

# A CLOUD SERVICE SELECTION MODEL BASED ON USER-SPECIFIED QUALITY OF SERVICE LEVEL

Chang-Ling Hsu

Department of Information Management, Ming Chuan University,  
Taoyuan, Taiwan  
johnny@mail.mcu.edu.tw

## ABSTRACT

*Recently, it emerges lots of cloud services in the cloud service market. After many candidate services are initially chosen by satisfying both the behavior and functional criteria of a target cloud service. Service consumers need a selection model to further evaluate nonfunctional QOS properties of the candidate services. Some prior works have focused on objective and quantitative benchmark-testing of QOS by some tools or trusted third-party brokers, as well as reputation from customers. Service levels have been offered and designated by cloud service providers in their Service Level Agreement (SLA). Conversely, in order to meet user requirement, it is important for users to discover their own optimal parameter portfolio for service level. However, some prior works focus only on specific kinds of cloud services, or require users to involve in some evaluating process. In general, the prior works cannot evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most either. Therefore, the aim of this study is to propose a cloud service selection model, CloudEval, to evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most. CloudEval applies a well-known multi-attribute decision making technique, Grey Relational Analysis, to the selection process. Finally, we conduct some experiments. The experimental results show that CloudEval is effective, especially while the quantity of the candidate cloud services is much larger than human raters can handle.*

## KEYWORDS

*Cloud Service, Cloud Service Selection, Multi-attribute Decision Model, Quality of Service, Cloud Computing*

## 1. INTRODUCTION

Emerging cloud computing and its application emphasize on lower Total Cost of Ownership (TCO), that pay as you go for the cloud services. It makes a user through the application of cloud services to reduce TCO and energy consumption. A cloud interactive model over Internet is composed of two parts: a cloud client and a cloud service. Common applications of a cloud client are such as web pages and mobile applications. As for categories of cloud services, NIST has defined three cloud service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1].

Recently, it emerges lots of cloud services in the cloud service market. Enterprises need to select suitable cloud services effectively to meet the requirements of an enterprise information systems and their integration. If a suitable cloud service could be integrated into an enterprise information system, the quality of the information system would be better than one with an un-suitable cloud service. Khezrian et al. [2] deem that two significant tasks in the process of a selection model are selection and ranking in which every solution for them is affected directly on description of services. During describing a service, three properties have to be considered: behavior, functional, and nonfunctional. The candidate services are initially chosen by satisfying both the behavior and functional criteria of a target cloud service. As there are often many cloud services that meet the functional and behavior requirements, the cloud service selection uses some criteria to select the optimal service. However, in real world practice, there are too large a number of possible candidate cloud services to select manually. Thus, service consumers need a selection model to further evaluate nonfunctional properties of the candidate services, such as Quality of Service (QOS), price and reputation.

Besides, service levels have been offered and designated by many cloud service providers in their Service Level Agreement (SLA). Conversely, in order to meet user requirement, it is important for users to discover their own optimal parameter portfolio for service level. It also depends on selection criteria of cloud services. And, the criteria of cloud service have dynamic, invisible, variant, subjective and vague characteristics. Therefore, it is a multi-attribute decision-making problem about discovering the optimal parameter portfolio. Still, cloud providers and third-party brokers have not had the selection and recommendation mechanisms. The criteria in previous researches focus on benchmarks by a trusted third-party broker, such as CloudHarmony [3], based on objective and quantitative testing of QOS, as well as reputation from customers. The criteria in previous researches focus on benchmarks by some tools (i.e. CloudCmp [4] and vCenter Hyperic [5]), or third-party brokers (i.e. CloudHarmony) based on objective and quantitative testing of QOS, as well as reputation from customers.

However, some prior works [6, 7] focus only on specific kinds of cloud services, or require users to involve in some evaluating process [7, 8]. In general, the prior works cannot evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most either. Therefore, based on user-specified QOS requirement for service level, and the objective and quantitative benchmarks, the aim of this study is to propose a new cloud service selection model, CloudEval (**Cloud Evaluator**), to evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most. CloudEval applies a well-known multi-attribute decision making technique, i.e. Grey Relational Analysis, to the selection process.

The remainder of this paper is organized as follows. Section 2 reviews the related research work. Section 3 describes CloudEval. Section 4 presents the experiments and an analysis of the results. Finally, Section 5 draws conclusions.

## **2. RELATED WORK**

### **2.1. Selection Models**

In view of the limits of the prior works that we have mentioned above, now we explained about why they cannot evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most. First, some prior works [6, 7] focus only on specific kinds of cloud services (such as SaaS, web server service, or cloud storage service) neglect wider scope benchmarks from the broker, and financial or service credit in SLA from providers need to be further quantified and integrated during the selection. Secondly, some prior

works related to AHP [7] or Fuzzy methods [8] with aggregated subjective and objective criteria need users to participate the evaluating process. They require the users to specify a preference manually for each decision alternative on each criterion. Thirdly, prior works [7, 8] rank the priorities of the candidate service list of alternatives calculated on only goals rather than calculated on both goals and user-specified service level.

## 2.2. Benchmarks and Attributes of Cloud Services

Benchmarking techniques have been used to measure the performances of the system components of a cloud service. The system components can be CPUs, storage services, server services, network and applications running in physical and virtual environment. Li et al. [4] indicate that recent benchmarking measurements are limited in scope; none of them cover enough of the dimensions (e.g., compute, storage, network, scaling) to yield meaningful conclusions. Further, some of the measurement methodologies do not extend to all providers, for example, they would not work for PaaS providers. Recently, CloudHarmony is a popular third-party trusted broker which offers both IaaS and PaaS of public cloud monitoring and benchmarking service [3]. The attributes of cloud service data that CloudHarmony has offered are availability, response time, system performances and network performances. As CloudHarmony has covered enough of the dimensions for measuring a cloud service, we adopt its cloud service data related to the attributes as part of the input data source of our selection model, CloudEval.

The criteria in prior works focus on benchmarks by some tools or third-party brokers based on objective and quantitative testing of QOS, as well as reputation from customers. The attributes of selection criteria that Menzel & Ranjan [7] have proposed used in their framework, CloudGenius, are price, maximum network latency, average network latency, performance, uptime (i.e. availability) and popularity (i.e. reputation). The Performance attribute has three subattributes, CPU, RAM and Disk performance. The attributes of criteria that Kalepu et al. have proposed are reputation and availability [9]. Considering the designing attributes of the criteria for CloudEval based on user-specified QOS requirement for service level, and the objective and quantitative benchmarks, we add two attributes, response time (as speed of service from CloudHarmony) and financial credit from SLA. In this paper, we design seven main attributes: availability, response time, price, reputation (as user rating), network performance (as network latency), system performance (as performance), and financial credit. Furthermore, we extend the subattributes of the network performance attribute and the system performance attribute to include all the benchmark testing in CloudHarmony.

## 2.3. Grey System theory and Grey Relational Analysis

Grey System theory has been widely applied to handle information concerned the systems that do not have enough information or is unknown. Deng indicates that the aims of Grey System theory are to provide theory, techniques, notions and ideas for resolving (analyzing) latent and intricate systems. The Grey Relational Space (GRS) and Grey Generating Space (GGS) are the essential contents and topics of Grey System theory [10]. Based on GRS and GGS, In addition, Grey Relational Analysis (GRA) in Grey System theory has been widely applied to analyzing multivariate data for decision making [10, 11]. GRA ranks alternatives, represented as compared sequences, by their nearness to the ideal criteria, represented as a referenced sequence.

GRA reflects a form of fuzzification of inputs, and uses different calculations, to include different calculation of norms [12]. Thus, GRA uses Grey Relational Generation (GRG) method to map all the data into GRS by normalizing disorderly raw data. Sallehuddin et al. indicate that the raw data can be turned to a regular list for the benefit of grey modelling, transferred to a dimensionless list in order to obtain an appropriate fundamental for grey analyzing and changed into a unidirectional

series for decision making [11]. GRA calculates a Grey Relational Coefficient (GRC) for each dimension (i.e. attribute), and then it calculates a grey relational grade by averaging all GRCs of each dimension for each compared sequence of the dimensionless list. Above all, GRA is simple, practical, and demands less precise information than other methods. Therefore, we adopt GRA method to select the optimal service which satisfies user-specified service level most.

### 3. THE CLOUD SERVICE SELECTION MODEL

The stakeholders of CloudEval are users, third-party brokers of cloud service and cloud service providers. The design of data sources for CloudEval is SLAs from providers and any trusted third-party brokers, such as CloudHarmony, which offers reputation of providers, some objective and quantitative benchmark-testing data. CloudEval consists of two components: selection process and data structure. We describe both components respectively as follows.

#### 3.1. Selection Process

We apply the well-known multi-attribute decision making technique, Grey Relational Analysis, to our selection process. The service selection process is as follows.

1st step. Setting user selection criteria, goals and their weights: a user sets one's selection criteria of cloud service, acting as a referenced sequence in GRA, and sets weight and goal for each attribute. The goals are represented with preference for value of an attribute of the selection criteria.

2nd step. Normalizing the candidate list: we normalize each cloud service acting as a compared sequence of the candidate list in GRG method.

3rd step. Calculating Grey Relational Coefficient (GRC) of the attributes of each service: we use Deng's method [13] to calculate all GRCs of the attributes of each cloud service based on the comparison between each compared sequence and the referenced sequence.

4th step. Calculating grey relational grade for each service: we calculate a grey relational grade for each cloud service by averaging all the grey relational coefficient of each attribute. As for the way of averaging all the grey relational coefficient, we use both Deng's equal-weighted average method [13] and weighted average method.

5th step. Ranking the list: we rank the candidate list by ordering grey relational grade of each service. Finally, we choose the largest grey relational grade in the ranked list as the optimal service which satisfies user-specified service level most.

#### 3.2. Data structure

Each cloud service of provider  $j$  is a compared sequence,  $X[j] = (x_1, x_2, \dots, x_m) \in \text{Domain}(A_1) \times \dots \times \text{Domain}(A_i) \times \dots \times \text{Domain}(A_m)$ , where  $j = 1..n$ .  $X[0]$  is a referenced sequence in GRA. Both  $X[0]$  and  $X[j]$  have a fixed-length vector with attribute-value pairs of a data instance,  $A_i$  is an attribute of  $X$ ,  $i = 1..m$ . As mentioned in Section 2.2, we have designed the seven main attributes of selection criteria. The attributes availability, response time, network performance, system performance and financial credit are QOS-related and the attribute user rating and price are not QOS-related. As for the goals for each attribute of the selection criteria, the bigger the better are the attributes *availability*, *user rating*, *network performance*, *system performance* and *financial credit*; the less the better are the attributes *response time* and *price*. We design the attributes in detail as follows.

### 3.2.1. Availability

It is also known as the uptime status, the percentage of available connectivity time during a time period from a service provider. When a remote cloud service is usable as a user connects the service online, we call the time *connectivity time*; When a remote cloud service is unusable, it might be offline, under maintenance, shutdown, breakdown or instable as a user connects the service online. We call the time *connectivity downtime*. We define *availability* as:

$$availability = (\text{connectivity Time} - \text{connectivity downtime}) / \text{connectivity time} \quad (1)$$

### 3.2.2. Response Time

It is also called as round trip time or speed to represent the time duration between sending out a request and receiving a response to a service user. Its measured unit maybe second or millisecond.

### 3.2.3. User Rating

Some customers have rated each cloud service according to their usage experiences of the service on some broker websites. It is often rated from 0 to 5. A user refers to the rating as a social reputation.

### 3.2.4. Price

Due to budget limit, a user will consider the announced pricing published by a provider. The pricing is on a per-use basis, which maybe per minute, hour, day, or month, under different system environment configurations of instance type of a cloud service. The environment configurations could contains number of CPU cores, size of RAM and storage, network throughput, etc.

### 3.2.5. Network Performance

Suppose each benchmark item of network performance of a cloud service collected from a broker,  $n[i, j]$ , where  $i$  is the  $i$ -th benchmark item, and  $j$  is the  $j$ -th provider. The measured unit of each benchmark item is MBS (Mega-Bits per Second) for throughput. Due to metric of each benchmark item has different value range, we set the scale of the network performance from 0 to 1. Thus, we normalize each benchmark item as:

$$net[i, j] = (n[i, j] - \min\_thput[i]) / (\max\_thput[i] - \min\_thput[i]) \quad (2)$$

where  $\max\_thput[i]$ : the maximum average summary network performance among all the  $i$ -th benchmark items of each provider,  $\min\_thput[i]$ : the minimum average summary network performance among all the  $i$ -th benchmark items of each provider. Then, we calculate the average summary network performance for the  $j$ -th provider by weighted average scoring as:

$$avg\_net\_scores[j] = \sum_{i=1}^n w[i] * net[i, j] \quad (3)$$

where  $w[i]$ : the user-specified weight of the  $i$ -th benchmark item.

### 3.2.6. System Performance

Suppose each benchmark item of system performance of a cloud service for a provider from a broker,  $s[i, j]$ , where  $i$  is the  $i$ -th benchmark item and  $j$  is the  $j$ -th provider. Due to metric of each

benchmark item has different measured unit and value range, for instance, it is IOPS (Input/Output Operations Per Second) for disk IO; there are ECUs (EC2 Compute Unit) [3] or CCUs (CloudHarmony Compute Unit) [3] for CPU performances. We set the scale of all the system performances from 0 to 1. Thus, we normalize each benchmark item as:

$$sys[i, j] = (s[i, j] - min\_val[i]) / (max\_val[i] - min\_val[i]) \quad (4)$$

where  $max\_val[i]$ : the maximum average summary system performance in all the  $i$ -th benchmark items of each provider,  $min\_val[i]$ : the minimum average summary system performance in all the  $i$ -th benchmark item of each provider.

Then, we calculate the average summary system performance for the  $j$ -th provider by weighted average scoring as:

$$avg\_sys\_scores[j] = \sum_{i=1}^n w[i] * sys[i, j] \quad (5)$$

where  $w[i]$ : the user-specified weight of the  $i$ -th benchmark item.

### 3.2.7. Financial Credit

It means that percentage of monthly bill (i.e. credit rate) for covered a cloud service or service credit which does not meet the availability level in SLA from a provider that will be credited to future monthly bills of customer [14, 15]. Each credit rate is counted on an availability interval. As each interval of provider may be different from a similar user-specified interval, for example, the availability intervals, [99.00%, 99.95%) vs. [98.00%, 99.96%) are shown in Table 2(a) and Table 1 respectively. For comparability of both the credit rates in different availability intervals, we design an algorithm, adjust-interval-credit as shown in Figure 1, to adjust each pair of the original credit of a provider: ([99.00%, 99.95%), 10%) as shown in Table 2(a) into pair of the adjusted credit: ([98.00%, 99.96%), 17.65%) as shown in Table 2(b).

In the algorithm, suppose each interval of financial credit or service credit of a cloud service in SLA from a provider,  $cred[k, j]$ , where  $k$  is the  $k$ -th availability interval and  $j$  is the  $j$ -th provider. The measured unit of each interval is credit rate for a user-specified interval of monthly availability percentage. The scale of the credit is from 0 to 1. The descriptions of some important symbols of  $cred[k, j]$  are specified as: (1) length: the length of an interval; (2) upperBound: the upper bound of an interval; (3) lowerBound: the lower bound of an interval; (4) newUpperBound: the adjusted upper bound of an interval; (5) newLowerBound: the adjusted lower bound of an interval; (6) upperBound. $\Delta$ : a new interval between upperBound and newUpperBound; (7) lowerBound. $\Delta$ : a new interval between lowerBound and newLowerBound; (8) upperBound. $\Delta$ .rate: the rate of upperBound. $\Delta$ ; (9) upperBound. $\Delta$ .length: the length of upperBound. $\Delta$ ; (10) newLength: the new length of an adjusted interval; (11) middle.length: the adjusted interval between lowerBound. $\Delta$  and upperBound. $\Delta$ .

Table 1. A user-specified financial credit list.

Interval No.	Availability Interval (Monthly)	Credit Rate
1	[98.00%, 99.96%)	10%
2	[94.00%, 98.00%)	25%
3	[0, 94.00%)	50%

Table 2. The adjusted credit based on the original credit.

	Original Credit List (a)		Adjusted Credit List (b)	
Interval No.	Availability Interval	Credit Rate	Availability Interval	Credit Rate
1	[99.00%, 99.95%)	10%	[98.00%, 99.96%)	17.65%
2	[95.00%, 99.00%)	25%	[94.00%, 98.00%)	31.25%
3	[0, 95.00%)	50%	[0, 94.00%)	50.00%

**Algorithm** *adjust-interval-credit*( $j, n, \text{user-credit-list}, \text{provider-credit-list}$ )

**Input:**  $j$ : the  $j$ -th provider;  $n$ : the number of user-specified availability intervals; *user-credit-list*: a list of each user-specified pair, (interval  $k$ , credit rate  $c$ ); *provider-credit-list*: a list of each pair of a service, (interval  $k$ , credit rate  $c$ ).

**Output:** an adjusted list of pair of a service, (interval  $k$ , credit rate  $c$ ).

- (1) initialize all the required variables;
- (2)  $\text{cred}[1, j].\text{length} = \text{cred}[1, j].\text{upperBound} - \text{cred}[1, j].\text{lowerBound}$ ;
- (3) **For** each interval of *provider-credit-list* and *user-credit-list*,  $k = 1$  to  $n$  **do loop**
- (4)  $\text{cred}[k+1, j].\text{length} = \text{cred}[k+1, j].\text{upperBound} - \text{cred}[k+1, j].\text{lowerBound}$ ;
- (5)  $\text{cred}[k, j].\text{newUpperBound} = \text{user}[k].\text{upperBound}$ ;
- (6) **If** ( $\text{cred}[k, j].\text{upperBound} < \text{user}[k].\text{upperBound}$ ) **Then**  
 $\text{cred}[k, j].\text{upperBound}.\Delta.\text{rate} = \text{cred}[k, j].\text{rate}$ ;  
 $\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} = \text{user}[k].\text{upperBound} - \text{cred}[k, j].\text{upperBound}$ ;  
 $\text{cred}[k, j].\text{newLength} = \text{cred}[k, j].\text{length} + \text{cred}[k, j].\text{upperBound}.\Delta.\text{length}$ ;
- (7)  $\text{cred}[k, j].\text{newLowerBound} = \text{user}[k].\text{lowerBound}$ ;
- (8) **If** ( $\text{cred}[k, j].\text{lowerBound} > \text{user}[k].\text{lowerBound}$ ) **Then**  
 $\text{cred}[k, j].\text{lowerBound}.\Delta.\text{rate} = \text{cred}[k+1, j].\text{rate}$ ;  
 $\text{cred}[k, j].\text{lowerBound}.\Delta.\text{length} = \text{cred}[k, j].\text{lowerBound} - \text{user}[k].\text{lowerBound}$ ;  
 $\text{cred}[k, j].\text{newLength} = \text{cred}[k, j].\text{newLength} + \text{cred}[k, j].\text{lowerBound}.\Delta.\text{length}$ ;
- (9) **If** ( $\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} < 0$  and  $\text{cred}[k, j].\text{lowerBound}.\Delta.\text{length} \geq 0$ ) **Then**  
 $\text{cred}[k, j].\text{middle.length} = \text{cred}[k, j].\text{length} + \text{cred}[k, j].\text{upperBound}.\Delta.\text{length}$ ;  
 $\text{cred}[k, j].\text{newRate} = (\text{cred}[k, j].\text{middle.length} / \text{cred}[k, j].\text{newLength}) * \text{cred}[k, j].\text{rate} +$   
 $(\text{cred}[k, j].\text{lowerBound}.\Delta.\text{length} / \text{cred}[k, j].\text{newLength}) * \text{cred}[k, j].\text{lowerBound}.\Delta.\text{rate}$ ;
- (10) **If** ( $\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} < 0$  and  $\text{cred}[k, j].\text{lowerBound}.\text{rate}.\Delta.\text{length} < 0$ ) **Then**  
 $\text{cred}[k, j].\text{middle.length} = \text{cred}[k, j].\text{length} + \text{cred}[k, j].\text{upperBound}.\text{rate}.\Delta.\text{length} +$   
 $\text{cred}[k, j].\text{lowerBound}.\text{rate}.\Delta.\text{length}$ ;  
 $\text{cred}[k, j].\text{newRate} = \text{cred}[k, j].\text{rate}$ ;
- (11) **If** ( $\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} \geq 0$  and  $\text{cred}[k, j].\text{lowerBound}.\text{rate}.\Delta.\text{length} < 0$ ) **Then**  
 $\text{cred}[k, j].\text{middle.length} = \text{cred}[k, j].\text{length} + \text{cred}[k, j].\text{lowerBound}.\text{rate}.\Delta.\text{length}$ ;  
 $\text{cred}[k, j].\text{newRate} = (\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} / \text{cred}[k, j].\text{newLength}) *$   
 $\text{cred}[k, j].\text{upperBound}.\Delta.\text{rate} + (\text{cred}[k, j].\text{middle.length} / \text{cred}[k, j].\text{newLength}) * \text{cred}[k, j].\text{rate}$ ;
- (12) **If** ( $\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} \geq 0$  and  $\text{cred}[k, j].\text{lowerBound}.\text{rate}.\Delta.\text{length} \geq 0$ ) **Then**  
 $\text{cred}[k, j].\text{newRate} = (\text{cred}[k, j].\text{upperBound}.\Delta.\text{length} / \text{cred}[k, j].\text{newLength}) *$   
 $\text{cred}[k, j].\text{upperBound}.\Delta.\text{rate} + (\text{cred}[k, j].\text{length} / \text{cred}[k, j].\text{newLength}) * \text{cred}[k, j].\text{rate} +$   
 $(\text{cred}[k, j].\text{lowerBound}.\Delta.\text{length} / \text{cred}[k, j].\text{newLength}) * \text{cred}[k, j].\text{lowerBound}.\Delta.\text{rate}$ ;
- (13) **End For Loop**
- (14) return the list of each ( $\text{newLowerBound}$ ,  $\text{newUpperBound}$ ,  $\text{newRate}$ ) of  $\text{cred}[k, j]$  of the service;

Figure 1. The *adjust-interval-credit* algorithm

After adjusting the intervals and credits of a service, we calculate the average credit rate for the  $j$ -th provider by weighted average method as:

$$avg\_credit\_scores[j] = \sum_{k=1}^n w[k] * cred[k, j] \quad (6)$$

, where  $w[k]$ : the user-specified weight of the  $k$ -th availability interval.

## 4. EXPERIMENTS

### 4.1. Design of experiments

We have conducted two groups of experiments. One group is for the comparison between CloudEval with adopting weighted attribute and human raters with adopting weighted attribute. The other group is for the comparison between CloudEval without adopting weighted attribute and human raters with adopting weighted attribute. The experimental steps have been conducted according to the process of CloudEval mentioned in Section 3.1. Besides, we have invited six raters to select by ranking the sample services manually. CloudEval used two toolboxes of GRG and GRC, mentioned in Section 2.3 and 3.1, written in MATLAB by [16]. Both the toolboxes were processed in MATLAB 7.0.1 for each experiment.

According to Table 3, we first generated a synthetic data set as shown in Table 4 with the seven attributes. All experiments use the dataset, in which were simulated as data from the broker, CloudHarmony, and SLA. The dataset is used as the input sample data for CloudEval and the raters in each experiment. The sample size of the data set is 30, numbered from X0 to X30. Service, X0, is the referenced sequence, whose values are the selection criteria of a cloud service; and all the services from X1 to X30 are the compared sequences, searched by CloudEval and the raters.

For evaluating the effectiveness of the experimental results, this study adopts the commonly used indicators, i.e. *Pearson correlation coefficient* (represented as  $\rho$ ) and *Spearman's rank correlation coefficient* (represented as  $\gamma_s$ ) to evaluate and compare the correlation between the rank lists of cloud services selected by the raters and by CloudEval. And, all the *correlation coefficients* were processed in PASW Statistics 18 (formerly SPSS Statistics) for each experiment.

Table 3. The attributes of experimental dataset.

Attributes	Value	Attributes	Value
<i>Id</i> (A0)	The identifier of the cloud services	<i>Price</i> (A4)	Randomly sampling data, normally distributed N(700, 350) from 1 cents to 1,500 cents
<i>Availability</i> (A1)	Randomly sampling data, uniformly distributed from 0.9 to 1	<i>network performance</i> (A5)	Randomly sampling data, normally distributed N(3, 1.5) from 0 to 5
<i>response time</i> (A2)	Randomly sampling data, normally distributed N(15, 6) from 1 to 30 seconds	<i>system performance</i> (A6)	Randomly sampling normally distributed N(3.2, 0.8) from 0 to 5
<i>user rating</i> (A3)	Randomly sampling data, normally distributed N(3, 1) from 0 to 5, increased by 0.5.	<i>financial credit</i> (A7)	Randomly sampling data, normally distributed N(2.5, 1) from 0 to 5

Table 4. The experimental dataset.

Cloud Service Id	Main Attributes						
	A1	A2	A3	A4	A5	A6	A7
X0	0.9500	15.0000	3	700	4.0000	4.0000	3.0000
X1	0.9806	13.5264	2.5	1000	2.8124	4.1928	3.1019
X2	0.9253	15.8717	2	900	3.2980	3.6547	3.5741
X3	0.9250	18.4351	2	600	2.8198	3.6701	4.3903
X4	0.9458	13.6599	4	1400	2.4443	4.2056	1.8874
X5	0.9081	9.7423	0	600	4.8067	3.0971	2.4509
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
X29	0.9109	18.2752	0.5	1000	4.0468	4.7359	2.9475
X30	0.9279	7.0281	5	300	1.9785	1.9483	2.5362

## 4.2. Experimental Results

The experimental results in Table 5 show that comparison of both the groups' correlation coefficients between the rank lists of cloud services selected by the raters and by CloudEval. As the sample size is 30, large enough, the values of both the correlation coefficients  $\rho$  and  $\gamma_s$  are the same values. Thus, we only illustrate *Spearman's rank correlation coefficient*,  $\gamma_s$ . At the significance level of  $\alpha = 0.01$ , all the experiments of  $\gamma_s$  in Table 5 illustrate that all the Bivariate Correlation tests are significantly different from zero in  $\rho$  between the rank-lists of user and CloudEval. It indicates that both groups of CloudEval are considerably correlative to the experimental results of the raters. As the average  $\gamma_s$  are increased from 0.6796 into 0.6952, it shows that CloudEval adopting weighted attribute can really improve both *Pearson correlation coefficients* and *Spearman's rank correlation coefficients* of CloudEval without adopting weighted attribute.

The experimental results in Table 6 show that each optimal service id, selected by user with adopting weighted attribute, by CloudEval without adopting weighted attribute, or by CloudEval with adopting weighted attribute. Both the groups of CloudEval have selected X9 as the optimal service. As for comparing with user's selection, three of the six raters have selected the same optimal service X9 as CloudEval have done; three of them have selected X26 or X27 as the optimal service. As the majority of the raters select the optimal services same as CloudEval, it shows that they are considerably correlative.

After all the discussion of the results above, therefore, we can say that CloudEval is effective, especially while the quantity of the candidate cloud services is much larger than human raters can handle.

Table 5. Comparison of both the groups' correlation coefficients.

	Cloud Service Selection Model	
	without weighted attribute	with weighted attribute
User ID	<i>Spearman's <math>\rho</math> coefficient</i>	<i>Spearman's <math>\rho</math> coefficient</i>
U1	0.7557**	0.7557**
U2	0.5205**	0.5272**
U3	0.6684**	0.6858**
U4	0.8700**	0.8752**
U5	0.6533**	0.6958**
U6	0.6093**	0.6315**
<b>Average</b>	<b>0.6796</b>	<b>0.6952</b>

\*\* : p-value < 0.01.

Table 6. The optimal service ids selected by the experiments.

User ID	The Optimal Service Id		
	by user with weighted attribute	by <i>CloudEval</i> without weighted attribute	by <i>CloudEval</i> with weighted attribute
U1	X9	X9	X9
U2	X27	X9	X9
U3	X9	X9	X9
U4	X9	X9	X9
U5	X26	X9	X9
U6	X26	X9	X9

## 5. CONCLUSIONS

For solving the problem of discovering a user's optimal parameter portfolio for service level and evaluating the nonfunctional properties of any kind of candidate cloud services, we have proposed the cloud service selection model, *CloudEval*, to evaluate the nonfunctional properties and select the optimal service which satisfies both user-specified service level and goals most. And, *CloudEval* adopting weighted attribute can improve the correlation with a rater's selection of *CloudEval* without adopting weighted attribute.

The design of data sources for *CloudEval* is SLAs from providers and any trusted third-party broker, such as *CloudHarmony*, which offers user rating, some objective and quantitative benchmark-testing data. We recommend *CloudEval* which will easily offering applications for industrial users to select any cloud services through real data from a trusted third-party broker, as well as price and SLA data from cloud service providers. For future work, as users feel more comfortable to use fuzzy concept to weight among attributes, we will combine fuzzy technique with grey relational analyzing technique for the weighting. In additions, we also plan to adapt *CloudEval* more automatically for users to apply it over the Internet.

## ACKNOWLEDGEMENTS

This work was supported in part by the Taiwan Ministry of Science and Technology under Grant No. NSC-102-2218-E-130-001.

## REFERENCES

- [1] Mell, P. and Grance, T., The NIST Definition of Cloud Computing, Special Publication 800-145. National Institute of Standards and Technology (NIST), U.S.: Department of Commerce, Sept. 2011, pp. 2-3.
- [2] Khezrian, M., Kadir, W. M. N. W., Ibrahim, S., Mohebbi, Munusamy, K., K., and Tabatabaei, S. G. H., "An evaluation of state-of-the-art approaches for web service selection," in Proc. Int. Conf. on Information Integration and Web-based Applications & Services (iiWAS'10), Paris, France, pp. 885-889, November 2010.
- [3] CloudHarmony, The CloudHarmony website, CloudHarmony Inc., December 2013. [Online]. Available: <https://cloudharmony.com>
- [4] Yang, A. Li, Kandula, X., S., and Zhang, M., "CloudCmp: comparing public cloud providers," in Proc. Int. Conf. on Internet Measurement Conference (IMC'10), Melbourne, Australia, pp. 1-14, November 2010.
- [5] VMware, VMware vCenter Hyperic, VMware Inc., April 2014. [Online]. Available: <http://www.vmware.com/products/vcenter-hyperic/>
- [6] Godse, M. and Mulik, S., "An approach for selecting Software-as-a-Service (SaaS) product," in Proc. IEEE Int. Conf. on Cloud Computing (CLOUD'09), Bangalore, India, pp. 155-158, September 2009.
- [7] Menzel, M. and Ranjan, R., "CloudGenius: decision support for web server cloud migration," in Proc. Int. Conf. on World Wide Web (WWW'12), Lyon, France, pp. 979-988, April 2012.
- [8] Qu, L., Wang, Y., and Orgun, M. A., "Cloud Service Selection Based on the Aggregation of User Feedback and Quantitative Performance Assessment," in Proc. IEEE Int. Conf. on Services Computing (SCC'13), Santa Clara Marriott, CA, USA, pp.152-159, June 28-July 3 2013.
- [9] S. Kalepu, S. Krishnaswamy, and S. Loke, "Verity: a QOS metric for selecting web services and providers," in Proc. Int. Conf. on Web Information Systems Engineering Workshops (WISE'03), Rome, Italy, pp.131-139, December 2003.
- [10] Deng, J., "Introduction to Grey System Theory," The Grey System Journal, vol. 1, pp. 1-24, 1989.
- [11] Sallehuddin, R., Shamsuddin, S. M. H., Hashim, S. Z. M., "Application of grey relational analysis for multivariate time series," in Proc. Int. Conf. on Intelligent Systems Design and Applications (ISDA'08), vol. 2, Kaohsiung, Taiwan, pp. 432-437, November 2008.
- [12] Muley, A. and Bajaj, V. H., "Applications of fuzzy multiple attribute decision making method solving by interval numbers," Advances in Computational Research, vol. 2, no. 1, pp. 1-5, 2010.
- [13] Deng, J., "Control problems of Grey Systems," Systems and Control Letters, vol. 5, pp. 288-294, 1982.
- [14] Google, Google Cloud Platform, Google Inc., March 2014. [Online]. Available: <https://cloud.google.com/products/>
- [15] Microsoft, Windows Azure Cloud Services, Virtual Machines, and Virtual Network, Service Level Agreement (SLA). U.S.: Microsoft, March 2014, pp. 1-6. [Online]. Available: <http://www.microsoft.com/en-us/download/details.aspx?id=38427>
- [16] Wnag, C.-W., Yeh, C- K., Wen, K.-L., and Jwo, W.-S., "The development of completed grey relational grade toolbox via Matlab," in Proc. IEEE Int. Conf. on Cybernetics & Intelligent Systems (CIS'06), Bangkok, Thailand, pp. 428-433, June 2006.

**AUTHOR**

Chang-Ling Hsu received the B.B.A. degree in Computer Science from Soochow University, Taiwan in 1989, the M.E. degree in Information and Electronic Engineering from National Central University, Taiwan in 1991, and the Ph.D. degree in Information Management from National Central University, Taiwan in 2004. He is currently an assistant professor at Department of Information Management in Ming Chuan University, Taiwan. His research interests include data mining, information retrieval and information system architecture.

