Collaborative Optimization of Service Composition for Data-Intensive Applications in a Hybrid Cloud

Hua Ma, *Member, IEEE*, Haibin Zhu, *Senior Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Wensheng Tang

Abstract—The multi-valued evaluations of quality of service (QoS), the complicated constraints between cloud services (CSs) and the collaborative resource assignments add many difficulties to the problem of **CS c**omposition for **d**ata-intensive applications (DiA) in a **h**ybrid **c**loud (CSCD-HC). Solving the CSCD-HC problem has become a challenging task due to the uncertain QoS, the diverse hardware configurations and the flexible pricing about CSs. This paper proposes a collaborative optimization approach for CSCD-HC. This approach models a DiA as a role-based collaboration (RBC) system and employs the **e**nvironments—**c**lasses, **a**gents, **r**oles, **g**roups, and **o**bjects (E-CARGO) model to formalize the CSCD-HC problem with complicated constraints. To deal with the multi-valued QoS evaluations, this paper exploits the cloud model theory to analyze the performance of CSs, and presents a new method utilizing the Mahalanobis distance to improve the similarity calculation of QoS cloud models. Based on it, the qualification of candidate CSs can be precisely measured for supporting CS composition. A solution via the IBM ILOG CPLEX optimization package is put forward to solve the CSCD-HC problem. The experimental results demonstrate that the proposed approach is effective and feasible for optimizing CSCD-HC.

Index Terms—Collaboration optimization, data-intensive, hybrid cloud, multi-valued QoS evaluations, service composition

1 Introduction

1.1 Motivation

With the increasing computation complexity and data scale, data-intensive applications (DiAs) have had the urgent needs for high performance computation and massive data storage to solve the challenging problems, such as DNA computing, astronomical observation and earthquake prediction [1], [2], [3]. A DiA system over big data completely depending on private infrastructures is too expensive. The exploitation of public cloud services (CSs) is appealing due to its costs reduction and resource elasticity [4]. For example, the data captured by image sensors is usually partitioned into the sensitive data (<20 percent) and the insensitive data (>80 percent), and the substantial cost can be saved when the latter is stored in public CSs [5]; A DiA with dynamic workload experiences the flash crowd load at rare time (e.g., the 5 percent-percentile heavy load time), and the hybrid cloud can provide the service provisioning in a cost-effective way [6], [7]. Now, the hybrid cloud integrating CSs from the private and public clouds has been a promising computing paradigm [8], [9]. Many enterprises and organizations (e.g., OpenText, Oxford University and SEGA) have successfully harnessed the hybrid cloud for their DiAs [10]. The leading CS providers (CSPs) (e.g., IBM[1], Cisco[2] and Tencent[3]) are devoting to helping users construct their hybrid clouds. However, in a hybrid cloud, how to achieve the CS composition optimization for a DiA consisting of multiple computation or storage tasks, with the consideration of quality of service (QoS) and cost, is still an open issue. This issue is facing a series of challenges as follows:

(1) The multi-valued QoS evaluations make it hard to objectively assess a CS's performance for DiAs in a hybrid cloud. The CS resources in a private cloud are limited and usually used to store or process the sensitive or critical data. Public cloud could provide enough CSs for any organization in theory. Unlike the reliable and stable private CSs, the QoS of public CSs is uncertain and dynamic due to the vulnerability of Internet and the diversity of user features [11], [12]. The quality of experience (QoE) of a public CS is usually different from its QoS declared by the CSP [13], [14]. Accurately predicting the QoS of public CSs has been a challenging problem due to the dynamic cloud environment. In a hybrid cloud, a DiA is composed of both private CSs and public CSs. The QoS of a private

• H. Ma and W. Tang are with the College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China. E-mail: {huama, tangws}@hunnu.edu.cn.
• H. Zhu is with the Collaborative Systems Laboratory, Nipissing University, North Bay, ON P1B8L7, Canada. E-mail: haibinz@nipissingu.ca.
• K. Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561. E-mail: lik@newpaltz.edu.

Manuscript received 17 June 2017; revised 24 Nov. 2017; accepted 26 Oct. 2018. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Haibin Zhu.)
Recommended for acceptance by Y. Lu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2018.2879603

1. www-01.ibm.com/software/cn/middleware/application-platform/hybrid-cloud.html
2. www.cisco.com/c/zh_cn/solutions/cloud/hybrid-cloud.html
3. www.qcloud.com/solution/hybridcloud

CS is definite and described easily with a single real number, while the QoS of a public CS may be uncertain and depicted with an interval number or the time series data obtained by continuous monitoring [11], [15]. Thus, how to objectively assess the overall performance of CS composition based on the multi-valued QoS evaluations [15] consisting of the single number, interval number and time series data, is a key problem for obtaining the optimal CS composition for a DiA in a hybrid cloud.

(2) With the increasing CSs from a hybrid cloud integrated into DiAs, the complicated constraints about CSs add more difficulties for optimizing CS composition. Recently, CSPs have constantly published diverse CSs with the same or similar functions. For example, Amazon EC2[4] provides dozens of products aiming at various requirements. The QoS and prices are not the same for different products. In a hybrid cloud, the CSs from different CSPs need to be selected and integrated into a DiA. To obtain the best execution performance and customer satisfaction, CS composition has to meet the various objective and subjective constraints about CSs. These constraints include the collaboration or conflict relationship between CSs determined by the compatibility of various CSP platforms, the users' preferences for CSPs influenced by the interoperability between CS platforms [1], [16]. Therefore, the abundant candidate CSs and the complicated constraints between them make it more challenging to optimize CS composition for DiAs in a hybrid cloud.

(3) The CS composition for DiAs is supposed to achieve the collaborative optimization of resource assignment in a hybrid cloud. To meet the changing and emerging demands of DiAs, an organization will maintain a sharing resource pool accommodating enough hybrid CSs in a hybrid cloud by reserving the public or private CSs. A DiA usually consists of multiple tasks. The CS with the suitable computation or storage capacity needs to be selected from candidates for every task. Then, a DiA can be viewed as a collaboration system involving multiple CSs from CSPs [17], [18]. The capacity of a CS can be depicted by its hardware configuration, cost and QoS. The capacity of different CSs determines their different collaborative abilities. The appropriate CSs should be assigned for a DiA according to its actual requirements. Thus, given the uncertainty of QoS, the diversity of hardware configurations and the flexibility of pricing, it is an intricate problem to optimize resource assignment based on available CSs in a hybrid cloud for maximizing the synthetically collaborative ability of a DiA.

Researchers have put considerable efforts on the service composition problem as it is related to web services, public CSs, mobile services and pervasive services [19], [20], [21], [22], [23], [24]. However, the hybrid cloud paradigm endows DiAs with new characteristics. The problem of **CS c**omposition for **D**iAs in a **h**ybrid **c**loud (denoted as **CSCD-HC**) needs to deal with the multi-valued QoS evaluations, complicated constraints and resource assignment optimization, for achieving the optimal resource utilization on the premise of meeting the computation and storage requirements of users within budget limits.

Inspired by the role-based collaboration (RBC) theory [25], [26], [27], [28], [29], [30], [31], [32], this paper models a DiA in a hybrid cloud as an RBC system, and proposes a collaborative optimization approach for the CSCD-HC problem. The cloud model theory is employed to analyze the characteristics of multi-valued QoS evaluations. By utilizing the Mahalanobis distance, a new similarity measurement method of cloud models is presented to evaluate the qualification of a CS for a task. The **e**nvironments—**c**lasses, **a**gents, **r**oles, **g**roups, and **o**bjects (E-CARGO) model is exploited to formalize the CSCD-HC problem with complicated constraints, and a solution using IBM ILOG CPLEX optimization package[5] is put forward to optimize CSCD-HC.

1.2 Our Contributions

The main contributions of this paper are as follows:

(1) Targeting the uncertain and dynamic characteristics of CSs, this paper exploits the cloud model theory to analyze the multi-valued QoS evaluations. To overcome the limitations of existing research, a new method utilizing the Mahalanobis distance is presented to measure the similarity of QoS cloud models. Based on it, the qualification of every candidate CS for every task is measured for supporting the decision-making of CS composition. The experimental results demonstrate that the proposed similarity measurement method is effective and can guarantee the high accuracy for assessing the CS's qualification.

(2) Inspired by the RBC theory, this paper innovatively models a DiA based on CS composition in a hybrid cloud as an RBC system, and utilizes the E-CARGO model to formalize the CS composition optimization problem for DiAs in a hybrid cloud. With the consideration of the multi-valued QoS evaluations and the complicated constraints, a solution using CPLEX is put forward to solve this problem. The experimental results demonstrate that the proposed approach is effective and feasible for optimizing the CS composition in a hybrid cloud.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 gives the problem statement. Section 4 utilizes the E-CARGO to define the problem model. Section 5 presents the qualification assessment method via cloud model theory. Section 6 proposes a solution to solve the CSCD-HC problem. Section 7 analyzes the experiments and results. Finally, the conclusions and further study are given in Section 8.

2 Related Works

2.1 CS Composition Problem

As an NP hard optimization problem [33], CS composition has been attracting much attention from the academic and

4. https://aws.amazon.com/ec2/

5. https://www.ibm.com/us-en/marketplace/ibm-ilog-cplex

industrial community. Some key issues in this problem, such as selecting appropriate CSs from a service pool, satisfying service composition constraints, coping with dynamic characteristics of CSs and network, must be addressed to assure the users' satisfaction [34].

Recently, a lot of achievements on service composition have been made aiming at the diverse constraints in the web services, public CSs, mobile services and pervasive services. Considering that the QoS correlations between services cause the performance issues for service composition, Deng et al. [19] proposed a correlation-aware service pruning method to select the candidate services. Targeting the mobility, unpredictability, and variation of mobile network's signal strength, Deng et al. [20] designed a mobility-enabled selection algorithm for composite services. To protect the privacy of CSs in the cross-cloud environment, Dou et al. [21] proposed a privacy-aware CS composition method for big data applications. Combining the service's past social interactions and popularity, Chen et al. [22] studied the strength of relationships between services and exploited the service's sociability to improve the quality of service composition. From the perspective of a developer, Deng et al. [23] studied the cost performance driven service mashup by taking the service package into account. To lower the communication cost and composition time, Siebert et al. [24] proposed a localized approach for service composition to interconnect various smart devices in pervasive computing environments. Aiming at the characteristics of DiAs, Liu et al. [35] indicated that the computation tasks related to the fixed datasets should be executed by the CSs of the same CSPs. In addition, the cloud workflow over Hadoop [16] and the Internet of things [36] bring new constraints for the CS composition problem related to DiAs.

Some classic algorithms, evolutionary algorithms and heuristic methods are applied to solve the large-scale service composition problem. Wu et al. [37] proposed a QoS-aware model and employed an extended genetic algorithm (GA) to optimize the composite service. Deng et al. [23] formulated the service mashup problem as a an integer-programming problem and proposed a GA-based method to solve it. To address the alliance relationship between services, Zhang et al. [38] presented a particle swarm optimization algorithm to solve service composition. Combining the greedy algorithm and ant colony optimization, Yu et al. [39] optimized the service compositions in a multi-cloud environment.

Although the above work is helpful, the CS composition optimization for DiAs in a hybrid cloud is still an open issue. To meet the requirements for the computation and storage of big data and obtain the best users' satisfaction, it is necessary to accurately assess the performance and QoS of CSs and to select the appropriate CSs for a DiA. The existing studies on DiAs focus on data placement and resource provision for improving the performance or reducing the cost [2], [16], [40]. Few of them take into account of the multi-valued QoS evaluations and the complicated constraints between CSs in a hybrid cloud.

2.2 Cloud Model Theory and Its Applications

Gaussian distributions are found widely in nature and society. The Gaussian distribution functions with the parameters of expectation (Ex) and standard variance (En) are often used as the membership functions in fuzzy sets. However, Li et al. [41] found that a concept might have the different meanings for different people, such that the membership degree is difficult to be identified precisely. Therefore, Li et al. introduced the hyper entropy (He) as the standard variance of En into the cloud model and proposed the cloud model theory. Cloud model theory [41] is an effective tool in transforming between the qualitative concepts and their quantitative expressions, and can represent the fuzziness, the randomness and the relationships of uncertain concepts. It has recently been applied successfully in many fields including the data processing [42], uncertainty measurement [43], performance evaluation [44] and decision analysis [45].

Cloud model theory can also provide the strong support for analyzing the latent features hidden in time series data [46], and clearly depict the global and local features of time series data [47], [48]. In a hybrid cloud environment, the time series data is the important component of multi-valued QoS evaluations. Considering the advantages in recognizing adaptively the relationships of the uncertain concepts, the cloud model theory could help to establish an effective mechanism to describe the characteristics of multi-valued QoS evaluations. Therefore, this paper employs the cloud model theory to analyze the multi-valued QoS evaluations, and puts forward a novel method by utilizing the Mahalanobis distance to improve the similarity measurement of QoS cloud models.

2.3 Role-Based Collaboration (RBC)

In view of the uncertain big data, Wang et al. [17] studied the evolution of a service-oriented system via different machine learning models. Liang et al. [18] employed large-system theory to model a DiA based on CS composition, and predicted its system performance via the identification and control technologies of time-varying system.

Although many achievements have been made on service composition, the existing approaches are limited to specific scenarios. To the best of our knowledge, no existing research has studied the CSCD-HC problem and modeled this problem from the perspective of RBC.

RBC is a promising computational methodology that utilizes roles as an underlying mechanism to facilitate collaboration and its model E-CARGO is valuable to model the components and processes of collaboration activities [25], [26]. E-CARGO model describes an RBC system and its key components in the form of formalized language. Based on it, researchers can employ its six core concepts, including environment, class, agent, role, group and object, to establish the standard mathematical model relevant to the assignment problem and combinatorial optimization problem. The research results on E-CARGO contribute to the theoretical models and solutions for group role assignment (GRA) [27], [28], GRA with conflicting agents (GRACA) [29], GRA with cooperation and conflict factors (GRACCF) [30] and group multi-role assignment (GMRA) [31], [32] problems. Recently, E-CARGO has been applied in different fields [49], [50].

In a hybrid cloud, a DiA is a RBC system involving the cooperation of CSs from various CSPs. We introduce the E-CARGO model to describe the CSCD-HC problem, propose its formal model with multi-constraints, and probe into a new idea of CS composition optimization.

CS	$task_1$	$task_2$	$task_3$	$task_4$
CS_1	0.84	0.79	0.93	0.76
CS_2	0.92	0.90	**0.97**	0.88
CS_3	0.67	0.74	0.82	**0.86**
CS_4	0.71	0.86	0.90	0.54
CS_5	0.86	**0.87**	0.68	0.85
CS_6	**0.91**	0.85	0.81	0.84

(a)

$$T = \begin{bmatrix} 0\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 1\,0\,0\,0 \end{bmatrix}$$

(b)

Fig. 1. An example of CS composition for a DiA in the private cloud. (a) A qualification matrix. (b) An assignment matrix.

3 Problem Statement

Assume that a DiA consists of n tasks and there are m candidate CSs in the hybrid cloud. The CSCD-HC problem is to achieve the combinatorial optimization by mapping between n tasks and m CSs. In a GRA problem, a group needs to be initiated by assigning roles to its members or agents to achieve its highest performance [27]. Similarly, a task of a DiA and a candidate CS can be directly modeled as a role and an agent in a GRA problem, respectively. To make this DiA work efficiently, n tasks must be assigned to a group of CSs. Every selected CS plays a specific role associated with one task. Every task may have the various demands about the hardware configuration, cost and QoS for the expected CS. On the premise of meeting the computation and storage requirements of DiAs within budget limits, the CS selected for a task is expected to have the best possible QoS. For one task, the competencies of different CSs are not identical. The qualification value can be used to measure one CS's competency for a task by evaluating its hardware configuration, cost and QoS. The optimization goal of CSCD-HC is to maximize the sum of qualification values of CSs that are selected for n tasks. Thus, the above characteristics of CSCD-HC problem make it become a special kind of GRA problem.

In a private cloud, no additional cost needs to be paid for CSs; the CSs' performance is definite and stable; the qualification of a CS for a task can be measured by directly evaluating its QoS. An example is given in Fig. 1a.

In this example, some suitable CSs need to be selected from 6 candidates to execute 4 tasks. The qualification value of a CS for a task is described with a decimal within [0,1]. There are $A_6^4(=360)$ permutations for this example and the different permutation associates the different sum of qualification. By utilizing the improved K-M algorithm [27], the optimization result can be obtained shown in Fig. 1b. T is an assignment matrix. $T_{i,j}=1$ means that CS $\#i$ is selected for executing task $\#j$. According to $T, \{\mathrm{CS}_6, \mathrm{CS}_5, \mathrm{CS}_2, \mathrm{CS}_3\}$ is the optimal permutation and gains the largest sum of qualification, namely, 3.61.

Considering the heavy financial burden of offering a vast private infrastructure, the CS composition architecture for DiAs in a hybrid cloud is designed in Fig. 2.

Fig. 2 illustrates a classic DiA—the GWAC light curve processing system [51]. A data-intensive computation platform is designed to support the CS composition for this DiA. The core components of this platform include:

(1) Resource manager: It manages the CSs in the sharing resource pool. The new CSs may be applied for users when the sharing resource pool cannot meet their requirements for computation and storage, and some long-term unused CSs with unsatisfactory QoS or low hardware configuration need to be eliminated from the pool.

(2) Resource monitor: It collects the QoS parameters and load status of every CS. The key QoS parameters include response time, throughput and so on. The load status is evaluated based on the real-time monitoring of CPU, memory, disk and network. Then, those CSs with low enough loads may be assigned the new computation tasks for improving the resource utilization.

(3) Cloud service composer: It provides the decision-making of cloud service composition for DiAs according to the specified optimization goals and constraint conditions. The hardware configuration, cost and QoS of CSs will be considered in the process of decision-making.

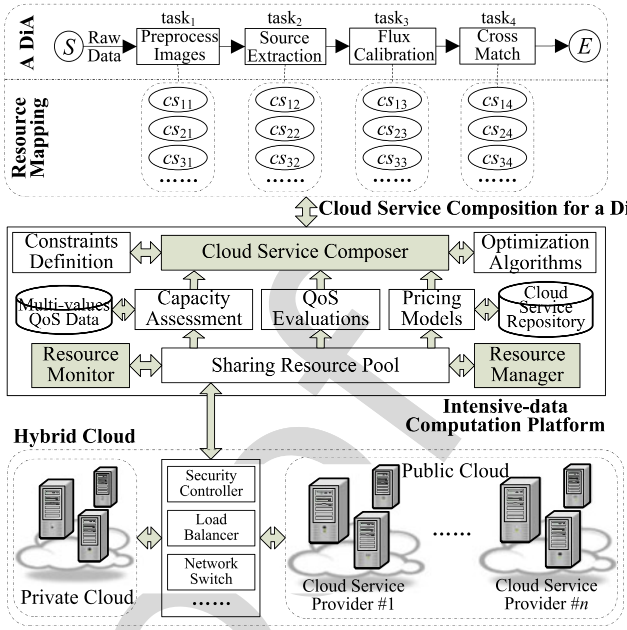


Fig. 2. CS composition architecture for DiAs in a hybrid cloud.

In a hybrid cloud, the complexity of CS composition for DiAs increases markedly due to the dynamic Internet network, the diverse CS products and the flexible pricing. We have to face the situation of multi-valued QoS evaluations [15] consisting of single real number, interval number and time series data. By referring to the hardware configuration and pricing of Amazon EC2, an example with the multi-valued QoS evaluations is shown in Table 1. In Table 1, CS_1 is a private CS with no extra cost for its usage.

Moreover, the constraints, such as the relationship of collaboration or conflict between CSs and the users' preferences for different CSPs, are the indispensable factors for the CSCD-HC problem. An example of constraints about CSs is shown in Table 2. From Table 2, 1 means the conflict relationship between two CSs, while 0 means that the two CSs can collaborate in a DiA; a value greater than 0 indicates a positive preference for one CS, a value smaller than 0 means a negative preference, and 0 means no preference. Thus, it is

TABLE 1
An Example with Multi-Valued QoS Evaluations

CS	Hardware Configuration			Cost ($/h)	QoS Evaluations	
	vCPU	Memory (GiB)	Storage (GB)		Response time for task$_1$ (s)	Throughput for task$_1$ (k/s)
CS$_1$	4	30.5	3×2000	\	1.9	400
CS$_2$	8	61	160	0.665	$[1.4, 1.7]$	$[330, 510]$
CS$_3$	16	122	320	1.33	$\{3.1, 3.2, 2.8, \ldots\}$	$\{320, 310, 300, \ldots\}$
CS$_4$	8	61	6×2000	1.38	$\{2.0, 1.7, 1.6, \ldots\}$	$\{340, 320, 330, \ldots\}$
CS$_5$	16	122	12×2000	2.76	$\{2.4, 2.1, 2.8, \ldots\}$	$\{180, 230, 390, \ldots\}$
CS$_6$	36	244	24×2000	5.52	$\{1.4, 1.1, 1.6, \ldots\}$	$\{380, 360, 370, \ldots\}$

TABLE 2
An Example of Constraints About CSs

Users	Collaboration or conflict						Use preference
	CS$_1$	CS$_2$	CS$_3$	CS$_4$	CS$_5$	CS$_6$	
CS$_1$	0	0	0	0	0	0	0
CS$_2$	0	0	0	0	0	0	0.2
CS$_3$	0	0	0	0	1	0	0
CS$_4$	0	0	0	0	0	0	0.3
CS$_5$	0	0	1	0	0	0	−0.3
CS$_6$	0	0	0	0	0	0	0

necessary to further study how to accurately measure the qualification of every candidate CS for every task and achieve the CS composition optimization for a DiA in a hybrid cloud.

Aiming at the characteristics of multi-valued QoS evaluations and the complicated constraints between CSs, this paper employs the cloud model theory to measure the qualification of every candidate CS for every task. Based on the E-CARGO model, this paper formulates the CSCD-HC problem as a GRA problem, and proposes a collaborative optimization approach for solving it.

4 Problem Model

A DiA based on CS composition in a hybrid cloud can be viewed as an RBC system. In this system, the CSs from different CSPs play different roles to execute a group of relevant tasks. As the fundamental model of RBC, E-CARGO brings in new visions to a collaboration system [25]. With the E-CARGO model, the CSCD-HC problem can be well defined and finally solved by virtue of the mature algorithm with high efficiency.

4.1 Basic Model

Aiming at the characteristics of a DiA in a hybrid cloud, the basic model of CSCD-HC problem based on E-CARGO [25], [27], [29], [30] can be defined as a six-tuple:

$$\sum ::= <E, C, O, R, A, G>,$$

whose components are explained below. E represents the problem environment. An environment denotes a plan or proposal to compose a set of CSs. C is a set of classes representing the definitions of abstract concepts relevant to E. O is a set of concrete objects connecting to C. R is a set of tasks in a DiA. A task corresponds to a task in GRA problem. A is a set of candidate CSs. A candidate CS corresponding to an agent in GRA problem can play one or several roles in a DiA. G is a set of groups. A group is a team of CSs to be established to fit an environment. In order to gain an optimal group, we need to assign the suitable candidate CSs to the appropriate tasks.

Assume the nonnegative integers $m = (|A|)$ expresses the size of $A, n = (|R|)$ expresses the size of $R, i, i_1, i_2, \ldots$ expresses the indices of candidate CSs, and $j, j_1, j_2, \ldots$ expresses the indices of tasks. Three supplemental components of the basic model include:

(1) Qualification matrix Q: It is an $m \times n$ matrix, where $Q[i,j]$ is the qualification value of CS $\#i$ for task $\#j$.

(2) Task assignment matrix T: It is an $m \times n$ matrix, where $T[i,j] \in \{0,1\}\,(0 \le i < m, 0 \le j < n)$ expresses if CS #i is assigned to task $\#j$. $T[i,j] = 1$ means yes and zero means no.

(3) Group performance ρ: It is the sum of qualification values of assigned CSs in a group, i.e., $\rho = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i,j] \times T[i,j]$.

4.2 Constraints Definitions

The CSCD-HC problem satisfies the following constraints:

(1) Weight vector of tasks $W : W[j] \in [0,1]$ is the weight of task #j, satisfying $\sum_{j=0}^{n-1} W[j] = 1$. In a DiA, the weights of some tasks processing the critical data are greater than other tasks processing the non-critical data.

(2) Lower bound vector of tasks $L : L[j]$ expresses how many CSs must be assigned to task $\#j$. $L[j] > 1$ means that task $\#j$ requires multiple CSs for the specific demands caused by the parallel computing or critical data backups.

(3) Conflicting CSs matrix C: It is an $m \times m$ matrix, where $C[i_1, i_2] \in \{0,1\}$. $C[i_1, i_2] = 0$ means that CS $\#i_1$ is in conflict with CS $\#i_2$ due to the incompatibility of CSP platforms, while $C[i,k] = 0$ means that CS $\#i_1$ can collaborate with CS $\#i_2$ in the same group.

(4) Preference vector $P : P[i] \in [-0.5, 0.5]\,(0 \le i < m)$. $P[i] = 0$ means no preference for CS $\#i; P[i] > 0$ means the positive preference; $P[i] < 0$ means the negative preference.

4.3 Objective Function

Given R, A, Q and the above constraints definitions, the CSCD-HC problem is to find a matrix T to:

$$\max \rho = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} Q[i,j] \times T[i,j] \times W[j] \times (1 + P[i]), \tag{1}$$

subject to:

$$T[i,j] \in \{0,1\}\,(0 \le i < m,\ 0 \le j < n), \tag{2}$$

$$\sum_{i=0}^{m-1} T[i,j] = L[j]\,(0 \le j < n), \tag{3}$$

$$\sum_{j=0}^{n-1} T[i,j] \le 1\,(0 \le i < m), \tag{4}$$

$$C[i_1, i_2] \times (T[i_1, j] + T[i_2, j]) \leq 1, \quad (0 \leq i_1, i_2 < m, i_1 \neq i_2, 0 \leq j < n), \tag{5}$$

$$C[i_1, i_2] \times (T[i_1, j_1] + T[i_2, j_2]) \leq 1, \quad (0 \leq i_1, i_2 < m, i_1 \neq i_2, 0 \leq j_1, j_2 < n), \tag{6}$$

where Eq. (2) shows that a CS can only be assigned or not; Eq. (3) requires that a group should satisfy the tasks' lower bound constraint; Eq. (4) requires that each CS can only be assigned to one task; Eq. (5) shows that the two CSs assigned to execute the same task should satisfy the conflicting constraints; Eq. (6) requires that two CSs in a group should meet the conflicting constraints.

To solve the objective function, we need to establish an optimal group with the maximum group performance $\boldsymbol{\rho}$ by selecting the appropriate candidate CSs for n tasks. Thus, the qualification assessment is the key precondition to identify the optimal solution. In a private cloud environment, the CSs' performance and QoS are definite and stable, and there is no additional cost and complicated constraints; the qualification value of a CS for a task can be easily assessed and described with a decimal by comparing the QoS of CSs. Taking Fig. 1a for example, based on an 6×4 qualification matrix, Fig. 1b is the optimal solution that makes the group work with $L = [1, 1, 1, 1, 1]$ and $W = [0.2, 0.2, 0.2, 0.2]$; the optimal ρ is 0.903. However, in a hybrid cloud environment, the virtual hardware configuration, uncertain QoS and flexible pricing of public CSs make it difficult to exactly assess the qualification values of CSs and identify the optimal solution; an example is shown in Tables 1 and 2. Thus, the qualification assessment model is proposed in Section 4.4.

4.4 Qualification Assessment Model

The CSs, possibly integrated into DiAs, are mainly classified into four types as follows: general storage CSs providing massive data storage capacity, such as Amazon S3[6]; general computation CSs providing high performance compute capacity, such as Amazon EC2; dedicated storage CSs providing massive storage capacity for specific data formats or types, such as Google Cloud SQL and Cloud Bigtable[7]; dedicated computation CSs providing high performance platform and efficient algorithms for specific computation tasks, such as Tencent-Cloud cloud recommendation engine[8]. The qualification of a CS for a task is determined by comparing the task's requirements with the CS's actual situation. Different tasks have various requirements for CSs in a DiA. These requirements are mainly divided into three aspects as follows:

(1) Hardware configuration expectation E^{H}: E^{H} expresses the minimal requirements to complete a task for hardware parameters, such as virtual CPU (vCPU), memory, storage and so on. $E_i^{\mathrm{H}} = \{e_{i,1}^{\mathrm{H}}, e_{i,2}^{\mathrm{H}}, \ldots\}$, where $e_{i,j}^{\mathrm{H}}$ is the ith task's expectation for the jth hardware parameter. For example, the E^{H} of a task is: $E_i^{\mathrm{H}} = \{2\,\text{vCPU}, 4\,\text{GiB of memory}, 1000\text{G storage}\}$.

(2) Cost expectation E^{C}: E^{C} expresses the cost expectation of CSs. $E^{\mathrm{C}} = \{e_1^{\mathrm{C}}, e_2^{\mathrm{C}}, \ldots, e_n^{\mathrm{C}}\}$, where e_i^{C} is the ith CS's cost expectation and its default value is the mean cost of all candidate CSs.

(3) QoS expectation: On the condition of satisfying E^{H} and E^{C}, the CS selected for a task usually is expected to obtain the best possible QoS.

The CSs' actual situations include: the real hardware status $R_i^{\mathrm{H}} = \{r_{i,1}^{\mathrm{H}}, r_{i,2}^{\mathrm{H}}, \ldots\}$ $(0 \leq i < m)$, the real usage cost $R^{\mathrm{C}} = \{r_1^{\mathrm{C}}, r_2^{\mathrm{C}}, \ldots, r_m^{\mathrm{C}}\}$ and the QoS evaluations $R_i^{\mathrm{Q}} (0 \leq i < m)$. $R_i^{\mathrm{Q}} = \{r_{i,1}^{\mathrm{Q}}, r_{i,2}^{\mathrm{Q}}, \ldots\}$, where $r_{i,k}^{\mathrm{Q}}$ is the CS #i's multi-valued evaluations relevant to the kth QoS parameter. Taking CS_3 from Table 1 for example, $R_3^{\mathrm{H}} =$ {16 vCPU, 122 GiB of memory, 320G storage}; $r_3^{\mathrm{C}} = 1.33\ \$/h$; the multi-valued evaluations of response time and throughput are collected, denoted as $r_{3,1}^{\mathrm{Q}}$ and $r_{3,2}^{\mathrm{Q}}$, respectively. $r_{3,1}^{\mathrm{Q}} = \{3.1s, 3.2s, 2.8s, \ldots\}$, $r_{3,2}^{\mathrm{Q}} = \{320k, 310k, 300k, \ldots\}$. Then, the qualification value of CS $\#i$ for task $\#j$ can be measured as follows:

$$Q_{i,j} = f_{i,j}^{\mathrm{H}} \times f_{i,j}^{\mathrm{Q}} \times f_{i,j}^{\mathrm{C}}, \tag{7}$$

where $f_{i,j}^{\mathrm{H}}$, $f_{i,j}^{\mathrm{C}}$ and $f_{i,j}^{\mathrm{Q}}$ are the hardware conformity, cost conformity and QoS conformity of CS $\#i$ for task $\#j$, respectively. Considering that E^{H} is the minimal hardware requirements to complete a task, a qualified CS should have more hardware resources than E^{H}. Then, the hardware conformity $f_{i,j}^{\mathrm{H}}$ is obtained by:

$$f_{i,j}^{\mathrm{H}} = \begin{cases} 0, & if\ \forall k, r_{i,k}^{\mathrm{H}} < e_{i,k}^{\mathrm{H}} \\ 1, & other. \end{cases} \tag{8}$$

The cost conformity $f_{i,j}^{\mathrm{C}}$ is calculated by:

$$f_{i,j}^{\mathrm{C}} = \begin{cases} 1, & \text{if } r_i^{\mathrm{C}} \leq e_j^{\mathrm{C}} \\ 1 - \left((r_i^{\mathrm{C}} - e_j^{\mathrm{C}}) \big/ e_j^{\mathrm{C}}\right)^2, & \text{if } e_j^{\mathrm{C}} < r_i^{\mathrm{C}} \leq 2e_j^{\mathrm{C}} \\ 0, & \text{if } r_i^{\mathrm{C}} > 2e_j^{\mathrm{C}} \\ \alpha, & \text{if } \text{CS}_i \text{ is a private CS,} \end{cases} \tag{9}$$

where $0 \leq f_{i,j}^{\mathrm{C}} \leq 1$; α denotes the cost coefficient of private CSs. α is set as a fixed value because there is no additional usage fee for private CSs. The value range of α is suggested from 1.0 to 1.3 according to the application scenarios. In practice, the public CSs with a high hardware configuration usually have a high price. Thus, we only assign the appropriate resources for a DiA according to its actual requirements.

To gain the QoS conformity $f_{i,j}^{\mathrm{Q}}$, we need to analyze the different QoS parameters individually in accordance with their types: gain or loss type. For example, response time is the loss type of QoS parameter, and throughput is the gain type of QoS parameter. When the evaluation data about multiple QoS parameters is available, $f_{i,j}^{\mathrm{Q}}$ is calculated by aggregating the conformity values of multiple QoS parameters with weighted operator. For example, if the tasks' QoS expectation of a DiA involves two QoS parameters, namely, response time and throughput, then

$$f_{i,j}^{\mathrm{Q}} = w^{rt} \times f_{i,j}^{rt} + w^{tp} \times f_{i,j}^{tp}, \tag{10}$$

where w^{rt} and w^{tp} are the weights of response time and throughput, respectively; $f_{i,j}^{rt}$ and $f_{i,j}^{tp}$ are the response time conformity and throughput conformity of CS $\#i$ for task $\#j$, respectively. In a hybrid cloud, we have to face the

6. https://aws.amazon.com/s3/
7. https://cloud.google.com/products/storage/
8. https://www.qcloud.com/product/cre

multi-valued QoS evaluations consisting of single real number, interval number and time series data. To calculate the QoS parameters' conformity and the CS's qualification, we introduce the cloud model theory to analyze the multi-valued QoS evaluations in Section 5.

5 Qualification Assessment via Cloud Model Theory

In this section, we define a QoS cloud model to analyze the multi-valued QoS evaluations and calculate the qualification value of candidate CSs for the tasks in a DiA.

5.1 QoS Cloud Model

A QoS cloud model [42] is composed of three numerical characteristics, namely Ex (expectation), En (entropy) and He (hyper entropy), defined as $cm = \{Ex, En, He\}$. Ex is the most representative value of QoS, En denotes the granularity scale of QoS, and He depicts the uncertainty of the QoS granularity. From the viewpoint of fuzzy set, Ex is the expected value of QoS with the membership degree 1, En represents the uncertainty of QoS values, which can be used to calculate the membership degree, and He depicts the uncertainty of membership degree. The QoS cloud models make it possible to get the distributing range of QoS by exploiting the continuous monitoring data.

A QoS cloud model consists of many cloud drops. The CSs' multi-valued QoS evaluations obtained in multiple timeslots can be viewed as the cloud drops and sent to a reverse cloud generator (RCG) [52], where the QoS cloud model's three-digit features can be calculated by:

$$\begin{cases} Ex = \overline{V} = \frac{1}{N}\sum_{k=1}^{total} v_k \\ En = \sqrt{\frac{\pi}{2}} \times \sigma = \sqrt{\frac{\pi}{2}} \times \frac{1}{N}\sum_{k=1}^{total} |v_k - Ex| \\ He = \sqrt{|S^2 - En^2|} = \sqrt{\left|\frac{1}{N-1}\sum_{k=1}^{total} (v_k - \overline{V})^2 - En^2\right|}, \end{cases} \tag{11}$$

where v_k is the QoS evaluation obtained in timeslot $\#k$; Ex is the mean value of QoS evaluations; σ is the standard deviation of Ex; S^2 is the sample variance of Ex, and *total* is the number of timeslots. For example, assume $Sa = \{0.305, 0.383, 0.331, 0.311, 0.338, 0.272, 0.262, 0.315, 0.329, 0.357\}$ is the response time data of CS *a*. The QoS cloud model related to it is $cm = \{0.3203, 0.0342, 0.0121\}$.

5.2 Similarity Measurement of QoS Cloud Model

To select the appropriate candidates for CS composition, it is crucial to identify the differences of the CSs' QoS by calculating the similarity of QoS cloud models. Some methods, such as DropCM method [53], LICM method [54], EDCM method [55], ECM method and MCM method [46], have been proposed to compute the similarity between two cloud models. However, these methods have their own limitations [15], such as the time-consuming computation, obvious calculation errors, and unsatisfactory calculation precision. To overcome the limitations of the existing methods, in the CS selection research, Ma et al. [15] presented a vector comparison method called as VCM method by combining the orientation similarity and dimension similarity; whereas this method cannot adaptively adjust the regulatory factor, which determines the weights of the orientation similarity and dimension similarity, in light of the diverse requirements of different tasks in a DiA. Therefore, this paper proposes a new measurement method utilizing the Mahalanobis distance to compute the similarity between two QoS cloud models for the qualification assessment in the CSCD-HC problem.

The Mahalanobis distance is a method of measuring the distance of data covariance that can effectively calculate the similarity between two unknown sample sets. The Mahalanobis distance is independent of the measurement scales unlike the Euclidean distance, and it remains unaffected by the different dimensions between coordinates. Recently, the Mahalanobis distance has been applied in many research fields [56], [57].

This paper utilizes the Mahalanobis distance to improve the computational accuracy of QoS cloud model similarity, noted as MaCM method. Let $cm_1 = \overrightarrow{V_1} = (v_1^1, v_2^1, v_3^1) = (Ex_1, En_1, He_1)$ and $cm_2 = \overrightarrow{V_2} = (v_1^2, v_2^2, v_3^2) = (Ex_2, En_2, He_2)$ be two QoS cloud models. Then, the Mahalanobis distance between cm_1 and cm_2 is calculated by:

$$MD(cm_1, cm_2) = \sqrt{(\overrightarrow{V_1} - \overrightarrow{V_2})S^{-1}(\overrightarrow{V_1} - \overrightarrow{V_2})^{\mathrm{T}}}, \tag{12}$$

where $\overrightarrow{V_1} - \overrightarrow{V_2} = (v_1^1 - v_1^2, v_2^1 - v_2^2, v_3^1 - v_3^2)$; T represents the transposition operation; S^{-1} is the inverse matrix of sample covariance matrix, and it is a symmetry positive definite matrix as follows:

$$S^{-1} = \begin{bmatrix} s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

Then, the Mahalanobis distance can also be defined by:

$$MD(cm_1, cm_2) = \sqrt{\sum_{1 \le m,n \le 3} s_{m,n}(v_m^1 - v_m^1)(v_n^2 - v_n^2)}, \tag{13}$$

where S^{-1} reveals the relationship between Mahalanobis distance and Euclidean distance. When $S^{-1} = I$ (an identity matrix), the three dimensions of cloud model, namely Ex, En and He, have the same fluctuation range, and we have

$$MD(cm_1, cm_2) = \sqrt{\sum_{m=1}^{3} (v_m^1 - v_m^2)^2},$$

which is equivalent to the Euclidean distance. When S^{-1} is a diagonal matrix, for example,

$$S^{-1} = \begin{bmatrix} s_{1,1} & 0 & 0 \\ 0 & s_{2,2} & 0 \\ 0 & 0 & s_{3,3} \end{bmatrix},$$

we have

$$MD(cm_1, cm_2) = \sqrt{\sum_{m=1}^{3} s_{m,m}(v_m^1 - v_m^2)^2},$$

which is equivalent to the weighted Euclidean distance.

The smaller the Mahalanobis distance, the more similar two QoS cloud models. Therefore, the similarity between two QoS cloud models can be obtained by:

$$MaCM\_sim(cm_1, cm_2) = \frac{1}{1 + MD(cm_1, cm_2)}. \tag{14}$$

5.3 Qualification Assessment Method

The qualification assessment method is stated as follows:

Step 1: Transform the multi-valued QoS data into the QoS cloud models. If the QoS evaluation is described as the time series data or interval numbers, all evaluations are sent to RCG, and a QoS cloud model including three numerical characteristics can established by Eq. (11). If the QoS evaluations are the single-valued data, let p^i be the unique evaluation value; then, a specific QoS cloud model $\{p^i, 0, 0\}$ is obtained. Thus, the cloud model matrix of QoS evaluations for m CSs can be described as follows:

$$CM = \begin{bmatrix} cm_{1,1} & cm_{1,2} & \dots & cm_{1,n} \\ cm_{2,1} & cm_{2,2} & \dots & cm_{2,n} \\ \vdots & \vdots & cm_{i,j} & \vdots \\ cm_{m,1} & cm_{m,2} & \dots & cm_{m,n} \end{bmatrix}, \tag{15}$$

where $cm_{i,j} = (Ex_i^j, En_i^j, He_i^j)$ is the QoS cloud model of service $\#i$ for task $\#j$. One task may experience the different QoS when running in various CS infrastructures. Due to the differences between task types, tasks may obtain the different QoS even if they are executed in the same CS.

Step 2: Identify the positive and negative ideal solutions for every task. An excellent CS should provide a steady QoS for users. The smaller the fluctuation ranges of En and He, the steadier the QoS. According to this principle, we define the ideal solutions. For the gains type, the positive and negative ideal solutions are identified by:

$$\begin{aligned} cm_j^+ &= \left\{ \max_{1\le i\le m}\{Ex_i^j\}, \min_{1\le i\le m}\{En_i^j\}, \min_{1\le i\le m}\{He_i^j\} \right\} \\ cm_j^- &= \left\{ \min_{1\le i\le m}\{Ex_i^j\}, \max_{1\le i\le m}\{En_i^j\}, \max_{1\le i\le m}\{He_i^j\} \right\}. \end{aligned} \tag{16}$$

For the loss type, the positive and negative ideal solutions can be defined by:

$$\begin{aligned} cm_j^+ &= \left\{ \min_{1\le i\le m}\{Ex_i^j\}, \min_{1\le i\le m}\{En_i^j\}, \min_{1\le i\le m}\{He_i^j\} \right\} \\ cm_j^- &= \left\{ \max_{1\le i\le m}\{Ex_i^j\}, \max_{1\le i\le m}\{En_i^j\}, \max_{1\le i\le m}\{He_i^j\} \right\}. \end{aligned} \tag{17}$$

Step 3: Calculate the CS's QoS conformity for every task. By utilizing the Mahalanobis distance to measure the similarity between the QoS cloud model and the ideal solutions, the QoS conformity of every CS for every task is calculated by:

$$f_{i,j}^{\mathrm{Q}} = \frac{MD(cm_{i,j}, cm_j^-)}{MD(cm_{i,j}, cm_j^+) + MD(cm_{i,j}, cm_j^-)}, \tag{18}$$

where $f_{i,j}^{\mathrm{Q}} \in [0,1]$. The larger $f_{i,j}^{\mathrm{Q}}$, the better the QoS of CS $\#i$ for task $\#j$.

Step 4: Employ the weighed operator to calculate the comprehensive QoS conformity. $f_{i,j}^{\mathrm{Q}}$ is obtained by:

$$f_{i,j}^{\mathrm{Q}} = \sum_{k=1}^{z} f_{i,j}^{\mathrm{Q},k} \times w^{\mathrm{Q},k}. \tag{19}$$

where z is the number of QoS parameters; $f_{i,j}^{\mathrm{Q},k}$ and $w^{\mathrm{Q},k}$ are the conformity of the kth QoS parameter and its weight. When a user has no explicit preferences for QoS parameters, $w^{\mathrm{Q},k} = 1/z$.

Step 5: Calculate the hardware conformity and cost conformity by Eqs. (8) and (9), respectively.

Step 6: Compute the qualification value of every CS for one task by Eq. (7). Then, the qualification matrix Q is available.

6 Solution to the CSCD-HC Problem

Our previous work [27], [32] provides a solid solution framework for solving the CSCD-HC problem, i.e., role negotiation, agent evaluation, and group role assignment. RBC and GRA are applied and extended to adapt solving the CSCD-HC problem from the following aspects:

(1) CSCD-HC can be taken as a specialized GRA problem, and the assessment of CSs' qualification is critical for solving it. The qualification of a CS is determined by its hardware configuration, cost and QoS; especially, the QoS of public CSs, affected by many factors, is usually dynamic and uncertain. Thus, we propose a cloud model theory-based method to assess the qualification of public CSs. This work extends the methodology of agent evaluation in RBC to solve problems in the same category.

(2) The existing solutions to GRA problem only support the single-valued qualification of the agents. In the CSCD-HC problem, we have to face the situation of multi-valued QoS evaluations. It is necessary to adopt the solving algorithm to support the multi-valued data including decimal, interval number and time series data.

(3) In the original GRA problem, the feasible solution may be available if $m > n$ and $0 \le \sum_{j=0}^{n-1} L[j] \le m$; whereas the conditions, $m \gg n$ and $0 \le \sum_{j=0}^{n-1} L[j] \ll m$, are satisfied in a CSCD-HC problem because there are enough candidate CSs from the public cloud. This work in fact adds a specific solution to a set of extended GRA problems.

(4) Considering that the private CSs have no extra usage cost, the private CSs should be used in precedence when they meet the basic requirements of tasks. Thus, we define a threshold vector of task (τ) to depict the tasks' requirements. τ states that a CS is qualified for task $\#j$ only if its qualification value is greater than $\tau[j]$.

6.1 Steps of Solving the CSCD-HC Problem

Based on the above analysis, the main steps of solving the CSCD-HC problem are described as follows:

Step 1: By analyzing the characteristics of a DiA, collect in the data-intensive computation platform its requirements including hardware configuration expectation

(E^{H}), cost expectation (E^{C}) and QoS parameter preferences ($w^{\mathrm{Q},k}$); determine the private CSs' cost coefficient (α); set the qualification threshold vector of task (τ); identify the weight vector of task (W), lower bound vector of task (L), conflicting CS matrix (C) and preference vector (P).

Step 2: By comparing the task's requirements with the CS's actual situations, calculate the hardware configuration conformity ($f^{\mathrm{H}}_{i,j}$) by Eq. (8) and the cost conformity $f^{\mathrm{C}}_{i,j}$ by Eq. (9), and the $f^{\mathrm{C}}_{i,j}$ of private CSs is set as α.

Step 3: Transform the multi-valued QoS evaluations into QoS cloud models and calculate every CS's QoS conformity for every task by Eqs. (18) and (19).

Step 4: Compute every CS's qualification for every task by Eq. (7). Then, the qualification matrix (Q) is obtained.

Step 5: Copy Q to a temporary matrix QT and update the values of QT according to τ, W and P. If $QT[i,j] < \tau[j]$, let $QT[i,j] = 0$. Next, $QT[i,j] = QT[i,j] \times W[j] \times (1 + P[i])$.

Step 6: Based on the given QT, W, L, C and P, solve the objective function of the CSCD-HC problem.

Step 7: Obtain an optimal solution if the role assignment matrix (T) is available. If there is no feasible solution, reduce the values of $\tau[j]$ and redo the above operations from Step 5. If a feasible solution is still unavailable although all acceptable $\tau[j]$ has been used, then the data-intensive computation platform notifies the administrators to lower the expectations for CSs or reserve more CSs with high performance from the public cloud.

6.2 Solving the CSCD-HC Problem

To solve the CSCD-HC problem, we use the IBM ILOG CPLEX optimization packages, which is different from the original usage of the Optimization Programming Language (OPL) of CPLEX optimization studio. Using the package by designing a Java program can result in better performance than using the OPL's compiler. To obtain a solution with the CPLEX package, we need to collect the four required elements and convert them into the forms required by CPLEX. The main steps are stated as follows:

(1) Identify four elements, including objective function coefficients, constraint coefficients, right-hand side constraint values, and upper and lower bounds, required by CPLEX package. We use QT, L, C and T to define a linear programming (LP) problem in CPLEX. QT is the objective function coefficient. T is the variables, and its upper and lower bounds are 1 and 0.

(2) Add the objective and constraint expressions. The objective of CSCD-HC problem should be described by a formula of the one-dimensional array forms of matrices QT, T and C and the linear expressions of W, L and P.

First, transform the matrices into 1-dimensional arrays as follows: $X[i \times n + j] = T[i,j], V[i \times n + j] = QT[i,j]$ and $F[i \times n + j] = C[i,j]\ (0 \le i < m, 0 \le j < n)$. Second, to add the optimization objective, the following methods need to be invoked:

```
IloIntVar[] X = cplex.intVarArray(m * n, 0, 1);
cplex.addMaximize(cplex.scalProd(X, V));
```

Third, to add the constraints to CPLEX, we iteratively add each constraint expression into CPLEX:

1) For Eq. (3):

```
IloLinearNumExpr expr1 = cplex.linearNumExpr();
for(int j = 0; j < m; j++)
   expr1.addTerm(1, X[i + j * n]);
cplex.addEq(expr1, L[i]);
```

2) For Eq. (4):

```
IloLinearNumExpr expr2 = cplex.linearNumExpr();
for(int j = 0; j < n; j++)
   expr2.addTerm(1, X[n * i + j]);
cplex.addLe(expr2, 1.0);
```

3) For Eqs. (5) and (6):

```
for(int i = 0; i < m * m; i++){
  int row = i/m; int col = i%m;
  if(row >= col)
    continue;
  if(1 == C[i]){
    IloLinearNumExpr conflict = cplex.linearNumExpr();
    for(int j = 0; j < n; j++){
         conflict.addTerm(1, X[row * n + j]);
         conflict.addTerm(1, X[col * n + j]);
    }
  cplex.addLe(conflict, 1);
  }
}
```

(4) Invoke the $cplex$.solve() method of the CPLEX package to maximize this formula based on the objective and constraint expressions.

7 Experiments

The related work [5], [6], [7], [8], [9], [10] indicates that the hybrid cloud computing paradigm facilitates to significantly save monetary cost and improve performance for the enterprises and organizations consumers. In this paper, the proposed collaborative optimization approach for CSCD-HC centers on helping consumers to achieve the optimal resource utilization, on the premise of meeting their computation and storage requirements for DiAs within budget limits. The experiments mainly answer the two questions:

(1) Could the proposed approach accurately assess the qualification of CSs in the multi-valued QoS evaluations environment? Assuming that both the hardware configuration and the cost expectation have been satisfied, Sections 7.1 and 7.2 verify whether the cloud model-based assessment

TABLE 3
Six QoS Cloud Models

	cm_1	cm_2	cm_3	cm_4	cm_5	cm_6
Ex	9.400	9.100	9.200	9.400	9.100	8.200
En	0.326	0.390	0.520	0.326	0.490	0.606
He	0.019	0.019	0.133	0.789	0.029	0.037

method can select the CS with the best QoS for a task of DiAs in a hybrid cloud.

(2) Could the proposed approach obtain the optimal CS composition solution for DiAs in a hybrid cloud? Considering that the CSs with a high hardware configuration usually have a high price in reality, only the appropriate resources are assigned for a DiA according to its actual requirements. Section 7.3 illustrates a complete case analysis for solving CSCD-HC. Assuming that the qualification values of CSs have been accurately assessed, Section 7.4 verifies the practicability and effectiveness of the proposed approach by the comparative analysis.

7.1 Case Analysis on Similarity Measurement Method of QoS Cloud Models

In this section, a case is analyzed to validate the similarity measurement method of QoS cloud model (MaCM method) proposed in Section 5.2. Assume that there are six QoS cloud models, noted as $cm_1 - cm_6$, as shown in Table 3. From Table 3, obviously, cm_2 is the more similar cloud model to cm_1 than cm_3. To compare the accuracy of different similarity measurement methods, the similarity values between cm_1 and other cloud models were calculated, with the results displayed in Table 4.

As shown in Table 4, we can analyze it as follows: (1) The DropCM method mistakenly identifies cm_3 as the most similar cloud model to cm_1. In addition, some cloud models that differ from cm_1, such as cm_4 and cm_6, also obtain high similarity values. The calculation results of the DropCM method fail to reflect the real differences between the QoS cloud models. (2) Although cm_2 is identified as the most similar cloud model to cm_1, the LICM method has the same flaw as the DropCM method. The maximum and minimum obtained by DropCM method are 1.0000 and 0.9967, respectively. The apparently different cloud models have the high similarity values which easily cause errors when complex calculations are based on these values. (3) The ECM method mistakenly identifies cm_4 as the most similar cloud model to cm_1, although the He of cm_4 has a very unusual value, about 41.5 times larger than that of cm_1, which demonstrates the irrationality of ignoring the influence of He. (4) The MCM method mistakenly identifies cm_5 as the most similar cloud model to cm_1. It is obvious that cm_2 is more similar to cm_1 than cm_5. The MCM method produces misleading results because the influence of He is exaggerated in the integral area calculation. (5) The EDCM method mistakenly identifies cm_3 as the most similar cloud model to cm_1 because three-digit features in a cloud model have various measurement scales. In addition, Ex is more than ten times larger than En and He, according to Table 4. Therefore, the subtle differences in Ex between two QoS cloud models will hide the drastic fluctuations in En and He. (6) The MaCM method accurately identifies cm_2 as the most similar cloud model to cm_1. The similarity measurement results can reflect the real differences between two QoS cloud models more precisely than the other five methods.

TABLE 4
Similarity Measurement between QoS Cloud Models

	cm_2	cm_3	cm_4	cm_5	cm_6
DropCM	0.9932	**0.9936**	0.9880	0.9925	0.9732
LICM	**1.0000**	0.9997	0.9967	0.9998	0.9992
ECM	0.6752	0.7437	**1.0000**	0.6996	0.1979
MCM	0.7197	0.5860	0.2490	**0.7300**	0.2753
EDCM	0.7653	**0.7686**	0.5650	0.7451	0.4480
MaCM	**0.5543**	0.2800	0.2467	0.3571	0.2328

7.2 Accuracy Analysis of Qualification Assessment

The proposed qualification assessment method utilizes the Mahalanobis distance to measure the similarity of QoS cloud models. The precise assessment of CSs' qualifications is the prerequisite for solving the CSCD-HC problem. Thus, the following experiments validate the accuracy of the proposed method in the multi-valued QoS evaluations environment. The experiments use the real WS-DREAM dataset #2,[9] which collects the QoS evaluations from 142 users of 4,532 services in 64 timeslots.

First, randomly select a potential user from the dataset, and use the method proposed in Ref. [15] to identify the neighboring users for the potential user and to predict the QoS values of the candidate CSs. Second, the hardware conformity and cost conformity of candidates are fixed as 1.0 because the hardware and cost information about CSs are unavailable in the dataset. Finally, employ the proposed qualification assessment method to evaluate the candidates and select the CS with the largest qualification value for the potential user.

The mean absolute error (MAE) is used to assess the accuracy of the proposed method. MAE is defined by:

$$MAE = \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{total}\left|v_k^{*} - v_k^{o}\right|, \tag{20}$$

where S denotes the number of service selection executed; v_k^{*} is the QoS value in timeslot $\#k$ of the optimal CS experienced by a potential user; v_k^{o} is the QoS value in timeslot $\#k$ of the predicted optimal CS.

We compare the service selection approach via the proposed qualification assessment method based on MaCM method, noted as SS_MaCM, with the other six approaches as follows: (1) the service selection approach using the DropCM method, noted as SS_DropCM; (2) the service selection approach using the LICM method, noted as SS_LICM; (3) the service selection approach using the ECM method, noted as SS_ECM; (4) the service selection approach using the MCM method, noted as SS_MCM; (5) the service selection approach using the EDCM method, noted as SS_EDCM; and (6) the

9. https://github.com/wsdream/WS-DREAM

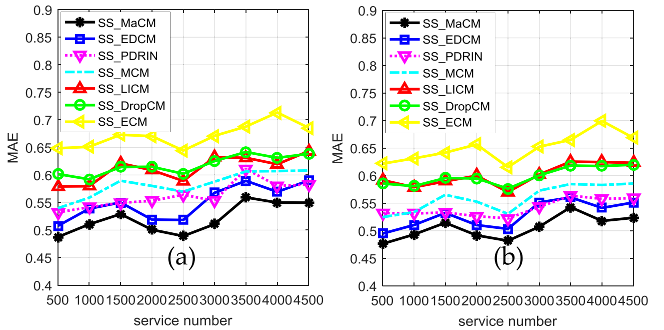


Fig. 3. Accuracy analysis. (a) the matrix density is 60%; (b) the matrix density is 80%.

service selection approach using possibility degree ranking of interval numbers [58], noted as SS_PDRIN. The experiments are performed by extracting the response time data from the dataset. Every experiment consists of 9 batches, in which 500 services are used in order. The first batch uses services #1–#500; the second batch uses services #501–#1000; and the last batch uses services #4,001–#4,500. Every batch is executed repeatedly for 50 rounds when the matrix density of the data is 60 and 80 percent, respectively. The MAE values are shown in Fig. 3.

From Fig. 3, SS_ECM obtained the largest MAEs in all approaches. The reason is that the QoS cloud models established based on the dataset are with the large hyper entropy values. The unsatisfactory results are certainly caused when the influence of hyper entropy is ignored. The MAEs of SS_DropCM and SS_LICM are also quite unsatisfactory, because they cannot identify the exact differences between QoS cloud models. The similarity between two significantly different cloud models is still very high, which can easily cause mistakes. Meanwhile, the MAEs of SS_LICM may be larger than SS_DropCM, because SS_DropCM can improve its accuracy to a certain extent through the massive calculations involving the large amounts of sampling cloud drops. SS_MCM achieved better results than SS_ECM, SS_DropCM, and SS_LICM. However, it is difficult to improve its accuracy significantly due to the errors caused by area integral calculations. The accuracy of SS_EDCM was better than that of SS_MCM overall, although lack of consideration of the differences between measurement scales reduced its accuracy. The accuracy of SS_PDRIN is somewhat similar to SS_EDCM because SS_PDRIN exploits the cloud model to identify the trustworthiness interval number. However, the possibility degree ranking of interval numbers used by SS_PDRIN also easily bring about errors in the multi-valued QoS evaluations environment. SS_MaCM obtained the highest accuracy in all approaches because it is capable of precisely measuring subtle differences between QoS cloud models. The results also demonstrate that the performance fluctuations of the original QoS data and the matrix density are closely related to MAEs. Greater performance uncertainty is bound to affect the precision of QoS cloud models, and then severely disrupts the accuracy of service selection approaches. The lower the matrix density, the larger the distortion of the QoS cloud model, which leads to the larger MAEs.

TABLE 5
E^{H} and E^{C} of a DiA

Tasks	E^{H}			E^{C} (\$/h)
	vCPU	Memory (GiB)	Storage (GB)	
Task$_1$	4	30	4000	4.0
Task$_2$	4	30	160	1.5
Task$_3$	4	20	100	1.0
Task$_4$	8	60	6000	5.0

7.3 Case Analysis for Solving CSCD-HC

A case analysis is given to illustrate the proposed collaborative optimization approach based on the example in Table 1 and Table 2. Let $m = 6, n = 4, L = [1, 2, 1, 1]$, $W = [0.3, 0.15,\ 0.15, 0.3], \alpha = 1.2$ and $\tau[j] = 0.3$. E^{H} and E^{C} are shown in Table 5. The complete multi-valued evaluations about response time are shown in Table 6.

The response time cloud models matrix is as follows:

$$CM = \begin{bmatrix} \{1.90, 0.00, 0.00\}\{1.40, 0.00, 0.00\}\{2.30, 0.00, 0.00\}\{2.70, 0.00, 0.00\} \\ \{1.55, 0.19, 0.10\}\{1.55, 0.31, 0.16\}\{1.45, 0.19, 0.10\}\{1.40, 0.25, 0.13\} \\ \{2.95, 0.19, 0.02\}\{2.30, 0.44, 0.06\}\{2.10, 0.54, 0.19\}\{1.56, 0.37, 0.13\} \\ \{1.80, 0.29, 0.04\}\{1.34, 0.20, 0.08\}\{2.73, 0.42, 0.11\}\{2.07, 0.26, 0.06\} \\ \{2.28, 0.36, 0.09\}\{2.43, 0.21, 0.04\}\{1.35, 0.17, 0.08\}\{3.13, 0.18, 0.07\} \\ \{1.48, 0.23, 0.13\}\{1.51, 0.24, 0.10\}\{1.48, 0.23, 0.13\}\{1.46, 0.23, 0.11\} \end{bmatrix}.$$

Based on CM, f^{H}, f^{C} and f^{Q} are obtained as follows:

$$f^{\mathrm{H}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad f^{\mathrm{C}} = \begin{bmatrix} 1.2 & 1.2 & 1.2 & 1.2 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0.89 & 1 \\ 1 & 1 & 0.86 & 1 \\ 1 & 0.29 & 0 & 1 \\ 0.86 & 0 & 0 & 0.99 \end{bmatrix} \quad f^{\mathrm{Q}} = \begin{bmatrix} 0.76 & 0.94 & 0.48 & 0.62 \\ 0.55 & 0.38 & 0.55 & 0.51 \\ 0.41 & 0.50 & 0.42 & 0.45 \\ 0.58 & 0.63 & 0.49 & 0.53 \\ 0.40 & 0.43 & 0.67 & 0.31 \\ 0.48 & 0.55 & 0.40 & 0.55 \end{bmatrix}.$$

Then, the Q matrix is obtained, and the final optimized CS composition is obtained with the proposed solution via CPLEX. The matrices Q, QT, and T are as follows:

TABLE 6
Multi-Valued QoS Evaluations About Response Time

	Task$_1$	Task$_2$	Task$_3$	Task$_4$
CS$_1$	1.9	1.4	2.3	2.7
CS$_2$	[1.4, 1.7]	[1.3, 1.8]	[1.3, 1.6]	[1.2, 1.6]
CS$_3$	{3.1, 3.2, 2.8, 2.9, 2.7, 3.0}	{2.7, 2.1, 1.5, 2.6, 2.1, 2.6, 2.0}	{2.5, 2.1, 1.4, 1.9, 1.7, 3.0}	{1.4, 1.7, 1.9, 1.3, 1.2, 2.1, 1.3}
CS$_4$	{2.0, 1.7, 1.6, 1.9, 1.4, 2.2}	{1.2, 1.3, 1.2, 1.5, 1.1, 1.5, 1.6}	{2.3, 3.3, 2.8, 2.4, 2.5, 3.1}	{2.2, 1.9, 1.7, 2.1, 1.9, 2.4, 2.3}
CS$_5$	{2.4, 2.1, 2.8, 2.5, 1.9, 2.0}	{2.7, 2.5, 2.4, 2.2, 2.6, 2.1, 2.5}	{1.2, 1.3, 1.7, 1.2, 1.4, 1.3}	{3.1, 3.3, 3.0, 3.3, 2.9, 3.0, 3.3}
CS$_6$	{1.4, 1.1, 1.6, 1.5, 1.4, 1.9}	{1.4, 1.1, 1.6, 1.5, 1.4, 1.9, 1.7}	{1.4, 1.1, 1.6, 1.5, 1.4, 1.9}	{1.4, 1.1, 1.6, 1.5, 1.4, 1.9.1.3}

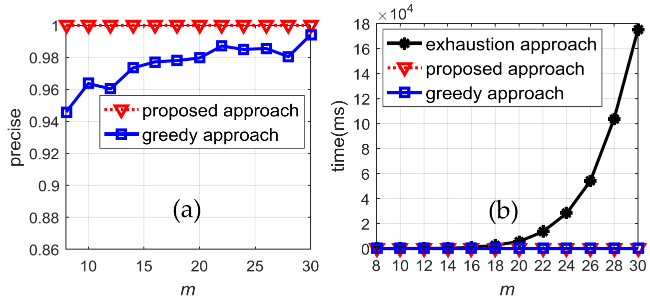


Fig. 4. Performance comparison. (a) precise; (b) execution time.

$$Q=\begin{bmatrix}0.91 & 1.13 & 0.58 & 0.74\\ 0 & 0.38 & 0.55 & 0\\ 0 & 0.50 & 0.37 & 0\\ 0.58 & 0.63 & 0.42 & 0.53\\ 0.40 & 0.13 & 0 & 0.31\\ 0.41 & 0 & 0 & 0.54\end{bmatrix}\ QT=\begin{bmatrix}0.27 & 0.17 & 0.09 & 0.22\\ 0 & 0.07 & 0.10 & 0\\ 0 & 0.07 & 0.06 & 0\\ 0.23 & 0.12 & 0.08 & 0.21\\ 0.08 & 0 & 0 & 0.06\\ 0.12 & 0 & 0 & 0.16\end{bmatrix}\ T=\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}.$$

Thus, the optimal CS composition is obtained by selecting five CSs, namely, $\{\mathrm{CS}_1, \{\mathrm{CS}_3, \mathrm{CS}_4\}, \mathrm{CS}_2, \mathrm{CS}_6\}$, for $\mathrm{task}_1-\mathrm{task}_4$, and the best group performance ρ is 0.73.

7.4 Performance Analysis for Solving CSCD-HC

For no related work on CSCD-HC, two typical approaches, the exhaustion approach and the greedy approach, are employed to compare with the proposed approach in the following experiments. The optimal solution obtained by the exhaustion approach is used as the baseline, and the precise of other two approaches can be calculated as:

$$precise = \rho^* / \rho^{\mathrm{b}}, \tag{21}$$

where ρ^{b} is the group performance of optimal solution obtained by the exhaustion approach and ρ^* represents the group performance obtained by other approach.

The experiments are executed in Dell notebook with Intel i7-6500U processor @2.5 GHz 2.6 Hz and 8G memory, and use MyEclipse (V2015 Stable 1.0) with JavaSE 1.7 in Windows 10 Home (64-bit). In the experiments, let $n=6$, $\tau[j]=0.2$ and $0 \le L[j] \le 3\,(0 \le j < n)$; m changes from 8 to 30 with a step of 2. In each step, the test is repeated for 50 rounds. In each round, Q, L, W, P and C are randomly generated. The proportion of elements assigned 1 in C and the proportion of nonzero elements in P are held within 10 percent. Fig. 4 shows the performance comparison about the precise and execution time of three approaches. The partial data about execution time is displayed in Table 7.

TABLE 7
The Partial Data About Execution Time (ms)

Approaches	m						
	8	10	12	14	16	18	20
Exhaustion approach	8.437	15.060	79.825	340.596	932.385	2740.844	5621.183
Proposed approach	5.000	5.000	5.000	5.000	5.000	5.800	5.600
Greedy approach	0.010	0.016	0.027	0.019	0.034	0.039	0.050

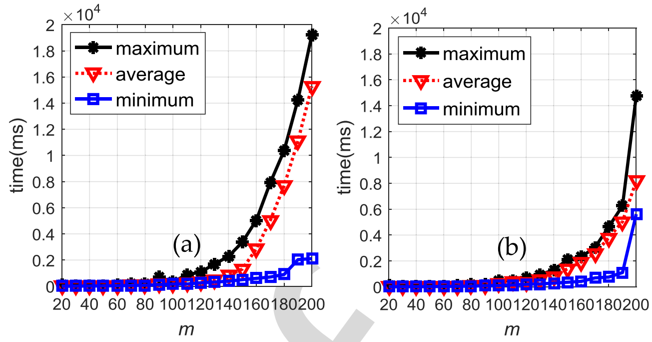


Fig. 5. Performance analysis. (a) $n = m/3$; (b) $n = m/5$.

The execution time of the exhaustion approach shows the trend of rapid growth in Fig. 4b, and it reaches 175.13s when $m = 30$. Especially, its execution time will increase exponentially when n is larger than 6, which means there are more tasks in a DiA. Fig. 4a shows that the greedy approach is enable to gain the larger precise value with the increasing candidate CSs. As shown, the greedy approach could quickly find the feasible solutions with the precise values more than 0.94. However, it cannot ensure to obtain the optimal solution in majority of cases. Fig. 4 demonstrates that the proposed approach is effective to acquire the optimal solution at a lower time cost, compared to the exhaustion approach.

m and n are the most important parameters to determine the problem complexity. The following experiments focus on the performance analysis for verifying the practicability of the proposed approach when m changes from 20 to 200 with a step of 10. To compare the impact of the ratio of n/m, we form two groups of tests whose n/m ratios are $1/3$ and $1/5$, respectively. The results are shown in Fig. 5. Fig. 5 demonstrates that the proposed approach is practical. The larger m and n require more time than that of a group with the smaller m and n. When m is smaller than 140, the time cost is within 2s. Thus, this proposed approach could provide the optimal CS composition solution within an acceptable computation time, and meet the vast majority of application requirements for solving the CSCD-HC problem.

In addition, the number of constraints also greatly affects the solution's performance. Fig. 6 compares the average execution time when the proportion of elements assigned 1 in C and the proportion of nonzero elements in P are confined within 5–20 percent, respectively. It is clear that a greater proportion leads to the dramatic increase in time for finding the feasible solution satisfying all constraints when m is larger than 160.

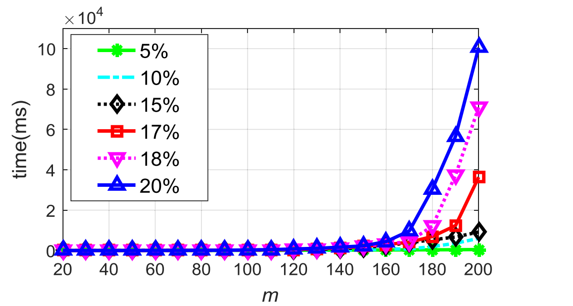


Fig. 6. Performance analysis when $n = m/4, 0 \le L[j] \le 3$, and $0 \le j < n$.

8 Conclusion and Further Study

Aiming at the characteristics of the CSCD-HC problem, this paper proposes a collaborative optimization approach. This approach models a DiA based on CS composition in a hybrid cloud as an RBC system and employs the E-CARGO model to formalize the CSCD-HC problem with the complicated constraints. From the perspective of RBC, the E-CARGO's utilization facilitates to improve the extendibility of the CSCD-HC problem model and the generality of solutions. To deal with the multi-valued QoS evaluations, this paper exploits the cloud model theory to analyze the dynamic performance of CSs, and present a new method utilizing the Mahalanobis distance to improve the similarity measurement of QoS cloud models. The precise assessment of CSs' qualification is available, and provides the strong supports for solving the CSCD-HC problem. The solution using IBM ILOG CPLEX package is put forward to optimize CSCD-HC. The experiments demonstrate that the proposed approach is effective and feasible for solving the CSCD-HC problem.

As for future work, we will study the following problems: (1) the dynamic extensibility mechanisms of CS resources in the sharing resource pool of the hybrid cloud for reducing the possibility of no solution and meeting the increasing requirements from DiAs; (2) the load balancing mechanisms and the parallel task scheduling strategies for enhancing the CS utilization.

Acknowledgments

Thanks go to Mike Brewes of Nipissing University for his assistance in proofreading this article. This work was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) (No. RGPIN2018-04818), Hunan Provincial Natural Science Foundation of China (No. 2017JJ2186) and MOE (Ministry of Education in China) Humanities and Social Science Research Youth Fund Project (No. 18YJCZH124).

References

[1] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 314–347, 2014.

[2] Z. Cai, X. Li, and J. N. Gupta, "Heuristics for provisioning services to workflows in XaaS Clouds," *IEEE Trans. Services Computing*, vol. 9, no. 2, pp. 250–263, Mar./Apr. 2016.

[3] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A survey of data-intensive scientific workflow management," *J. Grid Comput.*, vol. 13, no. 4, pp. 457–493, 2015.

[4] F. Amato and F. Moscato, "Automatic cloud services composition for big data management," in *Proc. 30th Int. Conf. Advanced Inf. Netw. Appl. Workshops*, 2016, pp. 46–51.

[5] Y. Zhang, H. Huang, Y. Xiang, L. Y. Zhang, and X. He, "Harnessing the hybrid cloud for secure big image data service," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1380–1388, Oct. 2017.

[6] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen, "Proactive workload management in hybrid cloud computing," *IEEE Trans. Netw. Serv. Manage.*, vol. 11, no. 1, pp. 90–100, Mar. 2014.

[7] Y. Niu, B. Luo, F. Liu, J. Liu, and B. Li, "When hybrid cloud meets flash crowd: Towards cost-effective service provisioning," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1044–1052.

[8] D. Huang, Y. Du, and Q. He, "Migration algorithm for big marine data in hybrid cloud storage," *J. Comput. Res. Develop.*, vol. 51, no. 1, pp. 199–205, 2014.

[9] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Generation Comput. Syst.*, vol. 79, pp. 849–861, 2018.

[10] S. Li, Y. Zhou, L. Jiao, X. Yan, X. Wang, and M. R.-T. Lyu, "Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization," *IEEE Trans. Serv. Comput.*, vol. 8, no. 3, pp. 398–409, 2015.

[11] H. Ma, Z. Hu, K. Li, and H. Zhang, "Toward trustworthy cloud service selection: A time-aware approach using interval neutrosophic set," *J. Parallel Distrib. Comput.*, vol. 96, pp. 75–94, 2016.

[12] H. Ma, Z. Hu, L. Yang, and T. Song, "User feature-aware trustworthiness measurement of cloud services via evidence synthesis for potential users," *J. Visual Languages Comput.*, vol. 25, no. 6, pp. 791–799, 2014.

[13] P. Casas and R. Schatz, "Quality of experience in cloud services: Survey and measurements," *Comput. Netw.*, vol. 68, no. 1, pp. 149–165, 2014.

[14] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Math. Comput. Model.*, vol. 57, no. 11, pp. 2883–2894, 2013.

[15] H. Ma, H. Zhu, Z. Hu, W. Tang, and P. Dong, "Multi-valued collaborative QoS prediction for cloud service via time series analysis," *Future Generation Comput. Syst.*, vol. 68, no. 3, pp. 275–288, 2017.

[16] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. Netto, and R. Buyya, "Big data computing and clouds: Trends and future directions," *J. Parallel Distrib. Comput.*, vol. 79, pp. 3–15, 2015.

[17] H. Wang, L. Wang, Q. Yu, and Z. Zheng, "Learning the evolution regularities for big service-oriented online reliability prediction," *IEEE Trans. Serv. Comput.*, 2017, doi: 10.1109/TSC.2016.2633264.

[18] X. Liang and B. Tang, "The coordination control without model of cloud computing services composition systems," *J. Syst. Sci. Math. Sci.*, vol. 34, no. 10, pp. 1172–1181, 2014.

[19] S. Deng, H. Wu, D. Hu, and J. L. Zhao, "Service selection for composition with QoS correlations," *IEEE Trans. Serv. Comput.*, vol. 9, no. 2, pp. 291–303, Mar./Apr. 2016.

[20] S. Deng, L. Huang, D. Hu, J. L. Zhao, and Z. Wu, "Mobility-enabled service selection for composite services," *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 394–407, May/Jun. 2016.

[21] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: Towards privacy-aware cross-cloud service composition for big data applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 455–466, Feb. 2015.

[22] W. Chen and I. Paik, "Toward better quality of service composition based on a global social service network," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1466–1476, May 2015.

[23] S. Deng, H. Wu, J. Taheri, and A. Y. Zomaya, "Cost performance driven service mashup: A developer perspective," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 8, pp. 2234–2247, Aug. 2016.

[24] J. Siebert, J. Cao, Y. Lai, P. Guo, and W. Zhu, "LASEC: A localized approach to service composition in pervasive computing environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 7, pp. 1948–1957, Jul. 2015.

[25] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst. Man Cybern. - Part C: Appl. Rev.*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[26] H. Zhu, "Role mechanisms in collaborative systems," *Int. J. Prod. Res.*, vol. 44, no. 1, pp. 181–193, 2006.

[27] H. Zhu, M. Zhou, and R. Alkins, "Group role assignment via a Kuhn–Munkres algorithm-based solution," *IEEE Trans. Syst. Man. Cybern. - Part A: Syst. Hum.*, vol. 42, no. 3, pp. 739–750, May 2012.

[28] Y. Sheng, H. Zhu, X. Zhou, and W. Hu, "Effective approaches to adaptive collaboration via dynamic role assignment," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 46, no. 1, pp. 76–92, Jan. 2016.

[29] H. Zhu, "Avoiding conflicts by group role assignment," *IEEE Trans. Syst. Man. Cybern.: Syst.*, vol. 46, no. 4, pp. 535–547, Apr. 2016.

[30] H. Zhu, Y. Sheng, X. Zhou, and Y. Zhu, "Group role assignment with cooperation and conflict factors," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 48, no. 6, pp. 851–863, Jun. 2018, doi: 10.1109/TSMC.2016.2633244.

[31] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the many to many assignment problem by improving the Kuhn–Munkres algorithm with backtracking," *Theoretical Comput. Sci.*, vol. 618, pp. 30–41, 2016.

[32] H. Zhu, D. Liu, S. Zhang, and S. Teng, "Solving the group multi-role assignment problem by improving the ILOG approach," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 47, no. 12, pp. 3418–3424, Dec. 2017, doi: 10.1109/TSMC.2016.2566680.

[33] A. Vakili and N. J. Navimipour, "Comprehensive and systematic review of the service composition mechanisms in the cloud environments," *J. Netw. Comput. Appl.*, vol. 81, pp. 24–36, 2017.

[34] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809–3824, 2014.

[35] S. Liu, L. Kong, K. Ren, J. Song, K. Deng, and H. Leng, "A two-step data placement and task scheduling strategy for optimizing scientific workflow performance on cloud computing platform," *Chin. J. Comput.*, vol. 34, no. 11, pp. 2121–2130, 2011.

[36] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: A survey," *Future Generation Comput. Syst.*, vol. 56, pp. 684–700, 2016.

[37] Q. Wu, F. Ishikawa, Q. Zhu, and D.-H. Shin, "QoS-aware multi-granularity service composition: Modeling and optimization," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 46, no. 11, pp. 1565–1577, Nov. 2016.

[38] Y. Zhang, G. Cui, S. Deng, and Q. He, "Alliance-aware service composition based on quotient space," in *Proc. IEEE Int. Conf. Web Serv.*, 2016, pp. 340–347.

[39] Q. Yu, L. Chen, and B. Li, "Ant colony optimization applied to web service compositions in cloud computing," *Comput. Elect. Eng.*, vol. 41, pp. 18–27, 2015.

[40] D. Yuan, Y. Yang, X. Liu, and J. Chen, "On-demand minimum cost benchmarking for intermediate dataset storage in scientific cloud workflow systems," *J. Parallel Distrib. Comput.*, vol. 71, no. 2, pp. 316–332, 2011.

[41] D. Li, H. Meng, and X. Shi, "Membership clouds and membership cloud generators," *J. Comput. Res. Develop.*, vol. 32, no. 6, pp. 15–20, 1995.

[42] G. Wang, C. Xu, and D. Li, "Generic normal cloud model," *Inf. Sci.*, vol. 280, pp. 1–15, 2014.

[43] H. Ma, H. Zhu, Z. Hu, K. Li, and W. Tang, "Time-aware trustworthiness ranking prediction for cloud services using interval neutrosophic set and ELECTRE," *Knowl.-Based Syst.*, vol. 138, pp. 27–45, 2017.

[44] X. Shang, P. Ma, and T. Chao, "Performance evaluation of electromagnetic railgun exterior ballistics based on cloud model," *IEEE Trans. Plasma Sci.*, vol. 45, no. 7, pp. 1614–1621, Jul. 2017.

[45] H. Peng, H. Zhang, and J. Wang, "Cloud decision support model for selecting hotels on TripAdvisor.com with probabilistic linguistic information," *Int. J. Hospitality Manage.*, vol. 68, pp. 124–138, 2018.

[46] H. Li, C. Guo, and W. Qiu, "Similarity measurement between normal cloud models," *Acta Electronica Sinica*, vol. 39, no. 11, pp. 2561–2567, 2011.

[47] H. Li and C. Guo, "Piecewise aggregate approximation method based on cloud model for time series," *Control Decision*, vol. 26, no. 10, pp. 1525–1529, 2011.

[48] S. Wang, L. Zhang, S. Wang, and X. Qiu, "A cloud-based trust model for evaluating quality of web services," *J. Comput. Sci. Technol.*, vol. 25, no. 6, pp. 1130–1142, 2010.

[49] L. Liu, H. Zhu, and Z. Huang, "Analysis of the minimal privacy disclosure for web services collaborations with role mechanisms," *Expert Syst. Appl.*, vol. 38, no. 4, pp. 4540–4549, 2011.

[50] H. Zhu, M. Hou, C. Wang, and M. Zhou, "An efficient outpatient scheduling approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 4, pp. 701–709, Oct. 2012.

[51] J. Li, Z. Shen, and X. Meng, "Scientific big data management: Concepts, technologies and system," *J. Comput. Res. Develop.*, vol. 54, no. 2, pp. 235–247, 2017.

[52] C. Liu, M. Feng, X. Dai, and D. Li, "A new algorithm of backward cloud," *J. Syst. Simul.*, vol. 16, no. 11, pp. 2417–2420, 2004.

[53] Y. Zhang, D. Zhao, and D. Li, "The similar cloud and the measurement method," *Inf. Control*, vol. 33, no. 2, pp. 129–132, 2004.

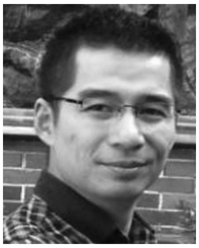
[54] G. Zhang, D. Li, P. Li, J. Kang, and G. Chen, "A collaborative filtering recommendation algorithm based on cloud model," *J. Softw.*, vol. 18, no. 10, pp. 2403–2411, 2007.

[55] L. Zhang, Y. Yang, and X. Zhao, "SaaS decision-making method based on cloud model," *Acta Electronica Sinica*, vol. 43, no. 5, pp. 987–992, 2015.

[56] L. Abdi and S. Hashemi, "To combat multi-class imbalanced problems by means of over-sampling techniques," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[57] F. Guo, W. Susilo, and Y. Mu, "Distance-based encryption: How to embed fuzziness in biometric-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 247–257, Feb. 2016.

[58] H. Ma, Z. Hu, and M. Cai, "Trustworthy service selection integrating cloud model and possibility degree ranking of interval numbers," *Chin. J. Electron.*, vol. 26, no. 6, pp. 1177–1183, 2017.



Hua Ma received the BS degree in computer science and technology, the MS degree in computer application technology, and the PhD degree in software engineering all from Central South University, Changsha, China. His research interests focus on cloud computing and high performance computing. He is currently an associate professor with the College of Information Science and Engineering, Hunan Normal University, Changsha, China. He is a member of the IEEE.

Haibin Zhu is a full professor of the Department of Computer Science and Mathematics, founding director of the Collaborative Systems Laboratory, Nipissing University, Canada. He has more than 150 research publications. He is a senior member of the IEEE and is serving and served as co-chair of the technical committee of *Distributed Intelligent Systems* of IEEE SMC Society, and associate editor of *IEEE SMC Magazine*. He is a founding researcher of Role-Based Collaboration and Adaptive Collaboration.

Keqin Li is a SUNY distinguished professor of computer science in the State University of New York. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPUGPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of things and cyber-physical systems. He has published more than 600 journal articles, book chapters, and refereed conference papers, and has received several best paper awards. He is currently serving or has served on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Transactions on Computers*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Services Computing*, and *IEEE Transactions on Sustainable Computing*. He is a fellow of the IEEE.

Wensheng Tang received the BS degree from Hunan Normal University, Changsha, China, in 1992, the MS and PhD degrees from the National University of Defense Technology, Changsha, China, in 1997 and in 2009, respectively. His research interests focus on the distributed computing and cloud computing. He is currently a professor with the College of Information Science and Engineering, Hunan Normal University, Changsha, China.