PEER- REVIEWED INTERNATIONAL JOURNAL

Aarhat Multidisciplinary International Education Research Journal (AMIERJ)

ISSN 2278-5655

Impact Factor: 0.948

Bi-Monthly

VOL - II | ISSUES - V | [2013-14]

Chief-Editor:

Ubale Amol Baban

[Editorial/Head Office: 108, Gokuldham Society, Dr.Ambedkar chowk, Near TV Towar,Badlapur, MS

REMOTE DISPLAY SOLUTIONS FOR MOBILE CLOUD COMPUTING

Rajesh S. Yemul

M.E. Computer Network

Student

Smt. Kashibai Navale College Of Engineering

Prof. Ms. Aradhana Deshmukh

M.E.(Computer)

Associate Professor

Smt. Kashibai Navale College Of Engineering

Abstract-

"Remote display solution for mobile cloud computing is an attempt to separate the input/output interface from the application logic for the mobile devices.

Essentially, the principle of mobile cloud computing physically separates the user interface from the application logic. Only a viewer component is executed on the mobile device, operating as a remote display for the applications running on distant servers in the cloud. Any remote display framework is composed of three components: a server side component that intercepts, encodes and transmits the application graphics to the client, a viewer component on the client and a remote display protocol that transfers display updates and user events between both endpoints.

Using standard thin client solutions, such as Remote Desktop Protocol (RDP), and Virtual Network Computing (VNC), in a mobile cloud computing context is not straightforward.

These architectures were originally designed for corporate environments, where users connect over a wired local area network to the central company server executing typical office applications. In this setting, the technical challenges are limited, because delay and jitter are small, bandwidth availability is seldom a limiting factor and office applications exhibit rather static displays when compared with multimedia applications. In a mobile cloud computing environment, the remote display protocol must be able to deliver complex multimedia graphics over wireless links and render these graphics on a resource constrained mobile device.

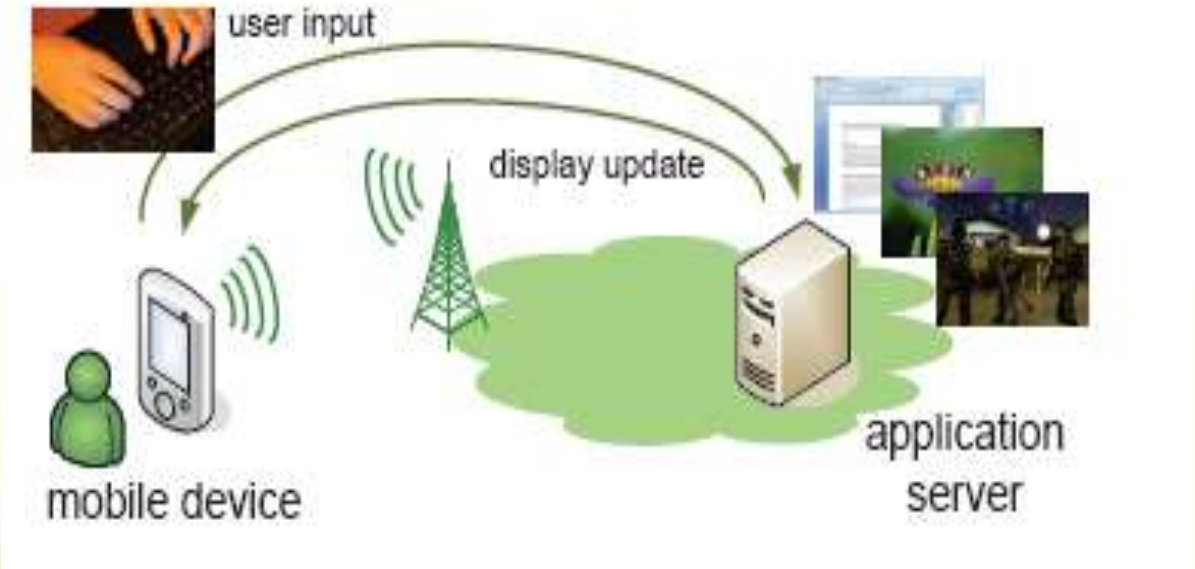
INTRODUCTION

Mobile devices have became an essential part of our daily life as mobile device popularity grows, end-user demands to run heavier applications are equally increasing. Although advances in miniaturization continue, the desire to preserve the advantages of weight, size and device autonomy will always impose intrinsic limits on processing power, storage opacity, battery lifetime and display size. Conventional desktop applications need to be redesigned to operate on mobile hardware platforms, thereby often losing functionality; whereas more demanding applications typically require specific hardware resources that are very unlikely to be available on mobile devices. At the same time, the web hosts increasingly powerful computing resources and has evolved to a ubiquitous computer, offering applications ranging from simple word processors, over all-encompassing enterprise resource planning suites to 3D games. Both Microsoft and Google, have developed complete online office suites, called Office Live and Google Apps respectively, that may evolve to all round alternatives for the mobile office suites. Beyond the conventional office applications, cloud computing broadens the range of applications offered to mobile end-users with demanding applications in terms of graphical hardware, such as 3D virtual environments, or storage capacity, such as 3D medical imaging applications. As the cloud infrastructure is shared among multiple users, these hardware resources can be provided in a cost-effective way. Essentially, the principle of mobile cloud computing physically separates the user interface from the application logic. Only a viewer component is executed on the mobile device, operating as a remote display for the applications running on distant servers in the cloud. Any remote display framework is composed of three components: a server side component that intercepts encodes and transmits the application graphics to the client, a viewer component on the client and a remote display protocol that transfers display updates and user events between both endpoints.

Using standard thin client solutions, such as Remote Desktop Protocol (RDP), and Virtual Network Computing (VNC), in a mobile cloud computing context is not straightforward.

These architectures were originally designed for corporate environments, where users connect over a wired local area network to the central company server executing typical office

applications. In this setting, the technical challenges are limited, because delay and jitter are small, bandwidth availability is seldom a limiting factor and office applications exhibit rather static displays when compared with multimedia applications.



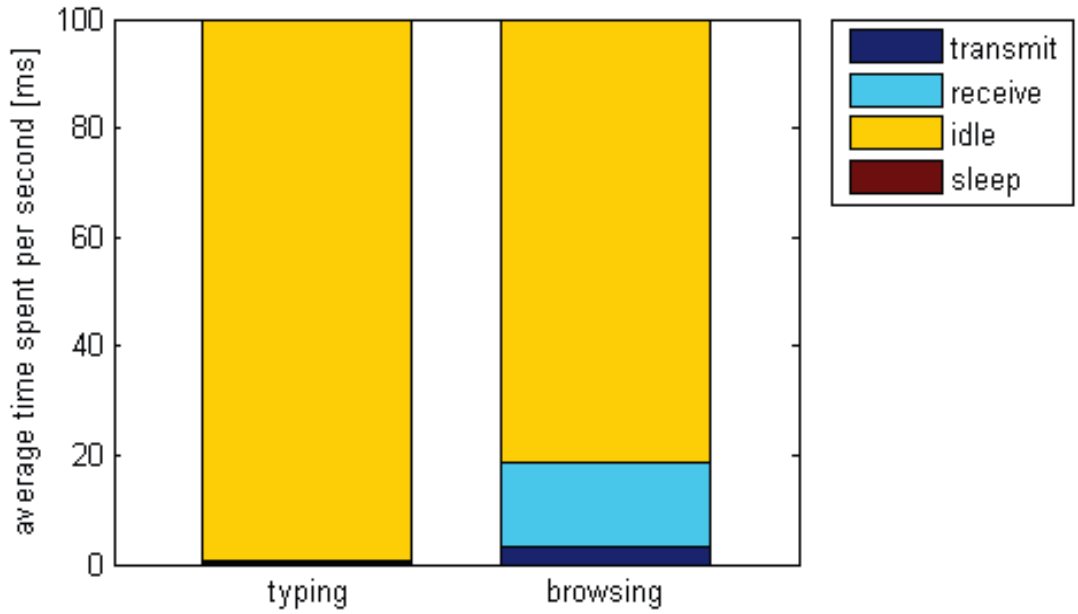
In a mobile cloud computing environment, the remote display protocol must be able to deliver complex multimedia graphics over wireless links and render these graphics on a resource constrained mobile device.

Mobile cloud computing provides a solution to meet the increasing functionality demands of end-users, as all application logic is executed on distant servers and only user interface functionalities reside on the mobile device. The mobile device acts as a remote display, capturing user input and rendering the display updates received from the distant server. Varying wireless channel conditions, short battery lifetime and interaction latency introduce major challenges for the remote display of cloud applications on mobile devices.

LIMITED MOBILE DEVICE BATTERY LIFETIME

Offloading applications to the cloud is a straightforward way to save on energy consumption because the amount of local processing is reduced. Offloading applications from mobile devices is mainly interesting when large amounts of computation are needed in combination with relatively small amounts of network communication. Demanding applications exchange a significant amount of data between client and server because they exhibit a high

degree of interactivity and detailed graphics, The WNIC energy consumption is the product of the number of bytes exchanged over the wireless interface, and the energy cost per byte. Efficient compression techniques to reduce the amount of exchanged data. The average energy cost per byte is determined by the distribution of the time over the four possible WNIC states: send, receive, idle and sleep mode. Because in each state a specific set of components is activated, the WNIC power consumption largely differs between the different states. Figure visualizes our measurements on the average WNIC time distribution in typical remote display scenarios.



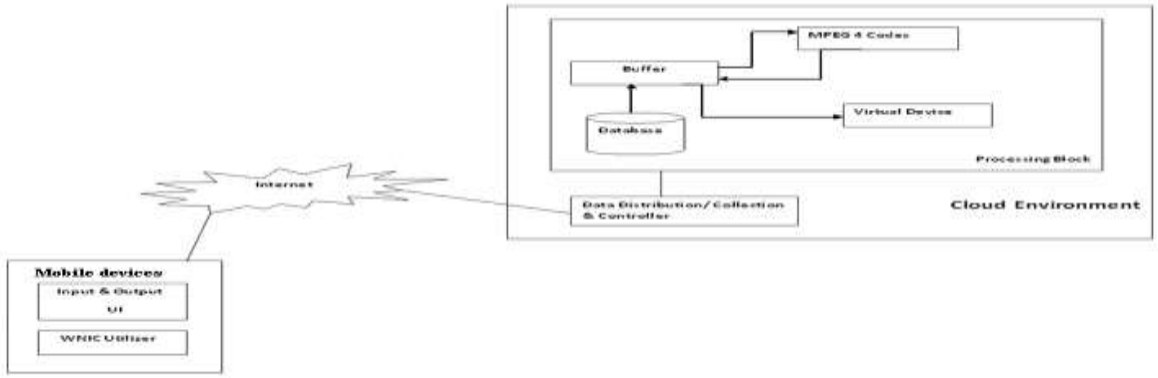
SYSTEM ARCHITECTURE

The system architecture contains 3 blocks

1. Mobile device

2. Communication medium

3. Cloud environment



the first block is mobile device which is nothing but any hand held device through which user wants to access the data stored in cloud environment. There are only two tasks of this block which are providing Input & displaying Output and WNIC card utilization. The first task helps us to overcome the battery problem as we are processing whole data on the cloud itself. So the task of the mobile is just to give the input and take the output. Which will helps us to solve the power consumptions problem. The application is provided to communicate among the internet. Task of the application is nothing but to just provide the User Interface for the communication.

Now next task of the first block is Wireless Network Interface Card (WNIC) utilization. The WNIC energy consumption is the product of the number of bytes exchanged over the wireless interface, and the energy cost per byte. The average energy cost per byte is determined by the distribution of the time over the four possible WNIC states: send, receive, idle and sleep mode. Because in each state a specific set of components is activated, the WNIC power consumption largely differs between the different states.

The next block which is nothing but the internet block that's task is just to provide the medium of communication.

The last block is the Cloud Environment which contains two more sub-blocks in it. The first block is Data Distribution/Collection & Controlling. Tasks of this block are controlling data, distributing data, gathering required data. In this block the data is distributed among the cloud parallel among the cloud for that certain algorithms are used which will simply divide data and

distribute among the available servers. Next task of this block is to gather the required data. Whenever request comes from the mobile device for the particular resources the information is gathered by this block. Here the database is filtered and the data is gathered then that data is forwarded to the next Processing block.

Now the task of the processing block is the important the simplification of the data is important. Separate database is provided to store filtered data gathered by the Data gathering block. Then buffer uses the stored data to provide input to the MPEG4 CODEC algorithm. The task of this algorithm is to separate the high frame rate data and low frame rate data and the compress them this will solve our bandwidth problem. Then the processed data is again forwarded to the buffer to store it temporarily before sending. Then that data is executed on the virtual device which is nothing but the Android simulator if the given data is executed successfully on the virtual device then it is clear to send.

INTERACTION LATENCY

Interaction latency, i.e. the delay a user experiences between generating some user input and having the result presented on his display, is key challenge of mobile cloud computing. Whereas bandwidth limitations are likely to disappear with technological advancements, interaction latency is an intrinsic key challenge of mobile cloud computing because even the most trivial user operations need to be communicated to the server.

Solutions to mitigate the interaction latency either target a reduction of the number of hops on the end-to-end path by moving the application closer to the client, or better synchronization mechanisms between client and server. Satyanarayanan et al. [9] introduce the concept of *cloudlets*: trusted, resource-rich computers that are dispersed over the Internet. Exploiting virtual machine technology, mobile devices rapidly deploy their services on the most nearby cloudlet by uploading an overlay virtual machine to customize one of the generic base virtual machines that are commonly available on all cloudlets. The physical proximity ensures low-latency, one hop, high-bandwidth wireless LAN access, e.g. over the latest WiFi 802.11n technology, instead of mobile radio technology access, such as HSDPA or LTE. In these cases, latency optimization strategies need to focus on a reduction of the number of roundtrip times that is required to resynchronize the client device display with the server. Given the current

application state, the application server can predict potential display updates and stream these in advance to the client. Contrary to video streaming, where the frame order is known in advance, in mobile cloud computing the next display update depends on user input. For example, when a user opens an application menu, the server could pre compute all dialog windows that can be opened by selecting one of the menu items. have applied this pre computing technique to the use case of virtual 3D environments. Given the current user position, the possible next user viewpoints are calculated in advance and provided to the client. When the user actually moves forward, the client fetches the correct viewpoint from its cache.

CONCLUSION

By physically separating the user interface from the application logic, the principle of mobile cloud computing allows to access even the most demanding applications in the cloud from intrinsically resource-constrained mobile devices.

REFERENCES

- Remote Display Solutions for Mobile Cloud Computing - Pieter Simoens, Filip De Turck *member, IEEE,* Bart Dhoedt *member, IEEE,* Piet Demeester *Fellow, IEEE, Feb 2011.*
- K.-J. Tan, J.-W. Gong, B.-T. Wu, D.-C. Chang, H.-Y. Li, Y.-M. Hsiao, Y.-C. Chen, S.-W. Lo, Y.-S. Chu, and J.-I. Guo, "A remote thin client system for real time multimedia streaming over VNC," in *2010 IEEE International Conference on Multimedia and Expo (ICME)*, 2010 2010, pp. 992–7.
- F. Lamberti and A. Sanna, "A streaming-based solution for remote visualization of 3D graphics on mobile devices," *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, vol. 13, no. 2, pp.* 247–260, MAR-APR 2007.
- H. Kawashima, K. Koshiba, K. Tuchimochi, K. Futamura, M. Enomoto, and M. Watanab , "Virtual PC-type thin client system," *NEC TECHNICAL JOURNAL, vol. 2, no. 3, pp. 42–47, SEP 2007.*