



*Research article*

## **Task arrival based energy efficient optimization in smart-IoT data center**

**Bin Wang and Fagui Liu\***

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

\* **Correspondence:** Email: [fgliu@scut.edu.cn](mailto:fgliu@scut.edu.cn); Tel: + 862039380283; Fax: + 862039380252.

**Abstract:** With the growth and expansion of cloud data centers, energy consumption has become an urgent issue for smart cities system. However, most of the current resource management approaches focus on the traditional cloud computing scheduling scenarios but fail to consider the feature of workloads from the Internet of Things (IoT) devices. In this paper, we analyze the characteristic of IoT requests and propose an improved Poisson task model with a novel mechanism to predict the arrivals of IoT requests. To achieve the trade-off between energy saving and service level agreement, we introduce an adaptive energy efficiency model to adjust the priority of the optimization objectives. Finally, an energy-efficient virtual machine scheduling algorithm is proposed to maximize the energy efficiency of the data center. The experimental results show that our strategy can achieve the best performance in comparison to other popular schemes.

**Keywords:** energy efficiency; task arrival; green data centers; smart cities; VM placement

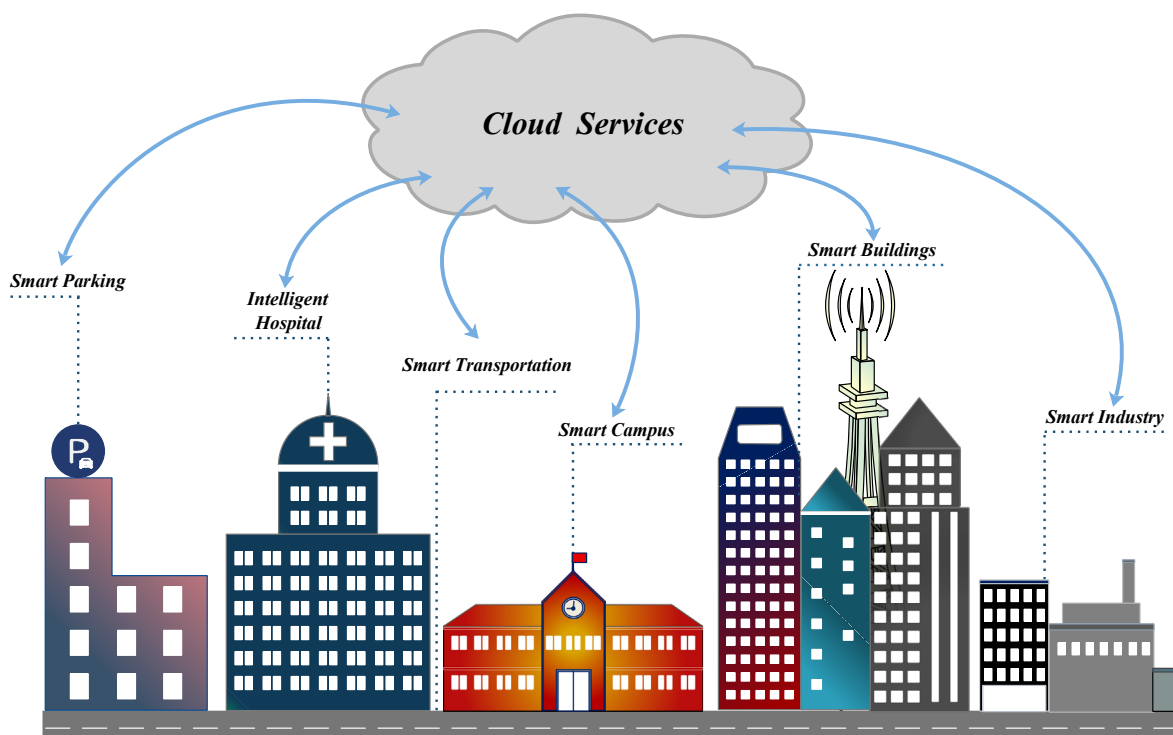
---

### **1. Introduction**

Due to the rapid development of technologies such as cloud computing, Internet of Things, and artificial intelligence, smart cities have become more than just a new concept in the past few years [1]. According to BCC Research report [2], the market size of smart cities has reached 342.4 billion in 2016 and is expected to reach 774.8 billion in 2021. The rapid growth of market size provides vast opportunities for governments and enterprises [3]. A smart city is a new form of urban development, which deeply integrated with digital technology. It can improve the efficiency of urban services and reduce resource consumption.

Smart cities utilize cloud computing and IoT to enhance urban infrastructure construction, enabling cities to achieve smart collaboration, resource sharing, interconnection, and comprehensive

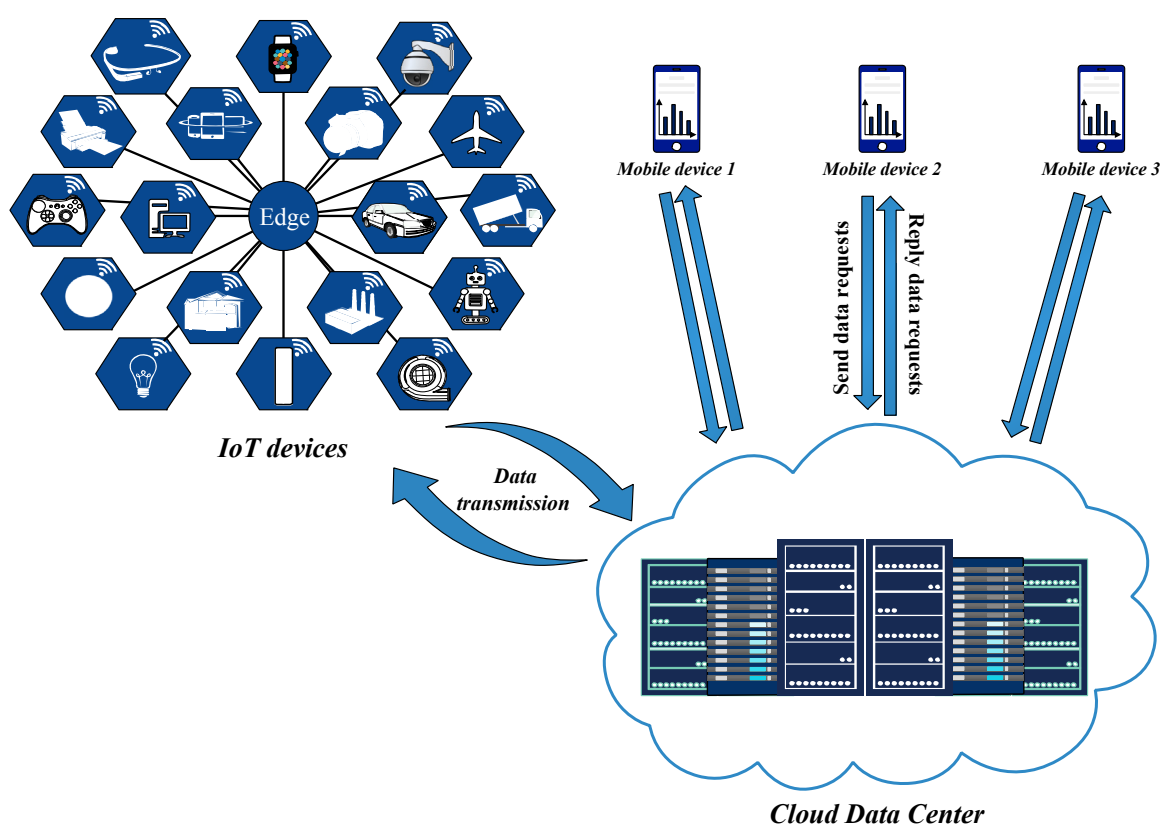
perception [4]. As shown in Figure 1, a smart city is composed of different smart-IoT networks. From the transportation sector to the medical sector, from smart campus to smart industry, the concept and technology of smart city have integrated into all fields of modern society, while closely linking with cloud computing platforms. More specifically, Figure 2 shows the data transmission architecture diagram between IoT devices and the data center in a smart city. The role of edge computing nodes here is like the nerve endings of the smart city brain. The edge nodes collect data from IoT devices nearby and transmit further requests that need to be processed to the cloud brain. As the brain center of smart cities, cloud computing analyzes and processes a large number of requests that cannot be processed by edge computing. With the increasing scale and complexity of smart city [5], the sensor networks scattered throughout the city will produce exponentially scaled data, which makes data centers become the backbone in smart city development. Cloud data centers need a customized resource management strategy to balance current demand and exponential growth of data. An inefficient data center will cause serious energy consumption problems and cannot guarantee the quality of service (QoS) (e.g., response time, latency, and throughput) [6]. In general, the QoS requirements of a cloud service can be specified in terms of a service level agreement (SLA) [7, 8]. Furthermore, with the growth and expansion of cloud data centers, energy consumption has become an urgent issue for smart cities system [9]. In order to prevent high energy consumption from becoming a bottleneck restricting the development of smart cities, high energy consumption problems need to be resolved.



**Figure 1.** Smart-IoT sectors in a smart city.

Many researches have proposed several resource management approaches to optimize the energy consumption and SLA of cloud data centers. Toor et al. [10] presented a scheduling framework for allocating jobs in the data center through dynamic voltage frequency scaling (DVFS) techniques.

Kulkarni et al. [11] proposed a green power management method to monitor and assign computing resources efficiently. Zhang et al. [12] proposed an energy-aware allocation algorithm using the conventional expression of Bin-Packing problem. The algorithm aims to reduce the number of used servers and turn off the servers that are in idle state. Aldossary et al. [13] proposed a novel cloud system architecture and energy-aware cost estimation framework to predict the resource usage and energy consumption of Cloud infrastructure services. These studies are looking for the best scheduling framework to fit all scenarios in a cloud data center. However, traditional cloud computing scheduling mechanisms have significant limitations when dealing with IoT requests. These approaches can no longer efficiently handle the massive requests generated by IoT devices in this scenario [14]. Furthermore, due to the high real-time constraints of IoT requests, SLAs in data centers are difficult to guarantee. In this paper, an efficient task arrival model is proposed for real-time prediction of massive IoT requests.



**Figure 2.** Smart City IoT data transmission equipment and data center architecture diagram.

How to achieve the trade-off between energy saving and SLAs is another significant concern for energy-efficient scheduling. Zhou et al. [15] defined energy efficiency as the reciprocal of the product of energy consumption and SLA, Chang et al. [16] minimized the energy consumption of the data center without exceeding the static SLA upper limit values (80 and 90%). Malekloo et al. [17] introduced a multi-objective scheduling mechanism to balance a trade-off among energy efficiency, performance and QoS. In literature [18], a novel multi-dimensional resource usage model and virtual machine (VM) placement algorithm are proposed to improve resource utilization in a balanced way. Since the generation and arrival of IoT requests have moderate fluctuations, the approaches mentioned above do not consider that the priorities of energy-saving and SLA optimization are not

static. SLA optimization is the primary concern of energy efficiency management when the load is high. Correspondingly, the hosts in the idle state need to be consolidated to save energy when the load is low.

In this paper, we concentrate on analyzing and modelling the arrivals of IoT requests. We take into account the massiveness of IoT requests and the stability of IoT load fluctuations. A novel Poisson task model based on upper and lower bound is proposed to predict the future arrival of IoT requests. In order to balance the trade-off between energy consumption and SLA, we introduce an adaptive energy efficiency scheduling policy and propose an energy-efficient VM placement algorithm.

The main contributions of this paper can be summarized as follows:

1) We propose an improved Poisson arrival model to predict the future arrival rates of IoT requests. We take into account the fluctuations of IoT requests and introduce a novel upper and lower bound mechanism to enhance the applicability of the arrival model.

2) We present the energy consumption model, SLA violation metrics and an adaptive energy efficiency model to analyze and monitor the variations in a cloud data center.

3) We propose an energy-efficient VM scheduling algorithm based on task arrival rates to achieve the trade-off between energy saving and SLA.

The rest of the paper is organized as follows. Section 2 discusses related work on energy-efficient technology analysis. Section 3 introduces a novel task arrival model with upper and lower bound. In Section 4, we propose an adaptive resource scheduling strategy. Experiments and results are given in Section 5. Section 6 concludes the paper with a summary and future research directions.

**Table 1.** Literature summary of energy-efficient optimization techniques.

Main uses	Project	Reliability	Load balancing	Energy	SLA	Description
<i>DVFS</i>						
Energy-aware DVFS	[19,30]	No	No	Yes	No	DVFS achieves energy saving by dynamically adjusting the operating frequency and voltage of the host CPU.
Hybrid-DVFS	[10]	No	No	Yes	Yes	
<i>Energy-aware heuristics</i>						
Evolutionary algorithm	[20,21]	Yes	Yes	No	Yes	Studies in this group solve multi-objective optimization problems of data centers through heuristic algorithms.
PSO-based	[23]	Yes	Yes	No	Yes	
Krill herd heuristic	[32]	Yes	Yes	Yes	No	
<i>Workload characterization and prediction</i>						
Failure-aware	[25]	Yes	Yes	No	Yes	These methods perform early scheduling by predicting overloaded or abnormal hosts.
Time-series analysis	[26,27]	Yes	Yes	Yes	Yes	
<i>Decentralized management</i>						
Multiagent system	[41,42]	Yes	Yes	Yes	Yes	These methods can optimize load balancing and improve the scalability and robustness of the system.
Trusted nodes	[43]	Yes	Yes	Yes	Yes	
Blockchain assisted	[44,45]	Yes	Yes	Yes	Yes	

## 2. Related works

With the rapid growth of the computing resources and applications, energy-saving is an immediate concern of consumers and industries. An increasing number of researchers devote themselves to the research on how to reduce energy consumption and improve utilization rates of data centers. The energy-efficient optimization can be divided into four categories: dynamic voltage frequency scaling technique [10,19] based on energy efficiency model, energy-aware resource allocation heuristics [20–24], cloud workload characterization and prediction [25–28] and decentralized management method [29]. The comparison of energy-efficient optimization techniques is shown in Table 1, which indicates the characteristics and differences of these strategies.

In [30], Krzywda et al. define a server power consumption model which mainly relying on the DVFS technique. It implies the existence of a cubic relationship between CPU frequency and power consumption. However, the DVFS technique does not consider the optimization of SLAs, which makes this technique has significant limitations in the existing cloud system. In [31], the authors deploy cloud resources in the data center under the client-level SLA requirements to minimize the overall power consumption. This research only concerns about the energy consumption of the environment. It does not take into account the computational performance provided by the cloud system when energy consumption is minimum. The literature [32] uses the Automatic Server Configuration System (ACES) to reduce energy consumption or improve energy efficiency to meet the load requirements to solve problems, thereby improving system energy efficiency. The author of the literature [33] proposes an online algorithm to reduce the energy consumption of the data center as the starting point, dynamically adjust the load of each central server and shut down some servers when the load is low. This form reduces the overall energy consumption for load balancing. In [34], an energy efficiency model for convenient calculation is proposed for the existence and condition of the maximum energy efficiency. In addition, through the necessary measurement and monitoring research on the system energy consumption to achieve research and management of energy consumption in the data center, the relevant results are more significant with the three research teams represented by Kansal et al. [35]. Focusing on the measurement of energy consumption in the cloud computing environment, the literature [36] introduces an energy efficiency model that can be achieved by the unit energy consumption of the data center. To define the performance of the system, it provides a robust basis for optimizing energy efficiency.

Buyya et al. [7] find that an idle server's energy consumption is equivalent to 70% of a fully-loaded server's. Based on this fact, they redesign the power consumption model of the data center. The CPU utilization of the server will change over time, and the authors define total energy consumption as an integral function in a time slice. Based on the power consumption model, virtual machines can be dynamically reallocated depending on the current CPU utilization, and it is more energy-efficient than a static method. Balasubramanian et al. [37] design an online control policy framework to improve resource scheduling efficiency in the cloud environment. This framework optimizes traffic allocation in each time slice and reduces the scheduling queue lengths to relieve network congestion in the environment. Xu et al. [38] propose an energy-aware virtual machine allocation policy which can dynamically schedule the VMs to minimize energy consumption. The policy first selects VMs that need to be migrated and then places the chosen VM on the host using the Modified Best Fit Decreasing (MBFD) algorithm. The authors set lower and upper utilization thresholds and keep the CPU utilization of hosts within the range of dual thresholds. Experiment results show that dual thresholds policy is

helpful to energy-efficiency and avoiding service level agreement violation. Hussain et al. [39] define an energy-aware service framework for cloud service providers. It enables cloud providers to manage the resources and services of the cloud platform efficiently.

In literature [40], the authors believe that a computing framework that combines cloud computing and ambient intelligence has become a key component of future Internet development. This framework enables the cloud environment to adapt to the demands of users and realizes adaptive dynamic resource management. Prieta et al. [41] describe the characteristics of multiagent system (such as dynamics, flexibility, and self-learning). The multiagent based cloud computing model can effectively allocate computing demands among nodes without global coordination of the system. The literature [42] believes that the variability of service demands in the cloud computing environment makes the decision-making process of resource allocation difficult. So they proposed a Multiagent system based on a virtual organization (VO) to manage the resources in cloud environments automatically. This method has global self-adaptability and can reduce the computational load of nodes in the resource management decision-making process. Tseng et al. [43] demonstrate the benefits of setting up trusted nodes when studying scheduling problems through several practical cases. The authors introduce that the trusted nodes can reduce the energy consumption and latency while maintaining the same resilience effect. In [44,45], the authors analyze the limitations of traditional centralized systems in resource allocation and management and introduce the advantages of resource optimization based on blockchain. These methods use decentralization to reduce the load of a centralized system. Experimental results show that the decentralized management method can improve the scalability and robustness of the system and effectively cope with the occurrence of failures.

Most of the existing energy efficiency models consider energy consumption to be as crucial as SLA. These researches are all concentrated on seeking the best framework to fit all constraints of a cloud data center in common scenarios. Since the generation and arrival of IoT requests have moderate fluctuations, the traditional energy efficiency models are no longer suitable in these scenarios.

### 3. Task arrival model

Suppose there is a set of IoT requests tasks with different constraints. Each task can be assigned to any given virtual machine. Assume that tasks occur randomly over a while and satisfy the following conditions: 1) the time interval can be infinitely divided into plenty of small time slices. The probability of the requests arriving once is proportional to the length of the time slice; the probability of more occurrences of the request is always equal to zero in each minimum interval. The requests arrive independently in different time slices. The probability of IoT requests arriving in each tiny time slice is independent of each other. Table 2 shows the summary of the main notations used in the paper.

In a cloud computing platform, within a short interval of time  $\Delta t$ , we assume that the probability of a new cloud task arriving is assumed to be  $\mu\Delta t$ . If  $\Delta t$  is small enough, the probability of reaching two or more tasks in  $\Delta t$  time is negligible. Suppose that the scheduling interval gap  $(t, 2t)$  is divided into  $n$  sub-time slices, each sub-time slice length is  $t/n$ . We further assume that the arrival of tasks in a given time slice is independent of the arrival of tasks in other time slices. Then, when  $n$  is large enough, the  $n$  time slices can be considered to constitute a Bernoulli independent test sequence, and the probability of a successful test is  $p = \mu t/n$ . From this, it can be seen that  $n$  sizes are  $t/n$ . The probability of having  $k$  tasks arriving during the time period is approximate:

$$\varphi\left(p; n, \frac{\mu t}{n}\right) = \binom{n}{p} \left(\frac{\mu t}{n}\right)^p \left(1 - \frac{\mu t}{n}\right)^{n-p}, \quad (1)$$

**Table 2.** Notation used in the article.

Symbol	Explanation
$t, \Delta t$	The scheduling interval gap, The small time slice
$n$	The number of sub-time slices
$\varphi$	The probability of tasks arriving during the time period
$\lambda$	The arrival rate of tasks in current time slice
$P_n^{cloud}$	The probability distribution of IoT devices requests
$P_0^\alpha(t)$	The probability distribution of zero task arrived at t moment
$\widehat{x}_m, x_{median}$	The predicted amounts of requests, The median value of series
$\sigma, \xi$	The standard deviation of the series, The scale level coefficient
$P_{host}^k$	The power of host $k$
$E_\tau^{dc}$	The real-time energy consumption of a cloud data center
$N_{vi\_allocMIPS}$	The MIPS value of processing the task per second
$N_{hj\_core}$	The number of CPU cores of host $j$
$N_{hi\_MIPS}$	The MIPS value of the current host $j$
$U_{host}^j$	The utilization model of host $j$
$E_{interval}^{dc}$	The total energy consumption generated by hosts
$\Theta_{hj}$	Energy consumption corresponding to the utilization of host $j$
$V_{SLA}^{dc}$	The SLA violation rate of data center
$T_{hj}$	The SLA violation time
$N_{hi\_MIPS\_not\_satisfied}$	The MIPS value that does not satisfy the resource requirement
$F, M$	Constraint problem, Energy efficiency model
$\alpha, \gamma$	The weight coefficients of the energy efficiency model
$\mu$	The average arrival rates of historical workload
$\hat{r}$	The standard deviation of historical arrival rates

We observe the probability of reaching  $p$  tasks in time  $t$ . During the time period  $(0, t]$ , we observe the arrival of tasks in a cloud computing platform server. In a short time slice  $\Delta t$ , the probability of a new task arriving is assumed to be  $\mu \Delta t$ .  $\mu$  is a fixed value, and the size depends on the total number of users on the server. If  $\Delta t$  is small enough, the probability of reaching two or more tasks in  $\Delta t$  time will be tiny (one or no arrival) that it is negligible.

$$\varphi\left(p; n, \frac{\mu t}{n}\right) = \frac{n(n-1)\cdots(n-p+1)}{n^p} \frac{(\mu t)^p}{p!} \left(1 - \frac{\mu t}{n}\right)^{-p} \left(1 - \frac{\mu t}{n}\right)^n. \quad (2)$$

Therefore, take the limit value of  $n$  for the Eq (2), let  $n \rightarrow \infty$ , now the equation can be transformed as:

$$\lim_{n \rightarrow \infty} \varphi\left(p; n, \frac{\mu t}{n}\right) = \lim_{n \rightarrow \infty} \left[ \frac{n(n-1)\cdots(n-p+1)}{n^p} \frac{(\mu t)^p}{p!} \left(1 - \frac{\mu t}{n}\right)^{-p} \left(1 - \frac{\mu t}{n}\right)^n \right] = \frac{e^{-\mu t} (\mu t)^p}{p!}. \quad (3)$$

Let  $\lambda$  denotes  $\mu t$ , according to  $p = \mu t/n$ , so  $\lambda = np$ , the Poisson distribution cloud task access request probability model is obtained:

$$f(p; \lambda) = e^{-\lambda} \frac{\lambda^p}{p!}. \quad (4)$$

**Lemma 1.** For any feasible  $\lambda t$ , the probability distribution of IoT devices requests number can be expressed as follows:

$$P_n^{cloud}(t) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}, \forall n \in N. \quad (5)$$

**Proof of Lemma 1.**

We define a minterm  $\theta(\mu)$  which satisfies:

$$\lim_{\mu=0} \frac{\theta(\mu)}{\mu} = 0. \quad (6)$$

When each time slice is partitioned small enough, the probability of cloud task reaching the time slice will be very low. Since the size of  $P_1^\alpha$  is proportional to the length of the time slice, the probability of a cloud task reaching more than 1 at this time will become smaller. The speed of probability approaching 0 is much faster than the reduction of time slice length. Therefore, it is confirmed that the probability that the number of cloud tasks reaching more than 1 is the minterm value. The equation of this probability can be expressed as:

$$P_1^\alpha(\mu) = \lambda\mu + \theta(\mu), \quad (7)$$

$$P_{>1}^\alpha(\mu) = \theta(\mu). \quad (8)$$

When  $n = 0$ , according to the independent definition, the following equations is obtained:

$$P_0^\alpha(t + \mu) = P_0^\alpha(t)P_0^\alpha(\mu). \quad (9)$$

The sufficient and necessary conditions for the number of cloud tasks is 0 at the  $t + h$  moment are given as follows: zero cloud tasks were reached at  $t$  moment and  $(t, t + h)$  interval. We can obtain the following equation regardless of whether any task arrives at the  $t$  moment or not.

$$1 = P_0^\alpha(\mu) + P_1^\alpha(\mu) + P_{>1}^\alpha(\mu). \quad (10)$$

According to Eqs (7)–(9),  $P_0^\alpha(t + \mu)$  can be expressed as:

$$P_0^\alpha(t + \mu) = P_0^\alpha(t)(1 - \lambda\mu + 2\theta(\mu)), \quad (11)$$

$$\frac{P_0^\alpha(t + \mu) - P_0^\alpha(t)}{\mu} = -\lambda P_0^\alpha(t) + \frac{2\theta(\mu)}{\mu} \quad (12)$$

According to the definitions and properties of derivation, Eq (12) can be transformed as:

$$\frac{d(P_0^\alpha(t))}{dt} = \lim_{\mu=0} \frac{P_0^\alpha(t+\mu) - P_0^\alpha(t)}{\mu}, \quad (13)$$



$$= -\lambda P_0^\alpha(t). \quad (14)$$

The constant value  $\lambda$  is shown in Eq (15):

$$\frac{d(P_0^{arrival}(t))}{P_0^{arrival}(t) dt} = -\lambda. \quad (15)$$

Solving the integral of Eq (15), the equation is simplified as follows:

$$\log P_0^{arrival}(t) = -\lambda t + a, \quad a \in R, \quad (16)$$

$$P_0^{arrival}(t) = M e^{-\lambda t}, \quad (17)$$

where  $a$  and  $M$  are constant values. Because the value of  $P_0^\alpha(0) = 1$ , the value of  $M$  is 1, the probability distribution of zero task arrived at  $t$  moment is shown in Eq (18):

$$P_0^\alpha(t) = e^{-\lambda t}. \quad (18)$$

In summary, definitions 1 and 2 are the necessary conditions of the proof. Especially the characteristics of service requests in smart home environment guarantee that the minterm  $\phi(\mu)$  does exist. Based on Eq (18) and mathematical induction, when  $n \geq 1$ , Eq (6) can be deduced. For brevity, the similar proof process is omitted. Lemma 1 is eventually proved.

After demonstrating that the Poisson distribution model is workable for the arrival series of IoT requests, we introduce an effective upper and lower bound mechanism and add it to the Poisson arrival model. The definition of the lower bound and upper bound can be expressed in Eqs (19) and (20):

$$Lower\_Bound = \widehat{x}_m - \left( \frac{1}{m} \sum_{i=1}^m |\widehat{x}_m - x_{median}| + \xi * \sigma \right) \quad (19)$$

$$Upper\_Bound = \widehat{x}_m + \left( \frac{1}{m} \sum_{i=1}^m |\widehat{x}_m - x_{median}| + \xi * \sigma \right) \quad (20)$$

where  $\widehat{x}_m$  denotes the predicted amounts of arrival requests;  $x_{median}$  denotes the median value of historical series;  $\sigma$  denotes the standard deviation of the series;  $\xi$  denotes the scale level coefficient and it is a constant value.

#### 4. Adaptive resource management approach

In this section, we define the energy consumption model and SLA violations metric of the data center. After the previous definition, an adaptive energy efficiency metric model is presented to balance the trade-off between energy consumption and SLA violations. The model can adaptively adjust the weight coefficients of energy consumption and SLA violations through the arrival rates given by the task arrival model.

#### 4.1. Energy consumption model

According to previous researches, a server power can be expressed as a function of power consumption by CPU, memory and disks. Moreover, recent studies show that the CPU power consumption generally dominates the energy model in a Xeon processor-based server. Considering the importance of CPUs for server power, we present a utilization-based energy consumption model for cloud data centers. Let  $N$  denotes the number of physical nodes in the data center, and  $P_{host}^k$  denotes the power of host  $k$  ( $1 \leq k \leq N$ ). It is obvious that energy consumption is the product of time and power. The real-time energy consumption of a cloud data center is defined in Eq (21):

$$E_{\tau}^{dc} = \sum_{k=1}^N \int_0^{\tau} P_{host}^k dt, \quad (21)$$

where  $E_{\tau}^{dc}$  denotes the overall energy consumption of the cloud data center.

Let  $N_{vi\_allocMIPS}$  denotes the Million Instructions Per Second (MIPS) value of the virtual machine executing the cloud task (the ability value of processing the task per second);  $N_{hj\_core}$  denotes the number of CPU cores of host  $j$ ; and  $N_{hi\_MIPS}$  denotes the MIPS value of the current host  $j$ . The value of  $N_{vi\_allocMIPS}$  depends on the allocation result of VMs. The value of  $N_{hj\_core}$  and  $N_{hj\_MIPS}$  are set in advance. Then the utilization model of host  $j$  is shown in Eq (22)

$$U_{host}^j = \frac{\sum_i^j N_{vi\_allocMIPS}}{N_{hj\_core} \cdot N_{hj\_MIPS}}. \quad (22)$$

In the cloud computing scheduling process, resource constraints represent the processing time of each subtask depends on the resources. Preemption is not allowed, each subtask must be completed without interruption. The problem is to assign each task to the appropriate resources and sort the tasks on the resources to minimize total energy consumption and SLA violation. The challenge is to carefully coordinate and optimize task scheduling and resource allocation to achieve overall cost and time optimization.

Resource scheduling in cloud computing is accomplished in the scheduling interval which preset by cloud service providers. According to the utilization model described above, the energy consumption model of the data center in each scheduling interval is:

$$E_{interval}^{dc} = \sum_i^{t_{sim}} \sum_j^{N_{host}} \theta_{hj} \left( \frac{\sum_i^j N_{vi\_allocMIPS}}{N_{hj\_core} \cdot N_{hj\_MIPS}} \right), \quad (23)$$

where  $E_{interval}^{dc}$  denotes the total energy consumption generated by hosts,  $t_{sim}$  denotes the simulation time in the data center, and  $\theta_{hj}$  denotes the energy consumption corresponding to the current utilization of the host  $j$ . The detail of  $\theta_{hj}$  under different load is illustrated in Table 3.

#### 4.2. SLA violations metric

In the scheduling interval, our scheduling goal is to minimize the SLA violations of each host while optimizing the total energy consumption of the data center. The SLA violation rate of all physical hosts in the data center is defined as:

$$V_{SLA}^{dc} = \sum_i^{t_{sim}} \sum_j^{N_{host}} T_{h_j} \left( \frac{N_{hi\_MIPS\_not\_satisfied}}{N_{hj\_MIPS}} \right), \quad (24)$$

where  $V_{SLA}^{dc}$  denotes the SLA violation rate of data center;  $T_{h_j}$  denotes the SLA time corresponding to the ratio of resource requirement not satisfied;  $N_{hi\_MIPS\_not\_satisfied}$  denotes the value of MIPS that not satisfies the resource requirement.

#### 4.3. Adaptive energy efficiency model

An energy efficiency metric model consists of energy consumption and SLA violations. Previous energy efficiency model studies considered that energy consumption is as important as SLA violations. However, the generation and arrival of IoT requests have moderate fluctuations. The traditional energy efficiency models are no longer suitable in this scenario.

In this subsection, we present an adaptive energy efficiency model to balance the trade-off between energy consumption and SLA violations. Since the data center's optimization goal is to reduce energy consumption and the number of SLA violations, the energy consumption and SLAV are inversely proportional to our optimization goal. In order to simplify the solving process, we make a log transformation of the energy consumption model and the SLAV model. Log transformation is a common way of data transformation, and the purpose is to make the presentation of data close to our desired assumptions. Since the log function is a monotonically increasing function in its domain, taking the logarithm will not change the feature and correlation of the data. In addition, the scale of the variables is compressed, making the data more stable and more convenient for our subsequent calculations.

And the energy efficiency model  $M$  are defined as follows:

$$M = \frac{1}{\alpha \log(E_{interval}^{dc}) + \gamma \log(V_{SLA}^{dc})} \quad (25)$$

The weight coefficients of  $\alpha$  and  $\gamma$  ( $\alpha + \gamma = 1$  and  $0 \leq \alpha, \gamma \leq 1$ ) respectively represent energy consumption and SLA violation rate in the model. In order to better demonstrate our energy efficiency model formula, we define:

$$F = \alpha \log(E_{interval}^{dc}) + \gamma \log(V_{SLA}^{dc}) \quad (26)$$

According to Eqs (25) and (26),  $F$  can be represented as follows:

$$F = \alpha \log \left[ \sum_i^{t_{sim}} \sum_j^{N_{host}} \theta_{h_j} \left( \frac{\sum_i^j N_{vi\_allocMIPS}}{N_{hj\_core} \cdot N_{hj\_MIPS}} \right) + \gamma \log \left[ \sum_i^{t_{sim}} \sum_j^{N_{host}} T_{h_j} \left( \frac{N_{hi\_MIPS\_not\_satisfied}}{N_{hj\_MIPS}} \right) \right] \right] \quad (27)$$

Apparently, the value of weight coefficients ( $\alpha$  and  $\gamma$ ) determines the emphasis of the energy efficiency model. According to the characteristics of  $\alpha$  and  $\gamma$ , we introduce the idea of sigmoid function to determine the value of weight coefficients, which can be expressed as follows:

$$\alpha = \frac{e^{\frac{\mu - \hat{r}}{\sigma}}}{1 + e^{\frac{\mu - \hat{r}}{\sigma}}} \quad (28)$$

$$\gamma = \frac{1}{1 + e^{\frac{\mu - \hat{r}}{\sigma}}} \quad (29)$$

where  $\hat{r}$  denotes the predicted arrival rates of IoT requests given by Poisson arrival model;  $\mu$  denotes the average arrival rates of historical workload;  $\sigma$  denotes the standard deviation of historical arrival rates. The value of  $\mu$  and  $\sigma$  will be updated after the end of each scheduling interval.

Regardless of how the IoT requests fluctuate, the proposed energy efficiency model can adaptively adjust the optimization priority of energy consumption and SLA violations. When the arrival rate of IoT requests  $\hat{r}$  is high, according to the Eq (29), the value of  $\gamma$  will be very close to 1, which means that the optimization of SLA violations is the primary concern of energy efficiency management in the current scheduling interval. Conversely, the energy efficiency model will focus on optimizing the energy consumption of the data center when the value of  $\hat{r}$  is low. If the value of  $\hat{r}$  is equal to the historical average arrival rate  $\mu$ ,  $\alpha = \gamma = 0.5$ .

The adaptive energy efficiency model  $M$  can be formalized as follows:

$$M = \frac{1}{F} \quad (30)$$

According to Eqs (27) and (30),  $F$  can be written in:

$$F = \frac{e^{\frac{\mu - \hat{r}}{\sigma}}}{1 + e^{\frac{\mu - \hat{r}}{\sigma}}} \cdot \log \left[ \sum_i^{t_{sim}} \sum_j^{N_{host}} \theta_{h_j} \left( \frac{\sum_i^j N_{vi\_allocMIPS}}{N_{hj\_core} \cdot N_{hj\_MIPS}} \right) \right] + \frac{1}{1 + e^{\frac{\mu - \hat{r}}{\sigma}}} \cdot \log \left[ \sum_i^{t_{sim}} \sum_j^{N_{host}} T_{h_j} \left( \frac{N_{hi\_MIPS\_not\_satisfied}}{N_{hj\_MIPS}} \right) \right] \quad (31)$$

#### 4.4. VM placement based on arrival rates

In this subsection, we propose a novel VM placement policy to balance the trade-off between energy consumption and SLA. We refer to this policy as VM placement based on arrival rates (VPBAR). The pseudo-code for the VPBAR algorithm is presented in Algorithm 1.

In the algorithm,  $vmList$  denotes the list of VMs;  $hostList$  denotes the set of hosts in the data center;  $\hat{r}$  denotes the predicted arrival rates of IoT requests given by Poisson arrival model;  $\mu$  denotes the average arrival rates of historical workload;  $\sigma$  denotes the standard deviation of historical arrival rates.

In each scheduling interval, the VPBAR algorithm first sorts the  $vmList$  in descending order of the CPU utilization and then calculates the value of weight coefficients  $\alpha$  and  $\gamma$  according to Eqs (28) and (29). VPBAR algorithm iterates the entire  $hostList$  and check whether the host satisfies the resource requirement of the VM. If it meets the requirement, we will calculate the utilization of

this host after an allocation of the VM. VPBAR finds a host with the optimal energy efficiency through VM allocation. The algorithm terminates when the allocation of VMs is complete. From the pseudo-code of the algorithm, we can know that the complexity of the VPBAR is  $O(M \times N)$ , where  $M$  represents the number of hosts and  $N$  represents the number of VMs.

---

**Algorithm 1.** The VPBAR Algorithm.

---

**Input:** vmList, hostList,  $\hat{r}$ ,  $\mu$ ,  $\sigma$

**Output:** allocation map of VMs

```

1: vmList.sortUtilizationByDecreasing()
2: for (vm: vmList) do
3:   allocatedHost = null
4:   minimumEnergyEfficiency=minimum
5:    $\alpha = \frac{e^{\frac{\mu-\hat{r}}{\sigma}}}{1+e^{\frac{\mu-\hat{r}}{\sigma}}}$ 
6:    $\gamma = \frac{1}{1+e^{\frac{\mu-\hat{r}}{\sigma}}}$ 
7:   for (host: hostList) do
8:     if (host satisfies the resource requirement of vm) then
9:       utilizationOfHost = getUtilAfterVM(host,vm)
10:      calculateSLAV()
11:      EnergyEfficiency = calculateEnergyEfficiency()
12:      if (minimumEnergyEfficiency < EnergyEfficiency) then
13:        minimumEnergyEfficiency = EnergyEfficiency
14:        allocatedHost = host
15:      end if
16:      if (allocatedHost  $\neq$  NULL) then
17:        allocate vm to allocatedHost
18:      end if
19:    end if
20:  end for
21: end for

```

---

## 5. Performance analysis

This section evaluates and tests the task arrival model for IoT requests described in Section 3 and VPBAR algorithm introduced in Section 4. It is necessary to evaluate the experiments on a hyper-scale virtualized cloud computing infrastructure. However, it is tough to perform repeatable large-scale experiments on real data centers [15,38,46]. The CloudSim Toolkit [47] is chosen as an experimental platform to evaluate our work. CloudSim is one of the most robust simulation frameworks in cloud computing. It supports on-demand virtualization resource management and simulation of large-scale cloud computing infrastructure.

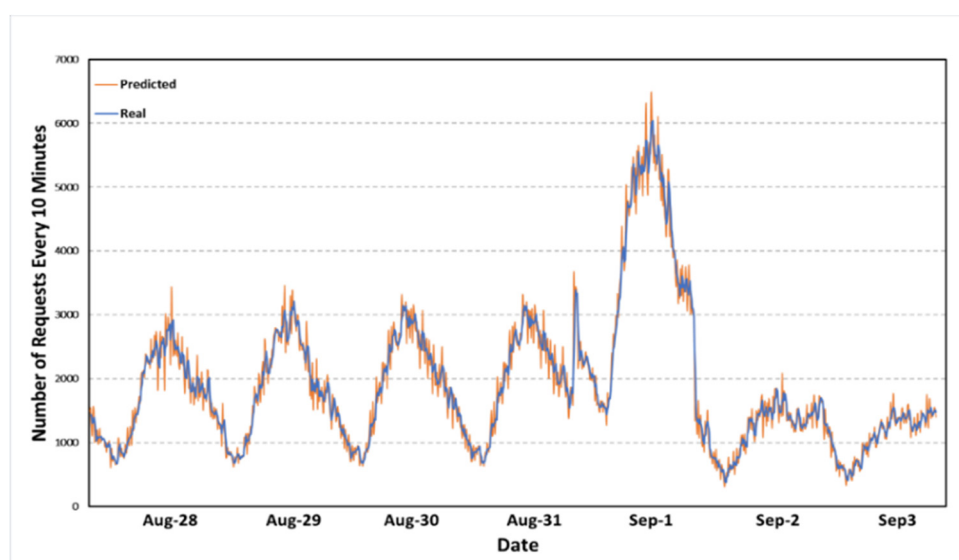
In the experiment, a real-world workload ClarkNet-HTTP [48] is used to evaluate our models and algorithms. The workload contains traces of requests from the cloud service provider for the Baltimore-Washington DC area. During two weeks period of time, the servers received 3,328,587 requests. The

ClarkNet-HTTP workload can simulate a realistic IoT request arrival in the smart building scenario as well as most of the situations that may be faced within the data center. The number of requests in this dataset will fluctuate continuously over time. It is ubiquitous to analyze IoT requests arrival scenario by using this workload.

Each experiment was repeated ten times to ensure the validity and accuracy of the experimental results. Table 3 shows the relationships between the power and CPU utilization of the servers used in our experiment.

**Table 3.** The power of two servers under different CPU utilization.

CPU utilization (%)	Power (W)	
	HP Proliant G4	HP Proliant G5
0	86	93.7
10	89.4	97
20	92.6	101
30	96	105
40	99.4	110
50	102	117
60	105	121
70	108	125
80	112	129
90	115	133
100	117	135

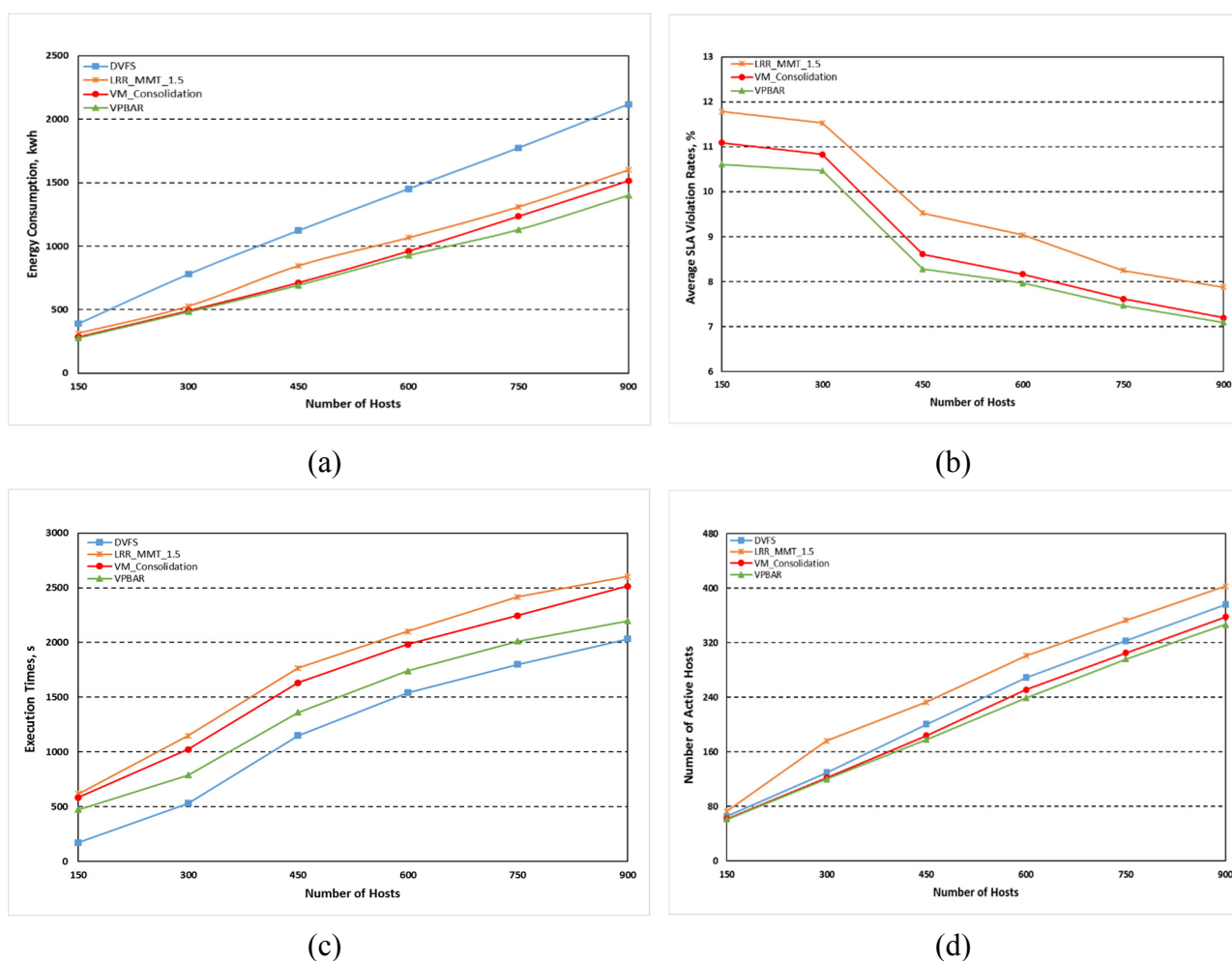


**Figure 3.** The prediction results of IoT requests.

The experiment in this paper is divided into two parts. The first part is to evaluate the effectiveness of the task arrival model, and the second part is to evaluate the performance of VPBAR algorithm. Figure 3 shows the prediction results of IoT requests by the Poisson arrival model with upper and lower bound in ClarkNet-HTTP dataset. The orange line in the figure represents the prediction result

of the arrival model between August 28 and September 3. After adding the upper and lower bounds to the Poisson arrival model, the predicted result is a curve with moderate fluctuations. When the result of the actual value is within the range of the upper and lower bounds, the prediction result will be considered accurate during this scheduling period. Since the IoT requests may fluctuate moderately, simply modelling the arrival of requests will result in a large deviation in task arrival rates. The results in the figure show that our arrival model performs very well on the predictions of IoT requests.

We test the scheduling algorithm with a different number of hosts. The number of hosts is proportional to the number of tasks. We compare the VPBAR algorithm with DVFS [10], LRR\_MMT\_1.5 (Local Regression Robust Minimum migration time) [49] and VM-CONSOLIDATION [38] algorithm. Figure 4(a–d) show the experimental results of VPBAR algorithm in different metrics.



**Figure 4.** Performance analysis on ClarkNet-HTTP dataset: (a) Energy consumption of cloud platform running on different number of PMs. (b) Average SLA violation rates of cloud platform running on different number of PMs. (c) Execution times of the algorithms. (d) Number of Active hosts used in the algorithms.

In Figure 4(a), we focus on the algorithms' results in terms of energy consumption. The experimental results show that VPBAR achieves the lower number of energy consumption, followed

by VM-CONSOLIDATION algorithm. DVFS yield the highest energy consumption because it does not consider the optimization of VMs. The DVFS algorithm mainly adjusts the CPU voltage and frequency according to the current load situation of each PM. This method can reduce part of the energy consumption, but there is no way to deal with those underloaded PMs. These physical machines will consume a lot of power even if they are entirely idle. LRR\_MMT\_1.5 wants to optimize energy consumption by reducing the migration time of VMs. However, due to the volatility of task arrivals, this method cannot effectively consolidate idle PMs to save energy. VPBAR algorithm has the lowest energy consumption. The figure reveals that when the number of hosts created in the data center is relative less, the advantage of VPBAR is not obvious. However, when the number of hosts in the working state is large, VPBAR can significantly reduce the energy consumption of the data center.

Figure 4(b) shows the results of average SLA violations in the data center. Since the DVFS technique does not involve the process of consolidation, the results of DVFS are not shown in this figure. The results show that LRR\_MMT\_1.5 cause a high number of SLA violations, followed by VM-CONSOLIDATION. LRR\_MMT\_1.5 can reduce the cost caused by the migration. However, it will result in many overloaded hosts in the data center, and these overload phenomena will cause serious SLA violations. Those algorithms do not consider the massiveness and time constraint of IoT requests, which causes performance degradation in the data center. VPBAR can model the arrival of cloud tasks. To avoid overloading as much as possible, it can schedule VMs in advance according to the number of tasks reached in a unit time slice.

In Figure 4(c), we compare the algorithms in terms of execution times. DVFS achieve the shortest execution times because it does not need to schedule VMs. Compared with LRR\_MMT\_1.5 and VM-CONSOLIDATION, the execution time of VPBAR algorithm is reduced by about 19% due to the efficiency of the task arrival model. As can be seen from the figure, the execution time of the LRR\_MMT\_1.5 algorithm is not short. The algorithm uses a local regression method that requires constant calculation when making decisions, which makes the algorithm more time-complex. The VM consolidation algorithm only needs to sort the PM lists and VM lists according to utilization, so the algorithm execution time is not too long. In most scenarios, the Poisson task arrival model can give the predictions of arrival rates immediately. It saves much time for resource scheduling.

Figure 4(d) illustrates the number of active hosts of all algorithms. The result shows that many hosts in the idle state are turned off to save energy through dynamic consolidation technique. VPBAR algorithm will gradually show its advantages when the number of requests is at a high level. It also reflects the excellent applicability of VPBAR algorithm for IoT requests.

## 6. Conclusions

In this era of a rapid increase in data volume, energy efficiency scheduling of cloud data centers has become extremely important. Existing scheduling methods do not consider the characteristics of IoT requests, which may lead to performance degradation or SLA violations in a data center. It is of great significance to design an efficient resource scheduling mechanism to handle IoT requests.

In this paper, we propose an improved Poisson arrival model to predict the future arrival rates of IoT requests. We take into account the fluctuations of IoT requests and introduce a novel upper and lower bound mechanism to enhance the applicability of the arrival model. Furthermore, an adaptive energy efficiency model is presented to adjust the priority between energy-saving and SLAs according to future arrival rates. Regardless of how the IoT requests fluctuate, the proposed energy efficiency



model can adaptively adjust the optimization priority of energy consumption and SLA violations. SLA optimization is the primary concern of energy efficiency management when the load is high. Correspondingly, the energy efficiency model will focus on optimizing the energy consumption of the data center when the load is low.

Furthermore, we propose an energy-efficient VM placement algorithm based on task arrival rates (VPBAR) to ensure high availability and energy efficiency. The objectives of our work are to help cloud service providers to provide a better service for their customers. The experimental results show that our arrival model performs very well on the predictions of IoT requests. With the upper and lower bound mechanism, the adaptability of the arrival model becomes stronger. VPBAR algorithm can also achieve excellent results than other solutions. The algorithm shows its advantages when the number of requests is at a high level. Based on the evaluation results, we can conclude that the resource optimization approaches based on task arrival models can achieve a better trade-off between energy saving and SLA optimization. The method is efficient and workable for IoT requests.

Due to the rapid rise of edge computing technology, edge computing nodes also occupy a critical position in the development of smart cities. Our next work goal is to consider the energy-saving problems of edge computing nodes and IoT devices, including considering the battery level and type of devices, data transmission costs, data flow oscillations and other factors.

## Acknowledgments

This work was supported in part by the Guangdong Major Project of Basic and Applied Basic Research under Project 2019B030302002, in part by the Science and Technology Major Project of Guangzhou under number 202007030006, in part by the Industrial Development Fund Project of Guangzhou under Project X2JSD8183470, and in part by the Engineering and Technology Research Center of Guangdong Province for Logistics Supply Chain and Internet of Things under Project GDDST[2016]176.

## Conflict of interest

The authors declare there is no conflict of interest.

## References

1. A. Camero, E. Alba, Smart city and information technology: A review, *Cities*, **93** (2019), 84–94.
2. J. Chin, V. Callaghan, S. B. Allouch, The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective, *J. Ambient Intell. Smart Environ.*, **11** (2019), 45–69.
3. C. J. Martin, J. Evans, A. Karvonen, Smart and sustainable? Five tensions in the visions and practices of the smart-sustainable city in Europe and North America, *Technol. Forecast. Soc. Change*, **133** (2018), 269–278.
4. P. Cardullo, R. Kitchin, Being a ‘citizen’ in the smart city: up and down the scaffold of smart citizen participation in Dublin, Ireland, *GeoJournal*, **84** (2019), 1–13.
5. V. Fernandez-Anez, J. M. Fernández-Güell, R. Giffinger, Smart city implementation and discourses: An integrated conceptual model. The case of Vienna, *Cities*, **78** (2018), 4–16.

6. S. Shashank, A. Gokhale, Dynamic resource management across cloud-edge resources for performance-sensitive applications, in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, IEEE, (2017).
7. A. Beloglazov, R. Buyya, Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints, *IEEE Trans. Parallel Distrib. Syst.*, **24** (2013), 1366–1379.
8. J. Abawajy, M. F. Fudzee, M. M. Hassan, M. Alrubaian, Service level agreement management framework for utility-oriented computing platforms, *J. Supercomput.*, **71** (2015), 4287–4303.
9. H. Haarstad, M. W. Wathne, Are smart city projects catalyzing urban energy sustainability?, *Energy Policy*, **129** (2019), 918–925.
10. A. Toor, S. Islam, N. Sohail, A. Akhunzada, J. Boudjadar, H. A. Khattak, et al., Energy and performance aware fog computing: A case of DVFS and green renewable energy, *Future Gener. Comput. Syst.*, **101** (2019), 1112–1121.
11. N. Kulkarni, S. V. Lalitha, S. A. Deokar, Real time control and monitoring of grid power systems using cloud computing, *Int. J. Electr. Comput. Eng.*, **9** (2019).
12. X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, et al., Energy-aware virtual machine allocation for cloud with resource reservation, *J. Syst. Software*, **147** (2019), 147–161.
13. M. Aldossary, K. Djemame, I. Alzamil, A. Kostopoulos, A. Dimakis, E. Agiatzidou, Energy-aware cost prediction and pricing of virtual machines in cloud computing environments, *Future Gener. Comput. Syst.*, **93** (2019), 442–459.
14. S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, N. Kato, A survey on network methodologies for real-time analytics of massive IoT data and open research issues, *IEEE Commun. Surv. Tutorials*, **19** (2017), 1457–1477.
15. Z. Zhou, J. Abawajy, M. Chowdhury, Z. Hu, K. Li, H Cheng, et al., Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms, *Future Gener. Comput. Syst.*, **86** (2018), 836–850.
16. K. Chang, S. Park, H. Kong, W. Kim, Optimizing energy consumption for a performance-aware cloud data center in the public sector, *Sustainable Comput. Inf. Syst.*, **20** (2018), 34–45.
17. M. H. Malekloo, N. Kara, M. Barachi, An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments, *Sustainable Comput. Inf. Syst.*, **17** (2018), 9–24.
18. M. K. Gupta, A. Jain, T. Amgoth, Power and resource-aware virtual machine placement for IaaS cloud, *Sustainable Comput. Inf. Syst.*, **19** (2018), 52–60.
19. Z. Tang, L. Qi, Z. Cheng, K. Li, S. Khan, K. Li, An energy-efficient task scheduling algorithm in DVFS-enabled cloud environment, *J. Grid Comput.*, **14** (2016), 55–74.
20. M. B. Gawali, S. K. Shinde, Task scheduling and resource allocation in cloud computing using a heuristic approach, *J. Cloud Comput.*, **7** (2018), 4.
21. Z. Zhu, G. Zhang, M. Li, X. Liu, Evolutionary multi-objective workflow scheduling in cloud, *IEEE Trans. Parallel Distrib. Syst.*, **27** (2016), 1344–1357.
22. Y. Peng, D. K. Kang, F. Hazemi, C. H. Youn, Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters, *Opt. Switching Networking*, **23** (2017), 225–240.
23. M. Masdari, F. Salehi, M. Jalali, M. Bidaki, A survey of PSO-based scheduling algorithms in cloud computing, *J. Network Syst. Manage.*, **25** (2017), 122–158.

24. L. Teylo, U. Paula, Y. Frota, D. Oliveira, L. M. A. Drummond, A hybrid evolutionary algorithm for task scheduling and data assignment of data-intensive scientific workflows on clouds, *Future Gener. Comput. Syst.*, **76** (2017), 1–17.
25. Y. Sharma, W. Si, D. Sun, B. Javadi, Failure-aware energy-efficient VM consolidation in cloud computing systems, *Future Gener. Comput. Syst.*, **94** (2019), 620–633.
26. Q. Zhang, H. Chen, Y. Shen, S. Ma, H. Lu, Optimization of virtual resource management for cloud applications to cope with traffic burst, *Future Gener. Comput. Syst.*, **58** (2016), 42–55.
27. F. J. Baldan, S. Ramirez-Gallego, C. Bergmeir, F. Herrera, J. M. Benitez, A forecasting methodology for workload forecasting in cloud systems, *IEEE Trans. Cloud Comput.*, **6** (2018), 929–941.
28. F. H. Tseng, X. Wang, L. Chou, H. Chao, V. C. M. Leung, Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm, *IEEE Syst. J.*, **12** (2018), 1688–1699.
29. M. Aloqaily, A. Boukerche, O. Bouachir, F. Khalid, S. Jangsher, An energy trade framework using smart contracts: Overview and challenges, *IEEE Network*, **34** (2020), 119–125.
30. J. Krzywda, A. Eldin, T. E. Carlson, P. O. Östberg, E. Elmroth, Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling, *Future Gener. Comput. Syst.*, **81** (2018), 114–128.
31. R. B. Uriarte, R. D. Nicola, V. Scoca, F. Tiezzi, Defining and guaranteeing dynamic service levels in clouds, *Future Gener. Comput. Syst.*, **99** (2019), 27–40.
32. M. Soltanshahi, R. Asemi, N. Shafiei, Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers, *Heliyon*, **5** (2019), e02066.
33. G. Kazdaridis, S. Keranidis, P. Symeonidis, P. S. Dias, P. Gonçalves, B. Loureiro, et al., Everun: Enabling power consumption monitoring in underwater networking platforms, in *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*, (2017), 83–90.
34. M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, A. K. Sangaiah, A hybrid model of internet of things and cloud computing to manage big data in health services applications, *Future Gener. Comput. Syst.*, **86** (2018), 1383–1394.
35. A. A. Khan, M. Zakarya, R. Buyya, R. Khan, M. Khan, O. Rana, An energy and performance aware consolidation technique for containerized datacenters, *IEEE Trans. Cloud Comput.*, (2019), forthcoming.
36. A. A. Beegom, M. S. Rajasree, Integer-pso: a discrete pso algorithm for task scheduling in cloud computing systems, *Evol. Intell.*, **12** (2019), 227–239.
37. V. Balasubramanian, M. Aloqaily, M. Reisslein, An SDN architecture for time sensitive industrial IoT, *Comput. Networks*, **186** (2021), 107739.
38. X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, S. Yu, A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems, *Future Gener. Comput. Syst.*, **105** (2020), 789–799.
39. A. Hussain, J. Chun, M. Khan, A novel framework towards viable cloud service selection as a service (cssaas) under a fuzzy environment, *Future Gener. Comput. Syst.*, **104** (2020), 74–91.
40. F. De la Prieta, S. Rodríguez, J. M. Corchado, J. Bajo, Infrastructure to simulate intelligent agents in cloud environments, *J. Intell. Fuzzy Syst.*, **28** (2015), 29–41.

41. F. De la Prieta, S. Rodríguez-González, P. Chamoso, J. M. Corchado, J. Bajo, Survey of agent-based cloud computing applications, *Future Gener. Comput. Syst.*, **100** (2019), 223–236.
42. F. De la Prieta, S. Rodríguez, J. Bajo, J. M. Corchado, A multiagent system for resource distribution into a Cloud Computing environment, in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, Berlin, Heidelberg, (2013), 37–48.
43. L. Tseng, Y. Wu, H. Pan, M. Aloqaily, A. Boukerche, Reliable broadcast with trusted nodes: Energy reduction, resilience, and speed, *Comput. Networks*, **182** (2020), 107486.
44. F. Ali, O. Bouachir, O. Ozkasap, M. Aloqaily, SynergyChain: Blockchain-assisted Adaptive Cyberphysical P2P Energy Trading, *IEEE Trans. Ind. Inf.*, (2020), forthcoming.
45. F. S. Ali, M. Aloqaily, O. Alfandi, O. Ozkasap, Cyberphysical blockchain-enabled peer-to-peer energy trading, *Computer*, **53** (2020), 56–65.
46. S. Ismaeel, R. Karim, A. Miri, Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres, *J. Cloud Comput.*, **7** (2018), 10.
47. E. Barbierato, M. Gribaudo, M. Iacono, A. Jakóbi, Exploiting CloudSim in a multiformalism modeling approach for cloud based systems, *Simul. Modell. Pract. Theory*, **93** (2019), 133–147.
48. P. Singh, P. Gupta, K. Jyoti, Tasm: technocrat arima and svr model for workload prediction of web applications in cloud, *Cluster Comput.*, **22** (2019), 619–633.
49. M. Mohammadhosseini, A. T. Haghghat, E. Mahdipour, An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm, *J. Supercomput.*, **75** (2019), 6904–6933.



AIMS Press

©2021 the Authors, licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)