Applied Mathematical Modelling 39 (2015) 1422–1449

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Online dispatching and parallel processing algorithms for saving money in systems with heterogeneous, single-buffered, speed-scalable processors

Rashid Khogali, Olivia Das, Kaamran Raahemifar *

Department of Electrical & Computer Engg., Ryerson University, Toronto, Ontario, Canada

ARTICLE INFO

Article history:
Received 4 April 2013
Received in revised form 21 April 2014
Accepted 1 September 2014
Available online 16 September 2014

Keywords:
Parallel computing
Dynamic speed scaling
Multi-processor scheduling
Heterogeneous processors
Optimization
Mobile computing

ABSTRACT

We propose three online dispatching and parallel processing algorithms to optimally assign arriving heterogeneous tasks to heterogeneous speed-scalable processors under the single-buffered computing architecture. The goal of our algorithms is to minimize the total financial cost (in dollars) of response time and energy consumption (TCRTEC) of tasks using dynamic speed-scaling, where each processor's speed can change dynamically within hardware and software processing constraints. Processors are heterogeneous in that they may differ in their hardware specifications with respect to maximum processing rate and power function parameters. Tasks are heterogeneous in terms of computation volume, memory and minimum processing requirements. We consider that the unit price of response time for each task is heterogeneous since the user may want to pay higher/lower unit prices for certain tasks. We assume that the unit price of energy for all tasks may be determined by the actual price of energy in a given geographical region and time of day. We model the overhead loading time incurred when a task is loaded by a given processor prior to its execution and assume it to be heterogeneous as well. We synthesize the *Single Buffer Decision and Parallel Processing* (SBDPP) algorithm and its two other versions. The first two versions allow the user to specify the unit price of energy and response time for executing each arriving task. The algorithm's second version extends the functionality of the first by allowing the user or the Operating Software (OS) of the computing device to further modify a task's unit price of time or energy in order to achieve a linearly controlled operation point that lies somewhere in the *economy-performance* mode continuum of the task's execution. The algorithm's third version operates exclusively on the latter. We briefly extend the SBDPP algorithm and its versions to consider migration, where an unfinished task is paused and resumed on another processor. The SBDPP algorithm is qualitatively compared against its two other versions. The SBDPP dispatcher is analytically shown to outperform the well-known Round Robin dispatcher in terms of the TCRTEC performance metric. Furthermore, numerical simulations suggest that the SBDPP dispatcher outperforms the Round Robin dispatcher with average cost savings exceeding 300% in both stochastic and deterministic traffic conditions even when processors are mildly heterogeneous (processor hardware parameters conservatively chosen not to differ from

Abbreviations: AMD, advanced micro devices (Technology Company); CMOS, complementary metal oxide semiconductor; FPDPA, Fixed Power Decision and Processing Algorithm; HMO, Horizontal Migratory Operation; IBM, international business machines corporation; OS, Operating Software; SBADPA, Single Buffer Assisted Decision and Processing Algorithm; SBDPP, single buffer decision and parallel processing algorithm; TCRTEC, total cost of response time and energy consumption; TCRTEC/N, average cost of response time and energy consumption for executing N tasks; TCSTEC/N, average cost of system time and energy consumption for executing N tasks.

* Corresponding author at: Electrical & Computer Engg. Dept., Ryerson University, 350 Victoria St., Toronto, Ontario M5B 2K3, Canada. Tel.: +1 (416) 979 5000x6097; fax: +1 (416) 979 5280.

E-mail address: kraahemi@ee.ryerson.ca (K. Raahemifar).

http://dx.doi.org/10.1016/j.apm.2014.09.004
0307-904X/© 2014 Elsevier Inc. All rights reserved.

mean by more than 10%). Simulations also suggest that even when processors are mildly heterogeneous, the SBDPP dispatcher robustly tolerates a higher arrival rate that is over five times that of the Round Robin dispatcher (without rejecting any task in the long run). We deduce a relationship among the number of processors, response time of tasks and the ideal deterministic arrival rate of tasks.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

1.1. Motivation

Battery energy in mobile computing devices is indeed a valuable and scarce resource. Desirable user experience, measured by sufficiently fast execution of tasks is equally important. The energy of mobile computing devices can be extended by higher capacity batteries or through remote execution [1]. On the go, it can also be extended by portable energy restoration devices such as solar panel chargers produced and sold by XTG Technology [2]. An online article [3] suggests that in 2009, Nokia worked on a technology to recharge their cellular phone battery by extracting energy emitted from ambient radio waves. In that same year, another online article reports that Samsung worked to develop a prototype of a solar powered cellular phone [4]. It is evident that energy in mobile computing devices is of great concern and companies that design or manufacture mobile computing devices invest in battery or energy technology to remain competitive.

Computing devices can use variable speed processors to regulate the energy consumption and completion time of executing tasks. AMD's *PowerNow* processor [5], Intel's *SpeedStep* processor [6] and IBM's *Power7* processor [7] are examples of modern processors capable of operating at variable speeds. The ability of a processor to operate at a variable speed is known as (dynamic) speed scaling. Speed scaling has been used as a strategy to reduce energy consumption [8–12]. It has been used to manage a processor's temperature and energy consumption [13] as well as to mitigate processor heat failure [14]. Some speed scaling algorithms factor both time and energy consumption of tasks [15,16].

Nowadays, many mobile computing devices such as cellular phones, Tablets, IPads and portable consoles (e.g. the PSP Vita [17]) utilize multiple processors/cores. This paper synthesizes the following three novel multi-processing speed scaling algorithms:

- Single Buffer Decision and Parallel Processing algorithm (SBDPP).
- Single Buffer Assisted Decision and Processing Algorithm (SBADPA).
- Fixed Power Decision and Processing Algorithm (FPDPA).

SBDPP, SBADPA and FPDPA run in real time to optimally allocate incoming heterogeneous tasks to be run on multiple, heterogeneous, single-buffered processors. The goal is to minimize the financial cost of both response time and energy consumption of tasks using intelligent dispatchers and dynamic speed scaling. The first two algorithms explicitly determine the optimum processing rate of a task as a function of the following: the processor's power function parameters; a task's user (or OS)-specified unit cost of energy; and a task's user-specified unit cost of response time.[1] The third algorithm determines the optimum processing rate as a function of the processor's power function parameters as well as a user-defined parameter that linearly controls the operation of a given processor in the *economy-performance* continuum.

A task's processing speed is positively related to its energy consumption, and inversely related to its execution time. Therefore, minimizing energy consumption (extreme economy mode) and minimizing execution time (extreme performance mode) of a task are orthogonal tradeoffs of each other. This trade-off leads to a variety of speed scaling problems, each with different constraints, objectives and assumptions. An example is the problem of *maximum lateness minimization* under an energy budget [10]. Another problem is minimizing the average response time of a collection of dynamically released tasks subject to a constraint on energy [18].

1.2. Previous work

A seminal paper proposed by Yao et al. [12] formally introduced a single possessor speed scaling problem to minimize energy consumption. Assuming the processor's power consumption ($P(s)$) is a convex function of processor speed (s), where $P(s) = s^{\alpha}$ for $\alpha > 1$, the authors considered scheduling a sequence of tasks on a single variable speed processor. Each task has a required deadline, release time and processing volume (analogous to the number of CPU cycles required to execute a task). They allow pre-emption, where a task is allowed to resume on the same processor after being interrupted. Yao et al. proposed an optimal offline algorithm (YDS for Yao, Demers and Shenker) to solve the task scheduling problem in polynomial

[1] The second algorithm (SBADPA) can also determine the optimum processing rate as a function of parameters that are identical to that of the third algorithm (FPDPA).

Download English Version:

https://daneshyari.com/en/article/1703388

Download Persian Version:

https://daneshyari.com/article/1703388

Daneshyari.com