Performance Analysis of Single-Server and Double-Server Queueing Systems with Normally Distributed Random Inputs

A partial requirement for the course

CSC133 (Modelling and Simulation)

2nd semester, A.Y. 2023-2024

Submitted to:

Prof. Llewelyn A. Elcana

Faculty of Department of Computing Sciences

MSU - College of Computing and Information Sciences

Marawi City

Submitted by:

Sittie Aisha C. Abdulmanan

June 2024

MAIN PROGRAM

Program Overview:

- Written in Python for readability.
- Requires libraries (pandas, prettytable, matplotlib) for exporting data.
- Organized into multiple folders and files for better structure (readable & modifiable):
    - 3 class files (Queue, Customer, Single_Server/Double_Server) in folder Classes
    - 5 function files (Normal_distribution, Display, Export, Snapshot, Plot) in folder Functions

Inputs:

Both the two main programs accept data in two ways:

1. Pre-defined data: Provide arrays for inter-arrival times, service times, and arrival times.
2. Random generation: Specify means and standard deviations for inter-arrival and service times. The system will generate them based on a normal distribution and calculate arrival times.

Outputs:

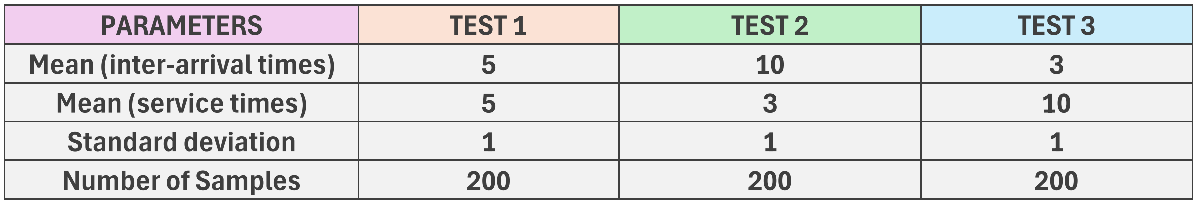
- Generates detailed information about the simulation.
- Saves output files to a separate folder named "Outputs".
    - Subfolders are created for each test name and output type.
    - There are 3 subfolders for each test folder:
        - Customer_Data - provides details about the customer
        - Simulation_Data – provides details about the overall simulation process
        - Snapshots – provides details every clock time
    - Creating new test folders requires including subfolders to prevent errors.
    - Existing files are overwritten during program execution.
    - The “Outputs” folder can be emptied of files, but not deleted. Deleting it will cause errors during program execution.

Plotting:

- Plotting functionality is disabled by default.
- Shows 3 graphs in one window:
    - Number of People in Queue over time
    - Server Status over time
    - Delay over time
- Multiple test runs will display graphs one at a time. Close the current graph to view the next.

METHODOLOGY

Three tests with varying parameters were conducted to determine the performance of single-server and double-server queueing systems. These tests used identical parameter sets for both server structure. Each test was run three times to identify any recurring patterns. The specific parameters for each test are listed below:



PARAMETERS	TEST 1	TEST 2	TEST 3
Mean (inter-arrival times)	5	10	3
Mean (service times)	5	3	10
Standard deviation	1	1	1
Number of Samples	200	200	200

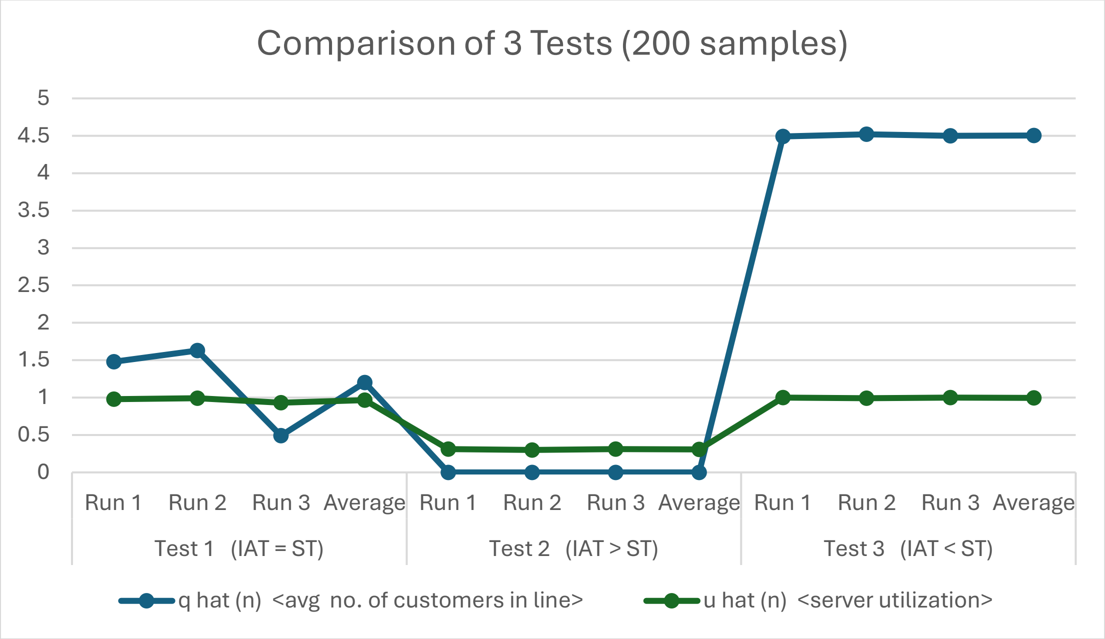
Three main measures are used to evaluate the performance of the both the single-server and double-server queueing systems namely: d_hat, q_hat and u_hat.

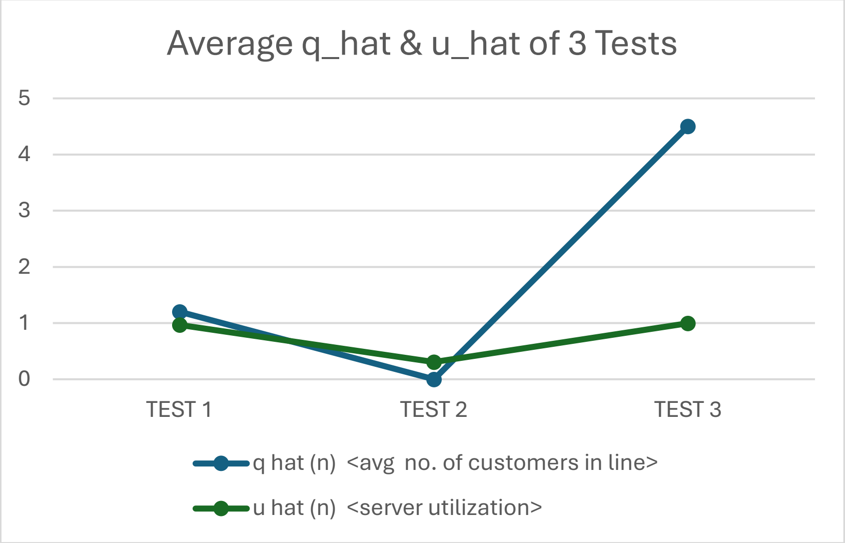
- The average delay time (d_hat) is the average amount of time that a customer spends in the queue waiting to be served.
- The average number of customers in the queue (q_hat) is a measure of how long customers wait on average before they are served.
- The server utilization (u_hat) is a measure of how busy the server is. A value of 1 means that the server is busy all the time.

RESULTS / DISCUSSIONS

❖ Single Server Queueing System

The table on the next page shows the value of both the q hat (average number of customers in the queue) and the u hat (expected utilization of the server) for each run of each test. Based on the graph results, there is an inconsistent pattern in Test 1, unlike in Tests 2 and 3. This can be contributed to the random generation of the inputs for Inter-arrival and Service times with a normal distribution. A standard deviation of 1 for both arrival and service times in Test 1 means there's a higher chance of values deviating significantly from the mean (5) compared to Tests 2 and 3 (standard deviation of 1 with a mean of 3 or 10). This can lead to more fluctuations in the queue length and waiting time during the simulation. The randomness inherent in using a normal distribution can definitely cause the initial inconsistency observed in Test 1. The specific combination of means and standard deviations used in Tests 2 and 3 likely led to more predictable arrival and service time patterns, resulting in a more consistent queue length and waiting time.





The graph above shows the average number of customers in the queue (q_hat) and the server utilization (u_hat) for three different tests. Based on the graph, the average number of customers in the queue (q_hat) and the server utilization (u_hat) are higher in Test 3 compared to Tests 1 and 2. In essence, the graph suggests that the single-server queuing system experiences the most workload in Test 3, which leads to longer queues and higher server utilization. Test 1 also shows some congestion compared to Test 2.

VALUES	TEST 1	TEST 2	TEST 3
d hat (n) <avg delay time>	6.08	0	45.23
q hat (n) <avg no. of customers in line>	1.2	0	4.503
u hat (n) <server utilization>	0.967	0.307	0.997
Total served customers	200	200	64.667
Total discarded customers	0	0	135.333
Total simulation time (Tn)	1020.9	2000.133	649.633

Based on the result of the simulation, Test 2 has the least busy server and the shortest queues. This can be seen from the fact that Test 2 has the lowest server utilization (u_hat) of 0.307 and the lowest average number of customers in the queue (q_hat) of 0. This indicates that the server was idle most of the time in Test 2, and the queue lengths were very short.

In contrast, Test 3 has the busiest server and the longest queues. This is evident from the fact that Test 3 has the highest server utilization (u_hat) of 0.997 and the highest average number of customers in the queue (q_hat) of 4.503. This suggests that the server was busy most of the time in Test 3, and the queues were significantly longer than in the other tests.

Test 1 has a server utilization and queue lengths in between those of Tests 2 and 3. The server utilization in Test 1 is 0.967, and the average number of customers in the queue is 1.2. This indicates that the server was very busy in Test 1, and the queues were moderate in length.

Among the three tests, Test 2 has the best overall performance. This is because the server was idle most of the time, which means that customers did not have to wait long for service. In queuing systems, the goal is to minimize queueing time and maximize server utilization, but not to the point where the server is overloaded. Test 2 achieves a good balance between these two goals.

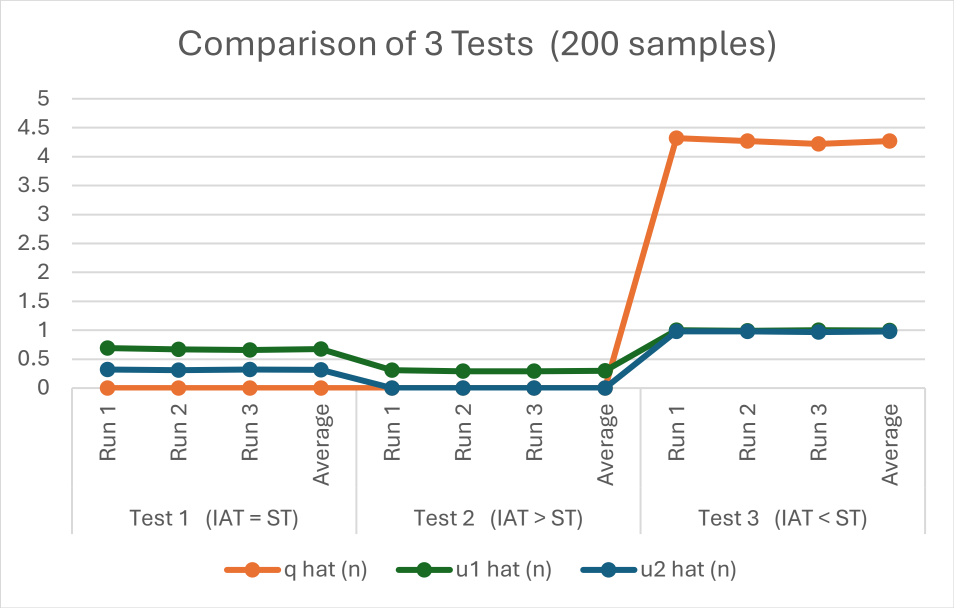
Compared to Tests 1 and 3, Test 2 has the lowest average number of customers in the queue (q_hat) and the lowest average delay time (d_hat). This indicates that customers experienced shorter waiting times in Test 2. Moreover, the server utilization (u_hat) in Test 2 is 0.307, which is a good balance between keeping the server busy and avoiding overload. In Test 1 (u_hat = 0.967), the server was nearing overload, and in Test 3 (u_hat = 0.997), the server was heavily overloaded.

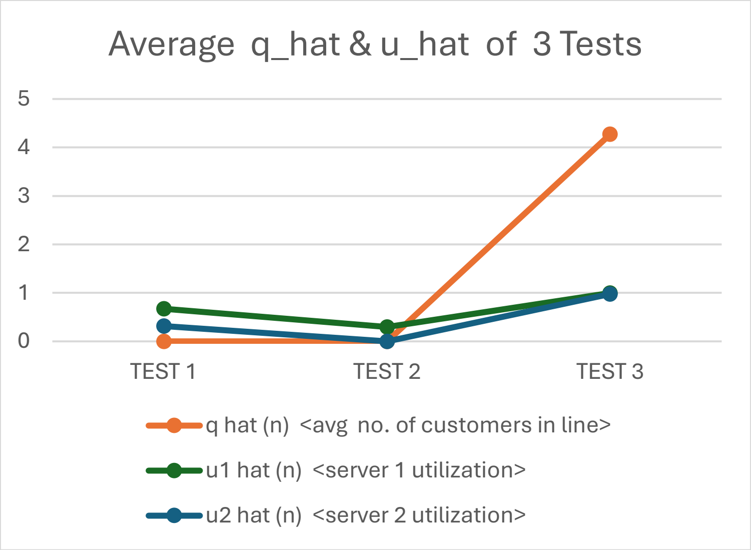
The average total number of customers served is the same in Tests 1 and 2 (200), but it is much lower in Test 3 (64.667). This is likely because the server was so busy in Test 3 that it could not serve as many customers. The average total number of discarded customers is 0 in Tests 1 and 2, but it is quite high in Test 3 (135.333). This suggests that some customers arrived in Test 3 cannot be accommodated due to the limit of queue, so they have to leave the system without being served.

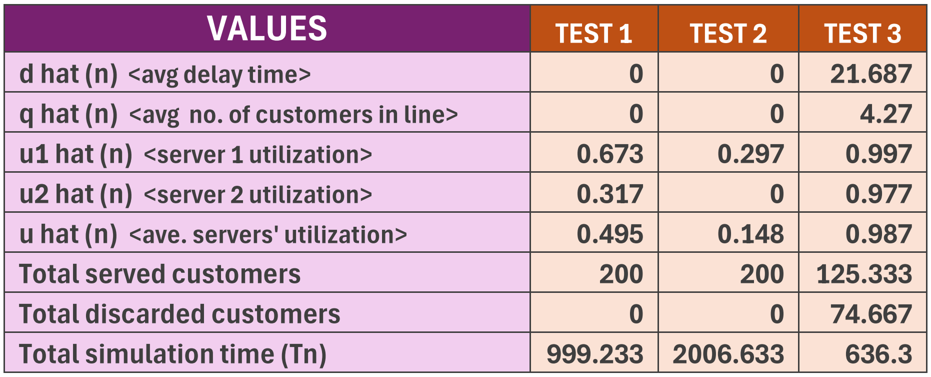
Overall, the simulation results show that the performance of the queuing system can vary depending on the arrival and service times of the customers. Test 2 represents a scenario where the server is well-balanced with the workload, while Tests 1 and 3 show scenarios where the server is either underutilized or overloaded.

- **Double Server Queueing System**

The table below shows the value of both the q hat (average number of customers in the queue) and the u hat (expected utilization of the server) for 3 runs of 3 tests. In both Tests 1 and 2, there is a significant gap between the servers' utilization in each run. Unlike in Test 3, the utilization of both two servers is almost the same. For the q hat, there is zero value for both Tests 1 and 2 in each run, while Test 3 has value ranging between 4 and 4.5 in each run.







VALUES	TEST 1	TEST 2	TEST 3
d hat (n) <avg delay time>	0	0	21.687
q hat (n) <avg no. of customers in line>	0	0	4.27
u1 hat (n) <server 1 utilization>	0.673	0.297	0.997
u2 hat (n) <server 2 utilization>	0.317	0	0.977
u hat (n) <ave. servers' utilization>	0.495	0.148	0.987
Total served customers	200	200	125.333
Total discarded customers	0	0	74.667
Total simulation time (Tn)	999.233	2006.633	636.3

As shown in the result, there is a significant difference between the utilization of the two servers for each test. In Test 1, server 1 utilization (u1_hat) is 0.673 and server 2 utilization (u2_hat) is 0.317. This indicates that both server 1 and server 2 were moderately busy. In Test 2, server 1 utilization (u1_hat) is 0.297 and server 2 utilization (u2_hat) is 0. This shows that Server 1 was moderately busy, while Server 2 was completely idle in Test 2. In Test 3, server 1 utilization (u1_hat) is 0.997 and Server 2 utilization (u2_hat) is 0.977. This indicates that both servers were very busy in Test 3, close to being overloaded.

Both Tests 1 and 2 have a q hat value of 0, indicating that there were no customers waiting in the queue on average. In contrast, Test 3 has an average total of 4.27 which means that there are at least 4 customers waiting in the queue on average. Moreover, since q hat is zero in Tests 1 and 2, it follows that there is also zero delay time in Tests 1 and 2. In Test 3, the average delay or time that a customer spends in the queue waiting to be served is 21.687.

In Tests 1 and 2, the average total number of customers served is 200; however, in Test 3, the average is substantially lower at 125.333. This is probably because Test 3's server was so busy that it was unable to service as many customers as possible. Furthermore, in Tests 1 and 2, the average total number of discarded clients is zero; however, in Test 3, it is rather large (74.667). This implies that some of the consumers who arrived in Test 3 were unable to be served because of the wait line's restricted capacity and were forced to exit the system.

Based on the results, all three tests are significantly different from one another. Server utilization patterns differ across the tests. In Test 1, both servers were moderately busy, indicating a good match between server capacity and workload. There's some idle time on server 2, which suggests the system could potentially handle a slightly higher workload without becoming overloaded. In Test 3, both servers were close to being overloaded, which could lead to longer waiting times and queues if the workload were to increase slightly. In Test 2, one server was completely idle while the other server was moderately busy. It's not always good to have one server in a double server queuing system fully idle (u_hat = 0); if the workload is constantly low, this could be viewed as a waste of resources. Nonetheless, given one idle server suggests effective workload allocation, no queues, and possible cost savings. It might be a positive sign of a well-balanced system.

With one server dedicated to handling the incoming requests, there's no need for customers to wait in a queue. In Test 2, one busy server has sufficient capacity to manage the burden as it is, and another server has some spare capacity to manage any future short increases in workload without significantly degrading performance.

Both Tests 1 and 2 avoids queues on average (q_hat = 0), indicating good customer service with minimal waiting time. However, it is only in Test 1 that both servers have some utilization (u1_hat = 0.673, u2_hat = 0.317). This suggests efficient use of resources compared to having one server completely idle in Test 2. However, it also means there's slightly less spare capacity to handle unexpected workload surges.

Test 3 highlights the limitations of the double server system under heavy workload. While queues and waiting times might be lower compared to the single server in Test 3, there are still customers waiting and some even discarded due to the long wait. Server utilization in Test 3 is very high, indicating the system is close to being overloaded.

COMPARISON

The double server system is expected to outperform the single server system in terms of reduced queue lengths**,** lower waiting times, and improved server efficiency. With two servers, queues are likely to be shorter or even nonexistent compared to the single server. Due to shorter queues, waiting times should also be significantly lower or even eliminated in the double server system. Moreover, server utilization might be lower in the double server system compared to the single server system, indicating better use of resources and potentially more room to handle short bursts of increased workload.

Overall, the double server system offers improved performance compared to the single server system, but it's not a perfect solution, and there are still trade-offs to consider. The double server system generally performs better than the single server system. However, having two servers compared to one obviously incurs a higher cost. Simulation analysis can help determine if the performance improvement justifies the additional cost.