

A QoS-Aware Web Service Selection Based on Clustering

R.Karthiban

PG scholar, Computer Science and Engineering, United Institute of Technology, Tamil Nadu, India

Abstract- Web service plays an important role in e-business and e-commerce applications. A web service applications are interoperable and can work on any platform, large scale distributed systems can be established easily. The selection of most suitable web service is very crucial for successful execution of applications due to the rapid growth of web service providers in the Internet. Researchers have proposed various techniques for service discovery like ranking and clustering the web services. These models lead to the selection and re-selection of improper web services due to the analysis of non-functional characteristics of web services. A novel technique to mine Web Service Description Language (WSDL) documents and cluster them into QoS similar Web service groups is proposed. The application of this approach to real Web service description files has shown good performance for clustering Web services based on QoS similarity, as a predecessor step to retrieving the relevant Web services for a user request by search engines. The various parameters of quality are trustworthiness, security, performance, availability, response time, scalability etc.

Index Terms- Web Services, QoS Parameter, Clustering, K-Means, Service Discovery.

- We introduce service clustering for grouping services with similar functionality into classes according to their QoS properties. The clustering result can make us obtain the service clustering information by which we know the characteristic of each atomic service. And this can help to speed up the algorithm while guaranteeing the near-optimal result by choosing the best services from each class. Besides, when an accident happens, it can assist to re-select atomic services according to the service clustering information which can be updated at run time.
- We demonstrate our proposed model's feasibility through some experiments, and analyze the experiment results with different parameters and in different situations.

The rest of this paper is organized as follows. Section II gives a brief background on related work. Section III describes the QoS parameters. Section IV introduces our proposed clustering approach. Section V explains the selection of web service from the cluster. Section VI discusses the experiments and results. Finally, Section VII concludes the paper and outlines future research avenues.

I. INTRODUCTION

The building block of web services architecture [1] are service provider, service registry and service users. The interaction involves publish, find and bind operations. Collectively, these roles and operations act upon the Web Service artifacts such as the Web service software module and its description. In a usual scenario, a service provider hosts a network-accessible software module. The service provider defines a service description and publishes it to a service registry. The service user queries the service registry for service selection and uses the service description to bind with the service provider. However, as the number of atomic services developed by different service providers grows rapidly, it is essential to select appropriate atomic services for complex business processes. However, as the number of atomic services developed by different service providers grows rapidly, it is essential to select appropriate atomic services for complex business processes.

In order to achieve better service selection the services are clustered based on the QoS values. We used K-means clustering algorithm to cluster the QoS values. So, we can able to differentiate the cluster values using mean values of the cluster. Our algorithm is used when service requestor needs to find an optimal choice of services for the purpose of satisfying the service requester's requirements, and it performs well in highly competitive environments. The main contributions of this paper are listed as follows:

II. BACKGROUND AND RELATED WORK

Research in Web mining has recently gained much attention due to the popularity of Web services and the potential benefits that can be achieved from mining Web services description files. Non-semantic Web services are described by WSDL documents while semantic Web services use Web ontology languages (OWL-S) [2] or Web Service Modeling Ontology (WSMO) [3] as a description language. Non-semantic Web services are more popular and supported by both the industry and development tools. The discovery process is quite different according to the Web services description method. Semantic Web services are discovered by high level match-making approaches [4], whereas non-semantic Web services discovery uses information retrieval techniques [5]. In our approach, we target the discovery of semantic Web services.

Yi Xia, Ping Chen, Liang Boa, Meng Wang and Jing Yang [6] proposed a method to improve the Web service discovery process using BPEL tree structure. He provides the special kind of algorithm called QSSAC that selects and reselects the proper Web Service for the user request. In contrast, our approach clusters Web Services based on their QoS values in order to reduce the search space and improve query matching. We make use of QoS values extracted from the description files to calculate the similarity among Web services. Many efforts have been made to overcome the drawbacks of UDDI-based discovery techniques. Khalid Elgazzar, Ahmed E.Hassan and Patrick

Martin [7] propose functionality based discovery mechanisms. Their proposed model follow the K- Means algorithm to cluster the Web Service which useful to bootstrap the Web Service discovery. The application of this approach to real Web service description files has shown good performance for clustering Web services based on function similarity, as a prior step to retrieving the relevant Web services for a user request by search engines. Rajini Mohana and Deepak Dahiya,. [8] propose an Algorithm for building a rule based model for ranking the web service based on quality of service (QoS) using fuzzy clustering and particle swarm optimization (POS). The number of quality attributes considered for the ranking but PSO reduces the number of rules by removing the rules that are having less weightage and will not affect the system.

Liu and Wong [9] use a proposal similar to ours and apply text mining techniques to extract features such as service content, context, host name, and name, from Web service description files

in order to cluster Web services. They propose an integrated feature mining and clustering approach for Web services as a predecessor to discovery, hoping to help in building a search engine to crawl and cluster non-semantic Web services. We differ in our choice of features. We believe that the service context and service host name features offer little help in the clustering process. Providers tend to advertise the services they provide on their own website, which means they provide different Web services on the same site. Hence, mining the surrounding Web pages (service context) or considering the host name does not help with the meaning of the Web service, which is not the case in UDDI. In addition, some Web services do not make use of the <documentation> element in the WSDL document, which means there is insufficient information for the content feature.

III. QoS PARAMETERS

Table-1: QoS Parameter and its units

S.No	QoS Attribute Name	Description	Unit
1.	Response Time	Maximum or Average Time taken to send a request and receive a response.	ms
2.	Availability	Number of successful invocations/total invocations.	%
3.	Throughput	Total Number of invocations for a given period of time.	Invokes/ second
4.	Successability	Number of response / number of request messages.	%
5.	Reliability	Ratio of the number of error messages to total messages.	%
6.	Latency	Time taken for the server to process a given request.	ms

There are some possible QoS metrics such as Performance, dependability, cost and reputation which is relevant to the web services. The performance of a web service represents how fast a service request can be completed. Depending on the stakeholder interests, it can be set to deal with performance such as throughput, response time, execution time and resource utilization. Throughput is the number of completions of web service requests during an observation time interval. Response time is the period of time necessary to complete a web service request. Execution time is the resource time consumed by a web service to process its composing activities. Resource utilization is the percentage of time a resource is busy serving web service activities. The dependability of Web Services is a property that integrates several attributes like reliability, availability and security.

Reliability represents the ability of a Web Service to perform its required functions under stated conditions for a specified time interval, Availability is the probability that the system is up and security.

Web service can be provided against payment of a certain amount of money. Usually the price of a Web Services is defined by the provider. The cost may be permanent for each invocation or proportional to the actual service demand to each Web Service method that is used. Providers may also request higher fees for services hosted on better/faster hardware. Reputation reflects a common perception of other WS or customer towards that service. It aggregates the ratings of the given service by other principals. Typically, a reputation would be established from a history of ratings by various parties.

IV. PROPOSED CLUSTERING APPROACH

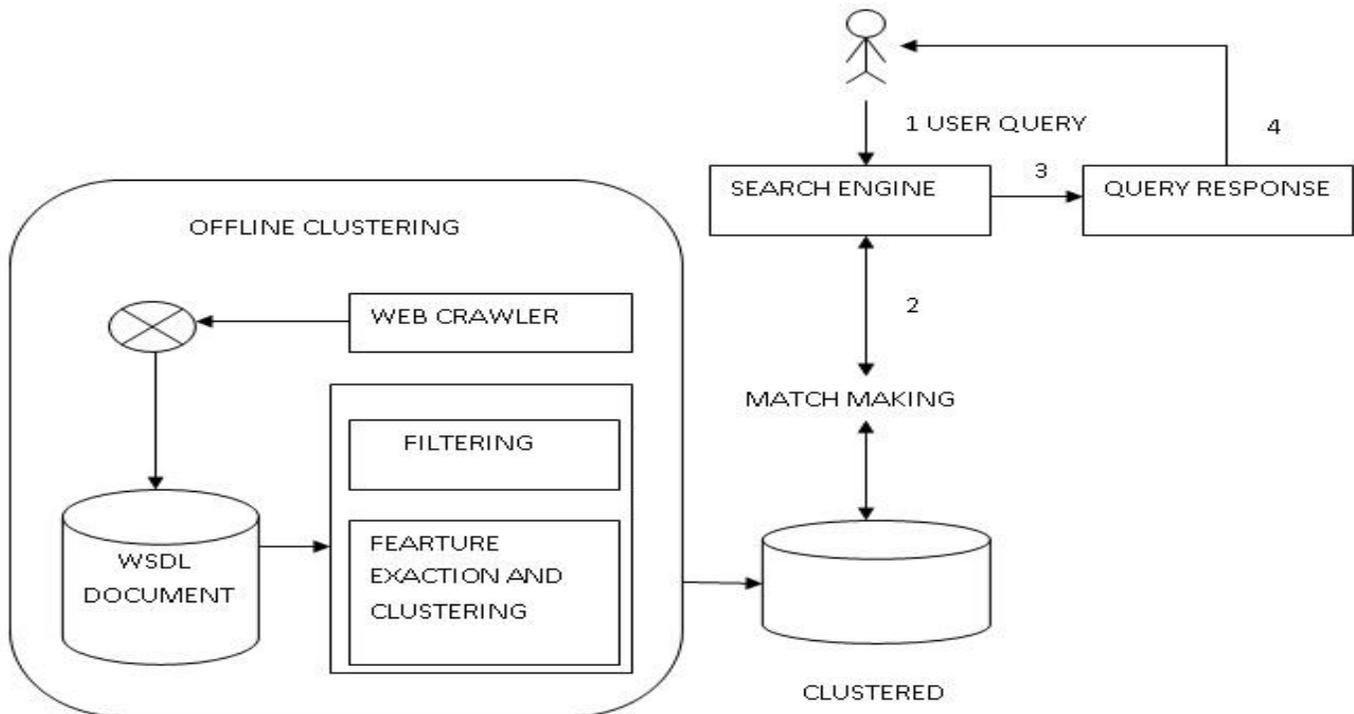


Fig-1: Architecture of proposed model

Clustering is based on QoS parameters which are requested by the user at the searching time. It significantly increases the performance of service discovery by analysing the best cluster value instead of analysing the huge amount of data. Our proposed model includes the four kinds of steps. They are,

- Filtering the web service based on keywords
- Extracting the QoS values from WSDL document
- Forming the cluster using the extracted QoS values
- Selecting the most suitable web service from the cluster.

Filtering is the very first step in web service clustering. It parses the WSDL document which contains a web service name and other QoS properties for the specific web service and also it has a URL for the web service. Finally it returns the web services Name which is specified by the user.

After the filtering process, the web service name will be displayed. The user specified QoS values are retrieved from the WSDL document based on filtered web service names. The retrieved QoS values are clustered using K-Means algorithm. The K-means algorithm proceeds as follows. First, it randomly selects k of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster. This process iterates

until the criterion function converges. It forms the cluster using the following procedure

Step 1: Make initial guesses for the means m_1, m_2, \dots, m_k

Step 2: Until there are no changes in any mean

Step 2.1: Use the estimated means to classify the samples into clusters

Step 2.2: For i from 1 to k

Replace m_i with the mean of all of the samples for cluster i

Step 2.3: end_for

Step 3: end_until

The clustered web Services are stored in the intermediate storage area. Which is contains the centroid as mean values of the cluster.

V. WEB SERVICES SELECTION

The ResponseTime QoS attributes comes under the minimization type and the remaining attributes falls under maximization type. If the user request the web service based on ResponseTime, the selection algorithm returns minimum value's URL from the cluster which has minimum centroid value. For the remaining QoS attributes request, the selection algorithm returns maximum value's URL from the cluster which has the maximum centroid value.

During the Web Service selection the selection algorithm checks for the first request. If the request is fresh then it from the cluster end stores in the intermediate storage area or otherwise the selection algorithm directly goes to the intermediate storage

area and analyse the already stored clusters and returns the proper Web Service for the user request and QoS parameter.

5.1 Merits of Proposed System

- Returns the correct web service based on requesting QoS attribute

- Performance of Web service selection will increase because of intermediate storage area between client and server.
- Updatons and modifications in the web service will not affect the performance of web service.

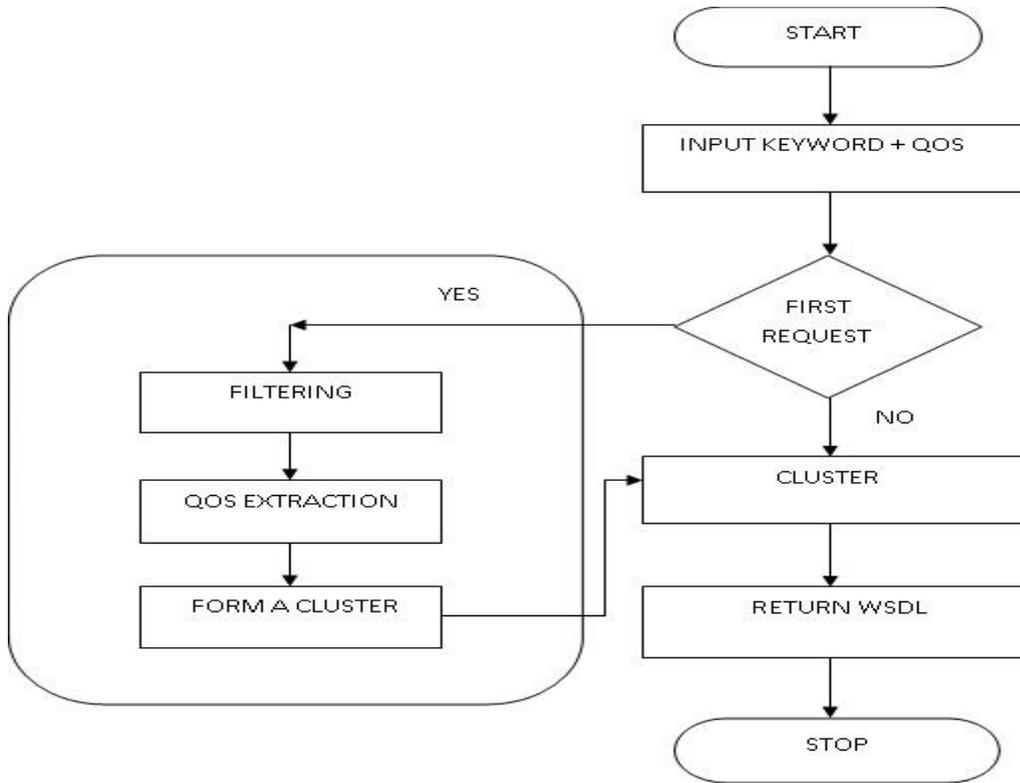


Fig-2: Data flow diagram for proposed model

VI. EXPERIMENTS AND RESULTS

This chapter focuses on the demonstration of the proposed model with single QoS constraint. The demonstration gives the best service from the huge volume of dataset based on QoS parameter. The QoS parameter is specified by the user at

searching time. The input dataset for the proposed model contains 2500 records.

Here, the specified QoS parameters are Availability, Throughput, Successability, Reliability, Latency and ResponseTime. At the same time number of cluster value is also specified by the user.

Table-2: Input for the Proposed System

Keywords	QoS Parameters	Number of Clusters
Holiday	Availability	Vary depending on the User
User	Throughput	
Web	Successability	
Google	Reliability	
Address	Latency	

Phone	ResponseTime	
Tax		
Wizard		
SMS		
Hotel		

The system returns the Web services and the specified QoS value for the request based on the keyword mentioned by the user. For example user wants to access the hotel web service with Quality of Service as Successability, it returns the output as

Table-3: Web Services Filtering

Service Name	Successability
K4THotelAvailWS	96
HotelsService	65
K4THotelSellWS	97
HotelFunctionsService	78

Based on the number of clusters and Web Service filtered outputs the system forms the cluster. For example, if the user enters a number of clusters equal to 3 then it forms a cluster in the following order

Table-4: Cluster Values

Cluster 1	Cluster 2	Cluster 3
65	78	96,97

Finally it returns the best cluster value, and the cluster value stored in the intermediate storage area.

96
97

It displays the URL by analysing the best QoS value from best cluster which stored in the intermediate storage area.

VII. CONCLUSION

Success of published web services depends on how it is getting discovered. Efficiency and accuracy factors must be considered while providing a discovery mechanism. Traditional UDDI based and search engine-based Web service discovery lacks the ability to recognize the content of the Web service description file. In this proposed model, filtering and QoS-based clustering is used to select the best web service for the customers. The procedure allows users to specify their QoS constraints which are used to filter off irrelevant services and takes the total

QoS utility score for each filtered candidate and it forms the cluster for relevant web services. From the cluster, the user assists to select the best web service in response to their specified preferences. Based on this project work, it can be concluded that, the proposed methodology provides a solution for dynamic web service selection at run time. The proposed model depicts that user will specify the QoS constraint and required specification to discover the best optimal service. Also, proposed model produce better quality result in comparison to the existing methods. The future scope of this work is to include more number of QoS parameters.

REFERENCES

- [1] Heather Kreger, May 2001, "Web Services conceptual architecture", IBM software Group, web service overview, pages 1-40.
- [2] M. Burstein, J. Hobbs, O. Lassila, D. Mcdermott, S. Mcilraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, K. Sycara, D. Martin (ed.) , 2004, "OWL-S: Semantic Markup for Web Services," W3C Member Submission.
- [3] H. Lausen, A. Polleres, 2005, "Web Service Modeling Ontology (WSMO)," W3C Member Submission.
- [4] Matthias Klusch, Benedikt Fries, Katia Sycara, 2006, "Automated semantic web service discovery with owlsmx," Proceedings of 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS).
- [5] John D. Garofalakis, Yannis Panagis, Evangelos Sakkopoulos and Athanasios K. Tsakalidis, September 2006, "Contemporary Web Service Discovery Mechanisms," Journal of Web Engineering, Vol. 5, No. 3, pp. 265-290.
- [6] Yi Xia, Ping Chen, Liang Boa, Meng Wang and Jing Yang, 2011, "A QoS-Aware Web Service Selection Algorithm Based On Clustering" in IEEE International Conference on Web Services, pp.428-435.
- [7] Khalid Elgazzar, Ahmed E.Hassan and Patrick Martin, 2010, "Clustering WSDL Documents to Bootstrap the Discovery of Web Services" in IEEE International Conference on Web Services, pp.147-154.
- [8] Rajini Mohana and Deepak Dahiya, 2011, "Optimized Service Discovery using QoS based Ranking: A Fuzzy Clustering and Particle Swarm Optimization Approach" in 35th IEEE Annual Computer Software and Application Conference Workshops, pp. 452-457.
- [9] Wei Liu, Wilson Wong, 2009, "Web service clustering using text mining techniques," International Journal of Agent- Oriented Software Engineering, Vol. 3, No. 1, pp. 6-26.

AUTHORS

First Author – R.Karthiban, PG scholar, Computer Science and Engineering, United Institute of Technology, Tamil Nadu, India, Email: karthibanlink@gmail.com