Real-Time Android Operating System Using Hypervisor

Nida Yasir  
Lecturer, Department of Computer Sciences, University of Lahore  
Sargodha Campus, Pakistan

Hina Khalid, Sidra Hassan and Kulsoom Fatima  
Students of MSCS, Department of Computer Sciences, University of Lahore  
Sargodha Campus, Pakistan

Abstract

From its launch to public, Android has gained the interest from organizations to applications developers and the universal audience. Until now this software platform continuously improved its features or in context to its hardware, and simultaneously extended its features of new types of devices which are different from initially developed mobiles. However, its real time capabilities are one of the features which are not explored yet. This paper directs to look at this gap and gives the basis for the suitability of android for use in the real time environment. With the main focus on the virtual machines, this paper describes a new hypervisor built to run Linux in a virtual machine. We suggest that android provides appropriate architecture for real time embedded system using Real-time hypervisor and feature paravirtualization embedded Linux. This suggested approach provides the basis of the future implementation of android operating system using capabilities of real time hypervisor.  

Keywords: Android, Dalvik VM, Real time system, Real time hyper visor, Paravirtualazation

I. Introduction

Android is an operating system relies on Linux Kernel. It is first developed by Android Inc. And the first Android Power Phone was available to consumers in December 2008, running on HTC Dream phone. Android delivers a complete set of software for mobile devices like it delivers an operating system, key mobile applications and Middleware [1]. It is open source software architecture the code was declared under Apache license. 

Its architecture is provided by the OHA (open handset Alliance), this is a group [3] of 71 technology and cell phone companies whose ambition is to provide a mobile software platform. It is designed primarily for touch screen mobile devices such as smart phones and tablet computers. Android is free to use for applications, become better day by day due to its high demands and popularity, and designed with multiple hardware implementations. It is one of the most popular used Operating System for mobile industry. The features Android integrates similar characteristics found now days in any mobile device platform, such as: Application framework, optimized graphics, media support, network technologies and broad cost receiver. 

Due to different hardware architectures, Android is gaining power in the mobile industry as well as in other industries. Due to its two core aspects (1) open source (2) architecture model, it is in the lime light now-a-days. Being an open source has made android easy to understand and analyzed. By using the Linux kernel in its architecture android uses all the Linux features in the mobile industry. 

Another important feature of android is its virtual machine environment. Android uses it own virtual machine that is Dalvik. Dalvik VM was specifically developed for mobile devices. It has advantages like: memory optimization, battery power saving and low frequency CPU. For memory management and scheduling (core OS features) Dalvik relies on the Linux kernel. Android applications are Java based. 

As android has many features, but still a feature that has not been explored yet is its real time capability. Android platform is not suitable for the real time environment. Although Linux provides a feature of basic preemptive fixed priority scheduling policy, but still it is impossible to achieve real
time behavior in the Android OS. To replace the fixed priority scheduling with dynamic scheduling schemes efforts have been made. These schemes use the concept of dynamic deadline. These scheduling schemes provide full CPU utilization. But at the same time when system is facing overload problem dynamic scheduling present unpredictable behavior [5].

This paper presents a direction to provide suitability of open real time environment in android platform. An approach of the real time hypervisor was present to provide the solution of the problem [5]. When hypervisor is used in android architecture, then a problem comes like when the hypervisor hangs all the system also hangs. This paper proposed a solution to this problem by using paravirtualization in Android architecture. The rest of the paper is divided into the following: section 2 discussed real time hyper visor, section 3 discussed paravirtualization, section 4 discussed related works, section 5 discussed android architecture, section 6 discussed proposed work and section 7 discussed conclusion.

II. Real-Time Hyper Visor

A hyper visor can be a software, firmware or may be hardware. It creates and runs virtual machine and also called a virtual machine monitor (VMM). The Host Machine is a computer on which hypervisor runs virtual machine. On a host machine one or more virtual machines (Guest Machine) can run [6]. A real time hypervisor has the capability to run android (as a guest operating system) in one of the partitions and real-time applications in a different partition. Real- hypervisor runs both Android and real time application in parallel manner. Hypervisor are classified into two different types. Type1: Hypervisor that installs directly onto a computer. There is not any host and the hypervisor has direct access to all kinds of hardware and to its different features. Type2 (Hosted): these required a host operating system and are mostly treated as installed software inside the host.

For interaction with hardware and background services Android uses a Linux OS, and then uses a virtual machine called Dalvik to run different software to which user interact with. Despite android not allowing a user to run multiple operating systems at a time, it is similar to type one hypervisor [10].

III. Paravirtualization

The main idea of virtualization is creating a virtual operating system that runs in the top of another operating system. Virtual operating system in this environment is called guest and the other one is being hosted. Virtualization enables today's X86 computers to run multiple operating systems and applications. It makes infrastructure simpler and more efficient. Applications get deployed faster, performance and availability increase and operations become automated. It is easier to implement and less costly to own and manage. [14]

In computing, Paravirtualization is a technique of virtualization that presents a software interface to virtual machines that is similar but not identical to that of the underlying hardware [11]. Paravirtualization the guest knows it does not run on hardware in contrast to full virtualization where
the guest acts like it runs on real hardware. A hardware environment is not simulated in a paravirtualization. The guest programs are executed in their own isolated domains, as if they are running on a separate system.

IV. Related Work

In [5] Maia et.al points out the current limitations of open real time environment in android OS platform and provides a hint on different perspectives. Maia provides different directions in order to make Android suitable for real time environment. Maia et.al mentioned four possible directions for implementing it as real time Operating System i.e.

1\textsuperscript{st} Direction: describes that Linux operating system should be replaced by an OS that provides real-time features with an addition of a real-time VM.

2\textsuperscript{nd} Direction: proposes the extension of Dalvik as well as the replacement of the standard operating system by a real-time Linux-based operating system.

3\textsuperscript{rd} Direction: substitute the Linux operating system with a Linux real-time version. Real-time applications use the kernel directly.

4\textsuperscript{th} Direction: proposes the addition of a real time hyper visor to support the execution of android platform and the real-time applications parallel.

V. Android Architecture

The android architecture, described in Figure 1 is composed of five layers such as Applications, Application framework, Libraries, Android runtime and finally the Linux kernel.

The topmost layer is the application layer provides the core set of application that is frequently offered out of a box with any mobile device. Such as SMS client map, dialer, web browser, contact manager.

After that there is an application framework written in Java language. It provides the framework application programming interfaces (APIs) used by the application executing on uppermost layer. Apart from the APIs, there is a set of services that enables the access to android’s core features like graphical components, information exchange, activity manager, and event manager as examples. Application framework enables reuse and replacement of component [2].

Below the application framework layer, there is another layer containing two essential parts: Libraries and the Android Runtime. The libraries provide the core features to the applications. Library is the layer that enables the device to handle different types of data. These libraries are written are in c or c++ language and are specific for particular hardware. Some of the important native libraries include the following such as surface manager, media framework, SQLite, Web kit, OpenGL. Among the libraries the most important are libc, the standard C system library turned for embedded Linux-based devices; the Media Libraries which maintain or support playback and recording of several audio and video formats; Graphics Engines, a lightweight relational database engine and 3D libraries based on OpenGL ES.

Regarding the android Runtime, besides the internal core libraries, Android Runtime consists of Dalvik Virtual machine and core Java libraries. The Dalvik virtual machine is developed by Dan Bornstein by Google. It is specially designed and optimized for android. Dalvik was designed from scratch (graphical programming language) and it is specifically targeted for memory-constrained and CPU-constrained devices. It runs Java applications on top of it; unlike the Java VM the Dalvik virtual machine does not run class files and JVM is stack based. It is optimized for mobile devices.

The Dalvik is an infinite register –based machine. Being a register machine, it shows two advantages when compared to stack-based machine. Namely, it requires 30\% less instruction to perform the same computation as a typical stack machine, causing the reduction of instruction send and memory access, and less computation time, which is also derived from the omission of common expression from the instruction. After all Dalvik presents 35\% more bytes in the instruction stream than a typical stack machine. This drawback is satisfied by the consumption of two bytes when consuming the instruction.

At the bottom of the layer is the Linux Kernel version 2.6 with approximately 115 patches. It is also a hardware abstraction layer that enables the interaction of the topmost layers with the hardware layer.
through device drivers. Furthermore, it is also provides the most fundamental system services such as security, memory management, process management and network stack. Android is built on a most popular and proven foundation. It made the porting to diversity of hardware, a relatively painless task.

VI. Purposed Work

As we describe all the features of android operating system above. An approach is to set out a real time hyper visor which is able of running two different partitions in parallel manner. In one partition Android as a guest operating system and real time applications in the second partition. The majority of the recent real time Linux solutions such as RTLinux [8] RTAI [9] are similar to this approach. These systems provide the facility to run real time applications in parallel to the Linux kernel, although the real time tasks have high priority instead the Linux kernel tasks, it expresses that hard real time can be used.

A virtualized system does not have full access to the hardware. In an x86 systems a guest mostly does not run in the most privileged ring. (E.g. When system needs all resources) and then calls the host to perform these operations. In fully virtualized environment the guest tries to perform the operations by itself and it fails to do so, failure would be captured by the host. On the other hand, in paravirtualized guest tells the host to perform the operation on its behalf, without trying the operation first. A communication from host to guest is called hyper call.

We choose paravirtualation because it makes easier to build a hypervisor (no need to follow all hardware in the host). One more advantage of paravirtualization over full virtualization is that it is faster than that. Linux supports many guest operating systems. Virtualizing Linux has already been done by many open source operating system, e.g. KVM [12] and XEN [13].

The idea we purposed for limitation and the problem we felt related to the architecture of the android operating system related to lack of real time capabilities. We have been using the technique of paravirtualization in which we added one extra hardware layer, of paravirtualizing Linux.

This works whenever the hypervisor will hang. Whenever the halt situation will be occur the interrupt vector to generate an interrupt to this hardware through hyper call, then that specific running application that halts, will transfer its control to an extra added layer paravirtualazation Linux for a while. After the system recovers from halting situation and continued its working, that running application will be transferred to the system after an interrupt. All the application work transferred from one layer of architecture to another layer through a special type of interrupts i.e. hyper call. Hypervisor receives hyper calls from the system and handle them. It’s the responsibility of the hypervisor to redirect these interrupts to the correct interrupt handler.

Our purposed design is given above for supporting the real time hypervisor feature using the paravirtualizing feature.
VII. Conclusion

Paravirtualizing Linux in a real time system is not difficult as we think, because Linux has its mechanism to simplicity the porting on a new hyper visor. Future work will be focused on the interface. Android has many features, but its architecture does not provide stability in real time Environment. Taking this feature into consideration many directions were proposed for evaluation of the android architecture to be used as a real time system. One direction was to use the real time hyper visor in android architecture, but it has some issues. One of them was, when the real time hyper visor hangs then real time application also hangs. This paper proposed a solution of this problem by using Real-time hypervisor and paravirtualization Linux.

References