Applied Mathematical Modelling 38 (2014) 1929–1947

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

A Monte Carlo algorithm for real time task scheduling on multi-core processors with software controlled dynamic voltage scaling

Abhishek Mishra [a,*], Anil Kumar Tripathi [b]

[a] Center of Excellence in Information & Communication Technology, Indian Institute of Technology Jodhpur, Old Residency Road, Ratanada, Jodhpur 342 011, India
[b] Department of Computer Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi 221 005, India

ARTICLE INFO

Article history:
Received 28 March 2012
Received in revised form 17 July 2013
Accepted 8 October 2013
Available online 25 October 2013

Keywords:
Dynamic voltage scaling
Energy efficient scheduling
Multi-core processor
Randomized algorithm

ABSTRACT

The task scheduling problem for multi-core processors is an important algorithm design issue. Dynamic voltage scaling (DVS) is used to reduce the energy consumption of cores. We ponder the problem of task scheduling on a multi-core processor with software controlled DVS where the objective is to reduce the energy consumption. We consider a system with a single multi-core processor with software controlled DVS having a finite set of core speeds and discuss a task scheduling problem associated with it. The problem that we address is to find a minimum energy task schedule for a given set of independent tasks that have to be completed within a given common deadline. We propose a Monte Carlo algorithm of complexity $O(t(mp+q+log(t))+p(t+q)(D^{pq}+n))$ for solving the task scheduling problem and compare it with the optimal algorithm. Here t is the number of tasks, p is the number of cores, q is the number of core speeds, m is an integer parameter that is the number of iterations we should try to get a feasible solution before declaring that no solution is possible, n is an integer parameter that is the number of iterations we should try to reduce the energy consumption when we get a feasible solution, and D is the common deadline of the tasks.

1. Introduction

In order to meet the challenges of high performance computing applications, a good solution is to use the multi-core processor architecture [1,2]. A multi-core processor is composed of two or more independent cores on a single chip. Multi-core processors are used in a variety of applications such as general purpose, embedded, network, digital signal processing (DSP), graphics etc.

Dynamic voltage scaling (DVS) is a power management technique in which varying the supply voltage can vary the speed of cores [3]. We can reduce the voltage (undervolting) to save the energy when computational requirement is low. We can also increase the voltage (overvolting) to improve the system performance when computational requirement is high.

This gives rise to the task scheduling problem [4] for multi-core processors in which the objective is to find a schedule of core speeds/voltages and also a schedule of tasks so as to minimize the energy consumption given a set of independent tasks to process and a deadline.

* Corresponding author. Tel.: +91 291 2449032.
E-mail addresses: amishra@iitj.ac.in, abhishek.rs.cse@itbhu.ac.in (A. Mishra), aktripathi.cse@itbhu.ac.in (A.K. Tripathi).

0307-904X/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2013.10.023

We consider systems with a single multi-core processor with software controlled DVS that has a finite set of core speeds. We address the problem of energy efficient task scheduling. The problem is to find a minimum energy task scheduling for a given set of independent tasks that have to be completed within a given common deadline. For solving this problem we propose a *Monte Carlo* algorithm of complexity $O(t(mp+q+log(t))+p(t+q)(D^{pq}+n))$ and compare it with the *optimal (OPT)* algorithm. Here t is the number of tasks, p is the number of cores, q is the number of core speeds, m is an integer parameter that is the number of iterations we should try to get a feasible solution before declaring that no solution is possible, n is an integer parameter that is the number of iterations we should try to reduce the energy consumption when we get a feasible solution, and D is the common deadline of the tasks.

The rest of the paper is organized as follows. Section 2 presents the literature overview. In Section 3 we consider systems with a single multi-core processor. In Section 4 we give some motivating examples. In Section 5 we address the *Energy Efficient Task Scheduling Problem (EETSP)*. In Section 6 we propose a Monte Carlo algorithm (the ENERGY-EFFICIENT-TASK-SCHEDULING-ALGORITHM - *EETSA*) for solving the EETSP problem. In Section 7 we compare the EETSA algorithm with the OPT (optimal) algorithm for dual-core processors. In Section 8 we compare the EETSA algorithm with the OPT algorithm for quad-core processors. Finally we conclude in Section 9.

2. Literature overview

There are a number of energy efficient task scheduling algorithms proposed in the literature for a wide range of system models. There are some examples of uniprocessor energy efficient algorithms [5–11]. There are some examples of multiprocessor energy efficient scheduling algorithms [12–18,4]. Some algorithms assume a continuously varying processor speed [6,4,17]. While some others assume a discretely available processor speeds [8,6,7,10,19].

Ishihara and Yasuura [8] consider the problem of voltage scheduling on a uniprocessor with DVS having a small number of discretely variable voltages. In their result they show that the voltage-scheduling problem of minimizing the energy consumption by a processor for processing a given computational load with a given deadline can be solved optimally in polynomial time with at most two voltages.

Yao et al. [10] solve the problem of minimum energy scheduling of independent jobs with arrival times, deadlines, and a given amount of computation on a uniprocessor with variable speeds. They assume that the power function is a convex function of the processor speed. They consider the case of discretely available processor speeds. They give an $O(nlog^2(n))$ time optimal offline algorithm for the problem where n is the number of jobs. Their result is extended by Irani et al. [19] to include the case in which a processor can go into a sleep state. In the sleep state, the processor speed and its power consumption is 0, but a constant amount of energy is required to bring back the processor into a non-sleep mode. They propose a 3-approximation algorithm for the offline version of the problem.

Chen et al. [7] extend the problem of Yao et al. [10] to include the case of jobs with precedence constraints. They consider the case of weakly dynamic voltage scheduling in which speed change is not allowed in the middle of processing a job. They prove the problem to be NP-Complete and also give fully polynomial-time approximation schemes for some special cases of the problem.

Yang et al. [4] consider the problem of energy efficient scheduling for a chip-multiprocessor with DVS that can use continuously varying processor speeds with no upper bound. The power function is assumed to be cubic in processor speed. They consider the problem of voltage scheduling for a set of independent tasks that share a common deadline. They give a 2.371-approximation algorithm for the problem.

Zhang et al. [17] consider the problem of energy efficient scheduling of real time dependent tasks on a given number of variable voltage processors. They give a two-phase framework to solve the problem. The first phase is for task assignment and ordering. The second phase is the voltage selection (VS) phase. They formulate the VS problem as an integer-programming (IP) problem. They prove that the IP problem for VS can be solved in polynomial time for the case of processors with continuous voltages.

3. Systems with a single multi-core processor

We are considering systems with a single multi-core processor with software controlled DVS having discretely available core speeds. Some examples of software controlled DVS are *Enhanced Intel SpeedStep Technology* [20], and *AMD PowerNow! technology* [21]. For example, *Enhanced Intel SpeedStep Technology* [20] for the Intel Pentium M processor supports processor speeds of 600 MHz to 1.6 GHz with a step of 200 MHz. For having low software overhead in switching the voltages, we assume that the DVS software is a periodic process that wakes up periodically to check if there is a need to change the voltage, and otherwise it sleeps.

We assume the power consumption function of the core to be cubic in the core speed [4]:

$$P(s)=\alpha s^3, \tag{1}$$

where α is a constant and s is speed of the core.