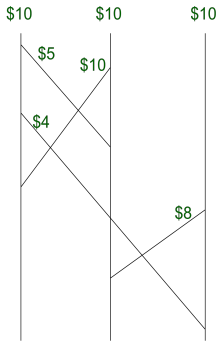
Lecture 8 – Consistent Distributed Snapshots

Distributed Snapshots

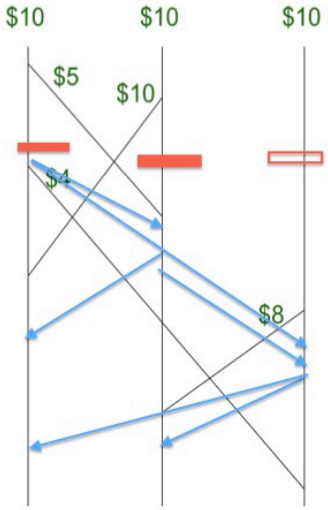
a. Example:
   - Distributed bank, money sent in reliable messages.
   - Audit problem:
     - Count the total money in the bank.
     - While money continues to flow around.
     - Assume total amount of money is conserved (no deposits or withdrawals).



b.

c. In picture below, start snap at first bar:
   i. Node 1 has $5
   ii. Node 2 has $0
   iii. Node 3 has $10
   iv. Channel 2->1 has $10
   v. Channel 1-0>2 has $5

- Distributed bank, money sent in reliable messages.
- Audit problem:
  - Count the total money in the bank.
  - While money continues to flow around.
  - Assume total amount of money is conserved (no deposits or withdrawals).



d.

e. In Chandy-Lamport snapshot:
   i. Node 1 records $5
   ii. Node 2 records $5
   iii. Node 3 records $2
   iv. Node 1 records 2->1: $10
   v. Node 2 records 3->2 $8

f. Why is this reordering correct?

i. Problem: process could change state asynchronously (internal events) before the markers it sends are received by other sites
ii. Has same events, can get from to this state with same events (in different order) from input
iii. Can get from this state to same output event with same events (in different order)
iv. Key idea:
    1. Reorder events in total order so that all pre-snapshot events happen, then snapshot, then post-snapshot events
v. Notion:
    1. Actual states = global states that occurred
    2. Feasible states = states that could occur according to local state machine at each process
vi. Based on logical time: can reorder logically concurrent events in the total order and get an equivalent output
vii. EXAMPLE:
    1. Real order:
        a. 1 sends 2 $5 - PRE
        b. 2 sends 1 $5 - PRE
        c. 1 sends 3 $4 - POST
        d. 2 receives $5 from 1 - PRE
        e. 1 receives $10 from 2 - POST
        f. 3 sends $8 to 2 - PRE
        g. 2 receives $8 from 3 - POST
        h. 3 receives $4 from 1 - POST
    2. So can reorder
        a. Move up d, f – could happen at any time
        b. REDRAW!
viii. Suppose we could not reorder:
    1. Means there is a "happens before" relationship between the things being reordered
    2. Implies either
        a. They are in the same process -> but not reordering anything in a single process
        b. There is a line of causal communication between them
    3. If causal communication, then must have been a message
        a. Would have an earlier (but post-snapshot) event followed by a later (but pre-snapshot) event with communication
        b. But by rule, always send marker after snapshot, so recipient (pre-snapshot) would have had to snapshot,
        c. CONTRADICTION!

g. Effectively picks a "virtual time" for snapshot, moves all events to be before or after that event by stretching/compressing timelines

i.

2. FLAWS:
    a. State external to the system not captured (e.g. clients of a distributed service)