International Journal of Trend in Research and Development, Volume 3(5), ISSN: 2394-9333
www.ijtrd.com

Overview of Cross-Platform Application Development Techniques for Smartphones

Rohit Kumar
Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, India.

Abstract— Smartphones provides us the capability of a typical computer with absolute mobility and small form factor. But these smartphones suffers from platform dependence i.e an application developed for a particular smartphone will not execute on other different smartphone (Smartphone with a different operating system). Due to this cost to the user per application increases. Efforts must to be made to tackle this problem and give user a seamless integration across multiple platforms. Solution to platform independence ranges from pure software implementation to pure hardware implementation. In this paper different ways to deal with platform dependence has been discussed across multiple width and dimensions. Comparative analysis of different approaches has been done against multiple performance parameters.

Index Terms— Smartphone platform independence, Cross-Platform application development for Smartphones, application portability for smartphones, smartphone operating systems.

I. INTRODUCTION

Smartphones are new age computing devices which have become very useful in day to day life. Looking at their capability we can call them computers with absolute mobility. These devices have features of typical mobile phone plus very powerful data computation capability. Present day smartphones have the capability to execute very complex applications which were initially possible on desktop computers. Typical functions performed by smartphones include internet surfing, word processing applications, multimedia applications, gaming etc. [1][2][3][4][5].

Since the smartphones are a young technology they open out a wide spectrum of research and development in hardware and software aspects of these small machines. The major problem associated with smartphones is that they don't provide platform independence i.e. an application written for Android phone will not work on blackberry phones supporting a different operating system. Due to this, same application has to be written according to the different smartphones software stacks or operating systems. Direct consequences of this are lots of rework and increased costs to the user [6][7].

Achievement of cross platform independence for smartphone has been the main objective of this discussion. Cross platform development can be achieved by software means or hardware means or combination of both. In this paper all major approaches has been discussed to achieve the platform independence. Software, Hardware, Distributed computation approach has been discussed in this paper to achieve cross-platform application development possible [8][9]. Numerous approaches to achieve cross platform independence have been proposed for normal computers. In recent time platform independence has been achieved by many software frameworks on desktop computers, but achieving the same on mobile platform has been not possible yet [10] [11][12][13][14].

Organization of the Paper: This paper has been broadly divided five major sections excluding the introduction. Section 2, 3, 4 discusses the three major approaches namely, virtual machine, distributed computation and hardware approach respectively. In section 5 comparative analyses of three approaches have been discussed. Section 6 the concluding section.

II. SOFTWARE APPROACH

Smartphones comes with number of different platforms e.g. Android, Symbian, and Windows and many more. Operating systems of smartphone are tailored according to the hardware that is to be used i.e according to the underlying hardware. This is done to take full advantage of the underlying hardware and to reduce energy consumptions.

Since hardware of smartphone is specialized and is less powerful as these devices are powered by batteries, so executing code meant for other smartphone is marginally possible, and in large number of cases produces poor results in terms of computational efficiency and power consumption. Recent work [2][15][16] on platform independence for smartphones have tried to achieve cross platform application development by developing virtual machines with various considerations. Partial platform independence has been achieved with web apps, PhoneGap, Titanium Mobile [2] platforms. Creation of virtual machine is a viable option available to achieve cross platform development by software means. The virtual interfaces and simulators can be provided as like the ones available for desktop PC's e.g. VMware.

A. Virtual Machine

A Virtual Machine is like a computer running within a computer. Actually the virtual machine is an interface between an application program and the kernel. Virtual machine takes different code developed for different machines as an input and provides executable code that can be executed on the underlying hardware.

There always remains a tradeoff between computational efficiency and cost. The computational efficiency of virtual machine is low because the code of virtual machine is very complex as it involves lots of translations and makes appropriate adjustments in the code. These overheads make the processing slow. But the obvious advantage is the cost savings it offers as the user saves the entire hardware cost. Figure 1.2 presents an abstract view of a system in which virtual machine is used.

This approach involving the use of virtual machines has the potential to solve the problem of cross platform development given rich set of resources like highly powered processors and large amount of caches, main memory and energy. But this setup has proved inefficient for smartphones which don't have rich set of resources as available for desktop computers. So, more refinements must have to be done to implement virtual machine in smartphones for making cross-platform development possible. In the recent work done on platform independence for smartphones some virtual machines has been

implemented PhoneGap, and Titanium Mobiles to name few. But none of the virtual machine framework have proved rigorous and complete.

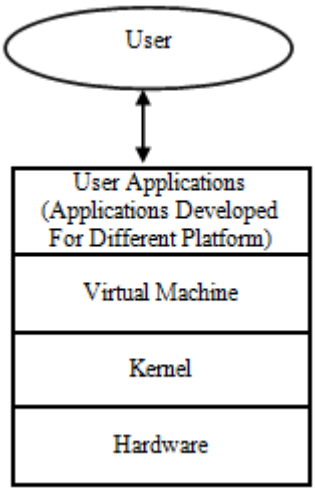

Figure 1: System architecture With Virtual machine

The major reasons contributing to the problem is that many flavors of operating systems exist. The problem can be solved if the operating system flavors offered are limited to two or three. The Titanium virtual machine provides platform independence for two flavors namely iOS and Android. Since for complete platform independence the virtual machine code will have to be written for all the operating systems available which will result in very large code and hence makes the task of platform independence difficult to achieve as smartphone are battery powered and have limited hardware resources. Though, in the coming time the virtual machine approach may become viable given the hardware kept evolving at the same rate.

III. Remote Computation Approach

Remote computation approach is another approach which can be used to achieve cross platform application development for smartphones. This approach is similar to distributed computing environment where large computations and resources provided by powerful servers are shared by other computers on the network. The idea is to get the computation power from the servers. The servers will provide the services and software's for the user's smartphones. In this scenario there is little or no load in the client's smartphones.

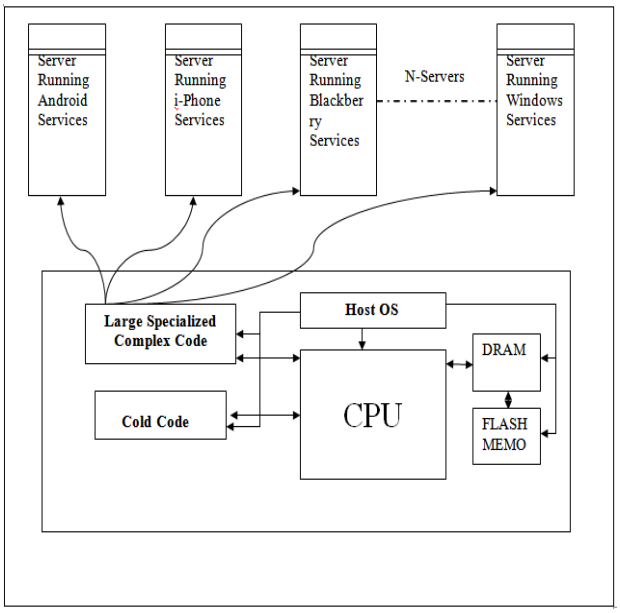

Figure 2: Architecture of remote computation for smartphones.

A. Architecture of Smartphone and Computation Methods

The remote computation approach relies on the powerful servers for providing computational capabilities to the smartphones for the purpose of achieving cross platform independence. Figure 1.2 presents the architecture for remote computation model. The idea is to take services from the servers which provide services for different smartphone operating systems. Depending on the application that the user wants to execute the concerned server proving the server will be called. Then it's the responsibility of the server to timely provide the results to the user.

B. Server Configuration

Servers used in this environment have very rich set of resources. They have powerful hardware in terms of processor power and memory availability. No power constrains exists as they are not dependent on batteries. In addition to this they have robust operating system and other system software's. They are also tightly integrated with the network by means of very high speed communication networks.

C. Client Architecture

The architecture of the client machines i.e the smartphones are inspired by the way the computations will be performed. The client can be of two types in first case only few resources are shared from the remote servers, in second case everything is shared form the servers as the client act as a dummy device.

The smartphones architecture depicted in Figure 1.2 is not of a dummy device type and favors partial services from the servers. The components of the Smartphone depicted in the figure have a central processing unit (CPU), a host operating system and main memory. CPU is supposed to execute general purpose and small code only, large and complex codes are executed by the remote servers. The computations can also be categorized to be of two types depending upon the computation performed at the smartphone end.

1. Partial computation at the smartphone end.
2. No Computation at smartphone end.

In partial computation, the CPU and other hardware are specialized according to the host operating system. So applications pertaining to that OS can be executed on the smartphone itself. Which means faster operation and fast response time. Non parent and large computations are performed at the respective remote servers.

The second case favors dummy smartphones where no application executes at the smartphone. Every computation is performed at respective servers. This approach is not followed as, due to network involvement as response time is always poor and any error in network renders the phone useless.

Many researchers in the past have emphasized that maximum computations should be performed at the phone end. It has been argued due to the fact that the network operation takes long time and are prone to other types of communication errors. As resources are put on network lots of communication takes place among devices. The following section puts some light on the communication scenario.

D. Communication Scenario and Code Offload

The communication between a smarphone and server usually consists of input provided by user to the application interface on his smartphone and the result sent by the server in response. The smartphone divides the computation in two parts namely local or small computation and remote or large computations.

Local computations are independent and have nothing to do with remote servers. On the other hand large and non parent computations can offloaded to remote servers for execution [17][18].

The user can also provide some code to the server for execution which cannot be executed on the client machine. So this process of sending code to remote server for execution is called code offload process. Figure 1.3 explains the code offload process. The smartphone side applications send requirements, tasks to the server then server finishes the tasks and sends back results to the smartphone. The network used for communication must be very fast and fault tolerant to accomplish code offload, even minor failure in network may let application running on smartphone to fail. Wireless networks used for communication must be optimized for good performance.

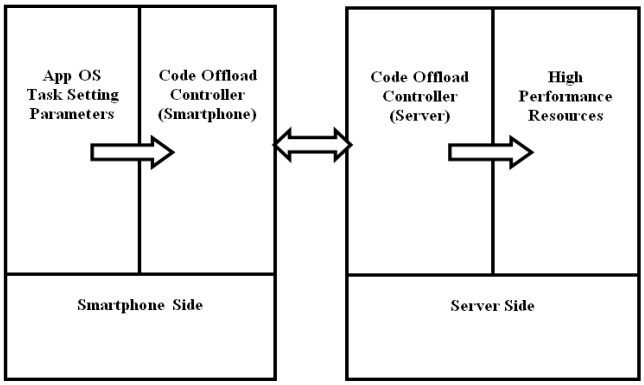

Figure 3: Code Offload Mechanism Diagram

E. Limitations of Remote Computation Approach

There are some inherent problems associated with network computation approach. As the operation of this approach heavily depends on communication networks so the performance of network plays a vital role in the success of this approach. Table 1.1 presents pros and cons of this approach. It can be seen from the table that this approach struggles with energy efficiency, latency, fault tolerance and robustness.

Owing to the above reasons this approach has not gained good acceptance, but for few services this approach has been used. Since some of the services anyway needs network access then those services can be better implemented with this approach. The partial computation approach can be easily adopted as it involves little communication overheads.

IV. HARDWARE APPROACH

This approach is concerned with design and implementation of hardware which is powerful and is very energy efficient. This approach if successfully implemented will be better than other approaches just studied as it will lead to efficient execution of processes in real time. Thus, making smartphones more versatile and capable. Even for implementing virtual machine powerful and energy efficient hardware is a prerequisite. So, efficient hardware development is a prerequisite for achieving cross platform independence on smartphones. In recent times work have been done on following factors to make energy efficient integrated circuits.

- Reduction in power consumption, heat dissipation.
- Reduction in leakage current.
- Dynamic clocking mechanism.
- Circuit design optimizations.
- Delay reduction.

Initially resistor-transistor-logic (RTL) was used to manufacture integrated circuits but now a day's complementary metal oxide semiconductor (CMOS) is used as it consumes very less energy as compared to initial RTL logics. CMOS consumes less energy as it has low power requirements and have low heat dissipation and very less current leakages. In addition to this CMOS circuits are very fast as they offer faster switching which also means lower delay.

Even after doing all these optimizations it's not possible to develop a powerful hardware for mobile phones. Advanced RISC machines (ARM) are used in present smartphones, though they consumes less energy but, fails when very large code is executed on mobile phones. So other approach for hardware design must be followed to make powerful processor which can work with battery powered devices.

In recent time problem of utilization wall and dark silicon has been observed by many researchers. It has been proved that by tackling the problem of utilization wall more than at least 7 times energy can be saved, that is not possible with present ARM processors.

A. Segmented Processor Design

Traditional processor has all its circuit components active all the times which contribute to lots of energy consumption although only a little area of processor is required for the current running application. If the segmented processor design is used then lots of energy preservation can be achieved by switching off the area of the processor which is not required.

In segmented processor design the processor area is divided into different segments of different sizes. Each of these segments is a complete processor in itself with all the circuitry like ALU and control unit and cache memory embedded in it. The processor segments come with reconfigurable hardware and can adapt to code changes. This adaptability is the key to success to achieve energy efficiency as the most frequently executed code can be embedded in hardware. With this design the energy consumption can be reduced at least 7 times than that of a conventional processor. Detailed discussion of this processor architecture is beyond the scope of this research paper.

Table 1: Comparison of various factors for Different approaches

Sr. No	Feasibility of Implementation	Energy Efficiency	Delay of Operation	Robustness	Operational Cost	Fault Tolerance
Virtual Machine Approach	Yes	Moderate	Moderate	Moderate	low	Good
Remote Computation Approach	Yes	Bad	Depends on the network Bandwidth	Moderate During Network up Time	High	Bad
Hardware Approach	Yes	Very Good	Very Less	Very Good	Moderate	Very Good

V. COMPARATIVE ANALYSIS OF DIFFERENT APPROACHES

There are various factors which are of utmost importance for evaluating the performance of different approaches discussed.

Table 1 presents these factors for various approaches. The importance of particular factor may vary from application to applications. E.g. the fault tolerance for some application like banking transactions or banking applications might be more important than that of energy efficiency. Form the table 1.1 it can be interpreted that the hardware approach offers better performances on almost all the desired factors ranging from energy efficiency to fault tolerance. The virtual machine approach stands next to it. This approach appears to be more flexible in terms of development and offers moderate parameters for major issues. The last approach namely the remote computation approach struggles mainly with fault tolerance. This approach isn't suitable for the mode of operation when no computation is performed locally at the smartphone. But for partial computation approach this approach is promising.

CONCLUSION

Each of the approach discussed has its own pros and cons and work is being done on all of these techniques. The most suitable method however is to use virtual machine approach but that in turn necessitates the development of powerful and energy efficient hardware or processor. It has been predicted by the recent researchers that such a powerful processor will be a reality soon. The third approach of network computing offers a great flexibility but is susceptible to network failures and delays and can't work in the absence of network. But this approach can be quite good when we have some constraints on mobility and a very robust network.

The segmented processor design with the capability to reconfigure its segments will be a major success for all the mobile platforms. Here the term reconfigurable means the processor segment can reconfigure itself according to the code changes. With this the energy savings will be enormous.

Many smartphones operating systems are proprietary software's and accessing their API's and Library is very difficult. Various vendors can be brought under one roof to discuss issues which pose hindrance in the process of platform independence. Their operating system architectures can be made more open for problem mitigation.

References

[1] Taneja, Kavita, Harmunish Taneja, and Rohit Kumar. "SPF: Segmented processor framework for energy efficient proactive routing based applications in MANET." *2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*. IEEE, 2015.

[2] K. Taneja et al., "Cross-Platform Application Development for Smartphones: Approaches and Implications", *3rd International Conference on "Computing for Sustainable Global Development", IEEE, 2016.*

[3] R. K. et. al., "Specialized hardware Architecture for Smartphones", *Intl. Journal of Engineering Research and Applications, ISSN: 2248-9622, pp.76-80.*

[4] R. Kumar et al., "Smartphones hardware Architectures and Their Issues", *International Journal of Engineering Research and Applications, ISSN: 2248-9622, pp.76-80.*

[5] Deepinder Kaur. et al., "Energy Named Entity Recognition, Extraction and Classification using Conditional Random Field with Kernel Approach*", IJEAR, ISSN: 0973-4562,Vol. 10, Issue:15, pp. 38193-38198.*

[6] S.K. et al., Comparing Routing Protocols, *IJEMER, ISSN: 2249-6645, Vol. 4, Issue: 2, pp. 36-42.*

[7] K. Taneja et. al., "EEGRP: Grid Based and Caching Equipped Energy Efficient Routing Protocol for MANET with Restricted Movement", *Far East Journal of Electronics and Communications, ISSN: 0973-6999 Special Volume 3, Part I, pp. 185-199, 2016.*

[8] R.K. et. al., "Smartphone's hardware and software development Issues", *IJES, ISSN: 2320-0332, pp. 69-75.*

[9] Wen-Hwa Liao et al., "GRID: A fully Location Aware Routing Protocols for MANET", *Telecommunications Systems, pp. 1548-1557 (2001).*

[10] Y.-C. Tseng et al., "Power Saving Protocol for IEEE 802.11 based Multi-hop MANET", *IEEE, pp. 37-60, (2001).*

[11] Shiv K. Verma et al., "Hybrid Image Fusion Algorithm using Laplace pyramid and PCA Method" *Proceedings of ICTS-2016 ACM Conference, ISBN 978-1-4503-3962-9, pp. 1-6, 2016.*

[12] L. Pawar et. al., "Optimized Route Selection On The Basis Of Discontinuity And Energy Consumption In Delay Tolerant Networks", *Springer: Advances in Intelligent Systems and Computing, ISSN: 2194-5357, 2016.*

[13] R. K. Bhullar et al. "Stress Calculation in Parallel Processor Systems Using Buddy Approach with Enhanced Short Term CPU Scheduling", *Proceedings of the International Conference on Communication and Computing Systems (ICCCS 2016), ISBN: 9781138029521, 2016.*

[14] M. Kaur et al., "Stochastic Approach for Energy-Efficient Clustering in WSN", *Global Journal of Computer Science and Technology, Vol. 14, Issue 7, pp. 1-8.*

[15] Rohit K. Bhullar et al., "Intelligent Stress Calculation in Segmented Processor Systems Using Buddy Approach", *Journal of intelligent and fuzzy systems" IOS press, ISSN: 1064-1246, In Press.*

[16] Rohit kumar Bhullar et al, "Test Your Programming Quotient for C", *Param Hans Publisher New Delhi, ISBN: 978-1-329-80946-8, Book.*

[17] Rohit Kumar et. al, "Pen Drive With OS Controlled Inbuilt Permanent Data Storage Partition", *Patent 2111/DEL/2015.*

[18] Rohit Kumar et. al, "Real Time Cleanliness Status Monitoring System For Public Toilets", *Indian Patent 1957/DEL/2015.*

[19] R.K. Bhullar et al., "A Novel Prime Numbers Based Hashing Technique for Minimizing Collisions" *2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016). IEEE, 2016*