The threshold based queueing system with hysteresis for performance analysis of clouds

Farah Aït-Salaht, Hind Castel-Taleb
INSTITUT TELECOM / Telecom SudParis
SAMOVAR, UMR 5157
9, rue Charles Fourier, 91011 Evry Cedex, France
Email: {farah.ait_salaht, hind.castel}@it-sudparis.eu

***Abstract*—In this paper, we propose to model a node cloud by a threshold based queueing system with hysteresis. The client requests (or jobs) arrive into the buffer, and are executed by service centers or Virtual Machines (VMs). We suppose that virtual machines are activated and deactivated according to the occupation of the queue, in order to model the dynamic behavior of the system. The queueing model is represented by a forward threshold sequence which has different values than the reverse threshold sequence. The forward thresholds represent the numbers of the customers in the queue from which we increase by one the number of VMs. Similarly, the backward thresholds are the values from which we decrease by one the number of VMs. When the forward thresholds are different then the backward thresholds then hysteresis is present. The relevance of hysteresis for the cloud model is to reduce the frequent transitions between activation and deactivation states of the VMs. We will use stochastic comparison method in order to prove and guarantee that the hysteresis model can be bounded by models with the same sequences of forward and backward threshold. The advantage of the bounding models is that the stationary probability distribution can be computed exactly and easily from a mathematical formula. We present some numerical results for the performance measures in order to show that the bounding values provide an accurate coverage for the exact values.**

I. Introduction

Cloud computing is an emerging distributed technology that promises to offer cost-effective scalable on-demand to users without the need for large up-front infrastructure investments. Cloud computing is proved to be profitable for a small scale or large scale business. A cloud computing platform can provide a variety of resources, including infrastructure, software, and services to users in an on-demand fashion.

Virtualization plays a key role in the success of cloud computing as the resources can be used more efficiently. 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, provision resource 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 [8], 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 [5], they develop an analytical model in order to evaluate the performance of cloud centers with 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 [4], 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 high degree of virtualization, and are categorized into three server pools: hot, warm, and cold. 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 queueing model based on queue-dependent virtual machines in order to represent the PM. With this model, virtual machines which are already provisioned are activated and deactivated in order to implement scaling up and down. The queueing model is a multi-server VMs with threshold queues and hysteresis [2]. The multi-server VMs with hysteresis is governed by 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. Moreover, the hysteresis prevents the frequent activation or deactivations of VMs which could be costly in energy consumption. We will use stochastic comparisons in order to bound the hysteresis system by models where the threshold sequence for activating is equal to the sequence for removing the VMs. The advantage of this models is that the Markov chain is very easy to solve as the stationary probability has a simple closed form. We derive bounds for performance measures as blocking probability, and mean number of customers in the buffer. We give some numerical values according to different values of input parameters: arrival rate and the number of VMs (called the degree of virtualization). The paper is organized as follows: next, we describe the cloud system, in section 3, we present the queueing model for the analysis. In section 4, we give the bounding models and we prove using the stochastic comparisons that they represent really

bounds. In the section V, we give numerical results of the performance measures. Finally, achieved results are discussed in the conclusion and comments about further research issues are given.

II. Cloud system description

We assume that the cloud center consists of many PM (Physical Machines), each of which can host a lot of VMs (Virtual Machines), as shown in Fig 1. Incoming requests are routed through a load balancing server to one of the PMs. Different users may share a PM (Physical Machine) using virtualization technique which provides a well defined set of resources (as CPU, RAM, storage). We focus our study on one PM, and we suppose that it is represented by a buffer which contains the requests waiting for service. The VMs provide service for customer requests, and each of them has been allocated with the different computing resource. The buffer has a finite capacity, so an arriving requests can be rejected if it finds the buffer full. This system provides the dynamicity of the service according to the scalability of user requests. In order to have a system able to handle the variability of the traffic intensity, the VM are activated and deactivated according to the system occupancy. In fact, the buffer management is defined by thresholds for the number of customer waiting in the queue, which activate or deactivate the VMs. Clearly, when the number of customers in the queue reaches a threshold, then a new VM is activated, and when it decreases below the threshold, a VM is deactivated. In the next section, we give the queueing model used for the analysis of the performance of the cloud node.

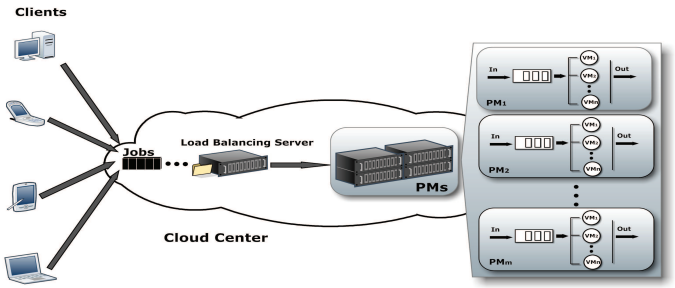


Fig. 1. Cloud center architecture.

III. Model description

We consider a finite buffer capacity with multi-homogeneous servers (VMs). We suppose a K multi-server with thresholds-based queueing system and hysteresis for which a set of forward thresholds $(F_1, F_2, \ldots, F_{K-1})$ and a set of reverse thresholds $(R_1, R_2, \ldots, R_{K-1})$ are defined. We assume that $F_1 < F_2 < \ldots < F_{K-1}$, $R_1 < R_2 < \ldots < R_{K-1}$, and $R_i < F_i, \forall 1 \leq i \leq K-1$. The behavior of this system is as follows. We assume that the first VM is still active in the system. If a customer arrives in the system, and finds F_i $(i = 1, \ldots, K-1)$ customers in the system, then an additional VM will be activated. When a customer leaves the system with R_i $(i = 1, \ldots, K-1)$ customers, then a VM will be removed from the active VMs. We denote by $X(t)$ the model where each state is represented by (x_1, x_2), with x_1 is the number of customers waiting in the system and x_2 is the number of active VMs. We suppose that client request arrivals follow Poisson distribution with rate λ, and servers (or VM) have an exponential service time distribution with mean rate $\mu_i = \mu$ $(i = 1, \ldots, K)$. We suppose that the system has a finite buffer capacity B. With these assumptions, we deduce that the system $X(t)$ is a continuous-time Markov chains defined over the state space A such that:

$$\begin{aligned} A = \{(x_1, x_2) \quad | \quad & 0 \leq x_1 \leq F_1, \text{ if } x_2 = 1; \\ & R_{i-1} < x_1 \leq F_i, \text{ if } x_2 = i \text{ and } 1 < i < K; \\ & R_{K-1} < x_1 \leq B, \text{ if } x_2 = K\}. \end{aligned}$$

The evolution equations of $X(t)$ are defined for $i = 1 \ldots K-1$ as follows:

$$\begin{aligned} (x_1, x_2) \quad \rightarrow \quad & (\min\{B, x_1 + 1\}, x_2), \\ & \text{with rate } \lambda, \\ & \text{if } (x_1 \neq F_i \text{ or } (x_1 = F_i \text{ and } x_2 \neq i)), \\ \rightarrow \quad & (\min\{B, x_1 + 1\}, \min\{K, x_2 + 1\}), \\ & \text{with rate } \lambda, \\ & \text{if } x_1 = F_i \text{ and } x_2 = i, \\ \rightarrow \quad & (\max\{0, x_1 - 1\}, x_2), \\ & \text{with rate } x_2\mu, \\ & \text{if } (x_1 \neq R_i + 1 \text{ or } (x_1 = R_i + 1 \text{ and } x_2 \neq i + 1)) \\ \rightarrow \quad & (\max\{0, x_1 - 1\}, \max\{0, x_2 - 1\}), \\ & \text{with rate } x_2\mu, \\ & \text{if } x_1 = R_i + 1, \text{ and } x_2 = i + 1, \end{aligned}$$

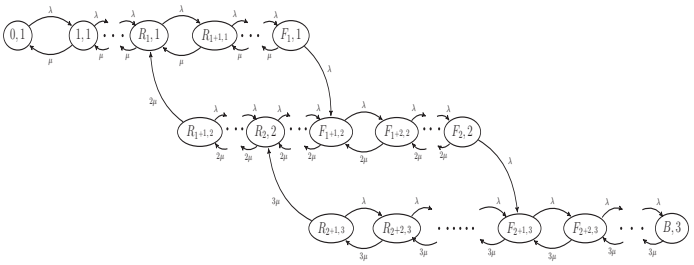


Fig. 2. Example of state transition graph for a three-servers system.

Obviously, this system has been already studied in the literature [2] and [6]. In [2], the authors propose to use Green's function method which is not so easy to apply as the formalism is not intuitive. In [6], the authors use the concept of stochastic complementation to solve the system. They propose to partition the state space in disjoints sets in order to aggregate the Markov chain. Contrary to these two approaches, we propose in this paper to define bounds rather than an exact resolution of the system. The relevance of using bounds is first to reduce the state space size of the system and to derive very simple closed-form of the steady state probability distribution. Next, we propose to define new systems by the modification of the exact system, and we prove that they represent bounds for some performance measures.

IV. Bounding systems

We propose to define bounding systems which are easier to solve. These systems are equivalents to the exact system,

except that the forward and the reverse thresholds are the same. For the upper bound, we take $(F_1, F_2, \ldots, F_{K-1})$ for the forward thresholds and the reverse thresholds. And, for the lower bound, we take $(R_1, R_2, ...R_{K-1})$ for the forward and the reverse thresholds. The behavior of each of these systems is represented by a Markov chain defined on state space $S = \{0, \ldots, B\}$. Moreover, the stationary probability distribution has a very simple closed form. We denote by $Y(t)$ the Markov chain associated to the upper bound (with $(F_1, F_2, \ldots, F_{K-1})$ for the forward and the reverse thresholds). The evolution equation of this model is given as follow:

$$\begin{aligned} x &\rightarrow \min(B,\, x+1), \text{ with rate } \lambda \\ &\rightarrow \max(0,\, x-1), \\ &\quad \text{with rate } i\mu, \text{ if } F_{i-1} < x \leq F_i,\ \forall i = 1 \ldots K-1 \\ &\quad \text{with rate } K\mu, \text{ if } F_{K-1} < x \leq B \end{aligned}$$

where $F_0 = 0$.

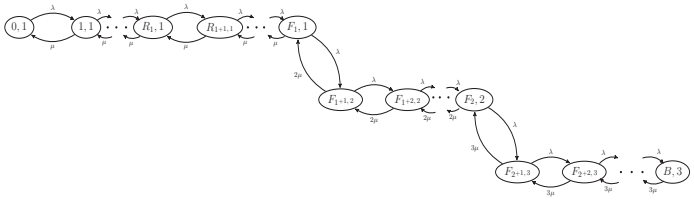


Fig. 3. Example of state transition graph of upper bound model for a three-servers system.

The steady state probability distribution of the system has a known closed-form [3]. Let P_i be the probability to have i customers in the system, and $\rho_i = \frac{\lambda}{i\mu}$. We have the following equations:

$$P_n = \rho_1^n P_0, \text{ if } 1 \leq n \leq F_1 \quad (1)$$

$$P_n = \Pi_{i=1}^{j-1} \rho_i^{F_i - F_{i-1}} \rho^{n - F_{j-1}} P_0, \text{ if } F_{j-1} < n \leq F_j,\ j = 2, \ldots, K-1 \quad (2)$$

$$P_n = \Pi_{i=1}^{K-1} \rho_i^{F_i - F_{i-1}} \rho_K^{n - F_{K-1}} P_0, \text{ if } F_{K-1} < n \leq B \quad (3)$$

where $F_0 = 0$, and P_0 is such that:

$$P_0 = \frac{1 - \rho_1^{F_1+1}}{1 - \rho_1} + \sum_{j=2}^{K-1} \Pi_{i=1}^{j-1} \rho_i^{F_i - F_{i-1}} \frac{\rho_j - \rho_j^{F_j - F_{j-1}+1}}{1 - \rho_j} + \Pi_{i=1}^{K-1} \rho_i^{F_i - F_{i-1}} \frac{\rho_K - \rho_K^{B - F_{K-1}+1}}{1 - \rho_K} \quad (4)$$

In the same way, we defined by $Z(t)$ the Markov chain which represents the lower bound (with $(R_1, R_2, \ldots, R_{K-1})$ for the forward and the reverse thresholds). In this case, the above equations (evolution equation and equations (1)-(4)) are also available by changing the sequence $F_{i,\, i=1 \ldots K-1}$, by the sequence $R_{i,\, i=1 \ldots K-1}$. Next, we will prove that $Y(t)$ (resp. $Z(t)$) is a stochastic upper bound (resp. lower bound) for $X(t)$. The bounding systems provide bounds for performance measures as mean number of customers, mean response times, and blocking probabilities. Now, we give some definitions and theorems about stochastic ordering theory.

A. Stochastic ordering theory

We give some theorems and definitions about stochastic orderings [7] used in this paper. We consider a discrete, and countable state space A, endowed by a binary relation $\preceq$ which is at least a preorder [7]. As an example, on the state space $A = \mathbb{R}^n$, component-wise order is a partial order, and on $A = \mathbb{R}$, $\leq$ is a total order. In the sequel, $\preceq$ denotes at least a preorder on A. We consider two independent random variables X and Y defined on A. The most known stochastic ordering is the strong stochastic ordering, and it is denoted by $\preceq_{st}$. It could be defined using increasing functions as follows [7].

Definition 1: $X \preceq_{st} Y \iff \mathbb{E}f(X) \leq \mathbb{E}f(Y)$, for all non decreasing functions $f : A \rightarrow \mathbb{R}^+$ whenever expectations exist.

We can also compare stochastic processes. Let $\{X(t), t \geq 0\}$ and $\{Y(t), t \geq 0\}$ be stochastic processes defined on A.

Definition 2: We say that $\{X(t), t \geq 0\} \preceq_{st} \{Y(t), t \geq 0\}$, if $X(t) \preceq_{st} Y(t), \forall t \geq 0$

When the processes are defined on different states spaces we can compare them on a common state space using mapping functions. Let $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$) defined on A (resp. B), g (resp. h) be a many to one mapping from A to S, (resp. $B \rightarrow S$). Next, we compare the mapping of the process $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$) by the mapping function g (resp. h), which means $g(X(t))$ (resp. $h(Y(t))$), on the common state space S.

The stochastic comparisons of processes by mapping functions is defined as follows [1]:

Definition 3: We say that $\{g(X(t)), t \geq 0\} \preceq_{st} \{h(Y(t)), t \geq 0\}$, if $g(X(t)) \preceq_{st} h(Y(t)), \forall t \geq 0$

We can use the coupling method for the stochastic comparison of the processes. For the $\preceq_{st}$ ordering, the coupling method can be used for the stochastic comparison of CTMCs. As presented in [1], it remains us to define two CTMCs: $\{\widehat{X}(t),\ t \geq 0\}$ and $\{\widehat{Y}(t),\ t \geq 0\}$ governed by the same infinitesimal generator matrix respectively as $\{X(t), t \geq 0\}$, and $\{Y(t), t \geq 0\}$, representing different realizations of these processes with different initial conditions. The following theorem establishes the $\preceq_{st}$-comparison using the coupling [1]:

Theorem 1:

$$\{g(X(t)), t \geq 0\} \preceq_{st} \{h(Y(t)), t \geq 0\} \quad (5)$$

if there exists the coupling $\{(\widehat{X}(t), \widehat{Y}(t)),\ t \geq 0\}$ such that:

$$g(\widehat{X}(0)) \preceq h(\widehat{Y}(0)) \Rightarrow g(\widehat{X}(t)) \preceq h(\widehat{Y}(t)),\ \forall t > 0 \quad (6)$$

B. Stochastic comparison proofs

In this section, we prove that the system with different thresholds for increasing and decreasing of the number of virtual machines (denoted $X(t)$) is a stochastic lower bound for $Y(t)$, and a stochastic upper bound for $Z(t)$. As $X(t)$ and $Y(t)$ represent Markov chains defined on different state spaces, then we will compare them by a mapping function on a common state space. We define the many to one mapping function $g : A \rightarrow S$, such that $g(x) = x_1$, and in the state

space S, we use the total order $\leq$. We apply the stochastic ordering theory presented before by considering the total order $\leq$ as a particular case of preorder $\preceq$. Next, we use the stochastic ordering $\leq_{st}$ instead of $\preceq_{st}$, and we will prove that $g(X(t)) \leq_{st} Y(t)$. We have the following theorem:

Theorem 2: If $X(t)$, $Y(t)$, and $Z(t)$ represent the systems defined previously, then we have:

1) $g(X(0)) \leq_{st} Y(0) \Rightarrow g(X(t)) \leq_{st} Y(t), \forall t > 0$
2) $Z(0) \leq_{st} g(X(0)) \Rightarrow Z(t) \leq_{st} g(X(t)), \forall t > 0$

Proof: For the proof, we apply the theorem 1, in order to use the coupling by the mapping function for the stochastic comparison. In our case, we have $X(t)$ defined on A, g a mapping function $A \rightarrow S$, $Y(t)$ defined on S, and h is the identity function. The proof is done by induction: we consider $X(t) = x$ and $Y(t) = y$, such that $g(x) \leq y$ and we prove that for any event, at time $t + dt$, if $X(t + dt) = x'$, and $Y(t + dt) = y'$, then $g(x') \leq y'$. As the transitions for the processes can be only increasing by one or decreasing by one, then for the comparison we take only states such that $g(x) = y$ (or $x_1 = y$, as $g(x) = x_1$) because for only these states, the order may be not verified after transitions due to arrivals or services. In fact, for states x and y such that $g(x) < y$, then we are sure that with transitions causing an increasing or a decreasing by 1, the process still verify $g(x) \leq y$.

We consider now the two events for the proof of the stochastic comparison by the mapping function g:

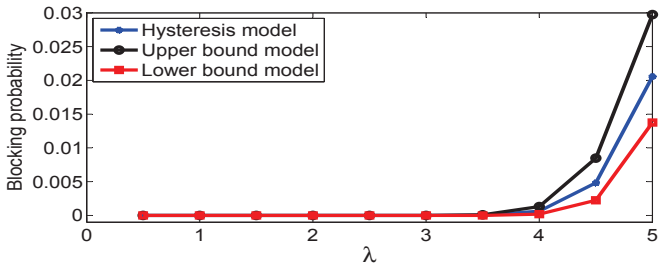
- Arrivals: if we have a transition from x to x' such that $x'_1 = x_1 + 1$, then we are sure that we can have also a transition from y to y' such that $y' = y + 1$. Because the arrival rate λ is the same in the two systems. So, as $x_1 = y$, then $x_1 + 1 = y + 1$, and we deduce that $g(x') \leq y'$.
- Services: if we have a transition from y to y' such that $y' = y - 1$, then we are sure that we have also a transition from x to x' such that $g(x') = x_1 - 1 = y - 1$. As the threshold for deactivation is lower in $X(t)$ than in $Y(t)$, then the rate for decreasing in $Y(t)$ is lower than in $X(t)$. So $g(x') \leq y'$.

So we deduce that for any events, we have: $g(X(t+dt)) \leq Y(t + dt)$, and from theorem 1, we deduce that in theorem 2, equation 1 is verified. For the lower bound, the proof is similar, except the inequalities are reversed. For a state $X(t) = (x_1, x_2)$, and a state $Z(t) = z$, such that $g(x) = z$, which corresponds to $x_1 = z$, then at time $t + dt$, we consider the two cases: arrivals, and services. For the arrivals, as the rates are the same, then the order is kept. For the services, then as the threshold for VM activation is lower in $Z(t)$, then $X(t)$, then the service rate is upper in $Z(t)$ than in $X(t)$, so $Z(t)$ is really a lower bound, and we deduce that in theorem 2, equation 2 is verified. ■

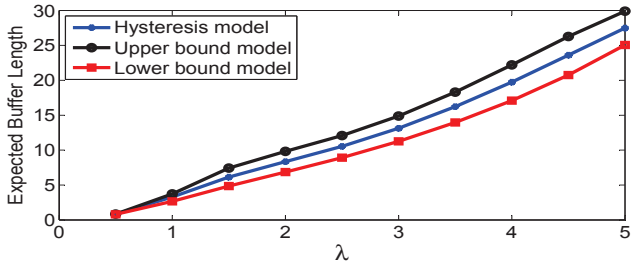
V. Numerical Examples

In this section, we present numerical examples which illustrate the behavior of the threshold-based queueing system with hysteresis and the bounding models presented in the paper. We compare these three models by computing some performance measures. We also present the activation and deactivation rate of the servers, associated with each model. Indeed, it is important to mention that the threshold-based queueing systems with hysteresis is introduced in order to avoid costly and frequent oscillations around the forward threshold. We note here that the hysteresis model is solved using the approach presented in [6]. We begin with a small example. We consider a system with number of servers, K, equal to 5. The forward and reverse threshold vectors are set to $F = (8, 12, 20, 30)$ and $R = (5, 9, 15, 23)$, respectively. The buffer size is $B = 40$, the service rate μ is set to 1.1, and the average arrival rate λ is varied from 0.5 to 5. For this example, we note that the length of the state space of the hysteresis model is 59 states while the bounding models are defined on 41 states. The solution of all models (i.e., hysteresis model, upper and lower bound models) is carried out using MATLAB.

We depict in Figures 4 and 5 the blocking probability, the expected buffer length and the expected departure computed for different values of arrival rate.



(a) Blocking probability



(b) Expected buffer length

Fig. 4. QoS parameters versus arrival rate.

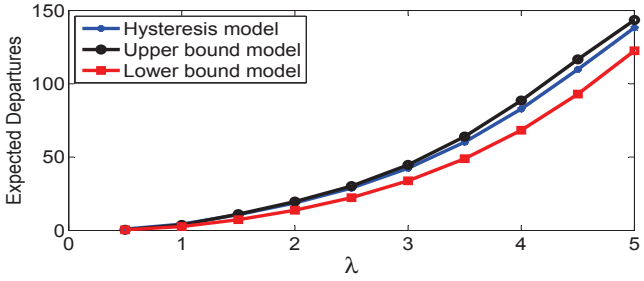


Fig. 5. Expected departure versus arrival rate.

From these figures, it is clear that the computed performance measures (blocking probability, mean buffer length and mean throughput) grow as the arrival rate increases. Moreover, we can observe that the bounding model really compute

bounds and gives a good coverage on performance measures of the hysteresis model. We note also that the average time required to resolve the original model is 0.0084 seconds while the lower and upper bound models require an average time of 1.5516×10^{-4} seconds. We present now an other example and study the activation and deactivation rate of servers in the different models. We consider queue with hysteresis with Poisson arrivals, where $K = 50$, $B = 5000$ and $\mu = 81$. The forward threshold is set to $F = (150 : 100 : 4950)$ and the reverse threshold vector is taken equal to $R = F - 100$. We vary the value of arrival rate from 100 to 600. We note that for this example, the length of the state space of the hysteresis model is 9901 states while the bounding models are defined on 5001 states. At this step, we can already make a first comment on the state space size of the hysteresis model. Indeed, we can notice that it is significantly larger than the state space size of the bounding models. We give respectively in Table I and Table II the activation rate of servers and deactivation rate of servers for the different values of arrival rate.

λ	Original model	U.B. model	L.B. model
100	0.23225	8.9773	8.9775
200	1.9802	16.6032	16.6032
300	2.9703	15.6474	15.6474
400	3.9604	7.7175	7.7175
500	4.9505	11.3768	11.3768
600	5.9406	18.2155	18.2155

TABLE I. ACTIVATION RATE OF SERVERS.

λ	Original model	U.B. model	L.B. model
100	1.7921	23.5600	23.5604
200	2.7822	24.5100	24.5100
300	3.7723	18.2549	18.2549
400	4.7624	8.5868	8.58678
500	5.7525	14.3824	14.3824
600	6.7426	21.2290	21.229

TABLE II. DEACTIVATION RATE OF SERVERS.

We note that the rates of activation and deactivation of the servers (VMs) in bounding models are significantly greater than those of the hysteresis model. Indeed, the use of threshold system with hysteresis allows us to avoid to switch too much, and stay more longer with an active server and therefore minimize the cost of activation and deactivation of the servers. As a result, we clearly observe the interest that may represent the threshold system with hysteresis for the modeling in cloud system. We give also some results on performance measures. In Table III, we present results on average buffer length while Table IV is devoted to the average departure. This experiment yields to the same conclusions as above. Indeed, we can see that the results provided by the bounding models are accurate and gives a good coverage on the results of the hysteresis model. Concerning the mean execution time, the computation of the upper (resp. lower) bound model takes 0.0775 second which is significantly lower than 65.1068 seconds needed to solve the original model.

VI. CONCLUSION

We propose in this paper a queue-dependent multiserver VMs with hysteresis in order to model the behavior of a PM in a cloud node. The relevance of this model is to represent the dynamicity of the resource according to the queue occupation, and with the hysteresis to reduce the cost due to activation/deactivation of the VMs. However, this system could be difficult to analyse when the number of VMs increases, so we propose to use bounding techniques in order to derive bounds for the performance measures. From the numerical results, we have seen clearly the accuracy of bounds. As a future work, we expect to investigate a more general system by considering batch arrivals and heterogeneous servers (VMs).

λ	Original model	U.B. model	L.B. model
100	97.3498	147.3500	47.3506
200	200.3880	250.3880	150.3880
300	308.2100	358.2100	258.2100
400	438.0840	488.0840	388.0840
500	577.5930	627.5930	527.5930
600	695.4770	745.4770	645.4770

TABLE III. EXPECTED BUFFER LENGTH VERSUS ARRIVAL RATE.

λ	Original model	U.B. model	L.B. model
100	9834.98	14735.0	4735.06
200	40277.60	50077.6	30077.60
300	92763.00	107463.0	77463.00
400	175633.00	195233.0	155233.00
500	289297.00	313797.0	263797.00
600	417886.00	447286.0	387286.00

TABLE IV. EXPECTED DEPARTURE VERSUS ARRIVAL RATE.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Doisy. A coupling technique for stochastic comparison of functions of markov processes. *JAMDS*, 4(1):39–64, 2000.

[2] O. C. Ibe and J. Keilson. Multi-server threshold queues with hysteresis. *Perform. Eval.*, 21(3):185–213, 1995.

[3] M. Jain. Finite capacity m/m/r queueing system with queue-dependent servers. *Computers & Mathematics with Applications*, 50(1-2):187 – 199, 2005.

[4] H. Khazaei, J. V. Misic, and V. B. Misic. A fine-grained performance model of cloud computing centers. *IEEE Trans. Parallel Distrib. Syst.*, 24(11):2138–2147, 2013.

[5] H. Khazaei, J. V. Misic, and V. B. Misic. Performance of cloud centers with high degree of virtualization under batch task arrivals. *IEEE Trans. Parallel Distrib. Syst.*, 24(12):2429–2438, 2013.

[6] J. C. S. Lui and L. Golubchik. Stochastic complement analysis of multi-server threshold queues with hysteresis. *Performance Evaluation*, 35:19–48, 1999.

[7] A. Müller and D. Stoyan. *Comparison methods for stochastic models and risks*. Wiley series in probability and statistics. J. Wiley & sons, 2002.

[8] P. Wang, W. Huang, and C. A. Varela. Impact of virtual machine granularity on cloud computing workloads performance. In *Proceedings of the 2010 11th IEEE/ACM International Conference on Grid Computing*, pages 393–400, 2010.