A Game Theory-based Virtual Machine Placement Algorithm in Hybrid Cloud Environment

Nawaf Alharbe
Applied College, Taibah University,
Saudi Arabia

Mohamed Ali Rakrouki
Applied College, Taibah University, Saudi Arabia
Business Analytics and DEcision Making Lab (BADEM)
at Tunis Business School, Tunisia

***Abstract*—This paper deals with the problem of virtual machine placement in hybrid cloud situation from an economic and QoS perspectives. Because excessive investment of resources in cloud computing environment will result in resources waste, and too few resources can generate QoS issues. This paper uses a game theory model to describe the problem and find the balance between these contradictions. Based on this model, a virtual machine placement algorithm for scheduling virtual resources is proposed. Compared to the traditional game theory, our LBOGT algorithm proposes a game between tripartite sides: users, individual providers and provider groups. Experiments show that our proposed algorithm reduces physical machines energy consumption by 6.16%, and increases by 10.6% in profit provider under the premise of users' QoS.**

***Keywords*—Cloud computing; virtual machine placement; game theory; quality of service; load balancing; energy consumption**

I. Introduction

From the point of view of security and cost, if a small and medium-sized enterprise only needs to meet the needs of its internal personnel, and its business on the data center does not need to be open to the public, this company can build its own private cloud. However, sometimes high-intensity workload for some reason can suddenly usher, and the local private cloud resources cannot normally meet the needs of all the new requests. At this time, the solution for this company is to rent cloud resources from other cloud providers in order to complete their tasks. In this way, the company uses both its own private cloud and the public cloud of other providers, thus combining into a hybrid cloud.

Over time, cloud computing develops more towards the direction of hybrid cloud and cloud alliance [1]. A single cloud provider is largely constrained by the market because of limited cloud resources, limited market radiation range, and limited system throughput. The cloud computing model as an economic model is dependent on the market, so small and medium cloud computing providers join forces with each other to form an alliance [2], [3]. Due to the change of this model, the game participants in the traditional game model are limited to users or a single cloud computing provider, and fail to reflect the collective interests of the entire hybrid cloud, which is flawed from the perspective of game theory. At the same time, as a hybrid cloud collective, it cannot reflect the collective interests well. The income of a collective is not simply to ensure the income of all the individual units to meet the maximum collective revenue.

Therefore, this paper proposes an improved game theory model: taking cloud computing users, a single cloud computing provider, and an aggregate of multiple providers as game theory participants, each representing its own interests. At the same time, under certain conditions, the two are also conflicting interests, and they need to fight for their own interests through games. Among them, cloud users hope that the response time of services is short and the quality of task completion is high, so they need more cloud resources. From the one hand, a single cloud computing provider hopes that its own interests are high, so it needs to reduce the investment of resources and maintenance costs. On the other hand, a provider group hope that cloud users will be satisfied with the service, and hope to increase the income of the entire group and reduce expenditure.

To sum up, this paper focuses on developing a virtual machine (VM) placement algorithm in a hybrid cloud environment. This scheme can describe the relationship between users, cloud resource providers and hybrid cloud systems composed of providers. The scheme comprehensively considers the QoS requirements of users and the benefits obtained by the provider, as well as energy consumption of the system. Finally, this paper proposes a VM placement algorithm based on this game theory to ensure user QoS and improve resource utilization.

II. Literature Review

From the perspective of providers and users, cloud computing is a dynamic process, and its management is a very important issue. For cloud computing providers, the management of cloud resources is very important for the 3 levels of services (IaaS, PaaS, SaaS) to ensure service-level agreements (SLA) [4], standardization, security, high availability, high energy consumption ratio, and resources utilization maximization.

From user's point of view, cloud computing services are divided into two stages [5]. The first one is the pre-processing stage, in which users are faced with the selection of cloud computing services. The second stage is a stage of continuous interaction with providers or cloud computing management, where users need to monitor the performance of their own services. Users will consider whether to continue to use the service or purchase other services according to their own evaluation results. User's QoS and system performance are considered by Calheiros et al. [6]. They designed a predictive load balancing model using queuing theory, and the virtual machine deployment is determined by using QoS target and predicted load. Different QoS requirements analysis for various users is provided by Xu et al. [7]. They proposed a multi-dimensional QoS scheduling strategy based on multi-workflow in cloud environment. According to this model, the task flow is scheduled to meet the preferences of different users. Starting from the needs of users, Bardsiri and Hashemi [8] designed various QoS indicators for service providers, and established a user QoS evaluation model according to these QoS indicators. The similarity of trustworthiness in the cloud environment is studied by Pan et al. [9] to find trusted adjacent services and predict missing QoS values by enhancing similarity of trusted services. Recently, Farzai et al. [10] considered the bandwidth as QoS indicator and proposed a hybrid genetic algorithm for minimizing power consumption, resource wastage, and bandwidth. The proposed method demonstrated high potential of scalability for large problem instances. In order to meet QoS requirements, Jing et al. [11] proposed a QoS-aware scheme based on a Particle Swarm Optimization (PSO) algorithm. Recently, Aloufi et al. [12] This paper discusses how cloud computing provides users with QoS by presenting the concept and characteristics of cloud computing and how machine learning techniques can be applied to resource management. In particular, this survey highlighted the advantages of utilizing machine learning to schedule forthcoming requests in order to achieve QoS and save energy. Edinat et al. [13] reviewed several previously proposed models that have been utilized in the literature to improve QoS and proposed a model based on Deep Reinforcement Learning and an enhanced DRL agent.

Load balancing is a difficult and an important problem in cloud computing. It's directly affects resource utilization, thereby affecting the profitability of cloud computing providers. References [14], [15] used live migration of virtual machines as an optimization goal to reduce energy consumption. A multi-objective ant colony optimization algorithm has been proposed by Fang and Qu [16] in order to balance simultaneously the load among the physical machines and the internal load of machines. Recently, a machine learning-based approach has been proposed by Ghasemi and Haghighat [17] for the same problem. A survey on load balancing algorithms and their classification is provided by Xu et al. [18].

Currently, there are currently two main methods to reduce the energy consumption of cloud computing: Dynamically adjust power consumption of physical server CPUs to save power; Turn off unnecessary services or energy-consuming resources in the cloud computing system to save power. In this way, Sotomayor et al. [19] suspended relatively low-priority tasks to reduce power consumption. Chen et al. [20] proposed power saving approaches by shutting down unnecessary virtual machines according to server load balancing. Recently, energy consumption in cloud computing environment has attracted more researchers [21]–[25].

III. Methodology

A. Member Interests in Hybrid Cloud

Due to the internal heterogeneity of the hybrid cloud system, resource management and scheduling difficulties of this system are greatly increased. There are many different cloud providers in a hybrid cloud system. After analyzing users needs, it is also difficult for the hybrid cloud to submit the analysis results to different providers. Furthermore, each provider must satisfy user's QoS requests. In order to ensure these requests, it is necessary to limit or even reduce the number of user task requests per unit time. However, the reduction of the number of processing tasks will reduce the profit of the provider. To deal with this contradictory relationship between user and provider, we need to build a game model.

This paper assumes that a provider in the hybrid cloud virtualizes physical resources through the hypervisor, and virtual resources will be provided to users in the form of virtual machines. The provider accepts a user's task application and finds that the remaining local resources are not enough to meet the user's needs, and cannot suspend or close tasks and virtual machines using local resources. At this time, the provider will have to follow some scheduling rules. Forcibly shut down some virtual machines, or the provider can transfer this local request to other providers in the hybrid cloud through the hybrid cloud management system, and use resources of other providers in order to complete the local request by outsourcing [26]. The final benefits are managed and distributed uniformly by the hybrid cloud system.

B. Solution Design

Hybrid cloud faces security, price, benefit distribution and other issues [27], [28]. To build a stable hybrid cloud system, we first need to establish a stable provider portfolio, then conduct business modeling, and finally develop a virtual machine deployment solution. Therefore, the design focus of this paper is divided into three parts:

1) Hybrid cloud group member selection model: A stable system must be stable internally, and all internal members of the system can bring benefits to the group. Therefore, the benefits of members in the group are greater than the benefits of running alone.
2) A game model: It is an economic model that can reflect the interests of multiple parties, can quantify and describe the relationship between QoS problem, interest problem and energy consumption problem of the hybrid cloud system. In addition, it can participate in solving the virtual machine deployment and revenue problems of the hybrid cloud. This paper adds a player to the traditional game-theoretic model, the individual resource provider, in order to make the model more flexible and able to safeguard the interests of all providers.
3) A virtual machine placement algorithm: It's an optimization algorithm proposed according to the game model. The purpose is to satisfy the user's QoS and improve resource utilization.

The overall design of the system is depicted in Fig. 1.

IV. Game Model Building

A. Metrics used by the Model

1) Stable Hybrid Cloud Membership - Pareto Optimization: Pareto optimization is an economic concept proposed by the Italian economist Vilfredo Pareto. Pareto optimization means that there are no losers in a change in a model, and at least one member can profit.

A cooperative game needs to meet a precondition, that is, to follow the principle of Pareto optimization. For game participants, the global benefit of the entire game must be

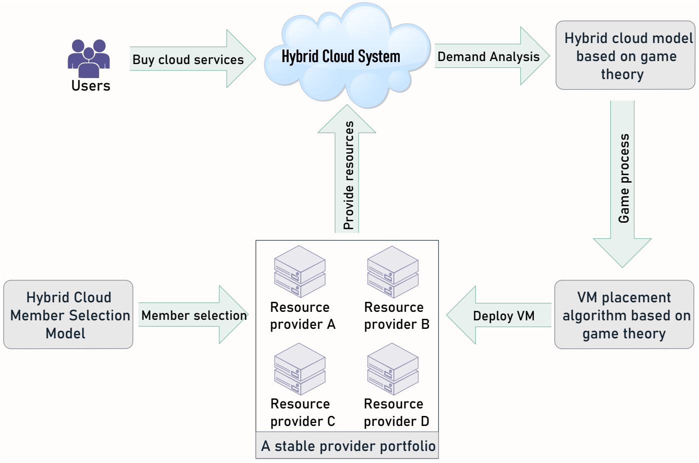

Fig. 1. Hybrid Cloud Virtual Machine Deployment Diagram.

greater than the benefit obtained by each player acting alone. Considering the stability of the hybrid cloud, there is a similar nature. Members of a stable team portfolio will not gain higher benefits if they leave the team to act alone or join a new team.

2) Hybrid Cloud Member Benefit Distribution Principle - Shapley Value: Shapley value refers to the fair distribution of the overall benefit according to the contribution value of members to the group. The Shapley value of a member represents the average expected contribution value of the member to the group.

Let $I = \{1, 2, ..., n\}$ be a set, any subset S of I corresponds to a function $V(S)$, if it satisfies:

$$\begin{cases} V(\emptyset) = 0 \\ V(S_i \cup S_j) \geq V(S_i) + V(S_j) \\ S_i \cap S_j = \emptyset \\ \forall S_i, S_j \in I \end{cases} \tag{1}$$

Then $[I, V]$ is called the n-player cooperative game, and V is the characteristic function of the game. $V(S)$ is the payoff of the cooperative strategy.

Here, $X_i (i = 1, 2, ..., n)$ is used to denote the income that member i in I should get from the maximum benefit $V(I)$ of cooperation, and $X = (X_1, X_2, ..., X_n)$ is the allocation of the maximum benefit in the cooperative strategy and satisfy:

$$\begin{cases} \sum\limits_{i=1}^{n} X_i = V(I) \\ X_i \geq V(i); \forall i = (1, 2, ..., n) \end{cases} \tag{2}$$

Then, the Shapley value can be written as $X = (X_1, X_2, ..., X_n)$ where:

- $X_i = \sum\limits_{s \in S_i} W(|s|)[V(s) - V(s/i)]$
- $W(|s|) = \dfrac{(n - |s|)!(|s| - 1)!}{n!}$
- S_i is all subsets of I that contain i
- $|s|$ is the number of elements in the subset
- $W(|s|)$ is the weighting factor.

Shapley value is used in this paper to measure the provider's contribution to the team, that is, how much the game player can add to the entire team combination when the game player joins the provider combination scheme. The higher the contribution value, the higher the benefit value allocated to the cloud provider. If the Shapley value is negative, it means that the provider's joining cannot increase the benefits of the alliance, and its joining will not be considered. The Shapley value of the cloud provider can be calculated by Eq. 3.

$$\varphi_i(v) = \sum_{S \subseteq RP} \frac{|S|!(|RP| - |S| - 1)!}{|RP|!}[v(S \cup \{RP_i\}) - v(S)] \tag{3}$$

where the cloud provider set is denoted by $RP = \{RP_1, RP_2, ..., RP_n\}(n \in \{1, 2, ..., N\})$, v corresponds to the characteristic function V of the game in the Shapley value, and the calculation method is shown in formula Eq. 4. A hybrid cloud is defined as a combination of providers, denoted by S, where $S \subseteq RP$ and $S > 1$.

B. Principles of Member Selection in Hybrid Cloud

The cloud computing service provider in this paper is not a single one, but a combination of several cloud computing providers. The choice between a group of providers and a single provider outside the group is a two-way street, with a single provider choosing whether to join the group, and the group considering whether to allow a single provider to join. The specific selection model of a single provider is not considered in this paper. This is because a single provider considers whether to join or not, and needs to integrate its own situation and whether the agreement between the two parties is beneficial to itself. There are too many variables to discuss. Therefore, the group composed of cloud computing providers needs to be stable, and the average revenue of a single provider is beneficial to the team, mainly satisfying the following points:

1) First, determine whether the resources owned by all providers in the provider team can meet the constraints of the system on resources.
2) For all providers in the group, the benefits obtained by joining the group must be higher than the benefits obtained by operating alone. In this way, during the operation of the group, no provider will consider withdrawing the group, so that this group is a stable combination.
3) If the team's combination scheme is optimal, on the premise of satisfying the above two points, a combination of providers needs to be selected. The average income obtained by this combination scheme is the highest, which can not only satisfy resource constraints and user QoS, but also guarantee the maximum benefit of members.

Users need to pay according to their own tasks and QoS preferences when purchasing cloud computing services. When providers consider service pricing, they need to price each resource used by users separately. The final price is calculated from the price of resources offered by a single provider. The price of k-type resources in the hybrid cloud S can be calculated by Eq. 2, where δ is a price correction parameter.

$$price_S^k = e^{\delta|S|} \cdot \overline{price_p^k} \tag{4}$$

Let R_k be the number of various resources in a hybrid cloud S, then the total profit $profit(S)$ per unit time of the cloud S can be calculated by Eq. 5.

$$profit(S) = \sum_k price_S^k \cdot R_k \tag{5}$$

Assume a stable hybrid cloud combination scheme $S = \{RP_1, RP_2, ..., RP_S\}$, the physical machine of all available resources is $\{m_1^1, ..., m_{M^1}^1, ..., m_i^k, ..., m_1^S, ..., m_{M^S}^S\}$, a physical machine can be represented by m_i^k, where $1 \leq i \leq M^k$, M^k is the number of available physical machines provided by the cloud provider RP_k, the remaining available resources on each physical machine m_i^k can be represented by a resource vector $\overrightarrow{R}(m_i^k)$. The tasks submitted by users $\{1, 2, ..., U\}$ are numbered from 1 to N, and each task corresponds to a virtual machine. The set $A = \{A_1^1, ..., A_x^u, A_y^u, ..., A_N^U\}$ represents all virtual machine allocation policies of the resource scheduling center, where $A_n^u = (A_n^u(m_1^1), ..., A_n^u(m_{M^1}^1), ..., A_n^u(m_i^k), ..., A_n^u(m_1^S), ..., A_n^u(m_{M^S}^S))^T$, $A_n^u(m_i^k) = (a_{n1}^u(m_i^k), ..., a_{n2}^u(m_i^k), ..., a_{nZ}^u(m_i^k))$ represents the number of resources allocated to task n on the physical machine m_i^k.

C. Hybrid Cloud Group Member Selection Model

The cloud resource provider in the hybrid cloud is a set, and there are multiple providers. Therefore, according to this set, multiple different subsets can be formed. These subsets are different composition methods of multiple providers, which means that some providers can be included in the hybrid cloud, and some cannot be included. So, we need to judge which providers can be included in the hybrid cloud and bring benefits to the whole. There are two main ways to judge:

1) **Stability:** A hybrid cloud is called stable, when any provider in the hybrid cloud benefits more if it leaves the cloud [29]. It can also be said that this hybrid cloud is the best choice for this provider. Generally speaking, the providers in a stable hybrid cloud should be mutually beneficial, and the providers should exist more in the form of cooperation rather than competition, and their competitors are outside the cloud system. A stable hybrid cloud should also bring more user resources to all providers, and users are more inclined to use the hybrid cloud rather than the system resources provided by a single provider. Fig. 2 shows the filtering process of a stable hybrid cloud combination scheme, in which Pareto optimization means that each member of the combination scheme brings positive benefits to all other members. At the same time, the income obtained in the group is also higher than from not participating in the group.

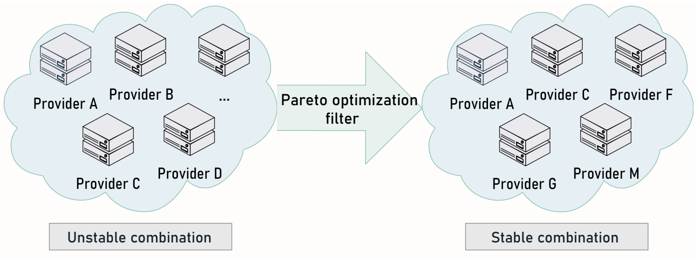

Fig. 2. Filter a Stable Provider Portfolio.

2) **Fairness:** A provider in the hybrid cloud hopes that the system is fair, which refers to the fairness of processing user requests, that is, the fairness of scheduling, and the fairness of benefit distribution. Each member should be fairly assigned tasks, and then the corresponding benefits should be distributed according to the completion of the tasks. The distribution of benefits in this paper is based on the Shapley value.

A hybrid cloud cooperation game can be expressed as $G = aP, v$, where aP is all available providers RP in the hybrid cloud, v is the characteristic function, for a provider combination S. Then, this paper defines its characteristic function as $v(S)$ as follows.

$$v(S) = \begin{cases} 0; & \text{(insufficient total resources)} \\ profit(S); & \text{(total resources met)} \end{cases} \tag{6}$$

When $v(s)$ is 0, it means that the hybrid cloud combination S is not feasible. Firstly, according to Eq. 6, we select the provider combinations that satisfy resource constraints, and then judge the overall benefits of these hybrid cloud combinations. The final total benefit is uniformly distributed by the contribution value calculated by the Shapley value. The contribution ability of the provider, that is, the ratio of $\varphi_i(v)$ to the total contribution is the measure. Then, the benefit value that each provider can obtain in the hybrid cloud combination scheme S is calculated as follows.

$$value_i(S) = \frac{\varphi_i(v)}{\sum\limits_{RP_x \in S} \varphi_x(v)} \cdot v(S) \tag{7}$$

The value of each provider's benefits when not participating in the group is:

$$value_i = \begin{cases} 0 & \text{; (Available resources meet demand)} \\ \sum\limits_k price_S^k \cdot R_k & \text{; (Available resources don't meet demand)} \end{cases} \tag{8}$$

The hybrid cloud combination scheme S is said to be stable, if $value_i(S) > value_i$; $\forall RP_i \in S$

D. User QoS Indicators in Game Model

Since the primary goal of cloud computing services is to ensure user QoS, it is necessary to model and evaluate user QoS preferences. Therefore, the user QoS indicators in this paper consist of the following five elements: Response time, cost, system availability, reliability and trustworthiness.

The user QoS indicators in a hybrid cloud environment are represented by a vector $\{Q_1, Q_2, Q_3, Q_4, Q_5\}$, in which all QoS indicators are in the $[0, 100]$ interval. Each indicator corresponds to a weight, which is determined according to the user's QoS preference and the type of service purchased when the user signs an SLA with the provider. The user's QoS preference and the type of service purchased will affect the division of QoS indicators. The weight value of QoS is also represented by a set of vectors $\{w_1, w_2, w_3, w_4, w_5\}$. Therefore, according to the above analysis, the overall QoS index of the service purchased by the user can be expressed by Eq. 9, where $\{q_1, q_2, q_3, q_4, q_5\}$ corresponding to the response time, cost, system availability, reliability, and trustworthiness, respectively.

$$QoS = \sum_{i=1}^{5} w_i \cdot q_i \tag{9}$$

One of the most important QoS indicators is service response time [30]. The computation amount of a task can be estimated from the task type submitted by the user and the environment variables combined with the historical data. It is then calculated according to the computing power $PC(m_i^k)$ of the virtual machine assigned to the user. The response time estimation for the user task is as follows.

$$RT_n = \frac{W_n}{PC(m_i^k)} \tag{10}$$

From the user's point of view, the shorter the response time, the better. This paper unifies the response time $q_1 = 100 - RT_n/Q_1$. It can be seen that the longer the response time, the less q_1.

From the provider's point of view, the provider hopes to maximize its own benefits, but high cost will affect the user experience. The resource pricing in this paper is represented by $price_k^S$, from which the cost required by the user is:

$$Cost(r_n) = \sum_{k=1}^{K} r_{nk} \cdot price_k^S \tag{11}$$

In this paper, in order to reflect the cost problem of the provider, when the resource price is less than a certain value, the user's QoS experience is certain and will not increase because of this. Therefore, the user service cost is divided into two levels. In the interval $[0, mark_u]$, the user QoS index q_2 is set to 100, and in $[mark_u, Q_2]$, the higher the service charge, the lower the q_2 value. The normalized q_2 can be represented by Eq. 12.

$$q_2 = \begin{cases} 100 & ;\text{if } Cost(r_n) \le mark_u \\ \dfrac{100 - Cost(r_n) - mark_u}{Q_2 - mark_u} & ;\text{if } mark_u < Cost(r_n) \le Q_2 \end{cases} \tag{12}$$

Since there are no specific data studies on effectiveness, reliability and trustworthiness in QoS, these three are not studied in depth in this paper. The three settings are as follows: The availability and reliability are mainly obtained by referring to the running status of the server in the historical data. The credibility can use the user's historical evaluation and the system itself to set the evaluation of the previous service completion.

The above three parameters q_3, q_4, q_5 give QoS ratings respectively according to the evaluation results. Then, the three indicators are unified according to the first two parameters, respectively, corresponding to a certain value in the interval $[0, 100]$, representing the scoring result of this attribute.

E. The Proposed Game Theory Model

The game theory model in this paper mainly considers the relationship between the following three in the cloud computing environment: cloud service users, a single provider, and a group of multiple providers. The purpose of using game theory is to use a model that can describe the interests of the three parties at the same time. This model must be objective, and can adapt to different applicable occasions through parameters adjustments. For example, in some occasions, it favors user QoS, and in some cases, it favors resource utilization, that is, multiple providers. Based on the above analysis, the game theory model in this paper establishes three optimization goals according to three stakeholders: user QoS optimization, server resource utilization optimization, and system maintenance cost optimization.

According to the relevant introduction above, we first establish the resource constraints of a hybrid cloud combination, and then establish the combination model of resource providers, and calculate the optimal provider combination through the constraints and the overall revenue value. Then a tripartite game model is established to calculate the overall total revenue of the system. The hybrid cloud game model in this paper is represented by a quadruple $\{Players, Req, A, Utility\}$ as follows:

1) $Players$ are players in the game, that is, users who submit resource requests, the hybrid cloud, and the providers in the hybrid cloud.
2) Req is the resource request matrix formed by the task submitted by the user.
3) $A = \{A^u | u = 1, 2, ..., U\}$ represents the deployment placement scheme of virtual machines allocated to users in the system.
4) $Utility$ is a global revenue function. The utility describes the total revenue that can be obtained by the entire system of a virtual machine deployment solution. The larger the value, the better.

Utility is divided into three parts according to three stakeholders, the first part is user QoS and has been introduced above. Then, the calculation method of the revenue value representing the resource utilization of the provider group is introduced. For a stable provider combination S, there are N tasks ready to be allocated at a certain time, and the virtual machine resource vector corresponding to the task is $\overrightarrow{r_n}$. The initial total resource vector on physical machine m_i^k is $\overrightarrow{TM}(m_i^k)$, $utl_t^{(m_i^k)}$ (calculated as in Eq. 13) is the utilization rate of resources of type t on physical machine m_i^k, and the average utilization rate of the resources on the physical machine is $\overline{utl_t^{(m_i^k)}}$.

$$utl_t^{(m_i^k)} = 1 - \frac{R_t(m_i^k) - \sum_n x_{nt}^{(m_i^k)}}{PM_t(m_i^k)} \tag{13}$$

where $R_t(m_i^k)$ represents the number of resources available on physical machine m_i^k, $x_{nt}^{(m_i^k)}$ is the number of resources of type n allocated to the task on physical machine m_i^k, and $PM_t(m_i^k)$ represents the total number of resources t on physical machine m_i^k.

The portion of the global benefit that represents the provider's interests is:

$$MT_i = \sum_{k=1}^{K} \omega_k^P \cdot N_k^P \cdot value_k^P \tag{14}$$

where $value_k^P$ represents the unit price of the corresponding resource, N_k^P represents the usage of the corresponding resource, and ω_k^P represents the price correction parameter of the corresponding resource. So the global revenue of the game is calculated by Eq. 15.

$$Utility = \alpha \cdot \frac{\sum_{n=1}^{N} \text{QoS}_n}{N} + \beta \cdot \frac{\sum_{y=1}^{M_N} MT_y}{M_N} + (1-\alpha-\beta) \cdot \frac{\sum_{i=1}^{M_N} \overline{utl^{(m_i^k)}}}{M_N} \tag{15}$$

where MT_y is the revenue of a single cloud resource provider.

In order to ensure the fairness of the three benefits, the three parts of the above equation can be normalized. From Eq. 15, it can be seen that the relationship between the three optimization objectives can be balanced by adjusting the value of the tripartite interest weights α and β.

V. Virtual Machine Placement Algorithm

A. Load Balancing Optimization in Physical Servers

When a cloud computing system creates a virtual machine on a physical server, because both are multi-dimensional resources, it cannot guarantee that the proportion of virtual machine resources matches the resource ratio of the physical machine. For example, the CPU resources may have been fully occupied by the virtual machine, but there are still a lot of memory resources left; or the memory resources are fully allocated but the CPU resources are still left. This part of the remaining resources will not be able to continue to be allocated, resulting in resource fragmentation and waste of resources. Therefore, for the sake of load balancing, the proportion of resources allocated to virtual machines should preferably be similar, so that resource fragmentation is not easy to generate. Thus, ensuring balanced operation of each physical server, reducing energy and resource waste, and improving utilization. Thereby, the system throughput can be increased at the same cost.

Because cloud users have various needs, in general, cloud resource providers will also provide virtual machine customization functions for cloud users, so the virtual machine requests of cloud users received by the hybrid cloud system will be various. The proportion of various resources in the machine is also uneven. First of all, if it is assumed that the resource ratio of all virtual machines is the same, then each physical server can ensure the load balancing of virtual machine resources. Therefore, this paper proposes an algorithm for load balancing optimization as follows:

1) When the hybrid cloud system is initialized, virtual machines with large difference in resource ratio are preferentially allocated, so that virtual machines with different resource ratios can complement each other's deficiencies.
2) When the hybrid cloud is running, if there is a new virtual machine assignment task, the Nash equilibrium of game theory is used to calculate whether to start a new physical server or not.
3) When the hybrid cloud is running, if a virtual machine needs to be shut down, the Nash equilibrium of the same calculation game theory determines whether to carry out the migration work and migration target of other virtual machines.

We calculate the average resources $\overline{Res}$ of physical servers as a reference standard for the proportion of virtual machine resources as follows.

$$\begin{aligned} \overline{Res} &= \{\overline{PE}, \overline{Mem}, \overline{Store}\}^T \\ &= \left\{ \frac{\sum_{i=1}^{n} PE_i}{n}, \frac{\sum_{i=1}^{n} Mem_i}{n}, \frac{\sum_{i=1}^{n} Store_i}{n} \right\}^T \end{aligned} \tag{16}$$

where PE represents the number of processor cores corresponding to the physical server, Mem represents the amount of memory, and $Store$ represents the number of external storage memory. The amount of resources required by a virtual machine can also be expressed as $VRes$.

The skewness of virtual machine resources is calculated from the Euclidean distance between virtual machine resources and physical machine resources as follows.

$$\begin{aligned} &VVari_t = \sqrt{(VPE_i - \overline{PE})^2 + (VMem_i - \overline{Mem})^2 + (VStore_i - \overline{Store})^2} \\ &\forall i = (1, 2, ..., n) \end{aligned} \tag{17}$$

The first step of the optimization algorithm is initialization. In this step, for each virtual machine to be created and its corresponding candidate physical machine, the utility value $Utility$ is calculated and sorted. All physical machines are initially marked as inactive. This optimization problem is solved using a greedy algorithm. Assuming that an optimal deployment scheme for $n-1$ virtual machines has been obtained, the optimal deployment scheme for the n virtual machine is to find a corresponding running state physical machine to ensure that the available resources are sufficient to create the virtual machine, and the benefits value is maximized.

The best physical machine that satisfies the conditions is directly enabled at the beginning. After several iterations of creation, the available resources on the best physical machine selected may not be enough to create the n virtual machines. At this time, the physical machine needs to make a decision, to determine whether it is necessary to delete some received virtual machines to make room to create new ones. The selection order is to delete from the maximum skewness between the virtual machine resource type and the total resources of the physical machine until the available resources are sufficient to create a new virtual machine.

The decision on whether to choose to replace a virtual machine is based on the utilization rate corresponding to the least used resource type (Eq. 18) on the physical machine before and after replacement. If the utilization rate increases after replacement, it means that the replacement is beneficial and the virtual machine can be replaced.

$$\begin{aligned} utl_{\min}^{(m_i^k)} &= \min_t\{utl_t^{(m_i^k)}\} \\ &= \min_t \left\{ 1 - \frac{R_t(m_i^k) - \sum\limits_n x_{nt}^{(m_i^k)}}{PM_t(m_i^k)} \right\} \end{aligned} \tag{18}$$

B. Algorithm Flow

Based on the above analysis, this paper proposes a game theory-based load balancing optimization algorithm (LBOGT). The idea of the algorithm: When placing virtual machines in the hybrid cloud system, it first obtains the resource information of the applied virtual machines from the application list. Next, the resource skewness of each virtual machine is calculated according to the resource information, and sorted in descending order of skewness. Then, put them in the $vmlist$ table and wait for the system to schedule. The system sequentially fetches the virtual machines to be scheduled from the $vmlist$, and calculates the virtual placement position with the greatest benefit according to the global revenue $Utility$. When the virtual machine is placed, there may be insufficient physical machine resources. At this time, calculate whether to migrate the virtual machine according to Eq. 18. If the virtual machine needs to be migrated, it is treated as a new virtual machine to be deployed. Otherwise, apply for new physical resources.

The detailed process of the LBOGT algorithm is given in Fig. 3.

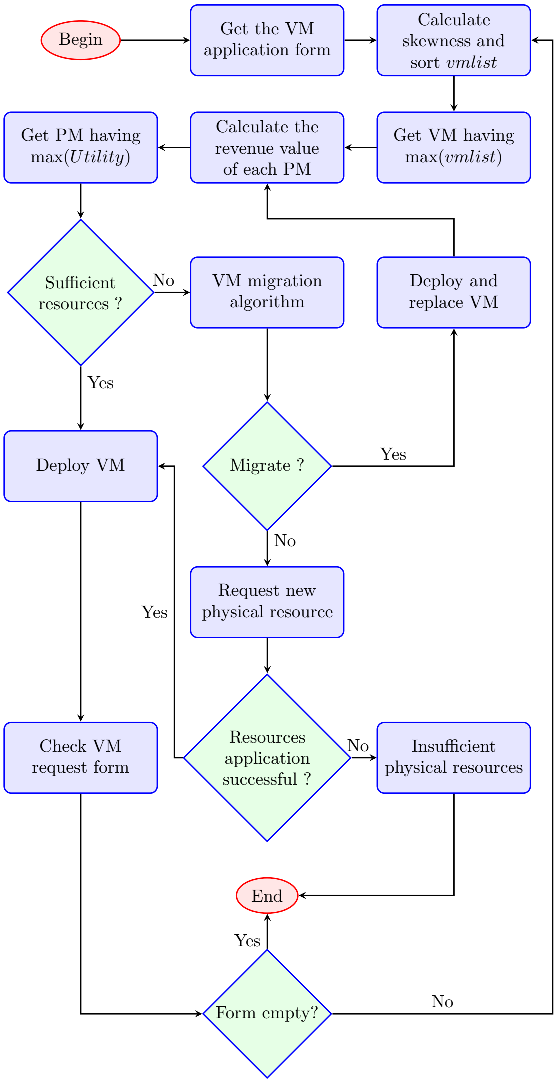
Begin
Get the VM application form
Calculate skewness and sort $vmlist$
Get VM having max($vmlist$)
Calculate the revenue value of each PM
Get PM having max($Utility$)
Sufficient resources ?
No
Yes
VM migration algorithm
Deploy and replace VM
Deploy VM
Migrate ?
Yes
No
Request new physical resource
Yes
Check VM request form
Resources application successful ?
No
Insufficient physical resources
End
Yes
Form empty?
No

Fig. 3. LBOGT Algorithm Flowchart.

VI. Experimental Results

This section will validate the aforementioned provider team member selection model and our proposed virtual machine placement algorithm based on load balancing optimization (LBOGT). This paper uses CloudSim [31] to build a cloud computing simulation experimental environment, and uses JAVA programming to test the algorithm. CloudSim is an open and extensible simulation framework that integrates modeling, simulation and experimentation into one cloud computing environment for experiments. The physical machine configuration in this paper is shown in Table I. Then, simulate 24 hours of data from a hybrid cloud data center. Our proposed algorithm LBOGT is compared to Inter Quartile Range (IQR), Median Absolute Deviation (MAD), Static Threshold (THR) and Game Theory (GT) algorithms.

A. Selection of Hybrid Cloud Group Members

TABLE I. Hybrid Cloud System Resource Constraints

Attributes	Ranges
Number of physical machines (host)	[400, 1000]
Number of processor (CPU) cores	[1600, 32000]
Processor (CPU) frequency	[0.50GHz, ∞]
Physical machine memory size	[6400GB, 128000GB]
Physical machine external memory size	[400TB, 10000TB]

There are currently 5 providers to be selected. The processor of the providers are Intel Xeon of the following models: E5-2620, E5-2603, E5-2609, E7-4807, E7-8837. The number of virtual machine requests is 800, and other configurations are shown in Table II.

TABLE II. The Candidate Providers of the Hybrid Cloud System

Provider	Number of CPU cores	CPU frequency	RAM size	External storage size	CPU unit price	RAM unit price	External storage unit price	Price correction parameter δ
1	1680	2.10GHz	8960GB	560	0.01	0.004	0.06	1.02140275
2	1280	1.80GHz	10240GB	640	0.008	0.003	0.06	1.16487403
3	1560	2.40GHz	8320GB	520	0.008	0.002	0.07	1.05145024
4	1680	1.86GHz	8960GB	560	0.015	0.003	0.05	1.13123516
5	1920	2.66GHz	15360GB	480	0.015	0.004	0.05	1.19472345

Based on the constraints shown in Table I, the configuration properties made up of all provider combinations are now calculated. There are a total of 5 providers, excluding the 5 possibilities of a single provider, then a total of 26 different provider combinations are possible.

From Table III, it can be seen that it is not that the more providers, the greater the average revenue value. This is because when a provider with a poorly configured physical server joins, the team will have an evaluation of the newly added provider's configuration, and a poorly configured server may affect all user experiences. Therefore, it is necessary to select an optimal combination among all combinations. This combination needs to meet the resource constraints of the team, and the average host revenue should be the highest. The average host revenue value refers to the team's total revenue evenly distributed to each host.

TABLE III. Restrictions Results

Provider combination	Total number of hosts	Total core	Total memory	Total external storage	Restrictions
1,2	600	2960	19200	1200	-
1,3	540	3240	17280	1080	Insufficient hosts
1,4	560	3360	17920	1120	Insufficient hosts
1,5	520	3600	24320	1040	Insufficient hosts
2,3	580	2840	18560	1160	Insufficient hosts
2,4	600	2960	19200	1200	-
2,5	560	3200	25600	1120	Insufficient hosts
3,4	540	3240	17280	1080	Insufficient hosts
3,5	500	3480	23680	1000	Insufficient hosts
4,5	520	3600	24320	1040	Insufficient hosts
1,2,3	860	4520	27520	1720	-
1,2,4	880	4640	28160	1760	-
1,2,5	840	4880	34560	1680	-
1,3,4	820	4920	26240	1640	-
1,3,5	780	5160	32640	1560	-
1,4,5	800	5280	33280	1600	-
2,3,4	860	4520	27520	1720	-
2,3,5	820	4760	33920	1640	-
2,4,5	840	4880	34560	1680	-
3,4,5	780	5160	32640	1560	-
1,2,3,4	1140	6200	36480	2280	-
1,2,3,5	1100	6440	42880	2200	-
1,2,4,5	1120	6560	43520	2240	-
1,3,4,5	1060	6840	41600	2120	-
2,3,4,5	1100	6440	42880	2200	-
1,2,3,4,5	1380	8120	51840	2760	-

In order to calculate the total revenue for each provider group, firstly we calculate the pricing of all system resources according to Eq. 4. The price correction parameter δ in the formula are obtained according to the resource evaluation of the resource provider as shown in Table II. The total value of the group's income is calculated by Eq. 5, and the final calculation results are shown in Table IV.

TABLE IV. Group Income Statement

Provider combination	CPU pricing	Memory pricing	External storage pricing	Total team revenue	Average hosting revenue
1,2	0.00982699	0.00377042	0.06587525	180.53017618	0.30088363
1,3	0.00934619	0.00313097	0.06721474	156.97679288	0.29069776
1,4	0.01189442	0.00373966	0.05892296	172.97365957	0.30888153
1,5	0.01432433	0.00452347	0.06056971	224.57098009	0.43186727
2,3	0.00882057	0.00287075	0.07155513	161.33542275	0.27816452
2,4	0.01173446	0.00344753	0.06367146	177.33228943	0.29555382
2,5	0.01448011	0.00426519	0.06553975	228.92960995	0.40880287
3,4	0.01108883	0.00277221	0.06476609	153.77890613	0.28477575
3,5	0.01365808	0.00383868	0.06694615	205.37622665	0.41075245
4,5	0.01589270	0.00426856	0.05802687	221.37309333	0.42571749
1,2,3	0.00933849	0.00326628	0.06821110	249.42119591	0.29002465
1,2,4	0.01118396	0.00365055	0.06291186	265.41806259	0.30161143
1,2,5	0.01301146	0.00421863	0.06412122	317.01538311	0.37739927
1,3,4	0.01079012	0.00322069	0.06357714	241.86467929	0.29495693
1,3,5	0.01253676	0.00390646	0.06491364	293.46199981	0.37623333
1,4,5	0.01408585	0.00421931	0.05916692	309.45886649	0.38682358
2,3,4	0.01058763	0.00304101	0.06667357	246.22330915	0.28630617
2,3,5	0.01249127	0.00373481	0.06809592	297.82062967	0.36319589
2,4,5	0.01416845	0.00403925	0.06254709	313.81749636	0.37359226
3,4,5	0.01363098	0.00371653	0.06321842	290.26411306	0.37213348
1,2,3,4	0.01048640	0.00329758	0.06534985	334.30908231	0.29325358
1,2,3,5	0.01189721	0.00380811	0.06636202	385.90640283	0.35082400
1,2,4,5	0.01315573	0.00404879	0.06223136	401.90326952	0.35884220
1,3,4,5	0.01279173	0.00379602	0.06270749	378.34988622	0.35693385
2,3,4,5	0.01277394	0.00366354	0.06515995	382.70851608	0.34791683
1,2,3,4,5	0.01224430	0.00373649	0.06437356	470.79428924	0.34115528

It can be seen from Table IV that the best combination that meets the constraints is the provider combination (1, 4, 5). The revenue of the host unit reached 0.38682358, and the average revenue of team members was the largest.

B. QoS Performance Test

The QoS performance test mainly compares IQR, MAD, THR, GT and LBOGT algorithms. In the comparison process, this paper sets the parameters α and β of Eq. 15 to 0.8 and 0.1, respectively, which can clearly reflect the QoS improvement of LBOGT algorithm. IQR, MAD, and THR algorithms are all built-in algorithms in CloudSim simulation software, which are used as reference in this paper.

THR algorithm is a short-answer threshold algorithm with fast algorithm speed and poor optimization results. Since IQR algorithm only tries to optimize the discrete degree of the deployment results during the virtual machine deployment process, the interquartile range of the deployment results is as small as possible, so that the load balance of the virtual machines can be guaranteed to a certain extent. IQR algorithm is a rough state optimization of the overall result, it is not specific to each physical machine, so the result is still far from the optimal solution. MAD algorithm weakens the individual extreme values in the deployment process, thereby ensuring the load balancing effect of most virtual machines. It is a deployment scheme that sacrifices a few and satisfies the overall deployment scheme. MAD sacrifices the performance of a few virtual machines. The optimization result also has a certain distance from the optimal solution.

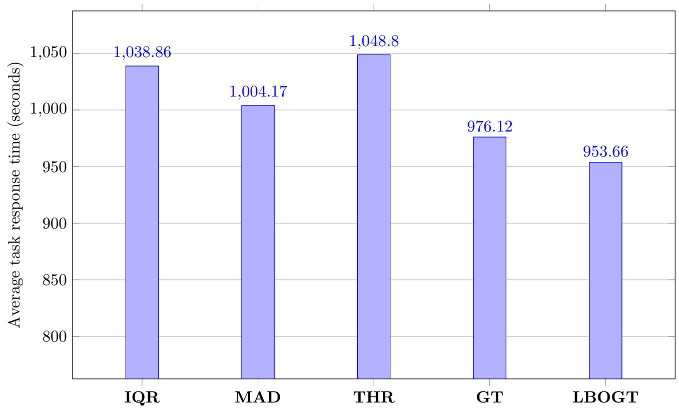

Fig. 4. Task Response Time Comparison.

From Fig. 4, it can be seen that the response time of LBOGT and GT algorithms is shorter, because compared with other reference algorithms, LBOGT and GT consider all QoS preferences of users, including five preference attributes of QoS, so the response time is better. Compared to GT, LBOGT increases the optimization of virtual machine deployment. Therefore, the performance of LBOGT's response time is slightly better than that of GT, and the effect is increased by 2.3%, because the response time of the system to tasks when there is not too much pressure is similar.

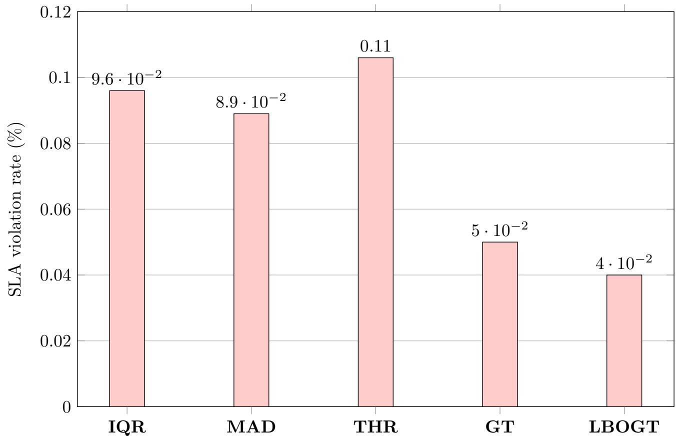

Fig. 5. SLA Violation Rate Comparison.

From Fig. 5, it can be seen that the SLA violation rate of the LBOGT model is 0.01% lower than that of GT model. This is because when the virtual machine placement algorithm is executed, there is an optimization process for LBOGT to select a virtual machine. The algorithm first selects virtual machines with larger resource skewness for deployment. In this way, virtual machines whose skewness can compensate each other can be deployed to a server as much as possible during the deployment process, so that the server can achieve load balancing. It reduces server resources waste, optimizes resource allocation, and allows users' tasks to be better executed. The higher the value of α in the utility function (Eq. 15), the higher the user's QoS weight is, so the game model will consider the user's QoS more. The smaller the α, the higher the individual provider's income, because the provider and the user have conflict of interest to a certain extent. The provider expects low cost and high income, while the user expects low consumption and high QoS. In the utility function, the lower the α and β, the higher the resource utilization of the hybrid cloud system, which can reduce resources waste and reduce maintenance costs. The other three algorithms have higher SLA violation rate values because they do not consider too much user QoS.

C. Power Consumption Performance Test

Before testing the power consumption performance of the LBOGT algorithm, it is necessary to set α and β parameters of Eq. 15 to 0.1. The performance of the algorithms in term of power consumption is presented in Fig. 6.

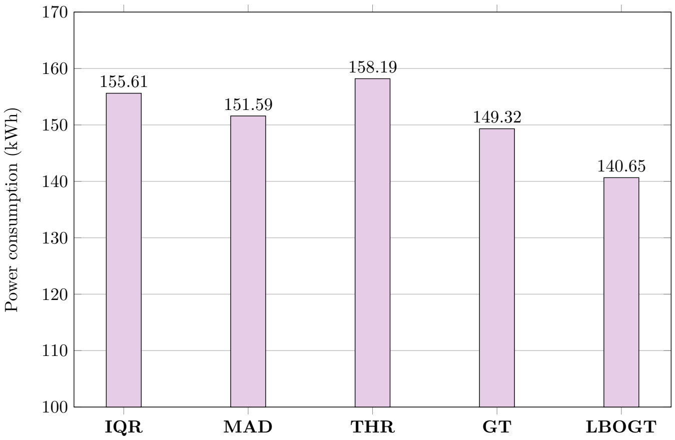

Fig. 6. Power Consumption Comparison.

Fig. 6 shows that LBOGT has great advantage in power consumption, saving 6.16% of energy than GT algorithm. Because there is a virtual machine optimization selection step in the process of LBOGT algorithm compared to GT. The virtual machine deployment order can be optimized to achieve better load balancing effect, thus reducing the waste of resources. On the other hand, LBOGT considers that there will be new requests during the operation of the hybrid cloud system, so there will be dynamic addition of virtual machines. LBOGT will also optimize the newly added virtual machine requests, so it is also conducive to the best deployment.

In the following, it is the comparison of the benefits of our proposed LBOGT algorithm and the traditional game theory-based algorithm GT. The comparison here refers to the income value obtained after pricing various resources when modeling the provider's interests. The income value is not added to the unit when setting it. It is a relative value and is used as the global income of the game theory. For judgment and comparison, see Table IV for details. Since the LBOGT algorithm has made improvements to the deficiencies of GT algorithm in the management of virtual machines to be deployed, and the provider's profit has been considered when formulating the global profit function of the game, it can be seen from Fig. 7 that the improved LBOGT algorithm can bring greater benefits. This benefit is more obvious when the task request is larger. Indeed, when the number reaches 1600, the benefit value increases by 10.6%.

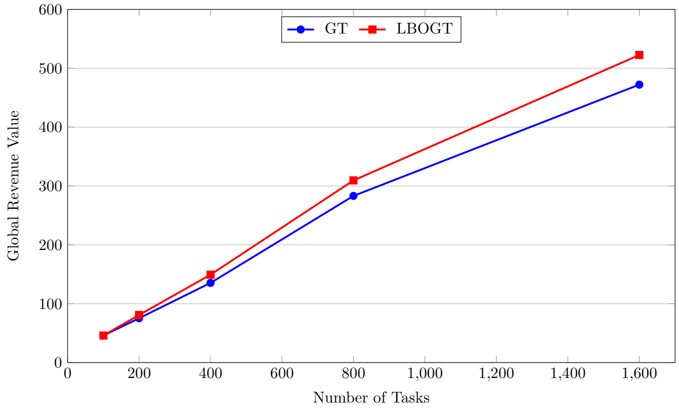

Fig. 7. Global Revenue Comparison.

VII. Conclusion

This paper considers the problem of virtual machine placement problem in hybrid cloud situation from the perspective of economics and QoS. In order to coordinate user QoS and cloud resource utilization, a game theory model of a three-party game is established, and the relationship between the three parties is reflected through a game's profit function. Finally, a virtual machine placement algorithm based on game theory and load balancing optimization is proposed. The algorithm mainly selects the virtual placement position through the game theory revenue function, and optimizes the placement order of the virtual machine through its resource application ratio. The experimental results show that our proposed LBOGT algorithm can coordinate and balance user QoS and cloud computing resource utilization. When the number of executed tasks reaches 1600, the cloud computing revenue is 10.6% higher than that obtained by the traditional game theory method.

Several various adjustments, testing, and experiments have been left to be completed in the future. More virtual machine placement objectives can be investigated in the future, in particular SLA-aware placement schemes.

References

[1] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures," *Computer*, vol. 45, pp. 65–72, 2012.

[2] H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, "Intelligent workload factoring for a hybrid cloud computing model," in *Proceedings - 5th 2009 World Congress on Services*, 2009, pp. 701–708.

[3] Z. Wu and H. V. Madhyastha, "Understanding the latency benefits of multi-cloud webservice deployments," *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 13–20, 2013.

[4] V. C. Emeakaroha, M. A. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. D. Rose, "Towards autonomic detection of sla violations in cloud infrastructures," vol. 28, pp. 1017–1029, 2012.

[5] Z. ur Rehman, O. K. Hussain, F. K. Hussain, E. Chang, and T. Dillon, "User-side qos forecasting and management of cloud services," *World Wide Web*, vol. 18, pp. 1677–1716, 2015.

[6] R. N. Calheiros, R. Ranjany, and R. Buyya, "Virtual machine provisioning based on analytical performance and qos in cloud computing environments," in *Proceedings - the International Conference on Parallel Processing*, 2011, pp. 295–304.

[7] M. Xu, L. Cui, H. Wang, and Y. Bi, "A multiple qos constrained scheduling strategy of multiple workflows for cloud computing," in *Proceedings - 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications*, 2009, pp. 629–634.

[8] A. K. Bardsiri and S. M. Hashemi, "Qos metrics for cloud computing services evaluation," *International Journal of Intelligent Systems and Applications*, vol. 6, pp. 27–33, 2014.

[9] Y. Pan, S. Ding, W. Fan, J. Li, and S. Yang, "Trust-enhanced cloud service selection model based on qos analysis," *PLoS ONE*, vol. 10, 2015.

[10] S. Farzai, M. H. Shirvani, and M. Rabbani, "Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters," *Sustainable Computing: Informatics and Systems*, vol. 28, 2020.

[11] W. Jing, C. Zhao, Q. Miao, H. Song, and G. Chen, "Qos-dpso: Qos-aware task scheduling for cloud computing system," *Journal of Network and Systems Management*, vol. 29, 2021.

[12] O. F. Aloufi, K. Djemame, F. Saeed, and F. Ghaban, "A survey on predicting workloads and optimizing qos in the cloud computing," in *Proceedings - 2021 International Congress of Advanced Technology and Engineering, ICOTEN 2021*. Institute of Electrical and Electronics Engineers Inc., 2021.

[13] A. Edinat, R. Al-Sayyed, and A. Hudaib, "A survey on improving qos in service level agreement for cloud computing environment," *International Journal of Interactive Mobile Technologies*, vol. 15, pp. 119–143, 2021.

[14] X. Wang and Y. Wang, "Coordinating power control and performance management for virtualized server clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 245–259, 2011.

[15] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in *Proceedings - International Conference on Distributed Computing Systems*, 2010, pp. 62–73.

[16] F. Fang and B.-B. Qu, "Multi-objective virtual machine placement for load balancing," *ITM Web of Conferences*, vol. 11, p. 01011, 2017.

[17] A. Ghasemi and A. T. Haghighat, "A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning," *Computing*, vol. 102, pp. 2049–2072, 2020.

[18] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 29, 2017.

[19] B. Sotomayor, K. Keahey, I. Foster, and T. Freeman, "Enabling cost-effective resource leases with virtual machines," in *Proceedings - 2007 ACM/IEEE International Symposium on High Performance Distributed Computing*, 2007.

[20] Y. Chen, A. Sivasubramaniam, A. Das, Q. Wang, W. Qin, and N. Gautam, "Managing server energy and operational costs in hosting centers," *Performance Evaluation Review*, vol. 33, pp. 303–314, 2005.

[21] A. S. Abohamama and E. Hamouda, "A hybrid energy–aware virtual machine placement algorithm for cloud environments," *Expert Systems with Applications*, vol. 150, 2020.

[22] C. Wei, Z. H. Hu, and Y. G. Wang, "Exact algorithms for energy-efficient virtual machine placement in data centers," *Future Generation Computer Systems*, vol. 106, pp. 77–91, 2020.

[23] S. Azizi, M. Zandsalimi, and D. Li, "An energy-efficient algorithm for virtual machine placement optimization in cloud data centers," *Cluster Computing*, vol. 23, pp. 3421–3434, 2020.

[24] H. Feng, Y. Deng, and J. Li, "A global-energy-aware virtual machine placement strategy for cloud data centers," *Journal of Systems Architecture*, vol. 116, 2021.

[25] R. B. Madhumala, H. Tiwari, and V. C. Devaraj, "Virtual machine placement using energy efficient particle swarm optimization in cloud datacenter," *Cybernetics and Information Technologies*, vol. 21, pp. 62–72, 2021.

[26] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, pp. 1–11, 2009.

[27] G. Chen, H. V. Jagadish, D. Jiang, D. Maier, B. C. Ooi, K. L. Tan, and W. C. Tan, "Federation in cloud data management: Challenges and opportunities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 1670–1678, 2014.

[28] D. C. Erdil, “Autonomic cloud resource sharing for intercloud federations,” *Future Generation Computer Systems*, vol. 29, pp. 1700–1708, 2013.

[29] F. Palmieri, L. Buonanno, S. Venticinque, R. Aversa, and B. D. Martino, “A distributed scheduling framework based on selfish autonomous agents for federated cloud environments,” *Future Generation Computer Systems*, vol. 29, pp. 1461–1472, 2013.

[30] C.-T. Fan, Y.-S. Chang, W.-J. Wang, and S.-M. Yuan, “Execution time prediction using rough set theory in hybrid cloud,” in *Proceedings - 2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*. IEEE, 2012, pp. 729–734.

[31] R. Buyya, R. Ranjan, and R. N. Calheiros, “Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities,” in *Proceedings - 2009 International Conference on High Performance Computing and Simulation*, 2009, pp. 1–11.