CSI3344 Distributed Systems

Workshop Solution 06

Q1. Refer to the following figure (Figure 1):

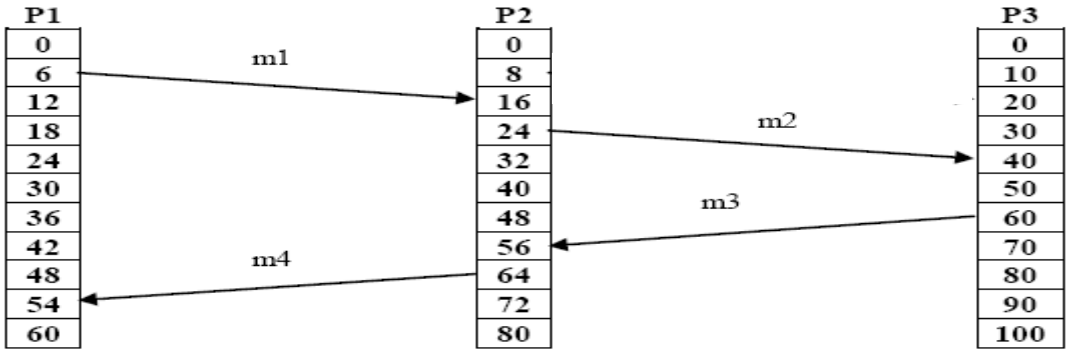
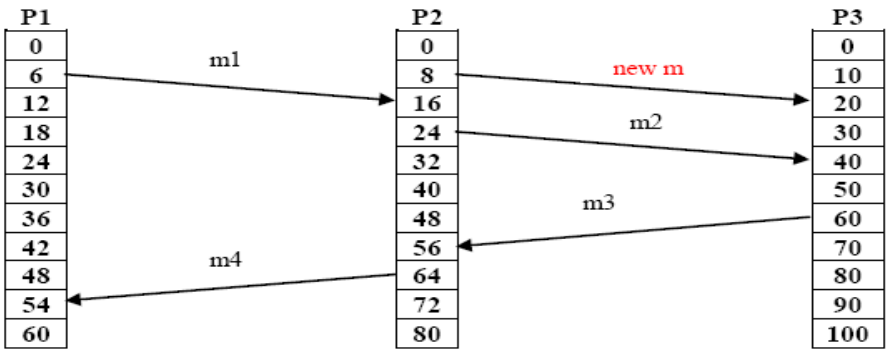


Figure 1. Three processes, each with its own clock, and four messages between processes. The clocks run at different rates.

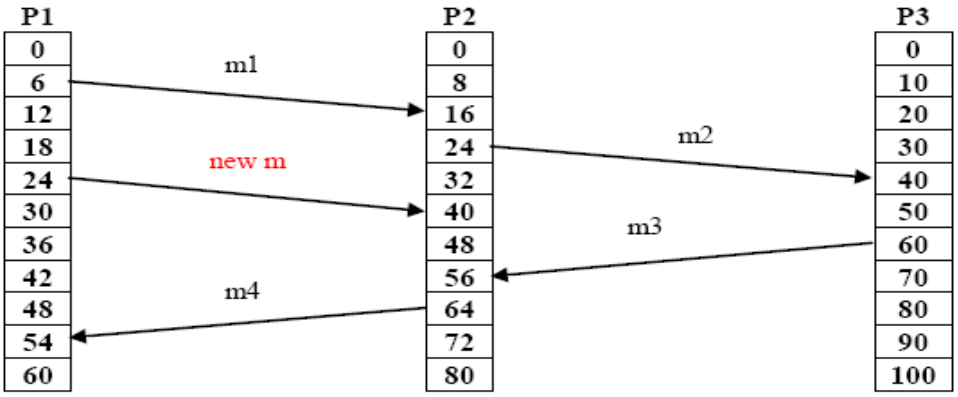
(a) Add a new message to the figure that is concurrent with message *m1*, that is, it happens neither before *m1* nor after *m1*.

A: The solution cannot involve P1 or it would be ordered. Thus it must be a message from P2 to P3 or from P3 to P2. If it departs or arrives from P2 after 16, it will be ordered with respect to *m1*, so it must depart or arrive before 16. The possibilities are a message leaving process P3 at 0 and arriving at process P2 at 8, or a message leaving process P2 at 0 and arriving at process P3 at 10. Both of these are concurrent with *m1*. One of these is as below:



(b) Add a new message to the figure that is after *m1* but before *m4.*

A: Based on the happened-before relation, there are many acceptable answers to this question – for example, any message sent from P1, departing after 6, or any message sent from P2, departing at or after 16 (however a message from P3 is unacceptable if it departs between 0 and 30, or departs after 60). The following figure shows one of the acceptable solutions.



Q2. The following diagram (Figure 2) shows that two updates on the same bank account at two replicated databases in different places take place at the same physical time, leaving it in an inconsistent state. Suppose that the account balance is $5000 before the updates take place. One update is to deposit $2000 to the account. The other update is to increase the interest by 1%.

The two copies of the same account must have the same balance after the completion of the two updates, no matter whichever commits first.

Describe how to use Vector Timestamps to keep this replicated database system consistent in the given conditions.

[*Hints: Refer to Chapter 6 in the Text. Two cases to be considered.*]

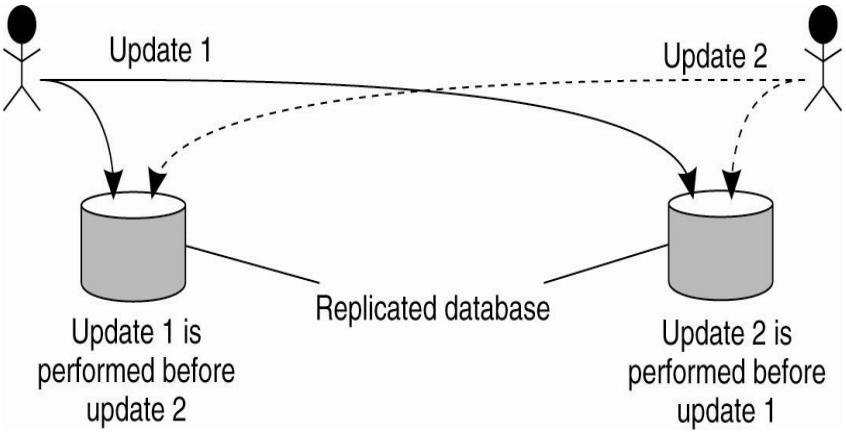


Figure 2: Updating a replicated database and leaving it in an inconsistent state

A: Refer to Chapter 6, from page 247.

END OF THE WORKSHOP SOLUTION