Let's continue our discussion.

**Properties of vector time**

- You still have:
  - Consistency
  - Total ordering
  - Event counting

- We also have:
  - Strong consistency and isomorphism

- Applications:
  - Distributed debugging
  - Global breakpoints
  - Failure recovery

**Global States**

And Snapshots
Distributed garbage collection

Distributed deadlock detection

Distributed termination detection

Can we just use clocks?

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 \( C \):

\[ \text{if } p_j \text{ has not recorded its state then} \]

- Record the state of \( C \) as the empty set
- Execute the “marker sending rule”
\[ \text{else} \]

- Record the state of \( C \) as the set of messages received along \( C \) after \( p_j \)’s state was recorded
  - 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