Performance Evaluation 73 (2014) 3–17

Contents lists available at ScienceDirect

Performance Evaluation

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

Decoupled speed scaling: Analysis and evaluation

Maryam Elahi *, Carey Williamson, Philipp Woelfel

Department of Computer Science, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4

ARTICLE INFO

Article history:
Available online 17 May 2013

Keywords:
Speed scaling systems
Fairness
Efficiency
Optimality

ABSTRACT

In this paper, we introduce the notion of decoupled speed scaling, wherein the speed scaling function is completely decoupled from the scheduling policy used in a simple single-server computer system. As an initial result, we first demonstrate that the Fair Sojourn Protocol (FSP) scheduling policy does not work properly with coupled (native) speed scaling, but that it can and does work well with decoupled speed scaling. We then compare the performance of PS, SRPT, and FSP scheduling policies under decoupled speed scaling, and demonstrate significant advantages for FSP. Our simulation results suggest that it might be possible to simultaneously achieve fairness, robustness, and near optimality with decoupled speed scaling.

1. Introduction

Many modern processors support dynamic speed scaling, which allows the CPU speed (e.g., clock frequency, voltage) to be adjusted dynamically via software control in the operating system [1]. The primary motivation for this feature is energy efficiency in the presence of highly-varying workloads. That is, the processor can be run at or near full speed to complete work quickly when demand is high, but can be reduced to a low-power or quiescent mode when demand is low or absent.

The dynamic adjustment of the processor speed is often referred to as "CPU speed scaling" in the literature, though the technically precise term is Dynamic Voltage and Frequency Scaling (DVFS). Modern processors typically support in the order of a dozen discrete operating states, rather than just two (e.g., On/Off in gated speed scaling) or three (e.g., On/Sleep/Off for a typical laptop).

The performance implications of speed scaling designs are interesting, even in the single processor case. An early paper on this topic was published by Weiser, Welch, Demers and Shenker in 1994 [2], who used simulation to explore the tradeoffs between response time and CPU energy consumption for empirical Unix workloads. This work pre-dated the widely-cited Yao, Demers, and Shenker (YDS) paper [3] on speed scaling from 1995, which provided the first formal treatment of the speed scaling problem, including several heuristic algorithms, as well as proofs that they were within a constant factor of the optimal. The latter paper triggered substantial follow-up work by Albers [4,5], Bansal [6–8], and others on improved algorithms, tighter bounds, and alternative metrics for evaluating speed scaling designs.

In speed scaling systems, the traditional performance metrics of throughput and response time become secondary to energy consumption, or a weighted combination of energy consumption and response time. The typical formulation[1] of the problem involves optimizing the total cost z, where:

$$z = E[T] + E[\varepsilon]/\beta. \tag{1}$$

In this expression, T represents response time, ε reflects energy cost, and β is a relative weighting factor.

* Corresponding author. Tel.: +1 4039662942; fax: +1 4032844707.
E-mail addresses: bmelahi@ucalgary.ca (M. Elahi), carey@ucalgary.ca (C. Williamson), woelfel@ucalgary.ca (P. Woelfel).

[1] An alternative cost function that is starting to receive more attention lately is the *energy-delay product* [9]. Because energy consumption is invariant under our definition of decoupled speed scaling, our results are applicable to either formulation of the weighted cost model.

0166-5316/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.peva.2013.04.002

Andrew, Lin and Wierman [10] presented an intriguing paper on this topic in ACM SIGMETRICS 2010. The authors demonstrate that there are inherent tradeoffs between optimality, fairness, and robustness in speed scaling systems. In particular, they prove that it is possible to provide *any two* of these properties simultaneously, but not all three. For example, Shortest Remaining Processing Time (SRPT) scheduling with dynamic speed scaling can provide optimal total cost, and is robust to uncertainties in the workload estimation, but it is unfair to large jobs. Conversely, Processor Sharing (PS) with dynamic speed scaling is fair to all jobs, and robust, but its mean response time and energy consumption can be much worse than SRPT.

In our paper, we investigate the tradeoffs inherent in speed scaling systems. In particular, we propose *decoupled speed scaling*, wherein the speed of the system is determined by an external speed scaling function that is agnostic about the current state of the system under a particular scheduling policy, and only depends on the incoming jobs and their sizes. This approach differs from the well-studied job-count-based speed scaling (coupled speed scaling), where the speed is determined dynamically based on the current state of the system, namely the number of jobs remaining in the system.

In particular, our speed scaling function is determined by a virtualized execution of a reference scheduling policy, such as PS, in the background. While this "virtual PS" idea has been used previously in the design of scheduling policies such as Weighted Fair Queueing (WFQ), Generalized Processor Sharing (GPS), and FSP, to the best of our knowledge we are the first to apply this principle to the speed scaling function itself. Furthermore, our approach is not restricted to using PS as the reference scheduler.

This idea is conceptually elegant, and simplifies the analysis of speed scaling systems. In particular, it facilitates the analysis of additional scheduling policies, including the Fair Sojourn Protocol (FSP) [11], and relaxes some of the restrictive assumptions in [10] regarding the structure of speed scaling systems (i.e., "natural" speed scaling).

The decoupled speed scaling model enables us to fix the speeds and then compare the behaviors (e.g., response time, fairness) of scheduling policies on a level playing field. The resulting system has a time-varying capacity, similar to stochastic capacity systems, which are well-studied in the literature (e.g., [12]). However, there are fundamental differences as well. In decoupled speed scaling, the capacity of the system varies deterministically, rather than stochastically. Furthermore, for a given speed scaling function, the CPU speed is identical at any point in time for any scheduling policy that governs the system. Thus the energy component of the cost function becomes equal for the policies under comparison, and the sole remaining focus is on the response time.

Our paper makes several key contributions. First, we define and propose the decoupled speed scaling model as a new approach to speed scaling system design. Second, we show that all policies are efficient in coupled speed scaling systems based on job count. Third, we show that the FSP scheduling policy is ill-suited for job-count-based coupled speed scaling, but that it can and does work well with decoupled speed scaling. Finally, using simulation, we compare the performance of PS, SRPT, and FSP scheduling policies under coupled and decoupled speed scaling. Our simulation results demonstrate pronounced advantages for FSP, which lead us to speculate that it might be possible to attain fairness, robustness, and near optimality with decoupled speed scaling. Note that the latter result does not directly contradict the results in [10], since the underlying assumptions about speed scaling differ.

The remainder of the paper is organized as follows. Section 2 presents a brief description of prior related work. Section 3 provides a simple pedagogical example to help motivate our work. Section 4 presents our formal system model, while Sections 5 and 6 present our theoretical results for coupled and decoupled speed scaling systems, respectively. Section 7 presents simulation results to validate the models. Finally, Section 8 concludes the paper.

2. Background and related work

Our paper builds upon substantial prior work in scheduling and speed scaling system design. In this section, we provide a concise summary of prior related work, particularly that most relevant to our current paper.

2.1. Scheduling policies

Processor Sharing (PS) is a well-known scheduling policy that epitomizes fairness [13,14]. PS shares a single processor simultaneously among n active jobs in the system by devoting a service rate of $1/n$ to each job. As jobs arrive and depart, PS dynamically adjusts the service rate provided to each job, while the aggregate rate is always fixed at unity.

Shortest Remaining Processing Time (SRPT) is a preemptive scheduling policy that optimizes mean response time. Using advance knowledge of job size information, the SRPT policy always selects for service the job that has the least remaining service time. With this approach, the mean waiting time and the mean response time are minimized [15–18]. SRPT scheduling has generated significant research interest, particularly in the context of request scheduling in Web servers [19–23]. Under certain job size distributions, SRPT even has a counter-intuitive "all can win" property, where all jobs prefer SRPT to PS [19]. However, this policy is sometimes unfair [24]. Furthermore, the performance of SRPT can deteriorate if it does not have accurate job size information [25].

Several other scheduling policies attempt to exploit the response time advantage of size-based scheduling, while also considering fairness and practical issues. Examples include Foreground–Background (FB) [26], Least Attained Service (LAS) [27], Resource Allocation Queueing Fairness Measure (RAQFM) [28], and the Fair Sojourn Protocol (FSP) [11]. FB and LAS approximate the effectiveness of SRPT, without the need to know job sizes in advance. RAQFM balances fairness based