A Stackelberg Game Approach toward Migration of Enterprise Applications to the Cloud

Shiyong Li, Wenzhe Li *, Huan Liu and Wei Sun

School of Economics and Management, Yanshan University, Qinhuangdao 066004, China; shiyongli@ysu.edu.cn (S.L.); liuhuan5011@foxmail.com (H.L.); wsun@ysu.edu.cn (W.S.)
* Correspondence: wenzheli0910@gmail.com; Tel.: +86-3358057025

Abstract: With the development of cloud computing, more and more cloud resources are rented or purchased by users. Using an economics approach to achieve cloud resource management has been thought of as a good choice for an enterprise user to complete an application’s migration and deployment into the public cloud. During an application’s migration process, it is important but very challenging to achieve the satisfaction of both the enterprise user and the public cloud provider at the same time. In this paper, we apply an economics approach to investigate the migration optimization problem during the migration process of applications from the enterprise user’s data center to the remote public cloud. We consider the application migration time of the enterprise user and the energy consumption of physical machines, and establish a single static round optimization problem for both the enterprise user and the cloud provider on the premise of satisfying the quality of experience (QoE) based on the Stackelberg game, where the public cloud provider is leader and the enterprise user is follower. Then we propose a novel algorithm to find the optimal physical machine placement for application migration. After that, we further consider that an enterprise user needs to migrate several applications, and extend the single-round static game to the multi-round dynamic game, where the energy consumption costs of the physical machines are reduced by adjusting the states of the physical machines in each round. We finally illustrate the performance of our scheme through some simulation results.

Keywords: cloud migration; energy consumption; Stackelberg game; quality of experience (QoE)

1. Introduction

With the development of cloud computing technologies, it is a better choice for enterprises to move part or all of their applications to the cloud [1]. There are abundant resources in the public cloud, such as CPU, memory, and bandwidth, which can provide enterprises with more professional services. As shown in Figure 1, enterprise applications consist of enterprise legacy applications and enterprise innovation applications. Enterprise legacy applications are divided into enterprise core applications and enterprise non-core applications. In the past, enterprises deployed core enterprise applications, such as Enterprise Resource Planning (ERP), Software Configuration Management (SCM), and Product Lifecycle Management (PLM), in the enterprise Data Center to ensure data security. Parts of non-core enterprise applications, such as office automation (OA), were also deployed locally. But with the development of cloud computing technologies and the improvement of cloud security, enterprises now trust cloud providers and have started to migrate their applications from the local data center to the cloud. They don’t need to build or maintain their own computer rooms, which greatly reduces the operation cost of enterprises, improves their competitiveness, and increases their own income. Cloud migration needs to consider the actual system environment of the enterprise. The cloud provider tailors enterprise migration solutions to meet multiple enterprises’ needs for system and application characteristics. Unlike enterprise legacy applications, other non-core enterprise applications, such as Portal and enterprise innovation applications such as Internet of Things
(IoT), are directly implemented in the public cloud. Instead of migrating applications from the local data center to the cloud, enterprises can use innovation applications provided by the cloud provider directly. For legacy applications, public cloud providers can obtain benefits by renting or selling cloud resources to process migration tasks for enterprise users. For innovation applications, the cloud provider rents or sells entire applications to the enterprise user rather than resources. In these processes, there are also costs associated with bandwidth, energy consumption, operation, and maintenance. In addition, as more and more enterprises migrate their applications to the public cloud, the public cloud provider needs larger data centers to process tasks and store data for enterprises. One of the main challenges of the public cloud data center is the large energy consumption of physical machines and cooling equipment, which is also a main consideration for achieving environmental greenness.

![Diagram](https://example.com/diagram.png)

**Figure 1.** Recommendations for enterprise application migration to the cloud.

Therefore, when an enterprise migrates its applications into the cloud, an important goal is to reduce the energy consumption of the physical machines as much as possible while guaranteeing the service quality requirements of the enterprise applications, so as to achieve benefit maximization for the cloud provider [2]. However, migrating enterprise applications to the cloud requires interaction between the cloud provider and the enterprise user. It is necessary to consider the benefits of the enterprise user and formulate a migration mechanism to meet the needs of both parties. In this study, we investigate how to rationally allocate physical machines to process applications migration in order to satisfy the interests of both the cloud provider and the enterprise user. We present a two-party Stackelberg game to obtain a satisfactory solution for both parties in a single migration process based on the energy consumption of the cloud provider and the application processing time of the enterprise user. Then, we also consider the whole application migration process and the energy consumption of the idle physical machines, and propose a dynamic physical machine allocation mechanism for migrating enterprise applications into the cloud. Specifically, we extend the conflict between the energy consumption of the physical machines and the application migration processing time of the enterprise applications to the competition between the cloud provider and the enterprise user, and then use the repeated Stackelberg game to model the interaction processes. The optimal resource allocation solution is obtained by solving the game equilibrium.

The rest of this article is organized as follows: Section 2 introduces the related work on cloud migration, Section 3 formulates the system model using the Stackelberg game,
Section 4 presents a resource allocation scheme for achieving cloud migration, Section 5 presents some numerical examples to illustrate the performance of the proposed scheme, and Section 6 concludes this article.

2. Related Work

2.1. Cloud Migration

Cloud migration requires the interaction between the enterprise users that need cloud resources to support their applications and the cloud provider that offers resources to users. The cloud provider has tradable resources, such as storage, computing, etc. Therefore, the cloud migration issue can be considered from two perspectives. On the one hand, a reasonable cloud migration process can improve the satisfaction of an enterprise user. For example, Li et al. [1] investigated the utility maximization for migrating enterprise applications into the cloud and proposed a novel migration scheme for achieving optimal cloud resource allocation. Huang et al. [3] proposed an automatic smart service migration outline through the ant colony optimization (ACO) algorithm on cloud-oriented e-commerce. In the presented model, they use the ACO algorithm to take the finest (near-optimal) service migration decisions. Aslam et al. [4] used the K-Means algorithm of the partitioning clustering method to analyze cloud migration strategies and helped consumers to make appropriate migration strategies by studying existing cloud provision strategies.

On the other hand, the interests of the cloud provider should not be ignored. Operation cost, transmission cost and time, delay, and the price of cloud resources directly or indirectly affect the interests of the cloud provider. For example, Ellison et al. [5] introduced a two-stage approach which accurately estimates the migration cost, migration duration, and cloud running cost of relational databases. Badshah et al. [6] proposed a mechanism for external cloud service providers to make efficient migration decisions to minimize the delay, running time, and transformation cost by searching an optimum data center, where the resources may be taken with optimum conditions. Some scholars have considered the interests of both the user and the cloud provider. For example, Wang [7] designed a cloud cooperative migration strategy based on the information exchange structure of the industrial Internet of Things and the delay mechanism caused by the migration.

Cloud migration is also used in enterprise disaster recovery and backup. Li et al. [8] proposed a dynamic multi-objective optimized replica placement and migration strategy for SaaS applications in edge cloud and designed a fast, non-dominated sorting genetic algorithm to solve the multi-objective optimized replica placement problem. Mseddi et al. [9] proposed an efficient replica migration scheme for distributed cloud storage systems, CRANE, which complements any replica placement algorithm by efficiently managing replica creation in geo-distributed infrastructures. In addition, enhanced learning has also been applied to optimize cloud migration problems. Ren et al. [10] proposed an evolutionary framework based on deep reinforcement learning for virtual service migration in large-scale mobile cloud centers. Li et al. [11] proposed a video stream session migration method based on deep reinforcement learning in a cloud computing environment to balance the cost and load of migration. Cloud migration has been realized for enterprise applications in life, but users may not have a clear understanding of cloud migration. Some scholars have proposed applicable cloud migration steps and frameworks from a more practical perspective. Nabavi et al. [12] proposed a plan-oriented migration approach by which the enterprise management is able to follow the migration steps of a valid plan. Bazi et al. [13] used a meta-synthesis method for the first time for the analysis and synthesis of previously published work and suggested a comprehensive cloud migration framework.

2.2. Stackelberg Game

The Stackelberg game is an economic theory in which the leader has a leadership advantage and can take the lead or advantageous position in the game. The followers should follow the leader to make the game work. There is an equilibrium solution to the Stackelberg game. Regardless of the leader’s chosen strategy, the best response set
for followers is non-empty. In this sub-section, we summarize research that uses the Stackelberg game to solve the cloud resource allocation problem. The Stackelberg game is used in pricing strategy for cloud resources. Chen et al. [14] jointly considered the principal resources with the goal of minimizing the total cost, then proposed a multi-resource combinatorial pricing mechanism to avoid the shortcomings of pricing mechanisms in heterogeneous cloud environments that consider only a single resource. Liu et al. [15] examined the complex interactions between users, online service providers and public cloud providers, and analyzed resource allocation in private clouds and pricing strategies in public clouds. Cardellini et al. [16] considered Software as a Service (SaaS) providers that use cloud resources provided by an Infrastructure as a Service (IaaS) provider to provide services and pay for them, and then studied the virtual machine provisioning and spot pricing strategies. Tang et al. [17] separately studied the problem of joint pricing and capacity planning in the IaaS provider market with a set of SaaS providers in monopolistic IaaS provider markets and multi-IaaS provider markets. Zhang et al. [18] considered communication and computational costs and built a multi-stage buyer-seller game to maximize the individual utility of mobile users or base stations for ad hoc mobile cloud computing in the software-defined mobile networks (SDMNs). In addition, there is also competition between cloud providers. Taghavi et al. [19] proposed the IaaS cloud providers had to compete not only over price, but also quality to make the cloud market open and increase cloud service demand.

Cloud providers and users are also concerned about the revenue of cloud computing besides the price of cloud resources. Zhu et al. [20] proposed a dynamic pricing mechanism to solve the conflict in maximizing revenue of both IaaS and SaaS providers and simultaneously consider the impact of resource price on users’ willingness to access service. Liu et al. [21] designed a service mechanism from a game theoretical perspective to achieve profit optimization for the cloud provider and its multiple users. Zhang et al. [22] formulated a hierarchical game to study the resource allocation problem of data center network systems when multiple data center operators (DCOs) served multiple service subscribers (SSs) at the same time. Qiu et al. [23] considered the dynamic arrival of user jobs and the unstable availability of spare resources on the private cloud, and modeled the interaction between the broker and the private clouds as a two-stage Stackelberg game that maximized the economic benefits for both the broker and the private cloud. Wei et al. [24] proposed a resource allocation model that guaranteed optimal gains for infrastructure providers based on the unit prices of different types of resources. Wang et al. [25] proposed a two-stage Stackelberg game model based on multi-leader/multi-follower to maximize the utility function of the MCC network. While focusing on benefits, energy awareness is also a critical problem in cloud computing. Fernández-Cerero et al. [26] defined a Stackelberg game based on the cloud scheduling model with the workload scheduler and energy-efficiency agent as the main players which achieved a fair balance between low energy consumption and short queuing and makespan. Now there are many case studies using the Stackelberg game to solve cloud computing issues in multimedia content dissemination. For example, Li et al. [27] proposed a multimedia cloud collaborative content dissemination method based on game theory to improve the user experience with weak system capability, reduce the average time of content dissemination, and reduce the total cost to multimedia cloud users. Chakareski [28] considered two expected P2P-Cloud interaction scenarios that can improve the performance of the reference P2P broadcast system and maximize the targeted functionality of the cloud and the clients. Cao et al. [29] proposed a new multimedia service optimization mechanism to ensure service reliability while meeting users’ needs for economy and security of services in hybrid cloud networks.

With the development of cloud computing technologies, the energy consumption of cloud data centers has become a critical concern to the whole world. Therefore, how to reduce the energy consumption of the cloud provider while achieving user’s satisfaction has become an important problem for the cloud provider.
3. System Model

In this section, we propose an energy-saving computing resource dynamic decision mechanism based on the idea of real-time adjustment of the physical machine’s operating state. With the objective of minimizing the overall energy consumption cost, the cloud provider dynamically adjusts the state of the physical machines and the enterprise user chooses a suitable physical machine for application migration with minimum application processing time. The cloud migration architecture is shown in Figure 2, including an enterprise user and its applications, a cloud migration manager, and several cloud physical machines. The cloud migration manager plays an intermediary role between the enterprise user and the cloud physical machines. It receives application migration requests from the enterprise user and adjusts the state of the physical machines of the cloud provider.

![Figure 2. Basic framework of enterprise application migration to the cloud.](image)

When the enterprise applications are migrated to the cloud, the enterprise user needs to determine the migration sequence in advance to prevent the applications from being unable to operate in time after the migration because the basic applications have not yet wholly been migrated. The cloud migration manager is responsible for the scheduling of physical machines and offers a set of available physical machines to the enterprise user. Then, the enterprise user chooses an appropriate physical machine that takes the shortest time to process tasks.

This paper establishes a Stackelberg game model between the cloud provider and the enterprise user. As a leader, the cloud provider gives an operating state strategy for physical machines. As a follower, the enterprise user chooses an appropriate physical machine to process the migration application. Since the migrated applications are multiple and non-simultaneous, the migration time is divided into several time periods. One round of the game between the enterprise user and the cloud provider is processed in each period. During each round, one application is migrated to the cloud and processed by only one physical machine. Due to the different physical machine running speeds, the processing time is different for each application. After the physical machine completes an application task, it becomes idle and can be allocated again to a new task at the beginning of the next round.

We denote the sets of physical machines of the cloud provider and applications of the enterprise user by $M$ and $N$, respectively. The corresponding numbers of physical machines and migration applications are $m$ and $n$. Define $i$ and $j$ as the physical machine index and migration application index. Assuming that applications are migrated at the start of each round, the number of rounds is equal to the number of migrated enterprise applications. The cloud provider offers an application migration strategy $X_j = \{X_{1j}, X_{2j}, X_{3j}, \ldots, X_{mj}\}$ for the enterprise user, where $X_{ij} = 1$ represents the physical machine $i$ is on and $X_{ij} = 0$ represents the physical machine $i$ is off when processing migration application $j$. The enterprise user gives a selection strategy $Y_j = \{Y_{1j}, Y_{2j}, Y_{3j}, \ldots, Y_{mj}\}$ in each round, where $Y_{ij} = 1$ represents the enterprise user selects the physical machine $i$ for migration application $j$ and $Y_{ij} = 0$ represents the enterprise user does not choose the physical machine $i$. If $X_{ij} = 0$, there should be $Y_{ij} = 0$. That is, the enterprise user should select a suitable physical machine to process the task among the physical machines in the “on” state provided by the cloud provider.
The notations are summarized in Table 1.

Table 1. Summary of notations.

| Notations | Definition |
|-----------|------------|
| $M$       | physical machines of the cloud provider |
| $N$       | migration applications of the enterprise user |
| $X_{ij}$  | strategy of the cloud provider |
| $Y_{ij}$  | strategy of the enterprise user |
| $T_{ij}$  | processing time of migration application $j$ by physical machine $i$ |
| $P_{ij}$  | energy consumption of physical machine $i$ to process application $j$ |
| $c_{ij}$  | energy consumption cost of physical machine $i$ to process application $j$ |
| $v_i$     | capacity of physical machine $i$ (For example: 1 TB, 1 GB) |
| $p_{idle}$| idle power of physical machine $i$ |
| $p_{peak}$| peak power of physical machine $i$ |
| $v_i$     | processing rate of physical machine $i$ |
| $c_j$     | total energy consumption cost in round $j$ |
| $C_j$     | energy consumption budget of cloud provider in round $j$ |
| $t_j$     | start time in round $j$ |
| $h_j$     | the workload of each migration application $j$ (For example: 1 TB, 1 GB) |
| $T_j$     | actual processing time of migration application $j$ |
| $R_j$     | the number of rounds occupied by enterprise application $j$ |
| $\sigma$ | time duration of each round |
| $\kappa, \phi$ | transformation factors |
| $\tau$   | physical machine idle endurance round |
| $\theta$ | quantity limitation of physical machine |

3.1. Application Processing Time of the Enterprise User

The process of application migration to the cloud can impact the operation of the enterprise. Specifically, some functions of migrated applications cannot be efficiently used by enterprise employees during cloud migration and customers of the enterprise may experience delayed access to company website. These may impact the benefits of enterprise. So migration time is a crucial factor for enterprise when migrating applications from local data center to the cloud. The migration time impacts the QoE of the enterprise user directly. In this subsection, we only choose the processing time of the migration applications to determine the QoE of the enterprise user. Application processing time is an important part of total application migration time. The enterprise user expects the shortest migration time for enterprise operation. The enterprise user selects a physical machine for each application to complete the migration task according to the available physical machine strategy given by the cloud provider. Assume the enterprise user chooses physical machine $i$ to process the migration application $j$. Hence, the processing time of the migration application $j$ by physical machine $i$ is formulated as:

$$T_{ij} = \frac{h_j}{v_i}$$  \hspace{1cm} (1)

where $h_j$ denotes the workload of the enterprise application $j$ and $v_i$ represents the processing speed of the physical machine $i$.

Finally, the enterprise user chooses an available physical machine $i$ provided by the cloud provider for application $j$, the actual processing time of application $j$ is formulated as:

$$T_j = \max_{i \in \mathcal{M}} \{ T_{ij} | X_{ij} | X_{ij} \in X^+_j \}, \forall j \in N$$  \hspace{1cm} (2)

3.2. Energy Consumption Cost to the Cloud Provider

The goal of the cloud provider is to minimize the energy consumption cost of physical machines. In this case, only one physical machine will be kept on and the others will be off. But for the enterprise user, when only one physical machine is turned on, it is difficult to derive their own optimal strategy based on the cloud provider’s strategy since
the enterprise user has no other choice at this time, resulting in the reduction of user’s satisfaction. Therefore, we assume that the cloud provider does not need to minimize its energy consumption. Still, the cloud provider’s physical machines need to meet their energy consumption budget. That is, if the energy consumption of the physical machine meets the budget, the cloud provider allows the user to use this physical machine to process its application migration.

In the case of a certain workload of application migration, a common server power function was studied by Google [30], as follows:

$$P_{ij} = \xi_i \times P_{idle}^i + \left( P_{peak}^i - P_{idle}^i \right) \times h_j$$

(3)

where $\xi_i$ represents the capacity of physical machine $i$ to support the workload; and $P_{idle}^i$ and $P_{peak}^i$ denote the idle and peak power associated with the physical machine $i$, respectively.

Hence, the cost of the physical machine $i$ for application $j$ is formulated as [31]

$$c_{ij} = \phi \times P_{ij} \times T_{ij}$$

(4)

And the total energy consumption cost for processing the migration of application $j$ can be formulated as:

$$c_j = \sum_{i=1}^{m} c_{ij}, \forall j \in N$$

(5)

where $\phi$ is defined as a factor to transform physical machine power into cost, and $T_{ij}$ is the processing time of migration application $j$ with physical machine $i$. For the migration of each application, the cloud provider needs to adjust the state of the physical machine in real time to achieve the optimization goal of the mechanism.

4. Energy Consumption Optimization: A Stackelberg Game Approach

In this section, we will present a specific cloud migration scheme based on the Stackelberg game. First, we will give a simple scheme with a single round of static requests. Then, we consider different volumes of migration tasks, and the cost of maintaining and restarting the physical machines, and further extend the original scheme to multi-round of dynamic requests

4.1. Single-Round of Static Cloud Migration Scheme
4.1.1. Problem Statement

Given the current migration workload of the enterprise user, a single period static decision mechanism of the cloud provider can be formulated as:

$$\max \sum_{i=1}^{m} (C_j - c_{ij}) X_{ij}, \forall j \in N$$

(6)

s.t. $1 \leq \sum_{i=1}^{m} X_{ij} \leq \theta, \forall j \in N$

(7)

$$X_{ij}(C_j - c_{ij}) \geq 0$$

(8)

$$X_{ij} \in \{0, 1\}$$

(9)

Here, the objective (6) is to maximize the energy utility of the cloud provider. This problem considers that the cloud provider turns on a certain number of physical machines that are within the energy budget, so the sum of the difference between the energy consumption cost of the physical machines and the budget is used as the objective function, and the cloud provider’s strategy is to select the appropriate set of physical machines that are within the budget and maximize the difference between the energy consumption cost.
and the budget. Constraint (7) means that there is at least one physical machine to perform the migration task for the enterprise user. For the cloud provider, the number of available physical machines that the cloud provider offers to the enterprise user should be reasonable and limited. Constraint (8) means the energy consumption of the physical machine should be less than the cloud provider’s energy budget when it is on. Constraint (9) shows that $X_{ij}$ is a binary variable.

Correspondingly, the optimal processing time of the enterprise user $j$ can be formulated as:

$$
\min \sum_{i=1}^{m} T_{ij} Y_{ij}, \forall j \in N \tag{10}
$$

subject to:

$$
\sum_{i=1}^{m} Y_{ij} = 1, \forall j \in N \tag{11}
$$

$$
0 < Y_{ij} < X_{ij} \tag{12}
$$

$$
Y_{ij} \in \{0, 1\} \tag{13}
$$

Here, the objective (10) minimizes the migration time of the enterprise user. Constraint (11) ensures that only one physical machine can be selected by the enterprise user at a time to process the migration task. Constraint (12) shows that the enterprise user’s strategy relies on the cloud provider’s strategy. Constraint (13) shows that $Y_{ij}$ is a binary variable.

Given the Stackelberg game between the cloud provider and the enterprise user, the equilibrium of this game can be solved by using the backpropagation solution method. In the second stage, given the cloud provider’s strategy $X_{j}$, the enterprise user, as the follower, needs to choose the physical machine that minimizes its migration time among the available physical machines for application migration and then respond with strategy $Y_{j}$. In the first stage, the cloud provider gives the set of physical machines that can process the migration task.

4.1.2. Analysis for Single-Round of Static Cloud Migration Problem

The energy consumption cost of physical machine $i$ to process migration task $j$ can be formulated as:

$$
c_{ij} = \varphi \left[ \xi_i P_{i}^{idle} + \kappa \left( P_{i}^{peak} - P_{i}^{idle} \right) h_j \right] T_{ij} \tag{14}
$$

The physical machine becomes idle when an application task is completed. The processing time $T_{ij}$ of the application migration is determined by the application migration workload $h_j$ of the enterprise user and the processing rate $v_i$ of the physical machine, that is,

$$
c_{ij} = \varphi \left[ \xi_i P_{i}^{idle} + \kappa \left( P_{i}^{peak} - P_{i}^{idle} \right) h_j \right] \frac{h_j}{v_i} \tag{15}
$$

The energy consumption cost $c_{ij}$ of the physical machine $i$ is determined by the idle power $P_{i}^{idle}$, peak power $P_{i}^{peak}$, processing rate $v_i$ of the physical machine, and the application workload $h_j$ of the enterprise user.

**Proposition 1.** The single period static decision mechanism of the cloud provider model is an Integer Linear Programming problem. The energy consumption cost function of the physical machine is a convex function on the migration application workload.

In order to study the relationship between the energy consumption cost of the physical machine and the workload of the enterprise user, we need to determine $\frac{\partial^2 c_{ij}}{\partial h_j^2}$.

Firstly, the first-order derivative is as follows:

$$
\frac{\partial c_{ij}}{\partial h_j} = \frac{\varphi}{v_i} \left[ \xi_i P_{i}^{idle} + 2\kappa h_j \left( P_{i}^{peak} - P_{i}^{idle} \right) \right] \tag{16}
$$
Obviously, \( c_{ij} \) is a monotonically increasing function of \( h_j \) in the interval. The second-order derivative is as follows:

\[
\frac{\partial^2 c_{ij}}{\partial h_j^2} = 2\phi k \frac{P_{\text{peak}} - P_{\text{idle}}}{v_i} > 0
\]

(17)

Obviously, \( \frac{\partial c_{ij}}{\partial h_j} \) is a convex function on \( h_j \) and is monotonically increasing on the interval. The larger the value of \( \frac{\partial^2 c_{ij}}{\partial h_j^2} \), the larger the slope of \( \frac{\partial c_{ij}}{\partial h_j} \), and the more energy consumption cost generated by the physical machine for every unit increase in the migration application’s workload.

Proposition 2. Given a certain workload \( h_j \), the inherent properties (i.e., idle power, peak power, and processing rate) of physical machines determine the order in which they are used by the cloud migration manager.

\[
\frac{\partial c_{ij}}{\partial P_{\text{peak}}^i} = \frac{\phi k}{v_i} h_j^2 \geq 0
\]

(18)

\[
\frac{\partial c_{ij}}{\partial P_{\text{idle}}^i} = \frac{\phi h_j}{v_i} (\xi_i - \kappa h_j)
\]

(19)

\[
\frac{\partial c_{ij}}{\partial v_i} = -\phi \left[ \xi_i P_{\text{idle}}^i + \kappa (P_{\text{peak}}^i - P_{\text{idle}}^i) h_j \right] \frac{h_j}{v_i^2} < 0
\]

(20)

If the inherent properties (i.e., idle power, peak power, and processing rate) of physical machines are known for a given enterprise application workload, the order in which the cloud migration manager calls the physical machines is determined.

Proposition 3. The optimal number of physical machines provided to the enterprise user in the cloud provider’s strategy is

\[
\min \{ \theta, |\arg\{ X_{ij}(C_j - c_{ij}) \geq 0 \} | \}
\]

The physical machines provided by the cloud provider are subject to the quantity constraints of the physical machines and the energy constraints set by the cloud provider. As the energy consumption cost function of the physical machine is a monotonically increasing convex function, an increase in the number of physical machines turned on leads to an increase in the total energy consumption cost of the cloud provider. As such, the cloud provider limits the number of physical machines that can be turned on. At the same time, however, the cloud provider also limits the energy consumption budget of physical machines, so the cloud provider’s strategy is bounded by

\[
\min \{ \theta, |\arg\{ X_{ij}(C_j - c_{ij}) \geq 0 \} | \}
\]

The optimization problem for the cloud provider can be described in Algorithm 1.

The cloud provider calculates the energy consumption cost of each physical machine when it processes the application migration task. Based on the energy consumption budget, the cloud provider determines whether the current application migration task can be performed on this physical machine. If it is below the energy consumption budget, the cloud provider can allow the enterprise user to select this physical machine to process the task. If the energy consumption cost is too high, it is unacceptable for the cloud provider to turn on this physical machine to provide services for the enterprise user, even if this physical machine is appropriate for the user. The cloud provider evaluates the energy consumption of each physical machine in turn and gives its final decision strategy \( X_f^* \).
Algorithm 1 Single-round static strategy algorithm

Input: Energy consumption $c_{ij}$ of physical machine; Quantity limitation $\theta$ of the cloud provider; Application processing time $v_j$
Output: cloud provider’s strategy $X^*_j$

1. For $i \in M$ do
2. Calculate the energy consumption cost of physical machine $i$ to process migration application $j$
3. sort $c_{ij}$ in ascending order
4. Select the first $\theta$ physical machines
5. Let $X_i = 1$
6. else
7. Let $X_i = 0$
8. End For

4.2. Multi-Round of Predictably Dynamic Migration Scheme

In this subsection, we consider a multi-round dynamic strategy mechanism. Multi-round predictability means dividing the service time into equal time slots, with one and only one enterprise application to be migrated at each time slot. It is assumed that the cloud provider knows all the information of the enterprise user’s application migration, including the workload of the application migration, the migration sequence, and so on. Under this assumption, with the goal of minimizing the total energy consumption cost of the whole migration, the cloud provider adjusts the state of the physical machines based on the demand at each round. For each round, different physical machines have the different time to process different migration tasks, which means that $T_{ij}$ is not necessarily equal to $\sigma$. During the time $T_{ij}$, if the physical machine $i$ processes the application migration task $j$, then it cannot process other application migration tasks. Therefore, in round $j$, if physical machine $i$ is occupied, the cloud provider’s strategy should be $X_{ij} = 0$.

The number of rounds occupied by the migration application $j$ can be formulated by the ceiling function as follows:

$$R_j = T_j = \inf \{ z \in \mathbb{Z} | z \geq T_j \}$$

that is, the physical machine that processes the migration task at round $j$ can further process the other task again at round $R_j + j$.

In practice, besides the cost used to process the application’s migration for the enterprise user, the loss associated with wear and tear resulting from the daily operation of the physical machines cannot be ignored. Especially the loss caused by changing the state of the physical machines may trigger more cost than that under the on-idle state. Based on this principle, we assume that the energy consumption cost of physical machine restart is larger than the energy consumption cost generated by $\tau$ rounds but less than the energy consumption cost of $(\tau + 1)$ rounds, where $\tau$ is the idle endurance rounds of the physical machine. If it is not used for the next $\tau$ rounds, the physical machine will be shut down. Otherwise, it will remain on, as shown in Figure 3.

In summary, the cloud provider excludes physical machines that are still processing tasks from previous rounds at first. Then, for other idle physical machines, the cloud provider’s strategy is obtained by using Algorithm 1, and the physical machine is assigned to process the application migration task after the enterprise user makes its strategy. Note that the cloud provider’s strategy is the optional set of physical machines. These physical machines are meeting their energy consumption budgets, and regardless of which one the enterprise user chooses, the energy consumption cost for processing tasks is acceptable for the cloud provider, whereas the enterprise user can only select one physical machine to process the task and responds to the cloud provider with its strategy. There is only one physical machine to serve the enterprise user, and the other physical machines within
the budget will remain in their original state. In the next stage, the cloud provider will re-evaluate the physical machine energy consumption. After getting the strategy for each round, the physical machine state is adjusted according to the idle endurance time $\tau$. The multi-round predictable dynamic strategy is shown in Algorithm 2.

![Flow chart of multi-round of physical machine state.](image)

**Algorithm 2 Multi-round predictable dynamic strategy algorithm**

| Input: idle duration time $\tau$ of physical machine  |
|-----------------------------------------------------|
| Single round static strategy of cloud provider      |

Output: Multi-round predictable dynamic strategy for cloud provider

1. For $j = 1, 2, 3, \ldots, n$ do
2.     For $i = 1, 2, 3, \ldots, m$ do
3.         The physical machine $i$ processes the previous migration task
4.         Let: $X_{ij} = 0$
5.     End for
6.     Get each round decision through Algorithm 1
7.     End for
8. For
9.     Use the process as shown in Figure 3 to adjust the physical machine state in each round
10. End for

**4.3. Cloud Migration Implementation and Architecture**

The implementation and architecture of migration from the enterprise into the cloud are illustrated in Figure 4. The architecture is composed of the enterprise user and the cloud provider. The enterprise user has several migration applications, such as the Web server and the Management server. The cloud provider consists of a cloud migration manager and several physical machines.

**4.3.1. The Enterprise User**

With the development of an enterprise, its local data center can no longer handle all of the data. Therefore, the enterprise chooses to migrate some applications to the cloud. The managers of the enterprise should not choose applications to be migrated to the cloud at random. The applications to be migrated are decided by enterprise managers based on several factors, such as the size of the enterprise, the importance of the application, and the migration complexity. Before an enterprise migrates its applications to the cloud, it selects the applications that need to be migrated to the cloud first and then decides the migration order of these applications. After that, the enterprise tells the cloud provider all of the migration information.
4.3.2. Cloud Migration Manager

The cloud migration manager is responsible for assigning each enterprise application that is being migrated to the appropriate physical machine by Algorithm 1. As the enterprise user sends all of the migration information for the applications to the cloud provider, the cloud migration manager calculates the energy consumption cost of each application by each physical machine based on the workload of the migration application firstly. These energy consumptions are then sorted incrementally. According to the constraints on the number of physical machines provided by the cloud provider, the cloud migration manager selects the appropriate set of physical machines. Finally, a physical machine with the lowest power consumption is selected to perform the migration task while meeting the goal of minimizing the migration time for enterprise user.

4.3.3. Physical Machine

After each application is migrated by the proper physical machine, the cloud provider dynamically adjusts the state of all physical machines by Algorithm 2. The cloud provider turns off the unwanted physical machines and turns on the physical machines that need to perform migration tasks in order to save energy.

5. Simulation Results

In this section, some simulations are conducted to illustrate the rationality of the model and to show the effectiveness of the proposed algorithm. We first consider a single-round migration from the enterprise user to the cloud and study the impact of migration time and the migration application workload on physical machines’ energy consumption cost. Then, we extend the single-round game into multi-round games and give a specific scenario for the optimal decision strategy for the state of the physical machine. We applied LINGO 11.0 to write the programs and drew the figures with MATLAB 2021a. All computations are run on a machine with 2.3 GHz Intel Core i5-6200U with 4 GB memory.

5.1. Single-Round Cloud Migration Scenario

This subsection will verify the rationality of the single-round static decision mechanism proposed in the paper through simulation experiments. The simulation experiment is set up with five physical machines, containing a series of physical machines with the same CPU but different capacity and performance, as shown in Table 2. The parameters
of physical machines shown in the table were obtained from the official website of the Standard Performance Evaluation Corporation.

Table 2. Physical machine parameters.

| Physical Machine | \( \xi_i \) | \( v_i \) | \( p_{idle}^i \) | \( p_{peak}^i \) |
|------------------|----------|--------|----------------|----------------|
| 1                | 5        | 82     | 430            | 1731           |
| 2                | 5        | 10     | 331            | 1192           |
| 3                | 5        | 57     | 352            | 1694           |
| 4                | 5        | 14     | 257            | 1173           |
| 5                | 5        | 35     | 523            | 2260           |

In this paper, we consider an enterprise user who intends to migrate its three applications to the cloud, as shown in Table 3.

Table 3. Enterprise application parameters.

| Enterprise Application | \( h_j \) | \( C_j \) |
|------------------------|----------|----------|
| 1                      | 1        | 150      |
| 2                      | 4        | 1500     |
| 3                      | 2        | 300      |

The values of notations involved in this paper are shown in Table 4.

Table 4. Values of notations.

| Notations | Value |
|-----------|-------|
| \( \varphi \) | 1     |
| \( \tau \)   | 1     |
| \( \sigma \) | 0.02  |

5.1.1. The Impact of the Application Processing Time on Cloud Provider’s Strategy

First, we investigate a scenario in which the cloud provider gives its strategy according to energy consumption of the physical machines, that is, does not consider the application processing time of the enterprise user. Figure 5 shows the energy consumption of physical machines when processing the same application. The energy consumption \( P_{ij} \) is a linear function about the application migration workload \( h_j \) of an enterprise user, and the more the application workload, the more the energy consumption of the physical machine. PM4 has the lowest energy consumption, while PM5 has the highest energy consumption. Compared with Figure 5, in Figure 6 we consider that the cloud provider makes its strategy according to the energy consumption cost of the physical machines. In this scenario, the physical machines have different energy consumption costs when processing the same application because of their different processing rate. Therefore, the strategy of the cloud provider will be different from the previous strategy. Each application being migrated by the enterprise user needs to be processed by one physical machine, so the runtime of the physical machine is proportional to the workload of the application and inversely proportional to the processing speed of the physical machine. The energy consumption cost of the physical machine is a quadratic function about the enterprise application workload, and the larger the application workload, the larger the energy consumption cost of the physical machine. PM1 has the lowest energy consumption cost, while PM2 has the highest energy consumption cost. Therefore, the cloud provider should consider the migration time when providing the migration strategy. There is a fixed sequence for the energy consumption cost of the physical machines regardless of the workload, so the cloud provider has a fixed order of selecting physical machines when determining its strategy.
5.1.2. Impact of Inherent Properties of Physical Machines on Cloud Provider Strategy

At a given workload, the inherent properties (i.e., idle power, peak power, and processing rate) of physical machines determine the order in which they are used by the cloud migration manager. According to the control variable method, the energy consumption cost of PM1 is used as an example to explore the effects of idle power, peak power, and processing time on energy consumption cost, respectively. Figure 7 shows that the idle power of the physical machine is positively correlated with the energy consumption cost. Figure 8 shows that the peak power of the physical machine is positively correlated with energy consumption cost. Figure 9 shows that the processing time of the physical machine is negatively correlated with energy consumption cost.

5.1.3. The Impact of $\theta$ on Energy Consumption Cost

Figure 10 shows the energy consumption cost of each physical machine while processing migration application 1. The energy consumption threshold of the physical machine was set to 150. The energy consumption cost of two physical machines, PM1 and PM3, are both much smaller than the threshold, while the energy consumption cost of PM5 is also smaller than the threshold. Meanwhile, the energy consumption cost of PM2 is much larger than the threshold, and the energy consumption cost of PM4 is slightly larger than the threshold. Since the energy consumption cost of the selected physical machines should be no larger than the threshold, only PM1, PM3, and PM5 can be selected. This is why we have the same cumulative energy consumption cost in Figure 11 when the cloud provider selects to turn on no less than three physical machines.
When $\theta = 3$, the cloud provider’s strategy is $X_1^* = (1, 0, 1, 0, 1)$ and the enterprise user can choose one physical machine from PM1, PM3, and PM5 to process the application migration task. The enterprise user needs the shortest application processing time, which is inversely proportional to the processing speed of the physical machines, so the enterprise user chooses PM1 to process the migration task, which means that the enterprise user’s strategy is $Y_1^* = (1, 0, 0, 0, 0)$.
5.2. Multiple Rounds Cloud Migration Scenario

In this subsection we extend the single round simulation into multiple rounds, where one application needs to be migrated at the beginning of each round and applications are migrated in rounds 1, 2, and 3 for application 1, application 2, and application 3, respectively. The workload of application 1 is only 1. Using $T_{ij} = \frac{h_j}{v_i}$ can calculate the migration time 0.012 for application 1. The migration task can be completed in the first round, and PM1 becomes idle in the second round. In the second round, PM1 is still selected to process the migration task because it has the lowest energy consumption. The workload of application 2 is 4 and the processing time is 0.049. PM1 will be occupied with three rounds. When processing the third application, PM1 is occupied and PM3 has the least energy consumption among the remaining physical machines, so PM3 is chosen to process application 3, and the processing time is 0.035. PM3 will be occupied for two rounds. Therefore, PM2, PM4, and PM5 can be turned off during task migration since they have been unused. The idle tolerance time $\tau$ is 1. That is, if a physical machine is not used at this round and is again not used in the next round, it should be shut down instead of remaining on. PM3 is not used in the first two rounds, so it should be turned off during the first two rounds, and restart to process the migration tasks in the third round. The specific migration scheme is shown in Figure 12.

To evaluate the effectiveness of Algorithm 1, we use LINGO, as shown in Table 5. The results obtained by Algorithm 1 are equal to those obtained by LINGO, thus it is obvious that Algorithm 1 is effective.
### Table 5. Comparison of Algorithm 1 and LINGO.

| Strategy     | 1st Round  | 2nd Round  | 3rd Round  |
|--------------|------------|------------|------------|
| Algorithm 1  | (1,0,1,0,1)| (1,0,1,0,1)| (0,0,1,0,0)|
| LINGO        | (1,0,1,0,1)| (1,0,1,0,1)| (0,0,1,0,0) |

6. Conclusions

In this paper, we apply the Stackelberg game to investigate the migration optimization problem during application migration processes from enterprise users’ data centers to the remote cloud. First we establish a single static round optimization problem wherein the cloud provider is leader and the enterprise user is follower, and propose a novel algorithm to find the optimal placement of the physical machines. Then, considering the enterprise user needs to migrate several applications, the single-round static game is extended into a multi-round static game, where the cloud provider dynamically adjusts the state of the physical machines at each round to achieve the reduction of energy consumption costs of the physical machines and also realize the satisfaction of the enterprise user. Numerical simulations are finally proposed to evaluate the performance of the novel algorithm.

In future research, we will consider a many-to-many competition between the cloud providers and the enterprise users. Additionally, beyond considering the operating cost of physical machines, other costs of the cloud data center, such as cooling costs and maintenance costs, also impact the operation of the cloud provider, and could be the subject of future research.

**Author Contributions:** S.L., W.L., H.L. and W.S. have contributed equally to the development of this model and algorithm and its writing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (71671159, 71971188), the Natural Science Foundation of Hebei Province (G2020203005), and the China Postdoctoral Science Foundation (2016M600196).

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the anonymous reviewers and Associate Editor for very detailed and helpful comments and suggestions to improve this work.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Li, S.; Sun, W. Utility maximisation for resource allocation of migrating enterprise applications into the cloud. *Enterp. Inf. Syst. 2021, 15*, 197–229. [CrossRef]
2. Li, S.; Zhang, Y.; Sun, W. Optimal Resource Allocation Model and Algorithm for Elastic Enterprise Applications Migration to the Cloud. *Mathematics 2019, 7*, 909. [CrossRef]
3. Huang, L.; Abnoosian, K. A new approach for service migration in cloud-based e-commerce using an optimization algorithm. *Int. J. Commun. Syst. 2020, 33*, e4457. [CrossRef]
4. Aslam, M.; Rahim, L.; Watada, J.; Hashmani, M. Clustering-based cloud migration strategies. *J. Adv. Comput. Intell. Intell. Inform.* 2018, 22, 295–305. [CrossRef]

5. Ellison, M.; Calinescu, R.; Paige, R.F. Evaluating cloud database migration options using workload models. *J. Cloud Comput.* 2018, 7, 6. [CrossRef]

6. Badshah, A.; Ghani, A.; Irshad, A.; Naqvi, H.; Kumari, S. Smart workload migration on external cloud service providers to minimize delay, running time, and transfer cost. *Int. J. Commun. Syst.* 2021, 34, e4680. [CrossRef]

7. Wang, K. Migration strategy of cloud collaborative computing for delay-sensitive industrial IoT applications in the context of intelligent manufacturing. *Comput. Commun.* 2020, 150, 413–420. [CrossRef]

8. Li, C.; Wang, Y.; Tang, H.; Luo, Y. Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud. *Future Gener. Comput. Syst.* 2019, 100, 921–937. [CrossRef]

9. Mseddi, A.; Salahuddin, M.A.; Zhan, M.F.; Elbiaze, H.; Glitho, R.H. Efficient Replica Migration Scheme for Distributed Cloud Storage Systems. *IEEE Trans. Cloud Comput.* 2021, 9, 155–167. [CrossRef]

10. Ren, H.; Wang, Y.; Xu, C.; Chen, X. SMig-RL: An evolutionary migration framework for cloud services based on deep reinforcement learning. *ACM Trans. Internet Technol.* 2020, 20, 43. [CrossRef]

11. Li, L.; Liu, H. Video Stream Session Migration Method Using Deep Reinforcement Learning in Cloud Computing Environment. *Wirel. Commun. Mob. Comput.* 2021, 2021, 5579637. [CrossRef]

12. Nabavi, S.Y.; Bushehrian, O. An adaptive plan-oriented and continuous software migration to cloud in dynamic enterprises. *Softw. Pract. Exp.* 2019, 49, 1365–1378. [CrossRef]

13. Bazi, H.R.; Hassanzadeh, A.; Moeini, A. A comprehensive framework for cloud computing migration using Meta-synthesis approach. *J. Syst Software* 2017, 128, 87–105. [CrossRef]

14. Tang, L.; Chen, H. Joint Pricing and Capacity Planning in the IaaS Cloud Market. *IEEE Trans. Cloud Comput.* 2017, 5, 57–70. [CrossRef]

15. Zhu, Z.; Peng, J.; Liu, K.; Zhang, X. A game-based resource pricing and allocation mechanism for profit maximization in cloud computing. *Softw. Pract. Exp.* 2020, 50, 173–189. [CrossRef]

16. Taghavi, M.; Bentahar, J.; Otrok, H. Two-stage game theoretical framework for IaaS market share dynamics. *Future Gener. Comput. Syst.* 2020, 201, 173–189. [CrossRef]

17. Zhu, Z.; Peng, J.; Liu, K.; Zhang, X. A game-based resource pricing and allocation mechanism for profit maximization in cloud computing. *Softw. Pract. Exp.* 2020, 50, 173–189. [CrossRef]

18. Zhang, Y.; Shi, Y.; Shen, F.; Yan, F.; Shen, L. Price-Based Joint Offloading and Resource Allocation for Ad Hoc Mobile Cloud. *IEEE Access* 2019, 7, 62769–62784. [CrossRef]

19. Zhang, H.; Xiao, Y.; Bu, S.; Yu, R.; Niyato, D.; Han, Z. Distributed Resource Allocation for Data center Networks: A Hierarchical Game Approach. *IEEE Trans. Cloud Comput.* 2020, 8, 778–789. [CrossRef]

20. Qiu, X.; Wu, C.; Li, H.; Li, Z.; Lau, F.C.M. Federated Private Clouds via Broker’s MarketPlace: A Stackelberg-Game Perspective. In *Proceedings of the 2014 IEEE 7th International Conference on Cloud Computing, CLOUD 2014, Anchorage, AK, USA, 27 June–July 2014*; pp. 296–303. [CrossRef]

21. Wei, W.; Fan, X.; Song, H.; Fan, X.; Yang, J. Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud. *IEEE Trans. Serv. Comput.* 2018, 11, 78–89. [CrossRef]

22. Wang, Y.; Meng, S.; Chen, Y.; Sun, R.; Wang, X.; Sun, K. Multi-leader Multi-follower Stackelberg Game Based Dynamic Resource Allocation for Mobile Cloud Computing Environment. *Wirel. Pers. Commun.* 2017, 93, 461–480. [CrossRef]

23. Fernández-Cerero, D.; Fernández-Montes, A.; Jakóbik, A.; Kolodziej, J. Stackelberg Game-based Models in Energy-aware Cloud Scheduling. In *Proceedings of the 32nd European Conference on Modelling and Simulation, ECMS 2018, Wilhelmshaven, Germany, 22–25 May 2018*; Volume 1, pp. 460–467. [CrossRef]

24. Li, C.; Liu, Y.; Luo, Y.; Zhou, M. Collaborative content dissemination based on game theory in multimedia cloud. *Knowl. Based Syst.* 2017, 124, 1–15. [CrossRef]

25. Chakareski, J. Cost and profit driven cloud-P2P interaction. *Peer Peer Netw. Appl.* 2013, 8, 244–259. [CrossRef]

26. Cao, T.; Zhong, L.; Xiao, H.; Song, C.; Yang, S.; Xu, C. Credible and economic multimedia service optimization based on game theoretic in hybrid cloud networks. *Trans. Emerg. Telecommun. Technol.* 2019, 20, e3779. [CrossRef]

27. Fan, X.; Weber, W.; Barroso, L.A. Power provisioning for a warehouse-sized computer. In *Conference Proceedings Annual International Symposium on Computer Architecture, Proceedings of the 34th Annual International Symposium on Computer Architecture, San Diego, CA, USA, 9 June 2007*; Assoc Computing Machinery: New York, NY, USA, 2007; pp. 13–23.

28. Xiong, Y.; Gan, J.; An, B.; Miao, C.; Bazzan, A.L.C. Optimal electric vehicle charging station placement. In *IJCAI 2015, Proceedings of the 24th International Joint Conference on Artificial Intelligence, 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015*; IEEE: New York, NY, USA, 2015; pp. 2662–2668.