Theoretical Computer Science 595 (2015) 46–64

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Online dynamic power management with hard real-time guarantees ☆

Jian-Jia Chen [a], Mong-Jen Kao [b,*], D.T. Lee [b,1], Ignaz Rutter [a], Dorothea Wagner [a]

[a] Faculty for Informatics, Karlsruhe Institute of Technology (KIT), Germany
[b] Institute for Information Science, Academia Sinica, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 12 April 2015
Received in revised form 1 June 2015
Accepted 4 June 2015
Available online 10 June 2015
Communicated by A. Marchetti-Spaccamela

Keywords:
Online energy-efficient scheduling
Dynamic power management

ABSTRACT

We consider the problem of online dynamic power management that provides hard real-time guarantees for multi-processor systems. In this problem, a set of jobs, each associated with an arrival time, a deadline, and an execution time, arrives to the system in an online fashion. The objective is to compute a non-migrative preemptive schedule of the jobs and a sequence of power on/off operations of the processors so as to minimize the total energy consumption while ensuring that all the deadlines of the jobs are met. We assume that we can use as many processors as necessary. In this paper we examine the complexity of this problem and provide online strategies that lead to practical energy-efficient solutions for real-time multi-processor systems.

First, we consider the case for which we know in advance that the set of jobs can be scheduled feasibly on a single processor. We show that, even in this case, the competitive ratio of any online algorithm is at least 2.06. On the other hand, we give a 4-competitive online algorithm that uses at most two processors. For jobs with unit execution times, the competitive ratio of this algorithm improves to 3.59.

Second, we relax our assumption by considering as input multiple streams of jobs, each of which can be scheduled feasibly on a single processor. We present a trade-off between the energy-efficiency of the schedule and the number of processors to be used. More specifically, for k given job streams and h processors with $h > k$, we give a scheduling strategy such that the energy usage is at most $4 \cdot \left\lceil \frac{k}{h-k} \right\rceil$ times that used by any schedule which schedules each of the k streams on a separate processor. Finally, we drop the assumptions on the input set of jobs. We show that the competitive ratio of any online algorithm is at least 2.28, even for the case of unit job execution times for which we further derive an $O(1)$-competitive algorithm.

1. Introduction

Reducing power consumption and improving energy efficiency have become important design requirements in computing systems. For mobile devices, effective power management can considerably extend the standby period and prolong battery

☆ An extended abstract of this work appeared in the 31st Symposium on Theoretical Aspects of Computer Science, STACS'14, Lyon, France.

* Corresponding author.
E-mail addresses: j.chen@kit.edu (J.-J. Chen), mong@iis.sinica.edu.tw (M.-J. Kao), dtlee@ieee.org (D.T. Lee), rutter@kit.edu (I. Rutter), dorothea.wagner@kit.edu (D. Wagner).
[1] Also with the Department of Computer Science and Information Engineering, National Chung-Hsing University, Tai-Chung, Taiwan.

http://dx.doi.org/10.1016/j.tcs.2015.06.014
0304-3975/

lifetime. For large-scale computing clusters, appropriately powering down the idling processing units can considerably reduce the electricity bill.

In order to increase the energy efficiency, two different mechanisms have been introduced to reduce the energy consumption. *(1) Power-down Mechanism:* When a processor is idling, it can be put into a low-power state, e.g., sleep or power-off. While the processor consumes less power in these states, a fixed amount of energy is required to switch the system back to work. In the literature, the problem of deciding the sequence of state transitions is referred to as *dynamic power management (DPM)*. *(2) Dynamic Speed Scaling:* The concept of dynamic speed scaling refers to the flexibility provided by a processor to adjust its processing speed dynamically. The rate of energy consumption is typically described by a convex function of the processing speed. This feature is also referred to as *dynamic voltage frequency scaling (DVFS)*, following its practical implementation scheme.

The majority of previous work regarding energy-efficient scheduling focuses mainly on uni-processor systems. For systems that support a power-down mechanism, Baptiste [7] considered hard real-time jobs, i.e., deadline misses of jobs are not allowed, with unit execution times and proposed the first polynomial-time algorithm that computes an optimal strategy for turning on and powering off a processor. In a follow-up paper, Baptiste et al. [8] further extended the result to jobs with arbitrary execution times and reduced the time complexity. When considering workload-conserving scheduling, i.e., the system is not allowed to enter low-power states when the ready queue is not empty, Augustine et al. [5] considered systems with multiple low-power states and provided online algorithms.

Dynamic speed scaling was introduced to allow computing systems to reach a balance between high performance and low power consumption dynamically. Hence, scheduling algorithms that assume dynamic speed scaling, e.g., Yao et al. [32], usually execute jobs as slowly as possible while ensuring that timing constraints are met. When the energy required to keep the processor active is not negligible, however, executing jobs too slowly may result in more energy consumption. For most of the realistic power-consumption functions, there exists a *critical speed*, which is the most energy-efficient for job execution [14,24].

Irani et al. [24] initiated the study of combining both mechanisms. For offline energy-minimization, they presented a 2-approximation. For the online version, they introduced a *greedy procrastinating principle*, which enables online algorithms that have certain properties and that are designed for speed scaling without power-down mechanism to additionally support the power-down mechanism. The idea behind this principle is to postpone job execution as much as possible in order to bundle workload for batch execution. The usage of job procrastination with dynamic speed scaling for periodic tasks has later been explored extensively in a series of studies [13,28,14].

The combination of the power-down mechanism with dynamic speed scaling suggests the philosophy of *racing-to-idle*: Execute jobs at higher speeds and gain longer quality sleeping intervals. Albers and Antoniadis [1] showed that the problem of minimizing the energy consumption for speed scaling with a sleep state is NP-hard and provided a $\frac{4}{3}$-approximation. Recently, this result was further improved by Antoniadis et al. [4] to a fully-polynomial time approximation scheme.

All of the aforementioned work mainly focuses on uni-processor systems. By contrast, for multi-processor systems, relatively fewer results are known. Demaine et al. [19] considered unit jobs and presented a polynomial-time algorithm based on dynamic programming for power-down mechanism. Approximations for several variations were also presented. In a follow-up paper, Demaine and Zadimoghaddam [20] presented logarithmic approximations for general formulations of scheduling problems with submodular objective functions, including energy consumption. Albers et al. [2] considered dynamic speed scaling with job migration and presented polynomial-time algorithms based on maximum flow problems.

As scheduling to meet deadline constraints is a long-standing difficult problem [17,16,18,22], additional augmentations on the hardware level, e.g., speed of the processors or number of the machines, have been considered to provide practical solutions. See, for example, [12,10,3,30]. In practice, machine augmentation follows the trends in multi-core systems, while speed augmentation has been shown its limits as overclocking is difficult to achieve due to the dramatic increase of power consumption and thermal dissipation.

Our focus and contribution. In this paper, we examine the problem of online dynamic power management that provides hard real-time guarantees, i.e., each job must finish its execution before its deadline, for multi-processor systems. We assume a system equipped with multiple processors that are identical and that operate independently from each other, and we can use as many processors as necessary. We assume that job executions can be preempted but cannot be migrated since job migration is often unaffordable in real-time system. That is, the execution of a job must be done on the same processor. The objective is to compute a schedule of the jobs and a sequence of switch on/off operations of the processors so as to minimize the total energy consumption. For this problem model we give an elaborate study that leads to practical energy-efficient solutions for real-time multi-processor systems.

First, we consider the case for which we know in advance that the set of jobs can be scheduled feasibly on one processor. We show that the competitive ratio of any online algorithm is at least 2.06, even for this restricted case. Then we propose the idea of *energy-efficient anchors*, which are defined for each of the jobs, to indicate a proper moment for which the online scheduler should no longer postpone the execution of the jobs. We show that this idea leads to a 4-competitive online algorithm which uses at most two processors. For jobs with unit execution times, we show that the competitive ratio improves to 3.59.

Second, we relax the conditions of our assumption by considering as input multiple streams of jobs, each of which can be scheduled feasibly on one processor. We present a simple strategy, as a byproduct of our first algorithm, to allow a