The Journal of Systems and Software 127 (2017) 168–176

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Energy minimization for on-line real-time scheduling with reliability awareness

Ming Fan[a,*], Qiushi Han[a], Xiaokun Yang[b]

[a] Broadcom Limited, 3151 Zanker Road, San Jose, CA 95134, United States
[b] Department of Engineering, College of Science and Engineering, University of Houston Clear Lake, 2700 Bay Area Blvd., Houston, TX 77058, United States

ARTICLE INFO

Article history:
Received 10 September 2016
Revised 7 January 2017
Accepted 7 February 2017
Available online 12 February 2017

Keywords:
Energy minimization
Reliability
Multi-core systems
Real-time scheduling

ABSTRACT

Under current development of semiconductor technology, there is an exponential increase in transistor density on a single processing chip. This aggressive transistor integration significantly boosts the computing performance. However, it also results in a power explosion, which immediately decreases the system reliability. Moreover, some well-known power/energy reduction techniques, i.e. Dynamic Voltage and Frequency Scaling (DVFS), can cause adverse impact on system reliability. How to effectively manage the power/energy consumption, meanwhile keep the system reliability under control, is critical for the design of high performance computing systems. In this paper, we present an online power management approach to minimize the energy consumption for single processor real-time scheduling under reliability constraint. We formally prove that the proposed algorithm can guarantee the system reliability requirement. Our simulation results show that, by exploiting the run-time dynamics, the proposed approach can achieve more energy savings over previous work under reliability constraint.

1. Introduction

Embedded computing systems have got a rapid growth in both scale and complexity in the last decade. This advancement is mainly rooted in the development of transistor scaling technology. Today, hundreds of billions of transistors can be integrated into a single chip, which directly results in a boost in the computing performance. However, one key problem, as a consequent of the aggressive scaling in the transistor size, is the huge amount of power increase within a single processing chip. The increased power consumption further poses severe constraints on both design and implementation of computing systems.

Real-time embedded systems, as one type of embedded systems that is dedicated to special applications with real-time constraints in an embedded environment, have been used in a wide range in our daily life. They can be easily found in mobile phones, electronic game devices, motor vehicles, medical equipments, etc. Take mobile phones as an example, these devices have essential restrictions on size, weight, thermal and power/energy. Power/energy is particularly important, as these portable devices largely depend upon the battery-life to deliver high performance and service quality (Zhang et al., 2009). Although computing performance has been continuously increased until today, power/energy issue is more critical in the design of real-time embedded systems.

Dynamic voltage and frequency scaling (DVFS) is one of the most commonly used techniques for power/energy management, which has been well studied in Jejurikar and Gupta (2004) and Yao et al. (1995). With DVFS enabled processors, the supply voltage and frequency are lowered at run-time to achieve energy savings. However, as shown in studies (Zhu and Aydin, 2006; Zhu et al., 2004), DVFS can adversely affect the system reliability. That means, reducing the voltage and operating frequency of a processor exacerbates the reliability problem. For example, it is reported that the transient fault rate occurred in a processor usually increases in several orders of magnitude under low power/frequency condition. Transient fault refers to the temporary malfunction of a processor, usually caused by electromagnetic interferences or cosmic ray radiations, that can lead to temporary errors in computation and corruptions in data (Srinivasan et al., 2004; Shivakumar et al., 2002; Ernst et al., 2004). Moreover, with the increased complexities in both system architecture and applications, the reliability issue is becoming more challenging. As we can see, appropriate real-time scheduling strategies, particularly for embedded system, are desired.

Several researches have been published on reliability-aware power/energy management for real-time embedded systems (Zhu and Aydin, 2006; Zhu et al., 2004; Zhao et al., 2011; Baoxian Zhao and Zhu, 2009; Han et al., 2016; Zheng et al., 2015; Shah et al., 2016). Zhu et al. (2004) proposed a new fault rate model by con-

* Corresponding author.
E-mail addresses: mingfan@broadcom.com (M. Fan), qiushiha@broadcom.com (Q. Han), yangxia@uhcl.edu (X. Yang).

http://dx.doi.org/10.1016/j.jss.2017.02.004
0164-1212/

sidering the frequency effects as well as the execution time together. Zhu and Aydin (2006) applied this fault model to manage the voltage and frequency by reserving backup blocks for specific tasks such that the energy could be minimized and the reliability requirement could be satisfied. Baoxian Zhao and Zhu (2009) further improved this approach by reserving processor resources that can be shared by multiple tasks. Han et al. (2016) presented an approach to pinpoint the peak temperature of a given periodic multicore DVFS schedule. All these approaches suffer a common drawback that task speed assignments are determined statically. In other words, the frequency assignment is predetermined and no run-time information is taken into account.

It is a well-known factor that, in real-time systems, there is usually a large difference between the worst case and the best case execution time for the same real-time task. Therefore the approach that can take advantage of the run-time dynamics can be very effective in saving energy. As a result, we want to study how to employ on-line scheduling techniques to save energy without degrading the system reliability. Specifically, in this paper, we present an on-line reliability-aware dynamic power management approach to schedule frame-based real-time tasks (which share the same deadline but with different execution time) on a single processor platform. The proposed algorithm reduces the energy consumption by dynamically recycling the redundant resources, and based on which, readjusting the frequency for the rest of the workload. Compared with the existing work, we have made a number of contributions:

- First, we made an interesting observation that the system reliability varies with the executions of real-time tasks. By taking the system on-line property into consideration, instead of guaranteeing the system original reliability through off-line approach, we satisfy the reliability requirement through on-line approach. To our best knowledge, this is the first paper that considers the system reliability from on-line perspective.
- Secondly, by recycling the preserved computing resource dynamically, our proposed algorithm can effectively exploit the run-time slacks to adjust the frequencies of real-time tasks such that the system energy consumption can be minimized without compromising the system reliability.
- Thirdly, we conducted extensive experiments to study the performance of our approach, and our experimental results demonstrate that our proposed algorithm can significantly reduce the energy consumption compared with the previous work.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 introduces the preliminary necessary for this paper. Section 4 motivates this research with an example, then formulates the research problem. Section 5 presents the proposed reliability-aware dynamic power management algorithm. Experiments and results are discussed in Section 6, and the conclusion of this work is presented in Section 7.

2. Related work

Many researchers have proposed approaches to dealing with energy related problems with consideration of fault tolerance. Elnozahy et al. (2002) derived a simple theory for power management in the context of duplex and triple modular redundancy systems. In their approach, the recovery back up blocks were used to reserve sufficient time to recover the duplex system from one fault. Unsal et al. (2002) proposed an energy-aware fault-tolerance heuristic, through which the backup tasks were postponed as late as possible such that the overlap between the primary task and its backup task was minimized. Zhang and Chakrabarty (2003) introduced an adaptive checkpointing scheme that dynamically adjusted the checkpointing interval during task execution. They modeled the faults by Poisson distribution, and the fault ratio was depended on the frequency and the amount of time remaining before the task deadline. Han et al. (2013) introduced a method for checkpointing determination to minimizes the worst-case response time for a task set that shares the reserved recoveries on a single processor, and then presented a fault-tolerant task assignment algorithm to minimize the overall energy.

Certain work talked about the power/energy management from the perspective of system reliability instead of the system fault tolerance. Zhu et al. (2004) studied the negative effects of energy management on system reliability and established two models (linear and exponential) to capture the fault rate changes with respect to supply voltages/frequencies. To ensure the reliability requirement, they proposed to reserve processor resource for recovery tasks in case transient faults occur. Then based on that model, work (Zhu and Aydin, 2006) proposed an energy management approach by reserving backup blocks for DVFS scaled tasks to maintain the system reliability requirement. This approach is further improved in work (Baoxian Zhao and Zhu, 2009), in which each reserved block can be shared by different tasks, thus more power-efficient voltage scale can be applied among all tasks. Niu et al. (2013) proposed a reliability-aware energy minimization scheme under certain window-constraints such that within any non-overlapped job sequence the system reliability is guaranteed. Zhao et al. (2012) proposed a task-level reliability model by taking the negative effects of DVFS on transient fault rate into consideration, and developed a single processor scheduling algorithm to minimize energy consumption under reliability restriction. Li et al. (2015) introduced a method to satisfy both applications reliability and deadline requirements by combining checkpointing, Dynamic Voltage Frequency Scaling (DVFS) and backward fault recovery techniques into consideration. Later, they extend their work by adding energy as one more objective, and developed a task scheduling algorithm to minimizes energy consumption under guaranteed systems reliability and deadline constraints (Li et al., 2016). All these approaches suffer a common drawback that task speed assignments are determined statically. The frequencies assignment to tasks is predetermined and no run-time information is taken into account.

3. Preliminary

In this section, we first introduce the system models used in this paper, which include task model, fault model, reliability model, power and energy models. Then we use an example to motivate our research.

3.1. The real-time model

The real-time system applications considered in this paper consists of N independent tasks, denoted as $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_N\}$. All tasks in Γ have the same deadline D, but with different execution requirements. We denote the execution time of τ_i as c_i. For the rest of this paper, we assume that Γ is sorted with increasing execution time. We also assume that all tasks within Γ are released at time 0, and share the same relative deadline after releasing.

We consider the uniprocessor system with DVFS operation capability. Assume that the available discrete frequencies can vary from the minimum value f_{min} to the maximum value f_{max}, and all frequency values are normalized with respect to f_{max}, (i.e. $f_{max} = 1$). Specifically, let $\mathcal{F}$ denote the discrete frequency set which consisting of L different frequencies, i.e. $\mathcal{F} = \{f_1, f_2, \ldots, f_L\}$, where $f_i < f_j$ if $1 \le i < j \le L$. The execution time of τ_i under the frequency f_i is given by c_i/f_i.