JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AutoML for Multi-Label Classification

Authors:-Kushal Saraf,
Department of Computer Science, NMIMS Computer Science, 2018-2022,

Abstract

As the scale of distributed computing continues to expand, its impact on energy consumption and the environment is becoming increasingly evident. According to statistics, data centers' energy usage has accounted for approximately half of their operational costs. The escalating energy consumption not only demands a significant amount of energy resources but also places a substantial strain on the environment. The substantial energy consumption of cloud data centers has become a focal concern in the field of information technology, drawing significant attention and requiring urgent resolution.

Presently, the high energy usage issue can be attributed to two primary factors. Firstly, the resource reservation mechanism, driven by the need to meet completion times, leads to low server utilization rates, inevitably resulting in inefficient usage for smaller tasks. Secondly, the current cooling infrastructure in data centers relies on a peak-value approach, which leads to excessive cooling supply, heightened operational costs, and a significant environmental impact.

Introduction

introduces a novel approach to addressing the high energy consumption challenge in data centers, particularly in the context of cloud computing. Leveraging techniques rooted in artificial intelligence, we propose the implementation of a reservation control engine and an intelligent cooling engine aimed at reducing energy consumption. Furthermore, we construct a platform for a green cloud data center, effectively deploy the reservation control engine, and validate the feasibility of the proposed framework. The results indicate that the framework can facilitate the creation of a cloud platform characterized by low power consumption and highly energy-efficient data center operations. Cloud computing has made an appearance in the ever-changing information technology world, garnering significant attention and adoption in recent years [1]. Fueled by its computational prowess, high reliability, storage capacity, and efficient service capabilities, cloud computing has found applications in diverse domains such as the Internet of Things (IoT) [2] and smart applications [3]. As the cloud computing ecosystem expands, the proliferation of cloud data centers has led to a pressing concern: the escalating energy consumption required to sustain these centers [4]. This energy consumption not only incurs substantial costs but also exerts a considerable strain on environmental resources. The need to solve this issue quickly is highlighted by the following data centers' energy consumption now accounts for over 50% of their operating expenses [5].

Undoubtedly, the prominence of cloud computing necessitates an urgent shift towards energy-efficient solutions. This challenge is two-fold. Firstly, the substantial energy consumption in cloud data centers stems from two primary factors. The first factor is the resource reservation mechanism, which, in pursuit of meeting completion time requirements, often leads to suboptimal server utilization. This phenomenon results in the underutilization of server resources and inefficient processing of smaller tasks. The second factor is the prevalent cooling approach employed in contemporary data centers, which frequently rely on top-tier solutions. This not only results in excessive cooling provision but also inflates operational costs, aggravating the challenge of energy inefficiency.

To tackle this pressing issue, the concept of a "green cloud data center" has emerged as a beacon of hope, representing the inevitable evolution of cloud data centers [6]. Green data centers seek to optimize IT equipment, refrigeration systems, power distribution, and information infrastructure to achieve enhanced energy efficiency while minimizing environmental impact. In this context, the term "green" signifies a holistic approach to data center design that champions sustainability and energy conservation. Consequently, green cloud data centers have emerged as pioneers in energy-efficient research.

While numerous approaches have been explored to optimize energy consumption within cloud data centers, they generally fall into two overarching categories:

1. **Resource Allocation and Scheduling:** This facet focuses on strategies to maximize the utilization of available resources, consequently improving overall efficiency. To address the dynamic and uncertain nature of user demands, real-time resource allocation decisions are made to optimize either task completion efficiency or resource utilization rates [7]. These approaches offer promise in domains such as dynamic workload allocation, where resource assignments must adapt to fluctuating demands to ensure optimal performance [8]. From allocating computation tasks in mobile networks [9] to designing optimal scheduling mechanisms for mobile agent paths [10], resource allocation and scheduling research play a pivotal role in enhancing data center efficiency.

2. **Power Supply and Cooling Optimization:** As the second significant area of research, power supply, and cooling mechanisms are explored to minimize operational costs and energy consumption. Cooling systems within data centers are a key contributor to energy expenditure, and optimizing their performance is imperative. The control systems of these cooling solutions play a pivotal role in driving energy optimization, as evidenced by absorption cooling machines employed to address technical and financial challenges [11].

The aforementioned challenges highlight the complexity of energy optimization in data centers, necessitating innovative solutions that extend beyond traditional methods. Herein lies the role of artificial intelligence (AI), a field that has demonstrated its efficacy in handling complex challenges. Recent studies have harnessed AI to tackle intricate real-world issues, ranging from utilizing deep learning for disease outbreak prediction [12] to crafting intelligent content distribution systems [13]. The integration of AI into networking and the Web of Vehicles (IoV) has also been touted as a transformative step [14], [15].

This paper underscores the role of AI in the pursuit of energy-efficient cloud data centers, advocating for AI-enabled green cloud infrastructure. This entails the deployment of a scheduling control engine and an intelligent refrigeration engine, each hailing from the domain of AI. These engines collaboratively address the root causes of energy inefficiency by optimizing resource allocation and cooling mechanisms. Reinforcement learning and combinatorial optimization techniques empower the engines to make autonomous decisions in complex environments, enriched by context recognition, demand prediction, and scheduling control mechanisms. AI, in this context, emerges as a catalyst for revolutionizing cloud computing.

In conclusion, this paper's primary contributions are threefold:

1. Introduction of a scheduling control engine and an intelligent refrigeration engine.
2. Integration of AI into the scheduling and refrigeration engines to address complex and dynamic resource environments, enhancing energy efficiency within data centers.
3. Proposal of an AI-enabled green cloud data center architecture, featuring the scheduling control engine, and validation of the design's feasibility through experimentation.

The subsequent sections delve deeper into these contributions. Section II outlines the architecture of the AI-enabled Green Cloud, delving into the intricate details of the scheduling control engine and the intelligent refrigeration engine. Section III presents the prediction model and resource allocation model employed by the scheduling control engine. Section IV introduces the experimental testbed, evaluating the system's scheduling latency and its impact on energy consumption optimization. Finally, Section V offers a comprehensive

summary of the paper's key insights and contributions. Through this exploration, it is evident that AI is a transformative force in the journey toward energy-efficient cloud data centers.

Architecture

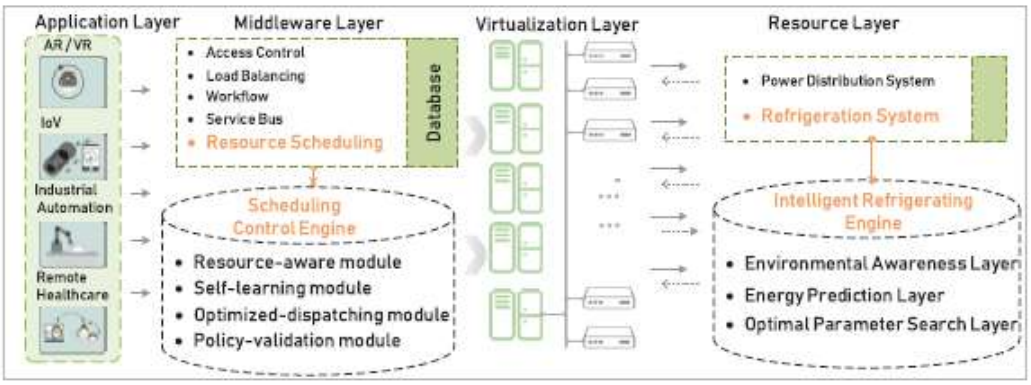

FIGURE 1. The architecture of ai-enable green cloud.

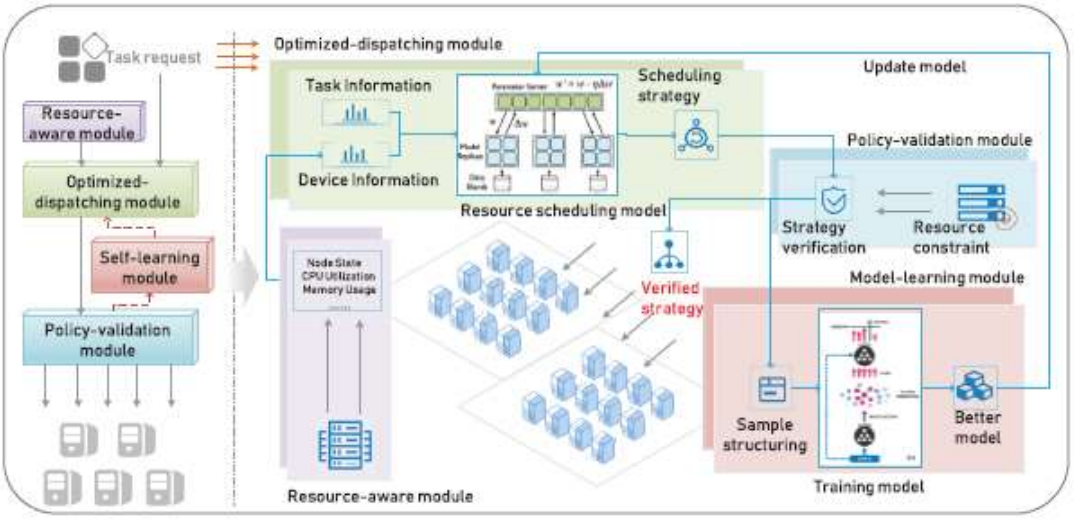

FIGURE 2. The structure of scheduling control engine.

temperatures within the equipment room. Given the high-density nature of cloud data centers, where heat generation is considerable, a robust refrigeration system becomes crucial. The energy consumption attributed to refrigeration constitutes one-third of the total energy consumption in a data center. To enhance energy efficiency and performance, this paper introduces an intelligent refrigerating engine, powered by deep learning, to analyze energy consumption patterns, forecast usage, and implement smart refrigeration strategies.

Environmental Perception Layer: Before the intelligent refrigerating engine optimizes data center cooling, it needs to acquire environmental information. The climate perception layer continuously captures real-time environmental data, including temperature, humidity, airflow, and the room's dimensions. Simultaneously, it obtains resource and equipment status information from the asset perception module within the scheduling engine. This collected data serves as input for the intelligent refrigerating engine.

The architecture of the intelligent refrigerating engine is further illustrated in Figure 3, with each functional layer elaborated upon. By intelligently analyzing environmental data, predicting energy usage, and implementing efficient cooling strategies, the intelligent refrigerating engine contributes significantly to optimizing energy consumption and ensuring the seamless operation of cloud data centers.

Ideal Boundary Search Layer: The size and structure of the machine room are fixed and can be statically set during installation. Dynamic data such as temperature and humidity can be obtained through sensors installed in the machine room. The environmental perception layer gathers real-time data from sensors and the resource booking module, integrates them, and forwards them to the energy consumption prediction layer.

Energy Consumption Prediction Layer: This layer utilizes climate and resource status data to predict the current energy consumption of the data center. Traditional algorithms rely on the opportunity to observe data and analyze the significance level between data and data center energy consumption. Deep learning is employed in this paper for energy consumption modeling without the need for explicit model learning. Initially, data from the data center is used to train a deep neural network model. This data corresponds to the data acquired from the resource perception module. The purpose of the model is to establish the relationship between energy consumption and environmental data. Energy consumption at any given time can be predicted based on real-time weather the issue of energy wastage in data centers is starkly evident, where physical servers, often in substantial numbers, are underutilized for storing and processing data. This inefficiency stems from the absence of prudent resource allocation, resulting in low asset usage rates. Such energy squandering is a matter of genuine concern, as a considerable number of physical servers dedicated to data storage and processing are not being utilized efficiently.

This paper's scheduling control engine is intricately designed to address the dual challenge of resource underutilization and low completion investment efficiency. In the context of multi-target resource allocation, the scheduling control engine ensures timely response to users' requests, elevates asset utilization rates and concurrently reduces energy consumption. The architecture of the scheduling control engine is presented in Figure 2, illustrating the sequence of operations involved.

As depicted in the figure, when a computational task triggers a resource request, the enhancement scheduling module is engaged to secure the necessary information about each device in the data center from the resource sensing layer. Armed with this data, the enhancement scheduling module employs AI-related technologies to fuse real-time task requirements and asset statuses, resulting in optimal resource allocation. Subsequently, the scheduling strategy is shared with the verification module to undergo validation. Upon successful validation, the approved scheduling strategy is executed, effectively enabling the cloud data center to perform efficient resource allocation. This approach maximizes the utilization of server resources, subsequently idling unused resources to minimize power consumption. This comprehensive workflow embodies the functionality of the scheduling control engine.

Furthermore, the self-learning module within the architecture continuously monitors the IT equipment load within the data center. It collects pertinent information and incorporates it into the experience base. The experience base undergoes constant training and refinement, progressively bolstering the perception model of asset intelligence. In tandem with the mentioned functionalities, the scheduling control engine consists of four key modules: the asset intelligence module, the scheduling enhancement module, the scheduling verification module, and the self-learning module.

In conclusion, the scheduling control engine is a pivotal component in addressing the prevailing energy inefficiencies in data centers. Optimizing resource allocation through AI-powered insights, not only ensures higher asset usage rates and efficient completion of tasks but also contributes to the overarching goal of energy conservation and sustainability. This engine's multifaceted approach, as illustrated in the architecture, holds promise in reshaping the landscape of data center resource management, offering a path toward greener and more resource-efficient cloud environments.
In the pursuit of effective resource utilization and reduction of energy consumption, the scheduling control engine plays a pivotal role. The comprehensive functionality of each module within the scheduling control engine is described below, as depicted in Figure 2.

Asset Intelligence Module

The initial step in achieving efficient resource allocation is understanding the status of each hardware or application within the data center. The asset intelligence module actively monitors and maintains the data regarding the entire data center's resources. Real-time updates on the status of each physical asset (operational or idle) and the available resources of each server or virtual node are provided by resource nodes. This information includes variables such as CPU utilization, memory usage, disk capacity, and projected uptime. To forecast the upcoming environment, deep learning techniques are employed. The scheduling control engine then judiciously allocates resources based on current and anticipated conditions, optimizing energy usage.

Prediction Model in Asset Perception Module

The asset perception module employs a prediction model to anticipate the data center's load shortly. This model, in turn, guides the scheduling enhancement module in resource allocation decisions. Accurate prediction of the data center's load is pivotal, as it directly influences the efficacy of resource allocation. Frequent interactions between the self-learning and asset intelligence modules occur. Their primary functions include recording the current resource statuses and continually relearning the asset intelligence module's prediction model. As real-time responsiveness to user requests is crucial, the in-service learning of the prediction model is impractical. Therefore, the self-learning module exclusively handles the prediction model's learning process.

Scheduling Enhancement Module

Traditional resource scheduling algorithms, such as First Come First Served (FCFS) or priority-based scheduling, primarily focus on timing aspects of resource allocation. The scheduling enhancement module, however, takes into account both task completion times and multi-target scheduling strategies for energy efficiency. Cloud task scheduling is intricate and real-time, making it impractical to search for the optimal combination using conventional methods. Leveraging reinforcement learning, the scheduling enhancement module continuously learns from each scheduling instance, amassing intelligence to guide combination optimization. By combining heuristic algorithms and reinforcement learning, this approach accelerates the search for optimal scheduling solutions.

Method Verification Module

The final step before execution is to validate the scheduling strategy. Ensuring the sufficiency of the scheduling enhancement module is critical before its deployment. Given the data center's complexity, it is risky to implement the module without rigorous verification. The verification module employs fundamental capacity conditions to scrutinize the scheduling strategy and guarantee the scheduling engine's reliability. For instance, the verification module ensures that the aggregate computing power used for examining existing and pre-assigned tasks on a computing node doesn't exceed its real-time available computing power. If the enhancement model's performance isn't up to the mark, traditional scheduling algorithms are temporarily employed until the enhancement model achieves optimal competence.

Intelligent Refrigerating Engine

A data center's operational processes, data storage, and computation are facilitated through server clusters, generating substantial heat during computation tasks. Failing to dissipate this heat promptly could lead to reduced computing power due to elevated conditions. For instance, Power Usage Effectiveness (PUE) is utilized as an indicator of energy consumption efficiency. The energy consumption layer forms a PUE prediction model based on real-time weather conditions. Previously, the PUE model employs deep learning to learn from and analyze vast amounts of data, extract significant features influencing energy consumption, and establish the correlation between environmental data and the PUE energy consumption indicator.

Boundary Optimization in Refrigeration System

The boundary optimization problem in the refrigeration system is a nonlinear problem requiring global optimization. Intelligent optimization algorithms can solve this problem efficiently. Therefore, this paper

employs intelligent optimization algorithms to address the optimal boundary set in the refrigeration system. In the ideal boundary search, the data from the environmental perception layer serves as input data, the energy consumption prediction model acts as the optimization algorithm's objective function, and a simulated optimization process yields the ideal boundary set. This set of ideal boundaries is then sent to the refrigeration control system as control commands. It's important to note that the refrigeration engine differs from the scheduling engine and is not prompted by computing tasks. Thus, the engine utilizes a time-triggering mechanism for intelligent refrigeration control. When the timer is activated, the environmental perception layer acquires real-time and predicted data from the data center and resource perception layer. The energy consumption prediction model is considered the objective function at the ideal boundary layer. An optimization search method is employed for global optimization and the generation of a set of optimal boundaries. Finally, this set of optimal boundaries is transmitted to the control system, which adjusts the parameters of the refrigeration equipment accordingly.

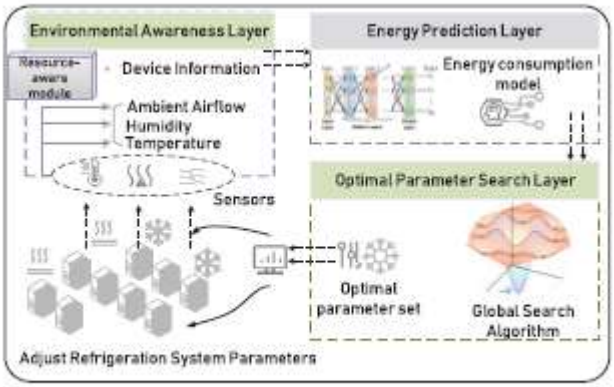

FIGURE 3. The architecture of cloud refrigeration engine.

Resource Planning of Cloud Data Center

In this section, we primarily introduce the LSTM-based (Long Short-Term Memory-based) predictive model and the RL-based (Reinforcement Learning-based) decision model associated with the Scheduling Control Engine.

LSTM-Based Predictive Model

For global resource planning of a cloud data center, it's essential to estimate the workload of each resource node in the next period. The workload prediction of the data center can be simplified as the prediction of each resource node's workload. This paper thus introduces a predictive model for the data center workload, which can be regarded as a time-sequential Recurrent Neural Network (RNN) model. RNNs have a strong capability to capture deep semantic expression and explore temporal sequence information within data. However, RNNs tend to have poor prediction effects for large changes in workload. To address this, we employ LSTM to enhance RNN by adding structures that maintain long-term memory, thereby reducing the prediction model's dependence on anomalous data. At time t, the prediction model based on LSTM is expressed as follows:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h)$$
$$o_t = \sigma(W_y h_t + b_y)$$

Here, x_t represents the input vector at time t, h_t is the hidden state at time t, o_t is the output at time t, σ is the sigmoid function, W_h is the weight matrix from input to hidden layer, U_h is the weight of the self-recurrent connection in the hidden layer, b_h is the bias, W_y is the weight matrix from hidden layer to the output layer, and b_y represents the output bias. The hidden state h_t at time t is determined by the previous state h_{t-1} and current input x_t. The output o_t at time t is based on the current hidden state h_t. Time series data serves as the input data for the RNN, and the output data from the application output layer serves as the prediction for the next time step.

RL-Based Decision Model: In this paper, the Discrete Particle Swarm Optimization (DPSO) algorithm is used to search for the optimal configuration of resource allocation. Given M resource nodes and expected N

tasks to be assigned, the goal of resource allocation is to distribute N tasks among M resource nodes, optimizing task completion times and energy consumption. The distribution grid and velocity grid can be represented as follows:

\[X_i = [x_{11}, x_{12}, ..., x_{1m}, ..., x_{nm}] \]
\[V_i = [v_{11}, v_{12}, ..., v_{1m}, ..., v_{nm}] \]

Here, \(x_{ij} \) represents the allocation of the ith task to the jth server, and \(x_{ij} \) takes values from the set \(\{0, 1\}\). \(v_{ij} \) represents the velocity of particle \(x_{ij} \), and to ensure the velocity of the particle lies within the range [0, 1], the sigmoid function \(\sigma \) is used to map \(v_{ij} \) to this range.

The fitness function and the position and velocity update equation are as follows:

Fitness Function: \(F = T_{cost} + P_{cost} \)
Position Update Equation: \(v_{ij}^{k+1} = wv_{ij}^k + c_1(p_{ij}^k - x_{ij}^k) + c_2(g_{j}^k - x_{ij}^k) \)

In these equations, \(T_{cost} \) is the total task completion time, \(P_{cost} \) is the total power consumption, \(p_{ij}^k \) represents the personal best position of the particle, \(g_j^k \) represents the global best position of the particle swarm, \(c_1 \) and \(c_2 \) are learning factors, \(w \) is the inertia weight, and \(k \) represents the iteration number. The particle velocity is bounded within the range [−\(v_{max} \), \(v_{max} \)].

However, DPSO's search process from particle initialization to optimization takes a long time due to dynamic conditions. To address this, reinforcement learning is applied in this paper to learn from each booking experience and accumulate knowledge. This greatly reduces the search time. Q-learning is employed to learn the expected reward after taking a specific action in a specific state. The Q-table records state-action pairs and is updated in each iteration. The Bellman Equation is used to update it. The Q-learning process involves assessing the current state's similarity with remembered states using a value-based evaluation technique. If their similarity reaches a certain threshold, the corresponding action is triggered; otherwise, the new state is added to the Q-table. This approach gives Q-learning predictive and exploratory capabilities.

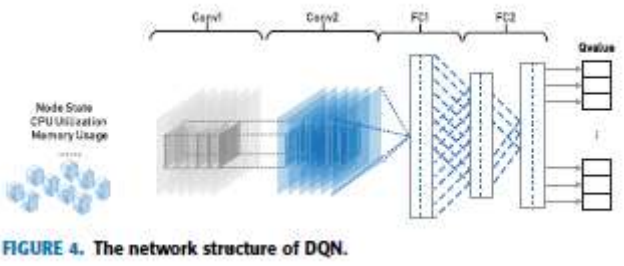

FIGURE 4. The network structure of DQN.

Deep Q Network (DQN): To establish the Q-table, the Deep Q Network (DQN) is employed, allowing end-to-end Q-fitting based on a convolutional neural network architecture.

The input data for the RL-based decision model consists of the current environmental state, the LSTM-predicted environmental state, and the requested task. These inputs are fed into a convolutional network, which mines the underlying nonlinear planning relationship deeply. The output value is the Q-value for each action in the current state, forming a vector of size 1xK, where K represents the number of actions in Q-learning. Actions with significant Q-value differences are prioritized in the action process. The initial positions of particles are selected based on the action process with large action differences. These initialized particles are strong solutions with natural variations. As a result, each unique particle corresponds to a locally optimal solution in different positions. The global optimal solution can be found by searching through particles, which accelerates the search process.

TABLE 1. Cloud computing simulation environment setting.

	Values	
SCHEDULING INTERVAL	300	
VIRTUAL MACHINE	TYPES	50
	MIPS	2500
	SIZE	2.5GB
	BANDWIDTH	100 Mbit/s
HOST MACHINE	TYPES	50
	MIPS	2500
	BANDWIDTH	1 Gbit/s
	STORAGE	1GB

Testbed

To verify the RL-based decision model, the proposed planning control engine is deployed in the CloudSim cloud computing simulation environment. CloudSim, written in Java, offers cloud computing features that support resource management and planning simulations. It enables researchers to avoid the complexities of real-world deployment, allowing the simulation of large-scale cloud clusters and the testing of corresponding algorithms on a single machine.

TABLE 2. The main parameter settings of the DPSO algorithm.

DPSO Algorithm	Parameter Settings
Particle swarm size P	P=30
Weight factor α	$\alpha = 0.5$
Learning factors c_1,c_2	$c_1 = 2$,$c_2 = 2$
Inertia factor w	$w = 1$
Maximum iterations $Maxiter$	$Maxiter = 1000$

CloudSim Simulation Environment: The first experiment investigates the CloudSim simulation environment. Request arrivals follow a Poisson distribution. The simulation environment parameters for Analysis 1 are presented in Table 1. The setup includes 50 physical machines and 50 virtual machines, each with the same configuration and a MIPS (Million Instructions Per Second) of 2,500.

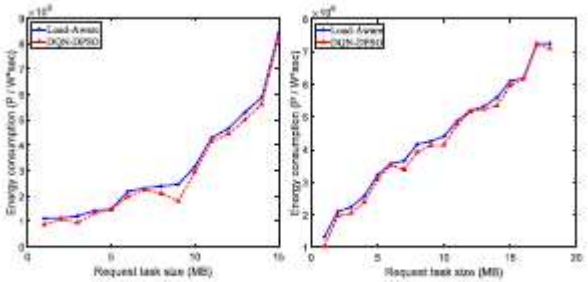

FIGURE 5. Experimental results for the performance evaluation of the energy consumption in cloudsim.

Furthermore, the reliability and security of the system are crucial, considering the dynamic and variable nature of real-world requests. To rigorously validate the model's reliability, we deploy the cloud data center platform in the Inspur data center to assess its stability in a real environment. Inspur's large data center comprises 2 management nodes and 7 data nodes, capable of storing 253 TB of data. It can be considered a small-scale data center, making it an effective platform for evaluating the availability of our model.

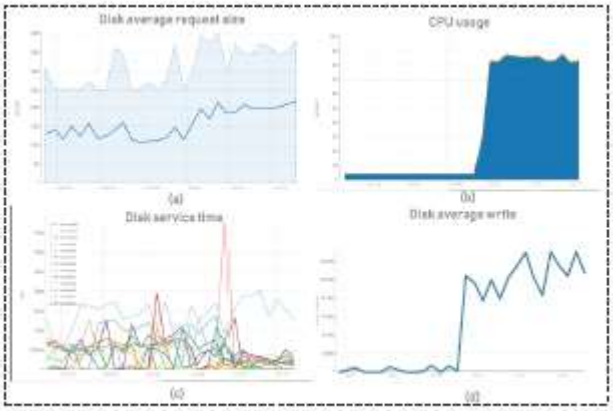

FIGURE 6. Real-time situation of wave data platform deployed with scheduling control engine.

Based on the configured environment, we initially simulate data center tasks on the simulation platform with the set parameters. The sample dataset is prepared to train a predictive model. The goal is to predict the number of tasks in the next time slot. Specifically, the data from the first 5 time slots are used to predict the number of tasks in the subsequent time slot, resulting in a vector of size 6 for each training instance.

Table 2 presents the key parameters used in the DPSO algorithm for both analyses.

Comparison of RL-Based Decision Model: We utilize a cloud computing simulation tool to compare the RL-based decision model with the resource allocation algorithm based on Load Aware. This comparison aims to evaluate the energy consumption optimization performance of the resource allocation based on the RL-based decision model. Figure 5 depicts the energy consumption of the CloudSim cloud platform using both resource allocation algorithms.

Figure 5 showcases the results of Analysis 1. The x-axis represents the request size, while the y-axis shows the average power consumption for executing the requested tasks. For clarity, we arrange the task sizes in ascending order and conduct two tests. As shown in the figure, our model demonstrates a superior impact on energy consumption optimization compared to the load-aware resource allocation algorithm.

Usability Testing on the Testbed: Subsequently, we evaluate the model's usability on the lab's Testbed. We load 5 computation tasks in the data center and monitor disk access, service time, and CPU utilization. Figure 6 displays the real-time status of the CPU and disk in the Inspur data center. Throughout the analysis, we gradually increase the number of computation tasks. As evident from Figure 6(a-d), the average number of disk requests consistently increases. There is minimal idle time during high CPU utilization and disk load balancing is achieved without persistent free disks. Based on the experimental results, the data center equipped with the planning control engine operates stably even under varying request loads.

Conclusion

CloudSim Simulation Environment: The first experiment investigates the CloudSim simulation environment. Request arrivals follow a Poisson distribution. The simulation environment parameters for Analysis 1 are presented in Table 1. The setup includes 50 physical machines and 50 virtual machines, each with the same configuration and a MIPS (Million Instructions Per Second) of 2,500.

Furthermore, the reliability and security of the system are crucial, considering the dynamic and variable nature of real-world requests. To rigorously validate the model's reliability, we deploy the cloud data center platform in the Inspur data center to assess its stability in a real environment. Inspur's large data center comprises 2 management nodes and 7 data nodes, capable of storing 253 TB of data. It can be considered a small-scale data center, making it an effective platform for evaluating the availability of our model.

Based on the configured environment, we initially simulate data center tasks on the simulation platform with the set parameters. The sample dataset is prepared to train a predictive model. The goal is to predict the number of tasks in the next time slot. Specifically, the data from the first 5 time slots are used to predict the number of tasks in the subsequent time slot, resulting in a vector of size 6 for each training instance.

Table 2 presents the key parameters used in the DPSO algorithm for both analyses.

Comparison of RL-Based Decision Model: We utilize a cloud computing simulation tool to compare the RL-based decision model with the resource allocation algorithm based on Load Aware. This comparison aims to evaluate the energy consumption optimization performance of the resource allocation based on the RL-based decision model. Figure 5 depicts the energy consumption of the CloudSim cloud platform using both resource allocation algorithms.

Figure 5 showcases the results of Analysis 1. The x-axis represents the request size, while the y-axis shows the average power consumption for executing the requested tasks. For clarity, we arrange the task sizes in ascending order and conduct two tests. As shown in the figure, our model demonstrates a superior impact on energy consumption optimization compared to the load-aware resource allocation algorithm.

Usability Testing on the Testbed: Subsequently, we evaluate the model's usability on the lab's Testbed. We load 5 computation tasks in the data centre and monitor disk access, service time, and CPU utilization. Figure 6 displays the real-time status of the CPU and disk in the Inspur data center. Throughout the analysis, we gradually increase the number of computation tasks. As evident from Figure 6(a-d), the average number of disk requests consistently increases. There is minimal idle time during high CPU utilization and disk load balancing is achieved without persistent free disks. Based on the experimental results, the data center equipped with the planning control engine operates stably even under varying request loads.

References

1. L. Hou et al., "Internet of Things cloud: Architecture and implementation," IEEE Commun. Mag., vol. 54, no. 12, pp. 32-39, Dec. 2016.

2. M. Chen, J. Yang, X. Zhu, X. Wang, M. Liu, and J. Song, "Smart home 2.0: Innovative smart home system powered by botanical IoT and emotion detection," Mobile Netw. Appl., vol. 22, no. 6, pp. 1159-1169, 2017.

3. M. Chen, J. Zhou, G. Tao, J. Yang, and L. Hu, "Wearable affective robot," IEEE Access, vol. 6, pp. 64766-64776, 2018.

4. W. Xiang, N. Wang, and Y. Zhou, "An energy-efficient routing algorithm for software-defined wireless sensor networks," IEEE Sensors J., vol. 16, no. 20, pp. 7393-7400, Oct. 2016.

5. L. Zhou, D. Wu, J. Chen, and Z. Dong, "Greening the smart cities: Energy-efficient massive content delivery via D2D communications," IEEE Trans. Ind. Informat., vol. 14, no. 4, pp. 1626-1634, Apr. 2018.

6. K. Bilal, S. U. Khan, and A. Y. Zomaya, "Green data center networks: Challenges and opportunities," in Proc. Int. Conf. Frontiers Inf. Technol., Dec. 2013, pp. 229-234.

7. Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, "Prometheus: Privacy-aware data retrieval on hybrid cloud," in Proc. IEEE INFOCOM, Turin, Italy, Apr. 2013, pp. 2643-2651.

8. Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving efficient and secure data acquisition for cloud-supported Internet of Things in smart grid," IEEE Internet Things J., vol. 4, no. 6, pp. 1934-1944, Dec. 2017.

9. Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (IPTV): The killer application for the next-generation Internet," IEEE Commun. Mag., vol. 45, no. 11, pp. 126-134, Nov. 2007.

10. Q. Deng, D. Meisner, and A. Bhattacharjee, "CoScale: Coordinating CPU and memory system DVFS in server systems," in Proc. IEEE/ACM Int. Symp. Microarchitecture, Dec. 2012, pp. 143-154.