REPORT: Simulation on different versions of a queue

Submitted by: Shivangi Saklani(16I190008) and Aakash Banik(16I190010)

Objective: *Come up with a simulation model that helps us in deciding the best queuing system to implement in case of variation in types of customers and their arrival rates and service times.*

Setting:

We have tried to consider three different types of queues in a two server system and we have assumed that the customers are of two types, Type I with daily ridership in the suburban railways and Type II with special demands (rare travellers). These type I customers enter the system at rate λ_1 and get serviced at a rate of μ_1. Similarly, the type II customers enter the system at rate λ_2 and get serviced at a rate of μ_2.

The three different types of queues are:

1. The first case deals with a system having single line (common queue) with two servers and a moderator to direct the first person in the queue to the available server out of the two.
2. The second case deals with two servers- two queue system i.e. each server has a separate queue but the two servers are exactly identical. Any person entering the system can join any queue.
3. The third case deals with two servers- two queue system but with specialized service for each type of customer at one dedicated counter.

Performance Measure:

1. *Average Waiting Time in queue*
2. *Probability that average waiting time exceeds tolerance levels.*

Assumptions:

We have made the following assumption which hold for all three queuing systems:

1. The system follows First Come First Serve (FCFS) discipline in each queue.
   (For cases when there are two queues, it doesn't follow FCFS in the system but in individual queues in front of the two servers).
2. An arrival in the system and a departure from a server can happen at the same time with a small difference Ɛ, since the time intervals are continuous.
3. The system in all cases is M/M/2
4. The total number of customers visiting the system in a day is 5000.
5. The results of this simulation are calculated over a span of a year.
6. The waiting time for first two customers in all systems is 0 i.e. the servers are ready before their first customer arrival.
7. Jockeying isn't allowed
8. Departure happens before arrival in any queue in subsequent times.

Approaching Model 1:

We approach this single queue model with the following pseudo algorithm:

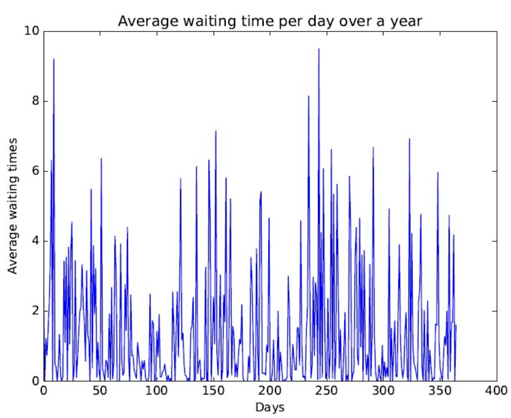
1. Generate the inter-arrival times t_i randomly by generating another random number say r between 0 and 1 and solving for x in the equation F(x)=r, where F(x) represents the cdf of exponential distribution with rate depending on the type of customer.
2. t_i represents the inter-arrival times in the system and T_i stores the time of arrival of i th customer in the system which is simply calculated as the cdf of inter-arrival times.
3. S_1 is the array consisting of all the system departure times from server 1 and similarly S_2.
4. The waiting times WQ_i is calculated by taking the minimum of the last two departures from the two servers and adding it to the total time that the customer i waits in the queue.

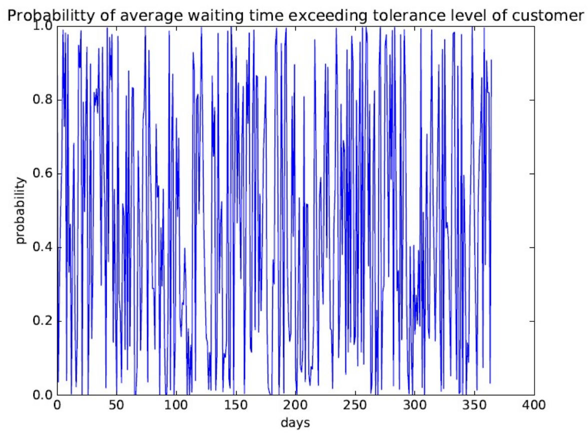
The service time is generated from an exponential distribution depending on the type of the customer and similar is done for its arrival rate as well. The above algorithm is repeated for 5000 customers a day and we have calculated the average waiting time in the queue for any customer in a day. These averages are calculated over a year to give us a rough estimate of the average waiting time in this model.

For a Sample Data, we made the following observations:

λ_1= 15/min, μ_1=20/min and λ_2= 5/min, μ_2=10/min then avg waiting time in queue= 1.371 minutes

For tolerance level, α=0.2 minutes, P(avg W_Q> α)=0.462





Approaching Model2:

The Basic idea behind this model is that the arrival in the system happens at rate λ_1+ λ_2 and hence randomized inter-arrival times are calculated accordingly. These arrival time in the system represents the time at which the customer i enters the system and decides on which queue to join depending on the length of the two queues.

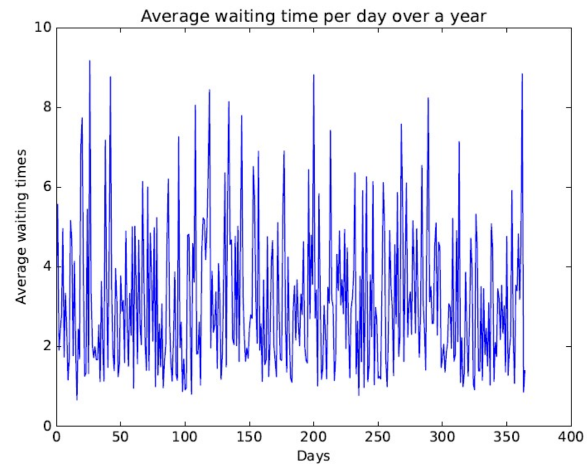
The Length of the two queues is calculated using the length of arrays S1 and S2 where S1(WLOG) contains the exit times of the people waiting in the queue **before i th arrival**. S1 keeps getting updated with every departure from queue 1 (draw analogy for S2).

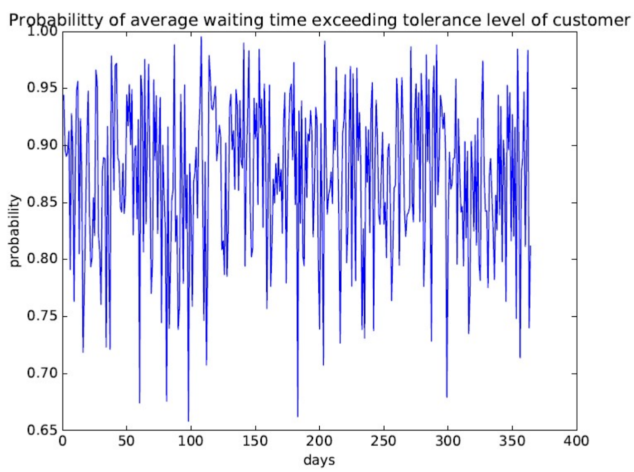
Once this customer i joins a particular queue then the time T_i i.e. the arrival time in the system would acts as its arrival time in the queue1 i.e. T_{1i} and thereafter we can continue with calculation of waiting times in each queue independently of each other and calculate average waiting time for a customer in a day. Average waiting time is then calculated over the average daily waiting times over a year.

For a Sample Data, we made the following observations:

λ_1= 15/min, μ_1=20/min and λ_2= 5/min, μ_2=10/min then avg waiting time in queue= 3.147 minutes

For tolerance level, α_1=0.2 minutes for type I customers and α_2=0.4 for type II customers, P(avg W_Q> α_{avg})=0.872





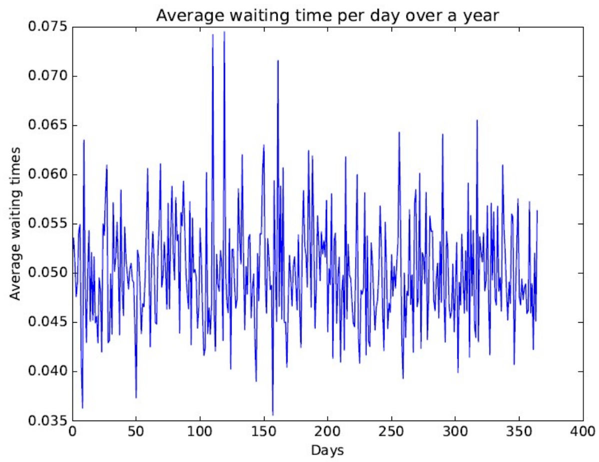
<u>Approaching Model3:</u>

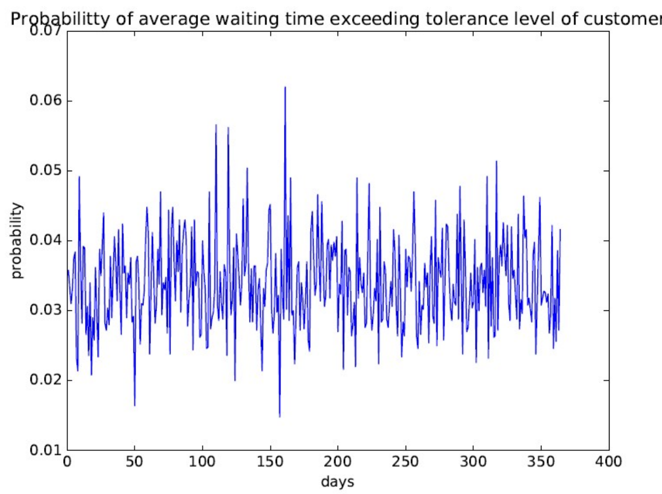
Since in this case we have specialized servers for each type of customers hence both the servers act as two different systems which leaves us with a much simpler model. At the end of each day in this system, we calculate the average waiting time of a customer in the system by summing up all the waiting times of queue1 and queue2 and calculating an average of it. Similar average is calculated over a year.

For a Sample Data, we made the following observations:

λ_1= 15/min, μ_1=20/min and λ_2= 5/min, μ_2=10/min then avg waiting time in queue= 0.05 minutes

For tolerance level, α_1=0.2 minutes for type I customers and α_2=0.4 for type II customers , P(avg W_Q> α_{avg})=0.034





<u>Conclusion</u>:

For the chosen parameters, we conclude that keeping segregated counters would serve the purpose in best possible way. Though if we only compare model1 and model2 since they have similarity in the arrival rate and just the queue arrangement varies, *a single queue for multiple identical servers **works better** than multiple queues identical server system.*