



Journal of  
**Software  
Engineering**

ISSN 1819-4311



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)

## A Web Service Personal Discovery Based on QoS

Zhou Yuan and Miao YaoFeng

Xi'an International University Modern Education Technology Center, 710077, Xi'an, China

*Corresponding Author: Miao YaoFeng, Xi'an International University Modern Education Technology Center, 710077, Xi'an, China*

### ABSTRACT

Traditional Web service discoveries mostly proceed from perspective of Web service function attributes. Web service companies provide many similar services that have same functions. Quality of Service (QoS) reflects the ability of a service matching users' needs and it is an important index for evaluating performance of Web service. Based on initial research results, this study proposes a clustering method of users and services according to users' QoS. Users having same preferences are clustered into a user cluster and services with similar QoS information are clustered into a service cluster. According to the direction and weight that the user cluster prefers, this study recommends a best service cluster whose matching value is the biggest to the user cluster by computing the comprehensive QoS value.

**Key words:** QoS, web service, web service cluster, clustering

### INTRODUCTION

Web service has become the focus of the community and it covers many fields, including data integration of Web service, Web service composition, search and discovery of Web service, semantic Web service and Web service security and privacy (Mahinderjit-Singh *et al.*, 2013; Vaughan, 2012). Searching and discovering Web service is one aspect closely associated with people's lives. The community of experts in this area is also exceptionally deep and broad.

Now, Web service discovery methods are mainly restricted to Universal Description Discovery and Integration (UDDI) and Web Service Description Language (WSDL) (Juric *et al.*, 2009; Pastore, 2008). It is mainly to match by describing the basic information of services which lacks the quality of services. Users can search Web service by finding the same functionality but mostly the quality of services is too different. Especially with the maturing of Web service, more and more people begin to focus on this regard, so the number of service providers greatly increases. Online registration rises and the overload phenomenon of traditional e-commerce information begins to appear (Selviaridis and Norrman, 2014; Ye *et al.*, 2013). Web service composition has become indispensable as a single Web service cannot satisfy complex functional requirements. An important component of the service composition is the discovery of relevant services. Web services have become common, if not essential, in the areas of business-to-business integration, distributed computing and enterprise application integration. In SOA environments, as the number of published Web services grows, the provision of a robust, scalable and efficient discovery service is still an interesting issue (Paulraj *et al.*, 2012; Di Modica *et al.*, 2011; Narock *et al.*, 2014).

For Web service in the big data environment, on one hand, the clustering of services is particularly important. The space can be reduced after clustering services which can greatly

improve the efficiency of service discovery; on the other hand, users with various background have a similar tendency to select services which are mostly similar. The clustering of users can improve the efficiency of service discovery to some extent. According to service features, clustering methods have increasingly shown their limitations. Many users have higher requirements on the Quality of Service. Therefore, personalized research of Web service based on QoS has become the majority of many scholars (Tran *et al.*, 2009).

This study aims at realizing users clustering from the perspective of users' preferences and calculating weights of users' preferences. The Web service architecture from the QoS perspective should be designed and the QoS can be used to normalize every representation of Web services. A contrast test is given to prove the method proposed in this study.

## **BASIC CONCEPTS OF WEB SERVICE**

Web service is actually an independent and modular application. It can be invoked through the Internet and be an API exposed to the world. It is also a loosely coupled and reusable software module. From the semantic point, it encapsulates a discrete function. It can be accessed by the standard Internet protocol after releasing on the Internet. Web service is an object deployed on the Web. It is built on an open and XML technology, so it is a rising platform with strong openness. The platform is actually a set of standards and describes how to use multiple languages and achieve interoperability of Web service. There are several features of Web service as follows:

- It is a reusable software module and can get different soft-wares by realizing different combinations. It has good encapsulation and customers can see a list of features it offers
- It is loosely coupled and can be much more easily achieved free allocation among different components
- It uses a standard protocol specification and has high integration capability
- It is published on the Internet so that it can communicate through the firewall
- It can be accessed in the program because it works in the code part. It can exchange and call data with other software

The wide use of Web service technology allows a large number of services, having similar functions appear on the Internet. The information overload often appears when discovering services and service requestors can not get their most needed services by one time. Therefore, the Quality of Service becomes the second largest factor following the service functions. The QoS of Web service has become a hot research field. The QoS mainly describes the information of service quality. How to describe its information is a crowd issue.

Many researchers have studied from different angles of the QoS of Web service and given different definitions. In general, characters of QoS are divided into two categories:

- Positive attributes, such as reputation, availability, the higher, the better
- Negative attributes, such as response time, price, the higher, the worse

Several common QoS properties are defined as follows:

- **Response time:** It refers to a time interval between the service request from a client providing a method and getting the request response. It includes the calculation of time for calling services, middle-ware communication overhead and round-trip time. Its equation is:

$$q_t = T_{com} + T_{mid} + T_{net}$$

where, the communication time can be obtained according to the implementation of the previous services. In expression:

$$T_{net} = \sum_{i=1}^n T_i / n$$

where,  $T_i$  is the time of the communication service and  $n$  is the number of observations

- **Service price:** It refers to the cost of the service requester that is paid for the service call
- **Availability:** It refers to the probability of the running service. Because Web service is pay-per-call which can describe its availability by using the ratio number and total number of successful calls. The  $q_a$  is the probability of available services:

$$q_a = T_a / \theta$$

where,  $T_a$  is the sum of recent  $\theta$  time ( $\theta$  is a constant set by the service manager) and the value of  $\theta$  depends on the particular application

- **Reliability:** It refers to the ability of services performing specific functions in the prescribed conditions in a specific time. Its attributes can be expressed by the MTBF. In equation:

$$q_r = su_t / sum_t$$

where,  $su_t$  means the successful number of services in the execution time  $t$  and  $sum_t$  means the total number of executions in the time  $t$

- **Reputation:** It is the credibility of a service reflecting their level of confidence. The greater value means the higher credibility which depends primarily on objective evaluation of consumers usage. Its equation is:

$$q_{rep} = \sum_{i=1}^n R_i / n$$

where,  $R_i$  represents the user service evaluation and  $n$  is the number of evaluated services

Based on above definitions, the QoS are described as a multi-dimensional vector  $QoS = \{q_t, q_c, q_a, q_r\}$ .

## RESEARCH OF USER PERSONAL PERFENCE MODEL BASED ON QoS

Benefited from the QoS of Web service, this study argues that the concept of QoS can be extended to technology preferences of individual users. Different users have different needs and service providers can be targeted to provide various products and services and also allocate service resources. Many studies have shown that similar user preferences are mostly similar in the choice of service selection.

Users want to have a way which can initiatively recommend services to uses, in this context, the recommender system comes into sight. It can recommend hobbies objects to users which can match their interests and be known as personalized recommendation system. Its essence is actually

Web service based on users' needs which is also an effective way to solve the current problem of information overload. Personalized recommendation technology can consider the basic information of some new users. They do not have historical operating information, so it is not comprehensive to mine Web service. Therefore, this study will introduce personalized technology to recommend mechanisms method.

**Personalized recommendation technology:** Before user clustering, the formal description of the user should be given. Many previous experimental studies indicate that users with some similar characteristics have same service function requests. Therefore, this study defines users' basic information such as age, sex, occupation, region, etc. To facilitate the presentation, formal description is defined as  $QoU = \{\text{age, gender, occupation, region}\}$ , where each value will be bound in several stages. Formal descriptions of uses selecting services will be clustered into a class of similar users, whose similarity can be calculated.

**Method of computing users cluster preference weight:** The BIRCH is a clustering algorithm based on the idea of clustering users' basic information which is that each object will initially be regarded as a separate new subcategory. Repeated in the next iteration, those similarity classes are gathered into a new large cluster until a certain condition is satisfied, where the similarity is the minimum of the two classes user' information similarities. The calculated equation based on the similarity of the user basic information is as follow:

$$\text{Sim User} = \text{Sim}(\text{User 1}, \text{User 2}) \times W \tag{1}$$

Similarity calculated equation is described in detail in this study. The threshold is connected with the body tree.

**Method of users clustering:** This study can calculate service quality weights of users' preferences using analytic hierarchy process and finally add a step by computing cluster weights of all users average. Algorithm 1 gives a detail description.

Algorithm 1: Analytic hierarchy process

---

**Step 1:** Establish a QoS attribute comparison matrix A:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

where,  $a_{ij}$  ( $1 \leq i, j \leq 4$ ) is considered as the importance of properties between the attribute  $i$  and  $j$ ,  $a_{ij}$  is assigned as follows:

$a_{ij} = 1$ , means  $i$  is equal to  $j$

$a_{ij} = 3$ , means  $i$  is obviously important than  $j$

$a_{ij} = 5$ , means  $j$  is obviously important than  $i$

So, if  $a_{ij} = 2m$ ,  $m = 1, 2, 3, 4$ , then the user considers that the properties of  $i$  and  $j$  are between  $a_{ij} = 2m-1$  and  $a_{ij} = 2m+1$

**Step 2:** A matrix consistency test

The consistency index of the matrix A is  $CI = (\lambda - n)/(n-1)$ , where,  $n$  is the dimension of the matrix, i.e., the number of QoS properties;  $\lambda$  is the largest absolute eigenvalue of the matrix A.

---

Algorithm 1: Continue

---

The random consistency index RI of the matrix A is connected with the matrix index. The consistency ratio is defined as  $CR = CI/RI$ , where CI is the consistency index of the matrix A, RI is the random consistency index of the matrix A.

When  $CR < 0.1$ , the relative matrix A has the satisfactory consistency and the consistency test can pass; otherwise, the relative matrix should be adjusted until it is satisfied.

**Step 3:** Get the feature vectors  $\alpha$  corresponding to maximum eigenvalues  $\lambda$  of the Matrix A and normalize  $\beta$ ,  $\beta$  is the weight vector of users' QoS preferences

**Step 4:** Repeat above steps and compare each user in a cluster using AHP to get many values  $\beta$

**Step 5:** Obtain  $\sum_{u=1}^j \beta_u / j$  that is the average of all weights of QoS preferences in the final cluster

Data pretreatments are mainly proposed before giving service discovery. The basic information of users are formally described after clustering and calculates attribute weights of users' preference QoS

---

## WEB SERVICE PERSONALIZED DISCOVERY MODEL BASED ON QoS

Since the quality of service is introduced as parameters of Web service discovery, many experts and scholars have achieved a variety of model architectures and studied them. These architectures are different from various emphases by using their industry experience and these experts have combined different models together to achieve a effective model. Most experts have roughly added a QoS role in the Web service architecture.

**Web service model architecture based on QoS:** Web service cluster is essentially based on a Web service clustering technology. In this stage, service registry center has tens of thousands varieties of services and many services are functionally similar or even identical. There are many bottlenecks in the face of choosing needed services from a vast similar services by service requestors. How to choose a suitable service is a problem in front of everyone.

Clustering refers that a data set is divided into a plurality of groups, so that, the same set has a high similarity and elements in different groups are very different.

Before service discovery, the clustering preprocess of the massive database can make the service match more efficient and reduce the space in the service discovery process. The efficiency of the service discovery can be improved. Figure 1a and b describe clustering based on service function and process by using ontology concept, respectively.

These two ways of clustering Web service can improve the efficiency of service discovery on different levels. The combination of two methods can compress search space to a very small range which is easily efficient for service requestors to select services. At present, service requestors in the selection process are very concerned about the functional requirements of the service to a large extent and they are also very concerned about the QoS of non-functional attributes of services.

Based on above, this study presents a Web service model based on functional requirements by the initial clustering of Web service which can get the parent service clusters. According to QoS attributes of Web service, a subset of service clusters can be obtained by the secondary clustering. Finally, Web service collection according with the functional and non-functional requirements for the service requestor can be obtained. The model is shown in Fig. 2a.

**Formation of Web service cluster:** The QoS of Web service contains many parameters, including price, response time, reputation, performance, availability, reliability and so on. These parameters have different meanings and it is necessary to standardize them. The result set should be collected in the interval (0, 1) and put them into quantifiable factors which can easily calculate

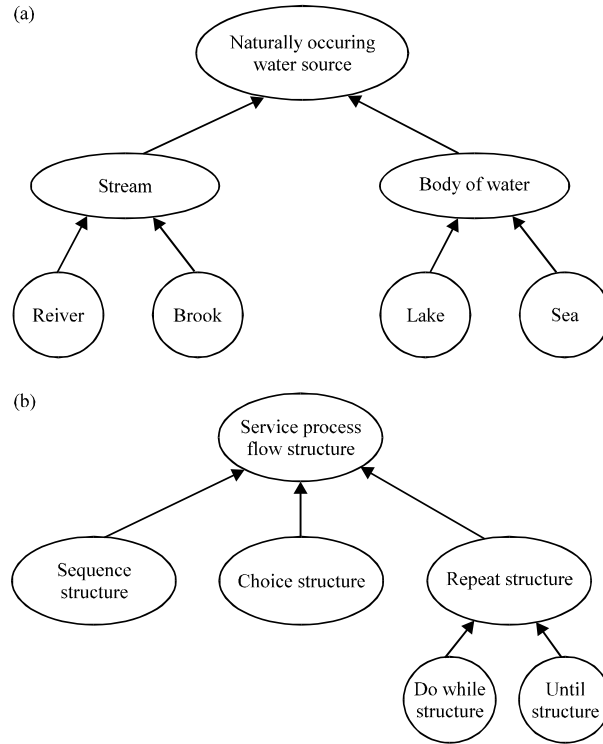


Fig. 1(a-b): Clustering ontology model based on (a) Service function and (b) Service process

the total service quality. Therefore, this study selects a simple data normalization method-Gauss law. This method can effectively avoid the impact of outliers (particularly the small or large value) relative to other normalization and extreme dimensional matrix methods:

$$QoS_{ij}^k = 0.5 + \frac{QoS_{ij}^k - \overline{QoS_i^k}}{2 * 3s_i} \quad (2)$$

where,  $\overline{QoS_i^k}$  represents the average of the k attribute for all the services of QoS in the feedback section by service requestors and  $s_i$  represents the standard deviation of the k data in the QoS attributes.

After the QoS is normalized by Gauss law, according to the Eq. 2, the comprehensive value of QoS can be calculated as follow:

$$Q = \sum_{j=1}^n QoS_{ij}^n \times W_j \quad (3)$$

where,  $W_j$  represents the weight value of each attribute,  $QoS_{ij}^k$  is the j-th attribute value of the service i after it is normalized. If users do not specify a value  $W_j$ , the value can be calculated based on the default weight. The value of  $W_j$  can be given by the method in research of user personal perfence module based on QoS.

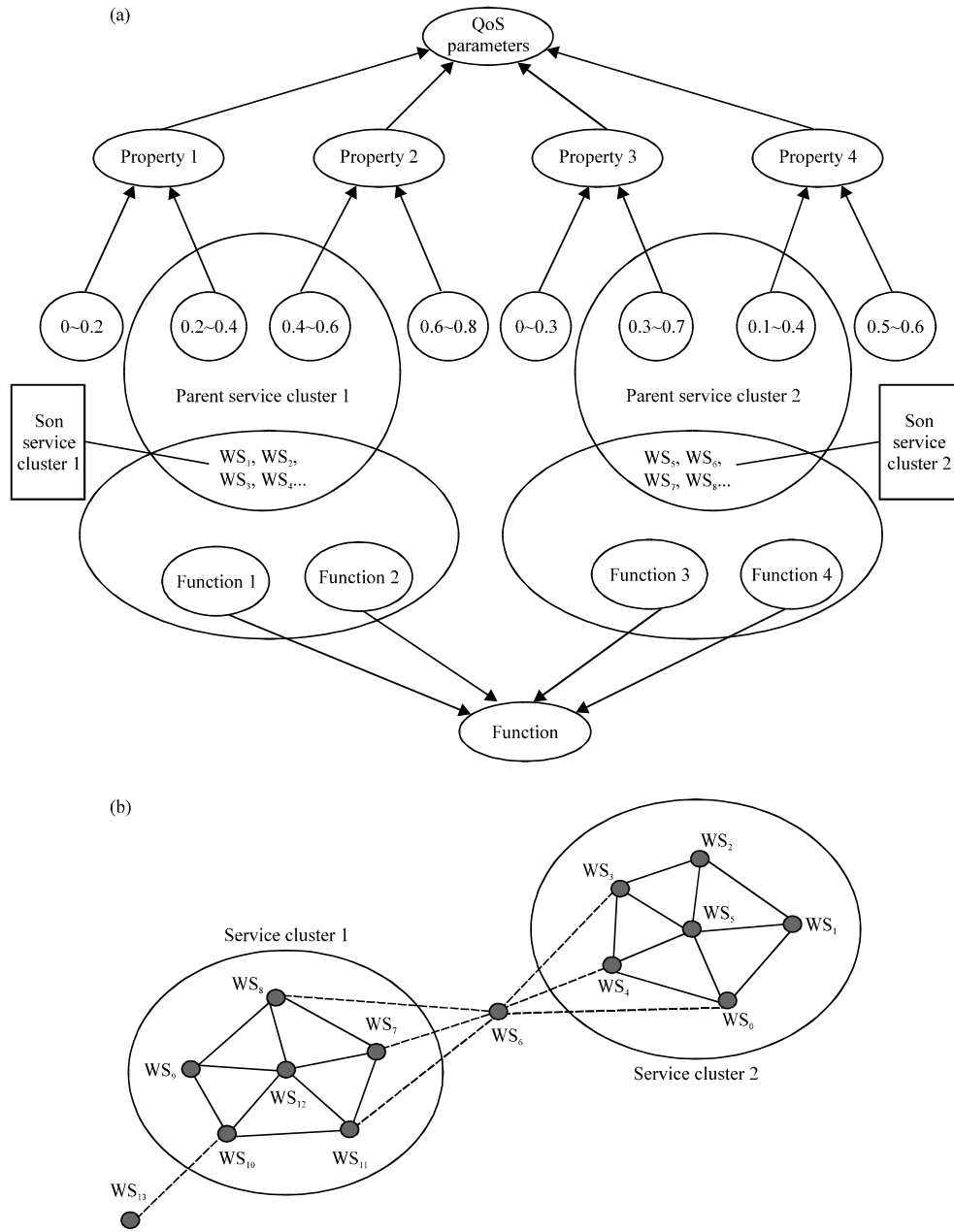


Fig. 2(a-b): Web service cluster formation (a) Model and (b) Process based on QoS

In this study, the QoS is expressed as a multi vector. Let  $QoS_{ws} = (pro_1, pro_2, pro_3, \dots, pro_n)$  and the similarity of the QoS of Web service is expressed as  $sim(QoS) = (QoS_{ws_i}, QoS_{ws_j})$  which can be obtained by calculating the cosine of the similarity of Web service  $i$  and  $j$ . According to the similarity parameter  $\sigma$  of the QoS, similarity graph  $G\sigma$  can be constructed. The parameter  $\sigma$  can be calculated by the similarity model of the QoS.  $G\sigma$  is composed by a number of services and each service is a node in  $G\sigma$ . For any two services  $ws_i, ws_j$ , if  $Sim(ws_i, ws_j) = \sigma$ , there is an edge connecting  $ws_i$  and  $ws_j$  and the specific model is shown in Fig. 2b.



The WS-SCAN algorithm can identify clusters, hubs and outliers in the Fig. 2b. This algorithm can ensure that each of the identified clusters are sufficiently similar and identify each cluster existing in one or more core services. These services can be regarded as the best candidate set for tickling serving requestors.

Each QoS attribute value will be integrated and obtained after the combination of service cluster. The equation is as follows:

$$Q' = \sum_{i=1}^k \sum_{j=1}^n QoS_{ij}^k \times W_j \quad (4)$$

where,  $i$  represents the number of services and  $j$  is the number of representatives of each service attribute. The best service clusters meeting users' demands can be obtained by calculating the largest integrated QoS.

Based on the calculation formula of QoS, the matrix by matching the user cluster and the service cluster can be calculated. Then, the result of service discovery can be obtained.

## **MATERIALS AND METHODS**

The preference of QoS based on users clustering is proposed in this study and the discovery process of personalized Web services can be described as follows.

Firstly, users will be clustered by using the BIRCH clustering algorithm and the basic information of registered users can be stored in the user information database. In the database, users will be divided into their corresponding clustering and a service usage frequency table with an inverted index can be built. Secondly, the parent service cluster can be obtained by using the service function clustering method. Required services can be provided from all provided services and the QoS similarity of required services can be calculated. The similarity of non functional attributes can be used to form the subset of service clusters. Finally, after calculating QoS attribute preference weights, the integrated QoS value can be obtained. By comparison, the optimal service cluster will be recommended to user clusters.

## **RESULTS AND DISCUSSION**

Wang *et al.* (2010), added impact factor to the semantic rules in the service request description to reflect the contribution of this rule to the matchmaking result and to facilitate the matching degree calculation. This traditional discovery algorithm does not consider users' clustering and just considers service functions and discoveries while ignoring the importance of QoS. Based on traditional researches, the QoS module is added in this study and the user clustering is considered. The personalized recommendation module is also added to the service discovery model which can improve the efficiency of discovery.

This study adopts 100 weather forecast services as the experimental data. To begin with, the information of users will be stored in the database. They will be divided into corresponding classes in the user database and the service frequency table can be inverted. Secondly, the service clustering function can find the parent service cluster set and screening out 200 services from provided services. The detailed experiment between the traditional method and the proposed method is shown as follow.

This study compares the personalized preferences of QoS based on our proposed method with the traditional method on the service discovery precision and the result is shown in Fig. 3a.

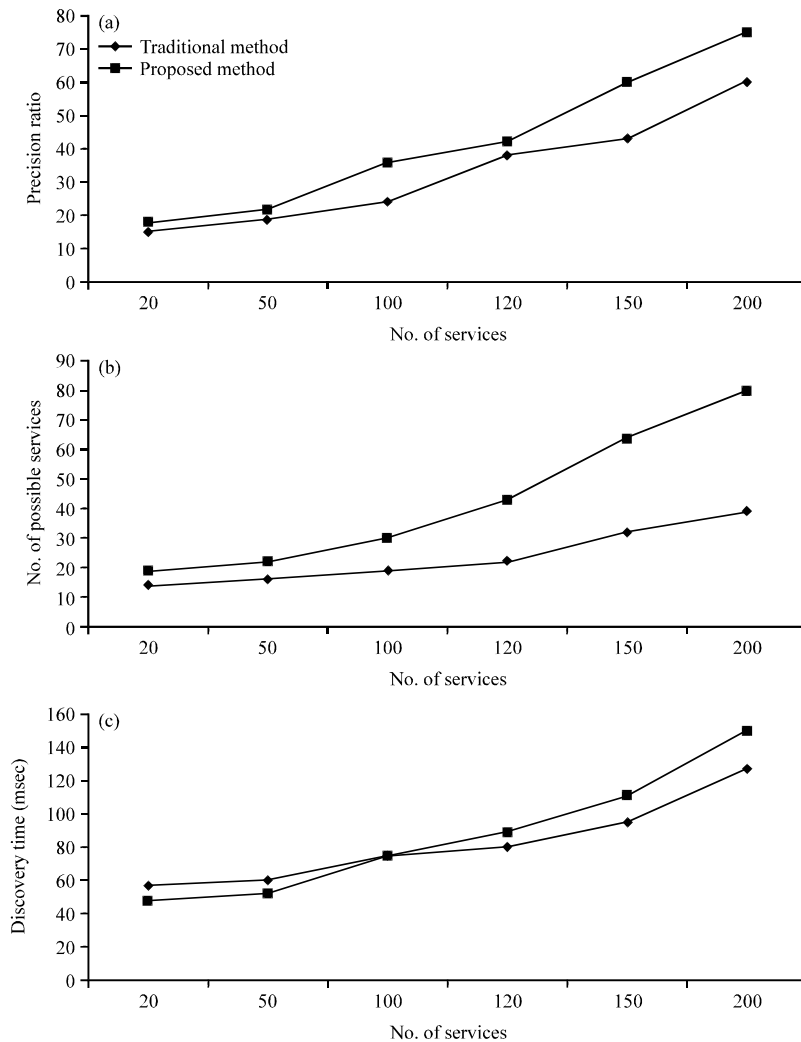


Fig. 3(a-c): Comparison of (a) Precision ratio, (b) No. of possible services and (c) Discovery time with traditional method

Then, the number of possible services and the efficiency of discovery time can be given by comparing the proposed method with the traditional method and results are shown in Fig. 3b and c.

From the three experiments, with the increase of web services, the proposed method by considering the the quality of QoS in this study can improve the precision ratio and reduce the discovery time and search space.

## CONCLUSION

This study presents users and services clustering based on users' preferences and QoS attributes. Attributes of service quality are formalized as a standardized formula language and can calculate the comprehensive service quality property by users' preferences and service quality attributes which can obtain quality attributes scope of certain users. Similar service quality

attributes are integrated into service clusters to facilitate new users to discovery services. Web service in this study is mentioned on the atomic level. In fact, complex services are now mostly used by the composed service.

## **ACKNOWLEDGMENT**

This study is supported by Natural Science Basic Research Program of Shanxi City of China under grant No. 2014JM8356.

## **REFERENCES**

- Di Modica, G., O. Tomarchio and L. Vita, 2011. A P2P based architecture for semantic web service discovery. *Int. J. Software Eng. Knowledge Eng.*, 21: 1013-1035.
- Juric, M.B., A. Sasa, B. Brumen and I. Rozman, 2009. WSDL and UDDI extensions for version support in web services. *J. Syst. Software*, 82: 1326-1343.
- Mahinderjit-Singh, M., X. Li and Z. Li, 2013. Context-aware web services for security control and privacy preservation in an RFID supply chain. *Int. J. Inform. Technol. Manage.*, 12: 39-66.
- Narock, T., V. Yoon and S. March, 2014. A provenance-based approach to semantic web service description and discovery. *Decis. Support Syst.*, 64: 90-99.
- Pastore, S., 2008. The service discovery methods issue: A web services UDDI specification framework integrated in a grid environment. *J. Network Comput. Applic.*, 31: 93-107.
- Paulraj, D., S. Swamynathan and M. Madhaiyan, 2012. Process model-based atomic service discovery and composition of composite semantic web services using web ontology language for services (OWL-S). *Enterprise Inform. Syst.*, 6: 445-471.
- Selviaridis, K. and A. Norrman, 2014. Performance-based contracting in service supply chains: A service provider risk perspective. *Supply Chain Manage.: Int. J.*, 19: 153-172.
- Tran, V.X., H. Tsuji and R. Masuda, 2009. A new QoS ontology and its QoS-based ranking algorithm for web services. *Simul. Model. Pract. Theor.*, 17: 1378-1398.
- Vaughan, J., 2012. Investigations into library web-scale discovery services. *Inform. Technol. Libraries*, 31: 32-82.
- Wang, H., L. Fan and Z.Z. Li, 2010. Method of web service discovery based on semantic rule. *Comput. Eng. Applic.*, 46: 80-84.
- Ye, N., S.S. Yang and B.M. Aranda, 2013. The analysis of service provider-user coordination for resource allocation in cloud computing. *Inform. Knowledge Syst. Manage.*, 12: 1-24.