Performance Analysis of Cloud Computing Centers Using Queuing Systems

S.Anjalai Devi*, R.Sudha

Department of Computer Science and Engineering
PRIST University, Tiruchirappalli, Tamil Nadu, India
*anjalidevi12@gmail.com

***Abstract*— Successful development of cloud computing paradigm necessitates accurate performance evaluation of cloud data centers. As exact modeling of cloud centers is not feasible due to the nature of cloud centers and diversity of user requests, we describe a novel approximate analytical model for performance evaluation of cloud server farms and solve it to obtain accurate estimation of the complete probability distribution of the request response time and other important performance indicators.**

***Keywords*— Cloud computing, performance analysis, response time, queuing theory.**

I. INTRODUCTION

Cloud Computing is a novel paradigm for the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the costs of management and maintenance of hardware and software resources. Cloud computing has a service-oriented architecture in which services are broadly divided into three categories: Infrastructure-as-a- Service (IaaS), which includes equipment such as hardware, storage, servers, and networking components are made accessible over the Internet; Platform-as-a-Service (PaaS), which includes hardware and software computing plat- forms such as virtualized servers, operating systems, and the like; and Software-as-a-Service (SaaS), which includes software applications and other hosted services.

A cloud service differs from traditional hosting in three principal aspects. First, it is provided on demand; second, it is elastic since users can use the service have as much or as little as they want at any given time (typically by the minute or the hour); and third, the service is fully managed by the provide. We assume that any task sent to the cloud center is serviced within a suitable facility node; upon finishing the service, the task leaves the center. A facility node may contain different computing resources such as web servers, database servers, directory servers, and others. A service level agreement, SLA, outlines all aspects of cloud service usage and the obligations of both service providers and clients, including various descriptors collectively referred to as Quality of Service (QoS). QoS includes availability, throughput, reliability, security, and many other parameters, but also performance indicators such as response time, task blocking probability, probability of immediate service, and mean number of tasks in the system, all of which may be determined using the tools of queuing theory. However, cloud centers differ from traditional queuing systems in a number of important aspects.

1. A cloud center can have a large number of facility (server) nodes, typically of the order of hundreds or thousands; traditional queuing analysis rarely considers systems of this size.
2. Task service times must be modeled by a general, rather than the more convenient exponential, probability distribution. Moreover, the coefficient of variation of task service time may be high—i.e., well over the value of one.
3. Due to the dynamic nature of cloud environments, diversity of user's requests and time dependency of load, cloud centers must provide expected quality of service at widely varying loads.

To fill this gap, in this work, we model the cloud center as an M/G/m/m + r queuing system with single task arrivals and a task buffer of finite capacity. We evaluate its performance using a combination of a transform-based analytical model and an approximate Markov chain model, which allows us to obtain a complete probability distribution of response time and number of tasks in the system. We also discuss the probability of immediate service (i.e., no waiting in the input buffer) and blocking probability, and determine the size of the buffer needed for the blocking probability to remain below a predefined value. Analytical results are validated through discrete-event simulation.

II. THE PROPOSED ANALYTICAL MODEL

We model a cloud server farm as a M/G/m/m+ r queuing system which indicates that the interarrival time of requests is exponentially distributed, while task service times are independent and identically distributed random variables that follow a general distribution with mean value of μ. The system under consideration contains m servers which render service in order of task request arrivals (FCFS). The capacity of system is m + r which means the buffer size for incoming request is equal to r. As the population size of a typical cloud center is relatively high while the probability that a given user will request service is relatively small, the arrival process can be modeled as a Markovian process. An M/G/m/m+ r queuing system may be considered as a semi-Markov process which can be analyzed by exploiting the embedded Markov chain

technique. Embedded Markov Chain technique requires selection of Markov points in which the state of the system is observed. Therefore, we model the number of the tasks in the system (both those in service and those queued but not yet serviced) at the moments immediately before task request arrivals; if we enumerate these instances as 0; 1; 2; ... ; m+ r, we obtain a homogeneous Markov chain. Therefore, the semi-Markov process records the state at arbitrary time while the embedded Markov chain only observes the state at which the system has an arrival.

Advantages:

Performance evaluation of server farms is an important aspect of cloud computing which is of crucial interest for both cloud providers and cloud customers. In this paper, We have proposed an analytical technique based on an approximate Markov chain model for performance evaluation of a cloud computing center. Due to the nature of the cloud environment, We assumed general service time for requests as well as large number of servers, which makes our model flexible in terms of scalability and diversity of service time.

2.1 The Embedded Markov Chain

The moments of task request arrivals are selected as Markov points. Two successive task request arrivals and task departures that (may) occur between them are shown schematically in Fig. 1. Note that the number of departures may be anywhere between 0 and 1, but it is likely to be low in fact, when the system is in the steady state, there will be on the average a single departure between every two successive arrivals. As our embedded Markov chain is homogeneous and ergodic, it has a steady-state solution. (Definition of ergodicity and the proof that the Markov chain is ergodic can be found in Appendix B, available in the online supplemental material.) Therefore, we can calculate the distribution of number of tasks in the system as well as the mean response tim**e.**

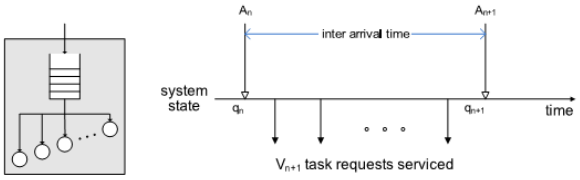


Fig.1. Embedded Markov Points

Let A_n and A_{n+1} indicate the moment of n^{th} and $(n+1)^{th}$ arrivals to the system, respectively, while q_n and q_{n+1} indicate the number of tasks found in the system immediately before these arrivals. If v_{n+1} indicates the number of tasks which depart from the system between A_n and A_{n+1}, then $q_{n+1}= q_n-v_{n-1}+1$.

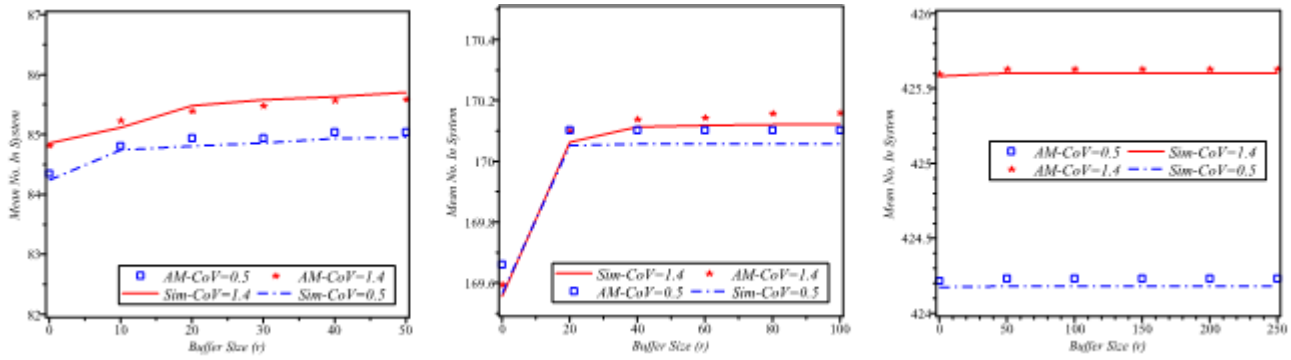
III. NUMERICAL VALIDATION

We have considered the system with different number of servers (m = 50, 100, and 200), while the input buffer was made variable from r = 0 to m/2 in five steps. In all cases, traffic intensity was ρ= 0:85, while the coefficient of variation CoV was assigned values of 0.5 and 1.4. These values give reasonable insight into the behavior and dimensioning of cloud centers. While the number of servers may be too low and the traffic intensity too high for a large provider such as Amazon (it is worth noting that no cloud provider publishes information regarding average traffic intensity, buffer space, number of servers, or the percentage of reserved, on-demand or spot instances), the values chosen may be quite applicable to small- to medium-sized providers that try to keep the utilization of their servers as high as possible.

Mean number of tasks in the system is shown in Fig. 4. As can be seen, it increases rather smoothly with buffer size when the number of servers is m = 50, but it is much less pronounced when the number of servers is higher. In fact, he impact of buffer size becomes virtually undetectable when the number of servers is high (m =200).

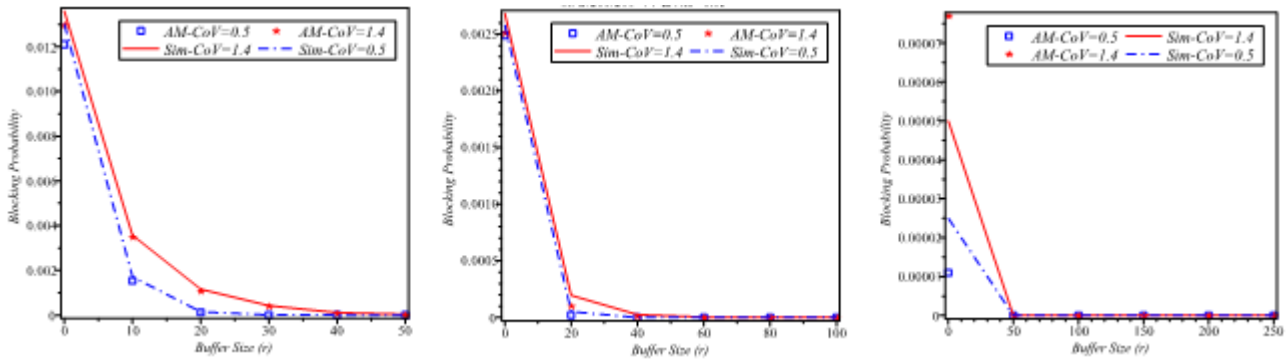
Blocking probability is shown in Fig. 5. As can be seen, it decreases rapidly when the buffer size increases. From the plot, we can estimate the minimum buffer size needed to keep the blocking probability below a given value ϵ. For ϵ = 0.002 (i.e., 0.2 percent), buffer size should be at least 10 for the system with m = 50 servers. For systems with higher number of servers, this minimum is lower, and even small extra buffer space results in virtually no blocking at all, as can be seen in Figs. 5b and 5c.

Finally, Fig. 6 shows the probability that a task will get immediate service without any queuing; this probability is an important non functional service property for cloud customers. Intuitively, this probability should not depend on the system capacity—it requires the presence of at least one idle server. However, its value is close to 1 at low buffer sizes, and then decreases and stabilizes as the number of spaces in the buffer r increases. This behaviour is due to the fact that, at low values of r (i.e., small buffer size), an arriving task is not very likely to get queued; instead, it will either get blocked (as per Fig. 5) or immediately serviced. Only when the buffer size exceeds a certain value will the probability of queuing become non negligible. In other words, increasing the capacity for queuing (i.e., the buffer size) will decrease both the probability of blocking and the probability of getting immediately into service. However, this trend ceases to hold beyond a certain value, and adding Fig. 4. Mean number of tasks in the system at traffic intensity ρ= 0:85. Fig. 5. Blocking probability. Fig. 6. Probability of immediate service. Extra buffer capacity will not affect either probability to a noticeable degree.



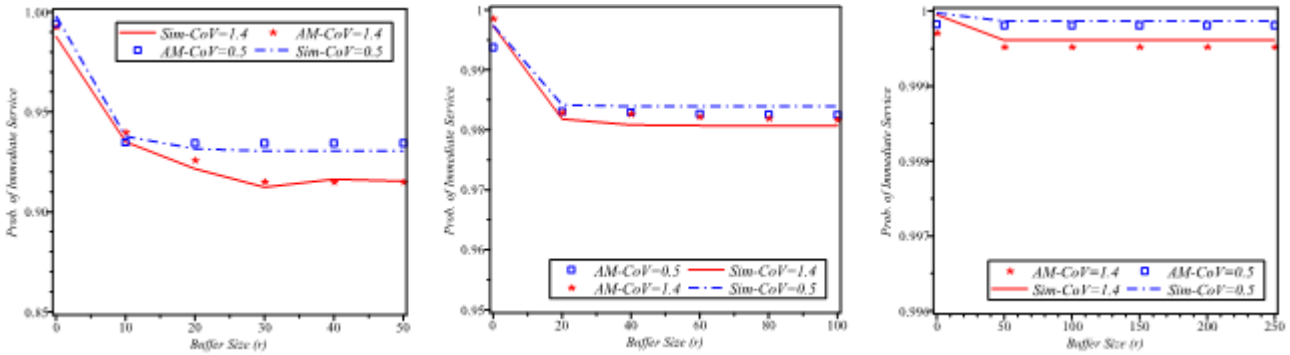
a) System with m=50 servers b) System with m=50 servers c) System with m=50 servers

Fig.4 Mean number of tasks in the system at traffic intensity ρ= 0:85



a) M/G/50/50+r b) M/G/100/100+r c) M/G/200/200+r

Fig.5 Blocking probability



a) M/G/50/50+r b) M/G/100/100+r c) M/G/200/200+r

Fig.6 probability of immediate service

We only show the results for M/G/50/50+r because the discussion aims to highlight the influences of coefficient of variation of service time on response time; for larger systems, the discussion is the same. Distinguishing between tasks and allocating dedicate buffer space to different classes of tasks can be a way to avoid sudden long delay. In other words, establishing a few parallel distinct homogeneous cloud centers instead of having one central heterogeneous center could be a solution which will decrease the waiting time; this provides another justification for providing performance results only for a cloud center with 50 servers. Finally, we note that the

agreement of simulation and analytical results is very good, which confirms the validity of our analytical model.

IV. CONCLUSION

Performance evaluation of server farms is an important aspect of cloud computing which is of crucial interest for both cloud providers and cloud customers. In this paper, we have proposed an analytical technique based on an approximate Markov chain model for performance evaluation of a cloud computing center. Due to the nature of the cloud environment, we assumed general service time for requests as well as large number of servers, which makes our model flexible in terms of scalability and diversity of service time. We have further conducted numerical experiments and simulation to validate our model. Numerical and simulation results showed that the proposed approximate method provides results with high degree of accuracy for the mean number of tasks in the system, blocking probability, probability of immediate service as well as the response time distribution characteristics such as mean, standard deviation, skewness, and kurtosis. Our results also indicate that a cloud center that accommodates heterogeneous services may impose longer waiting time for its clients compared to its homogeneous equivalent with the same traffic intensity.

REFERENCES

1. B. Furht, "Cloud Computing Fundamentals," Handbook of Cloud Computing, pp. 3-19, Springer, 2010.
2. L. Wang, G. von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, "Cloud Computing: A Perspective Study," New Generation Computing, vol. 28, pp. 137-146, 2010.
3. K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing," Proc. IEEE World Conf. Services, pp. 693-700, 2009.
4. B. Yang, F. Tan, Y. Dai, and S. Guo, "Performance Evaluation of Cloud Service Considering Fault Recovery," Proc. First Int'l Conf. Cloud Computing (CloudCom '09), pp. 571-576, Dec. 2009.
5. L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards a Cloud Definition," ACM SIGCOMM Computer Comm. Rev., vol. 39, pp. 50-55, Dec. 2008.
6. J.M. Smith, "M/G/c/K Blocking Probability Models and System Performance," Performance Evaluation, vol. 52, pp. 237-267, May 2003.
7. RSoft Design, Artifex v.4.4.2, San Jose, CA: RSoft Design Group, Inc., 2003.
8. D.N. Joanes and C.A. Gill, "Comparing Measures of Sample Skewness and Kurtosis," J. Royal Statistical Soc.: Series D (The Statistician), vol. 47, no. 1, pp. 183-189, 1998.
9. H.C. Tijms, "Heuristics for Finite-Buffer Queues," Probability in the Eng. and Informational Sciences, vol. 6, pp. 277-285, 1992.
10. M. Miyazawa, "Approximation of the Queue-Length Distribution of an M/GI/s Queue by the Basic Equations," J. Applied Probability, vol. 23, pp. 443-458, 1986.
11. D.D. Yao, "Refining the Diffusion Approximation for the M/G/m Queue," Operations Research, vol. 33, pp. 1266-1277, 1985.
12. K.T. Marshall and R.W. Wolff, "Customer Average and Time Average Queue Lengths and Waiting Times," J. Applied Probability, vol. 8, pp. 535-542, 1971

