Time Warp

Local Control

Richard M. Fujimoto
Professor

Computational Science and Engineering Division
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0765, USA

http://www.cc.gatech.edu/~fujimoto/

The Synchronization Problem

Local causality constraint: Events within each logical process must be processed in time stamp order

Observation: Adherence to the local causality constraint is sufficient to ensure that the parallel simulation will produce exactly the same results as the corresponding sequential simulation*

Synchronization Algorithms

- Conservative synchronization: avoid violating the local causality constraint (wait until it's safe)
  - 1st generation: null messages (Chandy/Misra/Bryant)
  - 2nd generation: time stamp of next event
- Optimistic synchronization: allow violations of local causality to occur, but detect them at runtime and recover using a rollback mechanism
  - Time Warp (Jefferson)
  - approaches limiting amount of optimistic execution

* provided events with the same time stamp are processed in the same order as in the sequential execution

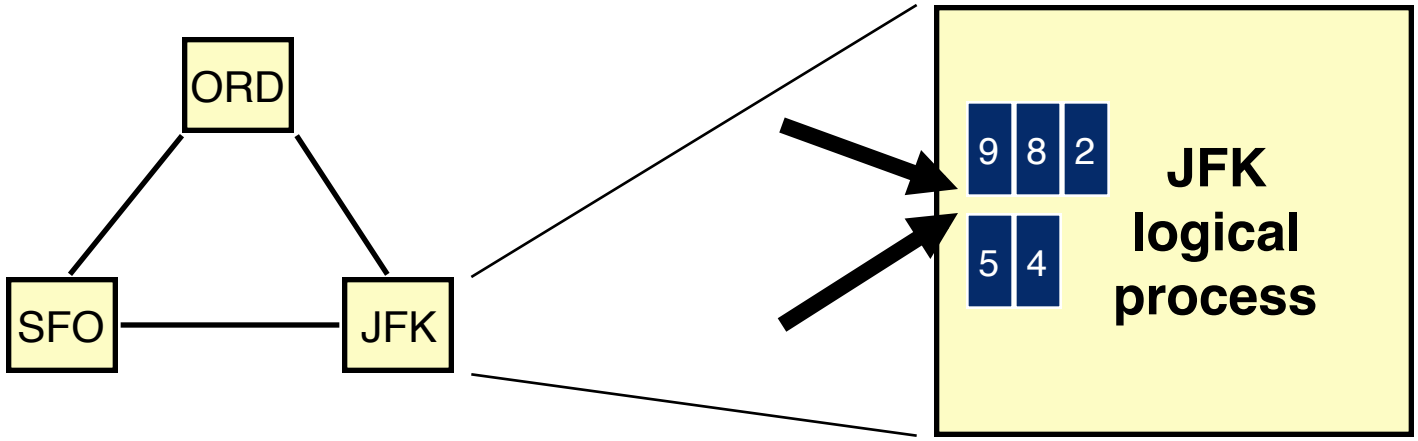
Time Warp Algorithm (Jefferson)

Assumptions

- logical processes (LPs) exchanging time stamped events (messages)
- dynamic network topology, dynamic creation of LPs OK
- messages sent on each link need not be sent in time stamp order
- network provides reliable delivery, but need not preserve order

Basic idea:

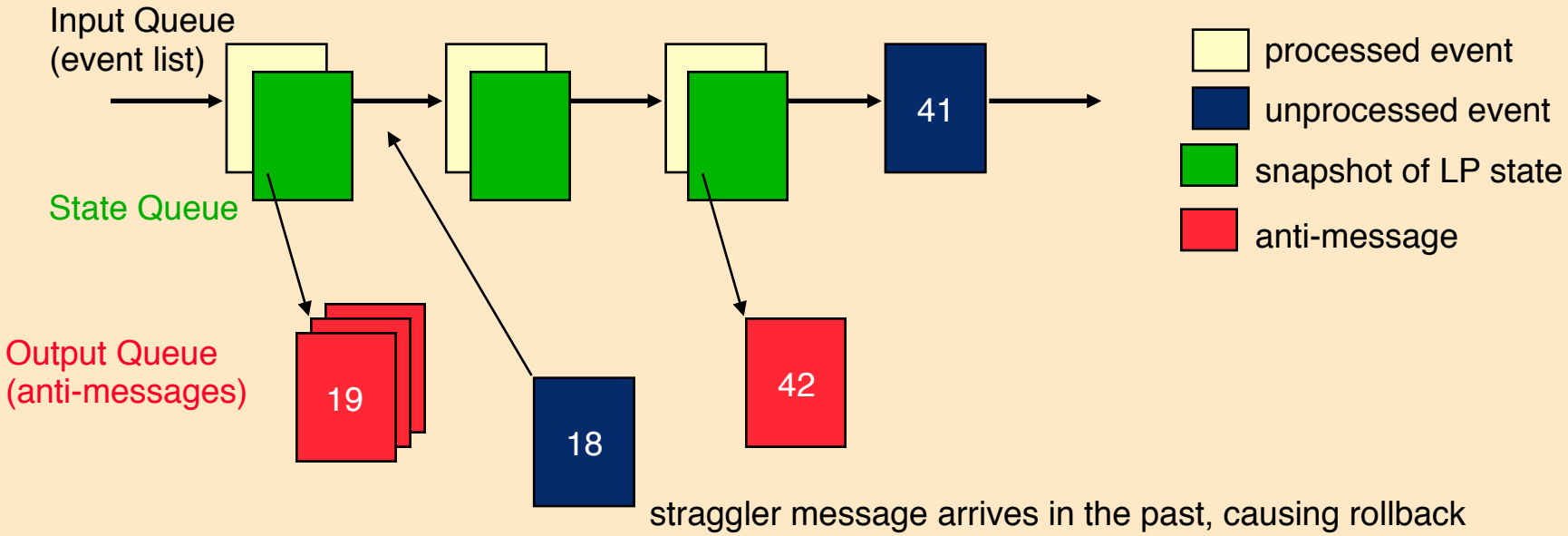
- process events w/o worrying about messages that will arrive later
- detect out of order execution, recover using rollback



process all available events (2, 4, 5, 8, 9) in time stamp order

Time Warp: Local Control Mechanism

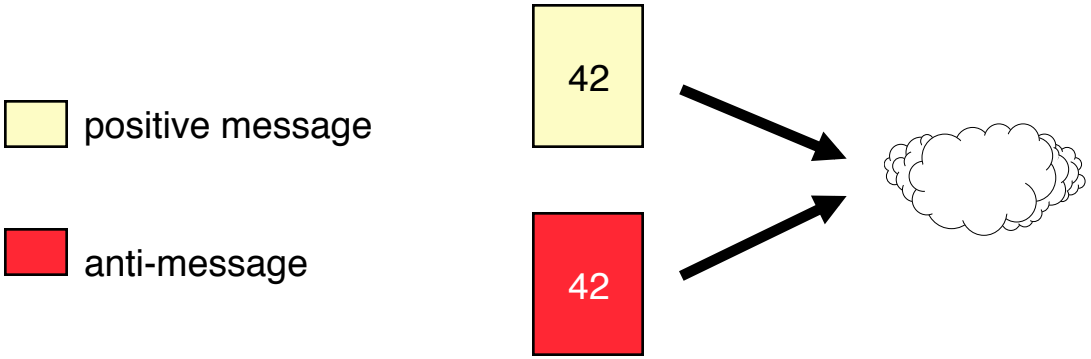
Each LP: process events in time stamp order, like a sequential simulator, except:
(1) do NOT discard processed events and (2) add a rollback mechanism



Adding rollback:

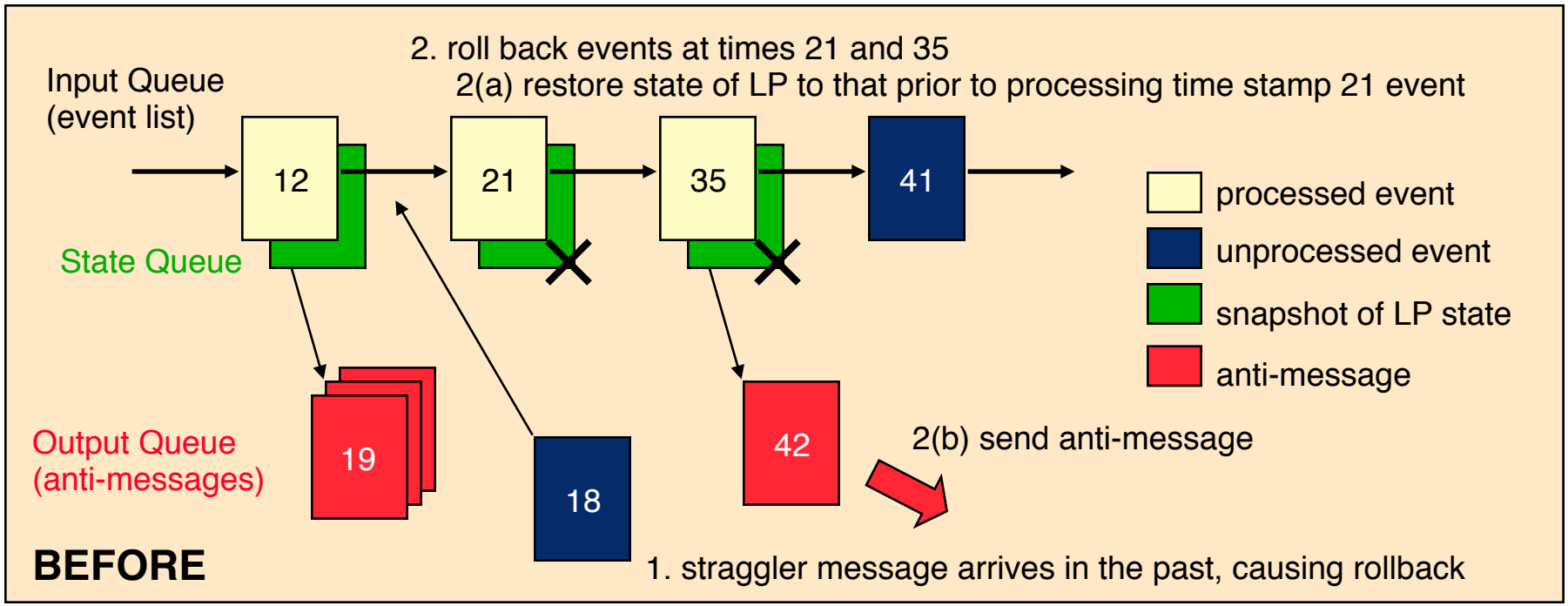
- a message arriving in the LP's past initiates rollback
- to roll back an event computation we must undo:
  - changes to state variables performed by the event;
    solution: checkpoint state or use incremental state saving (state queue)
  - message sends
    solution: anti-messages and message annihilation (output queue)

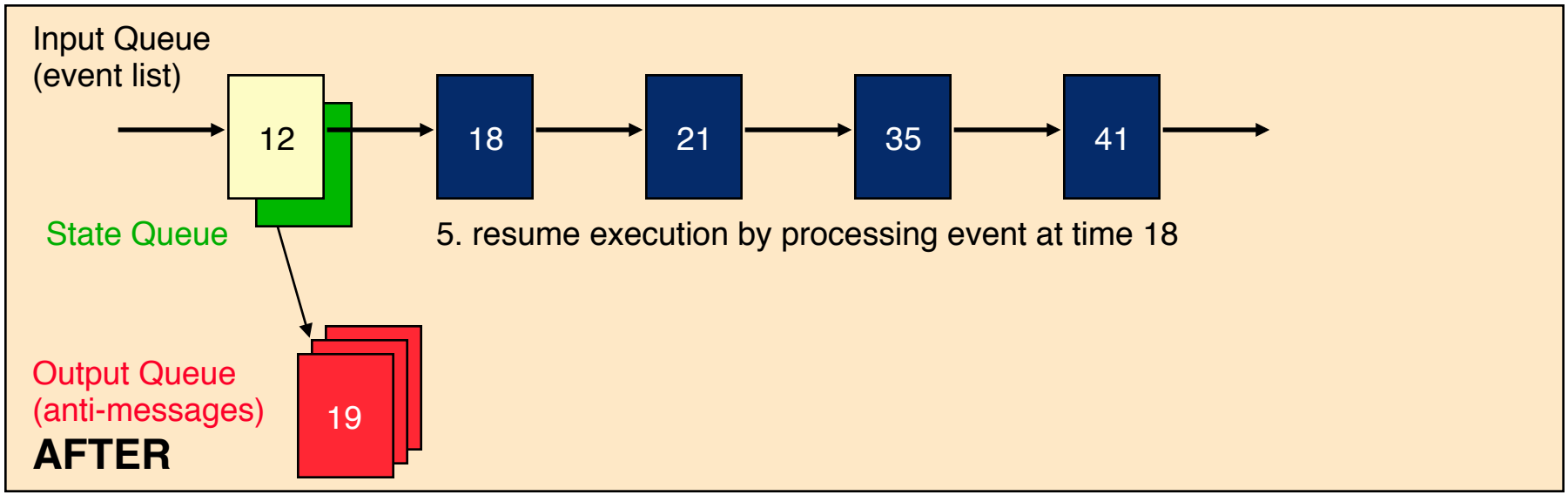
Anti-Messages



- Used to cancel a previously sent message
- Each positive message sent by an LP has a corresponding anti-message
- Anti-message is identical to positive message, except for a sign bit
- When an anti-message and its matching positive message meet in the same queue, the two annihilate each other (analogous to matter and anti-matter)
- To undo the effects of a previously sent (positive) message, the LP need only send the corresponding anti-message
- Message send: in addition to sending the message, leave a copy of the corresponding anti-message in a data structure in the sending LP called the output queue.

Rollback: Receiving a Straggler Message





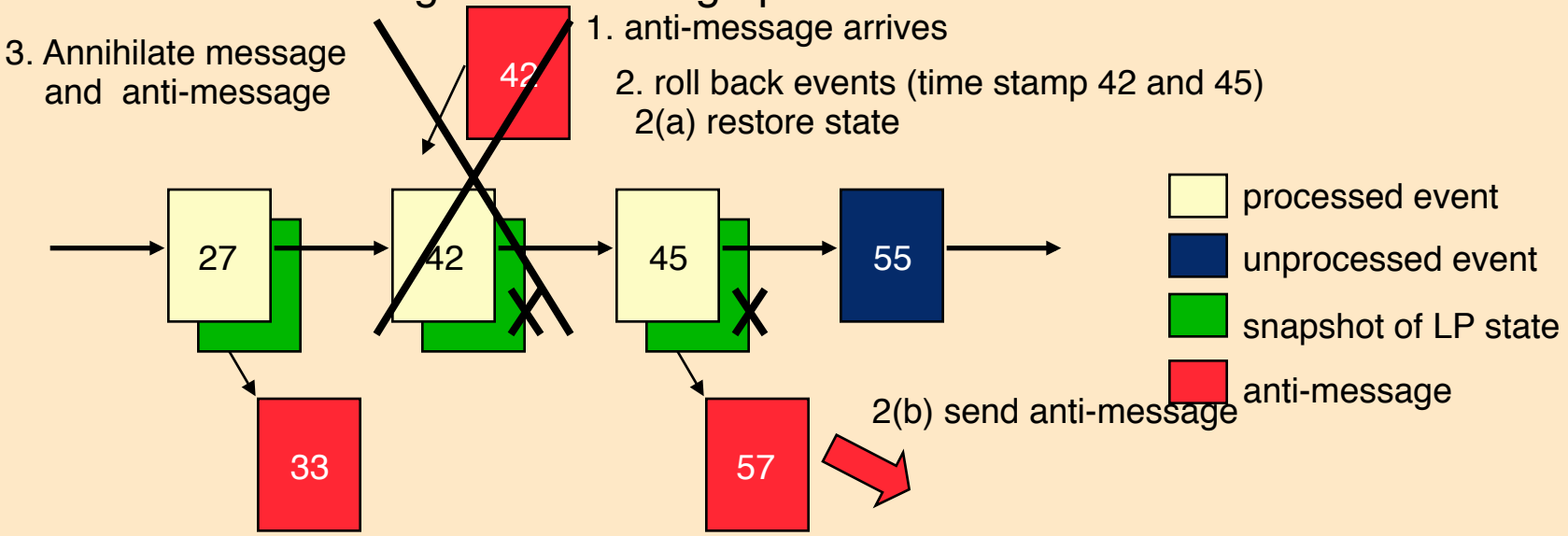
Processing Incoming Anti-Messages

Case I: corresponding message has not yet been processed

- annihilate message/anti-message pair

Case II: corresponding message has already been processed

- roll back to time prior to processing message (secondary rollback)
- annihilate message/anti-message pair



may cause "cascaded" rollbacks; recursively applying eliminates all effects of error

Case III: corresponding message has not yet been received

- queue anti-message
- annihilate message/anti-message pair when message is received

Global Virtual Time and Fossil Collection

A mechanism is needed to:

- reclaim memory resources (e.g., old state and events)
- perform irrevocable operations (e.g., I/O)

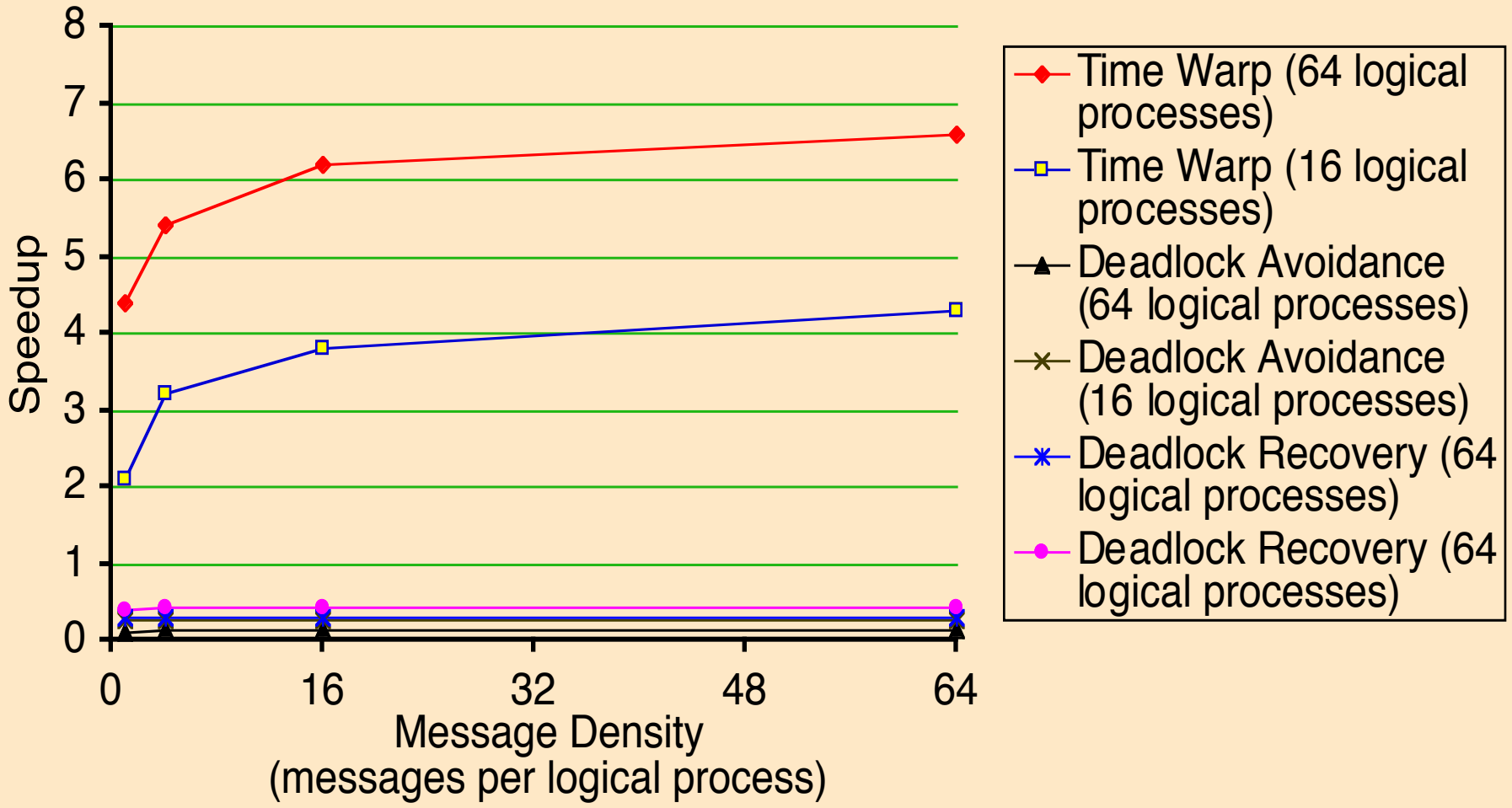
Observation: A lower bound on the time stamp of any rollback that can occur in the future is needed.

Global Virtual Time (GVT) is defined as the minimum time stamp of any unprocessed (or partially processed) message or anti-message in the system. GVT provides a lower bound on the time stamp of any future rollback.

- storage for events and state vectors older than GVT (except one state vector) can be reclaimed
- I/O operations with time stamp less than GVT can be performed.

Observation: The computation corresponding to GVT will not be rolled back, guaranteeing forward progress.

Time Warp and Chandy/Misra Performance



- eight processors
- closed queueing network, hypercube topology
- high priority jobs preempt service from low priority jobs (1% high priority)
- exponential service time (poor lookahead)

Summary

- Time Warp
  - Basic idea: detect out-of-order event execution and use a rollback mechanism to recover
  - Local control mechanism
    - Rollback
    - State saving
    - Anti-messages
    - Cascaded rollbacks