Optimal Web Service Selection According To Response Time

U. B. Bagal[1], R. N. Pawar[2], R. S. Jamgekar[3]
[1]Assistant Professor, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur, India*
[2]ME- CSE, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur, MS, India*
[3]Assistant Professor, *NBNSCOE, Solapur, Solapur University, Solapur, MS, India*
[1]umaji.bagal@sknscoe.ac.in, [2]pawarrupa22@gmail.com, [3]rs.jamgekar@gmail.com

Abstract— Selecting an optimal web service among a list of functionally equivalent web services still remains a challenging issue. Some issue of internet service is poor service quality, low performance servers, high latency can lead to lost sales, customer lost and user's frustration. In these studies we proposed a method QoS metrication which is based on Hidden Markov Model (HMM) and which is suggest optimal path for the execute user request efficiently. In the term of response time these technique measure and predict behaviour of Web services also used for rank services quantitatively rather than just qualitatively.by some experiments of real world data we define reliability and usefulness of our proposed method. The results have shown how our proposed method can help the user to automatically select the most reliable Web Service taking into account several metrics, among them, system predictability and response time variability module .

Keywords— Service composition, service selection, quality of service, quality of experience, Response Time, QoS.

I. INTRODUCTION

Internet makes the world into a smaller place. Companies from all around the world may now compete over different services offerings not only with their local adversaries, but now under a global scale. Escalating the competition and lead in industry segment can often be a matter of offering and perhaps even most importantly, assuring the good quality of the services offered. In the Web this should not be different; QoS (Quality of Service) policies help for control on quality of web service and taken assurance of quality conditions of web services are always met. A web service is method in which we communicate between two electronic devices over the network. As in the concept describe in computing it is a software function provided on web server over the network. The W3C defines a Web service generally as: -To increase the machine to machine interaction over the network a software system has been designed. Web services are modular, self-contained, distributed and dynamic application, implore over the network which generate product, process. Application will be distributed or web base and local. Web services build on the above of open standard like as HTTP, TCP-IP, XML file and HTML. For the direct application to application interaction used web services which are XML based information exchange system. These systems can include programs, objects, messages, or documents. Web service used for exchanging data between system and application as web service collection of standard and open protocol. Software applications will be written in many different programming languages and running on different platform, to exchange data used wed service over the computer networks like as internet in case of internally communication on a single computer. This interoperability (e.g. between Java and Python, or Windows and Linux applications) is due to the use of open standards.

1.1 Web Service Architecture

Many organizations use multiple software systems for management. Various software systems need to be exchange the data with each other and web service is used for the communication between two systems or applications over the network. Service requester is request data to software system where system processes the request and provides the data called service. Using different language software might be developed and however there is need a method to exchange data efficiently that doesn't depend on that particular programming language. Most of the software use XML tags. However by using XML tags web service can exchange the data. The web service architecture can be described as below in fig 1.and define work of service broker, service provider and service requester. Web service discrimination language file (WSDL) can be send to Universal description, discovery and integration (UDDI) by service provider. If the requester need data then he contact with UDDI for the find out who is provider and by using SOAP (Simple object Access Protocol) Protocol requester can contact with service provider. Then the service provider validate the request and process request and send it is an XML file using the SOAP (Simple Object Access Protocol) protocol. This XML file would be validated again by the service requester using an XSD file.

Universal Description, Discovery and Integration (UDDI) are a platform-independent, Extensible Mark-up Language (XML)-based on registry by which world wide businesses can list themselves on the Internet, and a mechanism to register and locate web service applications. UDDI was originally defined as core web service standard. It will be design to cross questioning by SOAP (Simple object Access Protocol) massage and provide access WSDL documents and message formats required to interact with the web services listed in its directory.

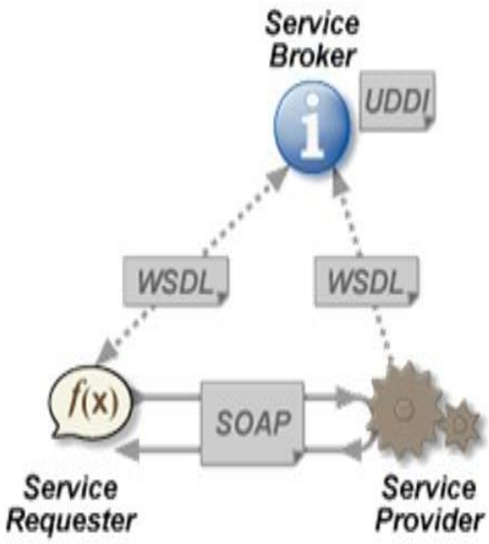


Fig.1 Web service architecture.

Universal Description, Discovery and Integration UDDI:

Web Services Description Language WSDL:

The Web Services Description Language (WSDL) is an XML-based interface definition language that is used for describing the functionality offered by a web service. The acronym is also used for any specific WSDL description of a web service (also referred to as a WSDL file), which provides a machine-readable description of on which way service will be called, which parameter required and result of data structure.

Simple Object Access protocol SOAP:

In computer network Simple Object Access protocol is used for the exchange structured information in the implementation of web services in computer networks. It will be used XML file for the massage format and use on another application protocol, mostly SMTP and HTTP for massage exchange.

Service Oriented Architecture (SOA):

It is design pattern based on the pieces of software which is provide functionality via protocol. This is known as service-orientation. It is independent of any vendor, product or technology. A service is a self-contained unit of functionality, such as retrieving an online bank statement. SOA is based on the concept of a service. On the base of service design approaches every Service Oriented Architecture system can design to perform more service operations. Every services discrete code of piece. With these help easy to reuse code in another way through application by changing the way an individual service with other services that make up the application another way service can change code itself. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. SOAP defines that as per the web base environment how to integrate largely change the applications behaviour instead of define API and SOA define protocols and functionality.

II. LITERATURE SURVEY

2.1 Title: Response Time Based Optimal Web Service Selection

Author: Waseem Ahmed, Yongwei Wu, Member, IEEE, and WeiminZheng, Member, IEEE

Year: 2015

Suggested [1] the overall behaviour of composite web service but it further provides the solution to complete user requests in the most efficient and reliable way. Predicted web services behaviour is by predict the status of underlying hidden states in terms of Response Time. Select optimal Web services and an optimal path at runtime for executing user request by identifying the status of underlying hidden states. The probabilistic insight of Web services used Hidden markov Models (HMM). In our model assumed that WS is deployed on a cluster of web servers and sometime the delay or crash during WS Invocation is because the bad node in sever clustering responds to user's requests.

2.2 Title: Quality Prediction of Service Compositions through Probabilistic Model Checking

Author: G. Stefano, C. Ghezzi, R. Mirandola, and G. Tamburrelli

Year:2008

Suggested [3] the Service-Oriented Architectures (SOAs) provide a new paradigm for the creation of business applications. It proposed a model driven approach, which automatically transforms a design model of service composition into an analysis model, which then feeds a probabilistic model checker for quality prediction. One of the first works in this area is proposed a framework for composed services modelling and QoS evaluation is presented. A composite service is modelled as a directed weighted graph where each node corresponds to a Web Service (WS) and edge weights represent the transition probabilities of two subsequent tasks. The author shows how to evaluate quality of service of a composed service from basic services characteristics and graph topology.

2.3 Title: Composing Web Services: A QoS View

Author: D.A. Menasce et al.

Year:2004

Suggested [4] Service-oriented architecture (SOA) is strong architecture for generated complex distributed systems by discovering and integrating web services provided by different organizations. Reliability of the Service related system based on remote web service as well as unpredicted internet. The results had shown how a collaborative reliability prediction approaches used for service-oriented systems. By using this approach, without requiring real world web service data predict the web service reliability for the current user.

2.4 Title: QoS Analysis for Web Service Compositions Based on Probabilistic QoS

Author:H. Zheng, J. Yang,W. Zhao, and A. Bouguettaya

Year:2011

Suggested [5] Reliability is an important feature for use in distributed applications to estimate the reliability of these applications using object-oriented design metrics validation techniques. Reliable distributed application is defined as a system whose behaviour is predictable, in spite of partial failures and reconfiguration. An obvious benefit of distributed systems is that they serve the global business and social environment in which we live and work. Another benefit is that they can improve the quality of services, in terms of reliability, availability and performance, for the complex systems.

III. PROPOSED SYSTEM

3.1 Hidden Markov Model

Hidden states of requested WS by using HMM. HMM is a powerful tool for modelling generative sequences that can be characterized by an underlying process generating observable sequences The HMM is a powerful statistical tool for modelling generative sequences that can be characterized by an underlying process generating observable sequences. Word hidden specifies that internal structure of the underlying system is hidden from the observer. Observer does not know in which state system may be in, but has only probabilistic insight where it should be. In HMM one does not know how many hidden states to use. Usually, based on domain knowledge there is only some guess about hidden states. Later training algorithm will find out how to connect these hidden states. HMM can solve three fundamental issues i.e. Evaluation, Decoding, Training. Using HMM to measure and predict WS behaviour with respect to response time, our model consists of a two-step process.

3.2 Analysing Current Status

The current status of underlying hidden states is analysed using HMM. Each hidden state is analysed in terms of QoS attribute i.e. response time. State of WS during time interval 't' can be considered as vector of probabilities that WS is in hidden state executing a certain request 'R' during time interval 't' having pattern of response time
O =O1;O2; . . .;On.

The current state of WS can lead us to predict state of WS during the n^{th} time interval under various operational conditions. State of WS during time interval 't' can be computed with the help of HMM i.e. d(i)=HS1,.....HS(n-1),HS(n)=i,01,.....o(n). Here d(i) represents the state of WS i.e. it represent maximum probability (computing maximum over all possible hidden states sequences) that themodel went through hidden states HS1; . . . ; HS(n-1) and the system is in state'I' at hidden state 'n'. i.e. HS(n-1) while observing O= O1;. . . ;On. Equation (3) in HMM is known as VITERBI algorithm, which can help users to build relations among hidden states Si and response time pattern generated by each hidden state S_i during time interval t_i. VITERBI algorithm can further be used to calculate the probability of a current status of any specific 'c' hidden state HS $_i$. Let Q_k j be the probability of observations from k1 to n such that $HS_{k=}j$.

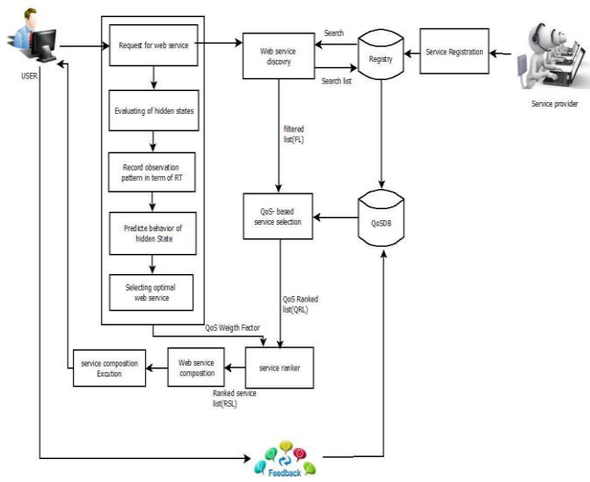


Fig.2 Proposed System

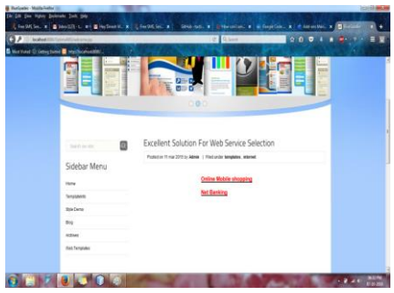
3.3 Predicting Future Behaviour of Component Service

To find the probabilistic behaviour of each hidden state of a WS for defined response time pattern against different user requests, we can calculate $P_{ur}(n)$ i.e. probability of hidden state HS_k in the k^{th} time interval is unreliable or hidden state HS_k producing results with delay during k^{th} time interval. This probability is calculated by First Passage Time Distribution. Let,T_k be the time in the k^{th} time interval, also known as First Passage Time, when hidden state HS_k produced delayed results, then Here HS_k represents the hidden state at time 'k' during the n^{th} time interval. Probability distribution among hidden states can be computed. PHS_ji is the probability of response time patterns that WS is in hidden state 'j' at current time as computed $PTkGn_jHS_j$i is the probability of going through hidden state during the nth time interval starting from j which can be computed recursively. Probability distribution that hidden state H S_{ur} produces unreliable results during time interval 'k' which can be further scrutinized using dynamic programming to efficiently compute for various time intervals.

3.4 Optimal Web Service Selection

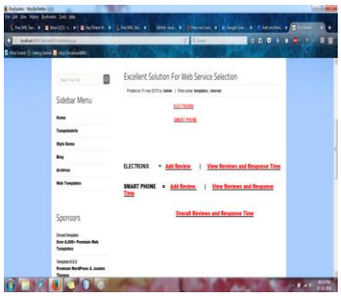
Here in this step select the optimal web service among the list of functionally equivalent web services. Predicting hidden states behaviour in terms of response time during nth time interval t. Then, finally selecting optimal web services used in composition based on hidden states behaviour.

IV. RESULTS AND EVALUATIONS



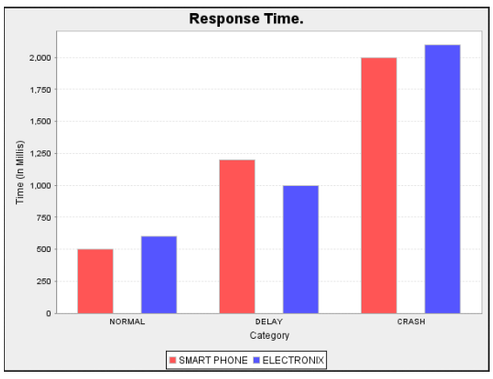
Snapshot:1. Home screen

Above figure shows the home page of our project user can login into the system then user can see the above results, Then user request for web service.



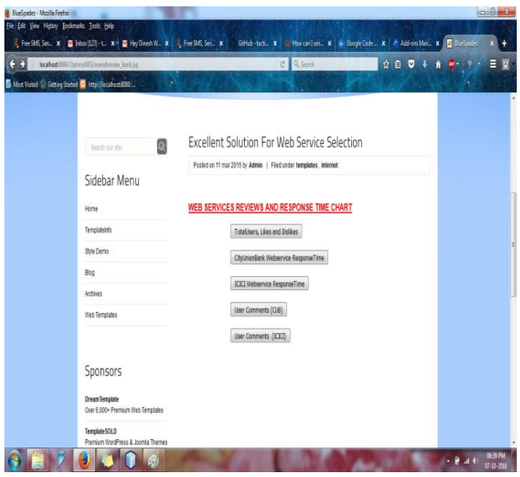
Snapshot 2: add and view review

Above figure shows the add and view review criteria. In this figure shows the user can add their review related to that service and his also view reviews.



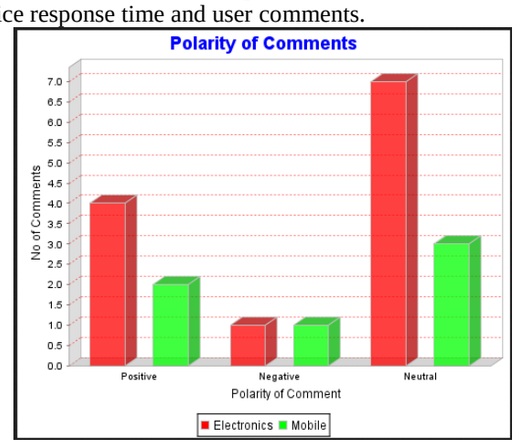
Snapshot 3: response time chart

Above figure shows the Smart Phone & Electronix response time chart. normal, delay, and crash response time chart.



Snapshot 4: web service review and response time chart

Above figure shows the all chart related to web service for Example like dislike, response time, total users, city union bank web service response time, ICICI bank web service response time and user comments.



Snapshot 5: Polarity Of Comments

Above figure shows the Polarity of Comments. Comment shows in the form of Positive, Negative, Neutral.

V. CONCLUSION

In this paper, a probabilistic model for predicting the response time of the web service. With the help of response time we select the optimal web service. To know the probabilistic insight of WSs we have used HMM. We have assumed that WS is deployed on a cluster of web servers and sometime the delay or crash during WS invocation is because the bad node in sever clustering responds to user's requests. With the help of HMM we have predicted the probabilistic behavior of these web servers and then selected the WS based on their probabilistic value. This not only predicts the overall behavior of composite web service but it further provides the solution to complete user requests in the most e client and reliable way.

FUTURE WORK

There are several directions for future research. A first point to investigate is related to the extension of the framework to support dynamic composition of web services. The basic change concerns essentially the construction of the potential compositions and their evaluations.

ACKNOWLEDGMENT

The author would like to thank the anonymous reviewers for their valuable comments and suggestions. We would like to express our special thanks of gratitude to Prof. R.S.Jamgekar for his- enthusiastic assistance in improving the clarity of this article.

REFERENCES

[1] A.V. Grassi, ''Architecture-Based Reliability Prediction for Service-Oriented Computing,'' in Architecting Dependable Systems III. Berlin, Germany: Springer-Verlag, 2005, pp. 279-299.

[2] V. Cortellessa and V. Grassi, ''Reliability Modeling and Analysis of Service-Oriented Architectures,'' in Test and Analysis of Web Services. Berlin, Germany: Springer Verlag, 2007, pp. 339-362.

[3] G. Stefano, C. Ghezzi, R. Mirandola, and G. Tamburrelli, ''Quality Prediction of Service Compositions through Probabilistic Model Checking,'' in Proc. 4th Int'l Conf. Quality Software- Architect., Models Architect., 2008, pp. 119-134.

[4] D.A. Menasce, ''Composing Web Services: A QoS View,'' IEEE Internet Comput., vol. 8, no. 6, pp. 80-90, Nov. 2004.

[5] H. Zheng, J. Yang,W. Zhao, and A. Bouguettaya, ''QoS Analysis for Web Service Compositions Based on Probabilistic QoS,'' in Service-Oriented Computing. Berlin, Germany: Springer-Verlag, 2011, pp. 47-61.

[6] Z. Zibin and R.L. Michael, ''Collaborative Reliability Prediction of Service-Oriented Systems,'' in Proc. 32nd ACM/IEEE Int'l Conf. Softw. Eng., Cape Town, Africa, 2010, vol. 1, pp. 35-44.

[7] R. Perrone, R. Macedo, G. Lima, and V. Lima, ''An Approach for Estimating Execution Time Probability Distributions of Component-Based Real-Time Systems,'' J. Universal Comput. Sci., vol. 15, no. 11, pp. 2142-2165, 2009.

[8] M. Cristescu and L. Ciovica, ''Estimation of the Reliability of Distributed Applications,'' Inf. Econ., vol. 14, no. 4, pp. 19-29, 2010.

[9] D. Zhong, Z. Qi, and X. Xu, ''Reliability Prediction and Sensitivity Analysis of WS Composition,'' in Petri Net: Theory and Applications, V. Kordic, Ed. Rijeka, Croatia: Intech, 2008, pp. 459-470.

[10] J. El Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz, ''QoSDriven Selection of Web Services for Transactional Composition,'' in Proc. IEEE ICWS, 2008, pp. 653-660.