A New Analytical Model for Calculating Elasticity in Cloud Computing

Assia Outamazirt, Kamel Barkaoui, Djamil Aissani

A New Analytical Model for Calculating Elasticity in Cloud Computing

Assia Outamazirt[1], Kamel Barkaoui[2], and Djamil Aïssani[3]

[1] Research Unit LaMOS of Bejaia University, Bejaia, Algeria
outamazirt.assia@gmail.com
[2] CEDRIC, CNAM, Paris, France
kamel.barkaoui@cnam.fr
[3] Research Unit LaMOS of Bejaia University, Bejaia, Algeria
djamil_aissani@hotmail.com

Abstract

One of the fundamental characteristics of Cloud Computing is its elasticity. It is about the ability to dynamically adapt computer resources consumption to workload while maintaining performance and quality of service. Most current industrial as well as academic solutions have limitations in terms of elasticity control, which affects the availability and performance of systems. In this paper, we propose a modeling of an elastic Cloud platform in terms of the markovian queuing model where the number of active servers depends on the current workload. A quantitative analysis of the steady state of our model allows to analyze and calculate the value of the elasticity in a precise way.

1 Introduction

The emergence of cloud computing has given rise to a new type of autonomic systems called elastic cloud systems, where elasticity is the main foundation.

Elasticity is one of the five characteristics of cloud computing [1]. It can be considered as a great advantage of the cloud computing paradigm that distinguishes it from other paradigms [2]. The concept of elasticity has its origin in the field of physics, where an object or a solid material is said elastic, if it is able to return to its original shape after being deformed. In cloud computing, elasticity is a key feature that dynamically adjusts the amount of allocated resources to meet changes in workload demands [3]. Indeed, we can compare the elasticity in cloud computing to the elasticity in the field of physics: an elastic cloud system (a cloud elastic platform) is a solid object; the resource (e.g., virtual machines (VMs)) utilization and quality of service (e.g. the average task response time) are properties and status of the platform. Dynamic workload (e.g., the number of service requests) is an external force. When the workload increases (respectively decreases), resource utilization increases (respectively decreases) and the quality of service decreases (respectively increases), i.e. the platform is deformed. To return to its original status, the platform must be able to adjust itself, for example by increasing (respectively decreasing) the number of virtual machines, so that the resources utilization and the quality of service can return to their original status [4].

In the literature, there are several definitions of elasticity in cloud computing (see [3, 5, 6, 7]). Some authors consider the concepts of scalability and elasticity as identical (e.g. [8, 9]), others as distinct (e.g. [7, 10]). According to [7], elasticity is defined as the degree to which a system is able to adapt to demands by provisioning and de-provisioning resources in an autonomic manner, such that the resources provided are consistent with the systems request. In [10], elasticity is also defined as the ability of a system to add and remove resources to adapt to the load variation in real time. On the other hand, scalability is defined as the ability of the system to sustain increasing workloads by making use of additional resources [7]. Scalability is time independent and it is similar to the provisioning state in elasticity but the time has no effect on the system [10]. In [11], scalability is defined as the ability of the system to meet the resource needs, without taking into account the speed, time, frequency

or granularity of its actions. The following equation summarizes the concept of elasticity in Cloud Computing [10]:

$$\text{Elasticity} = \underbrace{\text{scalability+automation}}_{\text{auto-scaling}} + \text{optimization},$$

this means that elasticity is based on scalability and can be considered as an automation of this concept, but it aims to optimize at best and as quickly as possible the resources at a given time. Another concept related to elasticity is efficiency, which characterizes how resources can be efficiently used. It is a measure linking demand capacity to services consumed over time [12]. There is also the concept of resilience which is the ability of a service to continue to operate despite the failure of one or more infrastructure elements [13].

From the point of view of the authors [4] and [14], the above definitions are classified in two categories. The first category includes definitions that are only qualitative, but not quantitative (e.g. [3, 7]) and the second category includes quantitative definitions, but not analytically treatable (e.g. [15]). Therefore, the authors presented a new quantitative and formal definition of elasticity in cloud computing. In other words, the authors considered that a cloud computing system is in (1) a normal state if the provided computing resources match the current workload; (2) an over-provisioning state if the provided computing resources exceed the current workload; (3) an under-provisioning state if the provided computing resources cannot handle the current workload. Thus, they defined the elasticity of a cloud computing platform with dynamically variable workload is the percentage of time (or, the probability) that the system is in the normal state and they have given the following equation for its numerical calculation:

$$\text{Elasticity} = \frac{T_{\text{normal}}}{T} = 1 - \frac{T_{\text{over}}+T_{\text{under}}}{T},$$

where $T = T_{\text{normal}} + T_{\text{over}} + T_{\text{under}}$ is a time period during which the system operate T_{normal} (respectively T_{over}, T_{under}) is the total time that the system is in the normal (respectively over-provisioning, under-provisioning).

If the system has been operating for a sufficiently long period of time and is in a stable state, then:

- $p_{\text{normal}} = \frac{T_{\text{normal}}}{T}$ is the probability that the system is in the normal state.
- $p_{\text{over}} = \frac{T_{\text{over}}}{T}$ is the probability that the system is in the over-provisioning state.
- $p_{\text{under}} = \frac{T_{\text{under}}}{T}$ is the probability that the system is in the under-provisioning state.
- Elasticity $= 1 - (p_{\text{over}} + p_{\text{under}})$.

Based on this new definition, the authors developed a Markov chain for computing the value of elasticity in the cloud computing.

In previous analytical modeling of cloud computing systems (see [16, 17, 18, 19, 20]), the authors considered non-markovian queuing systems in order to capture and analyze the dynamic behavior of the cloud computing environment. However, none of these models captured the real behavior of the elastic services. Specifically, all these models analyzed the cloud systems without taking into account the characteristic of elasticity and its analytical calculation.
In [4] and [14], the cloud platform has been traited as a markovian queuing system. In these works, the authors focused on the calculation of the value of elasticity in the cloud computing (as it is mentioned above). In fact, they considered an $M/M/c$ queue with an infinite number of servers that activate/deactivate at any time depending on the state of the system (normal, over-provisioning and under-provisioning states) to study an elastic cloud computing platform. However, it is important to minimize the number of active servers and to minimize the passages from on to off and vice versa in order to reduce the power consumption. Otherwise, the Markov chain developed in [4] and [14] can not lead to closed-form expressions for elasticity metrics, such as p_{normal}, p_{over}, p_{under}. According to the author of [4], this makes the analytical study of an elastic cloud computing platform very difficult. Therefore, in this paper, we develop a queuing model for the analytical modeling of cloud platform. This model allows us to obtain closed-form expressions for elasticity metrics. In addition, we perform a

quantitative steady-state analysis of our model in order to analyze and calculate the value of elasticity in a precise way.

To sum up, we treat a cloud platform as a queuing system. To model the elasticity in the cloud computing, we develop a new analytical model that we noted by $M/M/s+r/k$ queue. In this model, we consider that the number of active VMs when the system is in the normal state is s and the number of VMs that can be added is r. So, a new VM can be added if the current state is under-provisioning and the number of service requests waiting reaches a specific level. An active VM can be removed if the current state is over-provisioning and the number of service requests is less than a specific level. Then, to calculate the value of elasticity, we perform a quantitative steady-state analysis of our model. Finally, we present some numerical data just to show the impact of system parameters on elasticity.

The remainder of the paper is organized as follows. Section 2.1, describes our analytical model. Section 3 is devoted to present the quantitative steady-state analysis of our model. In Section 4, we compute the value of elasticity in the cloud computing and we present some numerical results. Finally, Section 5 concludes the work.

2 M/M/s+r/k queuing model

There are no analytical approaches that can analyze and predict elasticity in cloud computing in a precise way. In this section, we propose and study a queuing model for modeling, analyzing and calculating the elasticity in cloud computing.

2.1 M/M/s+r/k queuing model

A cloud computing platform is a multi-server system which has c (we note $c = s + r$) identical servers (VMs). In this paper, we consider a multi-server system as an $M/M/s+r/k$ queuing model (see Figure 1), where the arriving service requests are assumed to follow a Poisson process. The task execution times are i.i.d. exponential random variables with mean $\frac{1}{\mu}$. The $s + r$ servers are homogeneous and have identical service rate. A multi-server system maintains a queue with finite capacity for waiting tasks when all the $s + r$ servers are busy. The First-Come-First-served (FCFS) queueing discipline is adopted, and when an incoming service request finds the system full, it will be lost.

As in an elastic cloud platform, the number of servers adapts to the current workload, so we consider the queuing model $M/M/s + r/k$ with a variable number of servers which activate according to a specified rule and provide service following an exponential distribution (as it is mentioned above) with mean service rate μ_j $(= \mu)$ for jth $(j = 1, 2, ..., s + r)$ server.

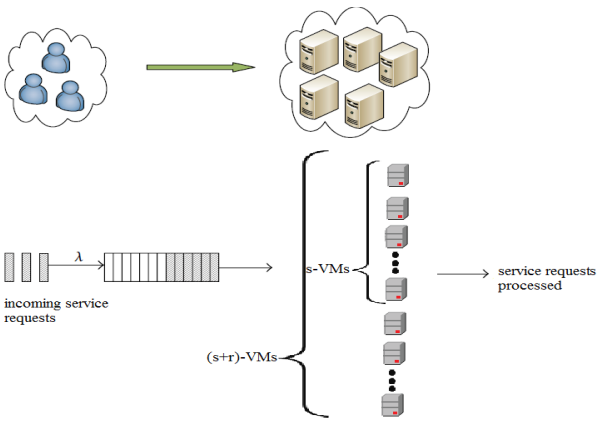

Figure 1: Modeling an elastic cloud computing platform as an $M/M/s + r/k$ queuing model.

2.2 Notations and hypotheses

We denote (a, b) a pair of integers which is used to determine the status of a system state with $a < b$ and $(i \geq 0)$ is the number of service requests present in the system at the instant t:

- A state is an over-provisioning state if: $0 \leq i \leq a$.
- A state is a normal state if: $a < i \leq b$.
- A state is an under-provisioning state if: $i > b$.

The number of servers employed (VMs used) depends upon the number of customers present in the system according to a threshold policy governed by the following rules:

- When we have the system is in a normal state, the first s servers are operating in the system.
- The time to initialize a new server is an exponential random variable with mean $\frac{1}{\alpha}$ (i.e., the VM start-up rate is α).
- As soon as there are L_1 service requests are waiting, the $(s+1)$-th server will start providing service, but it will be removed from the system if the number of service requests becomes less than L_1. In this case, $L_1 = b$.
- In general, when the number of service requests waiting reaches a specific level L_j , the $(s+j)$-th server $(j = 1, 2, 3, ..., r)$ will be available for service. As soon as the number of service requests is less than L_j , the $(s+j)$-th server will be removed from the system.
- Also, when the number of service requests is less than less than equal to a, the (j)-th server $(j = 1, 2, 3, ..., s)$ will be removed from the system.

The state-transition-rate diagram of our Markov chain is shown in the Figure 2.

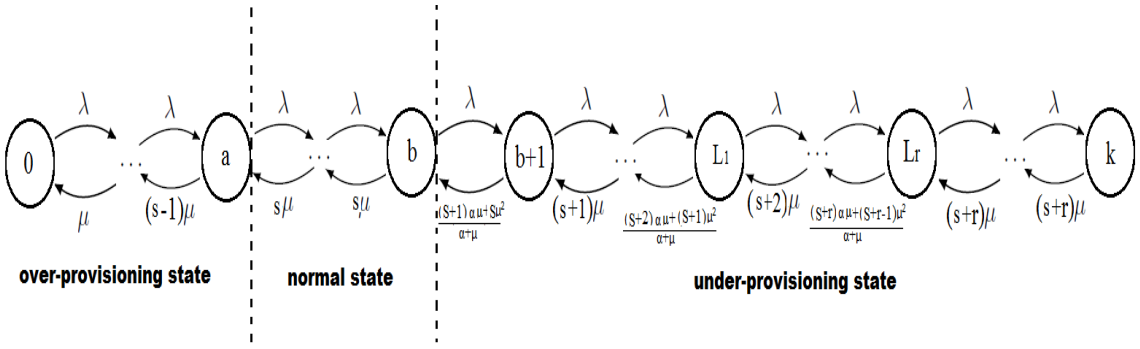

Figure 2: A state-transition-rate diagram.

3 Steady state probabilities

Since the $M/M/s+r/k$ queuing model has a finite number of states, then the Markov process describing the evolution of the number of service requests in the system is always ergodic, so the system is stable regardless of the arrival rate λ and the service rate μ.

3.1 Balance equations

We define π_i as the steady state probability that there are i service requests in the system. By using birth-and-death process, the steady-state equations for the finite capacity queuing system considered

in this paper are given as follows:

$$\lambda\pi_0 = \mu\pi_1. \tag{1}$$

$$(\lambda + min(s-1,i)\mu)\,\pi_i = \lambda\pi_{i-1} + min(s-1,i+1)\mu\pi_{i+1},\ \text{for } 0 \le i \le a. \tag{2}$$

$$(\lambda + s\mu)\pi_i = \lambda\pi_{i-1} + s\mu\pi_{i+1},\ \text{for } a < i \le b. \tag{3}$$

$$(\lambda + (s+j-1)\mu)\pi_{L_j-1} = \lambda\pi_{L_j-2} + \frac{(s+1)\alpha\mu + (s+j-1)\mu^2}{\alpha+\mu}\pi_{L_j},\ \text{for } j = \overline{1,r-1}. \tag{4}$$

$$\left(\lambda + \frac{(s+j)\alpha\mu + (s+j-1)\mu^2}{\alpha+\mu}\right)\pi_{L_j} = \lambda\pi_{L_j-1} + (s+j)\mu\pi_{L_j+1},\ \text{for } j = \overline{1,r}. \tag{5}$$

$$(\lambda + (s+j-1)\mu)\pi_i = \lambda\pi_{i-1} + (s+j-1)\mu\pi_{i+1},\ \text{for } j = \overline{2,r},\ L_{j-1}+1 \le i \le L_j - 2. \tag{6}$$

$$(\lambda + (s+r)\mu)\pi_i = \lambda\pi_{i-1} + (s+r)\mu\pi_{i+1},\ \text{for } L_r + 1 \le i < k. \tag{7}$$

$$(s+r)\mu\pi_k = \lambda\pi_{k-1}. \tag{8}$$

From equations (1) to (8), we can obtain the closed-form expressions for π_i, with $i = \overline{0,k}$. First, we derive π_i, for $i = \overline{1,k}$ in terms of π_0 as follows:

$$\pi_i = \begin{cases} \frac{1}{i!}\left(\frac{\lambda}{\mu}\right)^i \pi_0, \quad \text{for } \ 0 \le i \le a+1 \le s; \\ \\ \frac{1}{s^{i-s}s!}\left(\frac{\lambda}{\mu}\right)^i \pi_0, \quad \text{for } \ s+1 \le i \le b; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+j)^{i-L_j}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-j}\pi_0, \\ \text{for } j=1,\ i=L_1, \text{where } L_1 = b+1 \text{ and for } j=\overline{1,r-1},\ L_j+1 \le i \le L_{j+1}-1; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+j-1)^{(i-L_{j-1})-1}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-j}\pi_0, \\ \text{for } j=\overline{2,r},\quad i=L_j; \\ \\ \left(\frac{1}{s^{b-s}s!}\left(\frac{\lambda}{\mu}\right)^b\right)\left(\prod_{i'=1}^{r}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\right)\left(\prod_{\kappa=0}^{r-2}\frac{1}{(s+\kappa)^\kappa}\right)\left(\frac{1}{(s+r-1)^{(L_r-1)-L_{r-1}}}\right)\left(\frac{1}{(s+r)^{i-L_r}}\right)\left(\frac{\lambda}{\mu}\right)^{i-b-r}\pi_0, \\ \text{for } L_r+1 \le i \le k. \end{cases} \tag{9}$$

Writing $\rho = \frac{\lambda}{\mu}$, $\varphi = \frac{1}{s^{b-s}s!}$, $\phi = \prod_{i'=1}^{r}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}$, $\sigma = \prod_{\kappa=0}^{r-2}\frac{1}{(s+\kappa)^\kappa}$ and $\varrho = \frac{1}{(s+r-1)^{(L_r-1)-L_{r-1}}}$, π_0 can be obtained using the normalizing condition:

$$\sum_{i=0}^{s}\pi_i + \sum_{i=s+1}^{b}\pi_i + \pi_{b+1} + \sum_{j=1}^{r-1}\left[\sum_{i=L_j+1}^{L_{j+1}-1}\pi_i\right] + \sum_{j=2}^{r}\pi_{i=L_j} + \sum_{i=L_r+1}^{k}\pi_i = 1 \tag{10}$$

whith:

$$\sum_{i=0}^{s}\pi_i \quad = \quad \gamma_1\pi_0, \quad \gamma_1 = \begin{cases} \sum_{i=0}^{s}\frac{1}{i!}\rho^i, & if \quad \rho \ne 1; \\ \\ \sum_{i=0}^{s}\frac{1}{i!}, & if \quad \rho = 1. \end{cases}$$

$$\sum_{i=s+1}^{b}\pi_i \quad = \quad \gamma_2\pi_0, \quad \gamma_2 = \begin{cases} \sum\limits_{i=s+1}^{b}\frac{1}{s^{i-s}s!}\rho^i, & if \quad \rho \neq 1; \\ \\ \sum\limits_{i=s+1}^{b}\frac{1}{s^{i-s}s!}, & if \quad \rho = 1. \end{cases}$$

$$\pi_{b+1} \quad = \quad \gamma_3\pi_0, \quad \gamma_3 = \begin{cases} \varphi\rho^b\frac{\lambda(\alpha+\mu)}{(s+1)\alpha\mu+s\mu^2}, & if \quad \rho \neq 1; \\ \\ \varphi\frac{\lambda(\alpha+\mu)}{(s+1)\alpha\mu+s\mu^2}, & if \quad \rho = 1. \end{cases}$$

$$\sum_{j=1}^{r-1}\left[\sum_{i=L_j+1}^{L_{j+1}-1}\pi_i\right] \quad = \quad \gamma_4\pi_0, \gamma_4 = \begin{cases} \varphi\rho^b\sum\limits_{j=1}^{r-1}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\Big)\Big]\times \\ \Big(\frac{1}{s+j}\frac{1-(\frac{1}{s+j})^{L_{j+1}-1-L_j}}{1-\frac{1}{s+j}}\Big)\Big(\rho\frac{1-\rho^{L_{j+1}-1-L_j}}{1-\rho}\Big)\Big\}, if \quad \rho \neq 1; \\ \\ \varphi\sum\limits_{j=1}^{r-1}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-1}\frac{1}{(s+\kappa)^\kappa}\Big)\Big]\times \\ \Big(\frac{1}{s+j}\frac{1-(\frac{1}{s+j})^{L_{j+1}-1-L_j}}{1-\frac{1}{s+j}}\Big)\Big(L_{j+1}-1-L_j\Big)\Big\}, if \quad \rho = 1. \end{cases}$$

$$\sum_{j=2}^{r}\pi_{i=L_j} \quad = \quad \gamma_5\pi_0, \gamma_5 = \begin{cases} \varphi\rho^b\sum\limits_{j=2}^{r}\Big\{\Big[\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\Big)\Big(\frac{1}{(s+j-1)^{(L_j-L_{j-1})-1}}\Big)\Big]\times \\ \rho^{L_j-b-j}\Big\}, if \quad \rho \neq 1; \\ \\ \varphi\sum\limits_{j=2}^{r}\Big\{\Big(\prod\limits_{i'=1}^{j}\frac{\lambda(\alpha+\mu)}{(s+i')\alpha\mu+(s+i'-1)\mu^2}\Big)\Big(\prod\limits_{\kappa=0}^{j-2}\frac{1}{(s+\kappa)^\kappa}\Big)\Big(\frac{1}{(s+j-1)^{(L_j-L_{j-1})-1}}\Big)\Big\}, \\ if \quad \rho = 1. \end{cases}$$

$$\sum_{i=L_r+1}^{k}\pi_i \quad = \quad \gamma_6\pi_0, \quad \gamma_6 = \begin{cases} \varphi\rho^b\phi\sigma\varrho\Big(\frac{1}{s+r}\frac{1-(\frac{1}{s+r})^{K-L_r+1}}{1-\frac{1}{s+r}}\Big)\Big(\rho\frac{1-\rho^{K-L_r}}{1-\rho}\Big), & if \quad \rho \neq 1; \\ \\ \varphi\phi\sigma\varrho\Big(\frac{1}{s+r}\frac{1-(\frac{1}{s+r})^{K-L_r+1}}{1-\frac{1}{s+r}}\Big), & if \quad \rho = 1. \end{cases}$$

Therefore, equation (10) can be written as:

$$\pi_0 = \frac{1}{\gamma_1+\gamma_2+\gamma_3+\gamma_4+\gamma_5+\gamma_6} \tag{11}$$

3.2 Other System Metrics

Now, we find some more performance characteristics by using steady state probabilities as follows:

- Probability that the $(s+j)$-th $(j = 1, 3, ..., r-1)$ server is operating in the system:

$$P(s+j) = P(i \geq b+1) = (\gamma_3+\gamma_4+\gamma_5+\gamma_6)\pi_0.$$

- Probability that all servers are operating in the system:

$$P(s+r) = P(L_r + 1 \leq i \leq k) = \gamma_6 \pi_0.$$

- Probability that the 1st server is operating in the system:

$$P(1) = P(i \geq 1) = 1 - \pi_0.$$

4 Calculation of elasticity in cloud computing

From the steady state probabilities, we can also calculate the value of elasticity in Cloud Computing. The probability that the system is in the over-provisioning state is:

$$p_{\text{over}} = \sum_{i=0}^{a} \pi_i. \tag{12}$$

The probability that the system is in the normal state is:

$$p_{\text{normal}} = \sum_{i=a+1}^{b} \pi_i. \tag{13}$$

The probability that the system is in the under-provisioning state is:

$$p_{\text{under}} = \sum_{i=b+1}^{k} \pi_i. \tag{14}$$

Using the probabilities 12, 13 and 14, the value of the elasticity can be obtained:

$$\text{Elasticity} = \sum_{i=a+1}^{b} \pi_i = 1 - (\sum_{i=0}^{a} \pi_i + \sum_{i=b+1}^{k} \pi_i). \tag{15}$$

4.1 Graphic illustration

We present some numerical data just to show the impact of system parameters on elasticity. In Figure 3, we show p_{over}, p_{normal} and p_{under} as functions of the task arrival rate λ, where $s = 10$, $r = 1$, $a = 9$, $b = 13$, $k = 20$, $\mu = 1$ and $\alpha = 2$.

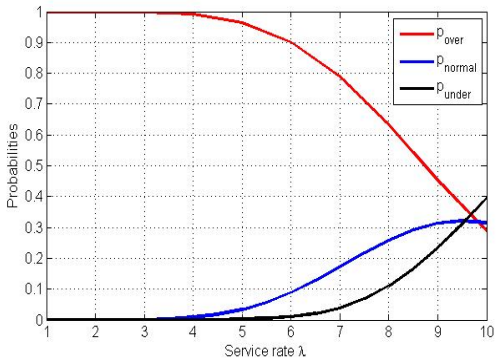

Figure 3: p_{over}, p_{normal} and p_{under} vs. λ.

In this Figure, we observe that when the arrival rate λ increases, p_{over} decreases, i.e., a large number of service requests reduces the probability of over-provisioning; p_{under} increases, i.e., more service requests increase the probability of under-provision and p_{normal} increases, i.e. the elasticity increases.

In Figure 4, we show p_{over}, p_{normal} and p_{under} as functions of the task service rate μ, where $s = 10$, $r = 1$, $a = 9$, $b = 13$, $k = 20$, $\lambda = 10$ and $\alpha = 2$.

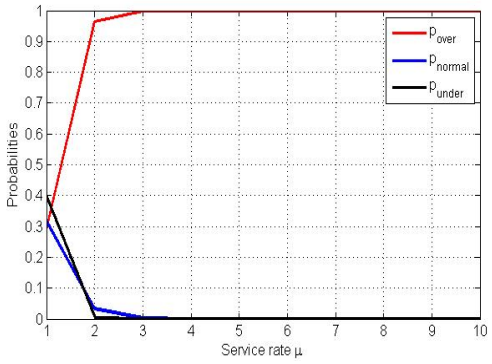

Figure 4: p_{over}, p_{normal} and p_{under} vs. μ.

In this Figure, we observe that as μ increases, p_{over} increases significantly, i.e., the increasing of service rate results in greater the probability of over-provisioning; p_{under} decreases, i.e., a higher service rate reduces the probability of under-provisioning and p_{normal} significantly decreases, i.e., the elasticity decreases significantly.

5 Conclusion

Elasticity is one of the five characteristics of Cloud Computing and can be considered a major advantage of cloud computing. However, there are no analytical approaches that can analyze and predict elasticity in a precise way. In this work, we considered a quantitative and formal definition of elasticity in cloud computing, and we developed an analytical model to study elasticity by treating a cloud platform as an $M/M/s + r/k$ queuing system with the number of active servers dependent on number of tasks present in the system. To analyze and calculate the value of elasticity in a precise way, we performed a quantitative steady-state analysis of our model. The model proposed in this paper can enable cloud service providers and its users to obtain accurate value of elasticity in cloud computing by using a few essential parameters of a cloud platform, such as the start-up rate of virtual machines, the arrival rate of cloud service requests and the service rate.

For the future, we plan to extend the model proposed in this paper to consider a doubly stochastic Poisson process to model the arrival process; this takes into account the arrival rate variation of cloud service user requests over time. We also plan to extend the proposed model to consider a general service time distribution to substantiate the actual service time of the cloud service request

References

[1] Peter Mell and Timothy Grance *"The NIST Definition of Cloud Computing"*, Special Publication 800-145, 2011. [online] https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published.

[2] Schahram Dustdar, Yike Guo, Benjamin Satzger and Hong-Linh Truong *"Principles of Elastic Processes"*, IEEE Internet Computing, 15(5), pp. 66-71, 2011.

[3] Lee Badger, Tim Grance, RobertPatt-Corner and JeffVoas *Draft cloud computing synopsis and recommendations*, NIST special publication 800-146, pp. 1-84, 2011.

[4] Keqin Li *Quantitative modeling and analytical calculation of elasticity in cloud computing*, IEEE Transactions on Cloud Computing, XX(YY), pp. 01-14, 2017.

[5] Emanuel Ferreira Coutinho, Flvio Rubens de Carvalho Sousa, Paulo Antonio Leal Rego, Danielo Gonalves Gomes and Jos Neuman de Souza *Elasticity in Cloud Computing: a Survey*, Annals of Telecommunications, 70(7-8), pp. 289-309, 2015.

[6] Guilherme Galante and Luis Carlos E. de Bona *A Survey on Cloud Computing Elasticity*, Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing, UCC 12. Washington, DC, USA: IEEE Computer Society, pp. 263-270, 2012.

[7] Nikolas Roman Herbst, Samuel Kounev, Ralf Reussner *Elasticity in Cloud Computing: What It Is, and What It Is Not*, Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13), San Jose, CA: USENIX, pp. 23-27, 2013.

[8] Doaa M. Shawky and Ahmed F. Ali *Defining a measure of cloud computing elasticity*, 2012 1st International Conference on Systems and Computer Science (ICSCS), Lille, France, pp. 1-5, 2012.

[9] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia *A view of cloud*, Communications of the ACM, 53(4), pp. 50-58, 2010.

[10] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah and Philippe Merle *Elasticity in Cloud Computing: State of the Art and Research Challenges*, IEEE Transactions on services computing, 11(2), pp. 430-447, 2017.

[11] Emanuel Ferreira Coutinho, Danielo Gonalves Gomes and Jos Neuman de Souza *An analysis of elasticity in cloud computing environments based on allocation time and resources*, 2nd IEEE Latin American Conference on Cloud Computing and Communications, Maceio, Brazil, pp. 7-12, 2013.

[12] Sebastian Lehrig, Hendrik Eikerling and Steffen Becker *Scalability, Elasticity, and Efficiency in Cloud Computing: a Systematic Literature Review of Definitions and Metrics*, Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures, QoSA15. New York, NY, USA: ACM, pp. 83-92, 2015.

[13] Rahul Ghosh, Dan Dongseong Kim and Kishor S Trivedi *System resiliency quantification using non-state-space and state-space analytic models*, Reliability Engineering and System Safety, 116, pp. 109-125, 2013.

[14] Wei Ai, Kenli Li, Shenglin Lan, Fan Zhang, Jing Mei, Keqin Li and Rajkumar Buyya *On Elasticity Measurement in Cloud Computing*, Hindawi Publishing Corporation, 2016, pp. 01-14, 2016.

[15] Sadeka Islam, Kevin Lee, Alan Fekete and Anna Liu *How a consumer can measure elasticity for cloud platforms*, Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, Boston, Massachusetts, USA, pp. 85-96, 2012.

[16] Assia Outamazirt, Mohamed Escheikh, Djamil Aïssani, Kamel Barkaoui and Ouiza Lekadir *Stochastic Model for Cloud Data Center with M/G/c/c+r Queue*, Proceedings of the 10th International Workshop on VECoS2016, Tunis, Tunisia, 1689, pp. 77-84, 2016.

[17] Assia Outamazirt, Mohamed Escheikh, Djamil Aïssani, Kamel Barkaoui and Ouiza Lekadir *Performance analysis of the M/G/c/c+r queuing system for cloud computing data centres*, Int. J. Critical Computer-Based Systems, 8(3/4), pp. 234-257, 2018.

[18] Assia Outamazirt, Kamel Barkaoui and Djamil Aïssani *Maximizing profit in cloud computing using M/G/c/k queuing model*, International Symposium on Programming and Systems (ISPS'2018), IEEE,Algiers, Algeria, pp. 1-6, 2018.

[19] Hamzeh Khazaei, Jelena Misic and Vojislav B. Misic *Performance analysis of cloud computing centers using M/G/m/m+r queueing systems*, Journal of IEEE Transactions on Parallel and Distributed Systems 23(5), pp. 936-943, 2012.

[20] Xiaolin Chang, Bin Wang, Jogesh K. Muppala, Jiqiang Liu *Modeling active virtual machines on IaaS clouds using an M/G/m/m+K queue*, Journal of IEEE Transactions on Services Computing 9(3), pp. 408-420, 2016.