

Performance of Web Services on Smart Phone Platforms

Rajitha B

Assistant Professor, Computer Science and Engineering Department, MNNIT Allahabad, Utter Pradesh, India

Abstract: This paper is proposed to investigate Web Services (WSs) performance on nomadic device like the Apple iPod touch. Because of the reduced capabilities of the mobile devices, clients residing in these terminals should be as simple as possible. So, we propose to move part of the business logic to an intermediate service provider. A WS can share a conceptual functionality between various WS, although they could be implemented in different locations and with different characteristics. A WS provider makes the mobile client independent of the implementation details as well as of their availability. A WS basically developed using two methods SOAP and REST. In this paper we measured the performance of these two Web Service's on the mobile device and analyzed which is efficient.

Keywords: *Web Services, Service Oriented Architectures, RESTful Services, Simple Object Access Protocol.*

I. INTRODUCTION

Rendering services over web through wireless technology has grown rapidly from the last decade, which lead to develop number of specific and personalized services for clients residing at various locations (such as at different networks). These services can be designed either by statically or dynamically. The static services are useful to resolve the issues of clients accessing the same patterns of service requests. But, it fails to resolve the problems of clients residing at different networks who try to consume the services depending on their respective preferences and context. This leads to think about deciding the context for requesting a service dynamically. So, in order to achieve this goal the internet specifies a new technology called Web Services (WS).

Web Services (WSs) [1] represent a black-box functionality which contributes the best aspects of component-based development and the Internet. Web Services can vary from simple service like "Hello World" to complicated business logic functionalities. One of the main advantages of using WS is reusability which leads not to worry about how the service is implemented. Web Services are accessed via Web protocols (ex: HTTP) and data formats (ex: XML). WS provides several standards to support both Business to Customer (B2C) and Business to Business (B2B) infrastructure, which lead to be more popular. WSDL, UDDI and SOAP are considered the core of this Web Service standards set. WSDL (Web Services Description Language [2]) defines a XML format for describing network services. These services are published and registered in an UDDI (Universal Description, Discovery and Integration [3]) registry; a platform-independent, XML-based registry for Web Services descriptions. Finally, the exchange of information between clients and WSs is supported by SOAP (Simple Object Access Protocol [4]), a XML-based lightweight protocol suitable for communicating in a decentralized, distributed environment.

II. PROBLEM STATEMENT

WS would be more appropriate for mobile devices. Since WS hides the internal details of the service, nomadic devices can be use this technology especially when they are non static. There are tremendous approaches described already for WS on nomadic devices based on SOAP protocol. A Few problems make it difficult to achieve this desirable open framework where flexibility is the key factor to communicate between the WSs and mobile clients.

- The SOAP specification doesn't mention security.
- XML needs to be processed (parse) for response. This is a limitation for mobile devices since they have limited processing power.
- No Interface flexibility.
- SOAP envelope may be heavy on Bandwidth.

To overcome such problems, another web architectural style has been proposed by Thomas Roy Fielding in his Ph.D. dissertation [6] called RESTful. REST focuses on component roles and constraints on interaction of components and ignores the protocol syntax. It mayn't even depend on XML document; it is based on HTTP method. It avoids the XML document parsing and it can be very easily developed by any person who has experience with network protocol HTTP. Light client solves the complexity problems of SOAP linked

to the capabilities of the mobile devices. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III. RELATED WORK

WS technology is changing the World Wide Web platform from the publication mechanism to software platform. This software platform are described, advertised, discovered, and invoked from the designed Application Programming Interface (API) as shown in Fig.2.1. The service requester requests the broker for the services to be invoked. This broker maintains a list of published and unpublished services (WS Providers) by verifying from the third party authorities. The service requester discovers the service provider via Web Service broker and service is invoked to the requester when the provider is an published service over wireless network.

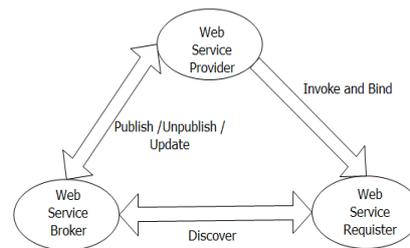


Fig.2.1 The web model

“The successful integration of the Wireless Communications and Web Service technology realms can lead to the creation of new and compelling business scenarios that expand on a broad array of endpoints (wireless devices, servers and PCs operating over the Internet, home devices)” as specified in [7]. Thus leads to “computing at the edge”.

Web Services are defined in a machine independent format such as XML [8] protocol. An advanced protocol designed for accessing Web services is SOAP (Simple Object Access Protocol) [4] based on XML. SOAP is defined on XML schema and their data-types. It contains around 40 data-types derived from XML and it also allows custom defined complex data-types. Due to which SOAP becomes powerful and rich language for exchanging information among object-oriented systems deployed at different locations. In addition to strong data-type support, SOAP also supports various messaging schemes.

A. Mobile Devices Area

Though there are wide collections of mobile devices like Nokia, RIM, Apple, Blackberry, Android, Palm Pre, Google Phone etc the general purpose of using a mobile devices is the same. The general goals to be achieved by mobile devices are as follows

- **Telecommunication Functions:** It mainly includes the voice services and then the text oriented messages such as SMS (short message services), Multimedia Services (MMS) for communication.
- **Email Client:** Enables to access the wireless communication protocols via a secured network. To provide such services a data service providers are needed which should be compatible for the mobile devices screen resolution. This plays a prominent role in classification of devices in terms of performance even when working as offline. Which some devices provide and some are not.
- **Organizing tools:** This includes the contact, notes, calendars, reminders etc.
- **Elaborating Documents:** Most mobile devices are providing this feature as advancement. This may be restricted based on the operating system type provided in the mobile devices.
- **Web Client:** This allows the users to check their business portals, news portals etc.
- **Entertainment:** This includes playing games, accessing audio and video files on the mobile devices, recording videos, taking pictures etc.
- **Chatting:** Chatting is one of the main uses of mobile devices in current generation.

B. Developing Platforms for Mobile Devices

Based on the above customers needs, mobile device applications are developed on different platforms respective to their company development tools. The most popular editions are Java Micro Edition, Windows Mobile, RIM Black Berry, and Apple Mac iPhone. The Java Micro Edition (JME) is the platform for developing applications for mobile devices. It provides two configurations; Connected Device Configuration (CDC) and Connection Limited Device Configuration (CLDC). CDC is for larger devices, typically for 2MB of memory size. CLDC is for smaller devices i.e. less than 512KB of memory. Sun Java Wireless Toolkit is set of tools; it provides an environment for creating java applications that can be deployed into mobile devices. This toolkit is

specific to CLDC. So, for the devices of larger memory the Sun provides Netbeans IDE. Sun Netbeans IDE with module plug-in seems to be the most popular tools for java platform enabled devices. There is another platform called Eclipse ME plug-in which is also the best for JME it is but less comfortable when compared to Netbeans. The current version of Netbeans is 6.5.1.

The tool for Windows Mobile is the eMbedded Microsoft Visual C++ and eMbedded Microsoft Visual Basic. These environments are later redesigned and developed as new integrated environments such as MS Visual Studio 2003, 2005 and at current as 2008 with .NET. The RIM BlackBerry uses its own development tool BlackBerry Java Development Environment based on Java ME with enriched Enterprise Server, Mobile Data Services and also it has the virtual terminal in the internal firm network with a secure channel. So, the applications developed in JME can also run in BlackBerry JDE with a secure data transfer. It uses the Rapid Application Development (RAD) from IBM to provide the web services client.

Apple uses the tool Xcode for developing the applications for iPhone or iPod touch using Apple's native language objective-c. The Present version of Xcode is 3.1.2; this comes with iPhone SDK (Software Development Kit) documentation for programming, current version is 2.2.1. This Xcode interacts with the tool Interface Builder (IB) provided by Apple to develop the GUI design. Basically Xcode is for coding and IB is for display view controllers to the user through which he interacts with the code.

C. Web Services for mobile Devices

Mobile devices are the most powerful and common used devices for communication in day to day life of a common human interaction. This lead to a challenging application integration and introduced new technologies for application interaction. The most common challenges are device size, bandwidth limit and processing power. Even the complex machine to machine interactions have been involved in mobile application integration.

Ad-hoc networking involves the creation of network temporarily for the wireless devices which can communicate with each other. This dynamic creation of network involves a large number of platform combinations such as laptops, mobile devices and processors which should not have a semantic mismatch in communicating software. XML (eXtensible Markup Language) has many features that can well-fits in this scenario. This provides a platform independent format for programming interface that can communicate between different mobile devices. Web Services are services provided over web for providing various functionalities for web clients. These services are provided via remote procedure calls over HTTP. Particularly Web Services uses Simple Object Access protocol (SOAP) [4].

Due to the nature of heterogeneity of mobile devices Web Services will well suites the mobile integration for multiplatform combinations by providing the platform independent nature via XML and WSDL (Web Service Description Language) [2]. The Services provide UDDI [3] for discovering and deploying any service uniquely for a mobile integration. Due to reduced processing capacities of mobile devices they cannot handle much server applications, so a better way is to provide Web Services as the Services for doing some business logic for the request given by the mobile client and return the results instead of doing processing at client side. In this way the overhead on devices can be moved to Web Services. Since mobile devices have very less memory so, they may not handle huge programs. In this scenario also Web Services plays an important role by residing at an end point as remote server for all the services.

A typical system with Web Services for mobile devices will look like the Fig.2.3. This shows that every service should be registered first using UDDI. The mobile client in the network first sends a request for the service discovery to the UDDI registry. This UDDI then checks for the service description (WSDL) in the repository and if found it responds with the respective services invoking address. Then Mobile Client sends the request to particular Web Service as per the WSDL.

D. Traditional changes in designing Web Services

The Web was particularly successful due to introduction of XML in 1998. This lead to the successful integration of SOAP based on Service Oriented Architectures (SOA). Technically, SOA is defined as identifying and designing the service providers and build an information system for consuming the service. SOA guides in designing information system architecture. There are two architectural styles for building a service-oriented system they are Representational State Transfer (REST) [6], and Message-Oriented Middleware (MOM) [4].

They are classified as loosely coupled and tightly coupled respectively. MOM i.e. SOAP is considered to be tightly coupled since it is based on well-defined relationships, where all peers are supposed to bind to a single network. The differences between these two techniques are underlined based on the issues like service discovery, identification of service, binding a service, shared data models, interface granularity, state management and independent development of service providers and consumers [12].

Traditional web application development depends on tightly coupled elements, often running in the same process. Its underlying architectural style is achieved by integration (making all services as group). Therefore the key design pattern in these systems is synchronous interactions. Tightly coupled systems require the interfaces between different components to be tightly interrelated in function and form, thus increasing the difficulty to make changes to part or whole application. Also it is difficult



Fig 2.3 Mobile Devices in ad-hoc network

to build tightly coupled applications, as the number of applications and services provided increases thereby requiring increase in number of interfaces to be created and maintained quickly. Furthermore, it requires lot of time in defining the connections and relationships between two cooperating applications.

A loosely coupled environment can be provided as an alternative to tightly coupled elements, where in the communication is achieved by cooperation [12]. These applications do not require knowing how the cooperating applications behave or implemented. Benefits of loosely coupled systems are in its agility and ability to survive changes in structure and implementation of the service. The main goal of service oriented computing is to create cooperating services in which application components are created with little efforts into a network of services that are loosely coupled to create flexible dynamic business processes and agile applications that span organizations and computing platforms [13]. The support for lightweight programming and loosely coupled systems is necessary in many contexts so that most of the applications can remain loosely coupled and fragile. However, these concepts are enclosed in the Web2.0. Web2.0 applications are dynamic; they invite user participation and are responsive to user requests as desktop applications. The standard Model View Controller (MVC) pattern seen in software engineering applies to the web application development. It helps in separating the business logic, data structures and physical interface for maintainability.

III. PROPOSED WORK

"Big Web Services" are the Simple Object Access Protocol (SOAP) standard derived from Extensible Markup Language (XML) messages, where messages are sent through SOAP envelope over network. But RESTful Web Services provide even better integration with HTTP. They are designed to work even without XML and WSDL Service API specification. RESTful WS meet the W3C definition; they gained much popularity than the SOAP WS.

A. SOAP

SOAP is a way to develop a web application under one kind of operating system and communicate it through same or some other kind of operating system by using World Wide Web's Hypertext Transfer Protocol (HTTP) and XML mechanisms for exchange of information. SOAP is message-level protocol where as HTTP is a transport-level communication protocols. All major operating systems comes with installed Web protocols, so programming running on such systems can easily communicate with each other via HTTP and XML. SOAP specifies exactly how to encode request and response messages using the SOAP envelope with clear headers, so that it can be called by heterogeneous applications. It can also be used in combination with SMTP, JMS and etc though the HTTP is widely used. SOAP envelope travels via different transport- layer protocols before reaching the actual destination.

Disadvantages

- The SOAP specification doesn't mention security.
- XML needs to be processed (parse) for response. This is a limitation for mobile devices since they have limited processing power.
- No Interface flexibility.

- Heavier on Bandwidth.
 - Because SOAP deals with objects serialized to plain text and not with string filed remote object references (interoperable object references, IORs, as defined in CORBA), distributed garbage collection has no meaning.
 - Client thread is blocked until the response is received.
 - SOAP clients do not hold any stateful references to remote objects.

B. RESTful

The Resource Oriented Architecture (ROA) is the name given to REST. Roy Fielding came up with a new name REST since it's an architectural style, not architecture. It's not even HTTP bounded. This ROA bridges undefined path between the RESTful principles and the HTTP based applications. The requirements are turned into resources in this ROA for designing web services. There are six data elements for REST they are resource, resource identifier, representation, representation metadata, resource metadata and control data. The first and key abstraction in REST is a resource; any kind of information can be treated as resource such as document, image, temporal service, non-virtual object, collection of other resources and etc. The second is resource identifier (URL/URN), identifies resource involved in an interaction between components. Third is representation consists sequence of bytes and metadata to describe these bytes it may includes document, file, HTTP message entity, instance, or variant. Next representation metadata and resource metadata is the information specific to the supplied representation (either requesting or response messages). Control data defines message purpose between components, such as the action/meaning of the resource is requested. It can parameterize requests and can override default behavior of some connecting elements such as cache behaviour can be modified.

The Connectors of the REST architecture are client, server, cache, resolver and tunnel. The primary ones are client and server, client initiates the communication by making a request and server is invoked based on this request and responds. Cache connector may be located either at server/client to provide cacheable data for later use. Mostly it resides on client to avoid repetitions of retrieving data from server to reduce latency. By default, the response to a retrieval request is cacheable and the responses to other requests are non-cacheable. These can be overridden by control data. Resolver translates resource identifier into network address readable format, needed to connect inter-component. Finally tunnel relays firewall or gateway at connection boundary between communications networks. It starts when connection is established and stops when connection is terminated.

The REST architecture has four components origin, gateway, proxy and user agent. A user agent initiates the request using client connector. An origin server uses a server connector to govern the namespace for a requested resource. Intermediary components act as both a client and a server in order to forward requests and responses. A proxy component is an intermediary selected by a client to provide interface encapsulation of other services, data translation, performance enhancement, or security protection. A gateway component is an intermediary imposed by the network or origin server to provide an interface encapsulation of other services, for data translation, performance enhancement, or security enforcement. RESTful web services are having the following four methods for communication unlike SOAP.

- GET performs execution of query on the server side requested by client and returns the data as per client representation.
- POST creates a new resource on the server. The web service response may be with data or the status indicating success or failure.
- PUT is used to update existing resources or data.
- DELETE is used to remove a resource or data.

In some cases, the update and delete actions may be performed with POST operations as well, for example, when the services are consumed by browsers that do not support PUT or DELETE.

Advantages

- The web services are completely stateless.
- Light Weight, no need for extra XML parsing.
- A caching infrastructure can be leveraged for performance and user can limit cache properties in GET method.
- Unlike SOAP, a RESTful web service doesn't need any particular tool for designing.
- 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.

- Mobile devices need minimum bandwidth so, REST is particularly useful for such limited-profile devices (PDAs), for which the overhead of headers and additional layers of SOAP elements on the XML restricts the load.
- RESTful Web service delivers response in the understandable format of user (not like XML response in SOAP).
- Scales well to large numbers of clients.
- Resources are retrieved not as character strings or BLOBs but as complete representations.
- Enables transfer of data in streams of unlimited size and type.

C. New Mobile World - iPhone

The iPhone is mainly designed to work with fingers. In SAN FRANCISCO, Jan. 10, 2007 Apple CEO Steve Jobs introduced this device as stating “First Macintosh, Then the iPod and now, he was lucky to introduce not one more, but three new revolutionary products -- a new iPod, a new phone and an Internet communicator”. He also said that the iPhone uses software that was at least five years ahead of any phone. This included so many features which were never seen in any single digital mobile device. iPhone is more like a mini-computer that can fit in our pocket and in hand like any other cell phone with wide features such as web browser, gaming kit, phone and music player. The Apple iPhone takes the third most leading place with sales of 3,938.8 million devices and occupies 10.8 per cent of market share in first quarter of 2009 where in year 2008 it had only 5.3 percent of market share, now it’s almost double in count [9].

The iPhone and iPhone3G are offered in 3.5-inch touch-screen that employs multi-touch technology. User can control items on this device with multiple fingers simultaneously thus named as multi-touch device. It’s most famous features, such as tapping the screen twice to zoom in or “pinching” and dragging your fingers to zoom out. The iPhone 3G offers a faster network connection, slightly rounder shape, a slightly heavier weight, improved battery life and two color choices. These phones use a trio of sensors to produce some of their best usability features, though neither model offers expandable or upgradeable memory.

The iPhone is similar to mini-computer, it supports wide set of features that a normal computer does. Some of the features are phone, Web browsing, E-Mail, calendar/PDA’s, iPod, 3-megapixel camera, video playback and App store. The iPhone has the feature of a PDA’s like calendar, address book, weather updates, business market share tracking and more. The iPhone uses built-in YouTube application for mobile video playback, due to its big and beautiful 3.5-inch screen it catches the user eye greatly. It allows shooting video, editing it, and sharing it with high-quality VGA video in portrait or landscape mode. App Store facilitates iPhones to run all kinds of third-party programs, including games to Facebook and Twitter to restaurant finders and productivity apps. These third-party programs may be paid or free.

The App Store makes the iPhone the most useful smartphone around. The new 3-megapixel camera has built-in auto-focus, helps to focus on anything and takes great photos. The keyboard of iPhone displays application-specific shortcuts. For example: for safari the keyboard appears with .com key, and @ key. This keyboard is adaptable, allows switching between more than 40 international layouts. iPhone also features sophisticated character recognition software that lets you draw a Chinese character with your finger and tap to choose the matching result.

D. iPhone SDK

The iPhone Software Development Kit (SDK) [16] provides the tools and resources required to develop the native iPhone applications. Objective-C is the base language for developing iPhone applications. Native applications run as standalone application on iPhone OS. Every iPhone application is built using the UIKit framework and therefore has essentially the same core architecture. The iPhone SDK includes access to documentation, resources and sample code in order to develop applications.

- *Xcode*: The iPhone provides Xcode as the power source editor and Integrated Development Environment (IDE) tool for project compiling, managing, running, and a graphically debugging. Xcode is integrated with iPhone OS. The advanced text/source editor, supports features such as code completion, syntax coloring, code folding and inline annotations for errors, warnings, and notes [15]. Xcode, have a choice of building application either on iPhone simulator or on real device.
- *iPhone simulator*: iPhone provides a great tool for locally running, testing and debugging an application on the Mac using this simulator.
- *Instruments*: It allows to collect, display and compare performance data graphically in real-time to optimize the application. Gathered data (about memory usage, disk activity, network activity and performance) is presented in a graphical display called timeline. It displays different types of information side by side, letting the user to correlate the overall behaviour for an application, not just in one specific area.

- *Interface Builder*: Interface Builder is the tool for assembling the application’s user interface visually eliminates the custom code needed to create, configures, and positions the objects. Application’s window is assembled by dragging and dropping preconfigured components.

IV. Implementation

The research deals with accessing Web Services on the mobile devices like Apple iPod touch. Three experiments are proposed to analyse Web Services performance on iPod touch. This project has been implemented in Objective-C for accessing service at client side and Netbeans IDE for designing service using REST architecture. Visual Studio .net tool for designing SOAP based service. Each service is located in different machine and client program is running on device (iPod touch). The following are the three experiments.

- Chatting application using RESTful web services.
- Experimental setup for comparing SOAP and REST style architectures.
- Performance Test for cache based systems.

A. Chatting application using RESTful web services.

The traditional chatting application uses the concept socket programming for message transfer over network. A socket contains the IP address and a port number. Socket programming is a networking program which is intended to design and run according to a specific protocol. Network communication protocol called internet protocol TCP/IP is the most widely used. Almost all the other protocols like HTTP, FTP, SMTP etc, are all based on TCP/IP. Sockets causes low network traffic and are very flexible to use. Socket based transfer allows to send the data as a raw information; it’s up to the server and client to make it according to the specified format to display. Security becomes overbearing because it just establishes the communication connection between two systems. And more importantly the data formats and protocols are system and application specific, so re-use of socket implementation is restricted and limited.

Due to limitations of socket programming this first project refers to use Web Services as the intermediaries for message transfers over network. Since the Web Services are independent in nature (not restricted to a specific protocol as of socket programming) they best suite this scenario. The client and server just need to know the URL of the web services and the set of parameters to be passed.

The clients are the Mobile devices and the service RESTful web service is designed in Netbeans 6.5.1 IDE. Web Services are designed in java, and there are different methods to process each request based on the client need. For authentication of user/client there is userAuthenticate service method to check whether user exists or not in the DB and respond to the client stating the status. If client gets successful in logging in he/she gets the list of online clients then client A can choose accordingly to whom he wants to chat and can send a request to other person. This message is sent with three parameters message, senderID and receiversID. This chatting is designed for university students and professors so; there exists a unique ID for each person (avoiding ambiguity). The RESTful service saves the messages in the DB including a flag stating that the client B has not read. The Client B checks pending message requests periodically with a timer setup, when finds any message that is retrieved and send to client B by modifying the flag information to read. RESTful web services well suites this scenario (avoids the XML parsing) unlike SOAP (overhead of parsing).

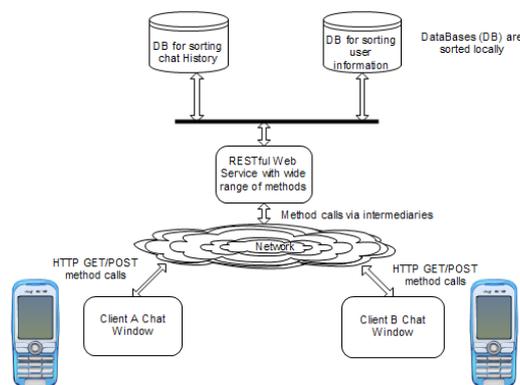


Fig.4.1 Architectural Design for Chatting application through Web Services

B. Experimental setup for comparing SOAP and REST architectures.

This experiment analyses the performance of SOAP and REST architectures on the iPod touch device. A simple service is designed to send the data as per the required bytes. If a user requests 4KB of data the service returns 4KB of information as a string. This experiment mainly focuses on how much time does mobile device take to download and parse the information. And analyze the performance of two architectural styles.

Mobile application contains a text filed to provide input for web service (the amount of Kilo Bytes of data to be retrieved). It contains to button actions one for SOAP call and other for RESTful call. And based on the action performed the total time taken for parsing and downloading, gets displayed on the device. The screen shots are shown in Appendix C.

The SOAP based web service is designed in .net platform and runs on IIS. The RESTful web service is designed using Netbeans 6.5.1 IDE tool. Both have same functionality i.e. to send requested kilo bytes of data to the service requester.

There is no functionally difference in service side. But on the client side the request is sent as the SOAP envelope for SOAP request and as a just QueryParam for GET method in RESTful service. For SOAP based service once the connection is established, first response is downloaded completely. And then SOAP result which is XML document is parsed and the exact information is retrieved. But for REST the data is retrieved directly and no need for parsing it, which reduces total retrieval time. Experimental results are shown in Table.2.1, various requests are been sent to the server ranging from 100KB to 10000KB. The response time is measured in milliseconds.

The Graph in Fig.4.3 shows the response time taken for both SOAP and REST architectural styles. This states that the retrieval time for REST is minimum compared to the SOAP service. The time taken for REST is almost same for all the requests, but for the SOAP it is getting increased for each request. At certain point like 10000KB of request the SOAP request gets crashed due to overload. But for REST it works fine. Hence the Experiment shows that RESTful web services best suites the Mobile devices rather than a soap.

Table.4.1 Experimental Results for response time

Data requested in Kilo Bytes	Parsing Time (ms) for SOAP service	Total response time (ms) for SOAP service	Total response time (ms) for RESTful service
100	80.504	3355.647	135.743
200	151.426	1389.58	172.731
300	197.729	4002.728	134.394
400	240.471	4784.679	1161.228
500	314.089	8473.51	123.677
600	359.526	7261.496	1778.143
700	412.052	8999.511	1155.853
800	471.03	10785.8	103.476
900	456.2119	12004.1	114.78
1000	577.418	12853.47	108.056
1100	635.507	15109.68	103.756
1200	686.658	16154.78	117.797
1300	737.038	17951.6	109.275
1400	813.254	18606.69	108.813
1500	848.96	13871.31	114.319
1600	1056.571	20652.98	1140.812
1700	951.901	27620.93	1206.617
1800	1022.469	49466.92	286.05
1900	1073.727	33499.68	1195.573
10000	6402.493	139705.9	130.349

C. Performance Test for RESTful service via Proxy.

There is a proposed system using RESTful web services developed in Netbeans. This system is proposed by Tracy, who was the master’s student in University of Saskatchewan. The system has a RESTful service which does the storing and retrieval of data from customer database, based on the client requests. This application is designed for browsing, shopping and ordering the products online. The client can be anyone like desktop, laptop and Mobile device. The proposed system was designed for desktop and any wireless clients. Now we have designed a Mobile device client on iPod touch.

The Request and Response is in XML format. The Client request is converted into XML and passed over network, similarly for Response the XML is parsed completely for actual result. The Client can request for as many records as needed. Since the data is transmitted over wired/wireless networks the response mayn't be guaranteed to be received at client and even the most of the client requests may be repeated. So, getting the same sorts of data from the service repeatedly is time consuming process. So the system architecture is changed as per Fig.4.2.

This architecture is designed with client proxy and server proxy. Client proxy is meant for caching the data and also to predict the next request. Server proxy is meant for sending the data to and flow from client proxy to main service and also to provide the locking for client writes to the database.

The iPod touch device client application is designed to perform all such operations. All the applications client proxy, server proxy, main service and client mobile application are distributed in different locations. Browsing, shopping and ordering the products online are performed via this iPod device from remote location.

The Experiment is conducted for measuring the performance of reads/writes from a client application. For these experiments the cache hit ratio is assumed to be 0, 0.2, 0.5, 0.8, and 1.0. So based on these cache hit ratio the data is cached into the client proxy. And the prediction is also assumed to be 0, 0.2, 0.5, 0.8 and 1.0 per cent. Where prediction 0 means the prediction accuracy is zero i.e. all the predicted results are incorrect, prediction 0.2 means that the prediction accuracy is 20 percent correct and the rest 80 percent is incorrect similarly for prediction accuracy 0.5, 0.8 and 1.0 percent. Based on this experimental setup there were different experiments conducted on iPod touch.

The Fig.4.4, Fig.4.5 and Fig.4.6 shows the experimental results observed for reads and writes for browsing, shopping and ordering the data online. It shows the results for all predictions and cache hit ratios combinations. The result shows that when the prediction is 100 and cache is 100 the reads takes very less time even than the conventional approach. Even writes also takes less time than the other predictions and cache hit ratios.

The Figs.4.7 and Fig.4.8 shows performance of cache for reads and writes in each category (i.e., shopping, browsing, and ordering). In the below shown figure, as the cache hit ratio increases the time taken for reading the information from shopping, browsing and ordering decreases. As seen in the graph, when cache hit ratio reaches 1.0, minimum time for processing the reads and writes also reduces. Similarly The Fig.4.9 and Fig.4.10 shows predictions rates of reads and writes for above mentioned categories. At 0 percent prediction the probability of reads and writes is more and it decreases as the percentage of prediction increases for each of them in the above mentioned categories.

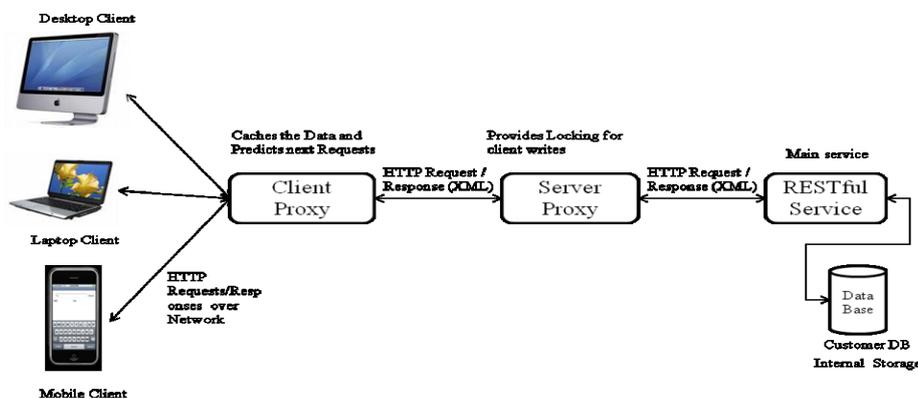


Fig.4.2 Architectural Design for Online shopping

V. CONCLUSION AND FUTURE SCOPE

Web Services is especially suitable for mobile environments, where devices are characterized by their low capabilities, nomadic use and need of dynamic selection of services. Therefore, we applied Web Services (used to hide complexity, availability and location problems to mobile clients) concept for mobile scenario which could be accessed over any communication infrastructure. To sum up, this paper proposes web services designed in RESTful style will best suite Mobile devices environment rather than SOAP web services. Unlike SOAP, RESTful Services are light clients, more adequate for mobile devices.

As far as the implementation is concerned we have developed basic experimental setups for comparing the SOAP and REST architectures and shown that REST is the best option than SOAP. But, the REST style can also be implemented with cache strategy on client side to save the data temporarily, which makes this system even better. The cache can either be saved on mobile devices or at some intermediately level, intermediately

level cache is already done in experimental setup three so next future scope can be concentrated on providing some cache on real handheld devices.

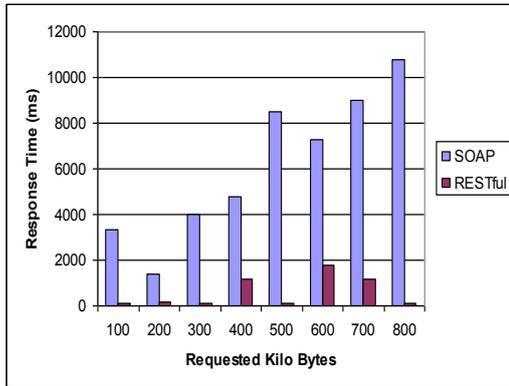


Fig.4.3 Comparison Graph for SOAP and REST

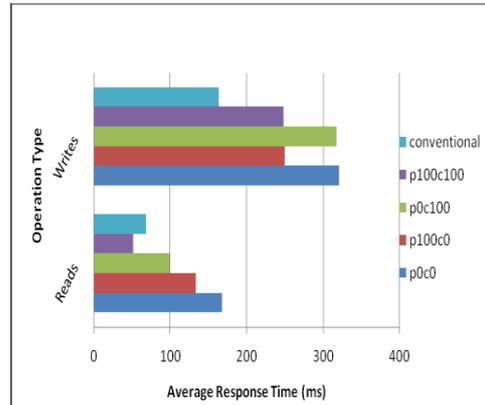


Fig.4.4. Overheads and Gains in Browsing

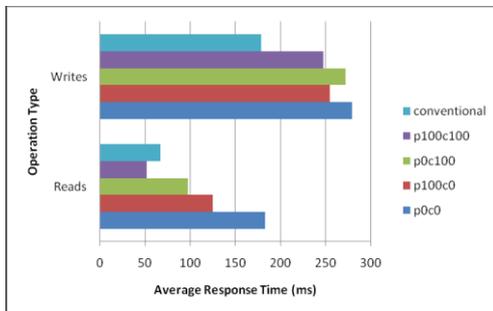


Fig.4.5. Overheads and Gains in Shopping

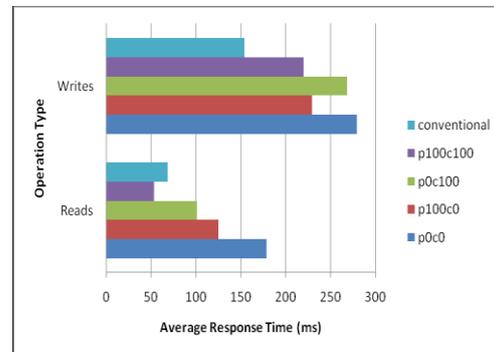


Fig.4.6. Overheads and Gains in Ordering

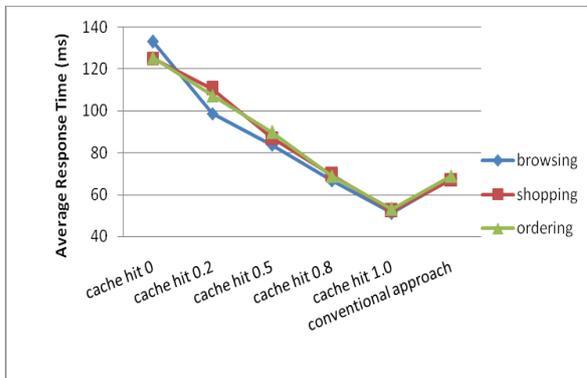


Fig.4.7. Performance of Cache for reads

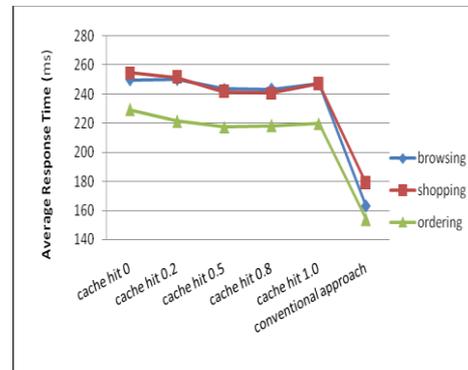


Fig.4.8. Performance of Cache for writes

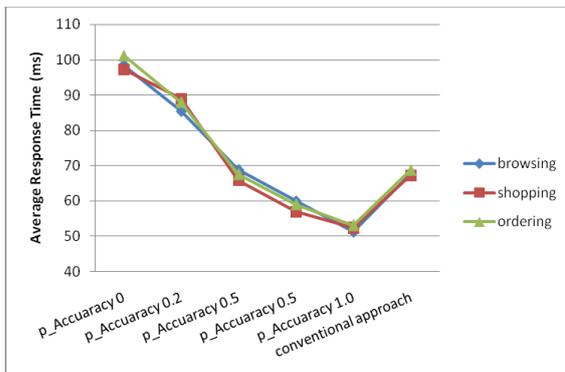


Fig.4.9. Performance of Prediction for reads

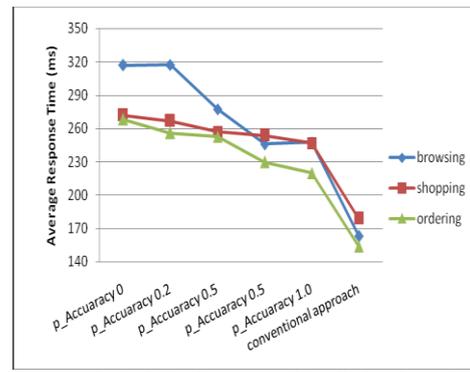


Fig.4.10. Performance of Prediction for writes

ACKNOWLEDGEMENT

I sincerely thank Dr. Ralph Deters, Professor, Department of Computer Science, University of Saskatchewan, Saskatchewan, Canada who was my guide for post graduate project work.

I sincerely thank Miss Yuting Tracy Leo for her kind help in my research. She was the post graduate student in department of computer science at the University of Saskatchewan, Saskatchewan, Canada.

REFERENCES

- [1] F. Curbera, M. Duffler, R. Khalaf, W. Nagy, N. Mukhi and S. Weerawarana. "Unraveling the Web Services Web. An Introduction to SOAP, WSDL, and UDDI". *IEEE Internet Computing*, vol. 6, no. 2, pp. 86-93, 2002.
- [2] "Web Services Description Language (WSDL) 1.1", Mar.2001. <http://www.w3.org/TR/wsdl>.
- [3] "Universal Description, Discovery and Integration (UDDI) v3.0", Feb. 2005. <http://www.uddi.org/>
- [4] "Simple Object Access Protocol (SOAP) Version 1.2", Jun, 2003. Available: <http://www.w3.org/TR/soap/>.
- [5] K. Barry, "Web Services and Service-Oriented Architectures", *Morgan Kaufman*, 2003.
- [6] Roy T. Fielding and Richard N. Taylor. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2):115-150, May 2002.
- [7] T. Pilioura, S. Hadjiefthymiades, A. Tsalgatidou, M Spanoudakis, "Using Web Services for supporting the users of wireless devices". *Department of Informatics and Telecommunications, National and Kalodistrian University of Athens, Panepistimopolis, TYPA building, Olisia, 157-84, Athens, Greece*, June 2005.
- [8] Thomas Erl, Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall, April 2004.
- [9] Market Share <http://www.gartner.com/it/>.
- [10] DVB Project, "Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H), ETSI EN 302 304 V1.1.1 (2004-11)". European Telecommunications Standards Institute, Nov. 2004. Available: <http://www.dvb-h-online.org>.
- [11] D. A. Menascé, "QoS Issues in Web Services", *IEEE Internet Computing*, vol. 6, no. 6, Nov./Dec. 2002, pp. 72-75
- [12] <http://dret.net/netdret/publications#wil08o>.
- [13] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann, Bernd J. Kramer, Service-Oriented Computing Research Roadmap, Service Oriented Computing(SOC), number 05432 in Dagstuhl Seminar Proceedings, Germany 2006.
- [14] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
- [15] <http://developer.apple.com/iphone/gettingstarted/docs/iPhonedevtools.action>.
- [16] http://en.wikipedia.org/wiki/iphone_OS.
- [17] <http://www.faqs.org/rfcs/rfc2396.html>.