Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Computer Engineering MSc 1st year

Computer Simulation and Modeling Project

Submitted by:

1. Meseret Dereje GSR/4274/13
2. Sebah Tewodros GSR/8418/13
3. Tinbilina Gashaw GSR/5005/13

Submitted to : Sosina Mengistu Gashaw, PhD

Assistant Professor,

Date :3/18/2021

Introduction

The project equations are a variation on a simple multi-server queuing model. The goal of the project was to design these 3 models as a simulation and gather the output data like response time and queueing length time. And finally reflect up on the results. Also in the project we were expected to design a custom exponential random variable generator. All the code for the project was done using python programing language. And the library's used are math and numpy for higher mathematical functions and matplotlib.pyplot to display our final result in the form of a graph

Random variable Generator

For this assignment we used a previously developed random generator that was tested using chi-square test which it has passed. This random generator is used to generate the inter-arrival and service time with a mean value of 3jobs/min and 2jobs/min respectively.

Model 1:

The first model consists of a load balancer that dispatches a job to the shortest queue. Each server has its own queue. It is assumed that the jobs are dispatched immediately, which means that no job waits in the load balancer.

The implementation is as follows: the program initializes values of variables associated with clock (that stores the current event time), queue length, server state, arrival, departure and response time to zero, setting the initial conditions of state variables and the two events, arrival and departure. The first arrival event is scheduled in the initialization method by setting the time to the first inter-arrival time that is generated by the random variation. Then the time is compared to select the smallest to execute the event method associated with the event type. Then, the future event list is created according to the time and the event type. Each queue belonging to the server is compared by the load balancer based on the queue length when a job arrives. A job is assigned to the server with the shortest queue length. In the arrival event, the number of customers arrived is increased and the server OS state is updated. A departure for the job is scheduled then, the next arrival is scheduled. The departure event is connected with the number of servers, therefore, there are three departure events, from the first, second, and third servers. The number of jobs departed is increased accordingly. The response time is calculated by subtracting the departure time from the current simulation time. The queue length is calculated by multiplying the number of jobs in the queues with the amount of time spent in the respective queues.

Model 2:

In this model, there is only a central queue. Whenever the servers become idle, they pull out a job from the queue and process.
The implementation differs only slightly from model one as there is only one queue and jobs are assigned based on which server is idle. As for the departure event, the implementation is the same since there are three servers.

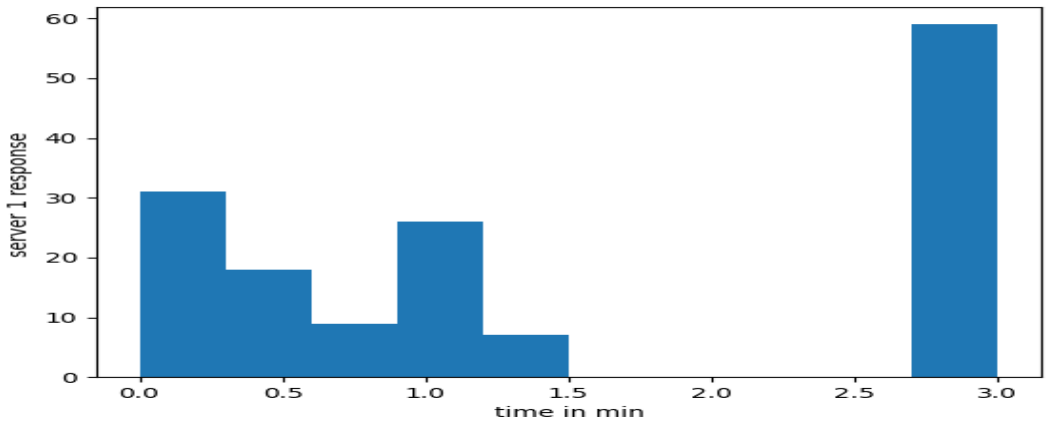
Model 3:

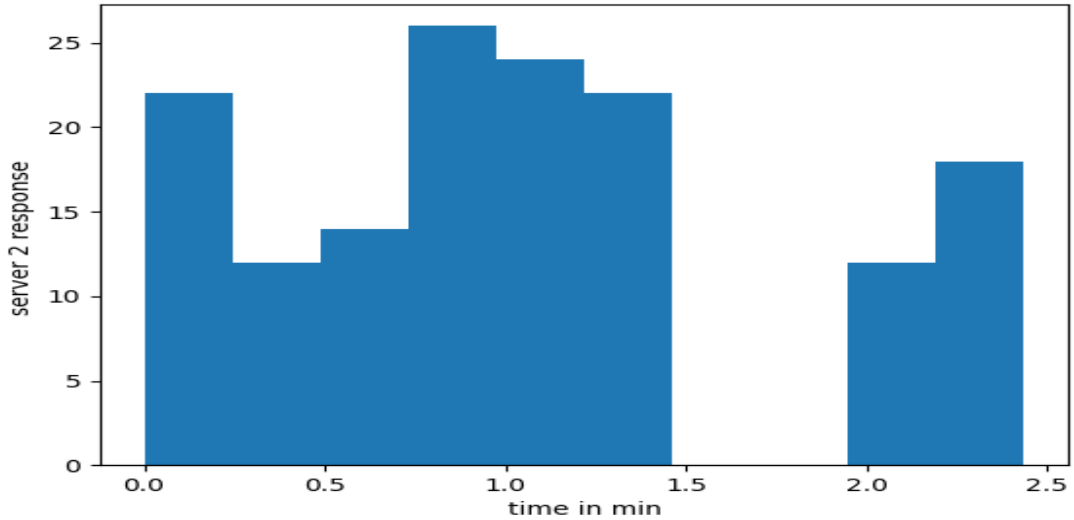
This model is a modification of model2. A power saving scheme is introduced. When a server is idle, it goes to sleep mode and wakes up when the number of jobs is greater than three.

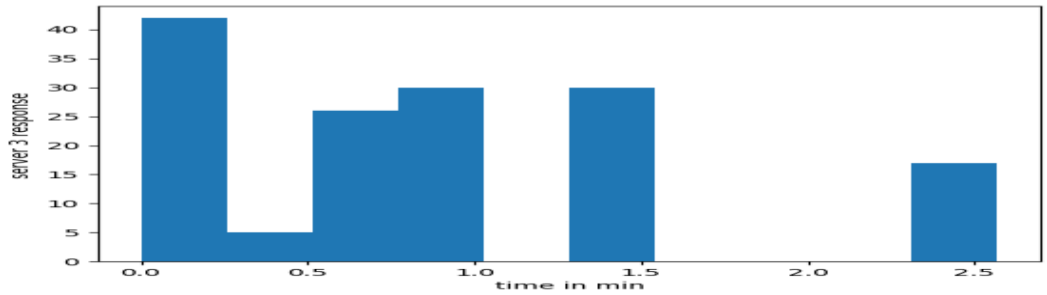
Results

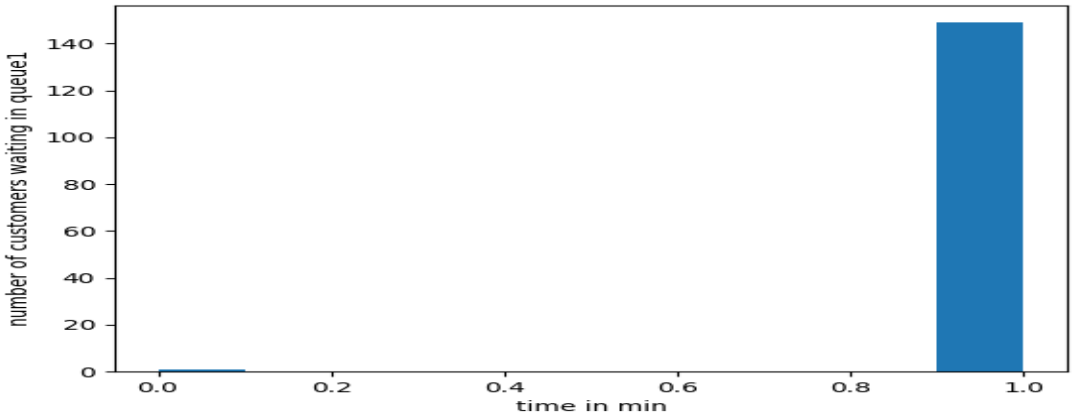
Model	Queue length time			Server one response time	Server two response time	Server 3 response time
	Q1	Q2	Q3			
1	1.1642	0.932	1.11	0.54	0.74	0.57
2	0.705			0.489	0.619	0.762
3	1.38			0.431	0.321	0.241

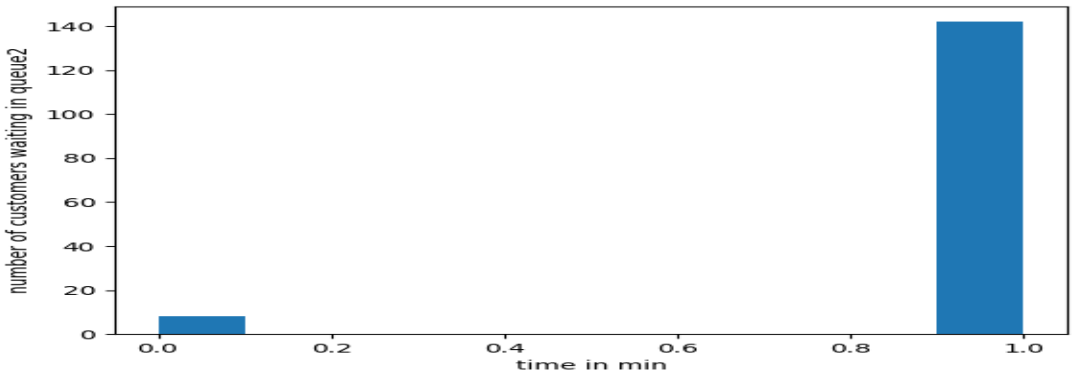
Model 1 graphs

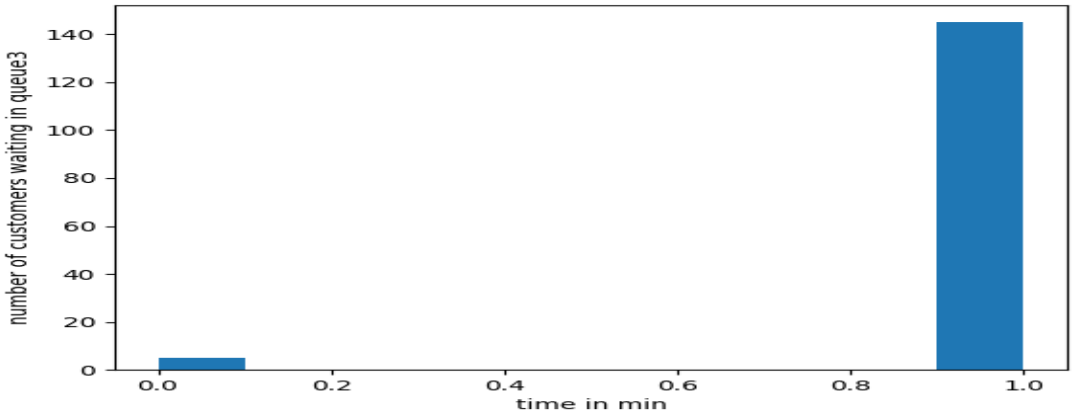




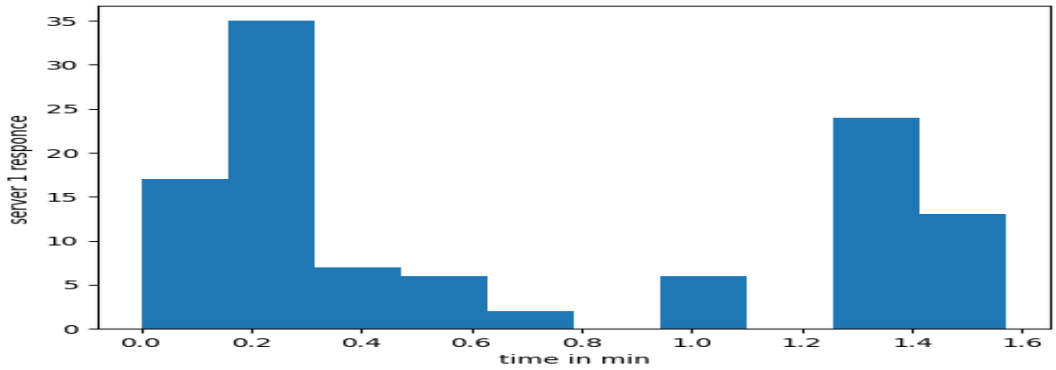

server 3 response
0
5
10
15
20
25
30
35
40
0.0
0.5
1.0
1.5
2.0
2.5
time in min

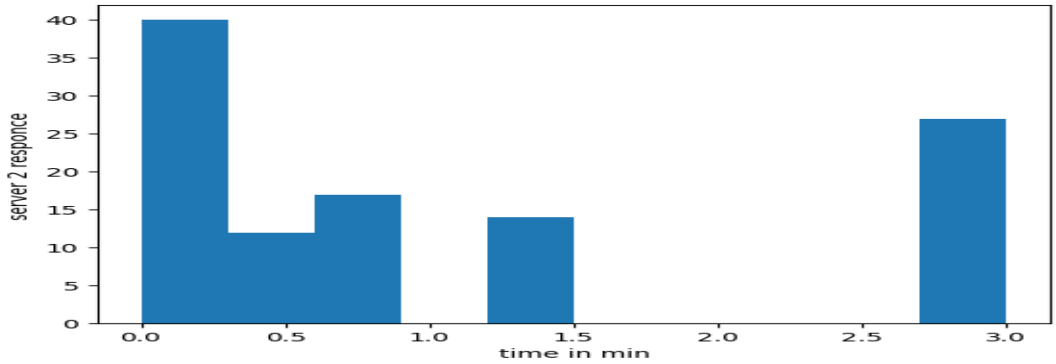

number of customers waiting in queue1
0
20
40
60
80
100
120
140
0.0
0.2
0.4
0.6
0.8
1.0
time in min

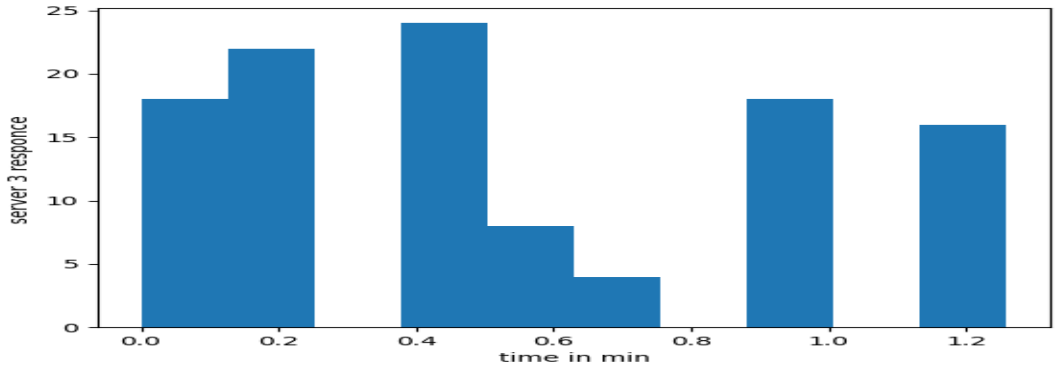

number of customers waiting in queue2
0
20
40
60
80
100
120
140
0.0
0.2
0.4
0.6
0.8
1.0
time in min

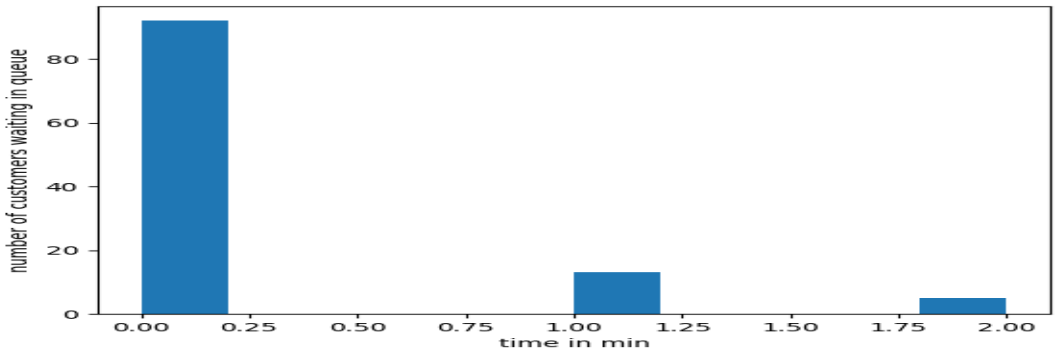

number of customers waiting in queue3
0
20
40
60
80
100
120
140
0.0
0.2
0.4
0.6
0.8
1.0
time in min

Graphs of model 2

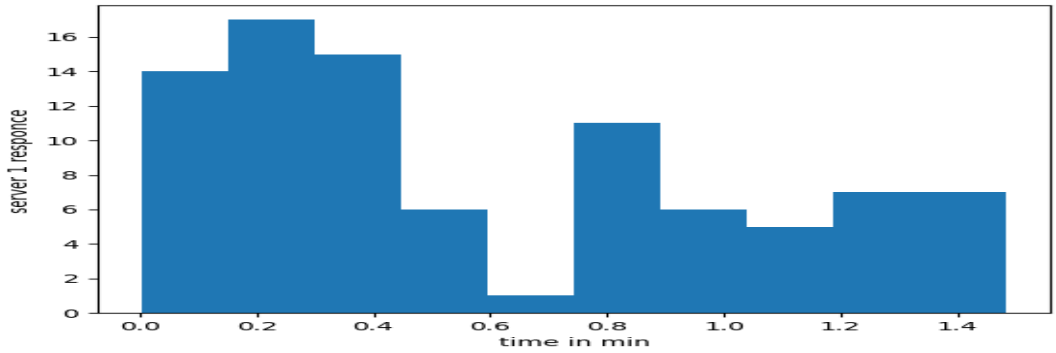


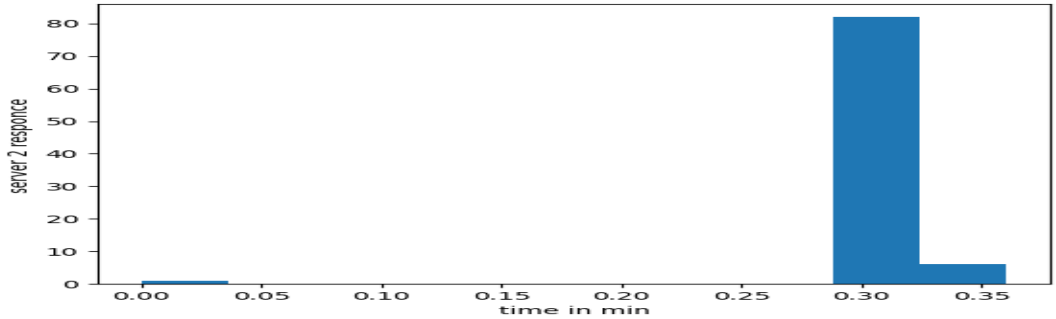


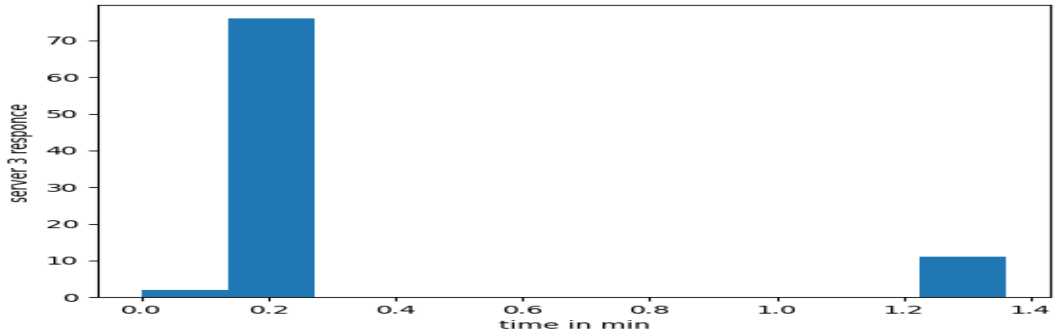


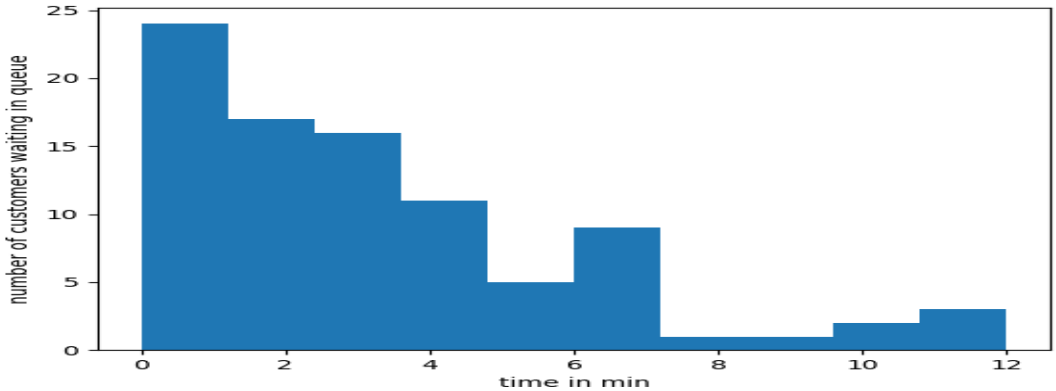


Graphs of model 3









Discussion on the result

From the above data we can easily see that model 1 and 3 have the best response time while having the longest queue line wait time. This is due to the fact in the case of model 1 the server is never idle and has its own queue that it doesn't share so one queue is for one server lowering the rate by which customers go into the server and giving a longer wait time and in the case of model 3 the longer queue line is formed when the server goes to sleep to save power during this time customers arrive and fill the queue before the server works at full capacity.

In the case of server two it has a much lower queue rate because customers are removed from the queue quickly by the server no server is idle so there is no down time.

The graphs also reflect this in model 1 the 3 queues show almost similar pattern while the queue graph of model two is a lower rate and on model 3 the queue graph spikes during the time where one or more server is down.

There for we recommend model 1 and 3 for services that require faster response time and model two for shorter queue length.