Discrete MM, practical task 4

Student: Grigoryev Mikhail

Group: J4133c

Task description

- Please modify the example (DEVSEXAMPLE.py) so that the accuracy of one of the parameters (for example, server utilization or customers queue length) is set to specific value.
- After that you should make several test runs and estimate sufficient number of replications to achieve the specified accuracy.
- If the accuracy is not achieved, repeat the procedure again. Computational technology is on your own decision.

Solution method

A multi-server queue model (Poisson process) was implemented in Python3 by modifying the given file in form of a simple iterative process of assigning clients to cashiers.

Metrics were calculated like shown below:

```
# 1) Average waiting time in queue
avTimeInQueue = sum([x.waitingTimeInQueue for x in S.stats]) / len(S.stats)

# 2) Probability that a customer has to wait
probToWait = len([x for x in S.stats if x.waitingTimeInQueue > 0]) / len(S.stats)

# 3) Probability of an Idle server
probIdle = sum([x.idleTimeOfServer for x in S.stats]) / S.GlobalTime

# 4) Average service time (theoretical 3.2)
avServiceTime = sum([x.serviceTime for x in S.stats]) / len(S.stats)

# 5) Average time between arrivals (theoretical 4.5)
avTimeBetwArr = sum([x.interArrivalTime for x in S.stats]) / (len(S.stats) - 1)

# 6) Number of waiters
numOfCustWhoWait = len([x for x in S.stats if x.waitingTimeInQueue > 0])

# 7) Average waiting time for those who wait
try:
    avTimeWhoWait = sum([x.waitingTimeInQueue for x in S.stats]) / numOfCustWhoWait
except:
    avTimeWhoWait = 0

```

```
# 8) Average time spent in the system
avTimeInTheSystem2 = avTimeInQueue + avServiceTime
```

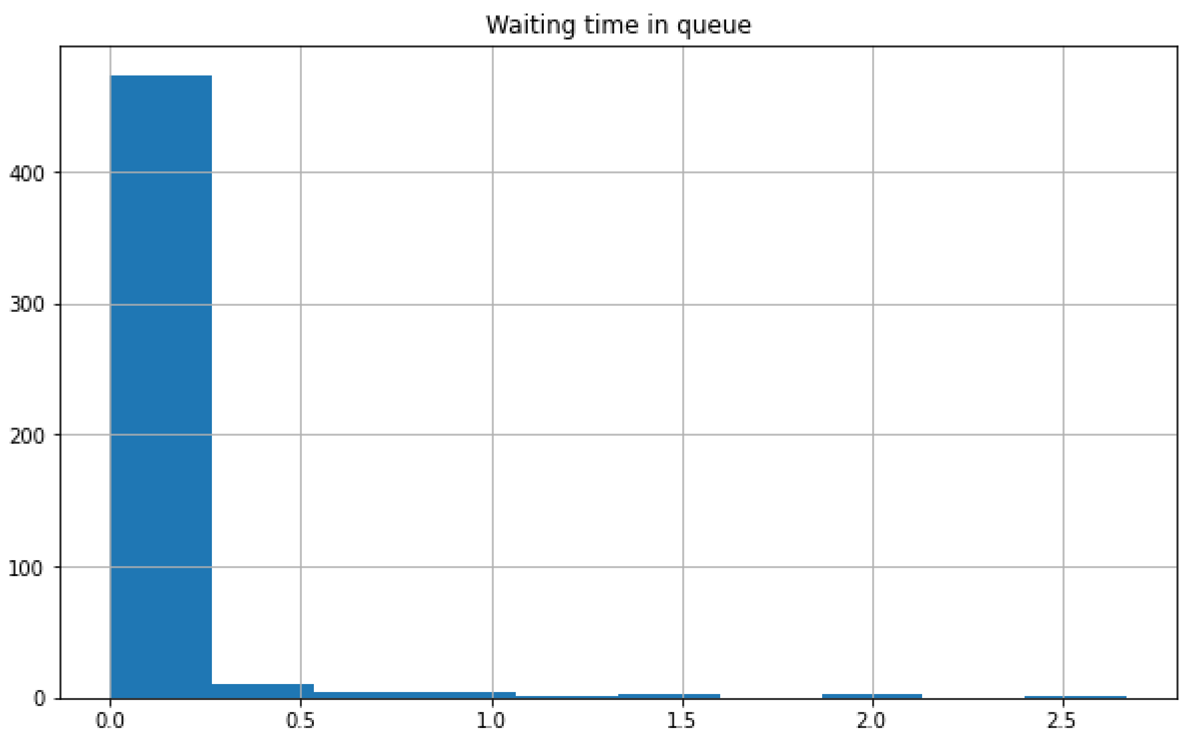
For calculating the confidence intervals, the following equations were used:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

$$\text{ConfInterval} = \bar{Y} \pm t_{\alpha/2,n-1}\frac{S}{\sqrt{n}}$$

Results

An example histogram for the metric "Waiting time in queue":



The output for the model achieving the accuracy of 1% on the metric "Average time in system":

```
m: 3.295 h: 0.048 r: 42 num of exps: 5
m: 3.316 h: 0.019 r: 55 num of exps: 44
m: 3.317 h: 0.017 r: 62 num of exps: 57
m: 3.312 h: 0.017 r: 64 num of exps: 64
m: 3.314 h: 0.016 r: 65 num of exps: 66
eps: 0.016
ans: 3.314+-0.016
R from: 5 to 66
```

Here the model attempts 5 experiments, finds the H-value too large, then adds more experiments so that the total number of them equals $R + \text{additional}$, where $\text{additional} = 2$ (bias for no attractors). As one can see, the calculated confidence interval (at 95%) is 3.314 ± 0.016 when $\varepsilon = 0.016$. Thus, 66 runs was enough for achieving the given accuracy iteratively.

Conclusions

In this practical work a placeholder queue simulation model (Poisson process) was implemented to showcase different metrics. This model was then used to calculate the values of those metrics with a given accuracy of ε in case when one doesn't know the total number of calculations needed for the accuracy. Based on the achieved accuracy, confidence intervals for the metrics were built.