Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Performance Analysis of a Cloud Computing System using Queuing Model with Correlated Task Reneging

To cite this article: Rakesh Kumar *et al* 2021 *J. Phys.: Conf. Ser.* **2091** 012003

View the article online for updates and enhancements.

Performance Analysis of a Cloud Computing System using Queuing Model with Correlated Task Reneging

Rakesh Kumar[1,2], Bhavneet Singh Soodan[1], Godlove Suila Kuaban[3], Piotr Czekalski[4] and Sapana Sharma[5]

[1] School of Mathematics, Shri Mata Vaishno Devi University, Katra, Jammu and Kashmir, 182320, India
[2] Department of Mathematics and Statistics, Namibia University of Science and Technology, Private Bag 13388 Windhoek, Namibia, 13 Jackson Kaujeua Street,Windhoek, Namibia
[3] Institute of Theoretical and Applied Informatics Polish Academy of Sciences, Baltycka 5, 44-100 Gliwice, Poland
[4] Department of Computer Graphics, Vision and Digital Systems, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology
[5] Department of Mathematics, Cluster University of Jammu, Jammu and Kashmir

E-mail: rakesh.kumar@smvdu.ac.in

Abstract. Queuing theory has been extensively used in the modelling and performance analysis of cloud computing systems. The phenomenon of the task (or request) reneging, that is, the dropping of requests from the request queue often occur in cloud computing systems, and it is important to consider it when developing performance evaluations models for cloud computing infrastructures. Majority of studies in the performance evaluation of cloud computing data centres with the use of queuing theory do not consider the fact that the tasks could be removed from queue without being serviced. The removal of tasks from the queue could be due to the user impatience, execution deadline expiration, security reasons, or as an active queue management strategy. The reneging could be correlated in nature, that is, if a request is dropped (or reneged) at any time epoch, and then there is a probability that a request may or may not be dropped at the next time epoch. This kind of dropping (or reneging) of requests is referred to as correlated request reneging. In this paper we have modelled a cloud computing infrastructure with correlated request reneging using queuing theory. An M/M/1/N queuing model with correlated reneging has been used to study the performance analysis of the load balancing server of a cloud computing system. The steady-state as well as the transient performance analyses have been carried out. Important measures of performance like average queue size, average delay, probability of task blocking, and the probability of no waiting in the queue are studied. Finally, some comparisons are performed which describe the effect of correlated task reneging over simple exponential reneging.

1. Introduction

Cloud computing is a distributed [5], dynamic, cost-effective, and scalable computing paradigm that enables on-demand remote access of computing resources such as software, storage, and infrastructure over the internet. The service-oriented architecture of cloud computing can be broadly classified into three categories such as Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS) [10]. The SaaS service model provides software applications directly to the end-user as a service. The IaaS service model provides computing infrastructure such as physical machines, virtual machines, virtual storage, and network to the user on-demand. The PaaS service model provides a runtime development environment and tools to developers as a service.

The most popular configuration for cloud computing data centres that has been proposed and discussed by research scholars and practitioners consists of a master server or load balancer, several computing servers, and high-speed transmission systems [19]. The load balancer receives all requests and then schedules them to the appropriate computing servers, which performs the required computation and then send the result of the computation to the transmission system, which transmits it to the users, or the result of the computation is sent to a storage device. Since requests or tasks are queued in buffers when they arrive, and the servers are busy, cloud computing networks have been abstracted into a queuing network, and then used queuing theory to analyze its performance [8]. The performance indicators considered includes response time (delay due to waiting), task blocking probability (due to buffer overflow), probability of immediate service, and the mean number of tasks in the system [13].

Analytical modelling techniques offer less expensive methods for the design and evaluation of cloud computing systems as there is no need for carrying out expensive test-bed experiments [4] or discrete event simulations which are time-consuming and expensive. Analytical models can be used to have high-level insights into the behavior of the systems within short time periods. But, the results obtained from analytical models are approximations of the relative trends of the performance indicators and may deviate from exact values [1]. The prediction and the estimation of the cost benefit of a strategy and the corresponding acceptable quality of service (QoS) may not be feasible by simulation or field measurements [2]; therefore there is a need of analytical methods for analyzing cloud computing systems.

The majority of studies in the performance evaluation of cloud computing data centre with the use of queuing theory do not consider the fact that tasks could be removed from queue without being serviced. The removal of a task from the queue could result from the impatience of the users. The removal of a task from the queue could be due to the expiration of its execution deadline, due to security reasons, or as an active queue management strategy (to avoid saturation or overflow of buffers). The dropping of requests before they are serviced is called the request or task reneging [11]. Request or task reneging in the context of cloud computing has been studied in [12, 7, 8, 3], but all the studies are based on steady-state queuing models, which necessitates the study of the transient behavior of cloud computing queuing systems with task reneging. Also, when a task is removed from the queue, other tasks in the queue that depend on it will equally be removed. Therefore, the removal or reneging of the task from the queue is correlated, and hence, a queuing model with task reneging can be applied to model correlated removal or reneging of tasks from queuing in a cloud computing data centre. The authors have discussed an analytical approach to model queuing systems with correlated reneging in [6, 9].

In this paper, we model a cloud computing data centre into a queuing network model with task reneging. The rest of the paper is organized as follows. In section 2, we present stochastic queuing model. Mathematical model is developed in section 3. Section 4 deals with steady-state solution. In section 5 the transient analysis is carried out. The conclusion is presented in section

2. Stochastic queuing model

We consider a finite capacity single server Markovian queuing model with correlated reneging. The model under investigation is based on the following assumptions:

- The customers arrive at a service facility one by one in accordance with Poisson process with parameter λ.
- There is a single queue and a single server. The service-times are independently, identically and exponentially distributed with parameter μ.
- The capacity of the system is finite (say, K), where $K=N+1$ and N is the queue capacity.

- After joining the queue and waiting for some time, a customer may get impatient and leave the queue (renege) without getting service. The reneging of customers can take place only at the transition marks $t_0, t_1, t_2, \ldots$, where $\vartheta_r = t_r - t_{r-1}$, $r = 1,2,3\ldots$, are random variables with $P[\vartheta_r \leq x]=1 - exp(-\xi x)$; $\xi \geq 0$, $r = 1,2,3,\ldots$ i.e. the distribution of inter-transition marks is negative exponential with parameter ξ.
- The reneging at two consecutive transition marks is governed by the following transition probability matrix:

$$\begin{array}{cc} & \text{to } t_r \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \text{from } t_{r-1} \begin{array}{c} 0 \\ 1 \end{array} & \begin{Vmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{Vmatrix} \end{array}, \text{ where } p_{00} + p_{01} = 1 \text{ and } p_{10} + p_{11} = 1$$

0 refers to no reneging and 1 refers to the occurrence of reneging. Thus, the reneging at two consecutive transition marks is correlated.

3. Mathematical model

Let us define the following probabilities as:

$Q_{0,r}(t)$ = Probability that at time t the queue length is zero, the server is idle, and r is an indicator whether a customer has reneged or not in previous transition mark ($r = 0$ refers to no reneging and $r = 1$ refers to the occurrence of reneging at previous transition mark).

$P_{0,r}(t)$ = Probability that at time t the queue length is zero, the server is not idle, and r is an indicator whether a customer has reneged or not in previous transition mark ($r = 0$ refers to no reneging and $r = 1$ refers to the occurrence of reneging at previous transition mark).

$P_{n,r}(t)$ = Probability that at time t the queue length is n ($1 \leq n \leq N$), the server is not idle, and r is an indicator whether a customer has reneged or not in previous transition mark ($r = 0$ refers to no reneging and $r = 1$ refers to the occurrence of reneging at previous transition mark).

The differential equations of the model are:

$$\frac{d}{dt}Q_{0,0}(t) = -\lambda Q_{0,0}(t) + \mu P_{0,0}(t) \tag{1}$$

$$\frac{d}{dt}Q_{0,1}(t) = -\lambda Q_{0,1}(t) + \mu P_{0,1}(t) \tag{2}$$

$$\frac{d}{dt}P_{0,0}(t) = -(\lambda + \mu)P_{0,0}(t) + \mu P_{1,0}(t) + \lambda Q_{0,0}(t) \tag{3}$$

$$\frac{d}{dt}P_{0,1}(t) = -(\lambda + \mu)P_{0,1}(t) + \mu P_{1,1}(t) + \lambda Q_{0,1}(t) + \xi[p_{11}P_{1,1}(t) + p_{01}P_{1,0}(t)] \tag{4}$$

$$\frac{d}{dt}P_{n,0}(t) = -(\lambda + \mu + n\xi)P_{n,0}(t) + \mu P_{n+1,0}(t) + \lambda P_{n-1,0}(t) + n\xi\big[p_{00}P_{n,0}(t) + p_{10}P_{n,1}(t)\big] \quad ,1 \leq n < N \tag{5}$$

$$\frac{d}{dt}P_{n,1}(t) = -(\lambda + \mu + n\xi)P_{n,1}(t) + \mu P_{n+1,1}(t) + \lambda P_{n-1,1}(t) + (n+1)\xi\big[p_{01}P_{n+1,0}(t) + p_{11}P_{n+1,1}(t)\big] \quad ,1 \leq n < N \tag{6}$$

$$\frac{d}{dt}P_{N,0}(t) = -(\mu + N\xi)P_{N,0}(t) + \lambda P_{N-1,0}(t) + N\xi\big[p_{00}P_{N,0}(t) + p_{10}P_{N,1}(t)\big] \tag{7}$$

$$\frac{d}{dt}P_{N,1}(t) = -(\mu + N\xi)P_{N,1}(t) + \lambda P_{N-1,1}(t) \tag{8}$$

The initial condition is $P_{0,0}(t) = 1$.

4. Steady-state solution

We obtain the steady-state solution of the model from the equations (1) to (8) by using the matrix-decomposition method as:

$$\boldsymbol{P}_0 = \boldsymbol{\psi}_1 P_{0,1}, \boldsymbol{\psi}_1 = \frac{(\lambda + \boldsymbol{A}_{34}\boldsymbol{A}^{-1}{}_{44}\boldsymbol{A}_{43})}{\boldsymbol{A}_{23} - \boldsymbol{A}_{24}\boldsymbol{A}^{-1}{}_{44}\boldsymbol{A}_{43}}, P_{0,0} = \frac{\boldsymbol{\psi}_1 \boldsymbol{A}_{21}}{\lambda} P_{0,1}, \boldsymbol{P}_1 = -(\boldsymbol{A}_{34} + \boldsymbol{\psi}_1 \boldsymbol{A}_{24})\boldsymbol{A}^{-1}{}_{44} P_{0,1},$$

$$Q_{0,0} = \frac{\mu}{\lambda^2}\ \boldsymbol{\psi}_1 \boldsymbol{A}_{21} P_{0,1}, Q_{0,1} = \frac{\mu}{\lambda} P_{0,1},\ P_{0,1} = \frac{1}{\frac{\mu}{\lambda^2}\ \boldsymbol{\psi}_1 \boldsymbol{A}_{21} + \frac{\mu}{\lambda} + \frac{\boldsymbol{\psi}_1 \boldsymbol{A}_{21}}{\lambda} + \boldsymbol{\psi}_1 \boldsymbol{e} + \boldsymbol{1} - (\boldsymbol{A}_{34} + \boldsymbol{\psi}_1 \boldsymbol{A}_{24})\boldsymbol{A}^{-1}{}_{44}\boldsymbol{e}}$$

5. Transient Analysis

Now, we perform the transient analysis of the model. Due to the complex nature of the model equations, it is quite difficult to obtain the transient solution analytically. Therefore, we use the Runge-Kutta method of fourth-order to obtain the transient solution. The “ode45” function of MATLAB software is used to compute the transient numerical results.

Key performance indicators

We study the following key performance indicators:

- Expected queue size $L_q(t) = \sum_{n=0}^{N}(n)[P_{n,0}(t) + P_{n,1}(t)]$
- Expected delay in the queue $W_q(t) = \frac{L_q(t)}{\mu[1 - Q_{0,0}(t) - Q_{0,1}(t) - P_{0,0}(t) - P_{0,1}(t)]}$

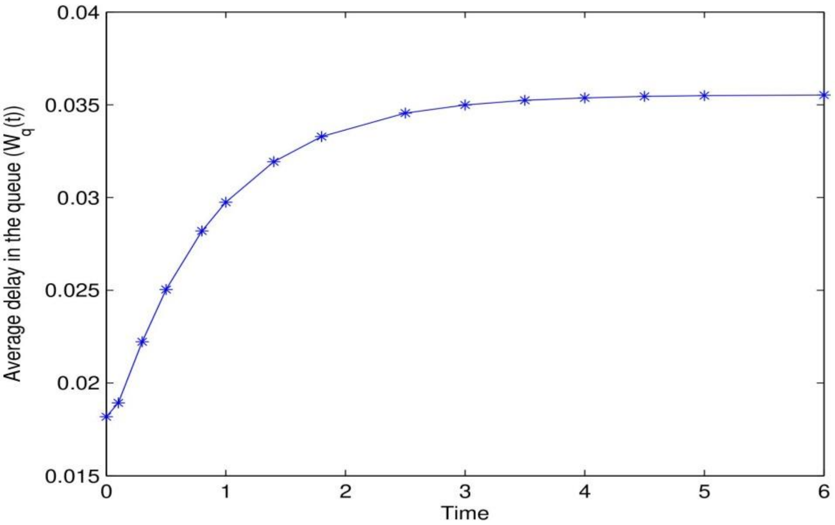

Figure 1: Average delay in the queue vs. Time.

The variation in average delay in the queue with time is presented in figure 1. One can observe that the average delay in the queue also increases as time progresses. But, after some time there is no change in average delay with time, that is, the system has achieved steady-state. In the figures 2 and 3, we compare three queuing models: M/M/1/N queuing model with correlated reneging, M/M/1/N queuing model with reneging and M/M/1/N queuing model. The above mentioned three models are compared with respect to the variation in average delay in the queue as shown in figure 2 with average arrival rate. It can be seen from the figure that the average delay in the queue of the M/M/1/N queuing model without reneging is higher than that of the M/M/1/N queuing model with correlated reneging. Further, the average delay in the queue in the case of M/M/1/N queuing model with reneging is lowest to the other two models. Thus, if the request dropping (reneging) in cloud computing systems is considered as correlated, then the average delay in the queue will be higher as compared to exponential reneging of requests.

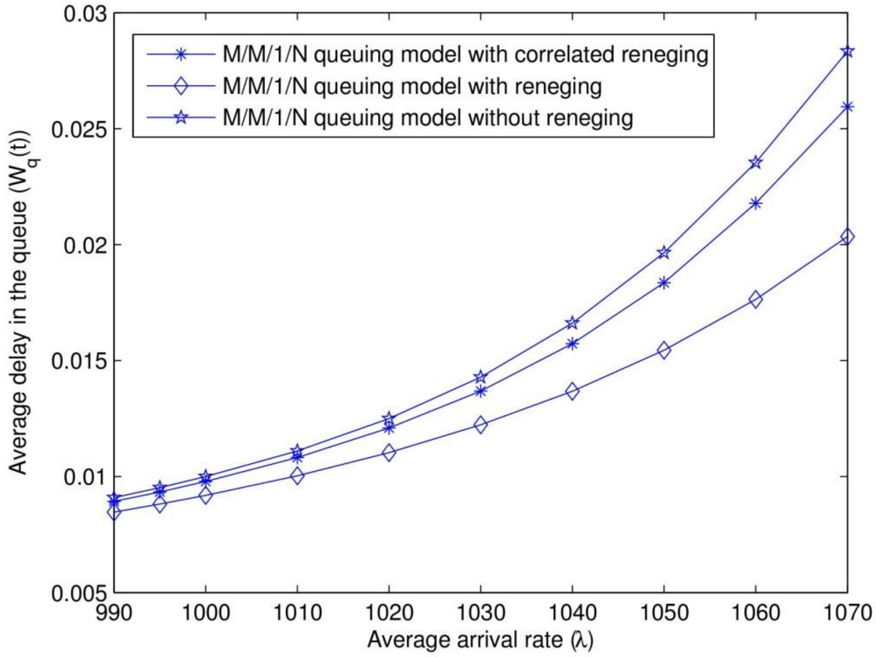

Figure 2: Average delay in the queue vs. Average arrival rate.

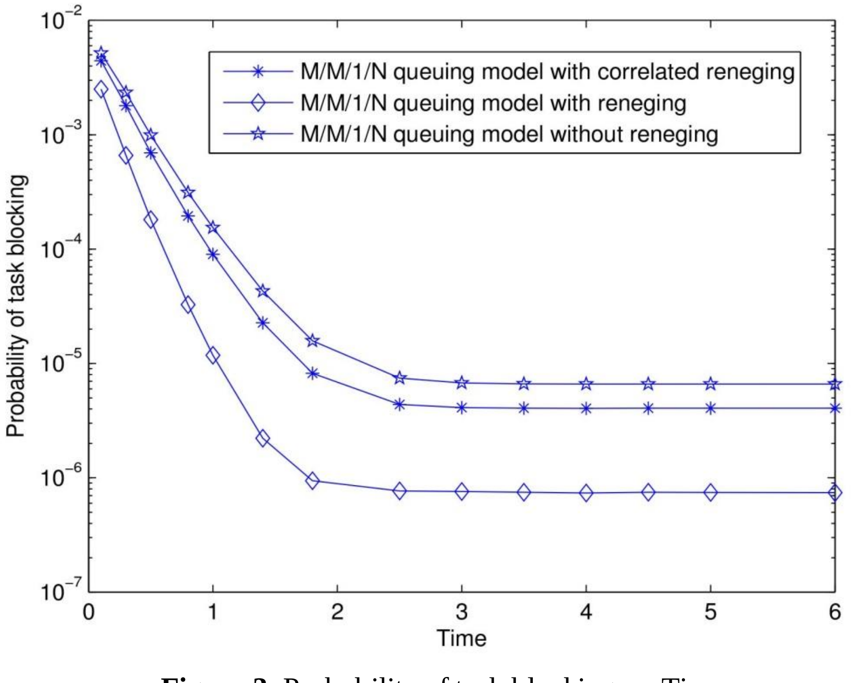

Figure 3: Probability of task blocking vs. Time

The variation in average delay in the queue with respect to the arrival rate as shown in figure 2 indicates that the performance indicators increases slowly with arrival rate until a certain value beyond which a small increase in the arrival rate or traffic load will result in the sharp or rapid increase in the average delay in the queue. Therefore, in designing, sizing and provisioning the load balancing server, it must be ensured that the average delay in queue be maintained within the slow variation regime to ensure QoS guarantee. The second comparison is carried out with respect to variation in the probability of task (request) blocking (P_{N,0}\left(t\right)+P_{N,1}\left(t\right)) with time as shown in figure 3. Initially, the task blocking probability in all the three models is high; it decreases slowly and after some time becomes stationary. The higher value of the task blocking probability at the initial stage is because of the initial condition we have taken\ P_{80,0}\left(0\right)=1, that is, there are 80 requests in the queue initially. Further, task blocking probability always remains higher in the case of M/M/1/N queuing model without reneging as compare to the M/M/1/N queuing model with correlated reneging. The task blocking probability is lowest in the case of M/M/1/N queuing model with reneging. This type of variation is due to the average queue sizes possessed by three queuing models.

6. Conclusion

In this paper, we have studied a single server, finite capacity Markovian queuing model with correlated reneging. We then demonstrate how they can be used in the design and performance evaluation of cloud computing data centres with correlated reneging of requests. The steady-state and the transient-state behaviours of the model are studied. We also investigated the influence of the design parameters such as the buffer sizes, the mean arrival rate, the mean processing speed, the average reneging rate on the key performance indicators (KPIs) such as response time (delay due to waiting), task blocking probability (due to buffer overflow), probability of immediate service, and the mean number of tasks in the system. Finally, some comparisons are performed which describe the effect of correlated request reneging over simple exponential reneging.

References

[1] Paya A and Marinescu D 2017 Energy-Aware Load Balancing and Application Scaling for the Cloud Ecosystem *IEEE Transactions on Cloud Computing* **1(5)** pp 15-27

[2] Bruneo D 2014 A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems *IEEE Transactions on Cloud Computing* **25** pp 560-69

[3] Chiang Y et al 2016 Performance and Cost-Effectiveness Analyses for Cloud Services Based on Rejected *IEEE Transaction on Services Computing* **9** pp 446-55

[4] Duan D, Yu S and Zhang A 2017 Cloud service performance evaluation: status, challenges, and opportunities – a survey from the system modeling perspective *Digital Communications and Networks* **3** pp 101-11

[5] Zanoon N, Kar J and Mishra M 2016 Mitigating Threats and Security Metrics in Cloud Computing *Internation Journal of Information Processing* **12** pp 226-33

[14] ElKafhali S and Salah K 2018 Modelling and Analysis of Performance and Consumption in Cloud Data Centers *Arabian Journal of Science and Enginering* **43** pp 7789-802

[6] Rodriguez V and Guillemin F 2016 Performance analysis of resource pooling for network function virtualization *17th International Telecommunications Network Strategy and Planning Symposium (Networks)* pp 158-63

[7] Chiang Y and Ouyang Y 2014 Profit Optimization in SLA-Aware Cloud Services with a Finite Capacity Queuing Model Mathematical Problems in Engineering *Hindawi Publishing Corporation*

[8] Nan X, He Y and Guan L 2014 Queueing model based resource optimization for multimedia cloud *J. Vis. Commun. Image R* **25** pp 928-42

[9] Kuaban G et al. 2020 A Multi-Server Queuing Model with Balking and Correlated Reneging With Application in Health Care Management *IEEE Access* **8** pp 169623-39

[10] Gupta S and Arora S 2018 Queueing system in cloud services management: a survey *International Journal of Pure and Applied Mathematics* **119** pp 12741-53

[11] Al-Seedyet R al 2009 Transient solution of the M/M/c queue with balking and reneging: a survey *Computers and Mathematics with Applications* **57** pp 1280-85

[12] Homsiet S al 2017 Workload Consolidation for Cloud Data Centers with Guaranteed QoS Using Request Reneging *IEEE Transactions on Parallel and Distributed Systems* **28** pp 2103-16

[13] Yang B et al 2009 Performance Evaluation of Cloud Service Considering Fault Recovery *IEEE International Conference on Cloud Computing* pp 571-76