

Investigating QoS of Real-World Web Services

Zibin Zheng, *Member, IEEE*, Yilei Zhang, *Student Member, IEEE*, and Michael R. Lyu, *Fellow, IEEE*

Abstract—Quality of service (QoS) is widely employed for describing nonfunctional characteristics of web services. Although QoS of web services has been investigated intensively in the field of service computing, there is a lack of real-world web service QoS data sets for validating various QoS-based techniques and models. To investigate QoS of real-world web services and to provide reusable research data sets for future research, we conduct several large-scale evaluations on real-world web services. First, addresses of 21,358 web services are obtained from the Internet. Then, three large-scale real-world evaluations are conducted. In our evaluations, more than 30 million real-world web service invocations are conducted on web services in more than 80 countries by users from more than 30 countries. Detailed evaluation results are presented in this paper and comprehensive web service QoS data sets are publicly released online.

Index Terms—Web service, quality of service, service evaluation, QoS data set

1 INTRODUCTION

WEB services have been emerging in recent years and are by now one of the most popular techniques for building distributed systems. Service-oriented systems can be built efficiently by dynamically composing different web services, which are provided by other organizations. The quality-of-service (QoS)-oriented systems are highly reliant on the quality of employed web services. With the prevalence of web services on the Internet, investigating quality of web services is becoming more and more important.

QoS is widely employed for describing nonfunctional characteristics of web services. With the increasing number of web services, QoS has become an important differentiating point of different functionally equivalent web services. Web service QoS includes a number of properties, such as response time, throughput, failure probability, availability, price, popularity, and so on [1]. Values of server-side QoS properties (e.g., price, popularity) are usually advertised by service providers and identical for different users. On the other hand, values of the user-observed QoS properties (e.g., user-observed response time, throughput, failure probability) can vary widely for different users, influenced by the unpredictable Internet connections and the heterogeneous user environments [1].

In the field of service computing [2], a number of 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. However, there is still a lack of comprehensive real-world web service QoS data sets for validating various QoS-based approaches.

To obtain user-observed QoS values of real-world web services, which are provided by different companies and actively used by other organizations, evaluations from different geographic locations under various network conditions are required. However, it is not an easy task to conduct large-scale web service evaluations from distributed locations, because 1) web service invocations consume resources of both service users and service providers; 2) it is time-consuming and expensive to conduct real-world evaluations on all the service candidates when the number of candidates is large; and 3) it is difficult to collect web service QoS data from distributed service users. However, without comprehensive real-world evaluations, sufficient web service QoS values cannot be collected. It is thus difficult to validate the feasibility and effectiveness of various QoS-based approaches in service computing.

To attack this critical challenge, we make a great effort to conduct three large-scale distributed evaluations on real-world web services, collect comprehensive web service QoS data sets, and publicly release these reusable data sets for future research. First, 21,358 web service addresses are obtained by crawling web service information from the Internet. Then three web service evaluations are conducted. In the first evaluation, failure probability of 100 web services is assessed by 150 distributed service users. In the second evaluation, response time and throughput of 5,825 web services are evaluated by 339 distributed service users. And in the third evaluation, QoS changing of 4,532 web services with time is studied by conducting 30,287,611 web service invocations by 142 users in 64 time slots with a time interval of 15 minutes. First hand experiences on real-world web service QoS are provided in this paper and reusable QoS research data sets are publicly released for future research.¹ Extended from its previous conference version [12], which reports the experimental results of the first two evaluations, the extensions of this journal version include: 1) providing detailed analysis and discussions on the relationship between QoS values and time, and 2) showing the applicability of our data sets while engaging research topics of QoS prediction, web service selection, web service search, and fault-tolerant web services.

• The authors are with the Shenzhen Key Laboratory of Rich Media Big Data Analytics and Applications, Shenzhen Research Institute, The Chinese University of Hong Kong, and with the Ministry of Education Key Laboratory of High Confidence Software Technologies (CUHK Sub-Lab) Computer Science & Engineering Department, The Chinese University of Hong Kong, Ho Sin-Hang Engineering Building, Shatin, N.T., Hong Kong. E-mail: zbzhen@cs.cuhk.edu.hk.

Manuscript received 1 Apr. 2012; revised 11 Aug. 2012; accepted 3 Nov. 2012; published online 16 Nov. 2012.

For information on obtaining reprints of this article, please send e-mail to: tsc@computer.org, and reference IEEECS Log Number TSC-2012-04-0033. Digital Object Identifier no. 10.1109/TSC.2012.34.

1. <http://www.wsdream.net>.



Fig. 1. Locations of web services.

The remainder of this paper is organized as follows: Section 2 introduces the information of web services. Section 3 presents 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 paper.

2 INFORMATION OF WEB SERVICES

Web services can be discovered from universal description, discovery, and integration (UDDI), which is an XML-based registry enabling company to publish and discover web services on the Internet), web service portals (e.g., xmethods.net, webservicex.net, webservicelist.com), and web service search engines (e.g., seekda.com, esynaps.com). By crawling web service information from UDDI, web service portals, and web service search engines, we obtain 21,358 addresses of WSDL (Web Service Description Language) files. Seekda.com² reports that there are a total of 28,606 public web services on the Internet. Therefore, the 21,358 web services in our experiments cover most of the publicly available real-world WSDL-based web services on the Internet. As shown in Fig. 1, these services are distributed all over the world. Most web services are located in North America and Europe. Among all the 89 countries, the top three countries provide 55.5 percent of the 21,358 obtained web services. These three countries are United States (8,867 web services), United Kingdom (1,657 web services), and Germany (1,246 web services).

By establishing HTTP connections to the 21,358 WSDL addresses obtained, we successfully download 16,514 (77.32 percent) WSDL files. The WSDL download failures are summarized in Table 1, where the first column lists the HTTP codes indicating different types of failures. The HTTP codes of the last four failure types in Table 1 are nonavailable (N/A), since we fail to establish HTTP connections and thus are unable to obtain the server returned HTTP codes. As shown in Table 1, there is a total of 4,844 failures. 48.49 percent of these failures are time-out failures caused by network connection problems, including 788 (16.27 percent) *Gateway Time-out*, 774 (15.98 percent) *Connection timed-out*, and 787 (16.25 percent) *Read timed-out*. Besides the time-out failures, there are also a lot of *File Not Found* failures (30.31 percent) and *Internal Server Error* failures (10.43 percent). The *File Not Found* failures are caused by the removal of WSDL files or update of WSDL addresses, while the *Internal Server Error* failures are caused by the fact that the servers encountered unexpected

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%

TABLE 2
Java Code Generation Failures

Failure Types	# WS	Percents
Empty File	249	7.31%
Invalid File Format	1,232	36.17%
Error Parsing WSDL	1,135	33.32%
Invocation Target Exception	764	22.43%
Null QName	22	0.65%
Databinding Unmatched Type Exception	4	0.12%
Total	3,406	100%

conditions which prevented them from fulfilling the request. These download failures indicate that WSDL files on the Internet can become unavailable easily, because 1) the Internet is highly dynamic, 2) some web service information on the Internet is out of date, and 3) some web services (e.g., web service made for experimental purposes) are removed from the Internet quickly.

Employing Axis2,³ we successfully generate client-side web service invocation Java codes for 13,108 (79.38 percent) web services among all the 16,514 web services. A total of 235,262,555 lines of Java codes are produced. There are 3,406 code generation failures, which are summarized in Table 2. As shown in Table 2, among all the 3,406 generation failures, 249 *Empty File* failures are caused by the fact that the WSDL files are empty; 1,232 *Invalid File Format* failures are due to that these WSDL files do not follow standard WSDL format; and 1,135 *Error Parsing* failures are caused by syntax errors of WSDL files. There are also 22 *Null QName* failures and four *Databinding Unmatched Type* failures. These generation failures indicate that the WSDL files on the Internet are fragile, which may contain empty content, invalid format, invalid syntax, and other various types of errors.

3 QoS EVALUATION OF WEB SERVICES

To obtain comprehensive QoS data sets of web services, we conduct several large-scale QoS evaluations of real-world web services. Axis2 is employed to generate client-side web service invocation codes and test cases automatically. To evaluate real-world web services from distributed locations,

2. <http://webservices.seekda.com>.

3. <http://ws.apache.org/axis2>.

TABLE 3
Statistics of Data Set 1

Statistics	Values
Num. of Web Service Invocations	1,542,884
Num. of Service Users	150
Num. of Web Services	100
Num. of User Countries	24
Num. of Web Service Countries	22
Mean of Failure Probability	4.05%
Standard Deviation of Failure Probability	17.32%

we employ a number of distributed computers from PlanetLab⁴ to serve as service users. PlanetLab is a global research network made up of more than 1,000 distributed computers globally. By deploying the web service evaluation codes to the PlanetLab computers, we can monitor the QoS of real-world web services from distributed locations. Since 2009, we have conducted three QoS evaluations and obtained three comprehensive research data sets. Detailed descriptions of these three data sets are provided in the following.

3.1 Data Set 1: Failure Probability

In the first evaluation, we randomly select 100 web services from the 13,108 web services obtained in Section 2 and employ 150 computers in 24 countries from PlanetLab to serve as service users. This evaluation focuses on studying QoS property of failure probability, which is defined as the probability that an invocation on a certain web service by a user will fail. The value of failure probability can be approximately calculated by dividing the number of failed invocations by the total number of invocations conducted by a user on a web service. In this evaluation, each service user invokes all the 100 selected web services for 100 times and records the detailed QoS values. We select 100 times of invocations since a large number of invocations consume too many resources of the real-world web services which are typically designed for business purpose, while a small number of invocations may not be able to obtain accurate failure probability values. A total of 1,542,884 web service invocations are conducted by the service users. By processing the experimental results, we obtain a 150×100 user-item matrix, where an entry $f_{a,i}$ in the matrix is the failure probability of web service i observed by service user a .

As shown in Table 3, the mean and standard deviation of all the 15,000 failure probabilities observed by 150 users on 100 web services are 4.05 and 17.32 percent, respectively, indicating that the failure probabilities of different web services observed by different service users exhibit a great variation. Fig. 2 shows the value distribution of failure probability. As shown in Fig. 2, although 85.68 percent of all the failure probability values are smaller than 1 percent, a large part (8.34 percent) of failure probabilities still encounter poor performance with values larger than 16 percent.

There are various types of web service invocation failures. HTTP codes of the web service responses can be employed for detecting the failure types (i.e., HTTP code 200 indicates invocation success, while other HTTP codes and exceptions stand for various types of failures). As shown in

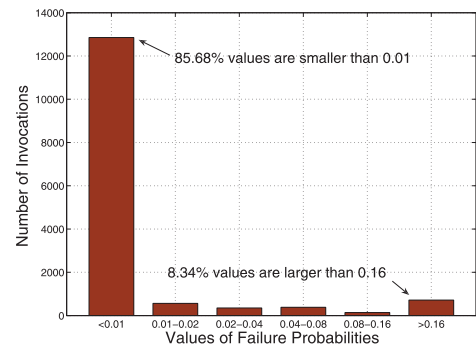


Fig. 2. Value distributions of data set 1.

Table 4, among all the 1,542,884 web service invocations, there are a total of 58,184 invocation failures. The detailed failure information is summarized in Table 4. Descriptions of different failure types are introduced as follows:

- (400)Bad Request. The web server was unable to understand the request since the client request did not respect the HTTP protocol completely.
- (500)Internal Server Error. The web server encountered an unexpected condition that prevented it from fulfilling the client request.
- (502)Bad Gateway. A gateway or proxy server received an invalid response from an upstream server it accessed to fulfil the request.
- (503)Service Unavailable. The web server was unable to handle the HTTP request due to a temporary overloading or maintenance of the server.
- Network is unreachable. A socket operation was attempted to an unreachable network, it did not get a response and there was no default gateway.
- Connection reset: The socket was closed unexpectedly from the server side.
- NoRouteToHostException. Socket connection failed caused by intervening firewall or intermediate router errors.
- Connection refused. An error occurred while attempting to connect a socket to a remote address and port. Typically, the connection was refused remotely (e.g., no process was listening on the remote address/port).
- Read timed-out. Time-out occurred on socket read.

TABLE 4
Invocation Failures of Data Set 1

Descriptions	Numbers
(400)Bad Request	3
(500)Internal Server Error	26
(502)Bad Gateway	33
(503)Service Unavailable	609
java.net.SocketException: Network is unreachable	3
java.net.SocketException: Connection reset	1,175
java.net.NoRouteToHostException: No route to host	415
java.net.ConnectException: Connection refused	619
java.net.SocketTimeoutException: Read timed out	4,606
java.net.UnknownHostException	5,847
java.net.SocketTimeoutException: Connect timed out	44,809
Other errors	39
Total	58,184

4. <http://www.planet-lab.org>.

TABLE 5
Statistics of Data Set 2

Statistics	Values
Num. of Web Service Invocations	1,974,675
Num. of Service Users	339
Num. of Web Services	5,825
Num. of User Countries	30
Num. of Web Service Countries	73
Mean of Response-Time	1.43 s
Standard Deviation of Response-Time	31.9 s
Mean of Throughput	102.86 kbps
Standard Deviation of Throughput	531.85 kbps

- *UnknownHostException*. The IP address of a host could not be determined.
- *Connect timed-out*. A time-out has occurred on a socket connect.
- *Other failures*. The invocation failure types cannot be identified due to lack of information.

As shown in Table 4, about 85 percent of these failures are due to socket connection problems, including 44,809 *Connect timed-out* and 4,606 *Read timed-out*. These timed-out exceptions are caused by network connection problems during socket connection and socket read. Besides the time-out exceptions, there are also a lot of other failures caused by network errors, including 33 *Bad Gateway*, 3 *Network is Unreachable*, 415 *No route to host*, and 5,847 *Unknown Host*. Some failures in Table 4 are caused by server-side errors, including 3 *Bad Request*, 26 *Internal Server Error*, 608 *Service Unavailable*, 1175 *Connection reset*, and 619 *Connection refused*. These experimental observations on invocation failures show that 1) web service invocations can fail easily, which can be caused by gateway errors, networking errors, and server errors; and 2) in the service-oriented environment, providing reliable web services is not enough for building reliable service-oriented system, since most invocation failures are caused by network errors.

3.2 Data Set 2: Response Time and Throughput

The second evaluation focuses on investigating the response time and throughput performance of web services. Response time is defined as the time duration between a service user sending a request and receiving the corresponding response, while throughput is defined as the average rate of successful message size (here in bits) delivery over a communication channel per second. The second evaluation was conducted in August 2009. As shown in Table 5, a total of 1,974,675 real-world web service invocations are executed by 339 service users from 30 countries on 5,825 real-world web services in 73 countries in this evaluation.

By processing the web service invocation results, we obtain two $339 \times 5,825$ matrices for response time and throughput, respectively. Each entry in a matrix represents the response-time value or throughput value observed by a user on a web service. As shown in Table 5, the mean and standard deviation of response time are 1.43 and 31.9 seconds, respectively, while the mean and standard deviation of throughput are 102.86 and 531.85 kbps, respectively. The large standard deviation values indicate

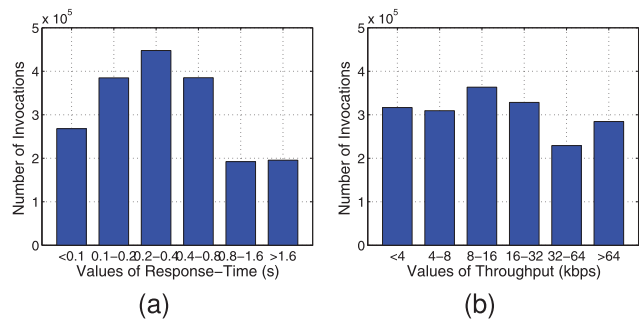


Fig. 3. Value distributions of data set 2.

that response time and throughput have a wide range of values which are quite different with each other. Fig. 3 shows the value distributions of response time and throughput. Fig. 3a shows that most of the response-time values are smaller than 1.6 seconds. Fig. 3b shows that most throughput values are smaller than 64 kbps.

To provide detailed illustration of the web service response-time and throughput values observed by different service users, we randomly choose two users (User 1 from the US and User 2 from Japan) to compare their response-time and throughput performance on different web services. We randomly select 100 web services and plot the response-time and throughput values of these web services observed by these two users. Figs. 4a and 4b show the performance comparison of response time and throughput, respectively. As shown in Fig. 4, although invoking the same web services, values of response time and throughput are quite different of these two users. For example, response-time values of User 1 are around 6 seconds on most of the web services, while response-time values of User 2 is less than 2 seconds on most of the web services. The long response time of user 1 may be caused by the poor client-side network condition. This experimental observation indicates that different users may have different usage experiences on the same web service, influenced by the network connections and the heterogenous client-side environments. Therefore, distributed web service evaluation is important for obtaining accurate user-observed QoS of web services.

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

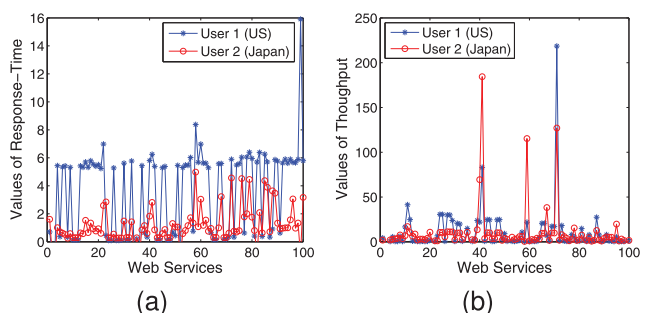


Fig. 4. Two users' QoS values.

TABLE 6
Statistics of Data Set 3

Statistics	Values
Num. of Web Service Invocations	30,287,611
Num. of Service Users	142
Num. of Web Services	4,532
Num. of User Countries	22
Num. of Web Service Countries	57
Num. of Time Slots	64
Interval of Time Slots	15 minutes
Mean of Response-Time	3.165s
Standard Deviation of Response-Time	6.12 s
Mean of Throughput	9.609kbps
Standard Deviation of Throughput	50.11 s

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.

As shown in Table 6, the means of response time and throughput are 3.165 seconds and 9.609 kbps, respectively. The standard deviations of response time and throughput are 6.12 seconds and 50.11 kbps, respectively. The large standard deviation indicates that these QoS properties includes a wide range of values. The distributions of the response time and throughput are shown in Fig. 5. From the figure, we can see that most response-time values are between 0.1 and 0.8 seconds and most throughput values are between 0.8 and 3.2 kbps.

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:

$$r_i = (q_i - q_{i-1})/q_{i-1}, \quad (1)$$

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. Fig. 6 shows the changing rate distributions of response time and throughput. As shown in

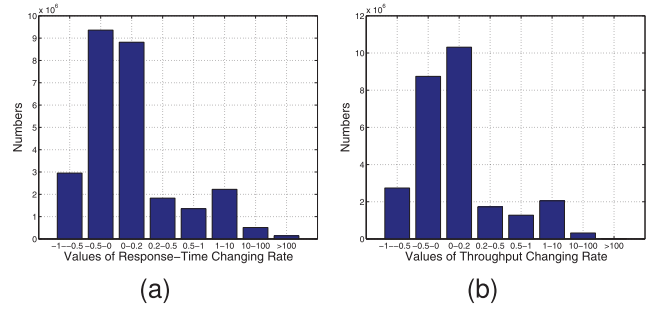


Fig. 6. Value distributions of QoS changing rate.

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 provide a detailed illustration of the web service response time changing with time, we randomly choose a user and plot his/her observed responsetime values on three different web services in the 64 time slots. Fig. 7 shows the response time values of these three web services in different time slots. From the figure, we can see that 1) the user-observed response time performance of web services can change dynamically with time. For example, web service 1 in the figure has quite different response time values at different time slots. 2) The same user may experience quite different response time changing patterns on different web services. For example, response-time performance of web service 1 is more dynamic than web service 3 in the figure. This research observation indicates that QoS changing relates to web services, since different web services demonstrate quite different changing patterns for the same user.

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). Another interesting observation is that different users have different

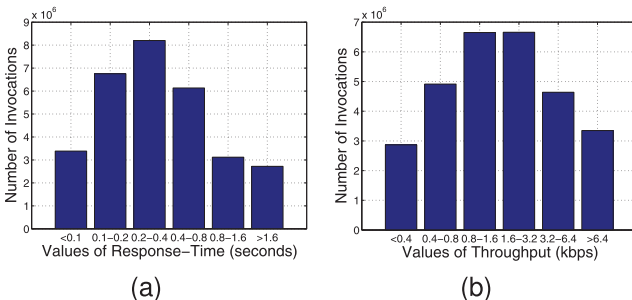


Fig. 5. Value distributions of data set 3.

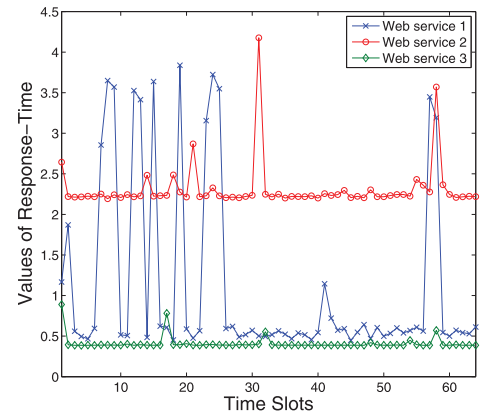


Fig. 7. Response time of three web services.

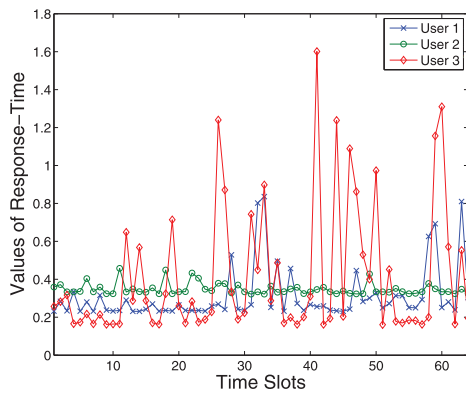


Fig. 8. Response time of three users.

response-time changing patterns on the same web service, indicating that QoS changing relates to users.

In summary, the experimental observations in this evaluation show that time is an important element when investigating QoS of web services. The QoS fluctuation with time brings a great challenge for various QoS-based approaches of web services. To achieve optimal system performance, dynamic adaptation and reconfiguration of service-oriented systems become necessary.

4 APPLICATION OF QoS DATASETS

4.1 Web Service QoS Prediction

Obtaining QoS values of web services is critical for various QoS-based approaches. Web service evaluation [13], [14], [15], [16] is a main approach for obtaining QoS values. However, conducting web service evaluation is difficult, since real-world web service invocations consume resources of service providers and impose costs of service users. Moreover, since there are a lot of web services in the Internet, it is impractical for service users to evaluate all the web service candidates.

Instead of conducting real-world web service invocations, web service QoS prediction aims at providing personalized QoS value prediction for service users, employing the historical QoS values. Web service QoS prediction usually includes a user-item matrix, where each entry in the matrix represents the value of a certain QoS property (e.g., *response time*) of a web service observed by a service user. The user-item matrix is usually very sparse, since a service user typically only invoked a small number of web services in the past. The research problem of web service QoS prediction is how to accurately predict the missing QoS values in the user-item matrix by employing the available QoS values. By predicting the web service QoS values in the user-item matrix, personalized QoS value prediction on the unused web services for service users can be achieved.

In service computing, web service QoS prediction has attracted a lot of attention in recent years. A number of QoS prediction approaches have been proposed, including user-based QoS prediction approach [17], UIPCC [4], matrix factorization approach [5], ranking-oriented approach [18], RegionKNN [3], and so on. To evaluate the prediction accuracy of various prediction approaches, real-world web

service QoS values from different users are needed. Our data sets can be employed in the experiments to evaluate the prediction accuracy of different prediction approaches.

4.2 Web Service Selection

In service computing, complex applications can be built efficiently by dynamically composing web services, which are selected at runtime from a set of functionally equivalent service candidates. These service candidates have the same functionality but differ for nonfunctional characteristics, which are described by QoS. The goal of service selection is to select the best set of services at runtime, considering process constraints, end-user preferences, as well as QoS of service candidates.

The service selection problem is usually modeled as an optimization problem. Local approaches [1], [7] select optimal web service for each abstract task independently, 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 values of web services are needed.

Our data sets include QoS values of a large number of web services, which can be employed for experimental studies of service selection. Moreover, our data set 3 includes the detailed QoS values of 64 different time slots. These time-aware QoS data provide valuable information to study the uncertainty of the highly dynamic Internet environment, where QoS is changing from time to time.

4.3 QoS-Aware Web Service Search

Web service discovery is a fundamental research problem in service computing. With the growing number of web services in the Internet, many web services provide similar functionalities to fulfil users' requests. UDDI and web service search engine are two major approaches for discovering suitable web services. Recently, the availability of web services in UDDI has decreased rapidly. Al-Masri and Mahmoud [21] show that more than 53 percent of the UDDI business registry registered services are invalid. Using search engine to discover web services has become more common nowadays.

Traditional web service search approaches [22] typically only exploit keyword-based search techniques without considering QoS of web services. In reality, web services sharing similar functionalities may have very different nonfunctionalities. To effectively provide personalized web service search results to different users, it is requisite to consider both functional and nonfunctional characteristics of web services when searching web services.

Zhang et al. [11] proposed a web service discovering approach, named WSExpress, by paying respect to functional attributes as well as QoS values of web services. Our released QoS data sets were employed in the experiments of this work to study the performance of different QoS-aware web service search approaches.

4.4 Fault-Tolerant Web Services

Comparing with the traditional stand-alone software systems, building reliable service-oriented systems is much more challenging, because 1) remote web services are

developed and hosted by other providers without any internal design and implementation details; 2) performance of web services may change frequently (e.g., caused by workload change of servers, internal updates of web services, performance update of communication links); and 3) the remote web services may become unavailable easily.

Software fault tolerance [23] is an important approach to build reliable systems. One approach of software fault tolerance, also known as design diversity, is to employ functionally equivalent yet independently designed components to tolerate faults. However, due to the cost of developing redundant software components, software fault tolerance is mainly used for critical systems. In the area of service computing, there is a number of functionally equivalent web services already diversely implemented by different organizations and publicly available on the Internet. These web services can be enclosed as alternative components for building fault-tolerant service-oriented systems. When designing optimal fault tolerance strategies for service-oriented systems, QoS of the alternative service candidates need to be considered to enhance the system performance. In our previous work [10], a QoS-aware fault-tolerant middleware is proposed for service-oriented systems. Our released QoS data sets have been employed in the experiments to study the performance of different fault tolerance strategies.

5 RELATED WORK AND DISCUSSION

In service computing [2], a lot of 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. However, there is a lack of real-world web service QoS data sets for verifying these approaches. Without large-scale web service data sets, characteristics of real-world web service QoS cannot be fully mined, and various QoS-based approaches are thus difficult to be realistic and practical.

In our previous work [9], a real-world web service evaluation has been conducted by five service users on eight publicly accessible web services. Since the scale of this experiment is too small, the experimental results are not much useful for future research. Al-Masri and Mahmoud [21] released a web service QoS data set that is observed by only one service user on 2,507 web services. The fact that different users will observe quite different QoS of the same web service limits the applicability of this data set. Our released data sets of this paper, on the other hand, include QoS information observed from distributed service users, and in different time slots. Vieira et al. [24] conducted an experimental evaluation of security vulnerabilities in 300 publicly available web services. Security vulnerabilities usually exist at the server-side and are user-independent (different users observe the same security vulnerabilities on the target web service). Different from Vieira's work [24], this paper mainly focuses on investigating user-observed QoS properties (i.e., failure probability, response time, and throughput), which can vary widely among different users.

6 CONCLUSION AND FUTURE WORK

This paper conducts evaluations on user-observed QoS of web services from distributed locations. A large number of web service invocations are executed by service users under heterogeneous environments on real-world web services. Comprehensive experimental results are presented and reusable data sets are released. In our future work, besides failure probability, response time, and throughput, more QoS properties will be investigated.

ACKNOWLEDGMENTS

The work described in this paper was supported by the National Basic Research Program of China (973 Project No. 2014CB347701), the National Natural Science Foundation of China (Project No. 61100078), the Shenzhen Basic Research Program (Project No. JCYJ20120619153834216), and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 415311).

REFERENCES

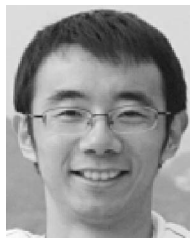
- [1] D.A. Menasce, "QoS Issues in Web Services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72-75, Nov./Dec. 2002.
- [2] L.-J. Zhang, J. Zhang, and H. Cai, *Services Computing*. Springer, 2007.
- [3] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-Aware Web Service Recommendation 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-Aware Search 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, 2003.
- [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, "On Testing 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.
- [21] E. Al-Masri and Q.H. Mahmoud, "Investigating Web Services on the World Wide Web," *Proc. 17th Int'l Conf. World Wide Web (WWW '08)*, pp. 795-804, 2008.
- [22] K. Gomadam, A. Ranabahu, M. Nagarajan, A. Sheth, and K. Verma, "A Faceted Classification Based Approach to Search and Rank Web APIs," *Proc. Sixth Int'l Conf. Web Services (ICWS '08)*, pp. 177-184, Sept. 2008.
- [23] M.R. Lyu, *Software Fault Tolerance*. Wiley, 1995.
- [24] M. Vieira, N. Antunes, and H. Madeira, "Using Web Security Scanners to Detect Vulnerabilities in Web Services," *Proc. 39th Int'l Conf. Dependable Systems and Networks (DSN '09)*, pp. 566-571, 2009.



Zibin Zheng is an associate research fellow at the Shenzhen Research Institute, the Chinese University of Hong Kong. His research interests include service computing and cloud computing. He received the Outstanding PhD Thesis Award of the Chinese University of Hong Kong in 2012, the ACM SIGSOFT Distinguished Paper Award at ICSE 2010, the Best Student Paper Award at ICWS 2010, and the IBM PhD Fellowship Award in 2010. He has served as

a PC member for conferences such as CLOUD 2009, SCC 2011, SCC 2012, and ICSOC 2012. He is a member of the IEEE.



Yilei Zhang received the BSc degree in computer science from the University of Science and Technology of China, Hefei, in 2009. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, the Chinese University of Hong Kong. He has served as a reviewer for international journals as well as conferences, including the

IEEE Transactions on Software Engineering, the *IEEE Transactions on Services Computing*, *WWW*, *KDD*, and *SCC*. His research interests include services computing and cloud computing. He is a student member of the IEEE.



Michael R. Lyu is currently a professor in the Department of Computer Science and Engineering, the Chinese University of Hong Kong. His research interests include software reliability engineering, distributed systems, service computing, information retrieval, social networks, and machine learning. He has published more than 400 refereed journal articles and conference papers in these areas. He is a fellow of the

IEEE and the American Association for the Advancement of Science for his contributions to software reliability engineering and software fault tolerance.