



Cloud-based power estimation and power-aware scheduling for embedded systems [☆]



Da-Ren Chen ^{*}, Kai-Feng Chiang

Department of Information Management, National Taichung University of Science and Technology, No. 129, Sec. 3, Sanmin Rd, North Dist., Taichung City 404, Taiwan, ROC

ARTICLE INFO

Article history:

Received 9 November 2014

Received in revised form 22 May 2015

Accepted 3 June 2015

Available online 7 July 2015

Keywords:

Embedded system

Power profiling

Power-aware cluster scheduling

Wireless sensor networks

Cloud computing

Directed acyclic graph

ABSTRACT

Power efficiency is a crucial issue for embedded systems, and effective power profiling and prediction tools are in high demand. This paper presents a cloud-based power profiling (CPP) tool for recording system calls and their associated parameters to predict hardware power consumption when running target applications. Based on hardware power consumption and system profiling from the operating system (OS) kernel, the proposed network model can effectively summarize running behavior of the target applications and the relationship among system calls. This model is also used to develop an energy efficient cluster scheduling for user-inactive processes to reduce the power consumption and extend the service time of embedded systems. These profiling data can be integrated into a cloud model to be maintained by software designers or OS developers to accommodate power estimation and scheduling data for a variety of platforms.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Energy consumption has emerged as one of the most critical issues for embedded devices (e.g., smartphones, wearable devices and vehicle monitor/control systems) and their applications. Recent developments in embedded system design have focused on multi-core architectures which perform applications in parallel, resulting in more complex energy consumption patterns than those found in single-chip devices. Although many improvements have been proposed for the design of low-power hardware, an application-dependent approach to energy management should be considered in higher system levels [1]. For example, a sensor network application that periodically returns sensing data can save a considerable amount of energy by temporarily turning off its radio device when sensing environment or computing data. Modern OSs can profile system calls to provide insight into application execution behavior. Such insight, in turn, can be used to predict and reduce hardware power consumption.

As transistors become more tightly packed, power and heat density levels on single chips operating at very high frequencies become unsustainably high. Therefore, heat and power-efficiency issues have emerged as major challenges for the development of single core processor systems. One solution to address the power and heat barriers in CMOS technology is to reduce the operating frequency and increase the number of computing cores. This provides greater system density and performance per watt as compared to single core processors, and thus allows new levels of performance and scalability

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Danielo Gomes.

^{*} Corresponding author. Tel.: +886 4 2219 6310, +886 933 205261.

E-mail address: danny@nutc.edu.tw (D.-R. Chen).

in computer systems. In addition, multicore processors allow parallel execution of the application threads and hence offer better concurrency. For example, MediaTek[®] clocked an octa-core mobile device SoC (MT6752) at 1.7 GHz, capable of concurrently processing tasks over eight cores [2]. MT6595 is an octa-core SoC with LTE connectivity capabilities, powered by four Cortex-A17 cores with heterogeneous multi-processing capabilities. However, this rapid progress in hardware technology has raised the urgent need for effective scheduling in multicore systems in terms of architecture-side technological advances including multiprocessing, symmetric multithreading, non-uniform memory access (NUMA) [3], virtualization, etc. Overall scheduling fairness such as the workload balance between user responsiveness and overall utilization must be considered. A scheduler needs to consider not only the distribution of threads on cores, but also the distribution of execution load. System calls and hardware-supported instruction profiling, available on modern OS platforms, can help to handle the scheduling problem. They provide similar functions on different platforms, providing information for characterizing process execution behavior. Linux provides rich opportunities for scheduling at the user level, along with information about the system state and workload. System calls provide an indispensable mechanism, allowing modern OSs to recognize most actions taken by applications. This mechanism is situated at an appropriate level of granularity to understand application execution behavior. At the system-call level, applications can be viewed as a set of fine-grained tasks containing a sequence of system calls, which can be used to make power-aware scheduling decisions.

Linux has already been widely applied in smartphones, wireless sensor networks, embedded devices, vehicle infotainment systems, and televisions. For example, Android [3,4] is built on top of the Linux kernel and was deployed in 79.3% of smartphones sold worldwide in 2013 [5]. The Linux kernel handles process control, networking, and peripheral and file system access. Device drivers are either integrated directly with the kernel or added as modules which are loaded while the system is running.

Cloud computing [6] is a model for on-demand network access to a group of configurable resources, such as communication network, storage, computing capability, applications and services. Cloud computing presents a promising business model as it allows users to outsource the provisioning of tasks while ensuring access to up-to-date resources or services on demand. Currently, Google Play[®] and the Apple App Store[®] respectively serve 1.3 and 1.2 million applications. Applications in the public cloud exhibit diverse execution behaviors as well as energy consumption properties. However, it is difficult to predict hardware power consumption based on application-level profiling in a variety of platform and software combinations. A generalized model of computing support for applications could be useful to record and analyze hardware power consumption and application execution behavior.

List scheduling is a standard technique used to schedule tasks with precedence constraints [7,8]. The tasks employed by classic list scheduling are scheduled on m identical processors or without power awareness. In a system with identical processors, each processor has the same specification and runs at identical speeds.

This paper presents the concept for a repository used for recording data about the relationship among system calls, hardware instructions and their power consumption. This repository could be situated in a cloud-based environment and equipped with powerful servers to analyze the recorded data. These data are then processed using a set of cloud-based power profiling tools called CPP to estimate the system power consumption of target applications. These analysis results are also applied to develop an energy-efficient multicore list scheduling algorithm. The proposed power prediction method relies on the proposed tools with a network model and table-based computation along with the repository which analyze the profiling data, and therefore provide improved precision for power prediction. Given numerous and diverse target applications and platforms, the proposed model will include a variety of tables and should be transferred to servers in a cloud computing environment to perform data intensive computing. In this paper we propose:

1. Tools which predict power consumption for target applications by means of profiling (retrieving) their system calls, essential micro instructions and parameters;
2. a cloud computing model for analyzing the profiling data for a variety of application and platform combinations to improve power consumption predictions; and
3. a cluster-based power-aware scheduling for user-interactive processes which organizes system calls using a network model to shorten execution time.

The remainder of this paper is organized as follows: Section 2 reviews the literature on power profiling. The proposed cloud-based model and power profiling are introduced in Section 3. Section 4 introduces a power-aware list scheduling. Section 5 presents the experimental results, and we conclude this paper in Section 6.

2. Literature review

Power profiling can help researchers understand the energy consumption behavior of computer devices and improve their energy efficiency. In general, power profiling is categorized into two classes. As shown in Fig. 1(a), *measurement-based* profiling [9,10] probes device circuits using instruments such as Volt-Ohm-Milliammeters (VOMs), oscilloscopes or data acquisition (DAQ). It is used to record system power consumption corresponding to time instances, and thus analyzes power consumption behavior corresponding to specific program execution. PowerScope [9] proposed by Flinn et al., measures the power consumption of embedded systems while recording system events. Their proposed software profiles the

Download English Version:

<https://daneshyari.com/en/article/455611>

Download Persian Version:

<https://daneshyari.com/article/455611>

[Daneshyari.com](https://daneshyari.com)