Next Generation Mobile Application in Cloud Computing using RESTful Web Services

Arvind D Meniya
Department of Information Technology,
Shantilal Shah Engineering College,
Bhavnagar, Gujarat, India.
arvind.meniya@gmail.com

Prof. H.B.Jethva
Department of Computer Engineering,
L D Engineering College,
Ahmedabad, Gujarat, India.
hbjethva@gmail.com

Abstract — Mobile devices, especially smart mobile devices, becoming very popular nowadays. Web services that are used to provide a service for an application running on smart mobile devices also becoming very popular. There are some limitations of the current smart mobile devices, such as limited storage capacity, low processor speed, limited memory, limited battery and slow wireless connection. The current mobile application running on a smart mobile device uses a services which traditionally built using SOAP based, which takes much size of the data to deliver as a service and time to deliver that service. This will lead to build a new type of web services over this traditional web services, and the solution is making a web services using REST way. This paper focus on the performance evaluation of the REST web services over the traditional web services by storing the RESTful web services in Cloud Computing.

Keywords-Web Service, SOAP,REST,WSDL

I. INTRODUCTION

Web Services [1,12] is a set of standards and a programming methods for sharing data between different software applications, moreover Web services is a standardized way to distribute services on the Internet. Web Services achieves its goal in a technology neutral manner; it provides well-defined interfaces for distributed functionalities, which are independent of the hardware platform, the operating system, and the programming language. So distributed functionalities, or services, which may be running on different hardware platforms, may be running in different operating systems, or may be written in different programming languages, can communicate through web Service interfaces.

Interoperability of Web Services mainly stems from its Extensible Markup Language (XML) based open standards. The Simple Object Access Protocol (SOAP) [2] is defined in XML. Since it is text-based and self describing SOAP messages can convey information between services in heterogeneous computing environments without worrying about conversion problems, there are many other Web Service specifications. Two of them, which are based on XML, are Web Service Description Language (WSDL) [3] and Universal Description, Discovery and Integration (UDDI) [8]. WSDL defines a standard method of describing a Web Service and its capability, and UDDI defines XML-based-rules for publishing Web Service information. Messages are exchanged through the SOAP protocol. SOAP works by exchanging information using GET/POST over HTTP. This allows the data to be exchanged regardless of where the client is in the network.

As Web Services technology has become an industry standard for connecting remote and heterogeneous resources, mobile devices have become a vital part of people’s everyday life. People use mobile devices anytime and anywhere, they may use their mobiles to check Email, access the Internet, or run other web applications. Web Services enable pervasive accessibility by allowing for user mobility as it overcomes the physical location constraints of
conventional computing. However, mobile computing also requires a technology that connects mobile systems to a conventional distributed computing environment.

The integration of mobile computing with Web Services technology will give many advantages to both sides. Mobile devices getting computationally capable, so mobile devices enabled with web services can be equal participant of web services architectures (can be web service client or web service provider).

Applying current Web Service communication models to mobile computing may result in unacceptable performance overheads. This potential problem comes from two factors. First, the encoding and decoding of verbose XML-based SOAP messages consumes resources. Therefore Web Service participants, particularly mobile clients, may suffer from poor performance. Second, the performance and quality gap between wireless and wired communication will not close quickly. It is caused by the mobile environment’s constraints like limited processor speed, limited battery lifetime, and slow unreliable and intermit connection.

The typical web application that requires the transmission of four to five times more bytes if implemented as a Web service compared to the same service implemented as a traditional dynamic program (e.g. Active Server Page application).

From this overall analysis of the web services, the performance of Representational state transfer (RESTful) web services [6] has not been evaluated on mobile devices. In this paper, we evaluate the performance of RESTful web services compared to the performance of conventional SOAP web services for mobile devices. Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. It is a style of web services use. It attempts to emulate HTTP and similar protocols by constraining the interface to a set of well-known, standard (generic) operations (e.g., GET, POST, PUT, DELETE). Here, the focus is on interacting with stateful resources, rather than messages or operations. RESTful offers a perfectly good solution for the majority of implementations, with greater flexibility and lower overhead.

The rest of the paper is organized as follow: Traditional Mobile Web service Development Problem, RESTful Web Services, Cloud Computing, Implementation Environment, and conclusion.

II. TRADITIONAL MOBILE WEB SERVICE DEVELOPMENT PROBLEM

Traditional environment of developing a web services for the mobile computing environment faces a problem, such as performance degradation. So, a primary research issue in the area of mobile Web Services is the attempt to provide an efficient message processing scheme while preserving XML’s interoperability.

Consuming WS from a mobile client is different compared to the standard WS scenarios, due to these factors. 1) Mobile devices have limited resources (e.g. CPU power, screen size). 2) The communication between client and service is established through wireless or cell network. 3) Existing WS in the Cloud do not support mobile clients.

There are several challenges in the process of consuming Web Services from mobile clients. 1) Loss of Connection 2) Bandwidth problem 3) Limited Resources. To overcome these challenges, a Mobile Cloud Computing (MCC) architecture is needed which connects mobile devices to the Cloud Computing. The MCC architecture includes a mobile client and a middleware design. An MCC is look like as given in fig.

The middleware [4] acts as a proxy that is hosted on the Cloud platforms which provide mobile clients access to Cloud services. The middleware improves interaction
between mobile clients and Cloud Services, for example, adaptation, optimization and caching.

III. RESTful WEB SERVICE

REST [5] is a software application architecture modeled after the way data is represented, accessed, and modified on the web. In the REST architecture, data and functionality are considered resources, and these resources are accessed using Uniform Resource Identifiers (URIs), typically links on the web. The resources are acted upon by using a set of simple, well-defined operations. The REST architecture is fundamentally client-server architecture, and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture, clients and servers exchange representations of resources using a standardized interface and protocol. These principles encourage REST applications to be simple, lightweight, and have high performance.

RESTful web services [5] are web applications built upon the REST architecture. They expose resources (data and functionality) through web URIs, and use the four main HTTP methods 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. Below table shows a mapping of HTTP methods to these CRUD actions.

<table>
<thead>
<tr>
<th>HTTP Methods</th>
<th>CRUD Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Retrieve a Resource</td>
</tr>
<tr>
<td>POST</td>
<td>Create a Resource</td>
</tr>
<tr>
<td>PUT</td>
<td>Update a Resource</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete a Resource</td>
</tr>
</tbody>
</table>

Table 1 : Http Methods and their Action

REST web services aim to be simple, and this is accomplished by limiting the types of operations one can perform on a resource. REST founders claimed that it [5]:

- 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.
- Provides equivalent functionality when compared to alternative approaches to communication.
- Does not require a separate resource discovery mechanism, due to the use of hyperlinks in content.
- 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).

IV. IMPLEMENTATION OF RESTFUL WEB SERVICE

To evaluate the performance of RESTful web services against conventional SOAP web services, we implement a RESTful web service and a SOAP web service and develop a web service client on a mobile device as well as on desktop client using console based application for each class of web services. Next, we shall illustrate the service implementation, the client implementation and emulator configuration, and the benchmarking environment.

A. Development of SOAP and RESTful Web Services

We implement RESTful and conventional SOAP web service and host them on the Glassfish application server as well as IIS 7. Glassfish [6] is a web service framework developed at Sun Microsystems[10,11]. Internet Information Services (IIS) [7] – formerly called Internet Information Server – is a web server application and set of feature extension modules created by Microsoft for use with Microsoft Windows.

B. Client Implementation

Mobile applications (service client) implemented using Eclipse indigo with Android SDK runs on Android Mobile device Emulator and desktop application (service Client) implemented using Microsoft Visual Studio 2008.

C. Benchmark Configuration

We implement three benchmarks using three different data types as parameters to the web service: float data type, string data type and integer data type. We measure a total session time and a message size of service call. Benchmarking web services are listed below:

1) String Representation
This web service will show the text that is passed by the user as a parameter and display the resultant string concatenation with default string parameter.

2) Integer Addition
This web service will show the addition of three integer values passed by user as a parameter and display the sum of these values.

3) Floating Addition
This web service will show the addition of three floating values passed by user as a parameter and display the sum of these values.
V. PERFORMANCE EVOLUTION OF WEB SERVICES

Table 3 shows the benchmarking results of the string concatenation, Integer Addition and float numbers addition web services. Table also shows the message size in bytes for our web services. The message size of RESTful web service is smaller than messages of Conventional SOAP web service. The response time of RESTful web service is smaller than messages of Conventional SOAP web service. From the result, it shows the higher advantage of using RESTful web service. The gap is very large between the response time of RESTful and the conventional SOAP web service.

<table>
<thead>
<tr>
<th>NAME OF SERVICE</th>
<th>MESSAGE SIZE (BYTES)</th>
<th>TIME (MILLI-SECONDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SOAP</td>
<td>REST</td>
</tr>
<tr>
<td>String Concatenation</td>
<td>148</td>
<td>19</td>
</tr>
<tr>
<td>Integer Addition</td>
<td>268</td>
<td>27</td>
</tr>
<tr>
<td>Floating Addition</td>
<td>363</td>
<td>43</td>
</tr>
</tbody>
</table>

VI. HOSTING OF WEB SERVICES IN MOBILE CLOUD COMPUTING

Cloud computing is web based processing, whereby shared resources, software, and information are provided to the users on demand over the internet. Now, Cloud Computing will have an impact on mobile industry also. The increased network connectivity and popularity of the smart phones and tablet devices like iPad have considerably increased the usage of applications on mobile devices. In Mobile Cloud Computing the data storage and data processing occurs outside the mobile device and results are displayed through screen or speakers. One of its major advantages is that users are no longer tethered to a desktop PC in their home and office, as they can access all the functionality of a cloud-based application from any location, as long as they have a Internet connection.

Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple servers. Cloud Platforms usually refer to application hosts that offer computational power, storage and Web access.

I have used the free cloud hosting environment that is provided by Ubuntu one [9]. The availability of the RESTful Web services becomes very high after storing it in the cloud environment.

VI. CONCLUSION

We have evaluated a RESTful web service for mobile devices, where we developed RESTful and conventional SOAP benchmarking web service. Benchmarking includes string concatenation, integer addition and float number addition web services. The performance evaluation results show the advantages of using RESTful web services over conventional web services for mobile devices. Advantages include less message sizes and response time. Results of performance comparison between conventional SOAP and RESTful show the obvious high performance RESTful over SOAP. The hosting of RESTful web services in the cloud environment will also improve the performance. Therefore, RESTful offers a perfectly good solution for the majority of implementations, with higher flexibility and lower overhead with mobile cloud computing.

VII. ACKNOWLEDGMENT

We would like to thanks Dr. Ravindra Patel for his comment. We would also like to thank Mr. Ravindra Makwana, Lecturer in IT Department , BPTI, Bhavnagar for his supportive help regarding this paper.

VIII. REFERENCES

[7] Internet Information Services /Server: http://www.iis.net/