Operations Research Letters 44 (2016) 101–106

Contents lists available at ScienceDirect

Operations Research Letters

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

Energy–performance trade-off for processor sharing queues with setup delay

Misikir Eyob Gebrehiwot [*], Samuli Aalto, Pasi Lassila

Aalto University School of Electrical Engineering, Finland

ARTICLE INFO

Article history:
Received 26 June 2015
Received in revised form
4 December 2015
Accepted 4 December 2015
Available online 11 December 2015

Keywords:
$M^X/G/1$-PS
Setup delay
Energy–performance trade-off

ABSTRACT

Despite the extensive literature on energy efficient control mechanisms for servers, only few studies address the processor sharing discipline. We study the energy–performance trade-off in an energy-aware $M^X/G/1$-PS system using two popular cost metrics. Among a family of control policies that can possibly stay idle before going to sleep to save energy, the optimal policy is found to be a simple control that either leaves the server idle, or puts it to sleep immediately whenever it becomes idle.

1. Introduction

In response to the increased desire to cut down energy costs of modern ICT systems, there is a growing focus on the study of energy efficient control mechanisms for servers [4,3,5,14]. Energy savings might be achieved by switching servers to lower energy states. However, this comes at a cost of reduced performance due to the long setup delay needed to switch a server between energy states. Thus, energy-aware control of servers involves optimization of the energy–performance trade-off. A variety of system cost metrics are utilized in the literature to quantify this trade-off. The two most popular cost metrics are the Energy–Response time Weighted Sum (ERWS) [4,14,13,6] and the Energy–Response time Product (ERP) [7,10,5].

For an energy-aware single server system with possible *busy*, *idle*, and *off* energy states, recent studies found the optimal control under the ERWS cost metric to be simple [13]. That is, when the server is not busy, it either stays idle or it is switched-off. This result was generalized in [6] by considering a system with multiple *sleep* states under both the ERWS and ERP cost metrics.

However, the service discipline considered in the study of energy-aware systems has mainly been FIFO with few notable exceptions. Gandhi et al. [5] studied the energy–performance trade-off for an M/M/1 system with a non-preemptive scheduling discipline, whereas Hyytiä et al. [8] considered an energy-aware M/D/1-PS system.

In this paper, we consider an energy-aware $M^X/G/1$-PS queue with batch arrivals that incorporates the *busy, idle, sleep and setup* energy states, and show that a simple control, which employs only one of the *sleep* or *idle* states whenever the server is not busy, still remains optimal. The optimal control happens to be the same as for the M/G/1-FIFO system studied in [13,6].

Essential in our analysis is an integral equation that characterizes the mean conditional response time of a processor sharing system. Similar integral equations have been studied for a variety of processor sharing models, see, e.g., [12,11,1,2].

The paper is organized as follows. In the following section we give a preliminary result related to our study. Section 3 describes the model, and in Section 4 we analyse the mean response time of the system. Optimization of the energy–performance trade-off is studied in Section 5, followed by a numerical illustration in Section 6. Section 7 concludes the paper.

2. Preliminaries

Consider the integral equation

$$g(t) = \bar{F}(t) + \lambda \int_0^\infty g(y)\bar{F}(t+y)dy + \lambda \int_0^t g(y)\bar{F}(t-y)\,dy, \tag{1}$$

where λ and $\bar{F}(t)$ are the job arrival rate and tail function of the service time distribution of a single server queueing system, so that $\int_0^\infty \bar{F}(t)\,dt = \mathrm{E}[S]$, with S representing a generic service time.

[*] Corresponding author.
E-mail addresses: misikir.gebrehiwot@aalto.fi (M.E. Gebrehiwot), samuli.aalto@aalto.fi (S. Aalto), pasi.lassila@aalto.fi (P. Lassila).

http://dx.doi.org/10.1016/j.orl.2015.12.004
0167-6377/© 2015 Elsevier B.V. All rights reserved.

Lemma 1. *For a stable system with $\rho = \lambda \mathrm{E}[S] < 1$ and non-negative $g(t)$, the integral equation given in (1) has a unique solution which depends only on the job arrival rate λ and the tail function $\bar{F}(t)$.*

Proof. See [1, Theorem 1] and the remark thereafter. □

Later, we will define $g(t)$ in relation to the mean conditional processing time of a job in an energy-aware $\mathrm{M}^X/\mathrm{G}/1$-PS queue. The *processing time* of a job is here defined to be the time elapsed between the beginning and end of service.

3. Model

We consider a single-server PS queue with general i.i.d. service times and batch arrivals. The arrival process is assumed to be a batch Poisson process where i.i.d. batches of random size $B_i \geq 1$ arrive with rate λ_b. Assume further that the mean $\mathrm{E}[B]$ and the second moment $\mathrm{E}[B^2]$ of the batch size distribution are finite, and denote

$$b = \frac{\mathrm{E}[B^2]}{\mathrm{E}[B]} - 1, \tag{2}$$

which refers to the mean number of other jobs in a batch with at least one (tagged) job. The load of the system, ρ, is given by

$$\rho = \lambda_b \mathrm{E}[B]\mathrm{E}[S],$$

and we assume that $\rho < 1$.

Consider a family of control policies where jobs are served in the *busy* state exhaustively and once the system becomes *idle*, a timer I is started, which is a random variable having a general distribution. We assume the timer is reset every time it is interrupted by an arriving batch of jobs or when it expires. Let I^{tot} denote the total idling time accumulated between two expirations of the timer. If the timer expires before a new batch of jobs arrives, the server will be put in a *sleep* state. Once in this state, the system waits for a turn-on threshold of k batches of jobs before starting the server again. (Note that, due to mathematical tractability, the threshold k refers to batches, instead of jobs.) There is a generally distributed random *setup* delay, D, involved before the accumulated jobs can be served in the *busy* state. Let Π denote the family of such control policies.

The four energy states have power consumption values of P_{busy}, P_{idle}, P_{sleep} and P_{setup}, respectively. We denote the mean power consumption of the system by $\mathrm{E}[P]$ and the mean response time of a job by $\mathrm{E}[T]$. To study the energy–performance trade-off, the well known Energy–Response time Weighted Sum (ERWS),

$$w_1 \mathrm{E}[T] + w_2 \mathrm{E}[P], \tag{3}$$

and Energy–Response time Product (ERP),

$$\mathrm{E}[T]\mathrm{E}[P], \tag{4}$$

cost metrics are utilized.

4. Analysis

Most known results for the energy–performance trade-off assume the FIFO service discipline. In this section, we derive the mean response time, $\mathrm{E}[T]$, for the energy-aware $\mathrm{M}^X/\mathrm{G}/1$-PS queue introduced above. Although explicit expressions will not be worked out, we derive an integral equation for the mean conditional processing time, which can still be utilized in the ERWS and ERP cost metrics to optimize the energy–performance trade-off.

The response time of a job can be decomposed into

$$T = Q + U,$$

where Q and U are the queueing and processing times of the job, respectively. The *queueing time* is defined to be the time elapsed between the arrival of the job and its beginning of service. Since the service discipline is PS, a job would, however, need to be queued only if it arrives in either the *sleep* or *setup* state.

Theorem 1. *For any control policy in Π, the mean conditional response time is given by*

$$\mathrm{E}[T|S=t] = \mathrm{E}[Q] + U(t), \tag{5}$$

where $\mathrm{E}[Q]$ is the expected queueing time, and is given by

$$\mathrm{E}[Q] = \frac{(1-\rho)\left(\frac{k(k-1)}{2\lambda_b} + k\mathrm{E}[D] + \frac{\lambda_b}{2}\mathrm{E}[D^2]\right)}{k + \lambda_b \mathrm{E}[D] + \lambda_b \mathrm{E}[I^{\mathrm{tot}}]}, \tag{6}$$

whereas $U(t)$ is the mean conditional processing time of a job of size t, i.e., $U(t) = \mathrm{E}[U|S=t]$, and its derivative $U'(t)$ is the unique solution of

$$\begin{aligned} U'(t) &= 1 + K_b \bar{F}(t) + \lambda_b \mathrm{E}[B] \int_0^\infty U'(y)\bar{F}(t+y)\,dy \\ &\quad + \lambda_b \mathrm{E}[B] \int_0^t U'(y)\bar{F}(t-y)\,dy, \end{aligned} \tag{7}$$

where $K_b = b + 2\lambda_b \mathrm{E}[B]\mathrm{E}[Q]$.

Proof. Let us take the expiration time of the idling timer I as a regeneration point. Between two regeneration points, the server traverses the *sleep* and *setup* states once and alternates between the *busy* and *idle* states multiple times.

The mean total idling time follows a geometric distribution with success probability $\mathrm{P}\{I < A\}$, where A refers to a generic interarrival time between batches of jobs, which is exponentially distributed with rate λ_b. This gives

$$\mathrm{E}[I^{\mathrm{tot}}] = \frac{\mathrm{E}[\min\{I, A\}]}{\mathrm{P}\{I < A\}},$$

see [6] for more detailed discussion.

The mean total number of batch arrivals during the *idle*, *sleep* and *setup* states in a single cycle is given by

$$k + \lambda_b \mathrm{E}[D] + \lambda_b \mathrm{E}[I^{\mathrm{tot}}].$$

Each of the arriving batches initiates its own busy period like a batch of jobs in an ordinary $\mathrm{M}^X/\mathrm{G}/1$-PS system would do. Hence, the mean number of jobs served in one regeneration cycle is given by

$$\mathrm{E}[N] = \frac{\mathrm{E}[B](k + \lambda_b \mathrm{E}[D] + \lambda_b \mathrm{E}[I^{\mathrm{tot}}])}{1-\rho}. \tag{8}$$

Queueing time. An arriving job would be in the first one of the k initiating batches with probability $\mathrm{E}[B]/\mathrm{E}[N]$ and if this happens, its expected queueing time before it gets service is $\frac{k-1}{\lambda_b} + \mathrm{E}[D]$ units of time. Similarly, a job would be in the second one of those k initiating batches with the same probability and a mean waiting time of $\frac{k-2}{\lambda_b} + \mathrm{E}[D]$ and so on. A job arrives during the *setup* state with probability $\lambda_b \mathrm{E}[B]\mathrm{E}[D]/\mathrm{E}[N]$ and its expected queueing time is $\mathrm{E}[D^2]/(2\mathrm{E}[D])$. This gives us

$$\mathrm{E}[Q] = \frac{\mathrm{E}[B]}{\mathrm{E}[N]}\left(\frac{k(k-1)}{2\lambda_b} + k\mathrm{E}[D] + \frac{\lambda_b}{2}\mathrm{E}[D^2]\right), \tag{9}$$

which is equivalent with (6).

Processing time. Here we apply an approach similar to [2]. Consider a tagged job whose size is greater than t. Due to the PS discipline, we have

$$U'(t) = 1 + \mathrm{E}[L_1(t)] + \mathrm{E}[L_2(t)], \tag{10}$$

where $L_1(t)$ denotes the number of jobs that were already in the system by the time the tagged job started service, whereas $L_2(t)$ is the number of jobs that arrive during the service of the tagged