Reduction of execution time and energy consumption using ternary decision for multiple off loading targets in handheld device

Dr. S. Meenakshi Sundaram[1], Tejaswini R M[2]

[1] *Dr. S. Meenakshi Sundaram, Professor & Head, Dept. of CSE, GSSS Institute of Engineering & Technology, Mysuru, India.*

[2] *Tejaswini R M, Assistant Professor, Dept. of CSE, GSSS Institute of Engineering & Technology, Mysuru, India.*

---***---

Abstract –

Performance and time management of battery operated devices such as laptops and Personal Data Assistants (PDAs), with energy conservation are the key factors considered in this paper. Energy management in handheld devices has become an important task with the growing number of accelerators, increasing memory demands and high computing capacities. Off loading computational workload to cloud platform provides a better solution to improve the performance and battery life of smart phones. This also consumes energy and time for uploading and retrieval of data from the cloud. In this paper we have developed a framework to minimize the response time and energy consumption. This method has less false offloading decision rate. The proposed method aims to reduce execution time and energy consumption using ternary decision for multiple off loading targets.

***Key Words*: Personal Data Assistants(PDAs),Smart Phones, Off loading framework, energy conservation**.

1. INTRODUCTION

The capabilities of handheld devices have increased and are capable of replacing laptops which have replaced desktop computers in many roles. Handheld devices have evolved from PDAs, Organizers, Cellular phones, and game machines into a variety of new forms. The increased functionality decreases the energy consumption of these devices. Energy saving methodologies for battery does not cope up with the energy requirements of handheld devices.

Among today's rechargeable batteries, lithium-ion cells offer the highest storage capacity. Introduced commercially by Sony in 1991, the capacity has seen an improvement of about 10 percent per year in the recent years [15]. There is a constant increase in storage capacities with the use of alternative materials and novel cell structures in batteries. Still major improvements in rechargeable batteries for better storage needs to be considered in the near future.

The disparity between energy requirements and energy sources in handheld devices is a result of several factors. First, users expect more functionality from their devices (more performance and memory, better displays, wireless capabilities, etc.). With the improvements in low-power electronics, there is an increased functionality which corresponds to increase in energy consumption. Second, as a consequence of the increased functionality the handheld devices consume power. Finally, battery technology has not improved at the same pace as the increase in energy requirements.

Battery suppliers and mobile system designers expect the eventual emergence of fuel-cell power sources and are investing accordingly. In its most basic form, a fuel cell combines hydrogen fuel with oxygen from the surrounding air to produce water and electricity. But hydrogen combusts easily and has low density, making it unattractive for most mobile applications. Instead, most efforts are directed towards using methanol as fuel and extracting the hydrogen with catalysts or high-temperature reforming [16].

The research community which includes the hardware manufacturers and OS designers have found positive solutions to extend the battery life of mobile handsets at different levels such as hardware, operating system, wireless technologies and applications. However, these efforts are limited by the heavy layering existing on Smartphone platforms that makes difficult exploiting cross layer optimizations, which might be fairly straightforward otherwise. One of the main reasons behind this limitation is a complex business ecosystem in which multiple players, cellular network providers, content providers, cloud service providers, hardware manufacturers and OS vendors compete to retain their share of the mobile business.

2. RELATED WORKS

Survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

[1] Pervasive computing allows a user to access an application on heterogeneous devices continuously and consistently. However, it is challenging to deliver complex applications on resource-constrained mobile devices such as cell phones. Application-based or system-based adaptations have been proposed to address the problem, but they often require application fidelity to be significantly degraded. This problem can be overcome by dynamically partitioning the application, and by offloading part of the application execution with data to a powerful nearby surrogate. This allows the application to be delivered in a pervasive computing environment without significant fidelity degradation or expensive application rewriting. Runtime offloading needs to adapt to different application execution patterns and resource fluctuations in the pervasive computing environment.

[2] Java-enabled wireless devices are preferred for various reasons. For example, users can dynamically download Java applications on demand. The dynamic download capability supports extensibility of the mobile client features and centralizes application maintenance at the server. Also, it enables service providers to customize features for the clients. In this work, we extend this client-server collaboration further by offloading some of the computations (i.e., method execution and dynamic compilation) normally performed by the mobile client to the resource-rich server in order to conserve energy consumed by the client in a wireless Java environment. In the proposed framework, the object serialization feature of Java is used to allow offloading of both method execution and byte code to native code compilation to the server when executing a Java application. Our framework takes into account communication, computation, and compilation energies to decide where to compile and execute a method (locally or remotely), and how to execute it. As both computation and communication energies vary based on external conditions, our decision must be done dynamically when a method is invoked.

[3] A framework is presented for making computation offloading decisions in computational grid settings in which schedulers determine when to move parts of a computation to more capable resources to improve performance. Such schedulers must predict when an offloaded computation will outperform one that is local by forecasting the local cost and remote cost. Typically, this decision amounts to predicting the bandwidth between the local and remote systems to estimate these costs. Our framework unifies such decision models by formulating the problem as a statistical decision problem that can either be treated "classically" or using a Bayesian approach. Using an implementation of this framework, we evaluate the efficacy of a number of different decision strategies.

[4] Extracting Mobile devices such as Personal Digital Assistants (PDAs) have gained popularity with their growing computation capability, flash memory, and various functions. Since most of these devices are equipped with cameras, organization and retrieval of images are important. Content-Based Image Retrieval (CBIR) provides a method to search images based on their contents and is a promising application for these devices. Several studies are dedicated to mobile CBIR. Most of them treat the mobile device as a “thin client”, and perform the actual search and feature database maintenance on the server side. Previous work performs CBIR entirely on the mobile device. This extends CBIR to the cases where network connectivity is limited. However, in other cases, partitioning computation between a mobile device and a grid powered server can save energy on the device. A dynamic decision should be made to decide when and what to offload to the server to save energy.

[5] Advances in computing hardware, communications technologies, and novel multimedia applications are spurring the development of smart phones and personal digital assistants. There is world-wide accessibility to 2G, 3G, and Wi-Fi networks. Some examples of well-known wireless handheld devices are BlackBerry, iPhone, iPad, iPod, and Kindle. On the one hand, small size and light weight are their attractive features for high mobility and accessibility. On the other hand, the same features impose significant constraints on their processing, memory, and energy storage capabilities, thereby limiting the device’s general functionalities and availability. User expectations in terms of performance from handheld devices are ever increasing. In addition to performance expectations, the requirement of portability imposes severe constraints on size and weight of a handheld system. Consequently, batteries too are small and light, and, therefore, the system energy budget is severely limited. The amount of energy in a fully charged battery is one of the important resources of a handheld system, and battery lifetime is an important characteristic. Unfortunately, improvements in energy density of batteries

have not kept pace with the advancements in microelectronics technology.

3. PROPOSED METHODOLOGY

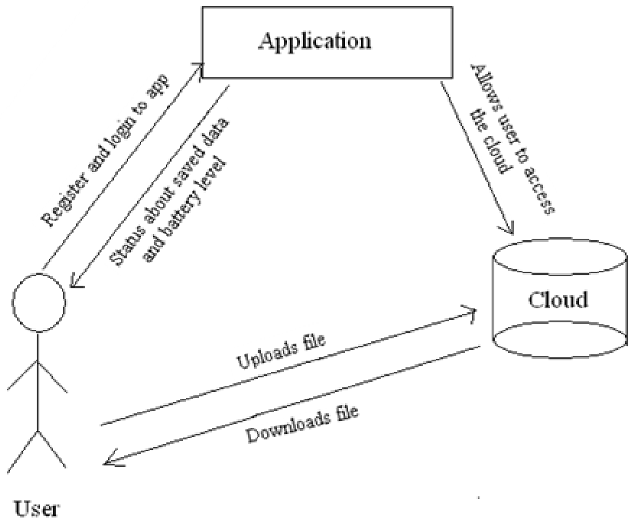

Fig -1: System Architecture

The users register with the application where each user logs in with a specific User-ID. The application allows the users to access the cloud. The users can upload and download the files from the cloud.

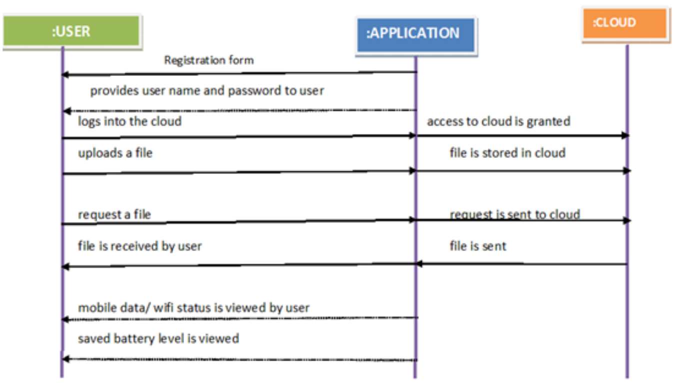

Fig -2: Sequence Diagram

Figure 2 shows a sequence diagram corresponding to the user and the cloud. Each actor, as well as the system is represented by a vertical line called lifeline and each message by a horizontal arrow from the sender to the receiver.

First, the application sends the registration form to the user where he has to enter his username and password then the user logs in to the application which in turn grants access to the cloud. The user can then upload or download files from the cloud. If the files are uploaded then it is stored in the cloud. If the file needs to be downloaded from the cloud, then the user sends the request to the application which in turn sends the request to the cloud. If the requested file exists in the cloud, it sends the requested file which is then received by the user. The user can then view the saved battery status and mobile data or Wi-Fi.

A) Upload Module

Energy and time consumption is more for data and program uploads to the cloud. In this paper, we have developed an off loading framework, which aims to shorten response time and reduce energy consumption at the same time.

1. Time and energy aware
2. Register page
3. View files
4. Current directory
5. Upload screen
6. View connection details

Login Procedure

This procedure describes the login method adopted in respect of logging into an application which is given in detail below:

Step1: The first page of the application shows the time and energy aware page in which we can login to any existing account or register to a new account.

Step2: Login using the username and password.

Step 3: The list of files is displayed which are already in the cloud.

Step 4: Go to menu and click on upload option.

Step 5: The files to be uploaded are displayed.

Step 6: Upload screen appears.

Step 7: As soon as the upload is completed the data pack or Wi-Fi which is on turns off automatically and the present battery level is displayed on the screen.

B) Download Module

This consumes both time and energy to download data or programs from the cloud and retrieve the results from the cloud. In this paper, we develop an offloading framework, which aims to shorten response time and reduce energy consumption at the same time.

1. Time and energy aware
2. Register page
3. View files
4. download
5. View connection details

View and Download Procedure

This procedure describes the methodology adopted to view and download applications which is given in detail below

Step 1: Files already available in the cloud are displayed.

Step 2 : Click on the download button next to the file which the users want to download.

Step 4: The file name being downloaded is displayed

Step 5: As soon as the download is completed the data pack or Wi-Fi which is on, turns off automatically and the present battery level is displayed on the screen.

4. RESULTS AND DISCUSSION

The login page for accessing using the cloud with due authentication with a valid user ID and password are shown in Figures 3 and 4. The list of files displayed after accessing the cloud is shown in fig.5. Once the user is authenticated successful upload or download of different types of files from the cloud is possible. Immediately after the completion of the tasks, the relevant applications are closed down in order to save the energy in the batteries of handheld devices as shown in figure 6.

Fig -3: Login Screen

Fig -4: User Registration

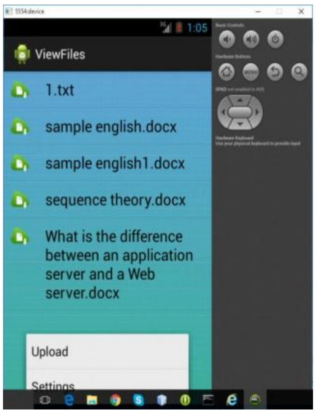

Fig -5: User views different files

Fig -6: Connection Details

5. CONCLUSION

Usage of advanced software in smart phones results in degraded performance and shortened battery life time due to use of limited resources. It also consumes both time and energy to upload data or programs to the cloud and retrieve the results from the cloud. The proposed work has shown considerable amount reduction of both time and energy consumption. The results show that the energy conservation in batteries is possible due to closing down of relevant applications used for execution. Further it can be extended to different wireless technologies

REFERENCES

[1] Lavendershih, "Booming popularity of smartphone helps to increase the flash demand," DRAMeXchange, Tech Rep.,2011.

[2] [2] X. Gu, A. Messer, I. Greenberg, D. Milojicic, and K. Nahrstedt, "Adaptive offloading for pervasive computing,"IEEE Pervasive Comput.,vol.3,no. 3, pp. 66–73, Jul./Sep. 2004.

[3] [3] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: A partition scheme," inProc. 2001 Int. CASES, 2001,pp. 238–246.

[4] [4] S. Ou, K. Yang, and J. Zhang, "An effective offloading middleware for pervasive services on mobile devices,"Pervasive Mobile Comput.,vol.3,no. 4, pp. 362–385, Aug. 2007.

[5] [5] G. Chen, B.-T. Kang, M. Kandemir, N. Vijaykrishnan, M. J. Irwin, and R. Chandramouli,"Studying energy trade offs in offloading computation/compilation in java-enabled mobile devices,"IEEE Trans. Parallel Distrib. Syst., vol. 15, no. 9, pp. 795–809, Sep. 2004.

[6] [6] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," inProc. IEEE IPDPS, Apr. 2008, pp. 1–8.

[7] [7] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui:Making smartphones last longer with code offload," inProc. 8th Int. Conf. MobiSys, Jun. 2010, pp. 49–62.

[8] [8] S. Han, S. Zhang, and Y. Zhang, "Energy saving of mobile devices based on component migration and replication in pervasive computing," inProc. Ubiquitous Intell. Comput., Aug. 2006, pp. 637–647.

[9] [9] Y. Hong, K. Kumar, and Y. Lu, "Energy efficient content-based image retrieval for mobile systems," in Proc. IEEE ISCAS, May 2009, pp. 1673–1676.

[10] [10] B. Seshasayee, R. Nathuji, and K. Schwan, "Energy-aware mobile service overlays: Cooperative dynamic power management in distributed mobile systems," inProc. 4th ICAC, Jun. 2007, p. 6.

[11] [11] Y. Wang, B. Donyanavard, and K. Cheng, "Energy-aware real-time face recognition system on mobile CPU–GPU platform," inProc. 11th ECCV, Sep.2010, pp. 411–422.

[12] [12] X. Zhao, P. Tao, S. Yang, and F. Kong,"Computation offloading for H.264 video encoder on mobile devices," inProc. IMACS, Oct. 2006, pp. 1426–1430.

[13] [13] Giurgiu, O. Riva, D. Juric, I. Krivulev, and G.Alonso, "Calling the cloud: Enabling mobile phones as interfaces to cloud applications," in Proc. 10th ACM/IFIP/USENIX Int. Conf. Middleware, Dec. 2009, pp. 1–20.

[14] [14] R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, and H. Bal, "eyedentify: Multimedia cyber foraging from a smartphone," in Proc. 11th IEEE ISM, Dec. 2009, pp. 392–399.

[15] [15] Linden, D. and Reddy, T. Secondary Batteries—Introduction. In Handbook of Batteries, ed. D. Linden and T. Reddy, pp. 22.3–22.24. New York, NY: McGraw-Hill, 3rd edition, 2002; http://www.knovel.com/knovel/databook/pdf/627/59788_22.pdf.

[16] [16] Riezenman, M. Mighty mites. IEEE Spectrum 40, 6 (June 2003), 30–33;http://www.spectrum.ieee.org/spectrum/jun03/features/mm.html (subscription required).

[17] [17] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Callingthe cloud: Enabling mobile phones as interfaces to cloud applications," in Proc. 10th ACM/IFIP/USENIX Int. Conf. Middleware, Dec. 2009,pp. 1–20.

[18] [18] R. Kemp, N. Palmer, T. Kielmann, F. Seinstra, N. Drost, J. Maassen, andH. Bal, "eyedentify: Multimedia cyber foraging from a smart phone," inProc. 11th IEEE ISM, Dec. 2009, pp. 392–399.

[19] [19] Y. Lin, Y. Lin, Y. Lai, and C. Lin, "VPN gateways over network processors: Implementation and evaluation,"J. Internet Technol., vol.11, no.4, pp. 457–463, Jul. 2010.

[20] [20] K. Yang, S. Ou, and H. Chen, "On effective offloading services for resource constrained mobile devices running heavier mobile Internet applications," IEEE Commun. Mag., vol. 46, no. 1, pp. 56–63, Jan. 2008.

[21] [21] Y. Zhang, X. Guan, T. Huang, and X. Cheng, "A heterogeneous auto offloading framework based on web browser for resource-constrained devices," inProc. 4th ICIW, May 2009, pp. 193–199.

[22] [22] A. Miettinen and J. Nurminen, "Energy efficiency of mobile clients in cloud computing," in Proc. 2nd USENIX Conf. HotCloud, Jun. 2010,p. 4-4.

[23] [23] C. Wang and Z. Li, "A computation offloading scheme on handheld devices," J. Parallel Distrib. Comput., vol. 64, no. 6, pp. 740–746,Jun. 2004.

[24] [24] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clone cloud:Elastic execution between mobile device and cloud," inProc. 6th ACMSIGOPS/EuroSys, 2011, pp. 301–314.

[25] [25] C. Cai, L. Wang, S. U. Khan, and J. Tao, "Energy-aware high performance computing: A taxonomy study," inProc. 17th IEEE ICPADS, 2011,pp. 953–958.