Homework 3

CS550 (V01)

Amit Nikam (A20470263)
anikam@hawk.iit.edu
Fall 2020

1. **(5 points)** The root node in hierarchical location services may become a potential bottleneck. How can this problem be effectively circumvented?

   Ans: Since random bit strings are used as identifiers, we can partition the identifier space and install a separate root node for each part. Then they should be spread across the network so that accesses to them will also be spread.

2. **(5 points)** 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?

   Ans: When a mobile entity changes its location, we update the new location records and then delete them from the previous location record. In the worst case scenario k+1 operation will done in each step, but since the root is same for both steps, total location records that need an update are 2k+1.

3. **(10 points)** 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?

   Ans: No since high-level name servers constitute the global layer of the DNS name space, so changes to the name space do not occur often. Recursive name resolution is highly beneficial at lower level; at higher levels using cache can be better.

4. **(10 points)** Explain how DNS can be used to implement a home-based approach to locating mobile hosts.

   Ans: The mobile host update the DNS server with its current address. The DNS name should be resolved and should return the current IP address. The DNS server providing ip will work as name server for the mobile host and mobile host must update the home server of its current address(away). This is a poor implementation over all.

5. **(10 points)** Outline an efficient implementation of globally unique identifiers.

   Ans: We first can take the network address of the machine where the identifier is generated, append the local time along with a generated pseudo-random number to it. It is possible that another machine in the world can generate this same number, although chances of this happening is negligible.

6. **(10 points)** 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?

Ans: The second machine has delay of 10 ticks per millisecond, so in a minute, i.e. 60000 milliseconds it will be delayed by 600,000 i.e. 600 milliseconds.

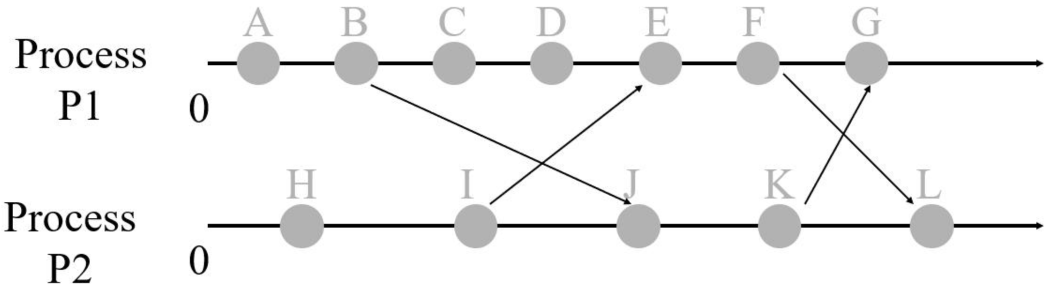
7. **(10 points)** Many distributed algorithms require the use of a coordinating process. To what extent can such algorithms be considered distributed? Discuss.

   Ans: In distributed algorithms, a coordinator is chosen among the processes that form part of the algorithm, this is somewhat like centralized algorithm. Having a coordinator does not make the algorithm less distributed since the coordinator is also picked in a distributed fashion.

8. **(10 points)** 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.

   Ans: The Ricart and Agrawala algorithm itself does not contribute to deadlock since each resource is handled independently of all the others. It all depends on the implementation of the system. If the rules are defined that don't let processing holding resources to wait for other blocked resource, deadlock will not occur.

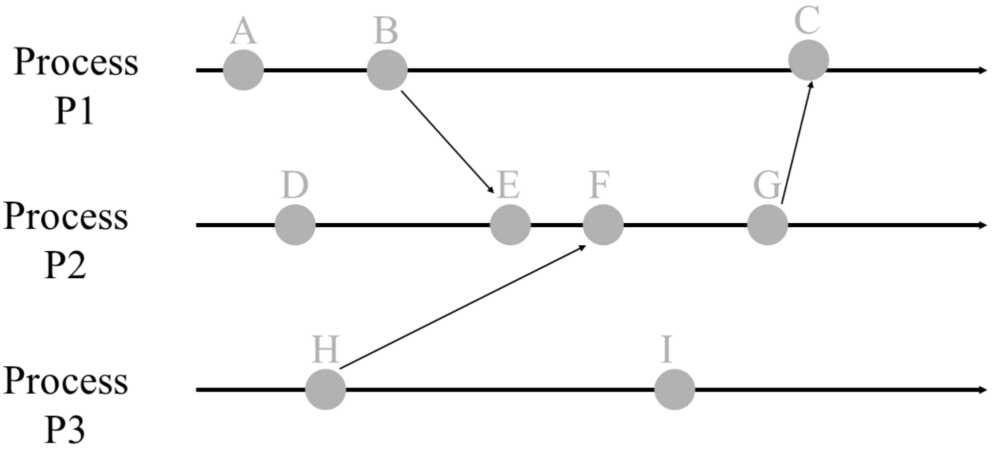
9. **(5 points)** Give the Lamport logical time for the above diagram:



Ans: The Lampart logical time for this is 7.

Process 1		Process 2	
A	1	H	1
B	2	I	2
C	3	J	3
D	4	K	4
E	5	L	7
F	6		
G	7		

10. **(5 points)** Please give the Vector logical time for the following diagram:



Ans: Process 1 = [3,4,1], Process 2 = [2,4,1], Process 3 = [0,0,2].

Process 1		Process 2		Process 3	
A	[1,0,0]	D	[0,1,0]	H	[0,0,1]
B	[2,0,0]	E	[2,2,0]	I	[0,0,2]
C	[3,4,1]	F	[2,3,1]		
		G	[2,4,1]		

11. **(10 points)** 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?

    Ans: All of them since the owner should always see the same mailbox. In fact, a primary-based local-write protocol implementation would be appropriate where the primary is always located on the user's mobile.

12. **(10 points)** 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?

    Ans: No it is not a necessary feature. A recovered process could check with processes if a new coordinator is elected, if not then become the coordinator again else send the higher identifiers to the new coordinator. Although a re-election can be beneficial if the difference in identifiers is significant.