



## INVESTIGATING QoS OF REAL – WORLD WEB SERVICES

Sivanesh R, GowthamP,Surya Raj M, Prakash P  
Department of Information Technology  
Dr. Mahalingam College of Engineering and  
Technology,Pollachi-642003  
Email: bssivanesh96@gmail.com

Mr. J. Dhyaneswaran M.E.,  
Assistant Professor  
Department of Information Technology  
Dr. Mahalingam College of Engineering and  
Technology,Pollachi – 642003

### ABSTRACT

This paper is entitled as “Investigating Quality of Service of Real World Web Services” Front End: PHP, Back End: MYSQL aimed at developing a Web service to provide service for the Public with best Quality. This is a Web based application that can be accessed throughout the world. This system can be used to automate the workflow of service requests for the various people.

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, 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 these 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

This is one integrated system that covers different kinds of facilities like Ambulance, Hospitals, Hostels, Police Station, Railway Station, Bus Stand, Temples, Banks, Schools and colleges, Post office, Couriers, Cinema Theatres etc. also provide details are latest news, rates of vegetable, fuel, jewelry, bus timing, etc. This Latest news contains news of local, national and international. Daily updates vegetables rate, fuel rate, jewelry rate and bus timing. User can search jobs in a particular locality. After getting their requested information, they can post their testimonials. User can share their festival videos, festival images, information to their friends, relatives, public by using social network.

### 1. INTRODUCTION

Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP). As described by the World Wide Web Consortium (W3C), web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks. Web services are characterized by their great interoperability and extensibility, as well as their machine-processable descriptions, thanks to the use of XML. Web services can be combined in a loosely coupled way to achieve complex operations. Programs providing simple services can interact with each other to deliver sophisticated added-value services. A Web Service is a standards-based, language-agnostic software entity that accepts specially formatted requests from other software entities on remote machines via vendor and transport neutral communication protocols producing application specific responses. Standards based Language agnostic formatted requests Remote machines Vendor neutral Transport neutral Application specific responses.



## ***1.1 BENEFITS AND TYPES OF WEB SERVICES***

Benefits of Web Services Loosely Coupled Each service exists independently of the other services that make up the application. Individual pieces of the application to be modified without impacting unrelated areas. Ease of Integration Data is isolated between applications creating 'silos'. Web Services act as glue between these and enable easier communications within and across organizations. Service Reuse Takes code reuse a step further. A specific function within the domain is only ever coded once and used over and over again by consuming applications.

The simplest Web service system has two participants: A service producer (provider) A service consumer (requester). The provider presents the interface and implementation of the service, and the requester uses the Web service.

On the conceptual level, a service is a software component provided through a network-accessible endpoint. The service consumer and provider use messages to exchange invocation request and response information in the form of self-containing documents that make very few assumptions about the technological capabilities of the receiver.

On a technical level, web services can be implemented in various ways. The two types of web services discussed in this section can be distinguished as “big” web services and “RESTFUL” web services.

### ***1.2 BIG WEB SERVICES***

In Java EE 6, JAX-WS provides the functionality for “big” web services. Big web services use XML messages that follow the Simple Object Access Protocol (SOAP) standard, an XML language defining a message architecture and message formats. Such systems often contain a machine-readable description of the operations offered by the service, written in the Web Services Description Language (WSDL), an XML language for defining interfaces syntactically.

The SOAP message format and the WSDL interface definition language have gained widespread adoption. Many development tools, such as Net Beans IDE, can reduce the complexity of developing web service applications.

A SOAP-based design must include the following elements. A formal contract must be established to describe the interface that the web service offers. WSDL can be used to describe the details of the contract, which may include messages, operations, bindings, and the location of the web service. You may also process SOAP messages in a JAX-WS service without publishing a WSDL.

The architecture must address complex non-functional requirements. Many web service specifications address such requirements and establish a common vocabulary for them. Examples include transactions, security, addressing, trust, coordination, and so on.

The architecture needs to handle asynchronous processing and invocation. In such cases, the infrastructure provided by standards, such as Web Services Reliable Messaging (WSRM), and APIs, such as JAX-WS, with their client-side asynchronous invocation support, can be leveraged out of the box.

### ***1.3 RESTFUL WEB SERVICES***



[www.ijirp.com](http://www.ijirp.com)

**International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)**

**ISSN: 2394-6364, Volume – 2, Issue – 3, March 2016**

In Java EE 6, JAX-RS provides the functionality for Representational State Transfer (Restful) web services. REST is well suited for basic, ad hoc integration scenarios. Restful web services, often better integrated with HTTP than SOAP-based services are, do not require XML messages or WSDL service-API definitions.

Jersey is the production-ready reference implementation for the JAX-RS specification. Jersey implements support for the annotations defined in the JAX-RS specification, making it easy for developers to build Restful web services with Java and the Java Virtual Machine (JVM).

Because Restful web services use existing well-known W3C and Internet Engineering Task Force (IETF) standards (HTTP, XML, URI, MIME) and have a lightweight infrastructure that allows services to be built with minimal tooling, developing Restful web services is inexpensive and thus has a very low barrier for adoption. You can use a development tool such as Net Beans IDE to further reduce the complexity of developing Restful web services.

A Restful design may be appropriate when the following conditions are met. The web services are completely stateless. A good test is to consider whether the interaction can survive a restart of the server. A caching infrastructure can be leveraged for performance. If the data that the web service returns is not dynamically generated and can be cached, the caching infrastructure that web servers and other intermediaries inherently provide can be leveraged to improve performance. However, the developer must take care because such caches are limited to the HTTP GET method for most servers.

The service producer and service consumer have a mutual understanding of the context and content being passed along. Because there is no formal way to describe the web services interface, both parties must agree out of band on the schemas that describe the data being exchanged and on ways to process it meaningfully. In the real world, most commercial applications that expose services as Restful implementations also distribute so-called value-added toolkits that describe the interfaces to developers in popular programming languages.

Bandwidth is particularly important and needs to be limited. REST is particularly useful for limited-profile devices, such as PDAs and mobile phones, for which the overhead of headers and additional layers of SOAP elements on the XML payload must be restricted.

Web service delivery or aggregation into existing web sites can be enabled easily with a RESTful style. Developers can use such technologies as JAX-RS and Asynchronous JavaScript with XML (AJAX) and such toolkits as Direct Web Remoting (DWR) to consume the services in their web applications. Rather than starting from scratch, services can be exposed with XML and consumed by HTML pages without significantly refactoring the existing web site architecture. Existing developers will be more productive because they are adding to something they are already familiar with rather than having to start from scratch with new technology.

## **2. RELATED WORK**

The survey describes various approaches and frameworks that have been developed in order to provide widely usable web service composition platforms. The aim of this section is to give a summary of what we have presented so far.

SchahramDustdarand WolfgangSchreiner deal with many different approaches and frameworks that have been developed in order to provide widely usable web service composition platforms. The aim of this section is to give a summary of what we have presented so far. This is achieved by creating a table, listing most of the frameworks that have been discussed in this paper, mapping them to the corresponding composition approach, and listing some of the most important features that were identified to categorize the composition frameworks. Some approaches, such as entries do not meet the requirements listed in the Table 1, since they do not provide a development platform for composite web services, but rather propose language or protocol specifications. By the analysis of the composition frameworks, some modules and



support features that are of great importance for designing and developing composite services have been identified. Those include an execution monitor to monitor and trace service execution, dynamic service selection, which is a profound part of service composition and may be used to choose appropriate service conversation protocols from a conversation repository, QoS modeling and evaluation, transaction support, and the possibility to graphically modeling service flows.

According to a recent McKinsey report[5], in 2010, enterprises and users stored more than 13 exabytes of new data. The potential value of global personal location data is estimated to be \$700 billion to end users, and it can result in an up to 50% decrease in product development and assembly costs. McKinsey also predicts an equally great effect of Big Data in employment, where 140,000-190,000 workers with "deep analytical" experience will be needed in the US; furthermore, 1.5 million managers will need to become data-literate [3]. EMC estimates that about 15 percent of all IT spending will move to the as-a-Service delivery models by 2015 and will grow to about 35 percent by 2021. The Big Data-as-a-Service market size is about 2.25 billion by 2015 and about 30 billion by 2021. In other words, about 4 percent of all IT spending will go into Big Data-as-a-Service by 2012 [4]. There is strong business motivation of Big Data-as-a-Service. Based on the ownership of big data, business models of Big Data-as-a-Service can be divided into the following two types:

(1) The owner of big data conducts data storage, management, and analysis and provides Web APIs for users to access the data or the analyzed results. For example, Google crawls a huge volume of data, conducts the data management and analysis, and provides various types of APIs with different functionality (e.g., APIs for Webpage search, APIs for map services, etc.) to users.

(2) Due to reasons such as cost reduction or lack of professional knowledge, the owner of big data outsources the big data processing (or part of it) to a third-party. It consumes the Big Data-as-a-Service provided by third-party and allows the service provider to work on it to extract values. The service providers typically charge users on a utility computing basis, i.e., the cost reflects the amount of resources allocated and consumed.

### **3. PROPOSED SYSTEM**

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, 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 web services is assessed by distributed service users. In the second evaluation, response time and throughput of web services are evaluated by distributed service users. And in the third evaluation, QoS changing of web services with time is studied by conducting web service invocations by users in time slots with a time interval of 15 minutes.

#### **3.1 ADVANTAGES OF PROPOSED SYSTEM**

Providing detailed analysis and discussions on the relationship between QoS values and time. 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.

Generating Client Side Web Service Invocation, Generating Failure messages, Empty File failures are caused by the fact that the WSDL files are empty.

Generating Invalid File Format failures are due to that these WSDL files do not follow standard WSDL format; and Error Parsing failures are caused by syntax errors of WSDL files.



The generation of this failure indicates Web Service Description Language files on the Internet are fragile, which may contain empty content, invalid format, invalid syntax, and other various types of errors. Analyzing and identifying error report will help in providing best quality of service.

TABLE 1
WSDL File Download Failures

Table with 4 columns: Codes, Descriptions, # WS, Percents. Rows include error codes like 400 (Bad Request), 401 (Unauthorized), 403 (Forbidden), 404 (File Not Found), 405 (Method Not Allowed), 500 (Internal Server Error), 502 (Bad Gateway), 503 (Service Unavailable), 504 (Gateway Timeout), 505 (HTTP Version Not Support), N/A (Connection Timed Out, Read Timed Out, Unknown Host, Redirected Too Many Times), and a Total row.

TABLE 2
Java Code Generation Failures

Table with 3 columns: Failure Types, # WS, Percents. Rows include failure types like Empty File, Invalid File Format, Error Parsing WSDL, Invocation Target Exception, Null QName, Databinding Unmatched Type Exception, and a Total row.

4. CONCLUSION

One greater advantage of hosting web is to make the resource to know all and to become familiar which paves a way for development of organization as well as for profit. While hosting the software it should be attractive, user-friendly, and easier to access. This auction warehouse software constitutes all this characteristics.

5. REFERENCE

List of references including: 1.D.A. Menasce, "QoS Issues in Web Services" IEEE Internet Computing, Nov/Dec. 2002. 2.X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-Aware Web Service Recommendation and Visualization," IEEE Trans. Services Computing, 2011. 3.Z. Zheng, H. Ma, M.R. Lyu, and I. King, "QoS-Aware Web Service" IEEE Trans. Service Computing, Apr.-June 2011. 4.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. 5. 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. 6. Software engineering By Roger S. Pressman Fifth Edition, Mc.Graw Hill International Edition 2001. 7. System Analysis and Design Elais M. Awad, Second Edit Galgotia Publications (P) Ltd Published by Sunil Galgotia, 2002.