DS: READING QUESTIONS 5

14 NOV. 2016

ALI ABDULMADZHIDOV

1. Consider the processes in Fig. 6.14 have continued execution and process P1 currently having VC1=(1,2,0) receives a message m from process P2, ts(m)=(2,1,1). What information does P1 have, and what will it do when receiving m?

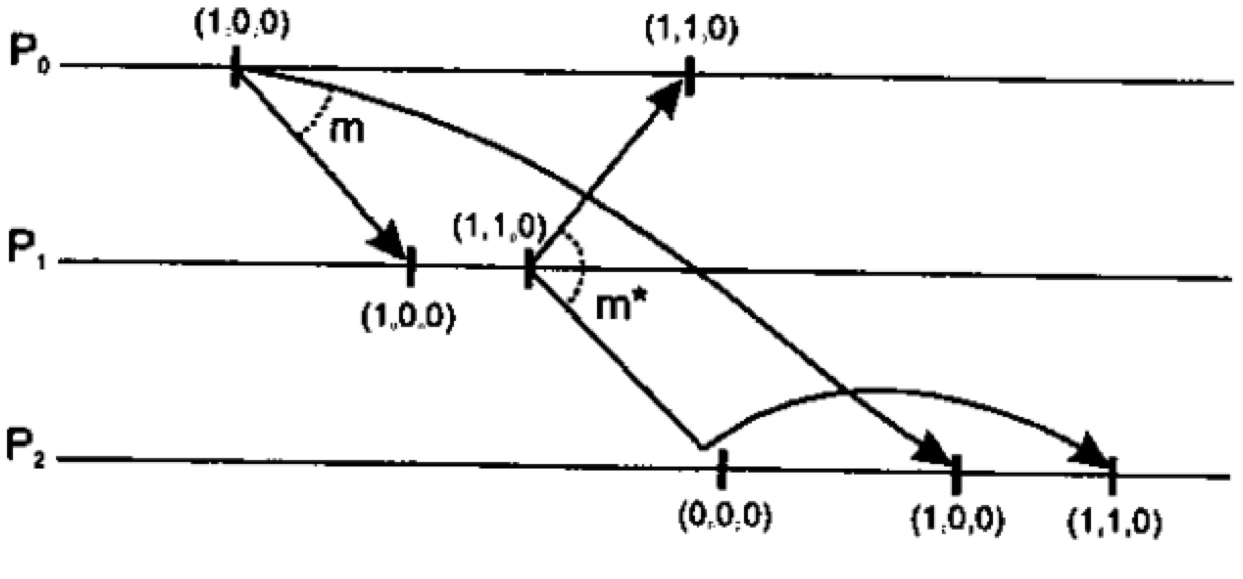

Figure 6.14: Enforcing causal communication.

P1 has (1.2.0). After receiving message (2.1.1) from P2, he'll see that one he missed one message from P0 and wait for it (2.2.0). Then he'll process (2.1.1) and become (2.2.1)

2. What would happen with the Ricart and Agrawala algorithm if a process crashes? Discuss

Crash of any process would fail all algorithm if we discuss by book. It is written that

> 'If any process crashes, it will fail to respond to requests. This silence will be interpreted (incorrectly) as denial of permission, thus blocking all subsequent attempts by all processes to enter any of their respective critical regions.'

But also there is the way to fix this problem. In default config process answers only if resource is safe to interact. But it's better to convince process answer either OK or NOTOK. If so, silence of crashed process wouldn't block all algorithm

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

Like a centrilized systems, distributed one should have coordinator to rule shared resources and avoid any conflicts But unlike of usual algorithms, distributed choose coordinator among them, but that doen't make that algorithm less distributed.

4. Which of the mutual exclusion solutions discussed in class requires the fewest messages under heavy contention?

The most effective algorithm is distributed. It needs only $2 * (n - 1)$ messages. But if number of processes is big, centralized show more efficient, cause always need only 3 message per entry/exit.

5. Consider a bully-based technique for electing a coordinator. What kind of failure would result in multiple coordinators being elected? Under what assumptions is this acceptable?

There is 2 ways when there can be 2 coordinators.

1) If somehow we have two nodes with same max id. In that situation they wouldn't send election message to each other, and nobody will interrupt with OK them.

2) If connection fails after node sends election poll to above. For example, node 6 sends election message to 7. 7 receives message, but when he tries to send OK response, connection fails. 6 be coordinator because he didn't get answer from 7, 7'll be coordinator because he send OK to 6.

6. Does the HTTP protocol use a push-based or pull-based approach for updates? Why?

HTTP is pull-based protocol because by default server waits for user http request GET/POST/.. and then process it and response with some answer.

7. When using a lease, is it necessary that the clocks of a client and the server, respectively, are tightly synchronized?

No. If client thinks, that his clock is not synchronized with server's one, he just tries to get lease before current one expires.

8. Consider a non blocking primary-backup protocol used to guarantee sequential consistency in a distributed data store. Does such a data store always provide read-your-writes consistency?

No. Unlike blocking protocols, where process ca disconnect only when update is successfully finished and reached to other replicas, in unblockin it can disconnect, just assured that its update is processed by server. There is no guarantees that update has reached to other replicas in that way.

9. A file is replicated on 10 servers. List all the combinations of read quorum and write quorum that are permitted by the voting algorithm

There're 2 rules for quorum gathering.

$Nr + Nw > N$

$Nw > N/2$

N = 10

(Nr, Nw) = (1, 10), (2, 9), (3,8), (4, 7), (5, 6)

10. How would you characterize the consistency model of Facebook?

Facebook provides per-object sequential and read-after write consistency in caches and eventual consistency across the caches. When user sessions are provided by one leaf cache, they receive per-object and read-and-write consistency, but if sessions are spread between several leaf caches (in situations when they are load-balanced between multiple web clusters) they receive eventual consistency.