

Resource Management for Minimizing Energy and Cost of Geo-Distributed Data Centers

Moh Moh THAN

University of Computer Studies, Yangon, Myanmar

(* Corresponding author's e-mail: mohmohthan@ucsy.edu.mm)

Received: 8 March 2020, Revised: 7 April 2021, Accepted: 14 April 2021

Abstract

Geo-distributed data centers (GDCs) house computing resources and provide cloud services across the world. As cloud computing flourishes, energy consumption and electricity cost for powering servers of GDCs also soar high. Energy consumption and cost minimization for GDCs has become the main challenge for the cloud service providers. This paper proposes a resource management framework that accomplishes resource demand prediction, ensuring service level objective (SLO), electricity price prediction, and energy-efficient and cost-effective resource allocation through GDCs. This paper also proposes an energy-efficient and cost-effective resource allocation (EECERA) algorithm which deploys energy efficiency factors and incorporates the electricity price diversity of GDCs. Extensive evaluations were performed based on real-world workload traces and real-life electricity price data of GDC locations. The evaluation results showed that the resource demand prediction model could predict the right amount of dynamic resource demand while achieving SLO, and also, the electricity price prediction model could provide promising accuracy. The performances of resource allocation algorithms were evaluated on CloudSim. This work contributes to minimizing the energy consumption and the average turnaround time taken to complete the task and offers cost-saving.

Keywords: CloudSim, Geo-distributed data centers, Resource management, Service level objective, Energy-efficient and cost-effective resource allocation, Power management techniques

Introduction

Cloud computing services are built on GDCs, which are spread worldwide to provide better performance and reliability. The service providers face high energy consumption and operating cost as servers in data centers consume a tremendous amount of energy. It is estimated that the electricity demand for data centers may rise more than 66 % over 2011 - 2035 [1], and the annual electricity bill is over \$40M [2]. Energy-related costs may amount to 41.6 % [3], which leads to the high operational cost of GDCs. Thus, it has been recognized as an emergent issue.

Resource management is crucial for efficient operation in large-scale data centers that contain thousands of servers. The resource management process has many aspects to consider, including handling workload dynamism, achieving SLO that user requirements are met, and allocating the requests to the available resources energy-efficiently and cost-effectively [4-6].

The amount of resources needed to allocate the requests is often dynamic due to its workload demand. Accurate resource demand prediction is essential to allocate resources dynamically and effectively. The cloud provider must ensure they have enough resources to meet the resource amount needed for incoming requests. Resource provisioning with the right amount of dynamic resource demand while achieving SLO becomes a critical issue [7]. Energy cost can be minimized at the geo-distributed level by distributing more workloads to the data center with a lower price since GDCs in multi-regional electricity markets are with various prices. Predicting the electricity price for GDCs in multi-regional

markets is required to reduce the electricity cost of GDCs. Machine learning (ML) is a method of computational learning that has shown promising results in the domain of prediction. ML algorithms build the models by extracting knowledge from the data, and ML model predictions allow making highly accurate estimates.

Resource allocation is the main cloud computing technology that uses computing resources to facilitate various user jobs. Both the energy efficiency of the servers inside data centers and the location-based electricity prices diversity need to be considered to reduce the total operational cost. CloudSim [8] is a generalized and extensible simulation toolkit that allows for cloud infrastructure modeling. It provides a user-defined policy for allocating virtual machines (VMs) to servers (hosts).

This paper proposes an energy-efficient and cost-effective resource management framework with four components: resource demand prediction, ensuring SLO, electricity price prediction, energy-efficient, and cost-effective resource allocation. The resource demand prediction model is developed by using ML techniques and SLO analysis is performed. Predicting electricity prices of GDCs in multi-regional electricity markets is performed to deploy for cost minimization of GDCs. EECERA algorithm is proposed based on the comparative results of two energy-saving resource allocation algorithms that consider energy efficiency factors and the electricity prices diversity of GDCs. EECERA algorithm is implemented on CloudSim to allocate GDCs' resources to support higher energy efficiency and lower electricity cost.

Literature review

This section discusses previous research and techniques related to predicting resource demand and ensuring SLO, predicting electricity price, and resource allocation methods that reduce energy consumption and lower the energy cost of data centers.

Rayan and Nah [9] applied ML techniques to predict the daily workload: the amount of power consumption (PC) and the number of physical machines (PMs) required to fulfill the demands. They observed that Random Forest Regression among three different ML methods provides the best performance, with an error of 11.68 for PMs and 4869.08 for PC prediction. Verma *et al.* [4] studied the prediction of the future CPU demand. They analyzed the performance of 5 ML algorithms and found that Reduced Error Pruning Tree produced a better result with 97.05 % accuracy.

Bobroff *et al.* [10] presented an Autoregressive (AR) prediction model to accommodate allocation decisions based on predicted resource demand and proposed the algorithm for dynamic allocation. The algorithm proactively adapts to demand changes while providing SLO guarantees. They solved the over-provision problem on the predicted resource demand. Their algorithm could provide a specified rate of SLO violations by reducing the amount of physical capacity required for a given workload and could achieve a substantial reduction in resource consumption up to 50 % compared to the static allocation.

Wormstrand [11] applied ML algorithms to predict the future electricity price of the Norwegian market. They compared the performance of 7 ML algorithms and found that the support vector machine is the best algorithm for their datasets, with an average error of 3.14 %. Filho *et al.* [12] used data mining techniques based on clustering and the Decision Tree algorithm to predict the electricity price of 4 markets: Center-east, Northeast, South, and North in Brazil. The overall accuracies of the classification and prediction for the short-term electricity price of the four markets are 94.02, 87.87, 89.55, and 97.01 %, respectively.

Hicham and Chaker [13] discussed the allocation algorithms: Shortest Job First (SJF) and First Come First Serve (FCFS). They evaluated the performance of these algorithms on CloudSim, and the results showed that SJF takes minimum average turnaround time compared to FCFS. Ali *et al.* [14] proposed an energy-efficient VM placement algorithm by selecting the most energy-efficient host first. They also applied power management techniques to their algorithm. Their algorithm achieved 9, 23, and 23 % more power-efficient than Minimum Power Difference, Best Resource Selection, and Round Robin algorithms.

Khosravi *et al.* [15] presented different energy and carbon-aware dynamic VM allocation methods that consider the parameters: availability of renewable sources, dynamic PUE, and changes of energy consumption that affect in reducing the total energy and carbon cost, and also the brown energy usage of

GDCs. Rawas and Zekri [16] proposed the Location-Aware and Energy-Efficient (LAEE) data-intensive workloads in GDCs. They combined the DVFS technique in LAEE to minimize the energy consumption of servers. Their proposed allocation method is evaluated using CloudSim. The results showed enhancements in the user's QoS, minimizing the total turnaround time as well as reducing the energy consumption of the data centers.

The above-mentioned resource demand prediction papers [4,9,10] did not consider enhancing the accuracy of the prediction model. In this paper, the resource demand prediction model is developed and is enhanced by hyper-parameter optimization to achieve a highly accurate predictor. This paper focuses on a higher priority to avoid under-provision than to avoid over-provision solved in [10] since the former is more likely to cause SLO violations. In this paper, the SLO analysis was performed in the prediction system and guaranteed SLO to meet the requested amount of CPU cores.

This paper illustrates price prediction for GDCs in US multi-regional electricity markets using ML techniques as the papers [11,12]. It also considers the time zone of the regions as [17]. After analyzing the performance of 3 ML algorithms, the most suitable prediction model is chosen to predict the future prices. Moreover, the predicted prices are applied in resource allocation for saving energy costs by incorporating the electricity price differences of GDCs.

Each of the mentioned papers [13-16] focused on minimizing the turnaround time for each request, PC of servers, energy consumption, and cost of data centers, respectively. This paper proposes the resource management framework to minimize energy consumption and also cost of data centers while satisfying SLO. For saving energy, it considers energy efficiency factors: allocation policies and power management technique in the proposed allocation algorithms, to minimize the turnaround time as well as to save the PC of the servers. For lowering the cost, it also exploits the electricity price diversity of GDCs.

Energy-efficient and cost-effective resource management framework

To manage the resources of GDCs in an energy-efficient and cost-saving manner while satisfying SLO, the proposed framework is implemented with four components: resource demand prediction, ensuring SLO, electricity price prediction, and energy-efficient and cost-effective resource allocation. It is illustrated in **Figure 1**, and the function of each component is as follows.

Resource Demand Prediction - It predicts the amount of resource (CPU core) for on-demand resource planning and efficient resource management of dynamic workload.

Ensuring SLO - SLO analysis is performed over the resource demand prediction results to avoid SLO violations.

Electricity Price Prediction - Electricity prices through GDCs in multi-regional electricity markets are predicted to reduce the energy cost by directing the requests to the favor of data center with minimal electricity price.

Energy-Efficient and Cost-Effective Resource Allocation - It allocates the workload requests with the proposed EECERA algorithm in energy-efficient and cost-effective manner. It monitors the power usage of resources and energy consumption of data centers using energy-saving techniques. It also attempts to reduce energy cost by exploiting the electricity price differences across GDCs.

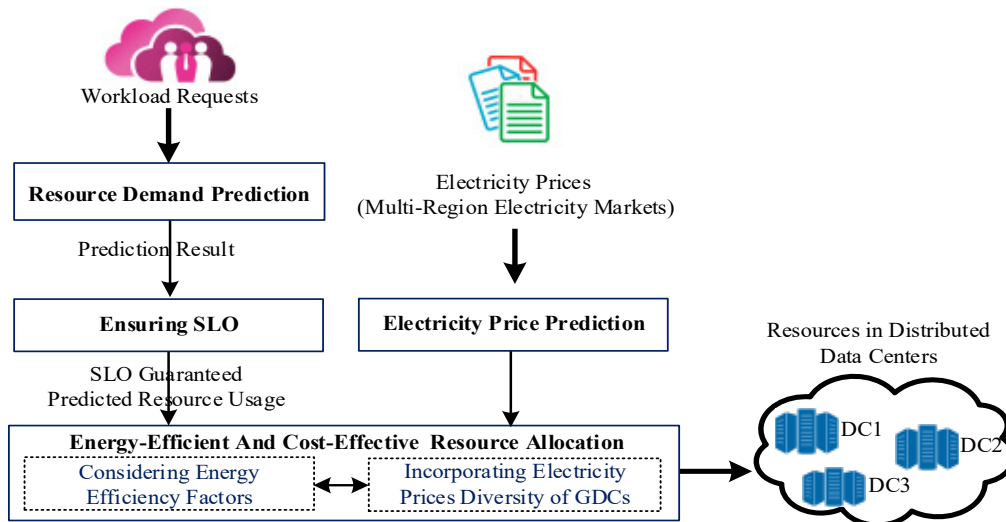


Figure 1 Proposed energy-efficient and cost-effective resource management framework.

Experimental environment

This section describes system specification for testing environment. It also presents the description of real workload traces and electricity price traces for the experiments.

System specification

The experiments were conducted on Intel ® Core i7-7500U CPU @ 2.90GHz, 8GB Memory, 1TB Hard Disk, HP machine.

Workload traces

The evaluation of the prediction model was conducted using data center workload traces of jobs submitted to an HPC (High-Performance Computing) Cluster available from the Parallel Workload Archive [18]. The summary of experiment workload traces (datasets) and their features are shown in **Table 1**.

Table 1 Summary of experiment workload traces and their features.

Datasets	No. of Jobs	Features
DAS (Distributed ASCI in Netherlands Supercomputer)	225,711	JobID SubmitTime RunTime NumCPUs UsedMemory
RICC (RIKEN Integrated Cluster of Clusters)	447,794	JobNo. ArrivalTime RunTime ReqCPUs ReqMemory
MetaCentrum (Czech National Grid Infrastructure MetaCentrum)	103,656	JobID SubmitTime RunTime NumCPUs UsedMemory

Electricity price traces

In the United States, electricity prices may vary per hour in some regions with deregulated electricity markets, while in most regions, electricity prices stay unchanged in a day [19]. To better capture the electricity price variations, it is assumed 3 GDCs in different regions where dynamic electricity prices are available under competitive electricity markets.

As the electricity prices for each data center, this research uses the publicly available data from the markets [20]: Electricity Market of Independent System Operator New England (ISONE), California (CAISO), and Electric Reliability Council of Texas (ERCOT) obtained from New England, California, and Texas, respectively. Hourly electricity prices (\$/MWh) in the years: 2016 and 2017 datasets for each of 3 markets are experimented. These datasets include 3 features: Date with time, Locational Marginal Price (LMP), and name of regional Hub.

Materials and methods

This research proposes SLO guaranteed, energy-efficient, and cost-effective resource management framework. The resource demands for the workload requests are predicted with the highly accurate resource demand prediction model, and it ensures SLO to meet the requested amount of CPU cores. SLO-guaranteed resource usage is applied in resource allocation for preparing the correct type of resources and planning the resource requirements in advance. The electricity prices of GDCs in multi-region electricity markets are predicted with the developed electricity price prediction models to route the incoming requests to the cheapest data center for minimizing the cost. The workload requests are allocated energy-efficiently and cost-effectively through GDCs. The detailed procedures of the components in the proposed framework and their evaluation results are described below.

Resource demand prediction

When VMs are statically provided according to the peak resource demand, it causes high resource cost for users and low resource utilization for cloud providers. So, it is important to implement a proactive resource provision to guarantee users' requirements. Accurate resource demand prediction becomes a key feature for efficient resource management of dynamic workload.

This section describes the development of the resource demand prediction model to predict the number of CPU cores for user requests. The CPU resource demand prediction model is developed on the

Decision Tree (DT) ML algorithm, where the hyper-parameter optimization was performed to achieve the prediction model with a high accuracy. The evaluation of the prediction model was conducted using real data center workload traces.

The performance of the model was measured by the mean absolute error (MAE) as in Eq. (1). The accuracy of a predictor was estimated by computing an error based on the difference between the predicted value y_i' and the actual known value y_i for each of test instances, where d is the number of instances.

$$MAE = \frac{\sum_{i=1}^d |y_i - y_i'|}{d} \quad (1)$$

Resource demand data pre-processing

This involves pre-processing historical workload traces by filtering out unnecessary information from raw data. It identifies or removes the outliers, and also resolves inconsistencies. The experimental workload traces are divided into 2 disjoint parts: training data and testing data.

Resource demand prediction model development

The resource demand prediction models are generated using the DT algorithm based on all possible combinations of hyper-parameters pairs and all features provided in training data. DT works reasonably well with default hyper-parameter values specified in software packages. The values of hyper-parameters affect the accuracy of the prediction model, and finding the best combination of hyper-parameters values for the DT plays a critical role in building the accurate predictor.

Hyper-parameter Optimization - It needs to determine the optimal hyper-parameter setting for the learning algorithm to produce the model with a minimum error rate. For the DT hyper-parameter setting, the parameters chosen for optimization are Maximum tree depth (MaxDepth) [21] and Minimum Information Gain (MinInfoGain) [22]. Many possible combinations of hyper-parameters (31 MaxDepth values: 0 - 30 and 11 MinInfoGain values: 0 - 1) are observed. The numbers of 341 models are developed using all possible combinations, as shown in lines 3 - 5 of **Figure 2**.

The resource demand prediction model with the lowest error rate is selected by analyzing the accuracy of the models generated with all possible combinations of hyper-parameter values as line 6 of **Figure 2**. The best model with minimum MAE is selected to predict the future CPU resource demand as line 10 of **Figure 2**. The procedure of resource demand prediction model development by hyper-parameter optimized DT algorithm is shown in **Figure 2**.

Procedure: Resource_Demand_Prediction

Input: pre-processed workload traces
Output: predicted resource demand

- 1 **BEGIN**
- 2 Develop the prediction model by using default hyper-parameter values of DT
 -Evaluate model accuracy ($minMAE \leftarrow MAE$)
- 3 **for** (MaxDepth=0, MaxDepth<=30, MaxDepth++)
- 4 **for** (MinInfoGain=0, MinInfoGain<=1, MinInfoGain+=0.1)
- 5 Develop the prediction model by using MaxDepth and MinInfoGain for DT
 -Evaluate model accuracy (print MAE)
- 6 **if** ($minMAE > MAE$) **then** $minMAE \leftarrow MAE$
- 7 **end if**
- 8 **end for**// MinInfoGain
- 9 **end for**// MaxDepth
- 10 Predict resource demand by the selected model with minMAE
- 11 **END**

Figure 2 Resource demand prediction procedure.

Experiment and result discussion of resource demand prediction

The accuracy varies depending on each set of parameter values and the characteristics of the workload dataset. Each set of all different combination parameter sets for each dataset are input to the DT algorithm to generate the prediction models based on the training data. The prediction performances of the considered models are evaluated on the testing data. The MAE results of the generated models against some combinations of hyper-parameter set pairs (0 to 10 MaxDepth values and 0.0 to 1.0 MinInfoGain values) among 341 models for the DAS dataset are shown in **Table 2**.

Table 2 MAE Results of Generated Models using Different Combinations of Hyper-Parameters (for DAS).

Max Depth	MinInfoGain										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16
1	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96
2	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37
3	1.93	1.93	1.93	1.93	1.93	1.88	1.88	1.88	1.88	1.88	1.88
4	1.49	1.49	1.49	1.49	1.49	1.59	1.59	1.59	1.59	1.59	1.59
5	1.59	1.59	1.52	1.52	1.51	1.60	1.60	1.60	1.60	1.60	1.60
6	1.34	1.34	1.46	1.46	1.46	1.55	1.55	1.55	1.55	1.55	1.55
7	1.26	1.28	1.46	1.46	1.46	1.56	1.56	1.56	1.56	1.56	1.56
8	1.27	1.18	1.45	1.46	1.46	1.55	1.55	1.55	1.55	1.56	1.56
9	1.80	1.17	1.44	1.44	1.44	1.54	1.54	1.55	1.55	1.54	1.54
10	1.92	1.20	1.46	1.47	1.47	1.52	1.52	1.54	1.54	1.54	1.54

After analyzing the MAE results of 341 generated models for the DAS dataset having the maximum MAE = 13.16 and the minimum MAE = 1.17, the model which has the minimum MAE is observed in MinInfoGain = 0.1 as shown in **Table 2**. For all MaxDepth values, the MinInfoGain value 0.5 and

beyond these cannot change the error rates significantly. The model, which has the minimum MAE for RICC dataset and MetaCentrum dataset are observed in MinInfoGain = 0.0. The MAE results of the generated models against 31 different MaxDepth values with the optimal MinInfoGain for each of 3 datasets are shown in **Figures 3 to 5**.

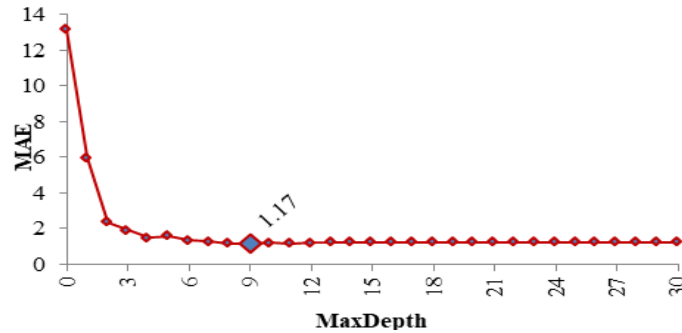


Figure 3 MAE results of the models generated using different MaxDepth values with MinInfoGain = 0.1 (for DAS).

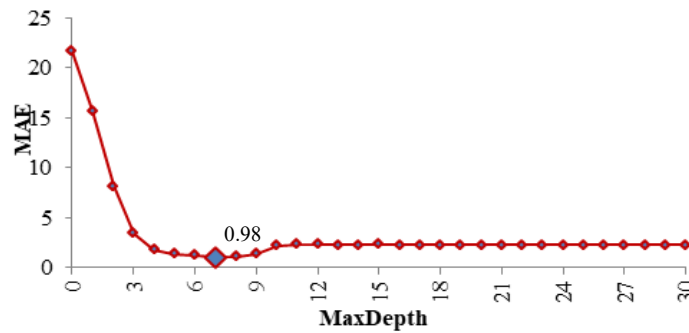


Figure 4 MAE results of the models generated using different MaxDepth values with MinInfoGain = 0 (for RICC).

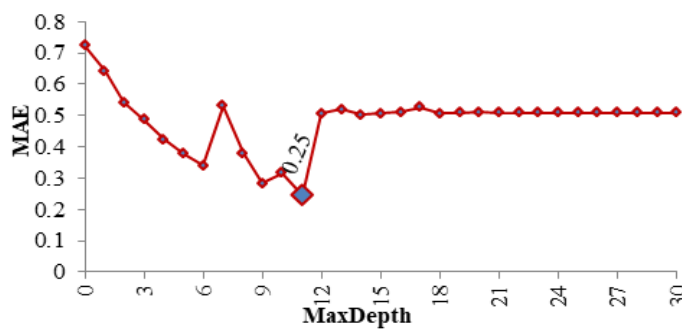


Figure 5 MAE results of the models generated using different MaxDepth values with MinInfoGain = 0 (for MetaCentrum).

From comparing the prediction accuracy of the generated models for the DAS dataset, the model with hyper-parameter pair (MaxDepth = 9 and MinInfoGain = 0.1) gives a better result MAE = 1.17. After analyzing the MAE results of 341 generated models (for the RICC dataset having the maximum MAE = 21.69 and the minimum MAE = 0.98, for the MetaCentrum dataset having the maximum MAE = 0.72 and the minimum MAE = 0.25), the minimum MAE is observed in the optimal hyper-parameter pair (MaxDepth = 7 and MinInfoGain = 0.0 for RICC dataset and MaxDepth = 11 and MinInfoGain = 0.0 for MetaCentrum dataset). The best prediction model with the lowest error rate was selected for the corresponding dataset.

Figure 6 presents MAE comparisons of prediction models generated by setting default hyper-parameters (MaxDepth = 5 and MinInfoGain = 0) and optimal hyper-parameters (that has the least MAE presented in **Figures 3 to 5**) for each dataset.

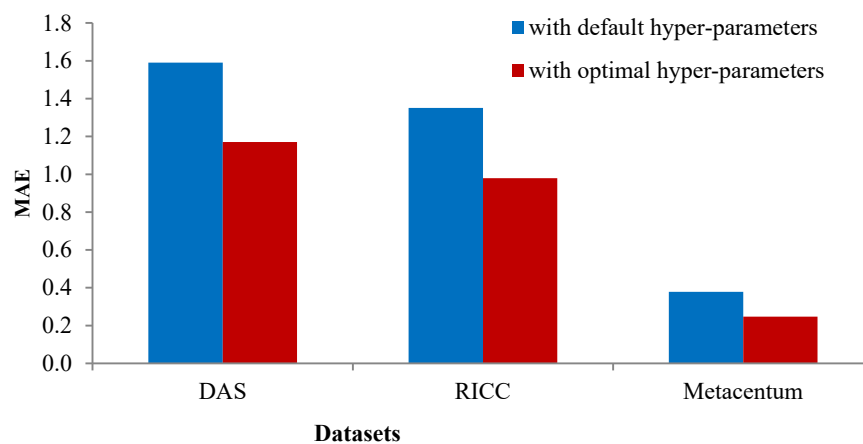


Figure 6 MAE comparisons of prediction models with default and optimal hyper-parameters.

The MAE value of the generated models with default hyper-parameters is 1.59, 1.35, 0.38 and with optimal hyper-parameters is 1.17, 0.98, 0.25 for DAS, RICC and MetaCentrum dataset respectively. Thus, setting the optimal hyper-parameters can significantly affect the resulting model's performance. The results show that setting the optimal hyper-parameters saves MAE 26, 28 and 35 % for the corresponding datasets respectively.

Ensuring SLO

The cloud provider must ensure they have enough resources to meet the customer demand. Otherwise, the provider will need to pay compensation to those customers whose performance criteria have not been met. This paper presents SLO based on the criteria to meet the requested CPU cores of the submitted jobs.

Predictive models that have a good predictive accuracy may generate prediction errors due to underestimation or overestimation. In order to avoid this risk, the predicted value generated from the model can be adjusted to accommodate these prediction errors. Under-provision due to underestimation causes SLO violations. Thus, SLO analysis is performed in the prediction system and guarantees SLO. **Figure 7** shows the under-provisioning frequencies after analyzing the actual and predicted value of the three datasets.

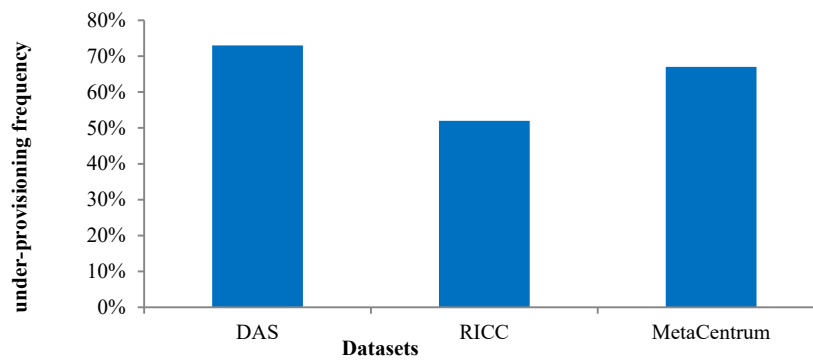


Figure 7 Under-provisioning frequency of predicted values for 3 datasets.

According to the results shown in **Figure 7**, the proposed predictors are more under-provision than over-provision. It eliminates the under-provisioning of the prediction system by padding a small percentage of the maximum predicted value to avoid SLO violations. The predicted value is increased with a small amount (1 - 5 % in our experiment) of the maximum predicted value. The increasing value is calculated as in Eq. (2). SLO analysis results evaluated on resource demand prediction results of 3 workload traces over one-day interval are shown in **Figure 8**.

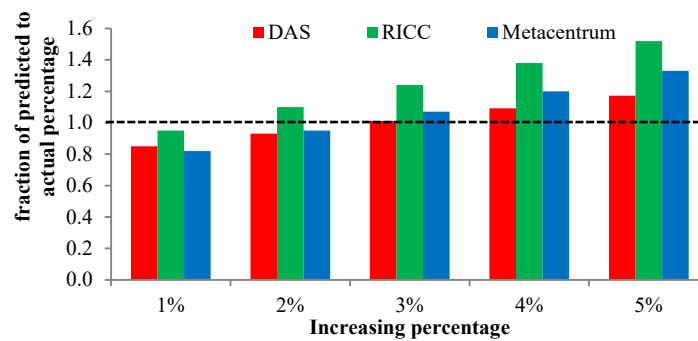


Figure 8 SLO analysis result of 3 workload datasets (one-day interval).

$$\text{Increasing value} = \text{maximum predicted value} \times \frac{\text{increasing percentage}}{100 \%} \quad (2)$$

The fraction of the predicted to the actual value for a one-day interval greater than or equals one means it meets SLO. The analysis results in **Figure 8** shows that it can remove under-provisioning and meet SLO by increasing a small percentage with 3 % of the maximum predicted value. The amount of SLO guaranteed predicted resource usage is applied for planning the resource requirements in advance.

Electricity price prediction

The cloud provider can reduce energy costs by directing the requests to the data center with a cheaper price if the electricity prices of GDCs in multi-regional electricity markets are predicted in advance. This section describes predicting electricity prices of GDCs in US multi-regional electricity markets to exploit in our proposed resource allocation algorithm.

Electricity price data pre-processing

Ideally, information about all the days for the whole period should be available, but some of this information was missing and some records duplicated. Electricity price datasets are pre-processed by deleting duplicated records to replace the missing value with the most commonly occurring value for that feature or with the most probable statistical value.

In this step, electricity price data is adjusted to refer to the time in the Eastern Time Zone as [17]. Three GDCs located in 3 different markets, e.g., New England, California, and Texas, are GMT-7, GMT-8, and GMT-6, respectively. It takes the time in the Eastern Time Zone for time-keeping and therefore adjusts the time difference for New England, California, and Texas by 0, 3, and 1 h, respectively.

The electricity price dataset for 2016 is divided into 90 % used for training and 10 % used for testing in each of the three markets to conduct the ML evaluation process. Validating dataset of the year 2017 for each of 3 markets is used for the selected model assessment.

Electricity price prediction model selection

The prediction models are generated using three different ML algorithms: M5P, Linear Regression, and Decision Table. This was applied by using training and testing data for every dataset. It uses the same input attributes for each dataset, which will make it easy to compare the results and decide which algorithm is the most accurate. After comparing the MAE results of the models generated using 3 ML algorithms, the most accurate model is selected to predict the future electricity price. The obtained model for each market is evaluated by applying the newly validating data in the same period for all markets to be compared for GDCs.

Experiment and result discussion of electricity price prediction

The comparative MAE results of the predictive models generated by 3 ML algorithms: M5P, Linear Regression and Decision Table for each of 3 electricity markets: ISONE, CAISO and ERCOT are presented in **Figure 9**.

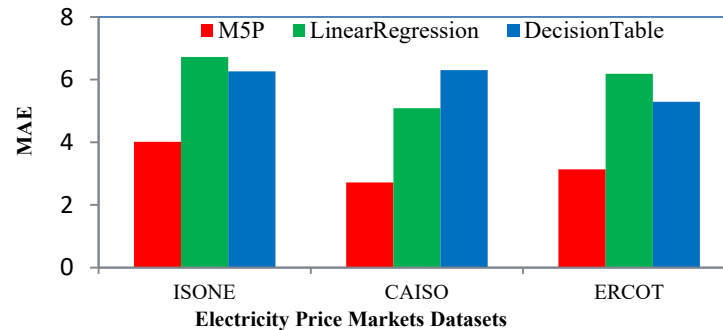


Figure 9 MAE comparisons of different ML algorithms for 3 electricity markets.

The models developed by the M5P algorithm achieve less MAE value than Linear Regression and Decision Table for all 3 markets datasets. As a result, the predictive model for each market generated by the M5P algorithm is selected to predict future electricity prices.

Electricity prices in each data center for one-day intervals are generated by applying to the obtained electricity price-prediction models. **Figure 10** illustrates the 24-hour real-life and predicted electricity price sequences of these data centers in 3 electricity markets on January 2, 2017.

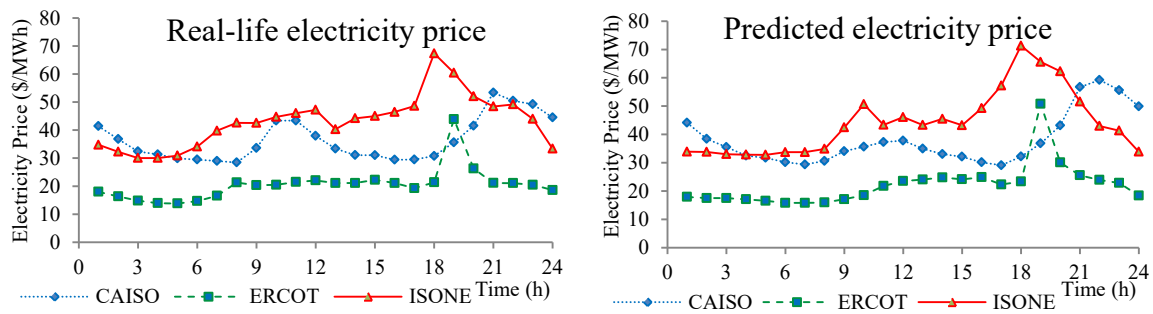


Figure 10 Electricity prices of GDCs in multi-regional electricity markets on January 2, 2017.

It can be observed that the relative order of the predicted electricity price sequences matches the relative order of the respective real-life electricity price sequences, although there is a little difference in real and predicted price values of each data center. As long as the relative order condition mentioned above holds, the data center with cheaper electricity price can be selected correctly at any time in a day. Using the predicted electricity price sequences of GDCs, energy cost minimization can be concerned.

Energy-efficient and cost-effective resource allocation

Energy-efficient and cost-effective resource allocation means allocating the incoming requests (tasks) to the servers through VMs in particular toward goals of saving energy and lowering cost.

$$E = P \cdot T \quad (3)$$

By considering the parameter of energy consumption (E) expressed in Eq. (3), it can be minimized by reducing the power consumption (P) as well as also the time period (T) needed to turn on the servers.

Energy consumption is reflected in the energy cost consumed by the system during the period of operation, which is the main component of a data center's operating cost. To save energy and cost of data centers, this paper deploys energy efficiency factors and electricity prices differences across GDCs depending on their locations.

Energy efficiency factors

The aim of energy efficiency is to reduce the energy consumption of servers in the data centers. There are some energy-saving techniques that can be deployed for monitoring and controlling energy consumption of servers. In this paper, energy efficiency factors: resource allocation policies and power management technique are considered for energy-saving resource allocation.

Resource allocation policies

Resource allocation is the process of creating VM instances that match with the incoming requests onto the hosts. This paper focuses on allocation policies to optimize processing time and energy consumption of servers. The optimal allocation policy decreases the time and availability of space in a productive manner without compensating the quality of the system [23]. This study emphasizes the following 2 allocation policies: First Come First Serve (FCFS) and Shortest Job First (SJF).

FCFS - The request which comes first to the data center is allocated to the VM first.

SJF - The request with the least runtime (length) among the requests in the ready queue is allocated to the VM first.

In evaluating allocation algorithms, turnaround time is important metric influencing energy consumption. Turnaround time is the time interval from the time since the task entered into the ready queue for execution until the task completed its execution. The average turnaround time is calculated as follow:

$$ATT = \frac{\sum_{i=1}^N TT_i}{N} \quad (4)$$

$$TT_i = FT_i - AT_i \quad (5)$$

where, ATT is Average Turnaround Time, TT_i is Turnaround Time of i^{th} job, N is the Number of jobs, FT_i is Finish Time of i^{th} job execution and AT_i is Arrival time of i^{th} job.

Power management technique

As servers are the primary power consumers of data centers, Dynamic Voltage and Frequency Scaling (DVFS) power management technique is used for minimizing PC of the servers without affecting the quality of services.

DVFS - It is the dynamic power management technique [24] that decreases the dynamic PC of a processor expressed in Eq. (6) by dynamically changing the voltage and the frequency of the processor during execution depending on the CPU utilization. It can adjust system voltage and frequency of a server without restarting [25]. The PC of the server can be reduced when it is in an idle state or low workload through DVFS technique. This method can decrease the PC of servers and enhance resource utilization.

$$P_{dynamic} = a \cdot c \cdot v^2 \cdot f \quad (6)$$

where $P_{dynamic}$ is dynamic power consumed, a is switching activity, c is capacitance, v is voltage, and f is frequency.

Experiment setup

This paper considers an IaaS provider with 3 data center sites located in 3 cities in US multi-regional electricity markets. These cities: Boston in New England, San Jose in California, and Dallas in Texas with 3 different time zones are chosen from Data centers Map website [26]. Each data center has 45 heterogeneous physical servers with 5 different characteristics given in **Table 3**.

Table 3 Characteristics of server types.

Server Type	Number of Core	Power
Type 1	16	750 W
Type 2	28	800 W
Type 3	32	750 W
Type 4	64	750 W
Type 5	64	1100 W

For research work on the cloud, the prototypical implementation is carried on CloudSim. It supports the integrated classes in Java to simulate the cloud environment. It allows modeling the cloud infrastructure providers with different system scale and hardware resources. It models the data centers with the broker, which acts on behalf of cloud users, physical servers, and virtual machines.

Even though CloudSim supports cloud infrastructure simulation, there is no consideration for minimizing energy consumption and cost of multi data center level. It is extended to enable energy-efficient and cost-effective resource allocation for GDCs.

A cloud infrastructure environment is implemented to verify the concept and validate the functionality of the proposed resource allocation algorithms. Then the own classes for the allocation algorithms are created to extend the basic CloudSim classes. The problem is modeled to enable the execution of HPC jobs from the logs of workload traces on parallel machines. The broker submits the job requests to the data centers for processing. For allocating the cloudlets to VMs, the space-shared policy is used so that the tasks are executed sequentially in each VM. Using this policy, each job unit has its own dedicated core. The allocation of VMs to hosts utilizes the proposed algorithms. Since the proposed algorithms use a non-preemptive method, the number of incoming jobs or the queue size did not affect the execution time of the individual job units. The power management technique is applied using the PowerModel class that provides a function called `getPower()`, which returns the PC of the host.

After implementing two energy-saving resource allocation algorithms and comparing their energy consumption, the EECERA algorithm is proposed for better resource allocation with less energy consumption and less cost of GDCs. The system consists of the following components and symbols are presented in **Table 4**.

Table 4 Description of symbols.

Symbol	Description
n	Number of VM requests
m	Number of Data Center sites
l	Number of hosts in each Data Center site
P	Server power consumption
E_k	Energy consumption of k^{th} host
Host_k	k^{th} host in Data Center j
P_{r_j}	Electricity price of Data Center j
Vm_i	i^{th} virtual machine
Length_i	runtime of i^{th} job request
DC_j	Data Center j
E_{total}	Total energy consumption of hosts in GDCs
C_{total}	Total energy cost of all hosts in GDCs

Energy-saving resource allocation algorithms

Two energy-saving resource allocation algorithms include DVFS enabled First Come First Serve (DFCFS) and DVFS enabled Shortest Job First (DSJF). These are implemented to address resource allocation in multiple VMs and maintain a minimized turnaround time and minimized energy consumption using CloudSim.

Algorithm 1 and 2 describe DFCFS and DSJF algorithms, respectively. The data center broker creates a list to receive the jobs (cloudlets). Virtual machines, where $\text{VM} = \{\text{vm1}, \text{vm2}, \dots, \text{vmn}\}$ are created, and the broker maps a job to a VM based on one-to-one mapping configuration. A job request contains the attributes such as the Runtime (Length), the number of CPU cores (NumCPUs, ReqCPUs), the required memory size (UsedMemory, ReqMemory), as mentioned in **Table 1**. The broker has (DC x Host) different VM placement options by the arrival of each job request. It submits the VM to the selected data center and host if the host has enough resources for the VM. It calculates the average turnaround time as in Eqs. (4) and (5). The energy consumption is computed as a function of CPU usage and is regulated automatically and dynamically based on DVFS.

DVFS enabled First Come First Serve (DFCFS) Algorithm

Algorithm 1 shows DFCFS algorithm based on FCFS allocation policy and deployed with DVFS power management technique.

Algorithm 1: DFCFS Algorithm

Input:
Job_List // the list of job requests
Host_list // the list of hosts in a data center
DC_list // the list of available data centers in geographic sites

Output:
Destinations [Host_k in DC_j]
ATT // average turnaround time
E_{total} // total energy consumption of hosts in data centers

```
1 BEGIN
2 Create VMs as the number of job requests in Job_List
3 foreach Job in Job_List do
4   Assigned jobi → VMi
5 end for
6 for all VM i ← 1 to n do
7   for all DC j ← 1 to m do
8     for all Host k ← 1 to l do
9       if CPU core of VMi ≤ remaining CPU core of Hostk then
10        Assigned VMi → Hostk in DCj
11        Calculate the remaining CPU core of Hostk
12      else Start Hostk+1 in DCj
13      end if
14    end for
15  end for
16 end for
17 Return Destinations
18 Calculate ATT
19 Calculate Etotal with DVFS
20 END
```

DVFS enabled Shortest Job First (DSJF) Algorithm

Algorithm 2 shows the DSJF algorithm applied with SJF policy and combined with the DVFS power management technique. It sorts the job list in ascending order based on their lengths, as in line 2 of Algorithm 2. If two requests have the same length, the next request can be allocated through FCFS allocation, where one that arrives first will be allocated to VM as in lines 3 - 5.

Algorithm 2: DSJF Algorithm

Input:

Job_List // the list of job requests
Host_list // the list of hosts in a data center
DC_list // the list of available data centers in geographic sites

Output:

Destinations [Host_k in DC_j]
ATT // average turnaround time
E_{total} // total energy consumption of hosts in data centers
1 BEGIN
2 Sort Job_List in ascending order based on their Length
3 **if** Length_i = Length_{i+1} **then**
4 Sort Job_List according to their Arrival Time
5 **end if**
6 Create VMs as the number of job requests in sorted Job_List
7 **foreach** Job in sorted Job_List **do**
8 Assigned job_i → VM_i
9 **end for**
10 **for all** VM $i \leftarrow 1$ to n **do**
11 **for all** DC $j \leftarrow 1$ to m **do**
12 **for all** Host $k \leftarrow 1$ to l **do**
13 **if** CPU core of VM_i ≤ remaining CPU core of Host_k **then**
14 Assigned VM_i → Host_k in DC_j
15 Calculate the remaining CPU core of Host_k
16 **else** Start Host_{k+1} in DC_j
17 **end if**
18 **end for**
19 **end for**
20 **end for**
21 Return Destinations
22 Calculate ATT
23 Calculate E_{total} with DVFS
24 **END**

Energy Consumption Comparison of DFCFS and DSJF

Figures 11 to 13 show the comparisons of energy consumption for the different number of requests from 3 workload traces: DAS, RICC and MetaCentrum under DFCFS and DSJF algorithms.

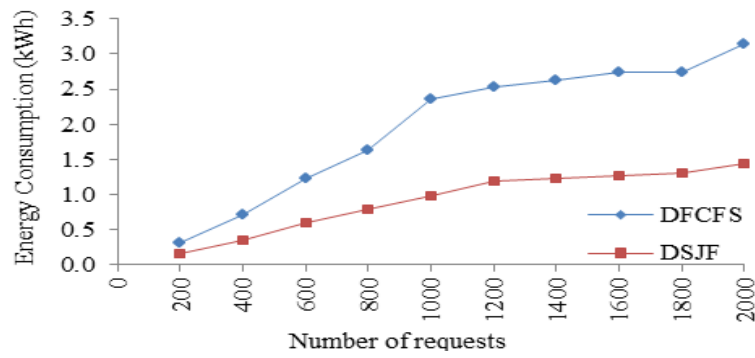


Figure 11 Energy consumption comparison of DFCFS and DSJF (for DAS).

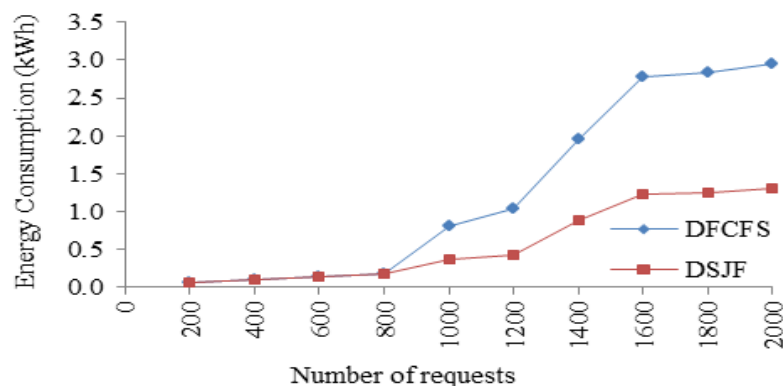


Figure 12 Energy consumption comparison of DFCFS and DSJF (for RICC).

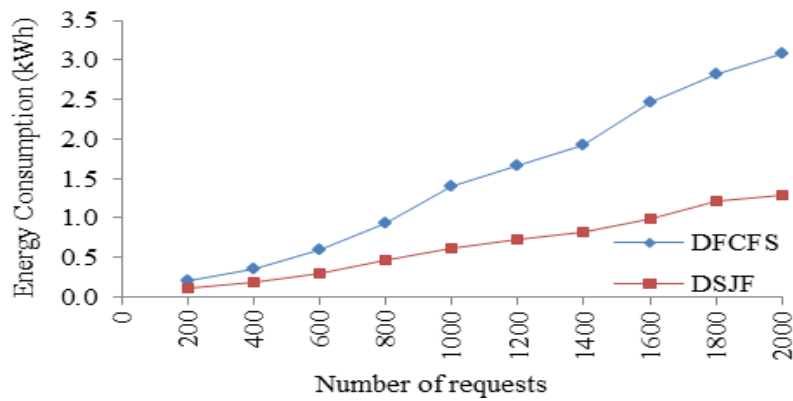


Figure 13 Energy consumption comparison of DFCFS and DSJF (for MetaCentrum).

The energy consumed for 200 to 800 requests of the RICC dataset under DFCFS and DSJF algorithms is almost indistinguishable. The reason is that the runtimes (lengths) of those requests are similar. As mentioned above, DSJF sorts the runtimes of the requests in ascending order before execution while DFCFS does not. Although the DSJF algorithm sorts the lengths of the requests having similar lengths, there is no effect of sorting the requests based on lengths. For the requests over 800 number of RICC dataset and all the requests of the other two datasets (DAS and MetaCentrum) having different lengths, it can be seen the effect of DSJF can save energy consumption compared to DFCFS algorithm. DSJF algorithm gains higher energy efficiency for the incoming requests with different lengths, and it can save up to 55 % of energy consumption compared to the DFCFS algorithm.

The average turnaround times (in seconds) for executing the requests of 3 datasets under DFCFS and DSJF algorithms are presented in **Table 5**.

Table 5 Average turnaround time of DFCFS and DSJF Algorithms.

Datasets	DFCFS	DSJF
DAS	204	86
RICC	164	77
MetaCentrum	244	113

By comparing the average turnaround time obtained from DFCFS and DSJF, as shown in Table 5, the turnaround time of DSJF is almost 55 % less than DFCFS. DSJF helps to reduce the turnaround time of the tasks as it first fulfills the request with the shortest length, and it takes the shortest possible time to finish. Then the VM can take up the next selected task. This will reduce the time period and the number of active VMs and active servers to reduce energy consumption. The shorter average turnaround time of the allocation method produces lower energy consumption. In this way, DSJF that takes the minimum turnaround time shows its contribution to energy-saving.

Proposed energy-efficient and cost-effective resource allocation (EECERA) algorithm

Lowering high operating costs of the data centers is one of the challenges faced by cloud providers. Fortunately, the geographical distribution of data centers exposes the opportunity for cost-saving, meaning that they pay different electricity prices exposed to different electricity markets.

Algorithm 3: EECERA Algorithm**Input:**

#Job_List // the list of job requests
 #Host_list // the list of hosts in a data center
 #DC_list //the list of available data centers in geographic sites

Output:

Destinations [Host_k in DC_j]
 # E_{total} // total energy consumption of hosts in data centers
 # C_{total} // total energy cost of hosts in data centers

```

1 BEGIN
2 Sort Job_List in ascending order based on their Length
3 if Lengthi = Lengthi+1 then
4   Sort Job_List according to their Arrival Time
5 end if
6 Create VMs as the number of job requests in sorted Job_List
7 foreach Job in sorted Job_List do
8   Assigned jobi → VMi // Add job request to VM based on one-to-one mapping configuration
9 end for
10 Sort DC_list in ascending order based on their Prj
11 for all VM i ← 1 to n do
12   for all DC j ← 1 to m do
13     for all Host k ← 1 to l do
14       if CPU core of VMi ≤ remaining CPU core of Hostk then
15         Assigned VMi → Hostk in DCj
16         Calculate the remaining CPU core of Hostk
17       else Start Hostk+1 in DCj
18       end if
19     end for
20   end for
21 end for
22 Return Destinations
23 Calculate Etotal with DVFS
24 Calculate Ctotal
25 END

```

EECERA algorithm, as described in Algorithm 3, is proposed for energy-efficient and cost-saving resource allocation for GDCs. The key idea is to shift resource allocation to energy-efficient servers as well as to locations associated with comparatively lower electricity prices. In order to reduce energy cost by exploiting the electricity price differences across regions under multiple electricity price market environments, our policies attempt to submit each arriving job to the data center that would lead to the lowest cost.

To be energy-efficient, EECERA deploys as a DSJF algorithm. Moreover, it attempts to minimize the total energy cost based on price diversities of GDCs. The electricity price for data center j is denoted as Pr_j and all servers in DC_j shall share the electricity price Pr_j . It routes the requests in favor of the data center with the cheapest electricity price by sorting the data center list in ascending order based on their prices as line number 10 of Algorithm 3. It calculates the total energy consumed by the servers within the data centers to support the VMs. The total energy cost is calculated as Eq. (7) by summing up the cost of all servers in all GDCs, i.e.,

$$C_{total} = \sum_{j=1}^m (\sum_{k=1}^l E_k) \cdot Pr_j \tag{7}$$

Three workload datasets are used to compare the proposed algorithm with respect to cost minimization with an energy-aware resource allocation (EERA) algorithm that considers only energy efficiency factors as DSJF but does not consider GDCs' electricity price order diversities as line number 10 of Algorithm 3.

Figures 14 to 16 show energy cost comparisons of EECERA and EERA that allocate 2000 VM requests from each of 3 workload datasets knowing hourly electricity prices of 3 electricity markets: ISONE, CAISO, and ERCOT in **Figure 10**.

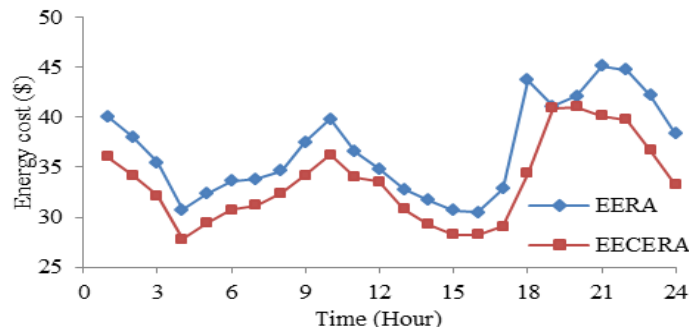


Figure 14 Total energy cost comparison of EERA and EECERA (for DAS).

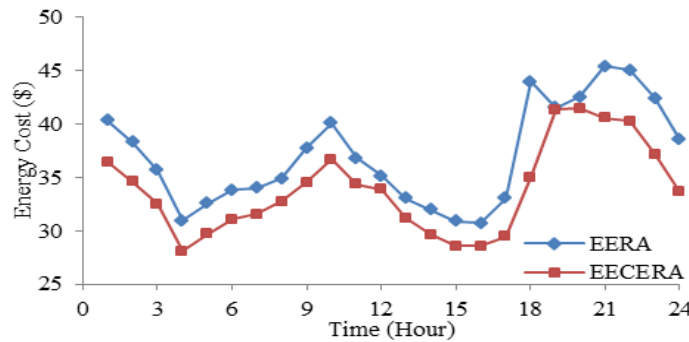


Figure 15 Total energy cost comparison of EERA and EECERA (for RICC).

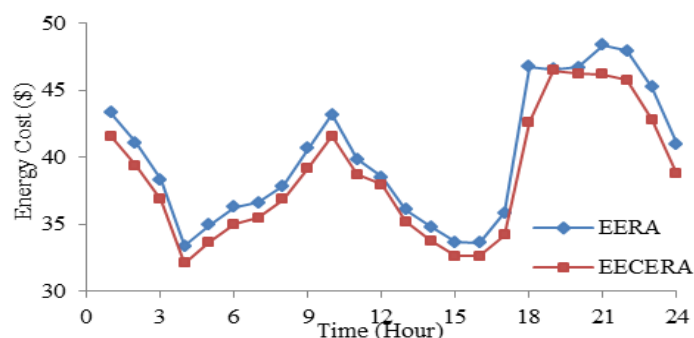


Figure 16 Total energy cost comparison of EERA and EECERA (for MetaCentrum).

EECERA Algorithm directs more workload requests to the data center with the cheapest electricity price first, knowing the future electricity prices of GDCs so that the total energy cost can be saved. According to the results experimented on three actual workload traces shown in **Figures 14 to 16**, the total energy cost of EECERA is significantly lower than EERA in every hour. With extensive simulations based on the electricity prices of GDCs shown in Figure 10, it shows that EECERA is able to reduce the electricity cost by 14, 13, and 6 % for each of the three datasets, respectively. This further proves that the EECERA algorithm provides a practical approach to lowering electricity costs for GDCs in multi-regional electricity markets with price diversities.

Conclusions

Resource management is getting popularity as it is paying attention for managing cloud resources to maximize the revenue of the cloud service providers. This paper proposes a framework for managing the cloud resources in GDCs cost-effectively through exploiting the electricity price diversity of GDCs and energy-efficiently while satisfying a certain level of SLO. Based on three real-world workload traces and three real-life electricity price data of GDCs in multi-regional electricity markets, implementation and extensive evaluations are performed for each component of the proposed framework.

Resource demand prediction model with high accuracy is developed using machine learning techniques with hyperparameter optimization to predict the resource demand with a low error rate. The results show that the resource demand prediction model generated by setting the optimal hyperparameters can significantly affect the resulting model's performance saving MAE around 30 %. SLO analysis is performed by padding 3 % of the maximum predicted value, eliminating the under-provisioning of the prediction system. After analyzing the MAE results of the models generated by 3 ML algorithms: M5P, Linear Regression, and Decision Table, the M5P model is selected to predict the future electricity prices of GDCs in multi-regional electricity markets in order to deliver the requests to the data center with cheaper electricity prices. By using CloudSim, the promising energy efficiency of 2 energy-saving resource allocation algorithms are investigated. It can be observed that the energy consumption of resource allocation with the DSJF algorithm is lower, about 55 %, than DFCFS. The proposed EECERA algorithm is compared with the EERA algorithm for energy cost minimization. Evaluation results show that it achieves both minimized energy consumption and cost reduction in the proposed resource allocation algorithms.

References

- [1] A Varasteh and M Goudarzi. Server consolidation techniques in virtualized data centers: A survey. *IEEE Syst. J.* 2017; **11**, 772-83.
- [2] A Qureshi, R Weber, H Balakrishnan, J Guttag and B Maggs. Cutting the electric bill for internet-scale systems volume XXXIX. *In: Proceedings of the ACM SIGCOMM Conference on Data Communication*, Barcelona, Spain. 2009, p. 123-34.
- [3] J Hamilton. Cooperative Expendable Micro-Slice Servers (CEMS): Low cost, low power servers for internet-scale services. *In: Proceedings of the 4th Biennial Conference on Innovative Data Systems Research*, California, USA. 2009, p. 1-8.
- [4] M Verma, GR Gangadharan, N Nanjangud, V Ravi, V Inamdar, L Ramachandran, R Calheiros and R Buyya. Dynamic resource demand prediction and allocation in multi-tenant service clouds. *Concurr. Comp. Pract. Exp.* 2016; **28**, 4429-42.
- [5] SK Garg, SK Gopalaiyengar and R Buyya. SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud datacenter. *In: Proceedings of the 11th International Conference on Algorithms and Architectures for Parallel Processing*, Melbourne, Australia. 2011, p. 371-84.
- [6] T Thein, MM Myo, S Parvin and A Gawanmeh. Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. *J. King Saud Univ. Comp. Info. Sci.* 2020; **32**, 1127-39.
- [7] J Liu. 2016, Towards an efficient cloud computing system: Data management, resource allocation and job scheduling. Ph. D. Dissertation. Clemson University, South Carolina, USA.
- [8] RN Calheiros, R Ranjan, A Beloglazov, CAD Rose and R Buyya. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exp.* 2011; **41**, 23-50.
- [9] A Rayan and Y Nah. Resource prediction for big data processing in a cloud data center: A machine learning approach. *IEIE Trans. Smart Process. Comput.* 2018; **7**, 478-88.
- [10] N Bobroff, A Kochut and K Beaty. Dynamic placement of virtual machines for managing SLA violations. *In: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management*, Munich, Germany. 2007, p. 119-28.
- [11] Ø Wormstrand. 2011, Electricity price prediction, a comparison of machine learning algorithms. Ph. D. Dissertation. Østfold University College, Halden, Norway.
- [12] JCR Filho, CM Affonsoe and RCL Oliveirae. Pricing analysis in the Brazilian energy market: A decision tree approach. *In: Proceedings of the IEEE Bucharest Power Tech*, Bucharest, Romania. 2009, p. 1-6.
- [13] GT Hicham and EA Chaker. Cloud computing CPU allocation and scheduling algorithms using CloudSim simulator. *Int. J. Electr. Comput. Eng.* 2016; **6**, 1866-79.
- [14] A Ali, L Lu, Y Zhu and J Yu. *An energy efficient algorithm for virtual machine allocation in cloud data centers.* *In: J Wu and L Li (Eds.). Advanced computer architecture.* Springer Science and Business Media Singapore, Singapore. 2016, p. 61-72.
- [15] A Khosravi, LLH Andrew and R Buyya. Dynamic VM placement method for minimizing energy and carbon cost in geographically distributed cloud data centers. *IEEE Trans. J. Sustain. Comput.* 2017; **2**, 183-96.
- [16] S Rawas and A Zekri. Location-aware energy-efficient workload allocation in geo distributed cloud environment. *J. Comput. Sci.* 2018; **14**, 334-50.
- [17] MA Adnan, R Sugihara and RK Gupta. Energy efficient geographical load balancing via dynamic deferral of workload. *In: Proceedings of the 5th IEEE International Conference on Cloud Computing*, Honolulu, HI, USA. 2012, p. 188-95.
- [18] Parallel Workloads Archive, Available at: <http://www.cs.huji.ac.il/labs/parallel/workload>, accessed January 2020.
- [19] A Mohsenian-Rad and A Leon-Garcia. Energy-information transmission tradeoff in green cloud computing. *In: Proceedings of the IEEE International Conference on Global Communications*, Toronto, Canada. 2010, p. 1-6.

- [20] Energy Online, Available at: <http://www.energyonline.com/Data/Default.aspx>, accessed January 2020.
- [21] Decision Trees - RDD-based API, Available at: <https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html>, accessed January 2020.
- [22] M Camilleri and F Neri. Parameter optimization in Decision Tree learning by using simple Genetic Algorithms. *WSEAS Trans. Comput.* 2014; **13**, 582-91.
- [23] AV Karthick, E Ramaraj and RG Subramanian. An efficient multi queue job scheduling for cloud computing. *In: Proceedings of the World Congress on Computing and Communication Technologies, Trichirappalli, India.* 2014, p. 164-6.
- [24] D Kliazovich, P Bouvry, Y Audzevich and SU Khan. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *In: Proceedings of the IEEE Global Telecommunications Conference GLOBECOME, Miami, USA.* 2010, p. 1-5.
- [25] LT Lee, KY Liu, HY Huang and CY Tseng. A dynamic resource management with energy saving mechanism for supporting cloud computing. *Int. J. Grid Distrib. Comput.* 2013; **6**, 67-76.
- [26] Data Center Map, Available at: <http://www.datacentermap.com>, accessed January 2020.