Selection Of Optimal Web Service Of User Request Basaed On Response Time

[1] Mr.Vyavahare M. P. , [2] Prof. Salve B. S.

Department Information Technology, Dattakala Group Of Institute ,Swami-Chincholi , Daund Pune, India

Abstract: *Now a day's web service plays very important role. It is widely used for communication between two different platforms. There are many web service providers that provide web services for users. It is challenging task to select best one from the list of functionally equivalent web services. In the internet world due to the low performance servers, large response time there is lots of sales and customers. To tackle with this problem, existing system consider the hidden states. For selecting optimal web service existing system consider the hidden states and response time parameters. They only consider the response time parameter. There are other QoS factors that also help to find out optimal path. We propose a method that selects optimal web service by considering the response time as well as other parameters like Price, Reputation, Latency, Accuracy and Accessibility of web service. Expected results of 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.

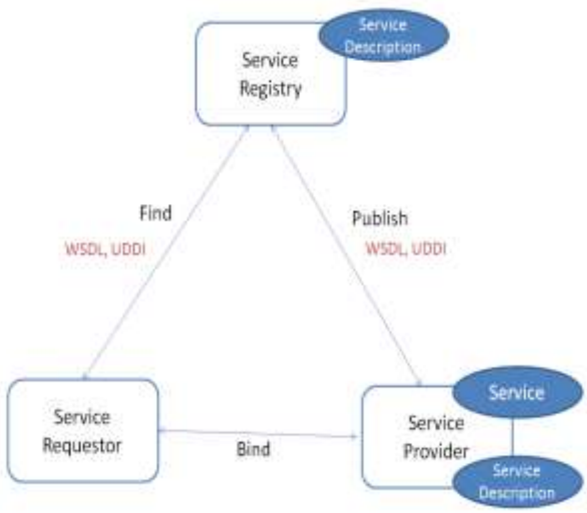

Fig 1 : Web Service Architecture

Now days the use of internet is increases day by day. While proving the services over internet Quality factor plays very important role. In the web, quality of web service is maintained by considering the Quality of Service factor.

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 he 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 retrieves 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. EXISTING SYSTEM

Below figure shows the architecture of Existing system. The service selection system is part of a service registry, in which service providers can publish their services into a Service Repository. When a user is searching for a service, he enters his functional as well as QoS requirements through the Selection UI, and he also has the option to define which decision strategy to follow when selecting services. All the user input is then passed on to the Selection and Ranking Component.

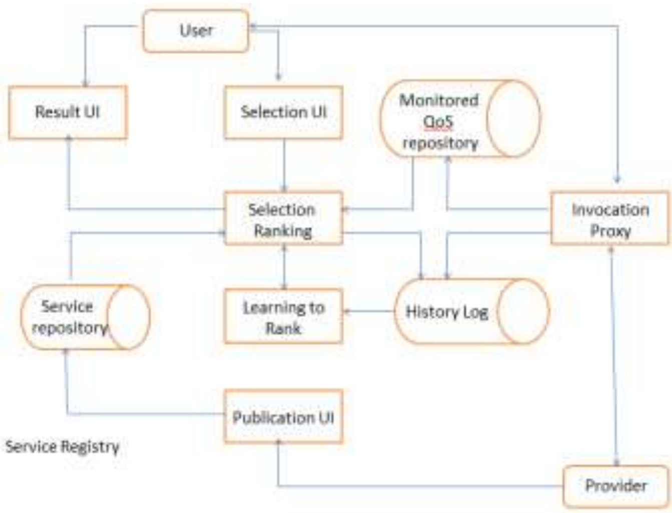

Fig 2 : Existing System Architecture

This component searches for the functionally matching services in the Service Repository first, and then rank these services based on the QoS requirements. If the decision strategy is explicitly defined by the user, one of the 12 decision-strategy-based ranking algorithms is used to rank the services. Otherwise, the personalized ranking algorithm learned through the Learning to Rank Component is used. The ranked list of result is returned to the user through the Result UI. The user then selects a service for the later invocation.

In order for the system to learn user's service selection pattern or decision strategy pattern, the service invocation request is sent to the Invocation Proxy first, and then forwarded to the service owner. The delivered service also goes through the Invocation Proxy first and then to the user, in this way, the actual QoS data can be monitored and saved into the Monitored QoS Repository. The History Log keeps the records of all the users' search requests as well as invocation requests. Every record has the following information: user ID, query, matching services in the result list, and invoked service. With the history data, the Learning to Rank Component can learn the personalized service ranking algorithm for individual users, which could identify user's pattern on following decision strategies.

In this system user select the optimal web service by using the ranking method. User can select web service based on his QoS requirements and his decision strategies. Here in this system hidden states are not considered. Hidden o also plays important role while selecting optimal web service. Service requestor does not know the hidden states of web services means in which state currently the service is, is there is any error occurred while retrieving the web service, etc. So to select optimal web service by analyzing the hidden states we proposed a new system. In this system we are using Hidden Markov Model to find out and analyze the hidden states and also consider the QoS parameters.

5. 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 Requirements:

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

Hardware Requirements:

1. Processor: Dual Core of 2.2 GHZ
2. Hard Disk: 40 GB.
3. Monitor: 14 Color Monitor.
4. RAM: 512 Mb.
5.

Software Requirements:

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

6. PROPOSED SYSTEM

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.

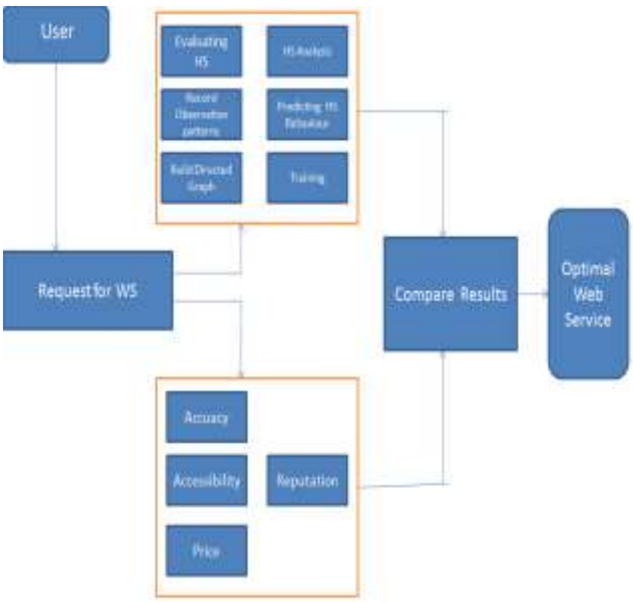

Fig 3 : Proposed System Architecture

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.

7. MATHAMATICAL 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. QoWS is presented in Table 1. 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:

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

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

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

Key Attribute Indicator	Most Important Measures	Symbol
Price	Price represents the amount of money paid for each web service using	*a*
Latency	Latency time between starting the request for a service and addressing the requirement	*B*
Accessibility	Accessibility represents the frequency of the service provider answering the request when the client asks for web service	*C*
Accuracy	Accuracy represents the probability of the request being responded correctly while providers answering the client's requests	*D*
Reputation	Reputation is the credibility of the web service	*E*

Table 1: Parameters of QoWS

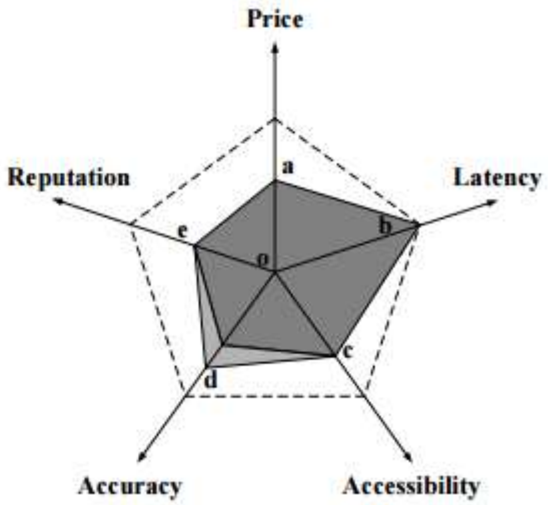

Fig4: Quality of Web Service Pentagram Model

A. Algorithms 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.

8. EXPERIMENTAL RESULTS

1. Home page
2. Online Banking
3. Website visiting
4. Web Services
5. Time Chart
6. Feed Back View

Fig 5: Home Page

Fig 6: Online Banking

Input:

Website visiting request

Output:

Selected website

. Fig 7: Website Visiting

Input:

Website access

Output:

Response time of the web service

Fig 8: Response time of the web service

Input:

Response time

Output:

Response time chart of the web service.

Fig 9: Response time chart

Input:

User feedback

Output:

Feedback view

Fig 9: Feedback View

9. CONCLUSION

As the number of similar functionality web services, the service selection issue is become more complicated. For that we propose a model for predicting the response time of the web service. Also we consider the QoS parameters like Accuracy, Price, Latency, Accessibility and Reputation.

With the help of HMM we predict the behaviour of web service in terms of response time. We record that result. We also record the result of web service QoS parameters. By considering the both results we select the best optimal web service.

10. ACKNOWLEDGMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am presenting this paper on “Selection of Optimal Web Service Of User Request Based On Response Time” as a part of the curriculum of M.E. Computer Engineering at DATTAKALA GROUP OF INSTITUTE,SWAMI-CHINCHOLI,DAUND,PUNE. It gives me proud privilege to complete this paper work under the valuable guidance of Prof. Salve B. S. I am also extremely grateful to Prof. Bere S.S. and Prof. Prof. Amrit Priyadarshi for providing all facilities and help for smooth progress of paper work. I would like to express my gratitude towards my parents all the Staff Members of Computer Engineering Department, for their kind co-operation and encouragement which help me in completion of this paper. My thanks and appreciations also go to my colleague and people who have willingly helped me out with their abilities.

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] B. Sami, G. Claude, and P. Olivier, ''Transactional Patterns for Reliable Web Services Compositions,'' in Proc. 6th Int'l Conf. Web Eng., Palo Alto, CA, USA, 2006, pp. 137-144.

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