Journal of Systems Architecture 61 (2015) 269–281

Contents lists available at ScienceDirect

Journal of Systems Architecture

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

Compositional power-aware real-time scheduling with discrete frequency levels

Guy Martin Tchamgoue [a,b], Junho Seo [a], Kyong Hoon Kim [a,*], Yong-Kee Jun [a]

[a] Department of Informatics, Gyeongsang National University, 501 Jinju-daero, Jinju 660-701, South Korea
[b] Department of Electrical and Computer Engineering, University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 26 July 2014
Received in revised form 23 April 2015
Accepted 23 May 2015
Available online 29 May 2015

Keywords:
Compositional real-time scheduling
Periodic resource model
Periodic task model
Power-aware scheduling
DVFS
Discrete frequencies

ABSTRACT

Power consumption remains a hot issue in all areas of computing ranging from embedded systems that rely on batteries to large scale data centers where reducing the power consumption of computing devices directly affects not only the management cost, but also contributes to a greener computing environment. The power-aware real-time scheduling problem has recently been addressed for a compositional framework with periodic task model under the assumption that a processor can continuously vary its operating frequency and voltage. However, in practice, this technique is only suboptimal and still produce the waste of computational resources. This paper introduces new frequency scaling schemes that statically determine optimal processor speeds at system, component, and task levels with the objective of minimizing the total energy consumption of the entire framework. Since real-world processors support only a finite set of operating frequencies, our algorithms also consider only discrete speed levels and guarantee still that each task meets its deadline. We implemented and evaluated the performance of a prototype framework that incorporates our algorithms on top of the RT-Xen hypervisor in order to provide power-aware compositional real-time scheduling framework to virtual machines.

1. Introduction

As communities and governments are now fighting to reduce the world carbon footprint, power-aware computing became more than necessary. Power consumption then remains a hot issue in all areas of computing ranging from embedded systems that rely on batteries to large scale data centers where reducing the energy consumption directly affects not only the management cost, but also contributes to a greener computing environment. One of the most commonly used techniques for reducing the power consumption in modern computing systems is based on the well-known Dynamic Voltage and Frequency Scaling (DVFS) scheme which allows to dynamically scale both the voltage and operating frequency of processors. On a DVFS-enabled processor, a reduced power consumption can easily be achieved whenever the full processing speed is not required by simply scaling down the operating frequency of the processor. However, a direct consequence of this speed reduction is that tasks take longer to complete.

Component-based design has been widely accepted as methodology to build large and complex embedded real-time systems by composing a system through systematic abstraction and composition [25,26,29]. With component-based design, a large system is broken into smaller subsystems referred to as *components*, allowing the reduction of a single complex design problem into multiple simpler design problems, and composing components into a flexible new system through component interfaces that abstract and hide the internal complexity of each component. This design model also facilitates the reuse of components that may now be developed in totally different environments.

In a *Compositional Real-Time Scheduling (CRTS) framework*, the main problem has been to define a scheduling interface model in order to specify the collective resource and deadline requirements of a component. Many frameworks and resource models [4,9,21,25,26,29] have been proposed for a variety of real-time task models. The periodic resource model [26] for instance was introduced for compositional real-time scheduling frameworks dealing with the periodic task model. The periodic resource model therefore describes the behavior of a periodic resource and calculates its minimum resource allocations in a time interval. A periodic resource $\Gamma(\Pi, \Theta)$ guarantees to supply at each resource period Π exactly Θ computational time units to a component.

To reduce the power consumption of compositional real-time scheduling frameworks, few work has addressed the real-time DVFS (RT-DVFS) scheduling problem. Tchamgoue et al. [28]

* Corresponding author.
E-mail addresses: guymt@ymail.com (G.M. Tchamgoue), joy2net@gnu.ac.kr (J. Seo), khkim@gnu.ac.kr (K.H. Kim), jun@gnu.ac.kr (Y.-K. Jun).

http://dx.doi.org/10.1016/j.sysarc.2015.05.003
1383-7621/

introduced static and dynamic scheduling algorithms that determine optimal processor speeds at system, component, and task levels to minimize the total power consumption with real-time guarantee for all tasks. Provided that the resource period and capacity are known for each component, existing algorithms yield optimal speed levels assuming that the processor can continuously adjust its operating frequency at runtime. However, in practice, processors only support a finite set of operating frequencies. Thus, assuming continuous speed adjustment obviously still causes a waste of computational resources due to the fact that whenever the computed optimal speed level is not supported by the underlying processor, the operating speed level must be set to the nearest greater available value, thus under-utilizing the system resources [16].

Assuming a compositional real-time scheduling framework with periodic task model and the periodic resource model, this paper concentrates on the RT-DVFS scheduling problem with discrete speed levels to minimize the power consumption of the framework while guaranteeing still its real-time requirements. Thus, the problem is to determine the optimal power-aware resource supply, hence the optimal power-aware component interface, that guarantees both the schedulability of the framework and its minimal energy consumption. The optimal speed level can therefore be determined either at *system level* for the whole framework, or at *component level* to guarantee the feasibility of each component, or finally at *task level* to guarantee the feasibility of each single task. For this purpose, we statically determine the optimal resource supply to each component for each supported processor speed level, and then select only the speed levels that minimize the total energy consumption and maximize the resource utilization, while guaranteeing the deadline requirements of the framework. In order to assign optimal speed levels, we formulate and solve an optimization problem at each scheduling level.

At component and task levels, solving the optimization problems requires the exploration of a large combinatorial space of solutions and was proved to be NP-Hard. Thus, to cope with this high computational cost, we propose a simple branch-and-bound algorithm that statically determines the processor speed levels in a computationally acceptable amount of time. We simulated the proposed schemes to validate their effectiveness. More, we implemented and evaluated the performance of a prototype of the new scheduling framework on top of the RT-Xen hypervisor to support the power-aware real-time scheduling of virtual machines.

The remainder of this paper is organized as follows. Section 2 presents our system, power, resource, and interface models. Section 3 describes a motivating example that supports the paper, defines and contextualizes our scheduling problem and presents the design architecture of our framework. Section 4 focuses on the proposed schemes for determining optimal processor speed levels. Section 5 presents our simulation results. Section 6 provides an overview of our prototype implementation inside the RT-Xen hypervisor. Related work is described in Section 7, and our conclusion and future work finally come in Section 8.

2. System and power models

This section provides an overview of the compositional real-time scheduling framework in our system and gives details on the resource, interface, and power models we consider in this paper.

2.1. Compositional real-time scheduling framework

In a *compositional* scheduling framework [26,29], a *component* is the basic scheduling unit and is defined by $C(W,A)$, where W is the workload and A the scheduling algorithm of the component. Components are organized in a tree-like hierarchy where a upper-layer component allocates resources to its child components, as shown in Fig. 1.

In this paper, we assume a compositional scheduling framework for periodic task model [26], although this results can easily be extended to other task models by selecting the appropriate resource demand and supply bound functions. Thus, the workload W of each component $C(W,A)$ consists of n periodic real-time tasks. Each task τ_i is defined by (p_i, e_i), where p_i represents the period of the task and e_i its worst-case execution time. A task τ_i releases a new job every p_i time units. Therefore, the j-th job of task τ_i should be finished by the next job's release time. Each component of the framework is abstracted through its interface and seeing by its upper-layer component as a single real-time task. This abstraction allows the upper-layer component to schedule its child component without any consideration of their internal real-time requirements. In Fig. 1, for example, two tasks of component C_1, scheduled with Earliest Deadline First (EDF), are abstracted into a new periodic task $I_1(10,3)$. Similarly, component C_2, which is scheduled using the Rate Monotonic (RM) policy, is converted into $I_1(15,4)$. The upper-layer component C_0 then only focuses on efficiently scheduling these two tasks, providing them with the appropriate resource supply.

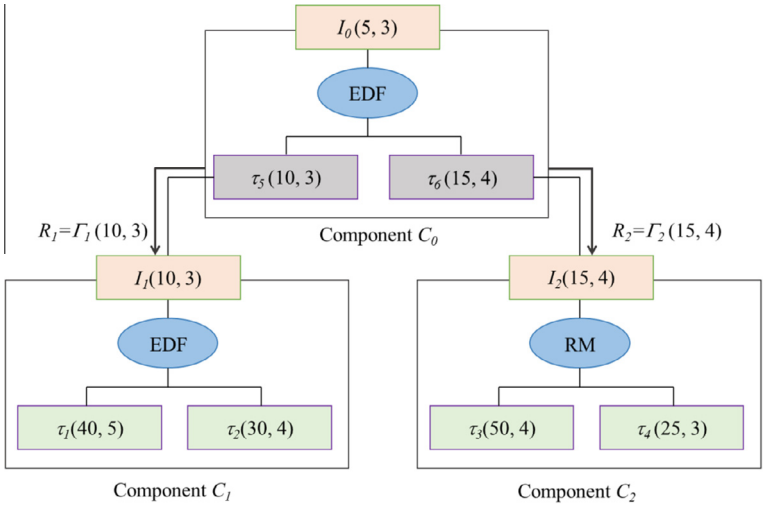

Fig. 1. An example of compositional real-time scheduling framework.