ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 3 (July - Sept. 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Research Paper on Self Learning Offloading Scheme for Android Phones

[I]Sheenam, [II]Sugandha Sharma, [III]Mayank Arora
[I,II]Dept. of CSE, CGC Gharuan Punjab, India
[III]Dept. of CSE, CCET Punjab University, Chandigarh, India.

Abstract
A new offloading scheme which is self learning that has been implemented in this paper. This offloading scheme creates a database of previously taken offloading decisions which helps in avoiding the decision making each time an application runs. In this paper, the focus has been made on enhancing the capabilities of smartphones by using virtualization techniques to shift the workload from smartphone to a resource rich computational Cloud environment.

Keywords
Cloud Computing, Mobile Computing, Offloading.

I. Introduction

Cloud Computing is the use of computational resources (hardware and software) that are delivered as a service over a network. It makes a virtual pool of resources such as storage, CPU, networks and memory to fulfill the user's resource requirement and provides on demand hardware and software. Cloud Computing enables seamless access to the user applications and data from anywhere anytime in the world, unleashing the user from the confines of the desktop and making it easier for group members in different locations to collaborate. Cloud Computing portends a major change in how we store information and run applications. Cloud Computing has emerged as the great technology in term of scalability and portability. It has changed our view of carrying data and communication.

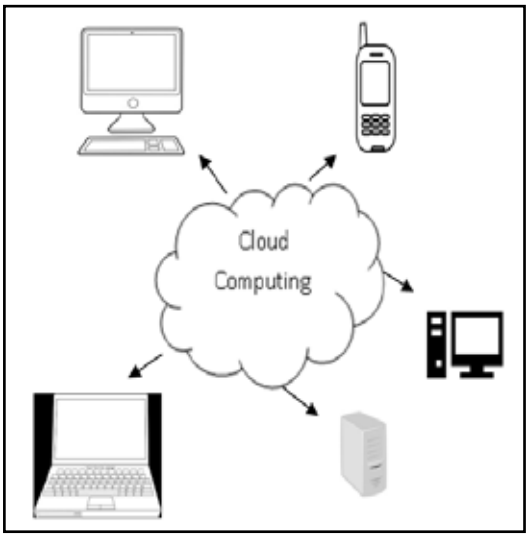

Fig. 1: Cloud Computing [8]

With the origin of cloud computing the idea of mobile cloud computing came forward. Mobile cloud computing means combination of mobile computing, cloud computing and mobile internet. In Mobile Cloud Computing, cloud's resources are used to increase the overall efficiency and performance of mobile phone [2].

Smartphones are the phones with feature rich operating system. Smartphones have advanced functionality from basic features phone like good internet connectivity, wifi, web browsing, and other applications. According to the requirement of user the developer develops the applications like 3D gaming, internet banking etc. These applications are resource exhaustive and smartphone have limited memory, battery life and computational power. To save phone's precious resources offloading concept came forward.

Android is Linux based operating system. Android is developed by Open Handset Alliance. Android architecture has 4 main layers i.e. Linux kernel, Libraries, Application Framework, Application, Android Runtime.

Features of Android

1. Beautiful User Interface: -Android operating system provide beautiful user interface.
2. Storage: - It uses SQLite databases for storage.
3. Multitasking: - User can use various applications at same time. User can easily jump from one application to the other.
4. Media support: - It support various extensions like MPEG-4, PNG, JPEG, BMP and GIF etc
5. Connectivity: - It uses Bluetooth, wifi etc. for connectivity purposes.

Offloading technology is used to offload the heavy computational to the cloud to save the phones precious resources. Many offloading techniques have been proposed which require an application need to be installed on phone which takes offloading decision to the other applications. The application which takes decision for the other application is compute intensive and it has some burden on phone. It requires changes to be made on operating system which is not practical [7].

Types of Offloading

There are two types of offloading

- Partial offloading
- Complete offloading

Partial offloading: - Those application is partially offloaded which is compute intensive and needed to be offloaded thus for reducing the load of the smartphone. In this the heavy computation task is being executed on the android virtual machines i.e. clone cloud and remaining part is executed on smartphone. When the android virtual machine has finished the execution of the compute intensive part of the application, it gets back the results to the smartphone and the smartphone collect all the results and provides to the customers in the required format.

Complete offloading: - In complete offloading unlike partial offloading the whole application is kept on the cloud. The cloud

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

acts as an exact replica of the smartphone with the same data. The application such as an antivirus runs on the cloud operating on the same data as in the smartphone. The results i.e. infected files if any are cleaned at the cloud side and the cleaned files are replaced on the actual smartphone. Doing this results in the user to have all the benefits of the applications without actually running it. The only drawback for this architecture is that the applications that fit into this architecture are very less, only the ones with very less or no user interaction.

Applications of mobile cloud computing

- Mobile Gaming: - A mobile game (m-game) is a video game played on a feature phone, smartphone, PDA, portable media player etc. M-game is completely offloading game. In mobile gaming, the engine wants large computing resources but the smartphones have no enough computing resources so it access the computing resources from the server of the cloud.
- Mobile Healthcare:- The main purpose of Mobile Cloud Computing in medical applications is to reduce the limitations of old medical treatment. Mobile healthcare (m-healthcare) provides convenient helps to mobile users to access resources (e.g., patient health records) easily and quickly. M-healthcare offers hospitals and healthcare organizations on-demand of patients from cloud servers.
- Mobile Learning:- Now a days, more and more mobile applications are coming up to provide educational content. As, these applications are running on cloud servers they solve many problems faced by the conventional mobile learning such as high device cost, less educational resources etc[6].

Architecture of Mobile Cloud Computing

- Access layer: Access layer include service interface to the user, service registration and reasonable service access. It provide all kinds of rules, service standards to mobile cloud computing, which is the bridge for cooperation between client and cloud end.
- Basic managing layer: The Managing layer provides management, managing system and services in mobile cloud computing system architecture. It can provide standard operations to services such as observing, authentication, directory, security and so on. It provide standard procedure interface and protocol to application service, hide dissimilarity between bottom hardware and operating system synchronizing, and manage network resources all together. The user management includes mobile account management, user environment configuration, user interaction management and accounting system. Task management includes task configuration, task execution and lifetime management. Resource management includes load balancing, error testing, error recovery and monitoring statistics. Security management includes user identification, access authentication, security audit and comprehensive protection.

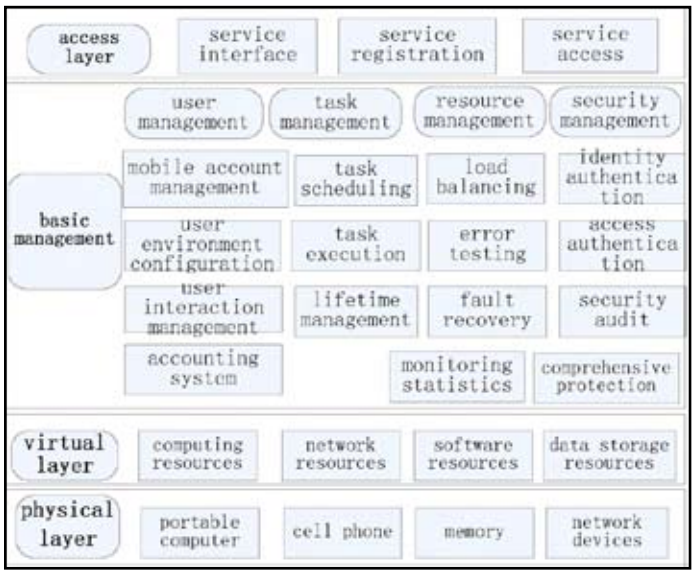

Fig. 2 : Architecture of mobile cloud computing [2]

- Virtual layer: Virtual layer include virtual items such as computing pool, storage pool and network pool. It has virtual environment, virtual system and so on.
- Physical layer: Physical layer indicates the hardware equipment such as portable computer, cell phones, memory, network devices and technology which supports mobile cloud services. A cloud with super service can be provided by distributed computers and by network technique and parallel technique.

II. Proposed Work

Through the survey it is identified that offloading a compute intensive application partially can improve the battery life of a smartphone, but the offloading system will acquire some overhead on the phone is main problem. At runtime of an application the overloading decision is made using input size and other important parameters that are needed by the cost model. The possibility is that the same decision is being taken again and again but only after checking all the parameters. The proposed solution for this problem is that if the offloading system is made self learning i.e. keeping a record of the previously taken decisions so that for future decisions could be made easily with the help of that data thus enabling the offloading system to self learn and making the system more reliable, energy efficient and fast.

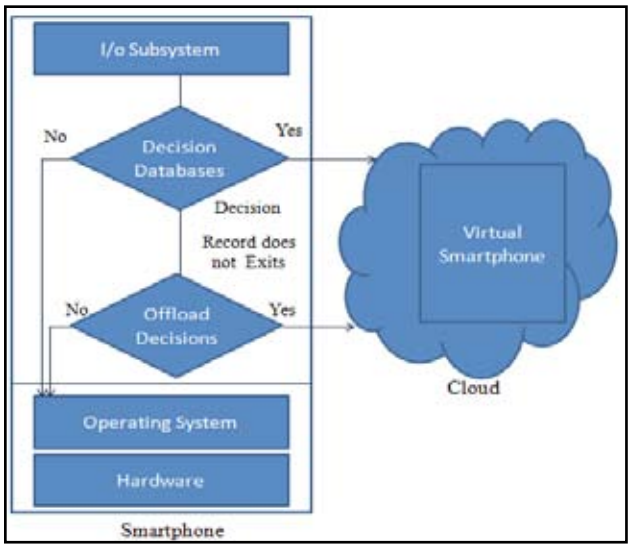

Fig. 3: Proposed Architecture[8]

The figure 3 shows the working model of the proposed architecture. The Input output Subsystem collects the inputs and extracts the parameters from those inputs and passes them to the decision database. If any previous decision entry is there containing the same parameters the decision is taken as it is, else the parameters are passed to the offloading decision unit which takes the offloading decision. The application is either offloaded or not depends upon that decision. The information about the decision is also saved in the decision database [7, 8].

III. Block Diagram

I/O Subsystem: - The I/O Subsystem collects the input from user validates then and passes the information to the subsequent user for processing. It also receives the processed data from the subsequent unit and provides the output in the required format to user.
Decision Database: - The unit keeps a record of the decision taken previously including all the parameters. During the first few runs the decision database populates itself with the decision information. In the subsequent runs the decision database directly provides the decision for the received parameter to the offloading subsystem. If the received parameters do not match the record of the decision tables, the parameters are as it is passed on to the offloading subsystem and the decision taken by the offloading system is saved as new entry.
Offloading Subsystem: - It is responsible for the offloading applications to the cloud. The subsystem runs the offloading decision, if the decision is not provided by the decision database, otherwise it realize on decision database.

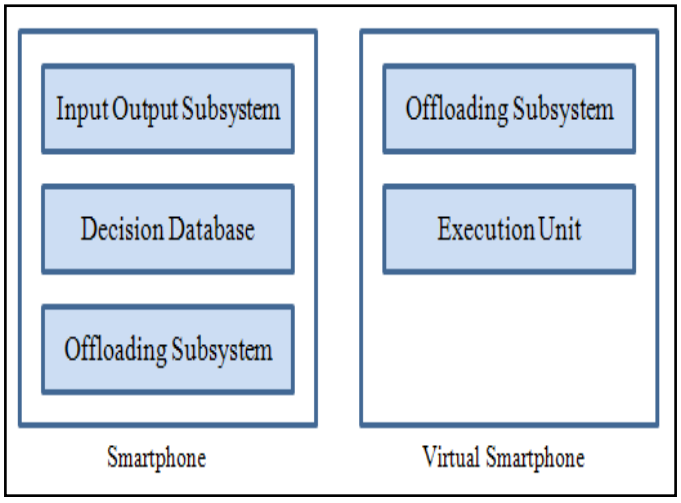

Fig. 4 : Block Diagram of Proposed Architecture.

Offloading Subsystem: - It receives the inputs from the smartphone and provides it to the execution unit for execution. After the processing is done, it receives the output from the execution unit and sends it to the smartphone.
Execution unit: - The execution unit simply execute the program after receiving its input from the offloading subsystem and provides the output to the offloading subsystem after the execution is done.

IV. Results and Discussion

We used Samsung Galaxy Y Duos Smart Phone whose to run our standalone application to generate prime numbers ranging between 8 to 1500 Bits. The table shows the time consumed by the prime number generator application while running in three different platforms i.e. first we ran it on the smartphone itself then we tried running the application on the virtual smartphone using the offloading technique in ACOF [7]. Then we executed the application using our proposed scheme. In order to get the desired results we ran the application several times in order to populate the database with the offloading decisions. As the number of runs goes on increasing the performance gets on increasing up to a certain level after which the performance stops increasing and a stagnant situation occurs. It is a must to keep in check that the application will not keep on saving data endlessly because this will lead in unnecessary overloading of the smartphone as this data will be occupying the precious memory of the smartphone.

Table 1: Execution Time

INPUT SIZE	EXECUTION TIME IN MILLISECONDS				
	SMART PHONE	VIRTUAL SMART PHONE	1RUN	10 RUN	20 RUN
8 Bits	13	16	16	16	14
16 Bits	17	20	20	20	18
32 Bits	39	32	32	30	39
64 Bits	56	46	46	44	40
128 Bits	131	73	73	70	50
256 Bits	255	115	118	104	100
512 Bits	6114	1801	1810	1756	1601
1024 Bits	95312	28731	28740	28400	26200
1500 Bits	238171	32120	32132	31610	30100

The input to the prime number generator application is the number of bits our desired prime number should be. For instance in the first case when the input size is 8 Bits the application will generate a prime number of 8 Bits in about 13 Milliseconds on th smartphone and in 16 Milliseconds if offloaded using ACOF and 14 milliseconds after 20 runs of our application. Initially our application takes about 16 milliseconds which is a little greater than the actual smartphone because it actually executes on the smartphone but before execution it takes a decision of whether to offload or not.

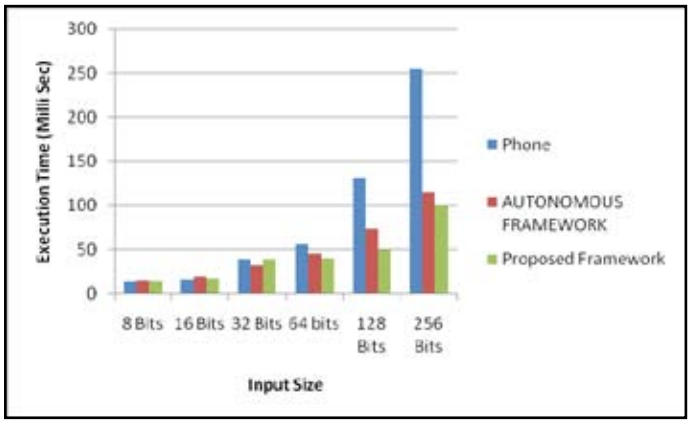

Fig. 5: Execution Time Graph Ranging from 8bits to 256bits.

As the size of the input goes on increasing the performance of both ACOF and our proposed architecture goes on increasing too. When the input size is 512 bits the smartphone takes 6114 milliseconds to generate the prime number but ACOF takes only 1801 milliseconds but after about 20 runs the proposed architecture took 1601 milliseconds to generate the same prime number.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

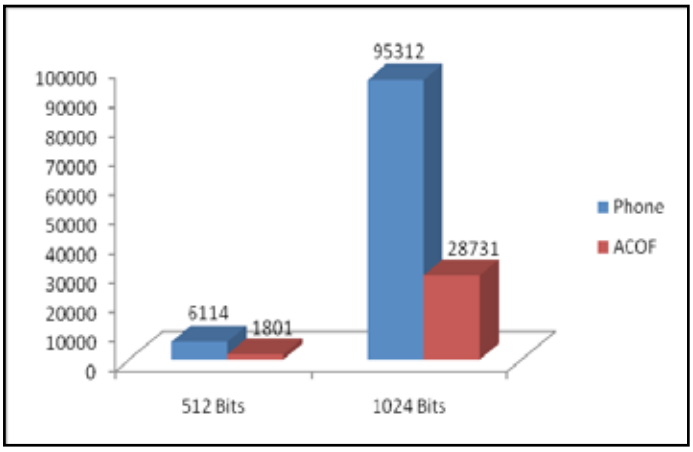

Fig. 6: Execution Time Graph Ranging from 512bits to1024bits

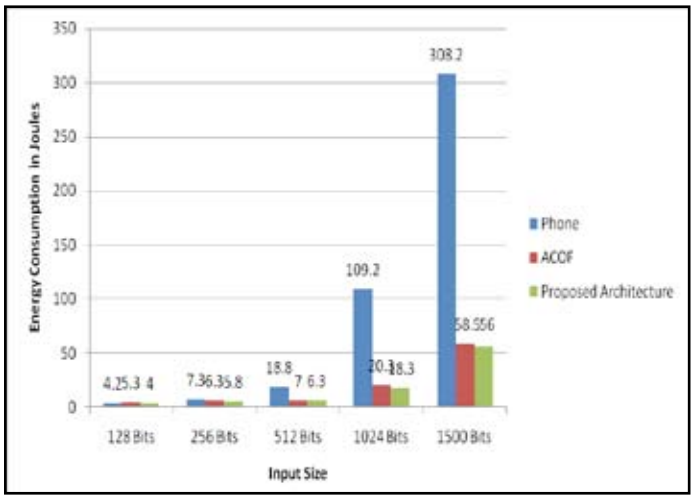

Fig. 7 : Energy Consumption Graph

Energy Consumption

The energy consumption is measured using a power tutor application. Most researchers use Power tutor for resource and battery management. The power tutor application provides accurate results of the energy consumed in joules by each application running on the smartphone. The key factors motivating the idea of offloading have been CPU and battery consumption. Thus, experiments were performed to analyse the battery consumption by the standalone application and the autonomous application using offloading techniques to generate prime numbers.

Table 2: Energy Consumption

INPUT SIZE	ENERGY CONSUMPTION IN JOULES				
	SMART PHONE	VIRTUAL SMART PHONE	1 RUN	10 RUN	20 RUN
128 Bits	4.2	5.3	5.3	5	4
256 Bits	7.3	6.3	6.3	6.1	5.8
512 Bits	18.8	7.0	7.2	6.9	6.3
1024 Bits	109.2	20.3	20.5	20	18.3
1500 Bits	308.2	58.5	59	58	56

To monitor the battery consumption by the applications we used a battery analyser tool named Power Tutor [15]. Power Tutor is an online power estimation system that has been implemented for Android platform smartphones. Power Tutor provides accurate, real- time power consumption estimates for power-intensive hardware components including CPU and LCD display as well as GPS, Wi-Fi, audio, and cellular interfaces. It uses power models and automated characterization techniques [16, 17].

V. Conclusion

This paper describe about the Cloud, Mobile Cloud Computing and the working of Mobile Cloud Computing. A new offloading scheme has been implemented in this paper which is self learning. This offloading scheme creates a database which collects the previously record, for taken offloading decisions which helps in avoiding the decision making each time an application runs. If the offloading system made self learning i.e. the applications made intelligent that lean from previously taken offloading decisions and thus reducing the overhead for offloading decisions and making the system more reliable, efficient and fast.

References

[1] *National Institute of Science and Technology , "The NIST Definition of Cloud Computing," July 2010. Available:http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.*

[2] *Weygand SONG and Xiaolong SU, " Review of Mobile Cloud Computing" in IEEE 3'rd international conference on Communication Software and Networks, May 2011, pp1-4.*

[3] *Niroshinie Fernando , Seng W. Loke Wenny Rahayu,"Mobile Cloud Computing: A Survey.*

[4] *Karthik Kumar · Jibang Liu · Yung - Hsiang Lu Bharat Bhargava, " A survey of Computation Offloading for Mobile Systems" Springer 2012.*

[5] *Xu Wang, Beizhan Wang, Jing Huang" Cloud Computing and its key techniques" 2011 IEEE*

[6] *Hoang T. Dinh, Chonho Lee, Dusit Niyato, and Ping Wang, "A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches" in proceedings conference of Wireless Communications and Mobile Computing – Wiley.*

[7] *Mayank Arora, Mala Kalra and Dr. Sarabjeet Singh,"ACOF: Autonomous Computation Offloading Framework for Android using Cloud", In Proceeding of IEEE 2nd International Conference on Information Management in the Knoweledge Economy,2013.*

[8] *Sheenam, Sugandha Sharma, Mayank Arora, "Self Learning Offloading Scheme for Android Phones" in Proceeding conference of International Journal of Advanced Research in Computer Science and Software Engineering, June 2104, pp. 1274-1279.*

[9] *A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning,*

"Saving portable computer battery power through remote process execution," in MCCR'98---ACM SIGMOBILE Mobile Computing and Communications Review Newsletter, Jan. 1998, Vol. 2,no. 1, pp. 19-26.

[10] B.G. Chun, P. Maniatis, " Augmented Smartphone Applications through Clone Cloud Execution," in Proceedings of 12th conference on Hot topics in Operating systems, 2009, pp. 8-8.

[11] Jaspreet Kaur, Sugandha Sharma, Mayank Arora," Mobile Cloud Computing Security Issues: Overview" in Proceeding conference of International Journal of Advanced Research in Computer Engineering and Technology, May 2104, pp. 1722-1726.

[12] B.G. Chun, S. Ihm, P. Manitis, M. Naik and A. Patti," Clone Cloud : Elastic Execution between Mobile Device and Cloud," in Proceedings of 6th Conference On Computer Systems, 2011, pp. 301-314.

[13] Android Phones When using the Wireless Communication Technologies" 2012

[14] Android (Operating System) , Wikipedia 2012. Available: http: // en .wikipedia.org/wiki/Android

[15] R. N. Mayo and P. Ranganathan, "Energy Consumption in Mobile Devices: Why Future Systems Need Requirements Aware Energy Scale-Down," in Proceedings of the Workshop on Power-Aware Computing Systems, October 2003.

[16] Sumandeep Kaur, Sugandha Sharma, Mayank Arora,"Data Synchronization between Mobile and Cloud" in Proceeding conference of International Journal of Advanced Research in Computer Science and Software Engineering, June 2104, pp. 1286-1292

[17] R. Kakerow, "Low power design methodologies for mobile communication," in Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors, pp. 8, January 2003.

[18] R. N. Mayo and P. Ranganathan, "Energy Consumption in Mobile Devices: Why Future Systems Need Requirements Aware Energy Scale-Down," in Proceedings of the Workshop on Power-Aware Computing Systems, October 2003.

[19] B.Zhao , Z.Xu, C.Chi , S.Zhu and G.Cao, Mirroring Smartphones for Good : A Feasibility Study. ZTE Communications.

Author Profile

Er. SHEENAM received B,Tech. degree in computer science and technology from Baba Isher Singh College of Engineering and Technology (PTU)and now pursuing M.Tech (CSE) from CGC Gharuan (Mohali). Research interest includes cloud computing and android.