OFFLOADING EXECUTION LOAD FROM A RESOURCE-CONSTRAINED DEVICE

Shruti Chordia
Student of P.V.P.P.C.O.E.

Niranjana Gajjelli
Student of P.V.P.P.C.O.E

Mr. Vaibhav Narawade
Associate Professor,
P.V.P.P.C.O.E., Mumbai

Abstract: **Modern smartphones have made many pervasive computing dreams come true. Still, many mobile applications do not perform well due to the shortage of resources for computation, data storage, network bandwidth and battery capacity. The purpose of our project is to provide the user with the capability to execute resource intensive applications on mobile devices. The mobile client provides input output interfaces while the processing is done in a remote location which is the server. Our application provides abstraction by hiding the processing from the user and only showing the input output interfaces. This makes user to think that all the processing is done on the mobile device itself. The application provides an environment where resource intensive applications, that required desktops or laptops to execute, can now be executed on mobile devices. This gives mobility to user as he can execute these applications whenever he wants through mobile.**

I. INTRODUCTION

Mobile devices such as smart phones have become ubiquitous. The article summarizes various uses of smart phones in class rooms as replacements of laptops. We are looking at using existing technologies to make smart-phones even useful by allowing them to run applications they are traditionally not built for. Mobile devices are unable to execute resource-intensive applications because of their low processing power, limited battery life or lack of software support.

Current approaches require development of special applications for different mobile operating systems such as Windows Mobile, iPhone O.S., and Android which is cumbersome and results in duplicate work because each smart-phone seems to support its own standard operating system. Furthermore, we speculate that resource intensive applications may simply not be able to execute efficiently on the processor of these mobile devices or may lack crucial software support to execute. For instance, many older smart-phones and certain new ones (such as iPhone) do not support java runtime environments.

A solution we are interested in is to provide an application which is small footprint of the main application on the mobile devices. This application while maintaining the same input and output interfaces as the main application will however, process the output at a remote location. The remote location can be one or more work stations provided in a classroom with software and other resource support needed to run the application. The contribution of our proposed research is

- Framework to support resource intensive application on mobile devices,
- Identifying the issues in developing and implementing such a framework, and,
- A survey of technologies available for supporting this framework.

II. LITERATURE SURVEY

Resource Intensive Applications are the applications that require software libraries, high computational power and high battery life. Executing Resource Intensive Application on Mobile Devices [1] works on the concept of offloading the resource intensive part of the application to remote location. The application makes the client a thin client. A thin client [3] is the one which depends heavily on some other server to fulfill its computational roles. Hypertext Transfer Protocol [4] is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web [15]. An assessment between the HTTP and FTP is done, in order to decide the best among them [6]. The suitable applications that are resource-intensive for mobile devices [8] [14] are then selected. We studied the progression of offloading the processes on remote locations i.e. the server [13]. Mobile communication using wireless media is done thoroughly for appropriate medium of communication [12]. How to work with TCP/IP connection for implementation purposes [11]. How the complete networking does works between the server and the client [10].

III. METHODOLOGY

A. *Study and analyze the application framework.*

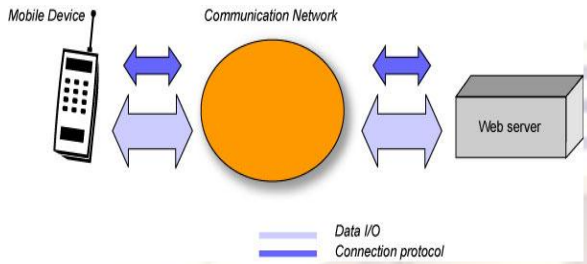

Fig.1 Application framework

We use client-server architecture for the purpose. The client is a thin client mobile device that communicates with the fat server via an internet medium to view the desired results for the input provided [16].

B. *Study of protocols such as Hypertext transfer protocol (HTTP), File transfer protocol (FTP).*

HTTP is convenient over the FTP in respect with the transfer speed of multiple files, meta-data about the files sent, persistent connections, and support for proxies, range, compression algorithms used [6].

C. *Finding resource intensive applications for mobile devices.*

Two resource-intensive applications we are interested in are: 1) The image processing application and 2) The database query analyzer. These need software support that could not be made available on the mobile device.

The image processing application would provide the service of carrying out the point processing, neighborhood processing and so, on the image given as input by the user. Whereas, the database query analyzer would help the user query a user defined database and display the result of the query on the user's mobile device.

D. *Designing the application.*

Applications are designed using the JEE SDK 6u2 version. The sun java wireless toolkit is used for the purpose with the default specifications provided by sun.

E. *Creating a Hypertext Transfer Protocol (HTTP) server.*

The server is created using Java on a stand-alone machine. We use the GlassFish Server Open Source Edition [7] which is the free community-supported application server. It has full Java EE 6 platform support. It is also the next generation modular and extensible architecture (OSGi).

F. *Creating a HTTP client using J2ME to request the server for services.*

The user must have to authenticate him before he could use any of the resource intensive service on his (mobile) device. The J2ME then sends a request to the server which would detect the device user and provide access to the services he wants to.

G. *Some of the services that the server will provide are as follows:*

○ **Database on mobile devices:**

In this, the mobile client can work on a database. The mobile client sees an interface for query to execute. The client user fires a query like create table or a select, update, delete query. This query will be sent to the server which will fire the query on the database server and sending the result of the query back to mobile client. And then the mobile client displays the output.

○ **Image Processor:**

Here, the mobile client is allowed to carry on the various image processing operations such as point processing, neighborhood processing and so. The client may give an input image via the Internet or browse through his phone to select a particular image. The processing is then decided by the client on input and a corresponding request would be sent to the server. The input image is processed and result is sent in the form of response from the server to the client's device. If any problem occurs then the client is notified about the error occurred [17]. The resource limitation of mobile devices causes the problem that the existing image processing software based on the centralized computing mode had difficulty running on mobile devices. A solution is given in the paper by adopting web service-based image processing method. For one thing, image processing tasks were distributed to service providers' service registry and service requesters. For another, what the service providers should do was only to invocate the specific image processing services [18] provided by service providers. Consequently, web service-based solution reduces the resource consumption of mobile devices by redistributing image processing tasks. Compared with traditional methods of image processing, web service-based processing has the advantages of loose coupling and component oriented also can take full advantage of computing resources in heterogeneous network.

H. *Offloading processes.*

The input being processed on a remote location giving a feel to the user that it is carried out on the client device itself is known as offloading processes [13] [15] on a remote device. It is the central part of the project we are working on.

I. *The HTTP transfers file from one server to client and vice versa using a TCP based network, such as Internet.*

Since the services provided by the application are resource intensive, hence they need to be offloaded to a remote server. For offloading, the MIDlet needs to make a connection with the Servlet. The connection can be made by using the Connector class and HttpConnection class in javax.microedition.io package. For the connection the Unique Resource Locator (URL) of the Servlet is required. Once the connection is successfully established, the parameters necessary for processing are sent to the Servlet. The Servlet does all the processing and then sends back the required output. The resources for implementing query analyzer as an application for the mobile device [8].

IV. PROPOSED WORK

In the survey of Resource Intensive Application On Mobile Devices, all research mentioned below is mainly to use Resource Intensive applications [15] on mobile devices. In this application we try to make smart-phones even smarter by allowing them to run applications they are traditionally not meant for. The survey presents you with technique, architecture or framework to accomplish the goal of running resource intensive application on mobile devices.

Our technique is simple. We will have a central server that will be running on a desktop and a mobile client. The mobile client will request the server for services and the server accepts the client's request and processes it. The services that the server offers are resource intensive services that are beyond the capability of the mobile device to execute because of its limitations in hardware and software.

The mobile device only displays an interface for input and output. The mobile device takes the input through its interfaces and gives it to the server. The server then does the processing on the input provided by the mobile device, generates the required output and then sends it back to the mobile device. And all that the mobile device needs to do is display the output.

As we see, the resource intensive part carried out on the remote server and the mobile device only displays the output. Hence the mobile device is not required to do any resource intensive processing rather just displaying it. And this gives the user a feel as if all the processing is done on the mobile device itself, since the user only sees the input and output interfaces and has no details about the actual processing. This abstraction makes the "Resource Intensive Applications" to run on mobile devices. The mobile device communicates with the server via wireless networks.

Block diagram of our proposed system can be shown as:

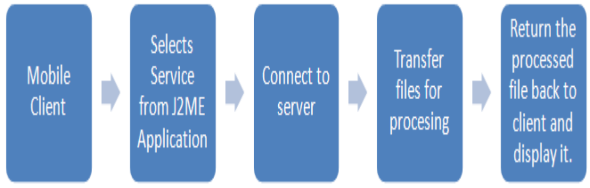
Mobile Client
Selects Service from J2ME Application
Connect to server
Transfer files for procesing
Return the processed file back to client and display it.

Fig.2 Block diagram of proposed system

V. DESIGN

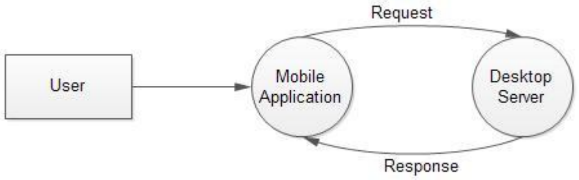

Fig.3 Level 0: DFD Basic Application Structure

In this we show the basic structure of our application. The user interacts with the mobile application which then requests the server for services and the server sends the response back to the client.

Here we explain the steps taking place in the Mobile Application. The user first needs to provide login details. The client is authenticated and then only the client can use the services available. The client must select from the two "Resource Intensive Services" our application provides.

➢ If "Database Query Analyzer" is selected then the user will see a textbox, where he is required to write the query. To fire the query, the query is read and then it is sent to the server. The server does the processing and then returns the output. The returned output is displayed by the mobile application.

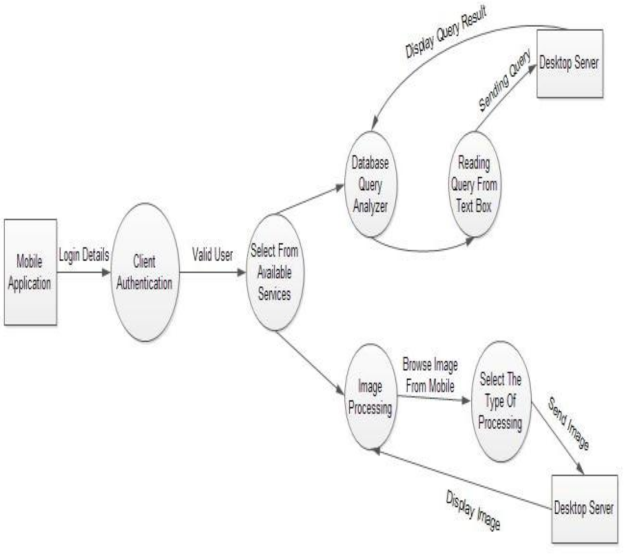

Fig.4 Level 1: DFD Mobile Application

- If "Image Processing" is selected then the user is required to give the path for the source image and also to select the type of processing to be done on the image. Then the image is sent to the server where the selected type of processing is done on the image. The output is sent back and displayed on the mobile.

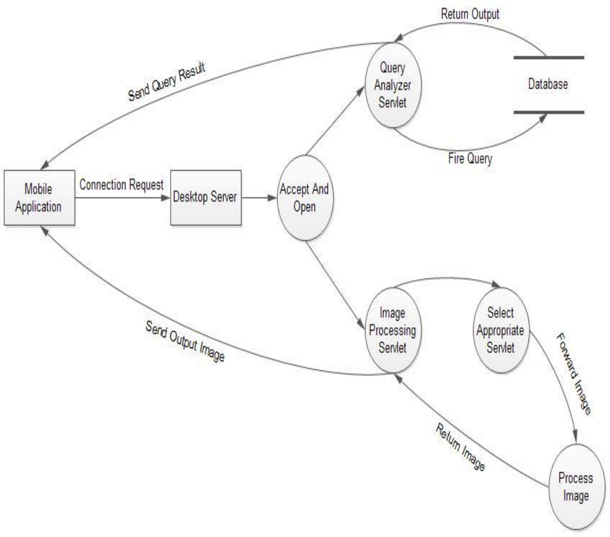

Fig.5 Level 2: DFD Desktop Server

Here the actual processing done by the server is shown. The server keeps on running. On receiving a connection request by the client, the server connects to the client and then receives the file.

- If "Query Analyzer Servlet" is called, then a query in the form of a String is received. This query is fired on a Database Server using JDBC. The output of the fired query is then sent back to the mobile client.

If "Image Processing Servlet" is called, then an image is received and also a String informing which type of processing is to be done on the image. Based on the string value another servlet is called, which performs the required image processing algorithms on the image. The processed image is sent back to the mobile client.

VI. CHALLENGES

- **Support from mobile operating systems:** The widely used mobile OSes include iPhone, Windows Mobile and Android differ widely in support they offer. The support we are interested in includes:
  - The ability to run process in background
  - Support for transfer of data using File Transfer Protocol over a TCP based network.
  - SDK supported by the phone.
- **Mobile specifications:** Many mobile devices have a range of specifications in terms of processing speed, pixels displayed on the screen, minimum size of the image that could be displayed, software support and so on.
- Displaying tabular output: Another challenge that may be faced while displaying the tabular output is to present a large number of records (say 100 or so). One of the solutions may be to fragment the output data but this may not be appealing to the user.

VI. CONCLUSION

The project was primarily a research topic whose objective was to make phones smarter by making "Resource Intensive Applications" to run on mobile devices. Our project allows the mobile devices to run applications they are customarily not meant or built for. The application provides mobility to the user as they are not required to run these applications on the computer. All they need is their mobile phones. We hope that the product developed after extensive research, continuous improvement, rigorous testing & after scrutiny of a large audience, will result in a unique application that will free users in running any application of their choice on the mobile device.

VII. REFERENCES

[1] L. Deboosere, P. Simoens and J. De Wachter B, Grid design for mobile thin client computing: Published in Future Generation Computer Systems Volume 27 Issue 6, June, 2011.

[2] Kenny King, Justin Rutherford and Prem Uppuluri, Executing resource intensive applications on mobile devices, Proceedings of the 48th Annual Southeast Regional Conference.

[3] Albert M. Lai and Jason Nieh, On the performance of wide-area thin-client computing, Published in ACM Transactions on Computer Systems (TOCS) Volume 24 Issue 2, May 2006.

[4] Wikipedia, File Transfer Protocol, http://en.wikipedia.org/wiki/File_Transfer_Protocol. Google. http://code.google.com/p/uHTTPlib/

[5] J. S. J. Daka, A. J. Waters, A high performance broadcast file transfer protocol, Symposium proceedings on Communications architectures and protocols, August 1988.

[6] Google, comparison of FTP and HTTP. http://daniel.haxx.se/docs/ftp-vs-http.html

[7] Wikipedia, GlassFish server. http://en.wikipedia.org/wiki/GlassFish

[8] Prashant Dhingra, Trent Swanson, Microsoft® sql server 2005, compact edition, First edition, July 2007.

[9] D. D. Clark, D. L. Tennenhouse, Architectural considerations for a new generation of protocols, Proceedings of the ACM symposium on Communications architectures & protocols, August 1990.

[10] Bruce Hallberg, Networking: A Beginner's Guide, September 2002.

[11] Kenneth L. Calvert, Michael J. Donahoo, **TCP/IP sockets in Java: practical guide for programmers,** October 2001.

[12] A. Nascimento, J. Rodriguez, S. Mumtaz, A. Gameiro, C. Politis
Dynamic resource allocation architecture for IEEE802.16e: design and performance analysis, August 2008.

[13] Alberto Cruz, Ignacio Marin, Joaquin Entrialgo, Antonio Campos, Offloading Approach for Flexible Provisioning and Execution of Remote Data-gathering Applications, Proceedings of the 2011 Fifth International Conference on Next Generation Mobile Applications, Services and Technologies.

[14] Shih-Wei Liao, Shih-Hao Hung, Chia-Heng Tu, Jen-Hao Chen, Scalable lossless high definition image coding on multicore platforms, Proceedings of the 2007 international conference on Embedded and ubiquitous computing.

[15] Feda Al Shahwan, Klaus Moessner, Francois Carrez, "Distributing Resource Intensive Mobile Web Services", 2011 International Conference on Innovations in Information Technology.

[16] José Rodr'iguez, Dominique Decouchant, Sonia Mendoza, "Java-based Framework for Implementing Soft Real-Time Distributed Applications", 2008 Mexican International Conference on Computer Science.

[17] Jaeyoung Kim, Heesung Jun, "Implementation of Image Processing and Augmented Reality Programs for Smart Mobile Device", 2011 The 6th International Forum on Strategic Technology.

[18] Yi Liang, Yingyuan Xiaoa, Jing Huang, "An Efficient Image Processing Method Based on Web Services for Mobile Devices", IEEE International Conference.