Mathematical methods for analyzing performance and energy consumption in the cloud

M. Kandi F. Aït Salaht H. Castel-Taleb
INSTITUT TELECOM/Telecom SudParis
SAMOVAR, UMR 5157
9, rue Charles Fourier, 91011 Evry Cedex, France
Email: hind.castel@it-sudparis.eu

E. Hyon
Sorbonne Universités,
UPMC Univ Paris 06,
CNRS, LIP6 Paris UMR 7606,
4 place Jussieu 75005 Paris
Université Paris Nanterre

***Abstract*—We propose in this paper to evaluate the performance and the energy consumption of cloud system using mathematical methods. We consider for the analysis a hysteresis queueing system, which 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 analyze when the size of the system is huge. We propose to use in this case different mathematical methods in order to compute the steady-state probability distribution : the SCA (Stochastic Complement Analysis) method in order to aggregate the state space, LDQBD (Level Dependent Quasi Birth and Death method), and the balance equations in order to derive exact formulas for the steady-state probability distribution. We compute both performance and energy consumption measures and we define an overall cost taking into account both aspects. We compare the methods from their computation time and we analyze the impact of some parameters as the thresholds, and the arrival rate on the behavior of the system.**

I. INTRODUCTION

One of the most significant recent progresses in the field of information and communication technology is Cloud computing, which may change the way people do computing and manage information. In this environment, a pool of abstracted, virtualized, dynamically-scalable computing functions and services are made accessible over the internet to remote users in an on-demand fashion, without the need for infrastructure investments and maintenance.

Virtualization plays a key role in the success of cloud computing because it simplifies the delivery of the services by providing a platform for resources in a scalable manner. One physical host can have more than one VM (Virtual Machine: it is a software that can run its own operating system and applications just like an operating system on a physical computer). With this flexibility, the cloud providers can rent the virtual machines depending on the demand and can gain more profit out of a single physical machine. With virtualization, service providers can ensure isolation of multiple user workloads, provide resources in a cost-effective manner by consolidating VMs onto fewer physical resources when system load is low, and quickly scale up workloads to more physical resources when system load is high. In [?], they study the right ratio of VM instances to physical processors that optimizes the workload's performance given a workload and a set of physical computing resources.

Performance evaluation of cloud centers is an important research task which becomes difficult because the dynamic nature of cloud environments and diversity of user requests. Then, it is not surprising that in the recent area of cloud computing, only a portion of research results has been devoted to performance evaluation. In [?], they develop an analytical model in order to evaluate the performance of cloud centers with a high degree of virtualization and Poisson batch arrivals. The model of the physical machine with m VMs is based on the $M^{[x]}/G/m/m+r$ queue. They derive exact formulas for performance measures as blocking probability and mean waiting time of tasks. In [?], they consider a cloud center with a number of physical machines that are allocated to users in the order of task arrivals. Physical Machines (PMs) are considered with a high degree of virtualization, and are categorized into three server pools: hot, warm, and cold. The authors implement the sub-models using interactive Continuous Time Markov Chain (CTMC). The sub-models are interactive such that the output of one sub-model is input to the other one.

In this paper, we propose to use a mathematical model in order to evaluate the performance of a cloud node, more precisely, a data center. We represent the system by a queueing model based on queue-dependent virtual machines in order to analyse quantitatively the dynamic behavior of the data center. The data center is represented by a set of PM (Physical Machines) hosting a set of VMs which are instanced according to user demand. In this paper, we represent the data center as a set VMs which could be very large, especially if the user demand is high. With this model, virtual machines are activated and deactivated according to the intensity of user demand. The queueing model is a multi-server with threshold queues and hysteresis [?]. We suppose that customer requests arrivals follow a bulk process. Each server represents a VM, and the multi-server queueing model with hysteresis is governed by a sequence of forward and reverse thresholds which are different. The forward (resp. the backward) thresholds represent the value of the number of customers from which an additional VM is activated (resp. deactivated). Obviously, the relevance of this model is to offer the flexibility of different thresholds for activating and removing VMs.

As the system is difficult to analyze exactly, especially when the number of VMs or the size of bulk arrivals is high, we propose to use stochastic comparisons in order to compute more easily, and so faster performance measure bounds. The bounding models are obtained by the simplification of the hysteresis model in order to compute easily the performance measures. We propose to simplify the batch arrival process by generating aggregated bounding processes. So the bounding systems are equivalent to the hysteresis system with aggregated bounding arrival process. We derive an upper bounding system (resp. a lower bounding system) from an upper bound batch arrival distribution (resp. a lower bound batch arrival distribution). We prove using stochastic comparisons that these processes provide really bounds for performance measures as blocking probabilities, expected buffer length and expected departure.

We give some numerical values according to different values of input parameters: arrival rate, size batches, and the number of VMs (called the degree of virtualization). The results show clearly the relevance of our approach to propose a tradeoff between computational complexity and accuracy of results. So it can efficiently solve the network dimensioning problem from QoS (Quality of Service) constraint requirements.

The paper is organized as follows: next, we describe the cloud system, and in section II, we present the queueing model for the analysis. In section III, we give some theoretical notions of the stochastic ordering theory and in section **??**, we give the bounding models and we prove using the stochastic comparisons that they represent really bounds. In the section **??**, we give numerical results of the performance measures. Finally, achieved results are discussed in the conclusion and comments about further research issues are given. The activation/deactivation one by one model has been extensively studied in the literature [1]–[3]. In [1], the authors use the concept of stochastic complementation to solve the system. They propose to partition the state space in disjoint sets in order to aggregate the Markov chain. The main advantage of this method is to obtain exact performance results, with reduced execution times. In [2], Le Ny et al. propose to compute the steady-state probabilities of a heterogeneous multi-server threshold queue with hysteresis by using a closed-form solution. And In [3], the authors propose to analyze this system through stochastic bounding theory. Confronted with a computational complexity problem in the Cloud system (the cloud systems are often defined on very large state spaces which makes their exact analysis very cumbersome or even impossible), the authors derived bounding models and defined an accurate bounds on performance measures. They offer a trade-off between the accuracy of the results and the computation time.

In this paper, we propose to study the Activation/Deactivation model block by block, which to our knowledge has never been considered and studied previously in the literature. So, we try in the following, to investigate this model and present some analysis and resolution methods.

II. Cloud system description

We analyse a data center in a cloud system composed by a set of Virtual Machines (VMs). We assume that the job requests arrive at the system following a Poisson process with rate λ, and are enqueued in a finite queue with capacity B. An arriving request can be rejected if it finds the buffer full. We model this system 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 are activated and deactivated according to the system occupancy. Actually, the buffer management is governed by thresholds vectors corresponding to the number of customer waiting in the system, which controls the operation of activating and deactivating the VMs. We suppose the case where the VMs are activated or deactivated by block, which means that several VMs can be simultaneously activated or deactivated.

We define K functioning level, where each level corresponds to a certain number of active servers. The number of active servers at level i is denoted by S_i, where $S_1 \leq S_2 \leq ... \leq S_K$. We suppose that $S_1 \geq 1$, so we have at least one active server by assumption. The transition from functioning level i to level $i+1$ allows to allocate (turn on) one or more additional servers, going from S_i to S_{i+1} active servers, while the transition from level i to level $i-1$ allows to remove (turn off) one or more active servers, going from S_i to S_{i-1} active servers. Depending on the system occupancy, we transit from the level i to level $i+1$ when the workload in the system exceeds a threshold F_i, and from level i to level $i-1$ when the workload in the system falls below a threshold R_{i-1}. So, the model is 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 $R_i < F_i, \forall i, 1 \leq i \leq K-1$.

We assume here that server deactivations occur at the end of the service, and when multiple servers are deactivated at the same times, all the customers who have not completed their service return to the queue.

The underlying model is described by the Continuous-Time Markov Chains (CTMCs), denoted $\{X(t)\}_{t\geq 0}$. A state is represented by a couple $(x_1,\, x_2)$ such that x_1 is the number of customers in the system and x_2 is the functioning level. The state space is denoted by A and is given as follows:

$$
\begin{aligned}
A = \{(x_1, x_2)\,|\, & 0 \leq x_1 \leq F_1, \text{ if } x_2 = 1; \\
& R_{x_2-1} + 1 \leq x_1 \leq F_{x_2}, \text{ s.t. } 1 < x_2 < K; \\
& R_{K-1} + 1 \leq x_1 \leq B, \text{ if } x_2 = K; \}
\end{aligned}
$$

Recalling that S_{x_2} represents the number of active servers at level x_2, the transitions between states follows:

$$
\begin{aligned}
(x_1,\, x_2) \;\rightarrow\;& (\min\{B, x_1+1\},\, x_2) \\
& \text{with rate } \lambda\,, \text{ if } x_1 < F_{x_2}; \\
\rightarrow\;& (\min\{B, x_1+1\},\, \min\{K,\, x_2+1\}) \\
& \text{with rate } \lambda, \text{ if } x_1 = F_{x_2}; \\
\rightarrow\;& (\max\{0, x_1-1\},\, x_2), \\
& \text{with rate } \min\{S_{x_2},\, x_1\}\cdot\mu, \\
& \text{if } x_1 > R_{x_2-1}+1; \\
\rightarrow\;& (\max\{0, x_1-1\},\, \max\{0, x_2-1\}) \\
& \text{with rate } \min\{S_{x_2},\, x_1\}\cdot\mu, \\
& \text{if } x_1 = R_{x_2-1}+1\,.
\end{aligned}
$$

An example of the transitions is given Figure 1.

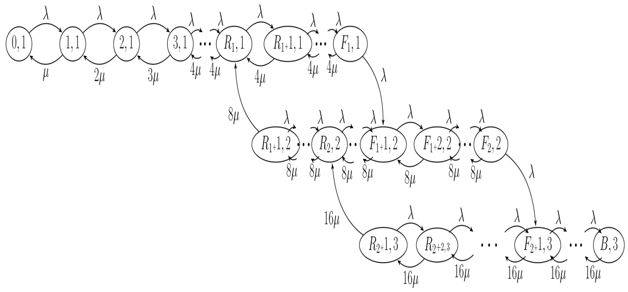

Fig. 1. Transition structure for $K = 3$, $S_1 = 4$, $S_2 = 8$; $S_3 = 16$, $R_1 + 1 \geq 8$ and $R_2 + 1 \geq 16$.

III. Resolution approaches

In order to compute the exact performance measures of the presented models, we expose hereafter some techniques to solve the CTMCs and compute the steady state probability vector. In this section, we present three resolution methods. A comparison in terms of execution times of these methods is presented in Section V.

A. Stochastic Complement Analysis (SCA)

To solve the $X(t)$ and $Y(t)$ Markov chains, the first approach consists to aggregate the underlying chain and use a numerical method to compute the steady state distribution. This approach has been proposed by Lui et al. [1] and works as follows. First, we aggregate the state space of the underlying chain by partitioning the set A into disjoint subsets. The number of derived subsets depends on the number of functioning level. From each subset, we define a corresponding Markov chain. These derived Markov chains are defined on reduced state spaces which makes their analysis less complex. The resolution of each Markov chain defines a conditional steady state probabilities. By applying the state aggregation technique, each subset 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. We note that the compute of the steady state probabilities of a Markov chain can be obtained using any chosen solution technique, as described in [4]. Lastly, a disaggregation technique is applied to compute the individual steady state probabilities of the original Markov process.

In the following, we present an important theorem stated by Lui et al. in their article [1].

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

$$Q = \begin{pmatrix} Q_{A_1A_1} & Q_{A_1A_2} \\ Q_{A_2A_1} & Q_{A_2A_2} \end{pmatrix} \tag{1}$$

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

Based on this theorem Lui et al. have investigated the "A/D One by One" model using some restrictions. Here, and without any restrictions, we propose to give first a general formulation for the "A/D One by One" model and then state some results for the "A/D by block" model.

Given the $\{X_t\}_{t\geq 0}$ Markov chain (resp. $\{Y_t\}_{t\geq 0}$ Markov chain) with state space A. We partition the state space A into K distinct sets denoted A_m, where:

$$A_m = \{(x_1, x_2)|(x_1, x_2) \in A \text{ and } x_2 = m\}; \forall m \in \{1, 2, ..., K\} \tag{2}$$

The set A_m contains the states belonging the level m.

Let $\{(X_m)_t\}_{t\geq 0}$ (resp. $\{Y_t\}_{t\geq 0}$) be a Markov chain defined on state space A_m, $\forall m \in \{1, 2, ..., K\}$. We denote by π_m the steady state probabilities of X_m (resp. Y_m) . The transitions in $\{(X_m)_t\}_{t\geq 0}$ (resp. $\{(Y_m)_t\}_{t\geq 0}$) are identical to those appear in the original process $\{X_t\}_{t\geq 0}$ (resp. $\{(X)_t\}_{t\geq 0}$) for the level m except some additional modifications. This modifications are set out below.

For $m = 1$:

- We add a transition from state $(F_1, 1)$ to state $(R_1, 1)$, with a rate λ.

For $m \in \{2, \ldots, K-1\}$:

- We add a transition from state (F_m, m) to state (R_m, m), with a rate λ.
- and a transition with a rate $(\min\{R_{m-1}+1, S_m\} * \mu)$ is added from $(R_{m-1}+1, m)$ to state $(F_{m-1}+1, m)$.

For $m = K$:

- We add a transition from state $(R_{K-1}+1,\ K)$ to state $(F_{K-1}+1,\ K)$, with a rate $\min\{R_{K-1}+1, S_K\} * \mu$.

The aggregated process that brings all aggregate states is a simple birth and death process, with:

- $\lambda_i = \lambda * \pi_i(F_i), \quad \forall i \in 1, ...K-1$ and
- $\mu_i = \min(R_{i-1}+1, S_i) * \mu * \pi_i(R_{i-1}+1), \quad \forall i \in 2, ...K.$

We denote by π the steady state probabilities of this aggregated process.

At this point we have all the necessary information to compute the steady probabilities of $\{X_t\}_{t\geq 0}$ (resp. $\{Y_t\}_{t\geq 0}$). Indeed, we determine: (1) for each level m ($m = 1 \ldots K$), the conditional state probabilities of all states, and (2) the steady state probability of the aggregated process. Hence, the steady state probability of each individual state (i, j) in $\{X_t\}_{t\geq 0}$ (resp. $\{Y_t\}_{t\geq 0}$), can be expressed as

$$\Pi(i,\ j) = \pi_j(i)\,\pi(j) \ \text{ where } \ (i,\ j) \in A_j.$$

B. Closed-form solution: QBD

The particular form of the Markov Chain generator suggest us to use the Quasi Birth and Death (QBD) processes framework to benefits of the numerous numerical methods to solve them [5]. For short a QBD process is a stochastic process in which the state space is two dimensionals 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 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 independant* but when the matrices are different the QBD is said *level dependant* (LDQBD).

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

Proposition 1: The Markov Chain $\{X_t\}_{t\geq 0}$ defined section II is a level dependant QBD with K levels, corresponding to the functioning levels, with a generator Q given by:

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

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

For $n = 1$ 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 $n \in 2, ...K-1$, we get:

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

also

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

and

$$Q_{n,n+1}(i,j) = \begin{cases} \lambda & \text{if } i = d_n \text{ and } j = F_n - R_n + 1 \\ 0 & \text{otherwise} \end{cases}.$$

Finally for $n = K$, it follows

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

and

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

Proof: Let i, j be the coordinate on line j and colum j of the generator Q. It records the transition from the ith state to the jth state. Let us describe now which is the ith state. Recall that, by definition, the number of customers on a given level n can vary from $R_{n-1}+1$ to F_n. Let n be the number such that $\sum_{l=1}^{n-1} d_l < i$ and such that $\sum_{l=1}^{n} d_l \geq i$ then the functionning level of the ith state is $x_2 = n$ and the number of customers is $x_1 = R_{n-1} + \left(i - \sum_{l=1}^{n-1} d_l\right)$.

We detail now the possible transitions from state (x_1, x_2). We first study the transition inside a level. We can jump in $\min\{B, x_1+1\}$ with rate λ and then $j = i+1$. We can also jump in state $\max\{0, x_1-1\}$ with rate $\min\{S_n, x_1\}$ and then $j = i-1$. At last, it can be noticed that, by construction of the matrices, the coordinate (i,i) of the generator Q is the coordinate (i', i') of the matrix $Q_{n,n}$ with $i' = x_1 - R_{n-1}$.

Let us study the transition to the upper level. If $x_1 = F_n$ then we can only jump to $(\min\{F_n+1, B\}, \max\{K, n+1\})$ with rate λ and In this case $j = \sum_{l=1}^{n} d_l + F_n - R_n + 1$. Furthermore, the coordinate $(i, i + F_n - R_n + 1)$ with $i = \sum_{l=1}^{n} d_l$ of the generator Q is the coordinate $(d_n, F_n - R_n + 1)$ of the matrix $Q_{n,n+1}$.

Let us study the transition to the lower level. If $x_1 = R_{n-1}+1$ then we can only jump to $(\max\{x_1 - 1, 0\}, \max\{1, x_2 - 1\})$ with rate $\mu \min\{S_n, x_1\}$ and in this case $j = \sum_{l=1}^{n-2} d_l + R_{n-1} - R_{n-2}$. Furthermore, the coordinate $((\sum_{l=1}^{n-1} d_l) + 1, \sum_{l=1}^{n-2} d_l + R_{n-1} - R_{n-2})$ of the generator Q is the coordinate $(1, R_{n-1} - R_{n-2})$ of the matrix $Q_{n,n-1}$.

There is no other possible transitions, the terms at the bounds of the state space should be adapted and the terms on the diagonal follow. Thus the generator has a tridiagonal form. Hence for a given n the matrix $Q_{n,n-1}$ (resp $Q_{n,n}$, $Q_{n,n+1}$ records the events associated with a decrease of the level (resp staying in same level and an increase of the level). ■

Numerically solving QBD as well as level dependant QBD is often based on a matrix geometric methods [**?**], [5] or kernel methods [6]. This is an hard computationnal task requiring to solve matrices equation. Equally, for LDQBD it exists numerical methods to solve them. Here the method proposed in [7] is used since it is shown that this method is efficient and numerically stable.

C. Mathematical analysis using balance equations

We give a closed form for the steady state probability using balance equations, and cuts on the state space. As in [2], we compute the probabilities level by level, where one level corresponds to a certain number of active servers. In the sequel, we give the algorithm which compute π.

Algorithm 1 Compute π

```
{probabilities of level 1}
for 1 ≤ m ≤ F_1 do
  if m < S_1 then
    compute π(m,1) from equation 3
  else if S_1 < m ≤ R_1 then
    compute π(m,1) from equation 4
  else
    compute π(m,1) from equation 5
  end if
end for
{probabilities of level k }
for 2 ≤ k ≤ K − 1 do
  compute π(R_{k−1} + 1, k) from equation 11
  compute π(R_k + 1, k + 1) from equation 29
  if R_k ≤ F_{k−1} then
    for R_{k−1} + 2 ≤ m ≤ F_k do
      if R_{k−1} + 2 ≤ m ≤ R_k then
        compute Π(m,k) from equation 30
      else if R_k + 1 ≤ m ≤ F_{k−1} + 1 then
        compute Π(m,k) from equation 31
      else
        compute Π(m,k) from equation 32
      end if
    end for
  else
    for R_{k−1} + 2 ≤ m ≤ F_k do
      if R_{k−1} + 2 ≤ m ≤ F_{k−1} + 1 then
        Compute Π(m,k) from equation 12
      else if F_{k−1} + 2 ≤ m ≤ R_k then
        Compute Π(m,k) from equation 13
      else
        Compute Π(m,k) from equation 14
      end if
    end for
  end if
end for
{probabilities of level K }
compute π(R_{K−1} + 1, K) from equation 11
for R_{k−1} + 2 ≤ m ≤ B do
  if R_{K−1} + 2 ≤ m ≤ F_{K−1} + 1 then
    Compute Π(m,k) from equation 33
  else
    Compute Π(m,k) from equation 34
  end if
end for
```

The relevance of our work is that we take more general cases for the thresholds for each level $2 \leq k \leq K$: $R_k \leq F_{k-1}$, and $R_k > F_{k-1}$. We suppose that for each level $2 \leq k \leq K$, $R_{k-1}+1 \geq S_k$, so the service rate for each level is $min(R_{k-1}+1, S_k) = S_k\mu$. The level one is a particular case, as the service rate depends on the number of customers in the system : so for a state $(m,1)$, where $1 \leq m < S_1$, the service rate is $m\mu$, and it is $S_1\mu$, if $m \geq S_1$.

In the sequel, we denote by $\mu_k = S_k\mu$, for $k = 1,\ldots,K$, and by $\rho_k = \frac{\lambda}{\mu_k}$, for $k = 1,\ldots K$. We consider also that $\rho = \frac{\lambda}{\mu}$. In the sequel, we give the probabilities 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 $2 \leq k \leq K$, 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 $\Pi(F_{k-1}+1,k-1)$ which has been already computed. After that the other probabilities of the level k are computed in terms of $\Pi(R_{k-1}+1,k)$, so 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.

1) Analysis of level 1: The following lemma gives the steady-state probabilities for level 1.

Lemma 1: We have three cases:

- if $0 \leq m \leq S_1$:

$$\pi(m,1) = \frac{\rho^m}{m!}\pi(0,1) \tag{3}$$

- if $S_1 < m \leq R_1$:

$$\pi(m,1) = \rho_1^{m-S_1}\frac{\rho^{S_1}}{S_1!}\pi(0,1) \tag{4}$$

- if $R_1 + 1 \leq m \leq F_1$:

$$\pi(m,1) = \frac{\rho^{S_1}}{S_1!}(\rho_1^{m-S_1} - \frac{\rho_1^{F_1-S_1+1}(1-\rho_1^{m-R_1})}{1-\rho_1^{F_1-R_1+1}})\pi(0,1) \tag{5}$$

We present now the proof of lemma 1. In the level one, it is logical to suppose that $R_1 \geq S_1$. We propose to make cuts in the Markov chain diagram around sets $\{(0,1),\ldots,(m,1)\}$. For $0 \leq m < R_1$, we have the following evolution equation: $\mu(m+1)\pi(m+1,1) = \lambda\pi(m,1)$.

So if $0 \leq m \leq S_1$, we can deduce equation 3, and if $S_1 < m \leq R_1$, we obtain equation 4.

Making cuts around the sets $\{(0,1),\ldots,(R_1,1),\ldots,(m,1)\}$, for $R_1 \leq m \leq F_1 - 1$, allows to derive the following balance equations : $\mu_1\pi(m+1,1) + \mu_2\pi(R_1+1,2) = \lambda\pi(m,1)$. So we obtain if $R_1 + 1 \leq m \leq F_1$:

$$\pi(m,1) = \rho_1^{m-R_1}\pi(R_1,1) - \pi(R_1+1,2)\frac{\rho_1}{\rho_2}\sum_{k=0}^{m-R_1-1}\rho_1^k \tag{6}$$

And we deduce that:

$$\pi(m,1) = \rho_1^{m-R_1}\pi(R_1,1) - \pi(R_1+1,2)\frac{\rho_1}{\rho_2}\frac{1-\rho_1^{m-R_1}}{1-\rho_1} \tag{7}$$

From equation 7, we deduce for $m = F_1$:

$$\pi(F_1,1) = \rho_1^{F_1-R_1}\pi(R_1,1) - \pi(R_1+1,2)\frac{\rho_1}{\rho_2}\frac{1-\rho_1^{F_1-R_1}}{1-\rho_1} \tag{8}$$

If we make a cut between states of level 1 and states of other levels, then we obtain the following evolution equation: $\lambda\pi(F_1,1) = \mu_2\pi(R_1+1,2)$, so we deduce that: $\pi(F_1,1) = \frac{1}{\rho_2}\pi(R_1+1,2)$.. From equation (4), for $m = R_1$, we have that:

$$\pi(R_1,1) = \rho_1^{R_1-S_1}\frac{\rho^{S_1}}{S_1!}\pi(0,1) \tag{9}$$

So, we deduce that:

$$\pi(R_1+1,2) = \frac{\rho_2\rho_1^{F_1-S_1}(1-\rho_1)}{1-\rho_1^{F_1-R_1+1}}\frac{\rho^{S_1}}{S_1!}\pi(0,1) \tag{10}$$

Then using equations 9, and equation 10, in equation 7, we deduce for $R_1 + 1 \leq m \leq F_1$, equation (5).

2) Analysis of level k: We consider now a level k such that $2 \leq k \leq K-1$. If we consider the cut of the state space between states of level $k-1$ and states of level k, we have the following evolution equation: $\pi(F_{k-1}, k-1)\lambda = \pi(R_{k-1}+1, k)\mu_k$. Which is equivalent to:

$$\pi(R_{k-1}+1, k) = \rho_k \pi(F_{k-1}, k-1) \quad (11)$$

Now, we compute the probabilities of each level, by expressed each of them in terms of $\pi(R_{k-1}+1, k)$. According to the threshold values, we have two cases to consider: $R_k \leq F_{k-1}$ or $R_k > F_{k-1}$. Note that the case of $R_k \leq F_{k-1}$, has been considered in [2], so we give just the main equations. We present in details the case $R_k > F_{k-1}$.

Case 1 : if $\mathbf{R_k > F_{k-1}}$:we have the following lemma :

Lemma 2: For the level k such that $2 \leq k < K$ we have three cases:

- if $R_{k-1}+2 \leq m \leq F_{k-1}+1$

$$\pi(m, k) = \frac{1-\rho^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \quad (12)$$

- if $F_{k-1}+2 \leq m \leq R_k$

$$\pi(m, k) = \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \quad (13)$$

- if $R_k+1 \leq m \leq F_k$:

$$\begin{aligned}\pi(m, k) &= \frac{\rho_k^{m-F_{k-1}-1} - \rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \\ &- \frac{\rho_k}{\rho_{k+1}} \frac{1-\rho_k^{m-R_k}}{1-\rho_k}\pi(R_k+1, k+1) \quad (14)\end{aligned}$$

If we consider cuts on the state space around the sets $\{(R_{k-1}+1, k), \ldots, (m, k)\}$ for $R_{k-1}+1 \leq m \leq F_{k-1}$, then we obtain the following equation : $\lambda\pi(m, k) + \mu_k\pi(R_{k-1}+1, k) = \mu_k\pi(m+1, k)$.

So we deduce if $R_{k-1}+2 \leq m \leq F_{k-1}+1$:

$$\pi(m, k) = \rho_k\pi(m-1, k) + \pi(R_{k-1}+1, k) \quad (15)$$

And we obtain from equation 15 by induction equation 12 if $R_{k-1}+2 \leq m \leq F_{k-1}+1$.

If we consider cuts on the state space around sets $\{(R_{k-1}+1, k), \ldots, (F_{k-1}+1, k), \ldots, (m, k)\}$, for $F_{k-1}+1 \leq m \leq R_k-1$, then we obtain the following balance equation : $\lambda\pi(m, k) + \mu_k\pi(R_{k-1}+1, k) = \mu_k\pi(m+1, k)$.

So we get for $F_{k-1}+2 \leq m \leq R_k$:

$$\begin{aligned}\pi(m, k) &= \rho_k\pi(m-1, k) - \rho_k\pi(F_{k-1}, k-1) \\ &+ \pi(R_{k-1}+1, k) \quad (16)\end{aligned}$$

So, we deduce from (16) that:

$$\begin{aligned}\pi(m, k) &= \rho_k^{m-F_{k-1}-1}\pi(F_{k-1}+1, k) \\ &+ \sum_{i=0}^{m-F_{k-1}-2} \rho_k^i \pi(R_{k-1}+1, k) \\ &- \sum_{i=1}^{m-F_{k-1}-1} \rho_k^i \pi(F_{k-1}, k-1) \quad (17)\end{aligned}$$

From Equation (12), for $m = F_{k-1}+1$, we obtain:

$$\pi(F_{k-1}+1, k) = \frac{1-\rho^{F_{k-1}+1-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \quad (18)$$

So from equations (18), (11), and (17) we obtain:

$$\begin{aligned}\pi(m, k) &= \frac{1-\rho_k^{m-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \\ &- \sum_{i=1}^{m-F_{k-1}-1} \rho_k^i \frac{1}{\rho_k}\pi(R_{k-1}+1, k) \quad (19)\end{aligned}$$

So we obtain equation 13 if $F_{k-1}+2 \leq m \leq R_k$. If we make cuts on sets of states $\{(R_{k-1}+1, k), \ldots, (R_k, k), \ldots, (m, k)\}$, for $R_k \leq m \leq F_k-1$, then we obtain the following balance equations:

$$\begin{aligned}\pi(m, k)\lambda + \pi(R_{k-1}+1, k)\mu_k &= \mu_k\pi(m+1, k) \\ &+ \mu_{k+1}\pi(R_{k-1}+1, k) \\ &- \mu_{k+1}\pi(R_k+1, k+1) \\ &+ \lambda\pi(F_{k-1}, k-1) \quad (20)\end{aligned}$$

From equation 20, we have for $R_k \leq m \leq F_k-1$

$$\begin{aligned}\pi(m+1, k) &= \rho_k\pi(m, k) + \pi(R_{k-1}+1, k) \\ &- \frac{\rho_k}{\rho_{k+1}}\pi(R_k+1, k+1) \\ &- \rho_k\pi(F_{k-1}, k-1) \quad (21)\end{aligned}$$

From equation 11, we have:

$$\pi(F_{k-1}, k-1) = \frac{1}{\rho_k}\pi(R_{k-1}+1, k) \quad (22)$$

Using 22 in equation 21, we obtain:

$$\pi(m+1, k) = \rho_k\pi(m, k) - \frac{\rho_k}{\rho_{k+1}}\pi(R_k+1, k+1) \quad (23)$$

By induction, we derive the following equation for $R_k+1 \leq m \leq F_k$:

$$\begin{aligned}\pi(m, k) &= \rho_k^{m-R_k}\pi(R_k, k) \\ &- \frac{\rho_k}{\rho_{k+1}} \sum_{i=0}^{m-R_k-1} \rho_k^i \pi(R_k+1, k+1) \quad (24)\end{aligned}$$

From equation 13, for $m = R_k$, we obtain:

$$\pi(R_k, k) = \frac{\rho_k^{R_k-F_{k-1}-1} - \rho_k^{R_k-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \quad (25)$$

So using equation 25 in equation 24, we obtain for $R_k+1 \leq m \leq F_k$:

$$\begin{aligned}\pi(m, k) &= \rho_k^{m-R_k}\frac{\rho_k^{R_k-F_{k-1}-1} - \rho_k^{R_k-R_{k-1}}}{1-\rho_k}\pi(R_{k-1}+1, k) \\ &- \frac{\rho_k}{\rho_{k+1}}\frac{1-\rho_k^{m-R_k}}{1-\rho_k}\pi(R_k+1, k+1) \quad (26)\end{aligned}$$

So from Equation (26), we derive equation 14 if $R_k + 1 \leq m \leq F_k$:

As we need to compute $\pi(R_k+1, k+1)$, then from Equation (11), we have:

$$\pi(F_k, k) = \frac{1}{\rho_{k+1}} \pi(R_k + 1, k + 1) \tag{27}$$

And using Equation (26) for $m = F_k$, we obtain:

$$\begin{aligned} \pi(F_k, k) \quad = \quad & \frac{\rho_k^{F_k - F_{k-1} - 1} - \rho_k^{F_k - R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) \\ & - \frac{\rho_k}{\rho_{k+1}} \frac{1 - \rho_k^{F_k - R_k}}{1 - \rho_k} \pi(R_k + 1, k + 1) \end{aligned} \tag{28}$$

So using equations 27 and 28, we derive:

$$\begin{aligned} & \pi(R_{k+1}, k+1) = \\ & \rho_{k+1} \frac{\rho_k^{F_k - F_{k-1} - 1} - \rho_k^{F_k - R_{k-1}}}{1 - \rho_k^{F_k - R_k + 1}} \pi(R_{k-1} + 1, k) \end{aligned} \tag{29}$$

Case 2 : if $\mathbf{R_k \leq F_{k-1}}$: we give the main equations, and the details are given in [2].

- if $R_{k-1} + 2 \leq m \leq R_k$:

$$\pi(m, k) = \frac{1 - \rho_k^{m - R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) \tag{30}$$

- if $R_k + 1 \leq m \leq F_{k-1} + 1$,

$$\begin{aligned} \pi(m, k) \quad = \quad & \frac{1 - \rho_k^{m - R_{k-1}}}{1 - \rho_k} \pi(R_{k-1} + 1, k) \\ & - \frac{\rho_k}{\rho_{k+1}} \frac{1 - \rho_k^{m - R_k}}{1 - \rho_k} \pi(R_k + 1, k + 1) \end{aligned} \tag{31}$$

- if $F_{k-1} + 2 \leq m \leq F_k$

$$\begin{aligned} \pi(m, k) \quad = \quad & \rho_k^{m - F_{k-1} - 1} \frac{1 - \rho_k^{F_{k-1} - R_{k-1} + 1}}{1 - \rho_k} \pi(R_{k-1} + 1, k) \\ & - \frac{\rho_k}{\rho_{k+1}} \frac{1 - \rho^{m - R_k}}{1 - \rho_k} \pi(R_k + 1, k + 1) \end{aligned} \tag{32}$$

3) For the level K: The steady-state probabilities are given in the following lemma :

Lemma 3: We have two cases:

- if $R_{K-1} + 2 \leq m \leq F_{K-1} + 1$:

$$\pi(m, K) = \frac{1 - \rho_k^{m - R_{K-1}}}{1 - \rho_K} \pi(R_{K-1} + 1, K) \tag{33}$$

- if $F_{K-1} + 2 \leq m \leq B$:

$$\pi(m, K) = \rho_K^{m - F_{K-1} - 1} \frac{1 - \rho_k^{F_{K-1} + 1 - R_{K-1}}}{1 - \rho_K} \pi(R_{K-1} + 1, K) \tag{34}$$

IV. Performance measures and energy cost parameters

We propose in this section to calculate the expected cost in terms of performances and energetic consumption for the models presented in this paper. This requires to compute the performance measures of the Markov Chain associated with all the parameters. Once the steady state vector known we can deduce various performance measures, indeed we are interested by the performance measures which can be expressed under the form of a Markov reward function $\mathcal{R}$, where $\mathcal{R} = \sum_{i,j} \pi(i,\, j)\, R(i,\, j)$ and $R(i,\, j)$ is the reward for state $(i,\, j)$.

We present now the metrics we consider. We denote by $\overline{N_C}$ the *mean number of customers* in the system. It is worth

$$\overline{N_C} = \sum_{(x_1, x_2) \in A} x_1 * \pi(x_1, x_2)\,. \tag{35}$$

We denote by $\overline{N_S}$ the *mean number of active servers* in the system. It is given by:

$$\overline{N_S} = \sum_{(x_1, x_2) \in A} S_{x_2} * \pi(x_1, x_2)\,. \tag{36}$$

We denote by $\overline{N_A}$ the *mean number of activations* triggered by time unit. It is given by:

$$\overline{N_A} = \lambda \sum_{(x_1, x_2) \in A} (S_{x_2+1} - S_{x_2}) \cdot 1_{\{x_1 = F_{x_2};\, 1 \leq x_2 \leq K-1\}} \cdot \pi(x_1, x_2). \tag{37}$$

We denote by $\overline{N_D}$ the *mean number of deactivations* triggered by time unit. It is given by:

$$\begin{aligned} \overline{N_D} = \sum_{(x_1, x_2) \in A} & \min\{S_{x_2}, x_1\} \cdot \mu \cdot (S_{x_2} - S_{x_2 - 1}) \cdot \\ & 1_{\{x_1 = R_{x_2 - 1} + 1 \;;\; 1 \leq x_2 \leq K-1\}} \cdot \pi(x_1, x_2) \end{aligned} \tag{38}$$

We denote by $\overline{N_R}$ the *mean number of losses* due to full queue, it is equal to:

$$\overline{N_R} = \lambda * \pi(B, K) \tag{39}$$

We denote by $\overline{R}$ the *mean response time* which is

$$\overline{R} = \frac{\overline{N_C}}{\lambda * (1 - \pi(B, K))} \tag{40}$$

The overall expected cost by time unit for the underlying model is given by:

$$\overline{C} = C_H * \overline{N_C} + C_S * \overline{N_S} + C_A * \overline{N_A} + C_D * \overline{N_D} + C_R * \overline{N_R}\,, \tag{41}$$

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 (cots of switching one server from deactivating mode to activating mode), $C_D$is the deactivating cost and C_R is the cost of losses jobs due to full queue.

TABLE I. COMPARISON BETWEEN RESOLUTION APPROACHES IN TERMS OF EXECUTION TIME

-	SCA + GTH	LDQBD	Balance equations
K = 5 B = 300 (521 stats)	0.246 sec	0.017 sec	0.006 sec
K = 10 B = 750 (1271 stats)	1.678 sec	0.041 sec	0.011 sec
K = 50 B = 3750 (6671 stats)	89.010 sec	0.496 sec	0.095 sec
K = 100 B = 7500 (13421 stats)	679.342 sec	2.775 sec	0.282 sec
K = 500 B = 37500 (67421 stats)	+30 min	304.437 sec	7.408 sec
K = 1000 B = 75000 (134921 stats)	+30 min	”Out of memory” (inversion of a very large matrix)	34.401 sec

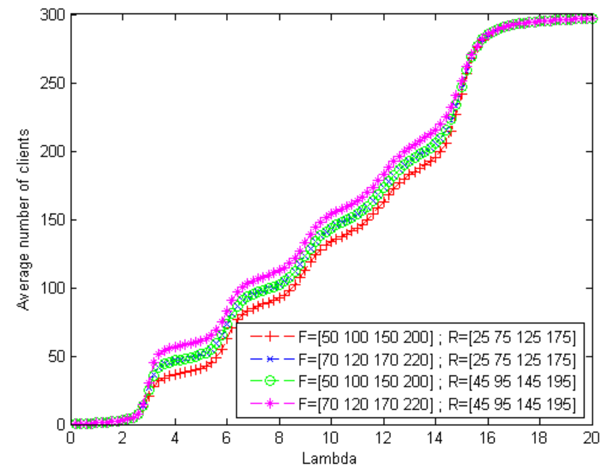

Fig. 2. Average number of clients in the system versus arrival rate (λ) : $\mu = 1$, $K = 5$ and $B = 300$

V. NUMERICAL RESULTS

We implemented the resolution approaches (SCA+GTH, LDQBD and balance equations) then we performed a set of experiments. In this section we first present (in part V-A) a comparison of the resolution approaches in terms of their respective execution time.Then we give (in part V-B) some case studies in which we show the evolution of performance and cost measures according to arrival rate and thresholds values. The observations that we extract from these case studies can be useful to design an optimization method to find thresholds values that minimise the cost.

A. Comparison of the resolution approaches in terms of execution time

Table I shows the execution time of each resolution approach for different values of : the number of levels (K) and the capacity of the system (B). We consider a model in which only one VM is activated/deactivated when we switch from one level to another ($S_1 = 1$, $S_{i+1} = S_i + 1\ \forall i < K$). In the first line we have the smallest instance (K=5, B=300 - markov chain with 521 stats), and in the last line we have the largest one (K=1000, B=75000 - markov chain with 134921 stats). All tests were implemented and performed on a machine with ”Intel i7” CPU and 8GB of RAM. We observe that the method using the balance equations is the fastest among the three for all instances. Indeed, the computation with this approach is made using formulas that contain basic operators. The SCA+GTH approach takes a lot of time for large chains. The reason is the complexity of GTH method that we use to resolve the generated sub chains. The approach based on the LDQBD structure uses matrix inversion. For $K = 1000$ and $B = 75000$, the program returned an error ”out of memory” because of the huge size of the matrix. We can conclude that the balance equations approach is the most appropriate for cloud systems with a very large number of VMs (thousands).

B. Performance and cost measures - Case Studies

In this section we consider a threshold-based queuing system for the Cloud and we present some experiments results. The goal is to observe the evolution of performance measures and the overall cost (that we defined in previous sections) according to arrival rate and thresholds values. In order to have a clear interpretation of results, we assume in all experiments that $\mu = 1$.

Experiment 1 : we analyze the behavior of the average number of clients in the system (figure 2) and the average response time (figure 3) when we vary the arrival rate (λ). We assume that the number of levels $(K) = 5$, the system capacity $(B) = 300$, $S_1 = 3$ and $S_{i+1} = S_i + 3\ \forall i$ (i.e. we activate three VMs when we switch from level i to $i+1$). Y-axis in figure 2 (resp. figure 3) represents the average number of clients in the system (resp. the average response time). We performed experiments for different configurations of thresholds F and R (a curve for each configuration).

In figure 2, the average number of clients increases according to λ. The system is saturated when λ goes beyond 15. The reason is that above $\lambda = 15$, we have $\frac{\lambda}{S_5 * \mu} >= 1$ (i.e. there are more clients arriving than the ability of the system to meet their needs). If we compare the curves of figure 2 (resp. 3), we notice that more the threshold values are significant then more the average number of clients (resp. average response time) is considerable. Indeed, when we choose larger thresholds, the VMs are activated following a larger number of customers in the system, which results in less performance. We also notice that when λ is not high the choice of the thresholds configuration has an impact (the difference between the curves is significant), then when λ grows the configuration has less impact on the values of performance results.

Experiment 2 : we analyze the behavior of the average number of servers (VMs) activations per time unit according to arrival rate (figures 4, 5 and 6). We assume that the number of levels $(K) = 7$, the system capacity $(B) = 250$, $S_1 = 3$ and $S_{i+1} = S_i + 3\ \forall i$ (i.e. we activate three VMs when we switch from level i to $i+1$). We vary in X-axis the arrival rate (λ). We performed tests for different values for thresholds F and R. The notation used to write thresholds in figures is $F = [a : b : c]$ (with $a = F_1$, $c = F_{K-1}$ and $\forall i\ b = F_{i+1} - F_i$). In figure 4, we assume a total overlap between the thresholds of activation F and deactivation R (i.e. $R_1 < F_1 < R_2 < F_2 < R_3 < F_3 < R_4 < F_4 < R_5 < F_5 < R_6 < F_6$). In figure 5 we assume no overlap between activation and deactivation thresholds (i.e.

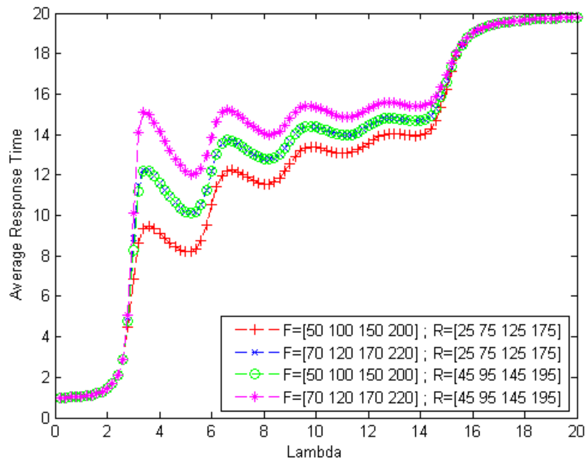

Fig. 3. Average response time in the system versus arrival rate (λ) : $\mu = 1$, $K = 5$ and $B = 300$

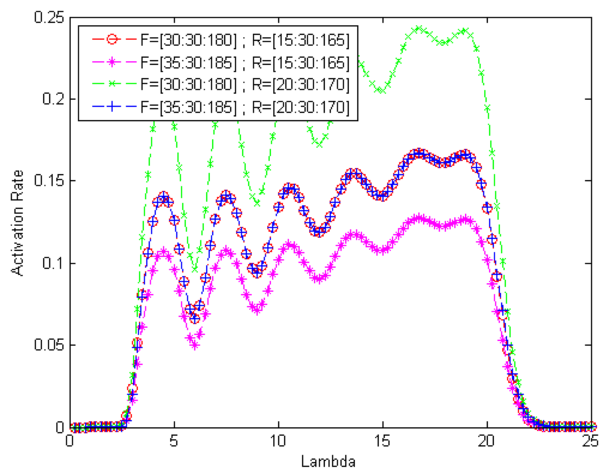

Fig. 4. Activation rate versus arrival rate (λ) : $\mu = 1$, $K = 7$ and $B = 250$ (total overlap between activation and deactivation the thresholds)

$R_1 < R_2 < R_3 < R_4 < R_5 < R_6 < F_1 < F_2 < F_3 < F_4 < F_5 < F_6$). And finally for Figure 6, we have a partial overlap between the activation and deactivation thresholds (i.e. $R_1 < R_2 < R_3 < F_1 < F_2 < F_3 < R_4 < R_5 < R_6 < F_4 < F_5 < F_6$).

Results (figures 4, 5 and 6) show that thresholds values and the difference between activation thresholds and deactivation thresholds influence the shape of the curve of activation rate. We notice that more activation thresholds (F) are far from deactivation thresholds (R) then less there are activations. Indeed, when thresholds are far from each other, we minimize oscillations between levels, which shows that it is interesting to use a hysteresis models for Cloud resources scaling.

Experiment 3 : we analyze the overall cost (figure 7). we assume that the arrival rate (λ) = 4, the number of levels (K) = 7 and system capacity (B) = 500, $S_1 = 3$ and $S_{i+1} = S_i + 3$ $\forall i$ (i.e. we activate three VMs when we switch from level i to $i+1$). We set $R = [45\ 95\ 145\ 195\ 245\ 295]$ and we vary in X-axis values of F. The initial value is $F_{init} =$

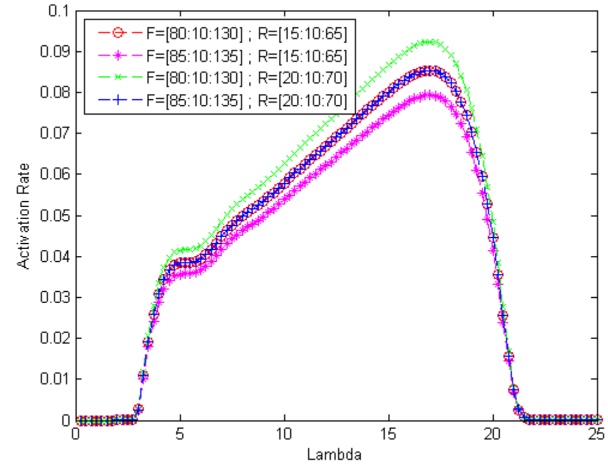

Fig. 5. Activation rate versus arrival rate (λ) : $\mu = 1$, $K = 7$ and $B = 250$ (no overlap between activation and deactivation thresholds)

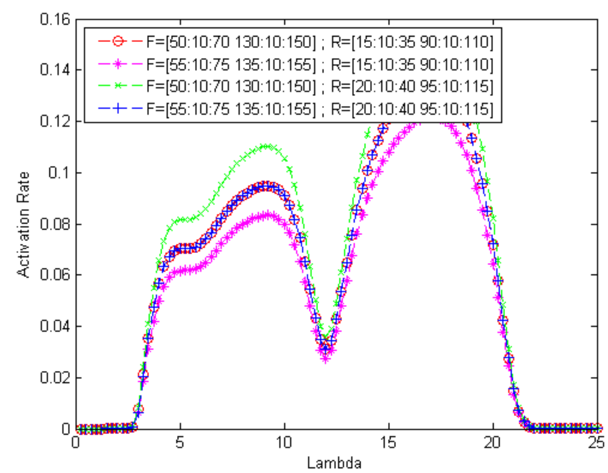

Fig. 6. Activation rate versus arrival rate (λ) : $\mu = 1$, $K = 7$ and $B = 250$ (partial overlap between the activation and deactivation thresholds)

$[50\ 100\ 150\ 200\ 250\ 300]$ (it corresponds to 0 in the x-axis) then we increase the values of F. For example, 10 in the X-axis corresponds to $F = F_{init} + 10 = [50\ 100\ 150\ 200\ 250\ 300] + 10 = [60\ 110\ 160\ 210\ 260\ 310]$. We measure the overall cost for different values of C_A (activationCost in figure 7).

We notice that when F increases the overall cost decreases in a first phase then increases after. The reason is that the formula of the overall cost includes both parameters that increases (for example : the average number of clients) and parameters that decreases (for example : the activation rate) according to F. The thresholds configuration that ensures the minimal cost depends on C_A (activationCost).

Experiment 4 : we compare a model (**conf 1**) in which only one VM is activated/deactivated when we switch from one level to another ($K = 32$, $S_1 = 1$, $S_{i+1} = S_i + 1$ $\forall i < K$, so $S_K = S_{32} = 32$) and two models (**conf2**, **conf3**) in which many VMs are activated/deactivated when we switch from one level to another ($K = 6$, $S_1 = 1$, $S_{i+1} = 2 * S_i$ $\forall i < K$, so

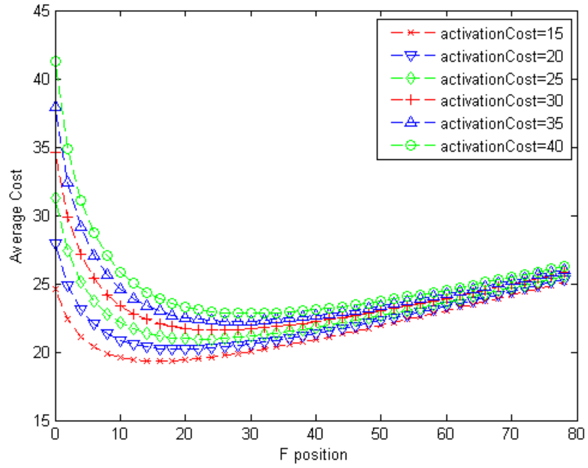

Fig. 7. Global cost versus F values : $\lambda = 4$, $\mu = 1$, $K = 7$, $B = 500$, $R = [45\ 95\ 145\ 195\ 245\ 295]$ and $F_{init} = [50\ 100\ 150\ 200\ 250\ 300]$

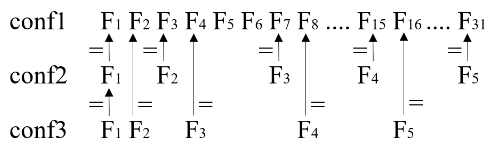

Fig. 8. The values of F for conf1, conf2 (upper bound) and conf3 (lower bound)

$S_K = S_6 = 32$). The overall number of servers (VMs) for the three models is 32 but we have less thresholds in **conf2** and **conf3**. In the experiment, we assume that the system capacity (B) = 400 and we vary the arrivals rate (λ). Thresholds of **conf2** (respectively **conf3**) were chosen so that the associated model is an upper bound (respectively lower bound) for the performances of the model **conf1**. Values of F thresholds are illustrated in figure 8 and we have $R_i = F_i - 5\ \forall i$ for all models.

Figure 9 illustrates the experiment results (the average number of clients of each model for different arrival rates (λ)). The result of **conf2** (resp. **conf3**) is always greater (resp. smaller) to the one-by-one model (**conf1**). **conf2** and **conf3** have less thresholds than **conf1**, so faster to analyse. This idea is useful to analyse large one-by-one models in which we can analyse bounding and smaller models to find a lower and upper bounds for performance rather than analysing the large (so complex) original model.

VI. Conclusion

We analyze a hysteresis queueing system using mathematical methods in order to evaluate the performance and the energy consumption in a cloud system. We consider in this model the general case where the VMs are activated/deactivated by blocks. The system is modeled as a Markov chain which becomes complex to analyze as the state space can grow very quickly. The relevance of this paper is use different mathematical methods : SCA, LDQBD, and closed form from equation evolutions in order to compare their efficiency in terms of accuracy and time computation. We give numerical values for both performance and energy consumption measures, and we

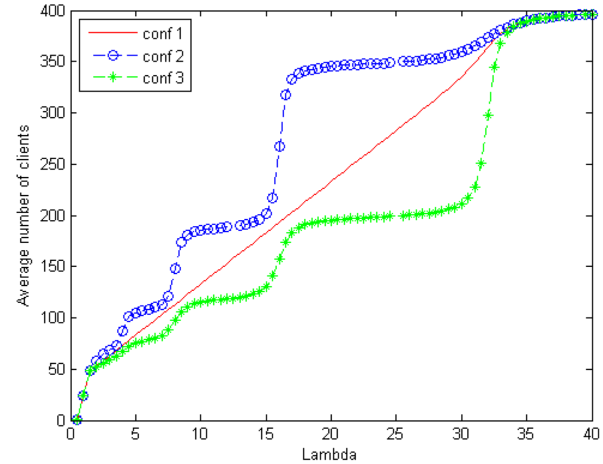

Fig. 9. Average response time in the system versus arrival rate (λ) : $\mu = 1$, $B = 400$, 32 servers - Comparison of the three models -

analyze the impact of the thresholds. One another important contribution of this paper is to suppose fewer constraints on the thresholds in order to analyze the impact on some measures as mean number of activations. We define a global cost for performance and energy consumption in order to propose a trade off between performance and energy consumption. As a future, we propose to develop optimization algorithms in order to obtain the thresholds with minimize the overall cost .

Acknowledgment

The authors would like to thank...

References

[1] J. C. Lui and L. Golubchik, "Stochastic complement analysis of multi-server threshold queues with hysteresis," *Performance Evaluation*, vol. 35, no. 1, pp. 19–48, 1999.

[2] L.-M. Le Ny and B. Tuffin, *A simple analysis of heterogeneous multi-server threshold queues with hysteresis*. IRISA, 2000.

[3] F. Ait-Salaht and H. Castel-Taleb, "The threshold based queueing system with hysteresis for performance analysis of clouds," in *Computer, Information and Telecommunication Systems (CITS), 2015 International Conference on*. IEEE, 2015, pp. 1–5.

[4] W. Stewart, *Introduction to the numerical Solution of Markov Chains*. New Jersey: Princeton University Press, 1995.

[5] M. F. Neuts, *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation, 1981.

[6] B. Gaujal, E. Hyon, and A. Jean-Marie, "Optimal routing in two parallel queues with exponential service times," *Discrete Event Dynamic Systems*, vol. 16, no. 1, pp. 71–107, 2006.

[7] H. Baumann and W. Sandmann, "Numerical solution of level dependent quasi-birth-and-death processes," *Procedia Computer Science*, vol. 1, no. 1, pp. 1561–1569, 2010.