The Journal of Systems and Software 112 (2016) 1–10

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Reliability guaranteed energy minimization on mixed-criticality systems

Zheng Li[a,*], Chunhui Guo[b], Xiayu Hua[b], Shangping Ren[b]

[a] Western Illinois University, Macomb, IL 61455, USA
[b] Illinois Institute of Technology, Chicago, IL 60616, USA

ARTICLE INFO

Article history:
Received 4 February 2015
Revised 16 October 2015
Accepted 17 October 2015
Available online 27 October 2015

Keywords:
Mixed-criticality
Reliability
Energy

ABSTRACT

This paper studies the energy minimization problem in mixed-criticality systems that have stringent reliability and deadline constraints. We first analyze the resource demand of a mixed-criticality task set that has both reliability and deadline requirements. Based on the analysis, we present a heuristic task scheduling algorithm that minimizes system's energy consumption and at the same time also guarantees system's reliability and deadline constraints. Extensive experiments are conducted to evaluate and validate the performance of the proposed algorithm. The empirical results show that the algorithm further improves energy saving by up to 10% compared with the approaches proposed in our earlier work.

1. Introduction

For real-time embedded systems, to further reduce system cost, more and more tasks with different functionality and of different levels of criticality are integrated on the same hardware platform (Burns and Davis, 2013; Ekberg and Yi, 2014). Examples include recent unmanned aerial vehicles (UAV) which integrate the HI-criticality tasks such as flight-control tasks and LO-criticality tasks such as photo capturing tasks on the same platform (Barhorst et al., 2009). Though the mixed-criticality design paradigm reduces system cost, tasks of different criticalities compete for the shared resource and cause system's timing behaviors become less predictable.

For a safety-critical system, high-criticality tasks are more crucial to the entire system than low-criticality tasks. To ensure that HI-criticality tasks always meet their deadlines, two worst case execution times are set for each HI-criticality task, i.e., *worst case execution time by design* and a more pessimistic one, *worst case execution time by certification*. When tasks' actual execution time is no more than their designed worst case execution time, the system is considered operating under the LO-mode. However, if any task executes beyond this limit, the system changes to the HI-mode immediately to signal that a situation beyond designed behaviors has occurred and some actions need to take place. A mixed-criticality system is schedulable if the following two conditions are satisfied (Ekberg and Yi, 2012): 1) both LO-criticality and HI-criticality tasks are guaranteed to meet their deadlines under the LO-mode; and 2) HI-criticality tasks are also guaranteed to meet their deadlines under the HI-mode. Determining whether a given mixed-criticality system is schedulable has been proven to be NP-hard (Baruah et al., 2012a) and different heuristic approaches are developed to addressing the schedulability issue.

In addition to guarantee task deadlines, power/energy efficiency and reliability issues are also critical for real-time embedded systems. As more and more transistors are integrated into a single chip, operation power/energy consumption of the chip has increased exponentially. Dynamic Voltage and Frequency Scaling (DVFS) technique, which dynamically lowers down the supply voltage and working frequency, is widely used for power/energy management. However, existing work (Zhu et al., 2004; Niu and Xu, 2015) has shown that transient fault rate increases when the supply voltage on the chip scales down. In other words, lowering down system's supply voltage potentially degrades the system's reliability. Hence, minimizing system's energy consumption without sacrificing the reliability requirement is another design challenge.

In this paper, we study how to schedule a mixed-criticality task set to minimize system's energy consumption under the following constraints:

1. schedulability constraint: both HI-criticality and LO-criticality tasks are guaranteed to meet their deadlines under LO-mode, and HI-criticality tasks are guaranteed to meet their deadlines under HI-mode;
2. reliability constraint: both HI-criticality and LO-criticality tasks are guaranteed to meet their reliability constraints under LO-mode, and HI-criticality tasks are guaranteed to meet their reliability constraints under HI-mode.

* Corresponding author. Tel. +1312-6509688.
E-mail addresses: z-li2@wiu.edu (Z. Li), cguo13@iit.edu (C. Guo), xhua@iit.edu (X. Hua), ren@iit.edu (S. Ren).

http://dx.doi.org/10.1016/j.jss.2015.10.029
0164-1212/

The main contributions of this paper are three-fold:

1. theoretically analyze the resource demand of mixed-criticality task set under both reliability and schedulability constraints;
2. develop a heuristic search based frequency assignment (HSFA) algorithm that decides the lowest task execution frequency which guarantees both deadline and reliability constraints;
3. empirically evaluate and compare the energy saving performance of the HSFA algorithm with the state-of-the-art approaches in the literature.

The rest of the paper is organized as follows. We first discuss related work in Section 2 and then present the models and definitions the paper is based upon in Section 3. The research problem to be addressed in the paper is formally defined in Section 4. The theoretical foundations are established in Section 5. Based on the theoretical analysis, we present a heuristic search based frequency assignment algorithm in Section 6. Experimental results are discussed in Section 7 and finally we conclude our work in Section 8.

2. Related work

The study of mixed-criticality task set scheduling issue started in the recent years. Baruah and Vestal (2008) first proposed to apply earliest deadline first (EDF) scheduling theory in mixed-criticality task set scheduling. To ensure the schedulability of a mixed-criticality task set, the earliest deadline first with virtual deadline (EDF-VD) scheduling algorithm was proposed (Baruah et al., 2012b). The EDF-VD algorithm assigns HI-criticality tasks reduced deadlines to ensure that HI-criticality tasks are schedulable even if they overrun their normal worst case execution time. Ekberg and Yi (2012); 2014) proposed a greedy approach by using the demand-bound function analysis to determine a task set's schedulability under EDF algorithm. Ekberg's greedy algorithm has a significant improvement over the EDF-VD, but it has higher time complexity. However, both Baruah's EDD-VD and Ekberg's greedy algorithm take the approach of terminating all LO-criticality tasks if any instance of HI-criticality task overruns its normal worst case execution time, i.e., when system enters into the HI-mode.

To provide a guaranteed minimum level of service to LO-criticality tasks when system enters into the HI-mode, Su and Zhu (2013) considered using elastic task models (Buttazzo et al., 1998) to increase LO-criticality tasks' period and hence reduce their competition against HI-criticality tasks but allow LO-criticality tasks to execute when possible. By noticing that postponing HI-criticality tasks' execution can promote earlier execution of LO-criticality tasks, Park and Kim (2011) developed a scheme called criticality based EDF which delays the execution of HI-criticality tasks as late as possible but without causing deadline violations.

Energy saving and reliability are two other major concerns for real-time embedded systems. For single criticality real-time systems, i.e. all tasks in the system have the same criticality, Zhu (2011) proposed a reliability-aware power management scheme which aims to minimize energy consumption while at the same time maintain system's reliability at the same level as if all tasks were executed with the highest processing frequency. Zhao et al. later improved the approach and developed a shared recovery technique that allows all tasks to share the same reserved recovery block, but only allows a single fault recovery during the entire task set execution (Zhao et al., 2009). Zhao et al. (2011) further extended the work and developed a generalized shared recovery technique which removed single recovery constraint and allow multiple fault recoveries by reserving multiple recovery blocks.

Recently, the study of energy saving and reliability issues in the context of mixed-criticality task sets has drawn increased attention. With the objective to minimize system's energy consumption and at the same time guarantee task deadlines, Huang et al. (2014) presented an approach by utilizing DVFS and EDF-VD techniques (Baruah et al., 2012b) to solve the problem. To satisfy system's reliability requirements, Axer et al. (2011) developed an approach to tolerating transient faults by duplicating high criticality tasks. Pathan (2014) also proposed a fixed-priority scheduling algorithm to tolerate transient faults in mixed-criticality systems.

As we can see, the aforementioned work treats reliability and energy consumption issues independently. However, system reliability and energy minimization are correlated when DVFS is used. In particular, reduced processing frequency reduces system energy consumption, but at the same time, reduced processing frequency also increases transient failure rate. Hence, they need to be addressed conjointly. In this paper, different with Huang et al. (2014) work that reliability constraint is not taken into consideration, we are to address the problem about how to schedule mixed-criticality task sets to minimize system's energy consumption while at the same time satisfying *both* deadline and reliability constraints.

3. Models and definitions

In this section, we introduce the models and definitions the research is based upon.

3.1. Models

3.1.1. Processor model

The processor is DVFS enabled with a finite set of available frequencies, i.e. $F = \{f_1, \ldots, f_q\}$. The frequency values in F are in a descending order with $f_1 = f_{\max}$ and $f_q = f_{\min}$. These frequencies are normalized with respected to $f_{\max}$, i.e., $f_{\max} = 1$.

3.1.2. Task model

In this paper, we make the same assumptions as in Ekberg and Yi (2014), Baruah et al. (2012b), i.e. there are two different criticality levels in a task set. In particular, for a given mixed-criticality task set $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$, each task τ_i is defined by a quadruple as $\tau_i = (L_i, C_i, T_i, D_i)$, where

- T_i is the task's period,
- D_i is the task's relative deadline and we assume $D_i \le T_i$,
- $L_i \in \{\text{LO}, \text{HI}\}$ is the task's criticality level,
- $C_i = \{C_i(\text{LO}), C_i(\text{HI})\}$ is task's worst-case execution time and $C_i(\chi)$ is the worst case execution time at criticality level χ under maximum processing frequency $f_{\max}$. If τ_i is a HI-criticality task, $C_i(\text{LO}) \le C_i(\text{HI})$; while if τ_i is a LO-criticality task, $C_i(\text{LO}) = C_i(\text{HI})$.

The LO-criticality subset and the HI-criticality subset are denoted as Γ_L and Γ_H, respectively.

3.1.3. Transient fault model

Although both permanent and transient faults may occur during task execution, transient faults are found more frequent than permanent faults (Niu and Xu, 2015; Guo et al., 2013). Hence, in this paper, we focus on transient faults. We adapt the same assumption as in the literature that the transient fault rate follows Poisson distribution with an average fault rate λ (Zhu, 2011; Zhao et al., 2011; Li et al., 2013). When a system is running under frequency f_i, the average transient fault rate is expressed as

$$\lambda(f_i) = \hat{\lambda}_0 10^{-\hat{d} f_i}, \quad (1)$$

where $\hat{\lambda}_0 = \lambda_0 10^{\frac{d}{1-f_{\min}}}$, $\hat{d} = \frac{d}{1-f_{\min}}$, and λ_0 is the average fault arrival rate when system running under the maximum frequency $f_{\max}$. The value $d\,(>0)$ is a system-dependent constant, which indicates the sensitivity of the system's fault arrival rate to system voltage and frequency scaling, the larger the d value is, the more sensitive the fault arrival rate to voltage and frequency scaling.