The Journal of Systems and Software 86 (2013) 2611–2619

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Power-aware scheduling algorithms for sporadic tasks in real-time systems

Yi-wen Zhang [a,b,*], Rui-feng Guo [a]

[a] Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China
[b] University of Chinese Academy of Sciences, Beijing 100039, China

ARTICLE INFO

Article history:
Received 8 March 2013
Received in revised form 25 April 2013
Accepted 25 April 2013
Available online 14 May 2013

Keywords:
Sporadic task
Dynamic voltage scaling
Real-time system

ABSTRACT

In this paper, we consider the canonical sporadic task model with the system-wide energy management problem. Our solution uses a generalized power model, in which the static power and the dynamic power are considered. We present a static solution to schedule the sporadic task set, assuming worst-case execution time for each sporadic tasks release, and propose a dynamic solution to reclaim the slacks left by the earlier completion of tasks than their worst-case estimations. The experimental results show that the proposed static algorithm can reduce the energy consumption by 20.63%–89.70% over the EDF* algorithm and the dynamic algorithm consumes 2.06%–24.89% less energy than that of the existing DVS algorithm.

1. Introduction

Power management is of primary importance in the operation of real-time systems, which can be attributed to longer battery life, reliability and packaging costs. The processor power is mainly from the CMOS circuits. It consists of the dynamic power (mainly due to switching activities) and the static power (mainly due to leakage current). Dynamic voltage scaling (DVS) aims at reducing the power consumption of the system by operating the processor at a lower frequency and thus on a lower voltage.

There are many substantial researches for scheduling real-time applications on DVS processors (Zhao et al., 2012; Jejurikar et al., 2004; Gong et al., 2007; Niu and Li, 2011; Aydin et al., 2001; Shin and Kim, 2006; Wang et al., 2012; Pillai and Shin, 2001; Jejurikar et al., 2005; Yao et al., 1995; Kim et al., 2002; Aydin et al., 2006; Mei et al., 2013; Chen and Qian, 2012; Zhuo and Chakrabarti, 2008). These approaches differ in many aspects, such as the scheduling considering periodic tasks/mix tasks, being static/dynamic, assuming worse-case execution time/best-case execution time, using RM policy/EDF policy, and using traditional DVS strategy/the critical speed strategy (Zhao et al., 2012; Jejurikar et al., 2004, 2005; Niu and Li, 2011). However, they still have common objective which reduces the energy consumption while meeting the timing constraints of real-time tasks. DVS uses slack time to adjust the processor speed. Many previous researches have been conducted regarding slack time analysis to schedule the real time task set (Gong et al., 2007; Niu and Li, 2011; Aydin et al., 2001; Shin and Kim, 2006; Wang et al., 2012; Pillai and Shin, 2001; Jejurikar et al., 2005; Kim et al., 2002; Aydin et al., 2006). Gong et al. (2007) have proposed the on-line scheduling method for the mixed task set, considering the dynamic power. Both Shin and Kim (2006) and Wang et al. (2012) have proposed a dynamic reclaiming algorithm oriented to the mixed tasks with the preemptive EDF scheduling policy. In addition, many previous researches have focused on scheduling method for the periodic tasks (Niu and Li, 2011; Aydin et al., 2001; Pillai and Shin, 2001; Jejurikar et al., 2005; Kim et al., 2002; Aydin et al., 2006; Wu and Kao, 2012; Tchamgoue et al., 2012). Aydin et al. (2001) have proposed a dynamic reclaiming scheduling algorithm oriented to the periodic tasks, when a higher-priority task instance completes early, it reclaims the slack time and scales down the processor speed. Pillai and Shin (2001) have proposed a new scheduling algorithm (Cycle-Conserving EDF) and a speculation-based algorithm (Look-Ahead EDF), assuming periodic tasks. Jejurikar et al. (2005) use the procrastination technology to reclaim the slack time of the higher-priority tasks which completes early. Kim et al. (2002) have proposed a dynamic scheduling algorithm to reclaim the slack time from the already completed higher-priority tasks as well as from the lower-priority tasks. Aydin et al. (2006) have proposed the scheduling algorithm with the preemptive EDF policy, considering a generalized power model. Niu and Li (2011) have provided a DVS algorithm under the RM policy. The CATSA algorithm has been proposed in the Wu and Kao (2012), to schedule real-time tasks with abortable critical sections in a non-ideal dynamic voltage scaling processor. Tchamgoue et al. (2012) have defined a new problem for power-aware scheduling in hierarchical framework with periodic resource model

* Corresponding author at: Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China. Tel.: +86 18698868960.
E-mail addresses: zywsy2010@gmail.com (Y.-w. Zhang), grf@sict.ac.cn (R.-f. Guo).

0164-1212/$ – see front matter
http://dx.doi.org/10.1016/j.jss.2013.04.075

and provide optimal task-level and component-level static DVS schemes.

Recently, the interest of the research focuses on the multiprocessor system (Chowdhury and Chakrabarti, 2005; Kim et al., 2007; Kumar and Palani, 2012). Chowdhury and Chakrabarti (2005) have proposed an efficient heuristic algorithm using a charge-based cost function derived from the analytical battery model. The power-aware scheduling algorithms for bag-of-tasks applications with deadline constraints on DVS-enabled cluster systems was presented in the Kim et al. (2007). Kumar and Palani (2012) have proposed a novel scheduling algorithm using the genetic algorithm to schedule the tasks and then find the optimal power supplies and determine the schedule length on the multiprocessor system.

The previous real time scheduling methods for sporadic tasks do not take the DVS technology, i.e. the tasks always execute with the maximum speed, leading to the high energy consumption. In order to decrease the energy consumption, there are many real time scheduling algorithms with DVS technology for sporadic tasks (Qadi et al., 2003; Mei et al., 2013; Huang et al., 2008; Zhong and Xu, 2007). Qadi et al. (2003) have proposed a DVS algorithm oriented to the sporadic tasks, assuming worst-case execution times and dynamic power. Zhong and Xu (2007) proposed a time-variant voltage scaling (TV-DVS) algorithm, which is more energy efficient than that of DVSST (Qadi et al., 2003); however, TV-DVS cannot meet the deadlines of tasks. Mei et al. (2013) proposed CC-DVSST algorithm which is an improvement to DVSST can meet the deadlines of tasks; however it only considers the dynamic power model. All in all, many real time scheduling algorithms have been proposed for periodic and aperiodic task models but none support the canonical sporadic task model with a generalized power model.

In this work, we propose a static algorithm oriented to the canonical sporadic task model while taking a generalized power model into consideration. In addition, we propose a dynamic scheme that utilizes dynamic slack for energy savings. To the best of our knowledge, this is the first work that addresses a dynamic scheme that utilizes dynamic slack for the canonical sporadic task model while considering a generalized power model.

The rest of the paper is organized as follows. The models and notations description are presented in Section 2. The static sporadic tasks scheduling algorithm is proposed and analyzed in Section 3 and Section 4 presents the dynamic sporadic tasks scheduling algorithm. Conclusions are presented in Section 5.

2. Model and notation

2.1. System model and notation

We consider a set of independent sporadic tasks $T=\{T_1, T_2, \ldots, T_n\}$ that are to be executed on a uni-processor system according to the EDF* scheduling policy. EDF* is the same as EDF (earliest deadline first), except that, among tasks whose deadlines are the same, the task with the earliest arrival time has the highest priority (FIFO policy); in case that both deadlines and arrival times are equal, the task with the lowest index has the highest priority. Each sporadic task T_i has three associated parameters, p_i, e_i and d_i. p_i is the minimum separation period between the release of two consecutive instances of a task; e_i is the worst-case execution time of the instance. d_i is the relative deadline for the instance. In this paper, we assume $d_i = p_i$. Thus, T_i can be described using the tuple (p_i, e_i). The ready time of T_i is denoted by r_i. The jth instance of task T_i is denoted by $T_{i,j}$. The ready time of $T_{i,j}$ is denoted by $r_{i,j}$. The actual execution time of T_i will be denoted by ae_i. We assume a variable speed (DVS-enabled) processor whose speed (frequency) S can vary between a lower bound $S_{\min}$ and an upper bound $S_{\max}$. For convenience, we normalize the CPU speed with respect to $S_{\max}$; that is, we assume that $S_{\max} = 1.0$. We denote U_{tot} as the total utilization of the task set under the maximum speed $S_{\max}$, and assume that $U_{tot} = \sum_{i=1}^{n} e_i/p_i \leq 1$.

2.2. Power model

In this paper, we adopt a generalized form of the system-level power model which was proposed by Zhao et al. (2012). In our general system-level power model, the power consumption P is given by

$$P = P_s + h(P_{ind} + P_{dep}) = P_s + h(P_{ind} + C_{ef}S^m) \tag{1}$$

Here, P_s is the static power, which includes the power to maintain basic circuits and keep the clock running and can be removed only by powering off the whole system. Due to the prohibitive overhead of turning off/on a system in sporadic real-time execution settings, we assume that the system is in on state at all times and that P_s always consumes and is not manageable. Different P_s will not affect the absolute energy savings and, for simplicity, we will ignore the static power P_s (i.e., $P_s = 0$). P_{ind} is the frequency-independent active power, which includes any active power that does not depend on running speed and can be effectively removed by putting the system to sleep. P_{dep} is the frequency-dependent active power. The effective switching capacitance C_{ef}, and the dynamic power exponent m (in general, $2 \leq m \leq 3$) are system/application dependent constants. The coefficient h is 1 when the system actively executes a task; otherwise, $h = 0$.

To obtain the maximal reduction in the overall energy consumption, the critical speed strategy is proposed by Zhao et al. (2012), Jejurikar et al. (2004), Niu and Li (2011), and Jejurikar et al. (2005), which is a minimal energy-efficient speed. From this model, one can derive the minimum energy-efficient speed value as $S_{crit} = \sqrt[m]{P_{ind}/(C_{ef} \cdot (m-1))}$ (Zhao et al., 2012). That is, for energy-efficiency, no job should be executed at a speed lower than S_{crit} as doing so would result in higher energy consumption. We develop our framework by assuming the continuous speed. The implications of having discrete speed levels can be solved by the two adjacent frequency levels policy (Gong et al., 2007) or the next higher discrete speed. Thus, we do not discuss it in this paper.

3. Static sporadic tasks scheduling algorithm

In this section, we present the static sporadic tasks scheduling algorithm (SSTSA) to the variable voltage scheduling problem for the sporadic task model, assuming that each task presents its worst-case work-load to the processor at every instance. A motivational example will be discussed in Section 3.1. We present the SSTSA algorithm in Section 3.2 and discuss the simulation results in Section 3.3.

3.1. A motivational example

This section presents an example to demonstrate the shortcoming of the EDF* algorithm without DVS technology. We consider a 3-task system T with the following parameters: (4, 2), (5, 1), (10, 3). The system utilization is $U_{tot} = 1$. Let us consider scheduling the instances that were released in the interval [0,20) under the preemptive EDF* algorithm. We assume that the tasks released instances as follows: T_1 at times 0, 5, 10, and 17; T_2 at times 0, 7, and 13; T_3 at times 4 and 15. According to (Niu and Li, 2011), we assume $S_{crit} = 0.3$ and $S_{\min} = 0.1$, the power consumption function can be modeled approximately as $P = 0.08 + 1.52{}^*S^3$, and the