Discrete Applied Mathematics 182 (2015) 150–161

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Solving a power-aware scheduling problem by grouping jobs with the same processing characteristic

Rafał Różycki *, Jan Węglarz

Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965, Poznań, Poland

A R T I C L E   I N F O

Article history:
Received 31 January 2013
Received in revised form 13 September 2013
Accepted 10 November 2013
Available online 12 December 2013

Keywords:
Power-aware scheduling
Doubly constrained resource
Modeling computational jobs

A B S T R A C T

In the paper a power-aware problem of scheduling preemptable jobs on parallel identical machines to minimize the schedule length is considered. Exact approaches utilizing the idea of grouping jobs are presented and compared from the viewpoint of the size of an appropriate non-linear programming problem.

1. Introduction

This paper deals with a scheduling problem from the green computing area, where the main idea is to find a good balance between a performance of computing and a consumption of natural resources. Since during the execution of a computer program (a computational job), energy is consumed as a main resource, appropriate power management is a basic technique for applying the green computing principles. It is obvious that energy consumption and computing performance (measured, e.g., as a completion time or a flow time of a set of computational jobs) are in opposition. There are two ways to find a good compromise. In the first one, the aim is to improve a given performance measure assuming some level of available energy, whereas in the second, the energy consumption is reduced supposing that the given value of a performance measure is maintained. The first problem, known as a *laptop problem* [3], is typical for portable electronic devices driven by electric energy accumulated in batteries with limited capacity. In the second problem (so-called *server problem*) the goal is to minimize the energy consumption in order, for example, to reduce the overall computing cost.

From a technological point of view, green computing is based on *variable speed processors* (VSP) and proper operating systems. A VSP is a kind of processor designed to dynamically control the speed of computation in order to manage the energy consumption. An operating system of a computer equipped with a VSP can decide not only which job to perform at the moment but also at what processing rate. Thus, scheduling rules implemented in the operating system should take into account the relation between the processing speed and electrical power used to process a computational job. Energy consumption of a computer system may be controlled by setting an appropriate speed of a processor for a computational job since this relation is non-linear. Otherwise, if the relation was linear, the processing rate would have no influence on the amount of energy consumed by the system.

* Corresponding author. Tel.: +48 616653025; fax: +48 618771525.
E-mail address: Rafal.Rozycki@cs.put.poznan.pl (R. Różycki).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.11.003

One of the research directions towards energy-aware computer systems is to model the problem of scheduling computational jobs as a deterministic scheduling problem. The relation between processing time (or speed) of a job and power used for its execution may be formalized by using different models of job processing. The power vs. processing speed model has been widely used up to now in the energy-aware scheduling problems [3,4]. The same *power usage function* characterizes the execution of all computational jobs in the model. The form of power usage function results from the features of most popular processors built on the base of CMOS technology, and is assumed to be a cubic function of its processing speed. Consequently, the jobs differ with a size—the parameter expressing an amount of computation (given e.g. by the number of CPU cycles) needed to complete a job.

On the other hand, there exist some experimental architectures of processors which allow even more accurate management of power/energy consumption during computation. Thus, it is advisable to use the job models which are more suitable for differentiating the characteristics of a computational job execution.

In this paper we use a dynamic *processing rate function* model originally developed for scheduling problems where the processing rate of a job depends on a temporal amount of a single continuous renewable resource [7,9,8]. This model has been consecutively extended to the case of a doubly constrained, continuous resource [10]. It has many advantages in comparison to the power usage model. They are enumerated in short in the next section.

The model with such a processing rate function may be applied to the general problems of power-aware scheduling of computational jobs. Notice that limited amounts of power and energy available in the laptop problem may be treated as a single, doubly constrained resource that computational jobs compete for. On the other hand, a power usage (with a known energy limit) or energy consumption (with a limited power level) may be minimized assuming a given level of a computer system performance in the case of the server problem. Whatever type of problem we solve, it is natural to extend it to the practical multicore processor case. It is justified by the fact that multicore processors become available even in the portable electronic devices. The processor cores share a common source of power to process a set of jobs.

Using a terminology of the deterministic scheduling theory, a multicore power-aware scheduling problem may be modeled as a parallel machine scheduling problem of preemptable jobs with an additional doubly constrained continuous resource, where each job is characterized by its processing rate function. The approach for solving the laptop problem with dependent jobs, parallel identical machines and schedule length minimization is described in [5], whereas the server problems of power or energy minimization with independent jobs and known due dates in [6]. In each case the best schedule is achieved by an optimal power allocation among combinations of jobs preliminary sequenced in the way that always leads to the optimal solution. Unfortunately, the number of variables of the non-linear programming problem which has to be solved grows exponentially with the size of the problem instance. As a consequence, such a numerical approach is intractable for realistic problem instances.

In this paper we present alternative approaches to solve the power-aware scheduling problem. One direction towards more efficient ways of finding optimal schedules is to identify special cases of the problem for which polynomial exact algorithms exist. We present such an algorithm for the case of identical job processing rate functions. Moreover, we propose an approach that may be attractive from the numerical point of view for the case of instances where the number of different job processing rate functions is relatively small in comparison with the number of jobs. We show that this approach may be further simplified if an instance of the problem fulfills some additional conditions. All the approaches and algorithms are demonstrated basing on an exemplary laptop problem of scheduling independent preemptive jobs on parallel identical machines with makespan as a scheduling criterion.

In Section 2 the considered scheduling problem is formulated. Section 3 recalls the idea of the general approach for solving the problem. In Section 4 a polynomial algorithm for solving a special case of the problem is presented. An idea of grouping jobs and its applications to exact algorithms are showed in Section 5. The paper is completed with some final remarks in Section 6.

2. Problem formulation

Let us consider a set J of n preemptive, independent jobs and a set M of m parallel identical machines. All jobs are ready to be processed at time 0. Each job requires an amount of power (unknown in advance), as well as a single machine to be processed at a moment. A processing rate of a job at time t depends on the temporal power allocation. This relation is formally expressed in the following form:

$$\dot{x}_i(t) = \frac{dx_i(t)}{dt} = s_i(p_i(t)), \quad x_i(0) = 0,\ x_i(C_i) = w_i \tag{1}$$

where

- $x_i(t)$—state of job i at time t,
- $s_i(\cdot)$—increasing, continuous processing rate function of job i, $s_i(0) = 0$,
- $p_i(t)$—amount of power allotted to job i at time t,
- w_i—size of job i,
- C_i—completion time (unknown in advance) of job i.