International Journal of Automation and Computing 8(3), August 2011, 297-308
DOI: 10.1007/s11633-011-0585-1

Trustworthy Assurance of Service Interoperation in Cloud Environment

Bing Li[1] Bu-Qing Cao[1, 3] Kun-Mei Wen[2] Rui-Xuan Li[2]

[1]State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, PRC

[2]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, PRC

[3]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, PRC

Abstract: Cloud computing can be realized by service interoperation and its essence is to provide cloud services through network. The development of effective methods to assure the trustworthiness of service interoperation in cloud environment is a very important problem. The essence of cloud security is trust and trust management. Combining quality of service (QoS) with trust model, this paper constructs a QoS-aware and quantitative trust-model that consists of initial trust value, direct trust value, and recommendatory trust value of service, making the provision, discovery, and aggregation of cloud services trustworthy. Hence, it can assure trustworthiness of service interoperation between users and services or among services in cloud environment. At the same time, based on this model, service discovery method based on QoS-aware and quantitative trust-model (TQoS-WSD) is proposed, which makes a solid trust relationship among service requestor, service provider and service recommender, and users can find trustworthy service whose total evaluation value is higher. Compared to QoS-based service discovery (QoS-WSD) method, it is proved by the experiment for TQoS-WSD method that more accurate result of service discovery will be achieved by service requestor, while reasonable time cost is increased. Meanwhile, TQoS-WSD method strongly resists the effect of service discovery by untrustworthy QoS values and improves service invocation success-rate and thus assures trustworthiness of services interoperation.

Keywords: Quality of service (QoS), trust, service discovery, service interoperation, service invocation success-rate, cloud computing.

1 Introduction

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services[1]. Its goal is to regard "computing power" as a public infrastructure, like water, electricity, financial systems, which can meet personal and social needs of information services with low-cost and high-efficiency, by the aggregation of large-scale calculation, storage, and communication resources, and to make "information" as a productive force of social progress, like material and energy. Socialization, intensification, and specialization are the three basic characteristics of cloud computing. Among them, socialization points that cloud computing is a computing model based on internet and public-participation, which provides various forms of cloud services resources, such as web services, application programming interface (API), infrastructure as a service (IaaS), platform as a service (Paas), and software as a service (SaaS) to users by building the infrastructure of services to publish, discovery, aggregation, and transaction. Intensification points that cloud computing can integrate and optimize various cloud services resources, such as computing, storage, and network resources, which enhance depth-sharing and reuse of cloud services resources, bringing scale-advantage, reduce costs, and quick response to requirements. Specialization points that cloud computing divides service resources into different categories to meet the requirements of individual users and realize fine, specialized, and domain-oriented service aggregation. Thus, cloud computing makes information services on internet social, intensive, and special, like the industrial production of material and energy, which can reduce overall operating costs of all society′s economy and promote the society to the development of "service" and "resource conservation".

Interconnection, intercommunication, and interoperability always has been a hot research topic for computer and network workers. Internet resolves network interconnection problem by TCP/IP protocol, world wide web, resolves information intercommunication problem by HTTP and HTML, likewise cloud computing can be realized by service interoperation, and its essence is to provide cloud services by network. Service interoperation refers to the capability of mutual cooperation and collaborative work between users and cloud services or among cloud services. Cloud services are massive, heterogeneous, distributed, and aggregated, and then, in order to accomplish the task given by users, the exchange of information, collaborative work, and mutual cooperation will happen between users and cloud services, so service interoperation between users and cloud services in cloud environment becomes very important. Meanwhile, the service provision capacity of a single service provider is limited, and it is necessary to aggregate some services provided by multiple service providers to meet users′ requirements. Thus, service interoperation in cloud environment among cloud services is necessary and important, which can realize service provision with larger area and more resources.

Manuscript received July 31, 2010; revised March 7, 2011

This work was supported by National Basic Research Program of China (973 Program) (No. 2007CB310801), National Natural Science Foundation of China (No. 60873083, No. 60803025, No. 60970017, No. 60903034, No. 60873225), Natural Science Foundation of Hubei Province for Distinguished Young Scholars (No. 2008CDB351), Natural Science Foundation of Hubei Province (No. 2008ABA358, No. 2008ABA379), Research Fund for the Doctoral Program of Higher Education of China (No. 20070486065), and Open Foundation of State Key Laboratory of Software Engineering (No. SKLSE20080718).

Currently, it has become a hot topic for service interoperation research, which includes service provision, discovery, and aggregation based on quality of service (QoS) in cloud environment. Among these, combined with service-oriented architecture (SOA) and multiagent technology, this paper proposes a new service-oriented QoS-assured cloud computing architecture, which includes physical device and virtual resource layer, cloud service provision layer, cloud service management, and multiagent layer to support QoS-assured cloud service provision, discovery, and interoperability research[2]. Rochwerger et al.[3] proposed a modular extensible cloud architecture with intrinsic support for business service management (BSM) and federation of clouds for interoperability among cloud providers, and its goal is to facilitate an open, service-based, online economy, where resources and services are transparently provided and managed across clouds in an on-demand basis at competitive costs with high QoS. Li et al.[4] presented a method for achieving optimization in clouds by using performance models in the development, deployment, and operations of the applications running in the cloud. The optimization here maximizes profits in the cloud constrained by QoS and service-level agreements (SLAs) across a large variety of workloads. Buyya et al.[5] pointed that consumers rely on cloud providers to supply all their computing needs, and they will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations. Cloud providers will need to consider and meet different QoS parameters of each individual consumer, as negotiated in specific SLAs. Zhang et al.[6] proposed a service composition approach based on sequence mining for migrating E-learning legacy system to SOA. Yu et al.[7] presented efficient algorithms for web services selection with end-to-end QoS constraints. Meanwhile, many other QoS-based web service selection methods have also been proposed in [8-10].

However, those works mostly assume that cloud services are trustworthy and consider that QoS values given by service providers and users are true and trustworthy. In fact, cloud environment is open, uncertain, and deceptive. In order to obtain users′ trust, service providers may provide incomplete, false, and even vicious service QoS values because of some unwarrantable interests. Meanwhile, some users will become service recommenders after they invoke service, and the feedback values of QoS (such as, availability, reliability, response time, etc.) given by service recommenders may not be their true wish. Thus, service requestors cannot completely trust the claimed QoS values provided by service providers and the feedback values of QoS given by service recommenders[11]. It is very difficult to find services that meet users′ requirements and trustworthiness, and services interoperation will also become untrustworthy, resulting in unsuccessful collaboration.

Therefore, it is a very important problem how to assure trustworthiness of service interoperation in cloud environment. The essence of cloud security is trust and trust management. In recent years, it has become a hot topic to make use of trust and trust model for selecting trustworthy services and the research of service interoperation trustworthiness[12, 13]. Li and Ping[14] proposed a trust model to enhance security and interoperability of cloud environment, in which cloud customer can choose different providers of services, and resources in heterogeneous domains can be cooperated. Simulation experiments show that the proposed model can establish a trust relationship between customer and provider and between different cloud platforms fast and safely. Maximilien and Singh[15] proposed a multiagent web service trust and selection method, which uses the model based on architecture and planning, and multiagent expresses specific application services. Cai et al.[16] attempted to build and maintain a "web of trust" by service registration repertory and obtain trustworthy users from it. Then, the trust degree of users can be obtained, which makes use of proposed Kalman filter algorithm to carry out excavation on "the web of trust" and analyze each user′s trust degree. Finally, the feedback of web services provided by the users with higher trustworthiness will be adopted, which makes web services selection process easier. Based on trust evaluation, from the perspective of software services selection, Wang et al.[17] proposed a networked-software architecture-oriented trust-driven mechanism to select service. Xu et al.[18] proposed a service discovery model based on QoS-enhanced trust, which makes use of matching and ranking web services to select the best service. Vu et al.[19] proposed a trust and reputation management method for QoS-based semantic service discovery and recommendation. These works consider the use of trust and trust model for services trustworthy selection and the research of service interoperation trustworthiness. However, they are all inadequate for the following aspects. First, how to properly initialize the initial trust value of service that has no interaction record and establish the initial trust relation, which makes service requester choose the service whose QoS values are higher in the first instance. Second, how to modify claimed QoS values of service providers and consider the dynamic attenuation of trust to accurately calculate QoS-aware direct trust value of service. Third, how to reasonably quantify recommendation weight and QoS feedback values of service recommender and effectively eliminate vicious recommendation, in order to accurately calculate QoS-aware recommendatory trust value of service.

To sum it up, cloud computing can be realized by service interoperation, and its essence is to provide cloud services by network. State Key Laboratory of Software Engineering of Wuhan University has proposed service interoperation metamodel framework based on role-goal-process-service (RGPS) from the perspective of role, goal, process, and service to research related key technologies and a series of international standards, which provides theoretical support and technical assurance for service provision, discovery, and aggregation. Meanwhile, based on the mode of registration, ontology, and mapping, related registration standard for metamodel framework for interoperability (MFI) has been established by making use of RGPS. Because of security problem of cloud environment, the interoperation between users and cloud services (that is represented by the provision and discovery of services) and the interoperation among cloud services (that is represented by the aggregation of services) will become untrustworthy. To solve these, under the guidance of service interoperation metamodel framework based on RGPS and MFI, combin-

ing QoS with trust model, this paper constructs a QoS-aware and quantitative trust model (TQoS-WSD), which consists of initial trust value, direct trust value, and recommendatory trust value of service, which make the provision, discovery, and aggregation of services trustworthy, assuring trustworthiness of service interoperation. First, initial trust value of service can be calculated by claimed QoS values of not-interactive-record published service. Second, taking the dynamic nature of trust into account, we use attenuation function, which takes interval and interaction-times as parameters, and successful historical interaction-times and unsuccessful historical interaction-times to calculate direct trust value of service. Third, in order to resist the vicious recommendation, an improved cross generation, heterogeneous recombination, and cataclysmic mutation (CHC) genetic algorithm is designed to extract and select optimum trust paths, and their trust values will be calculated and regarded as recommendation weight composing of service recommender, and then, recommendatory trust value of service will be gained. Finally, according to users′ weight, the total evaluation value of service can be synthesized by initial trust value, direct trust value, and recommendatory trust value of service. In that way, making use of TQoS-WSD to discover services, a solid trust relationship will be built among service requestor, service provider, and service recommender. Users can find trustworthy services whose total evaluation is higher, and then, they may be trustworthy aggregated. Therefore, the interoperation between users and cloud services or among cloud services becomes trustworthy.

Section 2 introduces service interoperation metamodel framework based on RGPS and MFI. Section 3 describes TQoS-WSD that consists of initial trust value, direct trust value, and recommendatory trust value of service in cloud environment. Section 4 proposes a service discovery method based on TQoS-WSD and discusses trust update, evaluation, and trustworthy assurance of service interoperation. Section 5 describes experiments that evaluate feasibility and effectiveness of TQoS-WSD. Finally, the content of the whole paper is summarized, and expectation is proposed in Section 6.

2 Service interoperation metamodel framework based on RGPS and MFI

The interoperability research among software systems has always been a hot topic. Chen et al.[20] proposed an ontology-based semantic interoperability framework of product data, and its feasibility and effectiveness has been demonstrated by a prototype system and an application instance. Lv et al.[21] proposed a heterogeneous system interoperation framework based on multilayer ontology to solve the problems of integrating heterogeneous systems during the business collaboration. Di and Fan[22] proposed a unified description of data model based on ontology to solve the gap in interoperation through mapping from logical data of enterprise′s information system. From these interoperability researches, it can be known that the businesses and organizations of various domains, all attempt to solve the integration and collaboration by making use of interoperability to satisfy their requirements. Currently, cloud computing is a computing model based on the internet, which can be realized by service interoperation, and its essence is to provide cloud services by network, thus service interoperation among software systems becomes very necessary and important to satisfy all kinds of requirements. Service interoperation metamodel framework based on RGPS from the perspective of role, goal, process, and service to research the related key technologies and a series of international standards, which provides theoretical support and technical assurance for service provision, discovery, and aggregation in cloud environment[23], is shown in Fig. 1.

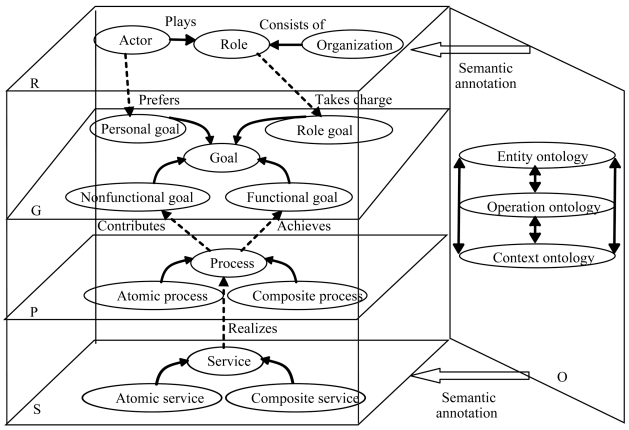


Fig. 1 Service interoperation meta-model based on RGPS

Making use of RGPS idea, ISO/IEC 19763 (MFI) has been built to solve service interoperation problem from the perspective of registration. Among MFI, ISO/IEC 19763-3 refers to the registration and management of domain ontology. Currently, a metamodel framework of service interoperation registration and management has been realized based on this ISO/IEC 19763 standard, which can realize service interoperation between users and web services or among web services, such as service discovery, aggregation[24]. However, it is possible that this service interoperation between users and cloud services is untrustworthy. Thus, it is necessary and important to make the provision, discovery, and aggregation of services trustworthy, assuring trustworthiness of service interoperation.

3 QoS-aware and quantitative trust-model (TQoS-WSD)

Web service is the basic element of distributed software applications, with the characteristics of self-contained, self-describing, modular, and loosely coupled, and so, it can be regarded as a black box. For simplicity, web service will be taken as an example of cloud services to build trust model. It is possible for service interoperation between users and web services, that is, service discovery problem for service computing and cloud computing is untrustworthy. Making use of some mathematics formula and computer algorithm (e.g., genetic algorithm) to build a TQoS-WSD, service discovery will become computable, so users can select trustworthy services, that is, service interoperation between users and services becomes trustworthy. Service entities provided by service providers are expressed as *Serv*, and their claimed QoS values are denoted by a set

of $A(a_1, a_2, \cdots, a_n)$; here, $0 \leqslant a_i \leqslant 1$, $1 \leqslant i \leqslant n$, and a_i represents dimensionless normalized QoS value of service. Service requestors are expressed as Req, and their preferred weights of QoS values are denoted by the set of $W(w_1, w_2, \cdots, w_n)$, here, $0 \leqslant w_i \leqslant 1$, $1 \leqslant i \leqslant n$, and $\sum_{i=1}^{n} w_i = 1$. Service recommenders are denoted by Rec_j, $1 \leqslant j \leqslant n$.

3.1 Initial trust value of service

Definition 1. Initial trust value of service is defined as the degree of trust of services provided or published by services provider while having no interaction record. It is denoted by $T_{\text{initial}}(Req, Serv)$. Here, based on claimed QoS values of published service, the initial trust value of service can be calculated as follows:

$$T_{\text{initial}}(Req, Serv) = \sum w_i \times a_i, \quad 1 \leqslant i \leqslant n. \tag{1}$$

By (1), the initial trust value of the newly published service can be calculated to avoid distrust because of noninteractive record. Meanwhile, service requestors can compare the initial trust value of services, which has the same functionality and has no interactive record, to select some services whose initial trust values are higher. Thus, the initial trust relationship can be built, which will improve QoS and trustworthiness of service.

3.2 Direct trust value of service

Definition 2. In specific circumstances, direct trust value of service points to the degree of trust that one service entity trusts another service entity according to direct historical interaction record, which is denoted by $T_{\text{direct}}(Req, Serv)$. As far as cloud environment is concerned, it represents the trust value generated by many historical interactions between service requestor and service.

QoS values are acquired by a direct historical interaction between service requestor and service, which can be denoted by the set of $D(d_1, d_2, \cdots, d_n)$; here, $0 \leqslant d_i \leqslant 1$, $1 \leqslant i \leqslant n$, and d_i represents dimensionless normalized QoS value of service. If a direct historical interaction between service requestor and service is successful; rewarded trust will be gained and denoted by S_x; here, $0 \leqslant S_x \leqslant 1$, $1 \leqslant x \leqslant P$, P represents successful historical interaction-times, and it can be calculated as follows:

$$S_x = 1 - \sqrt{\frac{\sum_{i=1}^{n} w_i (d_i - a_i)^2}{n}}. \tag{2}$$

On the contrary, if a direct historical interaction between service requestor and service is unsuccessful, punished trust will be gained and denoted by F_y; here, $-1 \leqslant F_y \leqslant 0$, $1 \leqslant y \leqslant Q$, Q represents unsuccessful historical interaction-times, and it can be calculated as $F_y = -k \times S_x$; here, k is penalty factor, $0 \leqslant k \leqslant 1$, and the reduction of trust value is more when k is greater.

In addition, trust will be dynamic attenuation, i.e., an agelong trust relationship will become attenuated over time. The trust relationship generated by recent interaction will be more solid, and so, it shall be given more weight in the calculation of direct trust value of service. Thus, attenuation function that takes interval and interaction-times as parameters are proposed as follows:

$$\lambda(j, t_j) = \begin{cases} 1, & j = l, t_j = t_l \\ \lambda(j-1, t_{j-1}) = \lambda(j, t_j) - \dfrac{t_j - t_{j-1}}{t_l - t_1}, & 1 \leqslant j \leqslant l, t_1 \leqslant t_j \leqslant t_l \end{cases} \tag{3}$$

where $j \in [1, l]$ represents interaction-times variable by order of history time, l represents total historical interaction-times, and $t_j \in [t_1, t_l]$ represents the time when historical j-th interaction-time occurs. Here, $\lambda(j, t_j) \in [0, 1]$, when recent l-th interaction-time occurs, attenuation function value is equal to 1, that is, trust has no attenuation. Attenuation function value acquired by agelong historical interaction will decrease, and it shall be given lower weight in the calculation of direct trust value of service.

Compared to other models[25−27], trust computation is based on interval and interaction-times, which can fully reflect the dynamic characteristics of trust relationship attenuation over time. If the historical interaction-times are more and the interval is smaller, trust attenuation is slow. Otherwise, trust attenuation is fast. By this, it increases the quantitative accuracy and dynamic adaptability of trust.

To sum it up, taking the dynamic characteristics of trust into account, we use attenuation function, which takes interval and interaction-times as parameters, and successful historical interaction-times and unsuccessful historical interaction-times, to calculate direct trust value of service as follows:

$$T_{\text{direct}}(Req, Serv) = \begin{cases} \dfrac{\sum_{x=1}^{P} (S_x \times \lambda(x, t_x)) + \sum_{y=1}^{Q} (F_y \times \lambda(y, t_y))}{P + Q}, & P, Q > 0 \\ 0, & P = Q = 0. \end{cases} \tag{4}$$

3.3 Recommendatory trust value of service

The feedback values of QoS given by service recommenders are possibly vicious or false data because of some known factors, while the average method has no effective standard to measure those QoS data[28]. Those untrustworthy QoS data will directly affect accuracy and reliability of QoS computation and final results of service selection. Therefore, how to reasonably quantify recommendation weight and feedback values of service recommender and effectively eliminate vicious recommendation, in order to more accurately calculate QoS-aware recommendatory trust value of service and improve accuracy and trustworthiness of service selection, has become very important. Li and Gui[29] has considered feedback′s level factor, where the recommendation weight of service recommender can be acquired by feedback weighting function whose value is equal to trust value of trust path. However, it has not considered how to extract effective and optimum trust paths when there are many trust paths. Liu et al.[30] has considered the existence of many trust paths, but it has used a simple arithmetic average method to calculate trust value of trust path

and has not considered the reliability of trust path, and so, it cannot effectively prevent vicious recommendation.

Therefore, this paper presents a computation method for finding the recommendation weight of service recommender. First of all, an improved CHC genetic algorithm is designed for the extraction and selection of the optimum trust paths from many trust paths between service requestor and service recommender. Then, considering joint-cheat problem in the optimum trust paths, these optimum trust paths will be sorted by their trust values and threshold value will be set, in order to select some optimum and reliable trust paths. Finally, the trust values of these optimum and reliable trust paths will be synthesized to take as recommendation weight of service recommender.

3.3.1 An improved CHC genetic algorithm for extraction and selection of optimum trust paths

Definition 3. Trust path points to direct trust relationship pathway from service requestor to service recommender, denoted by path.

Definition 4. The trust value of trust path points to the trust value of trust path from service requestor to service recommender, denoted by T_{Path}. It is a direct trust relationship among service entities included in this trust path, so T_{Path} can be calculated as follows:

$$T_{\mathrm{Path}}(Req \to \cdots \to Med_i \to \cdots \to Rec_j) = \prod_{\substack{D_g, D_k \in \{Req, Rec_j, Med_i\} \\ 1 \leqslant i \leqslant n,\ D_k \in Succ(D_g)}} T_{\mathrm{direct}}(D_g, D_k), \quad 1 \leqslant j \leqslant n. \tag{5}$$

Among them, $D_k \in Succ(D_g)$ represents that D_k is a direct successor entity of D_g in this trust path. $T_{\mathrm{direct}}(D_g, D_k)$ represents the direct trust value from D_g to D_k, which can be gained by (4). Med_i, $1 \leqslant i \leqslant n$, represents middle service entities in this trust path. Rec_j, $1 \leqslant j \leqslant n$, represents service recommender.

It is possible that the number of trust paths from service requestor to service recommender is zero, one, or more, which constitute a complex trust paths network. Thus, the time and space complexity to find all trust paths are relatively high. At the same time, different trust paths are different in certain aspects, such as length, reliability, etc. It is possible that some long trust paths only have small contribution to their T_{Path}. As a result, it is necessary to find some optimum trust paths whose lengths are limited, which not only can represent the characteristic of trust but also can prevent joint cheat. Therefore, this paper proposes to make use of improved CHC genetic algorithm to find some optimum trust paths whose lengths are limited, and their T_{Path} are higher.

Trust paths network can be expressed in the form of directed acyclic graph, that is, $G = ((V, E, W), Maxlength)$. In this expression, node $V \in \{Req, Rec_j, Med_i\}$ represents various kinds of service entities. E represents directed edge set from Req to Rec_j, which points to the direct trust relationship among service entities, and so, trust paths can be built by it. W represents weight value of directed edge E, which expresses the direct trust relation, and its value is equal to the direct trust value among service entities, such as $T_{\mathrm{direct}}(D_g, D_k)$, $W \in (0, 1)$. $Maxlength$ represents maximum length of trust path. A simple trust paths network expressed by directed acyclic graph G can be shown as in Fig. 2, and the number of trust paths from Req to Rec_j is six. Among this, $T_{\mathrm{Path}}(Req \to Med_2 \to Rec_j) = 0.6 \times 0.4 = 0.24$, $T_{\mathrm{Path}}(Req \to Med_1 \to Med_7 \to Rec_j) = 0.4 \times 0.6 \times 0.9 = 0.216$, and $T_{\mathrm{Path}}(Req \to Med_3 \to Med_8 \to Rec_j) = 0.8 \times 0.5 \times 0.4 = 0.16$, $T_{\mathrm{Path}}(Req \to Med_1 \to Med_7 \to Rec_j) = 0.4 \times 0.6 \times 0.9 = 0.216$.

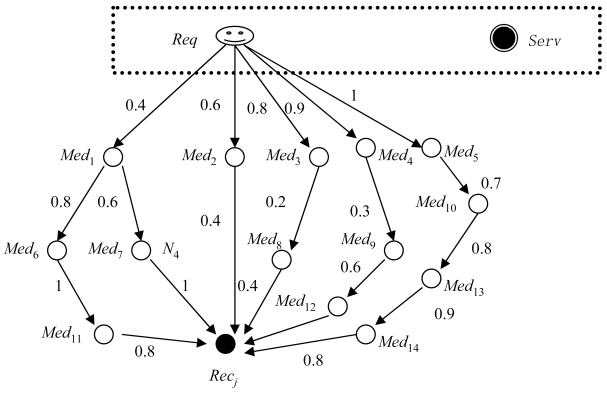


Fig. 2 Trust paths network

As mentioned above, an improved CHC genetic algorithm is designed to extract and select the optimum trust paths, whose steps are as follows:

Step 1. Gene encoding. Before carrying through search, data of solutions space will be represented as genotype data structure of genetic space by genetic algorithm. Because of making use of directed acyclic graph to find the optimum paths problem, the individual or chromosome will correspond to a path from Req to Rec_j in G. Each gene value of individual will be expressed by decision variable value of path. Here, decision variable, that is, service node number in path, can be expressed by real-number encoding. The encoding length of individual is equal to the number of decision variables. Because path length is variable, the length of individual or chromosome is also variable and is set to less than or equal to $Maxlength$.

Step 2. Population initialization. If the size of population is too large, fitness evaluation times will increase, which affects algorithmic performance. If the size of population is too small, this will make search space of genetic algorithm limited and cause premature convergence. Hereon, first, certain number of individuals will randomly generate. Second, based on the length of trust path from 1 to $Maxlength$, the corresponding individuals can be selected with the same proportion selected and joined in the initial population. This process will be iterative until the size of initial population reaches predetermined size, which is set to N_1.

Step 3. Fitness function selection. Genetic algorithm does not make use of external information to search, but only is dependent on fitness function value of individuals. Therefore, fitness function selection is very important, which directly affects the convergence speed of genetic algorithm and the optimal solution selection. The goal of problem is to find optimum trust paths whose length are limited from many trust paths between service requestor and ser-

vice recommender, that is, objective function is equal to trust value of a single trust path T_{Path}, and it can be used as fitness function:

$$f(x) = \prod_{\substack{D_g, D_k \in \{Req, Rec_j, Med_i\} \\ 1 \leqslant i \leqslant n,\ D_k \in Succ(D_g)}} T_{\text{direct}}(D_g, D_k). \quad (6)$$

In (6), $1 \leqslant x \leqslant N_1$ expresses the number of individuals or chromosomes, $f(x)$, namely, the fitness value of individual x. The higher the fitness value, the corresponding trust value of the trust path T_{Path} is greater.

Step 4. Genetic operations. It includes selection, crossover, and mutation, and other genetic operation:

1) Selection

Cross generational elitist selection strategy is adopted for selection operation, which mixes up previous-generation population and individuals population generated by new crossover approach, calculates, and sorts individual′s fitness value, then select N_1 individuals by order of decreasing fitness value and form next-generation population.

2) Crossover

The crossover operation of CHC algorithm is an improvement of uniform crossover. When different amount of bits of two parent individuals is equal to U, $U/2$ positions will be randomly selected, and then the bits value of two parent individuals in these positions will be exchanged. Because the length of individual or chromosome is variable, its hamming-distance calculation and matching is complex. Therefore, first, the mixing up of current-generation population and previous-generation population will take place. Second, some individuals will be randomly selected in the mixed population, which carry multipoint crossover operation according to a certain crossover probability, that is, $U/2$ positions will be randomly selected, then the bits value of two parent individuals in these positions will be exchanged and form new N_2 individuals.

3) Mutation

In the early evolution, CHC algorithm does not adopt mutation operation, but when the population evolves to a certain generation, such as $T/2$ (here, T represents the maximal evolution generations), it will select N_3 outstanding individuals by decreasing order according to the value of $f(x)/\sum_{i=1}^{N} f(x)$, then choose gene bits and randomly determine their value according to a certain proportion K. Generally speaking, K is equal to 0.35.

Step 5. Control parameters selection. The selection of control parameters is critical, which greatly affects the performance and convergence of genetic algorithm. These parameters include population size, crossover probability P_c, mutation probability P_m, and maximal evolution generations T. Population size can be selected from 10 to 200. The scope of T is generally from 100 to 1000. Generally, a larger P_c is chosen, and its value is good from 0.4 to 0.99, but if it is too high, and it may also lead to premature convergence. By experience, P_m is lesser, which is usually chosen between 0.1 and 0.6.

Step 6. Population update. By selection, crossover, and mutation operations, the resulting total number of new individuals is $(N_1 + N_2 + N_3)$. Then, N_1 outstanding individuals will be selected by decreasing order according to their fitness value and form next-generation population. As a result, the trust paths whose T_{Path} is lesser will be eliminated, while the trust paths whose T_{Path} is larger will be saved. Therefore, the best individual can be found by above steps. Then, one can get corresponding optimal trust path after the best individual is decoded.

Here, specific algorithm is not described.

3.3.2 The reliable trust paths selection and their integrated trust value calculation

Definition 5. Integrated trust value of trust paths. It is defined as the integrated trust value of optimum and reliable trust paths from service requestor to service recommender, denoted by ST_{Path}.

The optimum trust path whose T_{Path} is the highest can be found in directed acyclic graph G by making use of an improved CHC genetic algorithm, denoted by A. Then, remove all the intermediate nodes S_i of A in G and form another directed acyclic graph G', and next, optimum trust path in G' by making use of an improved CHC genetic algorithm can be found, denoted by B. According to this cycle, M number of optimum trust paths can be gained. It is possible that there are joint-cheats in the optimum trust paths. Thus, sort operations will be carried out for these optimum trust paths whose amount is M according to their trust values T_{Path}, and then, threshold value is set to select some reliable and optimum trust paths whose amount is $K(K \leqslant M)$. Finally, T_{Path} of these reliable and optimum trust paths whose amount is K will be integrated to form ST_{Path}, which justifiably will be taken as recommendation weight composing of service recommender. Algorithm realization is described as follows.

Algorithm 1. The calculation of ST_{Path}

Begin;
PathBest[ ]=Φ;
// The initialization of optimum trust paths set //
TrustPathBest[]= Φ;
// The initialization of reliable trust paths set //
For ($i = 1$; $i \leqslant M$; i++)
{ CHC-GA(G) algorithm;
//An improved CHC genetic algorithm for extraction and selection of the optimum trust paths//
Decode the best individual to obtain the optimum trust paths; Path
PathBest[i]=Path;
//Form the optimum trust paths set//
T_{Path}(PathBest[i]) = $\prod_{\substack{D_g, D_k \in \{Req, Rec_j, Med_i\} \\ 1 \leqslant i \leqslant n,\ D_k \in Succ(D_g)}} T_{\text{direct}}(D_g, D_k)$;
//Calculate T_{Path} of the optimum trust paths//
Delete all middle-node S_i of path from G and Form G';
$G = G'$;
}
Value[$1 \cdots M$]=Rank(T_{Path}(PathBest[$1 \cdots M$]);
// Sort by increasing //
$\alpha = [Value[M] - Value[1]]/(M-1)$
//The average interval taken as threshold value //
$P = 0$;
For ($q = M$; $q \geqslant 2$; q− −)
{ If ($Value[q] - Value[q-1] >= \lambda \times \frac{Value[M]-Value[1]}{M-1}$)
//$\lambda \geqslant 2$ represents the constant according to actual situation //
Delete PathBest[q] from PathBest[$1 \cdots M$];
Else
{ P++;
TrustPathBest[P]=Pathbest[q];
//Otherwise, save these paths and take them as the op-

timum and reliable trust paths//
$T_{\text{Path}}(\text{PathBest}[P]) = Value[q]$;
}
}
$ST_{\text{Path}} = 0$;
If ($k \leqslant P$)
{ for ($i = 0$; $i \leqslant k$; i++)
$ST_{\text{Path}} += T_{\text{Path}}(\text{PathBest}[i])$;
$ST_{\text{Path}} = ST_{\text{Path}}/k$;
}
Else
{ for ($i = 0$; $i \leqslant P$; i++)
$ST_{\text{Path}} += T_{\text{Path}}(\text{PathBest}[i])$;
$ST_{\text{Path}} = ST_{\text{Path}}/P$;
}
End.

Thus, ST_{Path} can be calculated by Algorithm 1, which justly will be taken as recommendation weight′s composing of service recommender.

3.3.3 The calculation of recommendatory trust value of service

Definition 5. Recommendatory trust value of service. It shows the formation of trust degree by a third party′s indirect recommendation among service entities, also known as reputation, denoted by $T_{\text{recommend}}(Req, Serv)$. In cloud environment, it indicates the degree of trust of service requestor when service requestor is indirectly recommended by service recommender.

Service recommender set is denoted by Rec_i, $1 \leqslant i \leqslant n$, $T_{\text{direct}}(Rec_i, Serv)$ represents the direct trust value which service recommender Rec_i interacts with specific service $Serv$. P_i', $1 \leqslant i \leqslant n$, represents successful interaction-times that Req interacts with $Serv$ when Req is indirectly recommended by Rec_i. Q_i', $1 \leqslant i \leqslant n$, represents unsuccessful interaction-times that Req interacts with $Serv$ when Req is indirectly recommended by Rec_i. Integrated trust value of trust paths ST_{Path} can be calculated from service requestor to service recommender, which justly can be taken as recommendation weight composing of service recommender. As a result, recommendatory trust value of service can be calculated as follows.

$$T_{\text{recommend}}(Req, Serv) = \begin{cases} \dfrac{\sum_{i=1}^{n} [\frac{P_i'}{P_i'+Q_i'} \times ST_{\text{Path}}(Req \to \cdots \to Rec_i) \times T_{\text{direct}}(Rec_i, Serv)]}{\sum_{i=1}^{n} [\frac{P_i'}{P_i'+Q_i'} \times ST_{\text{Path}}(Req \to \cdots \to Rec_i)]}, & i > 0 \\ 0, & i = 0. \end{cases} \tag{7}$$

4 Service discovery method based on TQoS-WSD

4.1 Service entity data structure description

As mentioned in Section 2, the involved entities of TQoS-WSD include service requestor Req, service provider $Serv$, middle service entity Med and service recommender Rec. Their data structure is described as follows.

1) $Req(ID, ServName, D(d_1, d_2, \cdots, d_n), T_{\text{direct}}(Req, Serv), T_{\text{direct}}(Req, Succ(Req)), SuccName, RecName, T_{\text{direct}}(Rec_i, Serv), ST_{\text{Path}}(Req \to \cdots \to Rec_i))$;

2) $Serv(ID, ServName, BasicAttribute, FunctionAttribute, ContactAttribute, A(a_1, a_2, \cdots, a_n))$;

3) $Med(ID, Succ(Med_i), Pre(Med_i), T_{\text{direct}}(Med_i, Succ(Med_i))$;

4) $Rec(ID, ServName, D(d_1, d_2, \cdots, d_n), T_{\text{direct}}(Rec_i, Serv))$.

In the above, ID represents serial number. $ServName$ points to service name. $BasicAttribute$ represents basic attributes of service. $FunctionAttribute$ points to function attributes of service. $ContactAttribute$ represents contact information of service. $SuccName$ represents subsequent entity name. $RecName$ represents the recommender entity name. $Succ(Req)$ represents subsequent entity of service requestor. $Succ(Med_i)$ represents subsequent entity of middle service entities. $Pre(Med_i)$ points to precursor entity of middle service entities. As mentioned above, the data structure of four service entities has been explained, and their extensible markup language (XML) description has not been described again. According to actual user requirement, data structure of those entities can be expanded.

4.2 Service discovery method based on TQoS-WSD

Traditional trust models for service selection often only consider the trust factor of service and even replace the QoS factor of service with the trust factor of service. However, proposed QoS models for service selection lack the consideration of trust factor of QoS. It can be seen from TQoS-WSD in Section 2 that the initial trust value of service actually is related to the QoS factor of service, and the direct trust value and recommendatory trust value of service are related to the trust factor of service. Thus, total evaluation value of service can be calculated as follows:

$$T(Req, Serv) = W_1' \times T_{\text{initial}}(Req, Serv) + W_2' \times T_{\text{direct}}(Req, Serv) + W_3' \times T_{\text{recommend}}(Req, Serv). \tag{8}$$

In (8), $T(Req, Serv)$ represents total evaluation value of $Serv$. W_1', W_2', and W_3' represent the preferred weights of service requestor Req; here, W_1', W_2', $W_3' \in (0, 1)$, and $W_1'+W_2'+W_3'=1$.

To sum up, the steps of service discovery method based on TQoS-WSD, can be described as follows:

First, when Req submit service request, function matching will be carried out, and some services whose function are the same and QoS are different will be found and form the service set Ω by those services.

Then, total evaluation value of service $T(Req, Serv)$ can be calculated as follows:

1) According to (1), the $T_{\text{initial}}(Req, Serv)$ of certain service $Serv$ in the service set Ω can be calculated. If Req has historical interaction with $Serv$, go to 2); otherwise, go to 3).

2) Taking dynamic nature, successful historical interaction-times and unsuccessful historical interaction-times into account, $T_{\text{direct}}(Req, Serv)$ can be obtained by (4). At the same time, Req sends a recommendation request to Rec; if none of the service recommender

entities, $T(Req, Serv)$ can be obtained by (8) according to the preferred weights of Req that are W_1', W_2', here, $W_1'+W_2'=1, W_3'=0$, then go to 6); otherwise, go to 4).

3) Req sends a recommendation request to Rec; if none of the service recommender entities, then $T(Req, Serv)$ is equal to $T_{\text{initial}}(Req, Serv)$, go to 6); otherwise go to 5).

4) According to the quantity of service recommender entities, $T_{\text{recommend}}(Req, Serv)$ can be acquired by (7); then, $T(Req, Serv)$ can be obtained by (8) on the basis of the preferred weights of Req that are W_1', W_2', and W_3', go to 6).

5) According to the quantity of service recommender entities, $T_{\text{recommend}}(Req, Serv)$ can be acquired by (7); then, $T(Req, Serv)$ can be gained by (8) on the basis of the preferred weights of Req that are W_1', W_3'; here, $W_1'+W_3'=1, W_2'=0$.

6) Total evaluation value of service $T(Req, Serv)$ can be calculated by above 1)–5) steps. Next, return to 1, to calculate total evaluation value of remaining other services of the service set Ω.

Finally, $T(Req, Serv)$ of all services in the service set Ω will be compared to select the best service whose $T(Req, Serv)$ is the highest to work. Meanwhile, if work is finished favorably, rewarded trust will be gained, so direct trust value and recommendatory trust value of service will be increased; otherwise, punished trust will be gained, so direct trust value and recommendatory trust value of service will be decreased.

The realization of specific algorithm is not detailed here.

4.3 Trust update and evaluation of service

The direct trust value and recommendatory trust value of service will be automatically updated to reward or punish according to service work state. If work is finished favorably, set $P=P+1$, and rewarded trust S_x will be gained by (2), so $T_{\text{direct}}(Req, Serv)$ will increase. Correspondingly, set $P_i'=P_i'+1$ and $T_{\text{recommend}}(Req, Serv)$ will increase by (7). On the contrary, set $Q = Q + 1$, and punished trust F_y will be gained, so $T_{\text{direct}}(Req, Serv)$ will decrease. Correspondingly, set $Q_i' = Q_i'+1$, and $T_{\text{recommend}}(Req, Serv)$ will decrease.

After service work finishes, trust update process as mentioned above will happen, and its results can guide service selection process next time. With the increase of service interaction times, the total evaluation value of some high-quality and reliable services will gradually increase, and then those services can be obtained for most Req. However, because of the limitation of individual service capabilities and the increase of Req, some requests will not be responded and make total evaluation value of those services decline. Thus, Req will seek other high quality and reliable services, and their total evaluation value will increase with the increase of service successful interaction-times. At the same time, total evaluation value of those original high quality and reliable services begin to rise after they decline to a certain extent. As a result, the whole trustworthy service environment is adaptive, adjustable, which in practice does not result in overload, and it also will not affect service quality and assures trustworthiness of service interoperation.

5 Experiment for comparison and performance analysis

5.1 Experiment platform and test data set

ISO/IEC 19763-3 international standard (MFI-3: meta-model frame work for interoperability (MFI) for ontology registration) is a meta-model framework of semantic interoperability management and service[24, 31], developed by State Key Laboratory of Software Engineering of Wuhan University, which includes ontology registration, mapping, storage and model selection for services and provide service registration, browse, query, etc. At present, under the guidance of this framework, platform of semantic interoperability for web services based on MFI-3 has been realized, which can support service interoperation between users and services or among services. It can be accessed through universal resource locator (URL): http://61.183.121.132/wsrr and shown as follows in Fig. 3.

Eyhab Al-Masri of Guelph University has researched several years to web services on the public internet and collected web services from universal description, discovery and integration (UDDI), search engines and services portal site by web service crawler engine (WSCE) , and measured more than ten kinds of their QoS attributes and made a final data set quality of web service (QWS)[32−34]. Making use of the platform of semantic interoperability for web services based on MFI-3, seven web services for E-mail validation can be acquired on the public Internet, and then set different QoS values to each service according to QWS and form the original test data set, named as EMAIL-QWS, which is shown in Table 1, where RT denotes response time, TP denotes throughput, AV denotes availability, AC denotes accessibility, IA denotes interoperability analysis (or reliability), and C denotes cost.

Table 1 The original test data set EMAIL-QWS[32]

ServID	Service name	RT (ms)	TP (req/min)	AV (%)	AC (%)	IA (%)	C (cents/invoke)
Serv1	XMLLogicValidateEmail	720	6.00	85	87	80	1.2
Serv2	XWebserviesXwebEmail-Validation	1100	1.74	81	79	100	1
Serv3	StrikeIronEmail Verification	710	12.00	98	96	100	1
Serv4	StrikeIronEmailAddressValidator	912	10.00	96	94	100	7
Serv5	CDYNE Email Verifier	910	11.00	90	91	70	2
Serv6	Webservicex-ValidateEmail	1232	4.00	87	83	90	0
Serv7	ServiceObjectsDOTSEmailValidation	391	9.00	99	99	90	5

Fig. 3 Platform of semantic interoperability for web services based on MFI-3

5.2 Experimental environment set up

After EMAIL-QWS has been gained by making use of the platform of semantic interoperability for web services based on MFI-3, experiment simulation can be done to evaluate the feasibility and effectiveness of TQoS-WSD. The involved entities of the experiment simulation include 10 *Req*, 20 *Rec*, 200 *Med*, and many *Serv*. Among this, the weights among nodes in the trust paths network that is constructed by *Req*, *Med*, and *Rec*, has direct trust value among service entities which can be generated by random function whose value is from 0 to 1, and each node has a direct trust relationship with at least two nodes, and maximum length of trust path $Maxlength$ is set to 6. During the calculation process of recommendation weight of service recommender, population size is set to 50, crossover rate is equal to 0.5, and mutation rate is set to 0.35, and maximal evolution generations is equal to 100. The values of M and K can be adaptively regulated by the experimental situation. W_i $(1 \leqslant i \leqslant 6)$ is equal to 0.1667, and W_i' $(1 \leqslant i \leqslant 3)$ is equal to 0.334.

5.3 Experiment results

According to user′s QoS requirements, QoS-based service discovery method can select the best web service from many web services whose function is the same[32], and it can be named as QoS-WSD method in this paper. This method only considers user′s QoS assurance and is not related to trust problem, which affects trustworthiness of final service discovery and services interoperation. Service response time and service invocation success-rate will be compared and analyzed to prove the feasibility and effectiveness of QoS-WSD and TQoS-WSD.

Experiment 1. The time performance of service discovery.

Based on EMAIL-QWS, a part of data will be incrementally selected from QWS to constitute new six test data set, whose web services amount respectively is 10, 20, 40, 80, 160, and 300. Then, according to 10 random service requests of E-mail validation, service discovery will be done, and their response time will be recorded, and their average values will be taken as experimental results, as shown in Fig. 4.

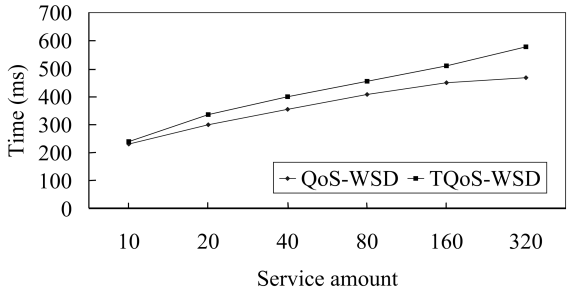


Fig. 4 The time performance of service discovery

Fig. 4 expresses the relationship between the total number of services and their services discovery response time. As seen from it, compared to QoS-WSD, the response time of TQoS-WSD is greater, and its performance is worse because of trust measurement. However, in general, service discovery response time is in milliseconds, this relationship is linear, and it can be accepted by users.

Experiment 2. The validity of service discovery.

Based on EMAIL-QWS, service discovery for E-mail validation can be done, which proves that the expected and trustworthy web services discovery results by TQoS-WSD

are better than QoS-WSD. The number of *Req* is set to 10, and each *Serv* is indirectly recommended by three service recommenders that are not vicious. Recommendation weight of service recommender can be calculated, as given in Section 2. Experiment simulation is repeated three times, and the interaction-times of each experiment simulation are set to 80 when *Req* and *Rec* interact with *Serv*; the true feedback values of QoS will be gained after each interaction time is finished. Representative experiment results are given as follows:

1) Making use of QoS-WSD method for service discovery, Fig. 5 expresses Serv7, whose web service relevancy function (WsRF) value is the maximal, which can be found when users′ preferred weights of QoS attributes W_i $(1 \leqslant i \leqslant 6)$ are the same and equal to 0.1667. Incidentally, WsRF value that is the weighted value of service QoS when users′ preferred weights of QoS attributes W_i $(1 \leqslant i \leqslant 6)$ are the same and equal to 0.1667.

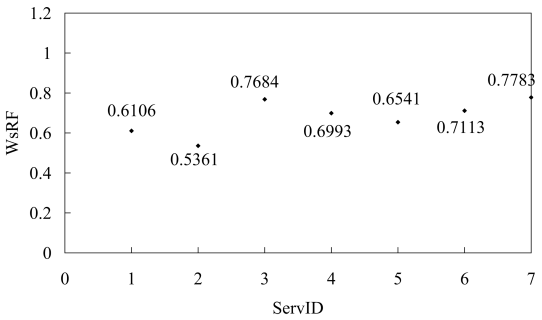


Fig. 5 Service discovery based on QoS-WSD ($W_i = 0.1667$)

2) Utilizing TQoS-WSD method for service discovery, Fig. 6 expresses the change trend of total evaluation value of each service after their multiinteraction-times finish. At first, the total evaluation value of each service is equal to their WsRF value because they have no historical interaction record. The total evaluation value of Serv3 has increased at the beginning and then started to decrease, which is possibly because of the limitations of individual service capabilities and the increase of their interaction-times, and so, some requests will not be responded. Total evaluation values of Serv1, Serv4, Serv5, and Serv7 are basically going to decline, which show their actual service capability is lower than their published service capability. The total evaluation values of Serv2 and Serv6 steadily increase because of their successful multiinteraction-times and good service capabilities. Meanwhile, during the process of interaction, *Req* can in turn find Serv7, Serv3, and Serv6, their total evaluation values are the maximal in a certain interval of interaction-times, which reflects adaptive and dynamic characteristic of trustworthy service environment.

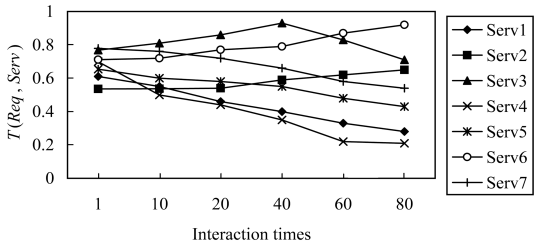


Fig. 6 Total evaluation value of service based on TQoS-WSD

3) As known from 1) and 2), *Req* may select optimum and trustworthy web services. Those web services selected by QoS-WSD and TQoS-WSD method will be implemented for service invocation experiment in order to prove the validity and accuracy of those methods. As seen in Fig. 7, the service invocation success-rate of TQoS-WSD method is higher than those of QoS-WSD, and its curve is in accordance with total evaluation value curve of optimum and trustworthy web services in Fig. 5.

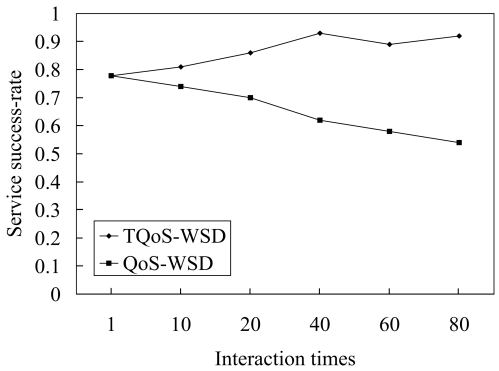


Fig. 7 Service invocation success-rate comparison

Experiment 3. The reliability of service discovery.

As known from Experiment 2, EMAIL-QWS of Table 1 is real-time data at that time, and a part of data is not trustworthy now, that is, there is a part of untrustworthy service provider. Based on the untrustworthy original test dataset EMAIL-QWS in Table 1, the reliability experiment of service discovery for E-mail validation will be implemented when there are vicious service recommenders. The experiment process is given as follows: First, set the number of service recommenders to 20, and the initial percent of vicious service recommenders that deliberately increase or decrease feedback values of QoS to 10%. We increase the percent of those by 10% until it reaches 80% in the experiment. Second, respectively utilizing QoS-WSD and TQoS-WSD, *Req* can interact with service entities and service recommenders for many times, and then, it can select corresponding optimum and trustworthy web services. Finally, service invocation experiment will be done for these optimum and trustworthy services and get service invocation success-rate, as shown in Fig. 8.

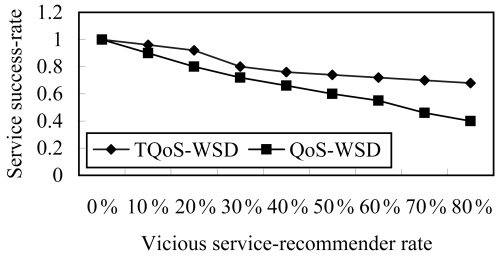


Fig. 8 Service invocation success-rate comparison of different vicious service-recommender rate

As seen from the above graph, the decline intensity of service invocation success-rate of QoS-WSD is larger than those of TQoS-WSD. When vicious service recommender rate reaches 40 %, and from then on until reaches 80 %,

the corresponding service invocation success-rate of TQoS-WSD gradually stabilizes at a certain value. This indicates that TQoS-WSD fully considers service trust, which strongly resists the effect of service discovery by untrustworthy QoS values and improves service invocation success-rate. Thus, the provision and discovery of services become trustworthy, and so, trustworthiness of services interoperation also can be sufficiently assured in cloud environment.

6 Conclusions and future work

To assure trustworthiness of service interoperation in cloud environment is a very important problem. Under the guidance of service interoperation metamodel framework based on RGPS and MFI, combining QoS with trust model, this paper constructs a QoS-aware and quantitative trust-model that consists of initial trust value, direct trust value, and recommendatory trust value of service. First, the initial trust value of service can be calculated by claimed QoS values of not-interactive-record published service. Second, taking the dynamic nature of trust into account, we use attenuation function, which takes interval and interaction-times as parameters, and successful historical interaction-times and unsuccessful historical interaction-times to calculate direct trust value of service. Third, in order to resist the vicious recommendation, an improved CHC genetic algorithm is designed to extract and select optimum trust paths, and their trust values will be calculated and regarded as the recommendation weight′s composing of service recommender, and then, the recommendatory trust value of service will be gained. Finally, according to users′ weight, the total evaluation value of service can be synthesized by initial trust value, direct trust value, and recommendatory trust value of service. At the same time, based on this model, the TQoS-WSD method for service discovery is proposed, which makes a solid trust relationship built among service requestor, service provider, and service recommender, and users can find trustworthy services whose total evaluation is higher. Therefore, the interoperation between users and cloud services or among cloud services becomes trustworthy. Making use of the platform of semantic interoperability for web services based on MFI-3, compared to QoS-WSD method, it is proved by the experiment for TQoS-WSD method that more accurate result of service discovery will be achieved for service requestor, whereas if reasonable time cost is increased, it strongly resists the impact of service discovery by untrustworthy QoS values and improves service invocation success-rate and so assures trustworthiness of service interoperation. In future research work, the emphasis will be to apply the model and method to actual environment.

Acknowledgments

We thank Eyhab Al-Masri and Qusay H. Mahmoud of University of Guelph, Canada for their QWS provision and thank the anonymous referees for their comments and suggestions which improved the quality of the paper.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing, Technical Report, EECS Department, University of California, Berkeley, USA, 2009.

[2] B. Q. Cao, B. Li, Q. M. Xia. A service-oriented QoS-assured and multi-agent cloud computing architecture. In *Proceedings of the 1st International Conference on Cloud Computing*, ACM, Beijing, PRC, pp. 644–649, 2009.

[3] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, F. Galn. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 535–545, 2009.

[4] J. Li, J. Chinnece, M. Woodside, M. Litoiu, G. IsZlai. Performance model driven QoS guarantees and optimization in clouds. In *Proceedings of ICSE Workshop on Software Engineering Challenges of Cloud Computing*, ACM, Vancouver, Canada, pp. 15–22, 2009.

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[6] Z. Zhang, D. D. Zhou, H. J. Yang, S. C. Zhong. A service composition approach based on sequence mining for migrating E-learning legacy system to SOA. *International Journal of Automation and Computing*, vol. 7, no. 4, pp. 584–595, 2010.

[7] T. Yu, Y. Zhang, K. J. Lin. Efficient algorithms for web services selection with end-to-end QoS constraints. *ACM Transactions on the Web*, vol. 1, no. 1, pp. 6–31, 2007.

[8] S. P. Ran. A model for web services discovery with QoS. *ACM SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, 2003.

[9] Y. T. Liu, A. H. Ngu, L. Z. Zeng. QoS computation and policing in dynamic web service selection. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, ACM, New York, USA, pp. 66–73, 2004.

[10] B. A. Soydan, M. P. Singh. A DAML-based repository for QoS-aware semantic web service selection. In *Proceedings of the IEEE International Conference on Web Services*, ACM, California, USA, pp. 368–375, 2004.

[11] H. H. Li, X. Y. Du, X. Tian. A capability enhanced trust evaluation model for web services. *Chinese Journal of Computers*, vol. 31, no. 8, pp. 1471–1477, 2008. (in Chinese)

[12] A. Josang, R. Ismail, C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[13] Y. Wang, J. Vassileva. Toward trust and reputation based web service selection: A survey. *Multi-agent and Grid Systems (MAGS) Journal*, vol. 3 , no. 2 , pp. 118–132, 2007.

[14] W. J. Li, L. D. Ping. Trust model to enhance security and interoperability of cloud environment. In *Proceedings of the 1st International Conference on Cloud Computing*, ACM, Beijing, PRC, pp. 69–79, 2009.

[15] E. M. Maximilien, M. P. Singh. Toward autonomic web services trust and selection. In *Proceedings of the 2nd International Conference on Service Oriented Computing*, ACM, New York, USA, pp. 212–221, 2004.

[16] S. B. Cai, Y. Z. Zou, B. Xie, W. Z. Shao. Mining the web of trust for web services selection. In *Proceedings of the IEEE International Conference on Web Services*, IEEE, Beijing, PRC, pp. 809–810, 2008.

[17] Y. Wang, J. Lv, F. Xu, L. Zhang. An internetware-software-architecture-oriented trust-driven mechanism for selecting services. *Journal of Software*, vol. 19, no. 6, pp. 1350–1362, 2008. (in Chinese)

[18] Z. Q. Xu, P. Martin, W. Powley, F. Zulkernine. Reputation-enhanced QoS-based web services discovery. In *Proceedings of IEEE International Conference on Web Services*, IEEE, Salt Lake City, USA, pp. 249–256, 2007.

[19] L. H. Vu, M. Hauswirth, K. Aberer. QoS-based service selection and ranking with trust and reputation management. In *Proceedings of the Cooperative Information System Conference*, Springerlink, Agia Napa, Cyprus, vol. 3760, pp. 466–483, 2005.

[20] L. Chen, X. Z. Ye, X. Pan, S. Y. Zhang, Y. Zhang, W. Peng. Ontology-based semantic interoperability of product data. *Computer Integrated Manufacturing Systems*, vol. 14, no. 4, pp. 821–828, 2008. (in Chinese)

[21] Y. Lv, Y. H. Ni, X. J. Gu, H. J. Hu, Z. X. Wang. Key technology of ontology-based heterogeneous system interoperation facing to business collaboration. *Journal of Zhejiang University*, vol. 43, no. 8, pp. 1485–1491, 2009. (in Chinese)

[22] X. F. Di, Y. S. Fan. Implementation of enterprises′ interoperation based on ontology. *International Journal of Automation and Computing*, vol. 7, no. 3, pp. 303–309, 2010.

[23] J. Wang, K. Q. He, P. Gong, C. Wang, R. Peng, B. Li. RGPS: A unified requirements meta-modeling frame for networked software. In *Proceedings of the 3rd International Workshop on Applications and Advances of Problem Frames*, ACM, Leipzig, Germany, pp. 29–35, 2008.

[24] Information Technology — Metamodel Framework for Interoperability (MFI), ISO/IEC Standard 19763-3:2007, 2008.

[25] R. F. Zhou, K. Hwang. Power-trust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.

[26] X. Li, L. Liu. Peer-trust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[27] Z. Q. Liang, W. S. Shi. Enforcing cooperative resource sharing in untrusted peer-to-peer environments. *Journal of Mobile Networks and Applications*, vol. 10, no. 6, pp. 771–783, 2005.

[28] S. Kalepu, S. Krishnaswamy, S. W. Loke. Verity: A QoS metric for selecting web services and providers. In *Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops*, IEEE, Roma, Italy, pp. 131–139, 2003.

[29] X. Y. Li, X. L. Gui. Trust quantitative model with multiple decision factors in trusted network. *Chinese Journal of Computers*, vol. 32, no. 3, pp. 405–416, 2009. (in Chinese)

[30] F. M. Liu, W. Y. Zhang, Y. S. Ding, X. Y. Liu, M. C. Zheng, Y. Liu. Dynamic trust: Three-dimensional dynamic computing model of trust in peer-to-peer networks. In *Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, Arnetminer, Shanghai, PRC, pp. 337–344, 2009.

[31] C. Zeng, K. Q. He, Z. T. Yu, C. P. Wan. Towards improving web service registry & repository model through ontology-based semantic interoperability. In *Proceedings of the 7th International Conference on Grid and Cooperative Computing*, ACM, Shenzhen, PRC, pp. 747–752, 2008.

[32] E. Al-Masri, Q. H. Mahmoud. QoS-based discovery and ranking of web services. In *Proceedings of the 16th International Conference on Computer Communications and Networks*, IEEE, Honolulu, USA, pp. 529–534, 2007.

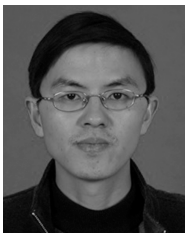
[33] E. Al-Masri, Q. H. Mahmoud. Discovering the best web service. In *Proceedings of the 16th International Conference on World Wide Web*, ACM, Banff, Canada, pp. 1257–1258, 2007.

[34] E. Al-Masri, Q. H. Mahmoud. Investigating web services on the world wide web. In *Proceedings of the 17th International Conference on World Wide Web*, ACM, Beijing, PRC, pp. 795–804, 2008.

Bing Li received the Ph. D. degree from the Computer Science School of Huazhong University of Science and Technology (HUST), PRC in 2003. He worked as a post doctoral researcher in the State Key Laboratory of Software Engineering (SKLSE), Wuhan University, PRC from 2003 to 2005. Now he is a professor of School of Computer and SKLSE, Wuhan University.

His research interests include cloud computing, networked software, service-oriented software engineering, complex system and complex networks, and system integration.

E-mail: bingli@whu.edu.cn (Corresponding author)



Bu-Qing Cao received M. Sc. degree from Central South University, PRC in 2007. He is currently a Ph. D. candidate in the State Key Laboratory of Software Engineering (SKLSE), Wuhan University, PRC. Now he is a lecturer of School of Computer Science and Engineering, Hunan University of Science and technology, PRC.

His research interests include cloud computing, networked software, and service computing.

E-mail: cao6990050@163.com

Kun-Mei Wen received the Ph. D. degree from the School of Computer Science and Technology of Huazhong University of Science and Technology (HUST), PRC in 2007. She is an assistant professor at the School of Computer Science and Technology, Huazhong University of Science and Technology, PRC.

Her research interests include web information search and mining, semantic search, and web information management.

E-mail: kmwen@hust.edu.cn

Rui-Xuan Li received the Ph. D. degree from the School of Computer Science and Technology of Huazhong University of Science and Technology (HUST), PRC in 2004. He is an associate professor of School of Computer Science and Technology, Huazhong University of Science and Technology.

His research interests include cloud computing, social network, semantic web and ontology, and information integration.

E-mail: rxli@hust.edu.cn