International Journal of Software Engineering for Smart Device
Vol.6, No.1 (2019), pp.27-34
http://dx.doi.org/10.21742/ijsesd.2019.6.1.04

Investigation of Queuing Model Based Cloud Computing Application's Performance with Matlab Software

Pilla Srinivas

Dept. of Computer Science and Engineering, Dadi Institute of Engineering & TechnologyVisakhapatnam, AP, India
srinivasp3@gmail.com

Abstract

Cloud computing was the technology developed to store the data and support the users with the access to the data stored by charging a minimal amount for the storage of data and for providing necessary steps for storing the data and for providing security to the data that was stored. The content stored in various servers at various locations based on the type and size of the content. The content can be accessed to the users with valid registrations and a set of security verifications entered by the customers. The problems at hand give rise to the task of evaluating the performance of data center with various queuing models to understand the distribution of the performance parameters with arrival and service rates, traffic intensity, number of servers and the associated probabilities. The major contribution in the current article is to provide a mathematical model with cloud-based network. In order to analyze the performance of the currently considered, the steady-state behaviour of the current model designed and developed by using the MATLAB® environment. The models considered for evaluation for single servers include M/M/c and M/M/1 under the flow of FIFO and FCFS. The results observed from the model are encouraging and the results are displayed in the results section.

Keywords: *Cloud computing, Queuing models, Exponential distribution, M/M/c, M/M/1, FIFO, FCFS*

1. Introduction

The internet connection was a must in the field of cloud data processing and its related application areas [1]. The major advantage for the customers and the users was the crash report. Whenever a crash occurs in the system or the system connected to the cloud, the data can be still available as the actual data was stored in the servers of the cloud not in the actual systems that were being connected to the cloud environment [2][3]. The data will not be disturbed or any damage to the actual data as it was stored at various servers connected to each other and located at various locations. The users might be customers from various numerous companies, numerous servers and their related applications and the numerous networks which could be used to connect all these servers and the machines [4][5]. Here, the data was stored in various servers with various configurations and various operating systems and all these servers were located at various locations, so that the data can be secured even though a major damage or loss occurs for servers located at one place on the country or on earth.

Article history:
Received (June 9, 2019), Review Result (July 12, 2019), Accepted (September 10, 2019)

Copyright ⓒ 2019 GV Press

1.1. Cloud data centers

The user can utilize the services and download the data or the programs or the set of codes that were stored in various servers and located at various locations. The data can be downloaded or the utilized by locating at various places by simply having an authorized access to the set of customers [5]. The customers were based on the set of applications they were likely to use or they wish to use the other set of data in the coming future also. The effort kept by users for providing data to the servers, maintaining the data in the servers and providing security to the data servers were very minimum. He most of the tasks and the security step for providing the system and the machines and servers will be taken care by the service providers and the people whoever maintaining the services[6][7]. The users related to that particular center can able to download the data at any point in time. The servers and the machines at each data center were internally connected to each other such that to maintain the data uniquely and to provide best service to the customers [8]. The data that was stored at various data centers were being shared by various research organizations, remote processing applications and other related applications.

1.2. Queuing systems

The queuing systems which were in the form of theoretical model were intended to develop and provide the various set of models for predicting and estimating the performances of various systems subject to the random in nature of the systems. The history of the queuing theory was first started in the years of 1908 [1][7][9][8]. The Copenhagen Telephone Company asked the Agner K.Erlang to study and identify the holding times of a telephone switch. Within short span of time he identified a great discovery of the queuing systems. He identified that the number of telephone conversations and the holding time of each telephone unit can be able to fit into the Poisson and exponential distributions namely. This discovery had made a great change in the next future for the vast development of latest technologies based on the present queuing theory models. The analysis of the waiting lines can be analyzed easily with the help of queuing models [4]. The input data that we are going to supply to a system or a machine was measured for an extended period of time. The queuing systems will assume that the arrival times and service times are always random in nature.

The contents of the queuing system are the customer arrivals i.e., the number of users or the customers who were using or entering into the queue for getting the service or to see something or get set of taking something from a point. When the customers enter, if the line is free, they can easily get their service, if the line us busy then certain time will be taken for each user to get their application or their task will be completed.

Various set of assumptions were involved in queuing systems whenever we are trying to solve the issues and the problems related to the queuing systems and its related applications. The arrivals to a queuing system or the system that was being solved by using the queuing system model are independent distribution, exponential distribution. The queues which were in heavy lines or small lines will not discourage the customers.

2. Problem description and solution

The problems at hand give rise to the task of evaluating the performance of data center with various queuing models to understand the distribution of the performance parameters with arrival and service rates, traffic intensity, number of servers and the associated probabilities. The major contribution in the current article is to provide a mathematical model with cloud-based network. In order to analyze the performance of the currently considered, the steady-state behaviour of the current model designed and developed by using the MATLAB® environment. The models considered for evaluation for single servers include M/M/c and M/M/1 under the flow of FIFO and FCFS. Hence there is a necessity to analytically model the cloud servers/data centers using queuing systems and estimate the performance parameters.

3. System design

System design includes parameters, performance measures, stability and properties considered for the data center performance evaluation using queuing models. System parameters are,

a. It is customary to introduce some notation for the performance measures of interest in queuing systems.

b. *Number of customers in the system (L_S)*: In steady-state, the expected value of the state distribution gives the mean number of customers in the system.

c. *Number of customers in the queue (L_Q)*: In steady-state, the expected value of the state distribution in a queue gives the mean number of customers in the queue.

d. *Throughput ( )*: The throughput for a queuing system with infinite capacity is the mean number of customers processed in a unit of time, i.e. the departure rate. Since the departure rate is equal to the arrival rate (and assuming $\rho < 1$), the throughput is =c ρ. For a queuing system with finite capacity, there can be loss in the systems, and so the throughput can be less than the arrival rate. In this case, throughput is often denoted differently (e.g. as *S*) to distinguish it from the arrival rate.

e. *Response Time* (W_S): (or sojourn time) It is the total time a customer spends in the system.

f. *Waiting Time* (W_Q): It is the time a job spends in the queue waiting to be serviced. Therefore, response time is the sum of the waiting time (W_Q) and the service time(*1/)* for a customer i.e. $W_S = W_Q + (1/\mu)$

To evaluate the performance parameters of data center in cloud architecture, the programs consider the following input values:

- Interarrival rates,
- Service rates,
- Number of servers, c
- Maximum number of customers allowed, K

The list of output parameters of the programs is highlighted below:

- Length of customers in a system, LS
- Length of customers in queue, LQ
- Waiting time of customers in a system, WS

4. System implementation

Whenever if there is any uncertainty in arrival and service times of any system or application, the queuing models are the best-fitted application or the model which could be used to estimate the presentation of the systems for various services to the users or customers. The simplest possible (single-stage) queuing systems have the following components: customers, servers, and a waiting area (queue). An arriving customer is placed in the queue until a server is available. For the present work it is assumed that customers are served in the order in which they arrive in the system (First-Come-First-Served or FCFS).

4.1. Queuing model - M/M/1

The M/M/1 queue has interarrival times which are exponentially distributed with parameter and also service times with Erlang distribution with parameter. The system has only a single server (c=1) and uses the FIFO service discipline with FIFO and FCFS flow of mode. The analysis of the M/M/1 queue is similar to that of the M/M/1queue. Steady-state performance measures for the above model are:

Steady state performance measures for the above model are:

$$R(t) = R_1.\beta_1 \left[1 - \exp\left[\alpha_1 \sum_{k_1=1}^{m_1} \sum_{r=1}^{k_1} \frac{{}^{m_1}C_{k_1} p_1^{k_1} (1-p_1)^{m_1-k_1}}{1-(1-p_1)^{m_1}} {}^{k_1}C_r \left(-1\right)^{3r} \frac{\left(1-e^{-r\beta_1 t}\right)}{r\beta_1} \right]\right] \qquad (1)$$

4.2. Queuing model - M/M/c

The M/M/c queue has the exponential Interarrival time and service time distributions as the M/M/c queues, however, there are c servers in the system, and waiting line is infinitely long and uses the FIFO and FCFS service discipline. There is no explicit input queue; an incoming call must be abandoned forthwith, Steady-state performance measures for the above model are:

$$P(t) = R_1.\beta_1 \left[1 - \exp\left[\alpha_1 \sum_{k_1=1}^{m_1} \sum_{r=1}^{k_1} \frac{{}^{m_1}C_{k_1} p_1^{k_1} (1-p_1)^{m_1-k_1}}{1-(1-p_1)^{m_1}} {}^{k_1}C_r \left(-1\right)^{3r} \frac{\left(1-e^{-r\beta_1 t}\right)}{r\beta_1} \right]\right] \qquad (2)$$

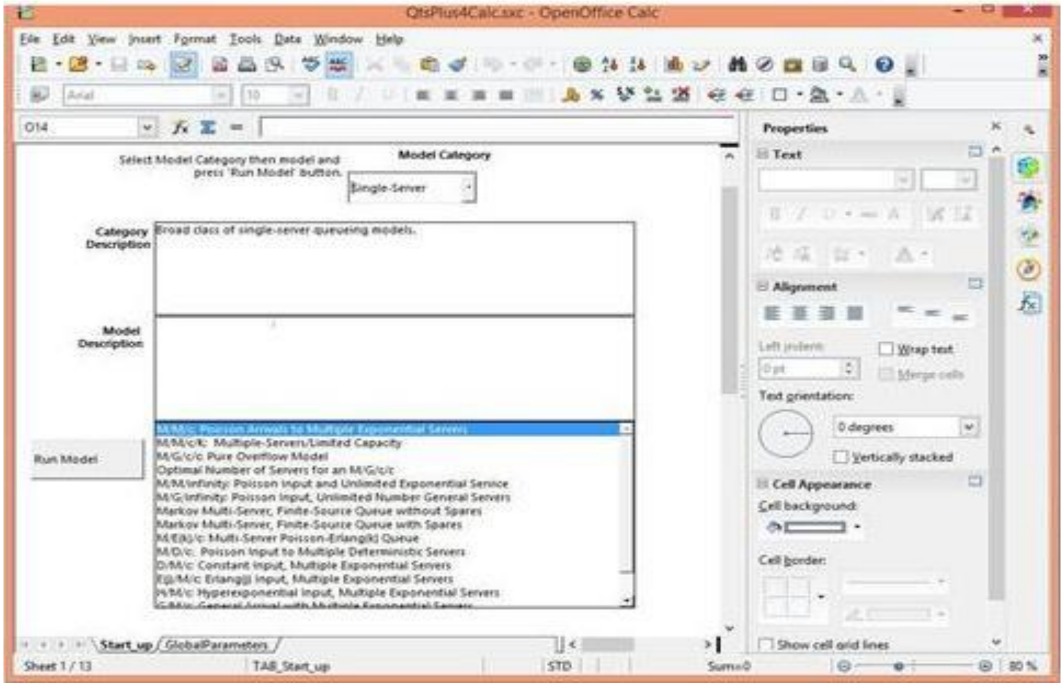
4.3. Validation

Validation studies for the present MATLAB codes generated was undertaken using the existing queuing softwares and published data. It was considered to validate the program first and then evaluate performance parameters for the input data of interest. The above approach was followed in the present work to prove the validity and reliability of the results generated. A list of validation software's published data utilized is highlighted below:

4.4. QtsPlus4Calc Software

QtsPlus4Calc, release 2006 is a freeware developed by Donald gross and Carl M. Harris of George Mason University. This software provides a platform to evaluate performance of various queuing models. A sample screenshot of the software environment is given at [Figure 2]. The calculator includes single server, multi-server, priority, bulk and network models.

Numerical Solutions to Queuing Systems *(http://queueing-systems.ens-lyon.fr):* Numerical solutions to queuing systems software provide a platform to evaluate performance of various queuing models with generalized service distributions. A sample screenshot of the software environment is given at [Figure 2]. The calculator includes single server models.



Select Model Category then model and press 'Run Model' button.

Model Category: Single-Server

Category Description: Broad class of single-server queueing models.

Model Description

Run Model

M/M/c: Poisson Arrivals to Multiple Exponential Servers
M/M/c/K: Multiple-Servers/Limited Capacity
M/G/c/c Pure Overflow Model
Optimal Number of Servers for an M/G/c/c
M/M/infinity: Poisson Input and Unlimited Exponential Service
M/G/infinity: Poisson Input, Unlimited Number General Servers
Markov Multi-Server, Finite-Source Queue without Spares
Markov Multi-Server, Finite-Source Queue with Spares
M/E(k)/c: Multi-Server Poisson-Erlang(k) Queue
M/D/c: Poisson Input to Multiple Deterministic Servers
D/M/c: Constant Input, Multiple Exponential Servers
E(j)/M/c: Erlang(j) Input, Multiple Exponential Servers
H/M/c: Hyperexponential Input, Multiple Exponential Servers
G/M/c: General Arrival with Multiple Exponential Servers

Figure 2. QtsPlus4Calc environment screenshot

5. Results and discussion

Based on the queuing models discussed in the previous chapters (4 and 5), input parameters limits were identified from literature for performance evaluation of small cloud computing data center farm (i.e. number of servers, *c* were limited to 2 and 4).The inter-arrival and service rates are chosen for a range of traffic intensity (or utilization) varying from 0 - 1. Each time service rate was varied to values 1, 2 and the respective performance was evaluated. Input parameter limits are given at [Table 1].

Table 1. Performance of data center - M/M/1, model = 1 with FCFS

Performance of data center - M/M/1 model= 1, *Er* = 2										
				Mean Number of Packets		Waiting Time			Variance	
		ρ	*Er-parameter*	*Ls*	*Lq*	*Ws*	*Wq*	*Var(Ls)*	*Var(Ws)*	*Var(Lq)*	*Var(Wq)*
1	0	0	2	0.00	0.00	1.00	0.00	0.00	0.50	0.00	0.00
1	0.1	0.1	2	0.11	0.01	1.08	0.08	0.11	0.62	0.01	0.12
1	0.2	0.2	2	0.24	0.04	1.19	0.19	0.27	0.79	0.08	0.29
1	0.3	0.3	2	0.40	0.10	1.32	0.32	0.49	1.03	0.26	0.53
1	0.4	0.4	2	0.60	0.20	1.50	0.50	0.83	1.42	0.67	0.92
1	0.5	0.5	2	0.88	0.38	1.75	0.75	1.39	2.06	1.52	1.56
1	0.6	0.6	2	1.28	0.68	2.13	1.13	2.45	3.27	3.29	2.77
1	0.7	0.7	2	1.93	1.23	2.75	1.75	4.81	5.90	7.30	5.40

Copyright © 2019 GV Press

1	0.8	0.8	2	3.20	2.40	4.00	3.00	11.84	13.50	18.40	13.00
1	0.9	0.9	2	6.98	6.08	7.75	6.75	51.58	55.06	72.14	54.56
1	1	1	2	Inf	Inf	Inf	Inf	Inf	NaN	Inf	NaN

Table 2. Performance of data center - M/M/c model=2, *Er* = 4 with FCFS

	Mean Number of Packets					Waiting Time			Variance		
	Λ	ρ	Er-parameter	Ls	Lq	Ws	Wq	Var(Ls)	Var(Ws)	Var(Lq)	Var(Wq)
2	0	0	4	0.00	0.00	0.50	0.00	0.00	0.06	0.00	0.00
2	0.2	0.1	4	0.11	0.01	0.53	0.03	0.11	0.08	0.02	0.02
2	0.4	0.2	4	0.23	0.03	0.58	0.08	0.25	0.11	0.13	0.05
2	0.6	0.3	4	0.38	0.08	0.63	0.13	0.43	0.15	0.47	0.08
2	0.8	0.4	4	0.57	0.17	0.71	0.21	0.70	0.21	1.26	0.15
2	1	0.5	4	0.81	0.31	0.81	0.31	1.13	0.32	2.91	0.25
2	1.2	0.6	4	1.16	0.56	0.97	0.47	1.91	0.52	6.28	0.45
2	1.4	0.7	4	1.72	1.02	1.23	0.73	3.60	0.96	13.50	0.90
2	1.6	0.8	4	2.80	2.00	1.75	1.25	8.56	2.25	31.60	2.19
2	1.8	0.9	4	5.96	5.06	3.31	2.81	36.35	9.38	103.59	9.32
2	2	1	4	Inf	Inf	Inf	Inf	Inf	NaN	Inf	NaN

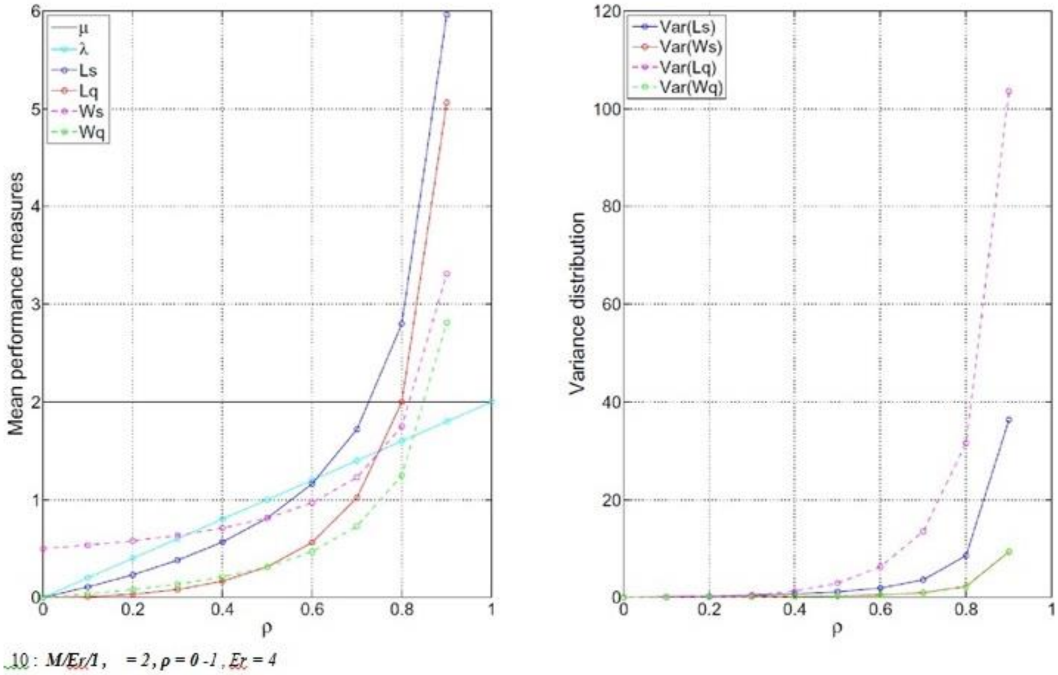


Figure 3. Graphical representation of second model M/M/1 for Variance and performance measures for FCFS mode

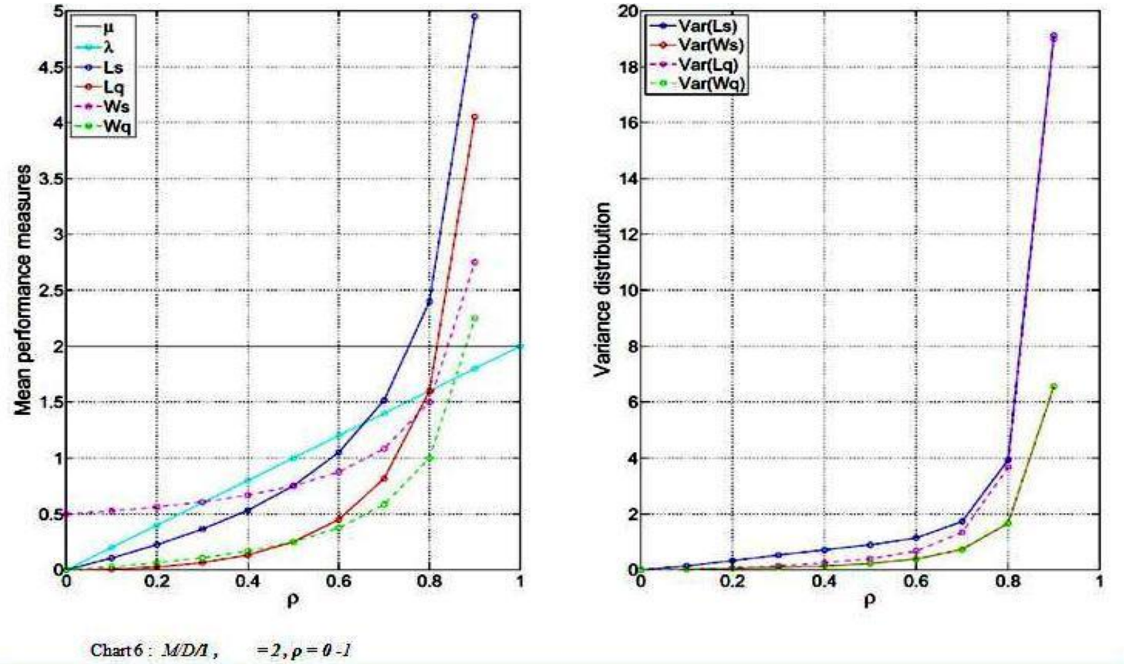


Figure 4. Graphical representation of first model M/M/c for Variance and Mean performance measures for FCFS mode

6. Conclusions

Performance evaluation for single servers indicate that as the service rate (r) increases for a constant range of traffic intensity (***ρ***) only waiting times of customers in the system (W_S) and queue (W_Q) decreases, whereas the length of customers in system (L_S) and queue (L_Q) remain unchanged as it is independent on. For the same input parameters *M/D/1* model shows optimum performance in terms of queue lengths and waiting times followed by *M/M/c* with the flow of FIFO and FCFS modes. Performance of *M/M/1* shows detrimental nature when compared with other queuing models, which is attributed to higher value of C_{oV}. Performance evaluation of small cloud computing data center is discussed with the theory based on queuing systems. Single server and multiple server models are presented along with their formulations for performance parameters. MATLAB programming/code generation and implementation for performance evaluation of cloud computing data server farm is accomplished. Comparisons among various models are attempted and relevant observations are highlighted.

References

[1] Hamzeh Khazaei, Jelena and Vojislav, “Performance analysis of cloud computing centers using M/G/M/M+R queuing systems,” IEEE Transactions on parallel and distributed systems, vol.23, pp.936-943, **(2012)** DOI: 10.1109/TPDS.2011.199

[2] Hamzeh Khazaei, “Performance modeling of cloud computing centers,” Doctoral dissertation, The University of Manitoba, Canada, Oct., **(2012)**

[3] B. Yang, F. Tan, Y. Dai, and S. Guo., “Performance evaluation of cloud service considering fault recovery,” First International Conference on Cloud Computing (CloudCom) 2009, pp.571-576, **(2009)** DOI: 10.1007/978-3-642-10665-1_54

[4] Myron Hlynka and Samantha Molinaro, "Comparing expected wait times of an M/M/1 queue," Department of Mathematics and Statistics, University of Winsor, June, **(2010)**

[5] Niloofar Khanghahi and Reza Ravanmehr, "Cloud computing performance evaluation: issues and challenges," International Journal on Cloud Computing Services and Architecture, vol.3, no.5, pp.29-41, Oct, **(2013)**

[6] Ivo Adan and Jacques Resing, "Queuing systems," Eindhoven University of Technology, The Netherlands, March, **(2015)**

[7] Dr. Janos Sztrik, "Basic queuing theory," University of Debrecen, Faculty of Informatics, Dec, **(2012)**

[8] T.SaiSowjanya, D.Praveen, K.Satish, and A Rahiman, "The queuing theory in cloud computing to reduce the waiting time," IJCSET, vol.1, no.3, pp.110-112, April, **(2011)**

[9] Chandrakala and Jyothi Shetty, "Survey on models to investigate data center performance and QoS in cloud computing infrastructure," First International Conference on Recent Advances in Science & Engineering, **(2014)**