Energy Efficient Resource Scheduling in Cloud Computing Based on Task Arrival Model

Bin Wang[1], Yongheng Liu[1], Fan Zhang[1], Jun Jiang[2*]
[1]Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China
[2]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
Email: wangb02@pcl.ac.cn, liuyh01@pcl.ac.cn, zhangf@pcl.ac.cn, junjiangscut@gmail.com

2022 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-5975-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCWkshps56602.2022.10008745

***Abstract*—High energy consumption has become a vital bottleneck restricting the development of cloud computing. Most of the current resource management frameworks focus on the scheduling module but fail to consider the burstiness of workloads adequately. In this paper, we present a resource management framework based on the arrival model switching mechanism to optimize the energy efficiency of cloud data centers. We analyze cloud task characteristics and propose two types of task arrival models. The arrival model based on the Poisson process is for common scenarios, and the grey model in traffic burst (GMTB) is for bursty scenarios. An anomaly detection module is introduced to detect the abnormal events and determine whether the model needs to be switched. Finally, we propose an integrated virtual machine scheduling policy to balance the energy consumption and service level agreement.**

***Index Terms*—Energy efficiency, task arrival, anomaly detection, resource scheduling.**

I. Introduction

As a distributed computing paradigm, cloud computing has the characteristics of the economy, elastic scaling, and high scalability, so it has quickly entered thousands of households [1]. Cloud computing technology has the characteristics of efficient configuration, fast release, and automatic arrangement in terms of operation and maintenance management. This technology allows users to access the shared pool of computing and storage resources efficiently, conveniently, and on-demand [2]. Due to the surge in the number and scale of cloud computing data centers around the world, the growth rate of energy consumption is also becoming increasingly alarming. From an economic perspective, it accounts for a large percentage of operational expenditure for cloud infrastructure [3]. To avoid the energy consumption of data centers becoming a bottleneck restricting the development of cloud computing, the major cloud service providers [4] are promoting the research on energy efficiency management of data centers.

Furthermore, a critical question for cloud providers is how to reduce the energy consumption of data centers while ensuring the quality of service (QoS) [5]. QoS requirements can be formalized in the service level agreements (SLAs) [6]. SLA is one of the critical factors to user experience and satisfaction. Hence, designing a practical energy efficiency approach in data centers requires finding a trade-off between energy consumption and SLA guarantee.

Many studies have proposed several resource management strategies to optimize the energy efficiency of cloud computing systems. Malekloo et al. [7] presented an QoS-aware Virtual Machine(VM) placement strategy to optimize the energy efficiency and other key aspects of appliances. Aldossary et al. [8] introduced a cloud system architecture to support energy-awareness and cost estimation of Cloud infrastructure services.However, the unstable fluctuations of task arrival has deemed the above frameworks not to explicitly handle the traffic bursts, which makes them ineffective in this scenario [9].

How to design an efficient VM scheduling algorithm is another significant concern for energy saving. A lot of researches [10] combined cognitive science with resource management in cloud computing. The method based on blockchain technology [11], [12] uses decentralization to reduce the load of the cloud system, and the decentralized management strategy can improve the scheduling process. Furthermore, previous resource management strategies did not take into account the classes of workloads running in current cloud computing environments [13], which would make the energy efficiency optimization insufficient under different workload scenarios.

In this paper, we focus on analyzing and modelling the different types of workload in the data center. We take into account the burstiness of workload and divide the cloud tasks into the conventional and bursty type. An arrival model switching mechanism is proposed to detect the abnormal events occurring in the task series and select an appropriate arrival model for workload prediction. We design an energy-efficient VM scheduling algorithm to treat different types of workload separately.

The remainder of the paper is organized as follows. Section II discusses the system architecture of our task arrival model. VM scheduler is discussed in Section III. Experiments and results are given in Section IV. Section V concludes the paper and discusses future directions.

II. SYSTEM ARCHITECTURE

Different from the traditional task arrival assumption in cloud computing centers, the arrival of workloads have bursti-

[*] Corresponding author.

This work was supported in part by the Major Key Project of PCL under Grant PCL2021A09, in part by the Guangdong Major Project of Basic and Applied Basic Research under Project 2019B030302002.

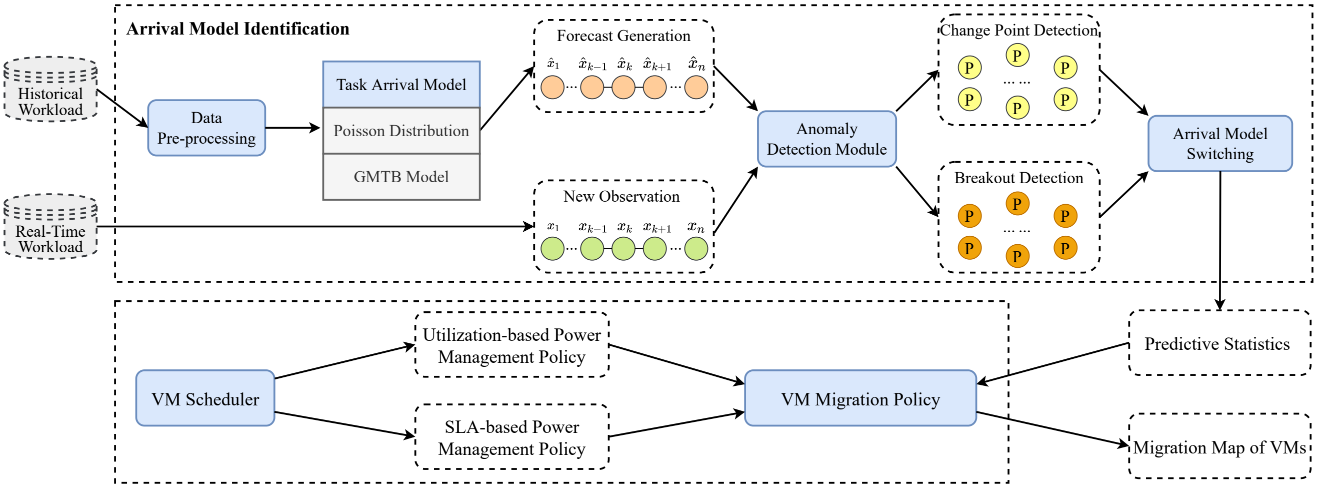

Fig. 1. A novel resource scheduling framework based on task arrival.

ness [14]. Conventional task arrival model based on the Poisson process is no longer suitable for bursty workloads [15]. Therefore, in this paper, we introduce a resource management framework based on task arrival model identification mechanism, as shown in Fig.(1). Our framework mainly consists of: the task arrival model, the anomaly detection module, the arrival model switching module and VM scheduler.

The task arrival model keeps track of historical workloads, establishes cloud task arrival model and makes predictions of future workload. The anomaly detection module is used to identify abnormal events occurring in the new observation task arrival series. The module is mainly divided into two parts: change point detection and breakout detection.

The arrival model switching module is responsible for switching the cloud task arrival model based on the anomaly in the real-time workload series. The VM Scheduler manages the virtual resources in every scheduling interval according to predictive statistics. The task arrival model gives a prediction of future workload. Two different power management policies (Utilization-based Power Management Policy and SLA-based Power Management Policy) are proposed to handle different predicted arrival rates.

A. *TASK ARRIVAL MODEL*

1) *TASK ARRIVAL BASED ON A POISSON PROCESS:*

Generally spealing, the resources of the cloud environment are limited. Task scheduling of cloud computing can be stated as follows.

Definition 1. Let $t = 0$ be the time when a scheduling period starts in the data center. The number of cloud task requests arriving at the data center in the x time period is denoted as $N_{\text{task}}(x)$ and $P_{dc}(n(t))$is defined as the probability that n cloud tasks arrive in data center at time t. First, there is a hypothesis 1. The probability of cloud tasks arriving in each very small time slice is independent of each other, namely:

$$\begin{aligned} & P_{dc}\left(N_{\text{task}}\left(t_1\right)=n_1, N_{\text{task}}\left(t_1+t_2\right)=n_2\right) \\ & =P_{dc}\left(N_{\text{task}}\left(t_1\right)=n_1\right) \cdot P_{dc}\left(N_{\text{task}}\left(t_2\right)=n_2\right). \end{aligned} \tag{1}$$

It is assumed that the probability of the request arrival is $\delta \Delta t$. Then the probability that k tasks arrive in this time interval is:

$$\psi\left(\rho ; n, \frac{\delta t}{n}\right)=\left(\begin{array}{l} n \\ \rho \end{array}\right)\left(\frac{\delta t}{n}\right)^\rho\left(1-\frac{\delta t}{n}\right)^{n-\rho}. \tag{2}$$

$$\psi\left(\rho ; n, \tfrac{\delta t}{n}\right)=\tfrac{n(n-1) \cdots(n-\rho+1)}{\rho ! n^\rho} \cdot \delta t^\rho \cdot\left(1-\tfrac{\delta t}{n}\right)^{n-\rho}. \tag{3}$$

$$\begin{aligned} \psi\left(\rho ; n, \frac{\delta t}{n}\right)= & \frac{n(n-1) \cdots(n-\rho+1)}{n^\rho} \\ & \cdot \frac{(\delta t)^\rho}{\rho !} \cdot\left(1-\frac{\delta t}{n}\right)^{-\rho} \cdot\left(1-\frac{\delta t}{n}\right)^n. \end{aligned} \tag{4}$$

Consequently, take the limit of n for Eq.(4), the formula is transformed into:

$$\begin{aligned} & \lim_{n \rightarrow \infty} \psi\left(\rho ; n, \tfrac{\delta t}{n}\right) \\ & =\lim_{n \rightarrow \infty}\left[\tfrac{n(n-1) \cdots(n-\rho+1)}{n^\rho} \cdot \tfrac{(\delta t)^\rho}{\rho !} \cdot\left(1-\tfrac{\delta t}{n}\right)^{-\rho}\left(1-\tfrac{\delta t}{n}\right)^n\right] \\ & =\tfrac{e^{-\delta t}(\delta t)^\rho}{\rho !}. \end{aligned} \tag{5}$$

Substituting a single parameter α for δt, according to $p=\frac{\delta t}{n}$, so $\alpha=n p$, the task arrival probability model can be expressed as:

$$f(\rho ; \alpha)=e^{-\alpha} \frac{\alpha^\rho}{\rho !}. \tag{6}$$

For any practicable δt, the probability distribution of the task arrival model can be formulated as :

$$P_n^{\text{cloud}}(t)=e^{-\delta t} \frac{(\delta t)^n}{n !}, \forall n \in N. \tag{7}$$

2) *TASK ARRIVAL WITH SUDDEN GROWTH OR BREAKDOWN EVENTS:*

In most cloud task arrival scenarios, the Poisson process can give a good predictions at a very low cost. However, the network traffic in the data center is known for its burstiness. When the traffic bursts instantaneously, it can easily lead to congestion at the receiving end. Once congestion occurs, the network latency and throughput of the traffic are affected, resulting in longer response latency and a worse user experience.

Autoregressive models are another common model for analyzing time series. but it is obviously not suitable for our workload scenarios. Workload prediction based on deep neural networks is one of the mainstream load prediction methods [16], [17], but there are problems, such as many network parameters and long model training times. The grey model predicts the workload growth trend of VMs and compares it with the previous period to determine the VM migration plan. This method can achieve an ideal prediction result when the total amount of data is limited. So we use the extension of the Grey forecasting model, the GMTB (Grey Model for Traffic Burst) model to make predictions. The GMTB model is suitable for workloads that are stable and sharply rising.

GM(1,1) [18] is an effective grey forecasting model that is most widely used to deal with uncertain and insufficient information. The most common and significant procedure in the grey forecasting is the accumulated generating operation (AGO), which is used to change the data variations and find regular patterns. The detailed definitions and steps of GMTB can be expressed as follows:

Definition 2. Let $X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n)\right)$ denotes the series of jobs arrival inside a data center, where $\left\{x^{(0)}(k) \mid \left\{x^{(0)}(k) \geq 0, k = 1, 2, \cdots, n\right)\right\}$ represents the kth observed value. By applying one time accumulated generating operation (AGO), the increasing series $X^{(1)}$ can be obtained as follows:

$$X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n)\right), \tag{8}$$

where accumulated generating operator $x^{(1)}(k)$ can be defined as follows:

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), k = 1, 2, \cdots, n. \tag{9}$$

Definition 3. Let $X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n)\right)$ and $X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n)\right)$, we define $z^{(1)}(k)$ as a background value of $x^{(1)}(k)$, then:

$$z^{(1)}(k) = \frac{1}{2}\left(x^{(1)}(k) + x^{(1)}(k-1)\right), k = 2, 3, \cdots, n. \tag{10}$$

And the first order grey differential equation of GMTB is obtained as follows:

$$x^{(0)}(k) + az^{(1)}(k) = b. \tag{11}$$

Definition 4. Let $x^{(0)}(k)$ denotes the kth observed value, and $z^{(1)}(k)$ denotes the background value of $x^{(1)}(k)$, then the whitenization equation is:

$$\frac{dx^{(1)}}{dt} + ax^{(1)}(k) = b. \tag{12}$$

Because of $x^{(1)}(0) = x^{(0)}(1)$, we can obtain the time responded function of GMTB model, as shown in:

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(1) - \frac{\widehat{b}}{\hat{a}}\right) e^{-\hat{a}(k-1)} + \frac{\widehat{b}}{\hat{a}}, (k = 1, \cdots, n). \tag{13}$$

According to the inverse accumulated generating operator (IAGO), the forecasting equation of GMTB can be written as follows:

$$\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \widehat{x}^{(1)}(k-1), (k = 2, 3, \cdots, n), \tag{14}$$

where $x^{(0)}(k)$ is the predicted value of the kth observations.

B. ANOMALY DETECTION MODULE

1) CHANGE POINT DETECTION:

The change point refers to a point at which observations or data follow two vastly different models. In statistical analysis, change point detection analyzes a series of time ordered data in order to identify times when a property of the series changes. The detection problem concerns detecting whether or not a change has occurred and determining the time of each change. It provides confidence levels and intervals for any such changes.

Definition 5. Let $Z_1^N = \{Z_1, Z_2, \cdots, Z_N\}$ denote the series of task arrival data inside a data center and let S_1^N denotes the statistical characteristics of the series from 1 to N. The change point is denoted by τ when the following condition is satisfied.

$$\exists \tau \in (1, N), S_{\tau+1}^N \notin S_1^\tau. \tag{15}$$

2) BREAKOUT DETECTION:

Breakout events in a time series can reveal abnormal activities that have occurred in the past, as well as other anomalous behaviors that may occur in the near future. In cloud data center, breakout have ramifications on the availability and performance of a service. Breakout typically have two features: a mean shift and ramp up. The mean shift represents a sudden jump that occurs in the time series. A sudden jump in CPU utilization from 45% to 70% would illustrate a mean shift. The ramp up represents a gradual increase in the value of metric from one steady state to another. A gradual increase in CPU utilization from 45% to 70% would illustrate a ramp up.

Because breakout events are often accompanied by a mean shift, we apply mean shift detection, due to its simplicity and effectiveness, to identify the time of each breakout in the task arrival series. The objective of mean shift detection is to find the sudden jump in the series. In the first phase, we need to calculate the mean shift value for each point. Let w denotes the detection window size and Z_t denotes the observation value at time t, the mean shift value of each point $gamma(t)$ is described as follows:

$$\gamma(t) = \frac{Z_t + Z_{t-1} + \cdots + Z_{t-w+1}}{Z_{t-w} + Z_{t-w-1} + \cdots + Z_{t-2w+1}}, t \in [2w, N]. \tag{16}$$

We can use this model to calculate the power consumption of each active PM. Constants provide reference power when the system is idle. Our model in the above is more appropriate than our previous models. In addition, it improves the average prediction accuracy of almost all indicators. We limit the model to linearity to limit the performance loss caused by the increasing complexity of the model equation.

3) ARRIVAL MODEL SWITCHING:

It has become common truth that the generation and arrival of workloads have moderate burstiness. Task arrival model based on poisson process is not applicable in some cases. An arrival model switching mechanism is proposed to select the appropriate task arrival model for cloud data centers.

Once an abnormal event occurs, the arrival model based on poisson process needs to be switched to the GMTB model or the poisson model needs to be remodelled. The policy for model switching finds a predictions as shown in Eq.(17):

$$\hat{y}_t = \begin{cases} e^{-\delta t}\frac{(\delta t)^n}{n!},\ b \le T \ \ and\ c > T \text{ for N times;} \\ \text{GMTB}(x^{(0)}(b)),\ b > T \text{ for N times;} \\ \text{Reserved},\ \ \text{otherwise.} \end{cases} \tag{17}$$

Where b is the number of detected breakout events; c is the number of change points; T is the threshold; $e^{-\delta t}\frac{(\delta t)^n}{n!}$ is the task arrival model based on poisson process. Once the number of breakouts exceed T for N times, the poisson distribution model is switched to the GMTB model, which implies a traffic burst event may occurs. The task arrival model will be reserved if both the number of change points and breakouts does not meets the conditions.

III. VM SCHEDULER

A. UTILIZATION-BASED POWER MANAGEMENT POLICY

We propose a new VMs selection method named UBPM (Utilization-Based Power Management) to select VMs for migration when a PM is overloaded in flat task arrival scenarios. The UBPM policy selects a list of VMs for migration from overloaded PMs that satisfies the following condition:

$$\frac{C^v_{vm}}{M^v_{vm}} > \frac{C^e_{vm}}{M^e_{vm}}, v \in S_{vm}, \forall e \in S_{vm} \text{ and } v \neq e \tag{18}$$

Where S_{vm} is the group of VMs of overloaded hosts. n this scenario, the energy-based power management policy will be applied when the arrival of the requests is flat. At the same time, we must ensure the overloaded PMs have been eliminated. UBPM policy means that there will be more PMs running near the overloaded threshold utilization through eliminating the overloaded PMs. In this scenario, the proportion of SLA violations is relatively low. The policy enables more PMs to run at a suitable high utilization rate and reduces the number of idle PMs.

B. SLA-BASED POWER MANAGEMENT POLICY

When a PM is overloaded in traffic bursts scenarios, we propose a VMs selection method named SBPM (SLA-Based Power Management Policy) to reduce the SLA violations. The SBPM policy selects a list of VMs for migration from overloaded PMs that meets the following condition:

$$C^v_{vm} \times M^v_{vm} < C^e_{vm} \times M^e_{vm}, v \in S_{vm}, \forall e \in S_{vm}, v \neq e^e \tag{19}$$

The SLA-based power management policy will be applied when the PM is overloaded. In this case, the SLA violation rates in the data center will rise sharply. The core method of the policy is minimizing the length of migration lists. Moreover, it will exacerbate the SLA violations caused by traffic bursts. By choosing the minimum number of VMs that can eliminate the overloaded PMs, we can reduce the risk of unnecessary migration in the scenarios of traffic bursts.

IV. EXPERIMENTS AND ANALYSIS

This section uses two real workloads in the experiments to evaluate the proposed load prediction model and scheduling algorithm. The workloads request data collected from the actual environment. These workloads can simulate realistic mission arrival scenarios and most situations a data center might encounter.

The first workload contained requests from Internet providers. The dataset has a specific seasonal trend and has appropriate differences between seasons. The second workload [19] originates from web traffic request data from the 1998 World Cup web server. The number of tasks consistently grows sharply on game day. This workload dataset can be used as request input in traffic burst scenarios [20].

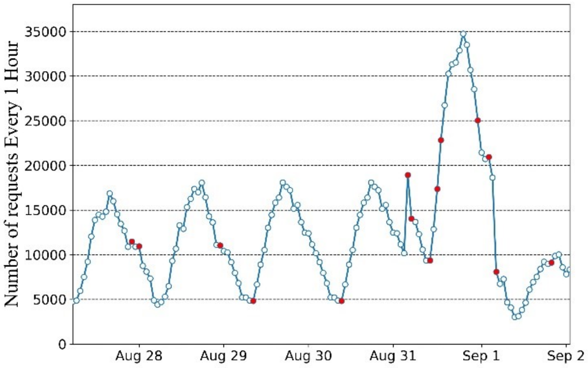

Fig. 2. The change points from August 28th to September 2nd.

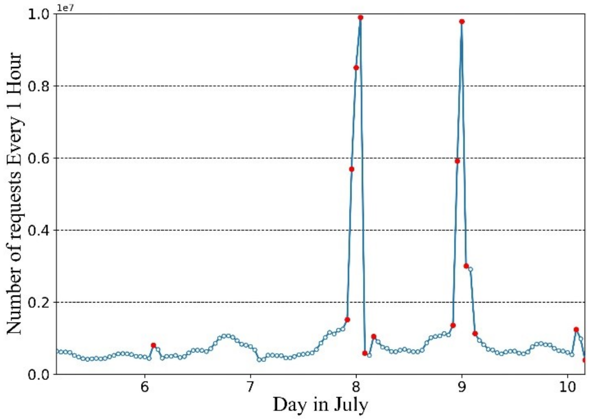

Fig. 3. The breakout points from July 6th to July 10th.

Fig.(2) shows the time series that contains multiple change points. The complete dataset illustrates two weeks' worth of all HTTP requests to the ClarkNet server. It is a real cloud task arrival scenario which we aim to tackle. The target of change point detection is to identify these state boundaries by discovering the change points.

Fig.(3) shows the time series that contains multiple breakout period. The number of cloud tasks always grows sharply on a game day. The objective of breakout detection is to determine the traffic bursts that occur in the cloud task series.

The performance of VM scheduling policy proposed in this paper (VMTA) is compared with the following algorithms.

- DthMf [3] is the Dynamic Threshold Maximum Fit strategy utilizing VM consolidation technique.
- MST [21] is the maximum sustainable throughput method.
- MFF [22] is the max flow forecast (MFF) model to forecast the trends of the series.

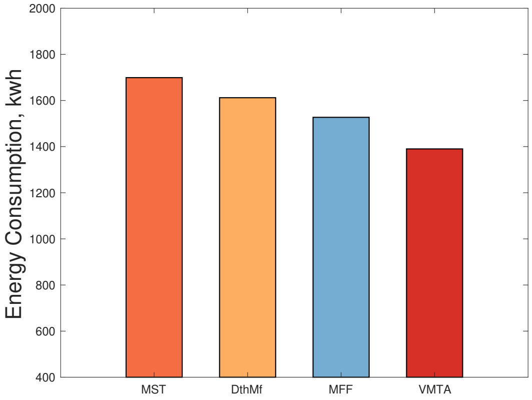

Fig. 4. The comparison of energy consumption.

Fig.(4-6) show the experimental results of each algorithm on the traffic burst data set (98 World Cup). In Fig.(4), we focus on the experimental results of each algorithm on the energy consumption index. The experimental results in the figure show that the total energy consumption of the data center using the VMTA scheduling algorithm is lower in this scenario, followed by the MFF algorithm. Since DthMf utilizes dynamic VM integration technology to achieve better energy saving, DthMf consumes less energy than MST.

Fig.(5) shows the experimental comparison results of the average SLA violation rate in the data center. The results show that DthMf leads to many SLA violations, followed by MFF. This is because these algorithms do not take into account the bursty nature of workloads and time constraints, resulting in degraded data center performance. Because the DthMf algorithm uses the dynamic virtual machine consolidation technology, a large number of hosts are shut down, which makes the SLA violation rate generated by the scheduling algorithm greatly increase in the case of traffic bursts. Since it can more accurately predict traffic bursts, the VMTA algorithm can better adjust and plan data center resources in advance, and this algorithm can achieve the best results in SLA optimization. The experimental results show that the VMTA scheduling algorithm proposed in this chapter can achieve a better trade-off between energy consumption and SLA in the traffic burst scenario.

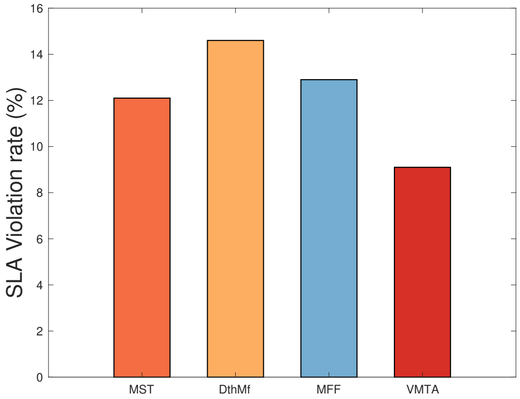

Fig. 5. The comparison of average SLA violation rates of cloud platform.

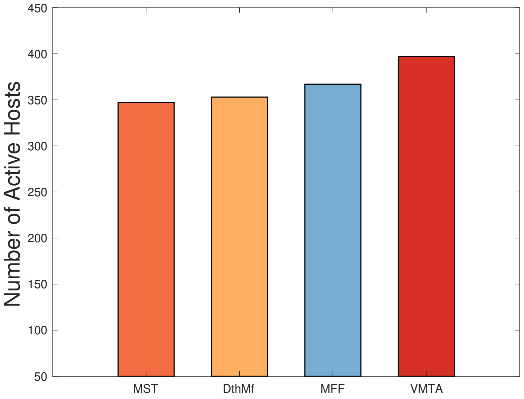

Fig. 6. The comparison of Number of Active hosts.

Fig.(6) shows the number of active physical machines in the data center. The results show that many hosts in an idle state can be shut down by the dynamic consolidation technique (DthMf) to save energy. In a traffic burst scenario, the VMTA algorithm will gradually show its advantages. It also reflects the excellent applicability.

V. Conclusion

In this paper, we have proposed a framework of resource management based on task arrival model switching for energy-efficient Cloud computing. We present a robust task arrival switching mechanism to resolve the prediction of the workloads and anomaly events. The task arrival model based on Poisson process and GMTB model can give a reliable prediction in most of cloud task arrival scenarios.

A VM migration policy based on the utilization and SLA power management policy is proposed to optimize high energy efficiency. We plan to verify the effectiveness and performance of the load forecasting and energy-efficient scheduling algorithms in a distributed cluster in a real environment.

REFERENCES

[1] Zhang, Y., Lan, X., Ren, J. and Cai, L., 2020. Efficient computing resource sharing for mobile edge-cloud computing networks. IEEE/ACM Transactions on Networking, 28(3), pp.1227-1240.

[2] Zhang, X., Wu, C., Li, Z. and Lau, F.C., 2018. A Truthful $(1-\epsilon)$-Optimal Mechanism for On-Demand Cloud Resource Provisioning. IEEE Transactions on Cloud Computing, 8(3), pp.735-748.

[3] Wang, B., Liu, F. and Lin, W., 2021. Energy-efficient VM scheduling based on deep reinforcement learning. Future Generation Computer Systems, 125, pp.616-628.

[4] Wang, B., Liu, F., Lin, W., Ma, Z. and Xu, D., 2021. Energy-efficient collaborative optimization for VM scheduling in cloud computing. Computer Networks, 201, p.108565.

[5] Hayyolalam, V. and Kazem, A.A.P., 2018. A systematic literature review on QoS-aware service composition and selection in cloud environment. Journal of Network and Computer Applications, 110, pp.52-74.

[6] Kumar, A. and Bawa, S., 2020. A comparative review of meta-heuristic approaches to optimize the SLA violation costs for dynamic execution of cloud services. Soft Computing, 24(6), pp.3909-3922.

[7] Malekloo, M.H., Kara, N. and El Barachi, M., 2018. An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. Sustainable Computing: Informatics and Systems, 17, pp.9-24.

[8] Aldossary, M., Djemame, K., Alzamil, I., Kostopoulos, A., Dimakis, A. and Agiatzidou, E., 2019. Energy-aware cost prediction and pricing of virtual machines in cloud computing environments. Future Generation Computer Systems, 93, pp.442-459.

[9] Wang, B. and Liu, F., 2021. Task arrival based energy efficient optimization in smart-IoT data center. Mathematical Biosciences and Engineering, 18(3), pp.2713-2732.

[10] Deebak, B.D., Memon, F.H., Khowaja, S.A., Dev, K., Wang, W. and Qureshi, N.M.F., 2022. In the Digital Age of 5G Networks: Seamless Privacy-Preserving Authentication for Cognitive-Inspired Internet of Medical Things. IEEE Transactions on Industrial Informatics.

[11] Wang, W., Xu, H., Alazab, M., Gadekallu, T.R., Han, Z. and Su, C., 2021. Blockchain-based reliable and efficient certificateless signature for IIoT devices. IEEE transactions on industrial informatics.

[12] Deebak, B.D., Memon, F.H., Khowaja, S.A., Dev, K., Wang, W., Qureshi, N.M.F. and Su, C., 2022. Lightweight blockchain based remote mutual authentication for AI-empowered IoT sustainable computing systems. IEEE Internet of Things Journal.

[13] Shen, H. and Chen, L., 2022. A Resource-Efficient Predictive Resource Provisioning System in Cloud Systems. IEEE Transactions on Parallel and Distributed Systems.

[14] Jiang, J., Liu, F., Ng, W.W., Tang, Q., Wang, W. and Pham, Q.V., 2022. Dynamic incremental ensemble fuzzy classifier for data streams in green internet of things. IEEE Transactions on Green Communications and Networking.

[15] Atmaca, T., Begin, T., Brandwajn, A. and Castel-Taleb, H., 2015. Performance evaluation of cloud computing centers with general arrivals and service. IEEE Transactions on parallel and distributed systems, 27(8), pp.2341-2348.

[16] Zhang, Y., Liu, F., Wang, B., Lin, W., Zhong, G., Xu, M. and Li, K., 2022. A multi-output prediction model for physical machine resource usage in cloud data centers. Future Generation Computer Systems, 130, pp.292-306.

[17] Ren, Y., Jiang, H., Ji, N. and Yu, H., 2022. TBSM: A traffic burst-sensitive model for short-term prediction under special events. Knowledge-Based Systems, 240, p.108120.

[18] Xiao, X. and Duan, H., 2020. A new grey model for traffic flow mechanics. Engineering Applications of Artificial Intelligence, 88, p.103350.

[19] Mallikharjuna Rao, K. and Rama Satish, A., 2022. A Comprehensive Study on Workloads in Cloud Computing. In Advanced Computing and Intelligent Technologies (pp. 505-514). Springer, Singapore.

[20] Wang, B., Liu, F., Lin, W., Ma, Z. and Xu, D., 2021. Energy-efficient collaborative optimization for VM scheduling in cloud computing. Computer Networks, 201, p.108565.

[21] Arkian, H., Pierre, G., Tordsson, J. and Elmroth, E., 2021, July. Model-based stream processing auto-scaling in geo-distributed environments. In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-10). IEEE.

[22] Liao, D., Sun, G., Yang, G. and Chang, V., 2018. Energy-efficient virtual content distribution network provisioning in cloud-based data centers. Future Generation Computer Systems, 83, pp.347-357.