Stochastic Modelling and Analysis of Cloud Computing Data Center

Said El Kafhali
Computer, Networks, Mobility and Modeling laboratory
National School of Applied Sciences
Hassan 1^{st} Univ, Settat, Morocco
Email: said.elkafhali@uhp.ac.ma

Khaled Salah
Electrical and Computer Engineering Department
Khalifa University of Science
Technology and Research (KUSTAR), Abu Dhabi, UAE
Email: khaled.salah@kustar.ac.ae

***Abstract*—Cloud data centers (CDC) are an integral part of today's internet services. Enterprises and Businesses around the world rely heavily on data centers for their daily computation and IT operations. In fact, every time we search for an information on the internet, or we use an application on our smartphones, we access data centers. In CDC, most compute resources are represented as virtual machines (VMs) which are mapped into physical machines (PMs). Performance is often is a key metric for CDC. This paper presents a stochastic model based on queuing theory to aid in studying and analyzing performance in CDC. CDC platforms are modeled with an open queuing system that can be used to estimate the expected Quality of Service (QoS) guarantees the cloud can offer. We give numerical examples to show how the model estimates the number of required VM instances needed to satisfy a given the QoS parameters. In particular, we plot the response time, drop rate and CPU utilization while varying the incoming request arrival rate, and for different number of VM instances. We cross-validate our analytical model using a DES (Discrete Event Simulator). Our analysis and simulation results show that the proposed model is able to estimate the number of VMs needed to achieve QoS targets when varying the arrival request rate.**

***Keywords*—Cloud Data Center, Queueing Theory, Performance Analysis**

I. Introduction

A cloud computing infrastructures consist of services that are offered and delivered through a data center, that can be accessed from a web browser anywhere in the world [1]. Cloud computing providers offer computing resources (servers, storage, networks, development platforms, and applications) to users either elastically or dynamically, according to user-demand and form of payment [2]. Cloud computing addresses three main areas of operation such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [3]. Compared to SaaS and PaaS, IaaS it is a form of cloud computing that provides virtualized computing resources over the Internet [4]. In a PaaS model, a cloud provider delivers applications, and other development, while providing cloud components to software over Internet [5]. On the other hand, SaaS uses the web to deliver applications that are managed by a third-party vendor and whose interface is accessed on the clients' side over Internet. Because of the web delivery model, SaaS eliminates the need for organizations to install and run applications on their own computers or in their own data centers [6].

In cloud computing, any efficient resource management scheme would seek to allocated computing, storage, networking and energy resources to a set of applications, in a manner that satisfies the QoS of the cloud-hosted application or service while minimizing the cost associated with using physical infrastructure resources of the CDC [7]. Performance evaluation of QoS cloud centers is a very crucial research task which becomes difficult due to the dynamic behavior of cloud environments and variability of client demand [8]. A typical QoS metric specifies a set of critical performance parameters, which may include mean response time, mean drop rate, mean request queuing length, mean waiting time, mean throughput, and blocking probability.

Queueing theory has been regularly used in the literature to study and estimate QoS parameters in cloud environments [9]. For example, in [10], an analytical performance model of single VM live migration is evaluated which show that an effective live migration can reduce service rejection probability scenarios and total delay. In [11], a statistical distribution is proposed with the extension of CloudSim, to overcome the virtualization layer overhead, insufficient trace logs and complex workload in cloud computing resource usage. Guo *et al.* [12] studied the performance of M/M/m queuing model to optimize the performance of services in an on-demand service in cloud computing. Ghosh *et al.* [13] proposed an interacting stochastic model approach to overcome the performance quantification of a large-scale virtualized IaaS CDC. Sun *et al.* [14] developed a new technique for efficient live migration of multiple VMs based on queuing models. To evaluate the blocking rate, they modeled the arrival request using the M/M/C/C queuing model. Similarly, to evaluate the average waiting queue length, the average waiting time, the average queue length and the average sojourn time of each migration request, they modeled the arrival request using the M/M/C queuing model. Salah *et al.* [15] proposed an analytical model, based on Markov chains, for cloud-hosted applications and services. The proposed model predict the number of VMs instances needed to satisfy a given Service Level Objective (SLO) performance requirement such as response time, throughput, or request loss probability.

978-1-5090-3672-1/17/$31.00 ©2017 IEEE

In this paper, we show how queueing theory can be used in estimating key performance metrics in CDC. The main contributions of this paper can be summarized as follows:

- A queuing analytical model for CDC is presented, and analytical equations are derived for key performance metrics.
- The analytical model is cross-validated with a simulation model based on Java Modeling Tools (JMT) simulator.
- Numerical examples are given to show how our model can be used to satisfy key QoS parameters, and also to determine the required number of VMs needed under variable workload conditions.

The rest of the paper is organized as follows: The proposed Data Center model is presented in section II. Section III presents the analytical model for the proposed model. Section IV presents the numerical and simulation results. Finally, section V is devoted to the conclusion.

II. Cloud Data Center Model

We consider a large data center in a cloud system composed of PMs with each PM hosting many VMs, as illustrated in Figure 1. Indeed, large data centers of Google, Microsoft, Yahoo and Amazon etc. contains tens of thousands of PMs [16]. Each VM is allocated to one PM, where as a PM can be allocated multiple VMs through a hypervisor. VM it is a software that can run its own operating system and applications just like an operating system on a physical computer. The Load Balancing (LB) server maintains the schedule queue to receive all requests from clients. A service request from a client is transmitted to the LB server running a service application [17], associated with an SLA. Client requests are submitted to a LB queue and then processed on the First-In First-Out (FIFO) basis. The arrivals of requests follow a Poisson process. Therefore, the inter-arrival times between successive arriving requests are independent and exponentially distributed random variables with rate $\frac{1}{\lambda}$. Queued requests are distributed to different PMs and the scheduling rate depends on the LB server capacity. We assume that the service time of the LB server queue is exponentially distributed with mean service time $\frac{1}{\mu}$. Thus, the LB server queue is modeled as an M/M/1/C queuing system [18]. Such a queue has a finite size C; so, an arriving request can be rejected if it finds the buffer full, otherwise it will be accepted.

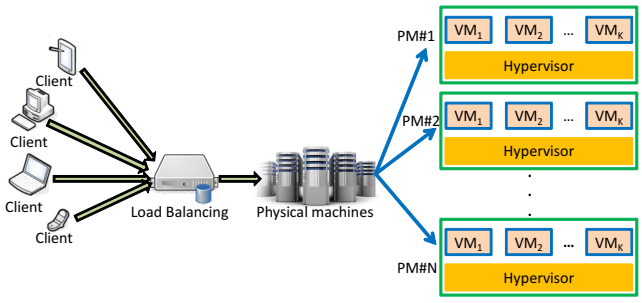

Fig. 1. The Architecture of the Cloud Data Center system

We suppose there are N PMs in the CDC. The requests are evenly distributed by the LB server to each PM with the same probability $\frac{1}{N}$. Consequently, the arrivals of requests at each PM follow a Poisson process with arrival rate $\frac{\lambda}{N}$. We assume that all PMs are homogeneous service. Therefore, the server time of each PM is exponentially distributed with mean service time server $\frac{1}{r}$. We model each PM in the CDC as an M/M/m/K ($K > m$) queuing system [18]. Each PM may run up to m VMs, and K is the maximum number of the requests in the PM. We assume that an inter-arrival time of requests and service times are exponentially distributed. If the queue reaches its maximum limit, the extra requests are dropped. If the resource is available then request is accepted and routed by the LB to the corresponding VM. It's assumed that all VMs allow the same web services with the same functionality to clients via Internet. As the maximum number of requests in the system is C, we assume that C is equal to the number of PM times the number of VMs that can be allocated to a single PM. So, C is given by the following formula

$$C = N \times K \tag{1}$$

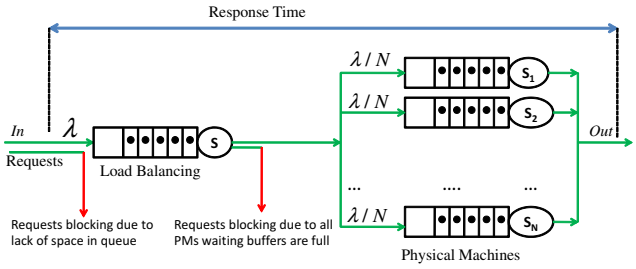

Fig. 2. Queuing Model

The queuing model of the CDC is shown in Figure 2. An arriving request that finds the LB queue full will be dropped. Once the request is admitted to the LB queue, it must wait until the LB processes it on a FIFO basis. We assume we operation in a homogenous cloud environment where by all PMs are equal in processing capacity and sizes with and each PM has a waiting buffer which can occupy, containing at most (K-1) requests. If a PM with free buffer spaces is found, the request is put into the PM waiting queue for further service. If all PMs waiting buffers are full, then the request is dropped. Thus, a client request may be assigned to a PM, dropped because all PMs waiting buffers are full, or dropped due to the insufficient LB buffer space.

III. Theoretical Analysis

A. Load Balancing Queueing Model

The LB is modeled as an M/M/1/C queue. The maximum number of requests in the system is C, which implies a maximum queue length of C-1. An arriving request enters the queue if it finds less than C requests in the system and is lost otherwise. Using the balanced equations and the normalization

condition, we obtain the steady-state probability of k requests in the system

$$\pi_k = \frac{1-a}{1-a^{C+1}} a^k \tag{2}$$

where $a = \frac{\lambda}{\mu}$ denotes the offered load in LB server.
The mean throughput service X is given by

$$X = \lambda \frac{1-a^C}{1-a^{C+1}} \tag{3}$$

The mean number of requests in the LB is

$$E(k) = \sum_{k=1}^{C} k\pi_k = \frac{a}{1-a} \frac{1-(C+1)a^C + Ca^{C+1}}{1-a^{C+1}} \tag{4}$$

We can obtain blocking probability due to lack of space in LB queue

$$P_{loss} = \pi_C = \frac{1-a}{1-a^{C+1}} a^C \tag{5}$$

We use Little's law formula [19] to obtain the mean response time of requests at the LB as

$$E(r_b) = \frac{E(k)}{X} \tag{6}$$

B. Cloud Data Center Queueing Model

We model access to each PM as an M/M/m/K ($K > m$) queue. The maximum number of requests in the PM is K, and each PM may run up to m VMs. Let us define the state of a PM as the total number of VMs in the PM. Figure 3 exhibits the transition diagram for the new request in a single PM.

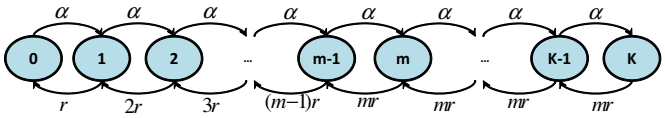

Fig. 3. Continuous Time Markov chain for new request in a single-PM

Let $\pi_i(n)$ denote the steady-state probability of having n requests in $PM_i (i = 1, 2, ..., N)$. Using the balanced equations and we note $\alpha = \frac{\lambda}{N}$, we find that

$$\pi_i(n) = \begin{cases} \frac{\pi_0(\alpha)^n}{n! r^n}, \forall n < m \\ \frac{\pi_0(\alpha)^n}{m! m^{n-m} r^n}, \forall n \geq m \end{cases} \tag{7}$$

where π_0 is given by

$$\pi_0 = (1 + \frac{(m\rho)^m (1-\rho^{k+1-m})}{m!(1-\rho)} + \sum_{i=1}^{m-1} \frac{(m\rho)^i}{i!})^{-1} \tag{8}$$

where ρ denotes the offered load and is expressed as

$$\rho = \frac{\alpha}{mr} \tag{9}$$

The effective rate of arrivals, λ_{eff}, to the service is given by

$$\lambda_{eff} = \alpha(1 - \pi_i(K)) \tag{10}$$

We then deduce the performances parameters as follows. First, the rate of loss can be obtained as follows

$$v = \alpha \pi_i(K) \tag{11}$$

The CPU utilization of each VM instance can be expressed as follows

$$U_{VM} = \frac{\lambda_{eff}}{mr} = \rho(1 - \pi_i(K)) \tag{12}$$

We compute $E(n)$, the mean number of requests in the PM_i as

$$E(n) = \sum_{j=1}^{K} j\pi_i(j) \tag{13}$$

The mean number of requests waiting in the PM_i can be obtained from

$$E(n_w) = \sum_{j=m+1}^{K} (j-m)\pi_i(j) \tag{14}$$

Finally, we use Little's formula [19] to obtain the mean response time and mean waiting time in the PM_i as follows

$$E(r_c) = \frac{E(n)}{\alpha(1 - \pi_i(K))}, E(w) = \frac{E(n_w)}{\alpha(1 - \pi_i(K))} \tag{15}$$

Based on Figure 2, we compute the response time T for a request on a system, the probability that a request is served below a specific time t is given by

$$T = E(rb) + E(rc) \tag{16}$$

IV. Simulation and Numerical Results

There are a number of available network simulation tools. Some of these simulators are designed specifically for cloud environments (e.g., CloudSim, iCanCloud, EMUSIM, MDCSim) [20], [21], and some are generic in natures (e.g., OPNET, NS, OMNeT, J-Sim) [22]. All of these available simulators did not have the capabilities to capture accurately the internal behavior and dynamics of the CDC. For this reason, we choose the Java Modeling Tools (JMT) to implement the performance of the proposed model [23], [24].

Simulation Environment: We consider a scenario of a small scale CDC where the CDC has 10 PMs. The average request arrival rate to the system is 1000 requests per second. The service times of each request in the LB are exponentially distributed with an average of 0.0001 seconds. Two types of commonly requests are considered; namely: (1) web requests, and (2) database requests. The average service time at the PMs of a web requests is 10 miliseconds while of a database requests require on average 15 miliseconds. The maximum number of requests in the system is 300, and the maximum number of the requests in the PM is 30. Note that, we

vary some of these parameters depending on the simulation scenarios whereas the others remain fixed.

Performance Analysis: The results obtained from simulation are represented by the black circles, whereas the curves represented by lines are those of analysis. The numerical results are shown in Figures 4 and 5. We have analyzed the web and database performance curves using multiple VM instances as function of requests arrival rate. Performance curves as those of the response time, drop rate and CPU utilization are plotted in the figures.

Figure 4 exhibits performance results obtained from simulation and analysis for web requests; whereas, Figure 5 exhibits those results for database requests. All of these figures depict clearly the impact of the number of VMs on key performance metrics. Specifically, Figure 4 (b) shows that when the requests arrival rate reaches the 2200 requests per second, we can see the impact of the number of the VM instances on the drop rate measure. The response time variation versus requests arrival rate depicted in Figure 4 (a) demonstrates that from 1600 requests per second, as the number of VM instances decreases, the response time increases and reaches 0.14 second. It is obvious that as the number of VMs increases the drop rate decreases. However in the database performance case (Figures 5(a) and (b)), when the arrival rate reaches the 1300 requests per second, we can observe that as the requests arrival rate and the number of VMs decrease, the response time and drop rate increase. Figure 4(c) exhibits the CPU utilization for multiple numbers of VMs. We observe that for the three values of number of VMs (namely: 21, 22 and 23), the curves are approximately linear, consequently as the number of VMs increases as the CPU utilization increases. As opposed to scenario of having 20 VMs, the CPU utilization parameter in these cases starts from 50% for 1000 requests per second and reaches 100% from 2200 requests per second and more.

Figures 5(a) and (b) show that when we increase the number of VMs and the requests arrival rate, the response time and the drop rate metrics decrease. Considering the CPU utilization measure depicted in Figure 5(c), we observe that when we have just twenty VMs, the CPU utilization percentage is higher and it reaches 100% from 1500 requests per second and more. For the three values of VMs (21, 22 and 23), the three curves are similar and when the requests arrival rate tends to 2300 requests per second, the CPU utilization reaches 100% value.

V. Conclusion

In this paper, we presented an analytical model that can be used in studying the performance of CDC and is able to estimate accurately the needed number VMs to achieve a target QoS metric. We have considered the typical architecture in which a CDC houses a collection of PMs that will be used to run VMs and also LBs. Scenarios were presented to illustrate the usefulness of our analytical model-specifically, in determining the impact of the number of allocated VMs on key performance and QoS parameters which included response time, drop rate and CPU utilization. We cross-validated the results obtained from our analytical model with simulation results obtained from the popular JMT simulator. The simulation and the analysis results are in agreement and thus implying that, our analytical model is correct. As a future work, we plan to conduct experimental work of an elastic-scaling mechanism on a real-world CDC in which our analytical formulas derived in this paper are used to scale resources automatically to meet QoS targets in accordance to variable workloads.

References

[1] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," *Cloud computing: Principles and paradigms*, pp. 1–44, 2011.

[2] B. Furht, "Cloud computing fundamentals," in *Handbook of cloud computing*. Springer, 2010, pp. 3–19.

[3] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[4] W. Huang, A. Ganjali, B. H. Kim, S. Oh, and D. Lie, "The state of public infrastructure-as-a-service cloud security," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 68, 2015.

[5] A. F. Alam, A. Soltanian, S. Yangui, M. A. Salahuddin, R. Glitho, and H. Elbiaze, "A cloud platform-as-a-service for multimedia conferencing service provisioning," in *21st IEEE Symposium on Computers and Communications*. IEEE, 2016.

[6] J. Schäfer and H. Lichter, "Changes in requirements engineering after migrating to the software as a service model," *Full-scale Software Engineering/Current Trends in Release Engineering*, p. 25, 2016.

[7] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," *Journal of Network and Systems Management*, vol. 23, no. 3, pp. 567–619, 2015.

[8] W. Kim, "Cloud computing: Today and tomorrow." *Journal of object technology*, vol. 8, no. 1, pp. 65–72, 2009.

[9] H. Chen and D. D. Yao, *Fundamentals of queueing networks: Performance, asymptotics, and optimization*. Springer Science & Business Media, 2013, vol. 46.

[10] H. Khazaei, J. Mišić, and V. B. Mišić, "Performance of an iaas cloud with live migration of virtual machines," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 9-13 Dec. 2013, pp. 2289–2293.

[11] D. Magalhães, R. N. Calheiros, R. Buyya, and D. G. Gomes, "Workload modeling for resource usage analysis and simulation in cloud computing," *Computers & Electrical Engineering*, vol. 47, pp. 69–81, 2015.

[12] L. Guo, T. Yan, S. Zhao, and C. Jiang, "Dynamic performance optimization for cloud computing using m/m/m queueing system," *Journal of Applied Mathematics*, vol. 2014, 2014.

[13] R. Ghosh, F. Longo, V. K. Naik, and K. S. Trivedi, "Modeling and performance analysis of large scale iaas clouds," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1216–1234, 2013.

[14] G. Sun, D. Liao, V. Anand, D. Zhao, and H. Yu, "A new technique for efficient live migration of multiple virtual machines," *Future Generation Computer Systems*, vol. 55, pp. 74–86, 2016.

[15] K. Salah, K. Elbadawi, and R. Boutaba, "An analytical model for estimating cloud resources of elastic services," *Journal of Network and Systems Management*, vol. 24, no. 2, pp. 285–308, 2016.

[16] R. H. Katz, "Tech titans building boom," *IEEE Spectrum*, vol. 2, no. 46, pp. 40–54, 2009.

[17] J. Vilaplana, F. Solsona, I. Teixidó, J. Mateo, F. Abella, and J. Rius, "A queuing theory model for cloud computing," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 492–507, 2014.

[18] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006.

[19] R. Nelson, *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*. Springer Science & Business Media, 2013.

[20] K. Bahwaireth, E. Benkhelifa, Y. Jararweh, M. A. Tawalbeh *et al.*, "Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 1–14, 2016.

[21] W. Tian, M. Xu, A. Chen, G. Li, X. Wang, and Y. Chen, "Open-source simulators for cloud computing: Comparative study and challenging issues," *Simulation Modelling Practice and Theory*, vol. 58, pp. 239–254, 2015.

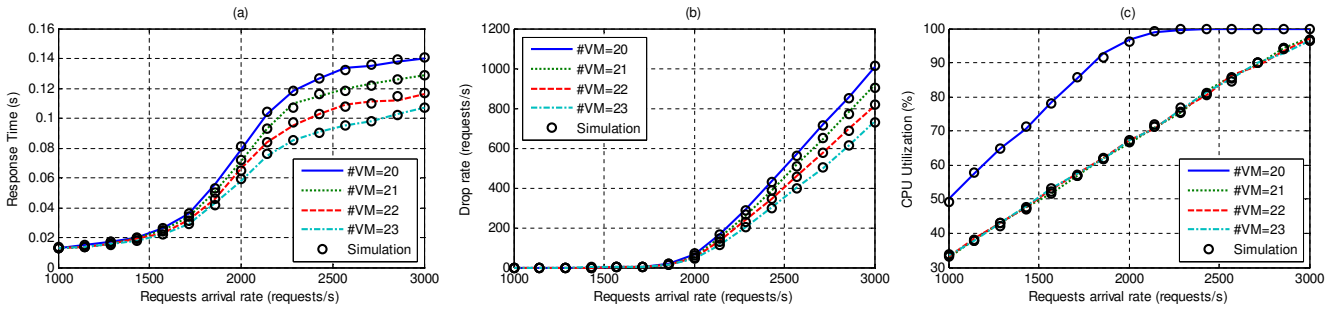

Fig. 4. Web performance curves using multiple VM instances as functions of requests arrival rate

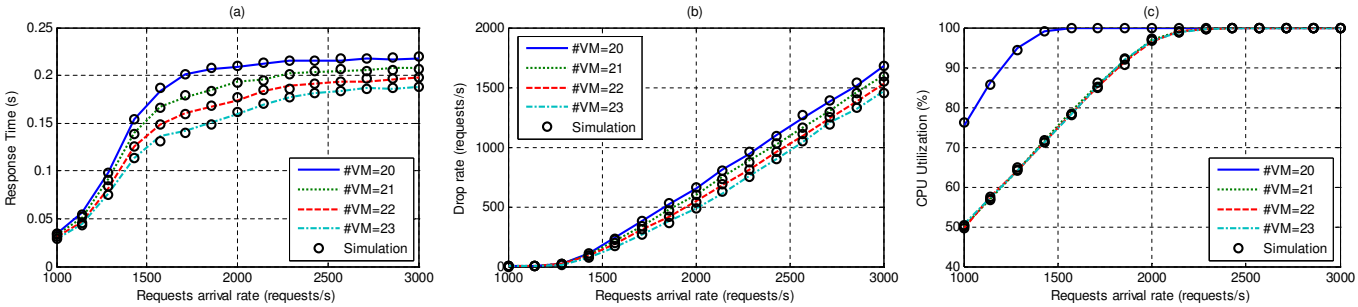

Fig. 5. Database performance curves using multiple VM instances as functions of requests arrival rate

[22] H. M. A. Fahmy, "Simulators and emulators for wsns," in *Wireless Sensor Networks*. Springer, 2016, pp. 381–491.

[23] M. Bertoli, G. Casale, and G. Serazzi, "Jmt: performance engineering tools for system modeling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 4, pp. 10–15, 2009.

[24] U. N. Bhat, *An introduction to queueing theory: modeling and analysis in applications*. Birkhäuser, 2015.