Distributed Computation

Jiarong Qian
Outline

• *Time, Clocks, and the Ordering of Events in a Distributed System* by Leslie Lamport

• *Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms* by Ozalp Babaoglu and Keith Marzullo
Distributed System

• “A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process.”

• Spatially separated computers
  – Multiprocessing on a single machine is similar to a distributed system but with negligible delays
Partial Order

- Process: a sequence of ordered events
- "Happened before" relationship (→)
  - "casually affect" relationship
- Concurrency: cannot tell which event happens first
Logical Clocks

• A function that assigns a value to each event on every process
• Clock Condition: For any events a, b: if a -> b then C(a) < C(b)
• Implementation Rules
  – At least a clock tick between two events on one process
  – Each message contains the timestamp when it was sent. Receiving event sets the clock value >= current time and > received timestamp
Total Order

- New “happens before” (=>) breaks the tie for concurrent events in partial order
  - Maintains the partial order relationship
  - Breaks the tie by comparing processes when two clock values are equal
Use Case: Synchronization

• Problem
  – Multiple processes request one resource
  – Only one process hold the resource at a time
  – FIFO
  – Process will relase the resource; all requests would be granted

• Assumption
  – Requests from the same process are received in the order of sent
  – Requests are broadcasted
  – All requests will be received
Algorithm

- Each process maintains a request queue initially containing $T_0:P_0$
- A process broadcasts $T_m:P_i$ to request for the resource
- Upon receiving a request, the request is added to the request queue and a timestamped response is sent
- Process $P_i$ releases a resource by removing $T_m:P_i$ from its request queue and broadcasts release resource
- When receiving release resource, a process removes the corresponding requests from the request queue
- Resource is granted when
  - There is no other requests happening before the request
  - Response later than the request timestamp has been received from other process
Anomalous Behavior

- Late messages can have smaller timestamp than earlier messages and the receiver cannot tell
- Strong clock condition (→)
- Logical clocks no longer work, need physical clocks
- Physical clock conditions
  - \( \left| \frac{dc}{dt} - 1 \right| < \kappa, \kappa << 1 \)
  - \( |C_i(t) - C_j(t)| < \epsilon \)
  - \( a \rightarrow b \) then \( b \) is at least \( \mu \) later than \( a \)
- \( \frac{\epsilon}{\kappa} \leq \mu \)
Distributed System States

• Each process can be considered as a state machine
• A monitor process monitors other process states
• The global state is the union of all process states
• A wait-for graph can be a format of the global state used to detect deadlock
• Measuring global state from messages can be inconsistent
Distributed Snapshot

- Snapshot means taking pictures of current process states
- Assuming each process has in-channels and out-channels
- For systems with physical clocks, inconsistency can be avoided by scheduling snapshots far enough in the future
- For systems with logical clocks, snapshots happen when receiving messages to do so.
- Non-monitor process records its state and notifies out processes, then it record messages of in processes
Discussion

• What’s the benefit of snapshot protocol 3 over snapshot protocol 2?