

Large-Scale Longitudinal Analysis of SOAP-Based and RESTful Web Services

Wei Jiang^{1,2}, Dongwon Lee³, Songlin Hu¹

¹Institute of Computing Technology, Chinese Academy of Sciences, China

²Graduate University of the Chinese Academy of Sciences, China

³The Pennsylvania State University, USA

{jiangwei,husonglin}@ict.ac.cn, dongwon@psu.edu

Abstract—While the usage of web services has increased explosively in recent years, very few studies examined the characteristics of web services using large-scale real data for a long period of time. In this paper, we present one such a large-scale longitudinal analysis of publicly available web services of SOAP-based and RESTful types. For the period of roughly one year and from five different world-wide locations, we closely monitor the ups and downs of various basic properties of web services and their QoS values using a total of 825,132 real web services.

I. INTRODUCTION

The W3C defines a *Web Service* (WS) as “a software system designed to support interoperable machine-to-machine interaction over a network.”¹ Since the introduction of the web service concept, abundant research works have been carried out and have significantly improved flexible and dynamic functionalities of the *service oriented architecture* (SOA) in the current web services. As the usage of web services increase explosively, questions about the “the status quo” of web services naturally arise such as: (1) How many web services are out there? Does the number increase rapidly? If so, how fast? (2) How good is the Quality-of-Service (QoS) of existing web services? and (3) What is the impact of time and location to the QoS factors of web services? etc.

Toward these research questions, a few small-scale snapshot-based studies have been proposed (e.g., [1], [2], [3], [4]). However, in this paper, we present a much more comprehensive empirical study that differs from existing approaches in that: (1) We have conducted our experiments for roughly a year, enabling a longitudinal analysis; (2) Using Amazon’s cloud computing environment, we repeated our experiments from five world-wide locations, enabling the location-specific analysis; (3) Our data set contains 825,132 real web services gathered from the Web; and (4) We closely monitor both functional and QoS aspects of SOAP-based and RESTful types of web services.

A. Types and Descriptions of Web Services

Web services are often categorized as two types: (1) *SOAP-based* and (2) *RESTful* web services. SOAP-based

web services are operation oriented and based on numerous W3C standards. On the other hand, RESTful web services refer to resources-based web services architecture that are often simpler to implement and use than SOAP-based web services are. However, RESTful web services themselves are *not* W3C standards. SOAP-based web services are often described by the *Web Services Description Language* (WSDL)², part of W3C standards. However, currently, there is no formal way to describe RESTful web services. Therefore, to describe RESTful web services, people often use different methods such as simple textual description (as in user manual), *Web Application Description Language* (WADL)³, or sometimes the latest WSDL 2.0⁴. In this study, as a method to describe a web service, we will use the notations— M_{wsdl} , M_{wadl} , and M_{text} :

- 1) M_{wsdl} : In this method, WSDL 1.1 is used to describe a web service. Originally, it does not support HTTP operations other than GET and POST, and exclusively used to describe SOAP-based web services type. However, the latest WSDL 2.0 supports all HTTP verbs and can describe RESTful web service type as well.
- 2) M_{wadl} : In this method, WADL is used to describe a web service. WADL is originally championed by Sun Microsystems and is an alternative description document for RESTful web services. Like WSDL, WADL is also an XML-based file format that provides a machine-readable description for web services. WADL is lightweight, and arguably easier to understand and write than WSDL is.
- 3) M_{text} : When describing RESTful web services (and sometimes even SOAP-based web services), companies often use a simple textual description on web pages to explain their APIs, instead of using XML-based WADL or WSDL 2.0. We refer to such a textual description method as M_{text} .

Example 1. Different companies tend to choose different methods to describe their web services. For instance, Google uses M_{text} to describe SOAP-based AdSense APIs⁵ and

²<http://www.w3.org/TR/wsdl>

³<http://wadl.java.net/>

⁴<http://www.w3.org/TR/wsdl20/>

⁵<http://code.google.com/apis/adsense/host/developer/>

¹<http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>

Table I
COMPARISON OF WEB SERVICES STUDIES IN LITERATURE.

Study	[1]	[2]	[3]	[4]
Year	2003	2004	2006	2007
# of WS	1,200	1,544	N/A	5,077
# of valid WS	804	640	3,435	4,033

RESTful web services such as Search or Map services. Similarly, Amazon uses M_{wsdl} to describe EC2⁶ while M_{text} to describe ElasticMapReduce⁷. Finally, Flickr⁸ supports both SOAP-based and RESTful services using both M_{wsdl} and M_{text} methods. On the other hand, other companies such as Microsoft Azure storage⁹ and management¹⁰, Facebook¹¹, and Twitter¹² currently support only RESTful web services via M_{text} method. \square

II. RELATED WORK

While abundant research efforts exist on technical aspects of web services (e.g., service search, composition and complexity [5], [6], [7]), comprehensive and longitudinal studies about the latest state of public web services are seldom. To our best knowledge, the following studies are known somewhat similar to our work: [1], [2], [3], [4], [8], [9]. Compared to these studies, our study has several novel contributions as follows.

First, our study is much more large-scale and longitudinal in nature. While the largest number of web services studied by previous studies is 4,033 SOAP-based web services [4], each week, we trace about 16,000 SOAP-based web services for 48 weeks (i.e., $16,000 \times 48$). In addition, we also closely trace about 2,500 RESTful web services described in M_{wadl} and M_{text} methods once a week for 36 weeks (i.e., $2,500 \times 36$). Table I compares existing SOAP-based web service studies in recent years. Table II, on the other hand, shows the scales of our research data sets.

Second, unlike previous works, our study has been carried out at five different geographic locations to be able to study location-dependent properties of web services (e.g., response time and throughput). For instance, [8] studied QoS aspects of public web services using a snapshot of data for one time. Such a survey is hard to detect any interesting location-specific patterns of web services.

Third, unlike all previous works that solely focus on SOAP-based web services, we study RESTful web services as well. Due to increasing popularity of RESTful web services (as demonstrated by its adoption by many major Internet companies), we believe the longitudinal analysis on RESTful web services becomes ever more important.

⁶<http://ec2.amazonaws.com/doc/2010-11-15/AmazonEC2.wsdl>

⁷<http://aws.amazon.com/documentation/elasticmapreduce/>

⁸<http://www.flickr.com/services/api/>

⁹<http://msdn.microsoft.com/en-us/library/dd179355.aspx>

¹⁰<http://msdn.microsoft.com/en-us/library/ee460799.aspx>

¹¹<http://developers.facebook.com/docs/>

¹²<http://dev.twitter.com/doc>

Table II
SUMMARIES OF SIX DATA SETS (# OF WEB SERVICES PER WEEK BELOW VARIES AND IS MARKED AS \approx).

Data Set	Start date	Weeks	Weekly Size	Total Size
BIG_{wsdl}	2010/11/22	48	$\approx 15,803$	$\approx 758,568$
BIG_{wadl}	2011/02/14	36	≈ 209	$\approx 7,525$
BIG_{text}	2011/05/08	24	$\approx 2,386$	$\approx 57,269$
QoS_{wsdl}	2011/01/31	38	1,548	1,548
QoS_{wadl}	2011/02/28	34	122	122
QoS_{text}	2011/04/10	28	100	100
				$\approx 825,132$

Finally, compared to the latest survey on web services (e.g., [1], [4]), we have identified several changes. For instance, while 89% of web services have less than 10 operations in [1], the ratio from our study decreases to about 75%. In 2004, 70% of web services adopted document style and 30% of them use RPC style. However, our study found that the ratio of document style grows to around 88% now.

III. DESIGN OF MEASUREMENTS

A. Data Sets

To cover three methods (i.e., M_{wsdl} , M_{wadl} , and M_{text}) to describe two types (i.e., SOAP-based and RESTful) of web services, in this study, we have collected and prepared six data sets. First, BIG_{wsdl} , BIG_{wadl} , and BIG_{text} are large data sets of web services in M_{wsdl} , M_{wadl} , and M_{text} methods, respectively. Note that BIG_{wsdl} , BIG_{wadl} , and BIG_{text} data sets are *dynamic* in that weekly snapshots contain *different* set of web services with some overlaps.

- 1) BIG_{wsdl} contains mainly SOAP-based web services described in M_{wsdl} , collected from three major web service repositories—`xmethods.net`, `webservicelist.com` and `webservices.seekda.com`. We repeatedly take a weekly snapshot of all web services available from these three repositories for 48 weeks.
- 2) BIG_{wadl} contains RESTful web service descriptions in M_{wadl} . We first issue a query “`filetype:wadl`” to a search engine and collect all matching results. Then, we validate all extracted files to check if they are really WADL files or not. For instance, a `filefacebook.com/eva.wadl` is not a WADL file but a Facebook page of a person “Eva Wadl.” We repeated this data collection of WADL files once a week for 36 weeks.
- 3) BIG_{text} contains textual descriptions of RESTful web services that we weekly collected from `programmableweb.com` for 24 weeks.

Second, to test QoS values of web services, we prepared three smaller data sets and named them as QoS_{wsdl} , QoS_{wadl} , and QoS_{text} , respectively. Since QoS measurement often takes more time and resources, smaller data sets are preferred. Unlike dynamically changing BIG_{wsdl} , BIG_{wadl} , and BIG_{text} data sets, QoS_{wsdl} , QoS_{wadl} , and

QoS_{text} are *static* so that each week we use the *identical* set of web services to measure and compare QoS values.

- 1) QoS_{wsdl} is created based on the well-known public QWS data set¹³ from [4]. The original QWS data set used 2,507 SOAP-based web services to measure their QoS parameters such as throughput and response time. From these 2,507 web services of QWS data set, we fixed simple errors (e.g., from “http://” to “https://”), removed redundant or invalid ones, and at the end obtained 1,548 web services in the QoS_{wsdl} data set.
- 2) QoS_{wadl} contains 122 randomly chosen valid WADL files from BIG_{wadl} .
- 3) QoS_{text} contains top-100 RESTful services and their textual descriptions from BIG_{text} .

We load web services in QoS_{wsdl} , QoS_{wadl} , and QoS_{text} into the open-source tool SoapUI¹⁴. For each loaded service, then we randomly choose one operation therein, set input parameters appropriately, and generate a load test instance. Based on the test instance, next, we invoke the services to obtain QoS measurements (to be explained in Section III-B). In our load tests, we simulate that multiple users invoke each web service with the interval of 1 second. Each load test lasts for 60 seconds.

Table II summarizes statistics of six data sets used in the experiments. All data sets of Table II are publicly available for download at: <http://debs.ict.ac.cn/jiangwei/serviceSurvey/>

B. The QoS Parameters

The Quality-of-Service (QoS) of web services mainly concern the quality aspect of a web service, and include many parameters related with the availability, scalability, robustness, exception handling, accuracy, integrity, interoperability, and network-related QoS requirements¹⁵. In this paper, in particular, we classify QoS parameters into two groups: (1) *Standalone QoS parameters (S-QoS)* are those that can be measured without the need of invoking a web service. For instance, whether a service is consistent with the WSDL specification and WS-I profile can be measured by validating tools such as Apache-cxf `wSDLValidator`¹⁶ and BSP 1.1 test tool¹⁷, respectively; and (2) *Invocation-required QoS parameters (I-QoS)* are those that can be only obtained by actually invoking a service such as response time and throughput.

Standalone QoS parameters (S-QoS): First, we measure the *S-QoS* parameters in three data sets— BIG_{wsdl} , BIG_{wadl} , and BIG_{text} . For the description in the M_{wsdl} method (mainly used for SOAP-based web services), we

measure: (1) compliance with the WSDL specification, (2) compliance with the WS-I Basic profile, (3) documentation ratio—what percentage of words appear between `<wsdl:documentation>` and `</wsdl:documentation>` (since authors often use `<wsdl:documentation>` as a container for human readable description, the higher ratio may indicate the better readability of a WSDL file), (4) document size, (5) service name, (6) service URL (i.e., whether the URL of the WSDL file is specified), and (7) style ratio (i.e., the ratio of # of SOAP binding style over # of total WSDL documents). The WSDL SOAP binding can be either Remote Procedure Call (RPC) style or document style binding, and can also have an encoded use or a literal use, yielding four combinations—RPC/encoded, RPC/literal, document/encoded, and document/literal. The differences among these four types are illustrated in [?].

Second, for the description in the M_{wadl} method, we measure: (1) document size, (2) service name, and (3) service URL. Third, for the M_{text} method web services, we measure (1) document size, (2) service name, (3) service URL, (4) category, (5) mashup size (i.e., the number of mashups use the service), and (6) message format (i.e., different web services support different response message formats such as JSON or XML).

Invocation-required QoS parameters (I-QoS): For BIG_{wsdl} , BIG_{wadl} and BIG_{text} data sets, we do *not* measure *I-QoS*. For QoS_{wsdl} , QoS_{wadl} and QoS_{text} data sets, instead, we measure the following QoS parameters: (1) max response time (if response time takes more than 60 seconds, we consider the service failed), (2) min response time, (3) average response time (i.e., the average time for all the requests made), (4) throughput (i.e., total number of invocations for a given period of time), (5) successability (i.e., number of successful invocations out of total invocations), and (6) response size (i.e., the average size, in bytes, of response message of all responses).

C. Survey Periods and Locations

To incorporate the impact of locations where measurements are done (especially QoS aspects), we conduct our experiments at five different locations. First, using Amazon’s EC2 cloud computing environment, we launched four MS Windows Server 2008 (1.7 GB RAM and 1 ECU) at USA, Ireland, Japan and Singapore. Second, with a similar set-up, we prepared the fifth server at China.

Each week, our workflow is as follows: (1) at China, we collect a new snapshot of BIG_{wsdl} , BIG_{wadl} , and BIG_{text} data sets, (2) against the new weekly snapshots, each week at US and China, we measure *S-QoS* and *I-QoS*, and (3) (once every four weeks) against QoS_{wsdl} , QoS_{wadl} , and QoS_{text} data sets, at Japan, Ireland, and Singapore, we concurrently measure *I-QoS*.

¹³<http://www.uoguelph.ca/~qmahmoud/qws/>

¹⁴<http://www.soapui.org/>

¹⁵<http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>

¹⁶<http://cxf.apache.org/docs/wSDLValidator.html>

¹⁷<http://www.ws-i.org/deliverables/workinggroup.aspx?wg=testingtools>

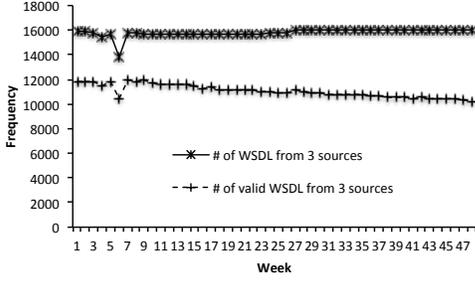


Figure 1. Change of # of WSDL files (BIG_{wsdl}).

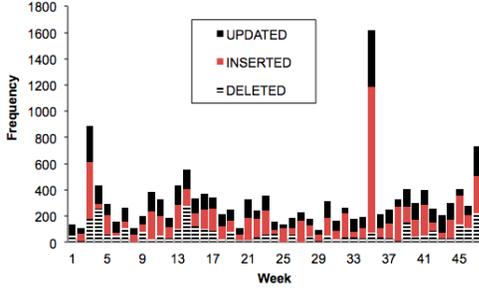


Figure 2. Change of edit operations to WSDL files (BIG_{wsdl}).

IV. RESULTS OF SOAP-BASED WEB SERVICES

A. Basic Analysis

First, we look at the longitudinal observations of web services. Figure 1 shows the weekly change of the number of SOAP-based web services described in M_{wsdl} method using BIG_{wsdl} data set. Throughout the weeks, the number of URLs collected in BIG_{wsdl} remains roughly 16,000 while the actual number of WSDL documents (i.e., valid URLs) that were retrieved is around 12,000 initially and 11,000 toward the end. The sudden drop at the week 6 is due to the network congestion occurred in Beijing, China, and considered as an outlier. Since the WSDL files in BIG_{wsdl} are drawn from three well-known web service repositories where human editors *manually* create the listing of web services, overall, the number of WSDL files is small and does not change dramatically.

On the other hand, when we estimate the number of WSDL files using Google with “filetype:wsl”, as of Jan. 2012, it is around 229,000, showing a large gap from 16,000 in BIG_{wsdl} . Further, the change of monthly numbers of newly indexed WSDL files during 2011 (estimated with “filetype:wsl daterange:xx-yy”) is captured in Table III. One can see the number is comparably small but still increasing slowly. For each two consecutive weeks, in Figure 1, 10,155–11,823 of WSDL files were common (out of roughly 12,000 valid WSDLs). Using these 10,155–11,823 of common WSDL files over consecutive weeks, Figure 2 traces *edit* operations across common web services—i.e. how many WSDL files have at least one inserted, deleted, or updated operation. Note that SOAP-based web services are dynamically changing.

With respect to encoding methods of M_{wsdl} method,

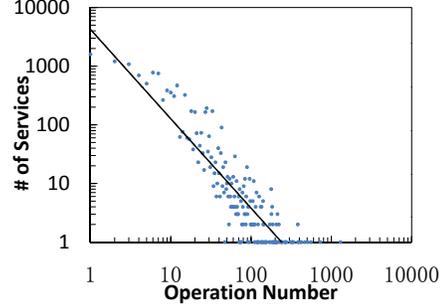


Figure 3. # of operations per web service in log-log plot (BIG_{wsdl}).

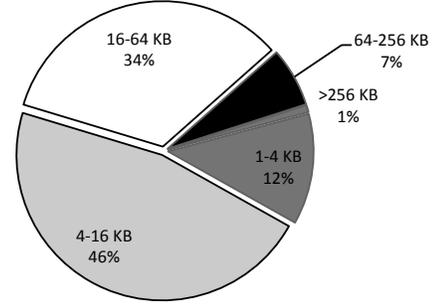


Figure 4. Distribution of WSDL file sizes (BIG_{wsdl}).

majority (87%–88%) of WSDL documents belong to document/literal encoding type. This percentage is much bigger than 70% of the study [1] done in 2004. Figure 3 shows the distribution of the number of operations per web service (using the latest weekly snapshot of BIG_{wsdl} data set). Note that the distribution roughly fits to a Power-Law function of $y = Cx^{-\alpha}$ (in log-log plot) with $\alpha = 1.52$. [9] reported a similar result with $\alpha = 1.49$. Figure 4 shows the break-down in terms of the WSDL file sizes. About 46% of WSDL files have the size of 4–16 KB, and 33% have 16–64 KB. That is, majority of WSDL files in BIG_{wsdl} data set is relatively small (4–64 KB). In our study, while 43% of WSDL files are less than 10 KB, 74% of them were in 2006 [3].

B. QoS Analysis

First, the documentation ratio (i.e., a fraction of documentation within a WSDL file) of WSDL files in the QoS_{wsdl} data set was examined. According to Figure 5, still, documentation ratios are relatively low. 57% of all WSDL documents have less than 2% of documentation ratios. Therefore, without extra information, WSDL files

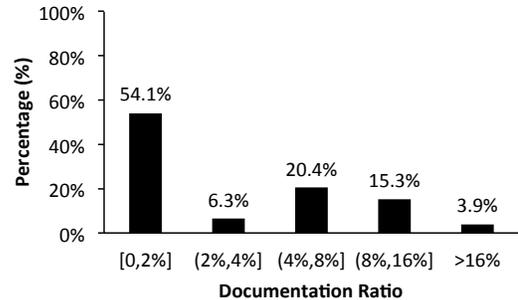


Figure 5. Documentation ratios of WSDL files (QoS_{wsdl}).

Table III
OF MONTHLY NEWLY-INDEXED WSDL FILES IN GOOGLE IN 2011.

Month	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
#	75	30	75	83	38	47	111	124	60	107	108	138

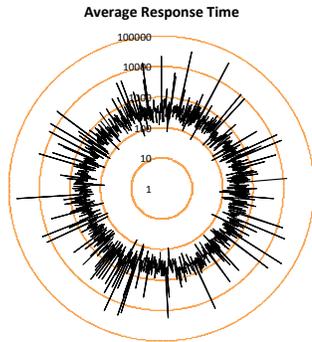


Figure 6. Distribution of average response time (QoS_{wSDL} @ US).

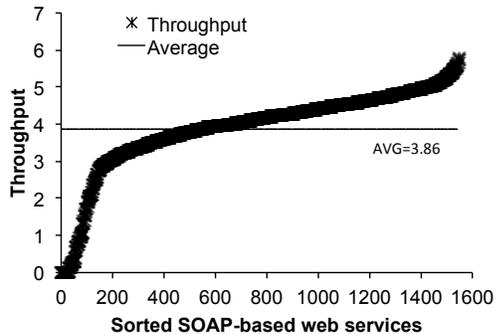


Figure 7. Distribution of throughput (QoS_{wSDL} @ US).

themselves are arguably hard to use. On the other hand, RESTful web services usually have extra information such as descriptions or examples on the Web, making them easier to use than SOAP-based web services.

Second, we examined whether or not these SOAP-based web services are compliant with W3C standards such as the WSDL Specification and WS-I basic profile. Around 71% of WSDL documents are fully consistent with the WSDL Specification, while about 51% of them are fully consistent with the WS-I profile. Although these standards were put forward years ago, there are still many WSDL web services that are not fully compliant with them. On the contrary, this also suggests that standards-compliant WSDL documents are arguably difficult to write.

Finally, weekly failure rates of web services in QoS_{wSDL} ranged from 1% to 7%—we considered a web service that did not respond within 60 sec “failed.” This indicates that the quality of web services vary greatly. For those that responded in time, their average response time are distributed as shown in Figure 6 (unit is ms). On the other hand, Figure 7 illustrates the throughput (i.e., # of successful invocation in a second) of all 1,548 web services in QoS_{wSDL} data set. The average throughput was 3.86. To see the pattern

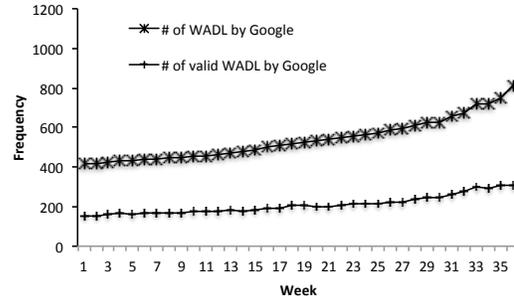


Figure 8. Change of # of RESTful services in M_{wadl} method (BIG_{wadl}).

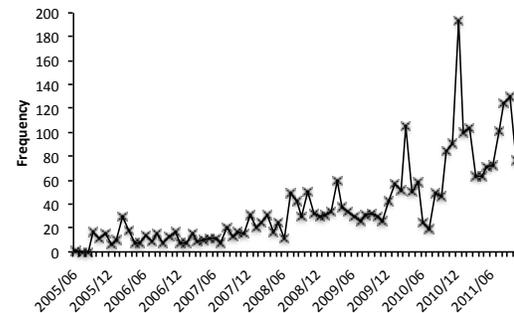


Figure 9. Change of # of RESTful services in M_{text} method (BIG_{text}).

more clearly, 1,548 SOAP-based web services are sorted in ascending order w.r.t their throughput on X-axis.

V. RESULTS OF RESTFUL WEB SERVICES

A. Basic Analysis

RESTful web services can be described by both M_{wadl} and M_{text} methods. First, the BIG_{wadl} data set contains 7,525 WADL files (a weekly download of 209 WADL files for 36 weeks), whose size is about 1/80 of that of WSDL files in BIG_{wSDL} data set. Figure 8 shows the weekly measurement of the number of WADL files returned by Google and valid ones therein. As to RESTful web services described in M_{text} method, due to its recent popularity, the data set grows faster than WADL-based RESTful web services grow. Figure 9 shows the trace of monthly increment of the number of RESTful services in M_{text} method. Note that the data goes back to as early as 2005 since all RESTful services in BIG_{text} have timestamps of their publication.

Figure 10 shows the histogram of mash-up sizes of RESTful services from the BIG_{text} data set. The mash-up size of a web service W is the number of other web services that use W as one of sources. That is, the bigger the mash-up size is, the more popular W becomes. While 75% of RESTful services in the BIG_{text} data set are currently not being utilized, 16% of them are being mashed-up by

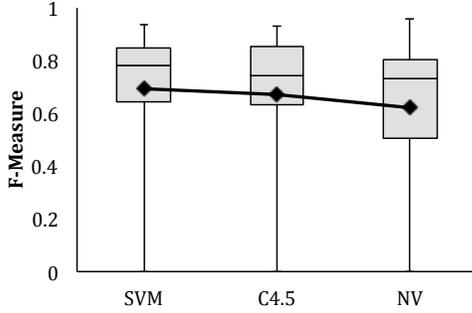


Figure 15. A comparison of three classifiers w.r.t. F-measure (BIG_{text}).

Table V
TOP-10 COUNTRIES W.R.T. % OF WEB SERVICES DISTRIBUTION.

Rank	M_{wsdl}		M_{wadi}		M_{text}	
	Country	%	Country	%	Country	%
1	US	37.9%	US	62.0%	US	41.9%
2	UK	15.9%	UK	7.7%	UK	25.5%
3	Germany	8.2%	Germany	7.7%	Germany	16.4%
4	France	3.6%	France	6.7%	Spain	5.3%
5	Australia	3.1%	Chile	3.0%	Canada	2.6%
6	Netherlands	2.9%	Spain	2.0%	France	1.3%
7	Spain	2.6%	Australia	1.7%	Denmark	1.0%
8	Canada	2.5%	Canada	1.7%	Netherlands	0.6%
9	Denmark	2.3%	Netherlands	1.0%	Finland	0.6%
10	China	2.0%	Czech Rep.	1.0%	Australia	0.4%

Figure 15 shows the comparison of three classification algorithms w.r.t. F-measure—i.e., the harmonic mean of precision and recall scores. 59 categories of web services were used as data points to show the box-and-whisker plot. Despite the challenge, three classical algorithms achieve relatively good mean F-measure scores: 0.692 for SVM, 0.672 for C4.5, and 0.621 for NV. Further details of top-10 and bottom-10 categories of the best performing approach (SVM) appear in Table IV.

VI. IMPACT OF LOCATIONS OF WEB SERVICES

In this section, we investigate the impact of “location” on the properties of web services. First, for each downloaded web service, we identified its corresponding country information (by tracing its URL using the SEO tool¹⁹). According to this identification, then, table V lists top-10 countries that have the largest shares of web service distributions in three methods. For all three methods, note that top-3 countries remain the same—i.e., United States dominates the rankings, being the #1 country with the most number of web services, while UK and Germany follow next. The combined shares of three countries range from 62% to 83.8%, indicating that the adoption of web services in other countries is much slower.

QoS aspects of web services are inherently influenced by the location of clients and servers. To see this impact first hand, we ran our experiments from five world-wide locations (i.e., US, Ireland, Japan, Singapore, and China). For instance, Figure 16 shows the number of failed web service invocations from five locations with BIG_{text} data set. In general, the quality of web services from BIG_{text} is

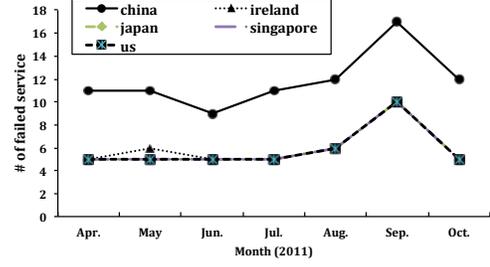
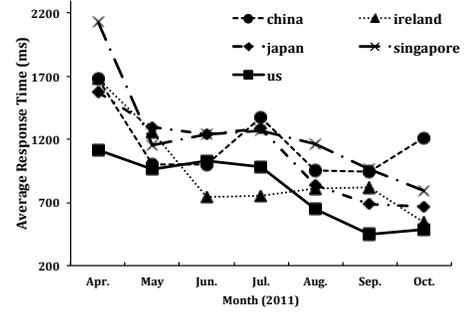
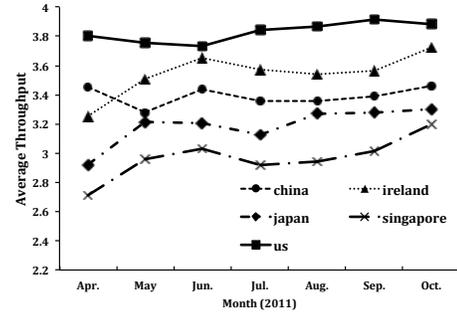


Figure 16. # of failed invocations from five locations (BIG_{text}).



(a) AVG response time



(b) AVG throughput

Figure 17. QoS results from five locations (BIG_{text}).

very high, yielding only a small fraction of failed invocations from all five locations (compared to the failure rate of 17% in BIG_{wadi}). In particular, note that there are more # of failed services measured in China than other locations. This is because the access to some web sites (e.g., Twitter and Facebook) are sometimes blocked in China.

Next, seven-month-long average response time and throughput of RESTful web services invoked from five locations are shown in Figure 17. With respect to response time, in Figure 17(a), invocations from US and Ireland appear to provide the fastest service. Since majority (i.e., 83.8%) of RESTful web services are originated from three countries—US, UK, and Germany (refer to Table V), the lower average response time for requests made from either US or Ireland makes sense. Reversely, invocations made from either China or Singapore had much higher average response time than US or Ireland. Average throughput from five locations is shown in Figure 17(b), showing the symmetric pattern from Figure 17(a)—i.e., while US and Ireland show the highest throughput, Singapore shows the lowest. Results for other

¹⁹http://www.selfseo.com/find_ip_address_of_a_website.php

Table IV
TOP-10 AND BOTTOM-10 CATEGORIES (OUT OF 59) W.R.T. F-MEASURE USING SVM.

Rank	Category	F-measure	Recall	Precision	ROC Area	#	%
1	Government	0.938	0.938	0.938	0.998	145	4.28%
2	Transportation	0.932	0.906	0.96	0.999	53	1.57%
3	Weather	0.914	0.889	0.941	1	18	0.53%
4	Real Estate	0.909	0.926	0.893	1	27	0.8%
5	Travel	0.894	0.873	0.917	0.994	63	1.86%
6	Financial	0.888	0.853	0.925	0.995	116	3.43%
7	Music	0.887	0.883	0.891	0.991	111	3.28%
8	Answers	0.88	0.846	0.917	1	13	0.38%
9	Payment	0.879	0.922	0.839	0.999	51	1.51%
10	Video	0.869	0.912	0.83	0.997	102	3.01%
...							
50	Wiki	0.476	0.455	0.5	0.998	11	0.32%
51	Tagging	0.471	0.364	0.667	0.908	11	0.32%
52	Other	0.454	0.496	0.419	0.783	135	3.99%
53	Blog Search	0.4	0.286	0.667	0.99	7	0.21%
54	Widgets	0.333	0.25	0.5	1	4	0.12%
55	Media Management	0.294	0.278	0.313	0.951	18	0.53%
56	News	0.277	0.225	0.36	0.881	40	1.18%
57	Retail	0	0	0	0.978	4	0.12%
58	Media Search	0	0	0	0.992	3	0.09%
59	Goal Setting	0	0	0	0.999	2	0.06%
AVG of 59 categories		0.692	0.683	0.713	0.977	3385	100%

data sets and methods show similar pattern and are omitted.

VII. CONCLUSION

Using a total of 825,132 real SOAP-based and RESTful web services, in this paper, we presented a comprehensive survey on diverse characteristics of web services. Unlike existing works, our experiments were repeated from five world-wide locations for almost a year. From our large-scale longitudinal analysis, we found that:

- Web services and their properties are dynamically changing. While both numbers of SOAP-based and RESTful services increase steadily, due to recent popularity, the number of RESTful services (described in M_{text} method) appears to increase more rapidly. Among three methods to describe web services (i.e., M_{wsdl} , M_{wadl} , and M_{text}), the following orders are found: w.r.t. usage scale, $M_{wsdl} > M_{text} > M_{wadl}$, w.r.t. growth rate, $M_{text} > M_{wsdl} \approx M_{wadl}$, w.r.t. standardization, $M_{wsdl} > M_{wadl} > M_{text}$, and w.r.t. service stability, $M_{text} > M_{wsdl} > M_{wadl}$.
- The adoption of web services across countries varies greatly. Three countries (i.e., US, UK, and Germany) produce majority of web services (i.e., 62%–83.8%) of the world. In turn, this skewness affects the QoS measures such as response time and throughput. When service requests are made from US or Europe, due to the physical proximity to the service locations, one tends to attain better response time and throughput.
- Using conventional classification algorithms (i.e., NV, C4.5, and SVM), we were able to predict the right category of each RESTful web service (out of 59 categories) with the average F-measure scores of 0.621–0.692. While there is a much room for improvement,

we leave the investigation for future work.

ACKNOWLEDGEMENT

Part of work by Wei Jiang was carried out when he visited Penn State in 2010. Dongwon Lee is in part supported by NSF DUE-0817376 and DUE-0937891 awards and Amazon web service research grant (2010, 2011). Wei Jiang and Songlin Hu are supported by the National Natural Science Foundation of China under Grant No.61070027, the Beijing Natural Science Foundation project under Grant No.4092043, the Science and Technology Project of State Grid Information & Telecommunication Company Ltd. under Grant No.SGIT[2011]567 and the Planned Science and Technology Project of Guangdong Province, China under Grant No.2010B050100009.

REFERENCES

- [1] S. M. Kim and M. C. Rosu, "A survey of public web services," in *WWW*, 2004, pp. 312–313.
- [2] J. Fan and S. Kambhampati, "A snapshot of public web services," *SIGMOD Rec.*, vol. 34, no. 1, pp. 24–32, 2005.
- [3] Y. Li, Y. Liu, L. Zhang, G. Li, B. Xie, and J. Sun, "An exploratory study of web services on the internet," in *ICWS*, July 2007, pp. 380–387.
- [4] E. Al-Masri and Q. H. Mahmoud, "Investigating web services on the world wide web," in *WWW*, 2008, pp. 795–804.
- [5] Y. Jianjun, G. Shengmin, S. Hao, Z. Hui, and X. Ke, "A kernel based structure matching for web services search," in *WWW*, 2007, pp. 1249–1250.
- [6] X. Dong, A. Halevy, J. Madhavan, E. Nemes, and J. Zhang, "Similarity search for web services," in *VLDB*, 2004, pp. 372–383.
- [7] W. Nam, H. Kil, and D. Lee, "On the computational complexity of behavioral description-based web service composition," *Theo. Computer Science (TCS)*, vol. 412, pp. 6736–6749, 2011.
- [8] E. Al-Masri and Q. H. Mahmoud, "Qos-based discovery and ranking of web services," in *ICCCN*, 2007, pp. 529–534.
- [9] H. Kil, S.-C. Oh, E. Elmacioglu, W. Nam, and D. Lee, "Graph theoretic topological analysis of web service networks," *World Wide Web J.*, vol. 12, pp. 321–343, 2009.