M/M/1 Queuing System Simulation using C

Ajay Chandra P[1] Kushagra Sinha[2] Praneeth Varma R[3] Venkata Mukesh M[4]
[1,2,3,4]Department of Computer Science
[1,2,3,4]School of Computing Sciences and Engineering Vellore Institute of Technology Vellore, India

Abstract— The queuing system is simulated where the inter-arrival times and the service times are assumed to be exponentially distributed. This simulation is going to be run on two criteria: based on the total runs and based on the total time duration of the simulation. This simulation is run based on an algorithm which is programmed in the C language. In the simulation, the average values derived such as the waiting time, idle time can be compared with the steady state parameters as well. Their difference or deviation from the steady state parameters will give us an insight into the accuracy of the algorithm being employed.

Key words: Simulation, Queuing System, Exponential Distribution

I. INTRODUCTION

A queuing system has the following characteristics: its calling population, the nature of the arrivals, the service mechanism, the system capacity, and the queuing discipline. The various examples for queuing systems are banks, airports, restaurants, post offices, traffic signals etc.

Simulation of a model by definition, is the imitation of operation of a real world model or system over some time. And these models are usually simulated or their steady state parameters are estimated using statistical methods which are also augmented with the real data from the practical examples[2]. This method has often been reliable. One of the statistical methods used is alluded to by Jerry banks, John S. Carson, Barry L Nelson and David M Nicol in the book Discrete Event System Simulation[1]. This very statistical method has been incorporated into our algorithm.

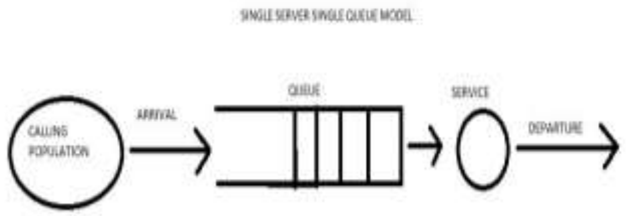

Fig. 1: Basic Queuing Model

II. M/M/1 QUEUE

The queuing system chosen is M/M/1.The M/M/1 queue follows the queuing notation A/B/c/N/K where the individual components are described as follows:[1]

- Inter-arrival time distribution
- Service time distribution
- c- Number of parallel servers
- N- System capacity
- K- Calling population size

Therefore, an M/M/1 can be thought of as a queuing system where the service time and inter-arrival time distributions are exponential, there is only one server and the system capacity and calling population are infinite.

An M/M/1 queue can be imagined to be a M/G/1 queue where the service times are exponentially distributed. Essentially, the mean is $1/\mu$, and the variance is $\sigma 2 = 1/\mu 2$.[1]

M/M/1 queue is an important paradigm where the service times' standard deviation is approximately same as its mean.

The steady state parameters of the M/M/1queue are given as following:[1]

$$\rho = \lambda/\mu, \quad P_n = (1-\rho)\rho^n$$

$$L = \frac{\lambda}{\mu-\lambda} = \frac{\rho}{1-\rho}, \quad L_Q = \frac{\lambda^2}{\mu(\mu-\lambda)} = \frac{\rho^2}{1-\rho}$$

$$w = \frac{1}{\mu-\lambda} = \frac{1}{\mu(1-\rho)}, \quad w_Q = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{\rho}{\mu(1-\rho)}$$

Where the notations are as follows[1]:
ρ- Server utilization,
Pn- Steady state probability of having n customers in the system,
L- Long run time-average number of customers in the system,
Lq- Long run time-average number of customers in the queue,
w- long-run average time spent in system per customer,
wq- long-run average time spent in queue per customer,
λ- arrival rate,
μ- service rate

III. RANDOM VARIATE GENERATION

Before we delve into the actual algorithm, it is imperative that an essential component of the algorithm is mentioned first. As sample data is to be fed into the simulation model, it should be random and should belong to the exponential distribution.

And the random variates are generated using the Inverse transform technique here, which is specified for exponential distribution as follows[1]:

$$1-e^{-\lambda X} = R$$
$$e^{-\lambda X} = 1-R$$
$$-\lambda X = \ln(1-R)$$
$$X = -\frac{1}{\lambda}\ln(1-R)$$

Where R is the random number input and X is the random variate for the distribution.

IV. ALGORITHM

1) Retrieve the mean values of both service times and the inter-arrival times.
   - If(mean(arrival)>mean(service))
   - Then "error"
2) The user choice for either simulation based on number of runs or trials or based on total simulation duration is provided.
3) Case1:
   - The total no. of trials and its total no. of runs are retrieved from the user.

– Random variates for arrival and service times are generated and the simulation is done.[3]

```
for(j=0;j<runs;j++) {

rand_iat1[i]=(float)rand() / (float)RAND_MAX ;
iat1[i]=(-l1)*log(rand_iat1[i]);
rand_st1[i]=(float)rand() / (float)RAND_MAX ;
st1[i]=(-l2)*log(rand_st1[i]);

iat= iat1[i];
st=st1[i];
```

– The output is displayed in the form of a table.

– The various factors such as average wait time, average idle time, probability (wait), probability (idle) are displayed.

– The user gets to choose if he wants to compare them to the steady state parameters.

4) Case 2:

– The total duration of the simulation is entered.

– The process similar to above is repeated for the stipulated time period.

– The output is displayed.

5) End of simulation.

V. SAMPLE INPUT AND OUTPUT

For a random input of mean (inter-arrival time)= 0.4 and mean(service time)= 0.6

The case 1 where the no. of trials being 200 and each trial containing 50 runs, the output is

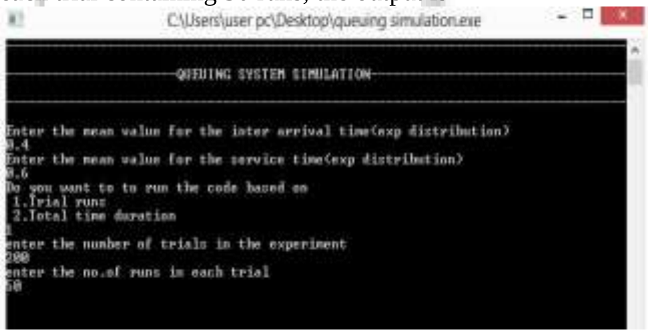

Fig. 2: Output

The output columns are as follows:

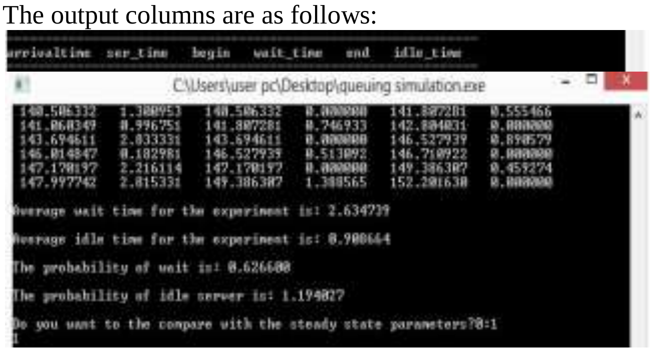

Fig. 3: Output columns

A. *Steady State Parameters*

Fig. 4: Steady State Parameters

For case 2, the duration given is 200 units, with the mean (arrival)=0.6 and mean(service)=0.8

The output is

Fig. 5: Output

VI. INFERENCE AND CONCLUSION

As it can be seen, the accuracy of the simulation (waiting times) with respect to the steady state parameters is about 80% which is decent. So, in conclusion, it can be said that the proposed algorithm for the simulation of M/M/1 queuing system is reliable in understanding the concepts of queuing system simulation and is consistent with the statistical methods already being employed to estimate the parameters.

VII. SUPPORT INFORMATION

The link added contains the source code for the simulation for reference:
https://drive.google.com/file/d/0B6J8TbJROUScZ2RXaGJyOHNpODg/view?usp=sharing

REFERENCES

[1] Jerry banks, John S. Carson, Barry L Nelson and David M Nicol -Discrete Event System Simulation, 4th Edition
[2] Irene K. Amponsah, Bennony K. Gordor, Francis Dogbey, Simulating a Single-Server Queue using the Q –Simulator, World Academy of Science, Engineering and Technology Vol:6 2012-06-29
[3] Simulation and Modeling Analysis, Law & Kelton (1991)

