Future Generation Computer Systems 56 (2016) 220–228

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Energy-aware task migration for multiprocessor real-time systems

Gang Zeng[a,*], Yutaka Matsubara[b], Hiroyuki Tomiyama[c], Hiroaki Takada[b]

[a] Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
[b] Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
[c] College of Sciences and Engineering, Ritsumeikan University, 1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan

ARTICLE INFO

Article history:
Received 15 January 2015
Received in revised form
6 June 2015
Accepted 18 July 2015
Available online 14 August 2015

Keywords:
Real-time systems
Energy optimization
Multiprocessor
Task migration
Dynamic voltage scaling

ABSTRACT

A task migration method is proposed for energy savings in multiprocessor real-time systems. The method is based on the portioned scheduling technique which classifies each task as a fixed task or a migratable task. The basic task migration problem with specific parameters is formulated as a linear programming problem to minimize average power. Then, the method is extended to more general case with a complete migration algorithm. Moreover, a scheduling algorithm is proposed for migratable tasks. Simulation results on two processor models demonstrated significant energy savings over existing methods.

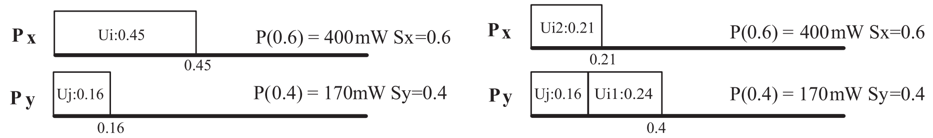
1. Introduction

Power consumption has become one of the major concerns in today's embedded system design. Reducing power or energy consumption can extend battery lifetime of portable systems, decrease chip cooling costs, and increase system reliability. As one of the most effective methods for energy savings, the dynamic voltage scaling (DVS) has been extensively studied on uniprocessor so far. However, the power problem still exists, which limits the further improvement of performance on the uniprocessor. As a promising approach to achieve higher performance and lower power consumption, multiprocessors have attracted much attention recently. Meanwhile, the energy-aware real-time task scheduling on multiprocessors is becoming an important issue. There are mainly two approaches to schedule real-time tasks on the multiprocessor: global scheduling and partitioned scheduling. The global scheme uses a global scheduler for all processors to assign tasks to the processor online, whereas the partitioned scheme uses a dedicated scheduler for each processor, and tasks are assigned to particular processors offline. Generally, the global scheme permits task migration between processors when dispatching a task. Conversely, normal partitioned scheme does not support task migration. Once a task is assigned to a processor, it cannot run on any other processors. In current real-time multiprocessor systems, the partitioned scheme is common because of its simplicity and ease of implementation [1,2]. In other words, it reduces a multiprocessor scheduling problem to a set of uniprocessor ones. For this reason, this work is targeted at the partitioned scheduling technique.

Several studies have been made on the energy-aware partitioned scheduling in the DVS-equipped multiprocessor systems. It was proven that the total energy consumption of the multiprocessor is minimized under the earliest-deadline-first (EDF) scheduling in the case that the workload is perfectly balanced among the processor cores [3,4]. Because the problem of energy-aware task scheduling is NP-hard [1], several heuristics were employed in the literature. It was known that the worst-fit decreasing (WFD) heuristics can achieve more energy savings than other heuristics because it leads to more balanced workload than other heuristics [1]. In fact, it has been proven that the WFD algorithm can achieve 1.5-approximation than the optimal algorithm for load balancing problem [5]. Approximate algorithm based on WFD with 2-approximation for the energy-efficient scheduling problem was also introduced in [6] by considering leakage energy. However, the above methods assume an ideal processor with continuously selectable speeds and zero idle power. As a result, for real processors with discrete speeds and nonzero idle power, these methods cannot achieve the maximal energy savings [7]. A significant advantage of the above algorithms is that the DVS overhead is avoided completely because they assign one speed for all tasks allocated on the same core. In contrast, algorithms for assigning different

[*] Corresponding author.
E-mail addresses: sogo@ertl.jp (G. Zeng), yutaka@ertl.jp (Y. Matsubara), ht@fc.ritsumei.ac.jp (H. Tomiyama), hiro@ertl.jp (H. Takada).

http://dx.doi.org/10.1016/j.future.2015.07.008



(a) Results of no migration: average power of multiprocessor = 368 mW.

(b) Results of migration: average power of multiprocessor = 310 mW.

Fig. 1. Motivational example for energy-aware task migration on XScale in case $P_{idle} = 0$ mW.

speeds to the tasks on the same core have also been proposed in [8–10]. However, the DVS overhead cannot be ignored in these algorithms due to frequently runtime voltage/frequency (V/F) transition. As summarized in the [11], commercial DVS uniprocessor chips typically require hundreds of microseconds for each V/F transition.

Recently, task migration between processors was recognized as an effective method for improving the worst-case system utilization bound on the partitioned scheduling [12,2]. For example, the Earliest Deadline Deferrable Portion (EDDP) method can improve the worst-case bound from 50% to 65% via task migration [2]. In addition, task migration was also proposed for thermal management [13]. Moreover, a workload aware task migration algorithm was proposed to reduce energy consumption, which is based on a global scheduling algorithm, and migrates a task from high speed processor to low speed one based on actual task execution time [14]. As mentioned earlier, this paper focus on the partitioned scheduling algorithm and assumes known worst case execution time for scheduling, which is significantly different with existing migration algorithm. In this paper we show that even for partitioned scheduling, there are opportunities for energy savings via task migration. As an example, Fig. 1 shows the potential for energy savings in the partitioned scheduling via task migration in which the width of rectangle indicates the magnitude of task utilization. In this example, although the utilizations of two tasks are only 0.45 and 0.16, higher speeds of 0.6 and 0.4 have to be used due to deadline constraints and discrete speeds. As can be seen, some energy has been wasted because even if the processor P_y has left capacity of 0.24 for running task at the low speed of 0.4, the large task has to be run at high speed of 0.6 on the processor P_x if task cannot be migrated. For this reason, if we can split the large task i into two portions, and migrate partial utilization of task i from the high speed processor to the low speed one, total energy can be saved. According to our approach, energy can be reduced by 16% when migrating 0.24 utilization of task i from processor P_x to processor P_y.

In this work, we propose algorithms to solve the above task migration problem for maximal energy savings on multiprocessor real-time systems. The approach includes two phases. In the design time phase, tasks are allocated to processors first without migration, then migration parameters are calculated and stored by using the proposed migration algorithm if energy can be saved via migration. In the runtime phase, a modified EDF based scheduler is employed to schedule the processor with migratable task according to the stored migration parameters. The main contributions of this paper are as follows. (1) A portioned based task migration model is proposed. The model has limited migration cost because task migration is applied only in case that energy can be saved, and the maximal number of migratable tasks is limited to $m/2$ where m is the number of processor cores. (2) For a selected task for migration, the energy-aware task migration problem is formulated as a linear programming problem and a novel optimal solution is proposed. Moreover, the above results are extended to more general case with a complete migration algorithm. (3) Energy savings are evaluated using simulation on a synthetic multiprocessor. The results showed that the proposed migration method can achieve better energy reduction than existing methods.

The rest of the paper is organized as follows. Section 2 introduces the system models. Section 3 presents the proposed migration algorithms. Section 4 gives experimental results. Finally, Section 5 concludes the paper.

2. System models and algorithms

2.1. Processor power model

Although many existing publications on energy-aware multiprocessor scheduling assumed a multiprocessor with dynamic core-level V/F scaling (DCVFS) capability, few commercial or even test embedded multiprocessors chip with this capability can be found in practice. This status was also declared in the [16]. The possible reasons include the increased design complexity, test cost, and reliability problem especially for the embedded systems. Recently, it has been reported that the ARM11 MPcore based test chip can provide static chip-level V/F scaling functionality [17], and Renesas and Waseda's RP2 test chip supports chip-level DVS, and core-level frequency scaling (but not voltage scaling) [18]. Consider the above reality, multiprocessor with DCVFS or static core-level V/F scaling (SCVFS) capability is assumed as the target multiprocessors in this work. The SCVFS means that the V/F can only be set offline, and they cannot be changed online. This type of multiprocessor could be an alternate of DCVFS with lower design complexity.

Homogeneous multiprocessor with m identical DVS-equipped processor cores is considered in this work. Without loss of generality, each processor core is supposed to have three power modes, i.e., run, idle, and inactive modes. Among three modes, the programs can only be run in the run mode. Generally, when the processor has no task to execute before the next task releases, OS can put the processor in an idle mode, which consumes less power than the run mode. If no task is allocated to a processor, the processor can be in the inactive mode and consumes zero power. The processor in run or idle mode is referred to as active state because both modes consume static power. The power consumption of processor in run mode is mainly determined by its operating frequency f, which are some discrete values within the range $[f_{\min}, f_{\max}]$.

Table 1 shows two commercial processor power models in which the speed rate S is the normalized frequency with respect to $f_{\max}$. The power values including both dynamic and static power are obtained by actual measurement using benchmark programs. The power model of XScale is quoted from the literature, and it has been employed in [19,20]. The DSP power model is based upon our measured results on a reference TMS320VC5509A DSP board [21]. While the two processors represent typical embedded processor with DVS capability, they have different features. As pointed out in [19], the XScale has ideal energy efficiency with respect to DVS which can be represented by a cubic or quadratic function of the speed rate. In contrast, the measured DSP has low energy efficiency. For the purpose of comparison, we calculate the ideal power of DSP by assuming a quadratic function of speed rate and