Optimal Speed Profile of a DVFS Processor under Soft Deadlines

Jonatha Anselmi[1], Bruno Gaujal[1], Louis-Sébastien Rebuffi[1]

jonatha.anselmi.inria.fr, bruno.gaujal@inria.fr,louis-sebastien.rebuffi@univ-grenoble-alpes.fr

[1] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000 Grenoble, France.

1. INTRODUCTION

Minimizing the energy consumption of embedded systems with real-time execution constraints is becoming more and more important. More functionalities and better performance/cost tradeoffs are expected from such systems because of the increased use of real-time applications and the fact that batteries are becoming standard power supplies. Dynamically changing the speed of the processor is a common and efficient way to reduce energy consumption and remarkable gains can be obtained when considering cache-intensive and/or CPU-bound applications as the CPU energy consumption may dominate the overall energy consumption. In fact, this is the reason why modern processors are equipped with Dynamic Voltage and Frequency Scaling (DVFS) technology [7]. In the deterministic case where job sizes and arrival times are known, a vast literature addressed the problem of designing both off-line and on-line algorithms to compute speed profiles that minimize the energy consumption subject to hard real-time constraints (deadlines) on job execution times; e.g., [5]. In a *stochastic* environment where only statistical information is available about job sizes and arrival times, it turns out that combining hard deadlines and energy minimization via DVFS-based techniques is much more difficult. In fact, forcing hard deadlines requires to be very conservative, i.e., to consider the worst cases. Matter of fact, existing approaches work within a finite number of jobs [6, 3].

The approach followed in this paper circumvents the difficulties described above by replacing the hard real-time constraints, i.e., jobs have hard deadlines that must be satisfied, by *soft* real-time constraints, i.e., jobs may miss their deadlines, at some cost. While the hard deadline of a job must be known at the job arrival, soft deadlines allow for a different information structure: here, only the deadline distribution is known at the job arrival. In this paper, we further assume that jobs missing their deadlines become obsolete and are dropped. Obsolescence is often found in real-time systems where the information carried by jobs may not be valid any longer after their deadline as it will be replaced by fresher input coming from other jobs. Therefore, obsolete jobs become useless and can get discarded from the queue. Dropping obsolete jobs can also model impatient customers: customers wait for service for some time (deadline) and quit (are dropped) if not served before that time. We formulate this problem as an MDP in continuous time where the state is the number of jobs in the system and the action is the processor speed. Our main result, Theorem 1, shows the existence of an optimal speed profile that is increasing in the number of jobs in the system and upper bounded by some constant. Surprisingly, our bound does not depend on the deadlines and arrival rates. In addition, it yields a simple approximation for the optimal policy and several numerical tests show that such approximation is accurate in heavy-traffic conditions.

2. METHODOLOGY AND MAIN RESULT

The system described here is a model for the dynamics of a real-time device composed of a single processor where incoming jobs need to be executed under a constraint on the amount of time that they spend in the system.

Processor. This is a DVFS processor whose speed can continuously vary in the interval $[0, S_{\max}]$. We consider that speed changes are immediate and induce no energy cost. When the processor works at speed s, it processes s units of work per second while its power dissipation is $w(s)$ watts. We require that $w(s)$ is continuous, increasing and strictly convex in the speed s.

Jobs. They form a stochastic point process, with Poisson arrivals with rate λ, i.i.d. deadlines exponentially distributed with rate δ and i.i.d. sizes exponentially distributed with rate μ. Without loss of generality, we assume that $\mu = 1$.

Dynamics. At any point in time t, the processor chooses its speed $s(t)$ and executes one of the jobs in its backlog queue. Now, three events can happen in continuous time: i) new job may join the queue, ii) the active job is completed before its deadline and leaves the system, iii) one job (active or inactive) reaches its deadline and is removed from the queue paying an immediate cost equal to C.

Cost Function. If we denote by M_T the number of missed deadlines in the time interval $[0, T]$, the objective of this paper is to study the speed profile $s(t)$ of the processor that minimizes the long-run average cost given by the missed deadlines plus power consumption, say $\mathcal{J}$. Specifically:

$$\mathcal{J} := \limsup_{T\to\infty} \frac{1}{T}\left(CM_T + \int_0^T w(s(t))\,\mathrm{d}t\right).$$

2.1 Markov Decision Process

We formulate the problem of interest as an MDP. The state space is $\mathbb{N}$ and a state represents the number of jobs in the system. The action space is $[0, S_{\max}]$, i.e., the set of available speeds for the DVFS processor. Let $\sigma = (\sigma_i)_{i\in\mathbb{N}}$ denote a stationary and deterministic speed policy adopted by the processor, i.e., $\sigma_i \in [0, S_{\max}]$ is the speed used in state i. It is well known that focusing on stationary and deterministic policies can be done with no loss of optimality in our case [4, Theorem 5.9]. Under the policy $\sigma = (\sigma_i)_{i\in\mathbb{N}}$, for $i, j \in \mathbb{N}$, the transition rates are given in Figure 1:

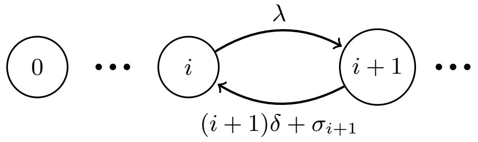

Figure 1: Markov chain diagram under policy σ.

By the ergodicity of the Markov chain X^σ under all policies σ, the long-run cost $\mathcal{J}$ is equal to the long-run expected cost and it is independent of the initial state. Letting $\mathbb{E}^\sigma$ denote the expectation given a speed policy σ, we have

$$\mathcal{J} = J(\sigma) := \limsup_{T\to\infty} \frac{1}{T} \int_0^T \mathbb{E}^\sigma c(X^\sigma(t), \sigma)\, \mathrm{d}t.$$

In this equation, the immediate cost function $c(\cdot,\cdot)$ is the expected cost incurred by the system at time t. It only depends on the current state and the current speed. Conditional on the state $(X^\sigma(t) = i)$, the obsolescence rate is $i\delta$. Thus, the expected cost is: $c(i, \sigma) := Ci\delta + w(\sigma_i)$. Stationary policies that minimize $J(\sigma)$ are optimal speed policies for the model. In particular, they are also optimal over all policies (history dependent and randomized) [4, Theorem 5.9] . Also, our MDP satisfies all the conditions given in [4, Theorem 5.9] to assert the existence of an optimal stationary deterministic policy σ^* and an optimality equation of the form

$$J^* = J(\sigma^*) = \min_{s\in[0,S_{\max}]} c(i, s) + \sum_j h^*(j) q_{i,j}(s), \; \forall i \in \mathbb{N} \quad (1)$$

where h^* is a real function defined on $\mathbb{N}$, usually referred to as *bias* of the optimal policy.

2.2 Main result

The goal of this paper is to investigate structural properties on σ^* and J^*. First, let us define B as

$$B := \operatorname*{arg\,min}_{s\in\mathbb{R}^+\cup\{+\infty\}} (w(s) + C(\lambda - \mu s)). \quad (2)$$

Then, our main result is the following.

THEOREM 1. *There exists a deterministic optimal policy $\sigma^* = (\sigma_i^*)_{i\in\mathbb{N}}$ that is increasing in i and upper bounded by B.*

The optimal speed policy of the processor is always bounded by a finite constant, namely $\min(B, S_{\max})$. We remark that B is *independent of the arrival rate, and the deadline distribution.* This is both surprising and helpful in practice. Indeed, if B is finite, one can set *a priori* the maximal speed of the processor to $S_{\max} := B$. This guarantees that in most cases, no cost reduction would be possible by using a more powerful processor. If the parameters and the power dissipation w were related to units of work instead of units of time, B would also be independent of μ and therefore of all parameters.

Underlying the proof of our main result, there are some technical challenges that we now discuss. The proposed MDP satisfies the regularity assumptions (stability, unichain) needed to establish an optimality equation as described in [4]. However, this is not enough to show structural properties of the optimal policy. In fact, the classical approach to do this is to uniformize the MDP and to investigate the properties of the corresponding discrete time value iteration operator. Unfortunately, this is not possible in our case because the transition rates are unbounded. To uniformize the MDP, a typical approach consists of truncating the state space. However, a naive truncation will not help here because the truncation barrier has a strong impact on the structure of the optimal policy in the sense that it would not preserve any monotonicity property that it may have without truncation. Instead, we use the technique proposed by Blok and Spieksma in [2], which smoothly scales down the upward rates of the truncated system as a function of the size of its state space. This technique has been already used, e.g. [1], though on discounted costs. Here, we use the same truncation technique but we apply it to the average cost. In our specific case, the convergence to the infinite system will be guaranteed by the monotone convergence theorem.

3. REFERENCES

[1] S. Bhulai, A. C. Brooms, and F. M. Spieksma. On structural properties of the value function for an unbounded jump markov process with an application to a processor sharing retrial queue. *Queueing Systems*, 76(4):425–446, 2014.

[2] H. Blok and F. M. Spieksma. Countable state Markov decision processes with unbounded jump rates and discounted cost: optimality equation and approximations. *Advances in Applied Probability*, 47(4):1088 – 1107, 2015.

[3] B. Gaujal, A. Girault, and S. Plassart. Dynamic Speed Scaling Minimizing Expected Energy Consumption for Real-Time Tasks. *Journal of Scheduling*, pages 1–25, July 2020.

[4] X. Guo and O. Hernandez-Lerma. *Continuous-time Markov decision processes. Theory and applications*, volume 62. 01 2009.

[5] M. Li, F. F. Yao, and H. Yuan. An $O(n^2)$ algorithm for computing optimal continuous voltage schedules. In *TAMC'17*, volume 10185 of *LNCS*, pages 389–400, Bern, Switzerland, Apr. 2017.

[6] J. R. Lorch and A. J. Smith. Improving dynamic voltage scaling algorithms with PACE. In *ACM SIGMETRICS 2001 Conference*, pages 50–61, 2001.

[7] D. C. Snowdon, S. Ruocco, and G. Heiser. Power management and dynamic voltage scaling: Myths and facts. In *Proc. of the 2005 Workshop on Power Aware Real-time Computing*, New Jersey, USA, Sept. 2005.