A Strategy for Server Management to Improve Cloud Service QoS

Binh Minh Nguyen, Dang Tran, and Quynh Nguyen
School of Information and Communication Technology
Hanoi University Science and Technology
Hanoi, Vietnam
minhnb@soict.hust.edu.vn, dangitbk@gmail.com, nguyen.nhu.quynh.1993@gmail.com

Abstract— **According to [1], cloud computing is one of ICT core areas in the next five years. The computing paradigm also is a research trend that has attracted strongly scientist community. Until now, there are many studies, which have focused on saving energy for servers in cloud systems. During the operation process of cloud data center, idle servers bring about power waste phenomenon. In order to overcome the problem, turning off idle servers is the popular solution applied in most researches. However, the turn on/off processes often affect significantly to quality of service (QoS) of cloud services because the server turning on always takes a long time and thus reduces the capability of quick response of services that run on these servers. In this paper, we present a strategy for effective server management. The strategy is developed based on a novel three-state model for physical servers belonging to cloud data centers. Our proposed model with an intermediate state will decrease the waiting service time for cloud appliances deployed on server machines. As a result, the approach will improve the cloud service QoS. We use CloudSim [2] to experiment and evaluate the solution. We also consider the power consumption of our model to examine its effectiveness.**

Keywords—cloud computing, green computing, queue theory, Markov chain, CloudSim, saving energy.

I. INTRODUCTION

Cloud computing is changing the way that IT services are being sourced and delivered. The motivation for this wave is the potential of cost savings and accelerating time to market. As a result, IT organizations of all sizes and types are becoming early adopters of cloud technologies. Consistent with this trend, the number of cloud services and applications are exponentially increasing every day. This trend also encourages cloud vendors to take interest in improving quality of their services. Once the providers ensure that their services offered with high quality, then they can scramble for profit in cloud market with many competitors today.

Inside clouds, resources and services that are abstracted from the underlying infrastructures and provided on demand at scale in a shared multitenant and elastic environment. It is easily to see that physical servers play the important role in clouds. The servers are strength and foundation for all services/appliances running on. However, in fact, the management and control of cloud servers in such a way that allow the reasonable balance between energy consumption and ensuring service quality are a difficult and complex problems. While the cloud providers would like to decrease the number of idle servers by turning off them (for saving power costs), they still have to keep the ability of quick response with user resource requirements. In the case of no or little available servers in ON state, the ability will be definitely effected.

So far, there are many studies that proposed different methods in order to resolve the problem of saving energy or controlling arriving jobs into cloud systems. In their paper, Saiqui Long, Zhao Yuelong, and Chen Wei [3] describe a strategy with three phases of energy-saving for cloud storage systems based on the Poisson distribution, queue length and Markov chain. In the similar way, Phung-Duc Tuan [4] also uses the Poisson distribution and Markov chain to optimize the job serving with two states of cloud servers: OFF (shutdown) and ON (serving jobs). In this system, the author assumes that if a server finishes a job, it (in the ON state) are turned off immediately and automatically. Otherwise, when another job arrives, the servers will switch its state. To change effectively states and reduce power consumption, the authors of [5] and [6] provide intelligent management approaches to turn on and off the physical servers. However, until now, there are still not yet researches dealing with the problem of decreasing waiting time while cloud system serve arriving jobs but simultaneously also consider with energy issue. Thus, it will improve the quality of cloud services in the aspect of more quickly and effectively serving jobs.

In order to achieve the ability of quick response with requirements of cloud resources, in this paper, beyond ON and OFF, we define an intermediate state (called MIDDLE) for the physical servers in cloud data centers. With the state, a reasonable number of servers is turned on and kept availably to wait and serve arriving jobs. Due to the servers at MIDDLE state, the clouds can eliminate booting process and thus decrease the waiting time while serving arriving jobs into the system. Like other researches, we also mainly rely on Poisson distribution and Markov chain to design algorithms, which enable to control when the number of servers in MIDDLE state is increased or decreased accordingly. Our three-state model and control algorithms to turn on/off servers are installed and experimented using CloudSim. Moreover, besides studying QoS for cloud services, we also evaluate the power consumption by comparing our model with no MIDDLE state model. In this way, we prove the model efficiency in relation to energy factor in cloud data centers.

The rest of the paper is organized as follows. In Section 2, we discuss the related works. In Section 3, we present the methodologies of our model in detail. In Section 4, we propose the resource allocation model and algorithms for managing MIDDLE servers in cloud data centers. In Section 5, we evaluate the performance in reducing serving time and energy consumption of our model. Finally, conclusions and future works are given in Section 6.

II. RELATED WORK

Recently, a large amount of research works in the field of energy saving for cloud infrastructures have been carried out. Hamilton [7] focuses on optimizing hardware energy consumption by putting forward an architecture for server racks that use low-power components like AMD Althon processors. In the similar way, the authors of [8] also design cloud storage system applying saving power ingredients for Hadoop platform.

Some other researches such as [9] and [10] exploit migration technologies in cloud environment to minimize the number of servers in the ON state. Thus, these techniques will evidently economize on data center power. Most of them aim at moving virtual machines (VMs), images or data in cloud compute/storage servers.

Another big set of works introduce algorithms, which can control and manage servers based on active and inactive hardware states. The authors of [4], [11] and [12] propose strategies to control servers, which have only two states ON and OFF. Their goal is to calculate exactly the waiting time of jobs inside the cloud systems using queue theory and thus optimize the energy consumptions. However, the waiting time for running jobs in the systems is always quite long because the machines are switched from ON to OFF state (in the case of there are not jobs running on the machines) and from OFF to ON (when jobs arrive into system). The process of turning on servers brings about spending a lot of time to boot, install and configure software platforms in order to serve the jobs. According to the intensity of arriving jobs, the work described in [3] presents an algorithm to turn on and off cloud storage servers. In contrast to others, this work defines active servers with three states: working, standby and idle. Although, the authors only develop an algorithm, which enables cloud system to automatically switch from idle to standby mode. In the case of the number of idle and standby servers is zero, the system of course has to turn on some servers to serve arriving jobs. This process also takes a long time until the system is ready to run jobs. Due to the goal of reducing power consumption, the authors still do not determine when and how many storage servers are in standby, idle and power off modes. Theoretically, Anshul Gandhi, Mor Harchol-Balter, and Ivo Adan [11] go deeply into mathematical models by listing several scenarios for server farms with setup costs. The authors consider totally three operational policies and two behavior models in proportion to the server states and system reactions with arriving jobs. Unfortunately, the work do not define any certain model with an intermediate state for servers.

Inspired by the works of [3], [4] and [11], we propose a novel three-state model for servers in cloud data centers. In comparison with the discussed studies above, our model has main differences and contributions as follows:

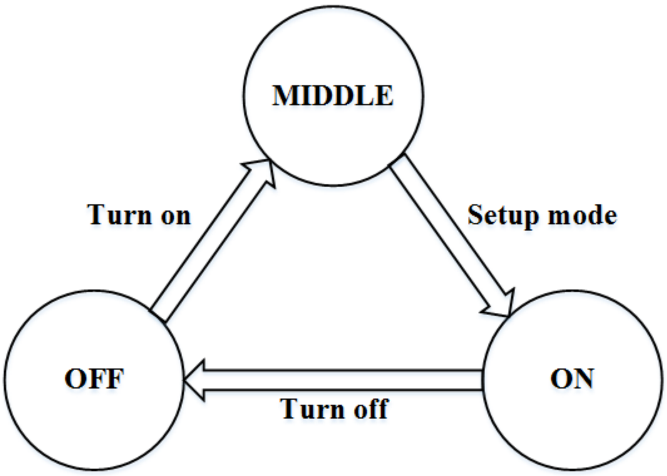


Fig. 1. State transition diagram of the physical servers

- We define a three-state model of servers in the data center of clouds, namely ON, MIDDLE and OFF. Furthermore, we calculate and keep an amount of servers in the MIDDLE state to guarantee that the waiting serving time will be decreased in all cases.
- We express and demonstrate the mathematical model based on Markov chain and queuing for three states of cloud servers above.
- Based on the mathematical formulae, we propose two policies: one only allows one server at MIDDLE serving one job at the same time. The other allows multiple servers at MIDDLE state simultaneously serving arriving jobs. The goal of the formulae are to evaluate their effectiveness in decreasing serving time with real cloud environment.
- Through carrying out experiments, gained results are shown to prove the ability of increasing QoS for cloud appliances by reducing service time. Otherwise, we also consider the power factor for each proposed policy with appearance of MIDDLE state. From the experimental outcomes, we can give remark about the correlation among ensuring QoS and energy while applying our model.

III. THEORY SYSTEM MODEL

In fact, depending on the server farm capacity, several strategies could be offered to manage effectively machines in the manner of optimizing service capability. In this direction, we consider two policies corresponding to two cases specifying that server setup process takes place alternately or simultaneously:

- System allows only one server, which can be in SETUP mode at any given time (ON/OFF/MIDDLE/∞/STAG model)
- System allows multiple servers that can be simultaneously in SETUP mode (ON/OFF/MIDDLE/∞ model).

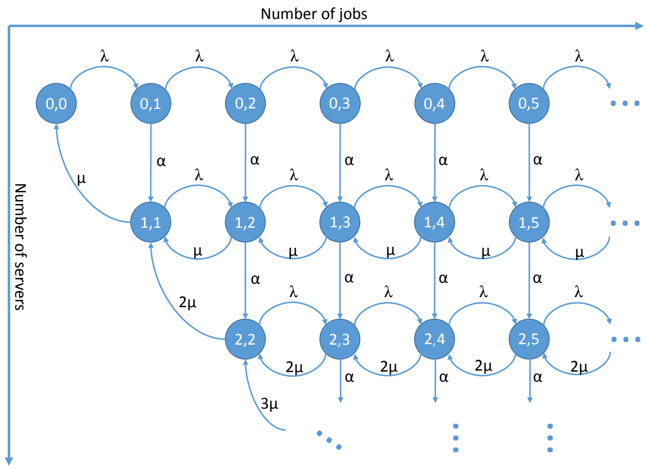


Fig. 2. Markov chain with ON/OFF/MIDDLE/∞/STAG model

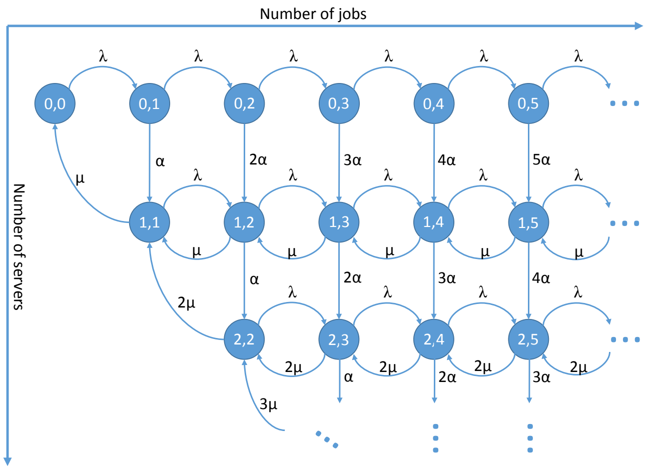


Fig. 3. Markov chain with ON/OFF/MIDDLE/∞ model

A. ON/OFF/MIDDLE/∞/STAG Model

In the data centers, idle servers (without serving jobs) will be turned off to save power. When a server is again turned on to serve arriving jobs, this process often takes a long time because the server has to go through boot and installation phases. To establish expected model, we regard that the number of servers is infinite. Indeed, large data centers of Google, Microsoft, Yahoo and Amazon etc. contains tens of thousands of servers [13]. In [11] Gandhi and his colleagues launch ON/OFF/∞/STAG model for server farms. STAG stands for staggered policy that means only one server is in SETUP mode at any point of time. As mentioned in first part, in this work, we define an intermediate state called MIDDLE among ON (job serving) and OFF (power off) states for physical servers in the cloud data centers. The MIDDLE is a state, in which servers are turned on and kept availably but they do not serve any jobs. Then we have novel model as follows: ON/OFF/MIDDLE/∞/STAG.

When a MIDDLE server is switched to ON state, it must be put into the SETUP mode. Fig. 1 illustrates our general three-state model. The needed time for putting a server from SETUP mode to ON state is called SETUP time. Similar to ON/OFF/∞/STAG, in the ON/OFF/MIDDLE/∞/STAG model, only one job can go to the SETUP mode at the same time. We assume that the SETUP time I is an exponentially distributed random variable with rate $\alpha = 1/E[I]$. The reason for the assumption is that jobs are diverse sizes. We consider the arriving jobs into system as deployment and running processes of VMs on physical servers. Due to the differences of jobs sizes, the job setup time (or VM deployment time) is different. We also suppose that arriving jobs into the system conform to Poisson process with rate λ, and X is time to finish a job having the exponential distribution with $E[X] = 1/\mu$. When a server is not used, it is immediately switched to OFF state. When a new job arrives into the system, this job is put into a queue.

In the ON/OFF/MIDDLE/∞/STAG model, if there are not servers in SETUP mode, a server in MIDDLE state (if it exists) will be switched to SETUP mode. When server j in ON state finishes a job, the first job in the queue will be migrated to the server j although this job is waiting for a certain server i in SETUP mode, then server i will be turned off.

In order to ensure that there are always servers in MIDDLE state when the system needs, we propose a control algorithm to switch OFF server to MIDDLE state. However, the number of MIDDLE servers must not too large to avoid wasting power. The switching process from OFF to MIDDLE state takes the identical time for all servers (called $t_{OFF\rightarrow MIDDLE}$). Assuming that there are always available servers in MIDDLE state, we can perform the state transition of system by Markov chain like ON/OFF/∞/STAG as described in Fig. 2 using *(i,j)* pair, where i is the number of ON servers, j is the number of jobs in the system and when $i = j$, no servers in the SETUP mode.

B. ON/OFF/MIDDLE/∞ model

This model is similar to the ON/OFF/MIDDLE/∞/STAG model presented above except allowing multiple servers can be simultaneously in SETUP mode, the state transition of system is performed in Fig. 3 according to Markov chain.

IV. Resource Allocation Model and Control Algorithms

As mentioned in previous part, it must control the switch process from OFF to MIDDLE state in the manner of always keeping some servers in the intermediate state. Concurrently, the amount of MIDDLE servers is not too large. With a view to ensuring balance between these two conditions, based on the mathematical model, we calculate an average number for servers, which has to switch from OFF to MIDDLE during a given time period.

A. Control algorithm for ON/OFF/MIDDLE/∞/STAG model

In the case of our system always has available servers in MIDDLE state, we can consider the system as the ON/OFF/∞/STAG Markov chain (Fig. 2). The limiting probabilities $\pi_{i,j}$ for this Markov chain are given by [11]:

$$\pi_{i,j} = \frac{\pi_{0,0}.\rho^i}{i!}\left(\frac{\lambda}{\lambda+\alpha}\right)^{j-1}$$

and

$$\pi_{0,0} = \sum_{i=0}^{k} \sum_{j \geq i} \frac{\rho^i}{i!} \left(\frac{\lambda}{\lambda + \alpha}\right)^{j-i}$$

Lemma 1: if $X_1, X_2, \ldots, X_n$ are independent exponential random variables with expectation $\mu_1^{-1}, \mu_1^{-1}, \ldots, \mu_n^{-1}$ then:

$$P\ \{X_i = \min(X_1, X_2 \ldots X_n)\} = \frac{\mu_i}{\mu_1 + \mu_1 + \cdots + \mu_n}$$

Proof. Let $y = \min(X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$, where y has the exponential distribution with expectation $(\mu_1 + \mu_2 + \cdots + \mu_{i-1}, \mu_{i+1} + \cdots + \mu_n)^{-1}$.

$$P\ \{X_i = \min(X_1, X_2, \ldots, X_n)\} = P(X_i < y)$$

If A, B are independent exponential random variables with expectation μ_A^{-1}, μ_B^{-1} then $P(X1 < X2) = \frac{\mu_A}{\mu_A + \mu_B}$.

Hence $P\ \{X_i = \min(X_1, X_2, \ldots, X_n)\} = P(X_i < y) = \frac{\mu_i}{\mu_1 + \mu_2 + \cdots + \mu_n}$

Lemma 2: in ON/OFF/MIDDLE/∞/STAG model, if at any time t the system is in state (θ, i) $(i > \theta)$ then the probability that the system in the state (θ', j) at time $(t + h)$ $(\theta' \neq \theta$ or $i \neq j)$ is:

$$P\ \{(\theta,\ i) \to (\theta', j)\} = \begin{cases} \alpha h + o(h),\ \theta' = \theta + 1 \text{ and } i = j \\ \lambda h + o(h),\ \theta' = \theta \text{ and } j = i + 1 \\ \theta\mu h + o(h),\ \theta' = \theta \text{ and } j = i - 1 \\ o(h),\ \textit{othercases} \end{cases}$$

Proof. If $i > \theta$, job queue is not empty. There are three types of events that can happen during $(t,\ t + h)$: having an arriving job, having a completed job, and having an additional server is switched to ON state. Jobs arrive under Poisson process with rate λ, which is the inter-arrival times are independent, exponentially distributed random variables with expectation λ^{-1}. The service times and setup times are also assumed to be independent and exponentially distributed with expectation μ^{-1} and α^{-1} respectively. Thus, the time from t to the next event has the exponential distribution with expectation $(\lambda + \alpha + \theta\mu)^{-1}$ (because the time to the next event is minimum of the exponential inter-arrival time, the exponential service time θ and the setup time). Thus, the probability that an event will occur in $(t,\ t + h)$ is $1 - e^{-(\lambda + \alpha + \theta\mu)h} = (\lambda + \alpha + \theta\mu)h + o(h)$.

When an event occurs, the probability that it will be caused by an arriving job is: $\lambda/(\lambda+\alpha+ \theta\mu)$ (according to lemma 1). Furthermore, the length of time required for the event to occur and the type of event are independent. Therefore, the probability of occurrence of the transition $(\theta, i) \to (\theta, i+1)$ is:

$$P\{(\theta,\ i) \to (\theta, i+1)\}$$
$$= [(\lambda + \alpha + \theta\mu)h + o(h)] \frac{\lambda}{\lambda + \alpha + \theta\mu} + o(h) = \lambda h + o(h)$$

By a similar argument

$$P\{(\theta,\ i) \to (\theta, i-1)\}$$
$$= [(\lambda + \alpha + \theta\mu)h + o(h)] \frac{\theta\mu}{\lambda + \alpha + \theta\mu} + o(h) = \theta\mu h + o(h)$$

And

$$P\{(\theta,\ i) \to (\theta+1, i)\}$$
$$= [(\lambda + \alpha + \theta\mu)h + o(h)] \frac{\alpha}{\lambda + \alpha + \theta\mu} + o(h) = \alpha h + o(h)$$

The probability that more than one event will occur in $(t,\ t+h)$ is $o(h)$ as $h \to 0$, it follows that $P\ \{(\theta,\ i) \to (\theta', j)\} = o(h)$, in the others cases.

Theorem 1: *in ON/OFF/MIDDLE/∞/STAG model, the expectation of the number of servers that are switched from MIDDLE state to ON state successfully in time period t is given by:*

$$E[K] = \frac{\lambda\alpha t}{\lambda + \alpha} \quad (1)$$

Proof. Assuming K_h is number of servers that are switched to ON state in time h. Thus, $K_h > 0$ when the transition $(\theta, i) \to (\theta', j)$ $(\theta' > \theta)$ occurs. According to lemma 2, $P\{(\theta, i) \to (\theta', j)\} = o(h)$ if $\theta' - \theta > 1$, hence $P(K_h > 1) = o(h)$. $K_h = 1$ when system switches from (θ, i) to $(\theta+1, i)$ with $i > \theta$. Therefore, according to the total probability formula:

$$P(K_h = 1) = \sum_{i>0} \pi_{\theta,i}.P\{(\theta, i) \to (\theta + 1, i)\}$$
$$= \left(1 - \sum_{i=0}^{\infty} \pi_{i,i}\right).(\alpha h + o(h))$$
$$= (\alpha h + o(h))\left(1 - \sum_{i=0}^{\infty} \frac{\pi_{0,0}.\rho^i}{i!}\right)$$
$$= (\alpha h + o(h))\left(1 - \pi_{0,0}.\sum_{i=0}^{\infty} \frac{.\rho^i}{i!}\right)$$
$$= (\alpha h + o(h))\left(1 - \frac{\sum_{i=0}^{\infty} \frac{\rho^i}{i!}}{\sum_{i=0}^{k} \sum_{j \geq i} \frac{\rho^i}{i!} \left(\frac{\lambda}{\lambda + \alpha}\right)^{j-i}}\right)$$
$$= (\alpha h + o(h))\left(1 - \frac{\sum_{i=0}^{\infty} \frac{\rho^i}{i!}}{\sum_{i=0}^{k} \sum_{l \geq 0} \frac{\rho^i}{i!} \left(\frac{\lambda}{\lambda + \alpha}\right)^{l}}\right)$$
$$= (\alpha h + o(h))\left(1 - \frac{1}{\sum_{l \geq 0} \left(\frac{\lambda}{\lambda + \alpha}\right)^{l}}\right)$$
$$= (\alpha h + o(h))\left(1 - \frac{1}{1/\left(1 - \frac{\lambda}{\lambda + \alpha}\right)}\right)$$
$$= \frac{\lambda}{\lambda + \alpha}(\alpha h + o(h))$$

Then we have:

$$E[K_h] = \frac{\lambda}{\lambda + \alpha}(\alpha h + o(h))$$

Dividing t into small enough time period h, assume K_i is the number of servers switched to ON state in time period i. We have:

$$E\left[K_i\right]=\frac{\lambda}{\lambda+\alpha}\left(\alpha h+o(h)\right)\approx\frac{\lambda}{\lambda+\alpha}\alpha h\,.$$

$$K=\sum K_i$$

Thus:

$$E\left[K\right]=E\left[\sum K_i\right]=\sum E\left[K_i\right]=\frac{\lambda}{\lambda+\alpha}\alpha h.\frac{t}{h}=\frac{\lambda\alpha t}{\lambda+\alpha}$$

The theorem 1 gives the formula for the expectation of the number of servers that switch from MIDDLE to ON successfully so that the average number of servers that must be switched to MIDDLE state is also calculated by formula (1). To satisfy this condition, our algorithm is that one OFF server is switched to MIDDLE state after time period τ given by:

$$\tau=\frac{t}{E\left[K\right]}=\frac{\lambda+\alpha}{\lambda\alpha}\quad(2)$$

The control algorithm to switch servers from OFF to MIDDLE:

Algorithm 1 Algorithm to turn on servers with ON/OFF/MIDDLE/∞/STAG model:

1. **while** true **do**
2. switch one server from OFF to MIDDLE
3. calculate time τ by the formula (2)
4. wait time period τ
5. **end while**

B. Control algorithm with ON/OFF/MIDDLE/∞ model

In this model, multiple servers can be simultaneously in SETUP mode. If there are always servers in MIDDLE state, the state transition of system is performed by Markov chain ON/OFF/∞ in Fig. 3. The limiting probabilities $\pi_{i,j}$ for Markov chain ON/OFF/∞ are given by the [11]:

$$\pi_{i,j}=\frac{\pi_{0,0}.\rho^i}{i!}\prod_{l=1}^{j-i}\frac{\lambda}{\lambda+l\alpha}\quad,i\ge 0,j\ge i$$

$$\pi_{0,0}=e^{-p}\left(\sum_{j=0}^{\infty}\prod_{l=1}^{j}\frac{\lambda}{\lambda+l\alpha}\right)^{-1}$$

Theorem 2**:** *in ON/OFF/MIDDLE/∞ model, the expectation of number of servers that are switched from MIDDLE state to ON state successfully in time period t is given by:*

$$E\left[K\right]=\alpha t.\frac{\sum_{k=0}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{\sum_{k=0}^{\infty}\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}\quad(3)$$

Proof. Similar to ON/OFF/MIDDLE/∞/STAG model:

$$P\left(K_h=1\right)=\sum_{j>i}\pi_{i,j}.P\left\{(i,j)\rightarrow(i+1,j)\right\}$$

and similar to lemma 2, in ON/OFF/MIDDLE/∞ model:

$$P\left\{(i,j)\rightarrow(i+1,j)\right\}=(j-i).\alpha h+o(h)\quad(j>i)$$

therefore:

$$P\left(K_h=1\right)=\sum_{j>i}\frac{\pi_{0,0}.\rho^i}{i!}\prod_{l=1}^{j-i}\frac{\lambda}{\lambda+l\alpha}\;.\left((j-i).\alpha h+o(h)\right)$$

$$\approx\pi_{0,0}.\sum_{j>i}\frac{\rho^i}{i!}\prod_{l=1}^{j-i}\frac{\lambda}{\lambda+l\alpha}\;.\left((j-i).\alpha h\right)$$

$$=\pi_{0,0}.\sum_{i=0}^{\infty}\frac{\rho^i}{i!}\sum_{j=i+1}^{\infty}\prod_{l=1}^{j-i}\frac{\lambda}{\lambda+l\alpha}\;.\left((j-i).\alpha h\right)$$

$$=\pi_{0,0}.\alpha h.\sum_{i=0}^{\infty}\frac{\rho^i}{i!}\sum_{k=1}^{\infty}\prod_{l=1}^{k}\frac{\lambda k}{\lambda+l\alpha}$$

$$=\pi_{0,0}.\alpha h.\left(\sum_{i=0}^{\infty}\frac{\rho^i}{i!}\right)\sum_{k=1}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}$$

$$=\alpha h.\frac{\left(\sum_{i=0}^{\infty}\frac{\rho^i}{i!}\right)\sum_{k=1}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{e^{p}\left(\sum_{j=0}^{\infty}\prod_{l=1}^{j}\frac{\lambda}{\lambda+l\alpha}\right)}$$

$$=\alpha h.\frac{\sum_{k=1}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{\sum_{k=0}^{\infty}\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}$$

Dividing t into small enough time period h, assume K_i is the number of servers switched to ON state in time period i. We have:

$$E\left[K\right]=\sum E[K_i]=\frac{t}{h}\alpha h.\frac{\sum_{k=0}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{\sum_{k=0}^{\infty}\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}$$

$$=\alpha t.\frac{\sum_{k=0}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{\sum_{k=0}^{\infty}\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}$$

The time period τ to switch one server from OFF to MIDDLE in ON/OFF/MIDDLE/∞ model is given by:

$$\tau=\frac{t}{E\left[K\right]}=\frac{1}{\alpha}.\frac{\sum_{k=0}^{\infty}\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}{\sum_{k=0}^{\infty}k.\prod_{l=1}^{k}\frac{\lambda}{\lambda+l\alpha}}\quad(4)$$

We have another algorithm for ON/OFF/MIDDLE/∞ model as follows:

Algorithm 2 Algorithm to turn on servers with ON/OFF/MIDDLE/∞ model

1. **while** true **do**
2. switch one server from OFF to MIDDLE
3. calculate time τ by the formula (4)
4. wait time period τ
5. **end while**

Note that with a complex systems, which require multiple queues model, we can use this strategy to determine time period to turn on servers for each queue, assuming the number of servers as unlimited.

V. EXPERIMENTS AND EVALUATIONS

In this section, we present the experiments and evaluations for our three-state model and proposed algorithms in view of improving QoS and energy benefits. We carry out three tests, namely:

- In order to prove the operations of our strategy, we determine the mean number of servers in MIDDLE state.
- We show the effectiveness of our model by comparing the mean waiting time for arriving jobs among three-state and two-state model. For the test, we experiment two model types: ON/OFF/MIDDLE/∞/STAG and ON/OFF/MIDDLE/∞ with/without staggered policy respectively.
- In proportion to each model type above, we continue to measure the power consumption to evaluate our proposed strategy. From the scope of this study, the quality of services depends on reducing the waiting time for arriving jobs while energy depends on power consumption of the system.

A. Experimental setup

To gain the results for evaluations, we use the CloudSim tool to simulate our model in cloud environment with a datacenter, servers, and job queue. The simulation is set up with a server farm (a.k.a data center) with 10000 hosts where each host is configured with 16 CPU cores (1200 MIPS/core), 32GB of RAM memory, and 1TB of storage. We consider an arriving job as the process of deploying a VM into the host farm. In all experiments, we fix μ=0.2 and $t_{OFF\rightarrow MIDDLE}$ = 200s.

B. Mean waiting time of arriving jobs

In the first tests, we aim at proving that our control algorithms work well and efficiency of three – state model. Fig. 4(a) and 4(b) present mean number of servers in MIDDLE state against traffic intensity λ with α = 0.1 and 0.01. Through experiments, we observe that this number increases with λ as we expected. Based on results, we can conclude that our algorithms worked well because some servers are kept in MIDDLE state.

In the second test, to evaluate our strategy in term of QoS, we compare the waiting time among three – state and two – state model when applying (Fig. 5) and non-applying (Fig. 6) the staggered setup policy. We make the following observations from these plots:

- In the case of using the staggered setup policy (ON/OFF/MIDDLE/∞/STAG): as can be seen from Fig. 5(a) and 5(b), the mean waiting time of three – state is significantly less than two – state model (ON/OFF/∞/STAG). For example, the difference between them is about 180s with α = 0.1 and approximates to $t_{OFF\rightarrow MIDDLE}$, these results manifest that in our three-state model, the waiting time for turning servers to MIDDLE state is eliminated. In this way, arriving jobs will be served immediately without booting process.
- In the case of without the staggered setup policy (ON/OFF/MIDDLE/∞): mean waiting time of three – state model also significantly decreases (Fig. 6(a) and 6(b)). The percentage decrease between ON/OFF/MIDDLE/∞ model and ON/OFF/∞ model is about 60% with α = 0.01 and 90% with α = 0.1.

In the both cases, the waiting time significantly decreased, and from the point of view QoS, it is improved by our strategy.

C. Power consumptions

In three – state model, the mean power consumption is given by the following formula:

$$P_{three-state\ model} = P_{ON} \,.\, E[n_{ON}] + P_{MIDDLE} \,.\, E[n_{MIDDLE}] + P_{SETUP} \,.\, E[n_{SETUP}] + + P_{OFF\rightarrow MIDDLE} \,.\, E[n_{OFF\rightarrow MIDDLE}]$$

where n_{ON}, n_{MIDDLE} , n_{SETUP} are number of ON servers, MIDDLE servers and SETUP servers respectively.

$n_{OFF\text{->}MIDDLE}$ is number of servers that are in turning process from OFF to MIDDLE state.

P_{ON} , P_{MIDDLE} , P_{SETUP} are power consumptions of a server in ON, MIDDLE and SETUP state respectively.

$P_{OFF\rightarrow MIDDLE}$ is power consumption of a server in turning process from OFF state to MIDDLE state.

The mean power consumption of two – state model is given by:

$$P_{\text{two – state model}} = P_{ON} \,.\, E[n_{ON}] + P_{OFF\rightarrow ON} \,.\, E[n_{OFF\rightarrow ON}]$$

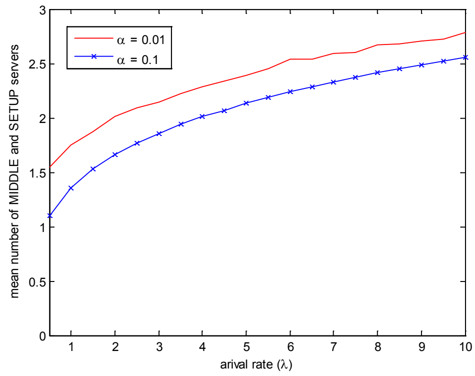
where $n_{OFF\rightarrow ON}$ is number of servers that are in turning process from OFF to ON state.

$P_{OFF\rightarrow ON}$ is power consumption of a server in turning process from OFF to ON state.

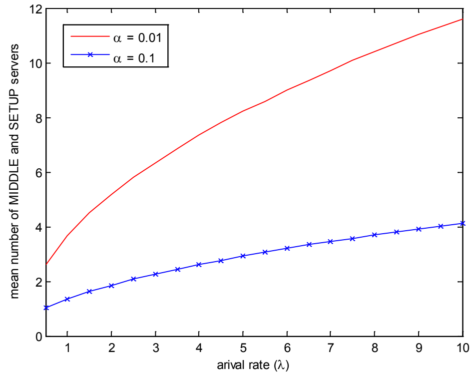
It is given as follows:

$$P_{OFF\rightarrow ON} = \frac{P_{OFF\rightarrow MIDDLE} * \bar{t}_{OFF\rightarrow MIDDLE} + P_{SETUP} * \bar{t}_{SETUP}}{\bar{t}_{OFF\rightarrow MIDDLE} + \bar{t}_{SETUP}}$$

where $\bar{t}_{OFF\rightarrow MIDDLE}, \bar{t}_{SETUP}$ are mean times of OFF to MIDDLE process and SETUP process respectively.

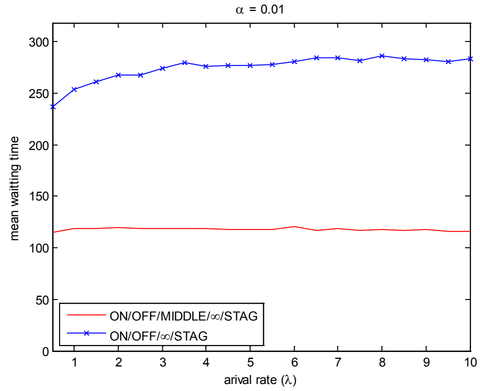


(a) ON/OFF/MIDDLE/∞/STAG model

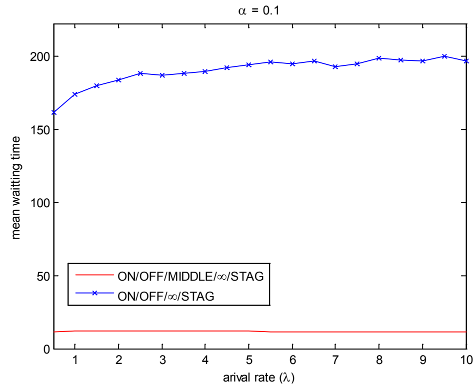


(b) ON/OFF/MIDDLE/∞ model

Fig. 4. Mean number of servers in MIDDLE state

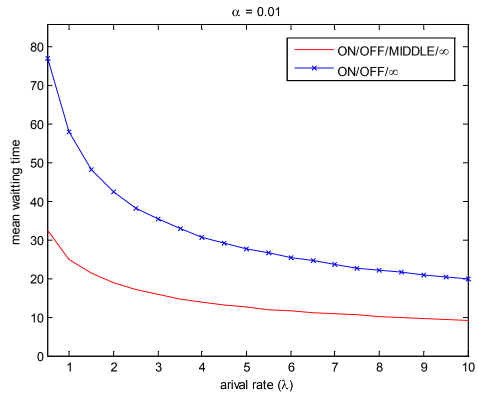


(a) α = 0.01

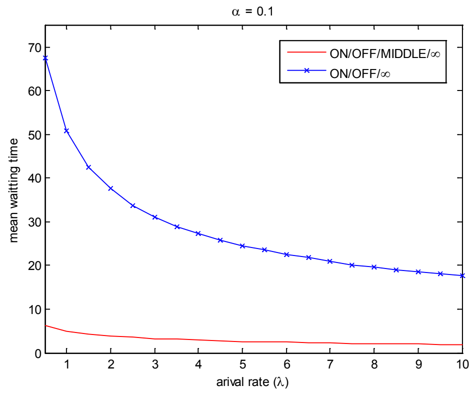


(b) α = 0.1

Fig. 5. Comparison of mean waiting time between three – state and two – state model (with staggered setup policy)



(a) α = 0.01



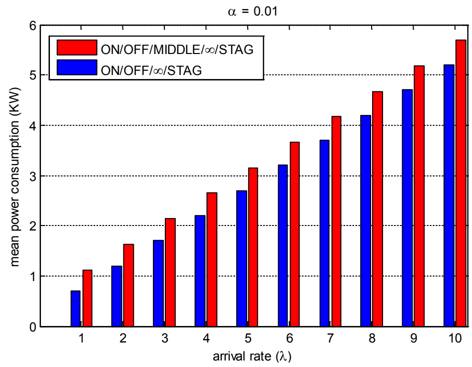
(b) α = 0.1

Fig. 6. Comparison of mean waiting time between three – state and two – state mode (without staggered setup policy)

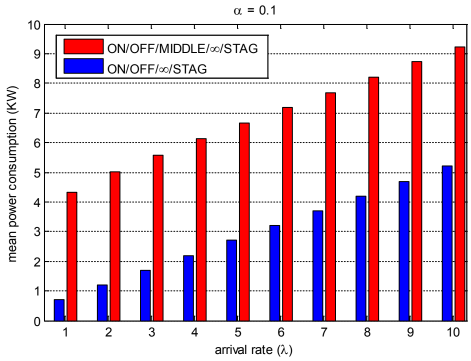
We assume that P_{ON} = 100 (W), $P_{SETUP} = 2.P_{ON}$, $P_{MIDDLE} = 0.6P_{ON}$ and experiment both policies.

2) ON/OFF/MIDDLE/∞/STAG model

For this test, we use staggered setup policy with ON/OFF/MIDDLE/∞/STAG model. Fig. 7 shows the test results and power consumption in comparison between the three – state and two – state model. We observe that the power consumption in the three – state is greater than two – state model. The power difference between two models is almost a constant when arrival rate increases in the both cases of α. Hence, the percentage increase of power consumption is less when arrival rate is greater (Fig. 8). Therefore, if the arrival rate is very large, the increase of power consumption is insignificant (it can be seen in the Fig. 8(a) and 8(b), the curves tend toward asymptotically to zero in both cases).

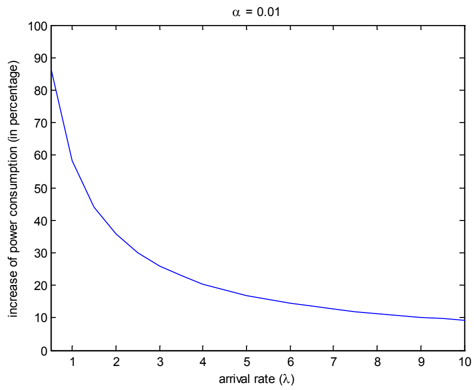


(a) α = 0.01

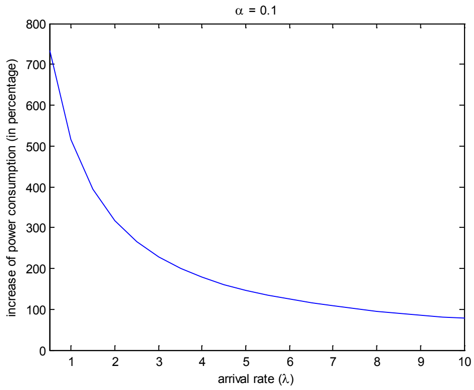


(b) α = 0.1

Fig. 7. Comparison of mean power consumption between three – state and two – state model (with staggered setup policy)

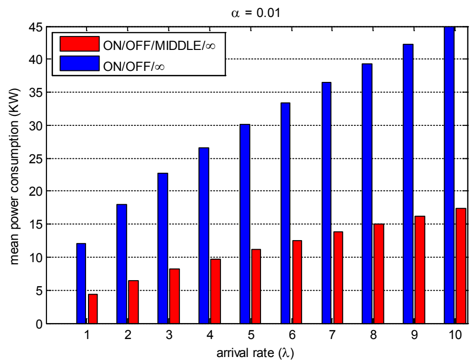


(a) α = 0.01

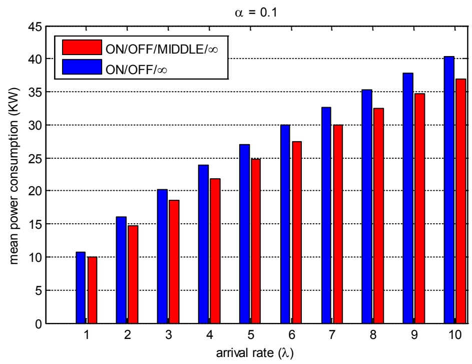


(b) α = 0.1

Fig. 8. Percentage increase of power consumption in *ON/OFF/MIDDLE/∞/STAG* model



(a) α = 0.01



(b) α = 0.1

Fig. 9. Comparison of mean power consumption between three – state and two – state model (without staggered setup policy)

3) ON/OFF/MIDDLE/∞ model

We also evaluate the ON/OFF/MIDDLE/∞ model with our strategy. Through achieved results, we realize that in the tests, the strategy not only decreases waiting time but also decreases power consumption. The explanation might be due to good performance of our strategy in comparison with two – state model.

For example, in two – state model, if there are three jobs arrive while the system has not idle ON servers. Then three OFF servers must turn into ON state. However, during these three OFF servers are in SETUP mode, some jobs have been completed and jobs queue size is less than three, so some servers in SETUP mode are turned off. This process will bring about wasting energy with the turned OFF servers. By contrast, in our model, the turning process is controlled by our strategy. Then the number of servers in SETUP mode are equal the number of servers that will be turned successfully into ON state. In this way, the number of servers in SETUP mode that are turned off is very small. The power consumption is shown in Fig. 9. The percentage decrease of power with our model is about 50-60% when α = 0.01 and about 10 % when α = 0.1.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presents a novel three-state model with control strategy, which allows managing servers in order to decrease deployment waiting time of cloud appliances when deploying on data centers. We also evaluate the energy waste phenomenon in cloud data centers with our proposed models. In this direction, we keep some servers in MIDDLE state using control algorithms, then we can eliminate the booting phase and decrease the waiting time.

We evaluate our model by performing simulations with CloudSim and the achieved experimental results show that our strategy consistently decreases the waiting time in comparison with traditional two-state model. In the term of quick service, we can improve the quality of services/application running on cloud data centers. Otherwise, we also conclude that the power consumption in ON/OFF/MIDDLE/∞ model is decreased and insignificantly increases in *ON/OFF/MIDDLE/∞/STAG* model.

We plan to extend this work in modeling the deployment of private, hybrid and federated cloud infrastructures (for both compute and storage services) in order to optimize service quality but still balance power usage. But firstly, because this work just stops in always keeping available servers in the MIDDLE state using proposed algorithms, in the near future, we have to determine which optimal number for the servers in the intermediate state. After that we will build the model with finite capacity of queue. Those are our further research subjects.

ACKNOWLEDGMENT

This research was carries out as part of the “Inferring the latent structures in big heterogeneous data” project, which is supported by American Air Force Office of Scientific Research (AFORS) and “Research on development and deployment of cloud-based bio-informatics services applying for metagenomics” project, which is supported by Vietnamese MOET.

REFERENCES

[1] Taisch, Marco, Bojan Stahl, and Giacomo Tavola. "ICT in manufacturing: Trends and challenges for 2020—An European view." In Industrial Informatics (INDIN), 2012 10th IEEE International Conference on, pp. 941-946. IEEE, 2012.

[2] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41, no. 1 (2011): 23-50.

[3] Long, Saiqin, Yuelong Zhao, and Wei Chen. "A three-phase energy-saving strategy for cloud storage systems." Journal of Systems and Software 87 (2014): 38-47.

[4] Phung-Duc, Tuan. "Server farms with batch arrival and staggered setup." In Proceedings of the Fifth Symposium on Information and Communication Technology, pp. 240-247. ACM, 2014.

[5] Baliga, Jayant, Robert WA Ayre, Kerry Hinton, and Rodney S. Tucker. "Green cloud computing: Balancing energy in processing, storage, and transport." Proceedings of the IEEE 99, no. 1 (2011): 149-167.

[6] Beloglazov, Anton, Jemal Abawajy, and Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." Future generation computer systems 28, no. 5 (2012): 755-768.

[7] Hamilton, James. "Cooperative expendable micro-slice servers (CEMS): low cost, low power servers for internet-scale services." In Conference on Innovative Data Systems Research (CIDR’09)(January 2009). 2009.

[8] Kim, Hyeong S., Dong In Shin, Young Jin Yu, Hyeonsang Eom, and Heon Y. Yeom. "Towards energy proportional cloud for data processing frameworks." In Proceedings of the First USENIX conference on Sustainable information technology, pp. 4-4. USENIX Association, 2010.

[9] Liu, Liang, Hao Wang, Xue Liu, Xing Jin, Wen Bo He, Qing Bo Wang, and Ying Chen. "GreenCloud: a new architecture for green data center." In Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, pp. 29-38. ACM, 2009.

[10] Milenkovic, Milan, Enrique Castro-Leon, and James R. Blakley. "Power-aware management in cloud data centers." In Cloud Computing, pp. 668-673. Springer Berlin Heidelberg, 2009.

[11] Gandhi, Anshul, Mor Harchol-Balter, and Ivo Adan. "Server farms with setup costs." Performance Evaluation 67, no. 11 (2010): 1123-1138.

[12] Gandhi, Anshul, Mor Harchol-Balter, and Ivo Adan. "Decomposition results for an M/M/k with staggered setup." ACM SIGMETRICS Performance Evaluation Review 38, no. 2 (2010): 48-50.

[13] Katz, Randy H. "Tech titans building boom." IEEE Spectrum 2, no. 46 (2009): 40-54.