Energy Consumption In Mobile Computing

Ricardo Isidro Ramírez
SEPI - ESCOM - IPN
ricardo.isidro.86@gmail.com

Erika Hernández Rubio
SEPI - ESCOM, IPN
ehernandezru@ipn.mx

Amilcar Meneses Viveros
CINVESTAV - IPN
ameneses@cs.cinvestav.mx

Abstract—**Mobile devices users have risen in recent years, by convergence of functionality and technology on it. Mobile devices also appear in market as interactive devices where users expects short-time response, usability and functionality. That convergence of digital entertainment, mobile communication technologies, and a set of new capabilities, becomes requirements for mobile application developers and hardware architects, thus both professionals have the dilemma of offer low power consumption and high performance on it. However, both concepts are opposite each other. That means, if I get a great performance, probably have attached high power consumption. Programmers have found in parallel computing, multithreading programming, and process scheduling, the way to achieve more efficiency in their programs. Moreover, hardware architects have found in multicore processors the way to achieve better performance and low-power consumption. This work wants to join both solutions and check if we can achieve less response time in applications and processes, and get a balance between performance and energy consumption.**

Keywords—Energy Consumption; Mobile Computing; Parallel Computing; Multithreading; Systems-On-Chip.

I. Introduction

In last years, mobile devices has gained popularity due to portability and convergence of many technologies included on it. You can include in this classification devices like smart phones, laptops of various sizes, tablets, mainly. By nature, these devices are bounded in some aspects like memory quantity, screen size and resolution, processor speed, battery life, and so on. Observe that few memory, reduced screen and slow processor just reduce capabilities of processes and applications executed on device. Consider also that RAM memory is a resource that can be recovered just killing unused applications; processor speed executes lowly some processes, but with some patience probably the process will finish; and screen size difficulties, but no prevents, the user experience when an application is used. But a battery with few energy will finish unexpectedly with all processes and applications running on device. Hence, a limited energy source limits capacities and reduce the operation of the entire device. This scenario will be always present independently of the kind of mobile device. It's necessary to understand how mobile devices drain energy by all the components and process presents on it. Energy is defined as the capacity to realize work, where work represents every process running in devices. Energy consumption is measured using Joules (J), but many authors prefer Watts (W) to measure power needed to run determined process. Because power consumption is very important, it is necessary to have the major possible energy efficiency. Energy efficiency is defined as the reduction of energy needs for achieve an objective. Usually, power consumption enhancements comes with technological advances, but also can be result of a improvement in organization or management of resources presents in mobile devices. That advances are product of research of hardware architects and software developers. On the one hand, hardware manufacturers have implemented multicore solutions inside mobile devices in recent years, in order to deliver smaller, faster and low-power processors with better performance too. On the other hand, software developers are working in multithreading and parallel techniques that offer better performance in their applications. But it isn't clear that multithreading paradigms are the better way for all kind of applications because not all of these have the same level of parallelism. Along this paper some topics about energy consumption and performance over mobile scenario will be discussed.

II. Hardware Energical Behavior

This section will discuss about how hardware components are involved in energy consumption. In section A the paper wants to give an idea of what components are the most battery consumers. Next, in section B, some device use scenarios are explained in order to understand how the use of device defines the consumed level of energy. And finally in section C, multicore hardware is presented as a new solution for give processors with potential better performance and reduced energy consumption. Also, it's important to say that some hardware manufacturers have been working in low-power technologies in order to keep the demand of new devices with better battery duration, which is a feature very appreciated by consumers.

A. Hardware Components That Drain Energy

If you want to have the best possible energy management, you need to know how, when and where the battery is consumed. You need to break down how principal hardware components in mobile devices drain battery.

The authors of this paper would thank to National Polytechnic Institute and 20131498 PIFI project, the given sponsorship for the coming conference.

978-1-4673-6155-2/13/$31.00 © 2013 IEEE

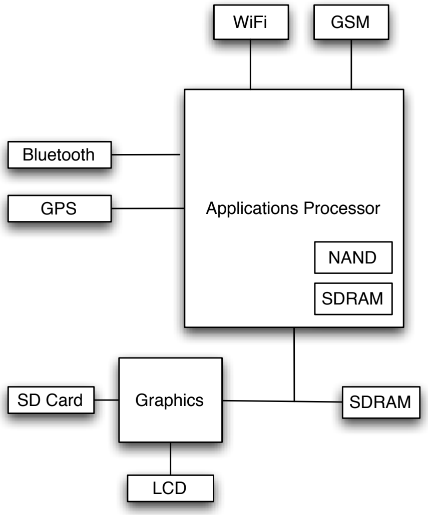

Figure 1: Important components in a mobile architecture [3].

Standard mobile devices are builded using some individual modules, just like a traditional computer. The main difference is the operating voltage supplied for each device, for example, between a large flat monitor and a reduced LCD screen. Principal hardware components that you can see in standard mobile devices are represented with their interconnections in Figure 1, where you can see: LCD, GPS module, Bluetooth, GSM and WiFi antennas, processor, memory, SDCard module, among others.

The major battery consumer of all these components is LCD screen with near of 50 - 80% of the energy consumption for a battery charge. That sound logic if you remember that screen usually is turned on, and for the touchscreen case, the screen is the preferred input/output device. The power necessary to keep screen in operation is shown in Figure 2, where you can observe that the backlight brightness level determines the power needed. Another variables that determines the power requirements for a screen is its size and the quantity of colors that screen are displaying. For example, a total white screen consumes 33.1 mW or 0.0331 J/s, and a black screen consumes 74.2 mW or 0.0742 J/s [3].

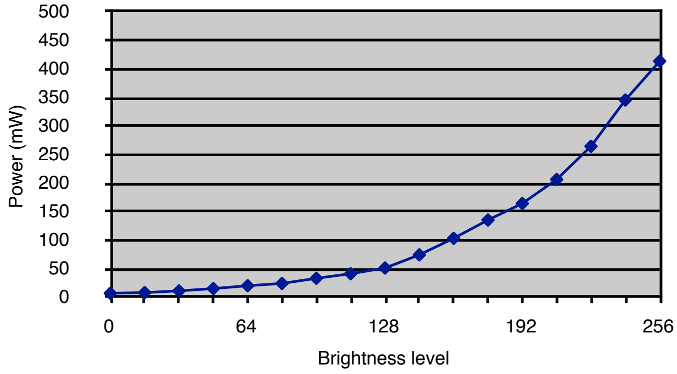

Figure 2: Display backlight power for varying brightness levels [3].

B. *Hardware Uses Stages and Power Consumption Associated*

All hardware components works together under different circumstances, depending of the goal of the user. For example if you need to take a photograph, you need to use camera, screen, and probably LED flash module. Moreover, if you want to make a call, you need to use GSM antenna, microphone, and keypad (virtual or physical), just for mention some components. All these hardware combinations can be called hardware usage scenarios. It is important consider these stages, because they are the basis for measure energy consumption. As the authors of [3] show, you need to establish a baseline state power of the device, when no applications are running. The baseline cases to consider are: mobile in suspended state and mobile in *idle* state. The difference between idle and suspended states are screen status.

A mobile is in suspended state if spends a large amount of time without interaction with user. That means some components like processor, audio, and RAM are idle but communications performs a low level of activity, and their energy consumption are low. As shown in Figure 3, it represents low use of RAM, audio (remember that this scenario considers that no one software is running in background, like music players) and graphics.

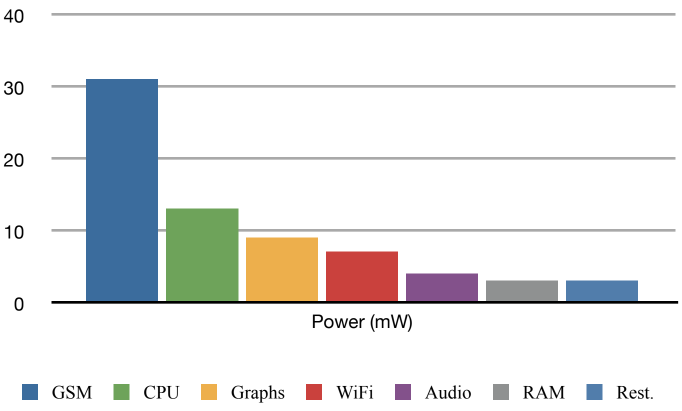

Figure 3: Power consumption given by a mobile device in suspended state [3].

So, the evident battery killer under this scenario is GSM module, that is because your mobile device only is active for communication applications, like SMS or calls. Consider that if you have programs that request information in background, the use of GSM module will grow, and maybe WiFi module could be used too. For Figure 3, the energy consumption under suspended case is 68.6 mW, or 0.0686 Joules/second.

The second scenario to review is idle state. The device is in idle state if it is fully awake, but no applications are active. Thus, screen becomes the major power consumer. The importance of consider power consumption of screen is because all graphics subsystem means around 50-80% of the total energy consumption in a standard mobile device. This fact can be seen in Figure 4 were you can note two facts: the first is GSM was relegated by graphics and LCD as the battery killer, and in second place, the total energy consumption grew to 268.8 mW, or 0.2688 Joules per second.

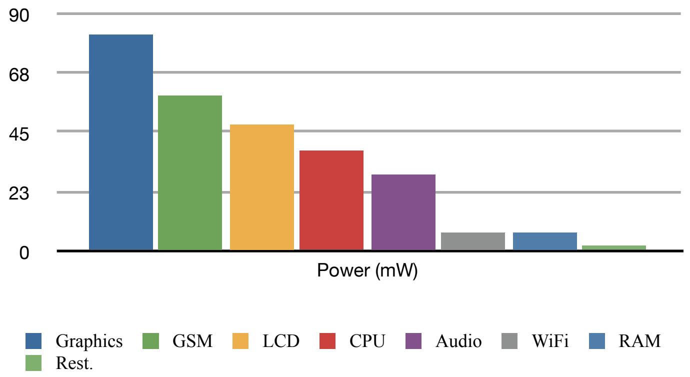

Figure 4: Power consumption given by a mobile device in idle state [3].

C. *Multicore Systems Analysis.*

Since 2005 year, the world have seen a growing trend to design multicore architectures, beginning by ARM, who announced in October 4th, 2005, a complete ARM Cortex-A8 processing solution, the first applications processor based on ARMv7 architecture that offered greater performance, energy efficiency and code density [1]. This way has been followed by other manufacturers like Intel that announced a few days, the 4th. generation of processors family Intel Core that will offer faster, finest, cooler, and safer systems, and will be used in new innovators designs of laptops like *Ultrabook* and tablets [2]. That new architectures options in market lets work with parallel algorithms or multithreading programs. But, that fact it's no necessary covering all software side, as you can see later.

Previous the appearance of these architectures, System-On-Chip designers used to follow another methods in the name of achieve short-time responses in their processors. According to [6], since 2005, the trend goes to the increase number of cores in the new designs of processors. Before 2005, processor manufacturers kept acceptable performance increments with only clock speed increments. However, that designs do not sustain the required performance for the growing market of processors. That increments in frequency achieve a 3 GHz frequency bound, so designers noticed that no more performance could be obtained and chips were heated. Hence, with adoption of multicore designs, the manufacturers can offer again solutions with increasing performance.

In multicore architectures, you can find two different architecture classifications for multiprocessors systems on chip (MPSoC), named homogeneous and heterogeneous as [7] explains and Figure 5 illustrates. An homogeneous architecture consist in a replication for the same core into a single die. As you can see in Figure 5, the MPSoC #1 have repeated the same processing element, and each processing element have the same processing capability, area, and energy consumption. An heterogeneous architecture demonstrates different processing elements despite having the same instruction set architecture (ISA), like you can see in Figure 5, MPSoC #2. This architecture style is widely used in the mobile domain, due to necessity for high performance in combination with energy efficiency. The fact that have or don't have different processing elements give the chance of use task scheduling approaches or not.

Multicore processors designs have forced to have a reference model to study the behavior of the power consumption. The base of this model is an extension of the Amdahl's Law [10][11][13][14][15]. The Amdahl's work [13] establish a parameter to predict speedup achievable for a potential parallelizable program. This prediction is based on the knowledge of the inherently sequential portion of the computation *f* in *n* processors, represented in (1) as follows:

$$Speedup(f,n) = \frac{1}{((1-f)+(f/n))} \,. \quad (1)$$

This law establish the upper bound of achievable speedup of parallel program. Work done in the extension of the Amdahl's Law is redefined the *f* term as the energy parameter of the sequential portion. Variations over this extension consider symmetric and asymmetric processors.

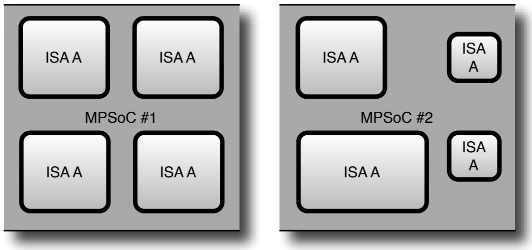

Figure 5: MPSoC #1 represents an homogeneous processor architecture, and MPSoC #2 represents an heterogeneous processor architecture [7].

III. Performance Features and Energy Consumption under Software approach.

Having examined the importance of consider the hardware layer in a energy consumption study, now let's continue with software approach. Basically, it exists two techniques that works with the mission of enhance performance in application execution: implement parallel programming and change scheduler policies. Section A talks about how parallel programming can enhance performance of certain applications, and Section B talks about how scheduling can contribute to speed and distribute tasks. Finally, section C presents an study made by [4] that shows the impact of web programming technologies in energy consumption while applications are surfing the web.

A. *Application focused energy consumption study.*

Also it is necessary to take in mind that not only the hardware will be the unique energy consumer on mobile devices. Techniques used to develop applications also contribute to stress the capacities of mobile phones by excessive use of various subsystems. One of the most common mistakes in the design of mobile programs is the excess of requests made by the applications itself. The authors of [4] presents the result of an analysis focused on web navigation through mobile phones, and they make some recommendations in order to save energy.

Based on the paper [4] we can highlight aspects like:

- Size and compression of images used in web pages, because some web page designers don't consider the difference between JPEG, PNG and GIF formats. Also, remember that mobile devices uses to have small screens and needs small interfaces. So the cost of render a page resizing the images attached, could be higher, as you can see in Figure 6.

- Implementation of Cascade Style Sheets in mobile environment must be omitted or reduced because the presence of a size reduced user interface don't let enjoy of a colorful web page.
- Use of Javascript embedded code in order to give dynamic appearance to web pages. Similar to use of CSS

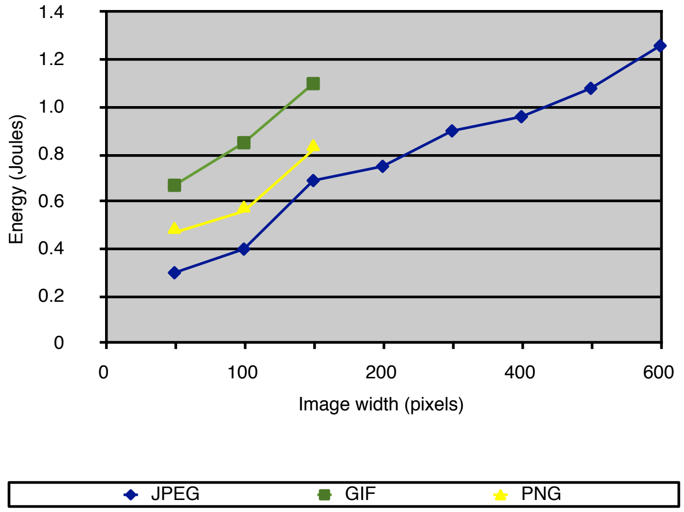

Figure 6: Comparative graphic between energy and width for the most used image formats in web pages [4].

The author of [4] recommends follow some rules if you are design a site specially designed for mobile browsers. Some of these recommendations are:

- Offer an special designed site for mobiles. Usually that sites have the subdomain .m or .mobile, just to name a few. Example of this practice is Gmail site. If you are going to access Gmail from your laptop, you are going to be redirected to http://www.gmail.com. But if you are going to access from you mobile phone or tablet, probably you are going to be redirected to http://m.gmail.com as a special domain only for reduced devices.
- Reduce as less as possible the use of Javascript and CSS if you page are going to be visualized with mobile devices. Also is recommended to analyze and optimize the use of that technologies in order to reduce code lines into scripts.
- Prefer the use of JPEG images over PNG, and mainly don't use GIF images. As you can see in Figure 6, the energy cost for render a 150 pixel PNG image is the same for render a 250 pixel JPEG image.

B. Schedulers

In computing, it exists situations when operating systems must decide among various process, which process will run first. The operating system component that check this is called scheduler, and the algorithm that scheduler utilize is called scheduling algorithm [8].

The use of adaptive CPU schedules that predict computing needs in mobile devices for specific tasks like graphic rendering, video reproduction, and so on, are a real requirement according to [9] and [10]. The existence of complex applications wants dynamic response to variations in workload and resource availability. For this reason, static solutions are difficult to apply in real world scenarios. As authors of [9] explain, adaptive CPU scheduling policies should address the needs of CPU scheduling and reservation for current multimedia applications.

An special task scheduling could be employed too if we are working under heterogeneous multiprocessor architectures, topic treated in previous chapter. Figure 5 illustrate both multicore approaches. On one hand, MPSoC #1 supports only naive task scheduling approaches, since all cores have equal features like energy consumption, capability, and processing, which disallow the use of special task scheduling. On the other hand, MPSoC #2 demonstrates different processing elements despite having the same instruction set architecture. For this reason, the second MPSoC provides flexibility on task scheduling. That's because if we have different tasks with different performance level requirements, MPSoC #1 approach just ignore the requirements and execute all tasks into their cores, causing a waste of energy. In contrast, MPSoC #2 has different cores with different capabilities. Hence, an efficient scheduling rule can distribute conveniently all work among all available cores. So, the result of join multicore heterogeneous architectures and scheduling algorithms, is a better use of available computation elements that give better performance and less energy consumption.

C. Parallel programming

According to [5], parallel computing is the use of a parallel computer to reduce the time needed to solve a single computational problem. That problem can include many knowledge areas like information technologies, molecular physics, climate, just to name a few. A parallel computer is a multiple-processor computer system supporting parallel programming. And finally, parallel programming is programming in a language that allows you to explicitly indicate how different portions of the computation can be executed concurrently by different processors.

Parallel programming is an important alternative in mobile computing because, as you seen in previous section, various mobile devices manufacturers have been implementing multicore processors, in order to supply more computing resources for new applications that aren't actually taking advantage of that fact. Not all applications can be benefited with parallelism approach. It exists a well defined class of applications called Digital Signal Processing or DSP, that could be parallelized thanks to their loop-based behavior [7]. Fast Fourier Transform, and Infinite or Finite Impulse Response filters are classical examples of DSP, usually applied in audio and video processing applications, just for mention some examples.

Models for power consumption based in process are beginning their study [14][15][17]. The main idea of our proposed model is generate an extension for Amdahl's Law, with an energy consumption approach. Thus, this model consider the power consumption of a whole computation of a problem of size *n*, *E(n)* is the sum of the sequential part $E\sigma(n)$ and the potential parallel part $E\rho(n)$:

$$E(n) = E_{\sigma}(n) + E_{\rho}(n) \text{ . (2)}$$

Now, in order to model energy consumption of a parallel program running on *m* processors, we have energy consumption for a *n*-size problem with *m* cores, *E(n,m)*, expressed as:

$$E(n,m) = E_{\sigma}(n) + \delta E_{\rho}(n) + E_{\kappa}(n,m) \text{ , (3)}$$

where the third adding on (3) is the power consumption required for parallel overhead, and the value of δ changes according to the energy behavior for the multicore processor. The equivalent of δ could be the inverse of *m* if processor keeps energy consumption as a constant, independently of the number of cores inside it; or *m* if processor cores only spends energy when they are active.

So, energy speedup is represented with $\Psi(n,m)$ and it is the division of *E(n)* by *E(n,m)*. And also if we consider:

$$f_c = \frac{E_{\sigma}(n)}{E_{\sigma}(n) + E_{\rho}(n)} \text{ , (4)}$$

we can write the energy speedup of a parallel program running on *n* processors as (4):

$$\Psi(n,m) \le \frac{1}{f_c + (1 - f_c)\delta} \text{ . (5)}$$

Hence, a multicore with capability of turn off their processing elements is being described. We can notice that the model predicts a reduction of power consumption if the parallel program runs in several processors.

On the one hand, Figure 7 shows the energy speedup when processor follows an architecture with core deactivation, that illustrate a negative speedup and increasing the use of energy. In that case, the expressions for consumption and speedup can be found in Table 1.

TABLE I. FORMULAS FOR ARCHITECTURE WITH CORE DEACTIVATION.

δ *value*	$E(n,m)$	$\psi(n,m)$
ρ	$E_{\sigma}(n) + mE_{\rho}(n) + E_{k}(n,m)$	$\Psi(n,m) \le \frac{1}{f + (1-f)m}$

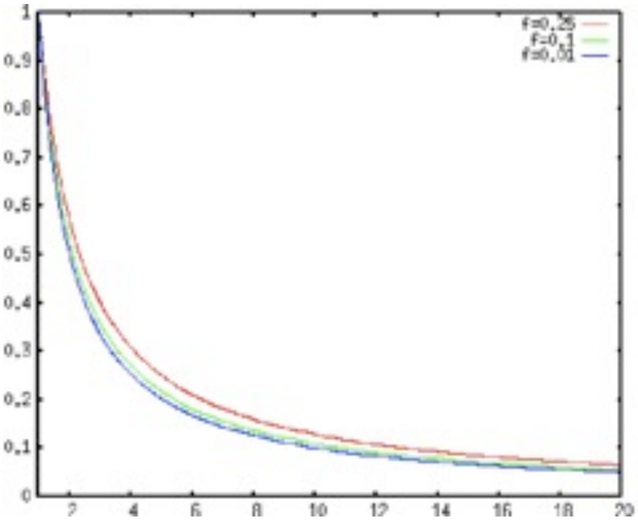

Figure 7: Energy speedup predicted for several *f* values.

On the other hand, Figure 8 shows the energy behavior when processors follows an architecture with a constant energy consumption independently of the number of cores, and a speedup that means gain in energy savings. In that scenario, the expressions can be found in Table 2:

TABLE II. FORMULAS FOR MULTICORE ARCHITECTURE WITH CONSTANT ENERGY CONSUMPTION.

δ *value*	$E(n,m)$	$\psi(n,m)$
$\frac{1}{\rho}$	$E_{\sigma}(n) + \frac{E_{\rho}(n)}{m} + E_{\kappa}(n)$	$\Psi(n,m) \le \frac{1}{f + \frac{(1-f)}{m}}$

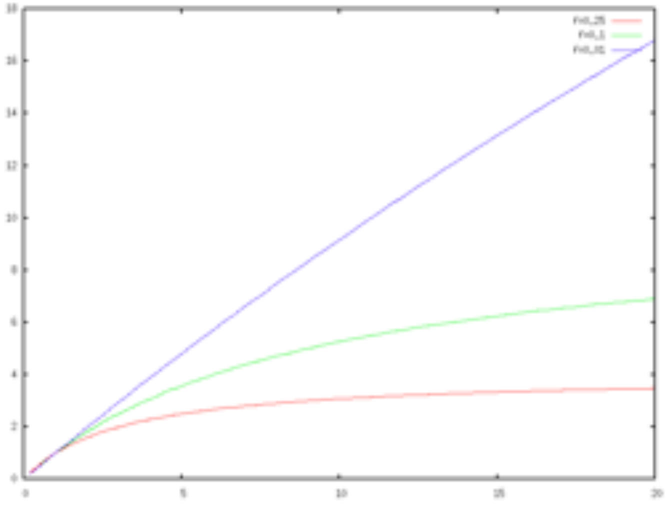

Figure 8: Energy speedup predicted for several f values.

Unfortunately, modern multicore processors operate in two different power levels. If a core is being used by a process, that core needs a distinct power level that another core in *idle* state. Hence, two different voltages could feed each core in order to reduce final energy consumption level. That operation case is not considered because more measurement and investigation is required here.

IV. Conclusion

It is necessary to establish an energy model software oriented in more detail, in order to get tools to software developers for optimize their projects for the actual scenario of multicore systems on a chip, and for predict if addition of parallel algorithms and multithreading programs can supply reductions in energy consumption. And hardware architects could be benefited with this work too because they could consider how energy is consumed by system processes. This paper showed that the major part of software studies are related only with achieve the best performance as possible. And the major part of hardware studies are focused on increment energy efficiency. The model for power consumption process-based require experimental results to be validate. Finally this model gives hope for a major adoption of use of parallel and multithreading programs in mobile devices.

References

[1] ARM. Arm introduces industry's fastest processor for low-power mobile and consumer applications. http://www.arm.com/about/newsroom/10548.php. (2005).

[2] BinHex 3.0. La 4ta. generación de procesadores Intel Core definirá el futuro del cómputo móvil en 2013. http://binhex3.com/?p=2913. (2012).

[3] Carroll, Aaron; Heiser Gernot; An analysis of power consumption in a smartphone, *Proceedings of the 2010 USENIX conference on USENIX Annual Technical Conference*, (2010).

[4] Thiagarajan, Naredran; Aggarwal, Gaurav et. al., Who killed my battery: analysing mobile browser energy consumption, *World Wide Web Conference 2012*, 41-50 p.p. (2012).

[5] Quinn, Michael J, Parallel Programming in C with MPI and OpenMPI, 1sr. Edition, Mc. Graw Hill, 529 p.p. (2004).

[6] Fuller, Samuel H; Millett, Lynette, Computing performance, game over or next level? *The National Academies Press*, Washington, D.C. 31-38 p.p. (2011).

[7] Carro, Luigi; Rutzig, Mateus Beck, Multi-core Systems on Chip, *Handbook of Signal Processing Systems*, 485-514 p.p. (2010).

[8] Tannenbaum, Andrew S., Woodhull, Albert S., Sistemas operativos: Diseño e Implementación, 2nd. Edition, Prentice Hall, 939 p.p. (1997).

[9] Wu, Fan; Agu, Emmanuel; Lindsay, Clifford; Adaptive CPU Scheduling to Conserve Energy in Real-Time mobile Graphics Applications, *Advances in Visual Computing*. *Lectures Notes in Computer Science*, Springer, 624-633 p.p. (2008).

[10] Moreno Londoño, Sebastián; Pineda de Gyvez, José; Extending Amdahl's Law for Energy-Efficience, *Energy Aware Computing (ICEAC), 2010 International Conference,* 1-4 p.p. (2010).

[11] Xian-He Sun; Yong Chen; Reevaluating Amdahl's Law in the multicore era, *J. Parallel Distributed Computing February 2010*, 188-210 p.p. (2010).

[12] Herlihy, Maurice; Shavit, Nir, The Art of Multiprocessor Programming, Ed. Morgan Kaufmann, 528 p.p. (2008)

[13] G. M. Amdahl, Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities; *Proc. Am. Federation of Information Processing Societies Conference; AFIPS Press,* 483-485 p.p. (1967)

[14] Mark D. Hill & Michael R. Mary, Amdahl´s Law in the Multicore Era, *Computer, IEEE,* (July 2008)

[15] Dong Hyuk & Hsien-Hsin S. Lee, Extending Amdahl´s Law for Energy-Efficient Computing in the Many-Core Era, *Computer, IEEE,* (December 2008)

[16] James Donald, Margaret Martonosi; Power efficiency for variation-tolerant multicore processors; *In Proceedings of the 2006 international symposium on Low power electronics and design (ISLPED '06). ACM, New York, NY, USA*, 304-309 p.p. (2006)

[17] Sangyeun Cho, Mani G. Melhem; On the interplay of Parallelization, Program Performance, and Energy Consumption; I*EEE Transactions On Parallel and Distributed Systems, Vol 21, No. 3*, (March 2010)

978-1-4673-6155-2/13/$31.00 © 2013 IEEE