Journal of Engineering Education Transformations ,
Volume 31 , No. 4, April 2018, ISSN 2349-2473, eISSN 2394-1707

Software Tool For Teaching Fundamentals of Queuing Theory

Rashpal Ahluwalia
Industrial and Management Systems Engineering Department
West Virginia University, Morgantown, WV, USA
rashpal.ahluwalia@mail.wvu.edu

Abstract: This paper presents a software tool that can be utilized to teach fundamentals of queuing theory. Arrival and service rates can be constant or from Exponential, Erlang, Hyper-exponential, or a general distribution. Number of servers, system capacity, and calling population size can be varied. The user can also select queue discipline and unit of time. The program computes average server utilization factor, average number of items in the queue, average number of items in the system, average time spent in the queue, average time spent in the system, probability of n items in the system, cost of service, and cost of waiting. The software tool can be utilized for in class exercises and for "what if" type analysis.

Keyword: Queuing Models, Queue Performance, Arrival Distribution, Service Distribution, Queuing Software

1. Background

A typical M/M/s queuing system is shown in Figure 1. Customers form a single queue to be serviced by any one of s servers. For the case of a single server we have an M/M/1 queuing system. The M/M/s system is different from the s*M/M/1 system where s separate queues are formed and each queue is served by a single server [1].

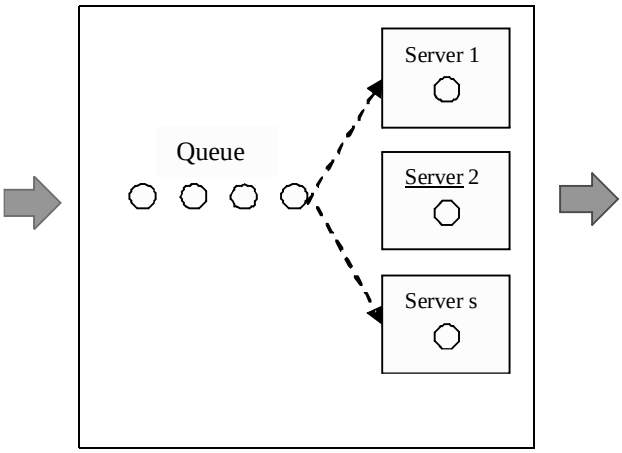

Figure 1: Single queue, multiple servers

Arrivals are assumed to follow a Poisson (discrete) distribution with parameter λ (average arrival rate). Equation (1) gives the probability of X arrivals, during a time interval.

Service times follow an Exponential (continuous) distributed with parameter μ (average service rate). The probability that a customer will be serviced during a time interval is given by equation (2). For a queuing system to be stable, arrival rate should be less than all of the service rates, that is, $\lambda < s*\mu$.

2. Queue Performance Measures

Common queue performance measures are:

Rashpal Ahluwalia
Industrial and Management Systems Engineering Department
West Virginia University, Morgantown, WV, USA
rashpal.ahluwalia@mail.wvu.edu

ρ Queue utilization factor

Lq Average number of items in the queue

L Average number of items in the system

Wq Average time an item spends waiting in queue

W Average time an item spends in the system

P0 Probability of zero items in the system

Pn Probability of n items in the system

Given arrival rate (λ), service rate (μ), and number of servers (s), queue performance measures can be computed as follows:

In order to determine the cost associated with a given queuing system one needs to identify the unit service cost (USC) and unit waiting cost (UWC). The overall service cost, waiting cost, and total cost can then be calculated according to equations (11), (12), and (13), respectively.

Service Cost = s * USC …… (11)

Wait Cost = Lq * UWC ……..…….. (12)

Total Cost = Service Cost + Waiting Cost … (13)

3. Program Implementation

A computer program in Visual Basic 2013 [2] was implemented to compute queue performance measures. A numeric example of a coffee shop as described in [3] was used to test the program. Table 1 shows the input parameters of the queuing system.

Figure 2 shows the output of the program for values of

Table 1: Input Parameters

Arrival rate (λ)	4	Customers/hour
Service rate (μ)	2	Customers/hour
Unit Service Cost (USC)	15	$/hour
Unit Wait Cost (UWC)	60	$/hour

s ranging from 1 to 6. Program pseudo code is shown in Appendix I. Top two lines of Figure 2 show the input parameters: 1) arrival rate, 2) service rate, 3) unit service cost, 4) unit wait cost, and 5) upper bound on number of servers. The user can select time unit associated with arrival rate from a drop down menu. Current choices are /minute, /hour, /day, /week, and /month. Additional time units can easily be added. Once the user selects the unit for arrival time the program automatically assigns the selected unit to service rate, unit service cost, and unit wait cost. This avoids the likely hood of choosing incorrect time units for these parameters. The data grid in Figure 2 shows queue performance measures for values of s ranging from 1 to the user specified upper limit. Description of the queue performance measures for the final value of s is shown below the data grid. Performance measures associated with different values of s enable the user to select the scenario that minimizes total cost. Table 2 summarizes the various parameters for 3, 4, 5, and 6 servers. The queuing system is unstable for the case of s = 1 and 2.

Server utilization for the four cases was 66.67%, 50.00%, 40.00%, and 33.33%, respectively. The total associated cost was $98.33, $70.43, $77.39, and $90.54, respectively. For the given dataset the case of four servers is most economical. Similarly, the impact of other parameters (arrival rate, service rate, unit service cost, and unit wait cost) can also be evaluated.

4. Conclusions

Table 2: Queue Performance for s = 3,4,5, and 6

	s = 3	s = 4	s = 5	s = 6
.	66%	50%	40%	33%
L_q	0.889	0.174	0.040	0.009
L	2.889	2.174	2.040	2.009
W_q	0.222	0.043	0.010	0.002
W	0.722	0.543	0.510	0.502
P_0	0.111	0.130	0.134	0.135
Serv. cost	$45.00	$60.00	$75.00	$90.00
Wait cost	$53.33	$10.43	$2.39	$0.54
Total cost	$98.33	$70.43	$77.39	$90.54

Queuing theory is applicable to a wide variety of situations, ranging from queues at a barber shop to flow of packets in a high-speed packet switching network. Most engineering and business courses that cover this topic simply introduce the rudimentary formulas, followed by homework problems. The paper suggests that queuing theory be taught as a decision-making tool. The students should be presented with a real-world queuing problem and asked to select an appropriate model. Once they have the model, they should identify its parameters and determine the values that will achieve a predetermined level of queue performance. Such

analysis can be time consuming if are carried out by hand calculations.

This paper presented a software tool with a variety of queuing models. The students should be asked to select a model appropriate for a given problem statement. They should then be asked to study the impact of model parameter on queue performance measures. Such "what-if" type analysis can help students gain an intuitive understanding of the model. In addition, the students could also be asked to write a computer program for a particular model.

References

[1] Gross, D., 2012, Fundamentals of Queuing Theory, Joh Wiley, 4th edition, ISBN-13:978-0471791270

[2] Visual Studio 2013, https://msdn.microsoft.com/en-us/library/dd831853(v=vs.120).aspx

[3] Jesse, E., Queue modeling in Excel, https://www.youtube.com/watch?v=irMhkCGXOXs&t=660s

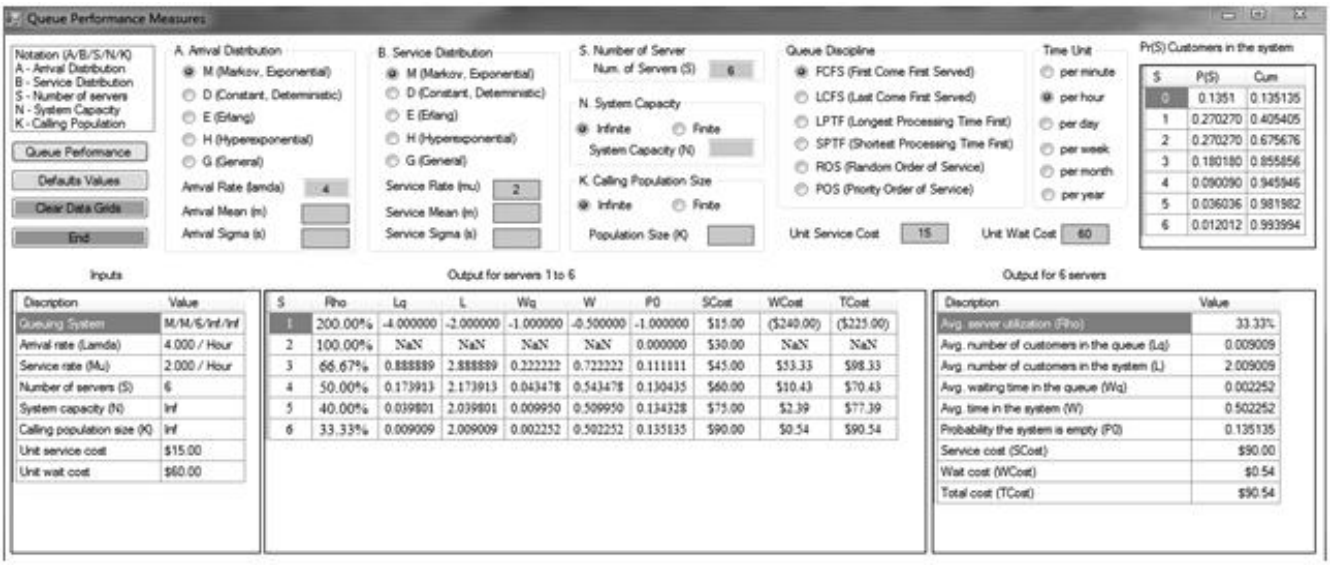

Figure 2: Software Screenshot for an M/M/s Queue

Appendix I: Pseudo Code

Input Variables

lamda	Arrival rate (double)
mu	Service rate (double)
s	Num. of servers (integer)
USCost	Unit service cost (double)
UWCost	Unit wait cost (double)
T0, T1, T2, R	Temporary variables (double)

Computed Values

Rho	Server utilization (double)
Lq	Avg. # in queue (double)
L	Avg. # in system (double)
Wq	Avg. time in queue (double)
W	Avg. time in system (double)
P0	Prob. of 0 in queue (double)
PN	Prob. of N in system (double)
SCost	Total service cost (double)
WCost	Total wait cost (double)
TCost	Total cost (double)

Performance Measures

```
T1      =1,     n = 1
Do While n <= s
T0      = Ri / i!
T2      = T0 / (1 - lamda / (n * mu))
P0      = 1 / (T1 + T2)
Rho     = lamda / (n * mu)
Lq      = P0 * T0 * Rho / ((1 - Rho) * (1 - Rho))
L       = Lq + R
Wq      = Lq / lamda
W       = Wq + (1 / mu)
SCost   = i * USCost
WCost   = Lq * UWCost
TCost   = SCost + WCost
Show values of Rho, Lq, L, Wq, W, P0,
SCost, WCost, and TCost
T1      = T1 + T0
n       = n + 1
Loop
Show final values of Rho, Lq, L, Wq, W, P0, SCost,
WCost, and TCost
```