

Two-Phase Locking & Atomic Transactions

Dr. Daniel Andresen

CIS520 – Operating Systems

Atomic Transactions

- An **atomic action** is a single, indivisible action that preserves the all-or-nothing property; that is, either the action is completely performed, or it is not performed at all.
- A collection of instructions that are used to perform a single logical function is called a **transaction**. An **atomic transaction** is a transaction in which either all or none of the instructions are performed.
- **Example:** A banking transaction to transfer funds between accounts.
 `withdraw($100, acct1);`
 `deposit ($100, acct2);`
- *An atomic transaction may have several operations, but they are all executed as a single, atomic action.*

Transaction System Model

- **Stable storage** can be implemented with a pair of ordinary disks. Once committed to stable storage, a transaction should never be lost.
- **Transaction Primitives**
 - *begin_transaction*
 - *end_transaction* and try to commit the transaction to stable storage
 - *read* a value from the database (read(X))
 - *write* a value to the database (write(X))
 - *abort* a transaction (A)
- **Concurrent Transactions** - allow transactions to be interleaved to improve performance.
 - **Note:** it is overly restrictive to simply solve the potential conflicting interleavings by using mutual exclusion...

Desirable Properties of Interleaved Transactions

- *Serializability* - transactions have the same effect as some serial execution of the same transactions.
- *Atomicity* - transactions appear to execute indivisibly (atomically).
- *Permanence* - once a transaction commits, the changes are permanent.

Two-Phase Locking Protocol

- In order to enforce **serializability**, the two-phase locking protocol is generally used.
- **2PL Protocol:**
 - **Growing Phase:** A transaction may obtain locks, but not release any locks.
 - **Shrinking Phase:** A transaction may release locks, but not obtain any more.
- If a transaction is not able to obtain a lock, then it must wait until the lock is released.
- Consequently, *the 2PL Protocol is prone to deadlock.*
 - If a deadlock is detected, then one transaction in the cycle may be aborted and restarted to break the deadlock.

The 2PL Theory & Competition

- **Serializability Theorem:** A 2PL Scheduler only produces serializable histories; that is, histories which are conflict equivalent to some serial execution.
- **Other Protocols:**
 - Timestamp-Based (operations from transactions with lower timestamps are processed first)
 - Serialization Graph Testing (no cycles \Rightarrow allow operation)