Performance Efficient Application Development in Mobile Cloud Computing

V. Vijayalakshmi
Assistant Professor, Dept. of CSE,
Christ College of Eng. & Tech,
Pondicherry, India.

D. Sundaranarayana
Final Year M.Tech, Dept. of CSE,
Christ College of Eng. & Tech,
Pondicherry, India.

I. Shamili
Final Year M.Tech, Dept. of CSE,
Christ College of Eng. & Tech,
Pondicherry, India.

Abstract: **Mobile cloud computing (MC2) is emerging as a new computing paradigm that seeks to augment resource constrained mobile devices for executing computing and data-intensive mobile applications. However, Mobile devices have limited resources such as memory, processing power, and battery. Moreover, complex applications may not be deployed on these devices. One of the remedy is to offload some functionality to more powerful servers to yield improved performance and low resource consumption [7]. The operation of the mobile computing applications which is relied on the server maintenance is not taken into consideration for improving the quality and performance of the mobile computing applications. In this paper we present two similar strategies such as execution of the mobile application data will be done on the mobile device and another one is the execution of the mobile application data on the cloud server. These strategies are used to avoid complex compression offloading mechanisms. The above two strategies can be compared to determine the performance efficient one such as one of them may be efficient for small applications and the other for large intensive applications. The results may vary according to number of resources in an application. Also in this paper we present a technique known as Dot Net Nuke (DNN) for managing and updating the mobile server as per the requirement of the user frequently without coding knowledge. This leads to improve the quality and updated operations on mobile computing.**

Keywords: Mobile Cloud Computing (MCC), Dot Net Nuke (DNN), Facility Management System, Offloading.

I. INTRODUCTION

Mobile Cloud Computing (MCC) is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to mobile users, network operators, as well as cloud computing providers. The precise goal of MCC (mean of MCC is Mobile Cloud Computing) is to enable execution of mobile applications on a large scale of mobile devices, with a rich user experience [5]. MCC provides business opportunities for mobile network operators as well as cloud providers. MCC can be defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle."

MCC uses augmentation approaches by which resource-constraint mobile devices can utilize computational resources of varied cloud-based resources. In MCC, there are four types of cloud-based resources, namely distant immobile clouds, proximate immobile computing entities, proximate mobile computing entities, and hybrid (combination of the other three models) [1]. Popular cloud such as Amazon EC2 is in the distant immobile groups whereas cloudlet or surrogates are member of proximate immobile computing entities. Smartphones, tablets, handheld devices, and wearable computing devices are part of the third group of cloud-based resources which is proximate mobile computing entities.

MCC is a landscape, where a combination of mobile computing, cloud computing, and communication networks creates several complex challenges such as Mobile Computation Offloading, Seamless Connectivity, Long WAN Latency, Mobility Management, Context-Processing, Energy Constraint, Vendor/data Lock-in, Security and Privacy, Elasticity that thwart the success of MCC and its adoption. The Architecture of Mobile Cloud Computing is simply the combination of the mobile computing architecture and the cloud computing scenario. The mobile devices access their servers or database normally through their access points or Base stations. In mobile Cloud Computing the Mobile devices access the cloud through the internet/LAN.

II. STRATEGY FOR MOBILE APPLIATION DEVELOPMENT & SERVER ENHANCEMENT USING DNN (Dot Net Nuke)

The Mobile Application development deals with certain kind of strategies for improving the performance of the mobile application which is also concerned with server enhancement [3]. Although several initiations have come into existence regarding the mobile application development, the performance testing under different size

of application are yet to be known. These testing are done upon two strategies namely Thin-Client & Thick-Client Execution of mobile application.

A. Thin-Client & Thick-Client Strategies For Mobile Application

The system consists of two similar Strategies for identifying the performance efficient method i.e. the response time for executing the process of each mobile application. The motive is to reduce the execution time of the mobile application processing by executing them in the below mentioned strategies.

i) Execution will be done on the mobile device (Thick-Client).
ii) Execution will be done on the mobile-cloud server (Thin-Client).

In the Thin-Client Method the mobile application which is created viz. Facility Management System (FMS), every request on the UI goes to the server for each and every transaction. Processing in client is not done and the mobile device experiences a minimalized usage. Similarly the Thick-Client method gets update from the server periodically so that the processing is done within the mobile device [3]. Comparison is done with the above two process to determine which one is suitable for which size of application. i.e. The same mobile application which is developed , is executed under both of the environment to measure its performance under different scenarios. One of them will be performance efficient for large set of application and the other will be efficient for small set of application. The results may vary according to number of resources (Size) in an application.

B. Server Enhancement By DNN (Dot Net Nuke)

This Deals with creating and maintaining a server side application with which the mobile application communicates to do its processing. This server side application is developed using a platform known as Dot Net Nuke (DNN).

It is a content management platform which allows the user to create and maintain web servers using Portal solutions. The content of the server application, editing and updating operation are easily done by the user as DNN supports it. This leads to improve the quality and updated operations on mobile computing.

DNN act as a *cloud* since it provides software services through different ports to different number of users. Portal solutions are those which provide service through different ports to same or different number of users. Through that port we are developing a web server application and making it to communicate with the mobile application.

The decision of whether to partly/fully utilize the cloud server is distinctly classified and also short task executions do not take place as like that of sample tasks in existing work [6]. The Proposed strategies does not involve with complex terms such as compression offloading to the server and the response time of the application is calculated and identified under different scenarios. Using DNN allows the modification and maintenance of the web server without any separate team and resources. This Manages and updates the server without taking longer durations, which results in updated operations on using mobile application. Coding Knowledge is not necessary to do so in the server, since DNN supports such operations. Maintenance team or any separate solutions are not necessary to maintain the web server which results in saving money, time and resources.

III. METHODOLOGY

The Significant purpose is to improve the performance of the mobile application processing. Previous systems uses compression algorithm for offloading the mobile application data to an external cloud, so that the data being transferred to the cloud for processing is reduced in size which may be suitable for applications with large intensive data sets. But the compression offloading scheme is an unnecessary action for small type of data applications which leads to additional complexities [6]. The proposed work excludes the compression algorithm and provides two strategies for identifying and minimizing the execution time of the mobile application. We also believe firmly that reducing the execution time of the mobile application will save the energy of the mobile device.

A. Web Server Application Development

The Server side application is developed by Dot Net Nuke for facility management system, with which the mobile application communicates.

The capabilities of DNN are discussed above with which the development of server makes it flexible and update intensive. The facility management system is built as both server application using DNN and in Mobile Application Using Android Studio. The FMS which is built for server application includes modules such as Resource Manager, Resource rate management and Booking reports.

These sub-modules are used for maintaining and updating the web server in minute's time, which are listed below along with their functions.

1) Resource Manager: The Resource manager module contains data of the resource location (place of location of the room, hall etc.), the particular resource, resource type and manages these entire datum. The resource manager distinctly classifies the location of the facility and the type of the facility to maintain them easily.

2) *Resource Rate Management:* It is because each and every resource (facility) differs from each other so that their rate of usage is also not similar. The frequency of the usage is based on the price and also the requirement of the individual organizations. The data of price of the resource

is maintained and classified by the resource rate management.

3) Booking Reports: The booking reports module consists of data of the booking/booked facilities by the mobile clients. It manages the booked resources by maintaining the records of the booking and it also helps to avoid overlapping of the booked facilities with more than one user.

B. Performance Evaluation

The Mobile application for FMS is developed using Android Application Studio. The application is developed for executing under both of the strategies such that the same application's performance under different scenarios can be evaluated. The execution time of the application can be identified in both strategies and so we can conclude that which strategy will be suitable for developing large intensive applications and which would rather be suitable for developing small applications. By identifying this, the future development of mobile applications could be more refined in their execution criteria according to their sizes. The mobile application development for FMS mainly involves with two sub-modules.

1) Display Resources: The Display resource is a client side module where this displays the facilities/resources available for the mobile user to make their booking. In the server side execution strategy, the display resource module retrieves every information from the server before displaying the available resource to the user. Therefore the communication happens frequently between mobile client and server.

Whereas in the Client Side execution strategy the display resource module does not use the server for displaying the available resource but the updates are there in the mobile itself to display the available facilities.

2) Book Resource: The Book resource module is for the actual booking of the available facilities and it syncs with the server side execution modules in order to make necessary updates in them.

IV. SYSTEM DESIGN

The Architecture of server side execution is as follows

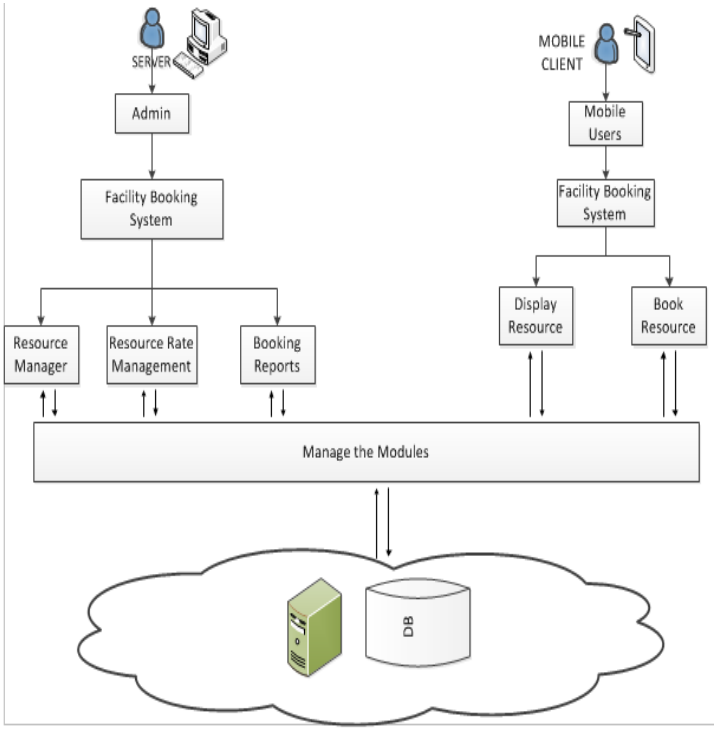

Fig. 1. Architecture of Server Side Execution.

This Architecture displays the server side execution strategy where the mobile users when using the FMS system, each time the display of the available resources go to the server as a request.

It comes back as a response to the user and displays the list of available resources for facility booking. Similarly for each and every operation which is done by the mobile user goes to the server and comes back to the mobile display. These modules and updates are stored back in the database periodically.

The Architecture of Client Side execution is as follows

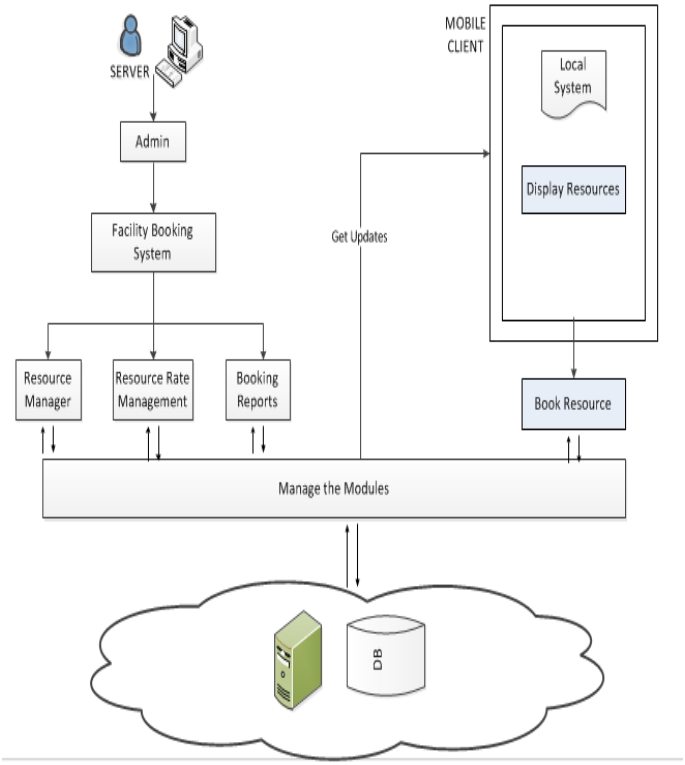

Fig. 2. Architecture of Client Side Execution.

This Architecture displays the Client side execution strategy where the mobile users when using the FMS system, the updates from the web server application are updated to the mobile device itself. So the user request for displaying the resource available for facility booking does not goes to the server but displays within the mobile device itself. Only at the final stage of processing the booking it goes and communicates with the server. The local system has an important role in this strategy.

V. CONCLUSION

In this paper, we propose a framework for performance efficient strategy for the execution of the mobile applications. The proposed framework consists of two strategies in which one of them uses the server for every execution of application processing. The other one is a client/ mobile oriented strategy which do not rely on server for each executions of the process. These two strategies are proposed to overcome compression complexities of the existing system. This method gives a broader choice for mobile application developers who create different size of applications to decide which strategy would be efficient to develop the application execution method. The proposed work also executes the developed mobile application in both of the strategies in order to compute which strategy spares less execution time for the processing of the application. The motive is to identify the strategy to minimize the processing time of the application and there is a firm belief that saving the processing time will also save the energy of the particular mobile device. Also the development of the web server (server side application) using DNN platform makes it easy for the users to maintain their own server/websites without having a maintenance team or additional resources. This would result in updated mobile computing operations.

REFERENCES

[1] K. Liechtenstein, E. Andre, "User-centered development of mobile interfaces to a pervasive computing environment," in Proceedings of the First International Conference on Advances in Computer–Human Interaction, pp. 114–119.
[2] J. Landay, T. Kaufmann "User interface issues in mobile computing," in Proceedings of the Fourth Workshop on Workstation Operating Systems, IEEE, 1993, pp. 40–47.
[3] K. Elbashir, R. Deters, "Transparent caching for nomadic WS clients," in Proceedings of the IEEE International Conference on Web Services, ICWS, 2005, vol. 1, pp. 177–184.
[4] M. Satyanarayanan, "Fundamental challenges in mobile computing," in Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, PODC'96, ACM, New York, NY, USA, 1996, pp. 1–7.
[5] G. Huerta-Canepa, D. Lee, "A virtual cloud computing provider for mobile devices," in Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, MCS'10, ACM, New York, NY, USA, 2010, pp. 61–65.
[6] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, "An autonomic cloud environment for hosting ecg data analysis services," Future Generation Computer Systems, 2012, pp. 147–154.
[7] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, "Maui: making smartphones last longer with code offload," in Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys'10, ACM, New York, NY, USA, 2010, pp. 49–62.
[8] R. Balan, M. Satyanarayanan, S. Park, T. Okoshi, "Tactics-based remote execution for mobile computing," in Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, ACM ,2003, pp. 273–286.
[9] L. Deboosere, P. Simoens, J.D. Wachter, B. Vankeirsbilck, F.D. Turck, B. Dhoedt, P. Demeester, "Grid design for mobile thin client computing," Future Generation Computer Systems, 2011, pp. 681–693.
[10] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in Proceedings of the Sixth Conference on Computer Systems, EuroSys'11, ACM, New York, NY, USA, 2011, pp. 301–314.