Motivation for Snapshot: Some Examples

- Write a distributed operating systems algorithm to detect whether an underlying user computation is deadlocked, or has terminated.
- The number of tokens in a computation never increases but may decrease. Design an algorithm to obtain an upper bound on the number of tokens.

Motivation Snapshots: Example

What happens when a computation dies while a snapshot is taken?

<table>
<thead>
<tr>
<th>active</th>
<th>terminated</th>
</tr>
</thead>
<tbody>
<tr>
<td>alive</td>
<td>dead</td>
</tr>
<tr>
<td>died</td>
<td></td>
</tr>
</tbody>
</table>

Motivation Snapshots: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Time

Detection algorithm starts

Detection algorithm ends

Should report: alive

Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Time

Detection algorithm starts

Detection algorithm ends

Should report: alive

Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Time

Detection algorithm starts

Detection algorithm ends

Should report: dead

Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Time

Detection algorithm starts

Detection algorithm ends

Should report: dead

Motivation: Specification?

In this scenario, should the detection algorithm report that the computation is alive? Or that is is dead?

Time

Detection algorithm starts

Detection algorithm ends

Either

Specification of Detection Algorithms

- Let P be a stable predicate of a system, i.e., stable(P)
- Specification of an algorithm that detects P
  - The detection algorithm must terminate
  - If the detection algorithm terminates when NOT P holds then the algorithm must report that NOT P holds
  - If the detection algorithm starts when P holds then the detection algorithm must report that P holds
  - (If the detection algorithm starts when NOT P holds and ends when P holds then it may report either)
**Motivation: Specification**

If algorithm reported dead then dead now

Detection algorithm starts  
Detection algorithm ends  

If alive now then algorithm reported alive

Detection algorithm starts  
Detection algorithm ends  

**Specification of Detection Algorithm**

- If the snapshot said that P holds then P holds now.
- If NOT P holds now then the snapshot said that NOT P holds.

**More general problem**

*Given*  

\[ P \text{ next } (P \text{ OR } Q) \]

where Q is an operating systems action, and P is in the underlying computation.

*Write an OS algorithm to detect P*

*Example:*

\[ P: \text{database computations are deadlocked} \]

\[ Q: \text{transactions have been aborted to break deadlock} \]

**Detecting a Stable Predicate P**

\[
\text{while( NOT P) } \{ \\
\text{ P = detectionAlgorithm(); } \\
\text{ sleep(T); } \\
\text{ } \\
\text{ // P holds } \\
\text{ Q } \\
\text{ // P may no longer hold } \\
\}
\]

**GENERAL APPROACH TO DETECTION ALGORITHMS**

**System Time Lines: Channel States**

- State of channel from R to P at time T (assume only one such channel)

**Consistency of Local Snapshots**

Events BEFORE snapshot  

Events AFTER snapshot  

**Pictorial Explanation: Consistency means the line can be straightened**
There exists a computation from the start state to the snapshot state, and from the snapshot state to the final state.

If a stable predicate holds in any state then it holds in all states reachable from that state.

If P holds in the start state (Stable P) then if P holds in the snapshot state then it holds in the finish state which implies that it also holds for the start state.
Global Snapshot: Prevents this situation

Error message sent after snapshot received before snapshot

Marker message: Prevents this situation

Correct: message sent after snapshot received after snapshot

Determining messages in flight

Marker line P to Q
Message sent before snapshot

Marker line Q to P
Message in flight

Determine messages in flight

Marker line P to Q
Message sent before snapshot

Marker line Q to P
Message in flight

Global Snapshot Algorithm

- When an agent takes a local snapshot it sends a marker on each of its outgoing channels.
- When an agent receives a marker, the agent takes a local snapshot if it hasn’t done so already.
- The messages in flight along a channel c to an agent Q are the messages received by Q after Q takes its snapshot and before Q receives a marker along c.