Profiling Power Consumption on Desktop Computer Systems*

Giuseppe Procaccianti Antonio Vetro' Luca Ardito
Maurizio Morisio
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino,Italy
name.surname@polito.it

Abstract

Background. Energy awareness in the ICT has become an important issue: ICT is both a key player in energy efficiency, and a power drainer. Focusing on software, recent work suggested the existence of a relationship between power consumption, software configuration and usage patterns in computer systems.
Aim. The aim of this work was collecting and analysing power consumption data of a general-purpose computer system, simulating common usage scenarios, in order to extract a power consumption profile for each scenario.
Methods. We selected a desktop system running Windows XP as a test machine. Meanwhile, we developed 11 usage scenarios, classified by their functionality, and automated by a GUI testing tool. Then, we conducted several test runs of the scenarios, collecting power consumption data by means of a power meter.
Results. Our analysis resulted in an estimation of a power consumption value for each scenario and software application used, obtaining that each single scenario introduced an overhead from 2 to 11 Watts, corresponding to an increase of about 12%.
Conclusions. We determined that software and its usage patterns impacts consistently on the power consumption of computer systems. Further work will be devoted to evaluate how power consumption is affected by the usage of specific system resources, like processors, disks, memory etc.
Keywords: Green Software, Energy Aware, Energy Profiling, Power Consumption.

1 Introduction

Energy efficiency is finally becoming a mainstream goal in a limited world where consumption of resources cannot grow forever. ICT is both a key player in energy efficiency, and a power drainer. The Climate Group reported that the total footprint of the ICT sector was 830 MtCO2e and that the ICT was responsible for 2% of global carbon emissions [4]. Even if the efficient technology was developed and implemented, this figure will still grow up at 6% each year until 2020. Recently, much of the attention in green IT discussions focuses on data centers. However, it is foreseen [4] that data centers will only add up to less than 20 percent of the total emissions of ICT in 2020. The majority (57 percent) will come from PCs, peripherals, and printers, as shown in Figure 1 [4]. This is because of the enormous number of machines used by individuals and businesses: it is estimated there will be 4 billion PCs in the world by 2020. So the vast number of PCs is going to dominate ICT energy consumption. It is essential to have precise figures of the current energy consumption of computer systems and ICT equipment to understand how to reduce their power consumption to design future energy efficient equipment. Today these figures are incomplete and not precise.

Considering each IT device, it has its own energy consumption which can range from 0, when it is turned off, to X when all its internal components are used simultaneously. Through the

*Post-Print draft published in Lecture Notes in Computer Science, Springer, Vol. 6868, pp. 110-123, ISSN 0302-9743

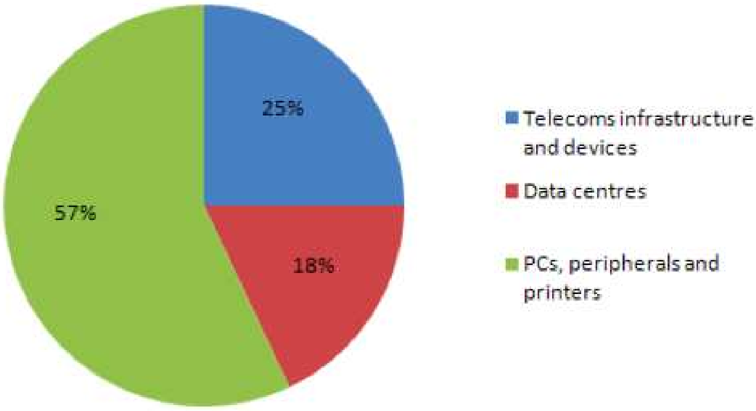

Figure 1: The 2020 global footprint by subsector

management of each part there is a variation Δ x of its consumption that is between 0 and X. The management of system components can be done either in hardware or software. When we buy a device and it is not programmable, we can not do anything to limit its energy consumption. The designers have already made choices in terms of selection of components and in terms of resource management. On the other hand, if a system can be programmed, choices made by developers will affect the management of energy the device consumes. Looking at embedded systems, all the responsibilities in terms of management of energy resources are dependent on the hardware management and on the firmware. Firmware optimizations have immediate effects that can be verified directly by measuring the current the device consumes. If we consider a general purpose device, the hardware and the operating system have an important role in global energy management, but it is not the only one. On this type of device is it possible to install a multitude of programs that will impact on the management of energy resources. For example, if a third party software uses a particular peripheral incorrectly, it could increase its energy demand even when not needed.

Motivated by this thought, we decided to run benchmarks on a common machine, simulating some typical scenarios and then measuring the energy consumption in order to make some statistical analysis on results. *A computer benchmark is typically a computer program that performs a strictly defined set of operations (a workload) and returns some form of result (a metric) describing how the tested computer performed.* [7]

In our benchmark the *workload* is a set of usage scenarios and the *metric* is the power consumption: we describe them in section 3, after a the description of the related work in section 2. Section 4 shows results, section 5 provides a discussion of the results and, finally, section 6 presents conclusions and future works.

2 Related Work

Recently a new post appeared on the MSDN Blog [8]: it concerns the energy consumption measurement of internet browsers. Authors measure power consumption and battery life of a common laptop across six scenarios and different browsers. They allow each scenario to run for 7 minutes and look at the average power consumption over that duration. The different scenarios were: Browsers navigated to about:blank (power consumption of the browser UI), loading a popular news Web sites (common HTML4 scenario), running the HTML5 Galactic experience (representative of graphical HTML5 scenario) and fish swimming around the FishIE Tank (what test is complete without FishIE). The baseline for scenarios comparison was the Windows 7 without any browsers running. Authors ran *IE9*, *Firefox*, *Opera* and *Safari* for each scenario and then they made a comparison of the obtained results. They executed the same operations with the different browsers, obtaining very different results on power consumption and laptop battery life. Hence, software can impact on energy consumption, as we also found in our previous work [11], where we monitored three servers for a whole year, observing that one of them consumed up to 75% more when used for graphical operations.

Kansal et al. [5] presented a solution for VM power metering. Since measuring the power consumption of a Virtual Machine is very hard and not always possible, authors built power models to get power consumption at runtime. This approach was designed to operate with low runtime overhead. It also adapts to changes in workload characteristics and hardware configuration. Results showed 8% to 12% of additional savings in virtualized data centers. Another related work is PowerScope [3]: this tool uses statistical sampling to profile the energy usage of a computer system. Profiles are created both during the data collection stage and during the analysis stage. During the first stage, the tool samples both the power consumption and the system activity of the profiling computer and then generates an energy profile from this data without profiling overhead. During data collection, authors use a digital multimeter to sample the current drawn by the profiling computer through its external power input. After that, they modified Odyssey platform for mobile computing. When there is a mismatch between predicted demand and available energy, Odyssey notifies applications to adapt. This is one of the first examples of *Energy-Aware* software. Yet in 1995 people are beginning to profile the energy performance of a computer. Lorch [6] in his M.S. thesis explained that there are two aspects to consider while measuring the breakdown of power consumption on a portable computer: I) Measuring how much power is consumed by each component, II) Profiling how often each component is in each state.

Other works about profiling and measuring energy consumption are related to embedded systems. For instance, JouleTrack [9] runs each instruction or short sequences of instruction in a loop and measure the current/power consumption. The user can upload his C source code to a Web Server which compiles, links and executes it on an ARM simulator. Program outputs, assembly listing and the run-time statistics (like execution time, cycle counts etc.) are then available and passed as parameters to an engine which estimates the energy consumed and produces graphs of different energy variables. Results showed that the error of predictions was between 2% and 6%.

3 Study Design

3.1 Goal Description and Research Questions

The aim of our research is to assess the impact of software and its usage on power consumption in computer systems. We define our goal through the Goal-Question-Metric (GQM) approach. [10]. This approach, applied to our experiment, lead to the definition of the model presented in Table 1. The first research question investigates whether and how much software impacts power consumption. We will test different applications and usage patterns. The second research question investigates whether a categorization of usage scenarios with respect to functionality is also valid for power consumption figures.

3.2 Variable selection

In order to answer our Research Questions, we planned and conducted a series of experiments, aimed at profiling how much power a computer system consumes when performing daily activities for a common user. We selected as independent variables the following 11 usage scenarios.

0 - Idle. This scenario aims at evaluating power consumption during idle states of the system. In order to avoid variations during the runs, most of OS'automatic services were disabled (i.e. Automatic Updates, Screen Saver, Anti-virus and such).

1 - Web Navigation. This scenario depicts one of the most common activities for a basic user - Web Navigation. During the simulation, the system user starts a web browser, inputs the URL of a web page and follows a determined navigation path. We chose Google Chrome as the browser for this scenario because of its better performance on the test system, which allowed us to increase navigation time. The website chosen for this scenario is the homepage of our research

Table 1: The GQM Model

Goal	*Evaluate*	software usage
	for the purpose	of assessing its energetic impact
	with respect to	power consumption
	from the viewpoint	of the System User
	in the context of	Desktop applications
Question 1	Does software impact power consumption?	
Metric	Consumed Power (Watts)	
Question 2	Can we introduce a classification for software usage scenarios basing upon power consumption?	
Metric	Consumed Power (Watts)	

group `http://softeng.polito.it`, so that we could maintain the same contents and navigation path during all the scenario runs.

2 - E-Mail. This scenario simulates sending and receiving E-Mails. For this scenario's purpose, a dedicated E-Mail account has been created in order to send and receive always the same message. In this scenario, the system user opens an E-Mail Client, writes a short message, sends it to himself, then starts checking for new messages by pushing on the send/receive button. Once the message has been received, the user reads it (the reading activity has been simulated with an idle period), then deletes the messages and starts over.

3 - Productivity Suite. This scenario evaluates power consumption during the usage of highly-interactive applications, such as office suites. For this scenario, we chose Microsoft Word 2007, the most used Word Processor application. During the scenario execution, the system user launches the application and creates a new document, filling it with content and applying several text editing/formatting functions, such as enlarge/shrink Font dimension, Bold, Italics, Underlined, Character and background colors, Text alignment and interline, lists. Then the document is saved on the machine's hard drive. For each execution a new file is produced, thus the old file gets deleted at the end of the scenario.

4 - Data Transfer (Disk). This scenario evaluates power consumption during operations that involve the File System, and in particular the displacement of a file over different positions of the hard drive, which is a very common operation. For this scenario's purpose, we prepared a data file of a relevant size (almost 2 GB) in order to match the file transfer time with our prefixed scenario duration (5 minutes). The scenario structure is as follows: the system user opens an Explorer window, selects the file and moves it to another location. It waits for file transfer to end, then closes Explorer and exits.

5 - Data Transfer (USB). As using portable data storage devices has become a very common practice, we developed this scenario to evaluate power consumption during a file transfer from the system hard drive to an USB Memory Device. This scenario is very similar to the previous one, exception given for the file size (which is slightly lower, near 1.8 GB) and the file destination, which is the logical drive of the USB Device.

6 - Image Browsing/Presentation. This scenario evaluates power consumption during another common usage pattern, which is a full-screen slide-show of medium-size images, which can simulate a presentation as well as browsing through a series of images. In this scenario, the system user opens a PDF File composed of several images, using the Acrobat Reader application. It sets the Full-Screen visualization, then manually switches through the images every 5 seconds (thus simulating a presentation for an audience).

7 - Skype Call (Video Disabled). For an average user, the Internet is without any doubt the most common resource accessed via a Computer System. Moreover, as broadband technologies become always more available, we thought it would be reductive not to consider usage scenarios that make a more intensive use of the Internet than Web Navigation and E-Mails. Thus, we developed the Skype scenario, which is the most used application for Video Calls and Video Conferences among private users. For this scenario's purposes, a Test Skype Account was created, and the Skype Application was deployed on the test machine. Then, for each run, a test call is made to another machine (which is a laptop situated in the same laboratory) for 5 minutes, which is our prefixed duration.

8 - Skype Call (Video Enabled). This scenario is similar to scenario 7, but the Video Camera is enabled during the call. This allows us to evaluate the impact of the Video Data Stream both on power consumption and on system resources.

9 - Multimedia Playback (Audio). This scenario aims to evaluate power consumption during the reproduction of an Audio content. For this scenario's purpose, we selected an MP3 file, with a length of 5 minutes, to reproduce through a common multimedia player. We chose Windows Media Player as it is the default player in Microsoft systems, and thus one of the most diffused.

10 - Multimedia Playback (Video). Same as above, but in this case the subject for reproduction is a Video File in AVI format, same duration.

11 - Peer-to-Peer. As for the Skype scenarios, we decided to take into account also a Peer-to-Peer scenario, which we believe is a very common practice among private users. For this scenario, we selected BitTorrent as a Peer-to-Peer application, because of its large diffusion and less-variant usage pattern if compared to other Peer-to-Peer networks with more complex architectures. During this scenario, the system user starts the BitTorrent client, opens a previously provided .torrent archive, related to an Ubuntu distribution, and starts the download, which proceeds for 5 minutes. After every execution, the partially downloaded file is deleted, in order to repeat the scenario with the same initial conditions.

In Table 2 we summarize all the scenarios with a brief description of each of them. The last column reports the category which scenarios belong to, from a functional point of view, according to the following:

- *Idle* (Scenario 0): it is the basis of our analysis, evaluates power consumption during the periods of inactivity of the system.
- *Network* (Scenarios 1,2,7,8,11): it represents activities that involve network subsystems and Internet.
- *Productivity* (Scenario 3): it is related to activities of personal productivity.
- *File System* (Scenarios 4,5): it concerns activities that involve storage devices and File System operations.
- *Multimedia* (Scenarios 6,9,10): it represents activities that involve audio/video peripherals and multimedia contents.

The dependent variable selected for the experiment is P i.e. the instant power consumption (W). Therefore, P_n is the average power consumption during Scenario $n = 1..11$ and $P_{idle|net|prod|file|MM}$ is the average power consumption of (respectively) Idle, Network, Productivity, File System and Multimedia scenarios.

Table 2: Software Usage Scenarios Overview

Nr.	Title	Description	Category
0	Idle	No user input, no applications running, most of OS'automated services disabled.	Idle
1	Web Navigation	Open browser, visit a web-page, operate, close browser.	Network
2	E-Mail	Open e-mail client, check e-mails, read new messages, write a short message, send, close client.	Network
3	Productivity Suite	Open word processor, write a small block of text, save, close.	Productivity
4	Data Transfer (disk)	Copy a large file from a disk position to another.	File System
5	Data Transfer (USB)	Copy a large file from an USB Device to disk.	File System
6	Presentation	Execute a full-screen slide-show of a series of medium-size images.	Multimedia
7	Skype Call (no video)	Open Skype client, execute a Skype conversation (video disabled), close Skype.	Network
8	Skype Call (video)	Open Skype client, execute a Skype conversation (video enabled), close Skype.	Network
9	Multimedia (Audio)	Open a common media player, play an Audio file, close player.	Multimedia
10	Multimedia (Video)	Open a common media player, play a Video file, close player.	Multimedia
11	Peer-to-Peer	Open a common peer-to-peer client, put a file into download queue, download for 5 minutes, close.	Network

3.3 Hypothesis Formulation

Basing upon our GQM Model, we can formalize our Research Question into Hypotheses.

- *RQ 1: Does Software impact Power Consumption?*

  $H1_0$: $P_{idle} \geq P_n$, $n \in [1, 11]$
  $H1_a$: $P_{idle} < P_n$, $n \in [1, 11]$

- *RQ 2: Can we introduce a classification for software usage scenarios basing upon power consumption?*

  $H2_0$: $P_{idle} = P_{net} = P_{prod} = P_{file} = P_{MM}$
  $H2_a$: $P_{idle} \neq P_{net} \neq P_{prod} \neq P_{file} \neq P_{MM}$

3.4 Instrumentation and Experiment Design

Every scenario has been executed automatically by means of a GUI Automation Software for 5 minutes. We obtained 30 runs per scenario, each composed of 300 observations (one per second) of the instant power consumption value (W).

Table 3: HW/SW Configuration of the test machine

CPU	AMD Athlon XP 1500+
Memory	768 MB DDR SDRAM
Display Adapter	ATI Radeon 9200 PRO 128 MB
HDD	Maxtor DiamondMax Plus 9 80GB Hard Drive
Network Adapter	NIC TX PCI 10/100 3Com EtherLink XL
OS	Microsoft Windows XP Professional SP3

The test machine is a Desktop PC running Windows XP, situated in the ISCBD Lab of the University of Cordoba, Escuela Politecnica Superior. In Table 2, the Hardware/Software configuration of the machine is presented.

We used two different software and hardware tools to do monitoring, measurement and test automation.The Software tool we used for test automation is Qaliber [1], which is mainly a GUI Testing Framework, composed of a Test Developer Component, that allows a developer to write a specific test case for an application, by means of "recording" GUI commands, and a Test Builder Component, which allows to create complex usage scenarios by combining the use cases.

The measurement of power consumption was done through a power monitoring device provided by an industrial partner, called PloggMeter [2]. This device is capable of computing Active and Reactive Power, Voltage, Current Intensity, $Cos\varphi$. The data is stored within the PloggMeters 64kB memory and can be downloaded in a text file format via the RF wireless connection to a Windows enabled PC or Laptop or viewed as instantaneous readings on the installed Plogg Manager software. We modified the device drivers to adapt the PloggMeter recording capability to our purposes, specifically to decrease the logging interval from 1 minute (which is too wide if compared to software time) to 1 second.

3.5 Analysis methodology

The goal of data analysis is to apply appropriate statistical tests to reject the null hypothesis. The analysis will be conducted separately for each scenario in order to evaluate which one has an actual impact on power consumption. First of all we will test the null hypothesis $H1_0$ for each scenario. Then we will group them into categories and we will test $H2_0$ for each category. Since we expect the values not to be normally distributed, we will adopt non parametric tests, in particular we selected the Mann-Whitney test [12]. The first hypothesis $H1_0$ is clearly directional, thus the one-tailed variant of the test will be applied. The second hypothesis $H2_0$ is not directional, therefore we will apply the two sided variant of the test. We will draw conclusions from our tests based on a significance level $\alpha = 0.05$, that is we accept a 5% risk of type I error – i.e. rejecting the null hypothesis when it is actually true. Moreover, since we perform multiple tests on the same data – precisely twice: first overall and then by category – we apply the Bonferroni correction to the significance level and we actually compare the test results versus a $\alpha_B = 0.05/2 = 0.025$.

3.6 Validity evaluation

We will classify threats of experiment validity in two categories: **internal** threats, derived from our treatments and instrumentation, and **external** threats, that regard the generalization of our work. There are three main internal threats. The first concerns the *measurement sampling*: our measurements were taken with a sampling rate of 1 second. This interval is a compromise between our power metering device capability and our software logging service. However, it could be a wide interval if compared to software time. Subsequently, we could have *network confounding factors*: as we included in our treatments several usage scenarios involving network activity and the Internet, the unpredictability of the network behaviour could affect some results. Another confounding factor is represented by *OS scheduling operations*: the scheduling of user activities

Table 4: Scenarios Statistics Overview

	Mean	**Median**	**S.E.**	**C.I.**	**Variance**	σ	**Var.Co.**	**VMR**
0 - Idle	86.81	86.69	0.007	0.013	0.424	0.650	0.007	0.005
1 - Web	89.09	88.57	0.011	0.022	3.372	1.836	0.021	0.038
2 - E-Mail	88.03	87.11	0.024	0.047	5.195	2.279	0.026	0.059
3 - Prod	90.12	89.40	0.025	0.500	5.862	2.421	0.027	0.065
4 - Disk	94.12	97.21	0.048	0.095	21.12	4.595	0.049	0.224
5 - USB	96.41	97.10	0.024	0.046	5.047	2.246	0.023	0.052
6 - Image	91.97	91.48	0.041	0.081	15.474	3.934	0.043	0.168
7 - Skype	91.87	91.69	0.015	0.029	1.981	1.407	0.015	0.022
8 - SkypeV	95.40	95.75	0.020	0.040	3.844	1.960	0.020	0.040
9 - Audio	88.14	87.94	0.013	0.025	1.429	1.195	0.013	0.016
10 - Video	88.61	88.57	0.009	0.017	0.677	0.823	0.009	0.008
11 - P2P	88.46	88.25	0.010	0.019	0.842	0.917	0.010	0.009

and system calls is out of our control. This may cause some additional variability in our scenarios, especially for those that involve the File System.

Finally, the main external threat concerns a possible *limited generalization* of results: this is due to the fact that it was conducted on a single test machine.

3.7 Preliminary Data Analysis

We present in Table 4 the following descriptive statistics about measurements for each scenario. Table 4 reports in this order mean (Watts), median (Watts), standard error on the mean, 95% confidence interval of the mean, variance, standard deviation (σ), variation coefficient (the standard deviation divided by the mean), index of dispersion (variance-to-mean ratio, VMR).

Power consumptions vary from a minimum median of 86.89 W of the Idle scenario up to a maximum median of 97.21 W of the Disk scenario. Therefore the excursion is about 11 W. Moreover, the different samples for each scenario are homogeneous because variability indexes are very low.

4 Results

We provide results of hypotheses testing of the two research questions. Table 5 and 6 report the scenarios tested, the p-value of Mann-Whitney test and the estimated difference of the medians between Idle scenario and the other ones.

Question 1: Does software impact power consumption?

$H1 : P_{idle} \neq P_n \forall n \in [1, 11]$.

Question 2: Can we introduce a classification for software usage scenarios basing upon power consumption?

$H2 : P_{idle} \neq P_{net} \neq P_{prod} \neq P_{file} \neq P_{MM}$

Table 5: Hypothesis $H1$ Test Results

Scenario Comparison	p-value	Est. Diff
0 - Idle vs. 1 - Web Navigation	< 0.0001	-1.87
0 - Idle vs. 2 - E-Mail	< 0.0001	-0.52
0 - Idle vs. 3 - Productivity Suite	< 0.0001	-2.71
0 - Idle vs. 4 - IO Operation (Disk)	< 0.0001	-10.41
0 - Idle vs. 5 - IO Operation (USB)	< 0.0001	-10.41
0 - Idle vs. 6 - Image Browsing	< 0.0001	-4.69
0 - Idle vs. 7 - Skype Call (No Video)	< 0.0001	-5.10
0 - Idle vs. 8 - Skype Call (Video)	< 0.0001	-9.05
0 - Idle vs. 9 - Multimedia Playback (Audio)	< 0.0001	-1.25
0 - Idle vs. 10 - Multimedia Playback (Video)	< 0.0001	-1.87
0 - Idle vs. 11 - Peer-to-Peer	< 0.0001	-1.66

Table 6: Hypothesis $H2$ Test Results

Scenario Category Comparison	p-value	Est. Diff
Idle vs. Network	< 0.0001	-2.08
Idle vs. Productivity	< 0.0001	-2.71
Idle vs. File System	< 0.0001	-10.41
Idle vs. Multimedia	< 0.0001	-1.67
Network vs. Productivity	< 0.0001	-0.31
Network vs. File System	< 0.0001	-6.97
Network vs. Multimedia	< 0.0001	0.31
Productivity vs. File System	< 0.0001	-6.87
Productivity vs. Multimedia	< 0.0001	0.73
File System vs. Multimedia	< 0.0001	8.53

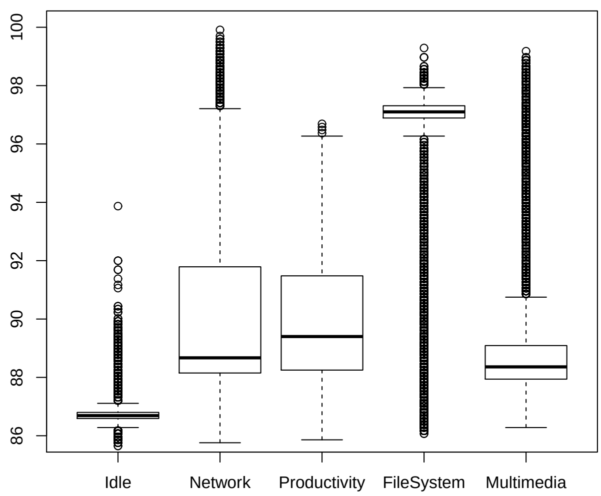

Figure 2: Box Plot of Scenario Categories

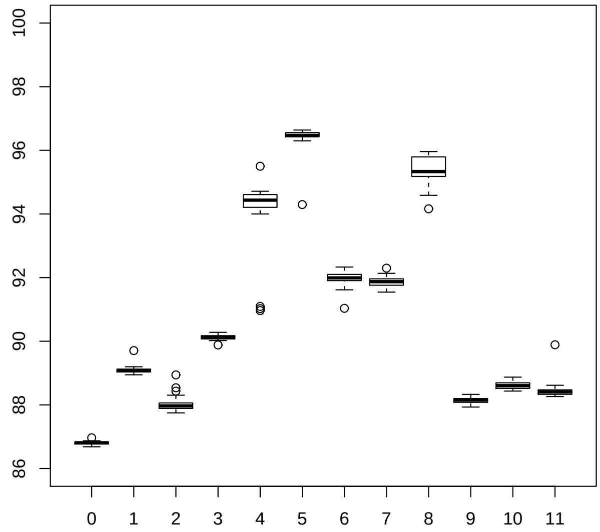

Figure 3: Box Plot of per-run Power Consumption values

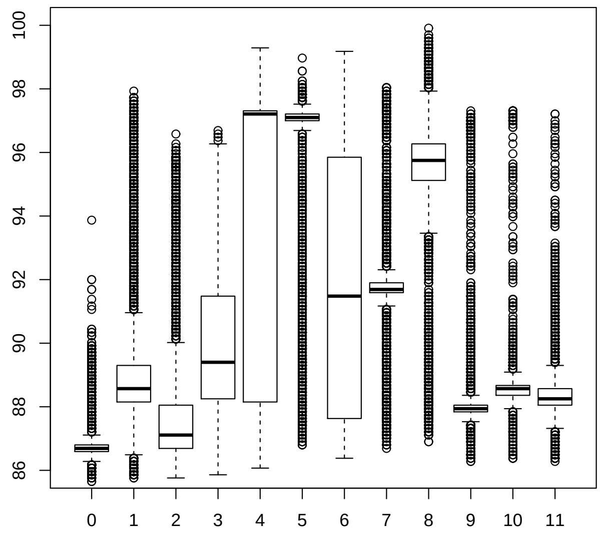

Figure 4: Box Plot of global Power Consumption values

5 Discussion

The collected data shows us several facts, and gives us the answers for our Research Questions. As we can observe in Table 5, every usage scenario consumes more power than the Idle Scenario. This difference rises up to 10.41 Watts, which represents 12% of the total Idle Power Consumption. Thus, we can undeniably affirm that software has a relevant impact upon Power Consumption, which was our first Research Question. As regards, however, our second RQ, Scenarios Classification, results are not homogeneous: for instance, we observe in Figure 2 that Network Category has a very wide range if compared to the others. Moreover, the comparison not always gives a clear distinction between the profiles. This suggests that a Classification based on functionality can be inadequate for Power Consumption. Another classification may arise from the analysis of every single Scenario. As we can see from Table 4 and 5, the most power-consuming scenarios are those that involve File System, followed by Skype (both with and without Video Enabled) and Image Browsing. From the hardware point of view, these scenarios are also the most expensive in terms of system resources. Thus, classifying our scenarios basing upon resource utilization can be a more accurate way to estimate their power consumption. For instance, the power consumption profile of Skype is very different (about 4-5 Watts in average) with and without enabling the Video Camera. Finally, another interesting question that arises from our analysis is, in case of applying these Scenarios in groups, if their power consumption would follow a linear composition rule (thus summing up the values). That is, for example, if we imagine a composed Usage Scenario S that involves a Skype Call, a Web Navigation and a Disk Operation performed simultaneously, their linear composition would give an estimated Power Consumption per second of $P_{idle} + \Delta P_S = 86.81W + 21.33W = 108.14W$, introducing about a 25% overhead power consumption.

6 Conclusions

Our experiment let us assess quantitatively the energetic impact of software usage. We built up common application usage scenarios (e.g.: Skype call, Web Navigation, Word writing) and executed them independently to collect power consumption data. Each single scenario introduced an overhead from 2 to 11 Watts (corresponding to an increase of about 12%): if their power consumption would follow a linear composition rule, the impact could be even higher.

Moreover, results set the basis for future works and research projects. First of all, our experiment will be replicated on different machines, thus making it possible for us to generalize our results. Meanwhile, we will proceed with the analysis of resource usage data, searching for statistical correlations between these values and the power consumption values. This analysis will hopefully help us to understand the relationships between resource utilization and power consumption. Our idea is that re-factoring applications by considering a more efficient resource utilization, the impact of software on power consumption could be easily reduced.

7 Acknowledgements

The authors of this article would like to thank Antonino Fiume for his precious help in the handling of the Plogg monitoring device, prof. Miguel-Angel Gomez Nieto and the ISCBD Group of the University of Cordoba (UCO) - Escuela Politecnica Superior for providing the test machine and the laboratory in which the experiment was performed.

References

[1] Qaliber - GUI Testing Framework, `http://www.qaliber.net/`

[2] Youmeter - Plogg Technologies. `http://www.youmeter.it/youmeter`, `http://www.plogginternational.com/`

[3] Flinn, J., Satyanarayanan, M.: Powerscope: A tool for profiling the energy usage of mobile applications. Mobile Computing Systems and Applications, IEEE Workshop on 0, 2 (1999)

[4] Group, T.C.: Smart 2020: Enabling the low carbon economy in the information age. Tech. rep., GeSi (2008)

[5] Kansal, A., Zhao, F., Liu, J., Kothari, N., Bhattacharya, A.A.: Virtual machine power metering and provisioning, http://research.microsoft.com/apps/pubs/default.aspx?id=120435

[6] Lorch, J.: A complete picture of the energy consumption of a portable computer. Tech. rep. (1995)

[7] Marcu, M., Vladutiu, M., Moldovan, H., Popa, M.: Thermal benchmark and power benchmark software. ArXiv e-prints (Sep 2007)

[8] Microsoft, C.: Browser Power Consumption - Leading the Industry with IE 9. http://blogs.msdn.com/b/ie/archive/2011/03/28/browser-power-consumption-leading-the-industry-with-internet-explorer-9.aspx

[9] Sinha, A., Chandrakasan, A.: Jouletrack-a web based tool for software energy profiling. In: Design Automation Conference, 2001. Proceedings. pp. 220 – 225 (2001)

[10] Van Solingen, R., Berghout, E.: Goal/Question/Metric Method. McGraw-Hill Inc.,US (January 1999)

[11] Vetro', A., Ardito, L., Morisio, M., Procaccianti, G.: Monitoring it power consumption in a research center: 7 facts. In: CPS, I. (ed.) Proceedings of Energy 2011 (2011)

[12] Wolfe, M.H..D.A.: Nonparametric Statistical Methods. John Wiley & Sons, New York (1973)