Modeling, Simulating, and Measuring the Performance of Queues

Anthony Hung

2019-03-01

Prerequisites

This vignette continues from concepts covered in the vignette: "Introduction to Queueing Theory and Queueing Models".

Introduction

In addition to simply being able to represent waiting lines mathematically, queueing theory allows for the evaluation of the behavior and performance of queues. Being able to measure the performance of queues also allows us to determine the effects of altering components of the queue on performance.

Simulating a M/M/1 queue

In simulating a M/M/1 queue, we want to keep track of three values of the queue over time.

1. Arrival times of customers
2. Departure times of customers
3. The number of customers in the system at every moment of arrivals or departures

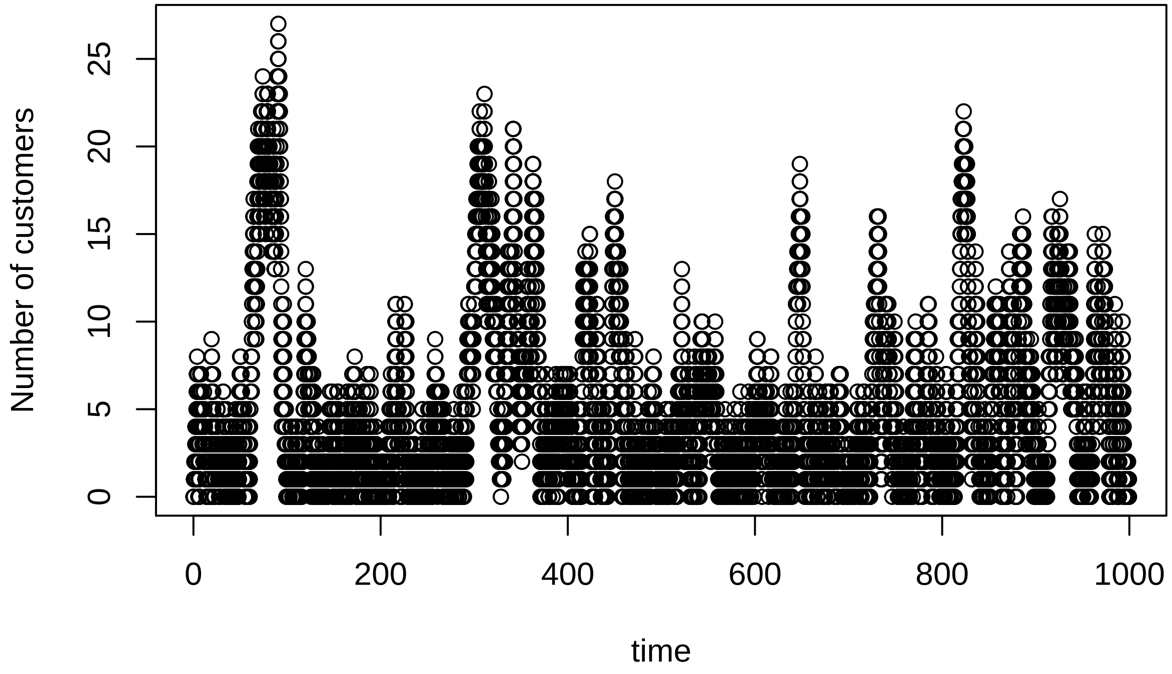
In simulating the queue behavior, we can take advantage of the superposition property of combined independent Poisson processes. Since arrivals and departures are indepdendent, the number of events in the combined process can be represented as a Poisson process with parameter $\lambda_{sum} = \lambda + \mu$. The probability of an event in this combined process being an arrival is $\frac{\lambda}{(\lambda+\mu)}$, and the probabilty of it being a departure is $\frac{\mu}{(\lambda+\mu)}$.

The function "simulate_MM1" simulates the number of customers in a M/M/1 queue over time given values for lambda, mu, and N_0 from T_0 to T_{max}. It also keeps track of when events (arrivals or departures) occur during the time periods and what type of event occurs at each of those moments.

```
lambda <- 4
mu <- 5

simulate_MM1 <- function(lambda=lambda, mu=mu, N0=0, Tmax=1000){
  #Initialize vectors to store each of the values of interest throughout the simulation
  events <- 0 #stores the type of event (1 for arrival, -1 for departure)
  Times <- 0 #times of events
  customers <- N0 #number of customers at each time in Times

  while(tail(Times,1) < Tmax){ #keep simulating until you have an event at a time greater than Tmax

    if(tail(customers,1)==0){ #separate behavior occurs if system currently has 0 customers
      tau <- rexp(1, rate=lambda) #interarrival intervals are exponentially distributed
      event <- 1 #only an arrival can occur if thre are 0 customers
    } else {
      tau <- rexp(1, rate=lambda+mu) #inter-event intervals are exponentially distributed
      if(runif(1,0,1) < lambda/(lambda+mu)){ #if runif is less than P(event = arrival)...
        event <- 1 #call the event an arrival
      } else{
        event <- -1 #otherwise, call the event a departure
      }
    }

    #now that we have simulatd one event, we need to do some accounting
    customers <- c(customers, tail(customers,1)+event)
    Times <- c(Times, tail(Times,1)+tau)
    events <- c(events, event)
  }

  #we need to toss out the information from the last event (it occured after Tmax)
  events <- head(events, -1)
  Times <- head(Times, -1)
  customers <- head(customers, -1)

  return(list(events,Times,customers))
}
```

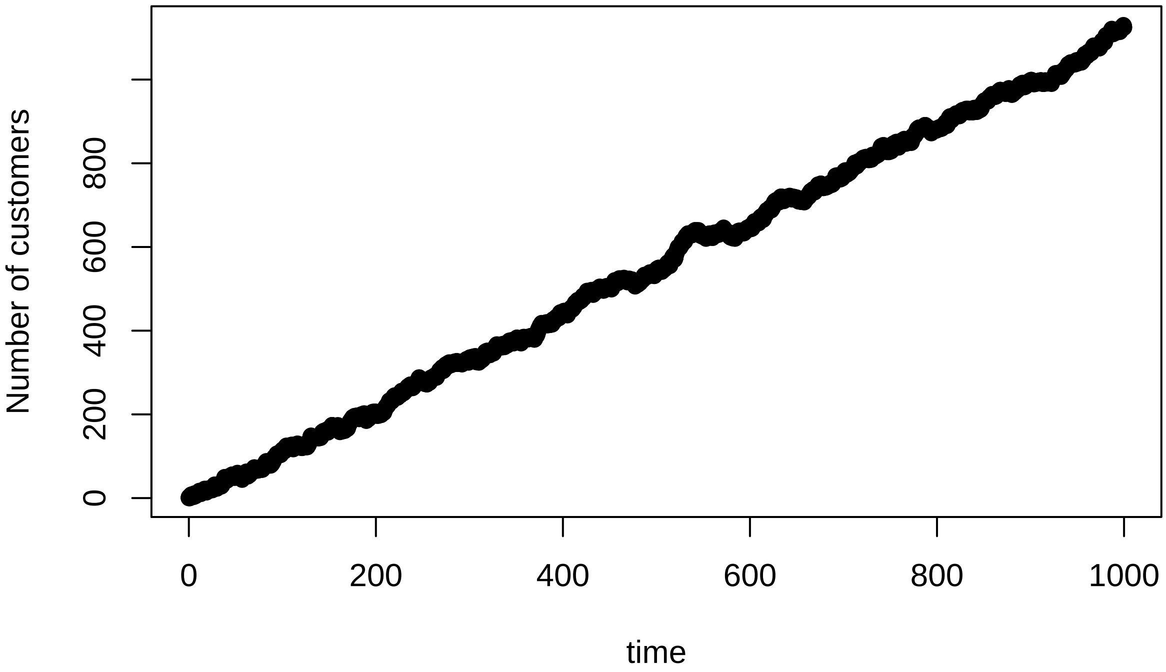
After simulating the number of customers in the queue over one run of the simulation, we can plot it.

```
sim <- simulate_MM1(lambda = 4, mu=5, N0=0, Tmax=1000)
plot(x=sim[[2]], y=sim[[3]], xlab="time", ylab="Number of customers")#number of customers vs time
```



Notice that if $\lambda > \mu$, the customer number explodes and will never reach a steady state.

```
sim2 <- simulate_MM1(lambda = 6, mu=5, N0=0, Tmax=1000)
plot(x=sim2[[2]], y=sim2[[3]], xlab="time", ylab="Number of customers")#number of customers vs time
```



Measuring the performance of queues

There are several formal quantities used to measure the performance of a queueing system (with c servers).

1. p_j := The stationary probability that there are j customers in the system
2. a := Offered load. The mean number of requests per service time.
3. ρ := Traffic intensity. Offered load per server (a/c).
4. a' := Carried load. Mean number of busy servers.
5. ρ' := Server occupancy. Carried load per server (a'/c).
6. W_s := Mean length of time between a customer's arrival and the customer's departure from the system.
7. W_q := Mean length of time between a customer's arrival and when the customer's service starts.
8. L_s := Mean number of customers in the system, including those in the buffer and at servers.
9. L_q := Mean number of customers waiting in the buffer.

Perform an analysis of the simulated M/M/1 queueing process including a steady state analysis of the system (stationary distribution) and performance measures (loss probability, average waiting time, idle time).

Multiple servers: the M/M/c queue