The Journal of Systems and Software 90 (2014) 128–137

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Power-aware fixed priority scheduling for sporadic tasks in hard real-time systems

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

[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 13 September 2013
Received in revised form
16 December 2013
Accepted 17 December 2013
Available online 7 January 2014

Keywords:
Sporadic task
Energy management
Real-time system

ABSTRACT

In this paper, we consider the generalized power model in which the focus is the dynamic power and the static power, and we study the problem of the canonical sporadic task scheduling based on the rate-monotonic (RM) scheme. Moreover, we combine with the dynamic voltage scaling (DVS) and dynamic power management (DPM). We present a static low power sporadic tasks scheduling algorithm (SSTLPSA), assuming that each task presents its worst-case work-load to the processor at every instance. In addition, a more energy efficient approach called a dynamic low power sporadic tasks scheduling algorithm (DSTLPSA) is proposed, based on reclaiming the dynamic slack and adjusting the speed of other tasks on-the-fly in order to reduce energy consumption while still meeting the deadlines. The experimental results show that the SSTLPSA algorithm consumes 26.55–38.67% less energy than that of the RM algorithm and the DSTLPSA algorithm reduces the energy consumption up to 18.38–30.51% over the existing DVS algorithm.

1. Introduction

The energy management remains an important problem for computer systems, and in particular, for real-time embedded systems. For CMOS circuits, two major sources of power consumption are the dynamic power (mainly due to switching activities) and the static power (mainly due to leakage current). Dynamic Voltage Scaling (DVS) and Dynamic Power Management (DPM) are two major techniques for reducing power dissipation in such systems. DVS adjusts the clock frequency and supply voltage based on the workload of systems. DPM aims to turn off system hardware that is not currently in use.

There are significant studies on DVS-enable real-time system for energy savings in the past decade. Yao et al. (1995) provided a static off-line scheduling algorithm, assuming aperiodic tasks and worst-case execution times. Pillai and Shin (2001) have proposed a reclaiming algorithm (Cycle-Conserving EDF) and a speculation-based algorithm (Look-Ahead EDF). They are based on updating and predicting the instantaneous utilization of the periodic task set. Aydin et al. (2001) have proposed a dynamic reclaiming algorithm oriented to the periodic tasks, which contains the off-line algorithm and the on-line algorithm. The off-line algorithm computes the optimal static speed. When a higher-priority task instance completes early, the on-line algorithm reclaims the slack time and scales down the processor speed. Kim et al. (2002) have proposed a novel DVS algorithm for periodic hard real-time tasks based on an improved slack estimation. It reclaims the slack time from the already completed higher-priority tasks as well as from the lower-priority tasks.

All of the above approaches only consider the dynamic power and ignore the static power. Jejurikar et al. (2004, 2005) proposed a critical speed policy to solve the system-level power problem. Aydin et al. (2006) have proposed a DVS algorithm that considers a generalized power model which consists of both the dynamic power and the static power. Procrastination scheduling and DVS are combined to reduce the energy consumption for the periodic task set in literature (Jejurikar et al., 2005). In addition, a fixed priority DVS algorithm is proposed in literature (Niu and Li, 2011). Kwon et al. (2013) have proposed an application-level energy-efficient scheduling approach for smart-phones operating systems. A new problem for power-aware scheduling in hierarchical framework with periodic resource model has been proposed in literature (Tchamgoue et al., 2012).

In fact, the real time system is a mixed task system, that is, there are periodic tasks and aperiodic tasks. The case of mixed real-time task scheduling on variable-speed processors was explored in literature (Wang et al., 2012; Lee and Shin, 2004; Gong et al., 2007; Shin and Kim, 2006; Ranvijay et al., 2009). Lee and Shin

[*] 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 © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.12.032

(2004) have proposed an on-line DVS algorithm called OLDVS that does not assume task periodicity, nor does it require any a priori information on the task set to be scheduled, but it only considers the dynamic power. Gong et al. (2007) have extended the OLDVS algorithm (Lee and Shin, 2004) to the processor with discrete frequency and voltage levels. In addition, Shin and Kim (2006) have proposed the two-phase approach to solve the mixed task scheduling problem. The proposed algorithms utilize the execution behaviors of scheduling servers for aperiodic tasks.

Recently, the interest of the research focuses on the multiprocessor system (Shieh and Pong, 2013; Kumar and Palani, 2012; Ansari et al., 2013; Singh et al., 2013; Geeraerts et al., 2013). Shieh and Pong (2013) aim to resolve this scheduling problem with voltage transition overhead consideration in multiprocessor and formalize this problem by an integer linear programming model and propose a heuristic algorithm for a runtime environment. Ansari et al. (Ansari et al., 2013) have proposed power aware scheduling of fixed priority tasks in soft real-time multicore systems scheduling algorithm (PASSRTMS). The PASSRTMS is a combination of offline and online scheduling with DVFS to schedule fixed priority tasks on soft real-time multicore systems. Singh et al. (2013) have proposed a methodology that applies DVFS for such cyclic dependent tasks in multiprocessor system. In addition, Geeraerts et al. (2013) have analyzed the schedulability problem of hard real-time, sporadic, arbitrary deadline and preemptive task systems upon identical multiprocessor platforms, and have used antichain techniques to prove to be efficient on problems with the same complexity.

All of the above approaches about low-power task scheduling focus only on reducing the CPU power by using DVS techniques. There are also researcher focusing on combining the DVS and DPM to reduce the energy consumption of the CPU and devices (Shin and Choi, 1999; Devadas and Aydin, 2012; Rong and Pedram, 2006; Zeng et al., 2008; Terzopoulos and Karatza, 2013). Shin and Choi (1999) have proposed a power efficient version of a widely used fixed priority scheduling method which DVS and DPM are combined. Rong and Pedram (2006) have addressed the problem of minimizing energy consumption of a computer system performing periodic hard real-time tasks with precedence constraints used dynamic priority scheduling. Terzopoulos and Karatza (2013) have studied energy gains that come from the application of two popular energy saving techniques, DVS and DPM, in a real-time 2-level heterogeneous grid system.

In this paper, we focus on a scheduling approach for the sporadic tasks set in the uni-processor system. The case of sporadic task scheduling on variable-speed processors was explored in literature (Qadi et al., 2003; Zhang and Guo, 2013; Zhong and Xu, 2007; Mei et al., 2013; Huang et al., 2008). A DVS algorithm, called DVSST, is presented in literature (Qadi et al., 2003), DVSST can be used with sporadic tasks in conjunction with preemptive EDF scheduling, assuming worst-case execution times and dynamic power. Zhong and Xu (2007) have proposed an analytical model of general tasks for DVS assuming job timing information is known only after a task release and designed the two efficient scaling algorithms, called TV-DVS, 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) have proposed CC-DVSST algorithm based on updating and predicting the instantaneous utilization of the sporadic task set, however it ignores the static power. Zhang and Guo (2013) have proposed a more efficient algorithm oriented to the canonical sporadic task model while taking a generalized power model into consideration, using the EDF scheduling scheme.

To the best of our knowledge, this is the first work that addresses a fixed priority scheduling approach for the canonical sporadic task model while considering a generalized power model. We proposed a static sporadic tasks low power scheduling algorithm, called SSTLPSA, assuming each task presents its worst-case work-load to the processor at every instance, and a dynamic sporadic tasks scheduling algorithm, called, DSTLPSA, combining the DVS and DPM technology to reduce the energy consumption. The experimental results show that the SSTLPSA algorithm consumes 26.55–38.67% less energy than that of the RM algorithm and the DSTLPSA algorithm reduces the energy consumption up to 18.38–30.51% over the existing DVS algorithm.

The rest of the paper is organized as follows. In Section 2, we introduce the preliminaries. We present a static sporadic tasks low power scheduling algorithm in Section 3. We present a dynamic sporadic tasks low power scheduling algorithm in Section 4 and conclude with the summary in Section 5.

2. Preliminaries

2.1. System model

The system consists of a task set of n sporadic real time tasks, represented as $T = \{T_1, T_2, \ldots, T_n\}$ where tasks are assumed to be mutually independent. A task T_i is a 3-tuple (e_i, p_i, d_i) (Mok A.K.-L., 1983) where e_i is the worst case execution time (WCET) of the task at the maximum processor speed, p_i is the minimum separation period between the release of two consecutive instances of a task, d_i is relative deadline. In this paper, we assume $d_i = p_i$ for all tasks. Thus, each task can be described using the tuple (e_i, p_i). We denote $T_{i,j}$ as the jth instance of task T_i. The ready time and the actual execution time of T_i are denoted by r_i and ae_i, respectively.

The variable speed (DVS-enabled) processor used in our system can be operate at a set of speed $[S_{\min}, S_{\max}]$. We normalize the processor speed by $S_{\max}$, that is, the speed set is $[S_{\min}, 1]$. The total utilization of the task set under the maximum speed $S_{\max}$ is denoted by $U_{tot} = \sum_{i=1}^{n} e_i/p_i$. In addition, we assume that the execution time of a task scales linearly with the processing speed. That is, at speed S_i, the execution time of task T_i is assumed to be $e_i(S_{\max}/S_i)$.

The sporadic task set is scheduled under the RM (Liu and Layland, 1973) scheduling policy. RM is a fixed priority scheduling algorithm where the shorter the period task, the higher the priority. It is assumed that tasks are arranged in the decreasing order of priorities according to RM, that is $p_1 < p_2 < \ldots < p_n$. According to (Liu and Layland, 1973), the feasibility condition states that a task set is schedulable under preemptive RM if the system utilization U_{tot} does not exceed the Liu–Layland bound (LLB) given by

$$U_{tot} \le n(2^{1/n} - 1) = LLB(n) \quad (1)$$

where n is the number of tasks. In this paper, we assume $U_{tot} \le LLB(n)$.

2.2. Power model

We adopt the system-level power model where the power consumption of the computing system considered is given by (Zhao et al., 2012; Zhang and Guo, 2013):

$$P = P_s + h(P_{ind} + P_{dep}) = P_s + h(P_{ind} + C_{ef}S^m) \quad (2)$$

Despite its simplicity, this power model captures the essential components for the system-wide energy management. Here, P_s is the static power, P_{ind} is the frequency-independent active power, P_{dep} is the frequency-dependent active power. C_{ef} is the effective switching capacitance, S is the running speed of the task, and m (in general, $2 \le m \le 3$) is the dynamic power exponent which is system/application dependent constants. The coefficient h is 1 when the system actively executes a task; otherwise, $h = 0$.

By setting the derivative of equation 2 to zero, a minimal energy-efficient speed S_{crit}, below which DVFS starts to consume more total