CSE535 Asynchronous Systems
Models of Computations

YoungMin Kwon

A Distributed Program

- A distributed program is composed of n asynchronous processes (p_1, p_2, …, p_n) communicate with messages

- C_{ij} denotes the channel from p_i to p_j
- m_{ij} denotes a message sent from p_i to p_j
  - Communication delay is finite and unpredictable

- The global state of a computation is composed of the states of the processes and the communication channels.

A Model of Distributed Executions

- Actions are modeled as three types of events
  - Internal event
  - Message send event: send(m)
  - Message receive event: rec(m)
  - e_i^x: x^{th} event of process p_i

- Execution of a process
  - A sequential execution of its actions

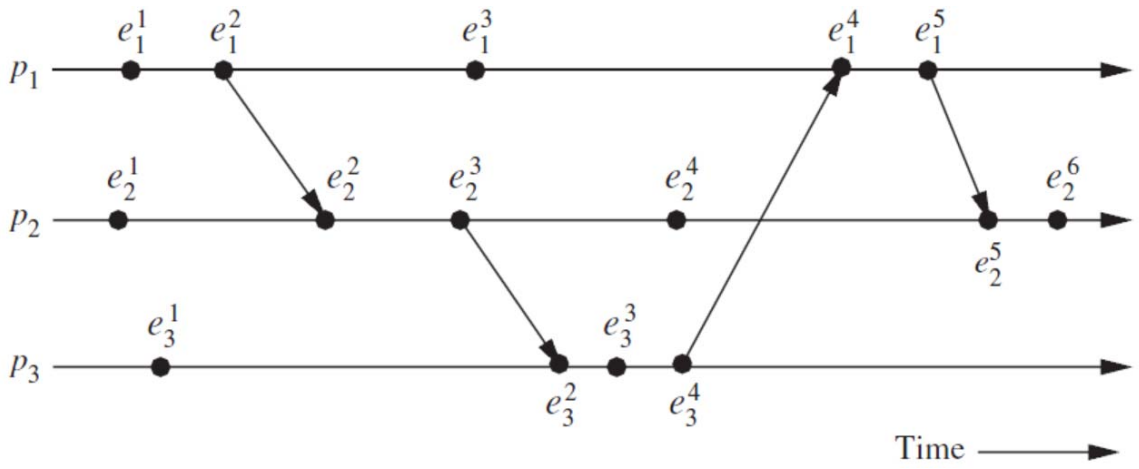
Order of Events

- Execution of a process p_i
  - Linear order of events by their occurrence

    $H_i = (h_i, \rightarrow_i)$
  - h_i: set of events of p_i
  - $\rightarrow_i$: binary relation defining a linear order on these events

- A binary relation $\rightarrow_{msg}$ captures the causal dependency due to message exchange
  - For every message m, send(m) $\rightarrow_{msg}$ rec(m)

Space-Time Diagram



- A space-time diagram of a distributed execution
  - A dot indicates an event
  - A slant arrow indicates a message transfer

Causal Precedence Relation

- Let $H = \cup_i h_i$ be the set of events executed
- The causal dependency relation $\rightarrow$ between events is

$$\forall e_i^x,\ \forall e_j^y \in H,\quad e_i^x \rightarrow e_j^y \quad \Leftrightarrow \quad \begin{cases} e_i^x \rightarrow_i e_j^y \text{ i.e., } (i=j) \wedge (x<y) \\ \text{or} \\ e_i^x \rightarrow_{msg} e_j^y \\ \text{or} \\ \exists e_k^z \in H : e_i^x \rightarrow e_k^z \ \wedge \ e_k^z \rightarrow e_j^y \end{cases}$$

- $\rightarrow$ is an irreflexive partial order relation
  - Partial order: a relation is reflexive, antisymmetric, and transitive.

Causal Precedence Relation

- For any two events e_i and e_j

$$e_i \not\rightarrow e_j \not\Rightarrow e_j \not\rightarrow e_i$$
$$e_i \rightarrow e_j \Rightarrow e_j \not\rightarrow e_i$$

- If $e_i \not\rightarrow e_j$ and $e_j \not\rightarrow e_i$ are said to be concurrent and denoted as $e_i \parallel e_j$
  - E.g. in the previous figure,

$$e_1^3 \parallel e_3^3 \text{ and } e_2^4 \parallel e_3^1$$

Models of Communication Networks

- FIFO (First-In, First-Out)
  - Each channel acts as a FIFO message queue.
- Non-FIFO
  - Senders add messages, Receivers remove messages in a random order.
- CO (Causal Ordering)
  - If send(m_{ij}) $\rightarrow$ send(m_{kj}) then rec(m_{ij}) $\rightarrow$ rec(m_{kj})
- CO $\subset$ FIFO $\subset$ Non-FIFO

Global State

- Definitions
  - LS_i^x: the state of a process p_i after event e_i^x and before e_i^{x+1}.
  - $send(m) \leq LS_i^x : \exists y: 1 \leq y \leq x :: e_i^y = send(m)$
  - $rec(m) \not\leq LS_i^x : \forall y: 1 \leq y \leq x :: e_i^y \neq rec(m)$
  - $SC_{ij}^{x,y}$: the state of a channel C_{ij}:

$$SC_{ij}^{x,y} = \{m_{ij} \mid send(m_{ij}) \leq LS_i^x \wedge rec(m_{ij}) \not\leq LS_j^y\}$$

Global State

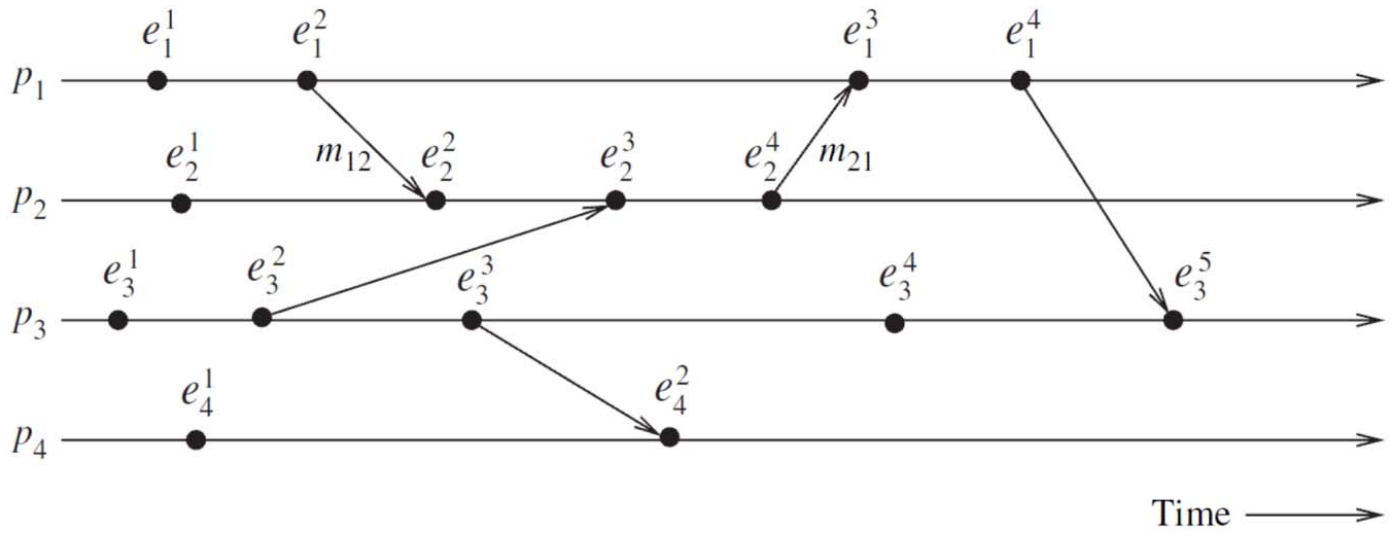
- Global State GS is defined as:

$$GS = \{\bigcup_i LS_i^{x_i}, \bigcup_{j,k} SC_{jk}^{y_j, z_k}\}$$

- A global State $GS = \{\bigcup_i LS_i^{x_i}, \bigcup_{j,k} SC_{jk}^{y_j, z_k}\}$ is a consistent global state iff

$$\forall m_{ij} : send(m_{ij}) \notin LS_i^{x_i} \Rightarrow m_{ij} \notin SC_{ij}^{x_i, y_j} \wedge rec(m_{ij}) \notin LS_j^{y_j}$$

Global State



$\{LS_1^1, LS_2^3, LS_3^3, LS_4^2\}$ is inconsistent

$\{LS_1^2, LS_2^4, LS_3^4, LS_4^2\}$ is consistent

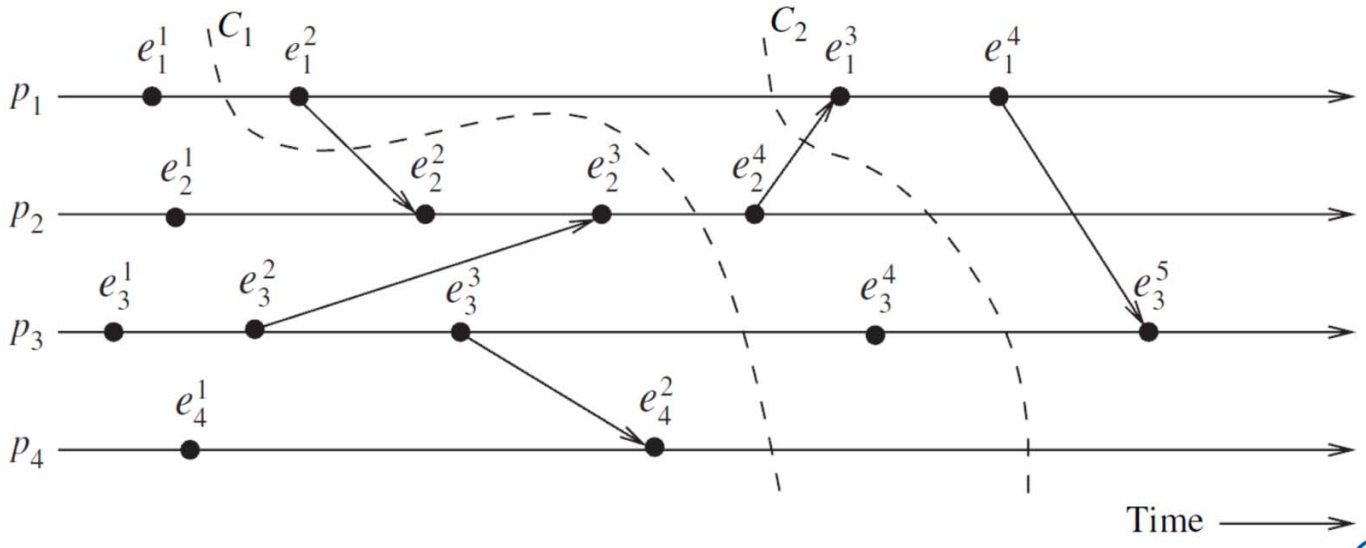
Global State

- A global state $GS = \{\bigcup_i LS_i^{x_i}, \bigcup_{j,k} SC_{jk}^{y_j,z_k}\}$ is transitless iff $\forall i, \forall j : 1 \leq i, j \leq n :: SC_{ij}^{y_i,z_j} = \phi$

- A global state is strongly consistent iff it is transitless as well as consistent

Cuts of a Distributed Computation

- Cut:
  - A zigzag line joining one arbitrary point on each process line in the space-time diagram.
  - PAST: the events to the left of a cut
  - FUTURE: the events to the right of a cut



Consistent Cut

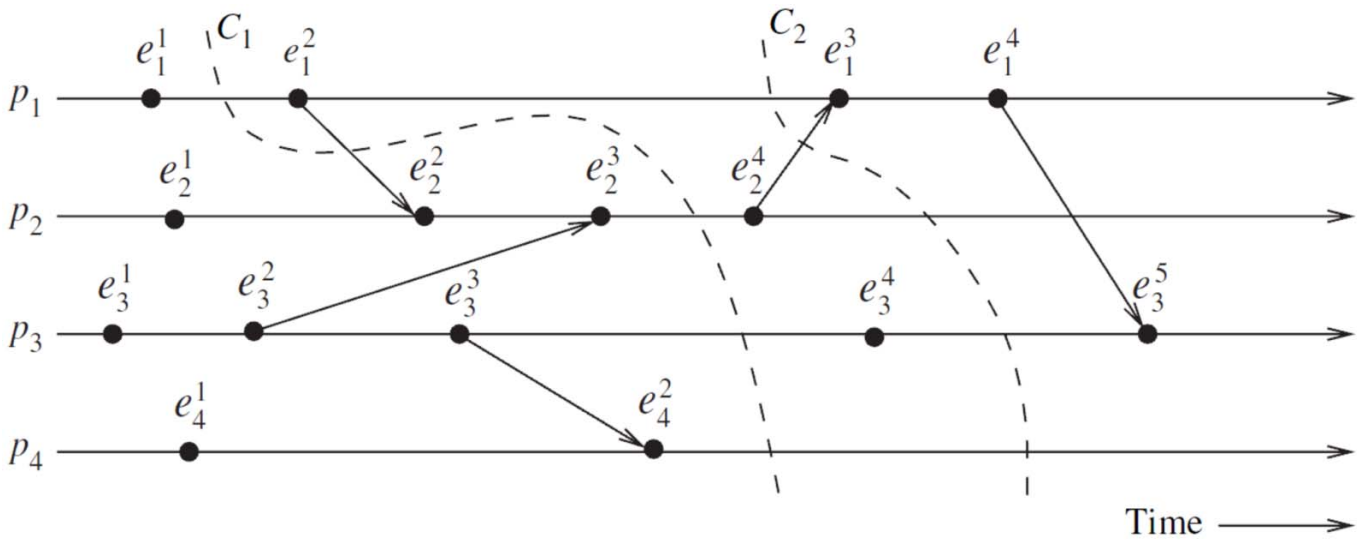
- Let $e_i^{Max\_PASTi(C)}$ be the latest event at process p_i that is in the PAST of a cut C
- The global state represented by a cut C is

$\{\bigcup_i LS_i^{Max\_PAST_i(C)}, \bigcup_{j,k} SC_{jk}^{y_j,z_k}\}$, where

$SC_{jk}^{y_j,z_k} = \{m \mid send(m) \in \mathbf{PAST(C)} \wedge rec(m) \in \mathbf{FUTURE}(C)\}$

- Consistent cut:
  - A cut that will result in a consistent global state
  - Every message received in the past was sent in the past.

Consistent Cut



- Is C_1 a consistent cut?
- Is C_2 a consistent cut?

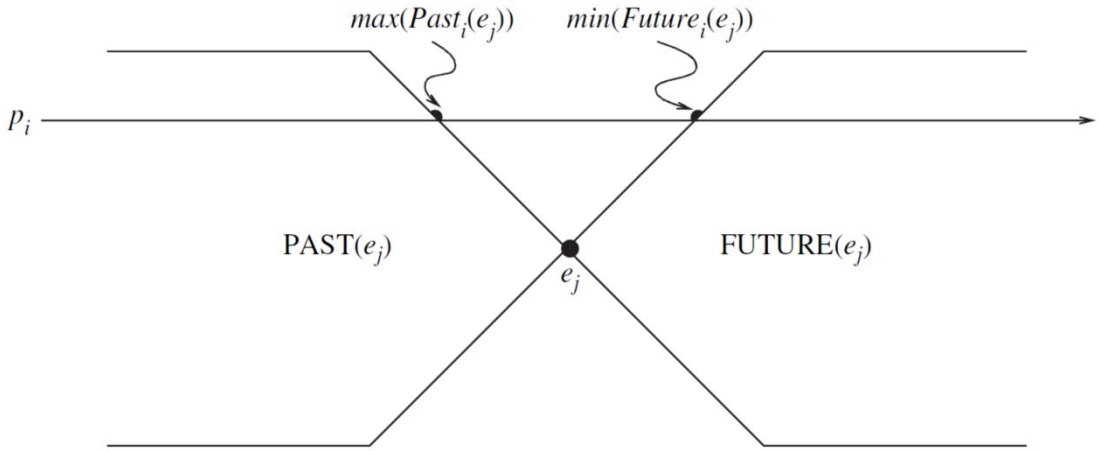
Past and Future Cones of Event

- Past(e_j): be all events in the past of e_j in a computation (H,$\rightarrow$)

$$Past(e_j) = \{e_i | \forall e_i \in H,\ e_i \rightarrow e_j\}$$

- Future(e_j) can be defined similarly.

- Concurrent events with e_j: H – Past(e_j) – Future(e_j)



Past and Future Cones of Event

- $Past_i(e_j)$: $Past(e_j)$ that are on the process P_i
- $max(Past_i(e_j))$: latest event at p_i that affected e_j

$$Max\_Past(e_j) = \bigcup_{(\forall i)} \{max(Past_i(e_j))\}$$

- Surface of the past cone of e_j