DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4349

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

PERFORMANCE EVALUATION OF RESTFUL WEB SERVICES AND SOAP / WSDL WEB SERVICES

Dr. Digvijaysinh Rathod
Institute of Forensic Science
Gujarat Forensic Sciences University
Gandhinagar, Gujarat (India)

Abstract: Service Oriented Architecture is style of software design for building software application that use services available in a network such as the web. RESTful HTTP and SOAP/WSDL services are two architectural styles for building web services, where SOAP/WSDL based services follows an operation centric approach and RESTful HTTP based services follows resource centric approach**.**These web services are well defined, self-contained in terms of its dependency on other web services components. Client has two choices for the purpose of communication with web services - Simple Object Access Protocol (SOAP) over HTTP and Representational State Transfer with HTTP. Today mobile devices and web services are very popular and choice of web service architectural style (SOAP/WSDL or REST) is very important, as mobile phone has physical constraint such as low processing speed, limited memory and slow intermits wireless connection. I have implemented a prototype systems and conducted extensive experiment with SOAP / WSDL web services and RESTful web services with same mobile client. The experiment results demonstrate that RESTful web services have much improved performance and scalability as compared to SOAP / WSDL web services.

Keywords: Web service, SOAP, RESTful, SOA, Android, WSDL, REST

1. INTRODUCTION

SOA is an architectural style Figure - 1 for building software applications that use services available in a network such as the web p **[1,2]**. It promotes loose coupling between software components, so that they can be reused. Applications in SOA are built based on services. A service is an implementation of well-defined business functionality, and such services can then be consumed by clients in different applications or business processes.

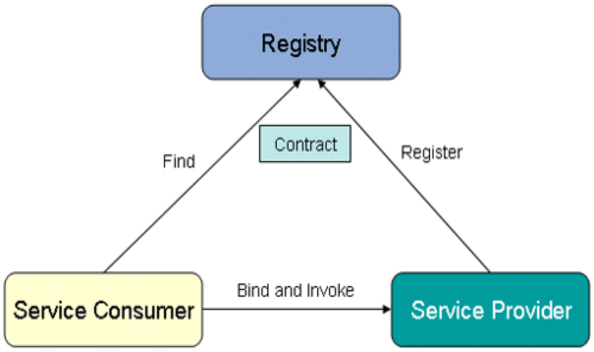

Figure-1 SOA Architecture

Web services have taken the concept of services introduced by Jini technology and implemented it as services delivered over the web using technologies such as XML, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Description, Discovery, and Integration(UDDI) [5].

2. REALIZING SOA WITH WEB SERVICES

Web services use XML to code and to decode data, and SOAP / WSDL to transport it (using open protocols). Besides these, HTTP, Web Services Description Language (WSDL), Universal Description, Discovery and Integration (UDDI), and SPARQL are the elements of Web Services. **[6,7]** Web services are software systems designed to support interoperable machine-to-machine interaction over a network. This interoperability is gained through a set of XML-based open standards, such as WSDL, SOAP, and UDDI. These standards provide a common approach for defining, publishing, and using web services. An increasing industry backlash to the mounting complexity of the WS-* SOAP specifications is resulting in many organizations considering RESTful Web Services. In fact, a recent poll by Information Week found that 58% of IT professionals believed that SOA introduced more complexity and resulted in cost overruns **[3,4]**.

A. Simple Object Access Protocol (SOAP)

SOAP is a simple XML-based protocol that allows communicating applications information over HTTP without the dependency of OS platform. SOAP uses HTTP and XML as the mechanisms for information exchange.

1) ***Universal Description, Discovery and Integration (UDDI)***

Universal Description, Discovery and Integration in short UDDI is a web based distributed directory like traditional phone book's yellow and white pages that enables businesses to list themselves on the Internet and discover each other. It defines a registry service – a Web service that manages information about service providers, service implementations, and service metadata – for Web services and for other electronic and non-electronic services. The service providers can use UDDI to advertise the services they offer while service consumers can use UDDI to discover services.

2) Web Services Description Language (WSDL)

The WSDL refers to Web Services Description Language, is an XML based protocol used for sending and receiving the information through decentralized and distributed environments. WSDL is an integral part of UDDI that was developed jointly by Microsoft and IBM **[8,9]**. It defines what services are available in its Web service and also defines the methods, parameter names, parameter data types, and return data types for the Web service. The WSDL document is quite reliable and applications that use web services accept it

B. Representational state transfer (REST)

REST or Representational State Transfer is based on a small set of widely-accepted standards such as Hypertext Transfer Protocol (HTTP), Uniform Resource Identifier (URI), and Extensible Markup Language (XML). REST requires far fewer development steps, toolkits, and execution engines than SOAP. Conceptual model of RESTful shows in Figure-2.

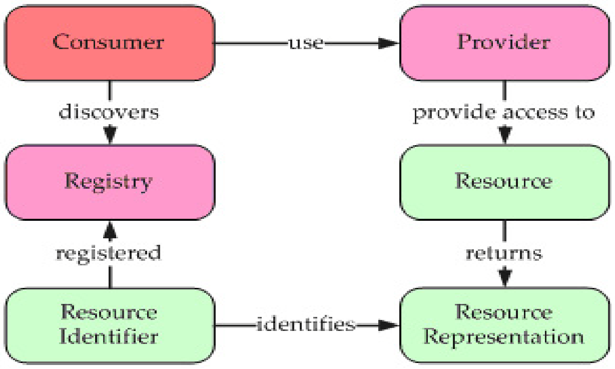

Figure-2 RESTful Conceptual Model

A RESTful web service provides access to a resource, identified by UR using HTTP. A resource is an abstraction of information. The HTTP methods, PUT, GET, POST, and DELETE define a uniform interface for accessing resources (i.e. **C**reate, **R**etrieve, **U**pdate, **D**elete or CRUD) [12]. HTTP is stateless request-response application protocol. Request and response messages are comprised of a command (method), a header, and a body.

3. BENCHMARK APPLICATION DETAIL

To evaluate the performance of RESTful web services with conventional SOAP / WSDL web services with mobile client presented in Figure-1. I developed prototype model which provide three types of services namely view student's information, image upload and integer array addition. Mobile client can use RESTfull or SOAP / WSDL web services to access these services. I illustrated performance evaluation configuration and emulator configuration in the next section.

4. PERFORMANCE EVALUATION CONFIGURATION

I developed prototype application using java programming language and deployed it on Glassfish web server. Glassfish web services container run on HP ProBook 4530s with 2.1 GHz 3i processor, 4 GB RAM and windows 10 operating system. Mobile client is developed using Android SDK 2.2. Vender of Android is Android Open Source Project. Android is a software stack for mobile devices that includes an operating system, middleware and key applications [13,14]. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. I created Android Virtual Device (AVD) with target Android 2.2, platform Android SDK 2.2 and API level is 8.0 [10,11].

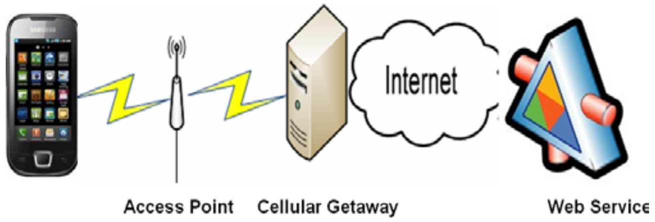

Figure-3 Setup for SOAP and RESTful web services with mobile client

A. Measurement Methodology

I measured the total message exchange time of both RESTfull web services and the conventional SOAP / WSDL webs services and compare them to evaluate the performances. I define the total message exchange time as the summation of sending request from android mobile client and receiving response from Glassfish web service container for RESTful and SOAP / WSDL web service. I measured total message exchange times for 50 to 100 repetitions varying the number of messages per stream (the first independent variable). I also vary the size of messages (the second independent variable). The resolution of the Java timer, MIDP 2.0/CLDC 1.1's System.currentTimeMillis(), used is 10 milliseconds

B. Performance evaluation implementation

I implemented three benchmark application and measure average response time (millisecond) and message size (byte) carried by the request for both RESTful and SOAP / WSDL web services for same experiment. Benchmark web services are discussed bellow:

1) View student's information services

The first benchmark web service is view student's information; which will provide the information of students like student id, student name, student address etc. Through mobile client we have to enter the number of student whose information, we want to extract from database. I categorized student information in two categories: Light information and Heavy Information. Light student information includes fixed length data size like Integer, Float, Char and Heavy student information includes fixed length data and Variable length data like Varchar data type. To measure the size of the record we used following formula [15,16]

$\mu = \alpha + \beta + £ + 4$
μ is the Record size
α is the Fixed data size
β is the Variable Data size
£ is the Null Bitmap

α = Total byte size of all fixed-length

β= 2 + (Number of Variable columns x 2) + Maximum variable size

Number of Variable Columns = number of variable-length columns and maximum variable size = Maximum byte size of all variable-length columns

Null Bitmap = 2 + ((Number of Columns + 7) / 8)

Number of Columns = Total number of columns (fixed-length and variable-length)

I measured response time and message size for variable number of student for both RESTful and SOAP / WSDL web service.

B) Image uploading services

Second benchmark web service is image upload services in a form of byte array with variable size. The aim of this benchmark web services is to compare variable length uploading speed of conventional SOAP / WSDL web services with RESTful Web services. I uploaded 100 images of same size and of same array byte domain to measure average upload time.

C) Integer array addition services

Third benchmark web service is integer array addition that returns a summation of array of integer. I also changed size of array elements. In this experiment, I measured response time in milliseconds and message size in byte for variable length array.

5. PERFORMANCE ANALYSIS

Table-1 shows benchmarking result of view student's information web service which is depicted in **Figure-4(a) and Figure-4(b)**. It show that for variable number of student size a message size (byte) and response time of RESTful web service is much better than SOAP / WSDL web service.

Table-1 Web service response time (ms) and message size (byte) for view student's information service

Number of Students	**Message Size (byte)**		**Time (Millisecond)**	
	SOAP / HTTP	**REST (HTTP)**	**SOAP / HTTP**	**REST (HTTP)**
Light information of 1 Student	3370	374	29.01	13.33
Heavy information of 2 Student	16055	2079	63.82	26.51
Heavy information of 3 Student	29155	5281	100.06	43.38
Heavy information of 5 Student	37001	7566	233.68	86.82
Heavy information of 6 Student	50455	13273	286.57	117.58

Figure 4(a). Message size (byte) comparison of RESTful and SOAP web service for for view student's information services

Figure 4 (b) .Response time (ms) comparison of RESTful and SOAP web service for view student’s information services.

Table-2 shows benchmarking results of upload images web service depicted in **Figure-5**, which shows that time (ms) to upload image in byte array domain take less time for RESTful web service as compared to conventional SOAP / WSDL web service. I uploaded 100 images of variable length.

Table – 2 Web service responses Time (ms) for Image Upload (Byte Array) web service

Bytes Array (kb)	**Time (Milliseconds)**	
	SOAP / HTTP	**RESTful (HTTP)**
10	8760	3950
20	12800	9560
30	19870	15689
40	28050	23900

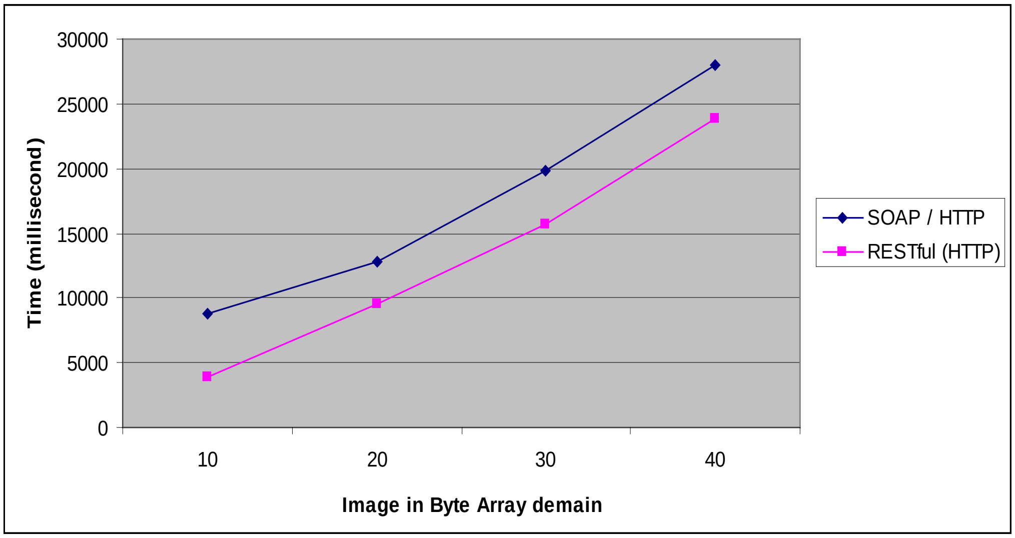

Figure 5 Response time (ms) comparison of RESTful and SOAP web service for Image Upload (Byte Array) web service

For third benchmark web service, benchmark result of integer array addition web service is listed in **Table-3** and depicted in **Figure-6(a) and Figure-6(b)**. It show that even for variable length integer array addition payloads of SOAP / WSDL web service is larger than RESTful web service for both GET/DELETE and PUT/POST varieties.

Table-3 Web service response time (ms) and message size (byte) for Integer Array addition web service

Number of Array Elements	**Message Size (byte)**			**Time (ms)**	
	SOAP / HTTP	**REST (HTTP)**		**SOAP / HTTP**	**REST (HTTP)**
10	1755	**195**		3905	1795
30	3710	480		8280	3440
60	5925	945		13200	5385
90	7524	1368		17442	6480
120	8860	1860		17500	7180
150	9964	2229		18750	10050
180	11093	2858		22140	9833

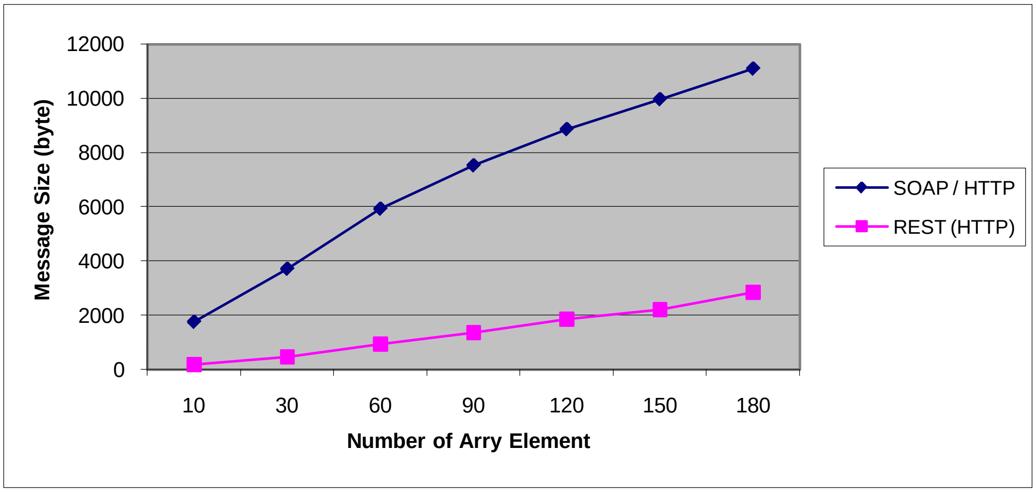

Figure 6(a) Message size (byte) comparison of RESTful and SOAP web service for Image Upload (Byte Array) web service

Figure – 6(b) Response time (ms) comparison of RESTful and SOAP web service for Image Upload (Byte Array) web service

6. CONCLUSION

In this paper, I presented benchmarking result of mobile based performance evaluation of RESTfull and SOAP / WSDL web services with the implementation of three different benchmark web services. It clearly shows that RESTful web service has better performance and scalability and REST provides true “Web services” based on URIs and HTTP. The analysis of the experimental results reveals the following findings

- REST is focused on accessing named resources through a single consistent interface.

- SOAP / WSDL is focused on accessing named operations, each implement some business logic through different interfaces.
- REST support verities of data formats such as XML and JSON while SOAP / WSDL support only XML.
- REST reads can be cached while SOAP / WSDL does not extends this feature
- Provides improved response times and server loading characteristics due to support for caching.
- Improves server scalability by reducing the need to maintain communication state.
- Requires less client-side software to be written than other approaches, because a single browser can access any application and any resource.
- Depends less on vendor software than mechanisms which layer additional messaging frameworks on top of HTTP.
- Does not require a separate resource discovery mechanism, due to the use of hyperlinks in content.
- Provides better long-term compatibility and resolvability characteristics than RPC. This is due to: The capability of document types such as HTML to evolve without breaking backwards- or forwards compatibility.
- The ability of resources to add support for new content types as they are defined without dropping or reducing support for older content types (MIME types).

7. REFERENCES

[1]. Fielding, R., Taylor, R.N.: Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology 2(2) (2006) 115{150)

[2]. Digvijasinh Rathod, Satyen Parikh, M. S. Dahiya, Goal-Based Constraint Driven Dynamic RESTful Web Service Composition Using AI Techniques. In Proceedings of International Conference on ICT for Sustainable Development (20156)- pp 745-754, DOI: 10.1007/978-981-10-0129-1_77.

[3]. Greenwood, D., Callisti, M. Engineering Web Service Agent Integration. In IEEE Conference of Systems, Man and Cybernetics, 2004. Available from http://www.whitestein.com/resources/papers/ieeesmc04.pdf.

[4]. Rathod, D. M., Parikh, S. M., & Buddhadev, B. V. (2013). Structural and behavioral modeling of RESTful web service interface using UML: Proceedings of 2013. In IEEE International Conference on Intelligent Systems and Signal Processing (ISSP) (pp. 28–33). doi:10.1109/ISSP.2013.6526869. ISBN:978-1-4799-0316-0.

[5]. Bematallah. B. Casati, F. Toumani, Representing, analyzing and managing web service protocols. Data ad Knowledge Engineering 58(3), 327-357 (2006)

[6]. Colombo M Di Nitto, E. Di Penta, M. Distante, D. Zuccala. Speaking a common language: A conceptual model for describing service-oriented systems. ICSOC 2005. LNCS, Vol.3826,pp. 48-60. Springer, Heidelberg (2005)

[7]. Nguyen, D K Van den Heuvel, W.J.Papazoglou, M P de Castro. A gap analysis methodology for engineering SOA-based applications.LNCS, Vol.5600,pp 293-318. Springer, Heidelberg (2009)

[8]. Apache AXIS2, http://ws.apache.org/axis2. Apache. Axis – Java implementation of the SOAP protocol. [Online]. Available: http://ws.apache.org/axis/.

[9]. Mani A. and Nagarajan A. "Understanding quality of service for Web services," http://www 106.ibm.com/developerworks/library/wsquality. html, 2002.

[10]. Sun Java Wireless Toolkit for CLDC: http://java.sun.com/products/sjwtoolkit.

[11]. http://developer.android.com/sdk/addingcomponents.html

[12]. Tian M, Voigt T, Naumowicz , Ritter H, Schiller J., "Performance Considerations for Mobile Web Services," Elsevier Computer Communications Journal, Volume 27, Issue 11 , Pages 1097-1105, 2004.

[13]. Beek, M., Bucchiarone, A. and Gnesi, S. Web Service Composition Approaches: From Industrial Standards to Formal Methods. Proceedings of Int'l Conf. On Internet and Web Application and Services (ICIW'07).2007 IEEE. 2007.

[14]. Milanovic, N. and Malek, M. Current solution for Web service composition. Proceedings of IEEE Internet Computing. November 2004. IEEE. 2004.

[15]. Srivastava, B. and Koehler, J. Web service composition: Current solutions and open problems. Proceedings of the ICAPS-2003 Workshop on Planning for Web Services. 2003. Trento, Italy.2003.

[16]. D. M. Rathod M. Dahiya S. M. Parikh "Towards composition of restful web services", 2015 6th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-6 2015.