Info

• Quiz 2 average was 21.36 / 30.
  • It will be returned at the end of class.

• Please check your grades in LMS regularly to make sure they are correct.

• Project will be handed out today.
Replicated Log Algorithm

• Initially:
  • $L_i = \emptyset$
  • $T_i(j,k) = 0$ for all $j$ and $k$
Replicated Log Algorithm (2)

On local event e

- Advance $C_i$ (local event counter)
- $T_i(i,i) = C_i$
- $L_i = L_i \cup \{e_R\}$

For send (m) to $N_k$

- $NP_i = \{ e_R \mid e_R \in L_i \text{ and } \neg \text{hasRec}(T_i, e_R, k) \}$
- Send $NP_i$ and $T_i$ to $N_k$
Replicated Log Algorithm (3)

On receive(m) from N_k

- m contains NP_k and T_k
- L_i = L_i U NP_k
The Dictionary Problem

• A dictionary is a set of entries (unordered)
  • Denoted by $V$
• We want to fully replicate the dictionary at all sites

• $V_i$ denotes the copy of $V$ at site $i$.
• Let $e$ takes place at site $i$.
  Then $V(e)$ denotes the contents of $V_i$ immediately after $e$ takes place.

• Types of local events: insert $(x)$, delete $(x)$
• Assumptions
  • 1. For each entry $x$, there is at most one insert($x$), over all sites.
  • 2. delete($x$) can only be invoked at site $i$ if $x \in V_i$ at time of invocation.
The Dictionary Problem (2)

• The dictionary problem is to find an algorithm for maintaining the dictionary such that, given an execution \( <E, \rightarrow> \), for every event \( e \), \( x \in V(e) \) if and only if

  • \( \text{insert}(x) \rightarrow e \)

  AND

  • There is no \( \text{delete}(x) \) event \( g \) such that \( g \rightarrow e \)
GLOBAL SNAPSHOTs

Motivation

• In a long distributed computation, fault tolerance is achieved by having processes periodically save their state.

• After a failure, the failed process rolls back from one of its saved states.
  • Reduces the amount of lost computation.

• Each saved state is called a checkpoint.
Checkpoint-Based Recovery

- **Uncoordinated checkpointing**: Each process takes its checkpoints independently.

- **Coordinated checkpointing**: Processes coordinate their checkpoints to save a consistent system-wide state.
Uncoordinated Checkpointing

- Processes take checkpoint independently
- Domino Effect!

Figure 5. Rollback propagation, recovery line and the domino effect.
Coordinated Blocking

- Processes are coordinated to form a consistent global state, and …

* okay, channels flushed
Coordinated Blocking

• Advantage
  • Always consistent
  • No Domino Effect
  • Less storage overhead

• Disadvantage
  • Large latency to checkpoint
Coordinated Non-blocking

- Processes are coordinated, but …
- Do we really need to block …?