International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 1, January-February 2015 **ISSN 2278-6856**

An Enhanced Web Service Recommendation System with Ranking QoS Information

S.Suria[1] and K.Palanivel[2]

[1]Department of Computer Science and Engineering,
Pondicherry University, Puducherry -605014, India

[2]Department of Computer Centre, Pondicherry University,
Puducherry -605014, India

Abstract

Web services are software components designed to support interoperable machine-to-machine interaction over a network. The adoption of web services as a delivery mode in business has fostered a new paradigm shift from the development of monolithic applications to the dynamic setup of business process. Service users are not knowledgement about all the different types of web services. Hence, Web Service Recommender System (WSRS) is needed to provide quality of service to the users. These web service recommendation systems use collaborative filtering algorithm to deliver the most relevant data for the given queries. In this paper, we include investigating the correlation between different QoS properties, and detecting malicious users with inaccurate QoS information. Service-oriented computing (SOC) enables organizations and individual users to discover openly-accessible capabilities realized as services over the Internet. We also introduce a web service recommender system that proactively discovers and manages web services. This paper focuses on the underlying search and ranking algorithms that enable the recommendations. This can be employed effectively by the service users for better web service selection. Our proposed techniques are effective and efficient when compared to the previous approaches through our experimental and simulation analysis.

Keywords: Web Databases, Data Alignment, Data Filtering, Multimode Text, Service recommendation, QoS, collaborative filtering, Ranking efficient

1. INTRODUCTION

A web service [1] is a method of communication between two electronic devices over the World Wide Web. A web service is a software function provided at a network address over the web or the cloud, it is a service that is "always on" as in the concept of utility computing.

"Web service" as software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Web services are distributed application components that are externally available. You can use them to integrate computer applications that are written in different languages and run on different platforms. Web services are language and platform independent because vendors have agreed on common web service standards.

Oracle is developing a java.net project called Metro. Metro is a complete web services stack, covering all of a developer's needs from simple "Hello, World!" demonstrations to reliable, secured, and transacted web services. For more information, see the Metro home page.

Metro includes Web Services Interoperability Technologies (WSIT). WSIT supports enterprise features such as security, reliability, and message optimization. WSIT ensures that Metro services with these features are interoperable with Microsoft .NET services. Within Metro, Project Tango develops and evolves the code base for WSIT. To see how WSIT works, use the Advanced Web Service Interoperability tutorial.

Several programming models are available to web service developers. These models fall into two categories, both supported by the IDE:

REST-based. REpresentational State Transfer is a new way to create and communicate with web services. In REST, resources have URIs and are manipulated through HTTP header operations. For more details, see RESTful Web Services.

SOAP/WSDL-based. In traditional web service models, web service interfaces are exposed through WSDL documents (a type of XML), which have URLs. Subsequent message exchange is in SOAP, another type of XML document. For more details, see SOAP-based Web Services.

The rest of the paper will be organised as follows: In section 2, we see about the web services. In section 3 we discuss about WRS function. In 4, 5 and 6, we see about objective of our system, system design and algorithm respectively. In 7, we see about the disadvantages of the existing system and advantages of our proposed system. The conclusion of our paper is in section 8.

2. WEB SERVICES

Web services are client and server applications that communicate over the World Wide Web's (WWW) Hyper Text Transfer Protocol (HTTP). As described by the World Wide Web Consortium (W3C), web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks [14]. Web services are characterized by

their great interoperability and extensibility, as well as their machine-processble descriptions, thanks to the use of XML. Web services can be combined in a loosely coupled way to achieve complex operations. Programs providing simple services can interact with each other to deliver sophisticated added-value services.

Web Services offer many benefits over other types of distributed computing architectures.

- **Interoperability -** This is the most important benefit of Web Services. Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. Services developed are likely, therefore, to have a longer life-span, offering better return on investment of the developed service. Web Services also let developers use their preferred programming languages. In addition, thanks to the use of standards-based communications methods, Web Services are virtually platform-independent.
- **Usability -** Web Services allow the business logic of many different systems to be exposed over the Web. This gives your applications the freedom to choose the Web Services that they need. Instead of re-inventing the wheel for each client, you need only include additional application-specific business logic on the client-side. This allows you to develop services and/or client-side code using the languages and tools that you want.
- **Reusability -** Web Services provide not a component-based model of application development, but the closest thing possible to zero-coding deployment of such services. This makes it easy to reuse Web Service components as appropriate in other services. It also makes it easy to deploy legacy code as a Web Service.
- **Deploy ability -** Web Services are deployed over standard Internet technologies. This makes it possible to deploy Web Services even over the fire wall to servers running on the Internet on the other side of the globe. Also thanks to the use of proven community standards, underlying security (such as SSL) is already built-in.

3. WRS FUNCTION

Recommendation systems changed the way inanimate websites communicate with their users. Rather than providing a static experience in which users search for and potentially buy products, recommender systems increase interaction to provide a richer experience. Recommender systems identify recommendations autonomously for individual users based on past purchases and searches, and on other users' behavior. This article introduces you to recommender systems and the algorithms that they implement.

A WRS includes the function as follows: a) A web service user provides his requirements using a semantic document. The requirements may vary from description of service and quality of service (QoS) parameters like max execution time, average execution time, max response time, average response time etc. b) The service providers of web services would need to register their services using service descriptions [3]. These service descriptions will contain the semantic service pro- file and QoS parameters like max execution time, average execution time, max response time, average response time etc. The service provider would also be required to specify the location of a WSDL document describing a web service. c) The semantic matcher will match the user request with registered service descriptions and provide a list of available services matching with requirements. This list will be given to the recommendation system. The recommendation system based on its learning through user's feedback orders the list and presents to the user. Each component of the list, finally provided to the user, may be a single service or a composition of registered services. The user can select a service from this list. After the execution is over, the user may provide a rating to this service using given metric. This rating indicates user's satisfaction level. It is stored in a repository and used as an input to the recommendation.

With the widespread proliferation of Web services, quality of service (QoS) will become a significant factor in distinguishing the success of service providers. QoS [7] determines the service usability and utility, both of which influence the popularity of the service. In this article, we look at the various Web service QoS requirements, bottlenecks affecting performance of Web services, approaches of providing service quality, transactional [13] services, and a simple method of measuring response time of your Web services using the service proxy.

Databases are established technologies for managing large amount of data. Web is a good way of presenting information. Alignment and annotation of data increases the efficiency of searching and updating information. Data alignment is the way of arranging data and accessing in computer memory. Data annotation is the methodology for adding information to a document, a word or phrase, paragraph or the entire document. In other words data unit annotation is the process of assigning meaningful labels. Data annotation [4] enables fast retrieval [12] of information in the deep web. A result page retrieved from a web database consists of several search result records (SRRs) and each result records consist of multiple data units. A data unit is defined as the values that represent real world entities. These data units are encoded dynamically into result pages for human browsing and converted into machine process able unit and assigned meaningful labels. The encoding of data units requires lot of human efforts to annotate data units manually. Thus, lack in scalability. To overcome this, automatic assigning of data units within the SRRs is required. Annotated in different aspects and aggregated to predict a final label. Finally, a wrapper is constructed. Wrappers are commonly used as translators which annotate new result pages from the same web database. This automatic annotation approach is highly effective and more scalable.

4. OBJECTIVE OF OUR SYSTEM

The basic idea of the existing model is that to locate users closely with each other who are more likely to have

similar service experience than those who live far away from each other. Inspired by the success of Web 2.0 websites that emphasize information sharing, collaboration, and interaction, we employ the idea of user-collaboration in our web service recommender system. Different from sharing information or knowledge on blogs or wikis, users are encouraged to share their observed web service QoS performance with others in this recommender system. The more QoS information the user contributes, the more accurate service recommendations [9] the user can obtain, since more user characteristics can be analyzed from the user contributed information. In web service recommender system, users usually provide QoS values on a small number of web services. Traditional memory-based CF algorithms suffer from the sparse user contributed data set, since it's hard to find similar users without enough knowledge of their service experience. The correlation between users' physical locations and QoS [15] properties is needed to solve this problem. In this it focus on the QoS properties that are prone to change and can be easily obtained and objectively measured by individual users, such as response time and availability. The Region as a group of users who are closely located with each other is likely to have similar QoS profiles. Each user is a member of exactly one region. Regions need to be internally coherent, but clearly different from each other. The region creation phase is designed as a three-step process. In the first step, users with similar IP addresses are categorized into a small region and extract region features. In the second step, we calculate the similarity between different regions. In the last step, we aggregate highly correlated regions to form a certain number of large regions. After the phase of region aggregation, queries are clustered [3] into a certain number of regions based on their physical locations and historical QoS similarities. The service experience of users in a region is represented by the region center. With the compressed QoS data, searching neighbours [16] and making predictions for an active user can be computed quickly. Traditionally, the QoS prediction methods need to search the entire data set, which is rather inefficient. In this approach, similarity between the active user and users of a region is computed by the similarity between the active user and the region center. Moreover, it is more reasonable to predict the QoS value for active users based on their regions, for users in the same region are more likely to have similar QoS experience on the same web service, especially on those region-sensitive ones. In order to overcome, the obstacles in the previous approach we implementing new concepts in this paper. A novel collaborative filtering [2] algorithm designed for large-scale web service recommendation. This collaborative filtering algorithm uses a rule-based mechanism to determine behavior consistent information based control strategies for route guidance in a dynamic [18] vehicular traffic system. These web service recommendation systems use collaborative filtering algorithm to deliver the most relevant data for the given queries. In an existing system, some disadvantages is there in order to overcome this, we proposed effective technique as CF algorithm to obtain the QoS value in Web service by Recommendation. We investigated the correlation between the different QoS properties in the webservices. We also propose detecting of malicious user in the service recommender system. The predicted QoS values can be employed for the Web service recommendation and selection by the following ways:

For functionally equivalent Web services, the one with best predicted QoS performance can be recommended to the active user. Service Recommender System can recommend the top best performing Web services, which may not have equivalent functionality, to the service users to help them discover the potential Web services. Service Recommender System can also recommend the top active service users, who have good predicted QoS values on a certain Web service, to the service provider to help the provider find its potential customers. Different from all other existing prediction methods, SRS not only provides the predicted QoS values for the active users, but also provides the confidences of the predicted values, which can be employed by the service users for better Web service selection [5]. Our proposed techniques are effective and efficient when compared to the previous approaches through our experimental and simulation analysis.

5. SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

This paper is intended to provide an insight of the annotation techniques and application of few techniques to provide the required results with the above stated advantages. A novel collaborative filtering algorithm [8] designed for large-scale web service recommendation. This collaborative filtering algorithm uses a rule-based mechanism to determine behaviour consistent information based control strategies for route guidance in a dynamic vehicular traffic system. This approach employs the characteristic of QoS and achieves considerable improvement on the recommendation accuracy.

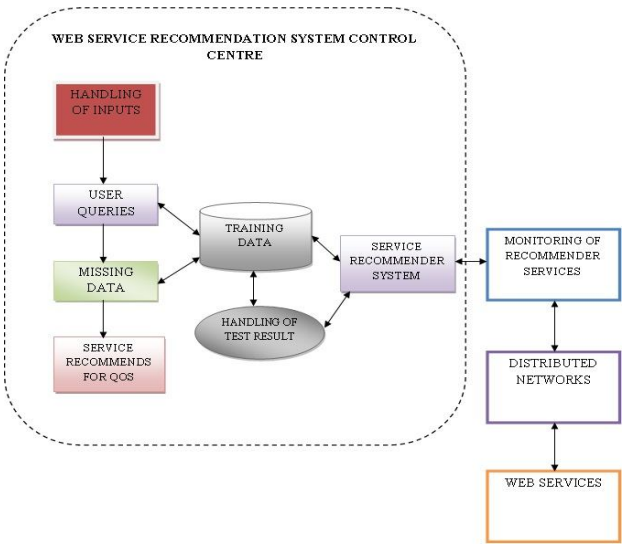

Figure 1 An Enhanced Web-service Recommendation System Architecture

In the presence of multiple Web services with identical or similar functionalities, Quality of Service (QoS) provides non-functional Web service characteristics for the optimal Web service selection. Since the service providers may not deliver the QoS it declared, and some QoS properties (e.g., network latency, invocation failure-rate, etc.) are highly related to the locations and network conditions of the service users, Web service evaluation by the service users can obtain more accurate results on whether the demanded. The service recommender system component is answering the global user query. The latter has to be split local queries (i.e., sub-queries) and has to determine which peer is able to solve a local query. Each sub-query is expressed in SQL. SRS handles a Local Query Processing Engine component. Then, it carries out all the interactions between the composed services and generates a set of composition plans to provide the requested data.

The basic function of this architecture is to obtain sufficient Web service QoS information from different service users crucial for making accurate Web service recommendations. The idea is that by contributing the individually observe Web service QoS information to Web Service Recommender System, the service users can obtain accurate Web service recommendation service. Apart from the user contribution mechanism, Web Service Recommender System also controls a number of distributed computers for monitoring the publicly available Web services. The system architecture of Web Service Recommender System, which includes the following procedures: An active service user provides the individually obtained Web service QoS information to the Web Service Recommender System, The Input Handler in the Web Service Recommender System processes the input data, The Find Similar Users finds similar users from the training data of Web Service Recommender System, The Predict Missing Data predicts [17] the missing QoS values for the active user using collaborative filtering algorithm [10] and saves the predicted values and The Recommender employs the predicted QoS values to recommend optimal Web services to the active user.

5.2 MODULES EXPLANATION

Region Creation

In web service recommender system, users usually provide QoS values on a small number of web services. Traditional memory-based CF algorithms suffer from the sparse user-contributed data set, since it's hard to find similar users without enough knowledge of their service experience. Different from existing methods, we employ the correlation between users' physical locations and QoS properties to solve this problem. In this paper, we focus on the QoS properties that are prone to change and can be easily obtained and objectively measured by individual users, such as response time and availability. To simplify the description of our approach, we use response time (also called round-trip time (RTT)) to describe our approach.

QoS Value Prediction

After the phase of region aggregation, thousands of users are clustered [11] into a certain number of regions based on their physical locations and historical QoS similarities. The service experience of users in a region is represented by the region center. With the compressed QoS data, searching neighbors and making predictions for an active user can be computed quickly. Traditionally, the QoS prediction meth-ods need to search the entire data set, which is rather inefficient. In our approach, similarity between the active user and users of a region is computed by the similarity between the active user and the region center. Moreover, it is more reasonable to predict the QoS value for active users based on their regions, for users in the same region are more likely to have similar QoS experience on the same web service, especially on those region-sensitive ones.

User-collaboration Idea

The basic idea of our approach is that users closely located with each other are more likely to have similar service experience than those who live far away from each other. Inspired by the success of Web 2.0 websites that emphasize information sharing, collaboration, and interaction, we employ the idea of user-collaboration in our web service recommender system. Different from sharing in-formation or knowledge on blogs or wikis, users are encouraged to share their observed web service QoS performance with others in our recommender system. The more QoS information the user contributes, the more accurate service recommendations the user can obtain, since more user characteristics can be analyzed from the user contributed information

Time Complexity Analysis

The time complexity of calculating the median and MAD of each service. Form services, the time complexity. With MAD and median, we identify the region-sensitive services from the service perspective. Since there are atmost n records for each service, the time complexity of each service. Therefore, total time complexity of region-sensitive service identification.

6. ALGORITHM

Pseudocode is a short hand way of describing a computer program. Rather than use the specific syntax of a computer language, more general wording is used. Using pseudocode, it is easier for a non-programmer to understand the general workings of the program.

This is usually made before a programmer starts making the actual program with Syntax, So as just to have the idea of the program. This should be written to the extend than every person, even without knowing any programming should still understand because it should be like reading some bullets points in simple and understandable language (like English)

Program Design Language for Web service using Pseudo code:

1) Consists of natural language-like statements that precisely describe the steps of an algorithm or program
2) Statements describe actions
3) Focuses on the logic of the algorithm or program
4) Avoids language-specific elements

5) Written at a level so that the desired programming code can be generated almost automatically from each statement
6) Steps are numbered. Subordinate numbers and/or indentation are used for dependent statements in selection and repetition structures.

7. DISADVANTAGES AND ADVANTAGES

7.1 DISADVANTAGES

However, three unsolved problems of the previous work affect the performance of current service recommender systems:

1. The first problem is that the existing approaches fail to recognize the QoS variation with users' physical locations.
2. The second problem is the online time complexity of memory-based CF recommender systems. The increasing number of web services and users will pose a great challenge to current systems.
3. The last problem is that current web service recommender systems are all black boxes, providing a list of ranked web services with no transparency into the reasoning behind the recommendation results .It is less likely for users to trust a recommendation when they have no knowledge of the underlying rationale.

7.2 ADVANTAGES

The main advantages of our proposed system:

First, we combine the model-based and memory based CF algorithms for web service recommendation, which significantly improves the recommendation accuracy and time complexity compared with previous service recommendation algorithms.

Second, we design a visually rich interface to browse the recommended web services, which enables a better understanding of the service performance.

8. CONCLUSION

An inventive draw near for web service commendation is designed, which considerably advances the calculation exactness to present Quality-Aware Service in Web. Web service recommender systems work by collecting user observed QoS records of different web services and matching together users this approach employs the characteristic of QoS and achieves considerable improvement on the recommendation accuracy. Our proposed work helps the user by updating the database and guiding the user to select the Quality-Aware services when difficulties arise. It also reduces the time complexity by providing an optimal value which ensures QoS-Aware Services. Our experimental result showed that our proposed novel technique works efficiently when compared to previous methods.

References

[1] M.B. Blake and M.F. Nowlan, "A Web Service Recommender System Using Enhanced Syntactical Matching," Proc. Int'l Conf. Web Services, pp. 575-582, 2007.

[2] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI '98), pp. 43-52, 1998.

[3] J. Himberg, "A SOM Based Cluster Visualization and Its Application for False Coloring," Proc. IEEE-INNS-ENNS Int'l Joint Conf. Neural Networks, pp. 587-592, 2000, vol. 3, doi:10.1109/IJCNN.2000.861379.

[4] Y. Pauline Jeba, "A survey on annotating search results from web databases," International journal of research in Computer Applications and Robotics.

[5] S. Wang,Q.Sun,andF. Yang,''Towards Web Service Selection Based on QoS Estimation,''Int'l J. Web Grid Serv.,vol.6,no.4, pp. 424-443, Nov. 2012.

[6] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS Prediction for Web Services via Collaborative Filtering," Proc. Int'l Conf. Web Services, pp. 439-446, 2007.

[7] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS Prediction for Web Services via Collaborative Filtering," Proc. Int'l Conf. Web Services, pp. 439-446, 2007.

[8] M.R. McLaughlin and J.L. Herlocker, ''A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience,'' inProc. 27th Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 2004, pp. 329-336.

[9] B. Mehta, C. Niederee, A. Stewart, C. Muscogiuri, and E.J. Neuhold, "An Architecture for Recommendation Based Service Mediation," Semantics of a Networked World, vol. 3226, pp. 250-262, 2004.

[10] M.R. McLaughlin and J.L. Herlocker, "A Collaborative Filtering Algorithm and Evaluation Metric That Accurately Model the User Experience," Proc. Ann. Int'l ACM SIGIR Conf., pp. 329-336, 2004.

[11] J. Zhu, Y. Kang, Z. Zheng, and M.R. Lyu, ''A Clustering-Based QoS Prediction Approach for Web Service Recommendation,'' in Proc. 15th IEEE Int'l Symp. Obj./Compon./Serv.-Oriented Real-Time Distrib. Comput. Workshops, Apr. 2012, pp. 93-98

[12] C.D. Mining, P. Raghavan, and H. Schu¨ tze, An Introduction to Information Retrieval. Cambridge Univ., 2009.

[13] J.E. Haddad, M. Manouvrier, and M. Rukoz, ''TQoS: Transac-tional and QoS-Aware Selection Algorithm for Automatic Web Service Composition, ''IEEE Trans. Serv. Comput, vol.3, no.1, pp. 73-85, Jan/Mar. 2010.

[14] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, ''An Algorithmic Framework for Performing Collaborative Filtering,'' in Proc. 22nd Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 1999, pp. 230-237.

[15] R.M. Sreenath and M.P. Singh, ''Agent-Based Service Selection,'' J. Web Semantics, vol. 1, no. 3,

pp. 261-279, Apr. 2003. Fig. 6. Impact of. chen et al.: web service recommendation via exploiting location and qos information 1923.

[16] Z.Zheng,H. Ma,M.R.Lyu, andI.King,''Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization,''IEEE Trans. Serv. Comput., vol. 6, no. 3, pp. 289-299, July/Sept. 2013.

[17] H.Ma, I.King, and M.R.Lyu, ''Effective Missing Data Prediction for Collaborative Filtering,'' in Proc. 30th Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 2007, pp. 39-46.

[18] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, and C.-H. Chen, ''Dynamic Web Service Selection for Reliable Web Service Composition,'' IEEE Trans. Serv. Comput., vol. 1, no. 2, pp. 104-116, Apr./June 2008