Performance Evaluations of a Cloud Computing Physical Machine with Task Reneging and Task Resubmission (Feedback)

Godlove Suila Kuaban[1], Bhavneet Singh Soodan[2(✉)], Rakesh Kumar[2(✉)], and Piotr Czekalski[3(✉)]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Baltycka 5, 44-100 Gliwice, Poland
gskuaban@iitis.pl

[2] School of Mathematics, Shri Mata Vaishno Devi University, Katra 182320, Jammu and Kashmir, India
bhavneet5678@gmail.com, rakesh.kumar@smvdu.ac.in

[3] Department of Computer Graphics, Vision and Digital Systems, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
piotr.czekalski@polsl.pl

Abstract. Cloud service providers (CSP) provide on-demand cloud computing services, reduces the cost of setting-up and scaling-up IT infrastructure and services, and stimulates shorter establishment times for start-ups that offer or use cloud-based services. Task reneging or dropping sometimes occur when a task waits in the queue longer than its timeout or execution deadline, or it is compromised and must be dropped from the queue or as an active queue management strategy to avoid tail dropping of tasks when the queues are full. Reneged or dropped tasks could be resubmitted provided they were not dropped due to security reasons. In this paper, we present a simple M/M/c/N queueing model of a cloud computing physical machine, where the interarrival times and the services times are exponentially distributed, with N buffer size and c virtual machines running in parallel. We present numerical examples to illustrate the effect of task reneging and task resubmission on the queueing delay, probability of task rejection, and the probability of immediate service.

Keywords: Transient-state · Steady-state · Performance evaluations · Cloud computing · Physical machines · Tasks reneging or dropping · Tasks resubmission or feedback

1 Introduction

Cloud service providers (CSP) provide on-demand cloud computing services such as software, platform and infrastructure to their customers. It enables the users

P. Gaj et al. (Eds.): CN 2020, CCIS 1231, pp. 185–198, 2020.
https://doi.org/10.1007/978-3-030-50719-0_14

to access these services anywhere, at any time and based on their needs without being concerned about the cost and time of setting up and running their infrastructure from scratch. Therefore, cloud computing has stimulated shorter establishment times for start-ups that offer or use cloud-based services and the creation of scalable enterprise applications [1]. Performance evaluations of cloud computing systems have been studied using queueing theory in [6,9–13]. The use of analytical modelling methods offer faster and less expensive performance evaluation tools when compared to testbed experiments and discrete event simulation, which are time-consuming and expensive [14]. The results obtained using analytical modelling may be an approximation of the relative trends of the performance parameters but can be used to derive high-level insight into the behaviour of the system [2]. The evaluation of cloud computing systems may require the prediction and estimation of the cost-benefit of a strategy and the corresponding acceptable quality of service (QoS) which may not be feasible by simulation or measurements [3].

Task reneging or dropping sometimes occur when a task waits in the queue longer than its timeout or execution deadline, or it is compromised and must be dropped from the queue or as an active queue management strategy to avoid tail dropping of tasks when the queues are full. Reneged or dropped tasks could be resubmitted provided they were not dropped due to security reasons. Dropping of tasks from the queue is called task reneging [15] while the resubmission of the dropped task is called feedback [16]. The authors in [4,5,17,18] studied task reneging in the context of cloud computing but their studies were limited to steady-state Markovian modelling without resubmission.

In this paper, we present a simple M/M/c/N queueing model of a cloud processing physical machine, where the interarrival times and the services times are exponentially distributed, with N buffer size and c virtual machines running in parallel. We present numerical examples to illustrate the effect of reneging and feedback on the queueing delay, probability of task rejection, and the probability of immediate service. The rest of the paper is arranged as follows: model description is presented in Sect. 2, performance modelling is presented in Sect. 3, some numerical examples are presented in Sect. 4 and conclusion in Sect. 5.

2 Model Description

The tasks submitted to a cloud computing infrastructure may be queued up in the load balancer and then scheduled to any of the available physical machines, provided the rate of arrival of tasks is far greater than the scheduling rate [7]. The load balancing mechanism detects the physical machines that are overloaded and those that are underutilised and strive to balance the load among them [7]. However, the evaluation of the load balancer is out of the scope of this paper. In the physical machines, the tasks can then be scheduled into any available VMs for processing. Because some of the tasks may be time-constrained or likely to fail or maybe have been compromised, it will renege or dropped from the queue or moved to another queue (jockeying) [8] depending on the queue management

strategy implemented. Figure 1 shows a general cloud computing model where users can submit tasks over the internet to a cloud computing data centre infrastructure, which consists of the load balancer and physical machines which are hosting virtual machines.

The use of effective tasks scheduling policies ensures that the potential of cloud computing is fully harnessed and exploited to meet the QoS requirements of cloud computing services. The authors in [20] presented a review of cloud computing scheduling methods which are categorised into QoS-based task scheduling, ant Colony optimisation Algorithm-based scheduling, particle swarm optimisation (PSO)-based task scheduling, Multiprocessor-based scheduling, Fuzzy-based scheduling, Clustering-based, task scheduling, Deadline-constrained scheduling, Cost-based, scheduling and other scheduling-based approaches. Scheduling algorithms which use techniques such as round-robin, allocation, a probabilistic allocation that seek to minimise the average response time, Random Neural Network (RNN) based allocation scheme that uses reinforcement learning and on-line greedy adaptive algorithm were presented in [22]. A discrete symbiotic organism search (DSOS) scheduling algorithm was proposed in [21] for optimal scheduling of tasks in cloud data centres.

Suppose that the tasks scheduled to a given physical machine are arriving with an arrival rate of λ as shown in Fig. 2. If the rate of arrival of tasks is greater than the rate at which they are processed, then those that arrive and when all the virtual machines (VMs), $\{VM_1, VM_2, VM_3, \cdots, VM_c\}$ that are running in parallel are busy, will have to wait and then later scheduled for execution. The processing server or physical machine is modelled as an M/M/c/N, where c is the number of VMs and N is the maximum number of tasks or the buffer size. It is assumed that all the VMs have the same processing rate, μ.

The model proposed in the paper are based on the following assumptions:

1. The arrival process of tasks into the task buffers in the processing servers follows a Poisson process with parameter λ.
2. The system has a single queue and finitely many numbers of VMs. The processing times of each VM are exponentially distributed with parameter μ. The mean processing rate of tasks is: $\mu_n = \begin{cases} n\mu, & 0 \leq n < c \\ c\mu & c \leq n \leq N \end{cases}$
3. The queue discipline is FCFS.
4. The capacity of the system is finite (say, N).
5. The reneging times or the times at which the tasks are dropped from the queue are exponentially distributed with parameter ξ.
6. When a task reneges or is dropped from the queue, it can be resubmitted with a probability p otherwise, with a probability $q = 1 - p$ it is not resubmitted.

3 Performance Evaluation Modelling: Steady-State and Transient-State Solutions

Defining the following probabilities:

$P_0(t)$ is the probability that at time t there is no task in the system.

$P_n(t)$ is the probability that at time t there are $1 \leq n \leq N$ tasks in the system.

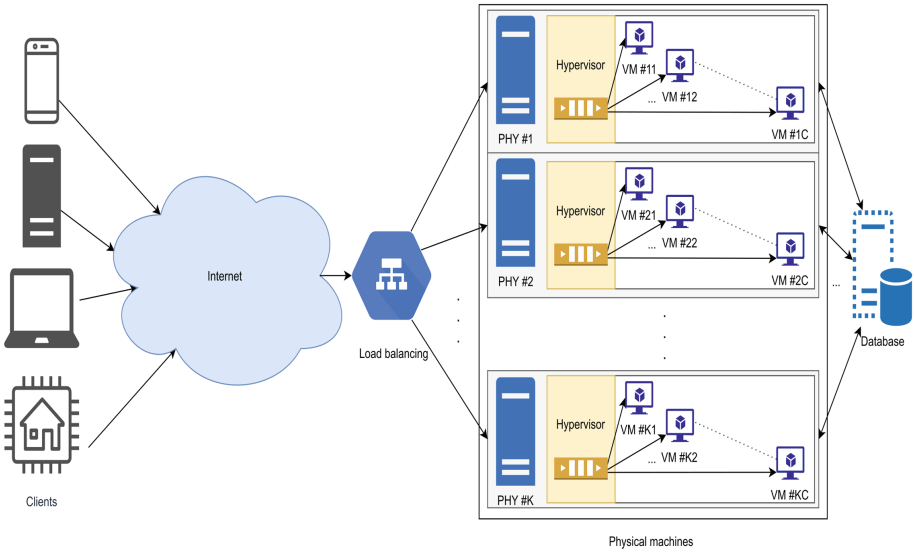

Fig. 1. Cloud computing model

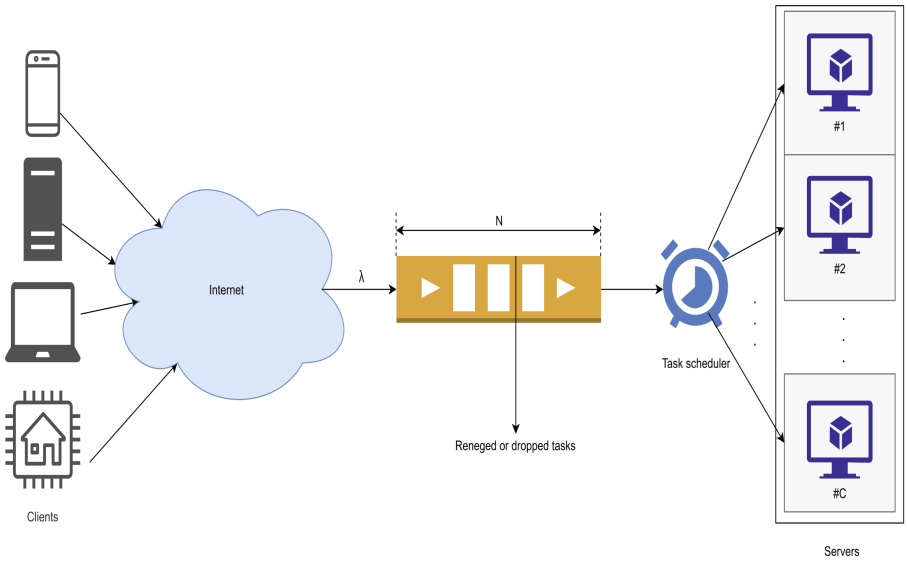

Fig. 2. Queueing model of a cloud computing physical machine

The difference-differential equations of the queuing model are:

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + q\mu P_1(t), n = 0 \tag{1}$$

$$\frac{dP_n(t)}{dt} = -(\lambda + nq\mu)P_n(t) + \lambda P_{n-1}(t) + (n+1)q\mu P_{n+1}(t), 1 \le n < c \tag{2}$$

$$\frac{dP_n(t)}{dt} = -(\lambda + cq\mu)P_n(t) + \lambda P_{n-1}(t) + (cq\mu + \xi)P_{n+1}(t), n = c \tag{3}$$

$$\frac{dP_n(t)}{dt} = -[\lambda + cq\mu + (n-c)\xi]P_n(t) + \lambda P_{n-1}(t) + [cq\mu + (n+1-c)\xi]P_{n+1}(t), c+1 \le n < N \tag{4}$$

$$\frac{dP_N(t)}{dt} = -[cq\mu + (N-c)\xi]P_N(t) + \lambda P_{N-1}(t), n = N \tag{5}$$

In steady-state, when $\lim_{t\to\infty} P_0(t) = P_0$, $\lim_{t\to\infty} p_n(t) = p_n$, $\lim_{t\to\infty} p_N(t) = p_N$, Eqs. (1)–(5) becomes:

$$0 = -\lambda P_0 + q\mu P_1, n = 0 \tag{6}$$

$$0 = -(\lambda + nq\mu)P_n + \lambda P_{n-1} + (n+1)q\mu P_{n+1}, 1 \le n < c \tag{7}$$

$$0 = -(\lambda + cq\mu)P_n + \lambda P_{n-1} + (cq\mu + \xi)P_{n+1}, n = c \tag{8}$$

$$0 = -[\lambda + cq\mu + (n-c)\xi]P_n + \lambda P_{n-1} + [cq\mu + (n+1-c)\xi]P_{n+1}, c+1 \le n < N \tag{9}$$

$$0 = -[cq\mu + (N-c)\xi]P_N + \lambda P_{N-1}, n = N \tag{10}$$

The above (N + 1) linear equations in the unknown probabilities $P_0, P_1 ... P_N$ are solved as follows:

Solving (6) and (7), we get

$$P_n = \frac{1}{n!}\left(\frac{\lambda}{q\mu}\right)^n P_0, 0 \le n \le c \tag{11}$$

Now, from Eqs. (8)–(10) and using relation (11), we get

$$P_n = \frac{1}{c!}\left(\frac{\lambda}{q\mu}\right)^c \frac{\lambda^{(n-c)}}{\prod_{m=c+1}^{N}[c\mu q + (m-c)\xi]} P_0, c+1 \le n \le N \tag{12}$$

Thus, P_n can be written as:

$$P_n = \begin{cases} \frac{1}{n!}\left(\frac{\lambda}{q\mu}\right)^n P_0 & 0 \le n \le c \\ \frac{1}{c!}\left(\frac{\lambda}{q\mu}\right)^c \frac{\lambda^{(n-c)}}{\prod_{m=c+1}^{N}[c\mu q + (m-c)\xi]} P_0 & c+1 \le n \le N \end{cases} \tag{13}$$

Where P_0 can be obtained using normalization equation, $\Sigma_{n=0}^{N} P_n = 1$.

We use a numerical method (Runge-Kutta method of fourth order) to obtain transient solution of the model. The "*ode*45" function of MATLAB software is

used to compute the transient numerical results. The mean number of tasks waiting in the queue, $L_q(t)$ and the mean waiting time $W_q(t)$ respectively are given by [24]:

$$L_q(t) = \sum_{n=c}^{N} (n-c) P_n(t) \tag{14}$$

$$W_q(t) = \frac{L_q(t)}{c\mu[1 - \sum_{n=0}^{c} P_n(t)]}$$

The transient state probabilities, including the probability that the queue is empty and the probability that the buffer is full can be obtained by solving set of equations in (5) numerically. If the queue is empty, incoming tasks will be processed immediately, it provides good quality of service (QoS) to the users but it is not profitable for the CSPs. If the buffer is full, then incoming tasks will be rejected, which results in poor QoS.

4 Numerical Examples

In this section we present numerical examples to illustrate the effect of reneging and feedback on the queueing delay, probability of task rejection, and probability of immediate service. We use a numerical method (Runge-Kutta method of fourth order) to obtain transient solution of the model. The "*ode*45" function of MATLAB software is used to compute the transient numerical results.

Figures 3 and 4 shows the variation of the state probabilities with time. Generally, the state probabilities increase sharply and then attains steady state. $P_0(t)$ is the probability that the queue is empty at the time, t, such that any packet that arrives is immediately scheduled into the VM for processing while $P_{10}(t)$ is the probability that there are 10 tasks in the queue. The values of the parameters are taken as: $\lambda = 12, \mu = 5, q = 0.9, \xi = 0.4, c = 3$.

Figure 5 shows the transient behaviour of the mean number of tasks in the queue with time. The mean queue size increases with time for a constant arrival rate and then attains a steady state after a long time. It can be observed that when tasks that are dropped from the queue are resubmitted, the queue size is relatively larger. Similar behaviour can be observed in Fig. 5 and 6, which shows the transient behaviour of the mean delay and the probability of task dropping when the buffer is full. It can also be observed that when the tasks that are dropped from the queue are resubmitted, the probability of task dropping or tail dropping of packets when the buffers are full is relatively higher. The values of parameters for used are: $\lambda = 85, \mu = 30, q = 0.85, \xi = 0.4, c = 3, N = 50$ and the initial condition is $P_7(0) = 1$.

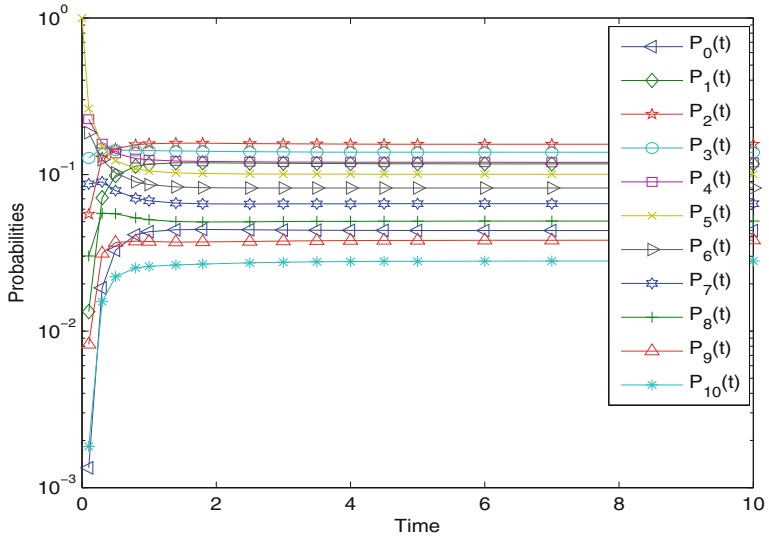

Fig. 3. Probabilities vs time

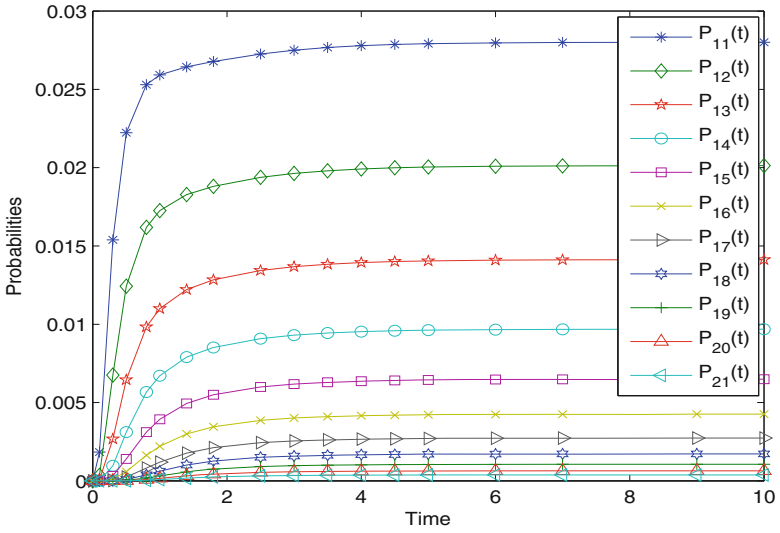

Fig. 4. Probabilities vs time

Figure 7 shows the effect of the average arrival rate of tasks on mean queueing delay. The mean delay increases as the rate of arrival of tasks increases slowly, and after a certain value of the arrival rate, a small increase in the arrival rate of tasks will result in a corresponding fast increase in the delay. A similar behaviour

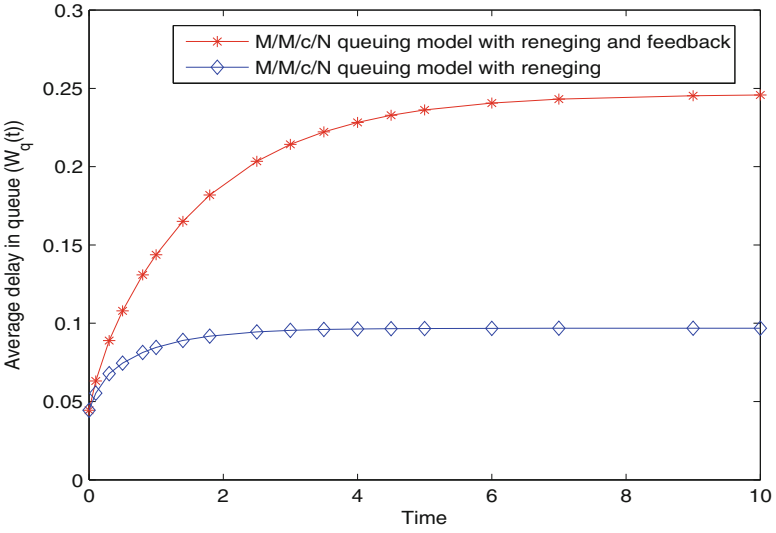

Fig. 5. Comparison of average delay in queue vs time

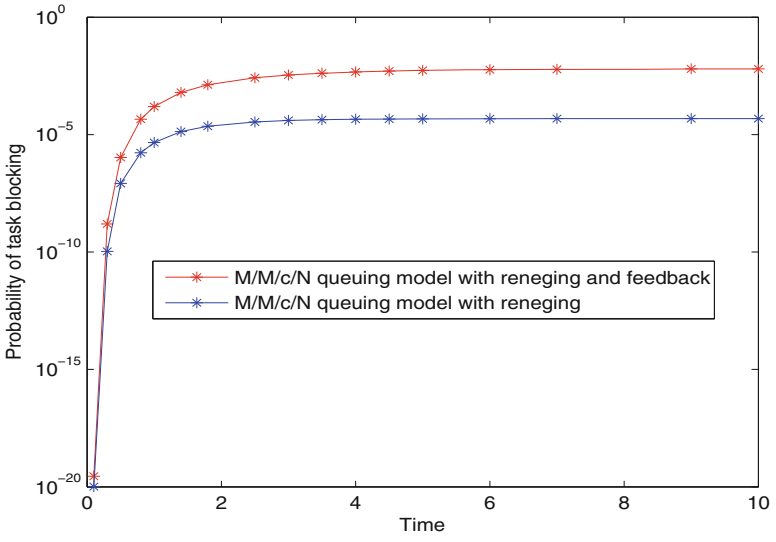

Fig. 6. Comparison of probability of task blocking vs time

of the probability of task blocking can be seen in Fig. 8. The values of parameters are: $\mu = 30, q = 0.85, \xi = 0.3, c = 3, N = 50$ at t = 3. Initial condition is $P_7(0) = 1$.

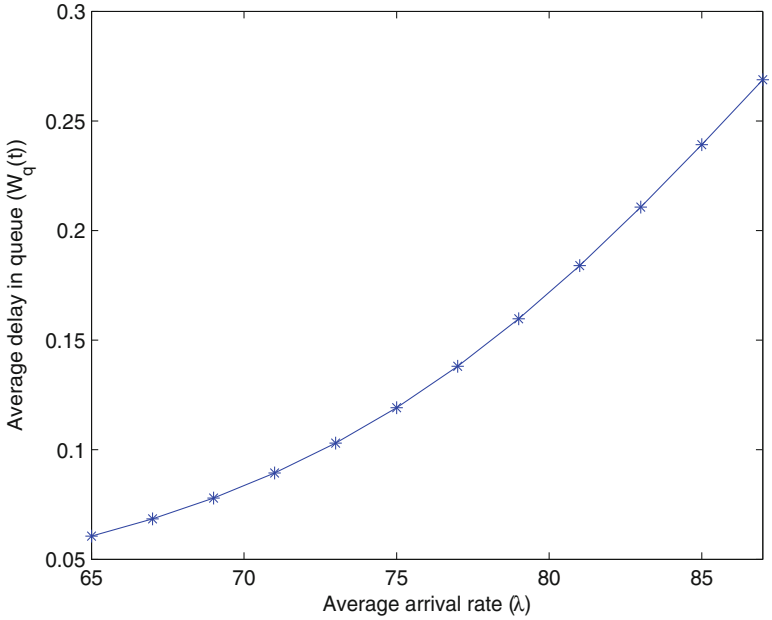

Fig. 7. Effect of average arrival rate on average delay in queue

Figure 9 shows the variation of the mean queueing delay with the probability of feedback. As the probability that tasks that are dropped from the queue are resubmitted increases, the higher the delay. Figure 10 shows that variation of the reneging rate with the mean queueing delay. Resubmission of tasks that reneged from the queue or a task that was rejected is very important to ensure QoS of some users; other users may have to wait longer in the queue. The values of parameters used are: $\lambda = 78, \mu = 30, \xi = 0.3, c = 3, N = 50$ at t = 3. Initial condition is $P_7(0) = 1$ and $\lambda = 88, \mu = 30, q = 0.8, c = 3, N = 50$ at t = 3. Initial condition is $P_7(0) = 1$ respectively.

Figure 11 shows that increasing the number of VMs will significantly decrease the queueing delays. In other to reduce energy consumption in cloud data centres, but the drawback of such a strategy is an increase in the queueing delay. Similar behaviour can be seen in Fig. 12, where increasing the number of VM also decreases the probability of task blocking. Therefore, increasing the number of VMs will improve the QoS but increases the energy consumption and hence the costs on the CSP. Other QoS and energy optimization methods such as task

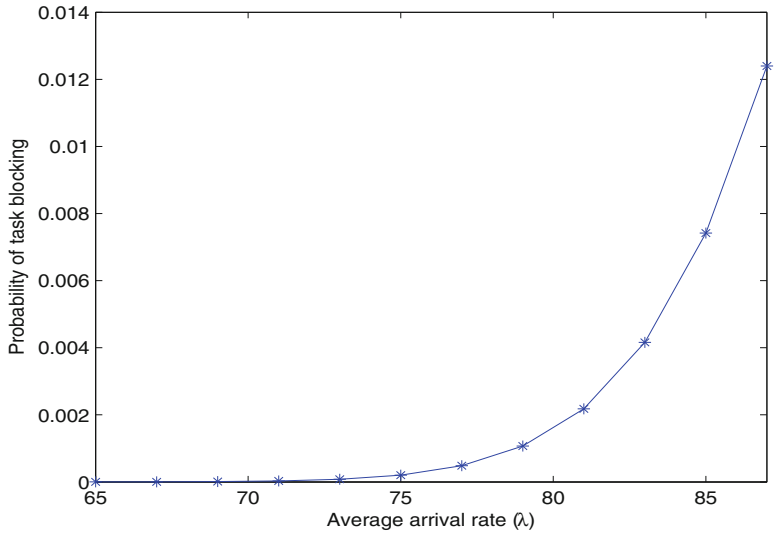

Fig. 8. Effect of average arrival rate on probability of task blocking

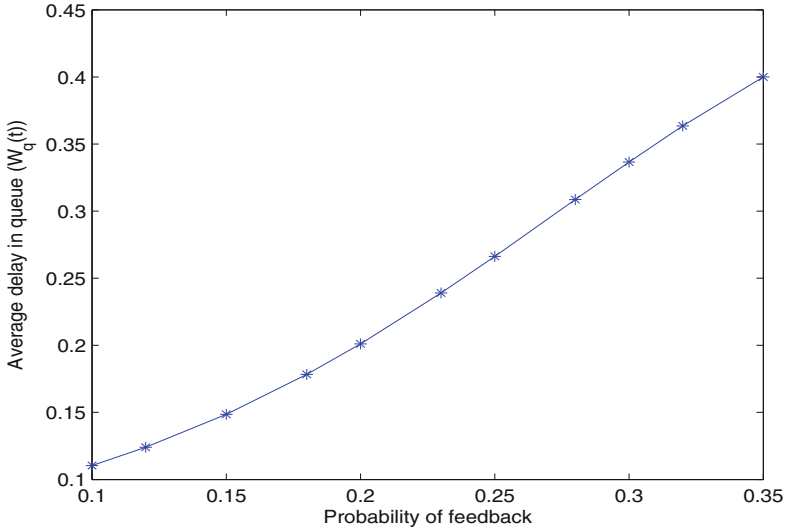

Fig. 9. Effect of probability of feedback on average delay in queue

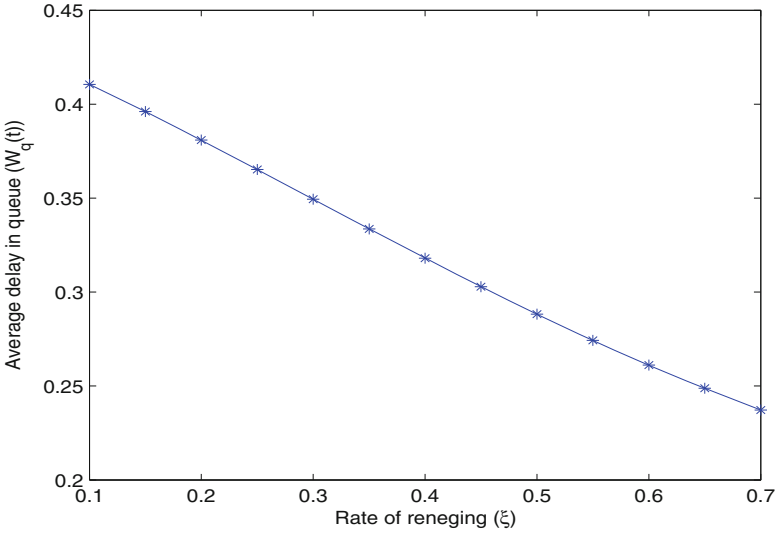

Fig. 10. Effect of rate of reneging on average delay in queue

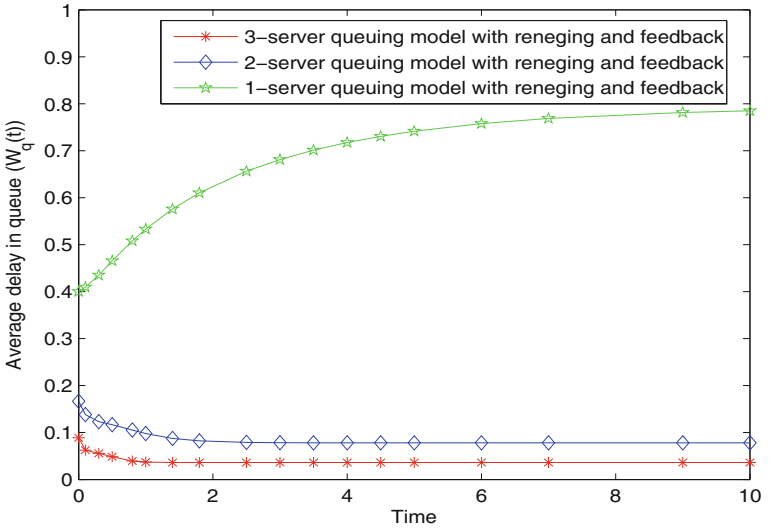

Fig. 11. Effect of number of servers on average delay in queue

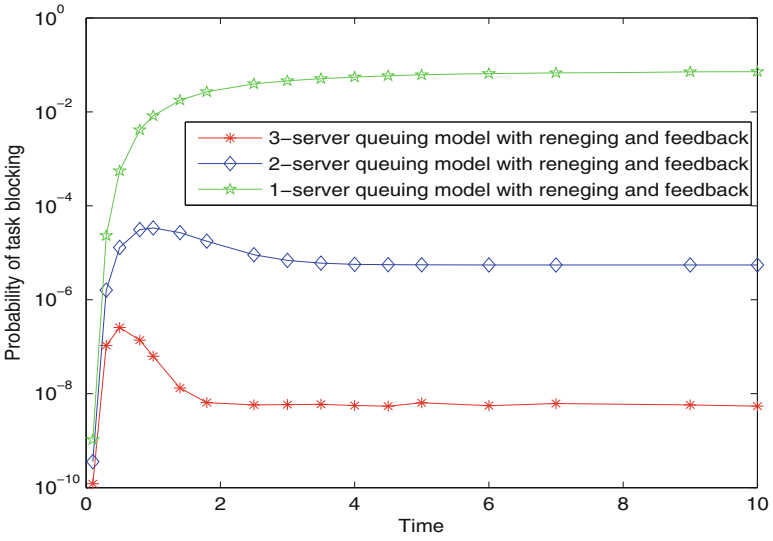

Fig. 12. Effect of number of servers on probability of task blocking

migration can be considered by are out of the scope of this work. The values of parameters used are: $\lambda = 14, \mu = 15, q = 0.8, \xi = 0.1, N = 20$. Initial condition is $P_7(0) = 1$.

5 Conclusion

We have presented a simple M/M/c/N queueing model of a cloud processing in which tasks could be dropped from the queue, the dropped tasks can be resubmitted for possible processing and if the buffer where the tasks are stored is full, subsequent tasks will be rejected. We have presented numerical examples to illustrate its utility by considering the effects of reneging and feedback on the queueing delay, probability of task rejection, and the probability of immediate service. We intend to extend this study to the evaluation of a cloud infrastructure with load balancing, where tasks are the first queue up and then scheduled into the various processing server and reneging and feedback will be considered both at the load balancer and the processing servers.

References

1. Buyya, R., et al.: A manifesto for future generation cloud computing: research directions for the next decade. ACM Comput. Surv. **51**(5), 38 (2018). https://doi.org/10.1145/3241737. Article no. 105
2. Paya, A., Marinescu, D.C.: Energy-aware load balancing and application scaling for the cloud ecosystem. IEEE Trans. Cloud Comput **5**(1), 15–27 (2017)

3. Bruneo, D.: A stochastic model to investigate data center performance and QoS in IaaS cloud computing systems. IEEE Trans. Cloud Comput. **25**(3), 560–569 (2014)
4. Chiang, Y.J., Ouyang, Y.C., Hsu, C.H.: Performance and cost-effectiveness analyses for cloud services based on rejected. IEEE Trans. Serv. Comput. **9**(3), 446–455 (2016)
5. Homsi, S., Liu, S., Chaparro-Baquero, A., Bai, O., Ren, S., Quan, G.: Workload consolidation for cloud data centers with guaranteed QoS using request reneging. IEEE Trans. Parallel Distrib. Syst. **28**(7), 2103–2116 (2017)
6. Ait El Mahjoub, Y., Fourneau, J.-M., Castel-Taleb, H.: Analysis of energy consumption in cloud center with tasks migrations. In: Gaj, P., Sawicki, M., Kwiecień, A. (eds.) CN 2019. CCIS, vol. 1039, pp. 301–315. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21952-9_23
7. Mishra, S.K., Sahoo, B., Parida, P.P.: Load balancing in cloud computing: a big picture. Advances in Big Data and Cloud Computing. J. King Saud Univ. - Comput. Inf. Sci. (2018)
8. Gupta, S., Arora, S.: Queueing system in cloud services management: a survey. Int. J. Pure Appl. Math. **119**(12), 12741–12753 (2018)
9. Vilaplana, J., et al.: A queueing theory model for cloud computing. J. Supercomput. **69**(1), 492–507 (2014)
10. Czachórski, T., Kuaban, G.S., Nycz, T.: Multichannel diffusion approximation models for the evaluation of multichannel communication networks. In: Vishnevskiy, V.M., Samouylov, K.E., Kozyrev, D.V. (eds.) DCCN 2019. LNCS, vol. 11965, pp. 43–57. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36614-8_4
11. Vetha, S., Devi, V.: Dynamic resource allocation in cloud using queueing model. J. Ind. Pollut. Control **33**(2), 1547–1554 (2017)
12. Cheng, C., Li, J., Wang, Y.: An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. Tsinghua Sci. Technol. **20**(1), 28–39 (2015)
13. ElKafhali, S., Salah, K.: Modelling and analysis of performance and consumption in cloud data centers. Arab. J. Sci. Eng. **43**, 7789–7802 (2018)
14. Duan, Q., Yu, S., Zhang, Z.: Cloud service performance evaluation: status, challenges, and opportunities - a survey from the system modeling perspective. Digit. Commun. Netw. **3**, 101–111 (2017)
15. Al-Seedy, R.O., El-Sherbiny, A.A., El-Shehawy, S.A., Ammar, S.I.: Transient solution of the $M/M/c$ queue with balking and reneging: a survey. Comput. Math. Appl. **57**(8), 1280–1285 (2009)
16. Kumar, R., Sharma, S.K.: M/M/1 feedback queueing models with retention of reneged customers and balking. Am. J. Oper. Res. **3**(2A), 1–6 (2013)
17. Karina, V., Rodriguez, Q., Guillemin, F.: Performance analysis of resource pooling for network function virtualization. Psicologia: Reflexao e Crítica, Universidade Federal do Rio Grande do Sul, 2016. hal-01621281 (2016)
18. Chiang, Y., Ouyang, Y.: Profit Optimization in SLA-Aware Cloud Services with a Finite Capacity Queuing Model Mathematical Problems in Engineering. Hindawi Publishing Corporation, London (2014)
19. Farahnakian, F., Pahikkala, T., Liljeberg, P., Plosila, J., Hieu, N.T., Tenhunen, H.: Energy-aware VM consolidation in cloud data centers using utilization prediction model. IEEE Trans. Cloud Comput. **7**(2), 524–536 (2019)
20. Arunarani, A., Manjula, D., Sugumaran, V.: Task scheduling techniques in cloud computing: a literature survey. Future Gener. Comput. Syst. **91**, 407–415 (2019)

21. Abdullahi, M., Ngadi, M.A., Abdulhamid, S.M.: Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Gener. Comput. Syst. **56**, 640650 (2016)
22. Wang, W., Gelenbe, E.: Adaptive dispatching of tasks in the cloud. IEEE Trans. Cloud Comput. **6**(1), 33–45 (2018)
23. Wei, L., Foh, C.H., He, B., Cai, J.: Towards efficient resource allocation for heterogeneous workloads in IaaS clouds. IEEE Trans. Cloud Comput. **6**(1), 264–275 (2018)
24. Kumar, R., Soodan, B.S.: Transient numerical analysis of a queueing model with correlated reneging, balking and feedback. Reliab.: Theory Appl. **14**(4), 46–54 (2019)