ISSN: 2320-1363

Preserving Digital Artifact Using Uri Fetch

K. Bhavani[*1], **M. Senbagavali**[*2], **Sakthivel**[*3]

Student, Master of Engineering, Dept of CS, PSV College of Engineering, Krishnagiri, Tamilnadu. India

Professor and Guide, Master of Engineering, Dept of CS, PSV College of Engineering, Krishnagiri, Tamilnadu, *India*

Professor and HOD, Master of Engineering, Dept of CS, PSV College of Engineering, Krishnagiri, Tamilnadu, India

Abstract –

The current web systems don't have the mechanism to preserve the content of the site and the URL being used were not verifiable and reliable. In this project, the concept of URL Fetch is used to verify the trustworthy of the URI and the reliability is assured to the users who access the sites. Generally WSC is a process of gathering a set of single web services together to create a more complex, value-added and cross-organizational business process. It needs the self-automated software program that is capable of automatically selecting, integrating and invoking the several web services to obtain and achieve the user defined goal.

I.INTRODUCTION

The Web Services are a set of programs which are non-dependent, modular and self-describing. It can be used by any domains usually the cross domains in order to access the methods that is present in web service. Web services have proven to be very useful among the Service level architectures. Much software companies outsource their service through the medium of Web services. It is widely used in sharing the services via web. That services may include the potential functionality that serves the best component to the other software's which are accessing this service. On the recent days, as the number of web services increase on its count, the need of effectiveness and efficiency towards the web services increases. So the component called Web services composition is introduced to group the web services in to a cluster. Generally WSC is a process of gathering a set of single web services together to create a more complex, value-added and cross-organizational business process.

It needs the self-automated software program that is capable of automatically selecting, integrating and invoking the several web services to obtain and achieve the user defined goal. A service which provides the same functionality, quality of service has been mostly applied to represent their non functional properties and differentiate them for service composition. Quality of service is a broad concept that encompasses a group of non functional properties, such as execution price, execution duration, availability, execution success rate, and reputation. Given a set of multiple global Qualities of service constraints and user preferences, the challenge is how to efficiently construct a composite service such that its overall Quality of service is optimal, while all the Quality of service constraints are satisfied.

II. LITERATURE SURVEY

2.1 [QoS Aggregation in Web Service Compositions] Michael C. Jaeger, GregorRojec-Goldmann, and GeroM¨uhl

For the composition of Web services non-functional characteristics are commonly considered criteria for finding and selecting available services. Our work focuses on a mechanism that determines the overall Quality-of-Service (QoS) of a composition by aggregating the QoS of the individual services. With aggregated QoS it can be verified whether a set of services satisfies the QoS requirements for the whole composition or not. The aggregation performed builds upon abstract composition patterns, which model basic structural elements of a composition like parallel paths, a sequence, or a looped execution. In this work we extend existing composition patterns with the abilities to consider dependencies between services. Furthermore we introduce how to use the pattern-based aggregation in the monitoring process during run-time. We will explain how the data derived from the monitoring process can be used to calculate a more accurate aggregation of QoS for the composition.

2.2 [TQoS: Transactional and QoS-aware selection algorithm for automatic Web service composition] Joyce El Haddad, Maude Manouvrier, and Marta Rukoz

Web Services are the most famous implementation of service oriented architectures that has brought some challenging research issues. One of these is the composition, i.e. the capability to recursively construct a composite Web service as a workflow of other existing Web services, which are developed by different organizations and offer diverse functionalities (e.g. ticket purchase, payment), transactional properties (e.g. compensatable or not) and Quality of Service (QoS) values (e.g. execution price, success rate). The selection of a Web service, for each activity of the workflow, meeting the user's requirements, is still an important challenge. Indeed, the selection of one Web service among a set of them that fulfill some functionalities is a critical task, generally depending on a combined evaluation of QoS. However, the conventional QoS-aware composition approaches do not consider the transactional constraints during the composition process.

2.3 [Preference-Based Web Service Composition: A Middle Ground between Execution and Search]

ShirinSohrabi and Sheila A. McIlraith

Much of the research on automated Web Service Composition (WSC) relates it to an AI planning task, where the composition is primarily done offline prior to execution. Recent research on WSC has argued convincingly for the importance of optimizing quality of service, trust, and user preferences. While some of this optimization can be done offline, many interesting and useful optimizations are data-dependent, and must be done following execution of at least some information gathering services. In this paper, we examine this class of WSC problems, attempting to balance the trade-off between offline composition and online information gathering with a view to producing high-quality compositions efficiently and without excessive data gathering. Our investigation is performed in the context of the semantic web employing an existing preference-based Hierarchical Task Network WSC system. Our experiments illustrate the potential improvement in both the quality and speed of composition generation afforded by our approach.

ISSN: 2320-1363

ARCHITECTURE DIAGRAM

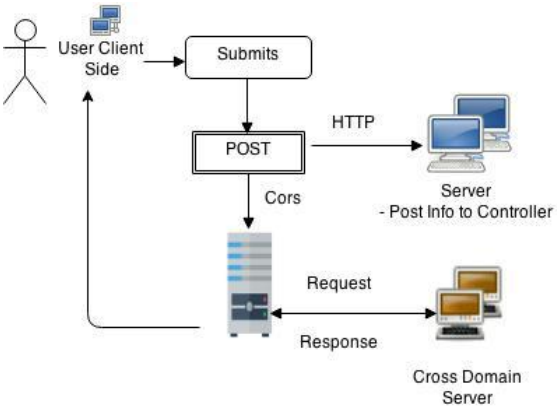

Fig 1: Architecture Diagram

III EXISTING SYSTEM

3.1 Existing System

The trusty URIs can be used for the verification of digital artifacts, in a manner that is independent of the serialization format in the case of structured data files such as nanopublications. Web services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Here, Web service composition uses numerical temporal planning method to enhance Quality of Service but their optimal QoS cannot defined. This increases computational cost and time for a composition task. Also it degrades the efficiency, reliability of QoS. Hence planning method cannot provide non-functional properties accurately when selecting the web services and globally optimized QoS value.

3.2 Disadvantages of Existing System

- For large instances, this process does not solve easily to obtain the QoS value.
- By using Local optimization and Integer programming, the global QoS constraints cannot achieved.
- Limited number of web services call only handle.
- It increases Computational cost and time for a composition task.
- So, it causes to degrades the efficiency, reliability of QoS when user selecting the web services.

IV PROPOSED SYSTEM

4.1 Proposed System

- The current system works on the basis of individual URI's where the time taken is high and verifialbity is likely to consistent.
- Hence the concept of URLFetch is used in the proposed approach. It is capable of processing bunch of URL's at a single time.

- URLFetch works on both GET and POST Method

Web Service Composition is the task of combining a set of single web services together to create more complex services. The proposed system uses CORS method where the browser and the server are interact to determine whether or not to allow the cross-origin request. Also, planning based approach method is enhanced which uses HTTP Redirect concept to make a cross domain call to the third party service. Additionally, exact Destination and http Response Status attributes allow you to configure the end-user experience of the redirection. Using this prototype, Open Authentication protocol 2.0 is used to make the cross domain call and once the request is successful, Channel API is used. In this project, channel API based multiuser chat is implemented to show the efficiency of web service call made on every chat which is exhibited using many different channels. This will update the information to users quickly. To minimize the computational time, an approximation algorithm is used and the composite service graph finds the optimal overall QoS value globally.

4.2 Advantages of Proposed System

- When the number of Web Services is increased, it is easy to obtain the services.
- The process of workflow model obtains the QoS value based on workflow time, workflow cost and the degree of workflow reliability.
- Using HTTP Redirect, the services are easily reached to the target space and obtain the fast data retrieval data from the server.
- Computational cost and time for a composition task is decreased.

It obtains the overall QoS constraints. Thereby the efficiency of QoS is increased.

V. PROBLEM DEFINITION

The use of web services has dominated software industry. Existing technologies of web services are extended to give value added customized services to customers through composition. Automated web service composition is a very challenging task. This paper proposed the solution of existing problems and proposed a technique by combination of interface based and functionality based rules. The proposed framework also solves the issues related to unavailability of updated information and inaccessibility of web services from repository / databases due to any fault/failure. It provides updated information problem by adding aging factor in repository/WSDB (Web Services Database) and inaccessibility is solved by replication of WSDB. We discussed data distribution techniques and proposed our framework by using one of these strategies by considering quality of service issues. Finally, our algorithm eliminates the dynamic service composition and execution issues, supports web service composition considering QoS (Quality of Service), efficient data retrieval and updating, fast service distribution and fault tolerance. Keywords---composition of services; dynamic composition; UDDI registry; web services.

VI. MODULE DESCRIPTION

- Open Authentication
- Web Service Identification
- Functionality Request

6.1 Open Authentication

In general, this module deals with authenticating the user inside the application, this is considered to be the efficient and secured method in order to allow the user to be authenticated inside the application.

An open authentication protocol is an open standard to authorization.OAuth provides client applications a 'secure delegated access' to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner, or end-user.

The client then uses the access token to access the protected resources hosted by the resource server. OAuth is commonly used as a way for web surfers to log into third party web sites using their Google, Facebook or Twitter accounts, without worrying about their access credentials being compromised. OAuth is a service that is complementary to, and therefore distinct from, Open ID. OAuth is also distinct from OATH, which is reference architecture for authentication, not a standard.

6.3 Web Service Identification

In this module, the web services based on the functionalities required are generated.

In general, each web services are intended to perform a separate operation. A web service repository is a set of disjoint services. We denote it as {w1, w2,}

6.2 Functionality Request

As the growth of web related requirements increases gradually, the users who interact with the web needs the system to system interaction, created the need for a structure which provided interaction not only with the user but also help application to application interaction.

This problem has a common name of Application Integration. It can be resolved using Web Services. Moreover, with the addition of the intelligence and autonomy of software agents, transactions may be equally automated for consumer-to-consumer, business-to-consumer, and business-to-business collaborations. But it is a clear observation that the world is a global village and the numbers of services on the web are increasing day by day. Because of this increasing number, a problem is raised to find an appropriate web service which satisfies user needs. The user wants an efficient way to find an appropriate web service which satisfied his needs in a short time.

CONCLUSION

The research lies in the field of dynamic web services composition selection. In this paper we discussed the main problems faced by dynamic web services composition. This paper proposed the dynamic web services composition algorithm to solve the composition issues related to data distribution, reliability, availability and QoS. It presented a framework in which multiple repositories and WSDBs have been introduced in order to make system more reliable and ensure data availability. By using multiple registries, data availability is guaranteed whereas by using aging factor users can retrieve up to date information. The proposed system is fault tolerant, reliable, performs fast data retrieval and Quality of service based. In future, the framework can be extended by crawling the web for searching web services instead of querying the UDDI registries. We will also be looking into deeper details of every component of the framework to ensure better and efficient composition.

REFERENCES

[1] Java, The Complete Reference, 8th Edition, by Herbert Schildt, Comprehensive Coverage of the Java Language.

[2]Professional JavaScript for Web Developers, by Nicholas C. Zakas, Third Edition.

[3]JavaScript and HTML5 Now, A New Paradigm for the Open Web, by O'Reilly, Kyle Simpson.

[4]http://www.kpmg.com/Global/en/IssuesAndInsights/ArticlesPublications/Documents/cloud-clarity.pdf (Accessed 2nd November 2014) **[5]**Mell, P., and Grance, T., 'The NIST definition of cloud computing', 2011

[6]Gartner: 'Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 2Q12 Update', in Editor (Ed.)^(Eds.): 'Book Forecast Overview: Public Cloud Services, Worldwide, 2011-2016, 2Q12 Update' (2012, edn.), pp. 18-19 .

[7]http://ec.europa.eu/digital-agenda/events/cf/e2wp2014/document.cfm?doc_id=23902. (Accessed 18th November 2014)

[8]Gracia, J., and Mena, E., 'Semantic heterogeneity issues on the web', Internet Computing, IEEE, 2012, 16, (5), pp. 60-67

[9]Kobayashi, M., and Takeda, K., 'Information retrieval on the web', ACM Computing Surveys (CSUR), 2000, 32, (2), pp. 144-173

[10]Nagireddi, V.S.K., and Mishra, S., 'An ontology based cloud service generic search engine', in Editor (Eds.): 'Book An ontology based cloud service generic search engine' (IEEE, 2013, edn.), pp. 335-340 .

[11]Afify, Y.M., Moawad, I.F., Badr, N.L., and Tolba, M.F., 'A semantic-based Software-as-a-Service (SaaS) discovery and selection system', in Editor (Eds.): 'Book A semantic-based Software-as-a-Service (SaaS) discovery and selection system' (2013, edn.), pp. 57-63 **[12]**Noor, T.H., Sheng, Q.Z., Alfazi, A., Ngu, A.H.H., and Law, J., 'CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web', in Editor (Eds.): 'Book CSCE: A Crawler Engine for Cloud Services Discovery on the World Wide Web' (2013, edn.), pp. 443-450

[13]Al-Masri, E., and Mahmoud, Q.H., 'Investigating web services on the world wide web', in Editor (Eds.): 'Book Investigating web services on the world wide web' (ACM, 2008, edn.), pp. 795-804

[14]Dong, H., Hussain, F.K., and Chang, E., 'Ontology-learning-based focused crawling for online service advertising information discovery and classification': 'Service-Oriented Computing' (Springer, 2012), pp. 591-598