ECON381 Fall 2024

Homework Assignment 1

In this assignment you are required to prepare a preliminary analysis about a A Prioritized Event Simulation. In this assignment you are not required to code in Java. It is intended to make you do some research and understand better the types of problems you will be facing in the real world versus the idealized problems we solve in the classroom.

Consider ABC Bank which is a typical deposit bank. Therefore the main task is to handle money transactions (EFT, or deposit/withdrawal at cashier).

- Each money transaction request is first handled by a load balancer (LB) that forwards the transaction to an available server.
- Each transaction takes time t, a function of the size of the transaction.
- Between two transactions, servers need some random duration to re-order their internal records.
- There are S servers serving banking transactions.
- There are queues inside each server, so that each Server can be assigned a number of transactions. However, the load balancer also has its own queue, so that servers' queues need not be utilized.
- All servers update their queues as they process requests.

ABC Bank so far has been assigning new requests on the least busy server. This is defined as the server with least number of active requests assigned. In case of equality, a random server is selected.

Questions

1. Try to model the load balancer's server selection process. Sketch the data structure for the problem.
2. Is there a sorting-related problem here?
3. Which type of algorithm would be better suited for this problem? Why?
4. Suppose ABC Bank decided to implement a "high priority customer" program. In this program these high priority customers' transactions are processed earlier than those of the ordinary customers. How would you handle this change in your data structure and your algorithm?
5. Suppose banking regulations now enforce all banks to prioritize transactions that have been waiting the longest duration. How would you handle this change in your data structure and your algorithm?
6. How would you classify the current design and your proposed designs in terms of sorting stability?