Analysis of performance and energy consumption in the cloud

M. Kandi[1] F. Aït-Salaht[2,3], H. Castel-Taleb[1], and E. Hyon[3]

[1] SAMOVAR, CNRS, Telecom SudParis, Université Paris-Saclay
9, rue Charles Fourier, 91011 Evry Cedex, France
medmehdikandi@gmail.com, hind.castel@telecom-sudparis.eu
[2] Crest-Ensai, Rennes, France
farah.ait-salaht@ensai.fr
[3] Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 Paris UMR 7606,
4 place Jussieu 75005 Paris France, Université Paris Nanterre, France
emmanuel.hyon@lip6.fr

Abstract. We analyze here a cloud system represented by hysteresis multi server queueing system. It is characterized by forward and backward thresholds for activation and deactivation of block of servers representing a set of VMs (Virtual Machines). The system is represented by a complex Markov Chain which is difficult to analyse when the size of the system is huge. We propose both analytical and numerical mathematical methods for deriving the steady-state probability distribution. We compute then performance and energy consumption measures and we define an overall cost taking into account both aspects. We compare the proposed methods with respect to the computation time and we analyse the impact of some parameters on the behaviour of the system.

1 Introduction

Improving the energy consumption of a cloud while guaranteeing a given quality of service is a problem encountered today by cloud providers. One way to achieve this is to adapt the capacities to demand which is made easier today with the virtualization of the servers. Hence, it is possible to modulate, in a transparent manner, the number of active Virtual Machines (VMs) over time. However, finding the policy that tailors resources to demand is a crucial point that requires accurate assessment of both the energy expended and the performance of the system. Multi server queuing models [2, 3] or server farms models [6, 15] have been proposed to represent dynamicity of a data center as well as to compute performance metrics. Multi-server threshold based-queueing system with hysteresis policy [4, 9, 13], in which activations and deactivations are governed by sequences of forward and reverse different thresholds, have been proposed, on the other hand, to efficiently manage the number of active VMs. For systems driven by hysteresis policies, the assessment of both performance and energy consumption requires the computation of the expected measures, but since cloud systems are often defined on very large state spaces such a computation is difficult. When the system is represented by a complex Markov chain, we face up a computational complexity problem which makes exact analysis very cumbersome or even impossible.

Under some assumptions, evaluation of hysteresis multiserver systems has been already studied in the literature and different resolution methods have been presented to compute efficiently the performance measures of the system. Among the most significant works, we can mention the work of Lui and Golubchik [13] which is widely used in the literature. It solves the model using the concept of Stochastic Complement Analysis (SCA). It is based on partitioning the state space in disjoint sets in order to aggregate the Markov chain. In [12], Le Ny et al. propose an other way to compute the steady-state probabilities of a heterogeneous multi-server threshold queue with hysteresis by using a closed-form solution. Otherwise, in [1] an aggregated bounding approach is proposed to derive accurate bounds on performance measures. However, in these papers, it had been only considered the case where one VM is activated (resp. deactivated) according to the demand and the threshold sequences. On the other hand, Mitrani [14, 15] defines server farm models in which several servers are activated at the same time. They are called activations by block. Such approaches allow to model more general practical models.

We propose in this paper to extend the current state of art and to couple the advantages of the activation by block and the advantages of hysteresis policy by considering a multi-server system with hysteresis in which activations/deactivations are made by block. This, up to our knowledge, has never been considered and studied previously in the literature.

This allows us to consider both performance and energy consumption in order to propose a trade-off between them. For the multi-server system with hysteresis and block activation, we establish and present three resolution methods. First method consists to adapt and extend the SCA aggregation method of [13]. The second investigated method is a numerical approach based on Level Dependent Quasi Birth and Death (LDQBD) method. At last, an analytical approach based on the balance equations method of [12] is presented in details. We adapt [12] and get closed form formulas for the steady-state probability distribution. Furthermore, by relaxing the former assumptions on the threshold sequences imposed by [13] or [12], we have generalized the set of threshold values. We then perform numerical results for Markov chains with large state space, as in cloud systems, and establish an overall cost taking into account both performance and energy consumption. Moreover, as we consider in this model more general assumptions for the thresholds, we can see in details the impact of their values on performance and energy consumption.

The paper is organized as follows: next (in section 2), we describe the cloud system and present the considered queueing model. In section 3, we detail the different methods to solve the model and compute the steady state probability vector. While part 4 presents the formulation used to express the expected costs in terms of performance and energetic consumption for the model, section 5 presents numerical results of performance and energy consumption measures.

Finally, achieved results are discussed in the conclusion and comments about further research issues are given.

2 Cloud system description

We analyse a cloud system composed by a set of Virtual Machines (VMs). We model it using a multi-server queue, with C homogeneous servers representing the VMs. The service time of each VM is exponential with mean rate μ. In order to represent the dynamicity of resource provisioning, the VMs can be activated and deactivated over time. We assume that the job requests arrive at the system following a Poisson process with rate λ, and are enqueued in a finite queue. An arriving request can be rejected if it finds the system, which have a whole capacity of B, full. The servers management is governed by threshold vectors which control the operation of activating and deactivating the VMs. These thresholds depend on the number of customers waiting in the system.

We suppose the case where several VMs can be simultaneously activated or deactivated what is called activated or deactivated by block. We define K functioning levels, where each level corresponds to a given number of active servers. The number of active servers at level k is fixed and denoted by S_k, where $S_1 \leq S_2 \leq ... \leq S_K = C$. We suppose that $S_1 \geq 1$, so we have at least one active server by assumption.

The transition from functioning level k to level $k+1$ allows to allocate (turn on) one or more additional servers, going from S_k to S_{k+1} active servers, while the transition from level k to level $k-1$ allows to remove (turn off) one or more active servers, going from S_k to S_{k-1} active servers. Depending on the system occupancy, we transit from the level k to level $k+1$ when the occupancy in the system exceeds a threshold F_k, and from level k to level $k-1$ when the occupancy in the system falls below a threshold R_{k-1}. The model is then characterized by activation thresholds $F = (F_1, F_2, ..., F_{K-1})$ (called also forward thresholds), and deactivation thresholds $R = (R_1, R_2, ..., R_{K-1})$ (called also reverse thresholds). These thresholds are fixed and can not be modified during the system works. We furthermore assume that $F_1 < F_2 < ... < F_{K-1}$, that $R_1 < R_2 < ... < R_{K-1}$ and that $R_k < F_k, \forall k, 1 \leq k \leq K-1$. We suppose that server deactivations occur at the end of the service, and when multiple servers are deactivated at the same time, all the customers who have not completed their service return to the queue.

The underlying model is described by the Continuous-Time Markov Chain (CTMC), denoted $\{X(t)\}_{t \geq 0}$. A state is represented by a couple $(m,\, k)$ such that m is the number of customers in the system and k is the functioning level. The state space is denoted by A and is given by :

$$\begin{aligned} A = \{(m,\, k) \,|\, & 0 \leq m \leq F_1, \text{ if } k = 1\,, \\ & R_{k-1} + 1 \leq m \leq F_k, \text{ if } 1 < k < K\,, \\ & R_{K-1} + 1 \leq m \leq B, \text{ if } k = K\}\,. \end{aligned}$$

The transitions between states then follows:

$$\begin{aligned}
(m,k) &\rightarrow (\min\{B, m+1\}, k), \text{ with rate } \lambda\,, \text{ if } m < F_k\,;\\
&\rightarrow (\min\{B, m+1\}, \min\{K,\ k+1\}), \text{ with rate } \lambda, \text{ if } m = F_k\,;\\
&\rightarrow (\max\{0, m-1\}, k), \text{ with rate } \mu\cdot\min\{S_k, m\}, \text{ if } m > R_{k-1}+1;\\
&\rightarrow (\max\{0, m-1\}, \max\{1, k-1\}) \text{ with rate } \mu\cdot\min\{S_k, m\}, \text{ if } m = R_{k-1}+1\,.
\end{aligned}$$

An example of the transitions is given Figure 1.

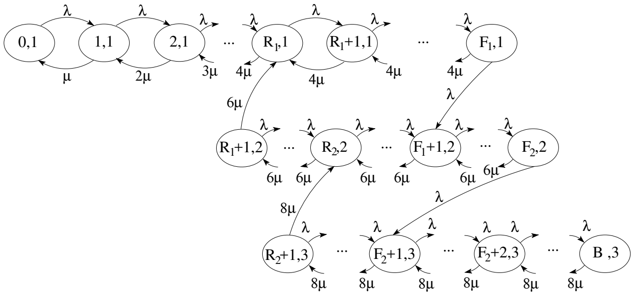


Fig. 1. Transition structure for $K = 3$, $S_1 = 4$, $S_2 = 6$, $S_3 = 8$, $R_1 \geq 5$ and $R_2 \geq 7$.

From a practical perspective, several other models fit with this block representation. For example, it can represent heterogeneous nodes of a cluster (possibly virtual), each node having a different number of cores. These nodes can be idle or activated. In this case, a node is represented by a level and S_k is the number of cores of the node. It can also represent a single physical component composed by many cores that can be activated or deactivated. On each core (represented by a level) a given number of S_k VMs are placed that share the CPU. These models follow the same markovian representation than the model studied here but their costs are different.

3 Resolution approaches

We expose hereafter three techniques to solve the CTMC and compute the steady-state probability vector. These resolution methods are either numerical or analytical or both analytical and numerical. They have been developed for the model and their correctness is shown. Some comparisons are presented Section 5.

3.1 Stochastic Complement Analysis (SCA)

To solve the $\{X(t)\}_{t\geq 0}$ Markov chain, the first approach, proposed by Lui et al. [13], consists to aggregate the underlying Markov chain and uses a numerical method to compute the steady-state distribution. The different restrictions of [13] (i.e. $R_k < F_{k-1}$, $\forall k$ and activation deactivation of a single server) can be relaxed without substantially modifying the framework. Our approach considers block activations and deactivations as well as different orders of the thresholds: $R_k < F_{k-1}$ and $R_k \geq F_{k-1}$, for all k. It is presented below and some details can be found in [8].

First, we aggregate the state space of the underlying Markov chain by partitioning the set A into disjoint subsets. These subsets depend here on the functioning levels. Hence, the state space A is partitioned into K distinct sets denoted A_k, where, for any k in $1, \ldots, K$, we have $A_k = \{(i,\, j) \mid (i,\, j) \in A, j = k\}$. The set A_k contains the states belonging the level k.

From each subset, we define a corresponding Markov chain. Let $\{X_k(t)\}_{t \geq 0}$ be the Markov chain defined on state space A_k. These derived Markov chains are defined on reduced state spaces which makes their analysis less complex. The resolution of each of the derived Markov chain defines a conditional steady state probabilities. For the whole chain $\{X(t)\}$, by applying the state aggregation technique, each subset A_k is now represented by a single state, and an aggregated process is defined. A resolution of this aggregated process is performed, i.e., the probabilities of the system being in any given set are computed. At last, a disaggregation technique is applied to compute the individual steady state probabilities for the original Markov process. The method correctness is based on the following theorem stated by Lui et al. in [13].

Theorem 1. *Given an irreducible Markov process with state space A, let us partition this state space into two disjoint sets A_1 and A_2. Then, the transition rate matrix (denoted by Q) is given as follows:*

$$Q = \begin{pmatrix} Q_{A_1A_1} & Q_{A_1A_2} \\ Q_{A_2A_1} & Q_{A_2A_2} \end{pmatrix},$$

where $Q_{i,j}$ is the transition rate sub-matrix corresponding to transitions from partition i to partition j.

We point out that the computation of the steady state probabilities of the derived Markov chains $\{X_k(t)\}$ is performed using a numerical resolution method.

3.2 Level Dependant Quasi Birth and Death Process

The particular form of the generator of $\{X(t)\}_{t \geq 0}$ suggests us to use the Quasi Birth and Death (QBD) processes in order to benefit from the numerous numerical methods to solve them [16]. For short, a QBD process is a stochastic process in which the state space is two dimensional and can be decomposed in disjoint sets such that transition may only occur inside a set or occur towards only two other sets. This results in a generator with a tridiagonal form (as the birth and death process) in which the terms on the diagonals are matrices. When the matrices are identical for each level, it is said *level independent* but when the matrices are different the QBD is said *level dependant* (LDQBD).

Let us define $Q_{k,k'}(i,j)$ that denotes the i-th line and j-th column element of matrix $Q_{k,k'}$. We have:

Proposition 1. *The Markov Chain $\{X(t)\}_{t\geq 0}$ is a Level Dependant QBD with K levels, corresponding to the functioning levels. Its generator Q is decomposed in:*

$$Q=\begin{pmatrix} Q_{1,1} & Q_{12,} & & & & \\ Q_{2,1} & Q_{2,2} & Q_{2,3} & & & \\ & Q_{3,2} & Q_{3,3} & Q_{3,4} & & \\ & & \ddots & \ddots & \ddots & \\ & & & Q_{K-1,K-2} & Q_{K-1,K-1} & Q_{K-1,K} \\ & & & & Q_{K,K-1} & Q_{K,K} \end{pmatrix}.$$

For all k, the inner matrices $Q_{k,k-1}$, $Q_{k,k}$ and $Q_{k,k+1}$ are respectively of dimension $d_k \times d_{k-1}$, $d_k \times d_k$ and $d_k \times d_{k+1}$, letting $d_k = F_k - R_{k-1}$, $R_0 = -1$ and $F_K = B$.

For $k = 1$ we have:

$$Q_{1,1}(i,j) = \begin{cases} \lambda & \text{if } j = i+1 \\ \mu \min\{S_1, i\} & \text{if } j = i-1 \\ -\lambda & \text{if } i = 1 \text{ and } j = 1 \\ -(\lambda + \mu \min\{S_1, i\}) & \text{if } i = j \text{ and } i \neq 1 \\ 0 & \text{otherwise} \end{cases},$$

and

$$Q_{1,2}(i,j) = \begin{cases} \lambda & \text{if } i = d_1 \text{ and } j = F_1 - R_1 + 1 \\ 0 & \text{otherwise} \end{cases}.$$

For $k \in \{2, \ldots, K-1\}$, we get:

$$Q_{k,k-1}(i,j) \quad = \quad \begin{cases} \mu \min\{S_k, R_{k-1}+1\} & \text{if } i=1 \text{ and } j=R_{k-1}-R_{k-2} \\ 0 & \text{otherwise} \end{cases},$$

also

$$Q_{k,k}(i,j) = \begin{cases} \lambda & \text{if } j = i+1 \\ \mu \min\{S_k, R_{k-1}+i\} & \text{if } j = i-1 \\ -(\lambda + \mu \min\{S_k, R_{k-1}+i\}) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

and

$$Q_{k,k+1}(i,j) = \begin{cases} \lambda & \text{if } i = d_k \text{ and } j = F_k - R_k + 1 \\ 0 & \text{otherwise} \end{cases}.$$

Finally for $k = K$, it follows

$$Q_{K,K-1}(i,j) \quad = \quad \begin{cases} \mu \min\{S_K, R_{K-1}+1\} & \text{if } i=1 \text{ and } j=R_{K-1}-R_{K-2} \\ 0 & \text{otherwise} \end{cases},$$

and

$$Q_{K,K}(i,j) \quad = \quad \begin{cases} \lambda & \textit{if } j=i+1 \\ \mu \min\{S_K, R_{K-1}+j\} & \textit{if } j=i-1 \\ -(\lambda+\mu\min\{S_K,R_{K-1}+j\}) & \textit{if } i=j \textit{ and } j\neq d_K \\ -\mu\min\{S_K,R_{K-1}+j\} & \textit{if } i=j=d_K \\ 0 & \textit{otherwise} \end{cases}.$$

The proof can be found in [8].

Numerically solving QBD is a hard computational task requiring to solve matrix equations and is often based on matrix geometric methods [11, 16] or kernel methods [7]. This is even more the case for LDQBD. Here, among the numerical existing methods to solve them, this one proposed in [5] is used since it is shown that this method is efficient and numerically stable.

3.3 Closed form solution using balance equations

We follow the approach of [12] and give a closed form for the steady state probability using balance equations and cuts on the state space. The relevance of our work is that we can take more general cases than [12] for the thresholds. We assume not only the case $R_k \leq F_{k-1}$, but also the case $R_k > F_{k-1}$ for each level $2 \leq k \leq K$. In this method, the probabilities are computed level by level, from level 1 to level K. For states of level 1, the steady-state probabilities are expressed in terms of $\pi(0,1)$. For a level $k \in \{2 \ldots K\}$, the process has two steps. First, the steady-state probability of the first state of the level: $\pi(R_{k-1}+1,k)$ is expressed in terms of the last state of the precedent level $\pi(F_{k-1},k-1)$ which has been already computed and which can be expressed in terms of $\pi(0,1)$. After that, the other probabilities of the level k are computed in terms of $\pi(R_{k-1}+1,k)$. Henceforth, it results that all the probabilities are computed in terms of $\pi(0,1)$. At the end, from the normalizing condition, $\pi(0,1)$ can be derived. From now on, for any $k \in \{1 \ldots K\}$, we define $\mu_k = \mu S_k$, $\rho = \frac{\lambda}{\mu}$ and $\rho_k = \frac{\lambda}{\mu_k}$. Next, we give the formulas for the level 1 probabilities.

Level 1 The following lemma gives the steady-state probabilities for level 1.

Lemma 1 (Level 1 probabilities). *In level one, the service rate depends on the number of customers in the system. So, for a state $(m,1)$, if $1 \leq m < S_1$, then the service rate is $m\mu$ and if $m \geq S_1$ it is $S_1\mu$. We can deduce $\pi(m,1)$ by :*

$$\pi(m,1)=\begin{cases} \dfrac{\rho^m}{m!}\pi(0,1) & \textit{if } 0\leq m\leq S_1, \quad (1) \\ \rho_1^{m-S_1}\dfrac{\rho^{S_1}}{S_1!}\pi(0,1) & \textit{if } S_1<m\leq R_1, (2) \\ \dfrac{\rho^{S_1}}{S_1!}\left(\rho_1^{m-S_1}-\dfrac{\rho_1^{F_1-S_1+1}(1-\rho_1^{m-R_1})}{1-\rho_1^{F_1-R_1+1}}\right)\pi(0,1) & \textit{if } R_1+1\leq m\leq F_1 (3) \end{cases}$$

The proof uses special cuts on state space from which one derives local balance equations. It is given in [8].

Level *k* Let us consider now k such that $2 \leq k \leq K-1$. We assume that $R_{k-1}+1 \geq S_k$, and thus the service rate for each level is $\min(R_{k-1}+1, S_k) = S_k\mu$. In order to express the relationship between level $k-1$ and level k, we should consider the cut of the state space between states of level $k-1$ and states of level k. This gives us the following evolution equation: $\pi(F_{k-1}, k-1)\lambda = \pi(R_{k-1}+1, k)\mu_k$, which is equivalent to:

$$\pi(R_{k-1}+1, k) = \rho_k \pi(F_{k-1}, k-1)\,. \tag{4}$$

All probabilities of level k can be expressed with respect to $\pi(R_{k-1}+1, k)$. However, these probabilities depend also of the level $k+1$ by the threshold value R_k. Therefore two cases should be considered: either $R_k \leq F_{k-1}$ or $R_k > F_{k-1}$.

We present now the case where $R_k > F_{k-1}$.

Lemma 2. *When $R_k > F_{k-1}$, for any $k \in \{2 \ldots K-1\}$, we have:*

$$\pi(m,k) = \frac{1-\rho^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \text{ if } R_{k-1}+2 \leq m \leq F_{k-1}+1, \tag{5}$$

$$\pi(m,k) = \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \text{ if } F_{k-1}+2 \leq m \leq R_k, \tag{6}$$

$$\begin{aligned}\pi(m,k) = {} & \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \\ & - \frac{\rho_k}{\rho_{k+1}}\frac{1-\rho_k^{m-R_k}}{1-\rho_k}\pi(R_k+1, k+1) \text{ if } R_k+1 \leq m \leq F_k.\end{aligned} \tag{7}$$

with

$$\pi(R_{k+1}, k+1) = \rho_{k+1}\frac{\rho_k^{F_k-F_{k-1}-1} - \rho_k^{F_k-R_{k-1}}}{1-\rho_k^{F_k-R_k+1}}\pi(R_{k-1}+1, k)\,. \tag{8}$$

The proof of lemma 2 is in [8].

We deduce from Eq. (8), that $\pi(R_k+1, k+1)$ is also expressed in terms of $\pi(R_{k-1}+1, k)$. Thus all the probabilities in Lemma 2 can be expressed in terms of the steady-state $\pi(R_{k-1}+1, k)$ which is the first state of the level. Since, furthermore, $\pi(R_{k-1}+1, k)$ is computed from $\pi(F_{k-1}, k-1)$, then it can be expressed in terms of $\pi(0, 1)$. So from the normalizing condition we derive all the probabilities.

Since the case $R_k \leq F_{k-1}$, has been considered in [12], it follows :

Lemma 3 ([12]). *When $R_k \leq F_{k-1}$, for any $k \in \{2 \ldots K-1\}$, we have*

$$\pi(m,k) = \frac{1-\rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1,k) \quad \textit{if } R_{k-1}+2 \leq m \leq R_k \tag{9}$$

$$\begin{aligned}\pi(m,k) = & \frac{1-\rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1,k) \\ & - \frac{\rho_k}{\rho_{k+1}}\frac{1-\rho_k^{m-R_k}}{1-\rho_k}\pi(R_k+1,k+1) \ \textit{if } R_k+1 \leq m \leq F_{k-1}+1,\end{aligned} \tag{10}$$

$$\begin{aligned}\pi(m,k) = & \rho_k^{m-F_{k-1}-1}\frac{1-\rho_k^{F_{k-1}-R_{k-1}+1}}{1-\rho_k}\pi(R_{k-1}+1,k) \\ & - \frac{\rho_k}{\rho_{k+1}}\frac{1-\rho^{m-R_k}}{1-\rho_k}\pi(R_k+1,k+1 \ \textit{if } F_{k-1}+2 \leq m \leq F_k\,.\end{aligned} \tag{11}$$

Proofs are given in [12].

Level $\boldsymbol{K}$ Let us consider newt the level $k = K$.

Lemma 4 ([12]). *The steady-state probabilities for the level K: $\pi(m,K)$ are equal to:*

$$\pi(m,K)=\left(\frac{1-\rho_k^{m-R_{K-1}}}{1-\rho_K}\right)\pi(R_{K-1}+1,K) \ \textit{if } R_{K-1}+2 \leq m \leq F_{K-1}+1,$$

$$\pi(m,K)=\left(\frac{1-\rho_k^{F_{K-1}+1-R_{K-1}}}{1-\rho_K}\right)\rho_K^{(m-F_{K-1}-1)}\pi(R_{K-1}+1,K) \ \textit{if } F_{K-1}+2 \leq m \leq B.$$

It is proved in [12].

4 Performance measures and energy cost parameters

We propose now to calculate the expected cost in terms of performance and energy consumption for the model presented in this paper. Once the steady-state vector is calculated, we get various performance and energy consumption measures. Indeed the cost is expressed as an expected Markov reward function $\mathcal{R}$, where $\mathcal{R} = \sum_{m,k} \pi(m,\, k)\, r(m,\, k)$ and $r(m,\, k)$ be the reward of state $(m,\, k)$. Metrics of interest are described hereafter.

First, we give the performance measures. These one are related to the Service Level Agreement (SLA) which defines several QoS (Quality of Service) constraints that the provider should guarantee. Losses, queue lengths and processing speed are the main parameters that are taken into account.

The *mean number of customers* in the system is denoted by $\overline{N_C}$ and is equal to: $\overline{N_C} = \sum_{(m,k)\in A} \pi(m,k) \cdot m$.

The *mean number of losses* due to full queue by time unit is denoted by $\overline{N_R}$ and is equal to: $\overline{N_R} = \lambda \cdot \pi(B,K)$.

The *mean response time* is denoted by $\overline{R}$ and is equal to: $\overline{R} = \overline{N_C}/\big(\lambda \cdot (1 - \pi(B,K))\big)$.

Energy consumption measures are defined now. The energy costs representation adopted here is mainly based on [10]. In this paper, the energy costs of a VM in use can be decomposed in two parts : static and dynamic costs. Static costs are mainly independent of the workload and comprise idle (or standby) consumption of the nodes, routers and consumption of the data center (cooling system, power distribution units,....) which is evaluated by the industrial metrics of the Power Unit Efficiency (PUE). On the other hand, dynamic costs include the energy consumption part of servers, storage devices and network that is induced by the resource usage and then depends on the workload. The hysteresis approach considers only the dynamic costs but static costs should be added in order to get the whole consumption of a VM. Hence, energy consumption is depending on both mean number of active servers (dynamic part of the cost) and mean number of their activation and deactivations, which represent the energy cost of the start (or pausing) and the data migration of a VM.

The *mean number of active servers* in the system is denoted by $\overline{N_S}$ and is equal to: $\overline{N_S} = \sum_{(m,k)\in A} S_k \cdot \pi(m,k)$.

The *mean number of activations triggered* by time unit, is denoted by $\overline{N_A}$ and is given by: $\overline{N_A} = \lambda \sum_{(m,k)\in A} (S_{k+1} - S_k) \cdot \mathbb{1}_{\{m=F_k; 1\leq k\leq K-1\}} \cdot \pi(m,k)$.

The *mean number of deactivations triggered* by time unit is denoted by $\overline{N_D}$ and is given by:

$$\overline{N_D} = \sum_{(m,k)\in A} \mu \min\{S_k, m\}(S_k - S_{k-1})\pi(m,k) \cdot \mathbb{1}_{\{m=R_{k-1}+1\,;\,1\leq k\leq K-1\}}$$

In order to consider both performance and energy consumption, then we define the overall expected cost by time unit for the underlying model as follows: $\overline{C} = C_H \cdot \overline{N_C} + C_S \cdot \overline{N_S} + C_A \cdot \overline{N_A} + C_D \cdot \overline{N_D} + C_R \cdot \overline{N_R}$. where, C_H is the per capita cost of holding one customer in the system within one time unit, C_S is the per capita cost of using one working server within one time unit, C_A is the activating cost (cost of switching one server from deactivating mode to activating mode), C_D is the deactivating cost and C_R is the cost of job losses due to full queue.

5 Numerical results

This section focuses on the analysis of the queueing model defined before (Section 2). We perform some numerical examples in order to show the interest of the model and the improvement of the resolution methods for the analysis of Cloud performance.

First, the three resolution approaches depicted in this work (SCA, LQBD and closed form solution) are compared and we observe which approach is the most relevant in terms of computational complexity and results accuracy. At last, using the most relevant resolution method, some use cases of cloud systems are analyzed and we observe some performance metrics. All evaluations were implemented in Matlab and performed on laptop with 64-bit Windows 10, 8 GB RAM and 2.00 GHZ Intel i7-4750HQ CPU.

5.1 Comparing the resolution approaches

Our objective here is to determine among the proposed resolution methods, the most relevant one. The relevance criteria are defined in terms of computation time and accuracy of the results. So, a set of experiments are performed for this purpose. We consider a threshold queueing model with hysteresis where the activation and deactivation of VMs are occurred one by one (ie. $S_1 = 1$, $S_{k+1} = S_k + 1$, $\forall k < K$, which means that $C = S_K = K$). This one by one activation case is considered since it represents the worst case in terms of computational complexity, and is thus the best way to compare the different proposed methods. We assume here that each server provides a service following an exponential distribution with rate 1. We generate several instances by increasing the size of the model (i.e. the number of levels (K) and the size of buffer B). We illustrate in Table 1, the computation times of each method for these instances. The forward and reverse thresholds are set as follow: $F = \{50,\ 100,\ 150, \ldots,\ B\}$ and $R = \{10,\ 40, 90,\ 140,\ 190, \ldots,\ B\}$. Threshold values have been taken arbitrarily and additional studies with different choices of threshold sequences will be the subject of future work.

	SCA with Power method	SCA with GTH method	LDQBD	Closed form
$\lambda = 2$, K = 10, B = 750, (1271 states)	2.933	0.0406	0.0121	0.00905
$\lambda = 10$, K = 100, B = 7500, (13421 states)	4117.59	0.9587	0.9889	0.0823
$\lambda = 10$, K = 500, B = 37500, (67421 states)	+3600	41.573	88.01	0.4979
$\lambda = 10$, K = 1000, B = 75000, (134921 states)	+3600	307.54	1330.27	1.0561

Table 1. Computational times (in seconds) of proposed resolution methods.

Through this table, one can clearly see that the closed form solution is the fastest one. This method is more than 1000 times faster than LQBD and 100 times than the SCA with GTH method. This result is expected because the closed form solution is based on a set of formulas containing basic operators contrary to LQBD method based on matrix inversion or SCA method with GTH approach where the numerical resolution approach GTH has a cubic complexity. It should be precised that since the SCA approach is a combination of state aggregation technique and numerical solution of Markov chain, then we propose to distinguish two numerical methods commonly used: the GTH [18] and power methods [17].

Considering the precision of the results, it could be noticed that even if the SCA and LQBD methods are numerical resolution approach, their precisions are not so far from the closed form method. Indeed, the gap on the stationary distribution vector between the different methods is smaller than 10^{-12}.

To confirm our conclusions, we propose to observe the relevance of the resolution methods according to the variation of the arrival rate λ. For this example, previously cited in the Table 1, parameters are $C = K = 100$, $B = 7500$ and $\mu = 1$. Then, we let vary the arrival rate from $\lambda = 1$ to $\lambda = 100$, and assess the computational resolution times of the closed form, LDQBD and SCA+GTH methods. In view of the computation times of the SCA + Power method (rather longer) this method is not considered in this comparative study. The obtained results are illustrated in the following figure.

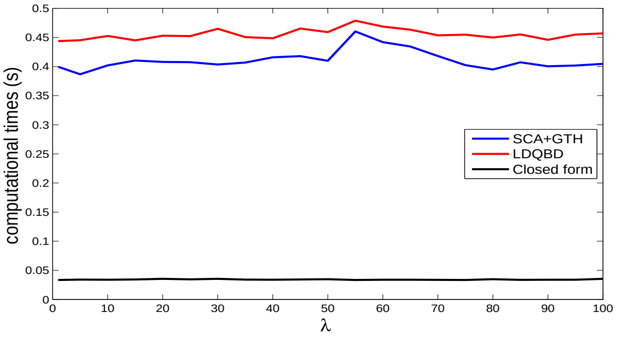
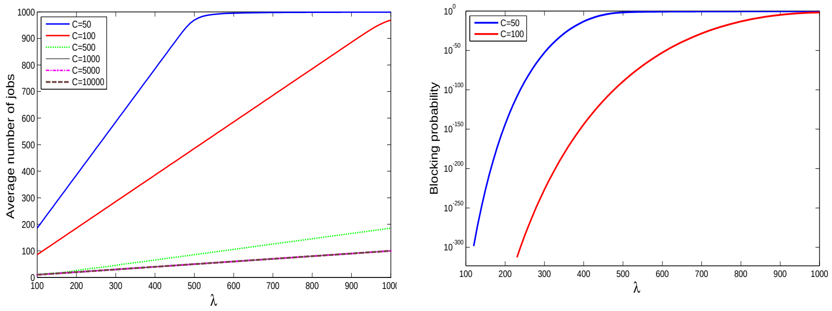


Fig. 2. Computational times (in seconds) versus arrival rate (λ).

In view of these results, it is clear that the closed form method is the most relevant resolution approach. However, for large ρ numerical methods could be more precise than the formal one due to the limits of the computer. This point should receive further investigations.

5.2 Performance and energy consumption measures

In this part, all computations are made with the closed form formula. We assess here the performance of a large cloud system and illustrate the trade-off between performance and energy consumption. We consider a multi server queue model driven by a hysteresis policy. We want to see the impact of the number of servers on the performance and the energy-efficiency of a cloud henceforth, the metrics defined Section 4 are used. We exhibit several cases in which our data center is composed by a pool of C virtual machines. It is assumed that the number C ranges from 50 to 10000 VMs, this last number being the size of a small data center. The buffer size is set to $B = 1000$ jobs, the service rate of each VM is set to $\mu = 10$ and we let vary the arrival rate varying between 100 and 1000 jobs/min. We assume that for each considered model there are fifty levels ($K = 50$). The forward thresholds and reverse thresholds are set respectively to $F = \{20, 40, 60, \ldots, 1000\}$ and $R = F - 10$. The sequence of service levels is taken as follows: $S = \{s \,|\, s = i \times \lfloor \frac{C}{50} \rfloor, \ \forall i = 0 \ldots 50\}$. Concerning the energy consumption parameters, we set the costs to 1: the energy consumption of one working server within one time unit is $C_S = 1$, the cost of holding one job in the system within one time unit is $C_H = 1$, the cost of activating or deactivating one server are respectively $C_A = 1$ and $C_D = 1$, and the cost of job losses due to full queue is $C_R = 1$.



(a) Average number of jobs in the system

(b) Blocking probability

Fig. 3. Performance metrics versus arrival rate (λ).

The performance results are illustrated in Figure 3(a) and 3(b). In Figure 3(b), one illustrates only the curves for $C = 50$ and $C = 100$, since the other models have a zero blocking probability. From these figures, we can obviously observe that the number of servers increase improves the performance. This is shown, in Figure 3, by the decrease of the number of jobs in the system and the blocking probability.

However, in terms of cost, one obviously observes the opposite when the system is moderately loaded. Hence, when the system is weakly / moderately loaded, the models that have a significant number of active VMs underperform comparatively to models with relatively few active VMs. Indeed, some VMs consume energy without performing any service. This can be seen on Figure 4 for $C = 10000$. On the other hand, when the system is overloaded, the cost of losses increases and affects the global cost. This can be clearly seen Figure 4 for the model with $C = 50$ when $\lambda > 450$ and for the one with $C = 100$ when

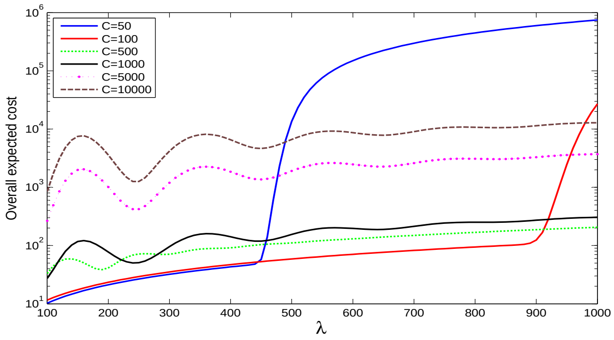


Fig. 4. Overall expected costs versus arrival rate (λ).

$\lambda > 900$. This is consistent with intuition. The oscillation phenomenon observed for large C remains unclear and deserves to be studied in more detail.

It could be noticed that the closed form resolution allows to compute the performance measures of all the instances in very short times (smaller than 2 seconds) even in cases where the number of VMs is 10000. Since concrete small cloud systems or cloud modules have a number of VMs around 10000, this shows the practical value of our method for answering rather quickly the questions about energy consumption and network dimensioning.

6 Conclusion

We develop numerical and analytical methods for the analysis of a hysteresis queueing system modelling a cloud system with activation/deactivation by block of VMs. One important contribution of this paper is to suppose few constraints on the thresholds. We give numerical values of the performance even in the case of large Markov chains, and show that our methods are hugely faster than the classical ones. We define a global cost for performance and energy consumption in order to propose a trade off between performance and energy consumption, and we analyse the impact of the thresholds on it. For the future, we need to analyze real cloud architectures with concrete energy consumptions for the VMs in order to compute relevant cost values. We also want to develop optimization algorithms to obtain the thresholds which minimize the overall cost.

Acknowledgement

This work was supported by grant ANR MARMOTE (ANR-12-MONU-0019).

References

1. Aït-Salaht, F., Castel-Taleb, H.: Bounding aggregations on phase-type arrivals for performance analysis of clouds. In: 24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016. pp. 319–324. IEEE (2016)
2. Ardagna, D., Casale, G., Ciavotta, M., Pérez, J., Wang, W.: Quality-of-service in cloud computing: modeling techniques and their applications. Journal of Internet Services and Applications 5 (2014)
3. 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)
4. Asghari, N., Mandjes, M., Walid, A.: Energy-efficient scheduling in multi-core servers. Computer Networks 59(11), 33–43 (2014)
5. Baumann, H., Sandmann, W.: Numerical solution of level dependent quasi-birth-and-death processes. Procedia Computer Science 1(1), 1561–1569 (2010)
6. Gandhi, A., Harchol-Balter, M., Adan, I.: Server farms with setup costs. Performance Evaluation 67(11), 1123–1138 (2010)
7. Gaujal, B., Hyon, E., Jean-Marie, A.: Optimal routing in two parallel queues with exponential service times. Discrete Event Dynamic Systems 16(1), 71–107 (2006)
8. Kandi, M., Aït-Salaht, F., Castel-Taleb, H., Hyon, E.: Mathematical methods for analyzing performance and energy consumption in the cloud. Tech. rep., Institut Mines-Telecom Telecom SudParis (2017)
9. Kitaev, M., Serfozo, R.: M/M/1 queues with switching costs and hysteretic optimal control. Operations Research 47, 310–312 (1999)
10. Kurpicz, M., Orgerie, A.C., Sobe, A.: How much does a vm cost? energy- proportional accounting in vm-based environments. In: PDP: Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. pp. 651–658 (2016)
11. Latouche, G., Ramaswami, V.: A logarithmic reduction algoritm for quasi-birth-death processes. Journal of Applied Probability 30, 650–674 (1993)
12. Le Ny, L.M., Tuffin, B.: A simple analysis of heterogeneous multi-server threshold queues with hysteresis. In: Applied Telecommunication Symposium (ATS) (2002)
13. Lui, J.C., Golubchik, L.: Stochastic complement analysis of multi-server threshold queues with hysteresis. Performance Evaluation 35(1), 19–48 (1999)
14. Mitrani, I.: Service center trade-offs between customer impatience and power consumption. Performance Evaluation 68(11), 1222–1231 (2011)
15. Mitrani, I.: Managing performance and power consumption in a server farm. Annals of Operations Research 202(1), 121–134 (2013)
16. Neuts, M.F.: Matrix-geometric solutions in stochastic models: an algorithmic approach. John Hopkins University Press (1981)
17. Philippe, B., Saad, Y., Stewart, W.J.: Numerical methods in markov chain modeling. Operations Research 40(6), 1156–1179 (1992)
18. Stewart, W.: Introduction to the numerical Solution of Markov Chains. Princeton University Press, New Jersey (1995)