Discussion 02

Spring 2019 – CS 188

Section 2B

Logical clocks

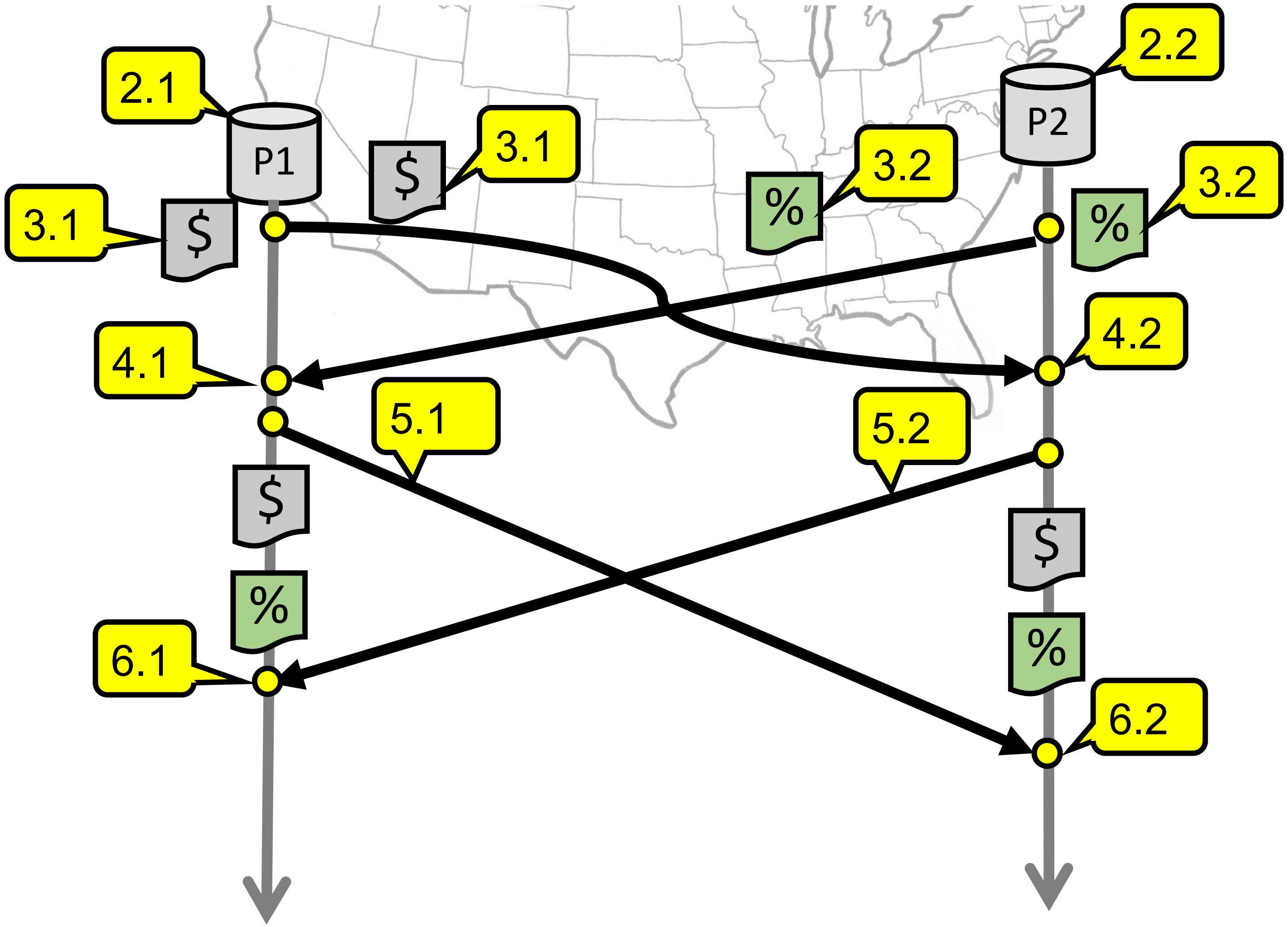
- Lamport clock
  - If a → b, then C(a) < C(b)
  - If a causally leads to b, then C(a) < C(b)
- Vector clock
  - C(a) < C(b) if and only if a → b

The Lamport Clock Rules

1. Each process P_i has their own local clock C_i
2. Before executing an event, $C_i \leftarrow C_i + 1$
3. Send the local clock value in the message m
4. On process P_j receiving message m:

   set C_j and time of receive event to $1 + \max\{C_j, C(m)\}$

Example 1: RSM with Lamport Clocks



P2 can execute % before receiving ack from P1, since anything that comes after is greater than or equal to 4.1

The Vector Clock Rules

- Initially, all vectors start at [0, 0,…, 0]

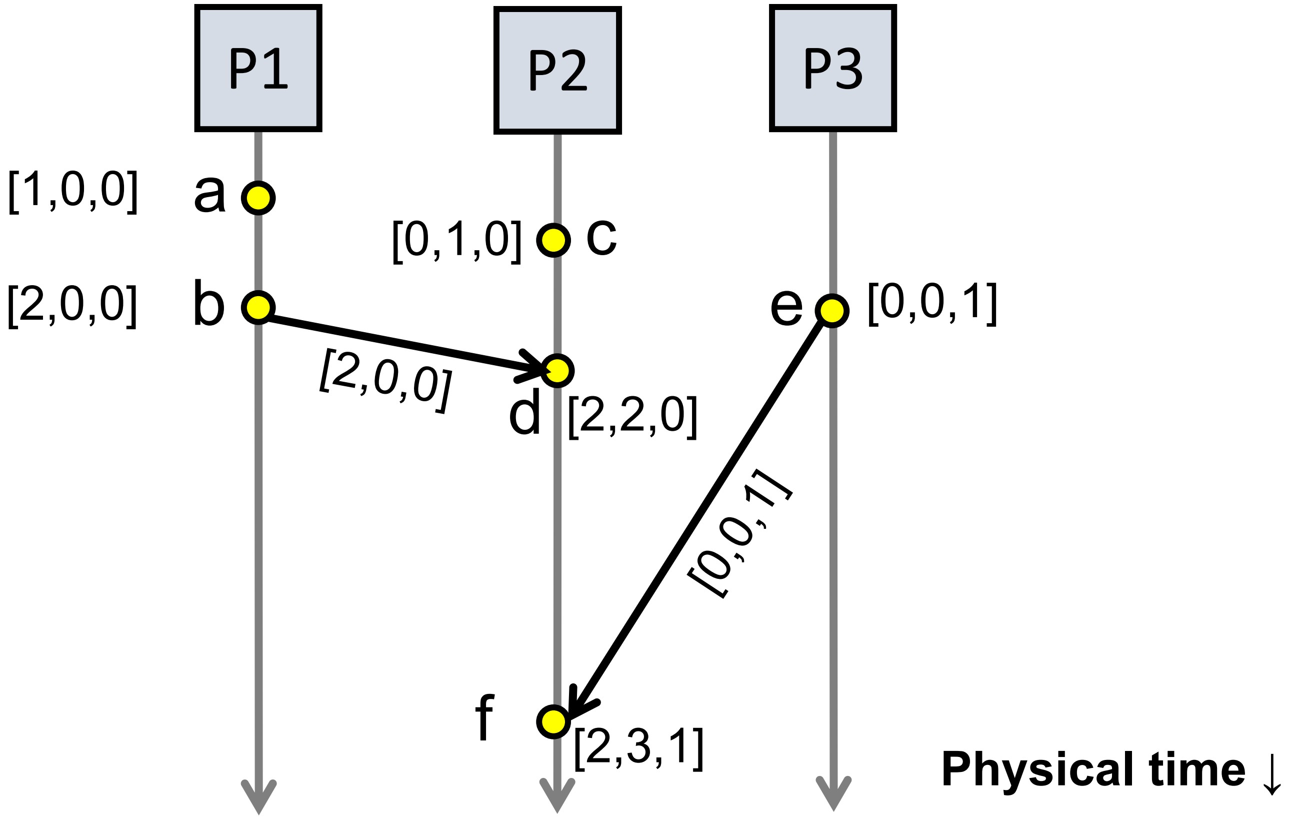
- Two rules for updating a vector
  1. For each local event on process i, increment local entry c_i in vector by 1
  2. If process j receives message with vector $[d_1, d_2, \ldots, d_n]$:
     - Set each local entry $c_k = \max\{c_k, d_k\}$
     - Increment local entry c_j in vector by 1

Example 2: Vector Clocks

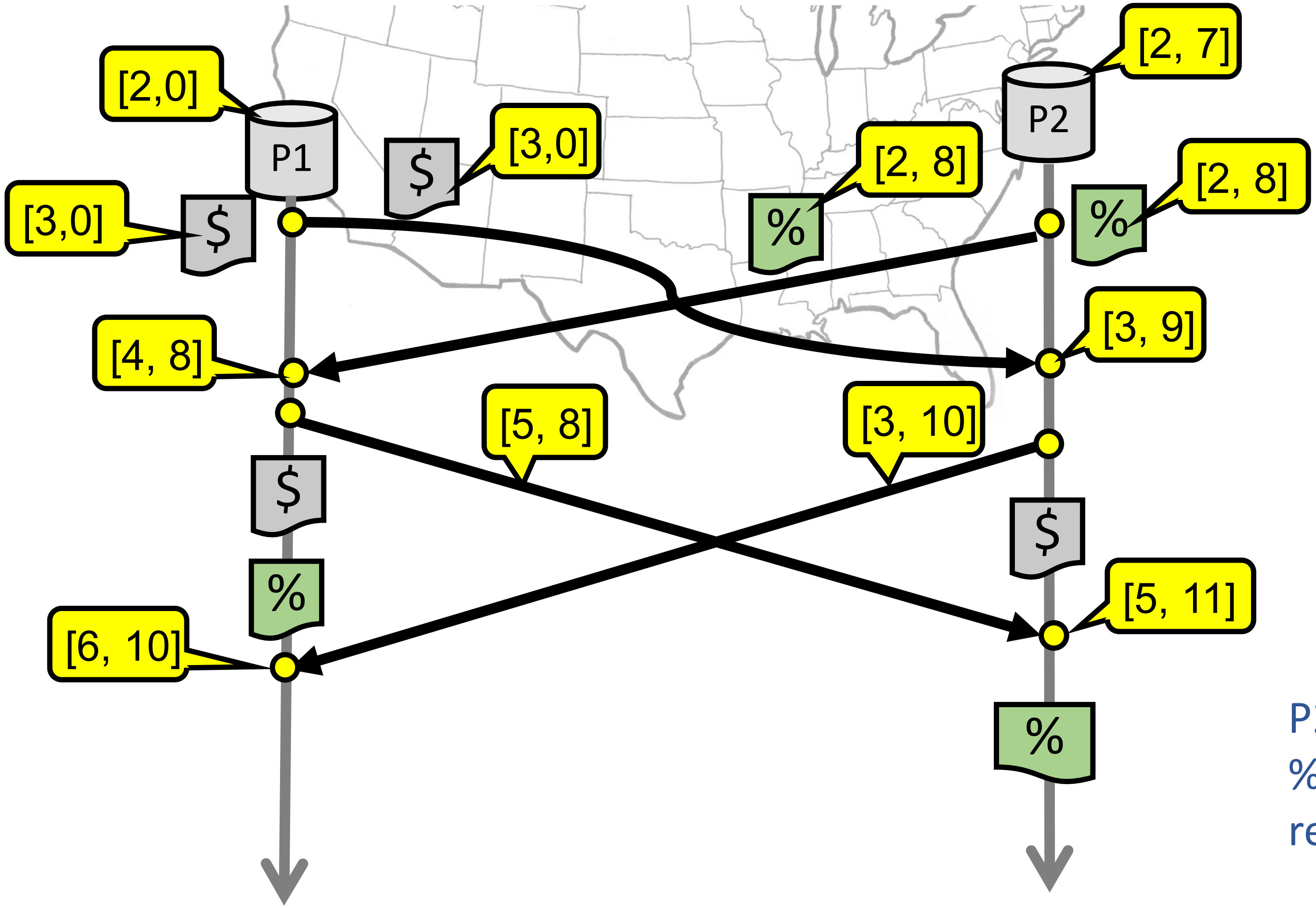
- All vectors start at [0, 0, 0]

- Applying local update rule

- Applying message rule



Example 3: RSM with Vector Clocks



P2 cannot execute % before [5, 11] is received.

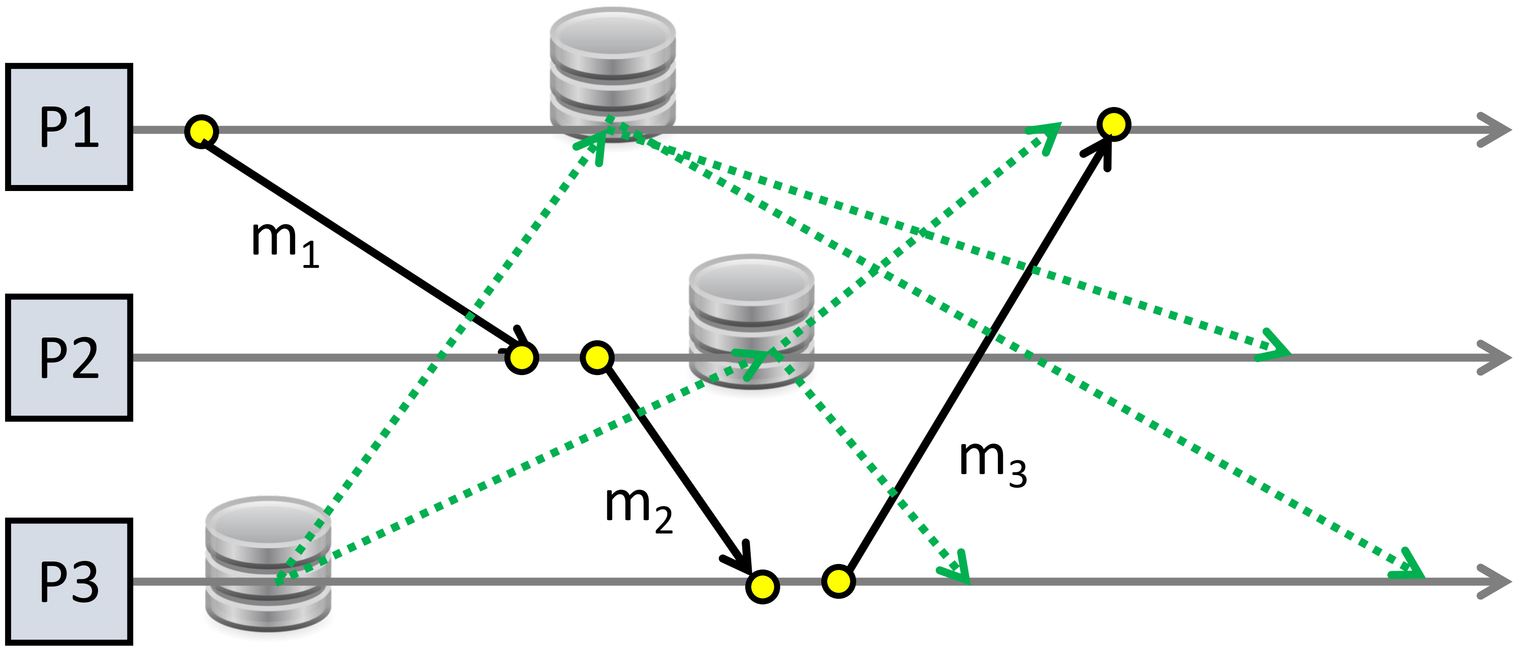
Recovering from crash failures

- Solution?
  - Checkpointing
    - Problem? Domino effect
    - Solution? Coordinated Checkpointing
      - Chandy-Lamport Snapshot

Chandy-Lamport Snapshot Rules

- If it is the first marker message received by P_i:
  - Record all local state (take checkpoint)
  - Record state of channel from P_j to P_i as empty
  - Send marker messages on all outgoing channels
- If duplicate marker message received by P_i:
  - Stop recording channel from P_j to P_i
  - Record state of channel as all messages received since marker
- Snapshot complete when every process has received marker on every incoming channel

Example 4: Chandy-Lamport example


P1
P2
P3
m_1
m_2
m_3