Computer Communications 50 (2014) 95–109

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Modeling power management in networked devices

Roberto Bruschi [a,*], Paolo Lago [b], Alfio Lombardo [c], Giovanni Schembra [c]

[a] National Inter-University Consortium for Telecommunications, Italy
[b] Department of Electrical, Electronic and Telecommunication Engineering, and Naval Architecture (DITEN), University of Genoa, Italy
[c] Dipartimento di Ingegneria Elettrica, Elettronica e Informatica (DIEEI), University of Catania, Italy

ARTICLE INFO

Article history:
Available online 5 April 2014

Keywords:
Power management
Networked devices
Markov models
Quality of service

ABSTRACT

This paper focuses on the power management of state-of-the-art networked devices (like common PCs, servers, set-top boxes, etc.) to evaluate their behavior, model their internal dynamics and possible sources of inefficiency, and optimize their performance and energy efficiency. To this purpose, we started from an experimental characterization of the power management schemes in common device platforms based on commercial off-the-shelf hardware and open-source software (i.e., common PCs/servers devices running the Linux operating system). The characterization allowed us to formalize an analytical model able to accurately capture the power management dynamics at hardware (at ACPI level and beyond) and software levels (Linux Governors). Finally, the proposed model has been applied to analyze the efficiency of networked devices according to various configurations of internal parameters and incoming workload. Thanks to its intrinsic accuracy and the representation of different fine-grained details, the model is able to provide precious information on the possible sources of inefficiency, and on how to act on policy parameters to optimize the system behavior.

1. Introduction

In the last few years, a number of studies have identified the eco-sustainability as one of the key aspects that may potentially constraint the Internet technology evolution, and its wide-adoption as public telecommunication infrastructure. Despite obvious environmental reasons, this interest springs from heavy and critical economical needs [1,2].

For supplying networking infrastructures and large-scale datacenters, telecom and service operators have so impressive and continuously increasing power requirements that appear to be among the major direct energy consumers in their nations. Telecom operators and third-parties estimated that such energy requirements will rapidly become no more sustainable, if no radical changes in Internet technology design will be undertaken [1]. Moreover, this projections becomes even more impressive if we consider not only "networking" devices (e.g., routers, switches, etc.) inside telecom and home networks, but also the "networked" ones (like common PCs, servers, set-top boxes, smartphones, etc.) [2–4], both in the user homes and in datacenters. In fact, consumer electronics are becoming smarter and even more "connected" through Internet technologies. Despite the proliferation of tablets and mobile phones, many other customer premises equipment are going to massively appear in the user homes (e.g., VoIP phones, set-top boxes, smart household appliances, etc.) [5].

At the same time, the largest part of data used by such devices, and related elaboration/processing, is moved inside powerful datacenters. It is a clear attempt to leverage "smart" devices from complex operations and high-end (energy-hungry) hardware requirements, while at the same time providing end-users with access to applications and data independently from the device and its location. To this purpose, applications are even more designed to be split between end-user and datacenter devices, and connected through the network. This application design trend is obviously causing a significant increase of network traffic, as well as the usage of network hardware and software components inside networked devices. Cisco forecasts that by 2015 Internet traffic will get very close to the impressive threshold of 1 zettabytes a year [5].

Starting from this scenario, it can be argued that the sustainability of the Internet strongly relies on the efficiency of network technologies working at the edges (customer homes and datacenters), and also inside networked devices [6].

To this goal, hardware platforms for networked devices should include advanced power management schemes, proving a certain proportionality between their energy consumption and actual

* Corresponding author. Tel.: +39 010 353 2057; fax: +39 010 353 2154.
E-mail addresses: roberto.bruschi@cnit.it (R. Bruschi), paolo@reti.dist.unige.it (P. Lago), lombardo@dieei.unict.it (A. Lombardo), schembra@dieei.unict.it (G. Schembra).

http://dx.doi.org/10.1016/j.comcom.2014.03.027
0140-3664/

workload. At the same time, the software (and especially the operating system) of such devices needs to be as optimized as possible to be aware of drawbacks of power management schemes, and to meet processing dynamics that are largely triggered by network traffic coming from the network.

It is worth noting that any optimization solutions, that even assure only very small power savings, may have a disruptive impact given the high density of networked hosts. However, the adoption of these power management capabilities in network devices affects both the performance of software applications, and of the Internet traffic, as well as the device energy savings [7].

In this paper, we deal with the efficiency of power management of state-of-the-art networked devices to deeply understand and evaluate their internal dynamics and possible sources of inefficiency. To this purpose, we started by focusing on the experimental characterization of common device platforms based on commercial off-the-shelf hardware and open-source software (i.e., common PCs/servers devices running the Linux operating system). This choice has been mainly driven by two main reasons: (*i*) commercial off-the-shelf hardware can well represent a large segment of technologies applied in networked devices, and already includes advanced power management mechanisms; (*ii*) the Linux operating system is open-source and allows us knowing/customizing every detail of the running software. Moreover, the selected software and hardware platforms are so widespread in consumer electronics, that the largest part of results and considerations can be easily extended in other scenarios (e.g., like in Android smart phones [8]).

The obtained measures allowed us to formalize a Markov based analytical model which is able to accurately capture the power management dynamics at hardware and software levels. In more detail, at the hardware level, ACPI power management primitives have been considered along with their fine-grained features and drawbacks (e.g., delay times in switching clock frequencies and in waking up hardware components, wake-up power consumption peaks, etc.). At the software level, we focused on the power management support of the Linux operating system. The orchestration policy of power management primitives and related operations (i.e., usage monitoring of hardware resources), both included in software modules also known as "Governors," has been accurately modeled.

Then, the proposed model has been applied to analyze the efficiency of the entire system according to various configurations of internal parameters and incoming workload. Thanks to its intrinsic accuracy and the representation of different fine-grained details, the model is useful to evaluate the system behavior in a number of incoming workload scenarios and power management configurations. The model output can provide a large set of precious information, which could be hardly obtained through experimentations. For instance, the model can be adopted to drive the parameter optimization of the Linux Governors, and to design and refine the performance of packet coalescing algorithms. Indeed, coalescing is a common approach at packet [9–13] and task scheduling levels [14–16] for improving processing performance. Despite of the different application scenarios, coalescing techniques are often designed on simple common idea to provide more efficient batch elaboration by sensibly reducing computational overhead. Recently, these kind of techniques have been successfully applied for boosting energy efficiency in network hardware (e.g., IEEE 802.3az interfaces [12]) and in networked devices (e.g., smartphones and tablets [13,16]).

The remainder of this paper is organized as follows. Section 2 introduces the anatomy of the power management subsystem in Linux devices based on commercial off-the-shelf hardware. Section 3 introduces how hardware and software components of the power management system interact among themselves at a fine-grained detail, with a particular reference to the case in which the PC workload is mainly driven by network traffic. Experimental results on the efficiency and dynamics of power management schemes are also shown in Section 3. The formalization of the proposed model is in Section 4, while the derivation of model-related performance indexes is in Section 5. Section 6 reports the performance evaluation results according to Governor parameters and workload levels. The conclusions are drawn in Section 7. Finally, an appendix section includes the description of the tools and instrumentation used to collect the experimental results.

2. The power management anatomy in linux PCs

In off-the-shelf PCs and similar devices (e.g., smart phones, tablets, etc.), the power management system is usually designed across three layers, namely hardware, firmware and software (see Fig. 1). At the *hardware* level, physical components (e.g., CPU, video cards, etc.) include specific mechanisms and solutions for dynamically modulating energy consumption and performance. These mechanisms are usually based on well-known hardware techniques for power management [18], like Dynamic Frequency Scaling (DFS), Dynamic Voltage and Frequency Scaling (DVFS), clock and supply voltage gating, etc. These mechanisms are usually implemented as power management "primitives" that can be enabled, disabled or configured by the software/firmware levels, which owns enough information to decide the most suitable trade-off between performance and energy consumption to meet application and user requirements.

The *firmware* level (i.e., the PC BIOS – Basic Input–Output System) includes the ACPI (Advanced Configuration and Power Interface) framework [19], which provides a standard interface between hardware-dependent power saving techniques and software layers.

Focusing on processors' power saving primitives, ACPI introduces two main types of abstraction for power management mechanisms, namely performance and idle states (*P*- and *C*-states), respectively. P- and C-states offer the possibility to the software level to neglect the specific implementation features and peculiarities of power management techniques at hardware level, and to expose them by means of intuitive functional abstractions.

In more detail, regarding C-states, C_0 is the active power state where the CPU executes instructions, while C_1 through C_n states correspond to low power idle modes, where the processor consumes less power and dissipates less heat. As the idle state becomes deeper ($C_1 \rightarrow C_n$), the transition between the active and an idle state (and vice versa) requires longer time and more energy [20]. In more detail, when a CPU is in the C_1 state, clock signals of the core are disabled. When the CPU is in the C_3 state, in addition to the core clocks turned off, the contents of Level 1 instruction cache as well as the Level 1 and 2 data caches of the processors are powered off. It is worth noting that transition times for moving to and from the C_3 state are significantly longer than in state C_1, since the CPU caches need to be flushed and restored from higher (and slower) memory units [23].

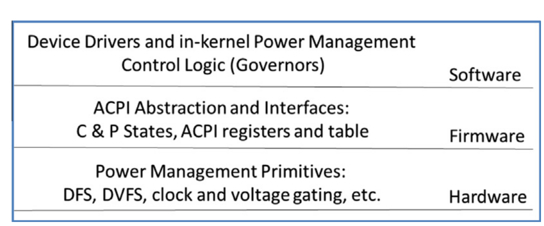

Fig. 1. Architecture and main components of the power management system in off-the-shelf PCs.