# DETECTING QUALITY OF WEB SERVICES (DQS)

[1] Viswesvar. S, [2] Balamurugan. V.H

*[1,2] Department of Information Technology, Panimalar Engineering College, Chennai (India)*

## ABSTRACT

*Non-functional characteristics of web services are described by Quality of Services. In current computing world, although Quality for web services can be determined, still there is lack of real world web services QoS data sets. In our project, we are going to integrate all available free web services in the internet, analyze quality of each web services and then provide the results of which web service is better for the consumers to make use of it .Quality of web service is evaluated based on Service evaluation technique containing an user-item matrix which includes various QoS properties like failure probability, response time, throughput, availability, cost, time aware performance, etc. We can compare different web services and provide the best possible web service to use it by providing ratings for those services and display these particulars visually as graphs*

***Keywords: Quality Of Service- Qos, Qos Data Set, Web Service***

## I. INTRODUCTION

In recent years, web services are emerging successfully and it is a renowned technique to build distributed systems. By dynamically composing different web services, we can build effective service-oriented systems. The Quality of service oriented systems is dependent on the employed web services quality. With availability of web services on the internet, determining quality of web services has become a major issue. Nonfunctional characteristics of web services are described by the Qos. As more web services arrives in internet, the Qos value has become a major point of differentiating various functionally equivalent web services. Properties in web service Qos includes failure probability, availability, price, response time, service ratings, throughput, popularity, type of browser supporting the service, and so on[1]. Server side Qos property values (price, popularity, etc) are provided by service providers and is same for various users. User observed Qos properties (response time, throughput, failure probability, etc) varies widely for different users, which is influenced by consumer’s internet connections and environment [1].

In service computing [2] field, many Qos based approaches have been engaged for web service recommendation [3], [4], [5], service composition [6], [7], Fault tolerant web services [8], [9], [10], web service Search [11], and so on. But still for Qos validation, there is a lack of comprehensive real time web services Qos data sets.

User observed Qos values can be obtained by evaluating from different locations using variety of different working environments. These values are provided by different companies and active users of those web services. However, it’s a tuff task for making large scale evaluation of Web services in different locations because

- Both service users and provider’s resources are consumed during web service invocation.

- Evaluating all the available web services is a time consuming and expensive task.
- It is a tedious process to collect web service Qos data from distributed service users.

But enough web service Qos values cannot be acquired without real world evaluations. In service computing, it is difficult to validate feasibility and effectiveness of various approaches in Qos.

To solve this challenge, we make effort to conduct distributed evaluations on real world web services, and released the data sets which are reusable for future enhancement. First, 72 web service addresses are obtained from the internet. Then three web service evaluations are conducted. In the initial evaluation, failure probability of available web services is assessed by limited distributed service users. In next evaluation, properties of available web services i.e., response time and throughput are evaluated with the available service users. In third evaluation, changing ratio of the web services with time is calculated by invoking those web services with available service users in various time slots with a time interval. Initial experiences in real world web service Qos are served in this project and reusable Qos are provided for future research. Extending from its previous journal version [*], which reports the above evaluations, this project includes

- Detailed analysis on relation between Qos values and time, and
- Data set applicability like Qos prediction, selection, search and fault- tolerant web services.

In this project, the remaining portion is as below: Section 2 introduces the information of web services of use. Section 3 gives our distributed QoS evaluations of
web services. Section 4 discusses the applications of the Web service QoS data sets. Section 5 introduces related work, and Section 6 concludes the project.

## II. WEB SERVICES INFORMATION

We can discover web services from Universal Description, Discovery, and Integration (UDDI), web service portals and web service search engines. UDDI is an XML based registry and it facilitates companies to publish and discover web services on the internet. We obtained 72 addresses of WSDL (Web Service Description Language) files by crawling information about web services from UDDI, web service search engines and web service portals. In our experiments we cover most of the free available web services on the internet and it counts about 72 in total which comes in five different general categories.

For the 72 WSDL addresses that are obtained by crawling, we established HTTP connections to them and we downloaded those WSDL files. There were few types of WSDL download failures and it is viewed in table 1, and first column lists the HTTP codes which indicate various types of failures. As from the table 1, these are the types of failure availability in the internet and most common type of failure is time-out. Apart from time-out failures, other failures include File Not Found failure, Internal Server Error failure, etc. The removal of WSDL files in the respective web address lead to the File Not Found failure, whereas Internal Server Error is caused due to servers which encountered unexpected errors preventing them from solving the request. These kinds of download failures depicts that in the internet WSDL files are easily prone to non-availability due to following reasons

- Highly dynamic nature of the internet
- Information about some web services are out dated
- Few web services are permanently removed from the internet as they are deployed only for experimental purposes.

TABLE 1
WSDL File Download Failures

| Codes | Descriptions | # WS | Percents |
|---|---|---|---|
| 400 | Bad Request | 173 | 3.57% |
| 401 | Unauthorized | 106 | 2.19% |
| 403 | Forbidden | 153 | 3.16% |
| 404 | File Not Found | 1,468 | 30.31% |
| 405 | Method Not Allowed | 1 | 0.02% |
| 500 | Internal Server Error | 505 | 10.43% |
| 502 | Bad Gateway | 51 | 1.05% |
| 503 | Service Unavailable | 22 | 0.45% |
| 504 | Gateway Timeout | 788 | 16.27% |
| 505 | HTTP Version Not Support | 1 | 0.02% |
| N/A | Connection Timed Out | 774 | 15.98% |
| N/A | Read Timed Out | 787 | 16.25% |
| N/A | Unknown Host | 12 | 0.25% |
| N/A | Redirected Too Many Times | 3 | 0.06% |
| Total | | 4,844 | 100.00% |

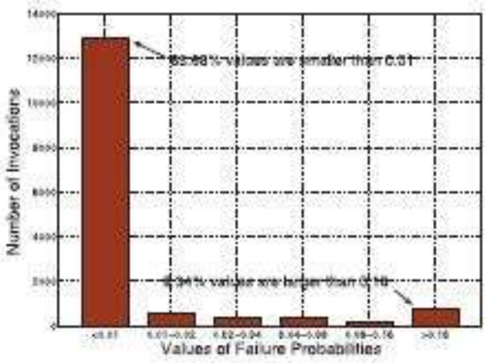
## III. WEB SERVICE QOS EVALUATION

To get data sets of Qos for web services, several large-scale evaluations of Qos for real-world web services should be conducted. Client-side web service invocation codes are employed by Axis2 to generate, test the test cases automatically.

From distributed locations to evaluate real-world web services, we should employ a lot of computers in distributed form from the PlanetLab4 so that it can serve as service to the users. A global research network like planet Lab should be made up of more than 1,000 computers in distributed form globally. The web service evaluation codes should be deployed so that the Planet Lab computer can be used to monitor the QoS from the web services of real-world from many distributed locations. From 2009, researchers conducted 3 evaluations of QOS and got three research sets of data. The following provides description of these three data sets.

### 3.1 Data Set 1: Failure Probability

Random selection of 100 web Services in the first evaluation from the 13,108 web services got from the Section 2 could employ at least 150 computers among 24 countries in the PlanetLab so that it would serve the service users. Their evaluation could focus on the studying of failure probability in QoS data sets, which can be defined with a probability which is an invocation of certain web service that a user may fail. By dividing the failed number of invocations by the invocations total number by a user, we can calculate value of failure probability.




Each web service user invoked all 100 web services which is selected for 100 times and detailed QoS values are recorded. Here 100 times of invocations are selected due to invocations of larger range consume lot of resources from the web services of real-world which is designed typically for business people purpose, while smaller number of invocations will not get a correct probability in failure values. A total range of 1,542,884 invocations in web service are conducted for the service users. Experimenting the results, we get a 150x100 item for user matrix, in which an entry $f_{a,i}$ in the user matrix is probability in failure of web service which can be observed by the service of user a. In the Table 3 format, standard and mean deviation for

All of the 15,000 probabilities in failure is got by 150 users on the web services(100) that are 17.32 and 4.03 percent, to indicate the failure in the probabilities of various web services got by various service users which may exhibit a variation in a great manner. Fig. 2 shows the failure of value distribution in probability. As in Fig. 2, almost85.68 percent of the probability in failure values are less than a percent, i.e., a larger part (8.34 percent) failure in probabilities can still encounter performance with poor values greater than 16 percent. There are many kinds of failures in web service invocation.

Web service responses in HTTP codes can be employed for the detection in the kinds of failure (i.e., HTTP code 200 will indicate the success in invocation, whereas the other HTTP c codes of failures and the exceptions stands for different kinds of failures).

### 3.2 Set of Data 2: Throughput and Response Time

The evaluation 2 focuses on investigating the throughput and response time of the web services performance. Duration of time between a service user which sends a request and then receiving a corresponding response is defined as a Response Time.

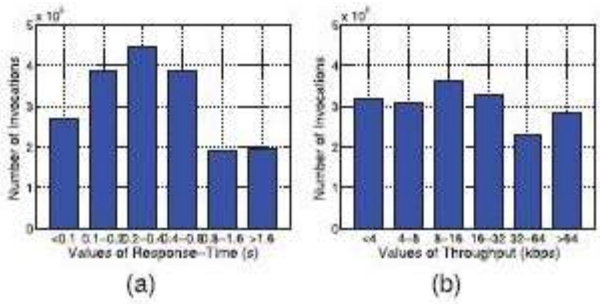



**Fig. 3 Value Distributions of Data Set 2.**

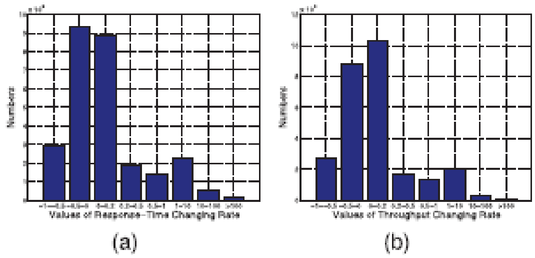
Throughput can be defined as the average successful message in its size of delivery over channel of communication per second. The evaluation 2 that was conducted in August 2009 shows Table 5,which is a total of 1,974,675 web service in the real-world invocations that are executed by nearly 339 service users from more than 30 countries with 5,825 real-world web services of 73 countries for the evaluation. By the web service invocation processing results, we get two 339 x 5,825 matrices for throughput and response time. Each matrix entry represents throughput value or the response-time value got by a web service by the user. In Table 5, the standard and mean deviation of the response time are 1.43 and 31.9 seconds, while the standard and mean deviation of throughput are531.85 kbps and102.86

### 3.3 Data Set 3: Time-Aware Performance

Since Internet is highly dynamic, the user-observed performance (e.g., response time, throughput) of web services is changing from time to time, influenced by the user environment, network condition, server workload, and so on. The third evaluation of web services focuses on investigating time-aware performance of web services. In

March 2011, we employed the distributed PlanetLab computers to monitor real-world web services continuously. A total of 4,532 publicly available real-world web services from 57 countries are monitored by 142 computers located in 22 countries in 64 different time slots. The time interval between neighboring time slots is 15 minutes. The detailed response-time and throughput values of the 64 time slots are collected. Totally 30,287,611 real-world web service invocations are conducted in this evaluation. In the highly dynamic Internet environment, QoS of web services may change from time to time. To investigate the QoS value changing with time, we employ the following equation to evaluate the changing rate of QoS values between two neighboring time slots: $r_i = (q_i - q_{i-1})/q_{i-1},$

Where $q_i$ and $q_{i-1}$ represent the QoS values of the time slots i and i-1, respectively, and $r_i$ represents the changing rate between these two time slots.




ig. 6. Value distributions of QoS changing rate.

Fig. 6 shows the changing rate distributions of response time and throughput. As shown in the figure, we can see that most changing rates of response time and throughput are between -0.5 and 0.2. Moreover, there is a small part of QoS changing rates with very large values (e.g., larger than 10 or even larger than 100), indicating that response time and throughput values of web services can seriously change at different time.

To further investigate the changing response time of different users, we randomly select three users and plot their observed response-time values on the same web services in the 64 time slots. Fig. 8 shows the response-time values of these three users. From the figure, we have a similar observation with Fig. 7, i.e., the user-observed response-time performance of web services can change dynamically with time (e.g., user 3 in the figure).

## IV. APPLICATIONS OF QoS DATASETS

### 4.1 Web Service Qos Prediction

Generating QoS values for different web service is more critical .web service evaluation [13][14][15] is the methods for retiring QoS values there are many web services available on internet it is actually a complex task to find out which is best

In order of surveying web services it aim to provide personalized QoS value for web service user dependently each entry of representing matrix of certain QoS property .the research of web service problem is to predict the values of missing QoS by calculating the available QoS

In providing service, web service has turned a lot of attractions in years large number of QoS predication took place.

### 4.2 Web Service Selection

Service computing many more good quality applications can be constructed by using web service ,used to selection of service candidate .the service candidates has similar functionality but alternate for non functional characters this problem is generally modeled as an optimization problem . Local approaches [1], [7] while global approaches [19], [7], [20] select a set of services that satisfy the process constraints and user preferences for the whole application together. To evaluate the performance of different selection approaches, real-world QoS.

### 4.3 Qos Aware Web Service Search

In service computing Web service discovery was found to be the fundamental research problem. With increase in number of web services, the functionalities to fulfil users' request will be same in many web service. The two Major approaches in order to discover suitable web service are UDDI and web service engine. In recent survey the availability of web service is decreased. Al-Masri and Mahmoud [21] has given enough proof that UDDI business registered services are invalid which is approximately equal to 53 percent. The common ways to discover web service is using search engine.

Traditional Web service search approaches [22] uses keyword-based techniques without the help of QoS web service. In reality, the web services sharing is different in functionalities and as well non functionalities. The functional and non functional characteristics are required to provide effective personalized web service search result.

Zhang et al [11] discovered a web service named WSExpress by giving importance to functional attributes as well as QoS values. The released data of QoS were taken as experiment in order to study the performance of different QoS-aware web service search approach.

### 4.4 Fault Tolerant Web Services

The traditional stand-alone software systems is much more challenging than building reliable service-oriented systems is due to several reasons such as 1) without any internal design and implementation details, remote web services are developed and hosted by other providers. 2) Performance of web service may change frequently. 3) the remote web service may not be available

Software fault tolerance [23] is said to be important approach to build reliable systems. The one approach of software fault tolerance is also said to be design diversity, to employ functionally equivalent yet independently designed components. It is due to the cost of developing redundant software components and software fault is mainly used for critical systems. There are already many different organizations in the area of service computing which is available on the internet. This can be used as alternative component for building fault tolerance service-oriented systems. The QoS is used as alternative service instead of designing optimal fault tolerance strategies.
In previous work [10] QoS-aware fault tolerant is used for service-oriented systems. The recently released QoS data is used as case study for next experiments of different tolerance strategies.

## V. RELATED WORK AND DISCUSSION

In Service computing[2] there are many QoS based approaches have been occupied by web service recommendation[3],[4],[5], service composition[6],[7], fault-tolerance web services[8],[9],[10], web service search[11],and so on. Even though, there is a lack in real world web service QoS data are used to verify these approaches. The characteristics of web service cannot be fully mined without large scale web service data sets many QoS based approaches are difficult to be realistic and practical.
In previous work [9], the five service users have been conducted by real world web service. The experimental results are not useful for future research as the scale of this experiment is too small. Al-Masri and Mahmoud [21] released a web service QoS data which will be useful for 2,507 web service. The different user will use different QoS as per the availability of the data set. The data sets of this paper will also include QoS information observed from distributed service users in different time slots. Vieira et al [24] conducted an experimental web service for 300 publicly available web services. The security vulnerabilities will exist at server side and it is user independent. Apart from Vieira's work [24], this paper mainly focuses on investigating user observed QoS properties which will vary widely from different users.

## VI. CONCLUSION AND FUTURE ENHANCEMENT

This project conducts evaluations in various web service Qos data sets observed by users from distributed locations. In real world web services, a large number of web services are invoked and these results are gathered. Some data sets which can be reused are released. In our future work, apart from response time, failure probability, throughput, Qos ratio, type of browser support, and its home source, additional Qos properties can be observed by the user and implemented in user friendly manner.

## REFERENCES

[1] D.A. Menasce, *"QoS Issues in Web Services,"* IEEE Internet Computing, vol. 6, no. 6, pp. 72-75, Nov./Dec. 2002.

[2] L.-J. Zhang, J. Zhang, and H. Cai, Services Computing. Springer,2007.

[3] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, *"Personalized QoS-Aware Web Service Recommendation and Visualization,"* IEEE Trans. Services Computing, vol. 6, no. 1, pp. 35-47, 2011.

[4] Z. Zheng, H. Ma, M.R. Lyu, and I. King, *"QoS Aware Web Service Recommendation by Collaborative Filtering,"* IEEE Trans. Service Computing, vol. 4, no. 2, pp. 140-152, Apr.-June 2011.

[5] Z. Zheng, H. Ma, M.R. Lyu, and I. King, *"Collaborative Web Service QoS Prediction via Neighborhood Integrated Matrix Factorization,"* IEEE Trans. Service Computing, vol. 6, no. 3,pp. 289-299, July-Sept. 2013.

[6] M. Alrifai and T. Risse, *"Combining Global Optimization with Local Selection for Efficient QoS-Aware Service Composition,"*Proc. 18th Int'l Conf. World Wide Web (WWW '09), pp. 881-890, 2009.

[7] L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, *"QoS-Aware Middleware for Web Services Composition,"* IEEE Trans. Software Eng., vol. 30, no. 5, pp. 311-327, May 2004.

[8] C.-L. Fang, D. Liang, F. Lin, and C.-C. Lin, "*Fault-Tolerant Web Services,*" J. System Architecture, vol. 53, no. 1, pp. 21-38, Jan. 2007.

[9] Z. Zheng and M.R. Lyu, *"A Distributed Replication Strategy Evaluation and Selection Framework for Fault-Tolerant Web Services,"* Proc. Sixth Int'l Conf. Web Services (ICWS '08), pp. 145-152, 2008.

[10] Z. Zheng and M.R. Lyu, *"A QoS-Aware Fault Tolerant Middleware for Dependable Service Composition,"* Proc. 39th Int'l Conf. Dependable Systems and Networks (DSN '09), pp. 239-248, 2009.

[11] Y. Zhang, Z. Zheng, and M.R. Lyu, *"WSExpress: A QoS-AwareSearch Engine for Web Services,"* Proc. Eighth Int'l Conf. Web Services (ICWS '10), pp. 91-98, 2010.

[12] Z. Zheng, Y. Zhang, and M.R. Lyu, *"Distributed QoS Evaluation for Real-World Web Services,"* Proc. Eighth Int'l Conf. Web Services (ICWS '10), pp. 83-90, 2010.

[13] V. Deora, J. Shao, W. Gray, and N. Fiddian, *"A Quality of Service Management Framework Based on User Expectations,"* Proc. First Int'l Conf. Service-Oriented Computing (ICSOC '03), pp. 104-114,

[14] E. Maximilien and M. Singh, *"Conceptual Model of Web Service Reputation,"* ACM SIGMOD Record, vol. 31, no. 4, pp. 36-41, 2002.

[15] W.-T. Tsai, X. Zhou, Y. Chen, and X. Bai, *"OnTesting and Evaluating Service-Oriented Software,"* IEEE Computer, vol. 41,no. 8, pp. 40-46, Aug. 2008.

[16] G. Wu, J. Wei, X. Qiao, and L. Li, *"A Bayesian Network Based QoS Assessment Model for Web Services,"* Proc. Int'l Conf. Services Computing (SCC '07), pp. 498-505, 2007.

[17] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, *"Personalized QoS Prediction for Web Services via Collaborative Filtering,"* Proc. Fifth Int'l Conf. Web Services (ICWS '07), pp. 439-446, 2007.

[18] Z. Zheng, Y. Zhang, and M.R. Lyu, *"CloudRank:A QoS-Driven Component Ranking Framework for Cloud Computing,"* Proc. Int'l Symp. Reliable Distributed Systems (SRDS '10), pp. 184-193,2010.

[19] T. Yu, Y. Zhang, and K.-J. Lin, *"Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints,"* ACM Trans.Web, vol. 1, no. 1, pp. 1-26, 2007.

[20] D. Ardagna and B. Pernici, *"Adaptive Service Composition in Flexible Processes,"* IEEE Trans. Software Eng., vol. 33, no. 6,pp. 369-384, June 2007.