"Ideal Web Service Selection in terms of Response Time and QoS Parameters"

Mahesh S. Jagtap, Prof. P. S. Patil

ME-II Student, Department of Computer Engineering,JSPM's Rajarshi Shahu School of Engineering &Research,Narhe,Pune, Maharashtra, India.

Assistant Professor, Department of Computer Engineering,JSPM's Rajarshi Shahu School of Engineering &Research,Narhe,Pune, Maharashtra, India.

---***---

Abstract - *Basically web service is used for the communication between the two different platform or communication between the same platform i.e java can communicate with php or php can communicate with perl. Due to the increasing popularity of web services there are lots of programmer motivates and they are build their own web services. There are lots of web services in UDDI which having similar functionality. Due to this, end user having difficulty in selecting the best one among them. Existing system uses the reviews and comments of previous user to select the optimal web service. But in this system user will not get accurate results because the reviews or comments of previous users may be fake. So to tackle with this problem we propose a system that selects the best optimal web service by considering the response time and other QoS parameters like Price, Reputation, Latency, Accuracy and Accessibility. Here in our proposed system we are analyzing the hidden states of web services with the help of Hidden Markov Model. The results of our proposed system are more accurate than the existing system.*

Keywords— Latency, Hidden Markov Model, QoS, SOAP, WSDL, UDDI,

1. INTRODUCTION

Web services are used for program to program interactions. Web services deploy the solution faster and open the new opportunities. Web services are expand on existing guidelines, for example, HTTP, Extensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI).

A web service is software or piece of software that is available over internet and uses XML for messaging system. XML is used to encode all communications to a web service. Consider the example that client invokes a web service by sending an XML message. After that he waits for corresponding XML response. As all communication is in XML, web services can interact with any programming language like java can talk with PHP, Unix applications can talk with Windows applications, etc. Web services are built on TCP/IP, HTTP, HTML, Java and XML.

1.1 Architecture of Web Service

Web service architecture is based on three roles: service provider, service registry and service requestor. The architecture involves three operations: publish, find and bind operation. Service provider provides the web services to the requested user. Service requestor is the person that requests for the web service to service provider. Service registry is the repository where all web service registration is there. Service registry contains the service description of all web services.

Service provider defines a service description for the Web service and publishes it into the service registry. The service requestor uses a find operation to retrieve the service description locally or from the service registry and uses the service description to bind with the service provider and interact with the Web service implementation. Service provider and service requestor roles are logical constructs and a service can exhibit characteristics of both. Figure 1 illustrates the architecture of web service.

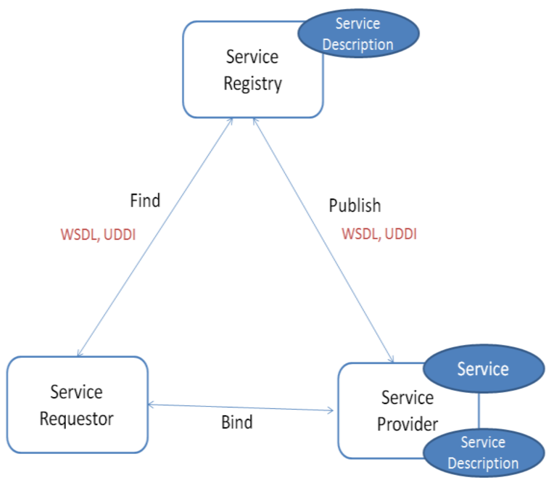

Fig. Architecture of Web service

On the users side, due to the increased number of services it is difficult choose the best web service among them. Currently user selects the best web service depending upon the previous user's reviews or comments. But this method will not give the accurate best web service. While selecting the best web services the hidden states must be consider. Unfortunately there is no previous method that considers the hidden states while selecting the best web service.

So here in this paper author proposes a method that selects the best web services by considering the hidden states of web services.

The remainder of the paper is organized as follows: Section II introduces related work, Section III describes problem statement, Section IV describes the Existing System, Section V describes Software Specification and requirements, Section VI describes the Proposed System, Section VII describes the Mathematical Model, Section VIII describes the Experimental results of proposed system, Section IX describes the Progress report of project and finally section X concludes the paper.

2. RELATED WORK

V. Grassi [2], analyzed the web service in both ways for predicting the reliability i.e. by considering service offered by software component and service offered by hardware devices.

For remote web services, he assumed that the vendor will provide details about the flow of executing user requests. To predict reliability author suggested the modification in WSDL. The limitation of this method is he only consider the reliability factor. We can also predict the performance of any web service.

Z. Zibin and R. Michael [3] have employed past failure data of real web services to find the reliability of current web services. They propose a collaborative reliability prediction approach for service oriented system. The main idea is to find out the past failure data of similar users when collecting the failure probabilities of service components, author assume service component failures are independent. In most cases this assumption is correct as web services are hosted on different servers of different organizations. But there might be case that the two web services are from the same servers. That is, here failure isolation occurs. Author of this paper will not consider this scenario in this paper[3].

Z. Yilei, Z. Zibin, and M.R. Lyu [4], Based on the intuition that a user's Web service QoS usage experience can be predicted by using the past usage experience from different users, author of this paper[4] propose a novel model based approach, called WSPred, for time-aware personalized QoS value prediction for Web services. By employing a collaborative framework, WSPred performs feature modeling on user, Web service and time based on the QoS usage experience collected from both local and global users. Requiring no additional invocation of Web services, WSPred makes the QoS prediction by evaluating how the user specific, service-specific and time-specific latent features apply to each other.

WSPred predicts missing QoS values based on the past QoS experience and the available QoS information in the current time interval. If no QoS information is available in the current time interval, WSPred purely depends on the past experience.

This is the drawback of WSPred method.

F. Salfner [5], this paper shows how hidden Markov Models (HMMs) could be used to predict failures of computer systems. The idea is to identify suspicious patterns of error events. These patterns indicate an upcoming failure. HMMs are used for pattern recognition tool. A machine learning approach has been proposed. In this case HMM is first trained offline using previously recorded log file data. After that it is used to predict failures online – while the system is running. The prediction is divided into two steps: First the current system state is find out from previous error events by applying Viterbi's algorithm. Based on the first state, the risk of failure is assessed by computing the first passage time distribution into a failure state. The critical part of the approach is handling of time. The solution presented in this paper [5] is : Time is split into equidistant slots and every slot without error or failure is filled with a symbol "silence". Other extensions base HMMs on continuous-time Markov chains. In this method, inter-event durations could be incorporated directly into transition timing.

Duhang Zhong, Zhichang Qi and Xishan Xu [13] "Reliability Prediction and Sensitivity Analysis of Web Services Composition". An important issue for business process built is how to assess the degree of trustworthiness, especially their performance and dependability characteristics. In this paper author focus on reliability aspects, and propose an approach to predict the reliability of web services composition. Stochastic Petri Nets (SPNs) can be used to specify the problem in a concise fashion and the underlying Markov chain can then be generated automatically. Author propose the usage of CSPN model, an extension of stochastic Petri nets as a solution to the problems of predicting the reliability of web service composition. The choice of Petri nets was motivated by the following reasons: (a) Petri nets are a graphic notation with formal semantics, (b) the state of a Petri net can be modeled explicitly, (c) the availability of many analysis techniques for Petri nets.

Sami B, Claude G, Olivier Perrin "Transactional Patterns for Reliable Web Services Compositions"

Reliability is one of the main challenge that encounter Web services compositions. Due to the inherent autonomy and heterogeneity of Web services it is difficult to predict the behavior of the overall composite service.

In this paper author propose a new solution that combines the business process adequacy of workflow systems and the reliability of transactional processing. Author introduces the concept of transactional patterns to ensure reliable composite services. A transactional pattern can be seen as a convergence concept between workflow patterns and advanced transactional models.

3. PROBLEM STATEMENT

In this Internet world web service is getting more and more popular technology today. Due to this popularity many web services are developed with similar functionality. When user searches service into UDDI directory, directory retrieve several web services with similar functionality. The problem become more complicated when discovery process return several web service with similar functionality. User has no way to select suitable web service. In this situation QoS parameters like Price, Reputation, Accuracy, Accessibility, Latency and also the hidden states of web service is consider as important approach for service selection. By considering these parameters we can find out the optimal web service from the list of functionally equivalent web services.

4. SOFTWARE REQUIREMENT AND SPECIFICATION

This Software Requirements Specification provides a complete description of all the functions and constraints of the Improved QoS based method for Optimal Path of user request execution .The document describes the issues related to the system and what actions are to be performed by the development team in order to come up with a better solution.

User Environment:

1. The application will be used on computer based on Windows Operating System.
2. Platform used will be Java, Where we used JDK1.6 as developing Environment.
3. Front end Java Swings or PHP and HTML and backend is My SQL Data Base.

Operating Environment:

1. Os: Windows Xp, Windows 7
2. Database: MYSQL

Hardware Requirement:

1. Processor: Dual Core of 2.2 GHZ
2. Hard Disk: 40 GB.
3. Monitor: 14 Color Monitor.
4. RAM: 512 Mb.

Software Requirement:

1. Operating system: Windows 7 Ultimate
2. Coding Language: Java
3. Front-End: Java /PHP and HTML
4. Data Base: MySQL

4. PRPOSED WORK

Below figure shows Proposed System Architecture. In the proposed system User first request for web service then Evaluating Hidden states of requested WS by using HMM. HMM is a tool that is used to find out the hidden states of web services. Hidden states mean that internal structure of the underlying system is hidden from the observer.

After the evaluation of hidden states the observation patterns are recorded. Based on that observation patterns the directed graph between the hidden states are drawn. After that hidden states are analysed. In this step the current state is analysed. With the help of current state, future behaviour of hidden state is predicted. For that we use the VITERBI Algorithm.

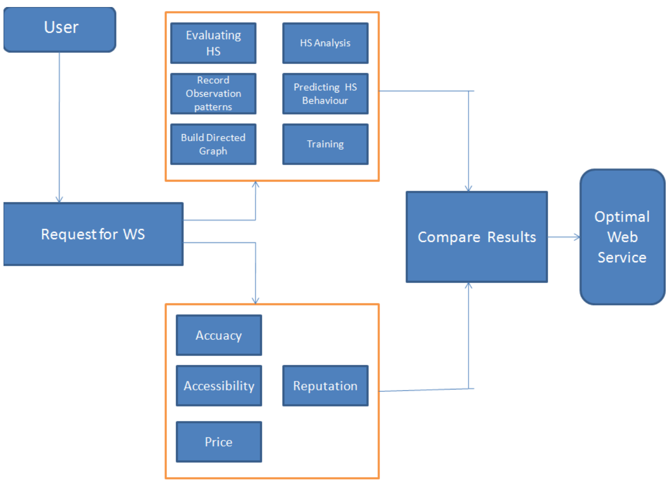

Fig. Proposed System

Training sequence in our proposed model is found by recording and labelling response time of a web service at regular intervals of time. Baum-Welch algorithm a particular case of expectation-maximization (EM) is used to train the model.

After that calculate the web service results by considering QoS parameters like Accuracy, Accessibility, Price and Reputation. And finally by comparing these results with the response time results user gets the optimal web service.

5.MATHEMATICAL MODEL

The purpose of our model is to be able to design quantifiable metrics for quality evaluations by measuring the QoWS of a given web service.

We consider the pentagram model. Each web service has its own key attribute indicator. Price, Latency, Accessibility, Accuracy and Reputations are the five important key attribute indicators of any web service.

For using any web service we have to pay some amount of price. Price factor is depended on the owner of web service.

We represent the price by denoting the symbol 'a'. Latency is the time between the request and getting the results. We define the latency by using symbol 'b'. Good Accessibility is the nothing but the service provider responds to the client request quickly. 'c' represents the accessibility. Accuracy means the probability of the correct response from the service provider to the client. We represent the accuracy by using symbol 'd'. 'e' represents the reputation.

As shown in Figure 4, QoS parameters are represented using a pentagram diagram. Each factor is measured based on the results of their refined factors during QoWS measure process.

Let's assume the measurement result of each factor is a value from 0 to 1. The value "1" indicates the maximum value for each factor, and "0" indicates the minimum value. The area of the pentagram is used as the measurement of QoWS. Clearly, the smallest value of this pentagram area is 0, and the maximum value is approximately 2.4. Since the pentagram consists of five triangles. The area of each triangle can be computed 0.5 X l_1 X l_2 X sinλ : where l_1, l_2, represent the sides of the triangle, and λ represents the 72-degree angle between the two sides. The letters a, b, c, d and e in Figure 4 are used to represent the five factors of QoWS respectively. When each factor is measured, then, QoWS can be computed as:

$$QoWs = \frac{1}{2}\sin 72°(ab + bc + cd + de + ea)$$

$$\cong \frac{1}{2} \times 0.9511 \times (ab + bc + cd + de + ea)$$

$$\cong 0.48 \times (ab + bc + cd + de + ea)$$

Algorithm for Proposed system:
Preprocessing Algorithm:

Step 0: Start

Step 1: Input - User request for web service.

Step 2: Evaluation

– Evaluating Hidden states of requested WS.

– Record observation in Terms of RT of WS

– Build Directed Graph between Hidden states

Step 3: Analysis - Analyzing the current status of each hidden State using HMM Algorithm i.e. behavior of HS

Step 4: Training - Note Down all status of each HS on nth input.

Step 5: Predicting - Predict the behavior of HS in terms of RT using HMM.

Step 6: Results of web service by considering QoS parameters.

Step 7: Comparison of two results.

Step 6: Output - Select optimal Web Service

This preprocessing algorithm is used for proposed system.

6. EXPERIMENTAL RESULTS

For selecting the best optimal web service it is necessary to have list of functionally equivalent web services. In our experiments we have select banking web service data set. There are lots banking web services are available. So it is difficult to select best one from them. In our model we record two results. First is by considering the hidden states i.e. by predicting the behavior of web service in terms of response time. And second is by considering the QoS parameters like Accuracy, Accessibility, Price and Reputation. By considering these two results we select the optimal web service. The collected results are listed in the following table.

	D_a	D_b	D_c	D_d	D_e
1	0.74	0.48	0.69	0.45	1.0
2	1.0	0.94	0.76	0.88	1.0

Table 1. Result Table

D_a, D_b, D_c, D_d and D_e respectively represents the price, latency, accessibility, accuracy and reputation metric of its requirements measurement.

The results of our proposed system are more accurate and more efficient than the existing system.

7. CONCLUSION

There are too many web services that having the similar functionality. So it is difficult to select optimal web service from the list of functionally equivalent web services. To solve this problem we propose a method that selects the best optimal web service with the help of response time and other Qos parameters like Price, Reputation, Accessibility, Accuracy and latency.

We are also using Hidden Markov Model to predict the behavior of web services in terms of response time.

Acknowledgment

It is matter of great pleasure for me to express my gratitude to my guide Prof. P. S. Patil of Computer Engineering, JSPM NTC, Narhe Pune, for his Inspiration, Adroit guidance, Constant supervision, Direction and Discussion in successful completion of this project. I am thankful to Prof. R. H. Kulkarni, Head of department of Computer Engineering, JSPM NTC, Pune, for their valuable support and guidance, also to Dr. D. M. Yadav, JSPM NTC Director and to all our staff members who encouraged me to do this.

REFERENCES

[1] Waseem Ahmed, Yongwei Wu,WeiminZheng, ”Response Time Based Optimal Web service Selection”, VOL. 26 NO. 2, FEBRUARY 2015

[2] V. Grassi, ”ArchitectureBased Reliability Prediction for Service Oriented Computing” 2005, pp. 279299

[3] Z. Zibin and R.L. Michael, “Collaborative Reliability Prediction of ServiceOriented Systems” 2010, vol. 1, pp. 3544.

[4] Z. Yilei, Z. Zibin, and M.R. Lyu, ‘‘WSPred: A Time-Aware Personalized QoS Prediction Framework for Web Services,’’ in Proc. IEEE 22nd ISSRE, 2011, pp. 210-219

[5] F. Salfner, ‘‘Predicting Failures with Hidden Markov Models,’’ in Proc. 5th Eur. Dependable Comput. Conf., 2005, pp. 41-46

[6] M. Cristescu and L. Ciovica, “Estimation of the Reliability of Distributed Applications”, vol. 14, no. 4, pp. 1929, 2010

[7] J. El Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz, “QoS Driven Selection of Web Services for Transactional Composition”,in Proc. IEEE ICWS, 2008, pp. 653660

[8] S. Maheswari, “QoS Based Efficient Web Service Selection”, Eur.J. Sci. Res., vol. 66, pp. 428440, 2011.

[9] C. Leilei, Q. Wang, W. Xu, and L. Zhang, Evaluating the Survivability of SOA Systems Based on HMM, in Proc. IEEE Intl Conf. Web Serv., 2010, pp. 673675.

[10] G. Rahnavard, M.S.A. Najjar, and S. Taherifar, A Method to Evaluate Web Services Anomaly Detection Using Hidden Markov Models, in Proc. ICCAIE, 2010, pp. 261265.

[11] W. Ahmed and Y.W. Wu, ‘‘A Survey on Reliability in Distributed Systems,’’ J. Comput. Syst. Sci., vol. 79, no. 8, pp. 1243-1255, Dec. 2013.

[12] S. Hosseini, S. Unankard, X. Zhou, and S. W. Sadiq, “Location oriented phrase detection in microblogs,” in Proc. 19thInt.Conf.Database Syst. Adv. Appl., 2014, pp. 495–509.

[13] Duhang Zhong, Zhichang Qi and Xishan Xu, ‘‘Reliability Prediction and Sensitivity Analysis of Web Services Composition.