Simulation of Single-Server Queuing Systems

Mark Randles
2005-01-27
CS464

Introduction

The goal of this lab was to explore the single server queuing system. The system works like so: there is a singe server that can process one object at a time. As the objects arrive they get queued up in the order they arrive until the server has processed the object in-front of it. When all the objects are processed the simulation ends.

The actual simulation, as provided, simulates the system as described above. New events (arrivals) get put in a simple queue, with the first event occurring at some time *t* after time 0. Each new event is given an attribute of it's departure time from the queue and processing (one in the same for this simulation). The simulation code looks to see which time is less, the time for a new arrival event to occur or for a departure to occur. The less time become the new time for the simulation and that event takes place. The simulation then repeats the steps until there are no more arrival or departure events.

Analysis

The single-server model is simple to theorize about, but can become complex to understand it's behavior. Since there are a number of values that can flux and produce a complex behavior. The lab asked us to specifically vary certain initial values and then analyze the results. The variables that we were asked to change were: the mean interarrival time, the mean service time, and the number of customers in the simulation.

The mean interarrival time is the average arrival time of a new customer to the simulation. Based on this time customers will arrive to the queue either faster or slower, and change the time that customer will wait in the queue before being served.

As you can see in Chart #1 (please see the Appendix), as the mean interarrival time increases the wait period in the queue drops almost to zero. This makes sense, since as the customers arrive slower, there is more time to process them before more arrive, and the queue becomes less filled. The actual shape of the graph suggest that the average time in the queue is related to the inverse or inverse squared of the mean interarrival time.

The mean service time, however, does not display this behavior. As displayed in Chart #2, as the mean service time increases so does the average delay of a customer in the queue. From the extreme slope of the graph, it can be surmised that the average delay time of a customer in the queue varies at some exponential value of the mean service time. This behavior makes sense, since the queue would become more full, since it takes longer to process each customer.

The last value that we varied was the number of customers. As you can see from Chart #3, as the average customers increased, we start to oscillate around a median value. This behavior can be explained since the number of customers in this simulation is essentially the number of times we repeat our simulation loop. It would only follow that as we ramp up the number of repeats to the simulation we would gain more and more accuracy to the actual average delay in queue if the simulation was run in an infinite loop. The value that the average delay in the queue should hover around is the average of the mean interarrival time and the mean service time.