

Article

# Multi-Objective Service Placement Scheme Based on Fuzzy-AHP System for Distributed Cloud Computing

A-Young Son and Eui-Nam Huh \*

Department of Computer Science and Engineering, Kyung Hee University, 1732, Deogyong-daero, Giheung-gu Yongin-si, Gyeonggi-do 17104, Korea

\* Correspondence: johnhuh@khu.ac.kr; Tel.: +82-31-201-2454

Received: 22 July 2019; Accepted: 23 August 2019; Published: 29 August 2019



**Abstract:** With the rapid increase in the development of the cloud data centers, it is expected that massive data will be generated, which will decrease service response time for the cloud data centers. To improve the service response time, distributed cloud computing has been designed and researched for placement and migration from mobile devices close to edge servers that have secure resource computing. However, most of the related studies did not provide sufficient service efficiency for multi-objective factors such as energy efficiency, resource efficiency, and performance improvement. In addition, most of the existing approaches did not consider various metrics. Thus, to maximize energy efficiency, maximize performance, and reduce costs, we consider multi-metric factors by combining decision methods, according to user requirements. In order to satisfy the user's requirements based on service, we propose an efficient service placement system named fuzzy-analytical hierarchical process and then analyze the metric that enables the decision and selection of a machine in the distributed cloud environment. Lastly, using different placement schemes, we demonstrate the performance of the proposed scheme.

**Keywords:** distributed cloud computing; service placement; fuzzy system; energy efficiency; resource management; Internet of Things

## 1. Introduction

In the past decade, traditional cloud computing technology, which relies heavily on the centralized cloud data center (CDC), has been widely used. Cloud services have also been offered in the form of centralized CDCs that can be remote from mobile users. This sometimes entails network delay due to the centralized CDC structure.

Recently, significant progress has been made in the deployment of cloud services nearer to users, which provides higher reliability and faster response time. In particular, in distributed cloud services, a small cloud, such as a distributed cloud, can be connected to provide services to the device users in order to provide the low latency and rapid response time of mobile devices. By placing and migrating to the cloud services that have a higher resource near to the mobile users, the distributed cloud offers faster response time, along with higher scalability and reliability than centralized CDCs [1–3].

One of the important issues in distributed cloud data centers (DDCs) is service placement (SP) with mobility. The limited coverage of a single server to support mobile users will result in service-level agreement (SLA) violation, which further leads to a decrease in performance and quality of service (QoS). Therefore, it is difficult to ensure seamless service [4,5].

Unlike conventional CDCs, DDCs are characterized by a relatively low computing resource and storage capacity. These DDCs consist of computing devices such as personal computers (PC), tablets, and laptops.

To support distributed cloud services efficiently, there is a need to manage and place various tasks in order to meet various SLAs parameters such as resource utilization, response time, service availability, latency, and costs. As many different end devices need to connect to the distributed cloud for SP, a decision should be made, when a service needs migration [6]. For this reason, various approaches for placement and migration have been studied and proposed for DDCs. To take advantage of nearby resources and resource management, SP and migration techniques have been playing an important role [7,8]. However, existing SP and migration schemes have some challenges, which are covered below.

First, placement and migration techniques in cloud computing have been developed to find various objectives to maximize energy efficiency [9], minimize costs [10], and maximize performance. However, in most of the previous approaches, objectives of services are not considered. For example, References [11,12] mainly focus on energy efficiency, while Reference [11] focuses on performance. Due to the high uncertainty of user mobility and requirements, it is very difficult to conduct efficient SP. Thus, in order to meet various requirements, we should also consider various metrics based on the objective.

Second, the cloud service provider (CSP) needs to balance conflicting objectives such as minimizing resources wastage, reducing overall operation costs, and minimizing service latency [13]. Specifically, multi-objective optimization investigates the search methods that are used to find solutions based on several conflicting performance objectives, such as waiting time minimization, resource utilization maximization, and profitability maximization [10–13].

It is difficult for the CSP to manage the trade-off between objectives. For example, energy efficiency is typically optimized at the expense of QoS. From the perspective of the cloud user, the only priority is the fulfillment of their application resource demands. The resource demand is typically specified as SLA. Although this approach can provide continuous service with a minimal delay, there are disadvantages, such as degraded virtual machine (VM) execution performance and increased SP cost. In addition, some SP algorithms provide a well-balanced SLA efficiently. However, they need to be avoided since they increase network traffic and energy costs [7,8].

To solve these problems, we propose an SP scheme based on a fuzzy-analytical hierarchical process (FAHP) and attempt to conduct research on the SP scheme for a multi-objective in DDCs. Especially, we focus on multi-objectives in energy efficiency and performance improvement. We also provide analysis of metrics related to efficiency for performance, efficiency for energy, and cost. The evaluation results show the effectiveness of the proposed scheme. Our contributions are as follow.

- (1) Categorization of metrics to improve energy efficiency and performance,
- (2) The SP decision whether or not based on the fuzzy system,
- (3) Selection for target machine with multi-metrics based on the analytical hierarchical process (AHP) system through the calculation of weights.

The rest of this paper is organized as follows. Section 2 provides an analysis on related work to solve the current challenges. In Section 3, we propose an SP system and scheme with steps. In Section 4, we evaluate and analyze the proposed system and scheme in terms of the efficiency of performance and energy in DDCs. Lastly, we present our conclusion.

## 2. Related Work

When service is placed or migrated, service users request their resource requirements from CDCs. The CSPs need to make a decision on resource allocation for the VMs and select the most suitable target machine. This process is known as placement. When service is executed, if the resource utilization is higher than the setting value, it can lead to performance degradation. However, if the resource utilization is lower than the setting value, it can lead to an increase in cost among CDCs. Thus, it needs to consider multi-metrics when SP is operated. To respond to changing workloads, there have been previous approaches for placement schemes [1–15] with various objectives.

Thus, this section provides a comprehensive review of existing placement schemes and illustrates the analysis of metrics, according to objectives with placement schemes.

### 2.1. SP

The objective of the SP is to find an efficient placement that does not violate the SLA while minimizing resource wastage and cost. To execute SP, as shown in Figure 1, SP steps are as shown in Figure 1 and described in the three subsections that follow.



**Figure 1.** Placement process.

#### 2.1.1. Load Detection

This step must set a threshold limit to decide when a certain server is over-utilization or under-utilization. The load detection step can be divided into overload (hotspot) detection and underload (coldspot) detection.

- **Hotspot Detection:** In the existing works, when the monitoring value or predictive value such as resource utilization, waiting time, and user mobility information exceeds the set threshold, this is considered a hotspot [16] in CDCs. Getting rid of hotspots can ensure the QoS and minimize resource wastage.
- **Cold Spot Detection:** If a specific VM detects low utilization than the set threshold, this state can be termed a “Cold Spot.” It is the aim of SP to remove this state through migration to other active hosts. For efficient SP, it needs to appropriately see that VMs are selected to solve the over-loaded or under-loaded host.

#### 2.1.2. Placement Decision

The VM decision step decides which VMs should be placed or migrated to minimize the resource wastage of the over-loaded or under-loaded host as well as to satisfy user requirements such as reducing service response time, energy consumption, and cost. Thus, it is important to detect the VM state such as resource usage, and energy consumption in the load detection step, and to choose optimized metrics depending on user requirements for the VM decision.

#### 2.1.3. Placement Action

In this step, the selected VM (service) in the previous step is placed on another VM. CDCs and DDCs need to place and migrate to minimize resource wastage, balance workloads, maximize energy efficiency, and save cost.

In this step, the proper VM is selected to place for resource management and efficiency service operation in CDCs and DDCs. Due to its importance, the problem of VM placement in the CDCs and DDCs became the focus of practical studies, and numerous VM placement methods have proposed algorithms for decisions, which included VMs (or services) to be placed (or migrated).

In previous work, for various user requirements, VM placement schemes have been proposed considering different metrics such as resource utilization, response time, etc. [3,4,9], according to the objective.

Due to the scaling of cloud infrastructures and dynamic workload, when services are placed or migrated, considering the target machine is another challenge. Accordingly, various research studies have been proposed for decision making for selecting between VMs and physical machine (PM) [9].

The decision for the target machine has to consider multi-metrics such as resource utilization, energy consumption, and latency [12,16], according to user requirements.

For selecting a source machine and target machine, it is important to detect an over-load and under-load state [17]. When a server does not have enough resources for the operating service, it is called a hotspot, which means it is overloaded and needs to place some services to another server [17]. Hotspots contribute to performance degradation as well as reduce energy efficiency. In addition, if the server state detected is a cold spot, it also contributes to performance degradation as well as reduced energy efficiency.

Thus, when a service is placed, it aims to find the best target machine, which leads to the improvement of performance, and energy efficiency of the CDCs and the reduced operation cost of the CDCs and DDCs. Large-scale CSPs face the challenging problem of how to place resources over areas so as to reduce the total cost while providing an adequate level of performance, by taking into account the dynamic demand.

## 2.2. SP Approaches

Recently, the problem of placement in CDCs has been studied extensively. In this section, we present a categorization of previous studies that have proposed architectures and approaches for VM placement. Furthermore, we present analyzed metrics from previous approaches, according to the objective.

### 2.2.1. Architecture

SP approaches have been studied in two different cloud architectures: the architecture of the solution, that uses a (1) centralized architecture or (2) a distributed architecture.

A centralized architecture usually has a long latency than a distributed architecture, and the performance may degrade as the demand increases. However, the most studied cloud architecture is the centralized CDCs. Thus, a distributed architecture study for SP schemes is definitely required in the future.

### 2.2.2. Approaches

This section describes the existing VM placement approaches in the previous work. To address limitations, we propose an SP scheme based on multi-criteria decision-making (MCDM) [8]. Figure 2 shows the classification for CDC. In other words, the metric affects the priority for selecting VM. According to a used metric in the proposed system, it can select the target machine.

Selecting the optimal services for SP from among the increasing number of various cloud services in the cloud data centers (CDCs) is a great challenge. Many cloud services will be available in the CDCs that will further complicate the service of selecting the best or most optimal services for the user from among many different types of services. Therefore, cloud service selection will be a great challenge. To solve this challenging research problem, a decision-making method is required to assist the users in service selection [11–15].

In References [15–17], the author proposes a new model for SP in CDCs using the AHP method, which is one of the most efficient MCDM techniques. This method can find the most suitable service in the CDCs for the SP. Typically, each alternative is evaluated with the decision metric, and each metric is assigned a weight based on its importance. The alternatives are ranked according to the relative closeness of the ideal solution. The closer the alternative is, the better. VMs are sorted from the highest overloaded to the idle ones based on some metrics such as CPU, RAM, and NET, etc. Each attribute is given a weight. There are many MCDM methods, as shown in Figure 2. However, the analysis layer process (AHP) is popular since the MCDM method is used in many areas to make decisions based on weights in order to meet conflicting goals.

However, almost all the proposed SP schemes consider one or two objectives and metrics, as shown in Table 1. In this paper, we consider multi-objectives and multi-metrics for satisfying the user’s requirements.

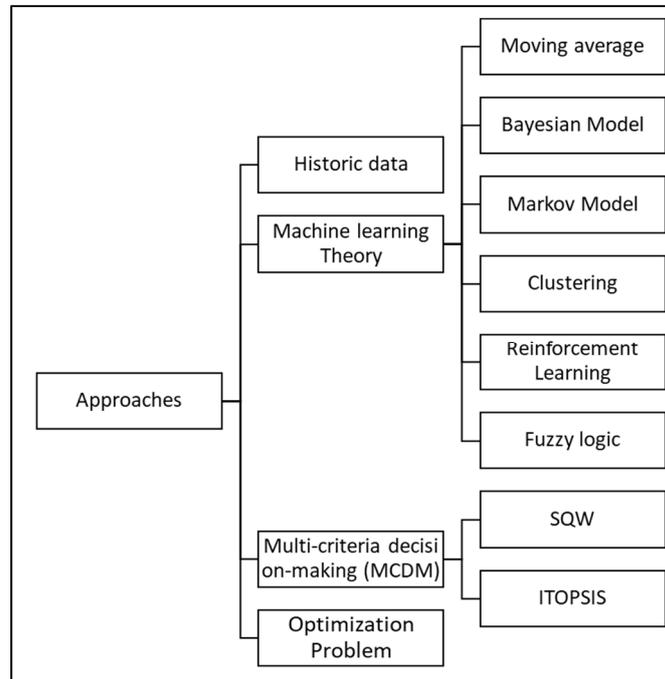


Figure 2. Approaches for decision-making.

Table 1. Related work on placement.

	[18]	[19]	[20]	[21]	Proposed Scheme
Goal	Minimize energy cost	Maximize energy efficiency	Minimize the number of active nodes	Maximize resource utilization	Energy efficiency and improvement of performance
Algorithm	Reinforcement	Constraint satisfaction problem	GA	AHP, Fuzzy, TOPSIS	Fuzzy-AHP
Method	Decision of the power mode	Selection of VM	Decision for the number of nodes	Combination of MCDM	Selection of Target machine
Metric	CPU	Latency	Runtime	CPU, RAM	resource, latency, etc.
Pros	Improvement of user satisfaction	Load balancing	Performance improvement	Resource utilization improvement	Consideration of multi-objectives
Cons	Consideration of CPU utilization	Consideration energy only	Small-Scale environment	Accuracy	Need for more tests in real environments
Common with our work	Use of metrics for resource utilization				
Different with our work	Consideration of multi-metrics with satisfying user requirements				

### 2.3. Service Placement Metric

The SP scheme helps reduce energy consumption and maximizes performance in CDCs. As shown in Table 2, we categorize related works according to the objective and metric.

**Table 2.** Related work on SP.

	Metric	Decision Value	Algorithm
[22]	Response time, QoS	SP	GA
[23]	Resource wastage	SP	GA
[24]	Cost	Load dispatching	Greedy, GA
[25]	Cost	Task distribution	Mixed ILP
[26]	Network latency	SP	ILP
[27]	Power consumption	SP	Weight
[28]	Delay	SP	Markov
[29]	Resource usage	Resource allocation	Consensus
[30]	Energy	SP	First-fit
[31]	Delay	SP	Own algorithms
Proposed scheme	Multi-metrics	SP	Fuzzy-AHP

We consider multi-metrics to find a good balance between metrics to minimize the trade-off. The performance degradations can increase the execution time, which, in turn, decreases the energy savings in order to define the balance between them.

We further summarize and compare the SP scheme based on the decision method described in Table 2. Most of these SP schemes attempt to influence decision making when researchers try to include one or two metrics such as resource usage, response time, and cost. To ensure that their SP meets the specific performance requirements, the algorithms must be improved to be more flexible. The SP process introduces various trade-offs, such as that between delay and cost. However, it is difficult to develop an efficient, high-performance SP approach that is energy-efficient and optimizes the balance between allocated resource and SP overhead.

There are many approaches for SP. Although these approaches address important and distinct aspects, to the best of our knowledge, each is insufficient for a scheme that applied current requirements considering conflicting goals.

To find a good balance between metrics, in this paper, we categorize metrics according to the objective: performance, energy efficiency, and cost.

The explanation for objectives is given in the following three sub sections.

### 2.3.1. Performance

An SP deals with fast and efficient approaches to provide resource management, maintenance, and fail-over. Many approaches are attempts to provide better performance.

Table 3 shows the suggested metrics to evaluate the performance of SP in previous studies. We have categorized the SP metrics for performance. In recent studies, the following six metrics were typically selected to evaluate SP performance: execution time, waiting time, resource utilization, transferred data, number of migrations, and throughput. These metrics are optimized in Table 3, according to different behaviors, such as the retrieval of maximal or minimal values. In addition, most of the approaches focus on minimization of the placement time. Time minimization reduces the execution time, downtime, and waiting time [1–7]. Time minimization is an important metric due to the affection of the performance of VMs.

**Table 3.** Performance metric.

Metric	Explanation	Optimization Behavior	Previous Research Reference Number
Execution time (ET)	The amount of time from start to finish during SP	Minimize ET	[7,8]
Waiting time (WT)	The amount of time that a certain VM is occupied resource, while the other VM is waiting for using the resource.	Reduce WT	[14,26]
Resource utilization (RU)	The average percentage for RU	Maximize RU	[9,15,23–25]
Transferred data (TD)	The amount of transfer data during the SP	Reduce TD	[25]
Number of migrations (NM)	The NM for SP	Minimize NM	[30]
Throughput (TP)	The number of tasks until completion of SP.	Maximize TP	[32]

### 2.3.2. Energy Efficiency

Recently, high EC has led to an increase in CDCs’ operating costs and a decrease in service performance. As a result, concerns about energy efficiency are also growing.

Therefore, the SP approaches for preserving server energy are important for CSP and service users.

Table 4 shows the metrics that have been used to evaluate the energy efficiency of SP in previous studies. Recent studies that have evaluated SP energy efficiency typically applied one or more of the following four metrics: RU, EO, NM, and EC.

**Table 4.** Energy efficiency metric.

Metric	Explanation	Optimization Behavior	Previous Research Reference Number
Resource utilization (RU)	The amount of resource usage for service execution	Maximize RU	[9,10,19,20]
Energy overhead (EO)	The amount of extra resources usage to perform SP	Reduce EO	[33–35]
Number of migration (NM)	The NM for SP	Minimize NM	[17]
Energy consumption (EC)	Amount of energy to perform SP in CDC	Reduce EC	[10,13,30,34]

Previous approaches to reduce EC have included RW minimization, and RU maximization for resource management [9,10,17,19,20]. Table 4 presents a list of the metrics adopted in previous research and their optimization behavior.

### 2.3.3. Cost

Cost efficiency of placement and migration try to reduce the CDCs’ service operation cost. In previous approaches, SP methods have been studied to minimize cost while considering QoS. Table 5 shows the cost metrics in the cost-efficient SP and migration approaches. We categorized the SP metrics. The following three metrics were typically applied in recent studies: VM cost, PM cost, and the distance between VMs and service users. These metrics were optimized according to different behaviors to minimize metrics. The approaches for cost saving are explained as follows. The following metrics were also considered for different service types.

Table 5. Cost metrics.

Metric	Explanation	Optimization Behavior	Previous Research
VM cost (VC)	The operation cost of using the VMs when services are executed in CDC	Minimize VC	[12,25,34]
PM cost (PC)	The cost of using the PM in the specific period	Minimize PC	[36–38]
Distance between VMs and service users (DS)	Distance between the VMs and the service users	Minimize DS	[39,40]

### 3. Proposed Method

Cloud computing supports several service types with resource management. Thus, SP schemes have to consider various service types and objectives (energy efficiency [41–46], performance [47], and cost). In the proposed system, the input metric for multi-objective factors, such as performance and energy efficiency, is not limited.

We propose SP schemes that include the following functions.

- Determine services that need to be placed through resource monitoring for the detection of overload or underload.
- Determine the target machine based on the AHP system by calculating weights such as prediction values for resources.

#### 3.1. Proposed Architecture

The fuzzy system has been used in many areas for decision-making. Thus, we used the fuzzy system to reflect the uncertainty of the placement condition. It can control SP, according to the objective, by determining the VMs that require placement and selection of the target machine, according to calculated weights. It helps improve performance and minimize resource wastage in DDCs. The fuzzy-AHP system architecture and phase for SP are as follows.

As shown in Figure 3, the proposed system consists of three main functions: Fuzzy System, AHP System, and placement manager.

- In the fuzzy system, it consists of two main functions: fuzzy inference system and rule configure system.
- Fuzzy inference system (FIS) provides an appropriate target machine to meet service requirements when resources reach the threshold assigned for the service. In this rule-configure system (RCS), the initial rule is to reflect the current uncertainty of the placement condition. After the adaptive rule has been configured, the stability of the range of RU is increased by reconfiguring the range. We can set the objective rule for application to the fuzzy system by reconfiguring the range. We present a detailed explanation fuzzy system in Section 3.2.
- In the AHP system, we describe how to measure the action. In exploring the optimal placement scheme, Table 2 defines relative value-based weight for our set of policies based on Table 4. The evaluated ranking is reflected by the rationality of the ranking scale. The ranking is a numerical scale that indicates the relative importance of metrics through a comparison. We use the standard Saaty 'AHP [48,49] 9-level scales, which Table 2 shows. Table 2 describes the notation for the proposed scheme. We send this result to the placement manager for weigh the calculation.
- The placement manager consists of two modules: decision and action for placement. In the decision module, it is the decision on the SP situation whether or not based on the fuzzy system values. In the action module, we can perform placement based on the result of the AHP system considering the target machine. As a result of our calculation, we can select the target machine. In the next section, we will explain the detailed proposed scheme by the step.

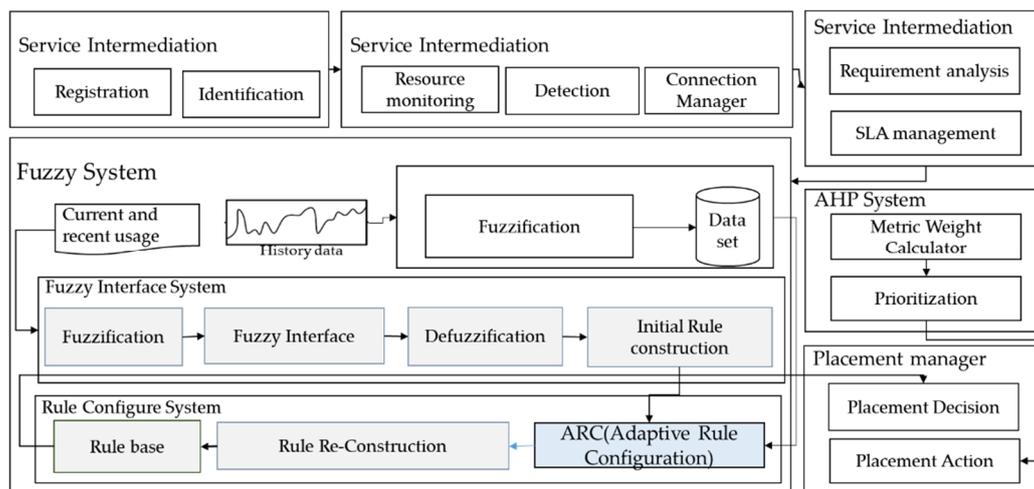


Figure 3. Service placement architecture.

### 3.2. Steps of the Proposed Scheme

We propose an efficient placement scheme considering the metric. We propose an efficient placement scheme considering the metric. In steps 1 and 2, we gather information based on requirements and the resource state for SP. In steps 3 to 7, a fuzzy system is conducted for the SP decision. Then an AHP system is conducted in steps 8 to 10 for target machine selection. As shown in Figure 4, the detailed steps for the proposed SP scheme are as follows.

- *Step 1. Resource monitoring and profiling.* The first step is to monitor the resource state of an ongoing service. It is needed for monitoring and profiling the SP decision and selecting the target machine.
- *Step 2. Service requirement analysis.* Based on the demand of the user requirement, we categorize objectives and needed metrics.
- *Step 3. Establish a fuzzy variable (create a membership function (MF)).* The concept of a fuzzy system is based on steps 3 to 5: fuzzification, rule construction and evaluation, and defuzzification [24]. The MF is used to calculate the probability values of an alternative for selecting the target machine. The input metric for the alternative is sent to the fuzzification phase in the fuzzy system, as shown in Figure 6.
- *Step 4. Fuzzy inference (rule construction).* Rules are designed by the importance of the metric relationship. Algorithm 1 represents our Fuzzy block with fuzzify and defuzzify to set MF.
- *Step 5. Defuzzifying.* In this step, we can obtain the output value of the “Placement decision” and “Target machine list” through the defuzzifying process.
- *Step 6. Placement decision.* In the configuring system, through the configuration of rules, we obtain a more objective decision value. In this step, if the “placement decision” is detected, we go to the AHP system.
- *Step 7. Perform the AHP system.* We construct the hierarchy structure based on AHP. AHP is an MCDM approach that can be used to solve complex decision problems. The main steps of AHP include making a hierarchy, assigning weights to each metric, investigating the consistency check of the system, and, ultimately, making a decision (determining the priorities of options). To select a service, this phase is the most important step. In this paper, we considered one scenario for energy and performance. A hierarchy structure of the performance metric is designed, as shown in Figure 5, based on previous approaches.
- *Step 8. Determine the decision table.* We explain how to measure the decision. When an efficient SP scheme is applied, Table 6 defines the relative value for the policies based on Table 7. According to the relative value, we can calculate the weight.

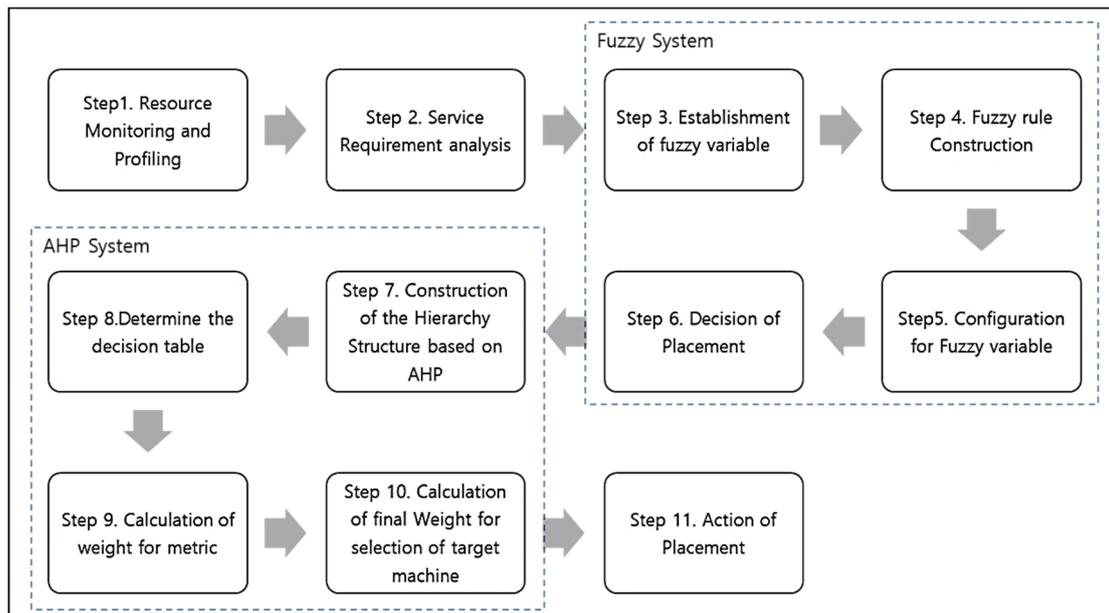
**Algorithm 1:** Fuzzy block (fuzzify and defuzzify setting).

```

FUZZIFY CPUConsumption
    TERM low: = (0, 1) (73, 1) (77, 0);
    TERM mediumLow: = (73, 0) (76, 1) (79, 1) (81, 0);
    TERM medium: = (79, 0) (81, 1) (84,1) (86, 0);
    TERM mediumHigh: = (84, 0) (86, 1) (89, 1) (91, 0);
    TERM high: = (89, 0) (91, 1) (100, 1);
END_FUZZIFY

FUZZIFY RAMConsumption
    TERM low: = (0, 1) (73, 1) (77, 0);
    TERM mediumLow: = (73, 0) (76, 1) (79, 1) (81, 0);
    TERM medium: = (79, 0) (81, 1) (84,1) (86, 0);
    TERM mediumHigh: = (84, 0) (86, 1) (89, 1) (91, 0);
    TERM high: = (89, 0) (91, 1) (100,1);
END_FUZZIFY

DEFUZZIFY NecessityOfPlacement
    TERM safe: = (0,0) (5,1) (10,0);
    TERM standBy: = (10,0) (15,1) (20,0);
    TERM nonUrgentSP: = (20,0) (25,1) (30,0);
    TERM urgentSP: = (30,0) (35,1) (40,0);
    METHOD: COG; // Use 'Center of Gravity' defuzzification method
    DEFAULT: = 0; // Default value is 0 (if no rule activates defuzzifier)
    
```



**Figure 4.** Proposed SP steps.

- Step 9. Calculate the weight for metric.* We use a ranking method by priority. A ranking is a numerical scale of the numbers that indicate how many numbers are more important with respect to the metric to which they are compared. We use the standard Saaty AHP 9-level scales, as shown in Table 6. In this system, we have to perform a comparison of every combination of two VMs at the same level (pairwise comparison) so that it is stated mathematically. The evaluation begins with the metric that represents the delimitation of the trade-off. We also use the S (Score) as the example

to represent how the weight is defined. As shown in Table 8, thus, we get the matrix-based Equations (1)–(3).

$$W_m = \sum W_n \tag{1}$$

$$W_n = \frac{V_{n-1}}{V_n} \tag{2}$$

$$\text{s.t max } \sum W_m \tag{3}$$

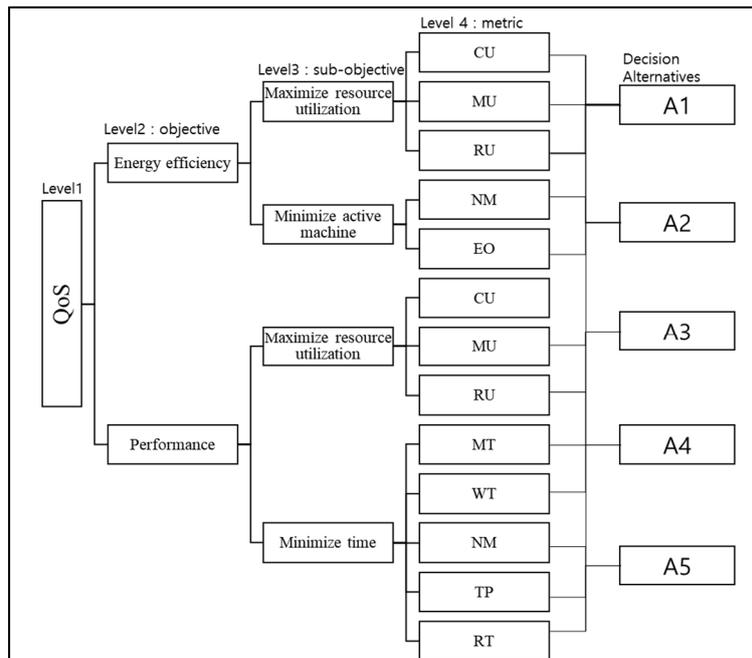


Figure 5. Hierarchical structure.

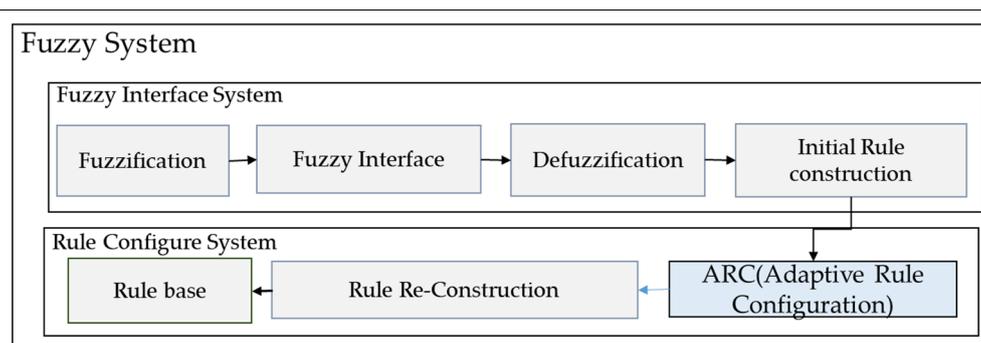


Figure 6. Fuzzy system.

Table 6. Relative value.

Relative Importance	Value
Equal importance	1
Somewhat more important	3
Definitely more important	5
Much more important	7
Extremely important	9

**Table 7.** Relative value-based weight.

	<b>M1</b>	<b>M2</b>	...	<b>Mn</b>
M1	S1/S1	S1/S2	...	S1/Sn
M2	S2/S1	S2/S2	...	S2/Sn
...	...	...	...	...
Mn	Sn/S1	Sn/S2	...	Sn/Sn

**Table 8.** Notation used in this paper.

<b>Notation</b>	<b>Description</b>
M1	Resource utilization of VM
M2	Capacity of VM
M3	The number of VM migrations for SP
M4	The time of delay during the SP
M5	Response time during SP
Mn	Metric n
Sn	Relative score n
Wn	Sum of weight
Wm	Weight for metric
RU	Resource utilization of VM
CU	CPU utilization of VM
MU	Memory utilization of VM
P	Percentage of resource utilization
RT	Response time of service
WT	Waiting time of service
NM	Number of VMs

Table 6 defines relative score-based policies. In addition, in Table 8, we present the used metric explanation. Figure 5 presents an example of the AHP structure of user requirements in order to decide among cloud service candidates.

A number of the previous studies [9–13,19–24,41–46] considered RU as a metric when services were placed or migrated. In this paper, we also considered the RU metric in the evaluation of all such cases with respect to parameters such as performance and energy efficiency. The CSP can decide to assign the weight according to relevant requirements. SP is performed from VM as well as other devices operated by the user. Therefore, RU is calculated to depend on the user’s mobility and the probability of the utilization of resources based on the previous usage demonstrated over time, which is shown in Equation (4).

The equation for RU is as follows.

$$Utilization_{VM_j} = \frac{\sum_{i=1}^n p}{time} \tag{4}$$

For example, low ET is better than a high ET, so we must get a high score. Thus, we present a mathematical method for a reciprocal number. In Equations (5) and (6), we represent two metrics, cpu utilization (CU) and memory utilization (MU), based on RU in order to apply them to situations, such as underload and overload.

CU: CU is the average percentage of utilization (p) of the resource for VM. CU is used for weight calculation. According to weight, it can decide priority. Based on priority, we can get an alternative for target machine selection. The equation for CU is as follows.

$$CU = \begin{cases} p & p \leq 70 \\ 140 - p & p > 70 \end{cases} \tag{5}$$

The score for CU is equal to or under 70%. If it is higher than the threshold, the score is decreased. If CU is 65%, we get a score of 65. If CU is 90%, we get a score of 50. The reason we calculate the equation this way is that, when the CU gets a hotspot, service leads to latency [1].

MU: MU is the average utilization (p) of VM resources. The following equation calculates the MU scores for weight.

$$MU = \begin{cases} p & p \leq 50 \\ 100 - p & p > 50 \end{cases} \quad (6)$$

We propose an SP scheme with metrics that combine score demand based on the user requirements. In this equation, t represents the average amount of time taken for the system to process a request. The following equation obtains a score for RT.

$$RT = \frac{1}{t} \quad (7)$$

- *Step 10.* Calculate the final weight for selecting the service.

Equation (8) determines the target machine with values such as RU, RT, and prediction of WT. In addition, we can calculate selection VM through Equations (8)–(10).

$$Target\ Service_p = RU + RT + WT \quad (8)$$

$$Target\ Service_e = RU + NM \quad (9)$$

$$Selection\ Service = \text{Max}(Target\ service_i) \quad (10)$$

- *Step 11. Action of placement.* After making decision of the target machine for SP, then we operate on SP.

#### 4. Evaluation

In this section, we evaluate our model. Through experiments, we prove the most influential metric categorized according to the objective in Section 2.3. To reach our goals of energy efficiency and performance improvement, SP is needed as well as definition and categorization metrics, according to user requirements, which is described in Section 2.3. Thus, with categorized metrics, we evaluate the SP according to the given case. Each experiment is repeated 20 times.

We divide the two cases used for the validation of the proposed scheme. Each case consists of energy efficiency and performance. We also compare our scheme with other policies.

##### 4.1. Evaluation Environment

In this section, we show the result of the evaluation of our SP scheme. To compare SP efficiency, we used the simulation tool called CloudSim. Table 9 detailed the test environment. In our experiment, we implemented our algorithms in CloudSim 3.0.3 and analyzed the performance of our proposed scheme. We considered heterogeneous data centers, half of which were HP ProLiant G4 and the rest were HP ProLiant G5 servers. The energy consumption was calculated based on HP ProLiant G4, HP ProLiant G5 CPU usage, and power consumption. The fuzzy rules were defined and integrated into CloudSim by using the JFuzzyLogic tool [13].

After creating the VMs on CloudSim, the VMs were placed in a different location. The scheme was able to evaluate the capability of such methods on SP. The reason for implementing test datasets and environments from different hosts is as follows. We wanted to test the generality and adaptability of new host CPU data to realize a multi-objective SP. In this work, we used real-world workload data that is provided from, PlanetLab [25]. These data were collected from the VMs of different servers. The workload is representative of an IaaS cloud environment such as Amazon EC2, where VMs are created and managed by several independent users. This dataset included the resource requirements of the service submitted by 100 users to a cluster of two hosts. With our settings, we generated

requests for each user as if the users were running the service in an IaaS cloud environment such as EC. For each user, we executed the proposed scheme that assigned a resource VM based on the resource requirements of the service. Whenever a user submitted requirements, our proposed SP scheme checked the feasibility corresponding to PlanetLab specifications, as if it could accommodate the service and conduct service SP to the target machine.

**Table 9.** Evaluation environment.

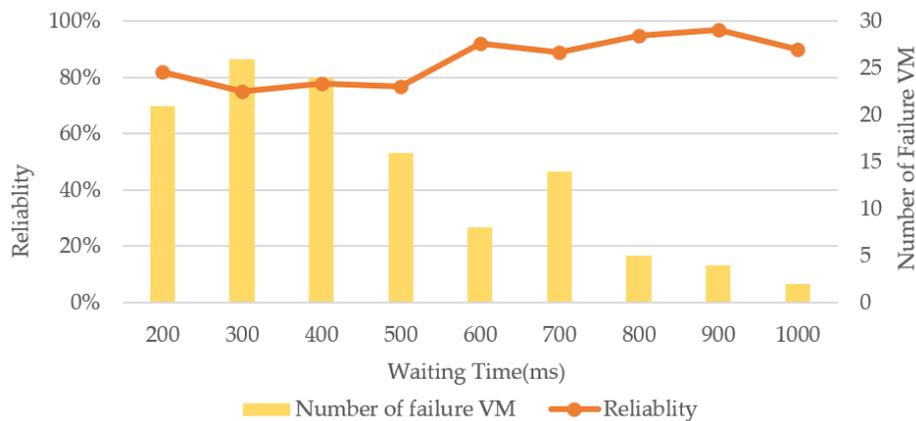
Type	Value
CPU	Intel core i5-4690 3.50 GHz
RAM	16GB
CDC environment	Number of CDCs 2 Number of hosts: 4 Number of VMs: 16 Number of cores: 2 Number of VM instance types: Small Instance: 1 EC2 Compute Unit, 1.6 GB *4 Number of services: 25

#### 4.2. Relation of Metric for Energy Efficiency

In this section, before verification of the efficiency of the proposed method, we present the importance of the relationship between metrics. In addition, we show membership function (MF) is used for the decision of placement. We can set the rule through MF and then make a decision for placement based on the rule.

##### 4.2.1. Importance of Balance between Metrics

Figure 7 shows the relationship of WT and reliability. For measurement, we set the WT per 100 ms. Then the placement measurement is conducted 200 times.



**Figure 7.** WT/reliability trade-off.

As shown in Figure 7, waiting time is increased while the number of failure VMs decreased. The result of the test indicate that we can find that WT is related to performance metrics such as service ET. Thus, we have to consider various metrics, according to objectives.

##### 4.2.2. Decision of Placement

For the decision of placement, we use the fuzzy system. In this section, we represent the decision of placement using MF.

- MF for the resource metric.

$$F = x, \mu F(x) \mid x \in X \tag{11}$$

At this step, metric values are mapped to the appropriate linguistic values by the corresponding universal set. For example, if  $X$  is the universal set, then its elements are denoted by  $x$ , such that the fuzzy set  $F$  in  $X$  is a set where  $\mu F(x)$  is an MF of  $x$  in  $F$ . This step maps each element of  $X$  onto a value between 0 and 1 [25]. In this paper, we used MF to represent the fuzzy number. Based on the linguistic input value, the interface module selects the appropriate rule to be applied and produces the linguistic output value. Both the fuzzifier and defuzzifier use an MF to convert numerical values to linguistic values and vice versa. The MF maps each numerical value to a membership value (certainty level) between 0 and 1 (0 completely uncertain, 1 completely certain). Algorithm 1 describes our fuzzy block, which uses fuzzify and defuzzify techniques to determine the MF. Figures 8 and 9 show the results of Algorithm 1.

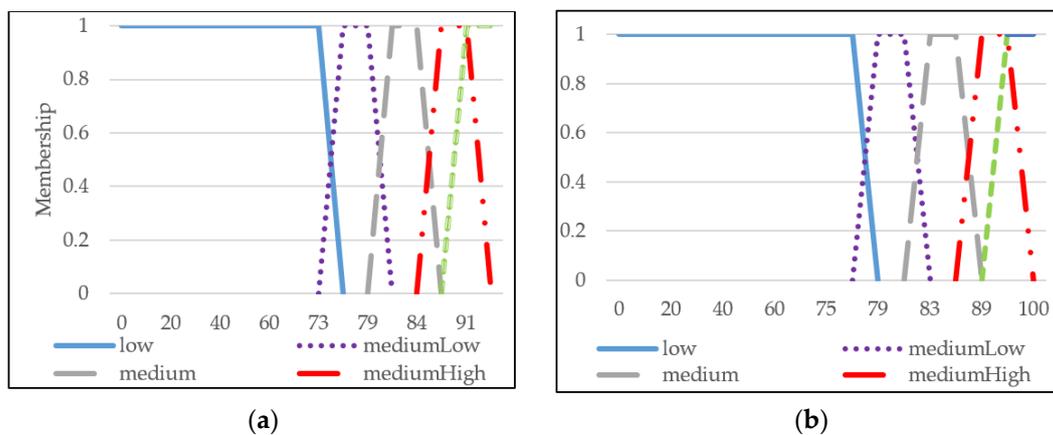


Figure 8. MF for RU. (a) Initial MF. (b) MF after configuration.

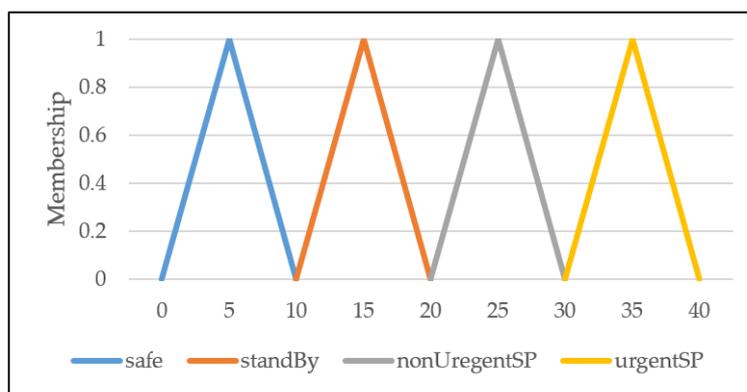


Figure 9. Fuzzy value range for decision of placement.

In other words, to allow the fuzzy value to map for any metric configuration, we set the range for metrics such as CPU consumption to value between 0 and 1. The MFs for CPU consumption and RAM consumption are shown in Figure 8.

As shown in Figure 8a, there are five rules ranging from low to high. We decided on an MF range from low to high. This represents essential parts for SP. In this step, the proposed scheme obtains more reliability than the existing model. The initial MF range is narrow, which is shown in Figure 8a. After configuring for MF, MF ranges of “medium” and “mediumHigh” are widened, as in Figure 8b. Through the proposed system, we can configure the fuzzy value. By fuzzy value configuring, we can obtain a more objective metric than the conventional fuzzy system in the decision step. The RU of cloud services changes dynamically. Because of uncertainty, it is difficult to satisfy user’s requirements from the mathematical representation.

To solve this limitation, we used a fuzzy system with a dynamically changed section. We called it a control point when the value of “Necessity of Placement” is detected. We present the fuzzy-value configuring stage in Figure 9.

- MF for the decision of placement

When “Necessity of Placement” is higher than the threshold, placement, or migration that occurs. Figure 8 shows the result of the MF decision of placement. The fuzzification stage maps an input metric to the appropriate MF. In the inference stage, we represent an appropriate if-then rule and make a result based on the rule.

The output value converts the combined result back into a specific control output value. To transform the results into the output value, defuzzification is performed. In these processes, we used the center of gravity method, which is a common and useful defuzzification stage in fuzzy systems. Figure 8 represents MF for a result of the decision of placement.

Through the output value, we can decide whether to perform the SP action.

### 4.3. Result of Evaluation

In this section, we report an experiment to examine and prove the effect of the metric.

#### 4.3.1. Fuzzy-Variable Configuration

We studied the effect of the NMs on the energy efficiency metric with the following scenario. As shown in Table 10, we set the fuzzy-rule for proving the categorized metric in Section 2.3.

Table 10. Scenario for energy efficiency.

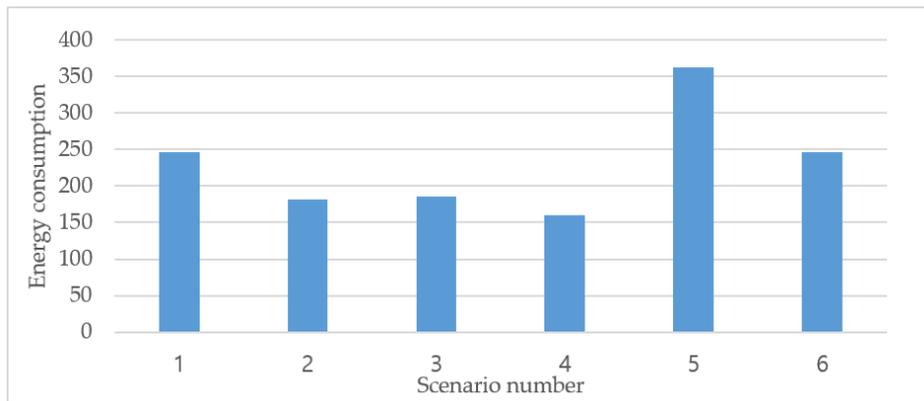
Scenario Number	Power	RU	NM	WT
1	Very high	Very high	Very high	Very high
2	Very high	Very low	Very low	Very low
3	Very low	Very high	Very low	Very low
4	Very low	Very low	Very low	Very low
5	Very low	Very low	Very high	Very low
6	Very low	Very low	Very low	Very high

Table 10 shows the scenarios for energy efficiency over 30 experiments. Each evaluation used a different rule value. Evaluations using the existing scheme were performed for a comparison. In Table 11, we present the explanation of the existing algorithm.

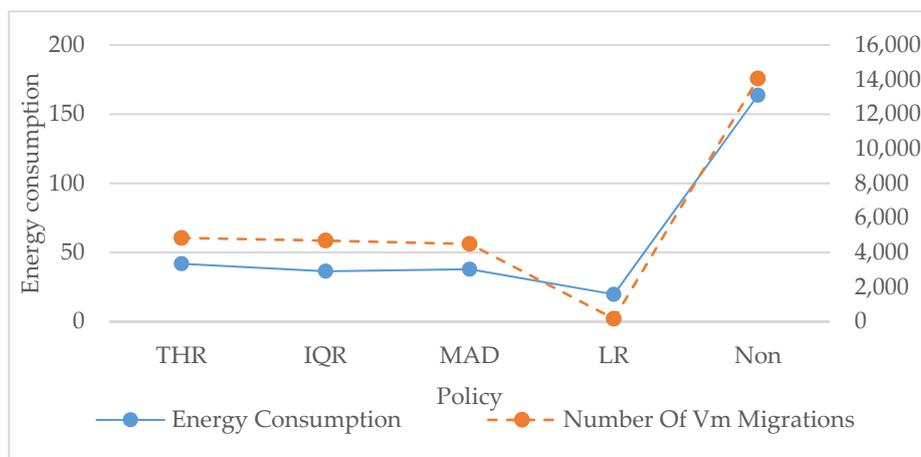
Table 11. Existing algorithm.

Policy	Description
THR	The static threshold policy
LR	A commonly used regression analysis
ARIMA	A is a forecasting method in which the future values a series based on the metric
LSTM	A learning algorithm based on time series data

Figure 10a presents EC according to a scenario such as the one in Table 10. We can analyze the effect of the weighted metric on the EC for SP of energy efficiency. In Figure 10a, we can see that EC of Scenario 5 is higher than those of other scenarios. We can also see from Figure 10b that trends in terms of EC and the NM of tasks increase with the rise of EC. These observations reveal that the NM metric has the most influence on energy efficiency.



(a)



(b)

Figure 10. EC with the number of VMs according to a (a) scenario and (b) policy.

#### 4.3.2. Proposed Placement Scheme Efficiency for Performance and Energy Efficiency

In this section, as shown in Table 12, we define the policy of each individual case. The fuzzy rule used in order to measure the effectiveness of metrics is as follows.

Table 12. Policy of case 1.

Metric	Fuzzy Rule	
CU		
MU		
RU	Very Low	Very High
Power		
NM		
WT		

Table 12 shows an explanation of the metric with policy. To measure the impact of policy considered in the rules on the target machine, we apply the calculation results for the case.

As shown in Figure 11, we calculated the weight values according to the alternatives in the AHP system based on Table 12. For example, it can be seen that, in M1 and M3, A5 is selected as the best alternative. To prove the efficiency of the proposed SP scheme, we divided them into two cases (performance and energy efficiency).

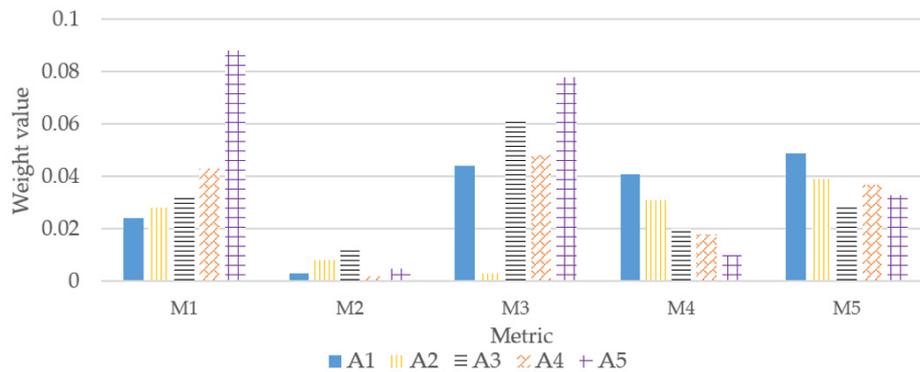
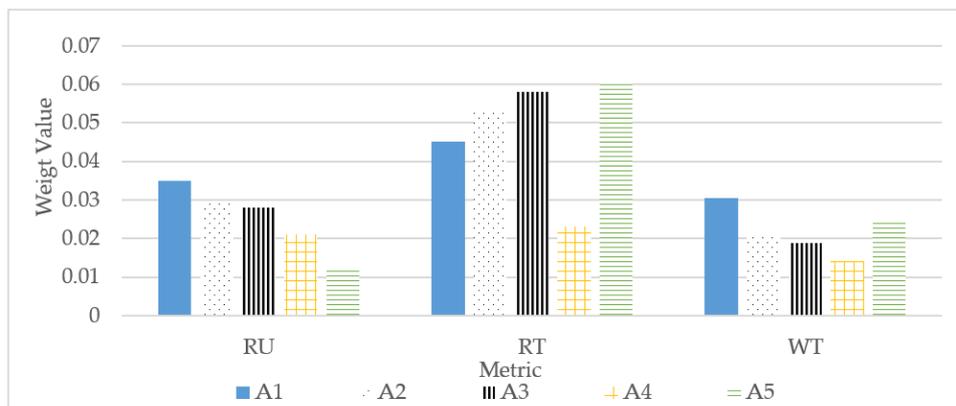


Figure 11. Result of weight.

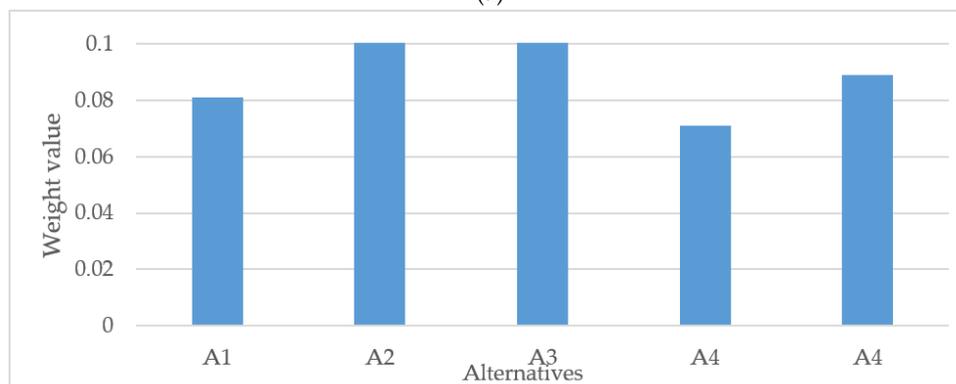
Although most of the existing work focused on the cloud service decision for SP in CDCs, the studies have been limited to a single metric. However, in this study, both single and multiple decision analysis approaches are performed by considering different metrics for performance, energy, and costs in the decision problem to determine the importance of different metrics. Our technique was also used to select the best service among different service providers based on the weights of different decision-making metrics.

- Case 1: Performance

In this case, for placement performance, we consider three metrics RU, RT, and WT. Figure 12a shows the selection VM for different alternatives. As shown in Figure 12, we calculate weight. Figure 12b shows total alternative ranks for case 1 for the performance, in the following order (from best to worst): A2, A1, A4, A1, and A5.



(a)



(b)

Figure 12. Weight rank for case 1 according to the metrics: (a) weight value and (b) alternative score.

We tested the performance of the scheme by measuring the ET during the entire experimental period. Each experiment was tested 30 times to ensure objectivity. Then we calculated the average for the worst-case execution time (WCET). We selected the target machine based on the results shown in Figure 12b. According to the result of the AHP score for the proposed scheme, we compared the results of the ET for SP with LR, Random, and LSTM. In Figure 13, we show that the decrease in ET of LSTM was approximately 9.5% less than that of the proposed scheme. This proves that our scheme increases SP efficiency better than existing schemes.

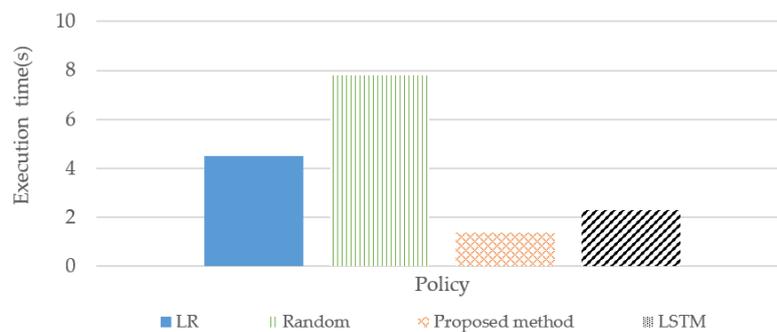
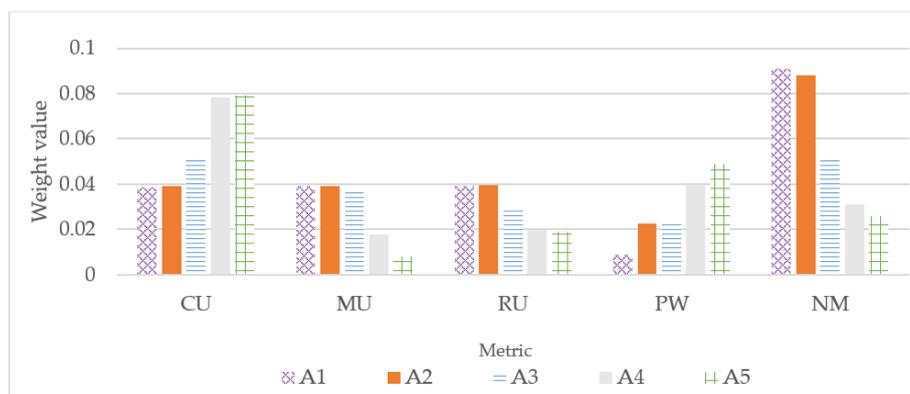


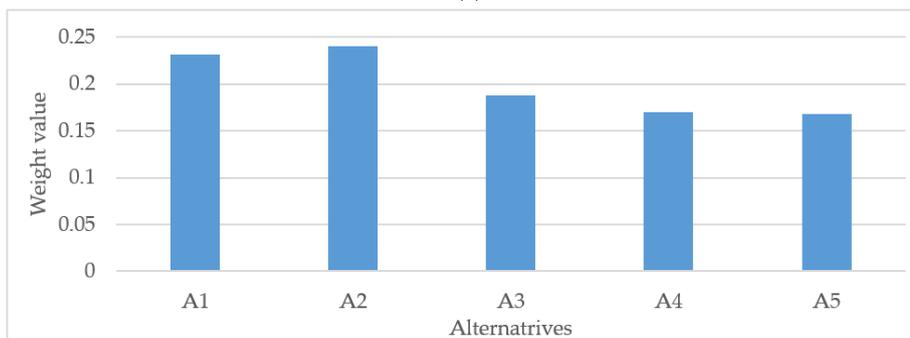
Figure 13. Execution time for case 1.

- Case 2: Energy efficiency

In this case, for placement energy efficiency, we consider five metrics CU, MU, RU, PW, and NM, as shown in Figure 14a. Figure 14b shows the total alternative ranks for case 2 for the energy efficiency, in the following order (from best to worst): A2, A1, A3, A4, and A5. We selected the target machine based on Figure 14b that shows the result of the AHP score. Then, we performed the placement by selecting the target machine, according to the result in Figure 14b.



(a)



(b)

Figure 14. Alternatives for case 2: (a) weight value and (b) alternative score.

Through our scheme, we selected the target machine based on Figure 14b and then performed SP. We compared the result of the EC with Threshold, LR, ARIMA, and LSTM in this paper. As shown in Figure 15, the evaluation results of the proposed scheme verified lower EC than the other methods.

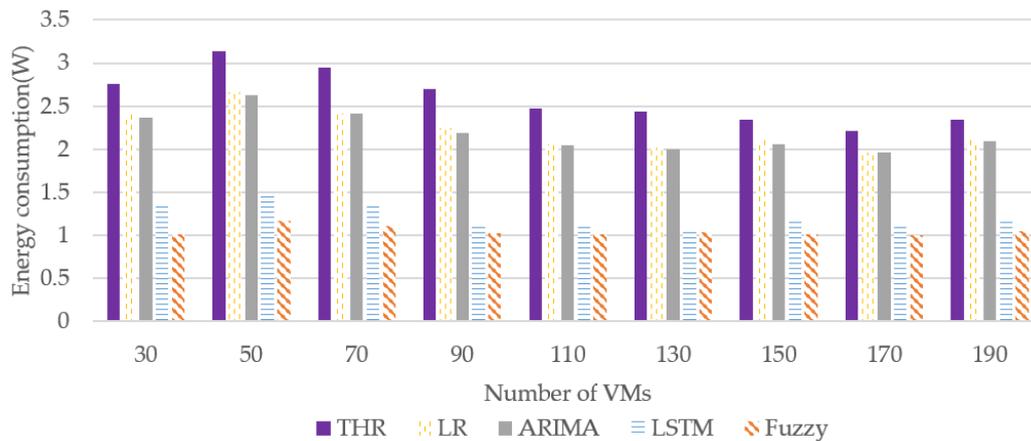


Figure 15. Energy consumption by policy.

Regarding energy, the proposed scheme showed an EC decrease about 23% lower when compared to the threshold scheme. For the test and evaluation, we constantly increased the number of VMs. When the number of VMs was smaller, the difference in EC became larger, as shown in Figure 15. Thus, the test result confirmed that our scheme is efficient in small data centers like distributed cloud environments.

Figure 16 also shows the importance of considering and configuring the metric from Fuzzy (sc1) to Fuzzy (sc3) by the policy. According to the result of the configuring metric, its weight changed. As a result, we saw the different result and verified the importance of the metric. Thus, our SP scheme maximizes the placement efficiency using a configuring metric, according to requirements.

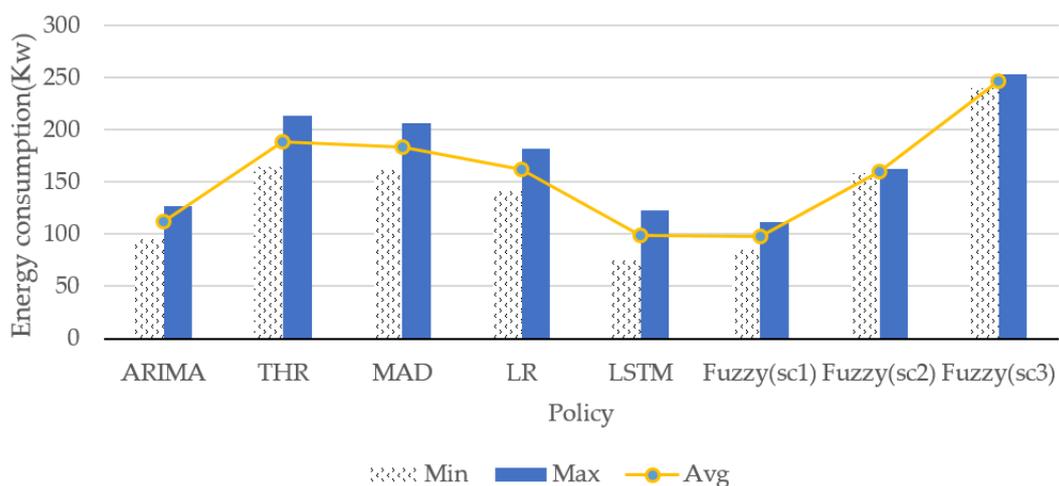


Figure 16. EC according to policy Fuzzy (sc1): Energy, Fuzzy (sc2): Performance, Fuzzy (sc3): Cost.

### 5. Conclusions

The development of huge CDCs and a variety of DDCs has led to the enormous degradation of performance and EC in DDCs [11]. In addition, users move frequently. Thus, it is essential for CDCs to conduct SP considering a variety of elements for energy efficiency and performance efficiency with resource management.

Therefore, various approaches have been studied for the DDCs. In particular, to take advantage of nearby resources and resource management, SP and migration techniques have been playing an important role. In addition, user requirements are increasing with the various service types.

We proposed an efficient SP scheme to solve these limitations and satisfy user requirements. In this paper, we focused on performance and energy efficiency. Moreover, the proposed system allowed efficient placement for a variety of requirements as well as tested cases such as performance and energy efficiency. Through the results of the test cases, we proved that the ET of the proposed scheme was about 39% faster than that of the threshold scheme. In addition, the proposed system reduced EC by up to around 32% in comparison with the threshold scheme. Using the results of the evaluation, our proposed scheme was verified to be more energy-efficient and performance-efficient than the general scheme.

Our proposed scheme can efficiently manage resources and improve user satisfaction according to services of various objectives. We hope this algorithm can contribute to distributed cloud environments when services are placed and migrated.

However, this study has a few limitations. More research and evaluation are needed to consider various environments and user requirements. In addition, the difficulty in making an optimal decision about a dynamically changed service state needs to be addressed. Therefore, in future studies, we will run our scheme with various services and compare it with diverse approaches to advance the scheme.

**Author Contributions:** Conceptualization, E.-N.H. Software, A.-Y.S. Supervision, E.-N.H. Validation, A.-Y.S. Visualization, A.-Y.S. Writing—original draft, A.-Y.S. Writing—review and editing, E.-N.H.

**Funding:** This research received no external funding.

**Acknowledgments:** This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICT Consilience Creative program(IITP-2019-2015-0-00742) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and This work was supported by Institute for Information & communications Technology Planning&Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00294, Service mobility support distributed cloud technology)

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

## Abbreviations

The following abbreviations are used in this manuscript.

DDC	Distributed Data Center
CDC	Cloud Data Center
SP	Service Placement
SLA	Service Level Agreement
AHP	Analytical Hierarchical Process
IoT	Internet of Things
QoS	Quality of Service
CSP	Cloud Service Provider
FAHP	Fuzzy-AHP system
VM	Virtual Machine
PM	Physical Machine
MMO	Multi objective optimization
MF	Membership Function
RU	Resource Utilization
NM	Number of Migrations
WT	Waiting Time

## References

1. Wang, S.; Xu, J.; Zhang, N.; Liu, Y. A survey on service migration in mobile edge computing. *IEEE Access* **2018**, *6*, 23511–23528. [[CrossRef](#)]

2. Chen, M.; Li, W.; Fortino, G.; Hao, Y.; Hu, L.; Humar, I. A Dynamic Service Migration Mechanism in Edge Cognitive Computing. *ACM Trans. Internet Technol. TOIT* **2019**, *19*, 30. [[CrossRef](#)]
3. Taneja, M.; Davy, A. Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017; pp. 1222–1228.
4. Abdah, H.; Barraca, J.P.; Aguiar, R.L. QoS-Aware Service Continuity in the Virtualized Edge. *IEEE Access* **2019**, *7*, 51570–51588. [[CrossRef](#)]
5. Kikuchi, J.; Wu, C.; Ji, Y.; Murase, T. Mobile edge computing based VM migration for QoS improvement. In Proceedings of the 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, 24–27 October 2017; pp. 1–5.
6. Machen, A.; Wang, S.; Leung, K.K.; Ko, B.J.; Salonidis, T. Live service migration in mobile edge clouds. *IEEE Wirel. Commun.* **2017**, *25*, 140–147. [[CrossRef](#)]
7. Wang, S.; Urgaonkar, R.; Zafer, M.; He, T.; Chan, K.; Leung, K.K. Dynamic service migration in mobile edge-clouds. In Proceedings of the 2015 IFIP Networking Conference (IFIP Networking), Toulouse, France, 20–22 May 2015; pp. 1–9.
8. Souza, V.B.; Masip-Bruin, X.; Marín-Tordera, E.; Sánchez-López, S.; Garcia, J.; Ren, G.J.; Ferrer, A.J. Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. *Future Gener. Comput. Syst.* **2018**, *87*, 1–15. [[CrossRef](#)]
9. Dashti, S.E.; Rahmani, A.M. Dynamic VMs placement for energy efficiency by PSO in cloud computing. *J. Exp. Theor. Artif. Intell.* **2016**, *28*, 97–112. [[CrossRef](#)]
10. Lucas-Simarro, J.L.; Moreno-Vozmediano, R.; Montero, R.S.; Llorente, I.M. Cost optimization of virtual infrastructures in dynamic multi-cloud scenarios. *Concurr. Comput. Pract. Exp.* **2015**, *27*, 2260–2277. [[CrossRef](#)]
11. Caglar, F.; Shekhar, S.; Gokhale, A.; Koutsoukos, X. Intelligent, performance interference-aware resource management for iot cloud backends. In Proceedings of the 2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI), Berlin, Germany, 4–8 April 2016; pp. 95–105.
12. Wang, Y.; Xia, Y. Energy optimal VM placement in the cloud. In Proceedings of the 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 27 June–2 July 2016; pp. 84–91.
13. Zhou, Z.; Abawajy, J.; Chowdhury, M.; Hu, Z.; Li, K.; Cheng, H.; Li, F. Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Gener. Comput. Syst.* **2018**, *86*, 836–850. [[CrossRef](#)]
14. Rodrigues, T.G.; Suto, K.; Nishiyama, H.; Kato, N. Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control. *IEEE Trans. Comput.* **2016**, *66*, 810–819. [[CrossRef](#)]
15. Liaqat, M.; Ninoriya, S.; Shuja, J.; Ahmad, R.W.; Gani, A. Virtual machine migration enabled cloud resource management: A challenging task. *arXiv* **2016**, arXiv:1601.03854.
16. Shaw, S.B.; Singh, A.K. Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center. *Comput. Electr. Eng.* **2015**, *47*, 241–254. [[CrossRef](#)]
17. Li, H.; Zhu, G.; Cui, C.; Tang, H.; Dou, Y.; He, C. Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. *Computing* **2016**, *98*, 303–317. [[CrossRef](#)]
18. Shaw, R.; Howley, E.; Barrett, E. An advanced reinforcement learning approach for energy-aware virtual machine consolidation in cloud data centers. In Proceedings of the 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, UK, 11–14 December 2017; pp. 61–66.
19. Ding, Y.; Qin, X.; Liu, L.; Wang, T. Energy efficient scheduling of virtual machines in cloud with deadline constraint. *Future Gener. Comput. Syst.* **2015**, *50*, 62–74. [[CrossRef](#)]
20. A-Shehri, H.A.; Hamdi, K. Multi-objective VM Placement Algorithms for Green Cloud Data Centers: An Overview. In Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–8.
21. Kumar, R.R.; Kumar, C. Designing an efficient methodology based on Entropy-TOPSIS for evaluating efficiency of cloud services. In Proceedings of the 7th International Conference on Computer and Communication Technology, ACM, Allahabad, India, 24–26 November 2017; pp. 117–122.

22. Skarlat, O.; Nardelli, M.; Schulte, S.; Dustdar, S. Towards qos-aware fog service placement. In Proceedings of the 2017 IEEE 1st international conference on Fog and Edge Computing (ICFEC), Madrid, Spain, 14–15 May 2017; pp. 89–96.
23. Gupta, M.K.; Amgoth, T. Resource-aware algorithm for virtual machine placement in cloud environment. In Proceedings of the 2016 Ninth International Conference on Contemporary Computing (IC3), Noida, India, 11–13 August 2016; pp. 1–6.
24. Donassolo, B.; Fajjari, I.; Legrand, A.; Mertikopoulos, P. Fog based framework for iot service provisioning. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6.
25. Jaumard, B.; Pouya, H. Migration plan with minimum overall migration time or cost. *J. Opt. Commun. Netw.* **2018**, *10*, 1–13. [[CrossRef](#)]
26. Velasquez, K.; Abreu, D.P.; Curado, M.; Monteiro, E. Service placement for latency reduction in the internet of things. *Ann. Telecommun.* **2017**, *72*, 105–115. [[CrossRef](#)]
27. Tang, Z.; Zhou, X.; Zhang, F.; Jia, W.; Zhao, W. Migration modeling and learning algorithms for containers in fog computing. *IEEE Trans. Serv. Comput.* **2018**. [[CrossRef](#)]
28. Shen, H.; Chen, L. Distributed autonomous virtual resource management in datacenters using finite-markov decision process. *IEEE ACM Trans. Netw.* **2017**, *25*, 3836–3849. [[CrossRef](#)]
29. Friedkin, N.E.; Proskurnikov, A.V.; Mei, W.; Bullo, F. Mathematical Structures in Group Decision-Making on Resource Allocation Distributions. *Sci. Rep.* **2019**, *9*, 1377. [[CrossRef](#)]
30. Ahmed, A.; Ibrahim, M. Analysis of energy saving approaches in cloud computing using ant colony and First Fit algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*. [[CrossRef](#)]
31. Rodrigues, T.G.; Suto, K.; Nishiyama, H.; Kato, N. A PSO model with VM migration and transmission power control for low Service Delay in the multiple cloudlets ECC scenario. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
32. Svärd, P.; Hudzia, B.; Walsh, S.; Tordsson, J.; Elmroth, E. Principles and performance characteristics of algorithms for live VM migration. *ACM SIGOPS Oper. Syst. Rev.* **2015**, *49*, 142–155. [[CrossRef](#)]
33. Nathan, S.; Bellur, U.; Kulkarni, P. Towards a comprehensive performance model of virtual machine live migration. In Proceedings of the Sixth ACM Symposium on Cloud Computing, Kohala Coast, HI, USA, 27–29 August 2015; pp. 288–301.
34. Khosravi, A.; Andrew, L.L.; Buyya, R. Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers. *IEEE Trans. Sustain. Comput.* **2017**, *2*, 183–196. [[CrossRef](#)]
35. Chen, Y.H.; Chen, C.Y. Service oriented cloud VM placement strategy for Internet of Things. *IEEE Access* **2017**, *5*, 25396–25407. [[CrossRef](#)]
36. Tian, H.; Wu, J.; Shen, H. Efficient algorithms for VM placement in cloud data centers. In Proceedings of the 2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Taipei, Taiwan, 18–20 December 2017; pp. 75–80.
37. Gao, C.; Wang, H.; Zhai, L.; Gao, Y.; Yi, S. An energy-aware ant colony algorithm for network-aware virtual machine placement in cloud computing. In Proceedings of the 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), Wuhan, China, 13–16 December 2016; pp. 669–676.
38. Grygorenko, D.; Farokhi, S.; Brandic, I. Cost-Aware VM Placement across Distributed DCs using Bayesian Networks. In Proceedings of the International Conference on the Economics of Grids, Clouds, Systems, and Services, Cluj-Napoca, Romania, 15–17 September 2015; Springer: Cham, Switzerland, 2015; pp. 32–48.
39. Zhang, W.; Han, S.; He, H.; Chen, H. Network-aware virtual machine migration in an overcommitted cloud. *Future Gener. Comput. Syst.* **2017**, *76*, 428–442. [[CrossRef](#)]
40. Zhao, D.; Yang, T.; Jin, Y.; Xu, Y. A service migration strategy based on multiple attribute decision in mobile edge computing. In Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 986–990.
41. Jiang, D.; Huang, P.; Lin, P.; Jiang, J. Energy efficient VM placement heuristic algorithms comparison for cloud with multidimensional resources. In Proceedings of the International Conference on Information Computing and Applications, Chengde, China, 14–16 September 2012; Volume 7473, pp. 413–420.
42. Wu, G.; Tang, M.; Tian, Y.C.; Li, W. Energy-efficient virtual machine placement in data centers by genetic algorithm. In Proceedings of the International Conference on Neural Information Processing, Doha, Qatar, 12–15 November 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 315–323.

43. Pascual, J.A.; Lorigo-Bostrán, T.; Miguel-Alonso, J.; Lozano, J.A. Towards a greener cloud infrastructure management using optimized placement policies. *J. Grid Comput.* **2015**, *13*, 375–389. [[CrossRef](#)]
44. Wang, S.; Liu, Z.; Zheng, Z.; Sun, Q.; Yang, F. Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers. In Proceedings of the 2013 International Conference on Parallel and Distributed Systems, Seoul, Korea, 15–18 December 2013; pp. 102–109.
45. Xiong, A.P.; Xu, C.X. Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center. *Math. Probl. Eng.* **2014**, *2014*, 816518. [[CrossRef](#)]
46. Beloglazov, A.; Abawajy, J.; Buyya, R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener. Comput. Syst.* **2012**, *28*, 755–768. [[CrossRef](#)]
47. Kuno, Y.; Nii, K.; Yamaguchi, S. A study on performance of processes in migrating virtual machines. In Proceedings of the 2011 Tenth International Symposium on Autonomous Decentralized Systems, Tokyo & Hiroshima, Japan, 23–27 March 2011; pp. 567–572.
48. Kikuchi, S.; Matsumoto, Y. Impact of live migration on multi-tier application performance in clouds. In Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, 24–29 June 2012; pp. 261–268.
49. Saaty, T.L. Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.* **2008**, *1*, 83–98. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).