ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 9, September 2016

Decentralized Auto-adaptive Methodology for Service-Based Applications

Shereena Thampi
Dept. of Computer Science and Engineering
Rajiv Gandhi Institute of Technology
Kottayam, India

Abstract: **The advancement in internet and web has brought about a tremendous change in the attitude of organizations. Based on the emerging trends in service oriented computing and web services most of the organizations have now begun to sell their products as services through the web. But this situation has brought with it so many issues also. As the number of services available is now growing at an enormous pace, it becomes difficult for the users to find the right service of their choice. Hence there arises the need for an efficient system that would help the users to find the service of their choice,. The paper proposes a novel architecture which utilizes the usage history, feedback from the user, location of the user, the QOS factors etc to make an efficient ranking of the available services. Thus it becomes easier for the user to select the most relevant service based on their requirement.**

Keywords—web services; location based; collaborative filtering; QOS utility; user behaviour; user feedback and preferences

I. INTRODUCTION

The web today is a growing universe of interlinked web pages and web apps, teeming with videos, photos, and interactive content. Over time web technologies have evolved to give web developers the ability to create new generations of useful and immersive web experiences [2]. Service-oriented computing is an emerging cross-disciplinary paradigm for distributed computing, which is changing the way software applications are designed, delivered and consumed. At the heart of service-oriented computing are services that provide autonomous, platform-independent, computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols to build networks of collaborating applications distributed within and across organizational boundaries. Web services are the internet enabled applications for performing business needs considered as the platform independent and loosely coupled. The use of the web services technology on the internet has increased widely as it has improved the efficiency and throughput for developers in developing applications [1].

There are a lot of interpretations of web services and we can say that web services are designed to compose various software components and provide machine-machine interactions over a network and the name itself suggests that it is a type of service that is deployed on the internet [1]. It allows two different applications running on different servers to interact with each other over the network and these applications can be implemented in different languages.

The service oriented computing and web services are becoming more and more popular that it enables most of the organizations to use web as a market for selling their services. The number of web services are increasing day by day that it has become difficult to choose the right service they require Thus it becomes necessary for a suitable system to choose the right service of one's choice.

The traditional web service discovery was based on UDDI registries. But this practice is no longer in use as the method got outdated due to the shutdown of the public UDDI registries. Several web services search engines were also in use. But they too are not efficient as they were exploiting the keyword base searching techniques.

A web service consists of both functional and non functional components and the selection of the web services are based on these two aspects. The web services are matched based on the functional requirements and then ranked based on the non functional components. One of the main aspects in non functional component is the QOS

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 9, September 2016

factors. Most of the selections are based on the assumption that the set of available services are functionally equivalent and thus ranking is based only on the non functional components. But this approach is not efficient as there may be keyword mismatch and thus functional components may be misinterpreted. Thus for making a selection both the components are to be considered simultaneously [4]. Most of the web service selection methods are now based on the QOS utility but the results may not be accurate as considering the user history is also important as most of the users resort to seek suggestion from users who are familiar with the requested web service [2].

One of the effective solutions is the use of recommendation systems to effectively find the required solution. Web service recommendation is the process of automatically identifying the usefulness of Web services and proactively discovering and recommending suitable Web services to end users only based on users' usage history. Currently, collaborative filtering (CF) is widely used for Web service recommendation techniques [6]. So far there have been only a limited number of research works using CF for Web service selection. Zhang et al. [7] propose to use CF for Web service ranking based on invocation histories. According to him, for the Web service selection process, the Web service is recommended to the user depending on its matching degree with the QOS requirement as well as its collaborative filtering ranking score calculated on its past invocation history from similar users. Our work is a web service ranking system based on the collaborative score of user history and feedback, QOS utility, functional relevance and location. The main contributions of the work include:

i) Computation of user similarity based on previous history of invocation, feedback from users which constitute users preference.

ii) Three aspects of web service collaborative filtering based score, functional relevance and QOS utility is considered for ranking.

iii) Rank aggregation is done and final ranking is based on the aggregated score and location.

iv) Evaluation of the real dataset to find the performance of the new system over the existing one.

II. RELATED WORK

Most of the researchers use QOS utility for discovering high-quality web services [4], [5], [8]. The main intention is to identify optimal Web services from a set of Web service candidates according to users' requests considering both functional and non-functional requirements. In these studies, a user is specifying his functional interest and QOS requirement, and submits them to the Web service discovery system. Then the service discovery system matches the user's functional and QOS requirements, and returns Web services with the best matching degrees to the user.

Yau et al. [8] propose a QOS-based service ranking approach to help users to select the service that best satisfies or matches users' QOS requirements. Zhang et al. [7] propose to use CF for service ranking based on invocation histories. Considering the above works, we can find that the existing service ranking and selection approaches focus on selecting the service with the best QOS from a set of services having already satisfied users' functional requirement. In fact, their functional relevance

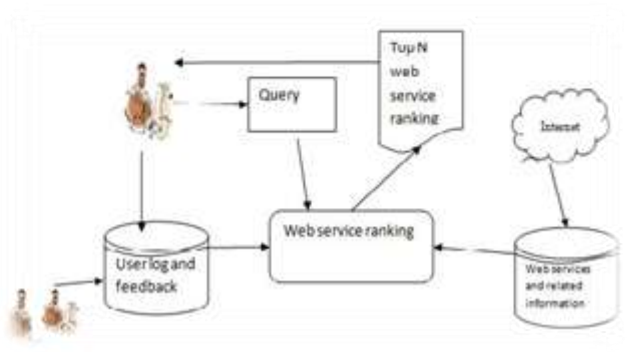
Top N web service ranking
Query
Internet
Web service ranking
User log and feedback
Web services and related information

Fig. 1. Framework of the web Service ranking system

may not be the same. Thus, functional relevance should be considered at the same time, since the functional matching may not necessarily return the accurate results due to the vocabulary mismatch or insufficient description information for function or QOS provided in WSDL files.

Collaborative filtering is a technique used by the recommender systems to make predictions and recommend relevant favorite items to a user by finding similar users to that user; CF is based on user-item matrix. The collaborative filtering approach is based on the assumption that if persons *A* and *B* has the same opinionon an issue, then *A* is more likely to have *B*'s opinion on a different issue *x* than to have the opinion on *x* of a person chosen randomly [3].

CF algorithms can be divided into two categories: memory based and model-based. Memory-based CF includes user-based and item-based approaches. CF based Web service recommendation mainly focus on QOS prediction. Shao et al. [9] propose a user-based CF algorithm using PCC (Pearson Correlation Coefficient) to compute user similarity. The missing QOS values of a Web service can be predicted by considering the corresponding QOS values of Web services used by his similar users. Zheng et al. [10] propose a novel hybrid collaborative filtering algorithm for QOS prediction of Web services by systematically combining both item-based PCC (IPCC) and user-based PCC (UPCC). Jiang et al. [11] presents an improved similarity measurement for users and Web services, which considers the personalized characteristics of users and Web services when calculating similarity using PCC.

G.Kang et al. [14] proposed a web service ranking based on user behavior. The collaborative filtering based score is calculated for the user behavior including the invocation history and the functional relevance. Then the QOS utility and functional relevance score is calculated and the three aspects are aggregated to obtain the final ranking. Thus the ranking will be more accurate than the previously existing methods.

III. FRAMEWORK AND ARCHITECTURE

To overcome the inefficiencies with the existing system we propose a novel architecture to rank the web services by considering the usage history, related users' preferences, the QOS utility, functional relevance and the location of the user requesting the service. The method is an effective one which helps the users to efficiently choose their required service from the large set of available web services.

The framework for the ranking system is shown in fig. 1. The active user makes a request and gives his query for the required web service. The system returns the top n web services to the user after processing. The ranking system takes the query from the user and processes it. The system considers the functional aspect, the QOS utility values, the feedback obtained for the available web services under the functional category and the history from the related users. While ranking the services the location of the requesting user is also considered and only those services which are allowed in the obtained region are included in the ranking list.

The architecture of the web service ranking system is shown in fig. 2. It contains a number of modules to perform the task. The main among are the different filters and their

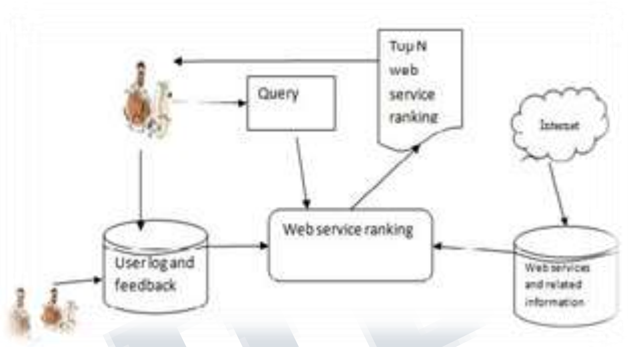
Top N web service ranking
Query
Internet
Web service ranking
User log and feedback
Web services and related information

Fig. 2. Framework of the web Service ranking system

Corresponding filter engines. The QOS and functional relevance filter and its corresponding filter engine obtain the query requested by the user. The filter collects the functional aspect of the query and calculates the similarity and QOS filter calculates the similarity for the request with the available services. The location filter obtains the location of the user and compares it with the location allowed for the available service. The collaborative filter score for the related user is also calculated by considering the history of invocation. The feedback filter collects the feedback from different users who have already used the services. Based on all these scores the web service ranking system find the aggregate score and then top n services are returned back to the user. This web service ranking is of very much use to the users as it reduces their burden of selecting a service from a large available set of services. This system also reduces the inefficiency that may arise due to the keyword mismatch while considering the functional relevance.

IV. DECENTRALIZED AUTO-ADAPTIVE SYSTEM

The decentralized auto adaptive system is a web service ranking system that calculates a ranking of the web service based on collaborative filtering, QOS utility, feedback, location and the related users and their preferences. For ranking let us consider the scenario.

Suppose there are m users and n web services collected from the internet. These users are using these m services. I.e., $u = \{u_1, \cdots, u_m\}$ and $s = \{s_1, \cdots, s_n\}$. In order to acquire the collaborative filtering based score of web services, service similarity is usually required. Service user similarity is considered based on invocation records, functional query, and QOS query, which indicate the historical user behavior together. QOS query contains QOS preference and QOS constraint. QOS preference is usually manifested as a weight vector, $wq^{k}_{ij} = \{w^{k}_{ij1}, \cdots, w^{k}_{ijm}\}$, where $\sigma^{m}_{l=1} w^{k}_{ijl} = 1$, $w^{k}_{ijl} = 1$, and m is number of QOS attributes considered in our web service selection system.

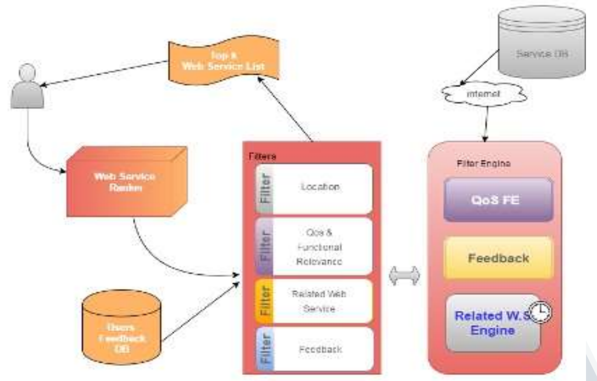

Fig. 3. Architectural Diagram Of Decentralized Auto Adaptive System

A. Functional Relevance

A user query for Web services includes the keywords, input and output. Thus, a vector *(keywords, input, output)* is used to represent the functionality part of a user query as well as the functionality part of Web service operations [13]. The keyword is obtained from the description of the service stored. The input is the query given by the user. The output is the result obtained. Semantic based and keyword based Web service search and discovery are widely studied. Ontology is utilized to compute the semantic similarity for semantic based approaches. Keyword based approaches usually use TF/IDF (Term Frequency/Inverse Document Frequency) or LDA (Latent Dirichlet Allocation) algorithm to compute the functional relevance. Therefore, these approaches can be used directly for computation of functional relevance.

$$SW = \frac{(sum\_wq_i \times wq_i)}{\left(\sqrt{sum\_wq_i^2 \times wq_i^2}\right)} \qquad (1)$$

B. QOS Utility

QOS constraint is expressed as interval data on each QOS attributes, e.g., reliability > 95%, which means 95% < reliability < 1, or time < 5 s, which means 0 < time < 5 s. Therefore, QOS constraint on each QOS attribute can always be expressed as a limited interval. Interval data on each QOS attribute can be viewed as sets, so we can use the jaccard coefficient [12] to compute similarity over two QOS constraints , which is shown as follows:

$$\boldsymbol{SC = \frac{1}{m}\sum_{j=1}^{m}\frac{\left|cq^{k1}_{i1jt} \cap cq^{k2}_{i2jt}\right|}{\left|cq^{k1}_{i1jt} \cup cq^{k2}_{i2jt}\right|}} \qquad \textbf{(2)}$$

Score _ α= SW+SC

C. Feedback

In the Service Web, customers' feedback constitutes a substantial component of Web Service reputation and trustworthiness, which in turn impacts the service uptake by consumers in the future. For assessing a Web Service reputation (FSR), we define reputation key metrics to aggregate the feedback of different aspects of the ratings.

$$FSR = \sum_{k=1}^{n} \binom{n}{k} \mathrm{FR_i U_i} \qquad (3)$$

Score_β= FSR

D. Users preferences

We partition Service Set S based on their topics to constitute Service Category Set, i.e., S_CATEGORY ={C1,C2,…,Ck}.

Then we can use P (U,S) to calculate the value of user preference, i.e. the association between user and service category, which is defined as

$$P(U,SC) = \sum_{i=0}^{n} \mathrm{eS_i} \quad (4)$$

Score_γ= P (U,SC)

E. Rank Aggregation

Finally we calculate the rank of top-n relevant Web services based on functional relevance and QOS utility,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 9, September 2016

User Relationships and Location Preferences and Feedback respectively [15].

$$F\ Score = score\_\alpha + score\_\beta + score\_\gamma \qquad (5)$$

F. Web Service Ranking Algorithm

After having explained the computations the generic algorithm for web service ranking is as follows:

Step 1: Given a user query, calculate the functional relevance based on the keyword and also the QOS score based on the constraints.

Step 2: The location of the user is considered and matched with the location allowed for the user.

Step 3: The score based on feedback is calculated.

Step 4: The score based on the user relationship and preference is calculated.

Step 5: After each calculation the result is combined. And aggregate score is obtained.

Step 6: a rank is obtained for each service and the top n services are listed to the user.

V. PERFORMANCE EVALUATION

In this section we describe the performance of system under various conditions. We consider score based on QOS utility only, QOS and feedback and QOS, feedback and user preference. The TABLE I show the score obtained for services under a particular category. The corresponding graph for a set of service for a particular category is shown in fig. 3.

TABLE 1 SERVICE SCORE

service	**Web service ranking**		
	QOS Score	***QOS +feedback***	***QOS +feedback+userpreference***
Km player	8.32	11.32	12.32
Gom player	6.02	8.02	9.02
Realp	8.32	11.32	12.32
layer			
Vlc palyer	8.32	13.32	14.32
azure	8.32	8.32	9.32

A. Data setup

The evaluation was conducted using real world dataset. The service information including the web service description for each service is also loaded into the database. The user invocation history is also saved. The experiment is conducted considering a period of time T.

B. Experiment setup

The experiment is conducted in a period of time T. The user inputs his query to the system along with constraints. The system on receiving the request checks for the category to which the requested service belongs. Then considers the QOS constraints and calculates the QOS score. Then score considering the feedback is calculated. Finally score considering user preferences and relationship is calculated. The graph is plotted for the score obtained against the set of services. The fig. 3 shows the graph for the category media player and the top 5 services and the score obtained by them.

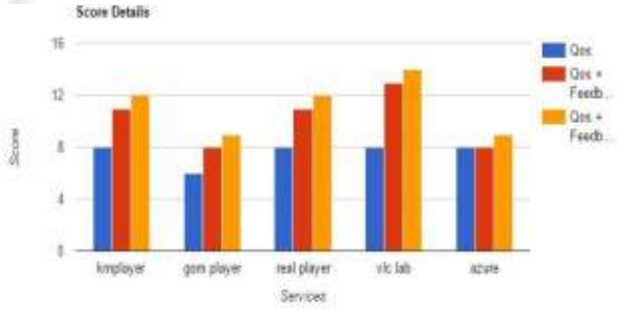

Fig. 4. Performance Graph

VI. CONCLUSION AND FUTURE WORK

The paper explained a hybrid web service ranking approach. This method considers the user query requesting a web service. The web service ranking considers the keyword which describes the category under which the service belongs. It checks for location matching and then calculates the QOS score, feedback score and scoring based on the related users and their preferences. The aggregated score is used for the final ranking and the top N services are listed back to the user. Our approach assumes that the feedbacks obtained are genuine. But there may be situation where the feedback is not genuine. Thus we can incorporate false reputation checking for considering the feedback score. We can add recommendation from the related users.

REFERENCES

[1] Introduction to Web Service: http://javabrains .koushik .org/courses/ javaee_jaxws/lessons/ Introduction-to-Web-Services.

[2] A. Birukou et al., "Improving web service discovery with usage data," IEEE Softw., vol. 24, no. 6, pp. 47–54, Nov./Dec. 2007.

[3] http://en.wikipedia.org/wiki/Collaborative filtering, Collaborative Filtering

[4] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSExpress: A QOS-aware search engine for web services," in Proc. Int. Conf. Web Serv., 2010, pp. 91–98.

[5] G. Kang et al., "Web service selection for resolving conflicting service requests," in Proc. Int. Conf. Web Serv., 2011, pp. 387–394.

[6] L. Yao et al., "Recommending web services via combining collaborative filtering with content-based features," in Proc. IEEE Int. Conf. Web Serv., 2013, pp. 42–49.

[7] Q. Zhang, C. Ding, and C. H. Chi, "Collaborative filtering based service ranking using invocation histories," in Proc. IEEE Int. Conf. Web Serv., 2011, pp. 195–202.

[8] S. S. Yau and Y. Yin, "QOS-based service ranking and selection for service-based systems," in Proc. Int. Conf. Serv. Comput., 2011, pp. 56–63.

[9] L. Shao et al., "Personalized QOS prediction for web services via collaborative filtering," in Proc. IEEE Int. Conf. Web Serv., 2007, pp. 439–446.

[10] Z. Zheng et al., "WSRec: A collaborative filtering based web service recommender system," in Proc. IEEE Int. Conf. Web Serv., 2009, pp. 437–444.

[11] Y. Jiang et al., "An effective Web service recommendation based on personalized collaborative filtering," in Proc. IEEE Int. Conf. Web Serv., 2011, pp. 211–218.

[12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to InformationRetrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008, vol 1..

[13] G. Kang et al., "Diversifying web service recommendation results via exploring service usage history," IEEE Trans. Serv. Comput., vol. 6, no. 1, pp. 35–47, 2015.

[14] Guosheng Kang, Buquing Cao, Jianxun Liu, Mingdong Tang and Yu Xu "An Effective Web Service Ranking Method via Exploring User Behaviour". IEEE Trans.net.serv. Mgmt., vol.112, no.4, Dec 2015.

[15] C. Dwork et al., "Rank aggregation methods for the web," in Proc. 10th Int. Conf. World Wide Web, 2001, pp. 613–622.