COS418 Precept 4 - Distributed Snapshots

10/6/17

Chandy-Lamport Snapshot Algorithm

Starting the snapshot process on a server:

- Record its local state
- Send ***marker messages*** on all outbound interfaces

When you receive a ***marker message***:

- If you haven't started the snapshot process yet, record your local state and send ***marker messages*** on all other interfaces
- Start recording messages you receive on all *other* interfaces
- Stop recording messages you receive on *this* interface

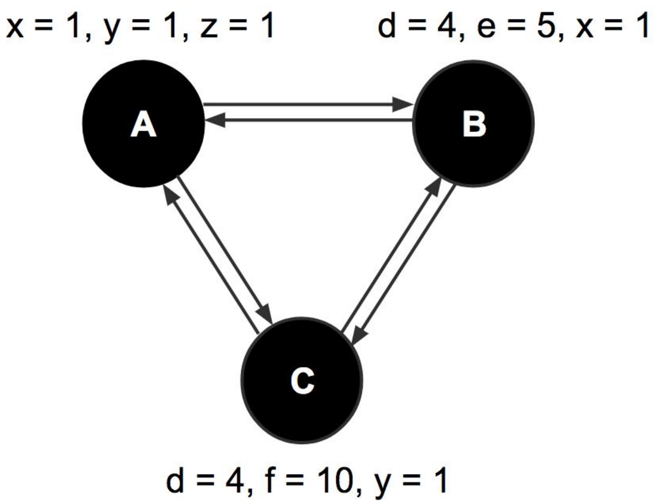
Terminate when all servers have received ***marker messages*** on all interfaces

Distributed Database Example

Below is a distributed database consisting of 3 servers, A, B, and C. Each server is responsible for storing a fraction of the data. Some keys are replicated on multiple servers for fault tolerance and availability.

The ***Set(key, value)*** message may be exchanged between servers if either:

- The client contacts the wrong server, in which case the contacted server will forward the request to the server that is responsible for the key of interest, or
- The client contacts the right server but the key is replicated, in which case the contacted server will forward the request to other replicas holding the same key



A	x = 1, y = 3, z = 1
B	d = 8, e = 10, x = 1
C	d = 4, f = 10, y = 3

Link(BA)	--
Link(CA)	--
Link(AB)	--
Link(CB)	--
Link(AC)	Set(d, 8)
Link(BC)	--

Run the chandy-lamport snapshot algorithm on the following sequence of events and record the final state recorded by the algorithm. Also mark the step at which the snapshot process finishes on each server.

B	e = 10
A → B	sends Set(d, 8)
A → C	sends Set(d, 8)
B	receives Set(d, 8) from A, d = 8
C	y = 3
C → A	sends Set(y, 3)
C	***starts snapshot***
C	receives Set(d, 8) from A, d = 8
A	receives Set(y, 3) from C, y = 3
B	receives Marker from C, ***starts snapshot***
B	e = 4
B → A	sends Set(x, 4)
A	receives Marker from C, ***starts snapshot***
A	receives Marker from B, ***ends snapshot***
A	receives Set(x, 4) from B, x = 4
C	receives Marker from B
C	receives Marker from A, ***ends snapshot***
B	receives Marker from A, ***ends snapshot***