Lecture 10: Global Snapshots

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
Consistent global state

- The global state of a distributed system is a collection of the local states of the processes and the channels.
- Notationally, global state \( GS \) is defined as,
\[
GS = \bigcup_i L Si, \bigcup_i SC_i
\]
- A global state \( GS \) is a consistent global state iff it satisfies the following two conditions:
  
  **C1:** \( \text{send}(m_{ij}) \in LS_i \Rightarrow m_{ij} \in SC_i \bigoplus \text{rec}(m_{ij}) \in LS_j \) (\( \bigoplus \) is Ex-OR operator.)
  
  **C2:** \( \text{send}(m_{ij}) \notin LS_i \Rightarrow m_{ij} \notin SC_i \land \text{rec}(m_{ij}) \notin LS_j \).

---

**Figure 4.1:** An Interpretation in Terms of a Cut.
Issues in recording a global state

The following two issues need to be addressed:

I1: How to distinguish between the messages to be recorded in the snapshot from those not to be recorded.

- Any message that is sent by a process before recording its snapshot, must be recorded in the global snapshot (from C1).
- Any message that is sent by a process after recording its snapshot, must not be recorded in the global snapshot (from C2).

I2: How to determine the instant when a process takes its snapshot.

- A process $p_j$ must record its snapshot before processing a message $m_j$ that was sent by process $p_i$ after recording its snapshot.

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 $C$:

- If $p_j$ has not recorded its state then
  - Record the state of $C$ as the empty set
  - Execute the "marker sending rule"
- 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

Correctness and Complexity

Correctness
- Due to FIFO property of channels, it follows that no message sent after the marker on that channel is recorded in the channel state. Thus, condition C2 is satisfied.
- When a process $p_j$ receives message $m_{ij}$ that precedes the marker on channel $C_j$, it acts as follows: If process $p_j$ has not taken its snapshot yet, then it includes $m_{ij}$ in its recorded snapshot. Otherwise, it records $m_{ij}$ in the state of the channel $C_j$. Thus, condition C1 is satisfied.

Complexity
- The recording part of a single instance of the algorithm requires $O(e)$ messages and $O(d)$ time, where $e$ is the number of edges in the network and $d$ is the diameter of the network.
Another example

The original source of this example is Indranil Gupta’s lecture notes for his CS425 distributed systems course at Illinois.
P1 is Initiator:
• Record local state S1,
• Send out markers
• Turn on recording on channels $C_{21}, C_{31}$

P1

P2

P3

$P_1$ $P_2$ $P_3$

A     B     C     D     E

E     F     G

H     I     J

$S_1$, Record $C_{21}, C_{31}$

• First Marker!
• Record own state as $S_3$
• Mark $C_{13}$ state as empty
• Turn on recording on other incoming $C_{23}$
• Send out Markers
• S1, Record $C_{21}=C_{24}$
• $C_{31}=<>$

• S3
• $C_{13}=<>$
• Record $C_{23}$

• First Marker!
• Record own state as S2
• Mark $C_{32}$ state as empty
• Turn on recording on $C_{12}$
• Send out Markers
P1
A B C D E

P2
E F G

P3
H I J

S1, Record C_{21}, C_{24}

C_{31} = <>

P2

S2
C_{32} = <>
Record C_{32}

P3

• S3
• C_{33} = <>
• Record C_{23}

Duplicate!

C_{12} = <>

P1

S1, Record C_{21}, C_{24}

C_{31} = <>

C_{31} = <>

Duplicate!

C_{21} = <message G→D>

P2

S2
C_{32} = <>

P3

• S3
• C_{33} = <>
• Record C_{23}

• Duplicate!
• C_{32} = <>
• C_{12} = <>
• Record C_{12}
Algorithm has Terminated
Collect the Global Snapshot Pieces

S1  \( C_{21} = \text{<message G\rightarrow D }> \)  \( C_{31} = \text{<}> \)
S2  \( C_{22} = \text{<}> \)  \( C_{12} = \text{<}> \)
S3  \( C_{13} = \text{<}> \)  \( C_{23} = \text{<}> \)