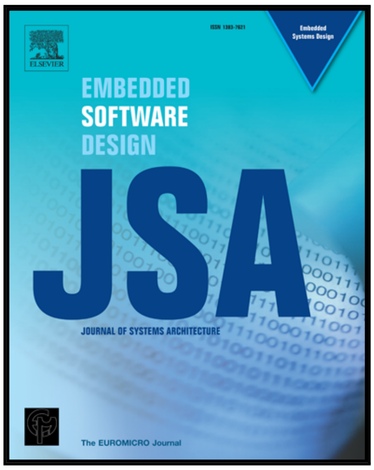
Accepted Manuscript



Exploit Primary/Backup Mechanism for Energy Efficiency in Dependable Real-Time Systems

Yifeng Guo, Dakai Zhu, Hakan Aydin, Jian-Jun Han, Laurence T. Yang

PII: S1383-7621(16)30262-4
DOI: 10.1016/j.sysarc.2017.06.008
Reference: SYSARC 1443

To appear in: *Journal of Systems Architecture*

Received date: 17 December 2016
Revised date: 20 March 2017
Accepted date: 29 June 2017

Please cite this article as: Yifeng Guo, Dakai Zhu, Hakan Aydin, Jian-Jun Han, Laurence T. Yang, Exploit Primary/Backup Mechanism for Energy Efficiency in Dependable Real-Time Systems, *Journal of Systems Architecture* (2017), doi: 10.1016/j.sysarc.2017.06.008

Exploit Primary/Backup Mechanism for Energy Efficiency in Dependable Real-Time Systems

Yifeng Guo, Dakai Zhu, Hakan Aydin, Jian-Jun Han and Laurence T. Yang

Abstract—Primary/Backup has been well studied as an effective fault-tolerance technique. In this paper, with the objectives of tolerating a single *permanent* fault and maintaining system reliability with respect to *transient* faults, we study dynamic-priority based energy-efficient fault-tolerance scheduling algorithms for periodic real-time tasks running on multiprocessor systems by exploiting the primary/backup technique while considering the negative effects of the widely deployed *Dynamic Voltage and Frequency Scaling (DVFS)* on transient faults. Specifically, by separating primary and backup tasks on their dedicated processors, we first devise two schemes based on the idea of *Standby-Sparing (SS)*: For *Paired-SS*, processors are organized as groups of two (i.e., pairs) and the existing SS scheme is applied within each pair of processors after partitioning tasks to the pairs. In *Generalized-SS*, processors are divided into two groups (of potentially different sizes), which are denoted as *primary* and *secondary* processor groups, respectively. The main (backup) tasks are scheduled on the primary (secondary) processor group under the *partitioned-EDF* (*partitioned-EDL*) with DVFS (DPM) to save energy. Moreover, we propose schemes that allocate primary and backup tasks in a *mixed* manner to better utilize system slack on all processors for more energy savings. On each processor, the *Preference-Oriented Earliest Deadline (POED)* scheduler is adopted to run primary tasks at scaled frequencies *as soon as possible (ASAP)* and backup tasks at the maximum frequency *as late as possible (ALAP)* to save energy. Our empirical evaluations show that, for systems with a given number of processors, there normally exists a configuration for Generalized-SS with different number of processors in primary and backup groups, which leads to better energy savings when compared to that of the Paired-SS scheme. Moreover, the POED-based schemes normally have more stable performance and can achieve better energy savings.

Index Terms—Real-Time Systems; Multiprocessor; Fault Tolerance; Primary/Backup; Energy Management; DVFS; DPM;

1 Introduction

Fault tolerance has been a traditional research topic in real-time systems as computing devices are subject to different types of faults at runtime. In general, to tolerate various faults and guarantee that real-time tasks can complete their executions successfully on time, the existing fault tolerance techniques normally adopt different forms of redundancy. For instance, as a simple and well-studied approach, *hot-standby* exploits *hardware/modular* redundancy and runs two copies of the same task *concurrently* on two processors to tolerate a single fault [33]. However, by their very nature, such redundancy-based fault-tolerance techniques demand more system resources, which can lead to excessive energy consumption (e.g., hot-standby has 100% energy overhead).

On the other hand, with the ever-increasing power density in modern computing systems, energy has been promoted as a first-class system resource and energy-aware computing has become an important research area [24]. As a common energy saving technique, *dynamic power management (DPM)* can power down (or turn off) components when they are not in use. Moreover, as a fine-grained power management technique, *dynamic voltage and frequency scaling (DVFS)* can operate computing systems at different low-performance (and thus low-power) states when the performance demand is not at the peak level by simultaneously scaling down their supply voltage and processing frequency [35].

Although both redundancy-based fault tolerance [6], [17] and DPM/DVFS-based energy management schemes [35], [52] have been independently studied extensively, the co-management of system reliability and energy consumption has caught researchers' attention only very recently [14], [31], [38]. Note that, fault tolerance and energy efficiency are normally conflicting design objectives in computing systems since redundancy generally results in increased energy consumption [2]. Moreover, recent studies show that DVFS has a negative effect on system reliability due to significantly increased transient fault rates at low supply voltages [11], [15], [49]. With this intriguing interplay between fault tolerance and energy efficiency, it becomes imperative to develop effective techniques that can address both dimensions while guaranteeing the timeliness of real-time tasks.

By taking the negative effects of DVFS on transient fault rates into consideration, a series of *reliability-aware power-management (RAPM)* schemes have been studied for various real-time task models based on the backward recovery technique [13], [19], [29], [34], [45], [46], [47], [48]. Basically, RAPM exploits system slack (i.e., *temporal redundancy*) for both reliability preservation and energy savings. RAPM ensures to schedule a recovery task before scaling down the processing frequency of any task, using the remaining slack time. By executing the recovery task at the maximum frequency, RAPM can achieve a desired system reliability level even if the task with scaled frequency incurs transient faults [29], [45], [48]. Although RAPM can guarantee system reliability with respect to transient faults (which were shown to be more common [26]), it does not offer provisions for tolerating permanent faults.

- *Y. Guo and D. Zhu are with Department of Computer Science, The University of Texas at San Antonio, San Antonio, TX 78249, USA.*
- *H. Aydin is with Department of Computer Science, George Mason University, Fairfax, VA 22030, USA.*
- *J. Han is with School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China.*
- *L. T. Yang is with Department of Computer Science, St. Francis Xavier University, Antigonish, NS, B2G 2W5, Canada.*