Analysis of an M/M/1 Retrial Queue with Speed Scaling

Tuan Phung-Duc and Wouter Rogiest

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 study the case without and with setup time. In all four cases, we obtain an exact solution for the stationary queue length distribution. This document presents the resulting expressions as well as their derivation.

Keywords Data center · Energy efficiency · Speed scaling · Setup time · Retrial queue

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 traditional measures like job latency, the overall energy consumption is a second important consideration. According to [4], data centers constitute about 40 % of the global ICT electricity consumption in 2012, or approximately 107 TWh. Concretely, a modern system needs mechanisms to handle the trade-off between performance and energy consumption [3].

T. Phung-Duc
Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Ookayama, Tokyo 152-8552, Japan
e-mail: tuan@is.titech.ac.jp

W. Rogiest (✉)
Department of Telecommunications and Information Processing, Ghent University, St.-Pietersnieuwstraat 41, 9000 Gent, Belgium
e-mail: wouter.rogiest@ugent.be

T.V. Do et al. (eds.), *Queueing Theory and Network Applications*,
Advances in Intelligent Systems and Computing 383,
DOI: 10.1007/978-3-319-22267-7_11

In response to this, speed scaling has been developed [6, 7, 15], slowing down server speed when the number of customers is low, and speeding up, in the converse case. As argued first in [7] (and later in [15]), this enables a better balance between performance and energy consumption. This is also argued in [19] in the context of data centers, and can be intuitively understood as follows. Assume that the speed of the system can be tuned by tuning the service rate ("speed scaling"). While power consumption rises more than proportionally with service rate (e.g., with the former approximately equal to the square of the latter [7]), this does not hold true for the mean number of customers in system. Specifically, the latter is approximately proportional to the mean service time (inverse of the service rate) in case of (very) low traffic load (with low arrival rate). Opposed to this, in case of high traffic load, speeding up can have a much larger than proportional impact on the number of customers in 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 traffic load is high than when traffic load is low, creating a trade-off. In this sense, it is useful to work at lower speed when 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 [5], presenting 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 [15], which features the concept of *switching delay* discussed also below.

While [5, 15] study a classic model without retrials, some retrial queues have been studied which also relate to the current work. Specifically, while [14] does not discuss speed scaling as such, it presents a generic study of the broad class of retrial queues with state-dependent rates, sharing many of the assumptions of this contribution. However, it is important to note that, on the one hand, [14] does not include any of the expressions derived below, and that, on the other hand, the concept of a setup time is not treated in [14], whereas it plays a key role in this contribution. Specifically, Sect. 4 and 5 below are devoted to models with a setup time, an important and realistic model extension defined below, studied earlier in e.g. [1, 10, 11, 12, 13, 16, 17, 18]. Further, the mentioned switching delay of [15] is identical to the setup time as defined in this work. Summarizing, speed scaling has already been considered in settings with setup times, and also indirectly in settings with retrial queues, but never in the combination of both. Since both phenomena are found in realistic data centers, it is useful to quantify their impact by means of the formulas derived in this contribution.

This paper is organized as follows. In Sect. 2 and 3, a speed scaling model without setup time is considered, either with classical linear retrial rate (Sect. 2) or with constant retrial rate (Sect. 3). In Sect. 4 and 5, the speed scaling model extension with finite setup time is considered, again with either linear (Sect. 4) or constant (Sect. 5) retrial rate. Sect. 6 presents a note on practical implementation. Conclusions are drawn in Sect. 7.

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, 8], 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 this section, we present a recursive scheme to calculate the joint stationary distribution. 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:

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

Let $\pi_{i,j} = \lim_{t\to\infty} \mathrm{P}(C(t) = i, N(t) = j)$ $((i, j) \in \mathcal{S})$ denote the joint stationary distribution of $\{X(t)\}$.

In this section, we derive a recursion for calculating the joint stationary distribution $\pi_{i,j}$ $((i, j) \in \mathcal{S})$. The balance equations for states with $i = 0$ read as follows.

$$(\lambda + n\nu)\pi_{0,n} = (n+1)\mu\pi_{1,n}, \qquad n \in \mathbb{Z}_+, \tag{1}$$

$$(\lambda + (n+1)\mu)\pi_{1,n} = \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + (n+1)\nu\pi_{0,n+1}, \qquad n \in \mathbb{Z}_+, \tag{2}$$

where $\mathbb{Z}_+$ denotes $\{1, 2, \ldots\}$. Using the notation of (5), we obtain the following system of equations for the partial generating functions $\Pi_0(z)$ and $\Pi_1(z)$.

$$\lambda\Pi_0(z) + \nu z\Pi_0'(z) = \mu z\Pi_1'(z) + \mu\Pi_1(z), \tag{3}$$

$$\lambda\Pi_1(z) + \mu z\Pi_1'(z) + \mu\Pi_1(z) = \lambda\Pi_0(z) + \lambda z\Pi_1(z) + \nu\Pi_0'(z). \tag{4}$$

Adding these two equations yields $\nu\Pi_0'(z) = \lambda\Pi_1(z)$. Substituting $\Pi_1(z)$ into the first equation we obtain

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

Coining 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

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

and

$$a_{(0)} = 1, \qquad a_{(n)} = a(a+1)\cdots(a+n-1), \qquad n \geq 1,$$

where $M(a,b,x)$ denotes the confluent hypergeometric function. We then have

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

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

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

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:

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

yielding

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

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.

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 and is given by $\nu(1 - \delta_{0,n})$ provided that there are n customers present in the orbit. Here, $\delta_{m,n}$ denotes the Kronecker delta, which returns 1 if $m = n$, and 0 otherwise. Again, *speed scaling* takes place: 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

In this section, we present a recursive scheme to calculate the joint stationary distribution. 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:

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

Let $\pi_{i,j} = \lim_{t\to\infty} \mathrm{P}(C(t) = i, N(t) = j)$ $((i, j) \in \mathcal{S})$ denote the joint stationary distribution of $\{X(t)\}$.

In this section, we derive a recursion for calculating the joint stationary distribution $\pi_{i,j}$ $((i, j) \in \mathcal{S})$. The balance equations for states with $i = 0$ read as follows.

$$\begin{aligned}
(\lambda + \nu(1 - \delta_{0,n})))\pi_{0,n} &= (n+1)\mu\pi_{1,n}, \qquad n \in \mathbb{Z}_+, \\
(\lambda + (n+1)\mu)\pi_{1,n} &= \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + \nu\pi_{0,n+1}, \qquad n \in \mathbb{Z}_+.
\end{aligned}$$

We define partial generating functions as follows.

$$\Pi_0(z) = \sum_{n=0}^{\infty} \pi_{0,n} z^n, \qquad \Pi_1(z) = \sum_{n=0}^{\infty} \pi_{1,n} z^n. \tag{5}$$

We obtain the following system of equations for generating functions.

$$\lambda \Pi_0(z) + \nu(\Pi_0'(z) - \pi_{0,0}) = \mu z \Pi_1'(z) + \mu \Pi_1(z), \tag{6}$$

$$\lambda \Pi_1(z) + \mu z \Pi_1'(z) + \mu \Pi_1(z) = \lambda \Pi_0(z) + \lambda z \Pi_1(z) + \frac{\nu}{z}(\Pi_0(z) - \pi_{0,0}). \tag{7}$$

Summing up these two equations yields

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

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 (6) yields

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

or

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

Solving this equation we obtain

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

where we coined 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 $\pi_{0,0} = \exp(-\gamma)$.

4 Linear Retrial Rate Model with Setup Time

In this section, we consider an extension of the model studied in Sect. 2, introducing the notion 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 Sect. 2 and 3) or the system may be in setup mode during a finite time called the setup time. In this section and the following, we assume finite iid setup times with exponential distribution with parameter α. Further, we assume that the first customer in the busy period immediately goes to the server without joining the orbit. Arriving customers who find the server occupied (either setting up or actually serving) join the orbit and repeat their attempt after some random time. Below, the terms "busy" and "active" are interchangeable, as well as "idle" and "sleeping".

Let $C(t)$ denote the state of the server and $N(t)$ denote the number of customers in the orbit at time t.

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

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

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

where $C(t) = 0, 1, 2$ implies that the server is idle, busy or in setup mode, respectively. It is easy to see that the system is always stable due to the speed scaling. Let $\pi_{i,j} = \lim_{t\to\infty} \mathrm{P}(C(t) = i, N(t) = j$. Our goal is to explicitly express all $\pi_{i,j}$ in terms of $\pi_{0,0}$ which is uniquely determined using the normalization condition.

More specially, let (0,0) denote the state corresponding to sleep mode (with thus an idle server), while $(0, j)$ $(j \geq 1)$ denotes states for which the server is idle while there are j customers in the orbit. Further, the states $(1, j)$ $(j \geq 1)$ correspond to a server busy serving a customer with j customers present in the orbit. Finally, the states $(2, j)$ $(j \geq 1)$ correspond to one customer awaiting setup in the server with j customers present in the orbit. The balance equation for an idle server reads

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

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

$$(\lambda + \mu)\pi_{1,0} = \nu\pi_{0,1} + \alpha\pi_{2,0}, \tag{8}$$

$$(\lambda + (n + 1)\mu)\pi_{1,n} = \lambda\pi_{0,n} + \lambda\pi_{1,n-1} + (n + 1)\nu\pi_{0,n+1} + \alpha\pi_{2,n}. \tag{9}$$

Introducing the generating function $\Pi_2(z) = \sum_{j=0}^{\infty} \pi_{2,j} z^j$, 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 setting up, with states $(2, j)$ are given by

$$(\lambda + \alpha)\pi_{2,0} = \lambda \pi_{0,0}, \tag{10}$$

$$(\lambda + \alpha)\pi_{2,j} = \lambda \pi_{2,j-1}, \qquad j = 1, 2 \ldots, \tag{11}$$

leading to

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

Taking the balance of flows in and out the orbit yields

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

Multiplying both sides by z and taking the derivative of both sides yields

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

Substituting $(z\Pi_1(z))'$ in terms of $\Pi_0(z)$ we find the following differential equation.

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

Reworking this equation, we obtain

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

where

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

This is a non-homogeneous confluent differential equation and its explicit solution seems difficult. But we can solve it by power expansion method.

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

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

where we have used

$$\Pi_2(z) = \frac{\lambda\pi_{0,0}}{\lambda+\alpha}\left(\sum_{j=0}^{\infty}\frac{\lambda z}{\lambda+\alpha}\right)^j, \tag{12}$$

and thus

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

Comparing the coefficients of z^0 in both sides yields,

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

Assuming that $\pi_{0,j} = \beta_j\pi_{0,0}$ ($j \in \mathbb{Z}_+$), it follows from the comparison between the coefficients of z^j that

$$\begin{aligned}(j+1)^2\beta_{j+1} - \frac{\lambda}{\mu}\left(j+\frac{\lambda}{\nu}\right)\beta_j &= \frac{\lambda^2}{\nu(\lambda+\alpha)}(j+1)\left(\frac{\lambda}{\lambda+\alpha}\right)^j, \\ &= \frac{\lambda}{\nu}(j+1)\left(\frac{\lambda}{\lambda+\alpha}\right)^{j+1}\end{aligned}$$

where $\beta_0 = 1$. This equation leads to

$$\beta_{j+1} = \frac{\lambda}{\mu}\frac{(j+\lambda/\nu)}{(j+1)^2}\beta_j + \frac{\lambda}{\nu}\frac{(\lambda/(\lambda+\alpha))^{j+1}}{j+1},$$

where $\beta_0 = 1$. This equation allows to calculate $\pi_{0,j}$ in terms of $\pi_{0,0}$ for any j. Thus, using (1), we can also calculate $\pi_{1,j}$ in terms of $\pi_{0,0}$ for any j. Determining $\pi_{0,0}$ can be done by means of the recursion explained below in Sect. 6.

5 Constant Retrial Rate Model with Setup Time

In this section, we extend the model of Sect. 3 with the notion of a setup time, an iid random variable with exponential distribution with parameter α. Further, the state space is the same as in the previous section. Finally, while the steady-state distribution is obviously different, we use the same notation as in the previous section. The balance equations are as follows.

$$\begin{aligned}\lambda\pi_{0,0} &= \mu\pi_{1,0}, \\ (\lambda+\nu)\pi_{0,n} &= (n+1)\mu\pi_{1,n}, \qquad n \geq 1.\end{aligned}$$

Transforming this equation to z-domain yields,

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

Balance of flows in and out the orbit yields

$$\lambda(\Pi_1(z)+\Pi_2(z))=\frac{\nu}{z}(\Pi_0(z)-\pi_{0,0}). \tag{13}$$

Multiplying both sides by z and taking the derivative of both sides arranging the result yields

$$\begin{aligned}\Pi_0'(z)&=\frac{\lambda(\lambda+\nu)}{\mu\nu}\Pi_0(z)-\frac{\lambda}{\mu}\pi_{0,0}+\frac{\lambda}{\nu}(z\Pi_2(z))',\\&=\gamma\Pi_0(z)+\pi_{0,0}Q(z),\end{aligned} \tag{14}$$

where

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

It should be noted that we have used

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

The solution of the differential equation (14) 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 (13), 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 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 next.

6 Recursive Approach

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

Practically, the approach for the model of Sect. 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.,

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

The probability $\pi_{2,n+1}$ is easily calculated in terms of $\pi_{0,0}$ for any n using (10) and (11).

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

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

The step from n to $n+1$ is taken in the same manner. As a result, we can calculate relative values of the $\pi_{i,n}$ ($i = 0, 1, 2$) for any value of n up to a certain value $n = N_0$ characterizing the accuracy (the larger the more accurate), 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 Sect. 2, 3 and 5. As a result, we can calculate any desired performance measure with high accuracy, by setting N_0 sufficiently high.

7 Conclusions

In this contribution, 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 four different cases: two without setup times (either linear or constant retrial rate), and two with setup times (again, linear or constant retrial rate).

With these results available, future work is to study the trade-off between performance and energy consumption, inherent to speed scaling systems. Here, a first route is by means of the existing cost function used in [7, 15]; however, this may ideally be contrasted with alternative formulations of the mentioned trade-off.

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 2673001. 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. The authors would like to thank the referees for their constructive comments which helped to improve this paper.

References

1. Artalejo, J.R., Economou, A., Lopez-Herrero, M.J.: Analysis of a multiserver queue with setup times. Queueing Systems **51**(1–2), 53–76 (2005)
2. Artalejo, J.R., Phung-Duc, T.: Markovian Retrial Queues with Two Way Communication. Journal of Industrial and Management Optimization **8**(4), 781–806 (2012)
3. Barroso, L.A., Holzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007)
4. Van Heddeghem, W., Lambert, S., Lannoo, B., Colle, D., Pickavet, M., Demeester, P.: Trends in worldwide ICT electricity consumption from 2007 to 2012. Computer Communications **50**(1), 64–76 (2014)
5. Conway, R.W., Maxwell, W.L.: A Queueing Model with State Dependent Service Rate. Journal of Industrial Engineering **12**, 132–136 (1961)
6. Yao, F., Demers, A., Shenker, S.: A scheduling model for reduced CPU energy. In: Proceedings 36th Annual Symposium on Foundations of Computer Science, pp. 374–382 (1995)
7. Wierman, A., Andrew, L., Tang, A.: Power-aware speed scaling in processor sharing systems. In: Proceedings of IEEE INFOCOM 2009, pp. 2007–2015 (2009)
8. Phung-Duc, T., Rogiest, W., Takahashi, Y., Bruneel, H.: Retrial queues with balanced call blending: analysis of single-server and multiserver model. Annals of Operations Research (2014). doi:10.1007/s10479-014-1598-2
9. Phung-Duc, T.: An explicit solution for a tandem queue with retrials and losses. Operational Research **12**(2), 189–207 (2012)
10. Phung-Duc, T.: Impatient customers in power-saving data centers. In: Sericola, B., Telek, M., Horváth, G. (eds.) ASMTA 2014. LNCS, vol. 8499, pp. 185–199. Springer, Heidelberg (2014)
11. Phung-Duc, T.: Server farms with batch arrival and staggered setup. In: Proceedings of the Fifth Symposium on Information and Communication Technology, pp. 240–247. ACM (2014)
12. Phung-Duc, T.: Exact solution for M/M/*c*/Setup queue (2014). Preprint: http://arxiv.org/abs/1406.3084
13. Phung-Duc, T.: Multiserver queues with finite capacity and setup time. In: Remke, A., Manini, D., Gribaudo, M. (eds.) ASMTA 2015. LNCS, vol. 9081, pp. 173–187. Springer, Heidelberg (2015)
14. Parthasarathy, P.R., Sudhesh, R.: Time-dependent analysis of a single-server retrial queue with state-dependent rates. Operations Research Letters **35**(5), 601–611 (2007)
15. Lu, X., Aalto, S., Lassila, P.: Performance-energy trade-off in data centers: impact of switching delay. In: Proceedings of 22nd IEEE ITC Specialist Seminar on Energy Efficient and Green Networking (SSEEGN), pp. 50–55 (2013)
16. Gandhi, A., Harchol-Balter, M., Adan, I.: Server farms with setup costs. Performance Evaluation **67**, 1123–1138 (2010)
17. Gandhi, A., Doroudi, S., Harchol-Balter, M., Scheller-Wolf, A.: Exact analysis of the M/M/k/setup class of markov chains via recursive renewal reward. In: Proceedings of the ACM SIGMETRICS, pp. 153–166 (2013)
18. Gandhi, A., Doroudi, S., Harchol-Balter, M., Scheller-Wolf, A.: Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward. Queueing Systems **77**(2), 177–209 (2014)
19. Mitrani, I.: Managing performance and power consumption in a server farm. Annals of Operations Research **202**(1), 121–134 (2013)