EFFICIENT SELECTION OF WEB SERVICE BASED ON RANKING USING QOS NON FUNCTIONAL REQUIREMENTS

Suchithra M. [1], Ramakrishnan M [2]

[1]Research Scholar, Department of Computer Science & Engineering, Sathyabama University, Chennai, India
[2]Chairperson, School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India
Email: [1]suchij24@gmail.com

Abstract —

Web service discovery process locates appropriate web services as per user preference from web. Since abundant numbers of web services are supplied for a single request, it becomes essential to rank them so that most relevant services can be supplied to the requester. In order to rank web services, various features are used and non-functional properties of a web service plays a vital role in this ranking process. In this paper, simple web services ranking algorithm is proposed which utilizes non-functional requirements. First web service filtering process is applied on selected web service set to remove unwanted and redundant web services. Then ranking process is applied on remaining web services and web service with highest score is supplied to the requester. Experimental results show that the proposed method outperforms the traditional methods of web service discovery process.

Key words: Service Discovery, ranking, QoS, Non-Functional Requirements, UDDI, SOAP, WSDL

I. INTRODUCTION

Web services are essential XML based components that can communicate with each other on different machines in internet irrespective of operating system, languages used and architecture settings. Web services can be created, published, advertised, located and accessed in internet using HTTP protocol[1]. The basic standards of web services include Simple Object Access Protocol (SOAP, Web Service Definition Language (WSDL), Universal Description, Discovery and Integration (UDDI).

Web services are collectively placed in the architecture called Service Oriented Architectures (SOA). The collection of web services communicate with each other, starting from simple data request / reply to complex operations[2]. It is also possible to integrate more than one web service to provide complex services. Examples of SOA are DCOM, CORBA, etc. Web services use XML to identify the data, SOAP to transfer message and WSDL to define the availability of services[2].

Web service discovery process locates service providers, retrieve web service descriptions and supply it to the requesters. Web service discovery is purely dependent on the user request query that contains details like type of desired service, price, number of returned results and information published by service provider[3]. The discovery process is also dependent on the architecture of service registry. After discovery process, the requestor must know the location and details of requested web service, and also methods to interact with them. Web service discovery can be static in which service implementation details are bound at design time or dynamic in which service implementation details are left unbound during design time[4].

When a request is invoked, a search is made in UDDI and web services that satisfy the user request are retrieved. Since the number of web services available is growing rapidly, it is difficult for the requestor to select most appropriate web service, manage web service compositions and understand requirements[5]. This leads to the task of finding best web services among available ones with the same functionalities. The functionalities of retrieved web services may be same but they are different in non-functional properties like time, availability, security, throughput, reliability and execution cost. The best way to select a web service is based on their QoS properties that a web service meets[6]. It is the measure of how well a web service serves the customer. QoS is having an impact on web service and service with highest QoS value is selected first. This ranking is done only after functional matching of web services with user request is over. This paper proposes novel web service ranking algorithm which uses QoS non functional properties of a web service. Section 1 presents introduction to web service and basic components of a web service. Section 2 provides literature review of the web service discovery process. QoS based non functional requirements of a web service and web service filtering process is briefly discussed in section 3. Experimental results and discussion are presented in section 4. Paper ends by providing the finding of the experiment and views as conclusion in section 5.

II. RELATED WORK

Hyun Sik Huang et al [7] presented agent based delegation model for the secure web service in ubiquitous computing environment. XML based encryption and a digital signature mechanism is proposed to provide security in web services environment. Shuping Ran [8] proposed a model for web services discovery with QoS. The proposed model considers both functional and non-functional requirements. The new roles in the framework called certifiers were introduced which verifies QoS values of a web service.

D. A. Mello and V. S. Ananthanarayana [9] presented quality driven web service selection and ranking. Four web service provider qualities are defined to categorize between similar and qualitatively competitive web services. Zhengdong Gao and Genfeng Wu [10] combined QoS based web service selection with performance prediction. The dynamic nature of web service performance is considered for web service ranking. New attributes are added to QoS model so that it can reflect performance of service.

Huang Lican and Jianfeng Nie [11] proposed QoS based efficient web service selection using Pareto principle. Generally huge amount of computation are needed to select best web service from multiple candidate web services. Pareto principle is used to reduce this computation.

Vuong Xuan Tran et al[12] proposed QoS ontology based web services ranking algorithm. QoS ontology is used to rank web services that facilitate various service participants to offer their QoS. Analytic Hierarchy Process (AHP) is used to rank web services in a dynamic way.

III. NON-FUNCTIONAL QOS

Generally non-functional properties of web service include performance, reusability, maintainability, security, reliability, availability, etc. QoS properties are very important non-functional characteristics of a web service. In the below section, we briefly discuss various QoS parameters

- Response Time - Time needed to submit the request and getting the response
- Accessibility - The degree of a web service to serve a request
- Compliance - Specifies the relationship between WSDL document and WSDL specification
- Success Ratio-Number of request messages responded successfully
- Availability – Percentage of time a web service is available
- Security – Represents suitable security mechanisms for a given web service with three parameters encryption, authentication and access control

Among the different QoS parameters, response time, reliability, availability and Success Ratio are selected in our method as these parameters are having great impact in assisting requesters for reasonable selection[13]. Moreover, these parameters are said to be fundamental qualities of a web service that are needed to fulfil web service objectives.

A. Web Service Filtering

As ample number of web service exists in web, filtering process selects relevant web services to form candidate service set. Filters control and validate the request to web service and response from web service. Filtering process also removes redundant web services[14]. The below algorithm performs filtering. Let {S} represents the set of web services and {Sf} represents the candidate web services set

Procedure filtering(S, C, n)

//S={WS_1,WS_2, WS_3….WS_n} set of web services

//S_f={S_1,S_2,S_3, ….S_n} candidate web services set

//C={C1, C2, C3,C4} set of QoS properties for each service such as response time, reliability, availability and SuccessRatio

n=total number of candidates

for each candidate web service

for j = 1 to 4

if q_{ij} (S_i) < C_j then

remove service S_i

end for

end for

return S_f

Procedure ranking (S_f, q)

// q_{max} is the maximum QoS value for web service candidate set

// q_{min} is the minimum QoS value for web service candidate set

// q is the QoS value for the parameter

$$q_p = \frac{(q - q_{min})}{(q_{max} - q_{min})} \qquad if\ qmax! = qmin$$

$$q_n = \frac{(q_{max} - q)}{(q_{max} - q_{min})} \qquad if\ qmax = qmin$$

$$q = \frac{(q - y)}{(q_{max} - y)} \qquad if\ qmax! = y$$

$$q' = \frac{(q_{max} - q_c)}{(q_{max} - y)} \qquad if\ qmax = y$$

return (qp, qn, q, q')

When a requester submits a query, web service discovery agent select services which meets requester's functional requirements. The selected web services form the candidate service sets. This set becomes the input for our proposed algorithm.

IV. RESULTS AND DISCUSSION

Five requesters were taken, with each of them having same functional requirements and different QoS constraints. The services returned by the discovery agents that satisfies user functional requirements are shown in the below table. The table shows various web services and their corresponding QoS values

Table 1 web services and QoS values

Web Service	Response Time	Availability	SuccessRatio	Reliability
WS_1	128.60	93	85	96
WS_2	144.25	92	93	45
WS_3	102.76	85	68	60
WS_4	553.36	68	72	73
WS_5	509.48	99	48	82
WS_6	241.62	76	90	98
WS_7	133.38	98	93	42
WS_8	177.56	98	100	53
WS_9	184.93	100	88	68
WS_{10}	141.46	76	62	79
WS_{11}	156.91	87	78	86

The QoS constraints given by the first requester are response time=600, availability=60, success = 60 and reliability = 66. The web services WS1, WS9, WS4 and WS8 are selected. The QFRSA value for the selected web services for each QoS parameter is calculated as listed in the below table

Table 2 QFRSA value for each QoS parameter

Web Service	Response Time	Availability	SuccessRatio	Reliability	QoS Score
WS_1	0.0698	0.0396	0.0296	0.1806	0.3596
WS_9	0.0706	0.0589	0.0563	0.0423	0.2599
WS_4	0.0766	0.0432	0.0384	0.0428	0.1268
WS_8	0.0685	0.0568	0.0448	0.0339	0.0869

Since WS1 is having highest score, it is recommended to the service requester. The next requester needs web service with different QoS constraints such as response time=550, availability=85, success = 70 and reliability = 60. The third requester QoS Constraints are response time = 480, availability=85, SuccessRatio=80 and reliability=40. The web services selected for second request are WS3, WS7 and WS4. And for the third requester WS1, WS2, WS5, WS6 are selected. The QFRSA scores for each request is shown in the below table

Table 3 QFRSA scores

Web Service	Response Time	Availability	SuccessRatio	Reliability	QoS Score
WS_3	0.0661	0.1469	0.1375	0.0421	0.4121
WS_7	0.0459	0.0719	0.1187	0.0031	0.3249
WS_4	0.0652	0.1094	0.1281	0.0126	0.2406

Table 4 QFRSA scores

Web Service	Response Time	Availability	SuccessRatio	Reliability	QoS Score
WS_1	0.0650	0.1469	0.1469	0.04212	0.4769
WS_2	0.0669	0.1187	0.1375	0.0323	0.3911
WS_5	0.0437	0.0719	0.0000	0.0000	0.2626
WS_6	0.0537	0.00548	0.1187	0.0769	0.1369

The web service WS3 is selected for the second request and WS1 is selected for the third request. Note that Success Ratio and reliability scores of WS5 is 0.0000, meaning that particular web service is not meeting the minimum threshold levels prescribed by the requester.

V. CONCLUSION

In this paper, ranking of web services based on non functional requirements of a web service is presented. The proposed method uses the combination of filtering and ranking processes. First filtering process is carried out, which controls and validates user Request of web service, and responses from web service. Unwanted and redundant web services are removed from the web service set and new candidate service set is generated. By using the QoS parameters such as response time, accessibility, availability and SuccessRatio, QoS score for each web service in candidate service set is calculated and service with highest QoS score is supplied to the user. The results of the proposed method are compared with traditional WSSRM method. It shows that the proposed method is more accurate in selecting the appropriate web service than the traditional method.

REFERENCES

[1] Zeng, Liangzhao, et al. "Qos-aware middleware for web services composition."Software Engineering, IEEE Transactions on 30.5 (2004): 311-327.

[2] M. Ramakrishnan and M. Suchithra, 2013. A Review on Semantic Web Service Discovery Algorithms. International Journal of Soft Computing, 8: 313-320.

[3] Chakraborty, Dipanjan, et al. "Toward distributed service discovery in pervasive computing environments." Mobile Computing, IEEE Transactions on 5.2 (2006): 97-112.

[4] Maximilien, E. Michael, and Munindar P. Singh. "A framework and ontology for dynamic web services selection." Internet Computing, IEEE 8.5 (2004): 84-93.

[5] M.Suchithra and M.Ramakrishnan, "Enhanced Web Service Ranking Approach Based on Non-Functional QoS Criterion with User Preference", International Journal of Technology and Engineering System (IJTES) Vol 7. No.1 2015 Pp. 89-97

[6] Yu, Tao, Yue Zhang, and Kwei-Jay Lin. "Efficient algorithms for Web services selection with end-to-end QoS constraints." ACM Transactions on the Web (TWEB) 1.1 (2007): 6.

[7] Hyun Sik Hwang; Hyuk Jin Ko; Kyu Il Kim; Ung Mo Kim, "Agent-Based Delegation Model for the Secure Web Service in Ubiquitous Computing Environments," Hybrid Information Technology, 2006. ICHIT '06. International Conference on , vol.1, no., pp.51,57, 9-11 Nov. 2006

[8] Ran, Shuping. "A model for web services discovery with QoS." ACM Sigecom exchanges 4.1 (2003): 1-10.

[9] D'Mello, D.A, Ananthanarayana, V.S. "Quality Driven Web Service Selection and Ranking," Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on , vol., no., pp.1175,1176, 7-9 April 2008

[10] M.Suchithra and M.Ramakrishnan "Non Functional QoS Criterion Based Web Service Ranking" International Conference On Soft Computing Systems ICSCS2015 April 2015, India

[11] Huang, Lican; Jianfeng Nie, "Using Pareto Principle to Improve Efficiency for Selection of Qos Web Services," Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE , vol., no., pp.1,2, 9-12 Jan. 2010

[12] Vuong Xuan Tran, Hidekazu Tsuji, Ryosuke Masuda, A new QoS ontology and its QoS-based ranking algorithm for Web services, Simulation Modelling Practice and Theory, Volume 17, Issue 8, September 2009, Pages 1378-1398, ISSN 1569-190X

[13] Badr, Youakim et al "Enhancing web service selection by user preferences of non-functional features" Next Generation Web Services Practices, 2008. NWESP'08, 4th International Conference on. IEEE, 2008.

[14] Abdelzaher, Tarek F., Kang G. Shin, and Nina Bhatti. "Performance guarantees for web server end-systems: A control-theoretical approach." Parallel and Distributed Systems, IEEE Transactions on 13.1 (2002): 80-96.