Modeling Decision Making to Control the Allocation of Virtual Machines in a Cloud Computing System with Reserve Machines

Abdellah Ouammou, Abdellah Zaaloul, Mohamed Hanini, and Abdelghani Bentahar

Abstract—**Due to the complex and dynamic nature of virtual machine allocation in the cloud, it is difficult to manage and control the resources or to choose the best allocation of these resources. In this paper, we propose an optimal model for allocating computing resources to assess the best management of system resources where a group of physical machines is defined as "reserves". The controller activates them one by one when the system has a high number of tasks. The objective is to maximize the reward of the cloud computing system. This reward is calculated based on the energy and execution time of each customer and the characteristics of the system. Finding the best allocation for such a complex system is a challenge. For this, we used a heuristic algorithm and dynamic programming approach. The results analysis showed the advantage of using our model to control the use of reserve machines to get high quality of service and low energy consumption.**

Index Terms—**Cloud computing, Dynamic programming, Resources management, Modeling decision, Reserve machines.**

I. Introduction

CLOUD computing is a new paradigm for the present and future generation of computing and technology as it provides appealing services such as flexible services, e-services and pay-as-use schemes. A cloud computing environment can provide a wide variety of resources, including infrastructure, software and services, to on-demand users. To have access to these resources, the cloud provider, provides the requested resources according to their availabilities and permits the user to access these resources for a specified period and a specific mission. In contrast to classical approaches "own and use", cloud computing services reduce the costs of purchasing and managing cloud infrastructure for cloud customers and allow them to control IT resources in real-time according to their needs[1]. Besides, dynamically, cloud resources constitute a heterogeneous and complicated system and the user's resource demand can change at any time. Managing resources in such an environment is a difficult task[2], but it is an essential function of any dynamic system. It demands sophisticated policies and decisions to manage complex multi-dimensional objectives such as CPU, memory and network bandwidth. Therefore, for the correct use of resources, efficient resource management can control the system resources and try to find an optimal balance between the QoS requirement and the quantity of energy consumption [3].

The energy consumption in the cloud data centers has grown exponentially in the IT sector in recent years. Data centres, as a principal component of current information and communication technology, have grown at an unprecedented rate as IT developers, including Microsoft, IBM, Google and similar large companies, have developed data centres to support their cloud and grid computing services[4]. These data centers are loaded with thousands of servers and switches that consume huge amounts of energy, increasing operating costs and carbon dioxide emissions into the environment. On the other hand, cooling equipment must be used to manage the heat released by these data centres, which also consume energy [5], [6]. Hence, the globalization of cloud computing requires the establishment of large-scale data centers that contain thousands of computing nodes, which in turn results in excessive energy consumption and negative environmental impacts [7], [8].

So, many of the operators have tried to find some algorithms that can minimize energy consumption. The available algorithms in resource management can be divided into two groups, depending on whether resource requests are dynamic or not, and most of the dynamic approaches are heuristic and lack theoretical performance guarantees.

Therefore, the realistic way to reduce significantly the energy consumption of a physical machine is to turn on the machine when this can be justified by the terms of the request. This is the policy that we propose to improve.

The idea in this paper is to consider a group of physical machines (PM) as "reserve". The reserve status is controlled by the number of clients in the system on which they are turned on when the number of tasks in the system becomes large enough. The question, of course, is how many reserves we have to use and when we turn on a new PM. The answer depends on the demand parameters and the cost function to minimize, and this function should include both a performance cost (e.g. execution time) and the energy consumed by the physical machines of the server. Obviously, there is a compromise between the two parameters, as performance is improved by increasing the number of physical machines powered, while power consumption is improved by increasing the number of physical machines powered off. That trade-off can be evaluated by analyzing

Manuscript received September 25, 2020; revised January 20, 2021.

Abdellah Ouammou is a PhD student at Hassan First University, Faculty of Sciences and Techniques, with Computer, Networks, Mobility and Modeling laboratory: IR2M, Settat, Morocco. E-mail: a.ouammou@uhp.ac.ma.

Abdellah Zaaloul is a professor at the department of Mathematics Ibn Zohr University, is with Complex Systems and Interactions Research Team, F.S.J.E.S, Ait melloul, Morocco. E-mail: a.zaaloul@uiz.ac.ma.

Mohamed Hanini is a professor at the department of Mathematics and Computer science, Hassan First University, Faculty of Sciences and Techniques, is with Computer, Networks, Mobility and Modeling laboratory: IR2M, Settat, Morocco. E-mail: mohamed.hanini@uhp.ac.ma.

Abdelghani Bentahar a professor at the department of Mathematics and Computer science, Hassan First University, Faculty of Sciences and Techniques, is with Computer, Networks, Mobility and Modeling laboratory: IR2M, Settat, Morocco. E-mail: bentahara@yahoo.fr.

and solving a Semi Markov Decision Process (SMDP) model of the system. The solution provides average performance and average consumption measurements. This makes it easy to calculate the cost function and minimize it by maximizing the total expected reward of the cloud computing system. However, the resolution of the SMDP problem on a large scale depends on the dimensionality constraint. Therefore, we make better use of the problem's characteristic and propose a dynamic resource management method. Since the system model contains a controller for allocation demand, the optimal policy of the requests is dynamically achieved by applying the value iteration algorithm (VI) [9], and it is complicated for a human to predict the best system configuration (assigning each client as long as the reward is maximized) without applying an algorithm as proposed in this document. As an implementation of our algorithm, we studied the optimal choice of the controller in order to control the performance of the system and to minimize energy consumed, and also to answer the question when we turn on a new physical machine.

The content of this document is organized as follows. It begins with a literature review in Section 2, followed by the development of the System Model in Section 3. Section 4 is reserved for the formulation of the system using the semi-markov decision process. The reward of the system is calculated and solved in section 5, followed by a numerical analysis of the performance in section 6. Finally, in section 7 a conclusion and future work are presented.

II. Related work

In the literature, there are some recent efforts devoted to managing resources in a cloud computing environment [10]. Gandhi et al. [11] analyse an ON-OFF policy in which a server is switched on if there is a pending task and is switched off if there are no pending tasks. The work in [12] proposed a novel fitness function to maximize the energy efficiency (i.e., profit) of cloud data centers while minimizing the overall energy consumption with reserved VM requests. Several studies combining the quality of service analysis and cloud energy consumption were proposed recently. Ouammou et al [13] have suggested an energy-efficient load balancing algorithm. They used a policy to determine the upper and lower thresholds for each physical machine. This approach has managed the number of migrations. But the main problem with this approach is that they used fixed values of the upper and lower thresholds. In [14], the authors have presented a mathematical queuing model to study the performance of multi-core virtual machines hosting SaaS cloud services. The authors in [15] maximize the use of resources by using two heuristics of consolidation noted energy-conscious taking into account both active and inactive energy consumption; the suggested model can be used to estimate the number of multi-core virtual machine instances needed to reach the QoS parameters under any additional workload. other recent works in the research area have addressed markov chain theory to estimate the efficiency of queues in the cloud servers. Hanini et al [16] introduced a continuous markov chain (CTMC) approach to manage the use of virtual machines with a controlled system workload. They examined the proposed model on the base of mathematical analysis of QoS parameters of the system. Also, there is some published works on the resources managements as [17], [18] and [19] that formulated their propositions using the semi markov decision process to manage the optimal action that will be taken under some conditions.The authors of [20] are interested in proposing an algorithm to schedule the client tasks to the resources of a data center in cloud computing. Their proposed solution, called OSACO algorithm, is concentrated on client QoS requirements and aimed to reduce the energy, cost and time.

In general, the majority of dynamic approaches are focused on heuristics, and consequently do not have sufficient theoretical guarantees of performance and, to our knowledge, a dynamic policy of the type proposed here has not been studied before. and perhaps the model closest to the one presented here is that examined by Isi mitarini in [21], where the author studied the problem of energy use when servers are powered up and down in a frame of the data centers. This means that after the setup time, all servers start running simultaneously even if the system does not require all servers. In [22] focus on an M/M/k queue system with setup costs M/M/k/Setup.Servers are divided in two types: in run mode and in sleep mode as a reserve when there is no work to be done. This gives us reason to reconsider the case where the configuration time of each server is considered different.

In this paper, we consider resources management as a stochastic optimization problem. Our purpose in this work can be summarized as follows:

- To propose a mechanism that helps to manage the resources by improving QoS and minimizing energy consumption based on a "PM reserve" approach.
- To formulate the dynamic virtual machine management problem as a markov decision process problem, in order to maximize the reward of the cloud computing system.

Based on the advanced algorithm, we show that our approach can significantly reduce the evaluation time of VM allocation solutions in each scalable iteration.

III. System Model

A. Problem statement

The studied system is a server containing M physical machines, of which Q are designated as Main Physical Machines (MPM) and $(M - Q)$ considered as Reserves Physical Machines (RPM) $0 \leq Q \leq M$. Considering that each PM can host many Virtual Machines (VM). In order to minimize the energy consumption even when we serve many clients at the same time, we optimally consolidate VMs in a minimum number of PMs. Clients arrive at the server in a Poisson process λ_p and find a controller in front of them that takes them through a MPM until the system gets n clients (Jobs) (n is called a "controller number"). At that time, the controller sends a feedback to the administrator to informe him to reduce the rate of clients sent to the servers from λ_p to λ_s, and then the controller must decide whether to send it to one of the MPMs that contain customers or to any of the RPM, based on system conditions. The service times are independent and distributed exponentially with the following parameters $\mu_1, \mu_2, ..., \mu_M$, such that $\mu_i = \mu_p$ with $i = 1, ..., Q$ and $\mu_j = \mu_s$ with $j = Q+1, ..., M$ with $Q < M$.

We notice that when the client arrives, and at least one of the MPM is available to host this client under some

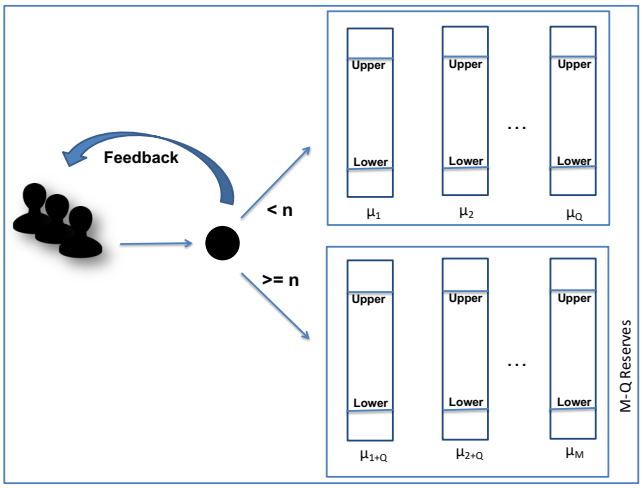

Fig. 1. Illustration of the proposed model

conditions related to the workload in these MPMs, the controller send it directly to one of the MPM. In the other case, when all MPM are noted empties, the controller can either send the customer to any of the MPMs. In this scenario, the system controller chooses to allocate multiple customers to a limited number of MPM in an attempt to reduce the total energy consumed. Alternatively, the controller can use a new physical machine from the reserves, thus increasing the consumption of energy and in return improving the quality of service. To help the controller to decide what decision should be made for the benefits of the system, a decision policy is proposed in this paper. For this, two thresholds are considered for each PM in order to maintain capacity below the upper threshold and above the lower, the details of this idea have been described in [13].So, if a client comes into the system, a VM describes the request of this client (job) and it has to be mapped to one of the available MPM as long as the capacity of this PM after allocation will not exceed the Upper threshold, and if this client has no place to mapped to any MPM, then the controller decides to turn on a new physical machine among the reserves and so on. Figure 1 illustrates the proposed model.

For our analysis, we assumed that the arrival of the clients follows a poisson process and the service times are exponentially distributed. These assumptions has been usually used in the literature and can provide adequate approximation of real systems such cloud computing environment [23] [24] [25][26].

B. Mathematical formulation

In this sub-section, using integer mixed linear programming, we will mathematically formulate the model that we have proposed. Table 1 presents the notation used in this model.
The objective is to reduce as much as possible the number of machines used from reserves.

The objective function is expressed as follows:

$$\min_{z_j} \sum_{j=1}^{M} z_j \tag{1}$$

where z_j are integer variables defined by the following equation

$$\forall j \in [1, M] \qquad z_j = \begin{cases} 1, & \text{The } j^{th} \text{ PM is used} \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

This optimization is based on linear constrains; each VM can be hosted just on a PM, and each PM is able to host VMs not only based on the amount of remaining capacity but also on the impact of hosting them on the performance of the system [27]. These constraints are modelled as follows:

TABLE I
LIST OF IMPORTANT NOTATIONS

Notation	Meaning
M	Number of all physical machines in the server
Q	Number of the PMs designated as Main PM
K	Maximal number of the VMs that the system can support
l	Number of the VMs that each PM can support
n	Number of clients used to decide when the system can turn on a new PM
x_k^p	Number of service requests that have been allocated in the k PM among the MPM
x_k^r	Number of service requests that have been allocated in the k PM among the RPM
λ_p	Arrival rate of a VM in MPM
μ_p	Service rate of a VM in MPM
λ_s	Arrival rate of a VM in RPM
μ_s	Service rate of a VM in RPM
C_{ij}	Capacity of the VM i located in j^{th} PM
$Upper_j$	Upper threshold for PM j
$Lower_j$	Down threshold for PM j

- C_j : The total capacity of all VMs located in a PM_d must be between its $Upper$ and $Lower$ thresholds.

$$Lower_j z_j \leq C_j = \sum_{i=1}^{l} C_{ij} V_{ij} \leq Upper_j z_j, \forall j \in [1, M], \tag{3}$$

where V_{ij} are decision variables defined by

$$V_{ij} = \begin{cases} 1, & \begin{cases} \text{The VM } i \text{ is mapped to } j^{th} PM \\ \forall j \in [1, M], \quad \forall i \in [1, l] \end{cases} \\ 0, & \text{Otherwise} \end{cases} \tag{4}$$

Note that if for a given i and j $V_{ij} = 1$ then $z_j = 1$

- The number of VMs that exist in a physical machines j, as long as the sum of their capacities do not exceed the upper threshold, must verify :

$$n_j = \sum_{i=1}^{l} V_{ij} \leq l, \quad \forall j \in [1, M] \tag{5}$$

- The number of VMs that exist in all MPM, as long as the sum of their capacities does not exceed the upper threshold of each PM, must verify:

$$n = \sum_{j=1}^{Q} n_j \leq l \times Q \tag{6}$$

- No virtual machine can exist in two physical machines.

$$\sum_{j=1}^{M} V_{ij} \leq 1, \quad \forall i \in [1, l], \tag{7}$$

We can summarize these by adding the objective function to all the constraints in the follows set of equations:

$$\min_{z_j} \sum_{j=1}^{M} z_j \quad \textbf{subject to}$$
$$\begin{cases} Lower_j z_j \ \leq \sum_{i=1}^{l} C_{ij} V_{ij} \leq Upper_j z_j \quad \forall j \in [1, M], \\ \sum_{j=1}^{M} V_{ij} \leq 1, \quad \forall i \in [1, l]. \\ n_j = \sum_{i=1}^{l} V_{ij} \leq l, \quad \forall j \in [1, M]. \\ n = \sum_{j=1}^{Q} n_j \leq l \times Q. \end{cases} \tag{8}$$

The system must always satisfy the constraints defined in the above equation. However, taking into account the dynamic character of the studied environment, it is also necessary that the chosen actions at any particular time be designed to minimise the cost caused by the operation of the system. For this, we will consider an expected reward to be maximized in the long run of the system. This maximization makes it possible to determine the optimal policy for deciding what action to take, this can be done using a markov decision process [28], [29]. The resolution of such a problem will require an approach from the theory of dynamic programming. This is the purpose of the following in this paper.

IV. System formulation using Semi Markov Decision process

A. System states

Let S be the set of states that present the current requests with a variable number of VMs available in the system :

$$S = \{s | s = (x^p, x^r, Q, \epsilon) = (x_1^p, x_2^p, ...x_Q^p, x_{Q+1}^r, ..., x_M^r, Q, \epsilon)\}$$

Both x_k^p and x_k^r are described in Table 1.Let Υ represents the set of events such as $\epsilon \in \Upsilon = \{A_p, A_r, D\}$ where A_p is the arrival of the customer when at least one of the MPMs is available, A_r is the arrival of the customer when all the MPMs already have at least one customer, and D denotes the leaving of a request allocated to one of the MPMs or RPMs.

B. Set of actions

We note that in our analysis model, a number of actions are possible to be taken as part of the set of actions A , i.e:

$$A = \{-1, 0, R, P_r\}$$

At the occurrence of an event, the system controller takes an action $a(s)$ from the set of actions A under system conditions.

$$a(s) = \begin{cases} -1 & \epsilon = D \in \{D_p, D_r\} \\ 0 \text{ or } R & \epsilon = A_r \\ P_r & \epsilon = A_p \end{cases} \tag{9}$$

When the system receives a new client, the following actions are determined by the system.

- $a(s) = P_r$: means that the request was accepted directly into a MPM before the system had n clients.
- $a(s) = R$: when the request became after the Q MPMs have already been used by at least one client, and that client has been accepted.
- $a(s) = -1$: represents the instances when the service request is terminated and the client leaves the system, and no service is requested. So, only the information of the number of virtual machines available in the system must be updated.
- $a(s) = 0$: when the request is rejected.

C. Decision time

When a state s and an action a are given, then the service time to get the next state is indicated by $\tau(s, a)$. That is determined by the average rate $\gamma(s, a)$ of the events, which is the inverse of the time interval between two decisions $\tau(s, a) = \gamma(s, a)^{-1}$.

So the average rate $\gamma(s, a)$ of the events is the sum of the rate of all system that can be expressed by:

$\gamma(s, a) =$

$$\begin{cases} Q\lambda_p + m\lambda_s + \sum_{i=1}^{Q} \mu_p + \sum_{i=Q+1}^{M} \mu_s & a(s) = 0, \ \ \epsilon = A_r \\ Q\lambda_p + m\lambda_s + \sum_{i=1}^{Q} z_i\mu_p + \sum_{i=Q+1}^{M} z_i\mu_s & a(s) = -1, \ \ \epsilon = D \\ Q\lambda_p + m\lambda_s + \sum_{i=1}^{Q} z_i\mu_p + \sum_{i=Q+1}^{M} z_i\mu_s + \mu_s & \begin{cases} a(s) = R, \epsilon = A_r, \\ \sum_{i=1}^{Q} V_{ij} > n \end{cases} \\ Q\lambda_p + m\lambda_s + \sum_{i=1}^{Q} z_i\mu_p + \mu_p & \begin{cases} a(s) = Pr, \epsilon = A_p \\ \sum_{i=1}^{Q} V_{ij} \leq n \end{cases} \end{cases} \tag{10}$$

In Eq (10), if the system does not accept any requests, there is $\sum_{i=1}^{Q} z_i$ services that exist in the system. Then, the service rate of the global system is $\sum_{i=1}^{Q} z_i\mu_p$. If a request is admitted, then there are $\sum_{i=1}^{Q} z_i + 1$ services in the system, and if the number of clients does not reach n clients, so the leaving rate is $\sum_{i=1}^{Q} z_i\mu_p + \mu_p$. Similarly, when the system have already received n clients and starts to use the RPM; then the leaving rate is $\sum_{i=1}^{Q} z_i\mu_p + \sum_{i=Q+1}^{M} z_i\mu_s + \mu_s$

D. Transition Probability

Now that the average event rate is calculated, the transition probability from the state s to the next state $\bar{s}$ when the action a is happens: $p(\bar{s}/s, a)$ can be calculated. In the following we will discuss possible cases of $p(\bar{s}/s, a)$ according to the s statements as listed below.

When $s = (x^p, x^r, Q, A_p)$ and $a(s) = P_r$:
$P(\bar{s}/s, a) =$

$$\begin{cases} \frac{(x^p - I_i)\mu_p}{\gamma(s,a)} & \begin{cases} \bar{s} = (x^p - I_i, x^r, Q, D_p) \\ a(s) = -1, \ \ i = 1, 2, ..., Q. \end{cases} \\ \frac{\lambda_s}{\gamma(s,a)} & \begin{cases} \bar{s} = (x^p, x^r + I_j, Q, A_r) \\ a(s) = R, \ \ j = 1, 2, ..., (M - Q). \end{cases} \\ \frac{\lambda_p}{\gamma(s,a)} & \begin{cases} \bar{s} = (x^p + I_i, x^r, Q, A_p) \\ a(s) = P_r, \ \ i = 1, 2, ..., Q. \end{cases} \end{cases} \tag{11}$$

We note that the vector I_i is a null vector with Q elements, except the i^{th} element which is 1, and the vector I_j is also a null vector of $M - Q$ elements, except the j^{th} element which is 1.

When $s = (x^p, x^r, Q, A_r)$ and $a(s) = 0$:
$P(\bar{s}/s, a) =$

$$\begin{cases} \frac{(x^p-I_i)\mu_p}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p-I_i,x^r,Q,D_p) \\ a(s)=-1, \quad i=1,2,...,Q. \end{cases} \\ \frac{(x^r-I_j)\mu_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r-I_j,Q,D_r) \\ a(s)=-1, \quad j=1,2,...,(M-Q). \end{cases} \\ \frac{\lambda_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r,Q,A_r) \\ a(s)=0. \end{cases} \end{cases} \tag{12}$$

When $s=(x^p,x^r,Q,A_r)$ and $a(s)=R$:
$P(\bar{s}/s,a)=$

$$\begin{cases} \frac{(x^p-I_i)\mu_p}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p-I_i,x^r,Q,D_p) \\ a(s)=-1, \quad i=1,2,...,Q. \end{cases} \\ \frac{(x^r-I_j)\mu_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r-I_j,Q,D_r) \\ a(s)=-1, \quad j=1,2,...,(M-Q). \end{cases} \\ \frac{\lambda_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r,Q,A_r) \\ a(s)=0. \end{cases} \\ \frac{\lambda_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r+I_j,Q,A_r) \\ a(s)=R, \quad j=1,2,...,(M-Q). \end{cases} \end{cases} \tag{13}$$

When the state $s=(x^p,x^r,A_p)$, and $\epsilon=D$, no specific action is required except an update of active VMs, as $a(s)=-1$,the associated transition probability is given as follows
$P(\bar{s}/s,a)=$

$$\begin{cases} \frac{(x^p-I_i)\mu_p}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p-I_i,x^r,Q,D_p) \\ a(s)=-1, \quad i=1,2,...,Q. \end{cases} \\ \frac{(x^r-I_j)\mu_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r-I_j,Q,D_r) \\ a(s)=-1, \quad j=1,2,...,(M-Q). \end{cases} \\ \frac{\lambda_s}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p,x^r+I_j,Q,A_r) \\ a(s)=R, \quad j=1,2,...,(M-Q). \end{cases} \\ \frac{\lambda_p}{\gamma(s,a)} & \begin{cases} \bar{s}=(x^p+I_i,x^r,Q,A_p) \\ a(s)=P_r, \quad i=1,2,...,Q. \end{cases} \end{cases} \tag{14}$$

V. Cost and optimal solution

The system cost under the state s and the taken action a is noted by :

$$c(s,a)=r(s,a)\tau(s,a) \tag{15}$$

The $c(s,a)$ is the system cost estimated by taking the action a under the state s in the event ϵ happens.
This combines both the revenue and the cost of the system. The main benefit of the system is to conserve energy consumption and ensure the quality of service. The function must reflect the impact of both [30].
To calculate the energy consumed in the system, the following equation is used :

$$E_{used}=E_Q+\sum_{j=1}^{M}\sum_{i=1}^{l}\delta\times C_{ij}t+\sum_{j=Q+1}^{M}P_jz_jt \tag{16}$$

where E_Q is the energy needed for monitoring the system when just the MPM is running, P_j is the power consumption to start a new PM from RPM, and δ is a powerful weight coefficient.
Besides, $r(s,a)$ is defined by the number of occupied VMs in the PMs taking into account the total capacity of the system i.e.

$c(s,a)=$

$$\begin{cases} E_Q+\sum_{i=1}^{M}C_i\delta t+\sum_{i=Q+1}^{M}P_it & a(s)=0, \quad \epsilon=A_r \\ E_Q+\sum_{i=1}^{M}C_iz_i\delta t+\sum_{i=Q+1}^{M}P_iz_it & a(s)=-1, \quad \epsilon=D \\ E_Q+\sum_{i=1}^{M}C_iz_i\delta t+\sum_{i=Q+1}^{M}P_iz_it+C_{ij}\delta t & \begin{cases} a(s)=R, \quad \epsilon=A_r, \\ j=1,...,Q,\sum_{i=1}^{Q}V_{ij}\le n. \end{cases} \\ E_Q+\sum_{i=1}^{Q}C_i\delta t+C_{ij}\delta t & \begin{cases} a(s)=P, \quad \epsilon=A_p, \\ jj=1,...,Q,\sum_{i=1}^{Q}V_{ij}>n. \end{cases} \end{cases} \tag{17}$$

A. Discounted cost model

Let us consider $\tau_n=t_n-t_{n-1}$ the n^{th} time of stay, between two successive instants of transition, and a_n is the decision taken at the moment t_n to get the moment t_{n+1}.
For a given policy π, and an instant t, a cost $c(s_t,a_t)$ is assumed and is defined by :

$$c(s_t,a_t)=r(s_t,a_t)\tau(s_t,a_t) \tag{18}$$

$(s_t)_{t\in\mathbb{R}^+}$ is a markov process of decision making for state spaces. $F=\mathbb{N}^M$ and actions $A=\{-1,0,P,R\}$.

$$V\delta(s,\pi)=E_\pi\left[\int_0^\infty e^{-\delta t}c\left(s_t,a_t\right)dt/s_0=s\right] \tag{19}$$

where E_π is the mean under the policy π, $\delta>0$ and s is an initial state of the system. the optimal policy exists because the cost function is positive so the set of actions is finite; according to assumption 2 of the proposition in subsection C [31].

B. Discretization of the problem

In this section, we are going to look for a transformation of the problem into an equivalent dynamic programming problem in discrete-time as a normalization procedure[32].
let $\nu=\lambda_p+\lambda_s+Q\mu_p+(M-Q)\mu_s$.
We define $0<t_0<t_1<...<t_n<...$ as the moments of transition in the state of the system. The time intervals are independents and exponentially distributed:

$$P\left(t_{k+1}-t_k>t\right)=e^{-t\nu} \quad k=0,1,2,\ldots \tag{20}$$

Proposition
The cost $V^\delta(s,\pi)$ for any policy π and any initial condition s is :

$$E_\pi\left(\int_0^{+\infty}e^{-\delta t}c\left(s_t,a_t\right)dt\right)=\frac{1}{\delta+\nu}\sum_{n=0}^{+\infty}\left(\frac{\nu}{\delta+\nu}\right)^nE_\pi\left(c\right)\left(s_n,a_n\right)) \tag{21}$$

where $s_n=s_{t_n}$ and $a_n=a_{t_n}$.

Proof: For any couple (s,a), and any policy π and any initial condition s, the cost $V^\delta(s,\pi)$ is :

$$\begin{aligned} &E_\pi\left(\int_0^{+\infty}e^{-\delta t}c\left(s_t,a_t\right)dt\right) \\ &=E_\pi\left(\sum_{n=0}^{+\infty}\int_{t_n}^{t_{n+1}}e^{-\delta t}c\left(s_t,a_t\right)dt\right) \\ &=\sum_{n=0}^{+\infty}E_\pi\left(\int_{t_n}^{t_{n+1}}e^{-\delta t}c\left(s_n,a_n\right)dt\right) \\ &=\sum_{n=0}^{+\infty}E_\pi\left(\int_{t_n}^{t_{n+1}}e^{-\delta t}dt\right)E_\pi\left(c\left(s_n,a_n\right)\right) \end{aligned} \tag{22}$$

The equality (22) results from the fact that the sequence of states are independents of the stay times because the policy π is deterministic: it does not depend on τ_n.
So, we have:

$$\begin{aligned} &E_\pi\left(\int_0^{+\infty}e^{-\delta t}c\left(x_t,a_t\right)dt\right) \\ &=\sum_{n=0}^{+\infty}E_\pi\left(\int_{t_n}^{t_{n+1}}e^{-\delta t}dt\right)E_\pi\left(c\left(s_n,a_n\right)\right) \end{aligned} \tag{23}$$

Since the variables τ_n are independents, so:

$$\begin{aligned} E_\pi\left(\int_{t_n}^{t_{n+1}}e^{-\delta t}\right) &=E_\pi\left(e^{-\delta t_n}\int_0^{\tau_{n+1}}e^{-\delta t}dt\right) \\ &=\tfrac{1}{\delta}E_\pi\left(e^{-\delta t_n}\right)\left(1-E_\pi\left(e^{-\delta\tau_{n+1}}\right)\right) \\ &=\tfrac{1}{\delta}\left(\tfrac{\nu}{\delta+\nu}\right)^n\left(1-\tfrac{\nu}{\delta+\nu}\right) \end{aligned} \tag{24}$$

Indeed, for all

$$n\ge0,\tau_{n+1}\sim\exp(\nu)\Longrightarrow E_\pi\left(e^{-\delta\tau_{n+1}}\right)=\frac{\nu}{\delta+\nu}$$

and

$$\begin{aligned} E_\pi\left(e^{-\delta t_n}\right) &=E_\pi\left(e^{-\delta\left(\sum_{k=1}^{n}t_k-t_{k-1}\right)}\right) \\ &=\textstyle\prod_{k=1}^{n}E_\pi\left(e^{-\delta\tau_k}\right)=\left(\frac{\nu}{\delta+\nu}\right)^n \end{aligned} \tag{25}$$

hence

$$E_\pi\left(\int_0^{+\infty} e^{-\delta t} c\left(s_t, a_t\right) dt\right) = \frac{1}{\delta+\nu} \sum_{n=0}^{+\infty} \left(\frac{\nu}{\delta+\nu}\right)^n E_\pi\left(c\left(s_n, a_n\right)\right) \quad (26)$$

■

Based on this proposal, we can write:

$$V^\delta(s,\pi) = \frac{1}{\delta+\nu} E_\pi\left(\sum_{n=0}^{+\infty}\left(\frac{\nu}{\delta+\nu}\right)^n c\left(s_n, a_n\right)/s_0 = s\right) \quad (27)$$

which is an expression of the discrete-time discounted cost under the policy π, with the discount rate $\beta = \frac{\upsilon}{\delta+\nu}, (0 < \beta < 1)$ and the cost function $\frac{c(s_n;a_n)}{\delta+\nu}$. Then the cost is defined by:

$$V_N^\beta(s,\pi) = E_\pi\left(\sum_{n=0}^{N-1} \beta^n c\left(s_n, a_n\right)/s_0 = s\right) \quad (28)$$

and the cost β-discounted for infinite horizon is defined by :

$$V^\beta(s,\pi) = E_\pi\left(\sum_{n=0}^{+\infty} \beta^n c\left(s_n, a_n\right)/s_0 = s\right) \quad (29)$$

C. Existence of the optimal policy

Proposition

i.) For all $s \in F$, we have:

$$\begin{cases} V_N^\beta(s) = \min\limits_{a\in A(s)}\left\{c(s,a) + \beta\sum\limits_{\bar{s}\in F} p(\bar{s}/s,a) V_{N-1}^\beta(\bar{s})\right\}, \forall N \geq 1. \\ V_0(\cdot) = 0 \end{cases}$$

ii.) If the cost is positive and the set of actions is finite, then:

$$\lim_{N\to+\infty} V_N^\beta(s) = V_\infty^\beta(s) = V^\beta(s)$$

and V^β so the only solution is given by the following equation:

$$V^\beta(s) = \min_{a\in A(s)}\left\{c(s,a) + \beta\sum_{\bar{s}\in F} P\left(\bar{s}/s,a\right) V^\beta(\bar{s})\right\}$$

For the proof of this proposition, see [31][32].

VI. Numerical Results and Analysis

In this section, the suggested scheme (SMDP) is compared with other schemes and the numerical results are obtained using Matlab and then discussed to support our performance analysis. In the meantime, the principal parameters of our analysis are given in Table 2.

Parameter	K	M	n	λ_p	λ_s	μ_p	μ_s	P_i	e_j
Value	64	16	8-32	1-9	7	8	6	40W	15W

TABLE II
System Parameters

In order to compare, we evaluate the performance of the following two reference schemes.

Greedy Algorithm (GA) scheme: this algorithm is designed to maximize the reward of the system at each moment in any decision [33].

Location Precedence (LP) scheme: which is a heuristic algorithm that applies a priority policy to send requests to the MPM or RPM. The LP scheme always assignes the more virtual machines in a physical machine that can support [33].

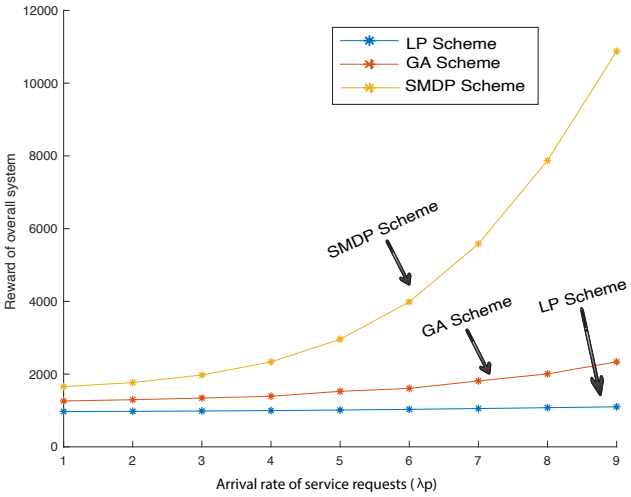

Fig. 2. System reward under different arrival rates for studied schemes.

Figure 2 illustrates the system reward under the three compared schemes. When λ_p is low, the differences between the schemes are slight, because the virtual machines in the MPM are sufficient. However, with the increase of λ_p, the reward of all schemes increases gradually and the benefit of the SMDP scheme becomes more evident. Besides, the LP scheme is less efficient, while the SMDP scheme is more efficient than the GA scheme. This is because the LP scheme does not consider the state of the reserves at all. However, the SMDP scheme sends the request to the RPM, which reduces the workload of the MPM in order to improve the quality of service expected from the system.

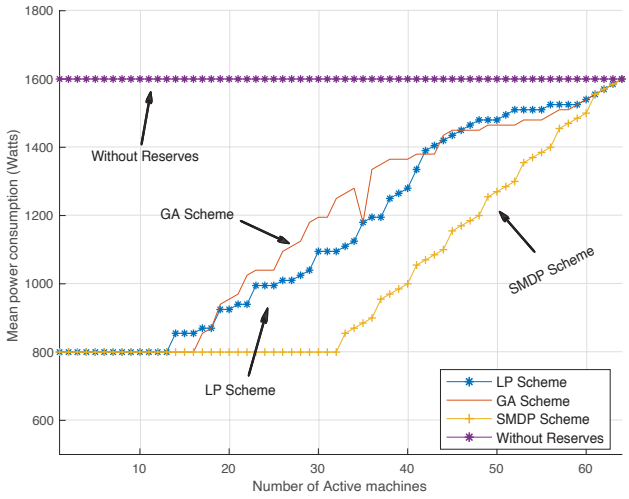

Fig. 3. System power consumption of each scheme under different number of active machines .

In figure 3, we present the mean power consumption in the system in relation to the number of activated machines. For all the three schemes, the system starts with a minimum consumption E_Q equal 800 W. According to the decisions made by the algorithms GA and LP to use a new machine among the reserves, their power consumption increases rapidly, because even if a machine is not fully loaded, the algorithm prefers to use a new machine for new customers until all machines are activated and then it recycles the possibility to send the incoming customers to the machines that still have the capacity. For our SMDP approach we observe that the power consumption remains minimal by using an optimal number of MPMs, after that the controller decides to use the RPMs one by one until all machines are activated. The fourth graph shows the energy consumption without using the concept of reserve machines and it is clear that the consumption is high even the number of activated machines is low. From this, we can say that our algorithm can conserve more energy over time compared to other approaches.

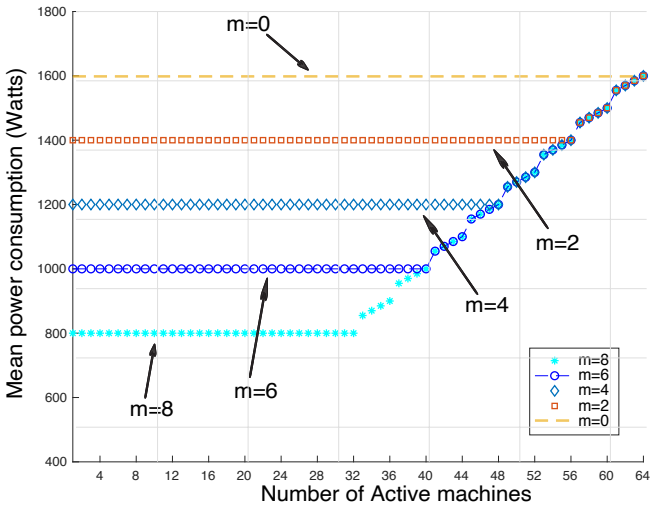

Fig. 4. Mean power consumption curves as function of number of reserves

Figure 4 shows the variation in the system's power consumption when the number of reserves $m = M - Q$ is changed and compares it to the static mode when all PMs are executed for the first time (there are no reservations). This figure shows that the last case, when all particles are running even if they are not necessary, is the worst in terms of energy consumption . In addition, power consumption becomes lower when a few PM are left in reserve. As shown in the figure, we notice that when the number of reserves increases, the power use is saved, and for example, when we use 8 PMs as reserves, we can save up to 50% of power (with 32 customers) and 37.5% when we reserve 6 PMs and 12.5% when we reserve 2PMs. This reservation of PM leads to a reduction in energy consumption and optimal use of resources.

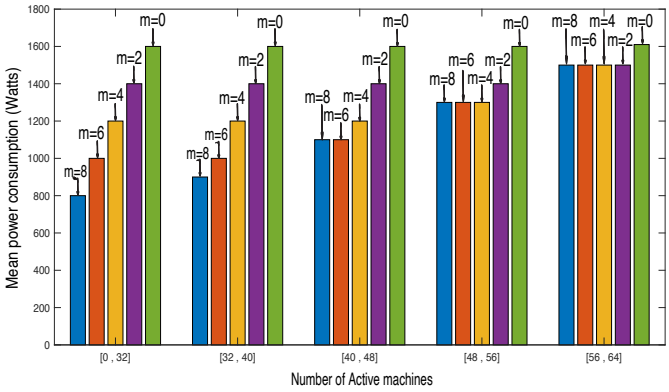

Fig. 5. Mean power consumption curves as function of set of active machines

The variation in power consumption as a function of the set of active VMs is illustrated in figure 5, this variation is presented in the form of intervals which present the role numbers of machines reserved in the system.

This figure confirms the effect of the number of active machines on system performance as well as the effect of the reservation of physical machines on energy consumption. But the most important fact deduced from this figure is that the number of RPMs has a good effect on the performance when the system load is not very large. However, the more the number of active machines increases the more the energy consumption tends to become independent of the number of reserved physical machines. This can be explained by the fact that, in order to ensure good performance, and when the number of requests increases, the system is forced to put RPMs in on mode, and these RPMs consume energy in the same way as the MPMs, which increases the energy consumption. In the case, without reserves (the green bar) the consumption remains high even if we have only one customer.

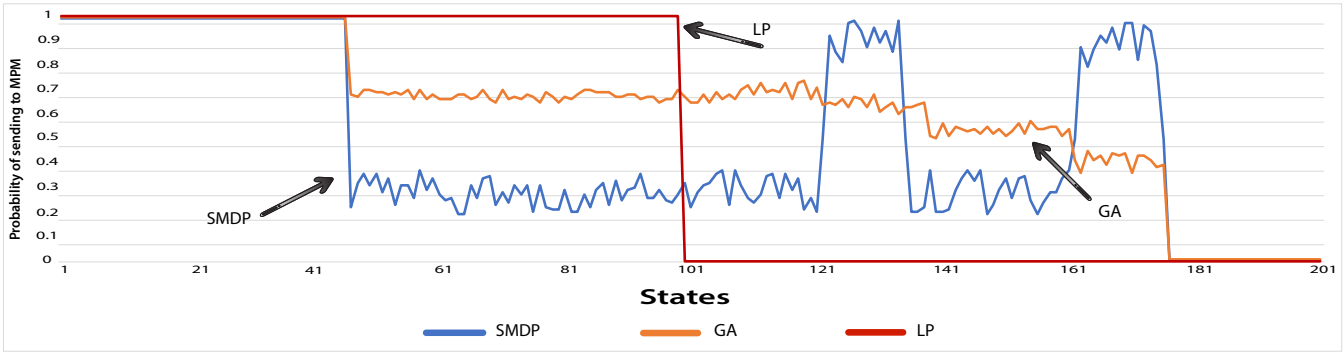

Fig. 6. The probability decision for sending the arrivals of customers to MPM with SMDP, GA and LP Schemes.

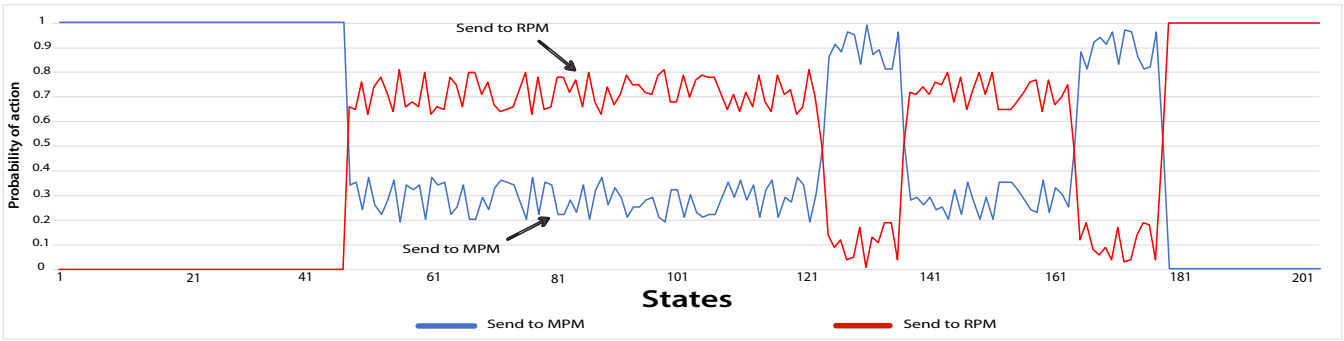

Fig. 7. The probability decision for controlling the arrivals of customers with SMDP Scheme.

Figure 6 presents the probability to send new clients to MPM for the three schemes. For our SMDP scheme we notice that the probability that the controller uses the main machines until the system has n clients is equal to 1. That means that the only decision possible for the controller is to use the MPM until he gets n clients. After that, the possibility to send the customers to any MPM decreased nearly to 0.3, which means that the probability to use a machine among the reserves increased to 0.6. This means that the controller prefer to use new machine from RPM to ensure the QoS rather than use only the MPM to save energy. When all the RPM are busy and the customers still arrive, the probability to send uniformly to one of the PM is increased until all the MPM are full and then the probability to send to any MPM is equal to 0. For the other two algorithms their probabilities decision results are presented in the same figure (orange color and red color). It is clear that the GA algorithm is not stable; it means that it is zigzag between the choices of decision to maximize the instantaneous cost of the system until all the MPM are full. For the LP scheme, it is noted that it has no interest to use new machines reserved, except the already used, until all MPM are filled and that is obvious in the figure as the probability to send to MPM equal 1 when all MPM are full and then the probability is equal to 0.

The probabilities of decision to make each one from the two possible actions for our proposed scheme are presented in figure 7. This latter shows that before having n customers, the obvious decision is to choose a MPM with a probability equal to 1 and to use a RPM with a probability equal to 0. After, when the system has n clients the controller allows the possibility to turn on the reserve machines, in this case we notice that the probability of sending a new client to RPM is increasing. On the contrary, the probability of sending clients to MPM is decreasing until the system reached a certain number of clients (when all MPM and RPM machines have at least $l * Q$ clients) where the controller chooses one more time to use the MPMs depending on the state of the system and taking into account the QoS. Towards the end, the probability to send to RPM is 1 and to send to MPM is 0, and this is normal because all the MPMs are busy and it only remains the possibility to install the VMs in RPM.

VII. Conclusion

The mapping of virtual machines to physical machines in a cloud computing system poses several challenges due to the dynamic nature of such an environment and to energy and performance constraints in the system. In this paper, we have been interested in modeling and evaluating the performance of a cloud computing system where some physical machines were considered as reserves and a controller turn them on one by one under some conditions. The optimal resource allocation problem with the objective of maximizing the expected reward is formulated as a markov decision process problem. The optimal policy for controlling customer arrivals has been proven by the theory of dynamic programming. The numerical results show that the proposed scheme significantly outperforms all the considered schemes and improves the expected reward. Moreover, the proposed approach can significantly reduce the energy consumption of the system. These results are demonstrated in various scenarios. In perspective, it will be more interesting to be able to determine the optimal number of physical machines that should be reserved for a given system workload.

References

[1] T. Abayomi-Zannu and I. Odun-Ayo, "Cloud identity management–a critical analysis," in *Lecture Notes in Engineering and Computer Science: Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2019, pp. 13–15.

[2] S. Kananizadeh and K. Kononenko, "Improving on linear scan register allocation," *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 228–238, 2018.

[3] X. Liu, Y. Zhang, and K. Zhang, "Optimization control of energy consumption in tunneling system of earth pressure balance shield tunneling machine." *Engineering Letters*, vol. 28, no. 2, pp. 551–558, 2020.

[4] H. El Ghor and M. Chetto, "Energy guarantee scheme for real-time systems with energy harvesting constraints," *International Journal of Automation and Computing*, vol. 16, no. 3, pp. 354–368, 2019.

[5] S. Mustafa, B. Nazir, A. Hayat, S. A. Madani *et al.*, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Computers & Electrical Engineering*, vol. 47, pp. 186–203, 2015.

[6] D. Liu, X. Sui, and L. Li, "An energy-efficient virtual machine placement algorithm in cloud data center," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2016, pp. 719–723.

[7] U. Wajid, C. Cappiello, P. Plebani, B. Pernici, N. Mehandjiev, M. Vitali, M. Gienger, K. Kavoussanakis, D. Margery, D. G. Perez *et al.*, "On achieving energy efficiency and reducing co2 footprint in cloud computing," *IEEE transactions on cloud computing*, vol. 4, no. 2, pp. 138–151, 2015.

[8] E. M. Kandoussi, M. Hanini, I. El Mir, and A. Haqiq, "Toward an integrated dynamic defense system for strategic detecting attacks in cloud networks using stochastic game," *Telecommunication Systems*, vol. 73, no. 3, pp. 397–417, 2020.

[9] B. Hajek, "Optimal control of two interacting service stations," *IEEE transactions on automatic control*, vol. 29, no. 6, pp. 491–499, 1984.

[10] I. Odun-Ayo, T.-A. Williams, O. Abayomi-Alli, and J. Yahaya, "Systematic mapping study of economic and business models of cloud services," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, pp. 987–994, 2020.

[11] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server farms with setup costs," *Performance Evaluation*, vol. 67, no. 11, pp. 1123–1138, 2010.

[12] X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, and R. Buyya, "Energy-aware virtual machine allocation for cloud with resource reservation," *Journal of Systems and Software*, vol. 147, pp. 147–161, 2019.

[13] A. Ouammou, A. BenTahar, M. Hanini, and S. El Kafhali, "Modeling and analysis of quality of service and energy consumption in cloud environment," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 10, 2018.

[14] S. El Kafhali and K. Salah, "Performance analysis of multi-core vms hosting cloud saas applications," *Computer Standards & Interfaces*, vol. 55, pp. 126–135, 2018.

[15] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268–280, 2012.

[16] M. Hanini, S. E. Kafhali, and K. Salah, "Dynamic vm allocation and traffic control to manage qos and energy consumption in cloud computing environment," *International Journal of Computer Applications in Technology*, vol. 60, no. 4, pp. 307–316, 2019.

[17] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An smdp-based resource allocation in vehicular cloud computing systems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7920–7928, 2015.

[18] H. Meng, K. Zheng, P. Chatzimisios, H. Zhao, and L. Ma, "A utility-based resource allocation scheme in cloud-assisted vehicular network architecture," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 1833–1838.

[19] S. El Kafhali and M. Hanini, "Stochastic modeling and analysis of feedback control on the qos voip traffic in a single cell ieee 802.16 e networks," *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 19–28, 2017.

[20] G. Bindu, K. Ramani, and C. S. Bindu, "Optimized resource scheduling using the meta heuristic algorithm in cloud computing." *IAENG International Journal of Computer Science*, vol. 47, no. 3, pp. 360–366, 2020.

[21] I. Mitrani, "Managing performance and power consumption in a server farm," *Annals of Operations Research*, vol. 202, no. 1, pp. 121–134, 2013.

[22] A. Ouammou, M. Hanini, A. B. Tahar, and S. El Kafhali, "Analysis of a m/m/k system with exponential setup times and reserves servers," in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, 2019, pp. 1–5.

[23] S. El Kafhali and K. Salah, "Efficient and dynamic scaling of fog nodes for iot devices," *The Journal of Supercomputing*, vol. 73, no. 12, pp. 5261–5284, 2017.

[24] M. Andersson, A. Bengtsson, M. Höst, and C. Nyberg, "Web server traffic in crisis conditions," in *In Proceedings of the Swedish National Computer Networking Workshop, SNCNW 2005*, 2005.

[25] K. Xiong and H. Perros, "Service performance and analysis in cloud computing," in *2009 Congress on Services-I*. IEEE, 2009, pp. 693–700.

[26] Z. Wang, Y. Chen, D. Gmach, S. Singhal, B. J. Watson, W. Rivera, X. Zhu, and C. D. Hyser, "Appraise: application-level performance management in virtualized server environments," *IEEE Transactions on Network and Service Management*, vol. 6, no. 4, pp. 240–254, 2009.

[27] A. Ouammou, M. Hanini, A. BenTahar, and S. El Kafhali, "A dynamic programming approach to manage virtual machines allocation in cloud computing," *International Journal of Engineering & Technology*, vol. 7, no. 4.6, pp. 128–132, 2018.
[28] C. Wei, "Almost sure exponential stability of nonlinear stochastic delayed systems with markovian switching and lévy noises," *IAENG International Journal of Applied Mathematics*, vol. 49, no. 3, pp. 351–358, 2019.
[29] D. Xu, H. Xian, X. Cui, and Y. Hong, "A new single-valued neutrosophic distance for topsis, mabac and new similarity measure in multi-attribute decision-making," *IAENG International Journal of Applied Mathematics*, vol. 50, no. 1, pp. 72–79, 2020.
[30] M. Nasseri, M. Alam, and R. C. Green, "Mdp based optimal policy for collaborative processing using mobile cloud computing," in *2013 IEEE 2nd international conference on cloud networking (CloudNet)*. IEEE, 2013, pp. 123–129.
[31] P. Whittle, *Optimal control: basics and beyond*. John Wiley & Sons, Inc., 1996.
[32] S. M. Ross, *Introduction to stochastic dynamic programming*. Academic press, 2014.
[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

Abdellah OUAMMOU is a PH.D. student in applied mathematics at Computer, Networks, Mobility and Modeling laboratory, Faculty of Sciences and Techniques, Hassan 1st university, Settat, Morocco. He has completed his bachelor degree in mathematics and applications at the faculty of sciences, Ibn zohr university, Agadir, Morocco, in 2013, and he received the Master degree in mathematics and applications from Faculty of Sciences and Techniques, Hassan 1st university, Settat, Morocco, in 2016. His current research interests are probability, stochastic optimization, discrete stochastic processes, discrete optimization, and cloud computing environments.

Abdellah Zaaloul has a PhD in applied mathematics, option modeling and performance evaluation of computer communication networks, at faculty of sciences and technologies (FSTS), settat, morocco. He received his M.Sc. degree in mathematics and applications from the Hassan 1st university, faculty of sciences and technologies in 2011. He has been working as professor of mathematics in high school since 1992 until 2018. And, since that date he has been working as a professor at the department of mathematics faculty of legal, economic and social sciences of aït melloul ,Ibn zohr university, agadir, morocco. Currently, he is member of e-NGN research group. Dr. Abdellah Zaaloul's interests lie to the areas of modeling and performance evaluation of communication networks, mobile communications networks, cloud computing and security, queuing theory and game theory, stochastic control and networking games.

Mohamed Hanini is currently a professor at the department of Mathematics and Computer science in the Faculty of Sciences and Techniques, Hassan 1st university Settat, Morocco. He obtained his PH.D degree in Mathematics and Computer in 2013. He is a member of e-NGN Africa and middle east research group, and an IAENG member. He is the author and co-author of several papers related to the fields of modeling and performance evaluation of communication networks, cloud computing and security. He participated as TPC member and as an organizing committee member in international conferences and workshops, and he worked as reviewer for several international journals.

Abdelghani Ben Tahar received his PhD degree in applied mathematics from Hassan II university, Casablanca, Morocco in 2001. He was in an INRIA post-doctoral at Rocquencourt and a CNRS post-doctoral fellows at LIRMM, Montpellier, France, and he works as a lecturer at LMRS (UMR 6085 CNRS-Univ. Rouen). Since 2009 he is a Professor at Hassan 1st University, Settat, Morocco. His current fields of research interests are focusing on networks performance evaluation and queuing network models.