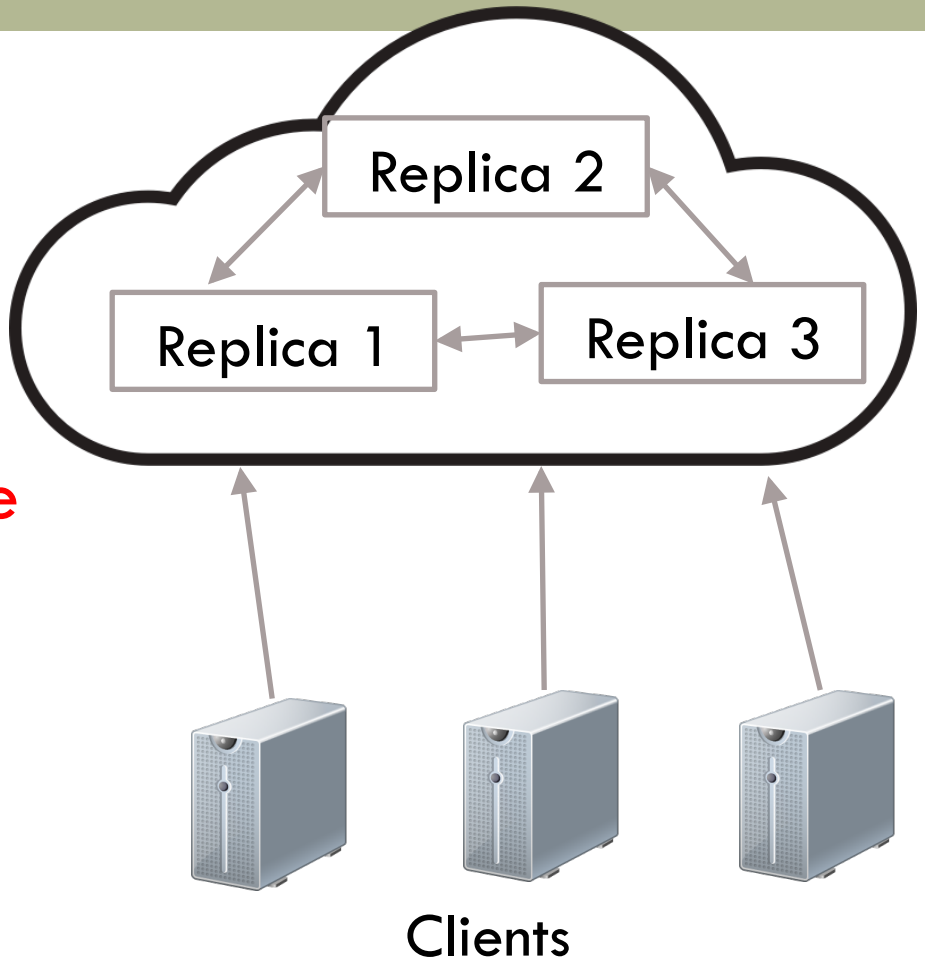
LECTURE 6

Distributed Snapshotting/Checkpointing

Replicated State Machine



Are we done now that we have logical clocks?

Failures!

RSMs with Logical Clocks

- Any replica can execute an update only after confirming clock is higher on all other replicas

- Implication: If any one replica is down, all other replicas cannot progress

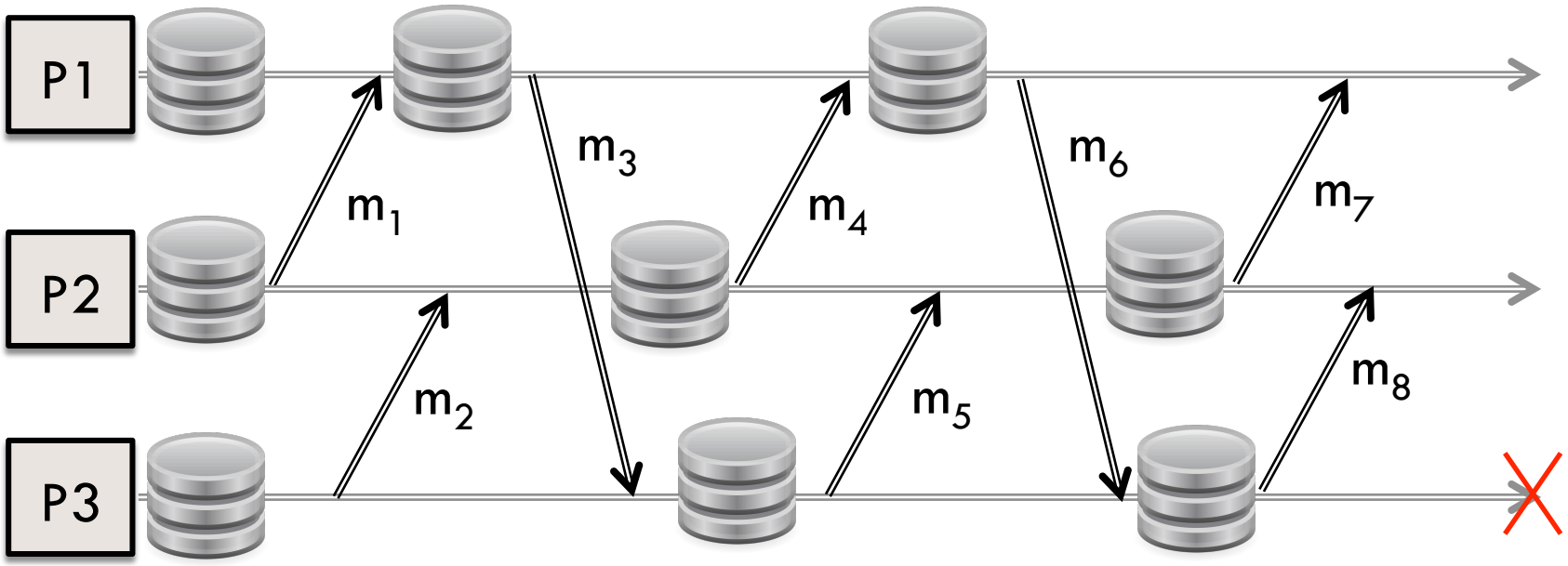
Types of failures

- Crash failures
  - Can resume with saved state
- Fail stop
  - All state is lost upon failure

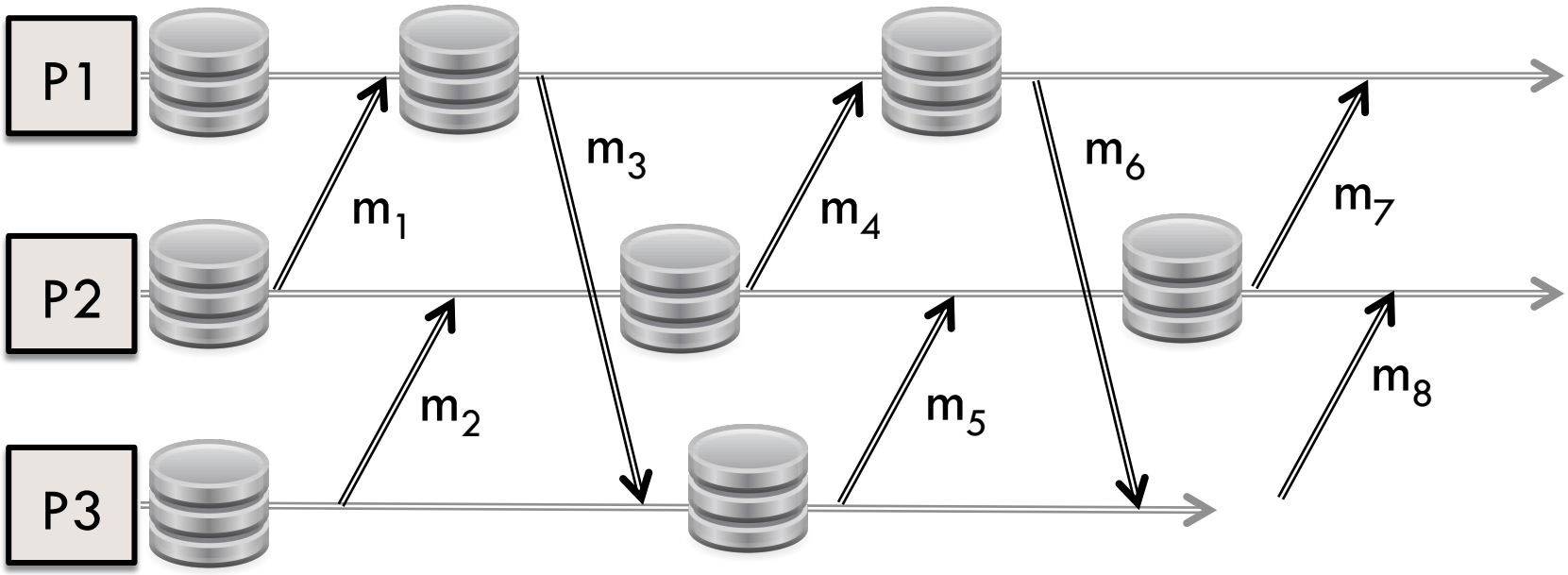
Recovering from failures

- Checkpoint process state periodically
  - Where to store checkpoint?
  - Persistent storage vs. volatile memory
  - Local machine vs. remote machine
- Where you store checkpoint depends on
  - Types of failures you want to tolerate
  - Performance overhead you are willing to bear
- Resume from last checkpoint after restart

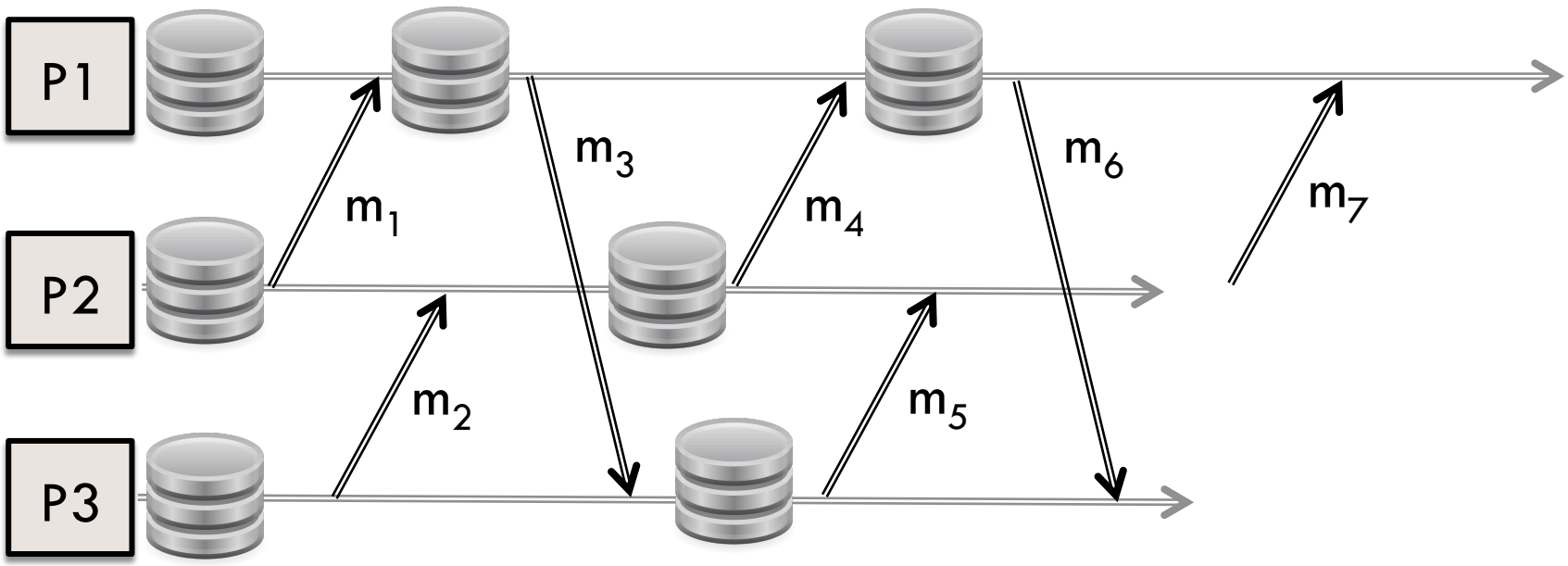
Challenge in Checkpointing



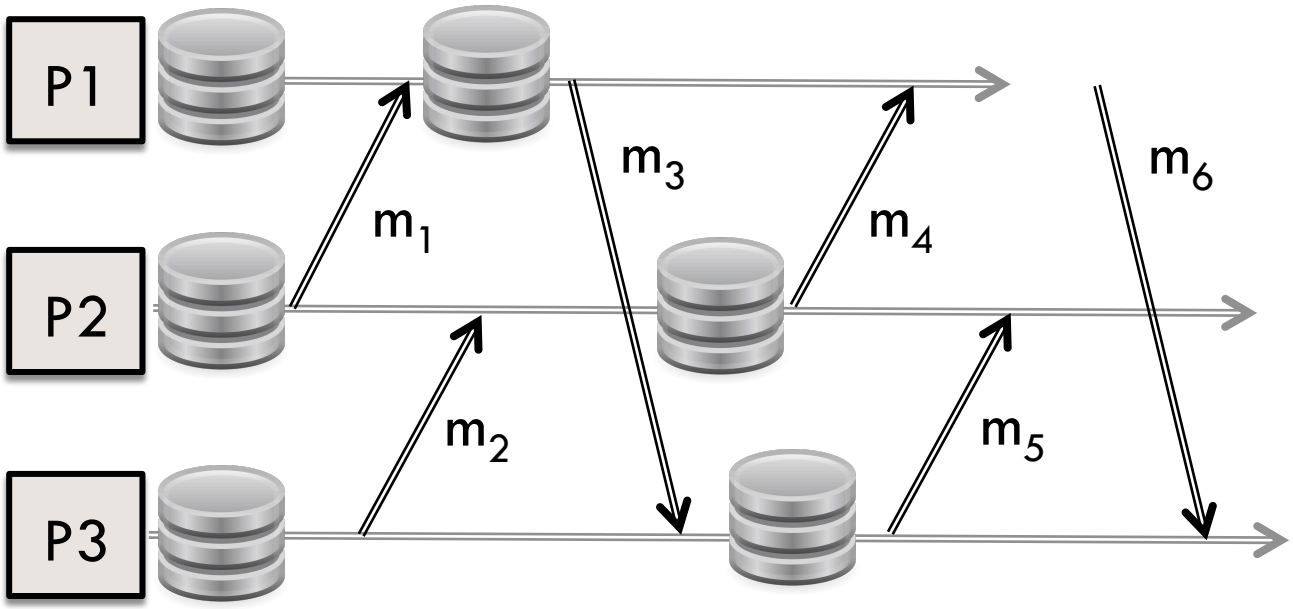
Challenge in Checkpointing



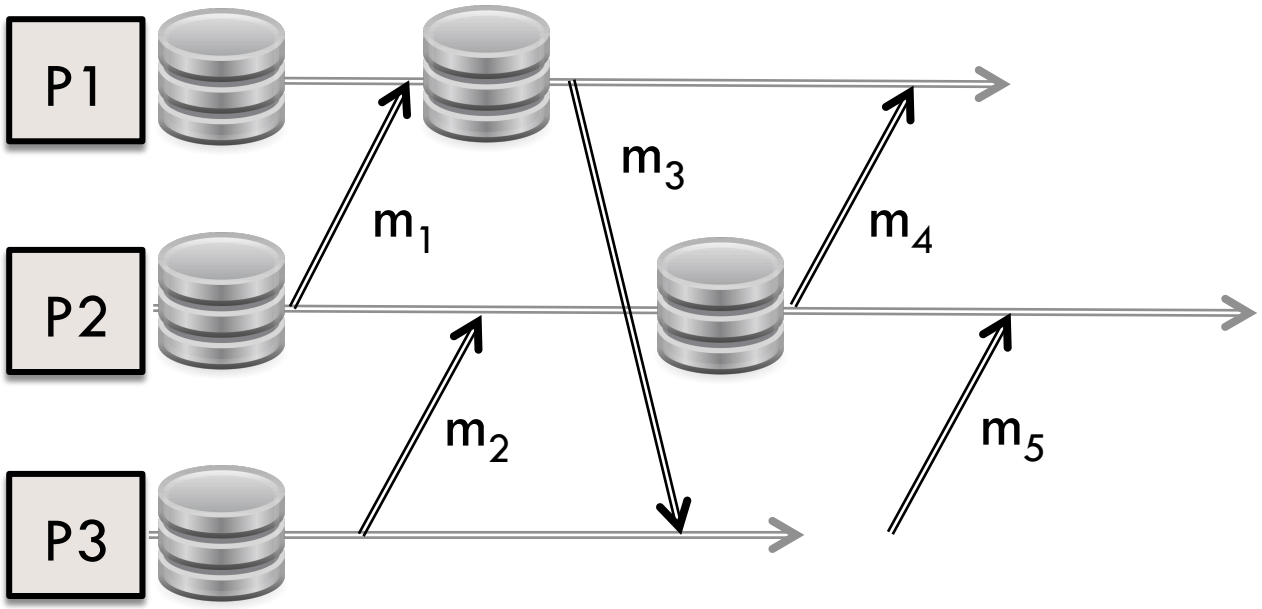
Challenge in Checkpointing



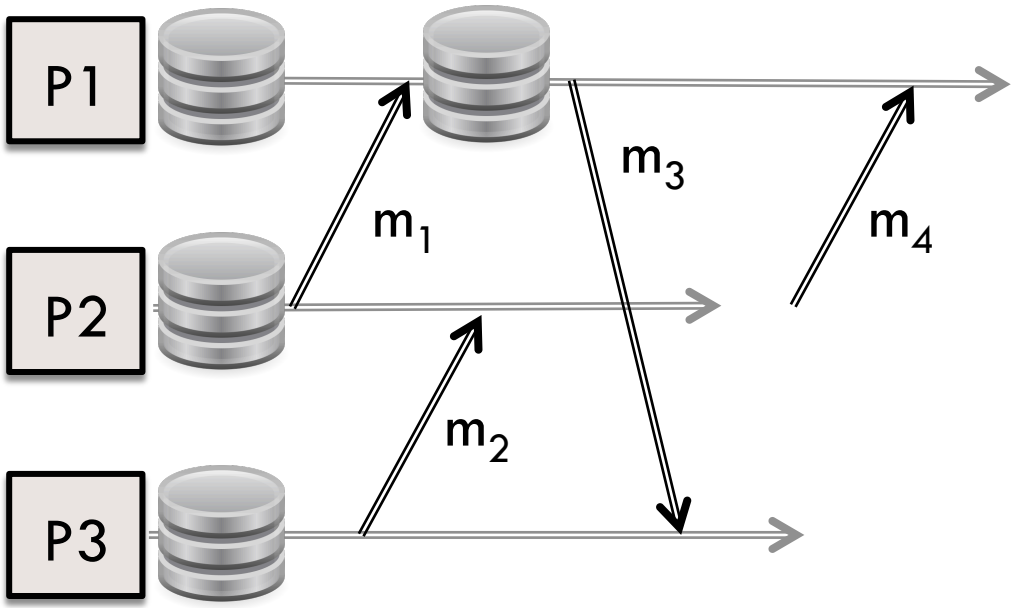
Challenge in Checkpointing



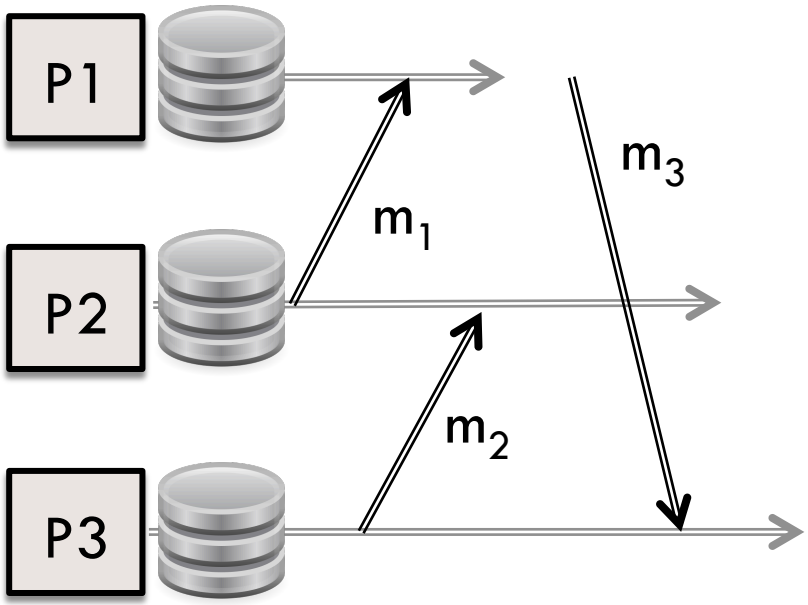
Challenge in Checkpointing



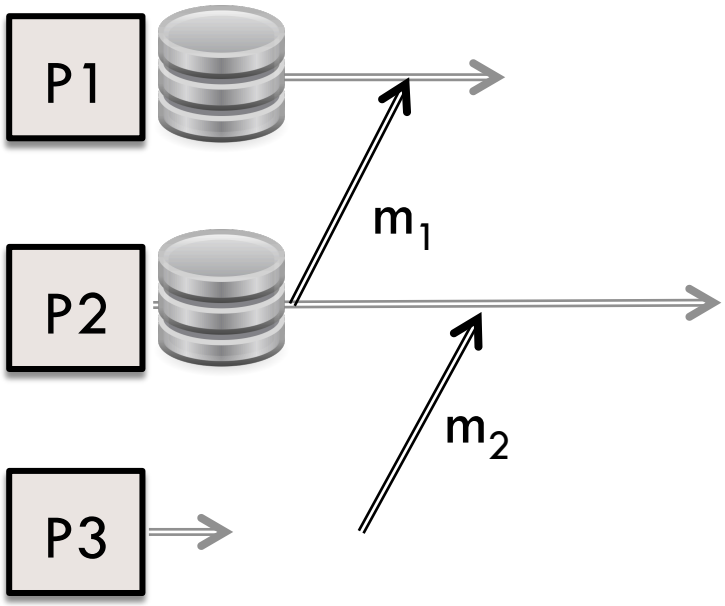
Challenge in Checkpointing



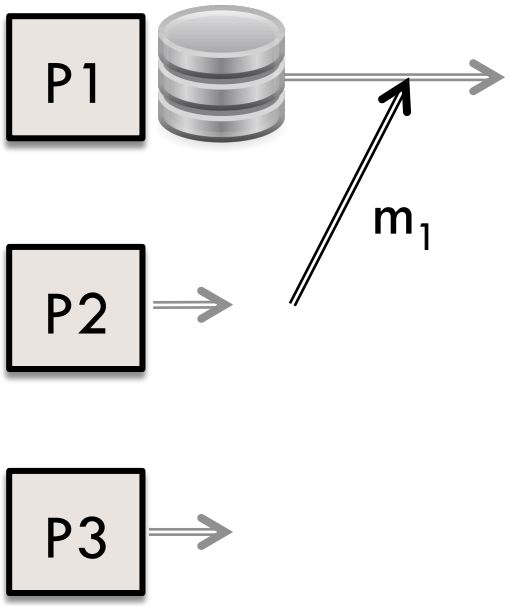
Challenge in Checkpointing



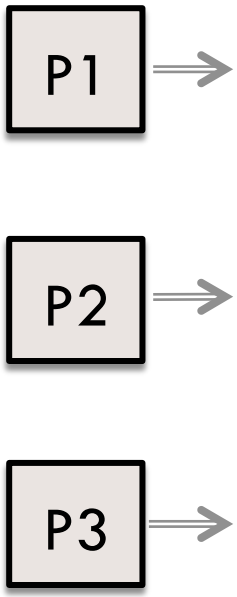
Challenge in Checkpointing



Challenge in Checkpointing



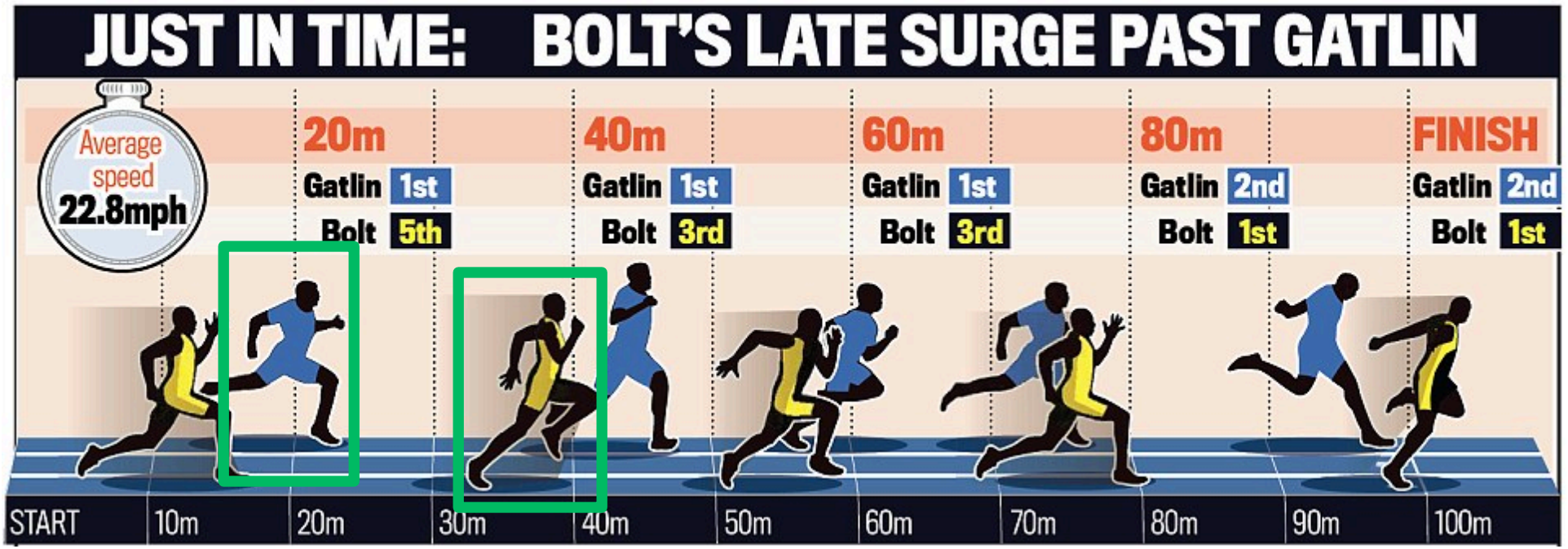
Challenge in Checkpointing



Checkpoint-based Recovery

- Problem: Domino effect
  - Cause: Uncoordinated checkpointing
- Coordinated checkpointing
  - Example: Chandy-Lamport snapshot
    - Distributed Snapshots: Determining Global States of Distributed Systems: ACM Transactions on Computer Systems 1985
  - Global snapshot of a distributed system

Example

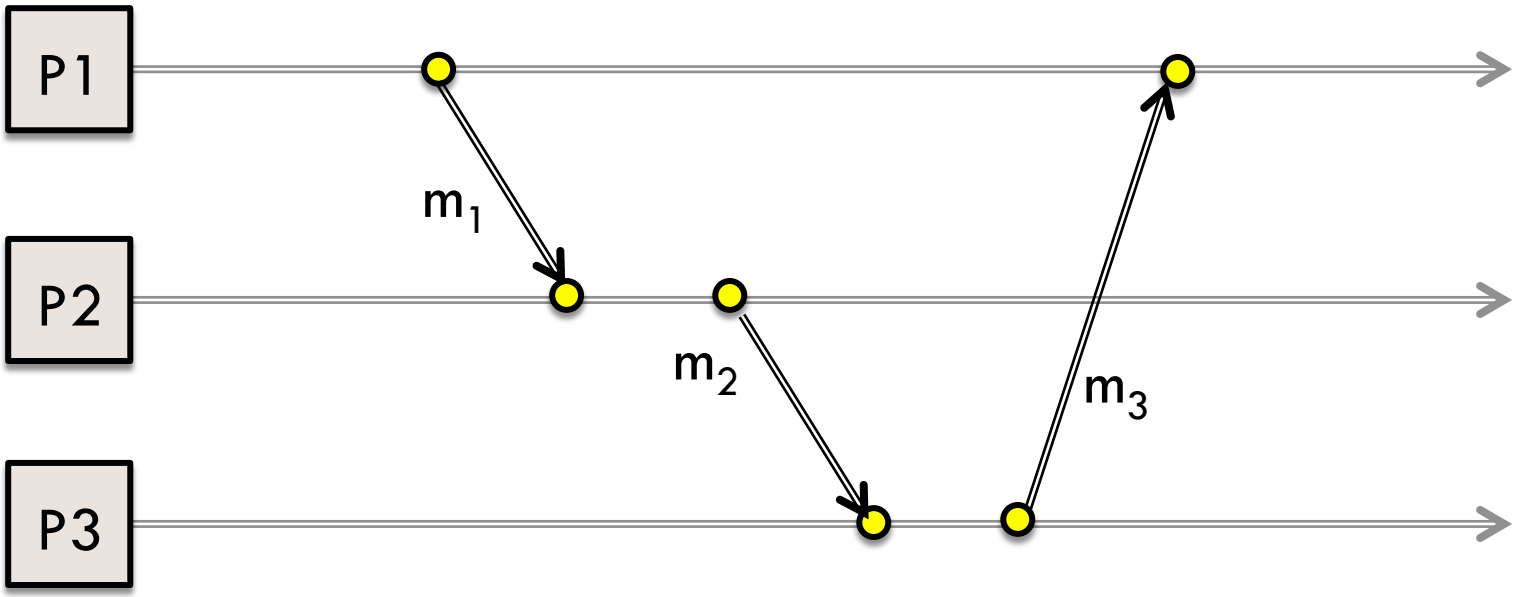

JUST IN TIME: BOLT'S LATE SURGE PAST GATLIN
Average speed 22.8mph
20m
Gatlin 1st
Bolt 5th
40m
Gatlin 1st
Bolt 3rd
60m
Gatlin 1st
Bolt 3rd
80m
Gatlin 2nd
Bolt 1st
FINISH
Gatlin 2nd
Bolt 1st
START
10m
20m
30m
40m
50m
60m
70m
80m
90m
100m

Chandy-Lamport algorithm

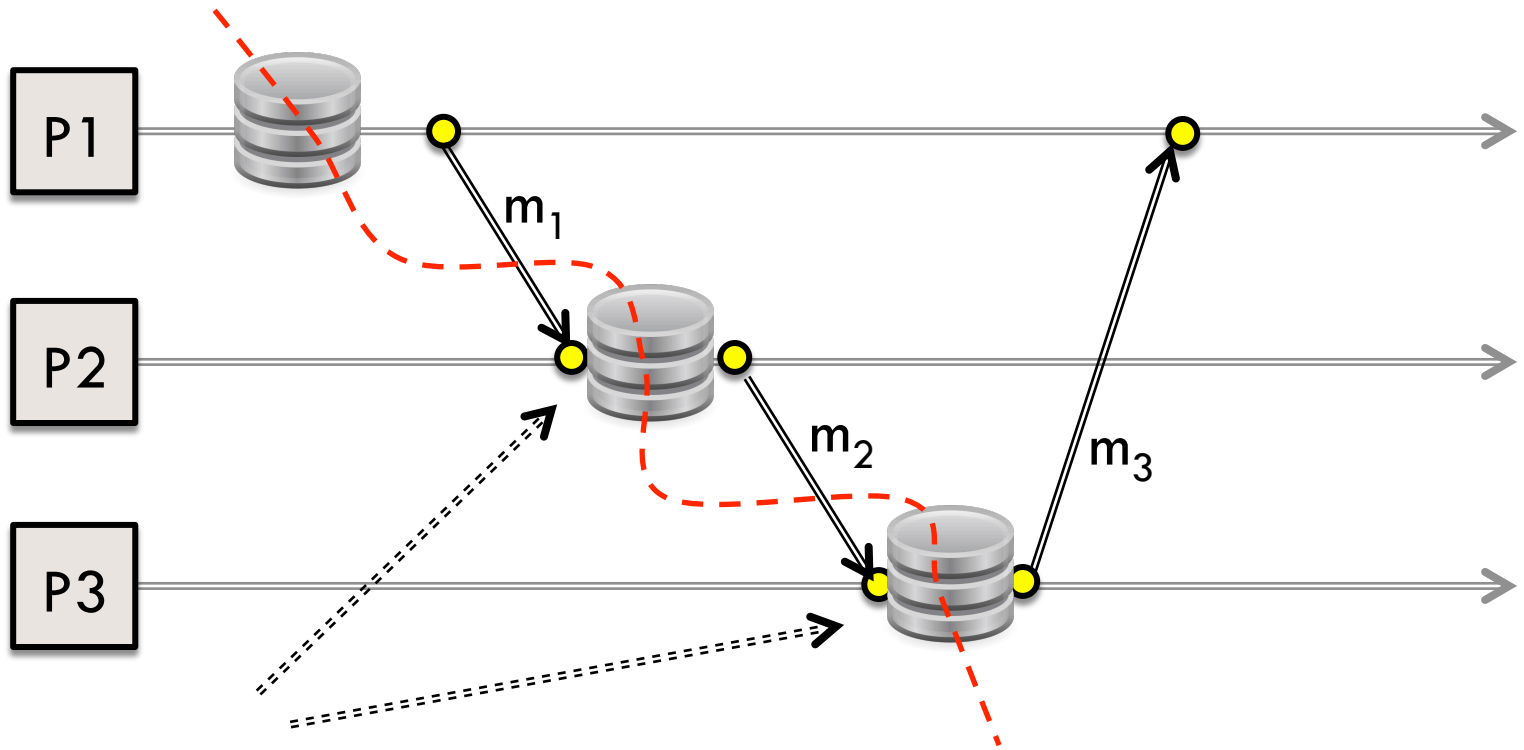
- How to take a snapshot of a distributed system?

- Example use cases:
  - Deadlock detection
  - Garbage collection
  - Evaluation of any stable property

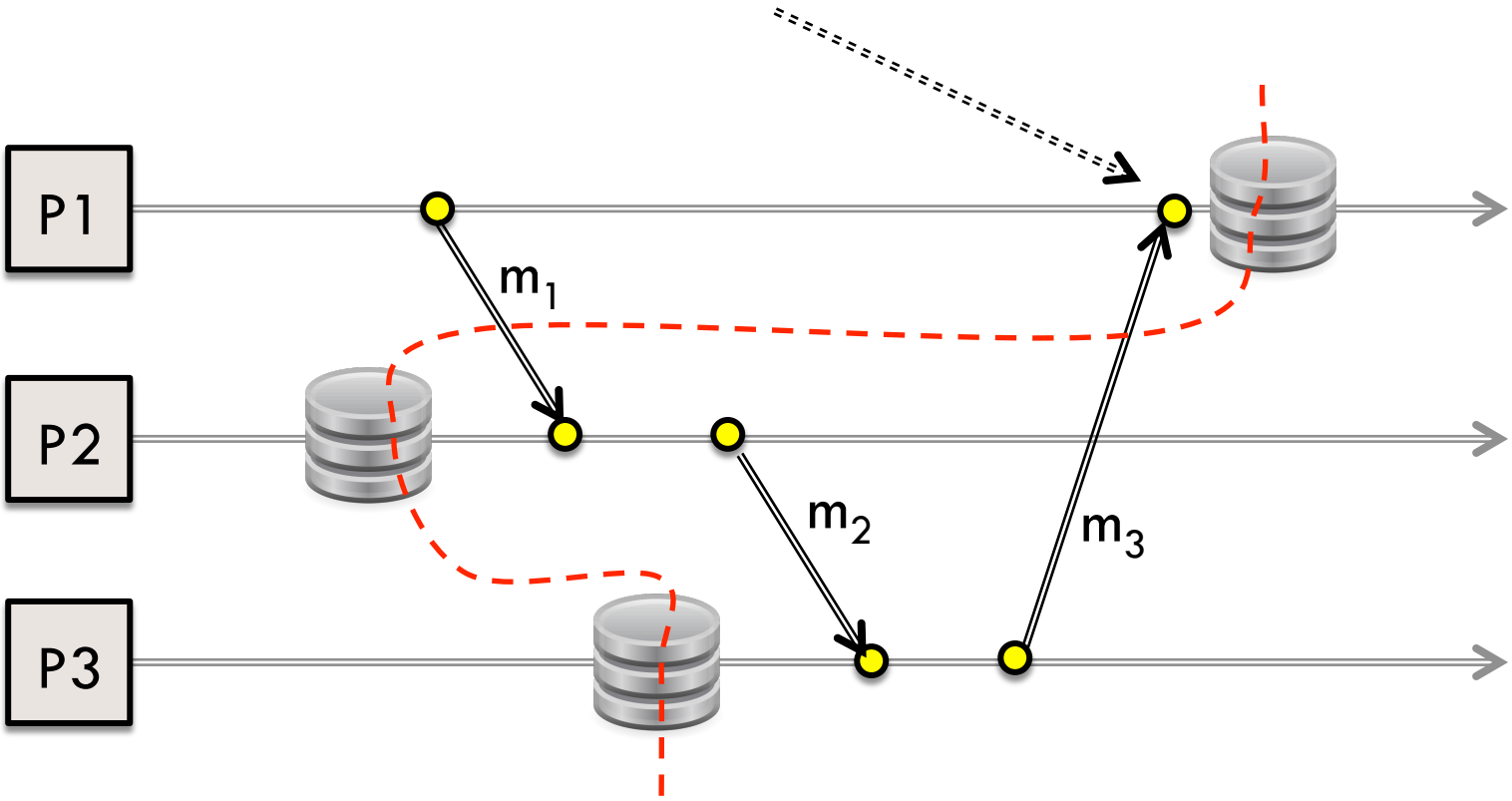
Example: Token ring



Example snapshot 1



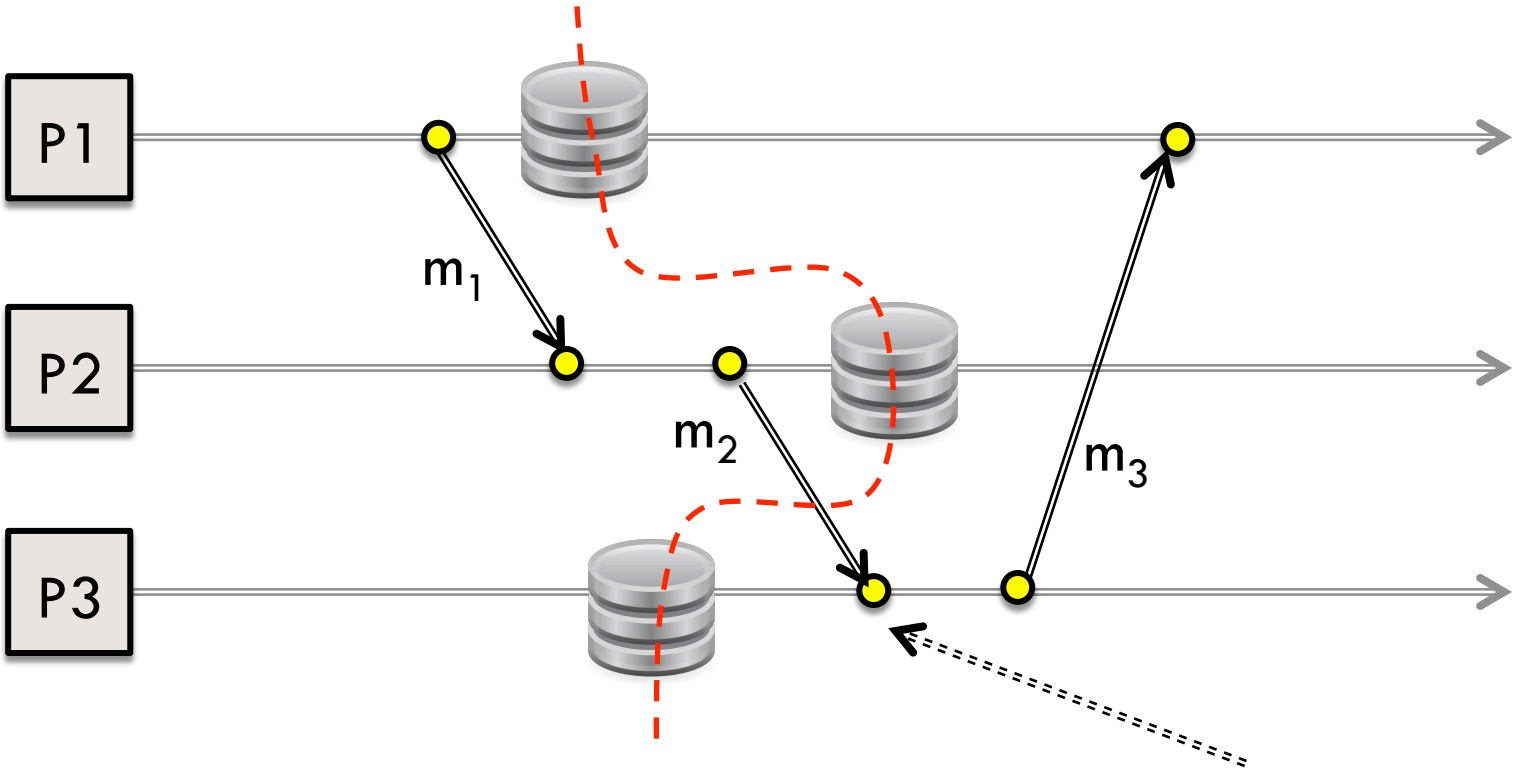
Example snapshot 2



Global snapshot

- Captured state must satisfy "happens before"
- If event **b** in snapshot and **a** → **b**, then event **a** must be in snapshot

Example desired snapshot



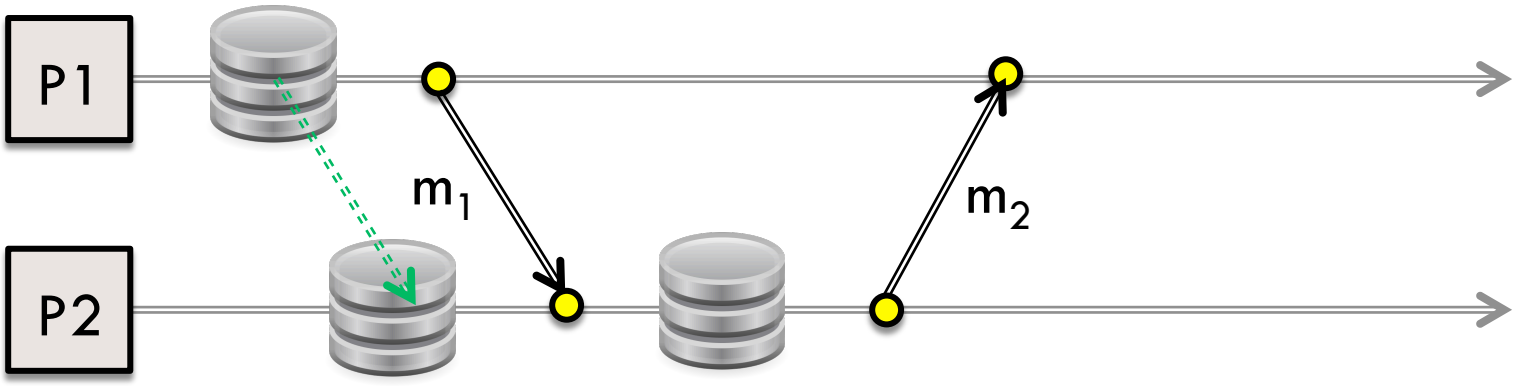
Global snapshot

- Goals of capturing global snapshot
  - Capture instantaneous state of every process
  - Capture relevant messages in transit
- What comprises a distributed system's state?
  - State of individual processes
  - State of every communication channel

Chandy-Lamport Snapshot

- N processes in the system
  - Processes don't fail while taking snapshot
  - Any process may initiate collection of snapshot
- One unidirectional channel in either direction between each pair of processes
  - All channels ensure FIFO delivery
  - Lossless and no duplication

Chandy-Lamport Snapshot



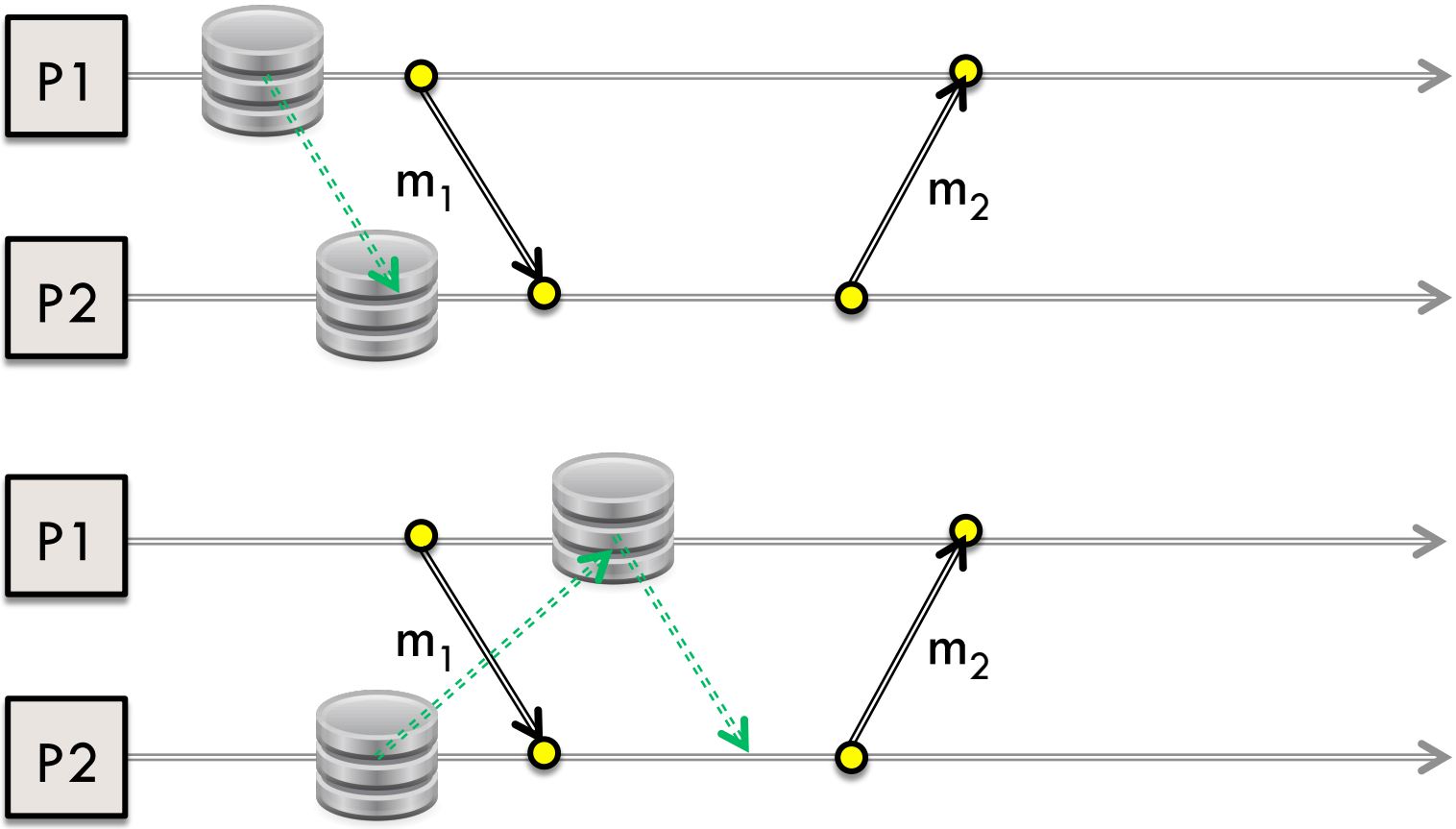
Chandy-Lamport: Intuition

- Enable snapshot by broadcasting *marker* message
  - Distinct from the application's messages

- Marker messages serve two purposes:

1. Enable processes to discover need for snapshot
2. Serve as a barrier on every channel
   - Record all messages received before marker

Chandy-Lamport Snapshot



Chandy-Lamport Snapshot

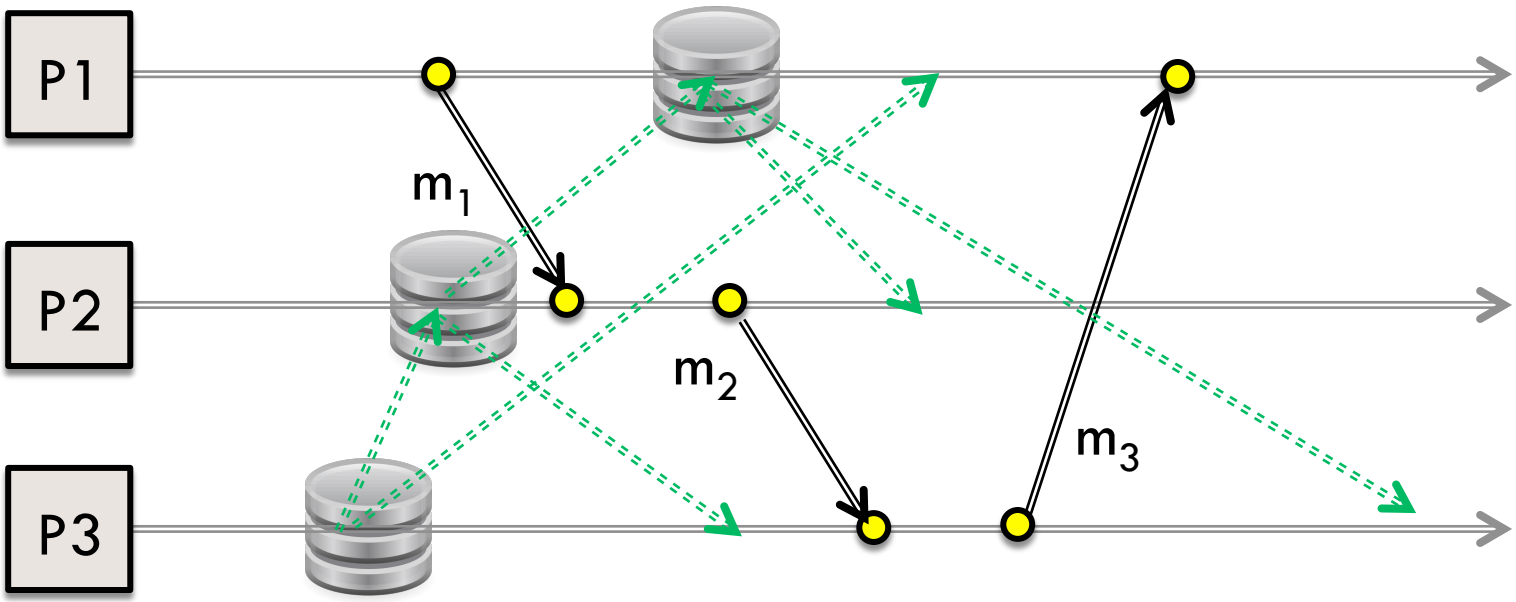
- Initiator process does the following:
  - Records its local state
  - Sends out *marker* messages on every outgoing channel
  - Starts recording messages on every incoming channel

- Two cases for how P_i should handle receipt of marker message from P_j

Chandy-Lamport Snapshot

- If the state has not been recorded by P_i upon the receipt of a marker along a channel c
  - Record local state
  - Record state of channel from P_j to P_i as empty
  - Send marker messages on all outgoing channels
- If state was previously recorded
  - 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

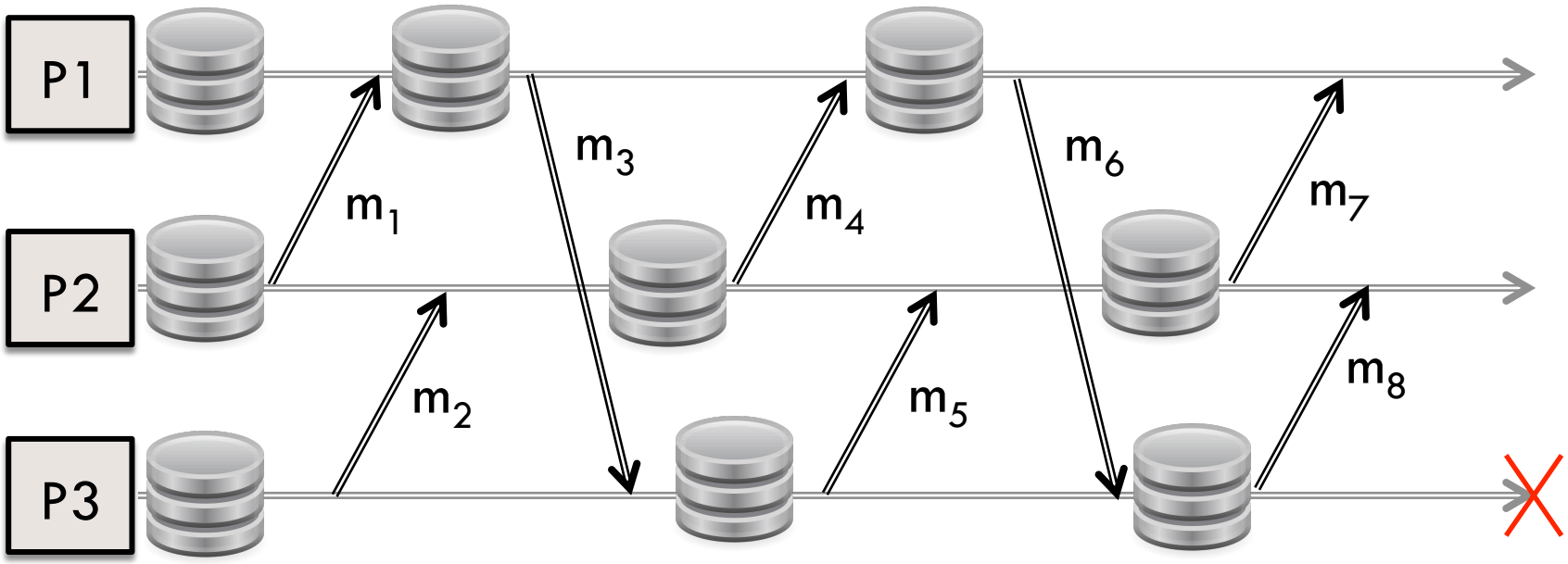
Chandy-Lamport in action



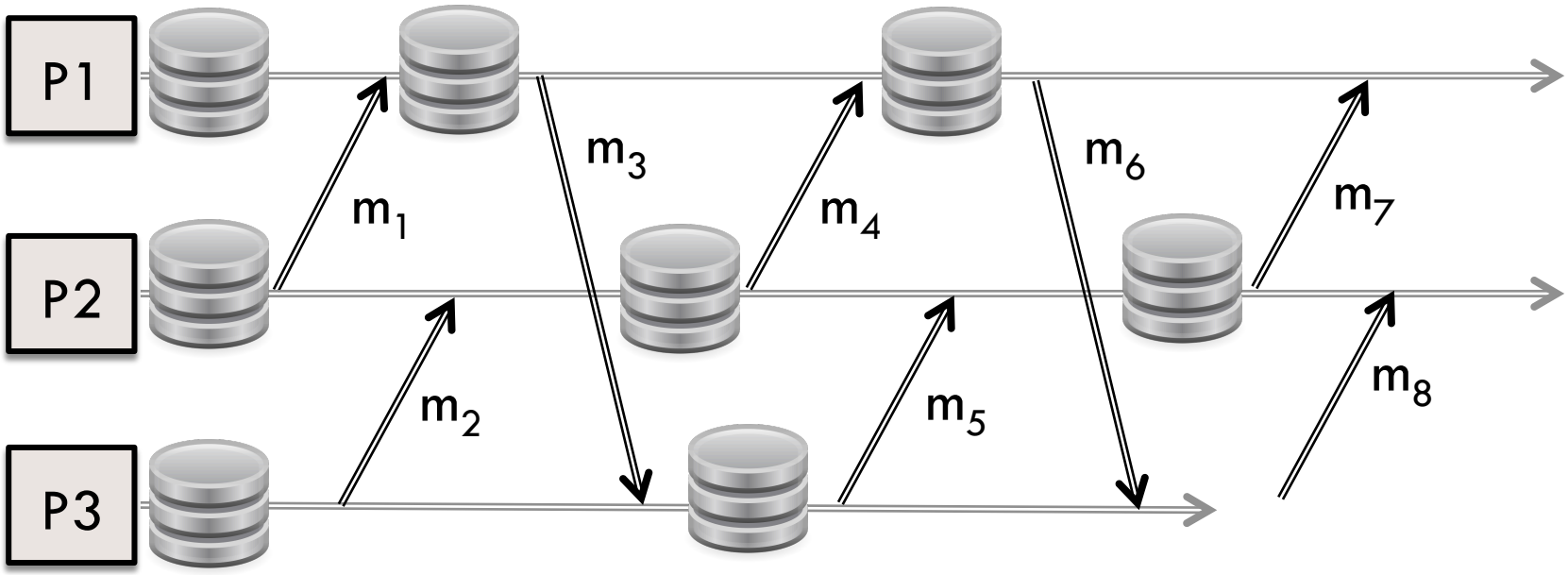
Coordinated checkpointing

- When to initiate global snapshot?
- External environment does not checkpoint
  - Cannot roll back output
  - Cannot re-issue inputs
- Implication: Need coordinated checkpoint upon input or output
  - Too slow!

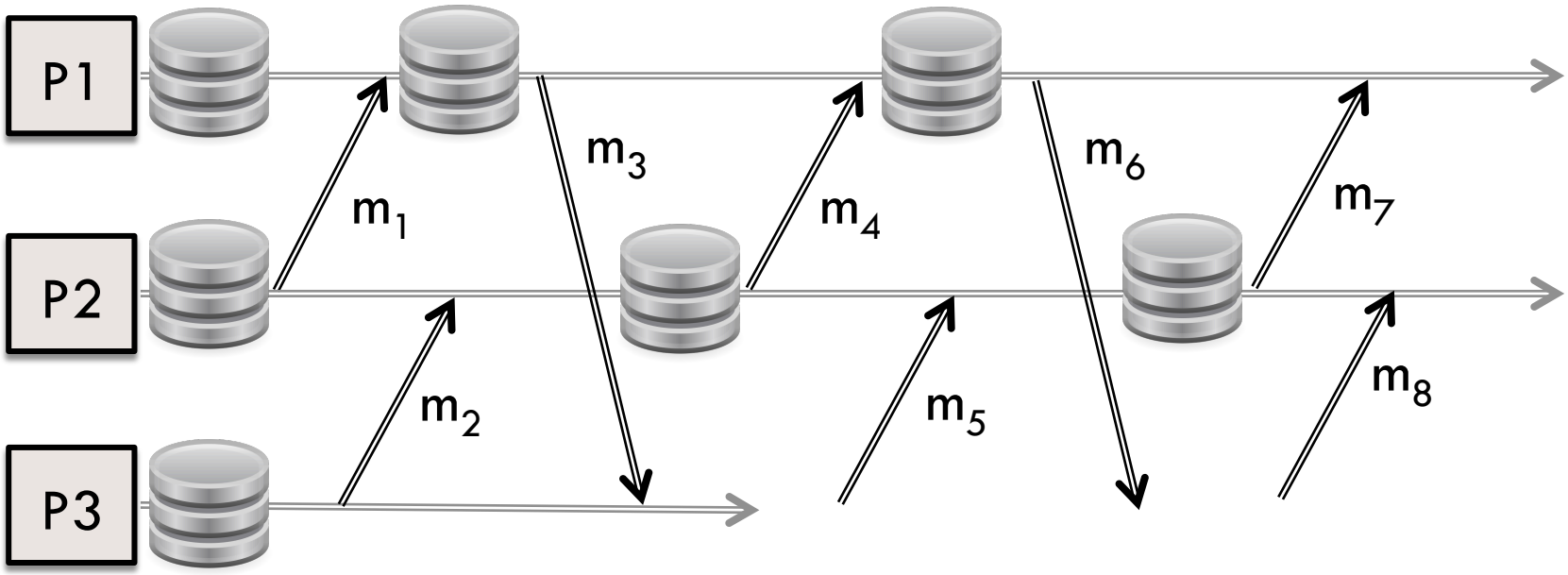
Challenge in Checkpointing



Challenge in Checkpointing



Challenge in Checkpointing



Message logging

- Between checkpoints, log all events that lead to non-determinism
- For example, log every message received
- Log everything necessary to ensure same messages are sent as in original execution