Snapshots

A snapshot of an execution of a distributed algorithm should return a configuration of an execution *in the same computation*.

Snapshots can be used for:

- Restarting after a failure.
- Debugging.
- Off-line determination of stable properties, which remain true as soon as they have become true.

  Examples: deadlock, garbage.

Challenge: Take a snapshot without (temporarily) freezing the execution.

Snapshots

We distinguish basic messages of the underlying distributed algorithm and control messages of the snapshot algorithm.

A snapshot of a (basic) execution consists of:

- a local snapshot of the (basic) state of each process, and
- the channel state of (basic) messages in transit for each channel.

A snapshot is meaningful if it is a configuration of an execution in the same computation as the actual execution.

Snapshots

We need to avoid the following situations.

1. Process p takes a local snapshot, and then sends a message m to process q, where:
   - q takes a local snapshot after the receipt of m,
   - or m is included in the channel state of pq.
2. p sends m to q, and then takes a local snapshot, where:
   - q takes a local snapshot before the receipt of m,
   - and m is not included in the channel state of pq.

Chandy-Lamport algorithm

Consider a directed network with *FIFO* channels.

Initiators take a local snapshot of their state, and send a control message $\langle\textbf{marker}\rangle$ to their neighbors.

When a process that hasn't yet taken a snapshot receives $\langle\textbf{marker}\rangle$, it

- takes a local snapshot of its state, and
- sends $\langle\textbf{marker}\rangle$ to all its neighbors.

Process q computes as channel state of pq the messages it receives via pq after taking its local snapshot and before receiving $\langle\textbf{marker}\rangle$ from p.

If channels are FIFO, this produces a meaningful snapshot.

Message complexity: $\Theta(E)$ (with E the number of edges)

Worst-case time complexity: $O(D)$ (with D the diameter)