IMPACT: International Journal of Research in
Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 2, Feb 2014, 167-174
© Impact Journals

OPTIMIZED CONSUMPTION AND ACCESS OF REMOTE DISPLAY ON MOBILE DEVICE ENVIRONMENT

J KUMARAN[1], PAWAR AJIT TANAJI[2] & ALLAMPATI RAKESH[3]

[1]Assistant Professor, Pondicherry Engineering College, Puducherry, India

[2,3]Research Scholar, Department of CSE, Pondicherry Engineering College, Puducherry, India

ABSTRACT

Mobile cloud computing has been introduced to be a potential technology for mobile services, Together with an explosive growth of the mobile applications and emerging of cloud computing concept. End-user demands to run heavier applications are equally increasing, as mobile device popularity grows. Mobile cloud computing integrates the cloud computing into the mobile environment and overcomes obstacles related to the performance environment and security discussed in mobile computing. MCC integrates the cloud computing into the mobile environment and overcomes obstacles related to the performance, environment in mobile computing. This paper gives an idea of MCC, which helps general readers have an overview of the MCC including the architecture, applications and solutions. In this paper, a number of adequate solutions that have been proposed to tackle the main issues such device battery lifetime, Bandwidth utilization associated with the consumption and access of remote display on mobile devices.

KEYWORDS: Mobile Cloud Computing, Mobile Devices, Device Battery Lifetime, Bandwidth, Interaction Latency

INTRODUCTION

Mobile cloud computing (MCC) is a combination between mobile network and cloud computing, thereby providing optimal services for mobile users. In mobile cloud computing, mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) since all the data and complicated computing modules can be processed in the clouds. This section provides an overview of MCC including definition, architecture, and advantages of MCC.

Mobile devices, ranging from smart phones to tablets, have recently become so pervasive that they are increasingly replacing personal computers in everyday activities related to both entertainment and work. The rapid progress of mobile computing becomes a powerful trend in the development of IT technology as well as commerce and industry fields.

The mobile devices are facing many challenges in their resources (e.g., battery life, storage, and bandwidth) and communications (e.g., mobility and security). The limited resources significantly impede the improvement of service qualities. “Mobile Cloud Computing at its simplest refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing to not just smart phone users but a much broader range of mobile subscribers”.

MCC as a new paradigm for mobile applications whereby the data processing and storage are moved from the mobile device to powerful and centralized computing platforms located in clouds. These centralized applications are then accessed over the wireless connection based on a thin native client or web browser on the mobile devices.

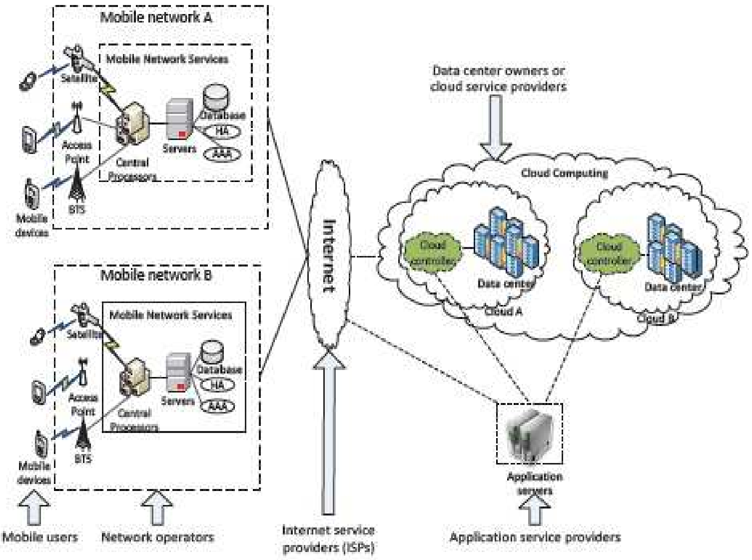

Figure 1: Mobile Cloud Computing (MCC) Architecture

Briefly, MCC provides mobile users with the data processing and storage services in clouds. The mobile devices do not need a powerful configuration (e.g., CPU speed and memory capacity) since all the complicated computing modules can be processed in the clouds. It has been attracting the attentions of entrepreneurs as a profitable business option that reduces the development and running cost of mobile applications, of mobile users as a new technology to achieve rich experience of a variety of mobile services at low cost, and of researchers as a promising solution for green IT. CC offers some advantages by allowing users to use infrastructure, platforms and software.

Existing Approaches/Solutions

- Mobile phones preserve the advantages of weight, size and device independence but will always impose basic limits on processing power, storage capacity, battery lifetime and display size.
- Conventional desktop applications are redesigned to operate on mobile hardware platforms, thereby often losing functionality.
- Demanding applications typically require specific hardware resources that are not available on mobile devices.
- To get the display users connect over a wired local area network to the central company server executing typical office applications.

Mobile Cloud Computing Challenges

Essentially, mobile cloud computing physically separates the user interface from the application logic. The mobile device executes only a viewer component, operating as a remote display for the applications running on distant servers in the cloud. As Figure shows, a remote display framework has 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.

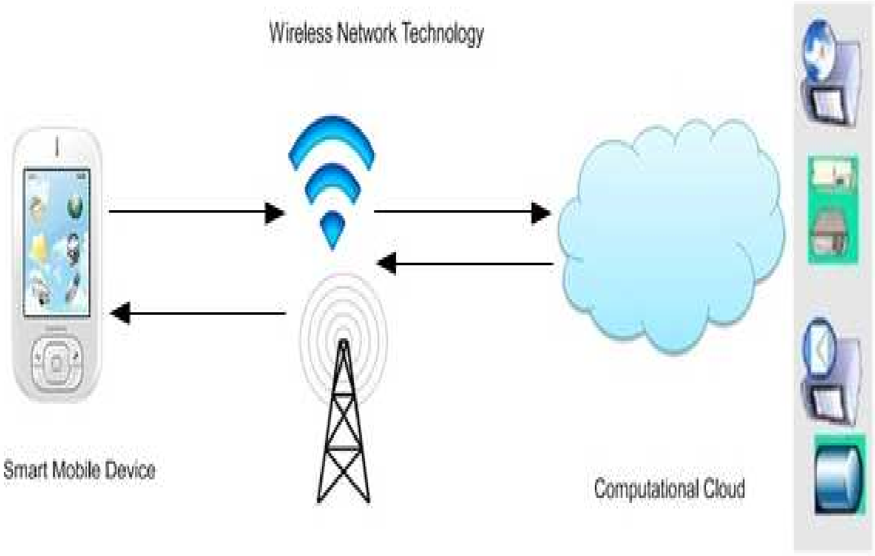

Figure 2: Model of Mobile Cloud Computing

A potential obstacle to mobile cloud computing success is the encumbered I/O functionality of mobile devices. Slide out keyboards and stylus devices facilitate user input and maximize display sizes without increasing overall device size, but they do not provide an adequate solution for convenient I/O. Novel user interfaces for mobile devices are, however, entering the market. Media tablets, such as Apple's I Pad, offer a larger screen with touch functionality and a keyboard close in size to regular keyboards. Other manufacturers, such as NEC, minimize their device size and support external keyboards and displays4 to augment I/O functionality.

More fundamental challenges for mobile cloud computing lie in the short battery lifetime of mobile devices, the limited and varying bandwidth on wireless links, and the interaction latency between user input and display updates. Table 1 presents an overview of several solutions proposed to address these challenges.

Proposed Solutions for Mobile Challenges

- The principle of mobile cloud computing physically separates the user interface from the application logic.
- Here, a Viewer component is executed on the mobile device, which is operating as a remote display for the applications running on distant servers in the cloud.
- 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.
- In a mobile cloud computing environment, the remote display protocol deliver complex multimedia graphics over wireless links and render these graphics on a resource constrained mobile device. Offloading save on energy consumption because the amount of local processing is reduced.
- Efficient compression techniques to reduce the amount of exchanged data are done using compression techniques and versatile graphics encoding, downstream data peak reduction and Optimization of upstream packetization overhead.

Table 1 presents an overview of several solutions proposed to address these challenges.

Table 1: Proposed Solutions to Mobile Cloud Computing Challenges

Challenge	Solutions
Device battery lifetime	Identification of wireless network interface card (WNIC) sleep intervals
Wireless bandwidth availability	Versatile graphics encoding Downstream data peak reduction Optimization of upstream packetization overhead
Interaction latency	Computing display updates in advance Image buffering for virtual environment streaming Scene object caching

Identification of WNIC Sleep Intervals

It is important to study WNIC energy consumption to develop strategies that optimize the energy balance, which 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 time over the four possible WNIC states: send, receive, idle, and sleep. Because a specific set of WNIC components are activated in each state, power consumption varies widely between the states.

Major additional energy savings are to be expected from approaches that put the WNIC in the energy conserving sleep mode during intervals without communication. Only a single network connection is required to communicate user events and display updates over a thin client protocol. This present a cross-layer algorithm that analyzes the thin client protocol information and puts the WNIC in sleep mode when no traffic from the server is expected.

A proposed cross-layer power-saving approach operates between the MAC layer and the remote display protocol layer. In this, the appropriate sleep intervals must therefore be determined at the remote display protocol layer, where the display update schedule is established—for example, via a push approach in which the server sends display updates with fixed intervals or a pull approach in which the client sends an explicit request. Correlating the transmission of user input to the network round-trip time predicts the arrival of the next display update. In between two display updates, the WNIC enters sleep mode. This sleep mode is interrupted at regular intervals to transmit user events.

Researchers have used cross-layer optimization to reduce WNIC energy consumption by up to 52 percent.

Wireless Bandwidth Availability

Compared to fixed access networks, modern broadband mobile and wireless technologies offer limited and variable bandwidth availability. Universal Mobile Telecommunications System (UMTS) users typically receive up to 384 kilobits per second, while Krishna Balachandran and colleagues9 reported practical throughputs of 347 Kbps for Long Term Evolution (LTE) and up to 6.1 Mbps for Wi MAX. Actual throughput depends on user mobility, interference, and fading effects.

Versatile Graphics Encoding

To design a windowed display encoder using H.264 video codec to compress screen updates, which will guarantee high fidelity desktop playback in a bandwidth constrained environment. In this scenario, encoder supports high performance interaction for applications with adequate window size.

An encoding method using H.264 video codec is introduced for high performance in compression to address the high latency and low bandwidth limitation of mobile Internet. The H.264 video codec is specially optimized for compression of motion pictures and multimedia content hence reducing network traffic greatly.

The H.264 Decoder would decode the display stream and update the display. Server takes control messages and provide service accordingly. H.264 is currently one of the best codec’s in terms of compression ratio. The goal of the system is to utilize H.264 video encoder in screen compression to optimize system performance in high motion scenarios.

In consideration of interactivity performance, client-driven update mode and the inter-prediction feature of H.264, we designed a H.264 screen encoder as such: once the server accepts a screen update request from the client, it will capture the screen and encode it as a frame in the H.264 video sequence, In the design, every frame is captured when the server accepts a request and the system latency depends on encoding time, network speed and latency. Cache is not added here because the encoder is slower than the decoder hence the time gap between two frames is sufficient for the decoder to process. Besides, encoder speed is less than real time (defined as □25 fps) thus introduces extra system latency.

The H.264 decoder is responsible for decoding display streaming data and updating it to client device’s frame buffer to perform display.

The mobile device regularly reports on the amount of data that the controller encodes per unit of time, a metric reflecting both the device hardware capabilities and the amount of data the device receives. By adjusting the resolution and image quality accordingly, the controller maintains a target frame rate that ensures a smooth visualization experience.

Downstream Data Peak Reduction

This paper revealed that repainting the graphical objects after recurring user actions causes considerable redundancy in remote display protocol traffic. Their proposed hybrid cache-compression scheme uses the cached data as history to better compress recurrent screen updates, with the cache containing various drawing orders and bitmaps. Using RDP, they reduced the number of data spikes by 27 to 42 percent, depending on cache size, resulting in global network traffic reductions of 10 to 21 percent.

Optimization of Upstream Packetization Overhead

Buffering user events at the client for a short period enables the joint transmission of multiple user events. The maximum buffering period is based on a tradeoff of remote display bandwidth reduction and interaction latency. A closed-loop controller running at the client integrates the interaction latency models in terms of this buffering period and the network round-trip time, ensuring that the average interaction latency does not exceed a predefined maximum value by continuously monitoring the current network status and adjusting the buffering period accordingly.

The highest bandwidth reductions are achieved for interactive applications with frequent user events and lower round-trip times. For a text-editing scenario and network round-trip times less than 50 ms, researchers have achieved bandwidth reductions up to 78 percent.

Interaction Latency

While technological advances are likely to overcome bandwidth limitations, interaction latency—the delay users experience between generating some input and seeing the result on their display—is an intrinsic challenge of mobile cloud computing because the device must communicate even the most trivial user operations to the server. Solutions to mitigate interaction latency try to either reduce the number of hops on the end-to-end path by moving the application closer to the client or provide better synchronization mechanisms between client and server.

Computing Display Updates in Advance

Given the current application state, the application server can predict potential display updates and stream them in advance to the client. Contrary to video streaming, in which 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 can pre-compute all dialog windows that can be opened by selecting one of the menu items.

As this approach applied to virtual 3D environments. Given the current user position, the system calculates the possible next user viewpoints in advance and provides them to the client. When the user actually moves forward, the client fetches the correct viewpoint from its cache.

Image Buffering for Virtual Environment Streaming

Due to limitations in mobile bandwidth and mobile device memory resources, in most cases, streaming all possible next display updates in advance is unfeasible. Furthermore, the gains of this pre-computing technique are highly dependent on prediction accuracy. A better strategy might be to buffer some key display updates, for which the server only needs to provide a differential update.

It is evaluated that the several cache management strategies and reduced the amount of requests during a 300-step movement in a 3D virtual environment from 300 to 145. Of course, in this case, the server response is still required to update the display.

Scene Object Caching

For more static applications, such as office applications, the potential next updates can be more accurately predicted as, for example, a menu layout will rarely change. Consequently, the number of corrective server updates will be more limited. A typical example would be the list of recently opened files in a text editor’s File menu. Scene description languages such as MPEG-4 BiFS are particularly suited to support this client-side handling of user input 13. The client not only receives graphic updates, but also is informed about the structure of the displayed scene and its composing objects, as well as how the user can manipulate these objects.

CONCLUSIONS

Mobile cloud computing is one of mobile technology trends in the future since it combines the advantages of both mobile computing and cloud computing, thereby providing optimal services for mobile users. This article has provided an overview of mobile cloud computing in which its definitions, architecture, and optimized solutions have been presented. The principle of mobile cloud computing allows to access even the most demanding applications in the cloud from intrinsically resource-constrained mobile devices by physically separating the user interface from the application logic. In this paper, we have surveyed contemporary remote display optimization techniques specifically tailored to the short mobile device battery lifetime, the varying and limited bandwidth availability on wireless links and the interaction latency. The context of mobile cloud computing is highly dynamic, owing to the user mobility, the wide diversity of applications, and the varying wireless channel status. Future work should therefore be devoted to the design of an overall framework, integrating all the presented solutions, and activating the most appropriate solutions dependent on the mobile device, network and cloud server status.

REFERENCES

1. C. Albanesius, “Smartphone Shipments Surpass PC Shipments for First Time. What’s Next?” PC Magazine, 8 Feb. 2011;
2. V.S. Pendyala and S.S.Y. Shim, “The Web as the Ubiquitous Computer,” Computer, Sept. 2009, pp. 90-92.
3. F. Lamberti and A. Sanna, “A Streaming- Based Solution for Remote Visualization of 3D Graphics on Mobile Devices,” IEEE Trans. Visualization and Computer Graphics, Mar./Apr. 2007, pp. 247-260.
4. H. Kawashima et al., “Virtual PC-Type Thin Client System,” NEC Technical J., Sept. 2007, pp. 42-47.
5. K. Pentikousis, “In Search of Energy-Efficient Mobile Networking,” IEEE Comm. Magazine, Jan. 2010, pp. 95-103.
6. K. Kumar and Y.-H. Lu, “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?” Computer, Apr. 2010, pp. 51-56.
7. T. Kim et al., “An Energy-Aware Transmission Mechanism for WiFi-Based Mobile Devices Handling Upload TCP Traffic,” Int’l J. Comm. Systems, May 2009, pp. 625-640.
8. P. Simoens et al., “Cross-Layer Optimization of Radio Sleep Intervals to Increase Thin Client Energy Efficiency,” IEEE Comm. Letters, Dec. 2010, pp. 1095-1097.
9. K. Balachandran et al., “Performance Assessment of Next-Generation Wireless Mobile Systems,” Bell Labs Technical J., Feb. 2009, pp. 35-58.
10. http://onlinelibrary.wiley.com/doi/10.1002/wcm.1203/abstract.
11. I. Nave et al., “Games@Large Graphics Streaming Architecture,” Proc. IEEE Int’l Symp. Consumer Electronics (ICSE 08), IEEE Press, 2008, pp. 205-208.
12. K.-J. Tan et al., “A Remote Thin Client System for Real-Time Multimedia Streaming over VNC,” Proc. IEEE Int’l Conf. Multimedia and *Expo* (ICME 10), IEEE Press, 2010, pp. 992-997.