International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||4 ||Pages|| 768-772 ||2013||
Website: www.ijsrm.in ISSN (e): 2321-3418

AUTONOMIC COMPUTING FOR PROCESSING THE COMPLEX WEB SERVICE

S.Karthikeyan[1], K.J.Jagdish [2]

[1]PG Student,
Affiliated to Anna University Chennai, Dept. of Computer Science and Engineering
Gnanamani College of Engineering,
Namakkal, India
sksrec@gmail.com

[2]Asst. Prof Dept. of CSE
Dept. of Computer Science and Engineering
Gnanamani College of Engineering,
Namakkal, India
kjjagdishpgp@gmail.com

Abstract:*Quality of Service (QoS) analysis for Web service compositions is an important and challenging issue in distributed computing. In existing work, there is a systematic approach is followed to calculate QoS for composite services with complex structures. This method is not effective to find the path probability information's as well as there is some restrictions to find the full path. In fact this existing method can't be the best solution for the optimum performance. Hence this project proposes QoS Ontology with ETQoS: Enhanced Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. The design which builds a general QoS ontology to support web services nonfunctional features. This process addresses the issue of selecting and composing Web services not only according to their functional requirements but also to their transactional properties and QoS characteristics. This method can also satisfy the user's preferences over QoS criteria. Experimental results show that the proposed ETQoS calculation approach significantly improves the efficiency in probabilistic QoS estimation.*

Keywords: Quality of Services, Web Services, Services Level Agreement, Asynchronous Transfer Mode

1.Introduction

WEB Services (WSs) are the most famous implementation of service oriented architectures allowing the construction and the sharing of independent and autonomous software's. Web service composition consists in combining Web services, developed by different organizations and offering diverse functional (e.g. ticket purchase, payment), behavioral (e.g. compensable or not) and non-functional properties (i.e. Quality of Service values – e.g. execution price, success rate), to offer more complex services.

A Web Service is a self-describing, self-contained software module available via a network, such as the Internet, which completes tasks, solves problems, or conducts transactions on behalf of a user or application. Web services constitute a distributed computer infrastructure, made up of many different interacting application modules trying to communicate over private or public networks to virtually form a single logical system.

A web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML.

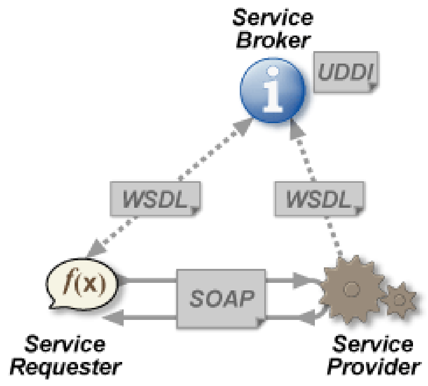

Fig 1: Web Service Architecture

Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML-based messages conveyed by Internet protocols.

In essence, a web service provides an interface defined in terms of XML messages and that can be accessed over the Internet (or, of course, an Intranet). It is an application that exposes a function which is accessible using standard Web technology and that adheres to Web services standards. This is significant because Web services are developed for and deployed onto any platform using any programming language.

2.Related Work

2.1 Quality of Services (QoS)

QoS refers to the ability of the Web service to respond to expected invocations and to perform them at the level commensurate with the mutual expectations of both its provider and its customers. Several quality factors that reflect customer expectations, such as constant service availability, connectivity, and high responsiveness, become key to keeping a business competitive and viable as they can have a serious impact upon service provision. QoS thus becomes an important criterion that determines the service usability and utility, both of which influence the popularity of a particular Web service, and an important selling and differentiating point between Web services providers.

Delivering QoS on the Internet is critical and significant because of its dynamic and unpredictable nature. Applications with very different characteristics and requirements compete for all kinds of network resources. Changes in traffic patterns, securing mission-critical business transactions, and the effects of infrastructure failures, low performance of Web protocols, and reliability issues over the Web create a need for Internet QoS standards. Often, unresolved QoS issues cause critical transactional applications to suffer from unacceptable levels of performance degradation.

A probability for transitions in a composite service is important information for QoS calculation. The method of estimating transition probabilities in a composite service has been given. The designer sets the initial transition probabilities at design time. At runtime, the transition probabilities are recomputed based on runtime data generated from past executions that have been stored in the database log. Cardoso et al. also give an example of estimating the transition probabilities in a loop. In the composite service for DNA sequencing test experiment, the experiment will start again if there are E. coli bacteria in the DNA sequence. The researcher sets the back transition probability for the loop by a prior knowledge of sequencing experiments, in which approximately 10 percent of the DNA sequence will contain E. coli bacteria. Therefore, the back transition probability for the loop is set to 10 percent. Aggregation method is proposed to calculate QoS for composite services. The QoS of a composite service is calculated as the aggregation of the QoS of composition patterns.

The authors of can only deal with simple patterns and cannot cover complex structures such as arbitrary loop and conditional patterns with component services shared by multiple branches. Only two types of loop patterns, i.e., self-loop and dual vertex-loop, have been considered. The QoS of a conditional pattern is calculated as the probability weighted sum of the QoS of the conditional branches, which is not always the truth. A loop is assumed to be executed a constant number of times. The maximum and minimum QoS of each type of patterns are calculated. For example, the maximum and minimum execution time for a conditional pattern are the execution time of the path with the maximum and minimum execution time, respectively, while the maximum and minimum execution time for a parallel pattern are calculated as the maximum execution time of the paths. The probability information of each execution path of the composite service is lost during the QoS calculation in both. Aggregation method has been widely used in calculating QoS for composite services in service selections. The efficiency of service selection has been improved by methods proposed. QoS-driven adaptive service selection approaches also adopt aggregation method to predict the QoS for a composite service and selects composition plan based on the predicted QoS at design time. Then, QoS monitoring and prediction mechanisms are used to trigger the preplanning of composite services at runtime. Another trend in QoS calculation is using the QoS of the critical path as the QoS of a composite service. A critical path of a composite service is the path that takes the longest time to finish.

A loop in is unfolded based on the maximum number of times that the loop is taken. The limitation of selecting services based on the QoS of the critical path has been discussed. To overcome the problems caused by calculating QoS based on critical path, the QoS of every path of the composite service is calculated and guaranteed to satisfy the user specified requirements. Ardagna and Pernici assume that the possible loop times and the occurring probabilities for the loop times are known in a priori. All the work mentioned above treats QoS as constant values (mean, maximum, or minimum of QoS) and do not consider the probabilities of execution paths of a composite service. The QoS of component services are seen as random variables and represented by either discrete or standard statistical distributions (e.g., normal and exponential). Hwang et al. adopt aggregation method in to calculate QoS distribution for composite services while uses simulation methods to estimate the QoS distributions.

3.Transactional Composite Web Service

The selected component WSs is properly ordered to obtain a TCWS which maximizes the user satisfaction in terms of QoS criterion and satisfies the transactional requirements set by the user and by the input workflow. The input of the assignation process is a workflow, a transactional requirement, expressed in term of risk and a set of weights over QoS criteria. The output of the process is a TCWS which assigned WS components maximize the QoS criteria. The process is sequential: WSs are assigned to each activity by analyzing the workflow from the left to the right. In a split pattern, services are assigned from top to down.

In order to explain the transactional WS selection process, it is necessary to establish how the user can express their transactional criteria. Although, expressing the QoS criteria is significant for the user, the risk or the possibility that an application will be unsuccessfully executed has a more significant effect on the user's decisions. The importance of the uncertainty of application completion and recovery is semantically expressed under a criterion called risk. For example, the set {!a,!ar, c, cr} of the behavioral properties of CWS can be divided into two subsets {!a,!ar} and {c, cr}, each one can be associated with a level of risk. For instance, in terms of the transactional properties, we believe that properties !a and

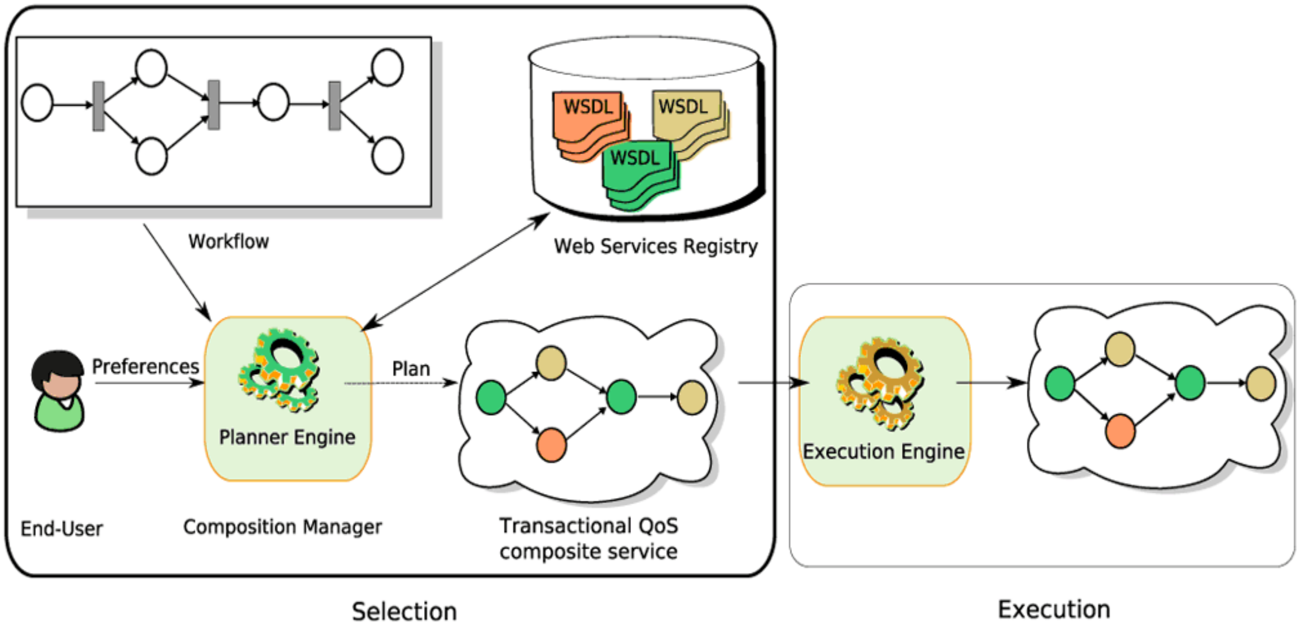

Fig 2: Transactional QOS Composite Service

!ar are riskier than c and cr. Indeed, properties! A and !ar mean that once a service has been executed, it cannot be rolled back.Therefore, to define two notions of execution risk in a transactional system like:

- Risk 0: the system guarantees that if the execution is successful, the obtained results can be compensated by the user.

- Risk 1: the system does not guarantee the successful execution and even if the execution is successful, the system does not guarantee the result can be compensated by the user.

The risk level 0 and 1 as defined above. However, other levels of risk could be defined in terms of compensation of the different components of the CWS. For example, the system can guarantee that if the execution is successful, some results can be compensated by the user or some results cannot be compensated by the user. In this case, the transactional properties must be relaxed.

A TQoS-driven approach is applied here. It consists of a Web service selection approach supporting transactional and quality driven WS composition. The selection of the component WSs is done by matching the WSs properties with the user's desires. More precisely, the selection is realized depending on transactional and QoS user requirements. The former is established by means of a risk notion that indicates if the results can be compensated or not. The latter is expressed as a weight over each QoS criterion. Our contribution is twofold. On the one hand, the composition manager selects WSs according to the quality and transactional behavior of the application, leaving to the user to focus on the construction of the desired functionalities. On the other hand, the proposed and formally analyzed a selection algorithm based on the workflow patterns and the transactional properties of the component WSs (elementary or composite). Five QoS criteria (execution price, execution duration, reputation, successful execution rate and availability) have been used and a local QoS-driven service selection related to these criteria has been chosen. However, other properties could have been taken into account, such as performance-related ones (e.g. machine resources), as done for example in. The local QoS aware selection is based on MCDM approach, however any other approach could have been used. The only constraint is to use a local optimization process in order to choose the WS having the best QoS among a set of potential WSs resulting from the transaction-aware selection process. In the experimentation, in order to give a semantic meaning to the risk notion, have considered two scenarios where the execution duration and execution price of a WS depend on additional operations required to guaranty their transactional properties. The risk notion to be used these scenarios. Under these conditions the implementation shows that the QoS of TCWS is in conformity with the user preferences. If the execution price criterion is important to the user (i.e. price minimum), then the better solutions are the ones with the lowest level of risk.

A contrario, if the execution duration criterion is more important to the user (i.e. execution time minimum), then the riskier solutions are the best ones. The results also show that risk 0 is equivalent to risk 1 if compensable services do not cost more than the others. Moreover, the evaluated the scalability of our TQoS algorithm. In fact, the experimental results, of two other scenarios, show that the number of activities does not affect the computation cost of the algorithm since the selection is done incrementally and therefore, the sets of candidate services for beginning activities is much bigger than the ones for ending activities. More experiments are needed to consider different scenarios and compare the performance of our algorithm with related ones.

4. Enchanced Transcation Quality of Services

4.1 Presentation of single service:

The service providers will advertise their atomic services at a global market place. There are several languages available for advertising, for example, UDDI or DAML-S Service Profile. The essential attributes to describe a Web service include the signature, states and the non-functional values. The signature is represented by the service's inputs, outputs and exceptions. It provides information about the data transformation during the execution of a Web service. The states are specified by precondition and post condition. Model it as the trans-formation from one set of states to another in the world. Non-functionality values are those attributes that are used for evaluating the services, such as the cost, service quality and security issues.

4.2 Translation of the languages:

The most service composition systems distinguish between the external and internal service specification languages. The external languages are used by the service users to enhanceaccessibility of the users in the sense that the users can express what they can offer or what they want in a relatively easy manner. They are usually different from the internal ones that are used by the composition process generator, because the process generator requires more formal and precise languages, for example, the logical programming languages. So far, the users have already get used to the standard Web service languages, such as WSDL and DAML-S. Thus the translation components between the standard Web service languages and the internal languages have to be developed.

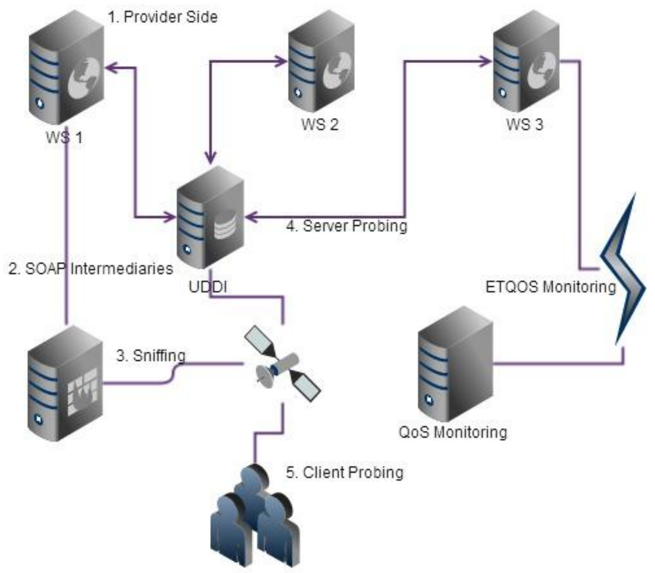

Fig 3: Overview of ETQoS Monitoring

4.3 Generation of composition process model:

In the meantime, the service requester can also express the requirement in a service specification language. A process generator then tries to solve the requirement by composing the atomic services advertised by the service providers. The process generator usually takes the functionalities of services as input, and outputs process model that describes the composite service. The process model contains a set of selected atomic services and the control flow and data flow among these services.

4.4 Evaluation of composite service:

It is quite common that many services have the same or similar functionalities. So it is possible that the planer generates more than one composite service fulfilling the requirement. In that case, the composite services are evaluated by their overall utilities using the information provided from the non-functional attributes. The most commonly used method is utility functions. The requester should specify weights to each non-functionality attributes and the best composite service is the one who is ranked on top.

4.5 Execution of composite service:

After a unique composite process is selected, the composite service is ready to be executed. Execution of a composite Web service can be thought as a sequence of message passing according to the process model. The dataflow of the composite service is defined as the actions that the output data of a former executed service transfers to the input of a later executed atomic service.

5. Conclusion

The QoS Ontology using ETQOS has been used to find the service to meet the QoS needs of users. This method can improve the effectiveness and accuracy of Web service discovery and it can return the best matched services to meet user's both functional and QoS needs. It can be used to support the automatic discovery of web services with QoS information. QoS ontology for Web services which is extensible and flexible for different domains and applications. ETQOS when using with QoS Ontology gives the accurate transaction information and it increases the scalability of QOS service of complex web services. The path probability information has also been got for the relevant QoS. The QoS metrics and values are designed in a way that can support transforming related QoS metrics between each other or computing a QoS metric from others. Various QoS attributes like tendency, mandatory, weighting, dynamism, relationship, grouping, etc. are integrated in the ontology in order to reason about QoS characteristics and facilitate efficient computation of QoS rank for Web services.

References

[1] R.Buvanesvari, V.Prasath, H.Sanofar Nisha, "A review of Fuzzy Based QoS Web Service Discovery," International Journal: Advanced Networking and Applications Volume: 04 Issue: 05 Pages: 1752-1759 (2013).

[2] Atif Alamri, Mohamad Eid and Abdulmotaleb El Saddik, "Classification of the state-of-the-art dynamic web services composition techniques," International Journal: Web and Grid Services, Vol. 2, No. 2, 2006.

[3] Ralph W. Feenstra, Marijn Janssen, and René W. Wagenaar,"Evaluating Web Service Composition Methods: the Need for Including Multi-Actor Elements," Faculty of Technology, Delft University of Technology, the Netherlands the Electronic Journal of e-Government Volume 5 Issue 2, pp. 153 – 164.

[4] Ajay Kattepur, Nikolaos Georgantas & Val´erie Issarny, "QoS Composition and Analysis in Reconfigurable Web Services Choreographies," ICWS 2013: 235-242.

[5] Yutu Liu, Anne H.H. Ngu, Liangzhao Zeng, "QoS Computation and Policing in Dynamic Web Service Selection," WWW2004, May 17–22, 2004, New York, New York, USA.ACM 1-58113-912-8/04/0005.

[6] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, "Quality Driven Web Services Composition,"May 20–24, 2003, Budapest, Hungary.ACM 1581136803/03/0005.

[7] Tang Yong, Zhang Zan-Bo, "Research in Enterprise Applications of Dynamic Web Service Composition Methods and Models," Electronic Commerce and Security, 2009. ISECS '09. Second International Symposium on (Volume: 1) 22-24 May 2009.

[8] Ching-She Wu, Wei-Chun Chang and Ibrahim Khoury,"Trade-Off Analysis on QoS-Aware Dynamic Web Services Composition with Evolutionary Optimization," Advances in Information Sciences & Service Sciences;Apr2012, Vol. 4 Issue 6, p9 April 2012.

[9] Joyce El Haddad, Maude Manouvrier, and Marta Rukoz,"TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition" Services Computing, IEEE Transactions on (Volume: 3, Issue: 1) Jan.-March 2010.

[10] Krithika V, Dr. Arshinder Kaur and Dr. K. Chandra Sekaran,"WEB SERVICES SUPPLY CHAINS: A LITERATURE REVIEW," International Journal of Web Services Computing, March 2012, Volume3, Issue 1, ISSN: 0976 – 9811

Author Profile

S.Karthikeyan received the B.TECH degree in Information Technology from Sri Ramakrishna Engineering College, Affiliated to Anna University, Chennai, in 2010.He is working towards the M.E degree in Computer Science and Engineering from Gnanamani College of Engineering, Affiliated to Anna University, Chennai since September 2012. His research interests include Web Services, Business Process Management, Web services composition, autonomic computing and Service-oriented computing.

K.J.JAGDISH received the M.E degree in Computer Science and Engineering from PGP COLLEGE OF ENGINEERING AND TECHNOLOGY, Affiliated to Anna University,Chennai. Received B.E degree in theComputer Science and Engineering from Mahendra Engineering College, Affiliated to Anna University,Chennai in 2008. Now working as Assistant Professor in Gnanamani College of Engineering,Affiliated to Anna University, Chennai Since June 2012. His research interest includes Cloud computing, Networking. He is a member of the ISTE.