Day 12: Snapshots
Snapshots Approaches

Synchronized
(all servers take a snapshot at a specified time, e.g., 2pm)

No guarantees that the two servers will actually take the snapshot slots at the exact time because of clock differences

Periodic Snapshots
Each server periodically captures a snapshot of the global state

The only consistent snapshot is at the beginning when no messages are sent

Chandy-Lamport protocol

Makes a lot of assumptions
Snapshot Assumptions

3 types of events
- Local
- Send
-Recv

Events that impact a snapshot's consistency

Includes the recv(e_2) but not send(e_3)

Global checkpoint includes send(e_3) but not recv(e_2)
Why is it okay to include only the send but not okay to include only the recv?

Versus

- Only send but no recv
  - Able to converge by resending msg

- Only recv but no send
  - No clear path to converge
Snapshot Assumptions

3-types of events

- local
- send
- recv

Events that impact a snapshot’s consistency

Snapshot consistency

All servers have the same state: for every message in the global snapshot, the snapshot for the servers include both the send & recv events.

Or all servers have the ability to reach the same state:

For every message in the global snapshot, either both send & recv events are included or only the send is included.
Consistent snapshots

only way for causality or to intertwine state on different servers is via msg

consistent = recv event also have then send event in the snapshot

Pick random snapshot \((e^5, e^2)\)

\(S_2\) Versus \([2, 0]\)

if pick \(e_i\) from \(S_i\)

\(VC_{e_i}[j] > VC_{e_j}[j]\)

\(j \neq i\)

Detecting inconsistent snapshot

if an event at \(S_i\) has a vector clock \(VC_i\)

and an event at \(S_j\) \((S_i \neq S_j)\) has a vector clock \(VC_j\)

then the snapshot is inconsistent if \(VC_i[j] > VC_j[j]\)

Thus \(S_i\) knows about more events at \(S_j\) than \(S_j\) does hence inconsistency
Chandy Lamport Assumptions

1. Network assumptions

a) All messages are delivered
b) No message is dropped
c) Messages are received in order sent
d) All servers can talk to each other
1. **Snapshot is triggered manually.**
   - one server who creates a snapshot & sends out markers & starts recording events

2. **Rules for processing a Marker from Sk**
   - if server hasn’t processed any markers
   - then create checkpoint of state
   - start recording events from others servers (\text{\textless} \text{Sk}\text{\textgreater})

   else
   - stop recording events from \text{Sk} & place recorded events in checkpoint

3. **Markers sending rules**
   - after creating initial checkpoint
   - send markers to all other servers before sending other message

4. **Algorithm end after all servers have received markers from every other server**
   - each server sends checkpoint to the initiator

5. **Only one snapshot at a time**
Snapshots Approaches

**Synchronized**
(all servers take a snapshot at a specified time, e.g., 2pm)

No guarantees that the two servers will actually take the snapshot shots at the exact time because of clock differences

**Periodic Snapshots**

Each server periodically captures a snapshot and we analyze snapshots to determine state

The only consistent snapshot is at the beginning when no messages are sent

**Chandy-Lamport protocol**

Makes a lot of assumptions