Server farms with batch arrival and staggered setup

Tuan Phung-Duc
Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
Ookayama, Meguro-ku, Tokyo 152-8552, Japan
tuan@is.titech.ac.jp

ABSTRACT

Cloud computing is a new paradigm where a company makes money by selling computer resources including both software and hardware. The core part of cloud computing is a data center where a huge number of servers are available. These servers consume a large amount of energy to run and to keep cool. Therefore, a reduction of a few percent of the power consumption means saving a large amount of money and the environment. In the current technology, an idle server still consumes about 60% of its peak processing a job. Thus, the only way to save energy is to turn off servers which are not processing a job. However, when there are some waiting jobs, we have to turn on the OFF servers. A server needs some setup time to be active during which it consumes energy but cannot process a job. Therefore, there exists a trade-off between power consumption and delay performance. In [8, 9], the authors analyze this trade-off using an M/M/c queue with setup time for which they present a decomposition property by solving difference equations. In this paper, using an alternative simple approach, we obtain generating functions for the joint stationary distribution of the number of active servers and that of jobs in the system for a more general model with batch arrivals. We further obtain moments for the queue size. Numerical results show some insights in to the performance of the system.

Keywords
Cloud computing, data centers, server farm, queue, setup time, staggered setup, power-saving, batch arrival

1. INTRODUCTION

Cloud computing is a new paradigm where companies make money by providing computing service through the Internet. In cloud computing, users buy software and hardware resources from a provider and access to these resources through the Internet so they do not have to install and maintain by themselves. The core part of cloud computing is a data center where there are a huge number of servers. The key issue for the management of data centers is to minimize the power consumption while keeping an acceptable service level for customers. It is reported that under the current technology an idle server still consumes about 60% of its peak processing jobs [3]. Thus, the only way to save power is to turn off idle servers. However, if the workload increases, OFF servers should be turned on to serve waiting customers. Servers need some setup time during which they consume energy but cannot process jobs. Therefore, customers have to wait a longer time in comparison with the case where the servers are always ON.

Although queues with setup time have been extensively investigated in the literature, most of papers deal with single server models [22, 24, 5, 6] where the service time follows a general distribution. Artalejo et al. [2] present a throughput analysis for multiserver queues with setup time where the authors consider the case in which at most one server can be in setup mode at a time. This policy is referred to as staggered setup in [9]. It is pointed out in [2] that the model belongs to a QBD class for which the rate matrix is explicitly obtainable. By solving difference equations, Artalejo et al. [2] derive an analytical solution where the stationary distribution is recursively obtained without any approximation. Recently, motivated by applications in data centers, multiserver queues with setup time have been extensively investigated in the literature. In particular, Gandhi et al. [10, 11, 12, 13] analyze multiserver queues with setup time. They consider the M/M/c system with staggered setup and derive some closed-form approximations for the ON/OFF policy where the number of servers in the setup mode at a time is not limited. Gandhi et al. [11] extend their analysis to the case where a free server waits for a while before shutdown. As a related model, Tian et al. [23] consider M/M/c model with vacation where after a service an idle server leaves for an exponentially distributed vacation.

In all the work on multiserver queue mentioned above, customers (jobs) are assumed to arrive individually according to a Poisson process. However, in cloud computing a big task might be divided into multiple subtasks to process in parallel [7]. This motivates us to consider a multiserver queuing system with setup time under batch arrival settings. In this paper, using a generating function approach, we derive a clear solution for all the partial generating functions for the joint stationary distribution of the number of active servers and that of customers in the system. The generating functions are obtained using recursive formulae. A special case of our model conforms to the model presented in [9]. Furthermore, we derive a recursion which allows cal-
Calculating all the moments of the queue length.

Some more related work are as follows. Mitra [19, 18] considers a model for server farms with setup cost. The author considers the case where group of reserve servers are shutdown concurrently if the workload is lower than some lower threshold and is powered up simultaneously when the workload exceeds some upper threshold. Because of this simultaneous setup, the underlying Markov chain in [19] has a simple birth-and-death structure allowing closed form solutions. The author investigates the optimal lower and upper thresholds for the system. The same author [17] extends their analysis to the case where each customer has an exponentially distributed random timer exceeding which the customer abandons the system. Muzzucho and Mitra [15] show that the theoretical results in [17, 18] fit that of real experiments. Schwartz et al. [21] consider a similar model to that in [17]. The main purpose of these papers is to find the optimal thresholds using which the group of reserve servers is turned ON or OFF. Muzzucho et al. [14] use the Erlang-C and Erlang-\(\Lambda\) formulae to approximate server farms with setup costs. For more literature on multiserver models with vacation, we refer to [25, 26]. In addition, for discussions on power consumption issue in data centers and cloud computing, we refer to [3, 4, 14, 15, 16, 20, 21].

The rest of our paper is organized as follows. First we present the model in Section 2. Section 3 is devoted to the detailed analysis where we derive the partial generating functions and the joint stationary distribution. In Section 4, we discuss the decomposition property for the queue length. In Section 5, we provide extensive numerical results to validate the decomposability in Section 4 and to show the performance of the system.

2. MODEL

We consider \(M/M/c\) queueing systems with staggered setup. Customers arrive at the system in batch according to a Poisson process with rate \(\lambda\). We assume that the batch size distribution is \(\beta_i\) \((i \in \mathbb{N} = \{1, 2, \ldots\}\) and its generating function is given by \(\beta(z)\). In this system, an idle server is turned off immediately. If there are some waiting customers, OFF servers are turned on one by one. Furthermore, a server needs some setup time to be active so as to serve a waiting customer. We assume that the setup time follows the exponential distribution with mean \(1/\alpha\). If a server finishes a job, this server picks a waiting customer if any. If there is not a waiting customer, the server in setup process and idle ones are turned off immediately. Let \(j\) denotes the number of customers in the system and \(i\) denotes the number of active servers. The number of servers in setup process is \(\min(j - i, 1)\). Under these assumptions, the number of active servers is smaller than or equal to the number of customers in the system. It should be noted that in this model a server is in either BUSY or OFF or SETUP. We assume that every customer that enters the system receives service and departs. This means that there is no abandonment. We assume that the service time of jobs follows an exponential distribution with mean \(1/\mu\).

Remark 1. The staggered setup is suitable for several technical reasons. First, setting up a server needs a instantaneously large amount of energy [6]. Thus, it is not desired to have multiple servers setting up at the same time. This model also fits for a manufacturing system context where all the servers are monitored by one administrator [2]. In this case, the administrator turns on the OFF servers one by one. Second, from an analytical point of view, the staggered setup makes the analysis simpler since the underlying Markov chain has a homogeneous structure which is suitable for the use of generating functions.

3. ANALYSIS OF THE MODEL

3.1 Generating functions

Let \(C(t)\) and \(N(t)\) denote the number of busy servers and the number of jobs in the system at time \(t\), respectively. Under the assumptions made in Section 2, it is easy to see that \(\{X(t) = (C(t), N(t)); t \geq 0\}\) forms a Markov chain in the state space

\[S = \{(i, j); j \in \mathbb{Z}_+, i = 0, 1, \ldots, \min(c, j)\},\]

where \(\mathbb{Z}_+ = \{0, 1, \ldots\}\). See Figure 1 for the transitions among states for the case of single arrival, i.e., \(\beta(z) = z\).

In this paper, we assume that \(\rho = \lambda/\alpha c < 1\) which is the necessary and sufficient condition for the stability of the Markov chain. In what follows, we assume that the Markov chain is ergodic. Under this ergodic condition, let

\[\pi_{i,j} = \lim_{t \to \infty} P(N(t) = i, C(t) = j), \quad (i, j) \in S,\]

denote the stationary probability of state \((i, j)\).

The balance equations for states \((0, j)\) \((j \in \mathbb{N})\) read as follows.

\[\lambda \sum_{t=1}^{j} \beta_t \pi_{0,j-t} = (\lambda + \alpha) \pi_{0,j}, \quad j \in \mathbb{N}.\]

Let \(\Pi_0(z) = \sum_{j=0}^{\infty} \pi_{0,j} z^j\). Multiplying the above equation by \(z^j\) and adding over \(j \in \mathbb{N}\) yields,

\[\lambda \beta(z) \Pi_0(z) = (\lambda + \alpha) (\Pi_0(z) - \pi_{0,0}),\]

or equivalently

\[\Pi_0(z) = \frac{(\lambda + \alpha) \pi_{0,0}}{\lambda + \alpha - \lambda \beta(z)}.\] (1)

The balance equation for state \((0, 0)\) is given by

\[\lambda \pi_{0,0} = \mu \pi_{1,1}.\]
This equation is also derived from the balance between flows in and out the group of states \( \{(0, j); j \in \mathbb{Z}_+\} \). Indeed, we have
\[
\alpha(\Pi_0(1) - \pi_{0,0}) = \mu \pi_{1,1},
\]
leading to
\[
\pi_{1,1} = \frac{\alpha(\Pi_0(1) - \pi_{0,0})}{\mu} - \frac{\lambda}{\mu} \pi_{0,0}.
\]
Now, we shift to the case where there is one active server, i.e., \( i = 1 \). We have
\[
(\lambda + \mu)\pi_{1,1} = \alpha \pi_{0,1} + \mu \pi_{1,2} + 2 \mu \pi_{2,2}, \quad j = 1, \quad (2)
\]
\[
(\lambda + \mu + \alpha)\pi_{1,j} = \lambda \sum_{i=1}^{j-1} \beta_i \pi_{1,j-i} + \alpha \pi_{0,j} + \mu \pi_{1,j+1}, \quad j \geq 2. \quad (3)
\]
We define the generating for the states with \( i = 1 \) as follows.
\[
\Pi_1(z) = \sum_{j=0}^{\infty} \pi_{1,j+1} z^j.
\]
\( \Pi_1(z) \) represents the generating function of the number of waiting customers while there is one active server.

Multiplying (2) by \( z^0 \) and (3) by \( z^{j-1} \) and taking the sum over \( j \in \mathbb{N} \) yields,
\[
(\lambda + \mu + \alpha)\Pi_1(z) - \alpha \pi_{1,1} = \lambda \beta(z)\Pi_1(z) + \frac{\alpha}{z}(\Pi_0(z) - \pi_{0,0}) + \mu \pi_{2,2}. \quad (4)
\]
Arranging (4) we obtain
\[
f_1(z) = \frac{\lambda + \mu + \alpha - \lambda \beta(z) - \mu}{\mu - \lambda \beta(z)} \pi_{1,1} \Pi_1(z) = \alpha \pi_{0,1} z + \frac{\alpha}{z}(\Pi_0(z) - \pi_{0,0}) + \mu \pi_{2,2},
\]
where \( f_1(z) = (\lambda + \mu + \alpha)z - \lambda \beta(z) - \mu \). Because \( f_1(0) = -\mu < 0 \) and \( f_1(1) = \alpha > 0 \), \( 0 < \lambda z_1 < 1 \) such that \( f_1(z_1) = 0 \). Furthermore, Rouche's theorem (see e.g., [1]) shows that \( z_1 \) is the unique root in the unit circle. Since \( \Pi_1(z) \) converges in \( |z| \leq 1 \), letting \( z = z_1 \) yields,
\[
\pi_{2,2} = \left( \frac{\mu - \alpha z_1}{\mu z_1} \right) \pi_{1,1} + \alpha \pi_{0,0} - \Pi_0(z_1).
\]
It should be noted that for the case \( \beta(z) = z \), i.e., single arrival, we have
\[
z_1 = \frac{\lambda + \mu + \alpha - \sqrt{(\lambda + \mu + \alpha)^2 - 4 \lambda \mu}}{2 \lambda}.
\]
Remark 2. At this point, we have expressed \( \Pi_1(z) \) and \( \pi_{2,2} \) in terms of \( \pi_{0,0} \).

Next, we shift to the case where there are \( i (2 \leq i \leq c-1) \) active servers. The balance equations read as follows.
\[
(\lambda + i \mu)\pi_{i,i} = \alpha \pi_{i-1,i} + i \mu \pi_{i,i+1} + (i + 1) \mu \pi_{i+1,i+1}, \quad (6)
\]
\[
(\lambda + i \mu + \alpha)\pi_{i,j} = \lambda \sum_{k=0}^{j-i} \beta_k \pi_{i,j-k} + i \mu \pi_{i,j+1} + \alpha \pi_{i-1,j}, \quad (7)
\]
for \( j \geq i + 1 \). We define the partial generating function for the case of having \( i \) active servers as follows.
\[
\Pi_i(z) = \sum_{j=i}^{\infty} \pi_{i,j} z^{j-i}, \quad i = 2, 3, \ldots, c - 1.
\]
Multiplying (6) by \( z^0 \) and (7) by \( z^{j-1} \) and adding over \( j = i, i + 1, \ldots \), we obtain
\[
(\lambda + i \mu + \alpha)\Pi_i(z) - \alpha \pi_{i,i} = \lambda \beta(z)\Pi_i(z) + \frac{\mu}{z}(\Pi_{i-1}(z) - \pi_{i-1,i}) + (i + 1) \mu \pi_{i+1,i+1},
\]
or equivalently
\[
f_i(z) = \Pi_i(z) - \alpha \pi_{i,i} = (i + 1) \mu \pi_{i+1,i+1} - iz \beta(z) - i \mu. \quad (8)
\]
where \( f_i(z) = (\lambda + i \mu + \alpha)z - \lambda \beta(z) - i \mu \). Since \( f_i(0) = -i \mu < 0 \) and \( f_i(1) = \alpha > 0, 0 < \lambda z_i < 1 \) such that \( f_i(z_i) = 0 \). Rouche's theorem also shows that \( z_i \) is the unique root inside the unit circle. For the case of single arrival, i.e., \( \beta(z) = z \), we have
\[
z_i = \frac{\lambda + i \mu + \alpha - \sqrt{(\lambda + i \mu + \alpha)^2 - 4 i \lambda \mu}}{2 \lambda}.
\]
Putting \( z = z_i \) into (8), we obtain
\[
\pi_{i+1,i+1} = \frac{(i \mu - \alpha z_i) \pi_{i,i} + \alpha (\pi_{i-1,i-1} - \Pi_{i-1}(z_i))}{(i + 1) \mu z_i}, \quad (9)
\]
\( i = 1, 2, \ldots, c - 1 \).

Remark 3. At this point, we have expressed the generating functions \( \Pi_i(z) \) \((i = o, 1, \ldots, c-1)\) and boundary probabilities \( \pi_{i,i} \) \((i = 0, 1, \ldots, c)\) in terms of \( \pi_{0,0} \).

Finally, we consider the case \( i = c \), i.e., all servers are active. Balance equations are given as follows.
\[
(\lambda + c \mu)\pi_{c,c} = \alpha \pi_{c-1,c} + c \mu \pi_{c,c-1}, \quad (10)
\]
\[
(\lambda + c \mu)\pi_{c,j} = \alpha \pi_{c-1,j} + \lambda \sum_{i=1}^{j-c} \beta_i \pi_{c,j-i} + c \mu \pi_{c,j+1}, \quad (11)
\]
\( j \geq c + 1 \).

We define the generating function for the case \( i = c \) as follows.
\[
\Pi_c(z) = \sum_{j=c}^{\infty} \pi_{c,j} z^{j-c}.
\]
Multiplying (10) by \( z^0 \) and (11) by \( z^{j-c} \) and summing over \( j \geq c \), we obtain
\[
(\lambda + c \mu)\Pi_c(z) = \frac{\alpha}{z}(\Pi_{c-1}(z) - \pi_{c-1,c-1}) + \frac{c \mu}{z}(\Pi_c(z) - \pi_{c,c}) + \lambda \beta(z)\Pi_c(z),
\]
or equivalently
\[
\Pi_c(z) = \frac{\alpha(\Pi_{c-1}(z) - \pi_{c-1,c-1}) - c \mu \pi_{c,c}}{f_c(z)},
\]
where \( f_c(z) = (\lambda + c \mu)z - \lambda \beta(z) - c \mu \). The numerator of \( \Pi_c(z) \) vanishes at \( z = 1 \) due to the balance between the flows in and out the group of states \( \{(c, j); j = c, c + 1, \ldots \} \). Thus, applying L'Hopital's rule and arranging the results yields
\[
\Pi_c(1) = \frac{\alpha \Pi_{c-1}(1)}{c \mu - \lambda \beta'(1)}, \quad (12)
\]
Remark 4. It should be noted that we have expressed \( \Pi_i(z) \) \((i = 0, \ldots, c)\) in terms of \( \pi_{0,0} \), which is uniquely determined by the following normalization condition:

\[
\sum_{i=0}^{c} \Pi_i(1) = 1.
\]  

(13)

According to (12), in order to calculate \( \Pi_i(1) \), we need \( \Pi_{i-1}(1) \) which is recursively obtained by Theorem 3.1.

In Section 3.2, we show some simple recursive formulae for the partial factorial moments.

### 3.2 Factorial moments

In this section, we derive simple recursive formulae for factorial moments. Because the generating function \( \Pi_0(z) \) is given in a simple form, its derivatives at \( z = 1 \) are also explicitly obtained in a simple form.

Theorem 3.1. The first partial moments of the queue length is recursively calculated as follows.

\[
\Pi_i'(1) = \Pi_{i-1}'(1) + \frac{\lambda \beta'(1) - \alpha - i \mu}{\alpha} \Pi_i(1)
\]

\[
+ \left( (i+1) \mu \pi_{i+1,1,i+1} + \alpha \pi_{i,i}, \quad i = 1, 2, \ldots, c - 1, \right.
\]

\[
\left. \frac{\alpha}{\alpha} \Pi_i(1) \right) = 1, 2, \ldots, c - 1,
\]

(14)

where \( \Pi_i(1) = \pi_{0,0} \beta'(1)(\lambda + \alpha)/\alpha^2 \). Furthermore, the \( n \)-th (\( n \geq 2 \)) partial factorial moment is given by

\[
\Pi^{(n)}(1) = \Pi^{(n-1)}(1) + \frac{n(\lambda \beta'(1) - i \mu - \alpha) \Pi^{(n-1)}(1)}{\alpha}
\]

\[
+ \sum_{k=2}^{n} n C_k \left( \lambda \beta(k)(1) + k \lambda \beta(k-1)(1) \right) \Pi^{(n-k)}(1),
\]

(15)

where \( \Pi^{(n)}(1) = n! \pi_{0,0} (\lambda \beta'(1))^n(\lambda + \alpha)/\alpha^{n+1} \).

Proof. Differentiating (8), we obtain

\[
f_i(z) \Pi_i'(z) = -(\mu + \alpha - \lambda \beta'(z)) \Pi_i(z) + \alpha \Pi_i(z) + \alpha \pi_{i,i}
\]

\[
+ (i+1) \mu \pi_{i+1,i+1,1,i+1}.
\]

Substituting \( z = 1 \) into the above equation and rearranging the result yields (14). Differentiating (8) for \( n \geq 2 \) times at \( z = 1 \) and arranging the result, we obtain (15). \( \square \)

Theorem 3.2. We have

\[
\Pi^{(n)}(1) = \frac{A_n}{n+1(\mu - \lambda \beta'(1))}, \quad n \in \mathbb{N},
\]

(16)

where

\[
A_n = \alpha \Pi^{(n+1)}(1)
\]

\[
+ \sum_{k=2}^{n+1} n C_k \left( k \lambda \beta(k+1)(1) + \lambda \beta(k)(1) \right) \Pi^{(n-k)}(1).
\]

Proof. We have

\[
f_c(z) \Pi_c(z) = \alpha (\pi_{c-1}(z) - \pi_{c-1,c-1}) - c \mu \pi_{c,c}.
\]

Differentiating this equation \( n \geq 1 \) times, we obtain

\[
f_c(z) \Pi_c^{(n)}(z) + \sum_{k=1}^{n} n C_k f_c^{(k)}(z) \Pi^{(n-k)}(z) = \alpha \Pi^{(n-k)}(z),
\]

where \( \Pi^{(n-k)}(z) = 0, \forall |z| < 1 \). Arranging this equation leads to

\[
\Pi^{(n)}(z) = \alpha \Pi^{(n-1)}(z) - \sum_{k=1}^{n} n C_k f_c^{(k)}(z) \Pi^{(n-k)}(z). \]  

(17)

We observe inductively that both the denominator and numerator in the right-hand side of (17) vanish at \( z = 1 \). Thus, applying L’Hospital’s rule and arranging the result, we obtain (16). \( \square \)

Remark 5. It should be noted that in order to obtain the \( n \)-th factorial moment \( \Pi^{(n)}(1) \), we need to have the \( (n+1) \)-th factorial moment \( \Pi^{(n+1)}(1) \). Fortunately, \( \Pi^{(n+1)}(1) \) is expressed in terms of \( \Pi^{(n+1)}(1) \) which is explicitly obtained for any \( n \) according to Theorem 3.1.

### 4. SPECIAL CASES

In this section, we devote to the decomposition property of the queue length where we show the single server system in Section 4.1 and discuss the multiserver model in Section 4.2. We restrict ourself to the case of single arrival, i.e., \( \beta(z) = z \).

#### 4.1 Single server

We consider the single server case. The partial generating functions are given as follows.

\[
\Pi_0(z) = \frac{(1-\rho)\alpha}{\lambda + \alpha - \lambda z}, \quad \Pi_1(z) = \frac{(1-\rho)\alpha^2}{(\mu - \lambda z)(\lambda + \alpha - \lambda z)}.
\]

where \( \rho = \lambda/\mu \). Let \( \Pi(z) \) denote the generating function of the number of waiting customers. We have

\[
\Pi(z) = \Pi_0(z) + \Pi_1(z) = (1-\rho) \left( 1 + \frac{\rho}{1-\rho^2} \right) \frac{\alpha}{\lambda + \alpha - \lambda z}.
\]

It should be noted that

\[
(1-\rho) \left( 1 + \frac{\rho}{1-\rho^2} \right)
\]

and

\[
\frac{\alpha}{\lambda + \alpha - \lambda z}
\]

represent the generating function of the number of waiting customers in the corresponding \( M/M/1 \) queue without setup time and that of customers arriving in the remaining setup time, respectively. Thus, we have

\[
L \overset{\triangle}{=} L_1 + L_2,
\]

where the \( L \) is the queue length of the current model while \( L_1 \) and \( L_2 \) represent the queue length of the conventional \( M/M/1 \) queue and the number of customers that arrive during the remaining setup time.

#### 4.2 Multiserver

In this section, we investigate the decomposability of the queue length. In particular we answer the question: does equation (18) hold?

\[
L \overset{\triangle}{=} L_1 + L_2, \tag{18}
\]
where $L_1$ is the queue length of the M/M/c without setup time and $L_2$ is the number of customers that arrive to the queue during the remaining setup time.

The generating function for the number of waiting customers in the conventional M/M/c queueing system is given by $1 - C(c,p,c) + C(c,p,c)(1 - \rho)/(1 - \rho z)$ where $\rho = \lambda/(c\mu)$ and $C(c,p,c)$ is the Erlang C formula for the waiting probability in the conventional M/M/c system without setup time. Therefore, if the decomposition result is established the generating function of the number of waiting customers in the system with setup time $\Pi(z)$ must be given by the following formula.

$$
\Pi(z) = \frac{\alpha}{\alpha + \lambda - \lambda z} \left(1 - C(c,p,c) + C(c,p,c)\frac{1 - \rho}{1 - \rho z}\right).
$$

In [9] the authors state that the decomposition property is held for the model meaning that (19) is true.

We prove this property. Indeed for the case where $\beta(z) = z$ after some tedious calculations, we find that

$$
\Pi_i(z) = \pi_{i,1} \frac{\lambda + \alpha}{\lambda + \alpha - \lambda z}, \quad \pi_{i,1} = \pi_{0,0} \left(\frac{\lambda}{\mu}\right)^i \frac{1}{i!},
$$

for $i = 0, 1, \ldots, c - 1$ and

$$
\pi_{c,c} = \pi_{0,0} \left(\frac{\lambda}{\mu}\right)^c \frac{1}{c!}, \quad \Pi_i(z) = \pi_{c,c} \frac{\lambda + \alpha}{(1 - \rho z)(\lambda + \alpha - \lambda z)}.
$$

It follows from $\Pi(z) = \sum_{i=0}^c \Pi_i(z)$ and $\Pi(1) = 1$ that (19) is true.

From the decomposition result for the queue length, we obtain the decomposition result for the waiting time via distributitional Little's law. In particular, we have

$$
W = W_1 + W_2,
$$

where $W$ denotes the waiting time in the current system while $W_1$ and $W_2$ are the waiting time in the corresponding M/M/c system without setup time and the setup time, respectively.

Remark 6. From the generating function, we obtain explicit expressions for the joint stationary distribution as follows.

$$
\pi_{i,j} = \pi_{i,1} \left(\frac{\lambda}{\lambda + \alpha}\right)^{j-i}, \quad j = i, i+1, \ldots, i = 0, 1, \ldots, c-1.
$$

Furthermore, if $\rho \neq \phi_0 = \lambda/(\lambda + \alpha)$, we have

$$
\pi_{c+1,c} = \pi_{c,c} \left(\frac{\rho_{k+1}}{\phi_0 - \rho_{k+1}}\right), \quad k \geq 0.
$$

If $\rho = \lambda/(\lambda + \alpha)$, we have

$$
\pi_{c+1,c} = \pi_{c,c} (k+1)^\rho, \quad k \geq 0.
$$

5. NUMERICAL EXAMPLES

In all numerical experiments we fix $\mu = 1$. Furthermore, we consider the case of Poisson arrival where $\beta(z) = z$ for which $z_i (i = 1, 2, \ldots, c - 1)$ is explicitly obtained.

5.1 Queue length decomposition

In this section, we show the decomposition property for the queue length of our system. We fix the number of servers by $c = 10$.

5.1.1 Mean number of waiting customers

In Figures 2, 3 and 4, we show the mean number of waiting customers $E[L]$ and its corresponding decomposition version $E[L_1] + E[L_2]$ against $\lambda$ for $\alpha = 0.001, 0.0001, \alpha = 0.1, 0.01$ and $\alpha = 1, 10, 100$, respectively. In these cases, we observe that

$$
E[L] = E[L_1] + E[L_2],
$$

implying the decomposition property for the mean.

5.1.2 Coefficient of variation

In Figures 5, 6 and 7, we show the coefficient of variation against $\lambda$ for $\alpha = 0.001, 0.0001, \alpha = 0.1, 0.01$ and $\alpha = 1, 10, 100$, respectively. We observe that the coefficients of variation for $L$ and $L_1 + L_2$ are the same meaning that the decomposition property is true.

5.1.3 Mean waiting time

Figures 8, 9 and 10 show the mean waiting time against the traffic intensity. We observe that the mean waiting time increases with the traffic intensity as expected. Furthermore, observe that the waiting time is dominated by the setup time for the cases where $\alpha$ is relatively small. We
Figure 4: Mean number of waiting customers for $\alpha = 1, 10, 100$.

Figure 5: Coefficient of variation for $\alpha = 0.001, 0.0001$.

Figure 6: Coefficient of variation for $\alpha = 0.1, 0.01$.

Figure 7: Coefficient of variation for $\alpha = 1, 10, 100$.

Figure 8: Mean waiting time for $\alpha = 0.001, 0.0001$.

Figure 9: Mean waiting time for $\alpha = 0.1, 0.01$. 

Mean Waiting Time
Arrival Rate (\lambda)
\alpha = 0.1 (Original)
\alpha = 0.1 (Decompose)
\alpha = 0.01 (Original)
\alpha = 0.01 (Decompose)
also observe that the waiting time calculated by our model is consistent with that by formula (20). This confirms that the decomposition is established.

5.2 Power consumption vs. traffic intensity

Figure 11 shows the total power consumption against the offered traffic load $\lambda/\mu = cp$ where the number of servers is fixed to $c = 10$. The cost per unit time for each state: SETUP, ON and IDLE of a server is set as follows: $C_{\text{setup}} = 5$, $C_{\text{run}} = 1$ and $C_{\text{idle}} = 0.6$. The power consumption of our system with staggered setup is given by

$$P_{\text{ON/OFF-staggered}} = C_{\text{setup}}(1 - \sum_{i=0}^{c-1} \pi_i - \Pi_c(1)) + C_{\text{run}}cp, \quad (21)$$

where $cp = \lambda/\mu$ is the mean number of running servers. We plot three curves corresponding to the cases $\alpha = 0.1, 1, 10$. For comparison, we also plot the curves for the conventional $M/M/c$ queue under the same setting. It should be noted that in the conventional $M/M/c$ system, an idle server is not turned off. As a result, the cost for power consumption is given by

$$P_{\text{ON/IDLE}} = C_{\text{run}}cp + C_{\text{idle}}(c - cp). \quad (22)$$

We observe in Figure 11 that a curve of the conventional $M/M/c$ system crosses the corresponding one of the $M/M/c$-Staggered at some points $\rho_0$ which is increases with the setup rate $\alpha$. This suggests that the shorter the mean setup time, the longer the range in which the $M/M/c$-Staggered outperforms the conventional $M/M/c$-ON/IDLE system. In the other words, if $\rho < \rho_0$, $M/M/c$-Staggered outperforms the conventional $M/M/c$-ON/IDLE system from a power consumption point of view, while the latter is more power saving than the former under a relatively high load, i.e., $\rho > \rho_0$.

6. CONCLUSION

In this paper, we have considered the $M^X/M/c$ queuing system with staggered setup where only one server can be in setup mode at a time. A server is turned off immediately after serving a job and there is no waiting customer. If there are some waiting customers, OFF servers are turned on one by one. Using a generating function approach, we have obtained the partial generating functions of the queue length. We also have obtained recursive formulae for computing the factorial moments of the number of waiting jobs. We have derived the decomposition property obtained in [8] using the generating function approach and have verified it by numerical examples. Numerical experiments have shown some insights into the performance of the system. For future work, it may be interesting to carry out experiments in real cloud or simulation environment like CloudSim [27] to validate the result of this paper. Furthermore, it is also important to consider the case where a fixed number of servers are always kept ON in order to reduce the delay of customers.

7. REFERENCES


