

Performance Analysis of Web Services for Android based Devices

Amarpreet Singh Johal

Baba Banda Singh Bahadur Engineering College,
Fatehgarh Sahib, Punjab, India

Baljit Singh

Baba Banda Singh Bahadur Engineering College,
Fatehgarh Sahib, Punjab, India

ABSTRACT

Android mobile devices and Web services are becoming very popular. As android devices and wireless technologies continue to rapidly grow over a short span of period, Web services technology recognizes mobile computing as an area to which it should expand. Web service can greatly increase the functionality of Android devices to interact with its environment. In this paper, the performance is analyzed for two of the most important approaches used for building and implementing Web services for an android based mobile devices i.e. SOAP-based Web services and RESTful Web services. REST services is identified by unique URI's (Uniform Resource Identifier) and accessed and manipulated using set of predefined methods: GET, POST, PUT, DELETE on the other side SOAP is XML based protocol for the decentralized, distributed environment, that use the power of the Internet. The experimental results show that RESTful web services outperform SOAP web services.

Keywords

Web services, Android, SOAP, REST.

1. INTRODUCTION

Web services have been there since at least 1999, making them a relatively new technology that has gotten lots of press and praise. There is no secret behind Web services that will instantly make everything better or work together. The most important factor that corresponds to the success and popularity of Web services is the fact that its backbone is XML. Services are self contained, dynamic, modular applications that can be stated, published, located, or called over the network to create products, processes, and supply chains. These services can be local, distributed, or Web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, and XML. They do not provide the user with a GUI, instead they share business logic, data and processes through a programmatic interface over a network. Developers can then add the Web service to a GUI by parsing the content from the Web service and displaying accordingly on the device screen.

Web service can greatly increase the functionality of Android devices to interact with its environment as they are considered to be the backbone of an Android application. In today's world when there is the boom of the smart phones, the user's are using more and more applications in every sector such as health, entertainment, office, college, banking etc. The dynamic content of an application comes from the Web services and hence it is very important to use an appropriate Web service to enhance the performance of an application. When choosing a type of Web service for Android devices, a lot of aspects have to be taken into considerations since the various standards differ from each

other. For example, performance, speed, payload, bandwidth and memory usage, one standard could be better than others to use. Time pressure for the development could be crucial as well. Therefore, choosing a suitable standard to implement is important.

Android devices require a technology that connects mobile systems to a distributed computing environment. Web services may be the key solution for such system, since a strong interoperable feature is the key requirement of the technology. Thus, the integration of Android devices with Web Services technology will give many advantages to both sides as mobile devices enabled with Web services can be equal participant of Web services architectures (can be web service client or web service provider). There are mainly two approaches for building and implementing Web services for Android based operating system i.e. SOAP-based Web services and RESTful Web services. Applications for Android devices can be made using anyone of the technique but it is mainly dependent on client which approach to follow.

Jeffrey V. Nickerson, Keith D. Swenson [1] distinguishes the two approaches for developing Web services, i.e., REST and SOAP. The main difference focused in this paper is that SOAP consists of a tightly coupled design similar to RPC (Remote Procedure Call), and REST consists of a loosely coupled design similar to navigating Web links. REST is said to be inexpensive, simple and easier to extend than XML-RPC implementations can offer [2]. REST has also been used to explain the excellence and scalability of HTTP Web Service performance is became an important factor.[3] SOAP based web services produces considerable network traffic, high latency and the message size is also large this is not in the case of RESTful. The RESTful web services have better performance than SOAP based web services in wired and wireless communication network. The RESTful web services are lightweight, easy and self-descriptive with higher flexibility and lower overhead. Numbers of alternatives are available for SOAP and RESTful web service on principal level, conceptual level and technology level [4]. The evaluation of SOAP vs. REST was difficult because coding with REST was challenging. In comparing the two technologies, SOAP is well developed with much industrial support but REST is still striving to establish itself and its importance [5].

The goal of this paper is to analyze the performance of SOAP and RESTful web services by comparing the two approaches using the different methods. The below section depicts an overview of SOAP, REST services.

1.1 SOAP

SOAP is designed to be a new protocol for the decentralized, distributed environment that uses the power of the Internet and XML to pass typed information between nodes. All SOAP messages consist of a common structure to them and are enclosed with SOAP envelope. The structure uses a voluntary header section which is followed by the payload. The header contain details of the encryption used on the payload, also may contain routing information for that message [6]. Figure 1 depicts the SOAP message.

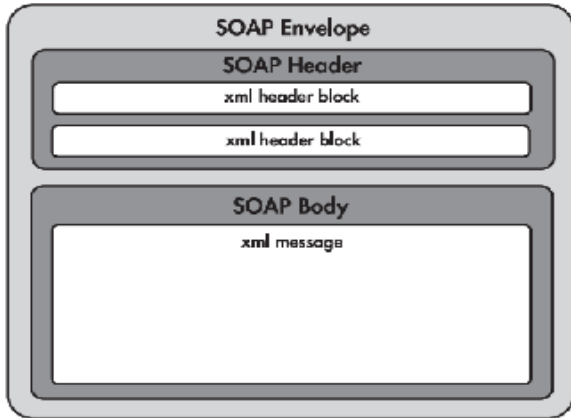


Figure 1: Diagram of a SOAP Message

1.2 REST

REST the abbreviation of “Representational State Transfer” designates an architecture style used to create networked applications. REST was introduced by Roy Fielding [7] in 2000 in the doctoral dissertation, one of the principal authors of the Hypertext Transfer Protocol (HTTP) [8] specification. REST uses a stateless, client-server, cacheable communications protocol which is almost always the HTTP protocol to create, retrieve, update, and delete resources. RESTful web services typically map the four main HTTP methods to the so-called CRUD actions: create, retrieve, update, and delete. Its architecture is based on client server model see Figure 2 in which client sends request to server then server process the request and return responses. The Table 1 shows REST HTTP methods and how they are typically used to implement a RESTful API.

Table 1. HTTP Methods and CRUP Actions

HTTP Method	CRUD Action
GET	Retrieve a resource
POST	Create a resource
PUT	Update a resource
DELETE	Delete a resource

While considering Android devices environment in particular, an execution of SOAP-based Web services may insert a noteworthy and an inappropriate performance overhead on the devices due to the increasingly thick SOAP messages and its expensive parsing requirements. Conversely, REST architecture style uses Uniform Resource Identifier (URI) and can greatly enhance application performance degradation

factors such as SOAP message length and parsing. Loosely coupled and flexibility of REST is also beneficial to devices

2. PURPOSED WORK

The work compared the performance analysis of Web services for Android devices using the two major Web service building technologies SOAP and REST. Both technologies work on HTTP as a layer of transport and SOAP is a protocol used for building Web services, whereas, REST is an approach for building Web services. XML is the medium of transporting messages from server to client for SOAP. The client-server model in Figure 2 shows the request response model. The client in this paper is Android devices and used server is Apache. The clients will increase from one to five and the payload will also vary to evaluate the comparison. The clients will also be tested on 2G, 3G and Wi-Fi networks. On server side, develop a Web service client for each class of Web services i.e. SOAP and REST. The calls from the client to server will be made to analyze the performance.

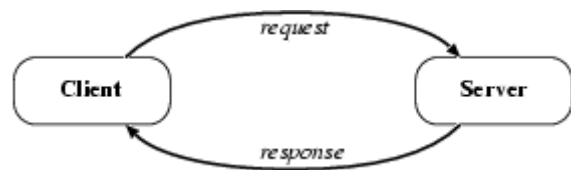


Figure 2: Client-Server model

3. METHODOLOGY

3.1 Service Implementation

Implement RESTful and SOAP web service and host them on the Apache server.[9] The Apache HTTP Server Project is a collaborative software development effort aimed at creating a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server. The server runs on, IBM compatible PC with 3.2 GHz processor and 2 GB RAM, runs on windows 8 platform. Client devices are Android Mobile emulators which are contained in the Android SDK. Emulator will develop and test Android applications without using a physical device. The Android emulator mimics all of the hardware and software features of a typical mobile device, except that it cannot place actual phone calls. To test application more easily, the emulator utilizes Android Virtual Device (AVD) configurations. Once your application is running on the emulator, it can use the services of the Android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes. [10]

3.2 Benchmark Configuration

To make the comparison, developed SOAP-based and REST-based Web services. In this paper, created an Employee service and created an ‘Employee’ table in MySQL database containing following attributes FirstName, LastName, Nationality, EmployeeId, Department, Email, Mobile, Salary, City and Country. The Employee service provides four methods: getEmployee which provides the particular employee record from the database, addEmployee which adds the particular details of an employee to the database, updateEmployee which updates the particular employee record using EmployeeId, deleteEmployee which deletes the record of employee on the basis of EmployeeId. On the basis of these four methods, calculate the response time which will be calculated as

$$Tr = Ct + Nt + St$$

Where:

Tr = Total Response Time

Ct = Client Time

Nt = Network Time

St = Server Time

Client applications were developed on Android devices (using Java) to invoke and interface with the appropriate service functionalities offered by Employee service. The interaction between the client application and Employee service was used as the basis for analyzing the performance of SOAP and REST.

4. EXPERIMENTAL RESULTS

Response time for all four methods of Employee service is evaluated for both REST based and SOAP based Web services.

4.1 Response time for getEmployee method

The getEmployee method provides a client (Android device) to request for information about an employee by using EmployeeId. The service gets the particular employee details from the database and sends the response to the client. Multithreading was used to depict multiple clients requesting the service at the same time. Figure 3 shows the response time for getEmployee method. It can be seen from the graph that for the getEmployee method, REST had better response time than SOAP as the number of simultaneous clients increased.

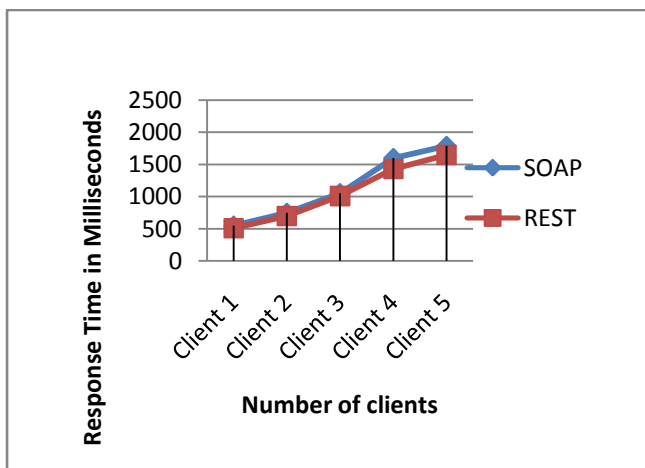


Figure 3: Response time for getEmployee method SOAP vs REST

4.2 Response time for addEmployee method

The addEmployee method provides a client to add new employee record to the database. The service adds the new employee details to the database and sends a response to the client to inform successful completion of the process. Figure 4 depicts SOAP vs. REST comparative chart for four clients. From the graphs, it can be observed that for the addEmployee method, REST had better response times than SOAP as the number of simultaneous clients increased. Similar to getEmployee method, REST had better response time.

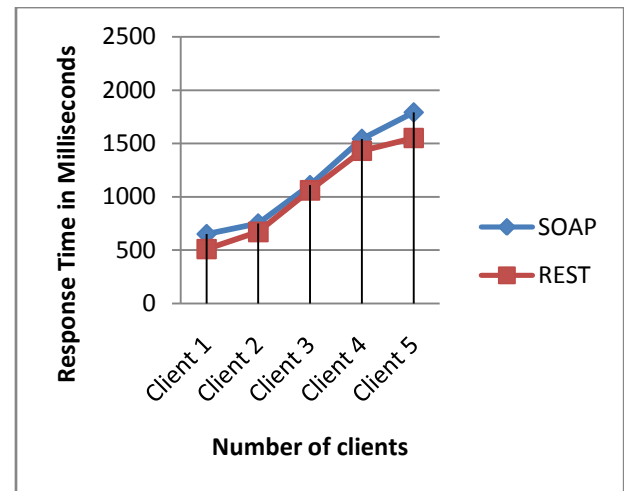


Figure 4: Response time for addEmployee method SOAP vs REST

4.3 Response time for all four methods

The all four methods involved invocation of addEmployee, getEmployee, updateEmployee, and deleteEmployee methods in a sequential order. Thus, the client makes service requests in a sequential order, and service fulfills each request and sends a response after completion of the request. Response time was calculated for completion of all four methods. Figure 5 show SOAP vs. REST comparative chart for five clients. From the graphs, it can be observed that for all the four methods, REST had better response times than SOAP as the number of simultaneous clients increased.



Figure 5: Response time for all four methods SOAP vs REST

5. CONCLUSION

The Table 2 shows the results of the experiments indicate that REST has better response time than SOAP. There can be various factors which affects the response time such as server processing capabilities, network bandwidth, payloads, distance between client and server and number of clients accessing the Web service. The methodology adopted for both REST and SOAP service is same and were tested using the same set of payload sizes, client-server machine and number of clients. As shown in experimental results while keep on increasing the number of clients, REST outperforms SOAP in all the requests. Therefore, Restful service provides high performance and perfectly good solution for the majority of

implementations on mobile devices. Thus as a part of future work, comparison can be made by increasing the number of clients and testing them using different payloads.

6. REFERENCES

- [1] Michael zur Muehlen, Jeffrey V. Nickerson, and Keith D. Swenson, 2005. Developing Web Services Choreography Standards - The Case of REST vs. SOAP, Decision Support Systems, 40(1):9-29.
- [2] Steve Vinoski, 2008. Serendipitous Reuse, IEEE Internet Computing, 12(1): 84-87.
- [3] Snehal Mumbaikar, Puja Padiya, 2013. Web Services Based On SOAP and REST Principles. In Proceedings of the International Journal of Scientific and Research Publications, 3(5):1-3.
- [4] Cesare Pautasso, Olaf Zimmermann, Frank Leymann, 2008. RESTful Web services vs. Big Web Services: Making the Right Architectural Decision, WWW, ACM April 21–25, 2008, Beijing, China.
- [5] Pavan Kumar Potti, 2011. On the Design of Web Services: SOAP vs. REST. Master Thesis, University of North Florida, pp. 20-42.
- [6] W3C. The SOAP Header Element. W3schools.com. Available: <http://www.w3schools.com/webservices/ws_soap_header.asp>. Accessed on: 2 March 2014.
- [7] Fielding, R. T., 2000. Architectural Styles and the Design of Network-based Software Architectures. Doctor of Philosophy, University of California, Irvine. Retrieved from: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
- [8] Group, Network Working. Hypertext Transfer Protocol - HTTP/1.1.061999. Available: <<http://www.w3.org/Protocols/rfc2616/rfc2616.html>>. Accessed on 10 March 2014.
- [9] Apache. About Apache. Available at: <http://httpd.apache.org/ABOUT_APACHE.html>. Accessed on 11 March 2014.
- [10] Android, Android Emulator. Android.com. Available at : <<http://developer.android.com/tools/help/emulator.html>>. Accessed on 11 March 2014.

Table 2. Service Response Time for REST and SOAP Methods

No.of Clients	Response Time (Milliseconds)					
	SOAP			REST		
	getEmployee	addEmployee	All four Methods	getEmployee	addEmployee	All four Methods
1	550	650	850	510	510	710
2	750	750	950	700	670	860
3	1050	1110	1260	1010	1060	1140
4	1600	1540	1490	1430	1430	1370
5	1790	1790	1810	1650	1550	1710