JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **13**, Number **4**, October **2017**

doi:10.3934/jimo.2017025

pp. **1927–1943**

SINGLE SERVER RETRIAL QUEUES WITH SPEED SCALING: ANALYSIS AND PERFORMANCE EVALUATION

Tuan Phung-Duc

Division of Policy and Planning Sciences
Faculty of Engineering, Information and Systems
University of Tsukuba, Ibaraki 305-8573, Japan

Wouter Rogiest* and Sabine Wittevrongel

Department of Telecommunications and Information Processing
Ghent University, St.-Pietersnieuwstraat 41
B-9000 Gent, Belgium

Abstract. Recently, queues with speed scaling have received considerable attention due to their applicability to data centers, enabling a better balance between performance and energy consumption. This paper proposes a new model where blocked customers must leave the service area and retry after a random time, with retrial rate either varying proportionally to the number of retrying customers (linear retrial rate) or non-varying (constant retrial rate). For both, we first study a basic case and then subsequently incorporate the concepts of a setup time and a deactivation time in extended versions of the model. In all cases, we obtain a full characterization of the stationary queue length distribution. This allows us to evaluate the performance in terms of the mentioned balance between performance and energy, using an existing cost function as well as a newly proposed variant thereof. This paper presents the derivation of the stationary distribution as well as several numerical examples of the cost-based performance evaluation.

1. **Introduction.** In current large-scale data centers, thousands of parallel servers are responsible for the processing of incoming jobs. While system performance is still measured by means of a traditional measure like job latency, the overall energy consumption is a second important consideration. According to [19], data centers constitute about 40% of the global ICT electricity consumption in 2012, or approximately 107 TWh. Therefore, a modern system needs mechanisms to handle the trade-off between performance and energy consumption [3].

In response to this, speed scaling has been developed [8, 20, 21], slowing down the server speed when the number of customers is low, and speeding up, in the converse case. As argued in [20] (and later in [8]), this enables a better balance between performance and energy consumption. This is also argued in [10] in the context of data centers, and can be intuitively understood as follows. Assume that

2010 *Mathematics Subject Classification.* Primary: 68M20, 60K25; Secondary: 90B22.

Key words and phrases. Data center, energy efficiency, speed scaling, setup time, deactivation time, retrial queue.

The reviewing process of the paper was handled by Wuyi Yue and Yutaka Takahashi as Guest Editors.

* Corresponding author.

the speed of the system can be tuned by tuning the service rate ("speed scaling"). While the power consumption rises more than proportionally with the service rate (e.g., with the former approximately equal to the square of the latter [20]), this does not hold true for the mean number of customers in the system. Specifically, the latter is approximately proportional to the mean service time in case of very low traffic load (with low arrival rate). Opposed to this, in case of high traffic load, speeding up can have in inverse a much larger than proportional impact on the number of customers in the system, while the relation between service rate and power consumption remains the same. In other words, the added value per additional unit of power is higher when the traffic load is high than when the traffic load is low, creating a trade-off. In this sense, it is useful to work at lower speed when the traffic load is low, and at higher speed in the converse case.

To the best of our knowledge, the first queueing model to address (a form of) speed scaling is [4], which presents the analysis of a single server system with Poisson arrivals and a service rate that depends on the number of customers n according to a formula $\mu_n = n^c\mu_1$, where μ_1 is a model parameter describing the service rate for a customer arriving at an idle system. An important recent contribution with speed scaling is [8], which features the concept of *switching delay* discussed also below.

While [4, 8] study a classic model without retrials, in this work, we assume a retrial queue for the incoming jobs (or customers). This reflects the distributed nature of a data center, in which the workload manager maintaining the queue is separated from the actual processing units. In this respect, it is useful to mention the related work on retrial queues of [11], which presents a generic study of the broad class of retrial queues with state-dependent rates, sharing many of the assumptions of this contribution. However, [11] does not discuss speed scaling as such and does not include any of the expressions derived below. Moreover, the concepts of setup time and deactivation time are not treated in [11], whereas both play a key role in this contribution.

The concept of setup time and its counterpart of deactivation time are important and realistic model extensions since these phenomena are found in realistic data centers. Setup times were studied earlier in different contexts in e.g. [1, 5–7, 13–16]. Furthermore, the mentioned switching delay of [8] is identical to the setup time as defined in this work. A similar concept to deactivation time was studied earlier in a context without retrial queues, in [9].

Summarizing the above, we conclude that speed scaling has already been considered in settings with setup times [8], and also indirectly in settings with retrial queues [11], but never directly in the combination with both setup times and retrial queues, and never with deactivation times. This is exactly the contribution of this work. Specifically, sections 4 to 7 of this paper are devoted to retrial models with speed scaling and a setup time, where the models in sections 5 and 7 additionally include a deactivation time. Based on the formulas derived in this paper, we also evaluate the mentioned balance between performance and energy consumption, using an existing cost function as well as a newly proposed variant thereof. This paper extends an earlier version [17], where neither the concept of deactivation time nor the cost-based evaluation were considered.

This paper is organized as follows. In sections 2 and 3, a speed scaling model without setup time is considered, either with classical linear retrial rate (section 2) or with constant retrial rate (section 3). In sections 4 to 7, speed scaling models with setup time are considered, either without deactivation (linear retrial rate in

section 4, constant retrial rate in section 6) or with deactivation (linear in section 5, constant in section 7). Section 8 presents a note on practical implementation. The cost functions, combining power consumption and delay performance, are discussed in section 9, followed by several numerical examples. Conclusions are drawn in section 10.

2. Linear retrial rate model.

2.1. **Assumptions.** We consider a single server retrial queueing system where blocked customers leave the server and retry after independent and identically distributed (iid) retrial times. Retrials take place at rate $n\nu$, where n is the number of customers in orbit: A so-called linear retrial rate model. Further, as is common in retrial queue terminology, see e.g. [1, 12], during consecutive retrials, the customer is said to be in the orbit. However, different from a classical retrial queue, *speed scaling* takes place: The service rate of the server is linear to the total number of customers in the system. In particular, if there are n customers in the orbit the customer in the server (if any) is served at rate $(n+1)\mu$. Customers arrive at the system according to a Poisson process with rate λ.

2.2. **Analysis.** In order to analyze the above queueing model, let $C(t)$ and $N(t)$ denote the number of active servers and the number of customers in the orbit, respectively. It is easy to see that $\{X(t) = (C(t), N(t)); t \geq 0\}$ forms a Markov chain on the state space

$$\mathcal{S} = \{(i,n); i = 0,1, n \in \mathbb{Z}_+\},$$

where $\mathbb{Z}_+ = \{0,1,2,\dots\}$. It is also easily seen that the system is always stable due to the speed scaling. Let $\pi_{i,n} = \lim_{t\to\infty} \Pr[C(t) = i, N(t) = n]$ $((i,n) \in \mathcal{S})$ denote the joint stationary distribution of $\{X(t)\}$.

In this section, we derive a recursion for calculating the joint stationary distribution $\pi_{i,n}$ $((i,n) \in \mathcal{S})$. The balance equations read as follows:

$$(\lambda + n\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}, \qquad n \geq 0, \tag{1}$$

$$(\lambda + \mu)\pi_{1,0} = \lambda\pi_{0,0} + \nu\pi_{0,1}, \tag{2}$$

$$(\lambda + (n+1)\mu)\pi_{1,n} = \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + (n+1)\nu\pi_{0,n+1}, \qquad n \geq 1. \tag{3}$$

We define the partial generating functions $\Pi_0(z)$ and $\Pi_1(z)$ as

$$\Pi_0(z) = \sum_{n=0}^{\infty} \pi_{0,n} z^n, \qquad \Pi_1(z) = \sum_{n=0}^{\infty} \pi_{1,n} z^n. \tag{4}$$

Using these definitions, we obtain the following system of equations for $\Pi_0(z)$ and $\Pi_1(z)$:

$$\lambda\Pi_0(z) + \nu z\Pi_0'(z) = \mu z\Pi_1'(z) + \mu\Pi_1(z), \tag{5}$$

$$\lambda\Pi_1(z) + \mu z\Pi_1'(z) + \mu\Pi_1(z) = \lambda\Pi_0(z) + \lambda z\Pi_1(z) + \nu\Pi_0'(z). \tag{6}$$

Adding these two equations, we find $\nu\Pi_0'(z) = \lambda\Pi_1(z)$. Substituting $\Pi_1(z)$ into (5), we obtain

$$z\Pi_0''(z) + \frac{\lambda}{\mu}\left(\frac{\mu}{\lambda} - z\right)\Pi_0'(z) - \frac{\lambda^2}{\mu\nu}\Pi_0(z) = 0. \tag{7}$$

Substituting $z = \mu x/\lambda$ and introducing the notation $p(x) = \Pi_0(\mu x/\lambda) = \Pi_0(x/\rho)$ ($\rho = \lambda/\mu$), we obtain the following equation:

$$xp''(x) + (1-x)\,p'(x) - \frac{\lambda}{\nu}p(x) = 0.$$

This is the confluent hypergeometric differential equation whose solution is a confluent hypergeometric function, a special case of the hypergeometric function also encountered in the analysis of some retrial queue models without speed scaling, such as the one studied in [2]. The solution for this equation is given by following expression:

$$p(x) = \pi_{0,0} M(a,b,x) = \pi_{0,0} \sum_{n=0}^{\infty} \frac{a_{(n)} x^n}{b_{(n)} n!},$$

where $M(a,b,x)$ denotes the confluent hypergeometric function, with

$$a = \frac{\lambda}{\nu}, \qquad b = 1,$$

and where for a real number x, the symbol $x_{(n)}$ denotes the Pochhammer symbol, defined as follows:

$$x_{(0)} = 1, \qquad x_{(n)} = x(x+1)\cdots(x+n-1), \qquad n \geq 1.$$

We then have

$$\Pi_0(z) = p(\lambda z/\mu) = \pi_{0,0} \sum_{n=0}^{\infty} \frac{a_{(n)} (\lambda z/\mu)^n}{b_{(n)} n!} = \pi_{0,0} \sum_{n=0}^{\infty} \frac{a_{(n)} (\lambda z/\mu)^n}{n!^2},$$

where we used $b_{(n)} = n!$ in the last equality. Thus, we get

$$\pi_{0,n} = \pi_{0,0} \frac{a_{(n)} \rho^n}{b_{(n)} n!} = \pi_{0,0} \frac{a_{(n)}}{n!^2} \left(\frac{\lambda}{\mu}\right)^n.$$

Furthermore, we have

$$\Pi_1(z) = \frac{\nu}{\lambda}\Pi_0'(z) = \pi_{0,0} \frac{\lambda}{\mu} M(a+1, b+1, \lambda z/\mu),$$

where we have used

$$M'(a,b,x) = \frac{a}{b} M(a+1, b+1, x).$$

Formally, the unknown number $\pi_{0,0}$ is determined using the normalization condition:

$$\Pi_0(1) + \Pi_1(1) = 1,$$

which yields

$$\pi_{0,0} = \frac{1}{M(a,b,\lambda/\mu) + \frac{\lambda}{\mu} M(a+1, b+1, \lambda/\mu)}.$$

Although this is an explicit expression for $\pi_{0,0}$, it still contains the confluent hypergeometric function, and thus, indirectly, infinite sums. This however poses no problem for the numerical calculation of $\pi_{0,0}$, since most scientific software packages are able to handle confluent hypergeometric functions directly.

Remark 1. It should be noted that in the analysis of an M/M/1 retrial queue with a linear retrial rate yet without speed scaling, one does not encounter a hypergeometric function, but merely a first-order differential equation. The reason for the hypergeometric function here is the linear increase of the service rate with the number of customers in the system. In this regard, speed scaling makes the analysis more complex.

3. Constant retrial rate model.

3.1. **Assumptions.** We consider a single server retrial queueing system where blocked customers leave the server and retry at a later time. As in the previous section, the retrial times are iid random variables. However, different from the previous section, the retrial rate is independent of the number of customers in the orbit. In particular, retrials occur at a constant retrial rate ν as soon as the orbit is non-empty. Again, *speed scaling* takes place: The service rate of the server is proportional to the total number of customers in the system. Just like in the linear retrial rate case studied in the previous section, if there are n customers in the orbit the customer in the server (if any) is served at rate $(n+1)\mu$. Customers arrive at the system according to a Poisson process with rate λ.

3.2. **Analysis.** As before, let $C(t)$ and $N(t)$ denote the number of active servers and the number of customers in the orbit, respectively. It is easy to see that $\{X(t) = (C(t), N(t)); t \geq 0\}$ forms a Markov chain on the state space

$$\mathcal{S} = \{(i,n); i = 0,1, n \in \mathbb{Z}_+\}.$$

Again, the system is stable due to the speed scaling. Furthermore, the joint stationary distribution of $\{X(t)\}$ is denoted by $\pi_{i,n} = \lim_{t\to\infty} \Pr[C(t) = i, N(t) = n]$, $((i,n) \in \mathcal{S})$.

In this section, we derive a recursive scheme for calculating the joint stationary distribution $\pi_{i,n}$ $((i,n) \in \mathcal{S})$. The balance equations now read as follows:

$$\lambda\pi_{0,0} = \mu\pi_{1,0}, \tag{8}$$

$$(\lambda+\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}, \qquad n \geq 1, \tag{9}$$

$$(\lambda+\mu)\pi_{1,0} = \lambda\pi_{0,0} + \nu\pi_{0,1}, \tag{10}$$

$$(\lambda+(n+1)\mu)\pi_{1,n} = \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + \nu\pi_{0,n+1}, \qquad n \geq 1. \tag{11}$$

Again introducing the partial generating functions $\Pi_0(z)$ and $\Pi_1(z)$, we obtain the following system of equations:

$$\lambda\Pi_0(z) + \nu(\Pi_0(z) - \pi_{0,0}) = \mu z\Pi_1'(z) + \mu\Pi_1(z), \tag{12}$$

$$\lambda\Pi_1(z) + \mu z\Pi_1'(z) + \mu\Pi_1(z) = \lambda\Pi_0(z) + \lambda z\Pi_1(z) + \frac{\nu}{z}(\Pi_0(z) - \pi_{0,0}). \tag{13}$$

Adding these two equations, we get

$$\lambda\Pi_1(z) = \frac{\nu(\Pi_0(z) - \pi_{0,0})}{z}$$

or

$$z\Pi_1(z) = \frac{\nu(\Pi_0(z) - \pi_{0,0})}{\lambda}.$$

Taking the first derivative of the latter equation with respect to z and substituting the result in the right-hand side of (12), we find

$$\lambda\Pi_0(z) + \nu(\Pi_0(z) - \pi_{0,0}) = \frac{\mu\nu}{\lambda}\Pi_0'(z)$$

or

$$\Pi_0'(z) = \frac{\lambda(\lambda+\nu)}{\mu\nu}\Pi_0(z) - \frac{\lambda}{\mu}\pi_{0,0}. \tag{14}$$

Solving this differential equation we obtain

$$\Pi_0(z) = \pi_{0,0}\left[\frac{\lambda}{\lambda+\nu}\exp(\gamma z) + \frac{\nu}{\lambda+\nu}\right],$$

where we introduced the notation

$$\gamma = \frac{\lambda(\lambda+\nu)}{\mu\nu}.$$

We also find that

$$\Pi_1(z) = \frac{\nu}{\lambda+\nu}\frac{\exp(\gamma z)-1}{z}\pi_{0,0}.$$

From the normalization condition,

$$\Pi_0(1) + \Pi_1(1) = 1,$$

we find that $\pi_{0,0} = \exp(-\gamma)$.

Remark 2. In the analysis of the simpler M/M/1 retrial queue with constant retrial rate yet without speed scaling, no differential equation is involved. The differential equation (14) in this analysis is brought about by the speed scaling.

4. **Linear retrial rate model with setup time.** In this section, we consider an extension of the model studied in section 2, introducing the concept of a setup time. As is the case in many realistic systems, upon turning idle (i.e., empty server and empty orbit), the system may go into sleep mode (or hybernation mode) to save energy, returning to active mode when triggered by the arrival of a new customer. Moving from idle to active mode may happen instantaneously (as in the models of sections 2 and 3) or the system may be in setup mode during a time interval called the setup time. In this section and the following, we assume iid setup times with exponential distribution with parameter α. Further, we assume that a new customer arriving at an idle system immediately goes to the server without joining the orbit and triggers the setup of the server. Arriving customers who find the server occupied (either setting up or actually serving) join the orbit and repeat their attempt after some random time.

Let $C(t)$ denote the state of the server and $N(t)$ denote the number of customers in the orbit at time t. There are 3 possible server states:

$$C(t) = \begin{cases} 0, & \text{the server is idle,} \\ 1, & \text{the server is busy,} \\ 2, & \text{the server is in setup mode.} \end{cases}$$

Here, $\{X(t) = (C(t), N(t)); t \geq 0\}$ forms a Markov chain on the state space

$$\mathcal{S} = \{(i,n); i \in \{0,1,2\}, n \in \mathbb{Z}_+\}.$$

Figure 1 presents the transitions among the states. Note that $(0,0)$ is the state corresponding to a system in sleep mode.

Let $\pi_{i,n} = \lim_{t\to\infty} \Pr[C(t) = i, N(t) = n]$ $((i,n) \in \mathcal{S})$. Our goal is to explicitly express all $\pi_{i,n}$ in terms of $\pi_{0,0}$, which is uniquely determined using the normalization condition. The balance equation for an idle server, with states $(0,n)$, reads

$$(\lambda + n\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}, \qquad n \geq 0,$$

which is identical to (1), the balance equation *without* setup time. As a result, the relation (5) between the partial generating functions $\Pi_0(z)$ and $\Pi_1(z)$ also holds true here. Opposed to this, the balance equations for a busy server, with states $(1,n)$, explicitly involve the setup parameter α, as follows:

$$(\lambda+\mu)\pi_{1,0} = \nu\pi_{0,1} + \alpha\pi_{2,0}, \tag{15}$$

$$(\lambda + (n+1)\mu)\pi_{1,n} = \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + (n+1)\nu\pi_{0,n+1} + \alpha\pi_{2,n}, \qquad n \geq 1. \tag{16}$$

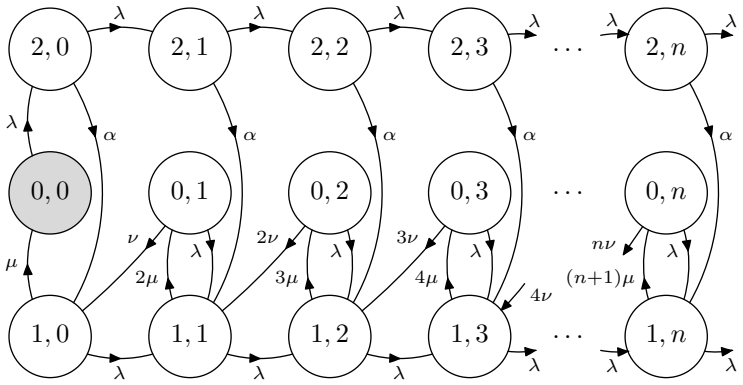


FIGURE 1. Transitions among states.

Introducing the partial generating function

$$\Pi_2(z) = \sum_{n=0}^{\infty} \pi_{2,n} z^n, \tag{17}$$

we then have

$$\lambda\Pi_1(z) + \mu\Pi_1(z) + \mu z\Pi_1'(z) = \lambda(\Pi_0(z) - \pi_{0,0}) + \lambda z\Pi_1(z) + \nu\Pi_0'(z) + \alpha\Pi_2(z).$$

The balance equations for a server in setup mode, with states $(2, n)$ are given by

$$(\lambda + \alpha)\pi_{2,0} = \lambda\pi_{0,0}, \tag{18}$$

$$(\lambda + \alpha)\pi_{2,n} = \lambda\pi_{2,n-1}, \qquad n \geq 1. \tag{19}$$

Transformation to the z-domain leads to

$$(\lambda + \alpha)\Pi_2(z) = \lambda z\Pi_2(z) + \lambda\pi_{0,0}$$

or

$$\Pi_2(z) = \frac{\lambda\pi_{0,0}}{\lambda + \alpha - \lambda z}. \tag{20}$$

Expressing the balance of flows in and out the orbit, we obtain

$$\lambda(\pi_{1,n} + \pi_{2,n}) = (n+1)\nu\pi_{0,n+1}, \qquad n \geq 0, \tag{21}$$

which yields

$$\lambda(\Pi_1(z) + \Pi_2(z)) = \nu\Pi_0'(z).$$

Multiplying both sides of the above equation by z and taking the derivative with respect to z of both sides, we get

$$\lambda[(z\Pi_1(z))' + (z\Pi_2(z))'] = \nu z\Pi_0''(z) + \nu\Pi_0'(z).$$

Using (5) to substitute $(z\Pi_1(z))'$ in terms of $\Pi_0(z)$, we find the following differential equation:

$$\lambda\frac{\lambda\Pi_0(z) + \nu z\Pi_0'(z)}{\mu} + \lambda(z\Pi_2(z))' = \nu z\Pi_0''(z) + \nu\Pi_0'(z).$$

Rearranging this equation, we obtain

$$z\Pi_0''(z) + (1 - \frac{\lambda}{\mu}z)\Pi_0'(z) - \frac{\lambda^2}{\mu\nu}\Pi_0(z) = \frac{\lambda}{\nu}(z\Pi_2(z))', \tag{22}$$

where $\Pi_2(z)$ is known, see (20). This is a non-homogeneous confluent differential equation and its explicit solution seems difficult, but we can solve it by means of a power expansion method.

In particular, substituting $\Pi_0(z) = \sum_{n=0}^{\infty} \pi_{0,n} z^n$ into the left-hand side of the differential equation (22), we obtain

$$\sum_{n=0}^{\infty} \left[(n+1)^2 \pi_{0,n+1} - \frac{\lambda}{\mu} \left(n + \frac{\lambda}{\nu} \right) \pi_{0,n} \right] z^n = \frac{\lambda^2 \pi_{0,0}}{\nu(\lambda+\alpha)} \sum_{n=0}^{\infty} (n+1) \left(\frac{\lambda}{\lambda+\alpha} \right)^n z^n,$$

where we have used

$$\Pi_2(z) = \frac{\lambda \pi_{0,0}}{\lambda+\alpha} \sum_{n=0}^{\infty} \left(\frac{\lambda z}{\lambda+\alpha} \right)^n,$$

and thus

$$(z\Pi_2(z))' = \frac{\lambda \pi_{0,0}}{\lambda+\alpha} \sum_{n=0}^{\infty} \left(\frac{\lambda}{\lambda+\alpha} \right)^n (n+1) z^n.$$

Comparison of the coefficients of z^0 in both sides yields

$$\pi_{0,1} = \frac{\lambda^2(\lambda+\mu+\alpha)}{\mu\nu(\lambda+\alpha)} \pi_{0,0}.$$

If we assume that $\pi_{0,n} = \beta_n \pi_{0,0}$ $(n \in \mathbb{Z}_+)$, it follows from the comparison of the coefficients of z^n that

$$\begin{aligned} (n+1)^2 \beta_{n+1} - \frac{\lambda}{\mu} \left(n + \frac{\lambda}{\nu} \right) \beta_n &= \frac{\lambda^2}{\nu(\lambda+\alpha)} (n+1) \left(\frac{\lambda}{\lambda+\alpha} \right)^n \\ &= \frac{\lambda}{\nu} (n+1) \left(\frac{\lambda}{\lambda+\alpha} \right)^{n+1}, \qquad n \geq 0, \end{aligned}$$

where $\beta_0 = 1$. Rearranging this equation, we obtain

$$\beta_{n+1} = \frac{\lambda}{\mu} \frac{(n+\lambda/\nu)}{(n+1)^2} \beta_n + \frac{\lambda}{\nu} \frac{(\lambda/(\lambda+\alpha))^{n+1}}{n+1}, \qquad n \geq 0,$$

with $\beta_0 = 1$. This equation allows to calculate $\pi_{0,n}$ in terms of $\pi_{0,0}$ for any $n \in \mathbb{Z}_+$. Using (1), we can also calculate $\pi_{1,n}$ in terms of $\pi_{0,0}$ for any $n \in \mathbb{Z}_+$. Determining $\pi_{0,0}$ can then be done by means of the recursion explained below in section 8.

Remark 3. As the introduction of a setup time makes the differential equation (22) non-homogeneous (as opposed to (7)), its solution is no longer a hypergeometric function. Fortunately, we are still able to solve this differential equation using a power expansion method.

5. **Linear retrial rate model with setup and deactivation.** In view of its usefulness in the numerical examples below, in this section, we extend the model of the previous section with the concept of a deactivation time. In particular, we assume that after the system becomes empty the server is not turned off immediately, but rather remains in a deactivation mode for a time interval called the deactivation time. During deactivation, new arrivals receive service immediately, without incurring a setup time. We assume that the deactivation time is exponentially distributed with mean $1/\beta$.

By adopting a convenient notation, the state space of the underlying Markov chain of the system is almost the same as in the previous section, except that the model now incorporates two states associated with an idle system, instead of one. First, there is one additional state OFF (O), corresponding to an idle system with

the server turned off. Second, the state $(0,0)$ corresponds to an idle system in deactivation. Let

$$X(t) = \begin{cases} O, & \text{the server is turned off,} \\ (C(t), N(t)), & \text{otherwise.} \end{cases}$$

It is easy to see that $\{X(t); t \geq 0\}$ forms a Markov chain on the state space $\mathcal{S}$ given by

$$\mathcal{S} = O \cup \{0,1,2\} \times \mathbb{Z}_+.$$

The transition diagram is presented in Fig. 2.

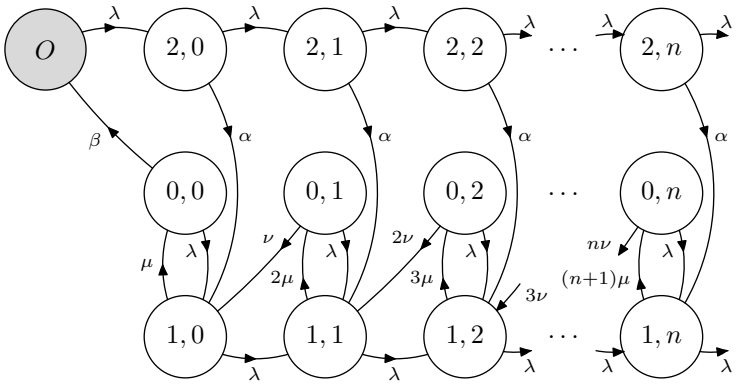


FIGURE 2. Transitions among states.

The stationary distribution is defined as before but with an additional probability $\pi_O = \lim_{t\to\infty} \Pr[X(t) = O]$. Partial generating functions are defined as before. After some tedious calculations we obtain the following results:

$$\Pi_2(z) = \frac{\lambda \pi_O}{\lambda + \alpha - \lambda z}, \qquad \lambda \pi_O = \beta \pi_{0,0}. \tag{23}$$

The differential equation for $\Pi_0(z)$ becomes the following:

$$z\Pi_0''(z) + \left(1 - \frac{\lambda}{\mu} z\right) \Pi_0'(z) - \frac{\lambda^2}{\mu\nu} \Pi_0(z) = \frac{\lambda\beta\pi_{0,0}}{\mu\nu} + \frac{\lambda}{\nu}(z\Pi_2(z))'. \tag{24}$$

From this differential equation, $\pi_{0,n}$ is expressed in terms of $\pi_{0,0}$ for all $n \in \mathbb{Z}_+$. Furthermore, $\pi_{1,n}$ is expressed in terms of $\pi_{0,n}$ as follows:

$$(\lambda + \beta)\pi_{0,0} = \mu\pi_{1,0}, \tag{25}$$

$$(\lambda + n\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}, \qquad n \geq 1. \tag{26}$$

The orbit balance equation is identical to (21), the equation *without* deactivation time. Thus, as detailed in section 8, we can use a recursion expressing all probabilities in terms of $\pi_{0,0}$ and then normalize these in order to calculate the stationary distribution.

Remark 4. Deactivation does not make an essential difference in the analysis. In particular, the differential equation (24) is almost the same as (22). Thus, we can solve this model using the same technique as for the model with setup time and without deactivation.

6. **Constant retrial rate model with setup time.** In this section, we extend the model of section 3 with the concept of a setup time, an iid random variable with exponential distribution with parameter α. Further, the state space is the same as in section 4. Finally, while the steady-state distribution is obviously different, we use the same notation as in section 4. The transition diagram is presented in Fig. 3.

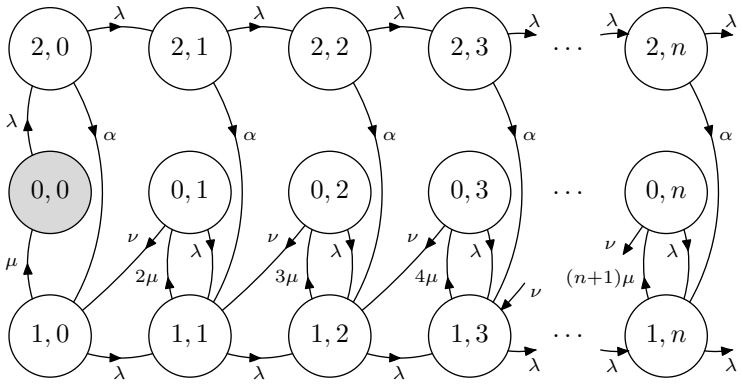


FIGURE 3. Transitions among states.

The balance equations for an idle server are identical to (8) and (9), the equations *without* setup time. Hence, the relation (12) between $\Pi_0(z)$ and $\Pi_1(z)$ remains valid here. Balance of flows in and out the orbit yields

$$\lambda(\pi_{1,n} + \pi_{2,n}) = \nu\pi_{0,n+1}, \qquad n \geq 0 \tag{27}$$

and hence,

$$\lambda(\Pi_1(z) + \Pi_2(z)) = \frac{\nu}{z}(\Pi_0(z) - \pi_{0,0}). \tag{28}$$

Multiplying both sides of the above equation by z, taking the derivative with respect to z of both sides, using (12) to substitute $(z\Pi_1(z))'$ and rearranging the result, we obtain the differential equation

$$\Pi_0'(z) = \frac{\lambda(\lambda+\nu)}{\mu\nu}\Pi_0(z) - \frac{\lambda}{\mu}\pi_{0,0} + \frac{\lambda}{\nu}(z\Pi_2(z))',$$

or

$$\Pi_0'(z) = \gamma\Pi_0(z) + \pi_{0,0}Q(z), \tag{29}$$

where $Q(z)$ and γ are defined as

$$Q(z) = -\frac{\lambda}{\mu} + \frac{\lambda}{\nu}\left(\frac{\lambda z}{\lambda + \alpha - \lambda z}\right)', \qquad \gamma = \frac{\lambda(\lambda+\nu)}{\mu\nu}.$$

It should be noted that we have used the expression (20) for $\Pi_2(z)$, which also holds here. The solution of the differential equation (29) has the form:

$$\Pi_0(z) = \pi_{0,0}\exp(\gamma z)\left(1 + \int_0^z \exp(-\gamma u)Q(u)du\right).$$

Hence, formally, we have $\Pi_0(1) = \kappa_0\pi_{0,0}$ where

$$\kappa_0 = \exp(\gamma)\left(1 + \int_0^1 \exp(-\gamma u)Q(u)du\right).$$

From (28), we also have $\Pi_1(1) + \Pi_2(1) = \kappa_1 \pi_{0,0}$, where

$$\kappa_1 = \frac{\nu}{\lambda}(\kappa_0 - 1).$$

From the normalization condition

$$\Pi_0(1) + \Pi_1(1) + \Pi_2(1) = 1,$$

we can then obtain

$$\pi_{0,0} = \frac{1}{\kappa_0 + \kappa_1}.$$

We can obtain κ_0 (and thus, also κ_1 and $\pi_{0,0}$) using numerical integration which is readily available in almost all scientific software packages. Furthermore, $\pi_{0,0}$ can also be obtained directly by means of the recursion explained in section 8.

Remark 5. The introduction of a setup time in the model implies that the differential equation involved in the analysis becomes more complicated. In particular, the second term on the right-hand side of (29) is a function of z, as opposed to a constant in (14). As a result, the solution is no longer explicit but can be formulated in an integral form.

7. **Constant retrial rate model with setup and deactivation.** Also for the constant retrial rate model, we devote a section to incorporating the deactivation time into the model presented in the previous section. Although the transition rates are different, the state space of the underlying Markov chain and the stationary distribution are defined as for the model in section 5. The transition diagram is presented in Fig. 4.

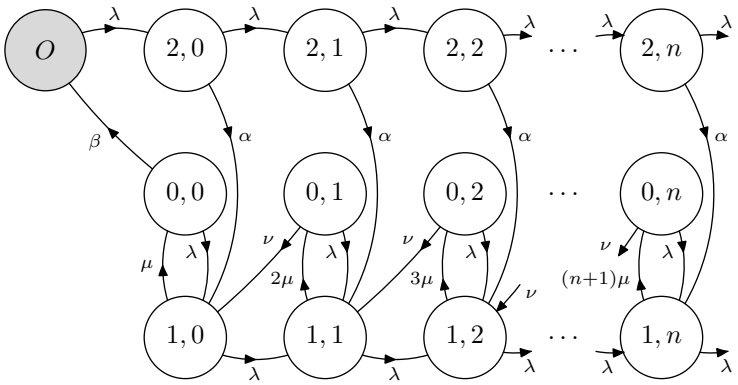


FIGURE 4. Transitions among states.

The differential equation for $\Pi_0(z)$ has the same form (29) as before, but now $Q(z)$ is given by

$$Q(z) = \frac{\lambda(\beta - \nu)}{\mu\nu} + \frac{\lambda}{\nu}\frac{(z\Pi_2(z))'}{\pi_{0,0}},$$

where

$$\Pi_2(z) = \frac{\lambda \pi_O}{\lambda + \alpha - \lambda z}, \qquad \lambda \pi_O = \beta \pi_{0,0}.$$

The probabilities $\pi_{1,n}$ can be calculated in terms of $\pi_{0,0}$ using the following equations:

$$(\lambda+\beta)\pi_{0,0} = \mu\pi_{1,0},$$
$$(\lambda+\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}. \qquad n \geq 1.$$

The orbit balance equation is identical to (27). As a result, as explained in the next section, a recursive scheme allows to calculate all probabilities $\pi_{i,n}$ in terms of $\pi_{0,0}$, which is determined using the normalization condition.

Remark 6. Similar to the case of a linear retrial rate, the introduction of a deactivation time in the model does not imply major changes in the analysis. The non-homogeneous differential equation can still be solved in either integral form or using a power expansion method.

8. **Recursive approach.** From a theoretical point of view, the results in the previous sections are nice since they are related to some well-known differential equations. However, from a practical point of view, it is more convenient to evaluate the stationary probabilities via some simple recursion.

Practically, the approach for the model of section 4 is as follows. In a first step, we set $\pi_{0,0} = 1$. In a second step, we can calculate $\pi_{2,0}$ and then $\pi_{1,0}$. Using these results, we can calculate $\pi_{0,1}$ using the balance equation in and out the orbit, i.e.,

$$(n+1)\nu\pi_{0,n+1} = \lambda(\pi_{1,n}+\pi_{2,n}), \qquad n \geq 0.$$

Next, the probability $\pi_{2,1}$ is easily calculated from the balance equation

$$(\lambda+\alpha)\pi_{2,n+1} = \lambda\pi_{2,n}, \qquad n \geq 0.$$

So, we can again use the following balance equation in order to determine $\pi_{1,1}$:

$$(\lambda+(n+1)\nu)\pi_{0,n+1} = (n+2)\mu\pi_{1,n+1}, \qquad n \geq 0.$$

The step from n to $n+1$ is taken in the same manner. As a result, we can calculate the relative values of the $\pi_{i,n}$ $(i = 0, 1, 2)$ for any value of n up to a certain value $n = N_0$, which characterizes the accuracy (the larger N_0, the better the accuracy), and then normalize the result by ensuring that the sum of the obtained probabilities is 1.

A similar procedure can be applied for the models of sections 2, 3, 5, 6 and 7. As a result, we can calculate any desired performance measure with high accuracy, by setting N_0 sufficiently high.

9. Energy-aware speed scaling.

9.1. Two cost functions.

As discussed in section 1 and in [8, 20], speed scaling enables a better balance between performance and energy consumption, involving a trade-off between the two. In this section, we use the obtained results to numerically evaluate this trade-off in different scenarios, in terms of the existing cost model applied also in [8,20]. In this model, the instantaneous cost (at an arbitrary instant) equals $N + S^\sigma/\tau$, where N (like above) denotes the number of customers in the orbit, S denotes the current service rate of the server, $\sigma > 1$ is the dynamic power parameter (denoted α in [8, 20]) and τ (denoted β in [8, 20]) controls the relative weight of delay, called delay aversion in [8, 20]. As discussed in [20], the factor S^σ models the dynamic power (excluding leakage power) used by chips when operating at speed S; correspondingly, here, it models the power consumption of the server

when running at rate S. In the analysis, we focus on the average cost per time unit, and therefore use as cost function

$$z = \mathrm{E}[N] + \mathrm{E}[S^{\sigma}]/\tau, \tag{30}$$

where z denotes the cost, $\mathrm{E}[N]$ denotes the average number of customers in the orbit, and the service rate S is a random variable that equals $(N+1)\mu$ if the server is busy ($C(t) = 1$) and zero otherwise ($C(t) \neq 1$), in all six models studied above. Accordingly, (30) can be rewritten as

$$z = \sum_{(i,n)\in\mathcal{S}} n\pi_{i,n} + \frac{\mu^{\sigma}}{\tau}\sum_{n=0}^{\infty}(n+1)^{\sigma}\pi_{1,n}. \tag{31}$$

Here, note that by virtue of Little's result, the performance-energy trade-off becomes explicit by dividing (31) by λ, since then, the first term is the mean delay, whereas the second is the mean energy per job. As in [8, 20], the focus below is on the case where $\sigma = 2$ to allow for comparison with the results reported there.

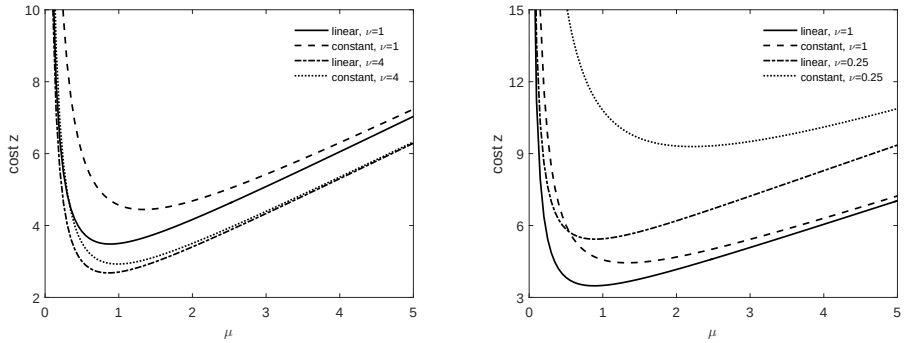
While the cost function given by (31) can be applied generically to any of the models considered in this paper, it is unable to capture the detailed behavior of the system, especially for the phases of setup and deactivation, during which the power consumption is assumed zero by (31), which is not realistic. To address this need, we propose a second cost function, which assumes that some power is consumed whenever the system is not switched off. Specifically, for the models presented in sections 5 and 7 (with deactivation), we propose the following cost function y,

$$y = \sum_{(i,n)\in\mathcal{S}} n\pi_{i,n} + \frac{\mu^{\sigma}}{\tau}\sum_{n=0}^{\infty}\{(n+1)^{\sigma}\pi_{1,n} + \varphi\pi_{2,n} + \psi\pi_{0,n}\}, \tag{32}$$

where φ and ψ represent the power consumption during setup ($C(t) = 2$) and during periods when the server is idle but the system is not switched off ($C(t) = 0$), respectively. The first parameter, the setup power consumption φ, is assumed to be matched with the power consumption for the processing of an isolated job (single job that is the only present in the system): $\varphi = 1$. For the second parameter, the idle power consumption ψ, we use the rough estimate proposed earlier in [10, 18], assuming that an idle server combined with a system that is not switched off consumes about 60% of the nominal power consumption of the mentioned isolated job: $\psi = 0.6$. Both cost functions (31) and (32) are applied in the following.

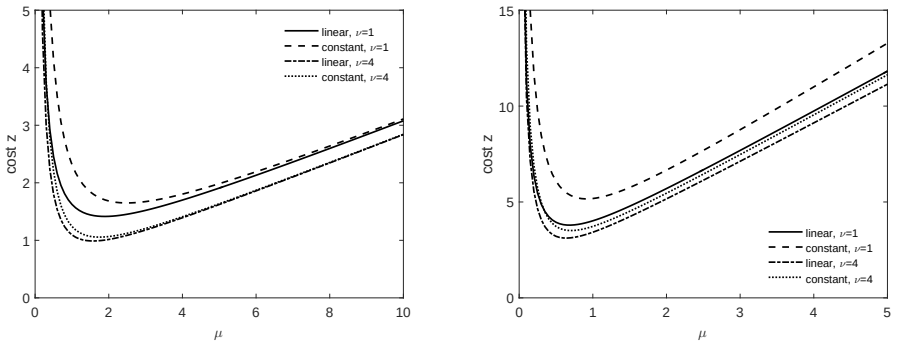
9.2. **Numerical examples.** In the numerical examples, we address the following optimization problem: Given a certain delay aversion value τ and assuming $\sigma = 2$, what is the speed parameter μ for which performance and energy are optimally balanced? To answer this question, we first use (31) as cost function in a first group of examples, and apply it to the models of sections 2, 3, 4 and 6 (all models safe those with deactivation time). As shown in [8], in the case without retrials and without setup time, this value is simply $\mu = \sqrt{\tau}$. In the case without retrials but with setup time, according to [8], while not optimal, $\mu = \sqrt{\tau}$ may perform reasonably, and provides for a robust choice in the sense that it does not take into account any a priori knowledge of the traffic statistics. In the case with retrials considered in this contribution, the retrial parameter ν inevitably impacts this relation, especially for small values of ν, when retrials take more time.

In Fig. 5(a), the cost function (31) is plotted as function of μ, with ν either equal to 1 or 4, $\tau = 1$ and $\lambda = 1$. A setting without setup times or deactivation



(a) for $\nu = 1, 4$, $\tau = 1$, $\lambda = 1$ (b) setting of Fig. 5(a) but with $\nu = 1, 0.25$

FIGURE 5. Cost z as function of the service rate μ



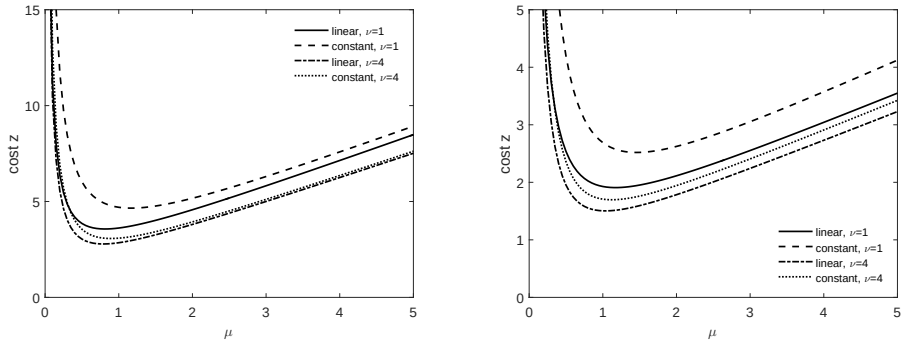
(a) setting of Fig. 5(a) but with $\tau = 4$ (b) setting of Fig. 5(a) but with setup time, $\alpha = 1$

FIGURE 6. Cost z as function of the service rate μ

times is assumed, both for a linear and a constant retrial rate, which corresponds to the models of sections 2 and 3, respectively. Curves are shown for both the case with linear retrial rate and the case with constant retrial rate. For all four curves, the trade-off is apparent, with low energy consumption on the left-hand side, low delay on the right-hand side, and an optimum (minimum) in between. In general, there is a lower cost level for a higher retrial rate, which is intuitive, as high retrial rates prevent large queues. Consequently, for a given value of ν, the curves for a constant retrial rate are above the ones for a linear retrial rate, since the effective retrial rate is higher in the linear case than in the constant case, for the same ν. Furthermore, for a large value of the retrial rate ($\nu = 4$), the optimum in the curves roughly occurs for $\mu = \sqrt{\tau} = 1$, which comes as expected. Indeed, the faster the retrials (larger ν), the closer the system resembles that of [8]. Beyond the figures, we may note that for the limit $\nu \to \infty$, the systems with linear and with constant retrial rate and that of [8] all coincide. At the other end of the scale, with small ν, Fig. 5(b) presents curves for a retrial rate value as small as $\nu = 0.25$. As can be

seen, the optimum for a linear retrial rate largely remains unimpacted, whereas the one for a constant retrial rate shifts to a higher value of μ.

To see the impact of the delay aversion, Fig. 6(a) presents curves for the same setting as Fig. 5(a) but with $\tau = 4$ instead of $\tau = 1$. Clearly, an increasing delay aversion leads to a higher optimal value of μ. In particular, the optimum in the curves still roughly occurs for $\mu = \sqrt{\tau}$, which now corresponds to $\mu = 2$.



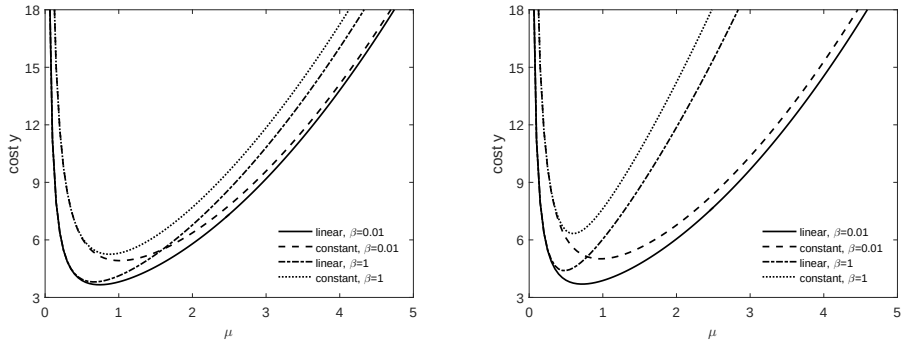
(a) setting of Fig. 6(b) but with $\alpha = 4$

(b) setting of Fig. 6(b) but with $\tau = 4$

FIGURE 7. Cost z as function of the service rate μ

While the figures discussed above consider models without setup time, we now turn to models with setup time but without deactivation time, treated in sections 4 and 6. Fig. 6(b) considers the same setting as Fig. 5(a) but with setup time, with setup rate $\alpha = 1$. Comparing both, the introduction of setup times leads to higher cost values, which is intuitive. Indeed, setup times lead to longer queues and more delay (direct effect), and therefore, given the speed scaling behavior, to higher energy consumption (indirect effect). Interestingly, the optimum for μ becomes somewhat more pronounced than without setup time, as is pointed out by Fig. 6(b). This trend is confirmed by the curves of Figs. 7(a) and 7(b), with larger setup rate for Fig. 7(a) and corresponding lower cost, and higher delay aversion in Fig. 7(b), leading to a higher optimal value of μ.

For a second group of numerical examples, we resort to the second cost function (32) to quantify the impact of the deactivation time on the balance of performance and energy consumption, using the models of sections 5 and 7. Fig. 8(a) considers the same setting as Fig. 6(b), with $\nu = 1$, but now with deactivation, with either a high deactivation rate and small average ($1/\beta$=1) or a very low rate and large average ($1/\beta$=100). In the comparison between Fig. 8(a) and Fig. 6(b), the impact of deactivation lies not so much in a cost increase or decrease (as both figures show similar cost levels) or in the position of the optimum (which remains roughly the same), but rather in a different behavior with respect to lower and higher values of μ. As can be seen in Fig. 8(a), for low values of μ, the two curves for a linear retrial rate coincide, despite the large difference in terms of deactivation rate. The same holds true for the two curves for a constant retrial rate. For high values of μ, however, the value of β is the decisive factor, with the two curves for $\beta = 1$ close together, and the two curves for $\beta = 0.01$ even closer. This can be understood as follows: The higher μ, the more often the system succeeds in emptying the queue,



(a) setting of Fig. 6(b), $\nu = 1$ but with cost y and deactivation

(b) setting of Fig. 8(a) but with $\alpha = 0.25$

FIGURE 8. Cost y (second cost function) as function of the service rate μ

and the more important the average duration of the deactivation time becomes. This effect also comes about if we stretch the periods of setup, with $\alpha = 0.25$ instead of $\alpha = 1$, the setting considered in Fig. 8(b). In such setting, the setup times are significantly larger than the deactivation times with $\beta = 1$, leading to a shift to a higher cost for the associated curves, particularly for higher values of μ, and a lowering of the optimal value of μ. The curves associated with $\beta = 0.01$ however largely remain unaffected, which can be explained by the fact that in this case, the deactivation times are still much larger on average than the setup times. As such, Figs. 8(a) and 8(b) together show that the relative durations of setup and deactivation times have a key impact on the system behavior.

10. **Conclusions.** In this paper, we studied an M/M/1 retrial queue model with speed scaling. The analysis yielded an exact solution for the steady-state queue length distribution, and this for six different cases: Two without setup times (either linear or constant retrial rate), and four with setup times, of which two augmented with a deactivation time (again, linear or constant retrial rate).

With these results available, two different cost functions enabled a detailed study of the trade-off between performance and energy consumption, inherent to speed scaling systems.

Acknowledgments. Tuan Phung-Duc was supported in part by Japan Society for the Promotion of Science, JSPS Grant-in-Aid for Young Scientists (B), Grant Number 26730011. Wouter Rogiest is Postdoctoral Fellow with the Research Foundation Flanders (FWO-Vlaanderen). Part of this research has been funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

REFERENCES

[1] J. R. Artalejo, A. Economou and M. J. Lopez-Herrero, Analysis of a multiserver queue with setup times, *Queueing Systems*, **51** (2005), 53–76.

[2] J. R. Artalejo and T. Phung-Duc, Markovian retrial queues with two way communication, *Journal of Industrial and Management Optimization*, **8** (2012), 781–806.

[3] L. A. Barroso and U. Holzle, The case for energy-proportional computing, *Computer*, **40** (2007), 33–37.
[4] R. Conway and W. L. Maxwell, A queueing model with state dependent service rate, *Journal of Industrial Engineering*, **12** (1961), 132–136.
[5] A. Gandhi, M. Harchol-Balter and I. Adan, Server farms with setup costs, *Performance Evaluation*, **67** (2010), 1123–1138.
[6] A. Gandhi, S. Doroudi, M. Harchol-Balter and A. Scheller-Wolf, Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward, *Proceedings of the ACM SIGMETRICS*, (2013), 153–166.
[7] A. Gandhi, S. Doroudi, M. Harchol-Balter and A. Scheller-Wolf, Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward, *Queueing Systems*, **77** (2014), 177–209.
[8] X. Lu, S. Aalto and P. Lassila, Performance-energy trade-off in data centers: Impact of switching delay, *Proceedings of 22nd IEEE ITC Specialist Seminar on Energy Efficient and Green Networking (SSEEGN)*, (2013), 50–55.
[9] V. J. Maccio and D. G. Down, On optimal policies for energy-aware servers, *Performance Evaluation*, **90** (2014), 36–52.
[10] I. Mitrani, Managing performance and power consumption in a server farm, *Annals of Operations Research*, **202** (2013), 121–134.
[11] P. R. Parthasarathy and R. Sudhesh, Time-dependent analysis of a single-server retrial queue with state-dependent rates, *Operations Research Letters*, **35** (2007), 601–611.
[12] T. Phung-Duc, W. Rogiest, Y. Takahashi and H. Bruneel, Retrial queues with balanced call blending: Analysis of single-server and multiserver model, *Annals of Operations Research*, **239** (2016), 429–449.
[13] T. Phung-Duc, Impatient customers in power-saving data centers, *Analytical and Stochastic Modeling Techniques and Applications, Lecture Notes in Computer Science, LNCS*, **8499** (2014), 185–199.
[14] T. Phung-Duc, Server farms with batch arrival and staggered setup, *Proceedings of the Fifth Symposium on Information and Communication Technology - ACM*, (2014), 240–247.
[15] T. Phung-Duc, Exact solutions for M/M/c/Setup queues, *Telecommunication Systems*, **64** (2017), 309–324.
[16] T. Phung-Duc, Multiserver queues with finite capacity and setup time, *Analytical and Stochastic Modeling Techniques and Applications, Lecture Notes in Computer Science, LNCS*, **9081** (2015), 173–187.
[17] T. Phung-Duc and W. Rogiest, Analysis of an M/M/1 retrial queue with speed scaling, *Proceedings of QTNA 2015, Advances in Intelligent Systems and Computing*, **383** (2015), 113–124.
[18] C. Schwartz, R. Pries and P. Tran-Gia, A queuing analysis of an energy-saving mechanism in data centers, *Proceedings of International Conference on Information Networking (ICOIN)*, (2012), 70–75.
[19] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet and P. Demeester, Trends in worldwide ICT electricity consumption from 2007 to 2012, *Computer Communications*, **50** (2014), 64–76.
[20] A. Wierman, L. Andrew and A. Tang, Power-aware speed scaling in processor sharing systems, *Proceedings of IEEE INFOCOM 2009*, (2009), 2007–2015.
[21] F. Yao, A. Demers and S. Shenker, A scheduling model for reduced CPU energy, *Proceedings 36th Annual Symposium on Foundations of Computer Science*, (1995), 374–382.'

Received October 2015; 1st revision February 2016; final revision June 2016.

E-mail address: tuan@sk.tsukuba.ac.jp
E-mail address: wouter.rogiest@ugent.be
E-mail address: sabine.wittevrongel@ugent.be