The Conjuration Of Web Services And Aggressive Ascertainment From Devices

K.Priya, R.Kavitha

Abstract: - The Mobile devices are getting more pervasive, and it is becoming increasingly necessary to integrate web services into applications that run on mobile devices. I introduce a novel approach for dynamically invoking web service methods from mobile devices with minimal user intervention that only involves entering a search phrase and values for the method parameters. Such tasks involve communication with servers over the Internet, XML-parsing of files, and on-the-fly compilation of source code. It perform extensive evaluations of the system performance to measure scalability as it relates to the capacity of the MIM server in handling mobile client requests, and device battery power savings resulting from delegating the service discovery tasks to the serverOn the other hand, using specialized alternative protocols restricts compatibility with traditional service applications. My implementation is integration of web services that will run on our mobile devices. Web services invocation in mobile is a novel approach.

Index Terms—Web service discovery, dynamic invocation, mobile devices, mobile computing.

1. INTRODUCTION

1.1 Mobile Computing – Overview

Mobile computing is the discipline for creating an information management platform, which is free from spatial and temporal constraints. The freedom from these constraints allows its users to access and process desired information from anywhere in the space. The state of the user, static or dynamic, does not affect the information management capability of the mobile platform. A user can continue to access and manipulate desired data while traveling on plane, in car, on ship etc. Thus, the discipline creates an illusion that the desired data and sufficient processing power are available on the spot, where as in reality they may be located far away.

The discipline of mobile computing has its origin in Personal Communications Services (PCS). PCS refers to a wide variety of wireless access and personal mobility services provided through a small terminal (e.g., cell phone), with the goal of enabling communications at any time, at any place, and in any form. PCS are connected to Public Switched Telephone Network (PSTN) to provide access to wired telephones. PCS include high-tier digital cellular systems for widespread vehicular and pedestrian services and low-tier telecommunication system standards for residential, business, and public cordless access applications.

1.2 Mobile Web Service Overview

A web service typically defines an interface between two or more software applications. Nowadays, standard web service technologies can be applied to several mobile devices almost without any problems. Mobile web services currently form one of the most promising approaches to apply well established service-oriented concepts to mobile environments.Basic mobile web services framework with the Mobile Host update the contents to a standard server, with each update of the device's state.

Mobile Host is also possible with other Java variants like Java 2 Micro Edition (J2ME), for smart phones. I also have developed a J2ME based Mobile Host and its performance was observed to be not so significantly different from that of the Personal Java version. Mobile Host opens up a new set of applications and it finds its use in several domains like mobile community support, collaborative learning, social systems etc. Primarily, the smart phone can act as a multi-user device without additional manual effort on part of the mobile carrier.

1.2.1 Adaptation of Standard Web Service Technology

A standard web service technologies are WSDL, SOAP, HTTP and UDDI can directly be applied to mobile systems assumed that these are relatively powerful, use reliable network connections and able to provide adequate addressing mechanisms.

The basic Web services platform is XML and HTTP. XML provides a language which can be used between different platforms and programming languages and still express complex messages and functions. The HTTP protocol is the most used Internet protocol. Web services platform elements:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

Proxy Server Overview

A proxy server is a computer or usually set of computers that act as an intermediary between a client and a web server. It enables client computers to make indirect requests through it for resources or services such as web pages, videos, PDF files, etc. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server and the proxy server evaluates the request as a way to simplify and control its complexity.

Proxy server create a virtual private network that encrypts data as it travels between your mobile and the proxy servers. While you do this, you keep your IP address anonymous and protect your data, as it cannot be traced back directly to you. Proxy services act as an intermediary between you and the rest of the internet, using their servers to send your information to the web and then sending the response back to you.

Proxy services are similar to a Virtual Private Network, or VPN, in this way. Both serve the purpose of connecting you to a third party via something other than your own direct internet connection. A proxy service provides the added security of anonymity, and typically allows you to choose from various server locations while simultaneously cycling through IP addresses, ensuring that your connection is always safe.

1.3.1 Overview of Directory Proxy Server Features

Directory Proxy Server provides the following features:

- Manageability
- Authentication and authorization
- Distribution
- Load-balancing/Fail-over

2. RELATED WORK

2.1 Analysis of Related Works

To better understand of improving reliability and energy efficient, it is useful to review and examine the existing research works in literature. Therefore, recent approaches and methodologies used for improving reliability and energy efficient have been discussed.

M. Bellare, D. Pointcheval, and P. Rogaway(2002) Password-based protocols for authenticated key exchange (AKE) are designed to work despite the use of passwords drawn from a space so small that an adversary might well enumerate, offline, all possible passwords. While several such protocols have been suggested, the underlying theory has been lagging. We begin by designing a model for this problem, one rich enough to deal with password guessing, for-ward secrecy, server compromise, and loss of session keys. The one model can be used to define various goals. We take AKE with \implicit” authentication as the \basic goal and we give definitions for it, and for Entity-authentication goals as well. Then we prove correctness for the idea at the center of the Encrypted Key-Exchange (EKE) protocol of Bellovin and Merritt: we prove security, in an ideal-cipher model, of the Two-flow protocol at the core of EKE.

J. Cao, M. Andersson, C. Nyberg, and M. Kihl(2005) Performance modeling is an important topic in capacity planning and overloads control for web servers. We present an M/G/1/K*PS queuing model of a web server. The arrival process of HTTP requests is assumed to be Poissonian and the service discipline is processor sharing. The total number of requests that can be processed at one time is limited to K. We obtain closed form expressions for web server performance metrics such as average response time, throughput and blocking probability. The average of the service time requirement and the limit of the number of Requests being served are model parameters. The parameters are estimated by maximizing the log-likelihood function of the measured average response time. Compared to other models, our model is conceptually simple and it is easy to estimate model parameters. The model has been validated through measurements in our lab. The performance metrics predicted by the model fit well to the experimental outcome.

G. Dattatreya(2008) In this work develops simple models and analytical methods from first principles to evaluate performance metrics of various configurations of computer systems and networks. It includes analytically tractable models, and presents models for complex systems as analyzable modifications and interconnections of simple models. Naturally, there is a trade off involved in choosing the optimal quantum size. A very small quantum size increases the load of the system due to overheads, whereas a big quantum exposes the effects of job size variability. The goal of this paper is to address the following question: What is the optimal quantum size? To be able to answer the question of optimal quantum size, we must first consider the question, how does the sensitivity of the mean response time to the job size, switching cost whenever the server finishes processing one quantum of a job and starts processing a different job. For example, in an operating system, at every preemption, the kernel data structures managing the run queues have to be modified. Also, switching to a job involves waiting for the cache to be filled with the relevant data and instructions.

J. Kurose and K. Ross(2009) Mobile Computing consist of end systems, packet switches, and communication links. End systems—also called hosts—include desktop PCs, laptops, hand-held network devices (including cell phones, PDAs, and Black Berries, sensors, and servers (such as Web and mail servers). Just as cities are interconnected by a network of roads and intersections, end systems of a computer network are interconnected by a network of communication links and packet switches. Communication links can be wired or wireless. A computer network enables distributed applications. A distributed application runs on end systems and exchanges data via the computer Network. Distributed applications include Web surfing, e-mail, instant messaging, Internet phone, distributed games, peer-to-peer file sharing, television distribution, and video conferencing. New distributed applications continue to be invented and deployed on the Internet.

Frank T. Johnsen(2011) A Web services technology for use in military networks. It used Web services for point-to-point connections between different systems. In these experiments, we found that using optimizations such as XML compression and optimized transport protocols, as well as store-and-forward functionality were a necessity in order to enable Web services in disadvantaged grids. we have surveyed central Web services standards and specifications , and found that Web services are well suited for building not only traditional "pull" type systems (i.e., request-response operations), but also "push" type systems (i.e., event driven operations). Also, we investigated proxy servers (i.e., intermediate nodes between clients and services) and found that they can be used to provide added value operations, such as compression, content filtering, and so on.

A key concern when adopting NBD is to keep costs down by using COTS technology when possible. Some of the techniques discussed in this report break Web services standards, but this is necessary to get Web services. By implementing the optimizations in proxies, it can continue to implement and use COTS technology in clients and servers.

The proxies intercept standard Web services and perform the necessary optimizations on inter-proxy traffic. Content filtering reduces the total information overhead, leading to less information that needs to be transmitted across the network. In addition, we have investigated various ways of reducing the XML overhead by comparing the compression performance of several algorithms. It found that a generic by compression algorithm like GZIP compresses XML well, but that the emerging for XML compression, EFX, performs slightly better than GZIP.

3. METHODOLOGY

3.1 MIM

Man in- the-Middle (MIM) servers which will be used by mobile devices to discover needed web services and build their proxies. After getting the proxy, a mobile device can invoke a particular method of the web service and get the desired results. More specifically, the MIM server offers a web service which exposes a web method that the mobile device invokes and passes to it a search string.

The MIM server's special service (or simply the MIM server) compares the submitted string to cached short-descriptions of Internet web services and generates a short list of services that best match the user's string. Next, the MIM server downloads the WSDL files of the short-listed services, and uses the included descriptions of the supported methods to identify the most appropriate service (i.e., the one whose method's description matched the user's query the most).

After this step, the MIM server generates a source code file from the WSDL file of the chosen service and then compiles it using libraries that target the mobile device platform to generate the client-side proxy and ship it to the mobile device. At this point, the call that was originally made by the mobile device application to the MIM server's web method returns with information about the Internet web service.

This includes name of service and its chosen method, number and types of input method parameters, and number and types of returned results. With such information, the mobile application generates a dynamic GUI for the user to supply values for the web method parameters, and then another GUI to display the results.

The cached short descriptions mentioned above are downloaded by an independent process on the MIM server which periodically queries UDDI servers. To start with, the MIM server has a process that wakes up periodically to download a list of web service descriptions and associated URIs from a designated set of UDDI registries. More notably, the MIM server offers a web service that interfaces to three main processes which jointly fulfill the user's request. The first of these is the Text Matching Process which serves two purposes: first, to generate a short list of candidate web services based on matching their cached short descriptions with the user's supplied search string, and second, to identify the most appropriate web service among those short listed based on matching the methods' description found in their downloaded WSDL files with the user's string.

3.2 Cache Hit Rate

The MIM Server caches WSDL files for its own use to reduce network traffic and speed up the processing of the mobile device's request. The cached WSDL files form a subset of the set of possible WSDL files that may be downloaded. To get an idea of the size relationship, about 570 files could be cached if a 10 MB cache size is allocated and an average WSDL size of 18 KB is used. Since the space allocated for caching the WSDL files is finite, a mechanism is needed for cache replacement.

For this, I use a version of the least recently used (LRU) policy that works with objects having different sizes. That is, when a file is to be added to a full cache, more than one WSDL files may need to be removed in order to create sufficient space. The size and a time stamp for each cached file are stored in a hash table, and each time there is an access for a WSDL file, a counter is incremented and the time stamp is updated. The dynamic frequency of a given WSDL file is the inverse of the number of accesses since the last access. To create space for an incoming WSDL file with size Sn, we remove from the cache the least number of files whose cumulative size is greater than Sn such that the sum of their dynamic frequencies is the minimum. This removal policy will not change in a major way the access distribution of WSDL files in the cache relative to the files on the Internet because it is the files that fall on the tail end of the distribution curve that will tend to be selected for replacement, and thus, the probability of hit remains generally valid.

3.3 Scalability Analysis

To analyze the MIM server's scalability in terms of the number of users, the operations of the server main processes and describe quantitatively the interactions between each process and the underlying hardware resources.

In this analysis, to define the main three hardware resources that affect the server operation: memory, processor, and network. Storage utilization was ignored as it poses no bottleneck in current server implementations. In our analysis, it is convenient to model the processor and network performances using queuing theory, but first, we need to decide on the appropriate queuing model. Considering processor performance, it is well established that an M/G/1-RR (round robin) queuing model would be suitable.

It is designed for round-robin systems (like operating systems) and is generic, as it requires the mean and variance without the full distribution of the service time. This model assumes that requests to the processor follow a Poisson distribution, so that the distribution of the inter arrival time between requests is exponential with mean - requests/second, and each request is given a time slice on the processor.

3.4 Adaptability to REST

The major vendors continue to build the core web services stack around SOAP, but REST is preferred by certain groups. Recently examines for publishing REST web service descriptions, such as, and evolving technologies, like WSDL 2.0, make migration of our design to the REST style quite possible.

The new WSDL 2.0 standard, which was designed with REST web services in mind, includes the semantics for describing such services. Therefore, to support REST ful services in parallel with REST less services. which is the likely future scenario, the

MIM server can access a system like SOA live to download REST service descriptions and use them for short listing services based on the user request, and utilize the WSDL 2.0 file to identify the desired REST service. Also similar to SOAP services, this WSDL 2.0 file can be used by a tool, such as Axis 2 from Apache, to generate a proxy class which can be used by the mobile device in the same manner as that generated from the SOAP service WSDL file. The only difference is that the proxy class generated from the REST WSDL will use the HTTP libraries, instead of the SOAP libraries.

3.5.2 Features and Specifications

Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardwares that support Android are mainly based on ARM architecture platform. Some of the current features and specifications of android are:

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

4. EXPERIMENTS AND RESULTS

Fig 4.1 Home page of the android mobile device

In this above screen shot it shows the home page of the android mobile device. In a lock screens on mobile devices often provide more functionality beyond unlocking the phone, such as notifications for email and text messages, a date and time display, or shortcuts to certain applications.

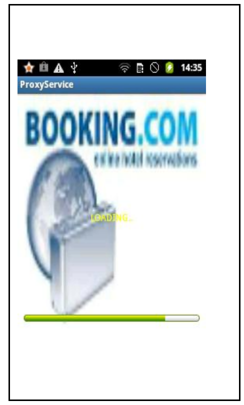

Fig 4.2 Airline Booking Application is loading

In this screen shot Airline booking application is loading in android mobile.

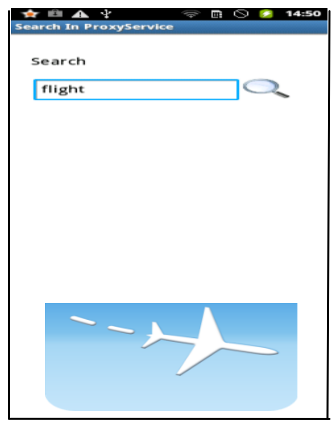

Fig 4.3 Airline Booking Search

In this screen shot , it seeks the related content and it performs first round of string matching.

Fig 4.4 Airline Booking Form

In this Screen Shot , the user can book the ticket to filling the booking form like the user name , email-id, source and destination .

5. CONCLUSION

The presented architecture makes it possible for mobile device users to dynamically invoke web service methods that meet their needs. The implemented solution overcomes technical limitations, and also saves device battery power, thus extending its participation in the wireless network. The scalability study can be used to decide on deployment of MIM servers in the network: given the capacity of the server, the number and distribution of MIM servers can be determined, knowing the cumulative expected request rate from users.Our design provides web service discovery services to personal applications running on mobile devices, where individual services can be used to extend the functionality of such applications. However, nothing precludes these applications from accessing composite web services that perform computationally intensive tasks, as in bioinformatics, data mining, and multimedia processing. But, since web service entities are usually autonomous and heterogeneous, how to connect and coordinate them is a challenging task that is not suited for mobile devices. As a future work, the Cloud Server can be programmed with the intelligence to identify a set of services whose collective functionality can serve the user's request. In fact, the Cloud Server may be suite to coordinate the functions of such services and provide an interface to the mobile device which is consistent with the current design.

REFERENCES

1) J. Cao, M. Andersson, C. Nyberg, and M. Kihl, "Web ServerPerformance Modeling Using an M/G/1/K*PS Queue" Proc. 10th Int'l Conf. Telecomm.

2) M. Chatti, S. Srirama, D. Kensche, and Y. Cao, "Mobile Web Services for Collaborative Learning" Proc. IEEE Int'l Workshop Wireless Mobile and Ubiquitous Technology in Education, pp.129-133,Nov. 2006.

3) R. Costello, "Building Web Services the REST Way" http:// www.xfront.com/REST-Web-Services.html, 2011.

4) G. Gehlen and L. Pham, "Mobile Web Services for Peer-to-Peer Applications" Proc. IEEE Conf. Consumer Comm. and Networking, pp. 427-433, Jan. 2005.

5) V. Gupta, "Finding the Optimal Quantum Size: Sensitivity Analysis of the M/G/1 Round-Robin Queue" ACM SIGMETRICS Performance Evaluation Rev., vol. 36, pp. 104-106, 2008.

6) A. Halteren and P. Pawar, "Mobile Service Platform: A Middleware for Nomadic Mobile Service Provisioning" Proc. IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Comm. (WIMOB), 2006.

7) R. Lee and R. Nathuji, "Power and Performance Analysis of PDA Architectures" technical report, MIT, http://www.cag.lcs.mit. edu/6.893-f2000/project/lee_final.pdf, Dec. 2000.

8) L. Li, M. Li, and X. Cui, "The Study on Mobile Phone-Oriented Application Integration Technology of Web Services" Proc. Int'l Conf. Grid and Cooperative Computing (GCC), Apr. 2004.

9) Y. Li, Y. Liu, L. Zhang, G. Li, B. Xie, and J. Sun, "An Exploratory Study of Web Services on the Internet" Proc. IEEE Int'l Conf. Web Services (ICWS '07), pp.380-387, 2007.

10) O. Rendo´ n, F. Pabo´n, M. Vargas, and J. Guaca, "Architectures for Web Services Access from Mobile Devices" Proc. Third Latin Am. Web Congress (LA-WEB '05), pp. 93-97, 2006.

11) R. Steele, K. Khankan, and T. Dillon, "Mobile Web Services Discovery and Invocation through Auto-Generation of Abstract Multimodal Interface" Proc. Int'l Conf. Information Technology: Coding and Computing (ITCC '05), vol. 2, pp. 35-41, 2005.

12) Sun Microsystems, "JINI Technology Surrogate Architecture Specification" http://surrogate.JINI.org/sa.pdf, Oct. 2003.

13) E. Sa´nchez-Nielsen, S. Martı´n-Ruiz, and J. Rodrı´guez-Pedrianes, "Mobile and Dynamic Web Services" Emerging Web Services Technology, pp. 117-133, Birkha¨user, 2007.

14) Q. Sheng, B. Benatallah, Z. Maamar, and A. Ngu, "Configurable Composition and Adaptive Provisioning of Web Services" IEEE Trans. Services Computing, vol. 2, no. 1, pp. 34-49, Jan.-Mar. 2009.

15) I. Silva-Lepc, R. Subramanian, I. Rouvcllou, T. Mikalson, J. Diament, and A. Iyengar, "SOAlive Service Catalog: A Simplified

ISSN: 2278 – 7798

Approach to Describing, Discovering and Composing Situational Enterprise Services" Proc. Int'l Conf. Service Oriented Computing (ICSOC '08), pp. 422-37, 2008.

16) W3C, "XForms - The Next Generation of Web Forms", 2007.

17) R. Wolff and A. Schuster, "Association Rule Mining in Peer-to- Peer Systems" IEEE Trans. Systems, Man, and Cybernetics, vol. 34, no. 6, pp. 2426-2438, Dec. 2004.

18) Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and Managing Web Services: Issues, Solutions, and Directions" The Very Large Databases J., vol. 17, no. 3, pp. 537-572, 2008.

K.Priya, currently pursuing M.Phil Computer Science under one of the college affiliated to Periyar University. I also received my BCA and MCA degrees from the affiliated Colleges under Periyar University and Anna University.

I have done two projects during my Post Graduate.

R.Kavitha, currently working as an Assistant Professor in Department of Computer Science in Vivekananda College for Women.

ISSN: 2278 – 7798