We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Energy Efficiency in the ICT - Profiling Power Consumption in Desktop Computer Systems

Giuseppe Procaccianti, Luca Ardito, Antonio Vetro' and Maurizio Morisio

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48237

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 [13]. Even if energy efficient IT technologies were developed and implemented, this figure would still grow up at a rate of 6% per year until 2020. Recently, much of the attention in green IT discussions focuses on data centers. However, it is foreseen 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 [13].

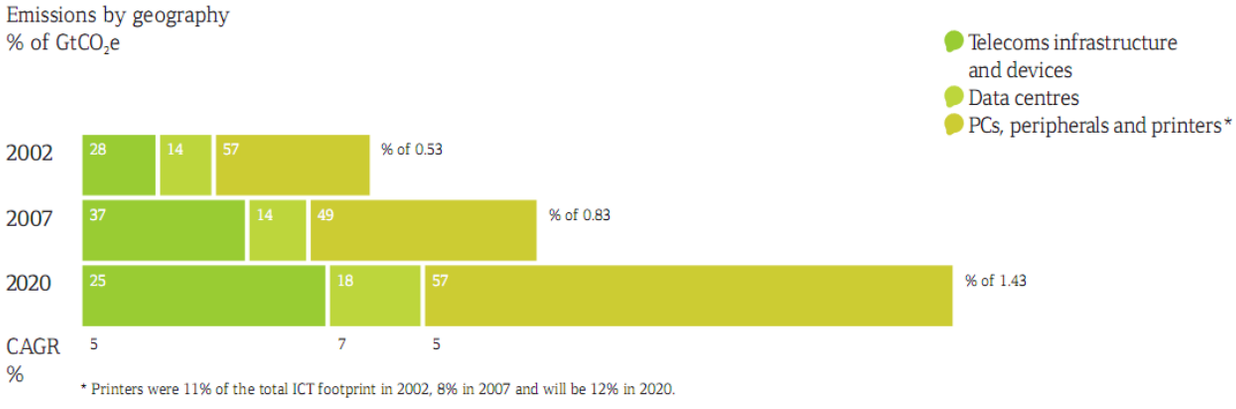


Figure 1. The 2020 global footprint by subsector

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. Finding precise figures of the current energy consumption of computer systems and ICT equipment is essential, in order to understand how to reduce their power consumption and improve their energy efficiency. Today these figures are incomplete and not precise.

INTECH
open science | open minds

©2012 Procaccianti et al., licensee InTech. This is an open access chapter distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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.

The underlying idea is that power consumption depends not only on hardware features, but also (and probably mostly) on software usage and software internal characteristics. For instance, a more complex software will require more CPU cycles, or a single long write operation on disk can be less power consuming rather than several small write operations. The energetic impact of software will also be analyzed in order to understand the effect of human behaviour on power consumption: for example we can check whether customized power management profiles are more efficient than default ones. Usually, if a person is aware of how much he is consuming, he is able to find his own solution that is the most appropriate for his device usage, and to save more power.

This chapter deals with the matter of finding relationships between software usage and power consumption. For this purpose, two experiments have been designed, consisting in running benchmarks[1] on two common desktop machines, simulating some typical scenarios and then measuring the energy consumption in order to make some statistical analysis on results.

This chapter is organized as follows:

- in Section 2, we present some related works in literature dealing with Green Software and Energy-Aware issues.
- Section 3 describes the experiment design process in all of its steps.
- in Section 4 we present the results of the experiments.
- in Section 5 we discuss the results of the experiments in detail, exposing the facts that became evident.
- in Section 6, we try to give some conclusions and present our future works.

2. Related work

In 2011, a post appeared on the MSDN Blog[2] : it concerned the energy consumption measurement of internet browsers. Authors measured power consumption and battery life of

[1] *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.* [11] In our benchmark the *workload* is a set of usage scenarios and the *metric* is the power consumption.

[2] 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`

a common laptop across six scenarios and different browsers. They allowed each scenario to run for 7 minutes and calculated the average power consumption over that duration. Authors ran *IE9*, *Firefox*, *Opera* and *Safari* for each scenario, maintaning a fixed running duration, and then they made a comparison of the obtained results. Notable differences arose regarding power consumption and laptop battery life. Namely, using IE9 the battery lasted 3:45 hours, while using Opera the battery ran out of power after 2:45 hours. Hence, software can impact on energy consumption, as we also found in our previous work [15], 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. [8] 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 [6]: 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.
In 1995, the first attempts were made to profile the energy performance of a computer. Lorch [9] 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 [12] 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%. The concept of energy-awareness is based upon a complete knowledge on how and where energy is consumed on a device. Carroll and Heiser [3] present a detailed analysis of power consumption in a mobile device, focusing on the hardware subsystems, through common and realistic usage scenarios. Results show that the GSM module and the display are the most power-consuming components: for example, a GSM phone call on OpenMoko Neo Freerunner, HTC Dream G1 and Google Nexus One consumes 1135 mW, 822 mW and 846 mW respectively.

Usually, an accurate power consumption analysis of mobile or embedded devices is component-based. However, instantaneous information about discharge current and remaining battery capacity is not always available, because most devices do not have built-in sensors to collect these data. PowerBooter [17] has been proposed as a technique to build a battery-based model automatically. Authors motivate this decision by considering that

different mobile devices of the same category show different power consumption, and a specific power consumption model for each device is difficult to obtain. Thus, instead of using external metering instrumentation to detect power consumption, only the internal battery voltage sensor is used, which is found across many modern smartphones. Authors indicate that for a 10-second interval, the PowerBooter technique has an accuracy of about 4.1% within measured values.

From a software engineering point of view, most contributions are devoted in developing frameworks and tools for energy metering and profiling. The authors of PowerBooter also propose an on-line power estimation tool called PowerTutor [17]. It implements the PowerBooter model in order to profile power consumption of applications, basing upon their component usage. Another example, which makes use of external metering devices, is ANEPROF [4], which authors define as a real-measurement-based energy profiler able to reach function-level granularity. It is developed for Android OS-based devices, thus it is aimed at profiling Java applications. It is based on JVM event profiling, using software probes to record runtime events and system calls. Authors had to address several design issues, such as overhead control and proper time synchronization. Power consumption profiling is made through correlation of real-time power measurements done by an external DAQ, connected to a ARM Computer-on-Module running Android 2.0. Authors also provide profiling data of four popular applications (Android Browser, GMail, Facebook, Youtube). The accuracy of ANEPROF depends on the hardware meter used. Its CPU overhead is stated to be less than 5%. Finally, SEMO [5] is a smart energy monitoring system, developed for Android, which provides also application-level consumption monitoring. This system is composed of three components: an *inspector*, which monitors the information on the battery, warning users when the battery reaches a critical condition; a *recorder*, which basically logs the actual charge of the battery and the running applications, and an *analyzer*, which calculates the energy consumption rate for each application and ranks them according to it.

Another alternative for energy measurement is low-level power-analysis using instruction-level models [14]. These models provide accurate power estimates for small kernels of code. An example of this kind of model is presented in Equation 1 where [10] Energy is the total energy dissipation of the program.

$$Energy = \sum_{i}(BC_i) + \sum_{i,j}(SC_{i,j}\, N_{i,j}\,) + \sum_{k}(OC_k) \tag{1}$$

The first part is the summation of the base energy cost of each instruction (BC_i is the base energy cost and N_i is the number of times instruction i is executed). The second part accounts for the circuit state ($SC_{i,j}$ is the energy cost when instruction i is followed by during the program execution). The third part accounts for energy contribution OC_k of other instruction effects such as stalls and cache misses during the program execution.

The study presented here is instead focused on the analysis of power consumption data, and it is designed to find out usage patterns of IT devices' energy consumption and to identify situations in which there is a waste of energy. Webber et al. [16] also collected data on devices, focusing on the after-hours power state of networked devices in office buildings: they showed that most of devices are left powered on during night, concluding that this is the first cause of energy waste.

Usage analysis is a crucial step to optimize the energy consumption: this task is even more necessary within data centers where the number of computers is large. In this field, Bein et al. [2] tried to improve the energy efficiency of data centers: they studied the cost of storing vast amounts of data on the servers in a data center and they proposed a cost measure together with an algorithm that minimizes such cost.

3. Study design

3.1. Goal description and research questions

The aim of this research is to assess the impact of software and its usage on power consumption in computer systems. The goal is defined through the Goal-Question-Metric (GQM) approach. [1]. This approach, applied to the experiment, led to the definition of the model presented in Table 1. The first research question investigates whether and how much software impacts power consumption. The second research question investigates whether a categorization of usage scenarios with respect to functionality is also valid for power consumption figures. The third research question tries to find a quantifiable relationship between power consumption and actual usage of the computer system, by selecting four metrics relative to the main system resources (CPU, Disk, Memory and Network).

Goal	*Evaluate* *for the purpose of* *with respect to* *from the viewpoint of* *in the context of*	software usage assessing its energetic impact power consumption the System User Desktop applications
Question 1	Does software impact power consumption?	
Metric	Consumed Power (Watts)	
Question 2	Is it possible to classify software usage scenarios basing upon power consumption?	
Metric	Consumed Power (Watts)	
Question 3	What is the relationship between usage and power consumption?	
Metrics	CPU Usage (percentage) Memory Usage (reads/writes) Disk Usage (reads/writes) Network Usage (Packets/sec) Consumed Power (Watts)	

Table 1. The GQM Model

3.2. Usage scenarios

The following usage scenarios, described in detail, will provide the basis for the analysis. The scenarios have been designed trying to simulate common operations for a desktop user, and they provide benchmarks (see Section 1) for the different resources of the computer system. This way, we will obtain useful information on the relationship between resource usage and power consumption.

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. Google Chrome has been chosen 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 the SoftEng research group `http://softeng.polito.it`, so that the same contents and navigation path could be maintained 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, Microsoft Word 2007 has been chosen, 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, a data file of a relevant size (almost 2 GB) has been prepared in order to match the file transfer time with the 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, this scenario has been developed 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,it has been thought to be reductive not to consider usage scenarios that make a more intensive use of the Internet than Web Navigation and E-Mails. Thus, the Skype scenario has been developed. Skype 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 the prefixed duration of all scenarios.

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

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

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

11 - *Peer-to-Peer Data Exchange.* As for the Skype scenarios, it has been made the decision to take into account also a Peer-to-Peer scenario, which has proven to be a very common practice among private users. For this scenario, BitTorrent was selected 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.

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

3.3. Variable selection

In order to answer the Research Questions, it is necessary to specify the independent variables that will characterize the experiment. As anticipated in the previous section, four metrics have been selected to evaluate the system usage. These metrics were measured by means of software logging (as will be explained in the *Instrumentation* section) considering the following values:

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	Audio Playback	Open a common media player, play an Audio file, close player.	Multimedia
10	Video Playback	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

Table 2. Software Usage Scenarios Overview

- CPU
  - CPU Time Percentage, intended as time spent by the CPU doing active work in a second
  - CPU User Time Percentage, intended as time spent by the CPU executing user instructions (i.e. applications) in a second
  - CPU Privileged Time Percentage, intended as time spent by the CPU executing system instructions (services, daemons) in a second
  - CPU Deferred Procedure Calls Percentage, intended as time spent by the CPU executing DPC in a second
  - CPU Interrupt Time Percentage, intended as time spent by the CPU serving interrupts in a second

  - CPU C1 Time Percentage, intended as time spent by the CPU in low-power (C1) State
  - CPU C2 Time Percentage, intended as time spent by the CPU in low-power (C2) State
  - CPU C3 Time Percentage, intended as time spent by the CPU in low-power (C3) State
- Memory
  - Memory Page Writings per second
  - Memory Page Readings per second
  - Memory Available (KiloBytes) per second
- Hard Disk
  - Physical Disk Transfers (Read/Write) per second
  - Logical Disk Transfers (Read/Write) per second
- Network
  - Network Packets per second as seen by the Network Interface Card

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.

3.4. Hypotheses formulation

Basing upon the GQM Model, the Research Questions can be formalized into Hypotheses. In order to formally express Research Question 3, $\rho(x, y)$ expresses the correlation coefficient between variables x and y. β represents a significant correlation value, which will be defined later in this Section.

- *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: Is it possible to classify software usage scenarios basing upon power consumption?*

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

- *RQ 3: What is the relationship between usage and power consumption?*

  $H3_0$: $\rho(I_{CPU}, P) = \rho(I_{Memory}, P) = \rho(I_{Disk}, P) = \rho(I_{Network}, P) = 0$
  $H3_a$: $max[\rho(I_{CPU}, P), \rho(I_{Memory}, P), \rho(I_{Disk}, P), \rho(I_{Network}, P)] > \beta$

3.5. Instrumentation

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

The test machines selected are two desktop PCs of different generations. In Table 3, the Hardware/Software configuration of the machines is presented. As can be seen, the difference

	Desktop 1 (old generation)	Desktop 2 (new generation)
CPU	AMD Athlon XP 1500+	Intel Core i7-2600
Memory	768 MB DDR SDRAM	4 GB DDR3 SDRAM
Display Adapter	ATI Radeon 9200 PRO 128 MB	ATI Radeon HD 5400
HDD	Maxtor DiamondMax Plus 9 80GB Hard Drive	Western Digital 1 TB
Network Adapter	NIC TX PCI 10/100 3Com EtherLink XL	Intel 82579V Gigabit Ethernet
OS	Microsoft Windows XP Professional SP3	Windows 7 Professional SP1

Table 3. HW/SW Configuration of the test machine

in terms of hardware is relevant; this will allow us to make some evaluations about how power consumption varied over the years, with the evolution of hardware architectures.

Different software and hardware tools have been used to do monitoring, measurement and test automation. The Software tool adopted is Qaliber[3], (see Figure 2) 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. One of the most important features of Qaliber is its possibility to log system information during scenario execution, using Microsoft's Performance Monitor Utility. By defining a specific Counter Log, adding all the variables of interest, it is possible to tell Qaliber to start Performance Monitor simultaneously with the Scenario, thus allowing a complete monitoring of all the statistics needed for this analysis.

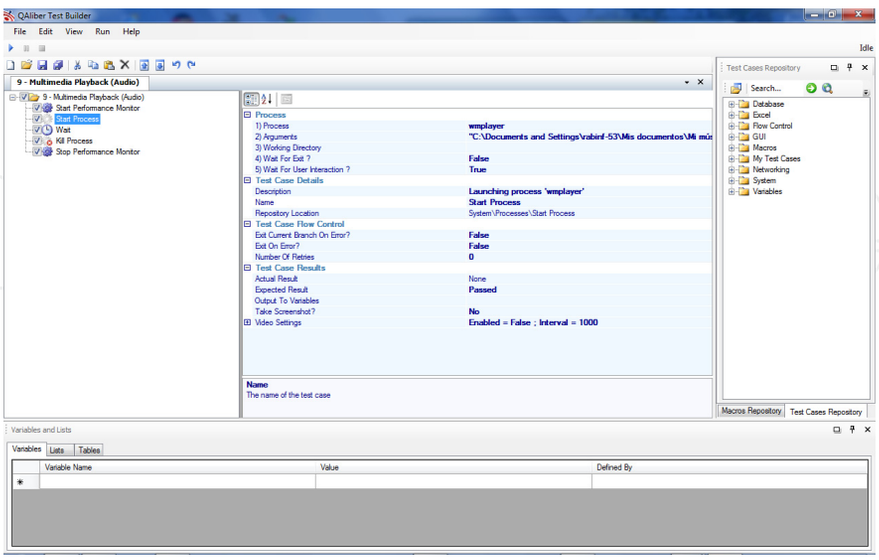


Figure 2. Qaliber Test Builder screenshot

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

The measurement of power consumption was done through two different devices. For the old-generation PC, PloggMeter[4] (see Figure 3) device was used. This device is capable of computing Active and Reactive Power, Voltage, Current Intensity, $Cos\varphi$. The data is stored within the PloggMeter's 64kB memory and can be downloaded in a text file format via Zigbee wireless connection to a Windows enabled PC or Laptop or viewed as instantaneous readings on the installed Plogg Manager software. The device drivers were slightly modified to adapt the PloggMeter recording capability to this analysis' purposes, specifically to decrease the logging interval from 1 minute (which is too wide if compared to software time) to 1 second.



Figure 3. The PloggMeter device

For the new-generation PC, WattsUp PRO ES[5] (see Figure 4) device was used. This device is capable of measuring current power consumption (Watts), power factor, line voltage and other metrics. The data is stored within the device internal memory, and then downloadable via USB interface. The sampling rate resolution is 1 second.



Figure 4. The WattsUp Pro ES device

[4] Youmeter - Plogg Technologies, `http://www.plogginternational.com/products.shtml`
[5] WattsUp Pro ES, `https://www.wattsupmeters.com/secure/products.php?pn=0&wai=0&spec=2`

3.6. 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.

In order to extract a Power Consumption profile for each Usage Scenario, a set of descriptive statistics was derived from the experimental data. For a single scenario, a total of 30 runs were executed, each composed of 300 observations (one per second) of the power consumption value. Thus, the calculations for the descriptive statistics were made using two approaches: firstly, the average of each run is extracted, obtaining a short vector of 30 elements, which was used as the subject of our analysis. This method allowed to speed up the calculations, and because of the decreased sampling rate, the data was less variant and showed an almost regular distribution.

Afterwards, the same analysis on the full datasets was applied, which means a total of 9000 observations. Comparing the results from these two approaches, focusing on the Index of Dispersion and the variance, the variability of a single scenario can be appreciated, which was also a useful tool for validating the experiment.

First of all, the null hypothesis $H1_0$ will be tested for each scenario. Then the scenarios will be grouped into categories and $H2_0$ will be tested for each category.

First of all, data distribution must be analysed, in order to determine the appropriate testing method for each hypothesis. The data distribution analysis was conducted using the Shapiro-Wilk normality test. Since its results pointed out that the data was not normally distributed, non parametric tests were adopted, in particular the Mann-Whitney test [7] for testing $H2_0$, and the Spearman's rank correlation coefficient (also known as Spearman's ρ) for testing $H3_0$.

The first hypothesis $H1_0$ is clearly directional, thus the one-tailed variant of the test will be applied. The second and third hypotheses $H2_0$, $H3_0$ are not directional, therefore the two-sided variant of the tests will be applied.

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 $\alpha_B = 0.05/2 = 0.025$. As regards Spearman's ρ significance, using 298 degrees of freedom (since 300 observations per scenario are available) the significance level of the ρ coefficient is 0.113. Thus, correlations coefficients resulting higher than this value can be considered as significant positive or negative correlations.

3.7. Validity evaluation

The threats of experiment validity can be classified in two categories: **internal** threats, derived from treatments and instrumentation, and **external** threats, that regard the generalization of the work.

There are three main internal threats that can affect this analysis. The first concerns the *measurement sampling*: measurements were taken with a sampling rate of 1 second. This

interval is a compromise between the power metering device capability and the software logging service. However, it could be a wide interval if compared to software time.

Subsequently, *network confounding factors* could arise: as several usage scenarios involving network activity and the Internet are included in our treatments, the unpredictability of the network behaviour could affect some results. Another confounding factor is represented by *OS scheduling operations*: the scheduling of user activities and system calls is out of the experiment control. This may cause some additional variability in the scenarios, especially for those that involve the File System.

In addition, the two machines on which our tests are performed are different in terms of hardware and software configuration. This is done on purpose, because we wanted to test devices which could represent common machines used in home and office scenarios, for both generations. Thus, installing an old version of an operating system on a new machine or viceversa would have altered this assumption. However, this will introduce another confounding factor, but still, will provide useful information regarding the evolution of these systems, even if no specific research hypotheses can be verified about the comparison.

Finally, the main external threat concerns a possible *limited generalization* of the results: this is due to the fact that the experiment was conducted on only two different test machines, which is a limited population to be representative of a whole category.

4. Results

4.1. Preliminary data analysis

We present in Table 4 and Table 5 the following descriptive statistics about measurements for each scenario. Tables 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 show an excursion of about 11 W for both PCs, even if the baseline is quite different (an average of 87 W in Idle scenario for the Old PC, 51 W for the New PC). Moreover, the very low variability indexes ensure that the different samples for each scenario are homogeneous.

4.2. Hypothesis testing

The results of hypotheses testing of the research questions are exposed in this section.

The testing of hypothesis H_1 and H_2 are exposed in Table 6 and 7. These table 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.

Figure 5 represents the bar plot of the power consumption increase (in watts), with respect to idle, of each scenario. Figure 6 shows the box plot of scenario categories for each PC. As regards hypothesis H_3, which evaluates correlations between resource usage and power consumption, more steps are needed. First of all, Table 8 reports the results of the Data Distribution Analysis. Then, in Table 9 and Table 10, are presented the results of the correlation

	Old-Generation PC							
	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

Table 4. Scenarios Statistics Overview: Old-Generation PC

	New-Generation PC							
	Mean	**Median**	**S.E.**	**C.I.**	**Variance**	σ	**Var.Co.**	**VMR**
0 - Idle	51.39	51.20	0.007	0.015	0.507	0.712	0.013	0.009
1 - Web	54.05	53.90	0.014	0.028	1.883	1.372	0.025	0.035
2 - E-Mail	53.40	53.40	0.011	0.021	1.123	1.059	0.019	0.021
3 - Prod	53.09	52.70	0.016	0.032	2.369	1.539	0.029	0.044
4 - Disk	60.24	62.10	0.037	0.072	12.38	3.518	0.058	0.205
5 - USB	61.29	61.90	0.023	0.046	4.901	2.214	0.036	0.080
6 - Image	52.75	52.50	0.011	0.023	1.214	1.102	0.021	0.023
7 - Skype	56.23	56.30	0.016	0.032	2.420	1.555	0.027	0.043
8 - SkypeV	62.13	62.90	0.036	0.070	11.428	3.380	0.054	0.184
9 - Audio	52.87	52.70	0.006	0.012	0.315	0.561	0.010	0.006
10 - Video	54.14	54.00	0.007	0.013	0.420	0.648	0.012	0.008
11 - P2P	54.32	54.50	0.008	0.016	0.609	0.780	0.014	0.011

Table 5. Scenarios Statistics Overview: New-Generation PC

test using Spearman's method, with a 95% confidence interval, applied to every couple $(watt, variable)$ for each scenario. Only the significant coefficients are listed.

4.2.1. Question 1: Does software impact power consumption?

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

4.2.2. Question 2: Is it possible to classify software usage scenarios basing upon power consumption?

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

	Old-Gen PC		*New-Gen PC*	
Scenario Comparison	**p-value**	**Est. Diff**	**p-value**	**Est. Diff**
0 - Idle vs. 1 - Web Navigation	< 0.0001	-1.87	< 0.0001	-2.60
0 - Idle vs. 2 - E-Mail	< 0.0001	-0.52	< 0.0001	-2.10
0 - Idle vs. 3 - Productivity Suite	< 0.0001	-2.71	< 0.0001	-1.50
0 - Idle vs. 4 - IO Operation (Disk)	< 0.0001	-10.41	< 0.0001	-10.80
0 - Idle vs. 5 - IO Operation (USB)	< 0.0001	-10.41	< 0.0001	-10.60
0 - Idle vs. 6 - Image Browsing	< 0.0001	-4.69	< 0.0001	-1.20
0 - Idle vs. 7 - Skype Call (No Video)	< 0.0001	-5.10	< 0.0001	-5.00
0 - Idle vs. 8 - Skype Call (Video)	< 0.0001	-9.05	< 0.0001	-11.50
0 - Idle vs. 9 - Audio Playback	< 0.0001	-1.25	< 0.0001	-1.50
0 - Idle vs. 10 - Video Playback	< 0.0001	-1.87	< 0.0001	-2.80
0 - Idle vs. 11 - Peer-to-Peer Data Exchange	< 0.0001	-1.66	< 0.0001	-3.30

Table 6. Hypotheses $H1$ Test Results

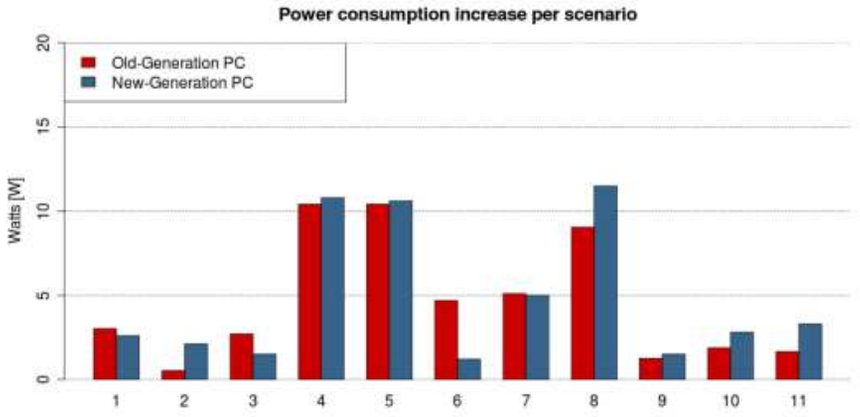


Figure 5. Bar Plot of per-scenario Power Consumption increase with respect to Idle

	Old-Gen PC		*New-Gen PC*	
Scenario Comparison	**p-value**	**Est. Diff**	**p-value**	**Est. Diff**
Idle vs. Network	< 0.0001	-2.08	< 0.0001	-3.20
Idle vs. Productivity	< 0.0001	-2.71	< 0.0001	-1.50
Idle vs. File System	< 0.0001	-10.41	< 0.0001	-10.60
Idle vs. Multimedia	< 0.0001	-1.67	< 0.0001	-1.60
Network vs. Productivity	< 0.0001	-0.31	< 0.0001	1.70
Network vs. File System	< 0.0001	-6.97	< 0.0001	-6.80
Network vs. Multimedia	< 0.0001	0.31	< 0.0001	1.60
Productivity vs. File System	< 0.0001	-6.87	< 0.0001	-9.10
Productivity vs. Multimedia	< 0.0001	0.73	< 0.0001	-0.20
File System vs. Multimedia	< 0.0001	8.53	< 0.0001	8.60

Table 7. Hypothesis $H2$ Test Results

4.2.3. Question 3: What is the relationship between usage and power consumption?

$H3_a :\neq max[\rho(I_{CPU}, P), \rho(I_{Memory}, P), \rho(I_{Disk}, P), \rho(I_{Network}, P)] > \beta$

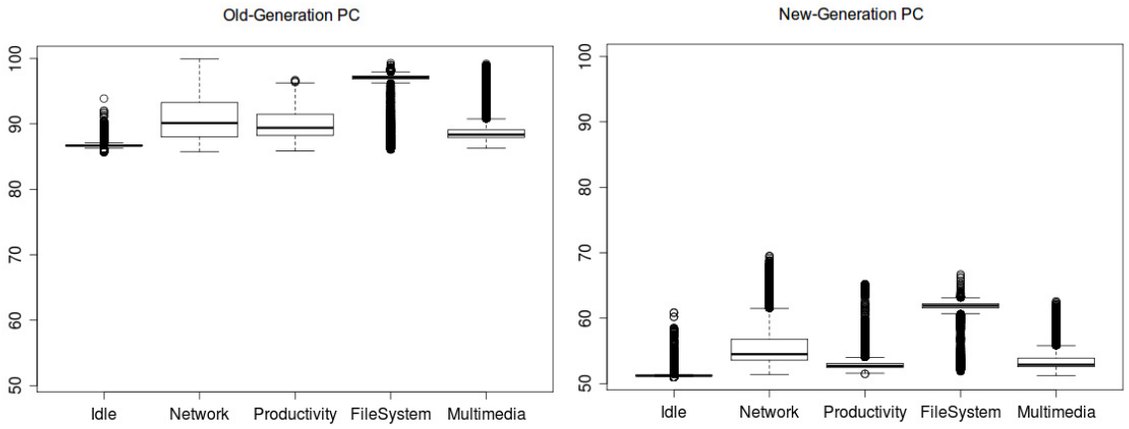


Figure 6. Box Plots of Scenario Categories

	Old-Gen PC		*New-Gen PC*	
Scenario	**Data Distr.**	**Max p-val.**	**Data Distr.**	**Max p-val.**
0 - Idle	Not Normal	1.5e-63	Not Normal	2.2e-39
1 - Web Navigation	Not Normal	4.4e-36	Not Normal	1.1e-20
2 - E-Mail	Not Normal	9e-73	Not Normal	1.2e-19
3 - Productivity Suite	Not Normal	1e-45	Not Normal	9.4e-29
4 - IO Operation (Disk)	Not Normal	1.2e-46	Not Normal	8.7e-51
5 - IO Operation (USB)	Not Normal	6.4e-52	Not Normal	2.5e-29
6 - Image Browsing	Not Normal	1.1e-35	Not Normal	6.7e-22
7 - Skype Call (No Video)	Not Normal	8.2e-30	Not Normal	3e-67
8 - Skype Call (Video)	Not Normal	1.3e-35	Not Normal	5.2e-36
9 - Audio Playback	Not Normal	7.9e-54	Not Normal	5.2e-44
10 - Video Playback	Not Normal	1.6e-44	Not Normal	6.6e-81
11 - Peer-to-Peer Data Exchange	Not Normal	8.9e-36	Not Normal	2.2e-35

Table 8. Data Distribution Analysis

Old-Generation PC				
Scenario Title	**Variable**	**p-value**	ρ	**R2**
2 - E-Mail	CPUC1Time.	< 0.0001	-0.36	13 %
4 - IO Operation (Disk)	CPUTime.	< 0.0001	0.35	12 %
4 - IO Operation (Disk)	CPUC1Time.	< 0.0001	-0.35	12 %
5 - IO Operation (USB)	CPUTime.	< 0.0001	0.47	22 %
5 - IO Operation (USB)	CPUC1Time.	< 0.0001	-0.47	22 %
7 - Skype Call (No Video)	CPUC1Time.	< 0.0001	-0.39	15 %
8 - Skype Call (Video)	CPUTime.	< 0.0001	0.63	40 %
8 - Skype Call (Video)	CPUUserTime.	< 0.0001	0.53	28 %
8 - Skype Call (Video)	CPUC1Time.	< 0.0001	-0.7	49 %
11 - Peer-to-Peer	MemoryKByteAvailable	< 0.0001	-0.34	12 %

Table 9. Spearman's ρ Coefficient between Power and Resource variables

5. Discussion

The collected data shows several facts, and gives the answers for the Research Questions.

New-Generation PC				
Scenario Title	**Variable**	**p-value**	ρ	**R2**
2 - E-Mail	CPUUserTime.	< 0.0001	0.42	17 %
2 - E-Mail	CPUPrivTime.	< 0.0001	0.43	18 %
3 - Productivity Suite	CPUUserTime.	< 0.0001	0.33	11 %
4 - IO Operation (Disk)	PhysicalDiskTransfers	< 0.0001	0.45	20 %
4 - IO Operation (Disk)	LogicalDiskTransfers	< 0.0001	0.45	20 %
4 - IO Operation (Disk)	MemoryPages	< 0.0001	0.44	19 %
4 - IO Operation (Disk)	MemoryKByteAvailable	< 0.0001	-0.54	29 %
4 - IO Operation (Disk)	CPUC3Time.	< 0.0001	-0.59	35 %
4 - IO Operation (Disk)	CPUTime.	< 0.0001	0.55	31 %
4 - IO Operation (Disk)	CPUUserTime.	< 0.0001	0.58	34 %
4 - IO Operation (Disk)	CPUPrivTime.	< 0.0001	0.39	15 %
6 - Image Browsing	CPUUserTime.	< 0.0001	0.34	12 %
7 - Skype Call (no video)	NetworkPkts	< 0.0001	0.62	39 %
7 - Skype Call (no video)	MemoryKByteAvailable	< 0.0001	-0.45	20 %
7 - Skype Call (no video)	CPUC3Time.	< 0.0001	-0.66	43 %
7 - Skype Call (no video)	CPUTime.	< 0.0001	0.52	27 %
7 - Skype Call (no video)	CPUUserTime.	< 0.0001	0.63	39 %
8 - Skype Call (Video)	NetworkPkts	< 0.0001	0.67	46 %
8 - Skype Call (Video)	MemoryKByteAvailable	< 0.0001	-0.62	39 %
8 - Skype Call (Video)	CPUC3Time.	< 0.0001	-0.88	77 %
8 - Skype Call (Video)	CPUTime.	< 0.0001	0.87	76 %
8 - Skype Call (Video)	CPUUserTime.	< 0.0001	0.9	81 %
9 - Audio Playback	MemoryKByteAvailable	< 0.0001	-0.34	12 %
11 - Peer-to-peer	NetworkPkts	< 0.0001	0.45	20 %
11 - Peer-to-peer	MemoryKByteAvailable	< 0.0001	-0.42	18 %
11 - Peer-to-peer	CPUPrivTime.	< 0.0001	0.35	12 %

Table 10. Spearman's ρ Coefficient between Power and Resource variables

5.1. Question 1: Does software impact power consumption?

As observed in Table 6, in both our test machines, every usage scenario consumes more power than the Idle scenario. This difference is even more evident in the New-Generation PC, where we obtain our highest increase percentage (up to 20%), as can be observed in Figure 7.

5.2. Question 2: Is it possible to classify software usage scenarios basing upon power consumption?

As regards our second RQ, scenarios classification, results are not homogeneous: for instance, in Figure 6 it can be observed 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 can be seen from Tables 4, 5 and 6, 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

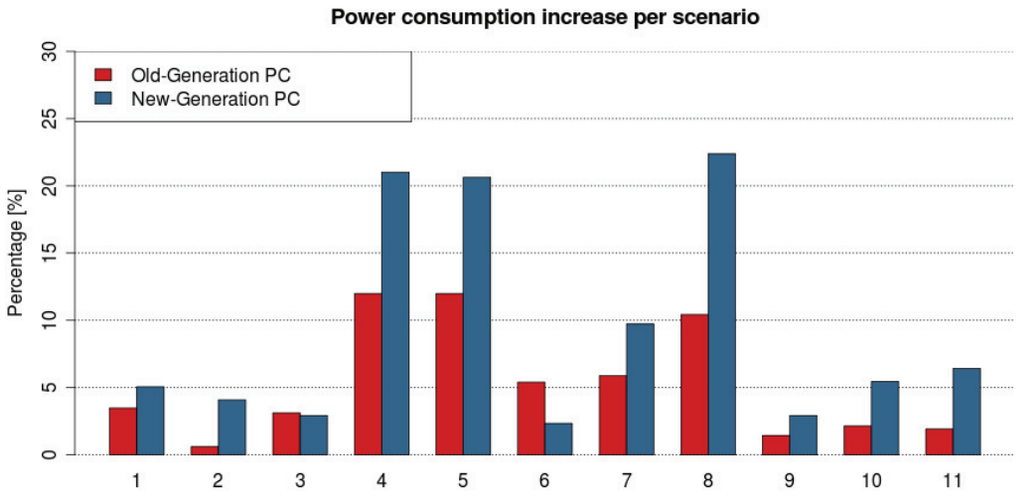


Figure 7. Per-scenario Power Consumption increase with respect to Idle (in percentage)

consumption profile of Skype is very different (about 4-5 Watts in average) with and without enabling the Video Camera.

Another interesting question that arises from the 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, supposing a composed Usage Scenario S that involves a Skype Call, a Web Navigation and a Disk Operation performed simultaneously, their linear composition would give, on our Old-Gen PC, an estimated Power Consumption per second of

$$P_{idle} + \Delta P_S = 86.81W + 21.33W = 108.14W$$

introducing a 25% overhead on power consumption. On the New-Gen PC, the estimated Power Consumption would be

$$P_{idle} + \Delta P_S = 51.39W + 24.90W = 76.29W$$

which gives a 48% overhead on power consumption.

5.3. Question 3: What is the relationship between usage and power consumption?

Taking a look at the results of the correlation analysis, represented by the third Research Questions, more conclusions can be made. First of all, we can observe that the coefficients related to the New-Gen PC are higher with respect to the Old-Gen PC. This may suggest that as hardware evolves, the software usage is even more significant for determining the power consumption of the system. This assumption is confirmed by Figure 7, where we can observe that the percentage increase of the New-Gen PC is higher, in most cases, with respect to the Old-Gen.

However, it is remarkable that, for both machines, the variables that show higher correlation coefficients are undoubtedly those related to CPU Usage and Memory Usage. High

coefficients are also present in the Hard Disk Index, but only in those scenarios that, clearly, involve File System operations. This means that those resources have a greater influence upon power consumption related to the others selected for the analysis. Further researches should probably focus upon these two variables.

As expected, power consumption has always a *negative* correlation with the time spent by the CPU in the C1 and C3 states, which are power saving, low-activity states, and with the available memory, which means that using more memory has a positive correlation with power, which is a reasonable and correct behaviour. This is also a confirmation that the analysis was conducted with the right premises.

Moreover, as expected, the scenarios who exhibit higher correlations are those who use more resources, such as Skype and IO scenarios. In particular, the Skype scenario with video enabled has a strong correlation with the CPU usage, probably because the real-time video elaboration makes the CPU the dominant resource for power consumption.

6. Conclusions and future works

This experiment assessed quantitatively the energetic impact of software usage. It consisted in building up common application usage scenarios (e.g.: Skype call, Web Navigation, Word writing) and executing them independently to collect power consumption data. Each single scenario introduced an overhead on power consumption, which may raise up to 20% for recent systems: if their power consumption would follow a linear composition rule, the impact could be even higher.

The relationship between usage and power consumption was also analysed in terms of correlation between resource usage. Although a clear linear relationship did not arise, the analysis showed that some resources drive power consumption more than others, such as memory and CPU usage. Using a precise control over how an application consumes these resources, it can be possible to predict its power consumption, thus including dedicated countermeasures in the Software Design Phase – which, by itself, is the essence of Energy-Aware Programming.

Our experiment also gives us the indication that modern desktop systems, although being more energy efficient in standby and idle states, due to their higher scalability, are even more sensible to the impact of software usage on power consumption. This indicates that research should focus on reducing this impact, as it will be always more significant as time goes by.

Moreover, results set the basis for future works and research projects. A more accurate correlation analysis will be conducted, focusing on the more relevant resources and taking into account also different kinds of relationships (not just linear). Moreover, we will focus our attention on battery-powered mobile devices, where software power consumption is a key issue. 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.

Author details

Giuseppe Procaccianti, Luca Ardito, Antonio Vetro' and Maurizio Morisio
Politecnico di Torino, Italy

7. References

[1] Basili, V. R., Caldiera, G. & Rombach, H. D. [1994]. The goal question metric approach, *Encyclopedia of Software Engineering*, Wiley.

[2] Bein, D., Bein, W. & Phoha, S. [2010]. Efficient data centers, cloud computing in the future of distributed computing, *Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations*, ITNG '10, IEEE Computer Society, Washington, DC, USA, pp. 70–75.

[3] Carroll, A. & Heiser, G. [2010]. An analysis of power consumption in a smartphone, *Usenix technical conference*, Boston, MA, USA, pp. 1–14.

[4] Chung, Y., Lin, C. & King, C. [2011]. ANEPROF: Energy Profiling for Android Java Virtual Machine and Applications, *Parallel and Distributed Systems, International Conference on* 0: 372–379.

[5] Ding, F., Xia, F., Zhang, W., Zhao, X. & Ma, C. [2012]. Monitoring energy consumption of smartphones, *CoRR* abs/1201.0218.

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

[7] Hollander, M. & Wolfe, D. A. [1973]. *Nonparametric Statistical Methods*, John Wiley & Sons, New York.

[8] Kansal, A., Zhao, F., Liu, J., Kothari, N. & Bhattacharya, A. A. [2010]. Virtual machine power metering and provisioning, *Proceedings of the 1st ACM symposium on Cloud computing*, SoCC '10, ACM, New York, NY, USA, pp. 39–50.

[9] Lorch, J. [1995]. A complete picture of the energy consumption of a portable computer, *Technical report*.

[10] Macii, E., Pedram, M. & Somenzi, F. [1998]. High-level power modeling, estimation, and optimization, *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 17(11): 1061 –1079.

[11] Marcu, M., Vladutiu, M., Moldovan, H. & Popa, M. [2007]. Thermal benchmark and power benchmark software, *ArXiv e-prints* .

[12] Sinha, A. & Chandrakasan, A. [2001]. Jouletrack-a web based tool for software energy profiling, *Design Automation Conference, 2001. Proceedings*, pp. 220 – 225.

[13] The Climate Group [2008]. Smart 2020: Enabling the low carbon economy in the information age, *Technical report*, GeSi.

[14] Tiwari, V., Malik, S. & Wolfe, A. [1994]. Power analysis of embedded software: a first step towards software power minimization, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 2(4): 437 –445.

[15] Vetro', A., Ardito, L., Morisio, M. & Procaccianti, G. [2011]. Monitoring IT power consumption in a research center: 7 facts, *in* IEEE CPS (ed.), *Proceedings of Energy 2011*.

[16] Webber, C. A., Roberson, J. A., McWhinney, M. C., Brown, R. E., Pinckard, M. J. & Busch, J. F. [2006]. After-hours power status of office equipment in the usa, *Energy* 31(14): 2823 – 2838.

[17] Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., Mao, Z. M. & Yang, L. [2010]. Accurate online power estimation and automatic battery behavior based power model generation for smartphones, *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, CODES/ISSS '10, ACM, New York, NY, USA, pp. 105–114.