

Embedded VMM for Portable Virtual Machines

Naveen Kalla, Patrice Guelah and Scott Armstrong
(nkalla2, pguela2, sarmstr2@uiuc.edu)

The biggest problems with embedded Linux-based mobile phones are

- **Licensing issues** – Linux is distributed under the GPL license and hence all derived code including the device drivers that are loaded into the kernel is subject to the same license and thus become open source. But for chip vendors who consider device interfaces as valuable proprietary IP, this could be a potential problem.
- **Security Threat** – If the operating system is compromised and the communication stack was running under this operating system, then there is a potential problem that the attacker can turn the phone into a jammer and thus disable communication in the entire cell. This could cause failure of even emergency calls for all mobile users in that cell.

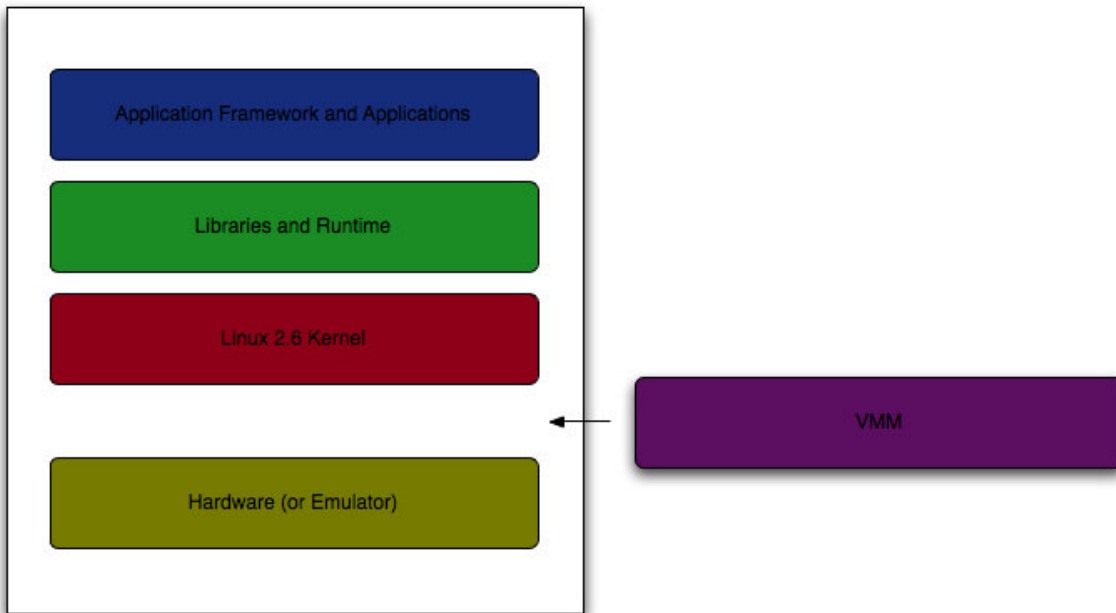
We haven't seen any of these problems earlier because most of the mobile phones used proprietary operating systems and home grown applications. Moreover most mobile phones used a different processor for running the communication stack (frequently called baseband processors) and a different processor for running the applications (frequently called application processor). The real time requirements of the communication protocols and the processing power of the legacy processors made it inevitable to have two processors. Extra hardware was needed to facilitate communication between the two processors thus resulting in bigger die sizes and higher bill of materials. Hence it was tougher to have high end phones in smaller and thinner form factors.

In the interest of miniaturization, a single processor could be used for both the communication stack and the applications with the help of Virtual Machine Monitors (VMM). VMMs have been used to solve a variety of problems or challenges over the years [6], [7], [8] and they will likely continue to be applied as a tool to discover and solve new problems. Server Consolidation, Testing and development, Dynamic Load Balancing, Disaster Recovery, Virtual Desktops, Improved System Reliability and Security are some of the areas where virtualization has shown its advantages. Embedded VMMs allow us to run multiple operating systems concurrently on the same hardware platform. As more processing power is being packed into a single processor, we could probably use a single processor for both the communication stack and the applications. We can solve the problems mentioned above by running the communication stack under a real-time operating system as a virtual machine and linux as a separate virtual machine on the embedded VMM. This would provide isolation to the communication stack in addition to numerous other benefits provided by the embedded VMMs [9], [12].

VMMs have been deployed for years by VMWare and Xen from the University of Cambridge, both aimed at the enterprise market. Recently virtualization solutions aiming at embedded designs have emerged, such as L4/Wombat from the University of New South Wales, and OK4 from Open Kernel Labs. Though VMMs are not new [4] [5], there has been very little work toward virtualizing ARM-based devices [3]. We hope to take the current research further and demonstrate a viable, real-world application of VMMs on portable devices.

We plan to develop a VMM for a mobile phone, bringing the benefits of a VMM to the handheld, portable device world. We will use the *Linux-based Android system* to demonstrate the benefits of the attributes of VMMs in the mobile phone environment. The Android system [1] architecture consists of an operating system layer running a modified Linux 2.6 kernel. This architecture is designed to run on the *ARM-based processor*. Development tools for the Android system include a hardware emulator to run and test applications developed for the Android system (Linux OS, libraries and applications) [1]. This emulator makes use of the QEMU ARM Instruction Set Architecture (ISA) [2] to emulate an ARM-based hardware telephone set. QEMU uses a dynamic translator [10]. The dynamic translator performs a runtime conversion of the target CPU instructions into the host instruction set. Since our primary goal is not to emulate the hardware architecture which QEMU and other solutions such as Skyeeye [11] already do, we will not do any translation. Instead we will focus on the primary goals listed below.

Our VMM that will run between an ARM-based phone (or Emulator) and the Linux OS of the Android system.



The main goal of this VMM development and enhancement effort is to demonstrate the VMM-enabled ability to run multiple software “phones” on the same device. The secondary goal is to develop a means to simplify the migration of the complete Android system software stack (OS with system state, libraries, and applications) between two hardware (or emulated hardware) environments, taking advantage of the check-pointing capabilities of VMMs.

Our proposed milestones and timeline for this project are as follows.

- February 2008 - Setup the emulated platform and run bootstrap code to initialize our VMM system on top of which the Android software stack will run.
- March 2008 - CPU scheduling when multiple Androids run on the VMM. Memory Abstraction and Inter virtual machine communication.
- April 2008 - Device Abstraction (mapping interrupts to events, multiplexing requests from different OSes). Checkpointing and restoring an android from a stored checkpoint.
- May 2008 – Demonstrate a fully functional, virtualized Android system.

Anticipated Results:

- We expect to incur performance overhead in our initial efforts to virtualize the Android system on the ARM ISA. Future efforts can be dedicated to performance enhancements.
- We expect to be able to boot a QEMU-based VMM, and from that VMM, boot the Linux OS of the Android system.
- We hope to be able to boot a second instance of the Linux OS of the Android system.
- We hope to checkpoint one version of android and restore the OS from a checkpointed version.

References and Related Work:

1. Android – Open Handset Alliance Project Web pages. <http://code.google.com/android>
2. Choices project Web Pages, Systems Research Group, Department of Computer Science, University of Illinois at Urbana-Champaign. <http://choices.cs.uiuc.edu>
3. Francis M. David and Jeffrey C. Carlyle and Ellick M. Chan and Roy H. Campbell, Porting Choices to ARM based platforms, Technical Report UIUCDCS-R-2007-2830, Department of Computer Science, University of Illinois at Urbana-Champaign, March, 2007. <http://choices.cs.uiuc.edu/choices-arm.pdf>
4. Philip A. Reames, Ellick M. Chan, Francis M. David, Jeffrey C. Caryle, and Roy H. Campbell, A Hypervisor for Embedded Computing, In Illinois Journal of Undergraduate Research, April, 2007. http://academic.ec.uiuc.edu/ijur/article3_2007.pdf
5. Rishi Bhardwaj and Phillip Reames and Russell Greenspan and Vijay Srinivas Nori and Ercan Ucan, A Choices Hypervisor on the ARM architecture, CS523 Course Project Report, Department of Computer Science, University of Illinois at Urbana-Champaign, 2006. <http://choices.cs.uiuc.edu/ChoicesHypervisor.pdf>
6. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the art of virtualization. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles (Bolton Landing, NY, USA, October 19 - 22, 2003). SOSP '03. ACM, New York, NY, 164-177. DOI=<http://doi.acm.org/10.1145/945445.945462>
7. Rosenblum, M. and Garfinkel, T. 2005. Virtual Machine Monitors: Current Technology and Future Trends. Computer 38, 5 (May. 2005), 39-47. DOI=<http://dx.doi.org/10.1109/MC.2005.176>
8. When virtual is better than real, Peter M. Chen, Brian D. Noble, Proceedings of the 2001 Workshop on Hot Topics in Operating Systems, May 2001.
9. Intel Virtualization Technology and embedded VMMs (<http://www.intel.com/technology/itj/2006/v10i3/5-communications/1-abstract.htm>)
10. QEMU, a fast and portable dynamic translator (http://www.usenix.org/publications/library/proceedings/usenix05/tech/freenix/full_papers/bellard/bellard_html/index.html)
11. Skyeye, integrated Simulation Environment for ARM-based processors (<http://www.skyeye.org/index.shtml>)
12. Virtualizing Embedded Linux (<http://www.embedded.com/design/opensource/205918954?pgno=1>)