Power Management Techniques in Smartphones Operating Systems

Ahmed Abdelmotalib, Zhibo Wu
School of Computer Science and Technology
Harbin Institute of Technology
China - Harbin

Abstract — Extend the battery life of mobile handsets at different levels such as operating system, wireless technologies and applications is the aim of the most hardware manufacturers and OS designers, so designing more energy efficient applications and operating systems is the best way to solve this problem, In this paper, we aim is to provide a summary of techniques employed in mobile computer and especially smartphones operating systems that can reduce the power consumption of today's mobile computing devices.

Keywords: Smartphones, Power management, Operating system.

I. INTRODUCTION

Smartphones are quickly becoming the main computing and (data) communication platform,
IDC (International Data Corporation-Worldwide Quarterly Mobile Phone Tracker) said 157.8 million smartphones sold worldwide in Q4 2011, bringing the total for the year to 491.4 million units [1].
Smartphone are as powerful as the PCs. Therefore, they are perfectly suitable to become the first real-life platforms for ubiquitous computing. For instance, they can be programmed to run location-aware applications that provide people with real-time information relevant to their current places. Deploying such devices in our daily life, however, requires a good understanding of their power requirements. Also the increasing popularity of power-hungry applications that take advantage of the smartphones resources and the diverse range of wireless interfaces and sensors can reduce the battery life of smartphones.
With this new paradigm however come new challenges in computer operating systems development, these challenges include many items such as frequent network disconnections, communications bandwidth limitations, resource restrictions, and power limitations.
In this paper we analyse general solutions for energy consumption on mobile devices and especially smartphones, concentrate on techniques can be employed in mobile computer operating systems that can reduce the power consumption of today's mobile computing devices.
The paper is organized as follows: In Section II we present Energy Management techniques in Smartphones operating systems, and finally Section III concludes the paper.

II. Energy Management in Smartphones OS

Energy efficiency is recognized as a paramount property of any mobile device, in particular for smartphones. However , applications and operating system both are responsible for energy management .At the operating system level, the main idea is to reduce energy consumption by unifying resource and energy management, however understanding of how resources are demanded by users and applications in the system is very important in order to manage energy. Previously we surveyed Power Consumption in Smartphones from hardware viewpoint [2], focused on what are typically the main culprits in terms of power consumption in Smartphones: CPU, memory devices and wireless communication and we showed that the most energy hungry parts of a mobile phone are the wireless technologies and not the display or the CPU.
This section describes the motivations behind considering energy as a fundamental resource in mobile operating systems.

1- **Smartphones Operating System's**

An operating system is a software component of a computer system that is responsible for the management of various activities of the computer and the sharing of computer resources. It hosts several applications that run on a computer and handles the operations of computer hardware. Users and application programs access the services offered by the operating systems, by means of system calls and application programming interfaces [3]. The most common mobile operating systems (OS) used by modern smartphones (as shown in Figure 1) include Apple's iOS, Google's Android, Microsoft's Windows Phone, Nokia's Symbian, RIM's BlackBerry OS, and embedded Linux distributions such as Maemo and MeeGo. Such operating systems can be installed on many different phone models, and typically each device can receive multiple OS software updates over its lifetime [1, 4].

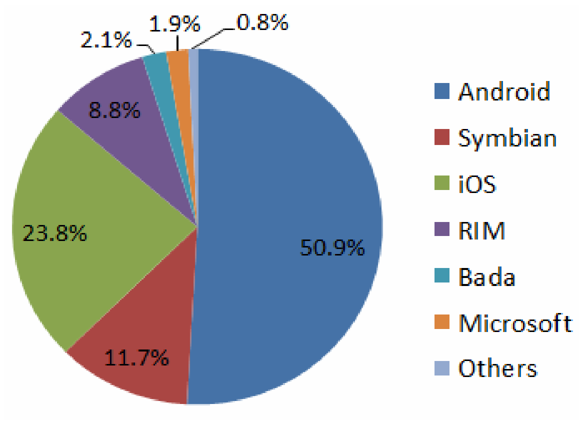

Figure 1: Smartphone operating systems

2- **energy-aware policies**

Some authors suggest that energy-aware policies in mobile devices should be performed by applications. So applications must adapt dynamically to energy limitations as in Chameleon [5] but this approach lacks of a central entity responsible for monitor all the resources consumption caused by other applications. On the other hand, other researchers suggest that resources and energy management should be entirely done at the operating system. However, this solution can present scalability problems. Others present an intermediate solution. They follow a hybrid approach [6] in which both applications and operating system collaborate to reduce the power consumption in a mobile phone. Ideally, the operating system must know applications' resource demands and the available energy resources until the next charging opportunity to reduce the power consumption while maximising user experience. However, new programming models, schedulers, energy measurement tools, and power-based APIs must be developed in order to support software-level energy management.

3- **Resource management techniques**

There are more efficient ways of managing resources from an energy-aware perspective. However, this requires an accurate knowledge of any task being performed inside the computer system (at the process or thread level) in order to be aware of the resource demands. efficient power management in mobile platforms is a complex and challenging research problem due to the multitude of possible hardware configuration options and power states. As Snowdon et al. claim in [7], power management in present mainstream operating systems tends to be simplistic and as a consequence, sub-optimal. Traditional operating systems run the workload to completion at the maximum performance setting and then they shift into a low-power mode (or to the lowest performance setting) to achieve energy savings.

Hardware resource monitors have proved to offer valuable information in the field of performance analysis.

The power management implementation is responsible for, on behalf of the operating system [8]:

1- Controlling the state and the power requirements of the hardware.

2- Extending the useful life of the battery component and the period the device can be used in between recharges.

3- Managing the user's perception of the phone operational state.

From this, it is clear that power management must be implemented at all levels of the operating system.for example:

1- There may be a UI-specific policy that decides to switch the display backlight or the display itself off after a period of user inactivity

2- A client of the services provided by an input port may decide to allow the controlling device driver to move the input port hardware to a low power state after a period of inactivity (no transactions through that port).

Symbian OS favors a distributed approach to power management, with components at different layers of the OS responsible both for managing their requirements on system power, and the impact of their actions on the availability of the phone. They achieve this in co-operation with other interdependent components, which can be at any level of the OS.

Android supports its own Power Management (on top of the standard Linux Power Management) designed with the premise that the CPU shouldn't consume power if no applications or services require power. Android requires that applications and services request CPU resources with "wake locks" through the Android application framework and native Linux libraries. If there are no active wake locks, Android will shut down the CPU [9].

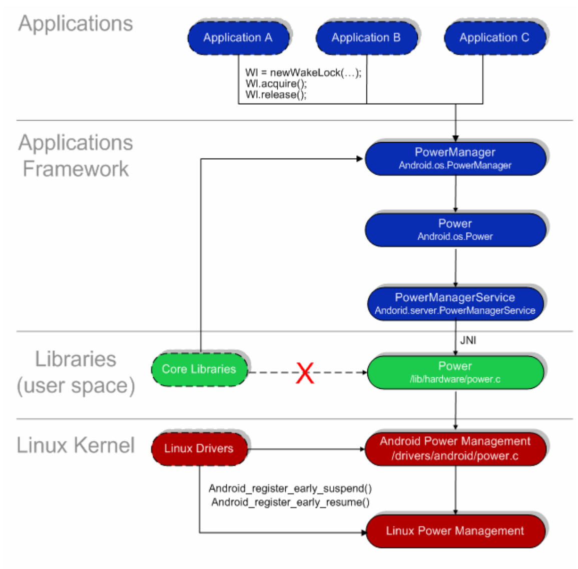

Figure 2: Android power management architecture

Odyssey OS utilizes PowerScope [10] to map energy consumption to program structure in a similar fashion to CPU profilers such as prof command in Unix machines. In other words, it maps energy to program structure at the procedure level so it can help to identify applications behaving as energy sinks. PowerScope combines both online and offline techniques to profile applications. It requires an external power meter and a second computer to reduce the energy overhead and any possible interference at the profile stage, using statistical sampling to collect traces. As it happens with energy models obtained using off-line methods, this approach is not scalable since it requires repeating the off-line training for every single hardware configuration and machine.
Nevertheless, this kind of tool has two advantages: it allows developers to re-implement applications in order to meet the design goals and it also allows the operating system to manage resources more efficiently.
PowerScope estimates the energy use of applications by measuring the time spent in each state and by counting the number of state transitions. The authors claim that, in addition to mapping energy costs to specific processes, it is necessary to identify thread-level activities. For instance, a task which blocks frequently (e.g. sockets) may expend larger amounts of power on other resources such as the screen, disk, and network when the processor is idle. An evolution of PowerScope is described in [11]. In this case, the authors introduce a predictive system to manage resources proactively in order to support multi-fidelity computation. The resources monitoring tool takes advantage of simple machine learning techniques so the resources manager can process larger number of adaptations with much more accuracy in a single step. Joule Watcher [12] is a fine-grained and event-driven energy profiler that also accounts the energy consumption at the thread-level. In this case, they use counters embedded in the target hardware drivers to register events that imply the consumption of energy by monitoring CPU, battery and memory. Those parameters are accessed in the thread and from the system context so it makes possible estimating the energy consumption of individual threads and the whole system. Obviously, because of the limited number of resources it monitors, its real integration on a modern energy-aware OS can be compromised.
Regarding resource management, earlier works described that energy-efficiency must be performed on the application side. Nevertheless, applications are usually responsible for performing transformations to increase the possible energy savings while the management policies are implemented in the operating system. Heath et al. described how application transformations can increase device idle times by informing the OS about the length of an upcoming period of idleness [13].
Self-tuning power manager (STPM) [14] is an energy-aware resource management middleware framework focused on I/O devices (i.e. wireless interfaces) which leverages collaboration between applications and system and also provides an energyaware caching system. The idea is that applications disclose ghost hints about their intentions of using any of the I/O devices while the system exposes context information about the general state of the system to applications. With this knowledge, STPM adapts the power management strategy to the observed pattern from the applications. The author claims that STPM makes better power management decisions because it tries to achieve the best combination of performance and energy conservation given a specific application workload. In theory, this approach does not limit which requests are serviced by hardware devices to applications as it might happen in ECOSystem [15]. Likewise, CIST [16] shows that it is

IJCSI
www.IJCSI.org

possible to achieve higher energy savings by considering the needs of the applications rather than analyzing the behaviour of resource.

III. CONCLUSION

Need of considering energy as a fundamental system resource in mobile devices is now the main researchers emphasizing. Researching on power consumption in smartphones has showed that this is a very hot topic nowadays, mostly because of the demand for longer stand-by times, combined with the demand for better performance. It clearly shows that a lot of research is being done, by several types of both dependent and independent organizations. Combined with the bombarding growth of not only the smartphone market, but also the bigger need for portability in general, the development of longer lasting batteries and low power consumption components is undoubtedly at its highest peak.
To this end we cover studies that analyse and compare general solutions for Power Consumption in Smartphones, we have seen that efficient operating system techniques can significantly reduce power consumption without considerably affecting the perceived performance.

REFERENCES

[1] http://www.email-marketing-reports.com /wireless-mobile/smartphone statistics.htm #smartphones
[2] A. Abdelmotalib and Z. Wu, "Power Consumption in Smartphones (Hardware Behaviourism)", International Journal of Computer Science Issues (IJCSI) 2012
[3]http://www.buzzle.com/articles/different-types-of-operating-systems.html
[4]http://en.wikipedia.org/wiki/Smartphone
[5] X. Liu, P. Shenoy, and M. Corner, "Chameleon: application level power management with performance isolation," in Proceedings of the 13th annual ACM international conference on Multimedia, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 839– 848.
[6] N. Vallina-Rodriguez and J. Crowcroft, "*Energy Management Techniques in Modern Mobile Handsets*", IEEE COMMUNICATIONS SURVEYS, February 2012
[7] D. C. Snowdon, E. Le Sueur, S. M. Petters, and G. Heiser, "Koala: a platform for os-level power management," in Proceedings of the 4th ACM European conference on Computer systems, ser. EuroSys '09. New York, NY, USA: ACM, 2009, pp. 289–302.
[8]http://www.developer.nokia.com/Community/Wiki/Symbian_OS_Internals/15._Power_Management
[9]http://www.netmite.com/android/mydroid/development/pdk/docs/power_management.html
[10] J. Flinn and M. Satyanarayanan, "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications," in Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, ser. WMCSA '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 2–.
[11] D. Narayanan and M. Satyanarayanan, "Predictive Resource Management for Wearable Computing," in Proceedings of the 1st international conference on Mobile systems, applications and services, ser. MobiSys '03. New York, NY, USA: ACM, 2003, pp. 113–128.
[12] F. Bellosa, "The benefits of event: driven energy accounting in powersensitive systems," in Proceedings of the 9th workshop on ACMSIGOPS European workshop: beyond the PC: new challenges for the operating system, ser. EW 9. New York, NY, USA: ACM, 2000, pp.
37–42.
[13] T. Heath, E. Pinheiro, J. Hom, U. Kremer, and R. Bianchini, "Code transformations for energy-efficient device management," Computers, IEEE Transactions on, vol. 53, no. 8, pp. 974 – 987, aug. 2004.
[14] M. Anand, E. B. Nightingale, and J. Flinn, "Ghosts in the machine: interfaces for better power management," in Proceedings of the 2nd international conference on Mobile systems, applications, and services , ser. MobiSys '04. New York, NY, USA: ACM, 2004, pp. 23–35.
[15] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, "ECOSystem: managing energy as a first class operating system resource," in Proceedings of the 10th international conference on Architectural support for programming languages and operating systems, ser. ASPLOS-X. New York, NY, USA: ACM, 2002, pp. 123–132.
[16] A. B. Lago and I. Larizgoitia, "An application-aware approach to efficient power management in mobile devices," in Proceedings of the Fourth International ICST Conference on Communication System software and middleware (COMSWARE), ser. COMSWARE '09. New York, NY, USA: ACM, 2009, pp. 11:1–11:10.
[17] N. Pettis, L. Cai, and Y.-H. Lu, "Statistically Optimal Dynamic Power Management for Streaming Data," IEEE Trans. Comput., vol. 55, pp. 800–814, July 2006.
[18] F. Fitzek, S. Rein, M. Perucci, T. Schneider, and C. Guhmann, "Low complex and power efficient text compressor for cellular and sensor networks," in 15th IST Mobile and Wireless Communication Summit,2006.