A Performance Model for Maintenance Tasks in an Environment of Virtualized Servers
(Work in Progress)

Tien Van Do[1,⋆] and Udo R. Krieger[2]

[1] Department of Telecommunications
Budapest University of Technology and Economics
H-1117, Magyar tudósok körútja 2., Budapest, Hungary
do@hit.bme.hu
[2] Faculty Information Systems and Applied Computer Sciences
Otto-Friedrich-Universität, D-96045 Bamberg, Germany
udo.krieger@ieee.org

Abstract. This paper introduces the CPP/M/c model with working vacations to describe queueing phenomena that arise in an advanced computing environment of virtualized servers operated by the infrastructure owners. In the proposed queue the inter-arrival times of jobs requesting servers follow a Generalized Exponential distribution. To model a maintenance activity, we assume that a certain number of servers simultaneously goes to a maintenance state for a random period when they complete the service of requests and find no further jobs in the waiting line. We derive an expression for the steady-state probabilities and prove a conditional stochastic decomposition property. By a relatively simple model we are able to prove a property which has a significant impact on the organization of maintenance activities of virtualized servers. It means that instead of migrating virtual servers to expensive physical backup servers during software maintenance, a wise and simple strategy based on the vacation approach can be used. Moreover, it is theoretically proved that the system is not overloaded if we organize the maintenance according to the vacation model. We believe that our model can be useful for administrators to choose an appropriate parameter set for the maintenance activities.

Keywords: Virtualized services, performance management, CPP/M/c model, working vacations policy.

1 Introduction

At present, virtualization constitutes a main trend in information systems and advanced business engineering. Recent studies have shown that a proportion of 39% among 808 of the largest companies worldwide apply server virtualization

[⋆] Corresponding author.

L. Fratta et al. (Eds.): NETWORKING 2009, LNCS 5550, pp. 931–942, 2009.

to achieve new business goals and to provide more efficient services to their customers. Disaster recovery, avoidance of service outage and dynamic load balancing represent some of the most important areas for the application of the rapidly evolving virtualization concepts. Compared to existing service technologies 25% of the cost or even more can be saved by these means.

In this context, virtualization means either to let a federation of servers appear as multiple computing entities or to let many computing entities appear as a single computer. The latter is commonly called server aggregation or grid computing. It is identified by an IDC research report (http://www.idc.com) that virtualization of system resources in severs with an x86 compatible instruction set is one disruptive technology. In the near future it may initiate a paradigm shift in IT industry providing new powerful services like enhanced server hosting.

As indicated by various studies it is the rationale behind this trend that virtualization can reduce the infrastructure and IT management cost. The reason is that it substantially improves the utilization of the physical infrastructure, i.e. servers, storage systems and network components, while it can provide the same safety and performance compared to a solution where each ASP obtains a separate physical machine/server from the owner of the infrastructure. It is another advantage that the infrastructure can provide in a flexible manner different service packages concerning specific operating systems running on top of the same hardware.

From a practical perspective, it is observed that virtualization is a well founded area. However, there are no theoretical investigations which consider contention problems arising in the virtualized environment of a server farm. To model the interaction between application service providers and an infrastructure provider this paper studies the CPP/M/c queue with a compound Poissonian arrival process (CPP) and working vacations.

Such vacation queues have been an intensively studied research topic of queueing theory, cf. [3,5,7,8,9]. However, most of those studies assume a Poissonian arrival model [3,5,7,9] or a model of single arrivals [8]. Regarding the performance evaluation of practical systems, this assumption limits the application of vacation queues.

Recently, queues with working vacations have obtained a big attention, see, e.g., the work of Servi et al. [7] and Liu et al. [5]. It is motivated by the performance evaluation of Wavelength Division Multiplexing (WDM) in optical systems. In this respect, the multi-server queue introduced here is indeed a generalization of the M/M/c system with synchronous vacations [9] regarding two different aspects, namely, the Poissonian batch arrivals and working vacations.

The rest of the paper is organized as follows. In Section 2, we first provide a description of the CPP/M/c model with working vacations (WV), develop then a matrix-analytic solution approach and prove some interesting property of this CPP/M/c WV-queue. Then some illustrative numerical results are presented in Section 3. Finally, we summarize our findings in the conclusions.

2 Analyzing the Maintenance Performance by a Versatile Queueing System

2.1 Description of the Maintenance Model

In a virtualization environment three different roles can be identified (see Figure 1):

- users/applications,
- application service providers and
- owners of the hardware infrastructure.

Applications and related services, e.g. Web servers with Web, information retrieval and business services, are provided by application service providers that require virtual machines from an infrastructure owner to run their virtualized application servers.

In this environment two interrelated categories of Service Level Agreement (SLA) can be defined:

- an SLA between users and application service providers specifying the service requirements, e.g. the response time and availability of a service, etc,
- an SLA between application service providers and an infrastructure owner.

The SLA between users and application service providers are complicated and they also depend on the nature of the hosted applications.

To operate the infrastructure efficiently, it is recognized that advanced management tools are needed. In this respect, system management activities should also include the tasks of managing both virtual servers and physical resources efficiently.

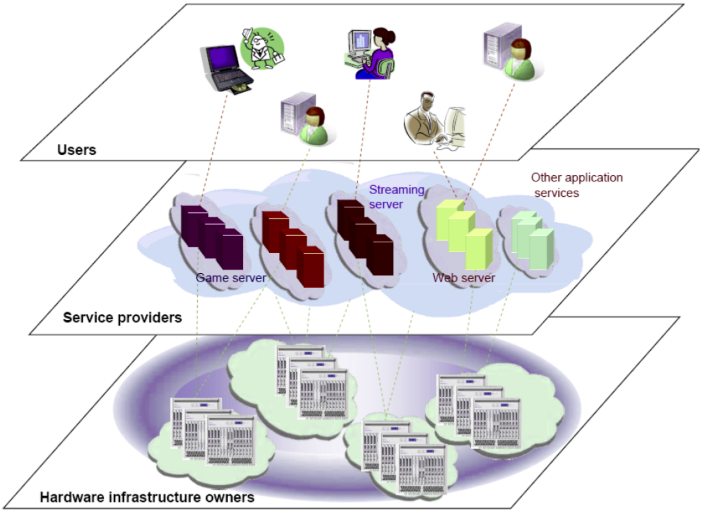


Fig. 1. Utility Computing Environment Based On Virtual Machines

In this paper, we consider the interaction between application service providers and an infrastructure owner. We suppose that there are c virtual servers available in the server pool of the infrastructure owner. To realize a pay-as-you-go approach, application service providers can initiate requests for servers to the provider of the infrastructure and server releases after task completion.

We assume that server requests arrive in batches following the Compound Poisson Process (CPP) (cf. [4]). This means that the inter-arrival times follow a Generalized Exponential (GE) distribution. The arrival process is motivated by the fact that GE is the only distribution of least bias [4] if only the mean and variance of inter-arrival times can be reliably computed by the available measurement data. This situation typically arises in virtualized computing environments exploiting the capabilities of monitoring systems. It has been shown by recent studies [1,2] that the CPP is sufficiently accurate to model Internet traffic in a Web server environment (i.e. the relevant CPP parameters have been estimated by the captured Internet traffic) and that it can be applied to the performance evaluation of wireless telecommunication systems.

To create a reliable computing system with these c servers, the provider of the infrastructure can initiate specific maintenance actions, e.g. software updates, a virtual server live migration etc., when any d servers become idle after a service completion instant. This kind of maintenance activities are modeled in such a way that d servers take a simultaneous vacation. During such a vacation period, the residual $c - d$ servers do not take a vacation even if they are idle. To ensure the mathematical tractability of the model, we assume that the durations of the vacation periods are independent, identical exponentially distributed random variables with parameter θ. The service rate of each server which is not in a vacation state is given by an independent exponential distribution with parameter μ. A server on vacation can serve customers following an independent exponential distribution with rate μ_v. Note that an application service provider who receives the allocation of a server which is on vacation may pay less as a form of compensation.

2.2 Analysis of an Advanced Multi-server Model with Working Vacations

Here, we consider the CPP/M/c multi-server queue with working vacations, infinite waiting room and First In First Out (FIFO) service principle that we have derived as performance model to analyze maintenance tasks in a virtualized server environment.

The arrival process of customer requests is determined by a Compound Poisson Process (CPP) with parameters (λ, ω). It means that the probability distribution function of the inter-arrival times τ is defined by $\mathbb{P}\{\tau = 0\} = \omega \in (0,1)$ and $\mathbb{P}\{0 < \tau < t\} = (1-\omega)(1-e^{-\lambda t})$. Therefore, the arrival process can be seen as a batch Poisson process whose batches of the random size S arriving at some epoch follow a geometric distribution $\mathbb{P}\{S = s\} = (1-\omega)\omega^{s-1}, s \geq 1$, with mean $\mathbb{E}(S) = 1/(1-\omega)$ and variance $\mathrm{Var}(S) = \omega/(1-\omega)^2$.

The requests are served by c servers following a specific working-vacations policy with independent, identical, exponentially distributed service and vacation times with rates μ, μ_v, θ, respectively. Let us suppose that there are no servers on vacation due to maintenance activities. Then a simultaneous vacation period of d servers starts if there are d idle servers after a service completion. At the end of a simultaneous vacation period of these d servers, three alternatives are possible:

– if there are no waiting customers, the d servers stay idle and are ready to serve any arriving new customers;
– if there are $c-d < j < c$, customers in the system, $j-c+d$ returning servers immediately start serving these customers and the other $c-j$ returning servers become idle;
– if there are $j \geq c$ customers in the system, the d returning servers all start serving these customers immediately.

At any time t the state of the system $Y(t) = (I(t), J(t))$ can be completely specified by two integer-valued random variables:

– $I(t) = \begin{cases} 0 & \text{if } d \text{ servers are on vacation at time } t \\ 1 & \text{if there are no servers on vacation at time } t \end{cases}$
– $J(t)$ represents the number of customers in the system at time t including any in service or the waiting room.

The system is now modeled by a continuous-time discrete state Markov process $Y = \{I(t), J(t)\}$ on a rectangular lattice strip $S = \{0,1\} \times \mathbb{N}_0$ due to our Markovian assumptions. We denote its corresponding steady-state probabilities by $\pi = \{\pi_{i,j}\}_{(i,j)\in S}$, where $\pi_{i,j} = \lim_{t\to\infty} P\{I(t) = i, J(t) = j\}$, and let $\mathbf{v}_j = (\pi_{0,j}, \pi_{1,j})$ be the partitioned vector of state probabilities.

The one-step transitions of the Markov chain Y have a specific tridiagonal block structure since the possible transitions are driven by following events:

(a) changing the status of $I(t)$, i.e. from the vacation to non-vacation of servers. Then $A_j(i,k)$ denotes the corresponding transition rate from state (i,j) to state (k,j), $i,k \in \{0,1\}, j \geq 0$. Let

$$A = A_j = \begin{bmatrix} 0 & \theta \\ 0 & 0 \end{bmatrix}, \ \forall j \geq 0; \quad \text{and} \quad A^* = \begin{bmatrix} -\theta & \theta \\ 0 & 0 \end{bmatrix}.$$

(b) the arrivals of customers. Then $B_{i,j,s}$ is the rate of the $s-$step upward transition from state (i,j) to state $(i,j+s)$, $i \in \{0,1\}, j \geq 0$, caused by a batch arrival of size s and

$$B_{i,j,s} = (1-\omega)\omega^{s-1}\lambda, \quad j \geq 0, i \in \{0,1\}, s \geq 1.$$

(c) the departures of customers. $C_j(i,k)$ is the transition rate from state (i,j) to state $(k,j-1)$; $i,k \in \{0,1\}, j \geq 0$. Then we get:

$$C_j = \begin{cases} \begin{bmatrix} j\mu & 0 \\ 0 & j\mu \end{bmatrix}, & 1 \le j \le c-d \\ \begin{bmatrix} (c-d)\mu + \mu_v & 0 \\ (c-d+1)\mu & 0 \end{bmatrix}, & j = c-d+1 \\ \begin{bmatrix} (c-d)\mu + (j-c+d)\mu_v & 0 \\ 0 & j\mu \end{bmatrix}, & c-d+1 < j \le c \\ \begin{bmatrix} (c-d)\mu + d\mu_v & 0 \\ 0 & c\mu \end{bmatrix} = C, & j > c. \end{cases}$$

Note that by a transition from $(1, c-d+1)$ to $(0, c-d)$ after a service completion with rate $(c-d+1)\mu$ we get a simultaneous vacation of d servers.

Let $\mathrm{Diag}(x)$ denote the diagonal matrix defined by a row vector x and $E \in \mathbb{R}^{2\times 2}$ be the identity matrix. We introduce the following notations

$$\begin{aligned} \Lambda &= \mathrm{Diag}[\lambda, \lambda] = \lambda E; \quad \Omega = \mathrm{Diag}[\omega, \omega] = \omega E; \\ B_s &= B_{j,s} = \mathrm{Diag}[(1-\omega)\omega^{s-1}\lambda, (1-\omega)\omega^{s-1}\lambda], \quad j \ge 0, \end{aligned}$$

and obtain

$$\begin{aligned} B_s &= \Omega^{s-1}(E-\Omega)\Lambda = \omega^{s-1}(1-\omega)\lambda E, \quad j \ge 1, \\ \Lambda &= \sum_{s=1}^{\infty} B_s = \lambda E. \end{aligned}$$

Lemma 1. *The necessary and sufficient condition for the existence of the steady-state probabilities of the process $Y = (I, J)$ is determined by*

$$\frac{\lambda}{c\mu} + \omega < 1 \quad \Leftrightarrow \quad \rho = \frac{\lambda}{(1-\omega)\cdot c\mu} < 1 \tag{1}$$

Remark 1. The standard condition (1) states that the traffic intensity ρ must be less than one to achieve the ergodicity of Y. Neither the rate θ of the vacations period nor the number d of simultaneous servers on vacations have an impact on the stability of the system. In other words, the system will not be overloaded due to a maintenance activity. Indeed, this is good news for a system administrator who shall organize the maintenance tasks of idle virtual machines.

Proof. The steady-state balance equation of the M/G/1-like upper Hessenberg system can be written as follows:

$$\sum_{s=1}^{j} \mathbf{v}_{j-s} B_s + \mathbf{v}_j \left[A^* - \Lambda - D^{C_j}\right] + \mathbf{v}_{j+1} C_{j+1} = 0, \qquad \forall j \ge 1. \tag{2}$$

Here D^{C_j} are diagonal matrices whose diagonal elements are the sum of the elements in the rows of C_j. Note that by construction $D^{C_j} = C_j$ holds for all $j \ne c-d+1$.

For $j \geq c+1$ we can write

$$\sum_{s=1}^{j} \mathbf{v}_{j-s} B_s + \mathbf{v}_j \left[A^* - \Lambda - C\right] + \mathbf{v}_{j+1} C = 0. \quad (3)$$

Substituting $B_s = \Omega^{s-1}(E-\Omega)\Lambda$ into this equation (3), we get

$$\sum_{s=1}^{j} \mathbf{v}_{j-s} \Omega^{s-1}(E-\Omega)\Lambda + \mathbf{v}_j \left[A^* - \Lambda - C\right] + \mathbf{v}_{j+1} C = 0 \qquad \forall j \geq c+1, \quad (4)$$

and

$$\sum_{s=1}^{j-1} \mathbf{v}_{j-1-s} \Omega^{s-1}(E-\Omega)\Lambda + \mathbf{v}_{j-1} \left[A^* - \Lambda - C\right] + \mathbf{v}_j C = 0 \qquad \forall j \geq c+2. \quad (5)$$

If we multiply equation (5) by Ω and then subtract the result from equation (4), we obtain the three-term recurrence equations

$$\mathbf{v}_{j-1}[\Lambda - A^*\Omega + C\Omega] + \mathbf{v}_j \left[A^* - \Lambda - C - C\Omega\right] + \mathbf{v}_{j+1} C = 0\ ,\ j \geq c+2,$$

$$\mathbf{v}_{j-1} Q_0 + \mathbf{v}_j Q_1 + \mathbf{v}_{j+1} Q_2 = 0\ ,\ j \geq c+2, \quad (6)$$

where $Q_0 = \Lambda - A^*\Omega + C\Omega,\ Q_1 = A^* - \Lambda - C - C\Omega,\ \ Q_2 = C.$

$Q(x) = Q_0 + Q_1 x + Q_2 x^2$ is defined as the characteristic matrix polynomial associated with the equations (6). It is proved in [6] that the solution of these matrix equations (6) is closely related to the eigenvalues and left-eigenvectors of the polynomial $Q(x)$. If $(x, \boldsymbol{\psi})$ is an eigenvalue-eigenvector pair of $Q(x)$, then it holds

$$\boldsymbol{\psi} Q(x) = 0,\ \ \det[Q(x)] = 0.$$

Consequently, we obtain:

$$\det[Q(x)] = \det \begin{bmatrix} q_{00}(x) & \theta x - \theta\omega \\ 0 & q_{11}(x) \end{bmatrix} = q_{00}(x) q_{11}(x)$$

$$q_{00}(x) = \lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta -$$
$$(\lambda + (c-d)\mu + d\mu_v + ((c-d)\mu + d\mu_v)\omega + \theta)x + ((c-d)\mu + d\mu_v)x^2$$
$$q_{11}(x) = \lambda + c\mu\omega - (\lambda + c\mu + c\mu\omega)x + c\mu x^2 = (1-x)(\lambda + c\mu\omega - c\mu x)$$

Therefore, $Q(x)$ has four eigenvalues

$$x_1 = \frac{1}{2G}\{H + G - \sqrt{(H+G)^2 - 4G(\lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta)}\}$$
$$x_2 = \frac{1}{2G}\{(H + G + \sqrt{(H+G)^2 - 4G(\lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta)}\}$$
$$x_3 = \lambda/(c\mu) + \omega,\ \ x_4 = 1,$$

where

$$\begin{aligned} G &= (c-d)\mu + d\mu_v \\ H &= \lambda + ((c-d)\mu + d\mu_v)\omega + \theta \end{aligned}$$

holds.

Note that $\psi_1 = (1, (\theta\omega - \theta x_1)/q_{11}(x_1))$ is the left-hand-side (LHS) eigenvector of $Q(x)$ for the eigenvalue x_1, and $\psi_3 = \psi_4 = (0, 1)$ are the LHS eigenvectors of $Q(x)$ for the eigenvalues x_3 and x_4, respectively.

Since $\omega < 1$ holds, we have

$$\begin{aligned} (\lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta) &< H, \\ 4G(\lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta) &< 4GH, \\ (H+G)^2 - 4G(\lambda + ((c-d)\mu + d\mu_v)\omega + \omega\theta) &> (H+G)^2 - 4GH, \\ 0 < x_1 &< \frac{1}{2G}(H + G - |H-G|) \leq 1, \\ x_2 &> \frac{1}{2G}(H + G + |H-G|) \geq 1. \end{aligned}$$

Applying results from [6], it is a necessary and sufficient condition for the ergodicity of the Markov chain Y that the number of eigenvalues of $Q(x)$ inside the unit disk is given by 2. Therefore, $x_3 < 1$ is required which yields condition (1). □

The steady-state balance equations of $J(t) \in \{0, \ldots, c+1\}$ can be written in the following form:

$$\mathbf{v}_0 \left[A^* - \Lambda\right] + \mathbf{v}_1 C_1 = 0$$

$$\sum_{s=1}^{j} \mathbf{v}_{j-s} B_s + \mathbf{v}_j \left[A^* - \Lambda - D^{C_j}\right] + \mathbf{v}_{j+1} C_{j+1} = 0\,, \quad 1 \leq j \leq c+1,$$

For $j \geq c+1$ the steady-state probabilities can be expressed as follows (cf. [6]):

$$\begin{aligned} \mathbf{v}_j &= \alpha \boldsymbol{\psi}_1 x_1^j + \beta \boldsymbol{\psi}_3 x_3^j \\ \pi_{0,j} &= \alpha x_1^j \\ \pi_{1,j} &= \alpha \frac{\theta\omega - \theta x_1}{q_{11}(x_1)} x_1^j + \beta x_3^j \end{aligned} \tag{7}$$

where α and β are coefficients that have to be determined by the boundary conditions.

Furthermore, we have to satisfy the normalization equation:

$$\sum_{j=0}^{\infty} \sum_{i=0}^{1} \pi_{i,j} = 1. \tag{8}$$

Consequently, we have to determine the vectors $\mathbf{v}_j$, $0 \leq j \leq c$, α and β. The total number of these unknowns is given by $2(c+1)+2 = 2(c+2)$. To determine these unknowns, we have the steady-state balance equations of the levels $j = 0, \ldots, c+1$ and the normalization equation. Thus, $2(c+2)+1$ is the number of boundary equations, among those only $2(c+2)$ equations are independent.

It can be observed from the steady-state balance equations of $J(t) \in \{0, \ldots, c\}$ that $\mathbf{v}_j$, $1 \leq j \leq c$ and $j \neq c-d+1$, can be expressed as a function of $\mathbf{v}_0$, i.e $\pi_{0,0}$ and $\pi_{1,0}$, and $\mathbf{v}_{c-d+1}$. Therefore, we have only six unknowns $(\pi_{0,0}, \pi_{1,0}, \pi_{0,c-d+1}, \pi_{1,c-d+1}, \alpha, \beta)$, which can be solved efficiently using the steady-state balance equations of the states $J(t) = c$, $J(t) = c-d$, $J(t) = c+1$ and the normalization equation.

2.3 Conditional Stochastic Decomposition

In the following, we prove a conditional stochastic decomposition property for the CPP/M/c queue with working vacations.

Lemma 2. *If the ergodicity condition for the CPP/M/c queue with working vacations holds, then the conditional steady-state queue length $J_b = lim_{t\rightarrow\infty}\{J(t) - c - 1 | J(t) > c, I(t) = 1\}$ provided that the server system is not on a working vacation can be decomposed into the sum of two independent random variables*

$$J_b = J_0 + J_c.$$

Here J_0 is the conditional steady-state queue length of the CPP/M/c queue without vacations and J_c is the additional steady-state queue length due to vacations.

Proof. The probability that the server is busy and the number of jobs is larger than c is determined by:

$$\begin{aligned} P_b = P\{J(t) > c, I(t) = 1\} = \sum_{j=c+1}^{\infty} \pi_{1,j} &= \sum_{j=c+1}^{\infty} \left(\alpha \frac{\theta\omega - \theta x_1}{q_{11}(x_1)} x_1^j + \beta x_3^j \right) \\ &= \alpha \frac{\theta\omega - \theta x_1}{q_{11}(x_1)} \frac{x_1^{c+1}}{1-x_1} + \beta \frac{x_3^{c+1}}{1-x_3} \end{aligned}$$

The probability generating function of J_b can be expressed as follows:

$$\begin{aligned} G_{J_b}(z) &= \sum_{j=0}^{\infty} P\{J_b = j\} z^j = \sum_{j=0}^{\infty} \frac{\pi_{1,j+c+1}}{P_b} z^j \\ &= \frac{1}{P_b} \sum_{j=0}^{\infty} \left(\alpha \frac{\theta\omega - \theta x_1}{q_{11}(x_1)} x_1^{j+c+1} + b x_3^{j+c+1} \right) z^j \\ &= \frac{1}{P_b} \left(\alpha \frac{\theta\omega - \theta x_1}{q_{11}(x_1)} x_1^{c+1}/(1 - x_1 z) + \beta x_3^{c+1}/(1 - x_3 z) \right) \end{aligned}$$

The steady-state probabilities of the CPP/M/c queue without vacations can be obtained by setting $\theta = 0$, $d = 0$ and $\mu_v = \mu$. The probability that the number of customers in the CPP/M/c queue without vacations is given by $\pi_j = \beta^* x_3^j$ for $j \geq c+1$, where β^* is an appropriate coefficient. Therefore, $G_{J_0}(z) = \beta^* \frac{x_3^{c+1}}{1-x_3 z}$ follows for the probability generating function of J_0. These relations yield the stated result. □

3 Illustrative Numerical Results

In this section we present some numerical results to illustrate the impact of the model parameters on the formulation of an effective maintenance policy, i.e. how many servers should be simultaneously on vacations. For demonstration purposes, we investigate the average number of customer requests waiting for free servers

$$\mathbb{E}(L_Q) = \sum_{j=c+1}^{\infty} (j-c) \cdot (\pi_{0,j} + \pi_{1,j}) = \sum_{j=c+1}^{\infty} (j-c) \cdot \left(\alpha[1 + \frac{\theta\omega - \theta x_1}{q_{11}(x_1)}] x_1^j + \beta x_3^j \right)$$
$$= \frac{\alpha[1 + \frac{\theta\omega - \theta x_1}{q_{11}(x_1)}] x_1^{(c+1)}}{(1-x_1)^2} + \frac{\beta x_3^{(c+1)}}{(1-x_3)^2}$$

as major performance metrics and select some illustrative parameter set. Other characteristics like the mean number of requests in the system

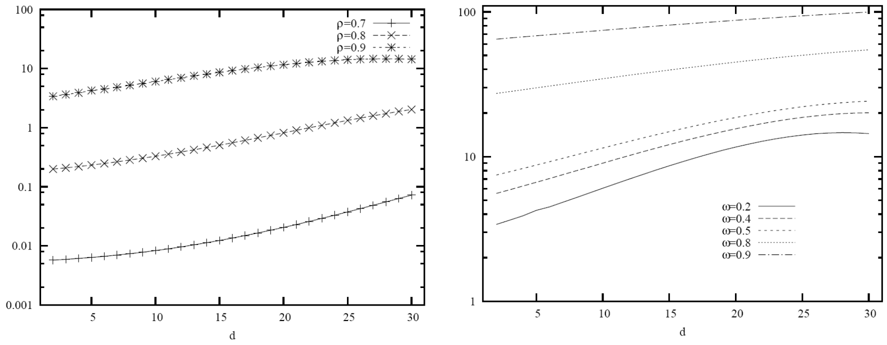
$$\mathbb{E}(L) = \sum_{j=1}^{\infty} j \cdot (\pi_{0,j} + \pi_{1,j}) \,,$$

or the mean number of active servers $\mathbb{E}(N_V) = \sum_{j=1}^{\infty} \min(j,c) \cdot (\pi_{0,j} + \pi_{1,j})$ and the throughput $\eta = \sum_{j=1}^{\infty} \mathbf{v}_j \cdot C_j \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \sum_{j=1}^{\infty} \sum_{i=0}^{1} \pi_{i,j} \cdot (C_j(i,0) + C_j(i,1))$ can be computed in a similar manner.

In Figure 2(a) we plot the average number of waiting requests $\mathbb{E}(L_Q)$ versus d for the following parameter set of a high load regime: $c = 100$ servers, $\omega = 0.2$, $\theta = 1.0$, $\mu = 5.0$, $\mu_v = 2.5$, $\rho = \lambda/[(1-\omega)c\mu] \in [0.7, 0.9]$. It generates batch arrivals of mean size $\mathbb{E}(S) = 1.25$ and variance $\mathrm{Var}(S) = 0.3125$ for a high traffic intensity $0.7 \leq \rho \leq 0.9$ and assumes that the average service time $1/\mu$ of requests needs only 20 % of the mean maintenance time $1/\theta$ while during these maintenance periods the latter service time is extended by 100 % compared to the normal operation mode.

Considering the average number $\mathbb{E}(L_Q)$ of requests waiting in the system, it is observed that increasing the load ρ from 0.7 to 0.8 or from 0.8 to 0.9 generates an increment of one order of magnitude. To show the impact of the size S of arriving batches and the influence of $\omega = 1 - 1/\mathbb{E}(S)$ and $\mathbb{E}(S) \in \{1.25, 1.67, 2, 5, 10\}$, respectively, we use the set of the same parameters but fix the load at $\rho = 0.8$. Figure 2(b) illustrates the average number of waiting requests $\mathbb{E}(L_Q)$ versus d and ω. In Figure 3 $\mathbb{E}(L_Q)$ is plotted against d and μ_v for the load $\rho = 0.9$ and a mean batch size of $\mathbb{E}(S) = 1/(1-\omega) = 1.25$.

It is observed that batch arrivals have the strongest impact on the average number of waiting customers. The impact of the offered load ρ and the service rate μ_v during maintenance can be handled by choosing an appropriate number d of servers under maintenance.



(a) (b)

Fig. 2. Average number $\mathbb{E}(L_Q)$ of waiting requests versus d for different traffic load ρ (left) and different control parameter $\omega = 1 - 1/\mathbb{E}(S)$ of the mean batch size $\mathbb{E}(S)$ (right)

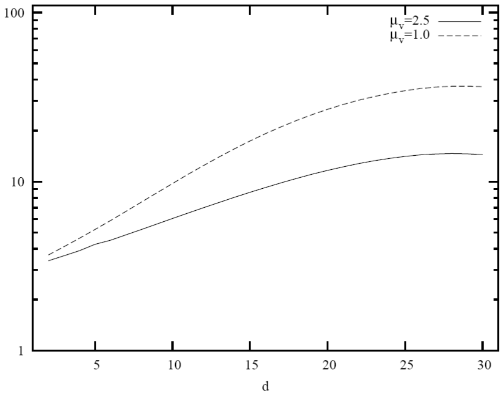


Fig. 3. Average number $\mathbb{E}(L_Q)$ of waiting requests versus d and μ_v

4 Conclusions

To model the queueing and congestion phenomena arising from maintenance tasks of a virtualized server environment, we have presented in this study a CPP/M/c multi-server system with Poissonian batch arrivals and working vacations.

In the proposed queueing system the inter-arrival times of jobs requesting service by a virtualized server follow a Generalized Exponential distribution. To model the maintenance activities, we have assumed that a certain number of servers goes simultaneously to a maintenance state for a random period when they have completed the service of jobs and find no further requests in the waiting line.

Analyzing the arising Markovian model by matrix-analytic methods, we have derived a new expression for the steady-state probabilities and proved a conditional stochastic decomposition property. The validation of the approach in a testbed and the estimation of the parameters by gathered data is a topic of future research.

In conclusion, we believe that the proposed Markovian multi-server system with working vacations can serve as a useful tool to define efficient maintenance policies in the virtualized environment of current server farms.

References

1. Do, T.V., Chakka, R., Harrison, P.G.: An integrated analytical model for computation and comparison of the throughputs of the UMTS/HSDPA user equipment categories. In: MSWiM 2007 Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, pp. 45–51. ACM, New York (2007)
2. Do, T.V., Krieger, U.R., Chakka, R.: Performance modeling of an Apache Web server with a dynamic pool of service processes. Telecommunication Systems 39(2), 117–129 (2008)
3. Doshi, B.T.: Queueing systems with vacations – a survey. Queueing Syst. Theory Appl. 1(1), 29–66 (1986)
4. Kouvatsos, D.: Entropy maximisation and queueing network models. Annals of Operations Research 48, 63–126 (1994)
5. Liu, W., Xu, X., Tian, N.: Stochastic decompositions in the M/M/1 queue with working vacations. Oper. Res. Lett. 35, 595–600 (2007)
6. Mitrani, I., Chakka, R.: Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method. Performance Evaluation 23, 241–260 (1995)
7. Servi, L.D., Finn, S.G.: M/M/1 queues with working vacations (M/M/1/WV). Perform. Eval. 50(1-4), 41–52 (2002)
8. Tian, N., Zhang, Z.G.: Stationary Distributions of GI/M/c Queue with PH Type Vacations. Queueing Syst. Theory Appl. 44(2), 183–202 (2003)
9. Zhang, Z.G., Tian, N.: Analysis on queueing systems with synchronous vacations of partial servers. Perform. Eval. 52(4), 269–282 (2003)