Lecture 9: Global Snapshots

Distributed garbage collection

Diagram showing object reference and garbage object interaction.
Distributed deadlock detection

\[ p_1 \xrightarrow{\text{wait-for}} p_2 \xleftarrow{\text{wait-for}} \]

Distributed termination detection

\[ p_1 \xleftarrow{\text{passive}} \xrightarrow{\text{activate}} p_2 \xrightarrow{\text{passive}} \]
What should we do if P2 fails at event X?

**Global state of the system**

- We need to keep track of not only the clocks.

- But also:
  - The state of the process
  - The state of all the channels
    - Messages sent and received on the channels

- Ok, that looks easy ...
Is it really?

Global snapshot problem

- Record the state of every process and channel at an instance of time.
- System model:
  - N process (machines)
  - 2 channels between every 2 processes
    - One in each direction
    - No shared memory
  - No global clock
  - Communication is FIFO
  - No messages are ever lost, but there could be some delay
Chandy-Lamport algorithm

Marker sending rule for process $p_i$
(1) Process $p_i$ records its state.
(2) For each outgoing channel $C$ on which a marker
has not been sent, $p_i$ sends a marker along $C$
before $p_i$ sends further messages along $C$.

Marker receiving rule for process $p_j$
On receiving a marker along channel $C$:
\[\text{if } p_j \text{ has not recorded its state then}\]
\hspace{1em} Record the state of $C$ as the empty set
\hspace{1em} Execute the “marker sending rule”
\[\text{else}\]
\hspace{1em} Record the state of $C$ as the set of messages
\hspace{1em} received along $C$ after $p_j$, state was recorded
\hspace{1em} and before $p_j$ received the marker along $C$

Algorithm 4.1 The Chandy-Lamport algorithm.

---

**Example time**

Let’s do this on the board together