

Homework 3

Md Ali

CS 550: Advanced Operating Systems

October 22, 2020

Exercise 1. The root node in hierarchical location services may become a potential bottleneck. How can this problem be effectively circumvented?

Proof. We can install separate root nodes for each section, this will lead to the partitioned root node should be spread across the network to insure that accesses to it will also be a spread across the network to avoid a potential bottleneck. \square

Exercise 2. In a hierarchical location service with a depth of k , how many location records need to be updated at most when a mobile entity changes its location?

Proof. With a mobile entity change its location we will essentially have two combination of insertion and deletion of operation. This ultimately leads to each process at the worst to be $k + 1$ to change the location of it. Since we have two operation we will be with the resulting changes needed at the worst is $2k + 1$ records. \square

Exercise 3. High-level name servers in DNS, that is, name servers implementing nodes in the DNS name space that are close to the root, generally do not support recursive name resolution. Can we expect much performance improvement if they did?

Proof. Not particularly, this is due to that the high-level name servers typically have global layers of the DNS name space, meaning changes don't happen often here. \square

Exercise 4. Explain how DNS can be used to implement a home-based approach to locating mobile hosts.

Proof. When the name is resolved, it should return the current IP address of the host, so each time the hosts moves location, it should contact a home server and provide it with its current address. This means that other name servers are told not to cache the address given. \square

Exercise 5. Outline an efficient implementation of globally unique identifiers.

Proof. To implement global unique identifiers the first thing is to take the network address of the machine where the identifier is generated, get the local time of the address and generate a random number with it. \square

Exercise 6. Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does tick, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

Proof. Since the clock that has the 10 millisecond off will be causing the error, in a minute this will result in a 600 msec per minute. Hence the maximum clock skew will be 600 msec per minute. \square

Exercise 7. Many distributed algorithms require the use of a coordinating process. To what extent can such algorithms be considered distributed? Discuss.

Proof. The fact that there is a coordinator does not make the algorithm any less distributed. In regards to distributed algorithms with a nonfixed coordinator, the coordinator is chosen among the processes that are part of the algorithm. \square

Exercise 8. A distributed system may have multiple, independent resources. Imagine that process 0 wants to access resource A and process 1 wants to access resource B. Can Ricart Agrawalas algorithm lead to deadlocks? Explain your answer.

Proof. The Ricart & Agrawalas algorithm does not have any deadlocks as each resource is handled independently from all others. \square

Exercise 9. Give the Lamport logical time for the diagram in the assignment.

Proof. Below is the processes with each events with a number of what it would go in together.

Process 1:

A - 1
B - 2; J - 3; K - 4; G - 5
C - 3
D - 4
E - 5
F - 6
G - 7

Process 2:

H - 1
I - 2; E - 3; F - 4; L - 5
J - 3
K - 4
L - 5

Since both processes have their own respective logical time and only sync up when they are requesting or sending data to one another. Below is Lamport's paper.

<https://lamport.azurewebsites.net/pubs/time-clocks.pdf>

\square

Exercise 10. Please give the Vector logical time for the diagram in the assignment.

Proof. Below is the matrix that corresponds with each process. Each row is an event (e.g. A, B, ... etc.) and the columns are for process 1, 2, and 3.

Process 1:

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 3 & 4 & 1 \end{bmatrix}$$

Process 2:

$$\begin{bmatrix} 0 & 1 & 0 \\ 2 & 2 & 0 \\ 2 & 3 & 1 \\ 2 & 4 & 1 \end{bmatrix}$$

Process 3:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

□

Exercise 11. Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?

Proof. Any one of them, it all depends if the owner should always see the same mailbox or not. I think the most appropriate and simplest would be that of a primary based local write protocol, where the primary is always located on the user's device. □

Exercise 12. In the Bully algorithm, a recovering process starts an election and will become the new coordinator if it has a higher identifier than the current incumbent. Is this a necessary feature of the algorithm?

Proof. This is not a necessary feature as instead of sending an election, the recovering process can send an inquiry asking who is the coordinator now, which would avoid the entire election as a whole. □