Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.038163

Article

Energy Cost Minimization Using String Matching Algorithm in Geo-Distributed Data Centers

Muhammad Imran Khan Khalil[1], Syed Adeel Ali Shah[1], Izaz Ahmad Khan[2], Mohammad Hijji[3], Muhammad Shiraz[4] and Qaisar Shaheen[5,*]

[1]Department of Computer Science & Information Technology, University of Engineering & Technology, Peshawar, 25000, Pakistan

[2]Department of Computer Science, Bacha Khan University, Charsadda, 24541, Pakistan

[3]Computer Science Department, University of Tabuk, Tabuk, 47512, Saudi Arabia

[4]Department of Computer Science, Federal Urdu University of Arts, Science and Technology Islamabad, 44000, Pakistan

[5]Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Rahim Yar Khan Campus, Punjab, Pakistan

*Corresponding Author: Qaisar Shaheen. Email: qaisar.shaheen@iub.edu.pk

Received: 29 November 2022; Accepted: 22 February 2023

Abstract: Data centers are being distributed worldwide by cloud service providers (CSPs) to save energy costs through efficient workload allocation strategies. Many CSPs are challenged by the significant rise in user demands due to their extensive energy consumption during workload processing. Numerous research studies have examined distinct operating cost mitigation techniques for geo-distributed data centers (DCs). However, operating cost savings during workload processing, which also considers string-matching techniques in geo-distributed DCs, remains unexplored. In this research, we propose a novel string matching-based geographical load balancing (SMGLB) technique to mitigate the operating cost of the geo-distributed DC. The primary goal of this study is to use a string-matching algorithm (i.e., Boyer Moore) to compare the contents of incoming workloads to those of documents that have already been processed in a data center. A successful match prevents the global load balancer from sending the user's request to a data center for processing and displaying the results of the previously processed workload to the user to save energy. On the contrary, if no match can be discovered, the global load balancer will allocate the incoming workload to a specific DC for processing considering variable energy prices, the number of active servers, on-site green energy, and traces of incoming workload. The results of numerical evaluations show that the SMGLB can minimize the operating expenses of the geo-distributed data centers more than the existing workload distribution techniques.

Keywords: String matching; optimization; geo-distributed data centers; geographical load balancing; green energy

1 Introduction

In recent years, energy usage in Information and Communication Technology (ICT) systems has been growing due to the massive adoption of cloud computing services such as the Internet of Things (IoT), social media, data analytics, big data, etc., [1,2]. Data centers (DCs) are one of the largest keepers of these computing services, which provides an open-access interface to enable big data analysis [3]. Keeping up with the burgeoning interest in cloud computing, the number of geographically distributed DCs has increased significantly. As a result, it is becoming a growing problem because cloud resources require a lot of energy to power the DCs. Recent studies predict that by 2022, annual data production will have quadrupled every year [3–6]. At the same time, considerable energy is consumed to execute this data. Over the past decade, measurements show that power usage of cloud DCs has climbed by nearly 10% [4]. As a result, the price of energy is rising rapidly to cover the expense of keeping computer resources active and cooled at all times. Moreover, eco-friendly and energy-efficient workload distribution strategies must be carefully planned.

To mitigate the operating cost of the geo-distributed DCs, many cloud service providers (CSPs), e.g., IBM Cloud, Google Cloud, Amazon Web Services, and Microsoft Azure, have focused on exploiting the geographical load balancing (GLB) by considering regional differences in energy prices to minimize the operating cost of the DCs [7]. GLB has been suggested to use the flexibility of geo-distributed DCs by routing incoming workloads to locations where renewable energy is available, thereby facilitating the integration of green energy sources into DCs.

In recent years have seen an increase in the study being done on using green energy in DCs. To power DCs using green energy is another way to lessen the load on the grid from traditional power plants and mitigate the overall cost of the DC [7,8]. Green power supply (such as solar or wind power) have the potential to minimize our reliance on fossil fuels significantly, but they are inherently unstable compared to traditional grid electricity [9]. The time of day, the season, the weather, and other variables may all affect their outputs. Predicting the production of green power supply before using them and then balancing the power sources with the incoming workload in the DC is an efficient way to use green energy.

In order to minimize the use of energy while guaranteeing high-quality service to customers during workload processing, several techniques (e.g., Virtualization, Consolidation, energy derivatives, and Data Deduplication) have been presented by experts in academia and the cloud industry [7–10]. By researching how cloud service providers with several DCs in various regions may distribute their workloads and conserve energy, we substantially contribute to the literature. Unlike prior works [1,8–11], we describe the operating cost minimization problem for the DC as an optimization problem considering the string-matching technique, renewable energy sources, and dynamic energy costs. String matching algorithms have had a significant impact on cloud computing and are crucial to solving many problems in the real world. It supports the completion of time-effective jobs across several disciplines. These approaches are helpful when looking for a specific pattern within a string. Detecting plagiarism, bioinformatics and DNA sequencing, digital forensics, spelling checks, spam filters, search engines, and intrusion detection systems are just a few examples of the practical applications of string matching.

In this paper, we examine green geographically dispersed DCs. At first, the incoming user requests are received by the Global Load Balancer (Global-LB). The function of the Global-LB is to make online decision for routing the incoming user requests to appropriate DC based on minimum energy consumption. The primary objective of this study is to use a string-matching algorithm (i.e., Boyer Moore) to compare the contents of incoming workloads to those of documents that have already been processed in a DC. A successful match prevents Global-LB from sending the user's incoming workload to a DC for execution and displaying the results of the previously processed workload to the user to save energy. On the contrary, if no match can be discovered, Global-LB will allocate the incoming workload to a specific DC, taking into account different factors such as the current renewable energy level, the dynamic price of electricity, the server load, and the expected delay. We formulate the operating cost reduction problem as a linear optimization program. We then develop an online algorithm based on a greedy algorithmic approach to solve the linear optimization problem. We perform extensive experiments based on real-world data and present the efficiency of our proposed algorithm. The contributions of our study are summed up as follows:

- By considering batch workloads and string-matching mechanisms, we formulate linear optimization problems to mitigate the total operating costs of the DC while utilizing renewable energy sources.
- To tackle the optimization problem, we suggest an online algorithm SMGLB, which is based on a greedy algorithmic approach and does not rely on the availability of future data like dynamic energy prices or incoming user requests.
- The effectiveness of our proposed algorithm is then evaluated by contrast with the benchmark algorithms.
- Finally, we demonstrate the usefulness and efficiency of SMGLB by rigorous numerical assessment utilizing real-world data of incoming workload, server utilization level, average delay, and renewable energy generation.

The remaining parts of the research work is organized as follows. The review of literature is presented in Section 2. GLB using string matching technique is formulated as an optimization problem in Sections 3 and 4, describing the problem setting. We propose an online algorithm based on a greedy algorithmic approach called SMGLB to solve the problem in Section 5. In Section 6, we describe the analytical performance outcomes of SMGLB based on real-world traces. Finally, Section 7 represents the conclusion of the paper.

2 Literature Review and Background

2.1 String Matching and Energy Efficiency

According to a recent survey conducted by cloud service providers (CSPs), almost 80% of businesses are investigating data deduplication and string-matching techniques for use in their storage systems, intending to reduce duplicate data and increase storage efficiency [10,12–16]. In order to conserve space in storage, data deduplication can eliminate identical copies of the same information. In many computing domains, such as data analytics, audio and video distribution, pattern recognition, image processing, and natural language processing, string matching is a universal approach for problem-solving [13,17]. CSPs' primary focus is finding and identifying patterns in incoming user requests. After an optical character recognition system recognizes the pattern, a string-matching technique is used to look for a redundant pattern in the database [15].

The key focus of this study is to overcome and extend the previous research studies related to string matching algorithms for energy cost minimization during workload processing in geo-distributed DCs. Reference [10] reviewed. Reviewed different string-matching algorithms regarding pattern length and time complexity. Reference [12] experimentally. examined online approximate for redundant data using string matching algorithms for energy cost minimization during data processing. Reference [13] designed a load balancing algorithm based on string matching techniques in geographically distributed DCs. The current research has mainly focused on removing redundant data to increase data storage efficiency in DCs. Hence, we explore a contemporary problem of operating cost reduction in geo-distributed DCs using a string-matching algorithm during workload processing.

2.2 Geo-Distributed Data Centers with Green Energy

Cloud computing should have reduced cloud DC energy usage. It is possible only if those cloud DCs are as efficient as they can be [4]. In this context, green DCs offer a variety of economical and environmentally friendly benefits. The long-term benefits of investing in a new-generation architecture for a green DC would outweigh any initial costs. The majority of the DCs use green energy sources, which has significantly reduced the cost of the electricity and fuel required to operate them. The green DC also put strategies in place to recycle DC e-waste, creating a new revenue stream [9].

Both in business and academics, interest in renewable-powered DCs is growing [6,8,9]. Previous research investigates the viability and advantages of employing GLB for interactive workloads processing that is delay-sensitive in order to assist the incorporation of green energy in DC [5,18–23]. Prototypes are created to demonstrate the efficacy of these task schedulers, and system implementation challenges with interactive workload schedulers that consider renewable energy sources are examined. Though all of the research, as mentioned earlier, considers only interactive workload, without energy storage devices, and with accurate future information of the DC. Our approach is online (without future information); in contrast, it handles batch workload in conjunction with green energy with energy storage devices in geo-distributed DCs.

2.3 GLB and Energy Efficiency

Reducing energy expenses for widely dispersed DCs is an essential topic of contemporary research [22,24–32]. The primary focus of this body of work is the development of methods for balancing workloads between geographically dispersed DCs to meet energy efficiency. Reference [24] provides. Provides a thorough overview of GLB for managing power consumption in DCs. [25]. Reference [26] conducts“Following the renewables” is the method of GLB that has received much interest recently. For this method to work, the dynamic load balancing mechanism must consider the green energy sources available to DCs [25]. Reference [26] conducts an early investigation into GLB. They demonstrate the impact of fluctuating power prices on brown energy use and suggest algorithms to enhance the use of green power supply using GLB. In order to test the efficacy of their algorithms, they employ numerical simulations based on real-world traces. Reference [27] has built upon this research by suggesting online algorithms to enhance the benefits of distributed green power resources across a wide geographic area. Their study demonstrates how to enhance the usage of green power supply like solar and wind. Reference [28] has proposed a load scheduling mechanism for DCs to improve their sustainability, where carbon emission expenses are represented as the social cost. Reference [22] suggested an algorithm for workload distribution using workload deadlines, the variability of renewable energy sources, the ambient temperature, and the cooling dynamics in DCs. In recent in-depth research work, [29] developed a complete model for workload allocation utilizing online algorithm approaches. A CSP with several DCs powered by brown and green power sources uses the on-site green energy production and geographical difference in energy price to save costs.

All previous research works consider geo-distributed DCs that generate energy from green sources. Similarly, we assume that GLB might help save energy expenses and optimize the use of green sources. While theoretical concerns are necessary, we strongly emphasize real-world applicability by testing our proposed system with electricity pricing, actual workload, and green energy. Compared with the existing research studies, we propose string matching geographical load balancing (SMGLB) based on online greedy algorithm design technique. SMGLB is the provable energy efficient solution that reduces the operating cost of the geo-distributed DC more than the existing workload distribution techniques. under time-varying system dynamics (e.g., considering variable energy prices, the number of active servers, on-site green energy, and traces of incoming workload).

3 Methodology

In this part, we define the problem and provide the system model for incoming workload allocation in geo-distributed DCs.

3.1 Overview and Problem Formulation

We consider a CSP to have N geo-distributed DCs; each DC $i \in N$ is powered by brown and green energy sources. Moreover, we assume that every DC i has homogeneous servers. In each time slot $t \in T$, the incoming workload $W_{in}(t)$ arrives at Global-LB. The primary objective of Global-LB is twofold, first, use a string-matching algorithm (i.e., Boyer Moore) to compare the contents of $W_{in}(t)$ to those documents that have already been processed (t) in a DC. A successful match prevents Global-LB from sending the $W_{in}(t)$ to a DC i for processing and displaying the results of the $W_p(t)$ to the user to save energy utilization. Secondly, suppose no match can be discovered; in that case, Global-LB will allocate the $W_{in}(t)$ to a DC i taking into account different optimization factors such as the current renewable energy level, the dynamic price of electricity, the server load, and the average delay to mitigate the operating cost of the geo-distributed DC. The user requests forwarding model is depicted in Fig. 1.

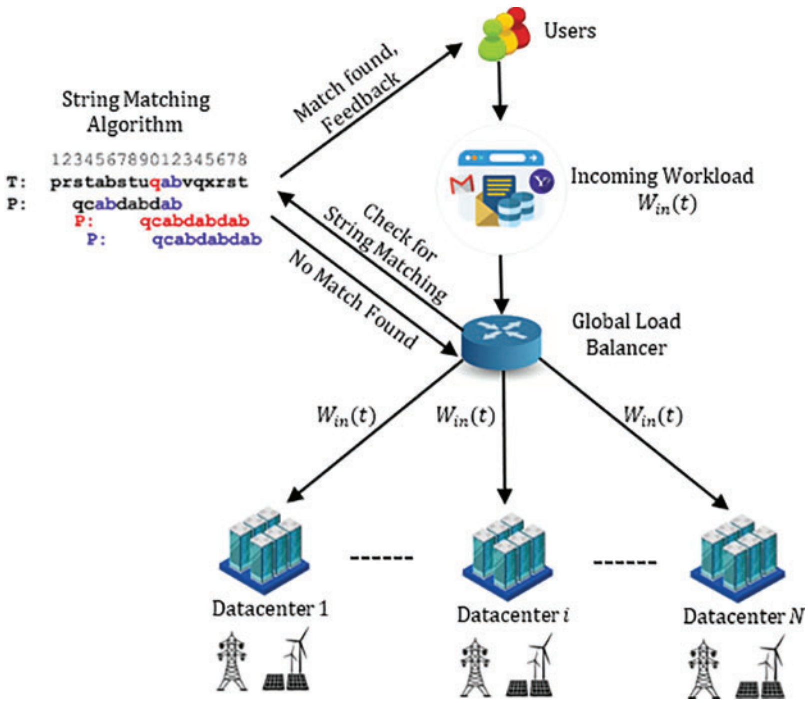

Figure 1: The incoming workload distribution model

3.2 The Workload Model

In cloud DCs, there are distinct workloads that fall into two categories: interactive or transactional workloads, which are delay sensitive, and non-interactive or batch workloads, which are delay tolerant. Internet web and multimedia streaming services are examples of interactive workloads that must be completed within a response time determined by service users. These workloads are frequently networked I/O heavy tasks that have no bearing on the power consumption of servers as long as all of the tasks are must be processed before the deadline. Moreover, batch workloads, such as extensive data analysis and scientific applications, may, in contrast, be scheduled to run whenever they are needed. They frequently include complex computations that demand high CPU use, which increases server power consumption. Since batch workloads that need much computation significantly impact server power consumption more than interactive workloads, we are particularly interested in them in this study. Let $W_{in}(t)$ is the incoming batch workload which will be allocated to datacenter i at time t in case of no match can be discovered during string matching, then we have;

$$\sum_{i=1}^{N} w_{in}^{i}(t) = W_{in}(t)\ \forall t \in [1, T] \tag{1}$$

A DC i may include hundreds or even thousands of servers to handle the massive amount of $W_{in}(t)$. The maximum number of servers S_i^{max} cannot be exceeded by the number of active servers $S_i^{ac}(t)$ in DC i. Consequently, we have;

$$0 \leq S_i^{ac}(t) \leq S_i^{max}\ \forall i \in [1, N] \tag{2}$$

3.3 Green Energy Generation Model

Our model allows DCs to be directly powered by the electrical grid or green energy sources like solar panels. Due to variations in weather, the number of solar panels at each DC should be adjusted accordingly. We assume that there is on-site green power production $R_i(t)$ using solar panels at time t in DC i. Consequently, we find;

$$0 \leq R_i(t) \leq R_i^{max}(t) \tag{3}$$

Eq. (3) assures that the amount of green energy can never be negative and never exceeds the available capacity.

3.4 Bandwidth Cost Model

Bandwidth cost occurred between the Global-LB and the DC for the incoming workload $W_{in}(t)$ processing. The linear bandwidth cost model $B_i(t) = b_i W_{in}(t)\, n_c$ is used in this study to depict the bandwidth cost between Global-LB and the DC.

Note that n_c is the communication demand between Global-LB and the DC, given that b_i is the coefficient of bandwidth cost between Global-LB and the DC. Distinct pairs of Global-LB and the DC have different bandwidth costs. We define the following upper limit of bandwidth cost $B_i^{max}(t)$ between Global-LB and DC:

$$0 \leq b_i(t) \leq B_i^{max}(t) \tag{4}$$

3.5 The Model of Energy Consumption

One of the significant cost-increasing factors in DCs is energy usage. Our model considers that energy rates in different DCs change over time when determining where to process the workload to cut energy costs. In this model, we emphasize the cooling system and IT equipment (i.e., servers) as the primary energy consumers in DCs. We use an empirical methodology based on production Computer Room Air Conditioning (CRAC) energy efficiency statistics for the cooling systems. A production CRAC automatically changes modes in response to the environment. Since the outside temperature is the input and the overall energy efficiency of the CRAC is the output, we analyze CRACs as a black box. We employ Power Usage Effectiveness in particular (PUE). PUE is characterized by $PUE = \frac{Total\,facility\ power}{IT\ equipment\ energy}$. A system with a lower PUE value is more effective and has less overhead [11]. PUE is often between 1 to 2 in cloud DCs [8]. Additionally, we gauge the DC's overall energy usage at the time t; $P_i(t) = PUE_i(t) \cdot\ S_i^{ac}(t)\left[P_i^{in}(t) + P_i^{ac}(t)\frac{w_{in}^i(t)}{S_i^{ac}(t)\,\mu_i}\right]$. Let $C(t)$ be the overall cost of energy for cloud DC operators and $q_i(t)$ be the electricity price at time t in DC i. Then, $C(t)$ can then be represented as; $C(t) = \sum_{i=1}^{N} q_i(t) \cdot P_i(t)$.

3.6 The Energy Cost Minimization Problem

To reduce DC operating expenses, we may use the above energy cost model to determine the optimal routing strategy considering energy consumption $C_i(t)$ and the bandwidth cost $B_i(t)$ in each DC i at any given time t. Consider the following optimization problem, which expresses this idea:

$$min \sum_{t=1}^{T} \sum_{i=1}^{N} C_i(t) + B_i(t) \tag{3.6}$$

Subject to;

$$\sum_{i=1}^{N} w_{in}^i(t) = W_{in}(t)\ \ \forall t \in [1, T] \tag{3.6a}$$

$$0 \le S_i^{ac}\,(\mathrm{t}) \le S_i^{max}\ \forall i \in [1, N] \tag{3.6b}$$

$$0 \le R_i(t) \le\ R_i^{max}(t)\ \ \forall i \in [1, N] \tag{3.6c}$$

$$0 \le b_i(t) \le\ B_i^{max}(t)\ \ \forall i \in [1, N] \tag{3.6d}$$

$$0 \le P_i(t) \le\ P_i^{max}\ \forall i \in [1, N] \tag{3.6e}$$

where (3.6a) is the incoming workload $W_{in}(t)$ at the global-LB equal to the total distributed workloads $w_{in}^i(t)$ among DCs. (3.6b) ensure that, active servers S_i^{ac} (t) is positive and upper bound by the upper limit of servers S_i^{max} at the respective locations. Constraint (3.6c) assures that the amount of green energy $R_i(t)$ can never be negative and not exceeds the available capacity $R_i^{max}(t)$. Constraint (3.6d) ensures the bandwidth cost between Global-LB and DC i cannot exceed the maximum limit. Finally, (3.6e) ensure that, at time t the energy usage of the DC $i\,P_i(t)$ cannot exceed the maximum power consumption limit P_i^{max}.

3.7 Algorithm Design and Proposed Solution

In order to tackle the optimization problem mentioned above, we construct an online algorithm based on the string-matching method "SMGLB" in this part. We describe SMGLB based on an online greedy algorithmic approach in Algorithm 1 to mitigate the operating cost of the DC in each discrete time slot *t*.

Algorithm 1: SMGLB–Minimization of Energy Cost

```
1.  for $t \in [1,\ T]$ do
2.        read $W_{in}(t)\ \forall\ i\ \in [1-N]$
Part A:   Searching for String Matching
3.  $n =$ length $\cdot W_p(t)$
4.  $m =$ length $\cdot W_{in}(t)$
5.  $\alpha =$ Compute last occurrence function $\left(W_{in}(t), m, \sum\right)$
6.  $\beta =$ Compute good suffix function $(W_{in}(t), m)$
7.  $s = 0$
8.  while $s \le n - m$ do
9.    $j = m$
10.     While $j > 0$ and $W_{in}(t)[j] = W_p(t)[s+j]$ do
11.            $j = j - 1$
12.     if $j = 0$ then
13.            $W_{in}(t) == W_p(t)$
14.            $s = s + \beta[0]$
15.     else
16.            $s = s + max\left(\beta[j], j - \alpha\left[W_p(t)[s+j]\right]\right)$
Part B: Selection of DC $i^*$ in case of mismatch occur
17. Calculate $P_i(t)$, $R_i(t)$, $S_i^{ac}(t)$, $S_i^{in}(t)$, $b_{ij}(t)\ \forall i \in [1,N]$ to select the $i^*$
            $i^* = \min_{\arg i \in [1,N]} \sum_{i=1}^{N} \left[C_i(t) + b_{ij}(t)\right]$
      Subject to constraints (3.6a), (3.6b), (3.6c), (3.6d), and (3.6e)
18. Assign $W_{in}(t)$ to DC $i^*$
19. Update $S_i^{ac}(t)$, $S_i^{in}(t)$ and $w_{in}^{i}(t)\ \forall i \in [1,N]$
20. end for
```

SMGLB has two parts. In part A (lines 3–16), a string-matching algorithm (i.e., Boyer Moore) is used to compare the contents of $W_{in}(t)$ to those documents that have already been processed $W_p(t)$ in a DC. A successful match prevents Global-LB from sending the $W_{in}(t)$ to a DC *i* for processing and displaying the results of the $W_p(t)$ to the user to save energy utilization. In line 2, we read the incoming workload $W_{in}(t)$ in each discrete time slot *t* for all DCs. We calculate the string length of the incoming workload $W_{in}(t)$ and already processed workload $W_p(t)$ and assign the value to *n* and *m*, respectively (lines 3 and 4).

The Boyer Moore algorithm using a 'backward' method aligns the beginning of the $W_{in}(t)$ at the beginning of the $W_p(t)$ and then compares the characters of $W_{in}(t)$ from right to left, starting with the character on the right. If a character of $W_{in}(t)$ is compared that does not belong to the $W_p(t)$, no match can be discovered by examining any other characteristics at this place. Therefore, beyond the mismatched character, the $W_p(t)$ can be changed entirely. The Boyer Moore algorithm uses two preprocessing methods simultaneously to identify probable shifts [33]. When there is a discrepancy, the algorithm computes a variation using both techniques and selects the most significant change by applying the most beneficial technique in each circumstance [34]. The following two heuristics of Boyer Moore are used to minimize the search in the incoming workload:

i. Bad Character Heuristics (Algorithm 1, Line 5**)**: Assume that a $\boldsymbol{W_{in}(t)}$ has a letter that never appears predictably. On the other hand, it could be a bad character present in the $W_{in}(t)$; in this case, aligning the nature of the pattern with a bad character in the $W_p(t)$. When a mismatch occurs at this character (called a bad character), the entire pattern can be changed, starting matching from substring next to this "bad character." Therefore, we require further data to facilitate a change in response when meeting a bad character. This data includes the final location of each component in the incoming workload and the alphabet of letters employed. If $W_{in}(t)$ not present in the DC, then it may result in a shift by m (length of $W_{in}(t)$). Therefore, the bad character heuristic takes $O\left(\frac{n}{m}\right)$ time in the best case and $O(nm)$ in the worst case. Algorithm 2 describes the pseudo-code of bad character heuristics.

Algorithm 2: Compute Last Occurrence Function $\left(W_{in}(t), m, \sum\right)$

```
$\forall a \in \sum$
**do** $\alpha[a] = 0$
**for** $j = 1$ *to m* **do**
$\alpha[W_{in}(t)[j]] = j$
*return* $\alpha$
```

ii. Good Suffix Heuristics (Algorithm 1, Line 6**):** To begin looking for the pattern $W_{in}(t)$ the algorithm uses the final character of the pattern. When a substring from the primary text matches a substring from the $W_{in}(t)$, it looks for further instances of the matched substring. It may also look for the prefix of a $W_{in}(t)$, which is the suffix of the primary $W_p(t)$. If not, it moves along the whole length of the $W_{in}(t)$ by m. Algorithm 3 describes the pseudo-code of good suffix heuristics. It can be described in the following steps:

Suppose $W_{in}(t)$ and $W_p(t)$ are aligned so that a substring $w_{in}^{s}(t)$ of $W_{in}(t)$ matches a suffix of $W_{in}(t)$, but the following left-to-right comparison yields a mismatch.

a. Then, if there is such a thing, locate the rightmost copy $w_{in}^{s'}(t)$ of $w_{in}^{s}(t)$ in $W_{in}(t)$, where $w_{in}^{s'}(t)$ is not a suffix of $W_{in}(t)$ and the character to the left of $w_{in}^{s'}(t)$ in $W_{in}(t)$ is different from the character to the left of $w_{in}^{s}(t)$ in $W_{in}(t)$. It is necessary to right-shift $W_{in}(t)$ such that the substring $w_{in}^{s'}(t)$ in $W_{in}(t)$ corresponds to the substring $w_{in}^{s}(t)$ in $W_p(t)$.
b. If $w_{in}^{s'}(t)$ does not exist, move $W_{in}(t)$ left end beyond $w_{in}^{s}(t)$ in $W_p(t)$ just enough so that the shifted pattern's prefix matches the suffix of $w_{in}^{s}(t)$ in $W_p(t)$.
c. Shift $W_{in}(t)$ to the right by m (length of $W_{in}(t)$) places if such a shift is not possible.
d. Whenever $W_{in}(t)$ is found in $W_p(t)$, we shift it by the smallest possible amount such that a correct prefix of the shifted $W_{in}(t)$ matches a suffix of the occurrence of $W_{in}(t)$ in $W_p(t)$.
e. Shift $W_{in}(t)$ by m places, or shift $W_{in}(t)$ past $w_{in}^{s}(t)$, if such a shift is not possible.

Algorithm 3: Compute Good Suffix Function $(W_{in}(t), m)$

```
$\gamma$ = Compute prefix function $(W_{in}(t))$
$W_{in}^{'}(t)$ = Reverse $W_{in}(t)$
$\gamma'$ = Compute prefix function $\left(W_{in}^{'}(t)\right)$
```

(Continued)

Algorithm 3: Continued

```
**for** $j = 0to\ m$ **do**
    $\beta[j] = m - \gamma[m]$
        **for** $l = 1to\ m$ **do**
            $j = m - \gamma'[L]$
        **if** $\beta[j] > 1 - \gamma'[L]$ **then**
            $\beta[j] = 1 - \gamma'[L]$
    *return* $\beta$
```

In Part B of Algorithm 1 (Lines 17–20), if no match can be discovered between $\boldsymbol{W_{in}}(\boldsymbol{t})$ and $W_p(t)$ then Global-LB will allocate the $\boldsymbol{W_{in}}(\boldsymbol{t})$ to a DC i^* considering different optimization factors such as the current renewable energy level, the dynamic price of electricity, the server load, and the bandwidth cost. In line 17, we compute energy consumption, green energy level, number of active and inactive servers, and bandwidth cost for all DCs in each discrete time slot t to select the DC i^* based on the lowest electricity and bandwidth costs with respecting the given set of constraints ((3.6a) to (3.6e)). After then, we assign the coming workload $\boldsymbol{W_{in}}(\boldsymbol{t})$ to DC i^* for processing (line 18). Finally, in line 19, we update the incoming workload, active, and inactive servers for all DCs in a discrete-time slot t for next incoming workload.

Running Time of Algorithm: The running time of the Boyer Moore algorithm in the worst case is $O(n+m)$ only if the $W_{in}(t)$ does not exist in the $W_p(t)$. When the $W_{in}(t)$ does occur in the $W_p(t)$, running time is $O(nm)$ in the worst case.

4 Experimental Setup

The rest of the research work is dedicated to a thorough analysis of the SMGLB's efficiency under simulated and real-world traces of incoming user requests, dynamic electricity prices, and on-site renewable energy generation, geo-distributed DCs. The numerical evaluation is described in the following sections:

- Workload arrival
- Datacenter features
- Prices of Electricity
- Green energy production
- Benchmark algorithms

4.1 Workload Arrival

We employ synthetic batch workloads modeled after actual queries made to Wikipedia for our experiments. The data we utilize includes, among other things, 10% of all queries made by users to Wikipedia over the one month between January 1 to January 30, 2020, UTC [7]. $T = 720$-time slots are produced from averaging the data across an hour. Since electricity prices change every hour, we utilize an interval of 1 h. Assuming that each user request uses 10% of a server's CPU, the workloads are normalized over many servers. The traces depict (see Fig. 2) the daily trend of normalized batch workloads in the actual world.

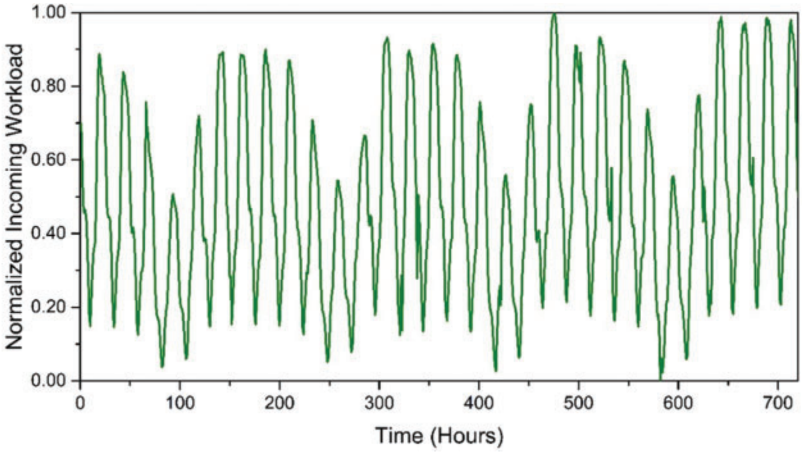

Figure 2: Hourly incoming workload

4.2 Data Center Features

We consider a single Global-LB serving $N = 3$ geographically dispersed DCs. According to assumptions, the DCs are in Utica, New York, Ontario, Canada, and Illinois, respectively. The Global-LB and DC bandwidth cost $B_i(t)$ is calibrated to increase city distance and to align with the cost of electricity [18,35]. Each DC i is assumed to have $S_i^{max} = 350$ accessible operational servers. Each server's power consumption rates in active and inactive states are predetermined to be $S_i^{in}(t) = 100\ W,\ \forall i$ and $S_i^{ac}(t) = 120\ W,\ \forall i$, respectively [8,36]. Each server's processing speed is assumed to be $\mu_i = 1$ [11]. Please note that to analyze the effects of other optimization factors (such as green energy availability, bandwidth cost, and electricity prices), we assume the homogeneous settings of the DCs. Additionally, the power usage effectiveness (PUE) is set to 1.2 in each DC i [36–38].

4.3 Prices of Electricity

We employ the day-ahead hourly prices of electricity $q_i(t)$ in \$ per MWh at the three DC sites as mentioned above. They come from publicly accessible government sources [36]. Fig. 3 depicts hourly electricity pricing at various locations. Electricity prices are considered from September 18 to October 17, 2017. We noticed that Ontario, Canada's electricity costs occasionally go down when there is a high and rigid energy supply, low electricity demand, and a high supply of renewable energy.

4.4 Green Energy Production

Green energy sources are flexible and incredibly sporadic. They might change throughout a single session (e.g., 1 h). Each DC i typically has energy storage devices (ESDs), that supply DCs with stable green energy [31]. As a result, we anticipated that throughout the one-time period ($t = 1$ h), on-site green energy generation would account for 18% of the total energy consumption of the DCs. Fig. 4 depicts the cumulative distribution functions (CDFs) of the green energy production of three geo-distributed DCs.

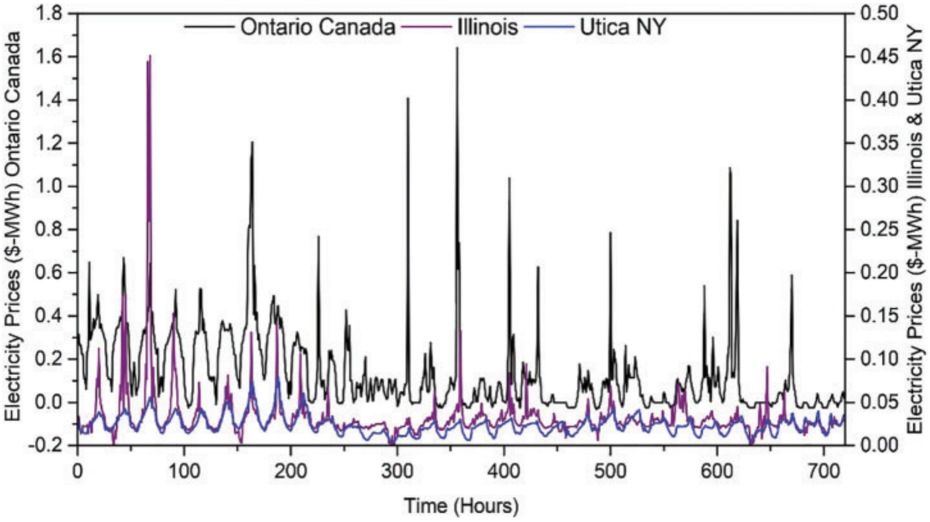

Figure 3: Hourly electricity price

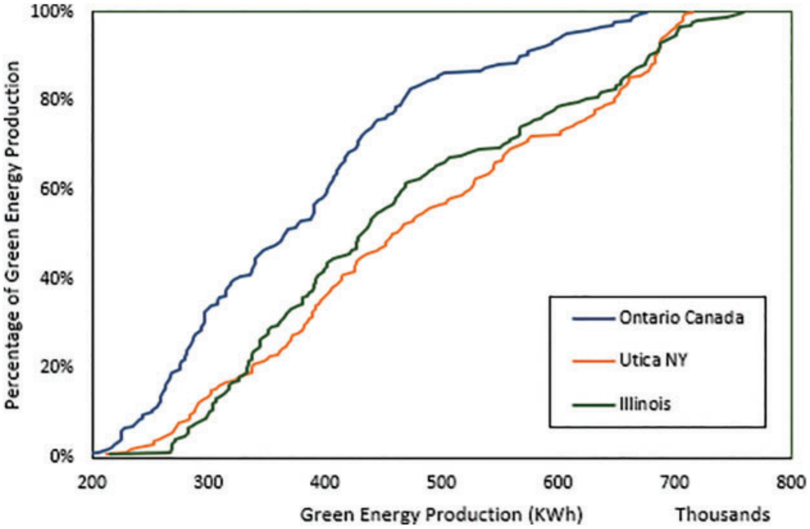

Figure 4: CDFs of the green energy production in DCs

4.5 Benchmark Algorithms

We evaluate SMGLB's performance to the three baselines listed below, which are either close to current practice or have recently been proposed, to establish benchmarks for the performance of SMGLB.

- **Benchmark 1 (B1)** [39]: The first benchmark, B1, uses a temperature-independent technique that considers $W_{in}(t)$ routing and capacity allocation. By setting a goal of lowering the total operating cost, it initially assigns capacity to batch jobs. The purpose is to resolve the request routing optimization using the remaining capacity. The sole variable employed is the variation in power prices, and a constant PUE of 1.2 is used to compute cooling energy. Despite being naive, such a method is frequently employed in current Internet services. Additionally, it permits an implicit evaluation of earlier efforts.

- **Benchmark 2 (B2)** [8]: The Round Robin (RR) method ensures that all geo-distributed DCs receive an equal share of $W_{in}(t)$ by setting equal weights for the global load balancer. A suitable benchmark policy to assess the efficacy of SMGLB is an RR strategy that fairly distributes $W_{in}(t)$ around DCs since Wikipedia workload and the green energy generated at each location follow a similar diurnal pattern.
- **Benchmark 3 (B3)**: We use suggested cost-aware dynamic provision (CDP) workload distribution strategy [40] as a third baseline to demonstrate the efficacy of the proposed algorithm. Remember that CDP can take advantage of the variation in power rates across different regions by routing $W_{in}(t)$. Green energy sources and batteries are ignored in this research work. The goal of CDP is to reduce the overall energy cost in each slot based on the observed system states, and the use of green energy makes for more fair comparisons.

5 Experimental Results

The experimental results under the conditions outlined in Section 4 are shown and discussed in this section.

5.1 Energy Cost Minimization Using String Matching Techniques

It is important to note that previous research studies mostly ignored string matching approaches for workload routing instead of focusing on lowering energy costs. With the use of delay-tolerant workloads, green power, and geographical load balancing, our algorithm SMGLB reduced energy consumption and associated costs. We first assume that the incoming workload $W_{in}(t)$ arrives at Global-LB with the objective to use a string-matching algorithm (i.e., Boyer Moore) to compare the contents of $W_{in}(t)$ to those documents that have already been processed $W_p(t)$ in a DC. A successful match prevents Global-LB from sending the $W_{in}(t)$ to a DC i for processing and displaying the results of the $W_p(t)$ to the user to save energy utilization. We choose $W_{in}(t)$ as an optimization factor in our benchmark algorithms so that the $d_i^Q(t)$ is identical in all benchmark algorithms if no match can be found by Global-LB because the performance of SMGLB depends on pattern matching for a fair comparison. Furthermore, the SMGLB dramatically reduces the DC's overall energy costs when the following parameter settings are used: string matching policy, availability of green energy, dynamic electricity rates, and GLB. The average operating cost in k of SMGLB and benchmark algorithms are shows in Table 1 and graphical representation are displayed in Fig. 5. The figure shows that SMGLB performs better than all benchmark algorithms. In particular, by contrasting SMGLB with B3, we can see that string matching can contribute to a decrease in the overall cost of power. Additionally, even though B2 considers the fluctuating power price over time, it is green energy oblivious and only tries to process $W_{in}(t)$ when the price of electricity is low. As a result, it performs poorly and wastes much green energy. This demonstrates the significance of workload distribution in DCs that have on-site green energy production. Finally, by contrasting B1 with B3, we observe that how $W_{in}(t)$may increase the usage of renewable power sources while decreasing cost of electricity in DCs.

Table 1: SMGLB improvement over benchmark algorithms

Comparison factor	SMGLB improvement over		
	B1	B2	B3
Operating cost	74.61%	83.67%	31.37%

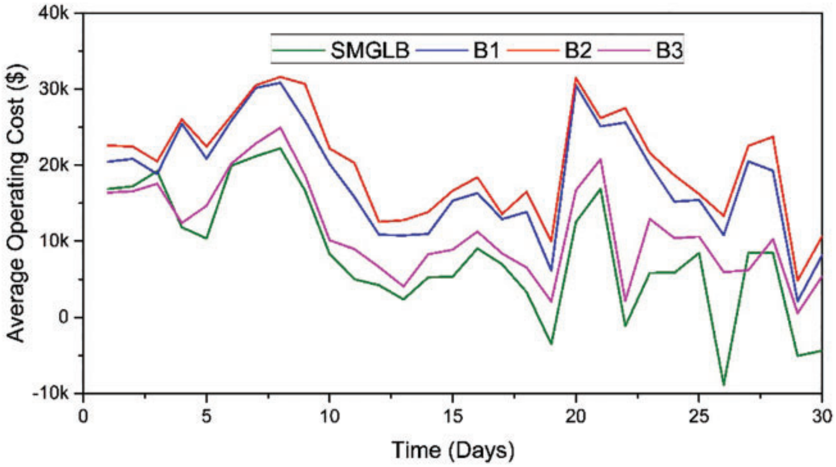

Figure 5: Average daily operating cost comparison between SMGLB and benchmark algorithms

Fig. 6 depicts the CDFs of the daily average operating cost of the SMGLB and benchmark algorithms. It shows the significant cost reduction of SMGLB over B1, B2, and B3. We can observe that SMGLB process approximately 100% of incoming workload between $(-10\ to\ 23)\ k\$$, while B3 $(0\ to\ 25)\ k\$$, B1 $(2\ to\ 30)\ k\$$, and B2 $(4\ to\ 31)\ k\$$, respectively. We noticed a negative operating cost in a few time slots. The Ontario, Canada, electricity rates cause negative energy costs. Remember that electricity prices go down when there is high energy production, low electricity demand, and a high supply of green energy.

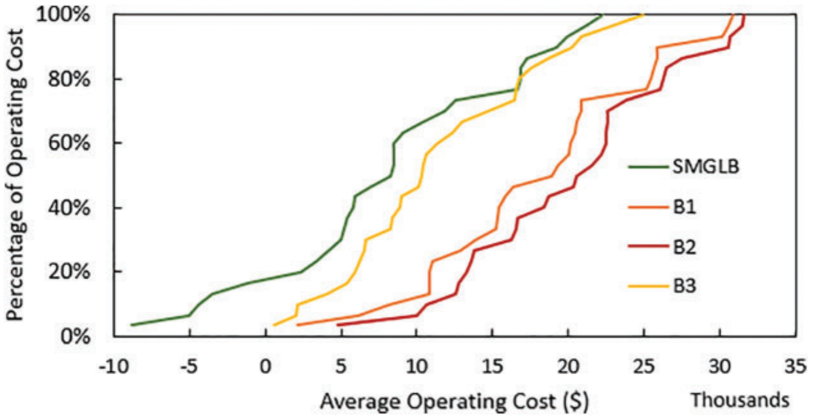

Figure 6: CDFs of the average operating cost of 30 days of algorithms

Table 1 summarizes and compares the increased efficiency of SMGLB to that of B1, B2, and B3. According to the data presented, SMGLB can boost performance by 74.61% compared to B1\percnt, 83.67% to B2%, and 31.37% to B3. Therefore, SMGLB significantly reduces the geo-distributed DCs\rsquo overall operating cost.

5.2 Impact of Energy Storage Devices Cost

In order to ascertain how the cost of energy storage devices affects operating cost reductions, we establish the parameters V and S_i^{max} for all DCs and evaluate SMGLB under various values of the loss of energy storage devices ω_i. The result is shown in Fig. 7. As the cost of energy storage devices increases, we can see that the operating cost savings diminish. When ω_i is high, SMGLB does not utilize energy storage mechanisms. However, even in this case, the GLB and string-matching-based workload load distribution still results in cost reductions when compared to B1 and B2.

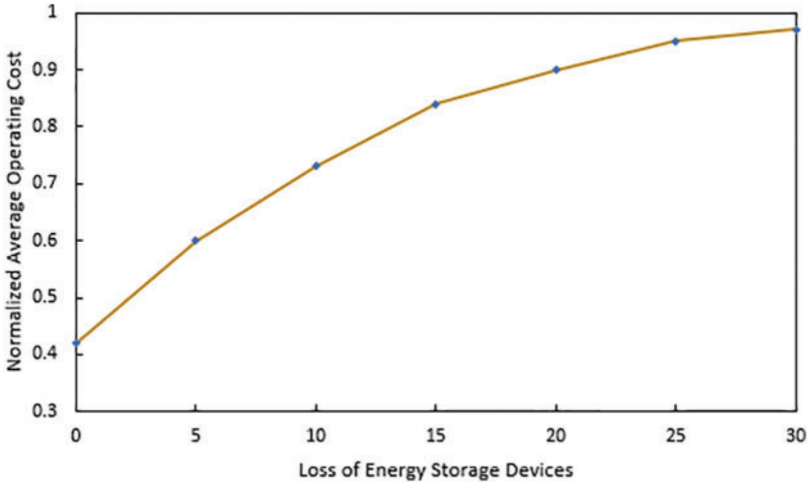

Figure 7: Impact of ESDs on the operating cost of the geo-distributed DC

5.3 Bandwidth Cost Savings

Now, we evaluate the bandwidth requirements of the abovementioned benchmark algorithms in contrast to our SMGLB algorithm. Assuming that B1 consistently distributes $W_{in}(t)$ to the closest DCs and bandwidth cost is negligible. On average, the other three algorithms cost more to run, as seen in Figs. 5 and 6. Our approach provides the most significant possible reduction in overall operating expenses by accounting for the disparity in bandwidth prices between Global-LB and DCs, as shown in the Fig. 5 above. It is essential to remember that B1 and B3 have comparable operational costs, despite B1's lower bandwidth cost and more significant energy cost than B3. Although B2 uses the least bandwidth, its energy costs are the greatest.

6 Conclusion

Large-scale commercial DCs have proliferated rapidly with the spread of cloud computing. Many CSPs frequently employ geographically dispersed DCs to guarantee customer service reliability and quality. Because of the massive volume of data being processed, energy efficiency is becoming a contemporary problem for geo-distributed DCs. Many research works have employed GLB and dynamic energy prices to process delay-tolerant workloads to lower the operating costs of geo-distributed DCs, but many issues remain unsolved. This motivation lies behind the fact that most prior efforts in processing incoming workloads have concentrated on minimizing these costs. To address the new problem of lowering the operational cost for geographically dispersed DCs in a multi-electricity market setting, we devise a cost-effective method based on the string-matching approach (SMGLB). The aim of SMGLB is twofold; first, we used a string-matching algorithm (i.e., Boyer Moore) to compare the contents of incoming user requests to those documents that have already been processed in a DC. A successful match prevents Global-LB from sending the workload to a DC for processing and displaying the result to the user to save energy utilization. Secondly, suppose no match can be discovered; in that case, Global-LB will allocate the workload to a DC, taking into account different optimization factors such as the current renewable energy level, the dynamic price of electricity, and the server load to mitigate the operating cost of the DC. Extensive evaluations using real-life data show that the SMGLB brings significant energy cost savings over existing workload distribution strategies.

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

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

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

References

[1] N. Hogade, S. Pasricha and H. J. Siegel, “Energy and network aware workload management for geographically distributed data centers,” *IEEE Transactions on Sustainable Computing*, vol. 7, no. 2, pp. 400–413, 2021.

[2] D. Jiang, Y. Wang, Z. Lv, W. Wang and H. Wang, “An energy-efficient networking approach in cloud services for IIoT networks,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 928–941, 2020.

[3] V. S. Shekhawat, A. Gautam and A. Thakrar, “Datacenter workload classification and characterization: An empirical approach,” in *IEEE 13th Int. Conf. on Industrial and Information Systems (ICIIS)*, Rupnagar, India, pp. 1–7, 2018.

[4] A. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat *et al.,* “Fog computing for smart grid systems in the 5G environment: Challenges and solutions,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 47–53, 2019.

[5] M. A. Albreem, A. M. Sheikh, M. H. Alsharif, M. Jusoh and M. N. M. Yasin, “Green internet of things (GIoT): Applications, practices, awareness, and challenges,” *IEEE Access*, vol. 9, pp. 38833–38858, 2021.

[6] V. D. Chakravarthy and B. Amutha, “A novel software-defined networking approach for load balancing in data center networks,” *International Journal of Communication Systems*, vol. 35, no. 2, pp. e4213, 2022.

[7] I. Ahmad, M. I. K. Khalil and S. A. A. Shah, “Optimization-based workload distribution in geographically distributed data centers: A survey,” *International Journal of Communication Systems*, vol. 33, no. 12, pp. e4453, 2020.

[8] M. I. K. Khalil, I. Ahmad and A. A. Almazroi, “Energy efficient indivisible workload distribution in geographically distributed data centers,” *IEEE Access*, vol. 7, pp. 82672–82680, 2019.

[9] J. A. Jeba, S. Roy, M. O. Rashid, S. T. Atik and M. Whaiduzzaman, “Towards green cloud computing an algorithmic approach for energy minimization in cloud data centers,” *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing: IGI Global*, vol. 24, pp. 846–872, 2021.

[10] A. Shakarami, M. Ghobaei-Arani, A. Shahidinejad, M. Masdari and H. Shakarami, “Data replication schemes in cloud computing: A survey,” *Cluster Computing*, vol. 24, no. 3, pp. 2545–2579, 2021.

[11] M. I. K. Khalil, I. Ahmad, S. A. A. Shah, S. Jan and F. Q. Khan, “Energy cost minimization for sustainable cloud computing using option pricing,” *Sustainable Cities and Society*, vol. 63, pp. 102440, 2020.

[12] A. Javadpour, A. M. H. Abadi, S. Rezaei, M. Zomorodian and A. S. Rostami, “Improving load balancing for data-duplication in big data cloud computing networks,” *Cluster Computing*, vol. 25, no. 4, pp. 2613–2631, 2022.

[13] F. T. Chong, M. J. R. Heck, P. Ranganathan, A. A. M. Saleh and H. M. G. Wassel, “Data center energy efficiency: Improving energy efficiency in data centers beyond technology scaling,” *IEEE Design & Test*, vol. 31, no. 1, pp. 93–104, 2013.

[14] S. Bird, A. Aijit, A. M. Othman, H. Wenjin, J. Kerop *et al.,* “Distributed (green) data centers: A new concept for energy, computing, and telecommunications,” *Energy for Sustainable Development*, vol. 19, pp. 83–91, 2014.

[15] G. Koutitas and P. Demestichas, “Challenges for energy efficiency in local and regional data centers,” *Journal of Green Engineering*, vol. 1, no. 1, pp. 1–32, 2010.

[16] M. Ernst, R. Hogue, C. Hollowell, W. Strecker-Kellog, A. Wong *et al.,* “Operating dedicated data centers–is it cost-effective?,” *Journal of Physics*, vol. 51, pp. 062053, 2014.

[17] R. Dhaya, R. Kanthavel and K. Venusamy, “Dynamic secure and automated infrastructure for private cloud data center,” *Annals of Operations Research*, vol. 25, pp. 1–21, 2021.

[18] Y. Guo, Y. Gong, Y. Fang, P. P. Khargonekar and X. Geng, “Energy and network aware workload management for sustainable data centers with thermal storage,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2030–2042, 2013.

[19] H. Lei, R. Wang, T. Zhang, Y. Liu and Y. Zha, “A Multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center,” *Computers & Operations Research*, vol. 75, pp. 103–117, 2016.

[20] L. Yu, T. Jiang and Y. Cao, “Energy cost minimization for distributed internet data centers in smart microgrids considering power outages,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 120–130, 2014.

[21] C. Gu, Z. Li, C. Liu and H. Huang, “Planning for green cloud data centers using sustainable energy,” in *IEEE Symp. on Computers and Communication (ISCC)*, Messina, Italy, pp. 804–809, 2016.

[22] T. Chen, A. G. Marques and G. B. Giannakis, “DGLB: Distributed stochastic geographical load balancing over cloud networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 1866–1880, 2016.

[23] A. Rahman, X. Liu and F. Kong, “A survey on geographic load balancing-based data center power management in the smart grid environment,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 214–233, 2013.

[24] J. Shuja, A. Gani, S. Shamshirband, R. W. Ahmad and K. Bilal, “Sustainable cloud data centers: A survey of enabling techniques and technologies,” *Renewable and Sustainable Energy Reviews*, vol. 62, pp. 195–214, 2016.

[25] Z. Liu, M. Lin, A. Wierman, S. H. Low and L. L. H. Andrew, “Greening geographical load balancing,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 193–204, 2011.

[26] M. Lin, Z. Liu, A. Wierman and L. L. H. Andrew, “Online algorithms for geographical load balancing,” in *Int. Green Computing Conf. (IGCC)*, IEEE, San Jose, CA, USA, pp. 1–10, 2012.

[27] M. A. Abdelghany, H. Mohsenian-Rad and M. Alizadeh, “Wholesale electricity pricing in the presence of geographical load balancing,” in *Asilomar Conf. on Signals, Systems, and Computers*, IEEE, Pacific Grove, CA, USA, pp. 653–658, 2017.

[28] D. Paul, W. -D. Zhong and S. K. Bose, “Energy efficiency aware load distribution and electricity cost volatility control for cloud service providers,” *Journal of Network and Computer Applications*, vol. 59, pp. 185–197, 2016.

[29] A. N. Toosi, C. Qu, M. D. de Assunção and R. Buyya, “Renewable-aware geographical load balancing of web applications for sustainable data centers,” *Journal of Network and Computer Applications*, vol. 83, pp. 155–168, 2017.

[30] X. Lu, F. Kong, X. Liu, J. Yin, Q. Xiang *et al.,* “Bulk savings for bulk transfers: Minimizing the energy-cost for geo-distributed data centers,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 73–85, 2017.

[31] Í. Goiri, M. E. Haque, K. Le, R. Beauchea, T. D. Nguyen *et al.,* “Matching renewable energy supply and demand in green datacenters,” *Ad Hoc Networks*, vol. 25, pp. 520–534, 2015.

[32] C. Qu, R. N. Calheiros and R. Buyya, “Mitigating impact of short-term overload on multi-cloud web applications through geographical load balancing,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 12, pp. e4126, 2017.

[33] A. Z. M. Saleh, N. A. Rozali, A. G. Buja, K. A. Jalil, F. H. M. Ali *et al.,* “A method for web application vulnerabilities detection by using Boyer-Moore string matching algorithm,” *Procedia Computer Science*, vol. 72, pp. 112–121, 2015.

[34] D. Cantone and S. Faro, “Fast-search: A new efficient variant of the Boyer-Moore string matching algorithm,” in *Int. Workshop on Experimental and Efficient Algorithms*, Springer, Berlin, Germany, pp. 47–58, 2003.

[35] Y. Peng, D. -K. Kang, F. Al-Hazemi and C. -H. Youn, “Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters,” *Optical Switching and Networking*, vol. 23, pp. 225–240, 2017.

[36] M. I. K. Khalil, S. A. A. Shah, A. Taj, M. Shiraz, B. Alamri *et al.,* “Renewable-aware geographical load balancing using option pricing for energy cost minimization in data centers,” *Processes*, vol. 10, pp. 1983, 2022.

[37] H. Hampton and A. Foley, “A review of current analytical methods, modelling tools and development frameworks applicable for future retail electricity market design,” *Energy*, vol. 260, pp. 124861, 2022.

[38] F. Khan, M. A. Jan and M. Alam, “Big data in healthcare: A survey,” in *Applications of Intelligent Technologies in Healthcare*, 1st ed., vol. 14, Bern, Switzerland: Springer Cham, pp. 143–152, 2019.

[39] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag and B. Maggs, “Cutting the electric bill for internet-scale systems,” in *Proc. of the ACM SIGCOMM 2009 Conf. on Data Communication*, New York, NY, USA, pp. 123–134, 2009.

[40] L. Rao, X. Liu, L. Xie and W. Liu, “Coordinated energy cost management of distributed internet data centers in smart grid,” *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 50–58, 2011.