Optimized solutions for mobile Cloud Computing

[1]Mr.J.Praveen Kumar,[2] Rajesh Badam
[1]Asst.Prof.CSE Dept.
[2]Mtech.(CSE Dept.)
Malla Reddy College of Engineering and Technology

Abstract: As mobile device popularity grows, end-user demands to run heavier applications are equally increasing. 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. This paper gives an idea of MCC, which helps general readers have an overview of the MCC including the architecture, applications and solutions. The issues, existing solutions and approaches are presented. In addition, the future research directions of MCC are discussed

Keywords- Mobile cloud computing, offloading, Mobile services.

Introduction

Mobile devices are becoming an important part of human life as the most effective and convenient communication tools not bounded by time and place. 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 mobilecomputing to not just smartphone users but a much broader range of mobile subscribers”.

The term “mobile cloud computing” was introduced not long after the concept of “cloud computing” launched in mid-2007. 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 . This section provides an overview of MCC including definition, architecture, and advantages of MCC.

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.
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.

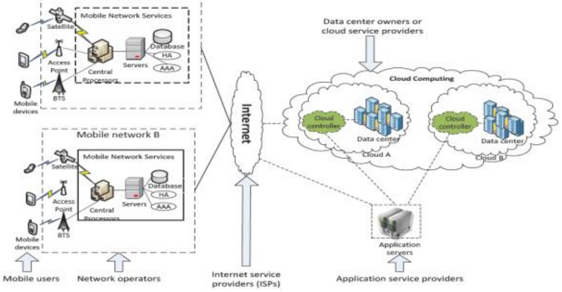
Proposed solutions for mobile challenges:

- 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 applications to the cloud is a straight forward way to

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.

Mobile cloud computing architecture



Cross-layer identification of WNIC sleep intervals

To develop strategies that optimize the energy balance,it is important to study WNIC energy consumption, 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.

Although the send and receive modes consume the most power, energy-saving approaches should focus on the large idle times observed in remote display scenarios. These idle times are a consequence of the limited frequency of user interactions imposed by the network round-trip time. After some interaction, users must wait until the results become visible on the screen before continuing their work. furthermore, interactive applications will only update their display when instructed by the user—for example, by entering a URL or clicking on a hyperlink.

A proposed cross-layer power-saving approach operates between the MAC layer and the remote display protocol layer (see Figure 2).8 Because the MAC layer operates on binary data and cannot discriminate between, for example, transmitted user input and Transmission Control Protocol (TCP) acknowledgments, it is unaware of the arrival of the next display update. 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.8

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 WiMAX. Actual throughput depends on user mobility, interference, and fading effects.

Versatile graphics encoding

The choice of codec to compress the intercepted application graphics at the server is a tradeoff between visual quality, compression efficiency, and decoding complexity.

Conventional remote display architectures including RDP, ICA, and VNC typically virtualize a layer of the graphic rendering stack at the server and forward intercepted drawing primitives to the client, such as instructions to draw a rectangle, display a bitmap, or put some text on the screen.

This approach is optimal for applications, such as typical office applications, that only update small regions of the display or have a slow refresh rate with respect to the network round-trip time. Bandwidth requirements to remotely display this type of graphics do not exceed 200 Kbps and can be adequately served over wireless links.

On the other hand, encoding multimedia graphics applications would require numerous drawing primitives because they update large parts of the screen at high refresh rates and they often contain fine-grained and complex color patterns. This kind of graphics can be more efficiently encoded using a video codec, such as H.264 or MPEG-4. Using video codecs for remote display purposes is referred to as interactive live streaming because the graphics are mainly the result of user interaction, in contrast to regular video streaming, which requires only limited user interaction—for example, to start and stop the video. Even when only a single application is used, the characteristics of the graphics on the user display might significantly differ when a user is accessing mobile cloud computing services. For example, a user browsing a Wikipedia page might click on a link that opens a YouTube video in the same browser window. Remote display frameworks must therefore be able to switch seamlessly between multiple encoding modes based on an analysis of graphics at the server.

Downstream data peak reduction

Interactive applications only update their display when instructed by the user. These display updates usually involve sending a large amount of data to the client in a short interval, which requires an instantaneous bandwidth much higher than the average bandwidth requirement. Furthermore, this bursty traffic pattern is unfavorable in wireless network environments, as it might induce additional collisions on the wireless channel.

Optimization of upstream Packetization overhead

User events are the principal source of upstream remote display traffic from client to server. Individually, each user event embodies only a small amount of information: a key or button ID, one bit to discriminate between the press and release actions, and possibly the current pointer coordinates. Nevertheless, user events induce important upstream traffic because they are often generated shortly after each other. Entering a single character results in two user events to indicate the press and release actions, whereas moving the mouse results in a sequence of pointer position updates.

Usually, the system transmits user events as they occur to minimize interaction latency. Because data packets sent upstream often contain a single user event, headers added at the TCP, IP, and wireless link layer cause large Packetization overhead. Table 3 quantifies the Packetization overhead of TCP/IP headers of three commonly used remote display protocols—the VNC Remote Framebuffer (RFB) Protocol, RDP, and ICA—when sending a single keystroke to the server. Optional headers and the wireless link layer header further increase total overhead.

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 precompute all dialog windows that can be opened by selecting one of the menu items.

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 precomputing 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.

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.

By physically separating the user interface from the application logic, mobile cloud computing allows access to even the most demanding applications from intrinsically resource-constrained mobile devices. Developers tailor contemporary remote display optimization techniques to mobile devices' short battery lifetime, the varying and limited bandwidth availability on wireless links, and interaction latency. Although each of these solutions adequately addresses specific mobile cloud computing challenges, an overall approach is currently lacking.

Because of user mobility, the wide diversity of applications, and the varying wireless channel status, the mobile cloud computing context is highly dynamic. Future research should therefore focus on the design of a comprehensive framework that integrates the existing solutions and activates the most appropriate one depending on the current device, network, and cloud server status.

Conclusion

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 applications supported by mobile cloud computing including mobile commerce, mobile learning, and mobile healthcare have been discussed which clearly show the applicability of the mobile cloud computing to a wide range of mobile services. Then, the issues and related approaches for mobile cloud computing have been discussed. Finally, the future research directions have been outlined.

References

[1.] C. Albanesius, “Smartphone Shipments Surpass PC Shipments for First Time. What’s Next?” *PC Magazine*, 8 Feb. 2011; www.pcmag.com/article2/0,2817,2379665,00.asp.

[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.

Author’s profile

Rajesh Badam: Pursuing Mtech. (CSE Dept.) Malla Reddy College of Engineering and Technology, Hyderabad.

Mr. J.Praveen Kumar

Asst.Prof. CSE Dept.
Malla Reddy College of Engineering and Technology, Hyderabad.