COMPSCI 711 Distributed Systems Assignment 3

Q1

a) Auditing System

Asynchronous time system. Don't need the information distributed to be in order.

b) Broadcast Scheme

Synchronous time system. It matters what order the messages are broadcast, so we need to keep track of the timestamps of the messages.

Q2

1. Before a view change, the coordinator blocks all operations.
2. Once it knows that all threads have completed their local view operations, the coordinator tells all nodes to stop what they are doing and block for a certain time period of time to ensure that all of the nodes have had a chance to respond. This will prevent deadlock if there is a node failure during the view change process.
3. When the wait time has elapsed, each node starts executing read-only operations until it has reached its point in the lock acquisition graph where it originally blocked. It then continues executing write operations to put its part of the state into the new views. After every node has completed this process, all locks are acquired and an update can be made safely without any risk of deadlock or data corruption.

Q3

Q4

The algorithm would be as follows:

1. Find the machines hosting all tasks.
2. From each machine find out the CPU load information and collect it.

3. Sort all the collected data and create a chart with the CPU Load on each of the machines.

Q5

We can know whether the probes used in the cyclic garbage collection have reached an object through all its predecessors, by recording the number of times that each object has been checked. This information is recorded as a boolean variable that records if the object has been visited before or not, initialized to false at the beginning of the algorithm.

At any time, there are three possible states for an object: it may be live, it may be recycled or there may be no information about it yet. To make sure that all objects are visited at least once before recycling any objects, we need to check whether each object is live before recycling it.

If the boolean record of the visited-ness is false, then we must set it to true and mark the object as recycled. Now we repeat this process for all its successors.

The procedure is as follows: Repeat While some objects remain unvisited Check if all predecessors of unvisited objects have been visited If they have not been visited yet, set them as unvisited Recycle unvisited objects Otherwise return

Q6

The algorithm needs to be modified so that it keeps track of the number of packets in flight. It should maintain the same functionality as the unmodified algorithm, with one exception: when a packet is transmitted on a channel, this modified algorithm will decrement by one the count of packets in flight on that channel.

Assume that there are two channels A and B. Channel A has a count of 3 packets in flight, and channel B has a count of 1 packet in flight. In both cases, when a packet is transmitted on a channel, this modified algorithm will decrement by one the count of packets in flight on that channel. So after a transmission from Channel A to Channel B, its count would be 2 packets in flight. And after a transmission from Channel B to Channel A, its count would be 1 packet in flight.

Q7

a) Accept request

No, it cannot ignore the prepare request message. This new message should be accepted as the previous one is outdated.

b) Respond to prepare request message

Without the version numbers acting as a priority system, a cycle of nodes can be formed, which leads to deadlock. Normally this would be easily resolved with the version numbers.