Cloud Computing in Mobile Networks with Middleware Technology

G.Srinivas Reddy[#1], D Khalandar Basha[*2]

[#1,] *Mahatma Gandhi Institute Of Technology, Hyderabad-INDIA*

[#2]*Institute of Aeronautical Engineering, Hyderabad-INDIA*

***Abstract*—The Web Services are of a vital role in the domain of mobile networks for their utility in various applications of communications field. However there are more challenging issues of functionalities of mobile technology with the current Web based resources. The work in the present paper deals with architectural implementation methodologies of the mobile technology for the enhancement of utility of the web services at the level of more reliability and scalability. Constrained resources, interruption of the connection and bandwidth limitation are the issues elevated in the attempt of the work. The middleware technology shows a way of convenient utility of personal interaction with the appropriate platforms for the betterment in the mobile service in the cloud platform. Mashup plotform with middleware in the cloud environment is found to be the most appropriate methodology for the reliable mobile service even with the limited resources of bandwidth and uninterrupted services.**

***Keywords*— Cloud computing, Middleware technology and Mobile architecture.**

I. Introduction

As the usage of hand phones increases continuously, the utility of Web services for the data transmission also needs the improvement of network infrastructure. There seems to more consumption of websites like Twitter, Google, and Facebook [4]. Services of mobile network are of many problems in the present Web Service Technology. It needs to optimize certain parameters like number of messages that are to be adjusted according to bandwidth, and the knowledge of customers to access the various kinds of Web Services. The customization and services of mashups meet the purpose of providing the best services to mobile clients increasing in large number [2]. The paper reveals implementation of cloud to improve the services of Web resources for the mobile clients. A process of interaction among the large number of machines with a networking mechanism can be treated as a Web Service. This interaction is incorporated by a certain standards and protocols using an XML and HTTP with other Web-oriented conditions [1].

II. Problem Statement and Aim of the paper

In the present attempt of this work an architecture is proposed in which middleware technology is used for supporting the mobile clients in accessing the services of cloud platform. The cloud computing is of flexibility of running the task on their Virtual Machines in different Operating Systems and hardware architectures. EC2 and GAE are the popular cloud platforms. These are supporting for the mobile clients for accessing the services under the limits of free of charge [6]. The objective of the attempt in the paper is to design a configuration and the architecture so that following can be feclitated.

- An establishment of connectivity between mobile clients and cloud Services
- Client and middleware technology connection
- Middleware involvement
- Protocol routing transformation method
- Getting an optimized outcome of the process

Issues of challenges in establishing the connectivity of Web Services and mobile clients.

- Challenge1. Stability of Connection: The interaction between client and cloud service needs a stable transmission of information among the clients and cloud service. Because of improper connectivity some times response or request may fail.
- Challenge 2**.** Bandwidth: It is more important to avoid simple SOAP messages that are not essential for clients and that may cause major network latency due to the consumption of large bandwidth for more XML data.
- Challenge 3. Limited power consumption: The solution aims at facilitating the data processing of mobile and enhancing the limitations of the power functioning beyond mobile clients.

III. Design and Architecture

The main thing to be achieved with the Mobile Cloud Computing (MCC) architecture is that a proxy is to involve in the process for the connectivity of mobile clients to Cloud services. Figure 1 gives outline of the MCC and components in the design. The architecture composed of three components, the mobile clients, the middleware partition and the Cloud services. As Cloud services are generally under control of service vendors, it is very comfort for the adjustments of processing methodologies in the middleware for the better connectivity to mobile clients. The middleware also excels advancements in the service to mobile clients [9].

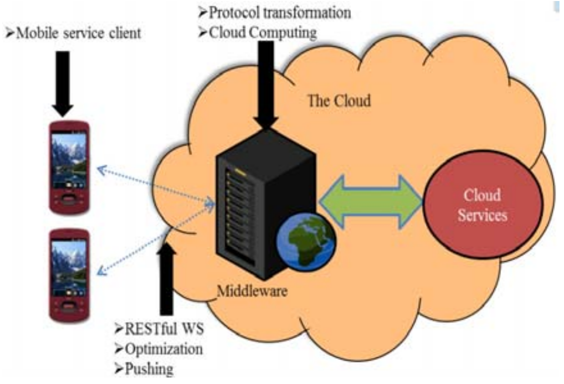

Fig 1 Outline of MCC

Figure 4.2 gives the details of the consumption of a pre-designed WS. It is observed that the simulation begins with a HTTP GET request. URL path of this request is already in the resource identifier to the WS. When WS are run through the middleware, the process is excuted with the certain steps are carriedout in the middleware.

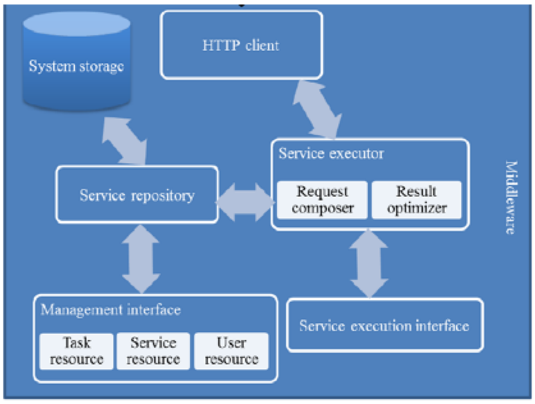

Fig 2 Mobile client architecture

Figure 2 gives a brief of the client architecture. It is a basic Model View Controller (MVC) pattern. The User Interface (UI) is framed in the embedded browser with the help of JavaScript, HTML, and CSS. When the UI components are indeed service data, they enable the custom JavaScript libraries to extract the data from local cache. Unless the local cache recent copy of inquired data, the RESTful client connects the middleware to process the data. Then the data are applied to the module and it gets stored in the local file system.It is a common thing that the data fed to the embedded browser is associated to JSON format [5]. The figure 2 shows Service Entities in hierarchical way that are designed as per the planning popose in the paper.

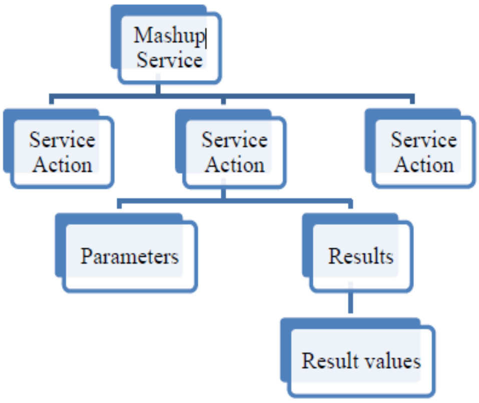

Fig 3 Hierarchical View of Service Entities

IV. Design and Architecture

The implementation includes Javascript libraries for the execution of different kinds of functionalities of the Blackberry OS [10]. The investigation of implementation has been made with the design of an existing iPhone application and Blackberry. The application that may be nedded for the customers can be categorized into three parts of the archecture, controlling function, cache technique for managing the tasks and User Interface (UI). Figure 4 shows the architecture of implementations. The controller is the very important coordinator among middleware, cache manager and UI. The controller establishes the UI and extracts data from the RESTful client or cache manager. Unless network connections are available, the controller is fed to cached data with the UI components. However, it enables the RESTful client to access data from the middleware. The cache manager stores recent newly processed data on a local file system [7].

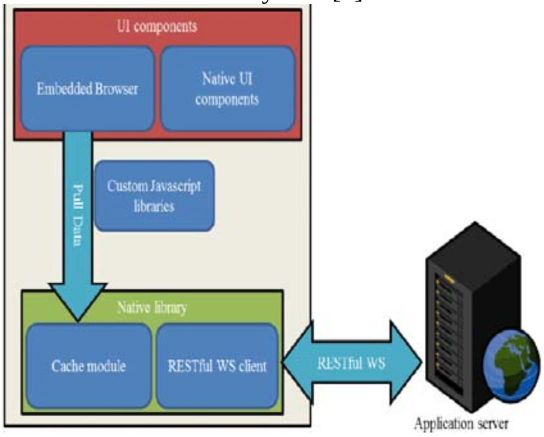

Fig 4 Native UI implementation

A. *The role of Amazon in network of EC2*

Most Java developers implements the middleware architecture as a Java Web application. The application provides RESTful WS interfaces to mobile clients [17]. The middleware is to be implemented on a Java HTTP server container [3]. The middleware makes the WS return XML responses, so that results can be taken out using the Java build-in XPath library.

V. Experimental Setup and Analysis

A. *Mashup Service through the Middleware Applied on EC2*

Experiment establishes and consumes a mashup services through the middleware hosted on EC2 in comparison with direct accessing the mashup service on the client side. The mashup service integrates two Yahoo Upcoming services [8].

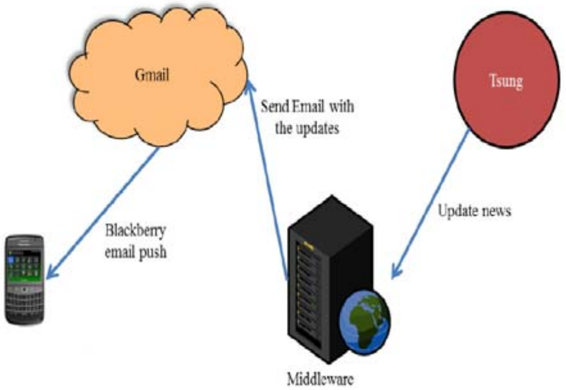

Fig 5 Email Push on the system of Blackberry

The mashup service identifies all events in the nearest city with reference of the user location. The middleware is implemented on an EC2. First, the client sendsrequest of HTTP POST and acknowledges the mashup service. The will generate some IDs . HTTP POST generates SERVICE

ACTION and REQUEST ACTION. *Figure5* depicts the overhead of consuming the mashup service on middleware series and client series [7]. The x-axis is the number of simulations carried out on the mashup service. The time period between sample request is 3 minutes. Processing time is taken on Y-axis. The average response time of the middleware mashup is 723ms with a standard deviation of 85. The average response time of client side mashup is 932.6ms with standard deviation of 92.4. Both of the two series are found to be of a lot of variations due to network latency. The result of simulation gives the details that the mashup implementation technique proposed in the paper is found to be a powerful methodology in the access of more bandwidth and processing power at optimum level. *Figure 5* gives the processing of the middleware on a mashup request. When the middleware receives a input of GET request, it first processes service results from the Cloud Services. The evaluation of CPU computation has been made. The analysis has been made for the evaluation of the time process in both Middleware and without Middleware deployment.

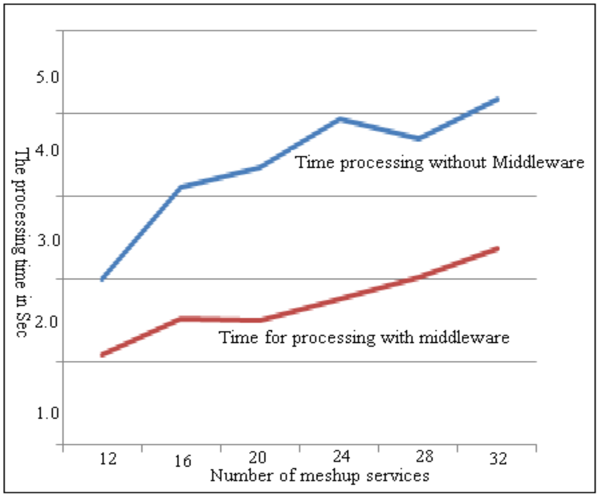

Fig 6 Processing time of services in the system of mashup

The duration of each load is 7 minutes. Response time is calculated every 3 second. HTTP requests will taken on the x-axis. The figure shows the comparison of processing time. So that the advantage of the applied technique can be justified its power in the services of its reliability and fast functioning of mobile service.

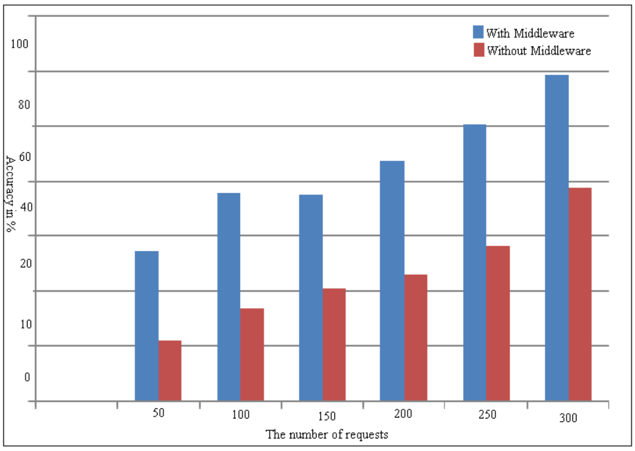

Fig 7 Response Time for Elastic Cloud

The figure 7 reveals the percentage of accuracy in its performance with proposed method in the paper. It is coined in work that Middleware implementation enhances the the functionality of the mobile services in cloud environment. EC2 can only have requests in the range of 50-300 per second without considerable errors due to the constraints of the Java programming. With 50request/s, the average response ime is 97s and half of the requests failed or dropped. The highest and average response time is very high, when the request rate is maximal. This can be revealed in the cloud mode of EC2. With the same availability of resource, it is expected that the response time is high as the load becomes larger.

VI. Analysis and Discussion

Mobile devices have limitations of less process power, bandwidth and loss connectivity. To overcome the limitations, the paper presents the Mobile Cloud Computing architecture for the connectivity of mobile device to the Cloud Services. Mashup services gives an interaction with middleware for the consumption of cloud services by the mobile clients. The middleware provides flexibility of adjusting the requirements for mobile clients to Cloud Services. The middleware pushes to efficiently deliver content to the mobile client through email. The middleware is a Personal Service Mashup Platform which fecilitates personal service mashup for mobile clients. Users can access mashup services on the middleware. The experiments suggests the following steps in designing an architecture for the mobile client and middleware.

- There is more accuracy in the enhancing the processing mechanism.
- There is considerable reduction of processing time in mobile functioning in cloud with Middleware implementation.
- The mobile client can access its applications on different mobile platforms.
- It is very easy and advantage to do mashup on the middleware side than the client-side.
- Implementation of the middleware process can be possible with GAE and EC2.
- As GAE is very much scalable, it becomes very easy for users to make system to be processed on the virtual machine in the environment of EC2.

References

[1] S. Wang and X. S. Wang, “In-Device Spatial Cloaking for Mobile User Privacy Assisted by the Cloud,” in Proceedings of the 11th International Conference on Mobile Data Management (MDM), pp. 381, June 2010.

[2] W. Itani, A. Kayssi, and A. Chehab, “Energy-efficient incremental integrity for securing storage in mobile cloud computing,”International Conference on Energy Aware Computing (ICEAC), pp. 1, January 2011.

[3] J. Shen, S. Yan, and X-S. Hua, “The e-recall environment for cloud based mobile rich media data management,” in Proceedings of the 2010 ACM multimedia workshop on Mobile cloud media computing, pp. 31-34, October 2010.

[4] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," in Proceedings of the ACM workshop on Cloud computing security, 2009, pp. 85-90.

[5] H. XuhuiLi and Y. Zhang, "Deploying Mobile Computation in Cloud Service?" in Proceedings of the First International Conference for Cloud Computing (CloudCom), 2009.

[6] Website available at http://www.abiresearch.com/research/1003385-Mobile+Cloud+Applications

[7] M. Samimi, F.A., Mckinley, P.K., Masoud Sadjadi, S.: Mobile Service Clouds: A self-managing infrastructure for autonomic mobile computing services. In: Proc. of the Second IEEE International Workshop on Self-Managed Networks, Systems and Services (2006)

[8] *Blair, G., Coulson, G., Robin, P., and Papathomas, M. 1998. An Architecture for Next Generation Middleware. In Proceedings of Middleware '98 (Sept. 1998), pp. 191{206.Springer Verlag.*, 1996.

[9] Capra, L., Emmerich, W., and Mascolo, C. 2001a. Middleware for Mobile Computing:

[10] A Demers, A., Petersen, K., Spreitzer, M., Terry, D., Theimer, M., and welch, B. 1994.The Bayou Architecture: Support for Data Sharing among Mobile Users. In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (Santa Cruz, Cali-fornia, Dec. 1994), pp. 2{7.142Switzerland.