ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

QOS Based Web Service Selection Using Ant Colony Optimization

T.Usha[1], M.Thamizharasi[2]

PG Scholar, Dept. of CSE, Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, India [1]

Associate Professor, Dept. of CSE, Dhanalakshmi Srinivasan College of Engineering & Technology, Chennai, India [2]

ABSTRACT: Selecting an optimal web service among a list of functionally equivalent web services still remains a challenging issue. For Internet services, the presence of low-performance servers, high latency or overall poor service quality can translate into lost sales, user frustration, and customers lost. Existing system based on Hidden Markov Models (HMM), which further suggests an optimal path for the execution of user requests. It only calculate response time. In this project propose three different algorithm Ant Colony (-based) Optimisation, genetic algorithm and Analytic Algorithm. The technique is show can be used to measure and predict the behavior of Web Services in terms of cost, availability and response time can thus be used to rank services quantitatively rather than just qualitatively. Anti-colony optimisation algorithm used to classify the response time.Genetic algorithm used to detect particular web service cost and analytic algorithm used to check the availability of webservices. It demonstrate the feasibility and usefulness of the methodology by drawing experiments on real world data. The results have shown how the 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.

I. INTRODUCTION

Web Services mean many things to many people. In the end, there will be a set of standards which allow us to do things could not do before, but in the meantime different people and companies approach them from different positions, and with different expectations. In 2001-2, Web Services have also been a buzzword used repeatedly and claimed to be one of the hot new technologies. The common themes are:-

- A departure from the web as a quasi-static information space to one in which interactions are the primary model
- A use of HTTP, XML and other standards from the web architecture as the building blocks
- A typical focus on enterprise wide and inter-enterprise operations

The Web in Web Services is, from the first point, a misuse: the term Internet Services would be more appropriate. The Web comes from the second point - the use of the HTTP and XML is already in use as a well-understood and well-debugged set of protocols which support the Web, and so it makes sense to reuse them in providing remote operations and those things connected with them. The third point is what makes web service requirements so different from a local RPC system. The fact that data is exchanged for business purposes and between different social entities means that accountability is required, rather than just reliable transmission.

- The vendors of software see web services as way to repackage existing capability in a way which makes it interoperable with other systems.
- The security requirements for web services are dictated by the trust environments, whether it is intranet or b2b or b2c, etc.
- For b2b one needs not just reliability but accountability.

The architecture of Web Services is the scope of the W3C Web Service Architecture Working Group.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

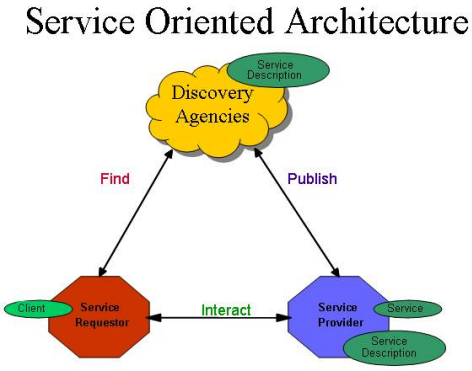


Fig. 1 Service oriented Architecture

However, the essential part of Web services is the Interact relationship between a Service provider and Service requestor. This is the Web Service. Discovery agencies need not be used - it will in some cases but not in others. The discovery agencies are well represented as a cloud, rather than being a well-defined module in the web services architecture. It will become interface to a huge world of data and query services which provide data about web services as well as many other things. The Interact between requestor and provider is the essential defining element to web services. As shall see, the metadata about web services

There is a mass of different pieces being bolted onto the foundations of Web Services provided by WSDL and SOAP 1.2 and the diagram implies things considerably. The management layer is a supervisory layer allowing the control of the many agents involved in a web services-based operation. The "Application semantics" layer indicates the necessity, for any useful interoperability, to have

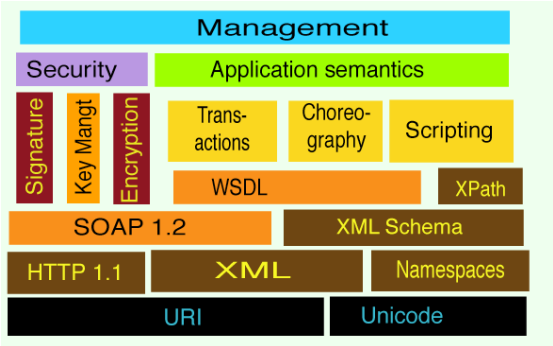


Fig. 2 Webservices Diagram

The question of the relationship between these two activities is constantly in the air.The whole description side is a clear semantic web application, and so long as XML languages are defined which introduced with English language specs but no RDF mapping, there is a potential ambiguity which will have to be resolved later in making that mapping, there is an inability to use common semantic web tools, and there is cost down the road assuming semantic web tools will eventually be used. Essentially, web serve ices become instant legacy technology for the semantic web.The DAML-services collation of researchers is tackling the job of service description at a higher level.Many things which are described as web services can in fact be described as the publication of a series of semantic web documents, just as the billing of a peer company is in reality effected by the issuance of an invoice.When Semantic Web agents query each other, it could use SOAP (though a direct encoding into an HTTP URI may also be effective).When Semantic Web agents update each other, it should use SOAP, running typically over HTTP POST.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

II. SYSTEM ARCHITECTURE

Design is a multi- step that focuses on data structure software architecture, procedural details, algorithm etc… and interface between modules. The design process also translate the requirements into presentation of software that can be accessed for quality before coding begins. Computer software design change continuously as new methods; better analysis and border understanding evolved. Software design is at relatively early stage in its revolution.

Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are normally associated with more classical engineering disciplines. However techniques for software designs do exit, criteria for design qualities are available and design notation can be applied.

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

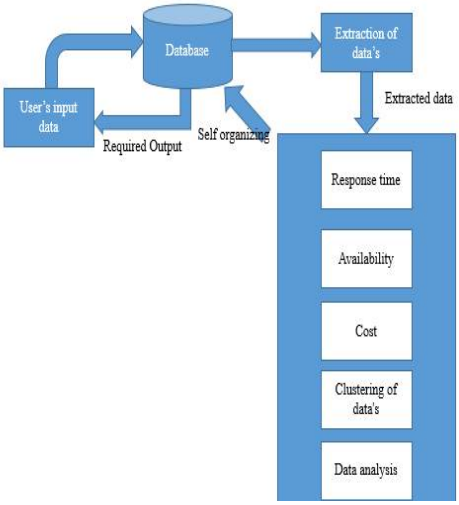


Fig. 3 System Architecture.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

III. INPUT DATA CONVERSION

Generally atomic web service is composed of different hidden states that are invisible to consumers. Each state is responsible to output certain results during time t. The probability of the response to certain requests depends on execution of these hidden states. When a consumer of a web service accesses a remote web service, sometimes he receives an unprecedented delay even under best operational conditions. It believe that this exceptional delay is because of unreliable hidden states responding to users' request at that particular time. Delay represents the state where system receives the results after long time, however, crash/error represents the state where WS crashes or user receives no response from WS.

Hidden states can entertain user's requests randomly and produce results anytime. As these hidden states can also be accessed from other hidden states while service invocation, hence the model is of ergodic type. There may be/exist certain hidden states that may produce results in similar patterns of time intervals of having different configurations. Communication delay or latency for calling another service inside target web service will also be involved in overall response time) clear identification among feature values is required which in machine learning is referred to as Feature Normalization. These features may be categorized in terms of memory requirements, network requirements, software service requirements etc. This will help to define the initial values of the probability of success or failure of hidden states. It shows general implementation of a WS used in exemplary scenario. User receives different observation symbols A, B, and C while he invokes a web service during a certain time interval t.

It iteratively improves the basic model which provides convergence to local optima, whereas second step, first requires us to compute current state of the system. Then based on current state, future behavior of the system is predicted. This can be computed using Ant colony optimization, genetic and analytical algorithm. Based on above two steps, for selecting an optimal WS and an optimal path for executing user requests strategy can be further divided into following steps:

1. Building a directed graph among hidden states of component web services used in composition.
2. Analyzing the current status of each vertex of directed graph i.e., underlying hidden states.
3. Predicting hidden states' behavior in terms of response time during nth time interval t.
4. Finally, selecting optimal web services used in composition based on hidden states' behavior.

Adjust the model parameters to analyze current state of the internal structure of web services. Build a directed graph using concept of hidden states to select optimal path for executing the user request. Predict the future behavior of these web services during nth time interval to select better web service for optimal composition. This can be useful in building a directed graph among hidden states of web service in group1 to the hidden states of web service in group2 at time t. A directed graph is developed among 12 hidden states of 3 web services WS3, WS4, and WS5 from group1 and 16 hidden states of 4 web services WS1, WS2, WS3, and WS4 of group 2.For simplicity we have shown only two web services with their corresponding hidden states. When user connects with the system, then based on prediction results system finds the WS, probabilistic behavior of whose hidden states are at their best level during that interval and then connect with it. Later algorithm1 is exploited to select an optimal path with minimum probabilistic value for executing user requests in the most efficient and reliable way.

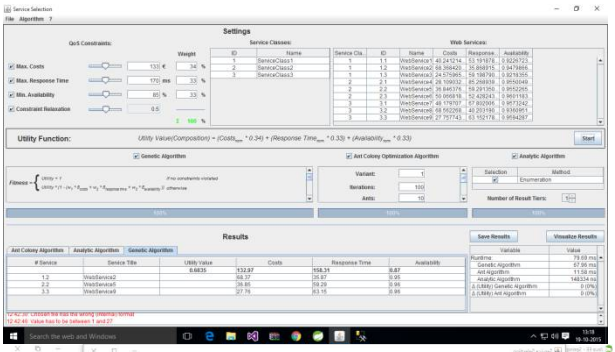
ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

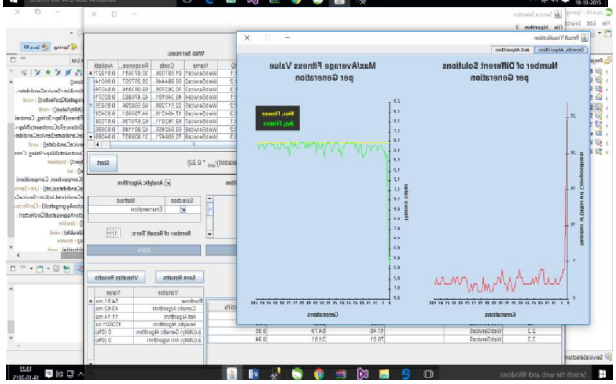
International Journal of Innovative Research in Science, Engineering and Technology

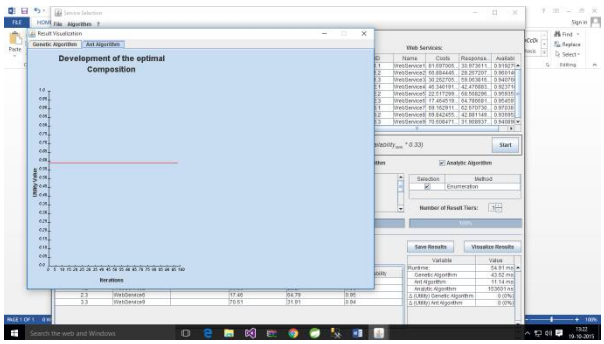
(An ISO 3297: 2007 Certified Organization)

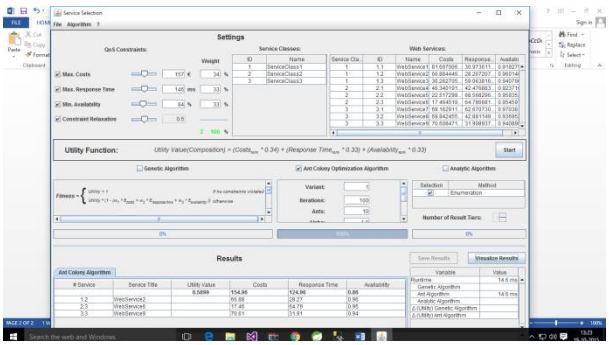
Vol. 5, Issue 4, April 2016

IV. RESULTS







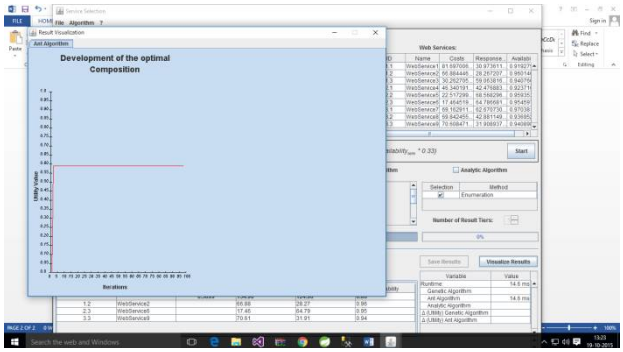


ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016



V. CONCLUSION

In this paper, we at first propose a probabilistic model for predicting response time of web service and then selected an optimal web service at runtime from the list of functionally equivalent web services. To know the probabilistic insight of WSshave used to Ant colony optimization, genetic and analytical algorithm. In the model 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 users' requests. With the help of Ant colony optimization, genetic and analytical algorithm have predicted the probabilistic behavior of these web servers and then selected the WS based on their probabilistic value. Experiment shows that the proposed model is more general and detailed in comparison to existing models. This not only predicts the overall behavior of composite web service but it further provides the solution to complete user requests in the most efficient and reliable way.

REFERENCES

1. V. Grassi, ''Architecture-Based Reliability Prediction for ServiceOriented Computing,'' inArchitecting 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,'' inTest 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 SoftwareArchitect., 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,'' inProc. 32nd ACM/IEEE Int'l Conf. Softw. Eng., Cape Town, Africa, 2010, vol. 1, pp. 35-44.
7. R.Perrone, R.Macedo, G.Lima,andV.Lima,''An Approachfor Estimating Execution Time Probability Distributions of ComponentBased 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,'' inPetri 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,'' inProc. IEEE ICWS, 2008, pp. 653-660.
11. B. Sami, G. Claude, and P. Olivier, ''Transactional Patterns for Reliable Web Services Compositions,'' inProc. 6th Int'l Conf. Web Eng., Palo Alto, CA, USA, 2006, pp. 137-144.
12. L. Li, L. Chengfei, and W. Junhu, ''Deriving Transactional Properties of Composite Web Services,'' inProc. IEEE ICWS, 2007, pp. 631-638.
13. K. Boumhamdi and Z. Jarir, ''A Flexible Approach to Compose Web Services in Dynamic Environment,''Int'l J. Digit. Soc., vol.1, no. 2, pp. 157-163, 2010.
14. Y. Tao, Z. Yue, and L. Kwei-Jay, ''Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints,'' ACM Trans. Web, vol. 1, no. 1, p. 6, May 2007.
15. 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.
16. 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.
17. S. Maheswari, ''QoS Based Efficient Web Service Selection, 'Eur.J. Sci. Res., vol. 66, pp. 428-440, 2011.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

18. C. Leilei, Q. Wang, W. Xu, and L. Zhang, ''Evaluating the Survivability of SOA Systems Based on HMM,'' inProc. IEEE Int'l Conf. Web Serv., 2010, pp. 673-675.
19. G. Rahnavard, M.S.A. Najjar, and S. Taherifar, ''A Method to Evaluate Web Services Anomaly Detection Using Hidden Markov Models,'' inProc. ICCAIE, 2010, pp. 261-265.
20. F. Salfner, ''Predicting Failures with Hidden Markov Models,'' in Proc. 5th Eur. Dependable Comput. Conf., 2005, pp. 41-46.
21. M. Zaki, A. Ihsan, and B. Athman, ''Web Services Reputation Assessment Using a Hidden Markov Model,'' inProc. 7th Int'l Joint Conf. Serv.-Oriented Comput., 2009, pp. 576-591.
22. 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.
23. P. Blunsom, Hidden Markov Models, Retrieved May 19, 2006, and 2004. [Online]. Available: www.cs.mu.oz.au/460/2004/materials/hmmtutorial.pdf.
24. L. Li, L. Chengfei, and W. Junhu, ''Deriving Transactional Properties of Composite Web Services,'' inProc. IEEE ICWS, 2007, pp. 631-638.
25. K. Boumhamdi and Z. Jarir, ''A Flexible Approach to Compose Web Services in Dynamic Environment,''Int'l J. Digit. Soc.,vol.1,no. 2, pp. 157-163, 2010.