Smartly Handling Renewable Energy Instability in Supporting A Cloud Datacenter

Jiechao Gao, Haoyu Wang and Haiying Shen
Department of Computer Science
University of Virginia
Charlottesville, VA, USA
{jg5ycn, hw8c, hs6ms}@virginia.edu

***Abstract*—The size and energy consumption of datacenters have been increasing significantly over the past years. As a result, datacenters' increasing electricity monetary cost, energy consumption and energy harmful gas emissions have become a severe problem. Renewable energy supply is widely seen as a promising solution. However, the instability of renewable energy brings about a new challenge since insufficient energy supply may lead to job running interruptions or failures. Though previous works attempt to more accurately predict the amount of produced renewable energy, due to the instability of its influencing factors (e.g., wind, temperature), sufficient renewable energy supply cannot be always guaranteed. To handle this problem, in this paper, we propose allocating jobs with the same service-level-objective (SLO) level to the same physical machine (PM) group, and power each PM group with renewable energy generators that have probability no less than its SLO to produce the amount no less than its energy demand. It ensures that insufficient renewable energy supply will not lead to SLO violations. We use a deep learning technique to predict the probability of producing amount no less than each value of each renewable energy source and predict the energy demands of each PM area. We formulate an optimization problem: how to match renewable energy resources with different instabilities to different PM groups as energy supply in order to minimize the number of SLO violations (due to interruption from insufficient renewable energy supply), total energy monetary cost and total carbon emission. We then use reinforcement learning method and linear programming method to solve the optimization problem. The real trace driven experiments show that our method can achieve much lower SLO violations, total energy monetary cost and total carbon emission compared to other methods.**

I. Introduction

Over the past years, more and more Internet services (e.g., e-commerce, content distribution, gaming, and social networking) have been deployed over the cloud datacenters, which are reliable, elastic, and cost-effective. Consequently, the size and energy consumption of datacenters have been increasing significantly. As a result, datacenters' increasing electricity monetary cost, energy consumption and energy harmful gas emissions have become a severe problem to the society. The average energy consumed by each datacenter is almost equal to the energy consumed by 25,000 homes in the U.S. [1] . For example, the Microsoft datacenter in Quincy, Washington consumes 48 megawatts which is enough to power 40,000 homes [2]. U.S. datacenters are projected to consume approximately 73 billion kWh in 2020 [3]. A 3% reduction in energy cost for a large company like Google can translate into over a million dollars in cost savings [4]. A large amount of datacenters around the world are powered by electricity generated by brown energy such as fuel fossil, coal and oil. The total volume of carbon emission will be 1430 Million metric tons, and datacenters will be responsible for 18% CO2 of emission in the entire world by 2020 [5].

To solve this problem, governments in many countries began to set up laws and regulations to brown energy utilization. The government and the Environmental Protection Agency (EPA) will punish the datacenters with severe fines based on the carbon emission [6]. As a solution, cloud service providers start using renewable energy such as solar, wind and hydro to power the cloud datacenters. For example, Apple has recently completed a massive 100-acre solar farm next to its iCloud datacenter in North Carolina, which will supply 84 million kWh of clean renewable energy annually [7].

In the future, a system with thousands of active consumers who own solar and wind energy generators and have an ability to sell the renewable energy is envisioned [8]. Consider the scenario with geo-distributed renewable energy sources (or generators), previous research works [9]–[15] focus on how to schedule jobs to the datacenters to use these renewable energy resources to minimize either energy monetary cost or carbon emission while satisfying time latency constraints of jobs. These methods predict the amount of generated renewable energy from each generator powering a specific datacenter, and the energy demands from jobs, and then schedule the jobs to different datacenters to achieve the goals.

However, renewable energy resources are featured by instability. For example, the amount of produced solar energy depends on solar irradiance, and the amount of produced wind energy depends on wind turbines; both factors depend on the time of a day, the season, the climate and so on. The energy amount instability brings about a new challenge since insufficient energy supply may lead to job running interruptions or failures as datacenters require uninterruptible energy supply to minimize the downtime for servers [16]. Though the previous works attempt to more accurately predict the amount of produced renewable energy, due to the energy instability, 100% accuracy may not be achieved, which means that sufficient renewable energy supply cannot be always guaranteed. *There has been no research effort on studying how to choose renewable energy generators to power a datacenter to mitigate the adverse effect on the datacenter jobs due to*

1530-2075/20/$31.00 ©2020 IEEE
DOI 10.1109/IPDPS47924.2020.00084

IEEE computer society

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

overestimation of the amount of produced renewable resources from the generators.

This paper is the first to handle this problem. In this paper, we consider that a job's service-level-objective (SLO) is specified by its successful running probability. To reduce the SLO violations due to overestimation of the amount renewable resources, we propose a method. It allocates jobs with the same SLO level to the same physical machine (PM) group, and power the PM group with an SLO=$x\%$ and predicted energy demand of ykWh with the renewable energy generator that is predicted to generate no less than ykWh amount with no less than $x\%$ probability at each time slot in the next time period.

Our system runs after each time period and schedule which energy generator will supply energy to which PM area in the next time period in order to minimize the number of SLO violations (due to interruption from insufficient renewable energy), total energy monetary cost and total carbon emission. It has three steps. First, we use long short term memory (LSTM) deep learning model to predict the tail distribution of the amount of generated renewable energy of each energy source represented, i.e., the probability that it will generate no less than each certain amount of energy at each time slot of the next time period. Second, we use LSTM to predict the energy demand of each PM area. Third, we allocate renewable energy generators to the PM areas based on the aforementioned rule. That is, if an energy source's generated renewable energy amount is no less than the PM area energy demand, and the probability of producing that amount is no less than the SLO value of the PM areas, it can be a candidate to be assigned to the PM area. Specifically, we use a reinforcement learning (RL) method and a linear programming method to solve the aforementioned problem.

We conduct comprehensive real trace driven experiments to compare our method with other three methods in terms of SLO satisfactory ratio, total energy monetary cost and total carbon emission. The experimental results show that our methods can achieve much lower number of SLO violations, total energy monetary cost and total carbon emission compared to the other methods. Our methods function as a complement to previous renewable energy management methods to reduce the number of SLO violations.

The rest of the paper is organized as follows. Section II presents the related work. Section III presents the system design. Section IV presents the performance evaluation of our system. Section V concludes the paper with remarks on our future work.

II. RELATED WORK

We classify the related work into two parts: energy-efficient resource management and renewable energy management.

Energy-Efficient Resource Management. Reducing the number of running servers is a common approach to reduce energy consumption of a datacenter.

Chen *et al.* [17] proposed algorithms to minimize the number of running servers via dynamically distributing workload to the servers, which saves up to 30% energy. Heller *et al.* [18] introduced ElasticTree, an energy manager with a focus on the datacenter network elements (links and switches). It monitors traffic conditions in the datacenter, and simply turns off the switches and links if they are not needed. Lin *et al.* [19] proposed an Energy-Efficient Adaptive File Replication System (EAFR). EAFR decreases the number of replicas for cold files without compromising their read efficiency, stores the cold files to servers and put these servers to the sleep mode to save energy. Dabbagh *et al.* [20] developed a framework for predicting future virtual machine requests and associated resource requirements. It puts unneeded machines into the sleep mode to reduce energy consumption.

Renewable Energy Management. Given a number of geo-distributed datacenters, with each datacenter being powered by certain renewable energy sources, several methods have been proposed to reduce total energy monetary or carbon emission.

The method in [13] aims to minimize monetary cost while giving higher priority to using renewable energy via rescheduling (or migrating) jobs between datacenters using RL based on neural network (NN) model. The method in [9] uses an integer linear programming method to allocate jobs to the different datacenters to minimize the carbon emissions of the datacenters by using renewable energy while satisfying: (1) the request processing time constraint; (2) the total electricity budget in each time slot; (3) the intermittent supply of the renewable resources; (4) the maximal number of servers in each datacenter. Since different types of energy resources have different carbon emission rate, the method prefers to use the energy resource with minimal carbon emission rate as much as possible. This method assumes that the amount of generated energy always follows the same pattern, and uses pattern-based method to predict the amount of generated renewable energy in each energy source.

Liu *et al.* [11] proposed an integrated workload management system for one datacenter. Since different renewable energy sources have dynamic energy generation and price through time, the system tries to minimize the monetary cost by scheduling jobs to different time slots while satisfying job processing time constraint and using solar energy as much as possible since it is easy to predict. It uses the k-nearest neighbor (k-NN) method to predict renewable energy and energy demand of each node based on historical data.

De Courchelle *et al.* [14] proposed a job scheduling method for a datacenter aiming to use renewable energy as much as possible.

Gu *et al.* [12] aim to minimize the brown energy usage via task allocation and renewable energy scheduling in edge computing. Given a set of jobs which need to run on the nodes, the method first schedules all the jobs on different nodes to minimize the total energy consumption of all the nodes. Considering the variance of renewable energy, the method schedules the renewable energy to different nodes, which tries to fully utilize renewable energy rather than brown energy. Finally, the authors proved that the problems above are NP-hardness and proposed a mixed integer linear programming to solve the problems.

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

Due to renewable energy instability, the above works may overestimate the amount of produced renewable resource from a generator. Different from the above works, our work is the first that handles how to choose renewable energy generators to power a datacenter to mitigate the adverse effect on the datacenter jobs from overestimation of the amount of produced renewable resources from the generators. It can complement other renewable energy management methods to reduce the number of SLO violations due to insufficient renewable energy supply.

III. System design

A. Background and Research Problem

Different renewable energy resources are influenced by different natural features [21], [22]. Solar energy is influenced by solar irradiance, and wind energy is influenced by wind speed. The renewable energy resources are featured by instability due to the environment and climate change. For example, in summer sunny days, solar can generate more stable energy resource than wind, and it can generate more energy at the daytime than that at night. In winter cloudy days, wind can generate more stable energy resource than solar. This instability feature brings a challenge when we use renewable energies as energy supply in local datacenters. To handle this problem, previous research attempts to increase the prediction accuracy of the produced energy amount, e.g., by using pattern match or machine learning methods, it still cannot guarantee 100% prediction accuracy due to the instability. In addition, prediction at a higher frequency is needed due to the instability, which generates high computation overhead. Therefore, renewable energy sources sometimes may not generate enough energy as predicted to power the datacenters.

In this case, a datacenter can use brown energy from the energy grid. However, it degrades the performance of achieving the goals and the energy supply switch may interrupt job running. Our goal is to mitigate the adverse effect on the running jobs from insufficient supplied renewable energy even if the datacenter does not resort to the brown energy supply.

We denote G_k as the k^{th} renewable energy resource, $g_{G_k,t}$ as the predicted energy generation amount of renewable energy resource G_k at time t. Complementary cumulative distribution function or simply tail distribution represents the probability distribution that the amount of the generated renewable energy is no less than each certain amount. We use P_{G_k,t_n} to denote the tail distribution of renewable energy resource G_k at t_n time.

$$P_{G_k,t_n} = \{< g^1_{G_k,t_n}, p(g^1_{G_k,t_n}) >, < g^2_{G_k,t_n}, p(g^2_{G_k,t_n}) >, ...\} \quad (1)$$

where $p(g^i_{G_k,t_n})$ means the probability that the amount of the generated energy is no less than $g^i_{G_k,t_n}$.

We consider that SLO specifies the deadline of a job and the maximum allowed probability of the job being completed beyond the deadline, or simply specifies the successful running probability of a job [23]. For example, a job must complete with a success probability of 99.9%. Note that different jobs have different SLOs. To avoid a job's SLO violation due to insufficient supplied renewable energy, we can assign the job the renewable energy generator that has probability no less than the SLO to produce the energy amount no less than the job's energy demand. A problem here is how to conduct such a mapping since energy is supplied to PMs rather than jobs. To handle this problem, we propose to divide PMs to PM areas, and each PM area host jobs with one SLO value. Then, we supply each PM area with renewable energy sources to avoid SLO violations. Note that when we allocate jobs to the PM areas, we need to consider the constraint of PM computing resource capacity and try to consolidate jobs to as few PMs as possible to save energy. We can reply on previous methods (e.g., [24]) for these purposes, which are out of the scope of this paper.

Specifically, our renewable energy resource assignment **problem** is as follows:

Given a datacenter and many geo-distributed renewable energy sources (or generators) that the datacenter can use, how the renewable energy sources should be mapped to the PM areas in order to minimize the number of SLO violations (due to interruption from insufficient renewable energy), total energy monetary cost and total carbon emission?

To handle this problem, we propose a system that conducts such mapping periodically (e.g., every hour). It incorporates the following three components.

(1) It predicts the tail distribution of each renewable energy source at each time slot in the next time period (Section III-B).
(2) It predicts the energy demand in each PM area by predicting the CPU utilization for each PM in a PM area (Section III-C).
(3) Based on the predicted renewable energy generation and predicted energy demand, it assigns renewable energy sources to PM areas to solve the above problem using RL-based method and linear programming method (Section III-D).

In this paper, we assume that a PM area has enough computing resources for the jobs with the corresponding SLO. We present each of the system components in the following.

B. Prediction for Renewable Energy Generation

We assume that our system conducts the mapping between renewable energy sources and PM areas every one hour, though it can be any time length. For each renewable energy source, it predicts the amount of generated energy every time slot t_i (e.g., 5 minutes) within the next hour, and for each PM area, it predicts the amount of energy demand every time slot within the next hour. Then, it conducts the mapping to make sure that the renewable energy sources mapped to a PM area will satisfy its energy demand at each time slot t_i within the next hour. That is, if a PM area demands ykWh at t_i and its SLO level is $x\%$, the matched renewable energy source must produce no less than ykWh with probability no less than $x\%$ at time t_i for each time slot within the next hour. Such a mapping strategy is to avoid SLO violation due to insufficient renewable energy supply.

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

Therefore, for each renewable energy source, we need to predict the tail distribution of each renewable energy source, e.g., the probability of generated energy amount no less than 1kWh is 95%, no less than 2kWh is 92%, and so on. To do this, we use the long short term memory (LSTM) deep learning model [25], [26] since it is effective in handling time series data and can observe the correlation between different time slots. The inputs of LSTM include a set of time sequence data, which records the historical amount of generated energy of a renewable energy source, and the factors affecting the amount of the renewable energy resource (e.g., solar irradiance for solar energy, wind speed for wind energy), and the output of LSTM is the tail distribution at each t_i in the next hour. We explained the tail distribution of energy resource G_k at time t_n (P_{G_k,t_n}) in the above. We use P_{G_k} to denote the tail distribution of renewable energy generator P_{G_k} for the next time period.

$$P_{G_k} = \{P_{G_k,t_1}, P_{G_k,t_2}, ..., P_{G_k,t_n}, ..., P_{G_k,t_N}\}, \quad (2)$$

where N is the number of time slots in the periodical time period T.

C. Prediction of Energy Demand

We need to predict the amount of energy consumption in each PM area at each t_n in the next hour. For each PM m, its energy consumption at time slot t_n (denoted by C_{m,t_n}) can be calculated as follows [27]:

$$C_{m,t_n} = C_{idle} + \varphi \times U^{cpu}_{m,t_n} \quad (3)$$

where U^{cpu}_{m,t_n} represents the CPU utilization of PM m, and φ is a calibrated constant, which is determined by the commercial model of the PM. C_{idle} represents the energy consumption of other types of resources such as memory or the energy consumption when the active PM is not being utilized, and can be a constant according to [28]. We use the LSTM deep learning technique for predicting the CPU utilization of a PM due to the aforementioned reasons. The historical CPU utilization time series data of each PM is used as the input and the output is the CPU utilization for a PM at each t_i in the next time period. Then, we calculate the energy consumption of each PM based on the predicted CPU utilization according to Equation (3). Finally, we add the energy consumption of each PM in a PM area to predict the total amount of the energy demand in this PM area.

D. Mapping Renewable Energy Sources and PM Areas

After we predict the tail distribution of the amount of generated energy in each renewable energy source, and the energy demand of each PM area, we need to map the renewable energy sources to the PM areas for energy supply to solve the problem in Section III-A. In this paper, we use two methods to solve this problem. First, we formulate this problem as a Markov Decision Process (MDP), and use a reinforcement learning (RL) method based on Deep Q-Network (DQN) [29] to solve the MDP problem [30]–[32]. Second, we formulate this problem as an optimization problem and then use integer linear programming approach to solve it. We present each method in the following.

1) RL-based Method: We first formulate this problem as a Markov Decision Process (MDP), denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is the state, $\mathcal{A}$ is the action, $\mathcal{P}$ is the probability between each two states and $\mathcal{R}$ is the reward. Below, we introduce the elements in the MDP for our problem.

a) State Space: The state space $\mathcal{S}$ is defined as the input of the RL model, and it consists of the information of renewable energy sources and PM areas in a datacenter.

We denote the renewable energy generators by:

$$\mathcal{G} = \{G_1, G_2, ...G_k, ...G_K\} \quad (4)$$

where G_k means the k^{th} renewable energy generator and K means the total number of renewable energy generators. The tail distribution at each time for the next time period (e.g., one hour) is denoted by $\mathcal{P}_G$.

$$\mathcal{P}_G = \{P_{G_1}, P_{G_2}, ..., P_{G_k}, ..., P_{G_K}\}. \quad (5)$$

The distance $\mathcal{D}_G$ vector indicates the distance between each energy source and the datacenter:

$$\mathcal{D}_G = D_{G_1}, D_{G_2}, ...D_{G_k}, ...D_{G_K} \quad (6)$$

The unit price vector indicates the unit price of each energy source at each time slot:

$$\mathcal{C}_G = C_{G_1}, C_{G_2}, ..., C_{G_k}, ..., C_{G_K}, \ where \quad (7)$$

$$\mathcal{C}_{G_k} = \{c_{G_k,t_1}, c_{G_k,t_2}, ..., c_{G_k,t_n}, ...c_{G_k,t_N}\}, \quad (8)$$

where c_{G_k,t_n} denotes the energy price of generator G_k at time slot t_n. As in [13], we assume that the unit price at each time slot of an energy resource is pre-known. If it is not pre-known, we can also use LSTM for the price prediction based on the factors influencing the price.

The features of the PM areas in the cloud datacenter include the predicted energy demand of each PM area at each time slot in the next time period and the SLO of each PM area. We use M_j to denote the j^{th} PM area in the cloud datacenter, use E_{M_j,t_n} to denote the predicted energy demand of the j^{th} PM area at time t_n, and use J to denote the total number of PM areas in the datacenter. We use $\mathcal{E}_M$ to denote the vector for the predicted energy demand of each PM area in the cloud datacenter, and use E_{M_j} to denote the predicted energy demand of PM area M_j at each time slot in the next time period.

$$\mathcal{E}_M = \left\{E_{M_1}, E_{M_2}, ..., E_{M_j}, ...E_{M_J}\right\}, \ where \quad (9)$$

$$E_{M_j} = \left\{E_{M_j,t_1}, E_{M_j,t_2}, ..., E_{M_j,t_n}, ..., E_{M_j,t_N}\right\} \quad (10)$$

where E_{M_j,t_n} denotes the predicted energy demand of PM area M_j at time slot t_n.

We use $\mathcal{L}_M$ to denote the vector of the SLO level of each PM area in the cloud datacenter.

$$\mathcal{L}_M = \left\{L_{M_1}, L_{M_2}, ...L_{M_j}, ...L_{M_J}\right\} \quad (11)$$

where L_{M_j} is the SLO of the j^{th} PM area. Thus, the whole state space can be defined as follow:

$$\mathcal{S} = \{S = (\mathcal{P}_G, \mathcal{D}_G, \mathcal{C}_G, \mathcal{E}_M, \mathcal{L}_M)\}. \quad (12)$$

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

b) Action Space: The action space is defined as the assignment plans to assign renewable energy sources to the PM areas as energy supply. We use $a_{k,j}$ to denote a binary variable; $a_{k,j} = 1$ means that renewable energy source G_k is assigned to PM area M_j, and $a_{k,j} = 0$ means otherwise. Action space $\mathcal{A}$ is expressed in the following:

$$\mathcal{A} = \{(A_1, A_2, ...A_i, ...A_{K(J+1)}) \,|A_i = (a_{G_1,M_1}, a_{G_1,M_2}, ...a_{k,j}, ...a_{G_K,M_J})\} \tag{13}$$

$$i \in \{1, 2, ...K(J+1)\}, k \in \{1, 2, ...K\}, j \in \{1, 2, ...J\} \tag{14}$$

where A_i is the i^{th} action, $A_{K(J+1)}$ means that there are total $K(J+1)$ actions in the action space because each renewable energy source has the chance to be assigned to each PM area or not.

c) Probability: When a decision about which energy source is assigned to which PM area is made, the state is changed with certainty, so the probability between states is always 1.

d) Reward: Based on the problem in Section III-A,we consider the following factors in the reward function.

Monetary cost: The unit price of energy source G_k at time t is denoted by $c_{G_k,t}$. The monetary cost for purchasing $g_{G_k,t}$ amount of energy from energy source G_k at time t to be used by the j^{th} PM area (denoted by $C_{j,k,t}$) is calculated by

$$C_{j,k,t} = c_{G_k,t} \cdot g_{G_k,t} \tag{15}$$

Carbon emission: The amount of carbon emission per kWh of energy source G_k at time t is denoted by $w_{k,t}$, The total carbon emission for $g_{G_k,t}$ amount of energy from energy source G_k at time t is calculated by

$$W_{j,k,t} = w_{k,t} \cdot g_{G_k,t} \tag{16}$$

SLO violations: Avoiding SLO violations due to insufficient computing resources has been handled in previous research. This is not the focus of this paper and we directly use the previous methods for this purpose in this work. In this paper, we focus on avoid SLO violations due to interruptions from insufficient supplied renewable energy. To avoid SLO violations, when we map single source G_k to the j^{th} PM area, we need to ensure that $b_{G_k,t} = g_{G_k,t} - g_{G_k,t} \cdot \epsilon \cdot D_{G_k} \geq E_{M_j,t}$ and $p(g_{G_k,t}) \geq L_{M_j,t}$ for each time slot in the next time period, in which $g_{G_k,t} \cdot \epsilon \cdot D_{G_k}$ is the energy loss in energy transmission from the source to the PM area, and ϵ is the loss rate. This can be easily extended to the case when multiple energy sources are mapped to one PM area to meet its energy demand. If either of the above conditions is not satisfied, the running jobs in the PM area may be interrupted or failed and experience SLO violations. We use V_{A_i} to denote the number of SLO violations for action A_i; that is, the number of jobs running in the PM areas, where either of the above conditions is not satisfied. Our RL-based method is distinguishing in that it can directly use these conditions to measure a variable in the reward function for an action rather than collecting the resulting variable value by taking many actions in practice, which reduces training time.

Based on the aforementioned problem, we define the reward $\mathcal{R}$ for action A_i by the following equation:

$$\mathcal{R} = \frac{1}{\sum_{t \in T} \sum_{j \in J} \sum_{k \in K} (C_{j,k,t} + W_{j,k,t}) + V_{A_i}} \tag{17}$$

Our principle of the reward function is setting a higher reward for reducing the number of SLO violations, total energy monetary cost and total carbon emission.

e) Reinforcement Learning Training: To collect the training data of DQN, the datacenter initially can use the optimization solution from our linear programming method. We can also select the action randomly and calculate the reward from the selected action offline. The data is used for the DQN training to train the DQN network. The RL agent iteratively makes the decisions and updates the network parameters, which is the training process of DQN. After the DQN is trained, we deploy it in a centralized server where the trained RL agent runs and generates the mapping plan between the energy sources and PM areas periodically.

2) Optimization Problem based Method: Given the features in renewable energy resources and PM areas, represented by the state in Formula (12), we formulate the renewable energy resource assignment problem as follow:

$$Min \sum_{t \in T} \sum_{j \in J} \sum_{k \in K} a_{k,j} \cdot (C_{j,k,t} + W_{j,k,t}) \tag{18}$$

$$Subject\ to: \sum_{k \in K} a_{k,j} \geq 1, \forall j \in J \tag{19}$$

$$\sum_{k \in K} a_{k,j} \cdot b_{G_k,t} \geq E_{M_j,t}\ \&\ a_{k,j} \cdot p(g_{G_k,t}) \geq L_{M_j,t}, \quad \forall t \in T, \forall k \in K, \forall j \in J \tag{20}$$

Equation (18) aims to minimize the total energy monetary cost and total carbon emission. Equation (19) ensures that a PM area must have no less than one energy sources to supply energy. Equation (20) aims to avoid SLO violations due to insufficient renewable energy supply in each PM area. Since in our scenario, the number of renewable energy sources and the number of PM areas are integers, the elements decision variables are integers too. Thus, our problem can be transformed into an optimization problem. We use integer linear programming method [33] to solve this problem.

IV. Performance Evaluation

A. Experiment Settings

In our experiment, we use the following real world datasets. All the datasets are from May 1, 2011 to May 30, 2011.

(1) Datacenter and job workload. The Google cluster trace [34] records resource utilization of CPU and memory usage of each job in about 12.5 thousand PMs. For each job, we assign it an SLO value randomly chosen from the rage of [90%,100%) [35]. Then we transfer the jobs with the same SLO value into the same PM area. The number of PM areas is determined by the SLO value. We vary the number of PM areas from 10 to 100 by controlling the division interval of the SLO range; 10 PM areas mean that the SLO range is divided by an interval of 1%, and 100 PM areas mean that the

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

SLO range is divided by an interval of 0.1%. Unless otherwise specified, the number of PM areas is 100, and the mapping time period is one hour.

(2) Renewable energy resources. In our experiments, We choose solar and wind as the renewable energy resources. We assume 500 renewable energy generators at the Virginia State (VA); half are solar energy generators and half are wind energy generators. To set the amount of energy produced by each generator at each time slot, we calculate the amount according to the methods in [36] and [37] as follows.

The amount of solar energy that can be generated in a time slot is calculated by:

$$E^{solar}(t) = \alpha \cdot A^{solar} s(t) \cdot \Delta t \tag{21}$$

where α is the ratio of how much solar energy can be transferred in to electricity, A^{solar} is the total active irradiation area of the solar panels, $s(t)$ is the solar irradiance, which means the energy per unit area (watt per square metre, W/m2), and Δt is the length of a time slot. The amount of wind energy that can be generated in a time slot is calculated by:

$$E^{wind}(t) = \beta \cdot (\frac{1}{2}) A^{wind} \rho^{air} v^3(t) \cdot \Delta t \tag{22}$$

where β is the ratio of how much wind energy can be transferred in to electricity, A^{wind} is the total rotor area of all wind turbines, ρ^{air} is the air density, and $v(t)$ is the wind speed.

We obtained the datasets about regional solar irradiance ($s(t)$) and wind speeds ($v(t)$) at the VA from the National Renewable Energy Laboratory (NREL) [38], [39]. The data was recorded daily per hour, and we assume that the value keeps similar in different time slots in an hour. ρ^{air} is set to $1.29Kg/m^3$ and this value usually does not change in normal environment. For other parameters, we use the parameter settings in [36] and [37] for our renewable generator model. For each energy generator, the conversion efficiency ratio (α and β) is set to a value randomly chosen from [20%,30%]. For each solar energy generator, A^{solar} is set to a value randomly chosen from [10000,15000]m^2. For each wind energy generator, A^{wind} is set to a value randomly chosen from [20000,25000]m^2.

(3) Carbon emission rate (gCO_2e/kWh) is to measure the amount of carbon emissions from the energy use. According to [40], coal has 968 carbon emission rate, wind has 22.5 carbon emission rate, solar has 53 carbon emission rate. Since coal is the most widely used brown energy for electricity, we use coal as our brown energy in our experiments.

(4) Electricity price. The electricity price is obtained from the websites of Energy Information Administration [41] and the Switch [42], which contain the price of brown energy and renewable energy electricity price respectively at each hour. The electricity price varies from each hour as well. In general, the price for the solar energy is in the rage of [250,350] USD/MWh, that of the wind energy is in the rage of [130,220] USD/MWh, and that of the brown energy is in the rage of [50,150] USD/MWh.

In our experiments, first we use 80% of the renewable energy generation data as training set, and the rest 20% data as testing data to predict the tail distribution of each energy source. Second, we use 80% of the CPU utilization historical data in each PM throughout time as the training data and the rest 20% data as testing data, and then add the values of all PMs in the same PM area to get the predicted energy consumption of each PM area.

B. Compared Methods

As our work is the first to handle the problem, we cannot find comparable methods within our knowledge. We choose the following three methods for comparison, and the details of the methods are described in Section II.

1) Renewable and Cooling Aware Workload Management (RCA) [11].
2) Green Scheduling for Cloud Datacenters (GS) [9].
3) Renewable Energy-Aware Reinforcement Learning (REA) [13].

As the works in [9], [13] are for multiple datacenters and the energy sources already supply energy to their associated datacenters, we make changes to adapt these works to our single-datacenter scenario. Specifically, we evenly distribute the energy sources to the PM areas, that is, each energy source group supplies energy to one PM area. Also, we set the parameter values based on these papers.

In our experiments, first, we compare our renewable energy generation prediction method with the other two renewable energy generation prediction methods, which are pattern matching [9] and K-nearest neighbors (KNN) [11]. Second, we compare our energy demand prediction method with the other three energy demand prediction methods, which are pattern matching [43], Neural Network (NN) [44] and Fast Fourier Transform (FFT) [45]. Third, we compare the performance of our method with the other compared methods in [9], [11], [13] with their own prediction methods and with our LSTM prediction method.

C. Performance Metrics

1. *Prediction accuracy.* To measure the prediction accuracy of renewable energy generation and energy demands, we measure the prediction accuracy as below: $A_n = 1 - \frac{|P_n - R_n|}{R_n}$ where A_n is the prediction accuracy of n^{th} prediction, P_n is the predicted value of n^{th} prediction and R_n is the real value of n^{th} prediction.

2. *Uninterrupted PM area ratio.* It is defined as the percentage of PM areas that always receive renewable energy no less than their demands. It is calculated by: $B = 1 - \frac{V}{J}$ where V is the number of PM areas that the energy demands cannot be satisfied and J is the total number of PM areas.

3. *SLO satisfaction ratio.* It is defined as the percentage of jobs whose SLOs are satisfied. If one PM area does not receive enough renewable energy for its energy demand in time period T, we assume that all the jobs running on this PM area cannot successfully run in that time period. We run the trace data 50 times in our experiment. In the experiment, if the number of successful running times for one job over

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

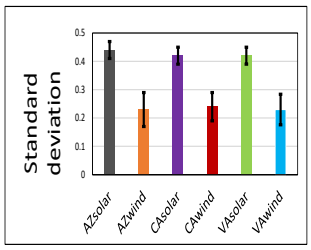


Fig. 1. Standard deviation of renewable energy in AZ, CA, and VA.

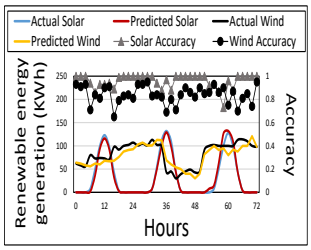


Fig. 2. Prediction of renewable energy generation.

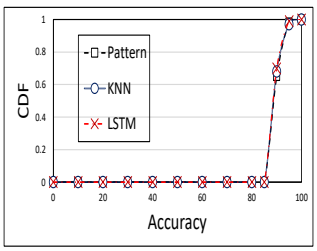


Fig. 3. Solar energy prediction accuracy.

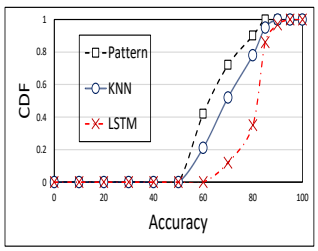


Fig. 4. Wind energy prediction accuracy.

the total number of the job running times is higher or equal to this job's SLO, we consider that this job's SLO is satisfied.

4. *Monetary cost and carbon emission.* We calculate the total monetary cost based on the real price dataset [41], [42] and Equation (15). Similar to monetary cost, we calculate the total amount of carbon emission based on the real dataset [40] and Equation (16).

5. *Time overhead.* We use training time latency and testing time latency to show the time overhead of the prediction methods and the source-PM area mapping methods.

D. Experimental Results

1) Renewable Energy Instability: In addition to VA, we also obtained the datasets about regional solar irradiance ($s(t)$) and wind speeds ($v(t)$) at the Arizona (AZ), California (CA) in May, 2011 from the National Renewable Energy Laboratory (NREL) [38], [39]. We choose these three locations because there are large datacenters in these states [46]. For each area, we calculate the standard deviation of $s(t)$ each day in the 30 days and then calculate the average standard deviation per day, and we also calculate the average standard deviation per day for $v(t)$.

Figure 1 shows the average standard deviation of solar irradiance and wind speed in a month in the three different areas. The error bar means the peak and valley standard deviation value in the month. The result follows AZsolar≈CAsolar≈VAsolar<AZwind≈CAwind≈VAwind. We see that the solar irradiations at different time slots in one day deviate greatly, so that the solar energy generation in one day is not stable. We found that the peak of the solar irradiation time is around 11 AM to 1PM each day, and in the rest of the day time, the solar irradiation varies. On the other hand, as the wind speed also varies in a day though it is more stable than the solar irradiation. Therefore, the energy generated by wind is also not stable.

2) Renewable Energy Generation Prediction Accuracy: Figure 2 shows the predicted and actual amount of renewable energy and the prediction accuracy for the solar energy and wind energy on one randomly selected solar generator and one wind energy generator using LSTM in 3 days randomly selected from one month. We see that the predicted values and actual values are almost overlapped and the accuracy stays above 0.8 most of the time. The result also shows that the accuracy of solar energy prediction is higher than wind energy prediction. However, we observe that the deviation of the generated energy amount at different time slots in one day of solar energy is larger than that of wind energy, which also is reflected by the result that the standard deviation of solar irradiance is larger than that of wind energy in Figure 1. The reason for the higher prediction accuracy of the solar energy compared to the wind energy is that since solar irradiation has certain time pattern, the amounts of the generated renewable energy at the same time in different days are similar, so it is easy to predict. But for wind energy, the relation between wind speed and time is not highly related between the same time points in different days. Therefore, for wind energy, the performance of prediction accuracy is not as high as that of the solar energy.

Figure 3 and Figure 4 show the CDF (Cumulative distribution function) of the solar energy prediction and wind energy prediction. The result for solar energy prediction follows: Pattern≈KNN≈LSTM. All prediction methods achieve high accuracy due to the same reason as explained in Figure 2. The result for wind energy prediction follows: Pattern<KNN<LSTM. The reason is that, for pattern matching, it only observes the wind energy for each time slot in each day and uses the same value for the same time slot in different days. Since the wind speed is not highly related between the same time point in different days, the accuracy for pattern matching is worse than KNN and LSTM. For KNN, it can classify the time period with similar wind speed, so in certain time slots, it achieves better accuracy than pattern matching. However, it can't consider the wind speed sequence in the entire time period. LSTM can consider the entire time period when predicting the wind speed in each time slot, thus producing the highest accuracy.

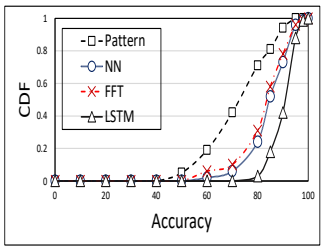


Fig. 5. Energy demand prediction accuracy.

3) Energy Demand Prediction Accuracy: Figure 5 shows the CDF of different prediction methods for energy consumption of PM areas. We choose 100 PM areas in this figure. The result follows: Pattern<NN≈FFT<LSTM. For pattern match-

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

ing, it observes the energy consumption for each PM area in each time slot each day and uses the same value for the same time slot in different days. FFT can process the time-series data and then find the most prominent pattern so that it achieves better performance than pattern matching. NN can consider the relationships between time points in the time sequence data, and LSTM can consider the relationship between time points in a longer time length, so LSTM can more accurately predict the energy consumption in each time slot than NN.

4) Performance Comparison with Compared Methods: We use RL to represent out RL-based method and use LP to represent out linear programming method. Figure 6 shows the energy utilization of different types of energies of each method. We observe that for brown energy: REA>RCA≈LP>RL≈GS. For REA, it aims to minimize the total energy monetary cost. Since the brown energy is cheaper than solar energy, it uses more brown energy to minimize the total price. For RCA, although it aims to minimize the total monetary cost, it tries to use more solar energy. LP consumes slightly more brown energy than RL. Since GS aims to minimize the total carbon emission, it uses renewable energy resource as much as possible which leads to low total brown energy usage.

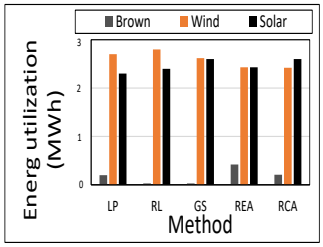
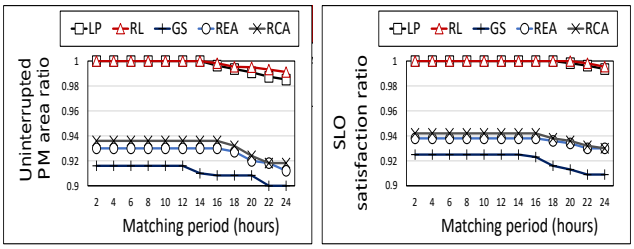


Fig. 6. Energy utilization of different energies.

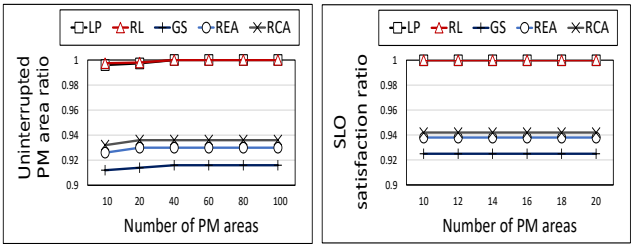
We also observe that the wind energy utilization follows RL>LP≈GS>RCA≈REA, and the solar energy utilization follows RCA≈GS>RL≈REA>LP. The wind energy is cheaper than the solar energy and also its carbon emission is lower than solar and brown energy. Therefore, since RL and LP aim to both reduce the total carbon emission and total energy monetary cost, wind is a better choice, so they use more wind energy than others. Since the total energy utilization is basically the same, RL and LP use less solar than others. Because LP uses more brown energy, it uses less wind and solar energy than RL. GS aims to minimize the total carbon emission, so it uses more wind energy and solar energy than others. REA and RCA aim to minimize the total energy monetary cost and RCA tries to use more solar energy. As a result, REA uses more brown energy (the cheapest energy) and RCA uses more solar energy than others.

Figure 7 shows the uninterrupted PM area ratio and SLO satisfaction ratio versus different matching time periods. In Figure 7(a), the result follows RL≈LP>RCA≈REA>GS. Our RL and LP use LSTM to more accurately predict the amount of generated renewable energy of each energy generator and energy demand of each PM area, so PM areas' energy demands are satisfied most of the time, thus producing the highest uninterrupted PM area ratio. As shown previously in Figures 3, 4 and 5, other prediction methods used in RCA, REA and GS have lower accuracy than LSTM, so more PM areas experience



(a) Uninterrupted PM area ratio (b) SLO satisfaction ratio

Fig. 7. Performance with different matching periods.



(a) Uninterrupted PM area ratio (b) SLO satisfaction ratio

Fig. 8. Performance with different number of PM areas.

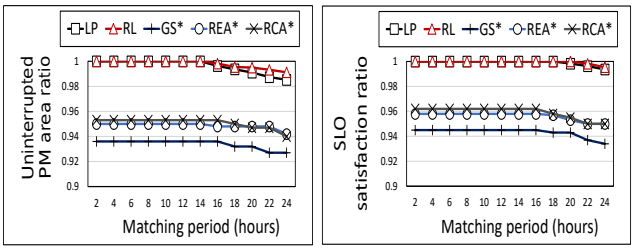
interruptions. Therefore, RCA, REA and GS generate lower uninterrupted PM area ratios. As shown in Figure 6, GS uses more renewable energies than REA and RCA, so GS has lower uninterrupted PM area ratio than those of REA and RCA because of the instability of the renewable energies.

In Figure 7(b), we can observe that the result follows RL≈LP>RCA≈REA>GS. Our RL and LP try to power each PM area with renewable energy generators that have probability no less than the PM area's SLO to produce the amount no less than its energy demand to ensure that insufficient renewable energy supply will not lead to SLO violations. As a result, they produce the highest SLO satisfaction ratio. REA, RCA and GS do not consider the possible SLO violations due to insufficient renewable energy supply and the subsequent energy supply switch. Therefore, they produce lower SLO satisfaction ratios. Since GS has the lowest uninterrupted PM area ratio, more jobs tend to experience SLO violations, so GS produces the lowest SLO satisfaction ratio.

In Figure 7, we also observe that as the matching period increases, the values of the two metrics of all methods decrease. This is because when the matching period is longer, the predicted values tend to be less inaccurate. Therefore, less frequent scheduling and longer scheduling time period make it less likely to guarantee that the supplied renewable energy is no less than the energy demand of a PM area. However, the slower decreasing rates of RL and LP mean that their scheduling time period can be longer than other methods, which saves computation resources for the scheduling.

Figure 8(a) and 8(b) show the uninterrupted PM area ratio and the SLO satisfaction ratio versus different number of PM areas. The results show RL≈LP>RCA>REA>GS, which are consistent with those in Figure 7 due to the same reasons as explained. For all the methods, as the number of PM

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.



(a) Uninterrupted PM area ratio (b) SLO satisfaction ratio

Fig. 9. Performance with different matching time periods using our prediction method.

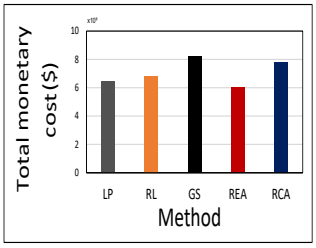


Fig. 10. Total monetary cost.

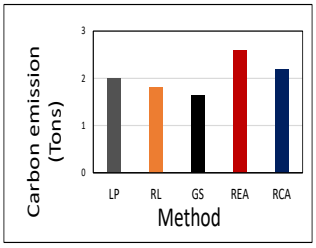


Fig. 11. Carbon emission.

areas increases, the uninterrupted PM area ratio and the SLO satisfaction ratio increase slightly. The reason is that, more PM areas mean fewer PMs in a PM area. Then, when a PM area does not receive sufficient renewable energy, fewer PMs and hence fewer jobs are interrupted. Therefore, more PM areas lead to higher uninterrupted PM area ratio and SLO satisfaction ratio.

Figure 9 shows the uninterrupted PM area ratio and the SLO satisfaction ratio versus different matching time period with our LSTM prediction method for energy demand. In the figure, GS*, REA* and RCA* means that we used LSTM and their scheduling methods as new methods. The results follow RL≈LP>RCA*>REA*>GS*, which is the same as in Figure 7 due to the same reasons. Comparing this figure and Figure 7, we observe that the uninterrupted PM area ratio and the SLO satisfaction ratio of GS*, REA* and RCA* are higher than those of GS, REA and RCA, respectively. This result indicates that our prediction method is more accurate and it can help achieve higher uninterrupted PM area ratio and SLO satisfaction ratio.

Figure 10 shows the total monetary cost of each method. The results show GS>RCA>RL>LP>REA. GS aims to minimize the total carbon emission, so it uses renewable energy resource as much as possible though the price of the solar energy is much more higher than the wind and brown energy, thus generating the highest energy monetary cost. RCA aims to minimize the total monetary cost while trying to use solar energy as much as possible. Our LP and RL tend to choose wind energy due to its cheaper price and lower carbon emission as explained above, their total energy monetary cost is lower than GS and RCA. REA tends to use more brown energy since the goal for REA is to only minimize the total energy monetary cost. Since the price of each energy resource follows Solar>brown≈wind, REA generates the lowest monetary cost.

Recall that the carbon emission rate for each energy resource follows Brown>Solar>Wind. Figure 11 shows the carbon emission of each method. The results show REA>RCA>LP≈RL>GS. REA tends to use more brown energy since it aims to minimize the total energy monetary cost. Since the carbon emission amount of the brown energy is much higher than the solar energy and wind energy, REA produces the most carbon emission. RCA uses more renewable energy than REA since RCA tries to use solar energy as much as possible, so that RCA generates less carbon emission than REA. Since both LP and RL aim to reduce carbon emission and also total energy monetary cost, they use more wind energy, thus producing less carbon emission. GS only aims to minimize the total carbon emission, so it uses renewable energy resource as much as possible, which produces the lowest carbon emission. Combining the results in Figures 10 and 11, we can conclude that our LP and RL perform well in both reducing total energy monetary cost and carbon emission, while other methods cannot achieve both goals simultaneously.

Figure 12 shows the time overhead for one decision of each method. The time overhead contains both training and testing for RL and REA. The results show REA≈RL>LP>GS≈RCA. REA and RL use the reinforcement learning model which needs a long time for training. After the model is trained, the decision making process takes a short time, which is shorter than other methods. Other methods need to solve an optimization problem without the need for training. LP considers more input features (including carbon emission, energy price and SLO violation) than GS and RCA, so its time overhead for making a decision is much more than GS and RCA. Since GS and RCA consider less input features, their optimization problems are easier to solve, so their time overheads of making a decision are lower than LP.

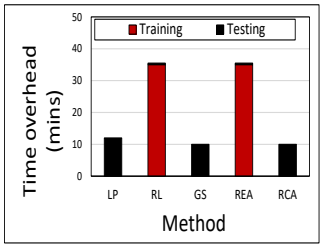


Fig. 12. Time overhead.

V. Conclusion

Renewable energy supply is a promising solution to current datacenter energy supply which is much more environment-friendly. However, the instability of renewable energy may lead to insufficient energy supply to the datacenter, resulting in job running interruption or even failures. Previous work attempting to achieve higher energy generation prediction cannot completely handle this problem since sufficient renewable energy supply cannot be guaranteed due to energy instability.

In this paper, we propose a renewable energy resource allocation system for a cloud datacenter. Our objective is to avoid SLO violations due to interruption from insufficient renewable energy supply while minimizing the total energy monetary cost and total carbon emission. First, using deep learning technique, the system predicts the tail distribution of

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.

each renewable energy source at each time slot in the next time period. Second, it predicts the energy demand in each PM area by predicting the CPU utilization for each PM in a PM area. Third, based on the predicted results, the system assigns renewable energy sources to PM areas to solve the above problem using RL-based method and linear programming method. Our extensive real trace driven experiments show that our system achieves superior performance than other methods in terms of the aforementioned goals. Since job migration between PMs in load balancing in a cloud datacenter can be very resource intensive, in our future work, we will further study how to minimize the energy cost of the job migration process.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants NSF-1827674, CCF-1822965, OAC-1724845, CNS-1733596, Microsoft Research Faculty Fellowship 8300751, and AWS Machine Learning Research Awards. We would like to thank Dr. Rajkumar Buyya for his valuable comments.

REFERENCES

[1] M. Chaudhry, T. Ling, S. Hussain, and X. Lu, "Thermal-aware relocation of servers in green data centers," *Frontiers of Information Technology & Electronic Engineering*, 2015.
[2] R. Katz, "Tech titans building boom," *Trans. on IEEE spectrum*, 2009.
[3] "United states data center energy usage report," *https://eta.lbl.gov/publications/united-states-data-center-energy*, [Accessed in Oct. 2019].
[4] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. of ACM SIGCOMM*, 2009.
[5] M. Webb, "Smart 2020: enabling the low carbon economy in the information age, a report by the climate group on behalf of the global esustainability initiative (gesi)," *Creative Commons*, 2008.
[6] "Emission regulation," *http://sapientservicesllc.com/regulations-govern-data-center-operations/*, [Accessed in Oct. 2019].
[7] "Apple expands n.c. solar farm to make data center use 100% renewable energy," *https://www.cultofmac.com/191838/apple-expands-n-c-solar-farm-to-make-data-center-use-100-renewable-energy/*, [Accessed in Oct. 2019].
[8] A. Malekpour and A. Pahwa, "Stochastic networked microgrid energy management with correlated wind generators," *Trans. on TPS*, 2017.
[9] C. Gu, C. Liu, J. Zhang, H. Huang, and X. Jia, "Green scheduling for cloud data centers using renewable resources," in *Proc. of INFOCOM WKSHPS*, 2015.
[10] H. Wang and H. Shen, "Proactive incast congestion control in a datacenter serving web applications," in *Proc. of INFOCOM*, 2018.
[11] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. of ACM SIGMETRICS*, 2012.
[12] L. Gu, J. Cai, D. Zeng, Y. Zhang, H. Jin, and W. Dai, "Energy efficient task allocation and energy scheduling in green energy powered edge computing," *Future Generation Computer Systems*, 2019.
[13] C. Xu, K. Wang, P. Li, R. Xia, S. Guo, and M. Guo, "Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning," *Trans. on NSE*, 2018.
[14] I. De Courchelle, T. Guérout, G. Da Costa, T. Monteil, and Y. Labit, "Green energy efficient scheduling management," *Simulation Modelling Practice and Theory*, 2019.
[15] H. Wang, J. Gong, Y. Zhuang, H. Shen, and J. Lach, "Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes," in *Proc. of BigData*, 2017.
[16] "Data centers: Jobs and opportunities in communities nationwide," *https://www.uschamber.com/report/data-centers-jobs-opportunities-communities-nationwide*, [Accessed in Oct. 2019].
[17] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services." in *Proc. of NSDI*, 2008.
[18] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks." in *Proc. of NSDI*, 2010.
[19] Y. Lin and H. Shen, "Eafr: An energy-efficient adaptive file replication system in data-intensive clusters," *Trans. on TPDS*, 2017.
[20] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient cloud resource management," in *Proc. of INFOCOM WKSHPS*, 2014.
[21] R. Komp, "Practical photovoltaics: Electricity from solar cell., 3rd edition," *Ann Arbor, aatec publications*, 2003.
[22] P. Gipe, "Wind power, revised edition: renewable energy for home, farm, and business," *New York, Chelsea Green*, 2004.
[23] G. Liu and H. Shen, "An economical and slo-guaranteed cloud storage service across multiple cloud service providers," in *Proc. of INFOCOM*, 2016.
[24] H. Shen and L. Chen, "Compvm: A complementary vm allocation mechanism for cloud systems," *IEEE/ACM Transactions on Networking (TON)*, 2018.
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.
[26] J. Gao, H. Wang, and H. Shen, "Task failure prediction in cloud data centers using deep learning," *Proc. of Bigdata*.
[27] X. Fan, C. Ellis, and A. Lebeck, "The synergy between power-aware memory systems and processor voltage scaling," in *Proc. of PACS*, 2003.
[28] L. Minas and B. Ellison, *Energy efficiency for information technology: How to reduce power consumption in servers and data centers*. Intel Press, 2009.
[29] V. Mnih, K. Kavukcuoglu, and D. Silver, "Human-level control through deep reinforcement learning," *Nature*, 2015.
[30] L. Tan, S. Song, P. Wu, Z. Chen, R. Ge, and D. Kerbyson, "Investigating the interplay between energy efficiency and resilience in high performance computing," in *Proc. of IPDPS*, 2015.
[31] A. Legrand, D. Trustram, and S. Zrigui, "Adapting batch scheduling to workload characteristics: What can we expect from online learning?" in *Proc. of IPDPS*, 2019.
[32] P. Kochovski, R. Sakellariou, and M. Bajec, "An architecture and stochastic method for database container placement in the edge-fog-cloud continuum," in *Proc. of IPDPS*, 2019.
[33] J. Abara, "Applying integer linear programming to the fleet assignment problem," *Interfaces*, 1989.
[34] C. Reiss, J. Wilkes, and J. Hellerstein, "Google cluster-usage traces: format+ schema," *Google Inc., White Paper*, 2011.
[35] "Oracle slo requirements," *https://docs.oracle.com/en/enterprise-manager/index.html*, [Accessed in Oct. 2019].
[36] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam, "Carbon-aware energy capacity planning for datacenters," in *Proc. of MASCOTS*, 2012.
[37] C. Stewart and K. Shen, "Some joules are more precious than others: Managing renewable energy in the datacenter," in *Proc. of HotPower*, 2009.
[38] "Nrel solar radiation research laboratory-solar dataset," *https://midcdmz.nrel.gov/apps/sitehome.pl?site=BMS*, [Accessed in Oct. 2019].
[39] "Nrel wind technology center-wind dataset," *https://midcdmz.nrel.gov/apps/sitehome.pl?site=NWTC*, [Accessed in Oct. 2019].
[40] "Measurement and instrumentation data center," *https://midcdmz.nrel.gov/l*, [Accessed in Oct. 2019].
[41] "Wholesale electricity and natural gas market data," *https://www.eia.gov/electricity/wholesale/*, [Accessed in Oct. 2019].
[42] "Which is the cheapest renewable energy supplier in 2019," *https://theswitch.co.uk/blog/energy/cheapest-green-supplier*, [Accessed in Oct. 2019].
[43] R. Karp and M. Rabin, "Efficient randomized pattern-matching algorithms," *IBM journal of research and development*, 1987.
[44] D. Specht, "A general regression neural network," *Trans. on NN*, 1991.
[45] H. Sorensen, "Real-valued fast fourier transform algorithms," *Trans. on ASSP*, 1987.
[46] M. Xua and R. Buyyab, "Managing renewable energy and carbon footprint in multi-cloud computing environments," in *Elsevier preprint*, 2019.

Authorized licensed use limited to: Northeastern University. Downloaded on April 16,2021 at 06:57:17 UTC from IEEE Xplore. Restrictions apply.