Karahan Sarıtaş - 2018400174
Yılmaz Burak - 2018400237
Mustafa Atay - 2020400333

IE 306 Group 11 - HW3 Report

Arena Code

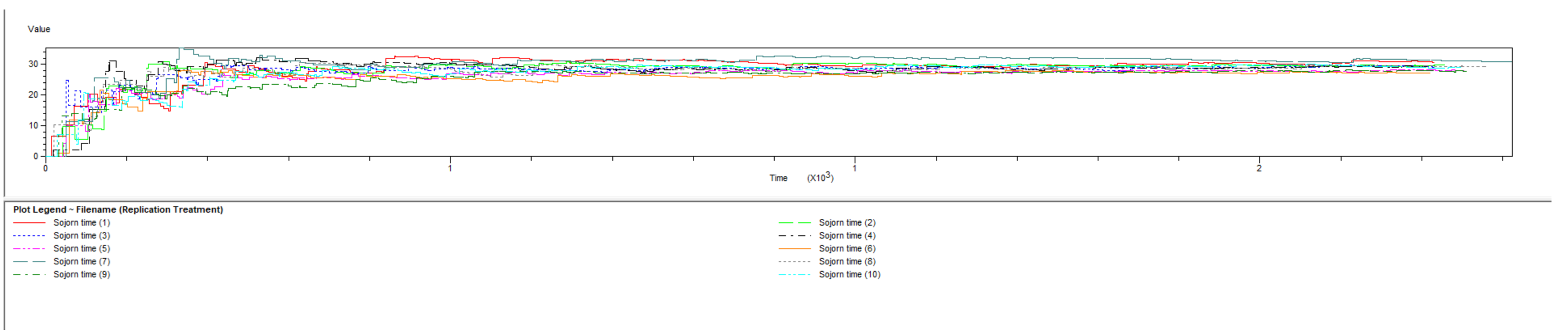
To perform the simulation in the Arena software, we have created an entity named as *Job*. *Job* entities correspond to the customers arriving to the system with uniform distribution. *Separate* block helps us to copy the incoming customers to simulate the reneging behavior. For each customer, a duplicate is created. Duplicates are forced to wait in a process block named as *Renege Time* while original *Job* entities wait in the Queue to get into the station #1 server. We created a variable named *vSearchNumber*. Whenever a *Job* object leaves the *Renege Time* block, the serial number of the *Job* object is assigned to this variable. Then we search for the *Job* object with the same serial number in the queue. There are two possibilities here: either the original *Job* object has already left the queue - meaning that the customer didn't wait longer than his renege time, or he is still in the queue - therefore he'll renege from the system. In the former case, we simply dispose of the *Job* object. In the latter case, we remove the original *Job* entity from the queue and dispose of it. This time the statistics is stored to keep the number of customers who reneged before entering into the station #1 server.

Customers who didn't renege from the queue of station #1 proceeds to the second stage. At the second stage they arrive at a *Decision* block. This block checks if the number of customers in the station #2 is equal to 4 or not. If it's 4, then the current customer immediately leaves the system. Otherwise, customer enters station #2 and then leaves the system. Overall we have 5 dispose units - 3 of which collect statistics while others don't contribute to the output statistics directly.

Notice that due to the *Separate* block, there are twice as many exiting *Job* entities as entering entities to the system. We run our simulations considering this design choice and calculated statistical outcomes accordingly. We also cross-checked our statistics by running the same simulations on Simpy.

Warm-up Period

Graph of ensemble averages of sojourn times can be seen below.



As it can be seen from the graph, about t = 400, ensemble averages stabilized. Therefore, we specified t = 400 hours as the warm-up period (t is in hours).

Simulation Results & Comments

Expected interarrival times - Uniform Distribution:

$$E[X] = \frac{10+4}{2} = 7$$

Expected renege time - Exponential Distribution with $\lambda = 0.04$

$$E[X] = \frac{1}{\lambda} = \frac{1}{0.04} = 25$$

Expected station #1 service time - $Erlang(k, \mu_1)$ with $k = 3$ and $\mu_1 = 0.468$:

$$E[X] = \frac{k}{\mu_1} = \frac{3}{0.468} = 6.41$$

Expected station #2 service time - Exponential Distribution with $\lambda = 0.03$

$$E[X] = \frac{1}{\lambda} = \frac{1}{0.03} = 33.3$$

The Conservation Equation:
L = λW (Little's Law) (λ = arrival rate) (W = average sojourn time)
Conservation equation holds for almost all queueing systems or subsystems regardless of the number of servers, the queue discipline, or any other special circumstances. As you see from the statistical results, when we increase the sample size, the statistical value of L (average number of jobs in the system per unit time) gets close to the calculated value. Calculated value is equal to the multiplication of
$\frac{1}{mean\ interarrival\ time} \cdot mean\ sojourn\ time$

10 Replications without excluding the warm-up - 500 exiting jobs

Average Sojourn:
(29.3000 + 28.4734 + 28.8084 + 28.5993 + 28.4619 + 27.4689 + 26.6532 + 30.1152 + 28.6336 + 27.2064) / 10 = 28.3723

Average renege rates:
1-0.6428 = 0.3572

Proportion of completed jobs:
0.6428

10 Replications excluding the warm-up - 500 exiting jobs

Average Sojourn -> (29.6470 + 30.0465 + 30.3251 + 30.1765 + 28.4029 + 29.0850 + 28.3147 + 30.1483 + 29.6661 + 30.0472) / 10 = 29.5859

Average renege rates:
1-0.65=0.35

Proportion of completed jobs:
0.65

Comparison between With/Without Warm Up - 10 Replications

When we eliminate the warm-up time, we see that average sojourn time gets closer to the theoretical value from 28.3723 to 29.5859.

We can see in the above graph that average sojourn time for the jobs before the warm up time are small; this might lead to a decrease in the total average sojourn time.

In our simulations without eliminating the warm-up period, the average output measures were biased down. That bias was expected since we are simulating a queueing-type model that eventually gets congested.

1 Replication - 5000 jobs without excluding the warm-up period

Average Sojourn time: 29.9495
Proportion of jobs completed: 0.6502
Proportion of reneged jobs: 0.3498

1 Replication - 5000 jobs excluding the warm-up period

Average Sojourn time: 29.9688
Proportion of jobs completed: 0.65
Proportion of reneged jobs: 0.35

Comparison between With/Without Warm Up - 1 Replication - 5000 Jobs:

At this point, we have a high amount of jobs, rendering the warm up period almost ignorable. That would explain the small difference between the sojourn times of the simulations.

Comparison between 5000 Jobs vs 10 replications - 500 excluding warmup

Average Sojourn time: 29.9495 vs 29.5859
Proportion of jobs completed: 0.65 vs 0.65

We can see that both of these simulations yield similar results in terms of their sojourn times and proportion of completed jobs. This also implies that the warmup period is effective and close to our findings.
After the warmup, 500-job simulation starts to behave as if it were in steady-state.

1 Replication - 5000 jobs with Incremented Server Capacity - 5 Servers:

Proportion of jobs completed: 0.75
Proportion of renege jobs: 0.25
Average sojourn time: 33.5065

As it can be seen, increasing the size of the simulation causes the sojourn time to also increase. We see that the proportion of jobs completed also increases. This might explain the rise in the sojourn time. Since the completed amount of jobs increase, instead of reneging, they wait to get their job done.

Below, the comparison between the simulation on ARENA and our modified Simpy module can be seen.

Comparison with Simpy

We modified our code from HW1 to reflect the modifications on the simulation environment in this project. Statistical outputs when there are 5000 exiting jobs can be observed below:

Statistics:
Average interarrival time: 7.019
Average renege time: 25.697
Total station #1 service time: 28834.007
Station #1 average service time: 6.443
Station #2 average service time: 32.921
Average waiting time in the queue of station #1: 2.507
Average sojourn time: 29.795
Average reneging rate: 0.347
Proportion of jobs that completed the service: 0.653
Average number of jobs in the system per unit time: 4.237
Little's Law - Calculated value of average number of jobs: 4.245

As it can be seen, average renege rate is 0.347 and proportions of jobs that completed the service is 0.653 which match with the results we gained from Arena simulation. Average sojourn time also complies with our result in the given confidence interval.

Expected interarrival times: 7.0
Expected renege time: 25.0
Expected station #1 service time: 6.41
Expected station #2 service time: 33.33

As we increase the input size, simulation results get close and closer to the theoretical values.

Statistical outputs of the simulation when there are 5 servers in the station #2 can be found below.

Statistics:
Average interarrival time: 7.0
Average renege time: 25.1
Total station #1 service time: 28832.557
Station #1 average service time: 6.439
Station #2 average service time: 34.476
Average waiting time in the queue of station #1: 2.622
Average sojourn time: 34.026
Average reneging rate: 0.257
Proportion of jobs that completed the service: 0.743
Average number of jobs in the system per unit time: 4.846
Little's Law - Calculated value of average number of jobs: 4.861

Again, values coloured in blue matches with the values we get from our Arena simulation.