Energy Profiler Framework

ENERGY PROFILER FRAMEWORK

Maurizio Morisio, Marco Torchiano, Antonio Vetro'
Politecnico di Torino
name.surname@polito.it

Abstract:

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 (10% of energy demand in 2009 - http://www.smart2020.org/).

To reduce the power consumption of both installed ICT equipment, and to design future energy efficient equipment it is essential to have precise figures of the current consumption. Today these figures are incomplete and not precise.

The goal of this proposal is to build a software framework capable of collecting power consumption data from fixed and mobile terminals in a network (PCs, servers, mobile phones, etc). The framework should be easy to install and operate, scalable and – of course – energy efficient. Data collection on power consumption is made on three levels: global, per hardware component (CPU, disk drive, peripherals, RF devices), per software function (OS, services, applications).

The framework offers (software) power gauges to be installed seamlessly on terminals, an XML data format to represent power consumption, and repositories to store the data collected. The goal of our idea is to understand:

- Real measurements on systems' power consumption,
- Relationships between power consumption and usage/characteristics of applications,
- Relationships between power consumption and user characteristics,
- Systems/applications that need improvements in order to be more efficient,
- The effect of machine based power reduction techniques,
- The effect of human based power reduction techniques.

Description:

Our idea is to develop an Energy Profiler Framework, starting from a Energy Profiler Software that can be installed on any electronic device that is connected to the network (notebook, desktop, mobile phones, printers, etc), and that has the capability of computing power consumption for such devices. The computation of the power information will be done by means of existent tools/API for the different Operating Systems installed in the nodes, while we'll use the network in order to transmit this information to a central server.

On the more complex electronic equipments, like computers and mobile phones, the Energy Profiler will be able to collect at regular time intervals, not only the power consumption, but also context information. For computers, our goal is to achieve the following context information:

- CPU utilization (overall, and by application/service that is running)
- average residency in C-States
- CPU frequency,
- number and causes of CPU wakeups
- type and causes of memory accesses
- type ad causes of disk accesses (internal and/or external)
- number and type of USB/ Bluetooth/serial devices connected
- network traffic (download/upload)

We will investigate whether this information can be obtained by means of existent tools/API or code from scratch must be developed. Furthermore, in the context of mobile phones, we are going to collect, at regular time intervals, the following information:

- average battery consumption
- applications used (frequency, average time, used resources)
- number and duration of phone calls
- number of short messages sent/received,
- data sent/received via GPRS/UMTS/HSDPA
- Internet connections (duration, data sent/received)
- GPS usage time.

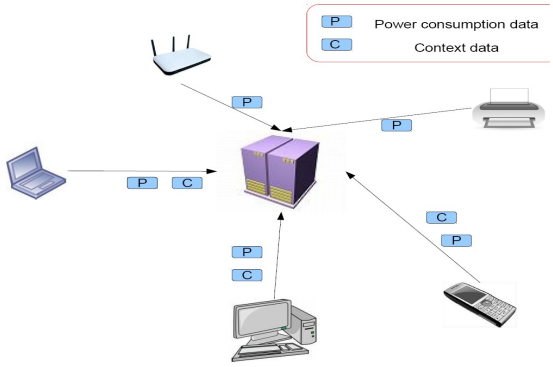
The underlying idea is that power consumption depends not only on hardware features, but also (and we suppose mostly) on software usage and software internal characteristics. For instance, a more complex software will require more CPU cycles, or a single long write operation on disk can be less power consuming rather than several small write operations.

Part of this information will be also used in order to understand the effect of human behavior 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's consuming, he's able to find his own solution that is the most appropriate for his device usage, and to save more power.

Information collected by the Energy Profiler Software will be sent to a central data center, where data can be statistically analyzed. The more suitable communication protocol, data format and transmission network will be studied during the research. The data format is particularly important: based on our previous experience on mobile phones environment, we suppose to use XML in order build the data format ad hoc. We will install and operate the Software Energy Profiler at the level of our Department, not only for practical reasons, but also because we believe that universities are the physical place of the research: "*turning on research universities into living laboratories of the greener future*" [http://www.computer.org/portal/web/computingnow/0110/theme/internetcomputing2], we will permit to quickly develop best practices and to make them available to industry and society in general. We expect, for this first experiment, about 100 installations in our university. Installations will be done with the end user agreement, and the following system will be built:

- a web site with the following functionalities/information:
  - description of the project
  - description of the tool
  - description of the information retrieved
  - end user agreement check-box
  - download button
- a mailing list with which we will inform the possible collaborators about the experiment (linking to the web site), asking their contribution and updating them about the progress of the experiment .

The model of the framework is that one of the Hackystat project [http://code.google.com/p/hackystat]: Hackystat users typically attach software 'sensors' to their development tools, which unobtrusively collect and send "raw" data about development to a web service for storage. The repository can be queried by other web services to form higher level abstractions of this raw data, and/or integrate it with other internet-based communication or coordination mechanisms, and/or generate visualizations of the raw data, abstractions, or annotations. So in our case the users will intentionally install our Profiler Software in their devices, and starting from that time these "software sensors" will collect both power consumption and context data and they will transmit them regularly to the central server.



Benefits to the software engineering community

This research will constitute a contribution in the comprehension of the power consumption of typical ICT products. We suppose our findings will be useful for the software engineering community in several ways:

- achieve a more precise estimation of ICT products ecological footprint,
- improve efficiency of those software that have higher power consumption,
- develop a suite of software configurations in order to minimize power consumption,
- develop a recommendation/profiling/auto-adaptive system for decreasing power consumption in devices' usage.