The Transaction Concept: Virtues and Limitations

Jim Gray, June 1981
Presented by Zhiyi Xu
ECS 265, UC Davis

Overview

Part 1:

- What is Transaction?
- A General Model

Part 2:

- NonStop
- Time-Domain Addressing
- Loggin and Locking

Part 3:

- Limitations of Known Techniques

What is Transaction?

Examples
- Contract Law
- Agreement
- “Christian Wedding Ceremony”

Component
- Negotiation
- Binding
- Intermediary

Properties
- Consistency
- Atomicity
- Durability

A General Model

“Transformations of a system state.”

Multiple actions can be done during transaction:

- Unprotected (no need to undone or redone)
- Protected (can or must be undone or redone)
- Real (once done, cannot be undone)

Outcomes of transaction:

- Committed
- Aborted

Types of Transaction:

- Simple (linear sequence of actions)
- Complex (concurrency, dependence, nested)

NonStop

No Perfect System and Other errors.

Unreliable (does wrong) and Unavailable (does not do right within time limited)

Solution (fault-tolerant application):

- Checkpoint and backup process
- Initial transaction state

Implementation of fault-tolerant applications:

- Time-domain addressing
- Logging plus locking

Time-Domain Addressing

<name, time>

- The old value continues to exist
- Can be addressed by specifying time interval

E: <V0, [T0, T1)>, <V1, [T1, T2)>, V2, <T2, *)>

- Most complete proposal for transaction system by Dave Reed
- Aborted if T2 >= T3
- Depending on commit record

Problems:

- Reads are writes
- Waits are aborts
- Timestamps force a single granularity
- Real operations and pseudo-time

Logging and Locking

All the undoable actions and redoable actions must have log records

Log records should be kept in stable storage for possible reconstruction

Sample log record:

- Name of transaction
- Previous log record of this transaction
- Next log record of this transaction
- Time
- Type of operation
- Object of operation
- Old value
- New value

Logging and Locking (cont.)

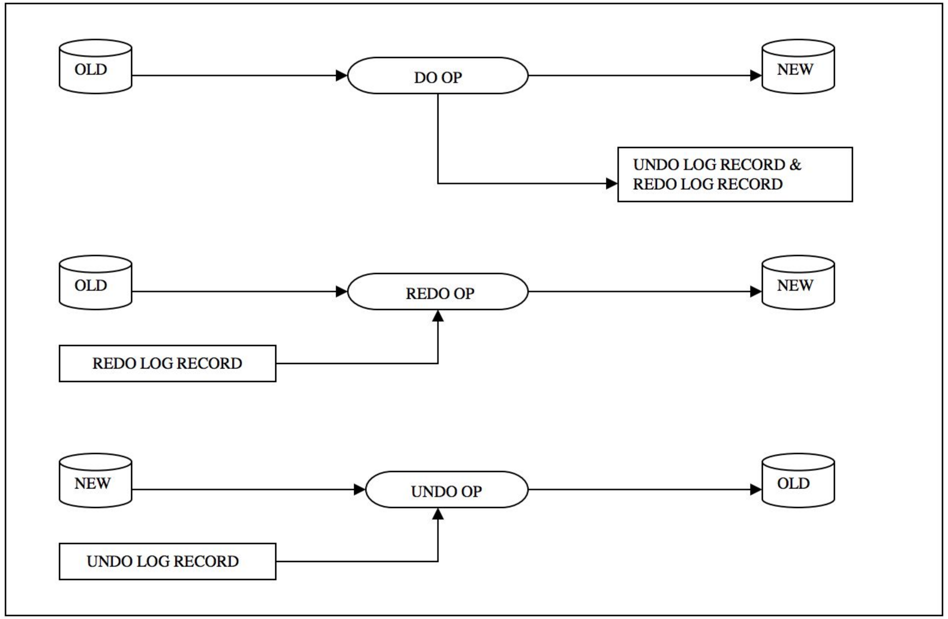

Figure 3. The DO-UNDO-REDO protocol. The execution of each protected action generates a log record which allows the action to be undone or redone. Unprotected actions need not generate log records. Actions which are not undoable (called real actions) use a related but slightly different protocol (see next figure).

Logging and Locking (cont.)

Other Details:

- Unprotected actions do not need log record.
- Real actions must be deferred until commit. Implemented by applying redo log.
- Undo and redo operations must be restartable.

Atomicity:

- Transaction that contributed to multiple logs.
- Multiple participants in one transaction (Ex: Wedding).

Concurrent transactions:

- Can read Input, but must not read or write output.
- Lock an object when it is accessed.
    - Lock confliction (predicate that covers the records).
    - Deadlock (Timeout or cycle detection).

Limitations of Known Techniques

Transactions cannot be nested inside transactions.

Transactions are assumed to last minutes rather than weeks.

Transactions are not unified with programming language.

Travel Agent Example

Nested transactions

- Might be undone by invoking compensating transaction.
- Visible effects to the outside world prior to the commit of parent transaction.

Long-lived transactions

- Concurrent transactions and deadlock.
    - Only active transactions hold locks.
    - UNDO and REDO commute with DO.
- Abortion when system restarts
    - Save point declaration.

Thank you!

Q&A