Performance Analysis of a Multi-Server Queueing System

Md Farhan Ishmam[1] and Md Toshaduc Rahman[2]

[1]Department of Computer Engineering, Islamic University of Technology, Gazipur, Bangladesh
farhanishmam@iut-dhaka.edu

[2]Department of Computer Engineering, Islamic University of Technology, Gazipur, Bangladesh
toshaduc@iut-dhaka.edu

Abstract

Multi-Server Queueing System is similar to the single server queueing system but passes the departing item from one server to the queue of the next server. In this paper, we address the performance of this system by plotting the traffic intensity against output statistics. We analyze how the output variables differ and thereby, state the characteristics of the produced graphs.

Keywords

Multi-Server Queueing System (MSQS), Simulation, Arrival Event, Departure Event, Exponential Distribution, Simulation Entity, Queue, Queueing Delay, Server Utilization, Traffic Intensity

1. Introduction

The Multi-Server Queueing System is a system consisting of two or more interconnected servers, with one or more queues. A multi-server queueing system can be more complex than a single server queueing system as the items from one server has to be passed to the next server. Usually, a multi-server queueing system varies from simpler designs of two servers that are linearly connected to more complex designs such as the Job Shop Model. For this experiment, we will look at two linearly connected servers with their separate queue. This can be considered as the simplest form of a Multi-Server Queueing System.

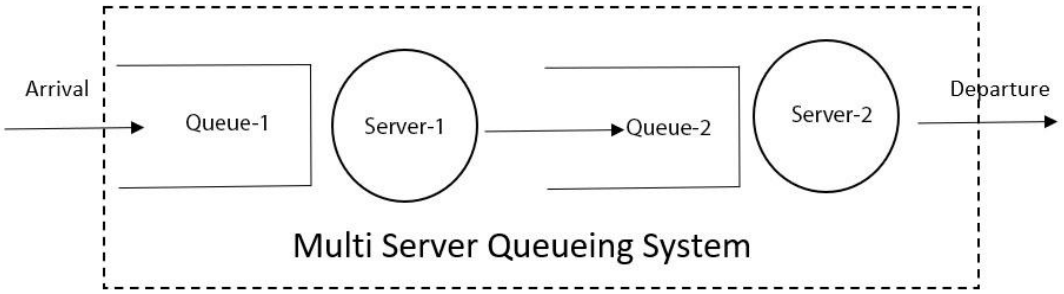
Our system consists of two servers and two queues. The first server receives items at an arrival rate following the exponential distribution and sends out items at a departure rate following the exponential distribution. The means of both distributions can be set manually in the program. The second server has no independent arrival rate. Instead, it only receives the items departed from the first server. The second server also has its own departure rate following the exponential distribution. The mean of the arrival rate of the second server is set to zero, and the mean of the departure rate is manually set to a fixed value.

In the upcoming sections, we go through the description of our multi-server queueing system, followed by the description of the simulation program, simulation results, graphical analysis and the conclusion of this paper.

2. System Description

The system is the combination of two single server queueing systems. The systems are linearly connected like a linked list. The model will take sets of arrival and departure times for each system as input and provide us the output statistics average queue waiting time, average queue length, and average server utilization. As the output of the first server is passed to the second one, the second server has no independent arrival time of its own, i.e. the arrival mean is set to zero.

Two trace files are also generated which keeps track of all the events occurring at the particular simulation clock time. The output statistics are generated in a report file which is calculated separately for each server. The output statistics of the reports depend on the manually set mean of the arrival and departure rate for each server.



3. Simulation Program Description

The program will consist of four major classes from the single server queueing system - **the event class, the server class, the scheduler class, and the queue class**. However, in addition to these classes, there will be a **SimEntity class** and a **ServiceFacility class**.

The major classes function similarly to the Single Server Queueing System classes. The server class is however modified as it inherits from the SimEntity class. The SimEntity class is basically a linked list of many servers. Each object has a pointer to the previous and next members. Functions are also defined to send and receive items from other instances of the class.

The ServiceFacility class is a composition of all the servers of the system. This class is helpful for the instantiation and initialization of all the servers, setting their arrival and departure times, and generating appropriate report files. A single ServiceFacility object has to be created and set to initialize the servers of the whole system.

To summarize the multiserver queuing system works similar to the single server queueing system, but has multiple instances of the queue and the server class. The two new classes SimEntity and ServerFacility helps to link the servers together and manage them easily. Each server independently generates its output statistics. The means are manually set to get various values of traffic intensity and to find the corresponding output statistics for those values.

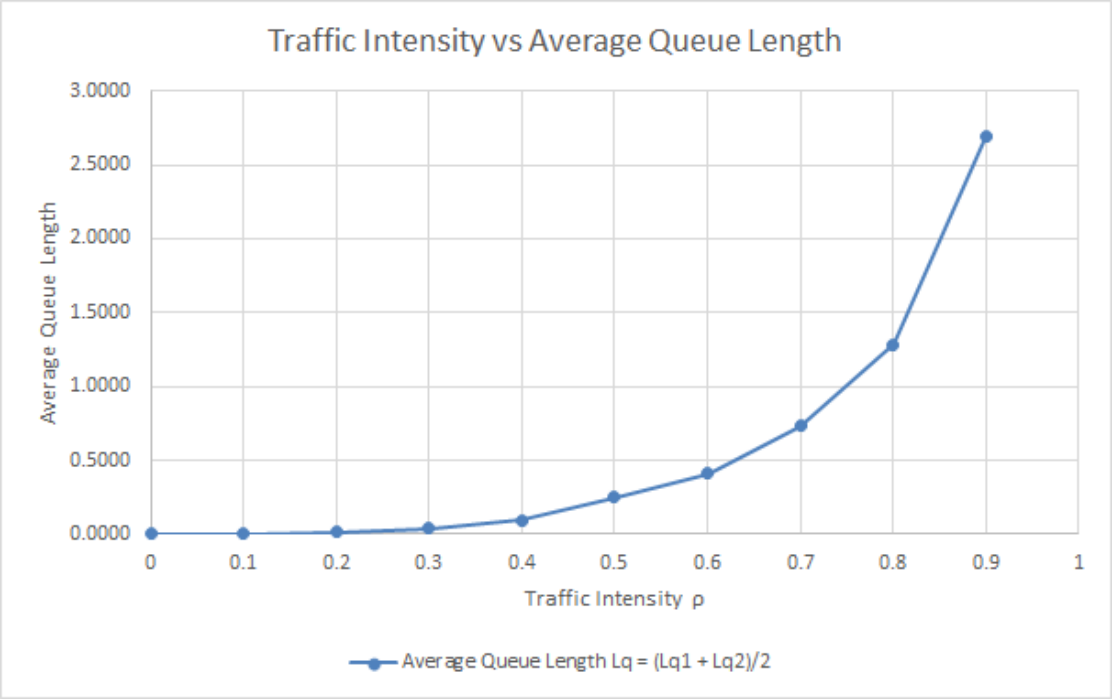
4. Simulation Results

The simulation results were taken by running the simulation 10 times for different values of traffic intensity ρ. The departure rate of server-1 is equal to the departure rate of server-2 and was taken at increments of 1. The departure rate of server-2 was fixed to 10 and by changing the arrival rate of server-1, we got different values of traffic intensity for server-1 and server-2. By multiplying the traffic intensity of server-1 by that of server-2, we get the system traffic intensity. The traffic intensity ranges from 0 to 0.9 with a 0.1 increase in each step. For various values of the traffic intensity, we get the average queue length, server waiting time, and queue waiting delay for each server. Then we take the average of all the servers for each output statistic.
The data from the simulation is given below:

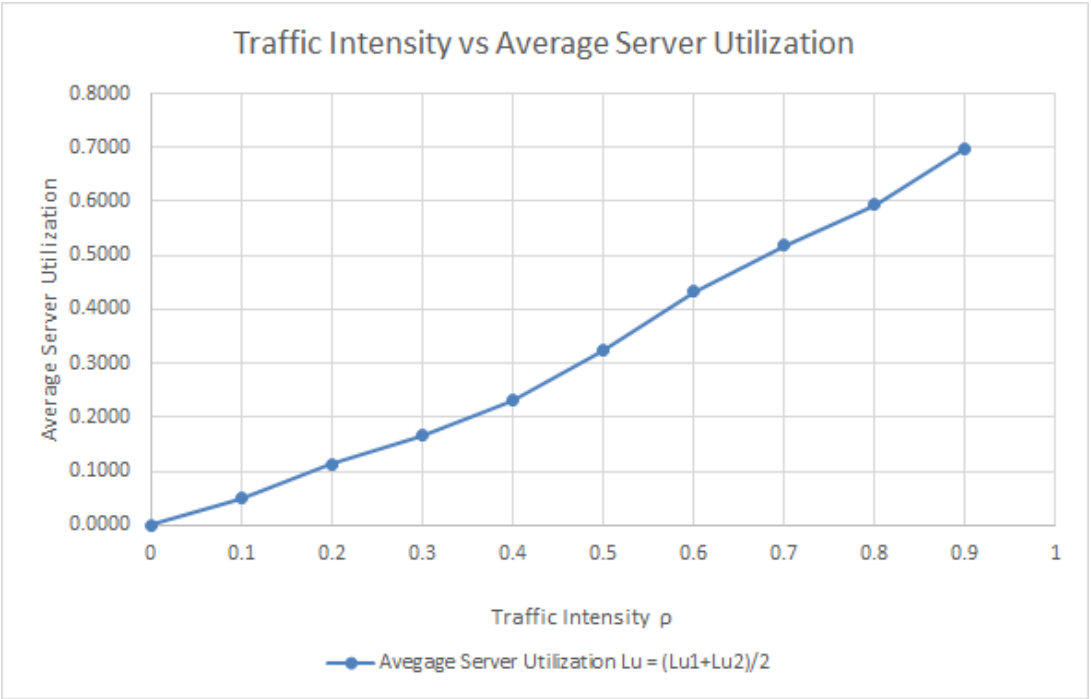
Name	Symbol, Eq	1	2	3	4	5	6	7	8	9	10
Server-1 Arrival Rate	λ1	0	1	2	3	4	5	6	7	8	9
Server-1 Departure Rate	μ1	1	2	3	4	5	6	7	8	9	10
Server-2 Arrival Rate	λ2	1	2	3	4	5	6	7	8	9	10
Server-2 Departure Rate	μ2	10	10	10	10	10	10	10	10	10	10
Server-1 Traffic Intensity	ρ1 = λ1/μ1	0	0.5	0.6667	0.75	0.8	0.8333	0.857	0.875	0.8889	0.9
Server-2 Traffic Intensity	ρ2 = λ2/μ2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Traffic Intensity	ρ = ρ1* ρ2	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Simulation Results											
Average Queue Length	Lq = (Lq1 + Lq2)/2	0	0.0024	0.0152	0.0413	0.0954	0.2500	0.4117	0.7338	1.2821	2.6901
Avegage Server Utilization	Lu = (Lu1+Lu2)/2	0	0.0505	0.1128	0.1659	0.2309	0.3238	0.4336	0.5182	0.5944	0.6982
Average System Waiting Time	W_s	0	0.0138	0.0518	0.0830	0.1159	0.1598	0.1936	0.2679	0.3724	0.5630
Average Queue Length	Lq = (Lq1 + Lq2)/2	0	0.0024	0.0152	0.0413	0.0954	0.2500	0.4117	0.7338	1.2821	2.6901

5. Graphical Analysis of Simulation Results

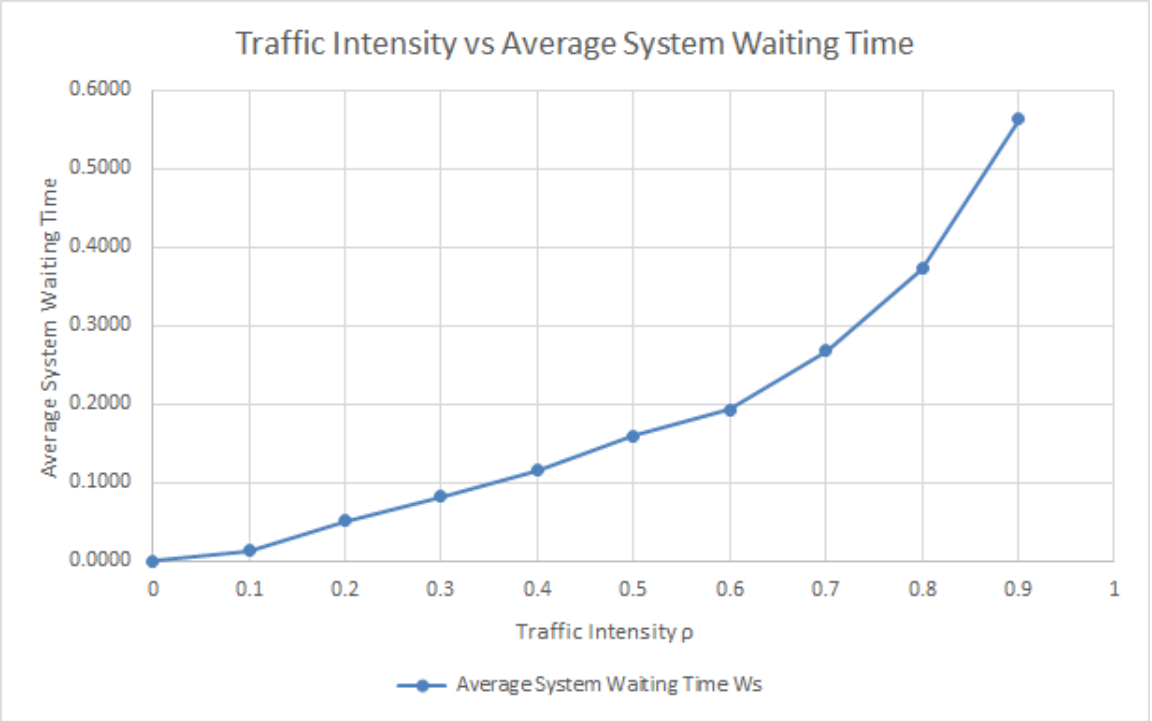
Average Queue Length: The queue length increases exponentially as shown in the graph. The queue length is really low at the lower values of traffic intensity as most of the time the servers stay idle. The value jumps drastically at higher values of traffic intensity as both of the servers are at nearly full utilization resulting in longer queues.



Average Server Utilization: The average server utilization is more or less linear with time. The server utilization has a much lower slope than the single server queuing system as the servers are more inactive at lower traffic intensity. But the slope rises quickly at greater values.



Average System Waiting Time: Similar to the average queue length, the average waiting time increases exponentially. However, the increase in waiting time is higher than the single server queueing system and increases sharply. That is due to the fact that there are two servers and two queues involved, the waiting time is much higher.



6. Conclusion:

The performance of the multi-server queuing system is identical to that of a single server queuing system. The average values tend to be lower for the server utilization and queue length than the single server counterparts. But, the average waiting time was higher. However, the intermediate arrival and departure means (the departure mean of server-1 or the arrival rate of server-2) can vary the result significantly. The plotting against traffic intensity keeps various other factors constant which can drastically vary the results. Plotting against two different values of traffic intensity for server-1 and server-2 might give us more insights if we vary the traffic intensities for both servers at a similar rate. But overall, we got expected results from the simulation program and the performance was satisfactory.

References:

[1] Simulation Modeling and Analysis by Averill Law, 4th Edition

[2] Discrete Event Simulation: A First Course by Lawrence Leemis, December 2004 Revision