads are categorized as underloaded, normally-loaded, and overloaded respectively. Underloaded nodes are powered off through DPM for energy efficiency, whereas the workload of overloaded nodes is balanced via a load balancer for optimal performance. However, neither underloaded nor overloaded nodes are considered while making decisions about the data replication placement. The data replication is hosted only upon the normally-loaded nodes that neither hinder the process of DPM nor degrade performance during the data replication process by becoming unresponsive. Data replication is hosted only on the normally-loaded nodes, ensuring they neither impede DPM processes nor compromise the performance during the replication process due to unresponsiveness.

Furthermore, the decision of replication placement is dependent on factors such as CPU utilization, proximity to requesting clients, available bandwidth, and available memory. The proposed approach outlines the rationale for initial placement of the replica as well as continuously monitors the host for these factors even after the placement. In case of the current host become unsustainable, the replica is automatically migrated to a new, more suitable node. The proposed EnE-Rep introduces several key features for achieving balanced resource utilization and energy efficiency in cloud data center given as following:

- Introduction of a framework that categorizes hosts within the cloud data centers into underloaded, normally-loaded, and overloaded based on the workload.
- Implementation of a double threshold mechanism that activates the load manager and energy monitor in response to the dynamic and unpredictable workloads typically encountered in cloud data centers.
- Integration of a replicator module capable of intelligently selecting an energy-efficient node for replica placement based on the factors such as CPU utilization, proximity, bandwidth, and memory.
- Development of an architecture incorporating VM selection methods for facilitating VM migration from overloaded and underloaded hosts.
- Evaluation of the proposed algorithm's performance using CloudSim, and Planetlab (a real-world workload consisting of 800 cloud data centers distributed across 500 distinct locations worldwide).
- Comparative analysis of the results with an approach that employs intelligent placement of data replication based on popularity for energy consumption.

Section 2 presents the related work; Section 3 clearly defines the problem statement; Section 4 introduces the proposed EnE-Rep model; Section 5 describes the experimental setup used for evaluation; Section 6 presents results and relevant discussion; and finally, Section 7 presents the conclusion based on the discussion in section 6 and potential avenues for future work.

2. Related Work

Data replication involves the decisions regarding the creation, storage, placement, and processing of a necessary replica. Replication decisions, which vary based on context including centralized, distributed, offline, or online, significantly affect the system performance and user experiences. Similarly, replication placement is an important aspect of data replication, particularly, the decision regarding the optimal location for transfer the replica poses a significant challenge. Therefore, placement scheduling should carefully be managed for preventing network congestion, ensuring replica availability, and maintaining efficient access times. A concise overview of the relevant studies is presented as following:

Atrey et al. (Atrey et al., 2019) proposed a scalable placement strategy for distributed cloud storage systems which partitions the data to manage large workloads efficiently. The researchers incorporate two scalable algorithms for efficiently addressing the computational demands. By partitioning data, the revised model enhances the system scalability and resource utilization. In addition, the model enhances system performance by reduces processing time and computational cost. However, the effectiveness of the partitioning model may vary depending upon data characteristics which necessitates the maintenance of data integrity and accessibility. Additionally, the algorithms may introduce complexity and potential trade-offs in terms of accuracy and resource usage. Similarly, Zhang (Zhang, 2020) introduced a time-efficient multi-objective approach for the replication placement problem in cloud storage systems by prioritizing Quality of Service (QoS) restrictions to minimize system response time. The proposed approach ensures an improved user experience and meets performance requirements by implementing QoS restrictions, as well as providing a balanced solution through the simultaneous optimization of various factors. However, potential drawbacks of the proposed approach include the complexity of the optimization process, challenges in meeting all QoS restrictions, and the assumption that minimizing the response time is always the primary objective, which may not align with other system requirements or trade-offs.

Subsequently, Ao and Psounis (2020) proposed a framework for efficient resource allocation in cloud computing systems for handling hierarchical and heterogeneous tasks. The framework minimizes task completion time by leveraging two key strategies including data replication for system reliability and a hierarchical resource management structure for optimizing performance. However, the framework's effectiveness depends on precise resource allocation algorithms and workload characterization. Inaccurate or inefficient allocation methods may lead to suboptimal task completion times. In addition, the hierarchical structure may introduce additional complexity and overhead. On the other hand, Huang et al. (Huang et al., 2020) proposed a mining-based approach for discovering interactions between data entities in cloud storage. The proposed approach aims to improve efficiency and reduce energy consumption by optimizing resource allocation. Additionally, the mining approach incorporates replica placement and backup for enhanced data availability and fault tolerance. However, inaccurate capture of interaction and relevant relationships may limit the potential efficiency gains. Moreover, replica placement and backup require additional storage space and computing resources.

Next, Bacis et al. (Bacis et al., 2019) proposed a data management approach for cloud storage that guarantees data availability and confidentiality during node failures. The proposed approach leverages "all-or-nothing" transformations and fountain codes. All-or-nothing transformation secures data integrity and confidentiality through encryption, whereas fountain codes enable data recovery from transmission errors or failures. However, weak encryption or inefficient fountain codes may compromise security or data availability in addition to the computational overhead by encryption and decoding processes. Similarly, Khalili Azimi (2019) proposed a data management approach based on a bee colony optimization for enhancing data availability in cloud storage. The bee colony optimization algorithm provides a decentralized and self-organizing approach, mimicking the behavior of a bee colony to efficiently search for optimal replication configurations. Consequently, the system demonstrates robustness against the changing conditions and optimize replica placement based on factors such as data importance, workload patterns, and resource availability. However, accurate decision-making is significant, as inefficient choices can result in wasted resources. Moreover, Edwin et al. (Edwin et al., 2019) introduced a dynamic and cost-effective data replication approach that enhances data availability and the replication process. The data replication approach utilizes a multi-objective optimization scheme that prioritizes cost-effective replication by considering replica costs in various data centers. In addition, the knapsack algorithm is enhanced to balance availability and load during replication, optimizing cost-effectiveness and load balancing. By dynamically adjusting replication levels based on cost and availability, the proposed approach optimizes resource utilization and reduces unnecessary overhead. However, performance of the data replication approach depends on the accuracy of the cost model and the knapsack algorithm; inaccurate cost estimates may result in suboptimal replication decisions, thereby impacting cost-effectiveness. Furthermore, Mostafa (2020) introduced a data replication consistency method for cloud-fog environments for improving system availability, fault tolerance, and Quality of Service (QoS). The research aims to prioritize the preparation of the system for potential availability issues to ensure continuous service. The implementation of data replication consistency enhances fault tolerance, minimizing data loss and disruptions, which leads to a more reliable and consistent user experience (QoS). However, inadequate or inconsistent replication can cause data inconsistencies. Additionally, the trade-off between system availability and resource utilization should be carefully managed to avoid excessive replication overhead.

On the other hand, Ramanan and Vivekanandan (2019) investigated the security vulnerabilities in cloud systems using a stochastic diffusion search algorithm for optimizing data replication costs. The stochastic algorithm promotes efficient resource utilization and cost savings by intelligently distributing replicas based on dynamic factors such as workload, resource availability, and network conditions. In addition, the stochastic diffusion algorithm strengthens cloud system security, safeguarding sensitive data from unauthorized access. However, aggressive cost reduction through inaccurate modeling or the algorithm's inherent randomness (stochastic nature) may pose scalability challenges in large cloud deployments. Conversely, Abbes et al. (Abbes et al., 2020) explored virtualizing container concepts for distributed applications in cloud storage. The research predicts replication factors (i.e., number of copies) needed for maintaining availability during container failures using experimental forecasting based on regression analysis. Although, virtualization improves resource utilization and scalability for containers, however, the regression approach used for replica placement relies heavily on the quality of data, assumed linear relationships, potentially overlooking various factors affecting availability.

Alternatively, Tahir et al. (Tahir et al., 2021) addresses user privacy and data integrity concerns in cloud systems using a Genetic Algorithm (GA) for generating encryption and decryption keys. The proposed tailored approach enhances data security and user privacy, ensuring the confidentiality of sensitive information. However, the computational complexity of the GA approach may strain system resources, potentially affecting performance. Furthermore, safeguarding the secure storage and management of generated keys is essential for upholding data integrity and privacy. Subsequently, Babar et al. (Nazir et al., 2018) proposed the CDSS-RPS data replication system, a two-phase approach for optimizing replica placement and file access time in cloud storage. In first phase, a centralized decision system, leverages node computing capacity for optimal replication placement. On the other hand, the second phase considers factors like access frequency, storage capacity, and response time for improving access time. Although, Gridsim-based implementation validates the effectiveness of the two-phase CDSS-RPS data replication system, however, accurate estimations of computing capacity and response time are significant in managing replica placement and balancing file access. Finally, Ebadi et al. (Tagne Fute et al., 2023) proposed a hybrid heuristic called, Hybrid Particle Swarm Optimization Tabu Search (HPSOTS) for intelligent data replica placement. Due to the trade-off between replication and energy efficiency, the proposed research categorizes the problem as NP-hard. HPSOTS is a nature-inspired algorithm that achieves significant improvements in Total Energy Consumption (TEC) and cost as compared to existing approaches. By evaluating multiple options for fulfilling read or write requests based on energy consumption, the study lays the foundation for our proposed work. A brief summary of most related studies is presented in Table I.

3. Problem Statement

Database replication is an important approach in cloud computing for improving data access times. However, in dynamic and heterogeneous cloud environments with unpredictable workloads, existing data replication scheduling approaches can lead to inefficiencies:

Overloaded Hosts: Replication tasks scheduled on overloaded hosts can increase data access times due to the computational overhead of complex replication algorithms, potentially violating Service Level Agreements (SLAs).

Underloaded Hosts: Replication tasks placed on underloaded hosts lead to wasted energy consumption. These hosts cannot be powered down for energy savings due to the ongoing replication tasks they support for other nodes. This combined effect leads to increased energy consumption and potential SLA violations in cloud data centers.

3.1. Research Objectives

This study aims to develop a novel data replication scheduling approach that addresses the limitations of existing methods by:

Optimizing Resource Allocation: The proposed approach seeks to consider real-time workload information for scheduling replication tasks on suitable hosts, avoiding overloaded nodes.

Minimizing Energy Consumption: Replication tasks are intended to be placed on underloaded hosts that are likely to be powered down for energy savings. By addressing these challenges, the proposed approach can lead to significant reductions in energy consumption and improve the overall efficiency of data replication in dynamic cloud environments.

Table 1. Comparative Summary of Related Work

Article	**Type of Handling**	**Place of Handling**	**Performance Metrics**	**Evaluation Tools**	**Limitations**
(Atrey et al,. 2019)	Placement	Server	Delay	CPP, Python	Weak presentation
(Zhang, 2020)	Placement	Server	Response time QoS	Implementation	Weak and inconclusive model implemented in unfamiliar environment.
(Ao and Psounis, 2020)	Modeling	Server	Cost Delay	Analytical	Highly complex optimization problem.
(Huang et al., 2020)	Placement	Multiple	Energy	MapReduce	Unrealistic system model
(Bacis et al., 2019)	Management	3rd party	Availability Security	Storj	Unclear model weak organization
(khalili azimi, 2019)	Management	Server	Throughput Delay	MATLAB	Unclear model weak organization
(Edwin et al., 2019)	Management	3rd party	Cost Availability Energy	Cloudsim	Increases the overall overhead when increasing replicas.
(Ramanan and Vivekanandan, 2019)	Security	Server	Security Cost QoS	Simulation	Shortage in experimental results
(Abbes et al.; 2020)	Modeling	3rd Party	Availability	Implementation	Malfunctioning of regression around 0 values of prediction.
(Tahir et al., 2021)	Security	Client	Security Delay	Implementation	Shortage in experimental results
(Fan et al., 2021)	Consistency	Multiple	Response time	Cloudsim	Not suitable for real-time data

4. Research Methodology of the Proposed EnE-Rep Model

This study introduces an inclusive model designed to optimize the benefits of the data replication process while simultaneously reducing the overall system energy consumption. The model, called EnE-Rep, categorizes nodes into three distinct groups: underloaded, normally-loaded, and overloaded. For each category, a tailored strategy is implemented such that overloaded nodes are balanced, whereas underloaded nodes are efficiently shut down using virtual machine consolidation and dynamic power management methods for energy efficiency. Similarly, this section provides a comprehensive overview of the components and nuances comprising the architecture of the proposed model. Major subsections present the discussion on topics such as replication request submission, SLA checking, utilization management, load management, energy management, sorting management, and periodic recursion monitoring respectively.

Nevertheless, the underlying system prioritizes honoring Service Level Agreements (SLAs) for time-constrained users. An initial check ensures any optimization process won't introduce delays that could violate these SLAs. Subsequently, the method leverages a heuristic-based approach to determine CPU utilization for each node (Hastie et al. 2009). The heuristic-based approach is then utilized for categorizing the nodes as overloaded, underloaded, or normally-loaded. CPU utilization exceeds 85% in the overloaded nodes, whereas it remains below 30% in the underloaded nodes (Beloglazov et al., 2012; Hastie et al., 2009).

Once the workload of a node is determined, the overloaded nodes are directed to the load balancer module. The load balancer module identifies the least occupied node from the entire host list and creates a new VM on a suitable node for transfering the excessive load. Similarly, the workload from underloaded nodes is migrated, and the nodes are vacated. These vacated nodes are subsequently powered off to reduce the overall energy consumption of the system. In the final phase of the proposed approach, normal nodes are sorted in ascending order based on their CPU utilization, proximity, bandwidth, and available memory. Percentile values from all sorted lists are standardized to bring them onto the same scale. The weighted average, calculated as the summation of the product of weights and quantities divided by the summation of weights, is determined according to Eq. (1).

$$\boldsymbol{Weighted\ Average} = \frac{\sum(\boldsymbol{Weights} \times \boldsymbol{Quantities})}{\sum \boldsymbol{Weights}} = \frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i x_i}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i}} \quad (1)$$

where $\boldsymbol{w_i}$ represents the weight of the objective in a priority-based arranged list and $\boldsymbol{n}$ indicates the total number of objectives. Subsequently, leveraging the weighted average calculations from Eq. (1), EnE-Rep creates a weighted average list for all normal nodes in contention for replica placement using Eq. (2).

$$\boldsymbol{W_{Avg}} = 40 * CPU + 30 * Prox + 20 * BW + 10 * \quad (2)$$

where weights ranging from 40 to 10 are assigned to factors influencing replica placement, with higher weights indicating greater influence on the final score. Similarly, $\boldsymbol{CPU}$ represents the CPU utilization of the node, reflecting the node's processing capacity. On the other hand, $\boldsymbol{Prox}$ shows the proximity of the node to the requester(s) who will access the replica, whereas the bandwidth $\boldsymbol{BW}$ represents data transfer capabilities between the node and requesters. Finally, available memory ($\boldsymbol{RAM}$) on the node is important for storing replica data effectively. The node with the lowest score on this list is selected to host the data replica. This selection approach prioritizes a balance between resource utilization, data access speed, and energy efficiency. A detailed explanation of each sub-module within EnE-Rep is provided in the following sections.

4.1. Replication Request Submission (RRS)

The RRS module initiates the data replication process under two primary conditions. Firstly, any modification made to the original data (denoted as "t") triggers replication, ensuring all replicas are updated with the latest version. Secondly, when a remote user accesses data from a remote replica repository and requests an updated copy, replication is triggered to provide the user with the most recent version of the data.

4.2. SLA Manager

The SLA Manager prioritizes adherence to Service Level Agreements (SLAs) for time-constrained users. Given the complex algorithms involved in data replication to ensure proximity to the requester, the SLA Manager identifies and separates cloudlets with time constraints from those operating under more flexible timeframes. Resultantly, the exclusion of time-constrained nodes from subsequent optimization steps effectively prevents potential SLA violations, thereby ensuring the fulfillment of their SLAs.

4.3. Utilization Manager

The utilization manager in EnE-Rep plays an important role by conducting a comprehensive analysis of CPU utilization across all nodes prior to scheduling data replication. This analysis serves as a critical filtering mechanism where overloaded nodes surpassing a utilization threshold are routed to the load balancer module for resource optimization, and underloaded nodes with low utilization are earmarked for potential migration and energy conservation through the energy monitor module. Finally, nodes with balanced CPU utilization are directed to the replicator section for data replication tasks. This intelligent allocation process ensures optimal resource utilization and prevents overloading nodes with replication tasks.

4.4. Load Monitor

The Load Monitor, receiving a list of overloaded nodes from the Utilization Manager, acts as a pivotal task reassignment unit for ensuring workload distribution across the system and prevent resource bottlenecks. The operations of Load Monitor consist of three main steps; first, it identifies suitable underloaded hosts from the entire host pool, considering factors like available CPU capacity, memory, and bandwidth. Secondly, Load Monitor assesses the projected workload on the candidate host post-load transfer, ensuring it remains below a predefined upper threshold to prevent overloading. Upon passing the feasibility check, the Load Monitor executes the workload transfer, potentially involving the creation of a new virtual machine (VM) on the underloaded host. Finally, after completing load balancing via VM consolidation and Dynamic Power Management (DPM), the Load Monitor forwards an updated list of "normalized" hosts—those with balanced workloads—to the Replicator module for optimal replica placement decisions.

4.5. Replicator

The replicator module serves as the focal point for determining data replication placement, considering four key factors: CPU utilization, proximity, bandwidth, and available memory across all hosts. To facilitate fair comparison, lists corresponding to each property are created and processed through a percentile calculator by aligning units across diverse metrics. Subsequently, weights for each property are computed using Eq.

(2), and their weighted average is calculated. This process identifies the optimal host for data replication, ensuring efficient resource utilization. Furthermore, the hosting VM's priority is elevated to mitigate any delays incurred during decision-making that guarantees swift data access.

4.6. Energy Monitor

The proposed model comprises several key components, each playing a significant role in optimizing system efficiency. Firstly, underloaded nodes identified by the utilization manager undergo dynamic power management, where idle nodes are shut down to decrease overall energy consumption. This process involves transferring the workload of nodes below the CPU utilization threshold to other suitable hosts based on CPU, bandwidth, and memory considerations before shutting them down, as depicted in Algorithm 1. Additionally, Figure 1 illustrates the comprehensive architecture of the model, depicting its major components and their interactions. Figure shows the end user's interaction with remote hosts where cloud computing services are accessed. Behind these remote hosts, the proposed methodology's operational intricacies are implemented. Following optimization, the relevant data is integrated into the central database.

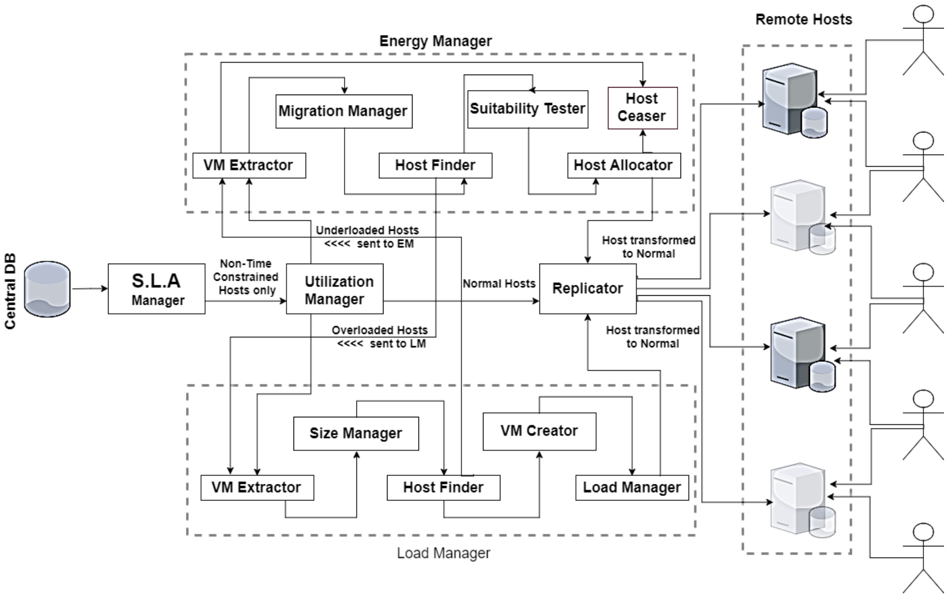

Figure 1. Detailed Architecture and Interaction Diagram of EnE-Rep

The algorithm for energy-efficient data replication outlines the optimization mechanism applied to non-time constrained cloudlets. CPU utilization is prioritized, with heavily utilized nodes given precedence. Lists for proximity, bandwidth, and memory are sorted and transformed into percentile lists to standardize units. The weighted average formula generates an energy-efficient list, with component weights determined by their perceived importance.

Algorithm 1: EnE-Replication for Normal Ranged Nodes

Input: Hosts within the utilization range <Normal >

Output: Replica placement on the BEST among <Normal >hosts

```
cloudlet.makeReplica()
forall hostsinhostList do
    if replica==True then
        if TC==False then
            cputilSorted=getUtil( hostList)
            Utilization list of hosts is created and sorted ascendingly
            As greater is preferred
            proxSorD= getProx(hostList)
            Proximity list of all hosts is created and sorted descendingly
            As lesser is
            bwSor=getBw (hostList )
            Bandwidth utilization list is created and sorted ascendingly
            asgreaterispreferred
            ramSor= getRam(hostList)
            RAM utilization list of all hosts is created and sorted
            Ascendingly as greater is preferred

            cputilPercen ← calPercen (cputilSorted)
             proxPercen ← calPercen (proxSorD)
             bwPercen ← calPercen (bwSor)
             ramPercen ← calPercen(ramSor)

            eeList.add(i)=40*cputilPercen.get(i)+30*proxPercen.get(i)
             +20*bwPercen.get(i)+10*ramPercen.get(i);
            forall itemineeList do
                if current < previous then
                    Best ← current
                                  // smallest element is searched
                end
                Best ← previous
            end
            allocateReplica(Best.getId())
               // replica is placed on Host which is BEST w.r.t
             all four (Util.Proximity.BW.RAM)parameters
            setPriorityHigh(getReplicatedVm())
              // Priority of VM dealing replica on BEST host is
             set to High so that scheduler gives it PE early
             and Max available to counter any data access delay
        end
    end
end
```

The algorithm then ranks normally-loaded nodes based on their total score across all parameters, selecting the most suitable host for data replication. This approach aims to maximize energy savings by efficiently utilizing system resources and minimizing idle states. To compensate for optimization time, the host node's priority is set to high.

The algorithm for energy-efficient data replication in data-intensive clouds outlines the operational framework of the proposed approach. Upon receiving a cloudlet with a data replication scheduling request, the model employs its optimization mechanism that is tailored for energy-efficient placement through exclusive attention to non-time constrained cloudlets. Initially, the algorithm retrieves CPU utilization data and arranges it in descending order to prioritize highly occupied nodes. Similarly, sorted lists are generated for proximity, bandwidth, and memory along with the percentile lists are established for standardizing the units. Nodes with the highest numerical values for each parameter top their respective lists, and an energy-efficient list is computed using a weighted average formula. Component weights are assigned based on perceived significance; however, CPU utilization is prioritized due to its relevance to workload segregation. The algorithm evaluates all four weighted average values for each node to rank them based on their collective scores. Among normally-loaded nodes, those with the highest scores are deemed optimal for data replication placement, striking a balance between workload and energy efficiency. This strategy facilitates the idling and shutdown of underloaded nodes, contributing to significant energy savings. Finally, host node priority is elevated for optimization time that ensures efficient scheduling of data replication tasks.

5. Experimental Setup

The Infrastructure as a Service (IaaS) model in cloud computing offers extensive computing resources with advantages like repeatability and resource control which necessitates thorough testing of proposed data replication approach on large-scale Data Centers (DCs). However, physical platforms of such magnitude are challenging to procure, prompting the use of simulation. Leveraging Cloudsim toolkit v3.0 proves ideal for this purpose that is tailored for cloud environments and sparing users from intricate details. Cloudsim facilitates dynamic workload integration through the inclusion of energy consumption modeling and accounting functionalities. Following is a detailed overview of the infrastructure setup and the submitted jobs for simulation:

5.1. Resource modeling

This study utilizes the CloudSim Toolkit 2.0 platform (Beloglazov *et al.* 2012), developed by Beloglazov and Buyya, for simulating a data center (DC) environment. The simulated DC comprises 800 Physical Machines (PMs), with half being HP ProLiant ML110 G4 servers and the other half HP ProLiant ML110 G5 servers. Table 2 provides detailed specifications regarding RAM and Processing Element (PE) for these server types. The server models such as HP ProLiant ML110 G4 and G5, demonstrate varying RAM and PE specifications, as presented in Table 2. Similarly, the processing power, measured in MIPS (Million Instructions Per Second), varies between the server models. The HP ProLiant ML110 G4 delivers 1860 MIPS, whereas the G5 model is more powerful at 2660 MIPS. Additionally, each server is allocated a bandwidth of 1000

MBs. Virtual Machines (VMs) in this experiment emulate Amazon EC2 instances, however, configured with a single core. The simulations are conducted on actual hardware platforms, comprising HP ProLiant and IBM GX3250 machines.

Table 2. Resource Specification of the Servers used in Simulation

Instance Type	Specification
Extra Large	2000 MIPs, 3750 MB
Medium	2500 MIPs, 850 MB
Small	1000 MIPs, 1700 MB
Micro	500 MIPs, 613 MB

5.2. Application modeling

Table 3 shows the parameters adjusted to create distinct workload scenarios, presented in ascending order of intensity. The application model utilizes authentic data sourced from the PlanetLab project that is specifically gleaned from traces of over 1000 VMs allocated to diverse users. These traces, derived from PlanetLab's CoMon Project spanning 10 days, depict authentic workload patterns. The rationale behind employing linear workload variations stems from the understanding that power consumption correlates linearly with factors such as CPU utilization, memory usage, storage access, and network activity. This methodology enables the evaluation of the EnE-Rep model's scalability across varying workload intensities.

5.3. Performance evaluation parameters

EnE-Rep differs from traditional replication considerations by placing a primary emphasis on energy efficiency over factors such as cost, response time, and reliability. On the other hand, conventional approaches prioritize various performance metrics as compared to EnE-Rep's novel strategy revolves around minimizing power consumption. Studies identify the direct correlation between a system's power usage and factors including CPU utilization and memory usage (Beloglazov et al. 2012; Fan et al., 2007; Kusic et al., 2009). In addition, data access time is influenced by factors like distance and available bandwidth. However, EnE-Rep introduces a unique energy conservation method by strategically migrating virtual machines (VMs) from specific hosts, enabling their shutdown to conserve energy. Subsequently, to assess the energy-saving benefits, EnE-Rep evaluates key metrics including the total number of VM migrations, successful host shutdowns, and components of data access time such as VM selection, host selection, and VM relocation time. EnE-Rep actively evaluates its energy-saving effectiveness through several key metrics. These metrics include the number of VM migrations enabling host shutdowns, and the various components of data access time including VM selection, host selection, and VM relocation time.

Table 3. Workload Variations Applied in the Experiment

Workload Set	Job file size (Bytes)	Job length (MI)	No. of VMs	No. of Hosts
1	300	2500	1000	1000
2	650	5000	2000	2000
3	1000	7500	3000	3000
4	1300	10000	4000	4000
5	1500	13000	5000	5000
6	1800	16000	6000	6000
7	2200	20000	7000	7000
8	2600	25000	8000	8000
9	3000	30000	10520	8000

5.4. Energy consumption

Cloud data centers are major consumers of energy that is primarily attributed to CPUs, storage disks, and network equipment, with CPUs being the most power-intensive components. Traditionally, techniques like Dynamic Voltage and Frequency Scaling (DVFS) have been employed to mitigate CPU power consumption through adjusting operating frequency and voltage. Despite its near-linear relationship between power and frequency, DVFS is limited by the finite number of available frequency states. In contrast, EnE-Rep adopts a more significant approach by powering down idle nodes based on the notion that around 70% of power is consumed by idle resources. Leveraging this strategy enables EnE-Rep to achieve greater energy savings compared to DVFS. Energy consumption is measured using Eq (3) (Cidon *et al.* 2013) given as following:

$$P(u) = K * P_{max} + (1 - K) * P_{max} * U \tag{3}$$

where P_{max} denotes the maximum power consumption under full server utilization, K represents the fraction of power consumed by the idle server, and U signifies the CPU utilization. Subsequently, energy is computed using Equation (4) (Bagheri and Mohsenzadeh, 2016) given as following:

$$E = \int_{0}^{\infty} P(u)\ dt \tag{4}$$

Where E represents the total energy consumption over the time period starting from t and extending indefinitely into the future. Similarly, $P(u)$ denotes the power consumption, which is a function of the CPU utilization $u(t)$. The function $P(u)$ gives the power consumed by the system at any given time t. Subsequently, t is the lower limit of the integral, representing the starting time from which the process of measuring energy consumption has begun. Finally, ∞ highlights the upper limit of the integral, indicating that the energy consumption is being considered over an infinite time period, essentially summing up the power consumption from time t to the end of time (or theoretically, forever).

Validation of the Equation

It is important to consider the context in which equation (4) is applied. The equation assumes that the system, such as a server, operates continuously starting from time t without a defined endpoint. This assumption is particularly relevant for systems like cloud servers, which are often designed to run indefinitely. Additionally, the power consumption $P(u)$ is time-dependent because the CPU utilization $u(t)$ varies over time. Equation (4) accounts for this variability, recognizing that power consumption is not constant but fluctuates with the level of CPU usage at any given moment. Furthermore, the integral in the equation accumulates the total energy consumed over the period from time t to ∞. Since energy is the product of power and time, integrating the power over this period yields the total energy consumption, providing a comprehensive measure of the system's energy usage.

The choice of ∞ as the upper limit in the integral can be justified on several grounds. Firstly, it ensures theoretical completeness by covering the entire potential lifespan of the system, thus accounting for all possible future energy consumption. This is particularly relevant in theoretical models where the system is assumed to operate indefinitely. Secondly, using ∞ as the upper limit is essential for modeling long-term energy consumption, especially in systems like cloud data centers which are designed for continuous operation. This employed approach aids in understanding long-term energy consumption patterns, which is important for making informed decisions about energy efficiency, sustainability, and cost management. Additionally, integrating up to ∞ enables worst-case scenario analysis by estimating the maximum possible energy consumption over time, which is valuable for planning purposes such as provisioning energy resources and designing cooling systems.

For comparison, the energy consumption is calculated as

$$EnergyConsumption\left(\frac{Kw}{h}\right)=\frac{EnergyConsumption}{3600*\ 1000} \tag{5}$$

6. Results and discussion

This section presents a detailed performance evaluation of the proposed EnE-Rep model against the classical scheduling policies and a metaheuristic technique called Hybrid Particle Swarm Optimization Tabu Search (HPSOTS). Figure 2 presents a heatmap that visually compares the number of VM migrations required by different scheduling techniques for all seven tested scheduling policies. Figure shows that the scheduling policies without optimization (thrrs, iqrmmt, iqrrs, lrrs, madmc, thrmc, and thrmu) suffer from significantly higher VM migrations, as indicated by the darker shades in the heatmap. This is due to their less effective approach of placing replications on the first available host without considering the load or suitability of the host. However, HPSOTS exhibits a reduction in VM migrations as compared to non-optimized techniques.

HPSOTS applies some level of intelligence in selecting hosts for replication placement, potentially considering factors that contribute to energy consumption. On the other hand, EnE-Rep demonstrates the most significant reduction in VM migrations as compared to both non-optimized scheduling methods and the HPSOTS metaheuristic technique. This prominent improvement is evidenced by the decrease in migrations from a staggering 44200 to a more manageable 25349. The success of EnE-Rep in minimizing migrations can be attributed to its intelligent approach to replica placement. EnE-Rep adopts a double threshold policy for CPU utilization, ensuring that replications are only placed on hosts with CPU usage within a specific, optimal range. By avoiding overloaded hosts, EnE-Rep eliminates the need for frequent migrations that is caused by the performance bottlenecks which results from insufficient resources. Additionally, by steering clear of underloaded hosts, EnE-Rep prevents unnecessary migrations triggered by inefficient resource allocation on underutilized machines.

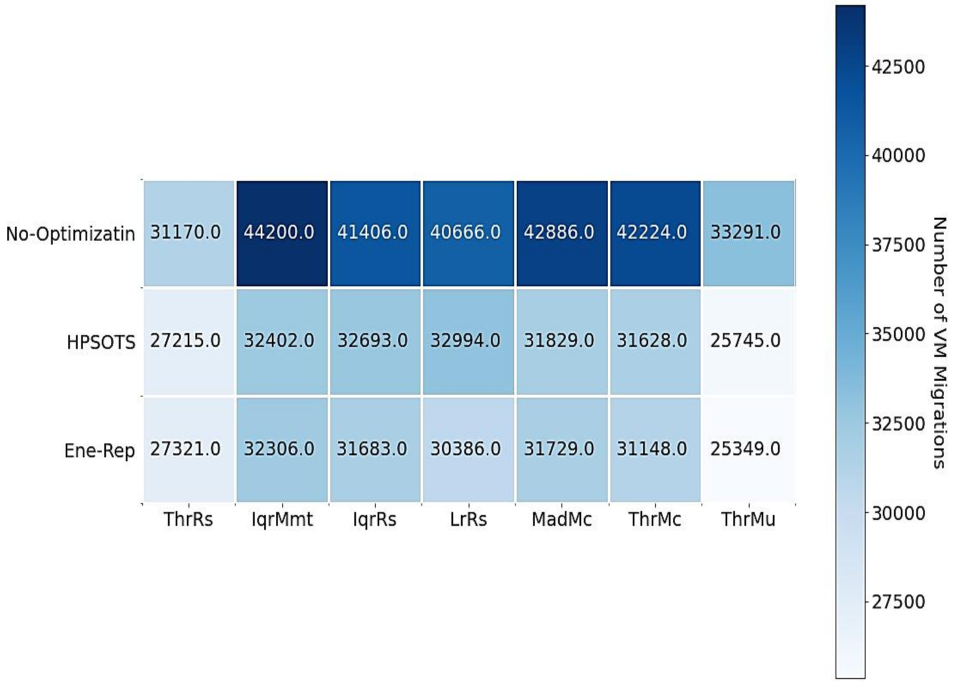

Figure 2. Heatmap for number of VM migrations during the execution

Similarly, Figure 3 explores another important aspect of VM migrations – the mean time before a VM migration becomes necessary. The analysis in Figure 3 compares how long VMs stay on a host before needing to be migrated. Unsurprisingly, non-optimized policies perform ineffictly due to their lack of migration consideration. The high frequency of unnecessary migrations in these policies directly affects their performance. However, HPSOTS prioritizes energy efficiency by evaluating the entire host population, nonetheless, it might not prioritize factors that directly reduce the number of

VM migrations. On the other hand, EnE-Rep shows better performance as compared to classical scheduling policies, particularly from HPSOTS by its adept optimization of VM migration frequency. The optimization is accomplished through a targeted approach: firstly, by assessing the CPU utilization of potential host candidates, and secondly, by prioritizing the placement of replications exclusively on hosts with normal CPU loads. This meticulous methodology yields several notable advantages. Firstly, it leads to reduced disruptions by allowing VMs to remain on suitable hosts for extended durations, thus mitigating the need for frequent migrations. Secondly, it enhances system performance by avoiding overloaded hosts, thereby averting potential performance degradation that is caused by the resource bottlenecks which culminates from frequent migrations.

Subsequently, Figure 4 presents a comparison of the time taken by each method to select a suitable host for data replication placement. HPSOTS exhibits the longest selection time because it evaluates the entire host population and ranks them based on energy consumption, thereby prioritizing comprehensive analysis over speed. In contrast, both EnE-Rep and the non-optimized methods demonstrate relatively similar selection times.

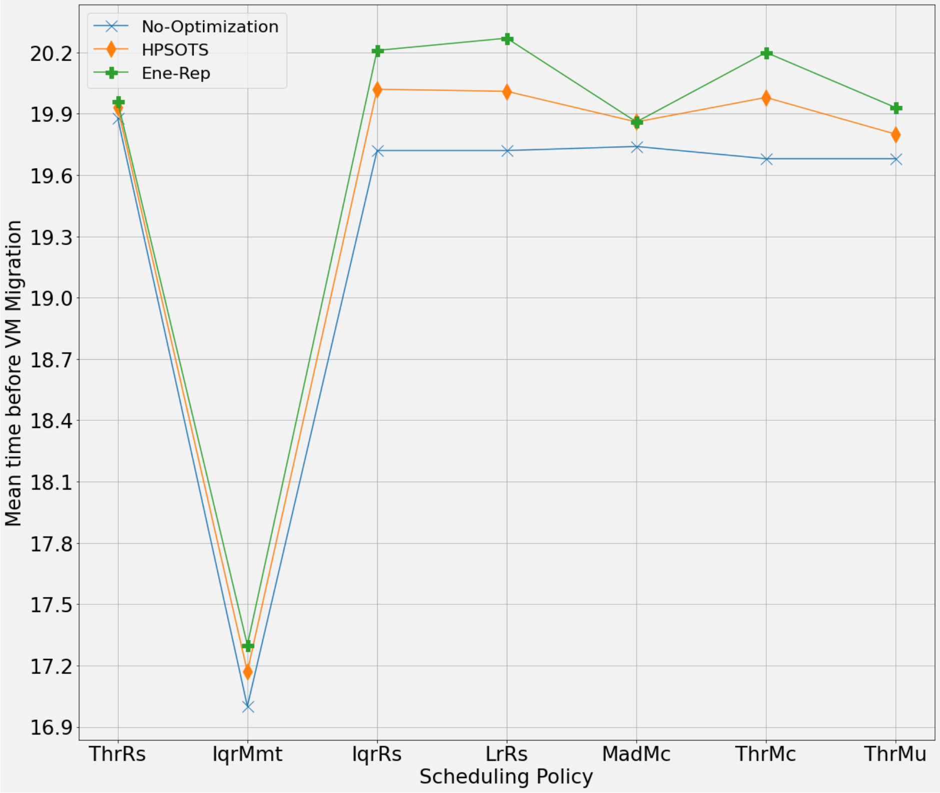

Figure 3. Mean time before a VM migration throughout the execution

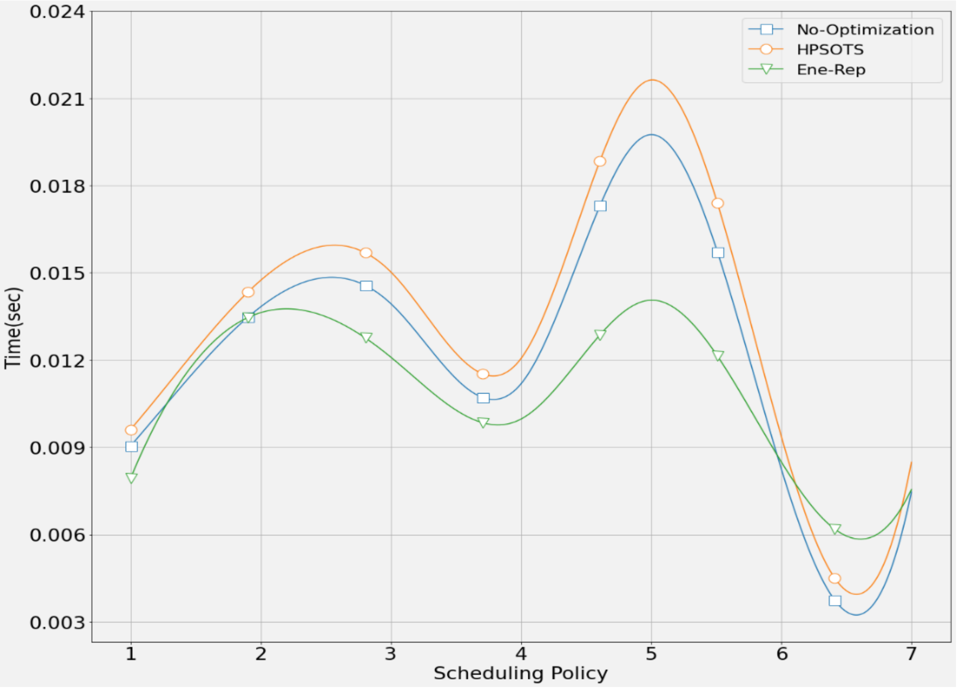

Figure 4. Mean time for host selection for placement of data replication

These approaches share a key characteristic, i.e., these methods focus on evaluating a single candidate host at a time, rather than the entire pool. Therefore, once a suitable host is found and meets the requirements, the replication is placed, and the selection process ends. However, in non-optimized techniques, if the initial candidate is unsuitable, an iterative search might be necessary to find an alternative that leads to longer selection times. EnE-Rep, on the other hand, leverages a predefined CPU utilization threshold, allowing it to identify suitable hosts faster as compared to the non-optimized technique's potentially time-consuming iterative search. By focusing on a specific CPU utilization range, EnE-Rep efficiently narrows down potential candidates that results in shorter selection time.

Finally, Figure 5 illustrates the energy consumption patterns of the proposed EnE-Rep against the other methods across all seven scheduling policies, providing a comprehensive view of their energy usage. The non-optimized techniques, represented by the blue bars, exhibit notably higher energy consumption in kilowatts. This heightened consumption can be attributed to two main factors. Firstly, non-optimized techniques trigger a substantial number of VM migrations, resulting in significant performance degradation. Additionally, these techniques adopt an unintelligent approach to replication placement, indiscriminately utilizing any available host regardless of its current workload. The utilized indiscriminate placement leads to disadvantages such as the replications placed on overloaded hosts trigger frequent migrations to address performance bottlenecks caused by insufficient resources. Conversely, placing replications on underloaded hosts results in wasted energy consumption because these machines remain powered-on despite having minimal workload. Resultantly, overall increase in energy consumption is observed in non-optimized techniques. In contrast, while HPSOTS focuses on energy reduction, it does not explicitly consider the impact of replication placement on migration frequency. This oversight may result in the selection of energy-efficient hosts that are not optimal in terms of migration that can limit HPSOTS's overall energy savings as compared to EnE-Rep.

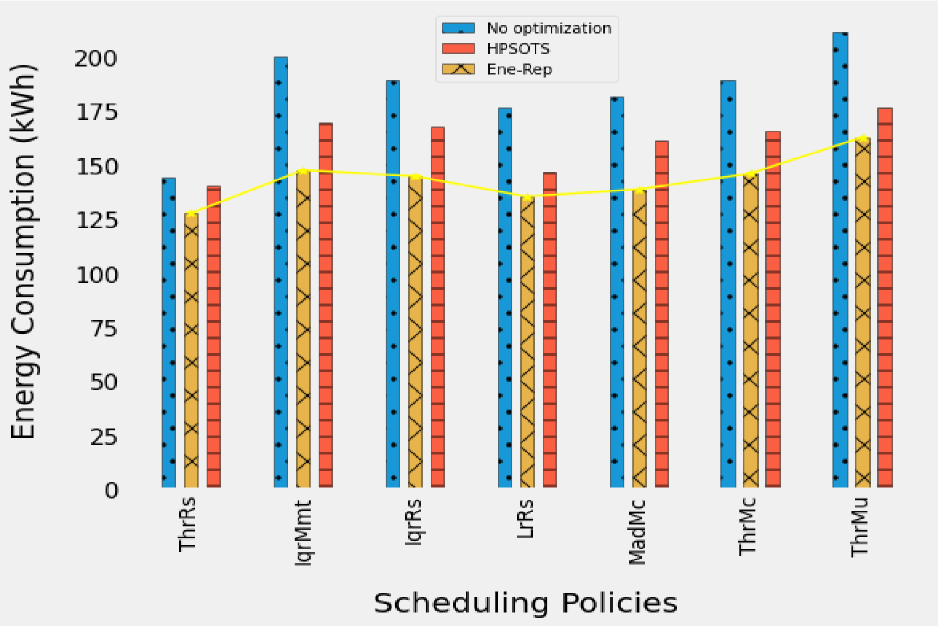

Figure 5. Energy consumption comparison of different strategies

7. Conclusion and future work

Cloud computing offers several advantages such as ease of use, affordability, adaptability, growth potential, and dependability, however, its growing infrastructure demands more energy and raises network distribution challenges. In this study, we have proposed EnE-Rep that integrates dynamic power management (DPM) and data replication for optimizing energy usage and enhance performance in simulations.

The performance evaluation of the EnE-Rep model against classical scheduling policies and the HPSOTS metaheuristic technique highlights its effectiveness in minimizing VM migrations and reducing energy consumption in cloud computing environments. EnE-Rep's intelligent replica placement strategy, guided by a double threshold policy for CPU utilization, effectively avoids overloaded and underloaded hosts, thereby mitigating the need for frequent migrations caused by performance bottlenecks. In contrast, non-optimized techniques exhibit higher VM migration frequencies and performance degradation. Although, HPSOTS prioritizes energy efficiency, however, its oversight of migration reduction may limit its overall energy savings compared to EnE-Rep. Additionally, EnE-Rep's efficient host selection process results in shorter selection times as compared to non-optimized techniques. The analysis highlights the drawbacks of non-optimized approaches and emphasizing the importance of intelligent replica placement in reducing energy consumption. Furthermore, the fusion of data replication and energy efficiency in EnE-Rep presents promising avenues for greener and more stable ICT infrastructures.

Future work could explore proactive threshold strategies and decentralized approaches to enhance performance in stochastic cloud computing environments, ultimately advancing the goal of sustainable and efficient technology infrastructure.

Abbreviations

DC	Data center
DPM	Dynamic power management
DVFS	Dynamic Voltage and Frequency Scaling
GA	Genetic Algorithm
HPSOTS	Hybrid Particle Swarm Optimization Tabu Search
MIPS	Million Instructions Per Second
PE	Processing Element
PMs	Physical Machines
QoS	Quality of Service
SLAs	Service Level Agreements
TEC	Total Energy Consumption
VMC	Virtual machine consolidation

Acknowledgments

The author extends his appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through general project under grant number GRP/51/44

References

Abbes, H., Louati, T., Cérin, C. (2020). Dynamic replication factor model for Linux containers-based cloud systems. *Journal of Supercomputing*, **76**(9), 7219–7241.

Ao, W. C., Psounis, K. (2020). Resource-Constrained Replication Strategies for Hierarchical and Heterogeneous Tasks. *IEEE Transactions on Parallel and Distributed Systems*, **31**(4), 793–804.

Atrey, A., Van Seghbroeck, G., Mora, H., De Turck, F., Volckaert, B. (2019). SpeCH: A scalable framework for data placement of data-intensive services in geo-distributed clouds. *Journal of Network and Computer Applications*, **142**, 1–14.

Bacis, E., De Capitani DI Vimercati, S., Foresti, S., Paraboschi, S., Rosa, M., Samarati, P. (2019). Dynamic allocation for resource protection in decentralized cloud storage. In *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, IEEE, , pp. 1–6.

Bagheri, K., Mohsenzadeh, M. (2016). E2DR : Energy Efficient Data Replication in Data. *Journal of Advances in Computer Engineering and Technology*, **2**(3), 27–34,,.

Balakrishnan, S. R., Veeramani, S., Leong, J. A., Murray, I., Sidhu, A. S. (2017). High Performance Computing on the Cloud via HPC+Cloud software framework. In *Proceedings on 5th International Conference on Eco-Friendly Computing and Communication Systems, ICECCS 2016*, IEEE, , pp. 48–52.

Beloglazov, A., Abawajy, J., Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, **28**(5), 755–768.

Bonzi, M. (2021). ASPEN GLOBAL CHANGE INSTITUTE ENERGY PROJECT October 2021 Research Review. Retrieved from https://policycommons.net/artifacts/2186496/aspen-global-change-institute-energy-project-october-2021-research-review/2942473/

Buyya, R. (2009). Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009*, **25**(6), 1.

Cidon, A., Stutsman, R., Rumble, S., Katti, S., Ousterhout, J., Rosenblum, M. (2013). MinCopysets: Derandomizing Replication In Cloud Storage. In *Proc. 10th USENIX Symp. Networked Systems Design and Impementation (NSDI)*, , pp. 1–5.

Ding, Y., Qin, X., Liu, L., Wang, T. (2015). Energy efficient scheduling of virtual machines in cloud with deadline constraint. *Future Generation Computer Systems*, **50**, 62–74.

Edwin, E. B., Umamaheswari, P., Thanka, M. R. (2019). An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center. *Cluster Computing*, **22**, 11119–11128.

Fan, C., Jiang, Y., Mostafavi, A. (2021). The Role of Local Influential Users in Spread of Situational Crisis Information. *Journal of Computer-Mediated Communication*, **26**(2), 108–127.

Fan, X., Weber, W. D., Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. *Proceedings - International Symposium on Computer Architecture*, **35**(2), 13–23.

Hastie, T., Rosset, S., Zhu, J., Zou, H. (2009). Multi-class adaboost. *Statistics and Its Interface*, **2**(3), 349–360.

Huang, Y., Huang, J., Liu, C., Zhang, C. (2020). PFPMine: A parallel approach for discovering interacting data entities in data-intensive cloud workflows. *Future Generation Computer Systems*, **113**, 474–487.

Jahangir, M. H., Mokhtari, R., Mousavi, S. A. (2021). Performance evaluation and financial analysis of applying hybrid renewable systems in cooling unit of data centers – A case study. *Sustainable Energy Technologies and Assessments*, **46**, 101220.

Jennings, B., Stadler, R. (2015). Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*, **23**(3), 567–619.

Kappelman, L., Torres, R., McLean, E. R., … Guerra, K. (2022). The 2021 SIM IT Issues and Trends Study. *MIS Quarterly Executive*, **21**(1), 75–114.

khalili azimi, S. (2019). A bee colony (beehive) based approach for data replication in cloud environments. In *Lecture Notes in Electrical Engineering*, Vol. 480, Springer, , pp. 1039–1052.

Kireev, V. S., Bochkaryov, P. V., Guseva, A. I., Kuznetsov, I. A., Filippov, S. A. (2019). Monitoring System for the Housing and Utility Services Based on the Digital Technologies IIoT, Big Data, Data Mining, Edge and Cloud Computing. In *Communications in Computer and Information Science*, Vol. 1054, Springer, , pp. 193–205.

Ksentini, A., Taleb, T., Messaoudi, F. (2014). A LISP-Based Implementation of Follow Me Cloud. *IEEE Access*, **2**, 1340–1347.

Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., Jiang, G. (2009). Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, **12**(1), 1–15.

Mostafa, N. (2020). A Dynamic Approach for Consistency Service in Cloud and Fog Environment. In *2020 5th International Conference on Fog and Mobile Edge Computing, FMEC 2020*, IEEE, , pp. 28–33.

Mytton, D. (2020). Assessing the suitability of the Greenhouse Gas Protocol for calculation of emissions from public cloud computing workloads. *Journal of Cloud Computing*, **9**(1), 1–11.

Nazir, B., Ishaq, F., Shamshirband, S., Chronopoulos, A. (2018). The Impact of the Implementation Cost of Replication in Data Grid Job Scheduling. *Mathematical and Computational Applications*, **23**(2), 28.

Pierson, J. M., Hlavacs, H. (2015). Introduction to energy efficiency in large-scale distributed systems. *Large-Scale Distributed Systems and Energy Efficiency: A Holistic View*, 1–15.

Ramanan, M., Vivekanandan, P. (2019). Efficient data integrity and data replication in cloud using stochastic diffusion method. *Cluster Computing*, **22**, 14999–15006.

Roser, M. (2022). AI Timelines: What Do Experts in Artificial Intelligence Expect for the Future? *Singularityhub*. Retrieved from https://singularityhub.com/2022/12/18/ai-timelines-what-do-experts-in-artificial-intelligence-expect-for-the-future/

Ruan, K., Carthy, J., Kechadi, T., Baggili, I. (2013). Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. *Digital Investigation*, **10**(1), 34–43.

Tagne Fute, E., Nyabeye Pangop, D. K., Tonye, E. (2023). A new hybrid localization approach in wireless sensor networks based on particle swarm optimization and tabu search. *Applied Intelligence*, **53**(7), 7546–7561.

Tahir, M., Sardaraz, M., Mehmood, Z., Muhammad, S. (2021). CryptoGA: a cryptosystem based on genetic algorithm for cloud data security. *Cluster Computing*, **24**(2), 739–752.

Zhang, H., Chen, G., Li, X. (2019). Resource management in cloud computing with optimal pricing policies. *Computer Systems Science and Engineering*, **34**(4), 249–254.

Zhang, T. (2020). A QoS-enhanced data replication service in virtualised cloud environments. *International Journal of Networking and Virtual Organisations*, **22**(1), 1–16.

Received April 29, 2024, accepted August 1, 2024