Single Server Queuing Model

Suraj Nath

25 February 2017

Single Server Queuing Model

Problem Statment

Write a R program to simulate a single-server queue, with the following assumptions.

- Assume inter-arrival times are independent and identically distributed (IID) random variables.
- Assume service times are IID, and are independent of inter-arrival times.
- Queue discipline is FIFO.
- Start empty and idle at time 0.
- First customer arrives after an inter-arrival time, not at time 0.
- Stopping rule: When nth customer has completed delay in queue, stop simulation.
- Update system, state variables, clock, event list, statistical counters, all after execution of each event. You are expected to create a function with three arguments

1. Inter arrival rate
2. Service rate, and
3. Stopping rule

Quantities to be estimated are

1. Expected average delay in queue (excluding service time) of the 20 customers completing their delays,
2. Expected average number of customers in queue (excluding any in service), and
3. Expected utilization (proportion of time busy) of the server.

Note: (a) The system begins at time 0 with no customers in the system and an idle server. (b) The times between arrivals are mutually independent and identically distributed exponential random variates with arrival rate lamda = 1 (you can assume initially). (c) The service times are mutually independent and identically distributed exponential random variates with service rate mu = 0.5 (you can assume initially). (d) The server does not take any breaks. (e) A customer departs the system once the service is complete. (f) When 20th customer has completed delay in queue, stop simulation.

Code

Function for SSQ

```
ssq <- function(ra, rs, numOfCustomers) {
  wait_time <- 0;
  total_arrival_time <- 0;
  total_wait_time<- 0;
  total_idle_time <- 0;
  prvious_service_time <- 0;
  for (i in 1:numOfCustomers)
  {
    serTime <- rexp(1, rs);
    interArrTime <- rexp(1, ra);
```

```
    total_arrival_time <- total_arrival_time + interArrTime;
    wait_time <- wait_time - interArrTime + prvious_service_time;
    prvious_service_time <- serTime;
    if (wait_time >= 0)
    {
      total_wait_time <- total_wait_time + wait_time
    }
    else
    {
      total_idle_time <- total_idle_time - wait_time;
      wait_time <- 0;
    }
  }
  print(paste("Expected Utilization is",1 - total_idle_time/total_arrival_time,sep = " "))
  print(paste("Expected Average delay in the queue is",total_wait_time/numOfCustomers,"minutes.", sep =
  print(paste("Expected Average Customers in the queue is",total_wait_time/total_arrival_time, sep = "
}
```

Output

```
# ssq(lamda, mu, number_of_cust)
ssq(1,0.5,20)
```

```
## [1] "Expected Utilization is 0.976774692216033"
## [1] "Expected Average delay in the queue is 14.8433542527212 minutes."
## [1] "Expected Average Customers in the queue is 11.3849987678368"
```

Conclusion

In this assignment we learned about simulation in depath and more about simulation system variables

I also created other implimentation using simmer of this which can be found on my my github account (https://github.com/electron0zero/R-projects/tree/master/single_server_queue)