Queuing System Simulator

Henrique Pedro Roschel
Politecnico di Torino
Student id: S306718
s306718@studenti.polito.it

I. Problem overview

The proposed activity for this laboratory practice was to implement a multi-server FIFO queuing system with a finite waiting line and simulate it under the following conditions:

- Number of servers: $K = 10$;
- Waiting line size: $N = 100$;
- Arrival process of clients are exponentially distributed with $\lambda \in \{5, 7, 9, 10, 12, 15\}$;
- Service time of servers are exponentially distributed with $\lambda = 1$;

The measures to be analyzed are the **average delay**, i.e. the time between the arrival of a client and the start of its service, and the **dropping probability**, the chance that a client gives up the service because the queue is bigger than N at the moment of its arrival.

II. Proposed approach

For the implementation of the queue using Python, the first step was to define how to represent the elements of the system. First, the **queue** is a list of client's records, in a way the FIFO policy can be performed with the function `queue.pop(0)`. The **FES** (future-event set) is also implemented with a list, in which `put` inserts a new event, and `pop` selects and remove the next event.

Events and **clients** are recorded with a specific `namedtuple` each, that contains the type of the event and its time or the client's time of arrival and its current event. The servers are implemented with a single variable to count the number of idle server.

Finally, for the end of the simulation, the possible stop conditions are based on time (stops when the clock reaches a given instant) or the number of arrivals (stops after the n^{th} arrival). A third possibility would be to end the simulation when the queue reaches its maximum capacity, but this was not considered since it would not be possible to compute the dropping probability.

III. Results

The simulation was run under the conditions presented above, varying not only the arrival rate, but also the stop condition. As shown in the tables, the waiting line size can almost be considered infinite, since the dropping probability is zero for lower arrival rates. As for the number of servers, $K = 10$ can be considered an adequate value for most of simulations with $\lambda \leq 10$, with average delay not greater than 10 minutes, but for $\lambda > 10$, a higher number of servers should be considered.

Arrival rate λ	Average delay (min)	Dropping probability
5	0.0085	0.00%
7	0.0736	0.00%
9	0.9483	0.00%
10	2.3282	0.00%
12	29.9894	0.00%
15	56.9461	12.82%

TABLE I
STOP CONDITION: 6 HOURS (360 MINUTES)

Arrival rate λ	Average delay (min)	Dropping probability
5	0.0113	0.00%
7	0.0786	0.00%
9	0.9276	0.00%
10	2.5715	0.00%
12	45.2243	0.00%
15	68.1367	19.79%

TABLE II
STOP CONDITION: 8 HOURS (480 MINUTES)

Arrival rate λ	Average delay (min)	Dropping probability
5	0.0047	0.00%
7	0.0395	0.00%
9	0.4903	0.00%
10	6.0127	0.00%
12	48.9380	2.17%
15	76.6124	23.05%

TABLE III
STOP CONDITION: 10 HOURS (1200 MINUTES)

Arrival rate λ	Average delay (min)	Dropping probability
5	0.0066	0.00%
7	0.0601	0.00%
9	0.5269	0.00%
10	8.0827	0.00%
12	32.3910	0.00%
15	49.1338	10.18%

TABLE IV
STOP CONDITION: 5000 ARRIVALS

Arrival rate λ	Average delay (min)	Dropping probability
5	0.0061	0.00%
7	0.0802	0.00%
9	0.6155	0.00%
10	1.2970	0.00%
12	60.7049	6.53%
15	77.0150	22.97%

TABLE V
STOP CONDITION: 10000 ARRIVALS