Presentation on Database Recovery Techniques

PURPLE PETAL EDU HUB

Introduction

Database Recovery

- **Pre-condition** : At any given point in time the database is in a consistent state
- **Condition** : Some kind of **system failure** occurs
- **Post-condition :** Restore the database to the consistent state that existed before the failure

- **Database recovery** is the process of restoring the database to the most recent consistent state that existed just before the failure
- Example:
  - If the system crashes before a fund transfer transaction completes its execution, then either one or both accounts may have incorrect value. Thus, the database must be restored to the state before the transaction modified any of the accounts

Failure Classification

- Types of failures
  - The database may become unavailable for use due to
    - **Transaction failure**: Transactions may fail because of incorrect input, deadlock, incorrect synchronization
    - **System failure**: System may fail because of addressing error, application error, operating system fault, RAM failure, etc
    - **Media failure**: Disk head crash, power disruption, etc

Transaction Log

- For recovery from any type of failure data values prior to modification (BFIM - BeFore Image) and the new value after modification (AFIM – After Image) are required
- These values and other information is stored in a sequential file called Transaction log. A sample log is given below. Back P and Next P point to the previous and next log records of the same transaction

Main Recovery Techniques

1. Deferred Update:

- **These techniques do not physically update the DB on disk until after a transaction reaches its commit point**
- **The updates are recorded in the local transaction buffer and in the log file for recovery**
- **These techniques need only to redo the committed transaction and no-undo is needed in case of failure (No-Undo/Redo)**

Deferred Update example in a single user environment

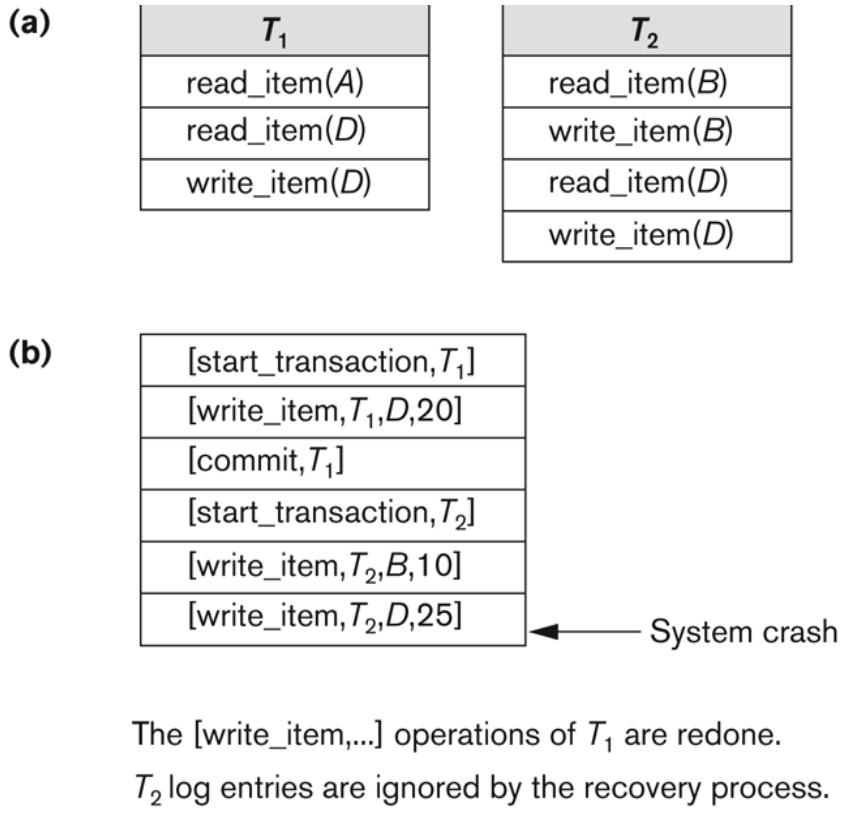

Figure 19.2
An example of recovery using deferred update in a single-user environment. (a) The READ and WRITE operations of two transactions. (b) The system log at the point of crash.

Main Recovery Techniques(Cont)

2. Immediate Update:

- **The DB may be updated by some operations of a transaction before the transaction reaches its commit point**
- **The updates are recorded in the log which must contain the old values (BFIM) and the new values (AFIM)**
- **These techniques need to undo the operations of the uncommitted transactions and redo the operations of the committed transactions (Undo/Redo)**
- **The Undo/No-Redo may be used in special case where all operations are recorded in the DB before the transaction commits**

Main Recovery Techniques(Cont)

3. Shadow Update:

- The modified version of a data item does not overwrite its disk copy but is written at a separate disk location

4. In-Place Update:

- The disk version of the data item is overwritten by the cache version

Main Recovery Techniques(Cont)

Example of Shadow paging:

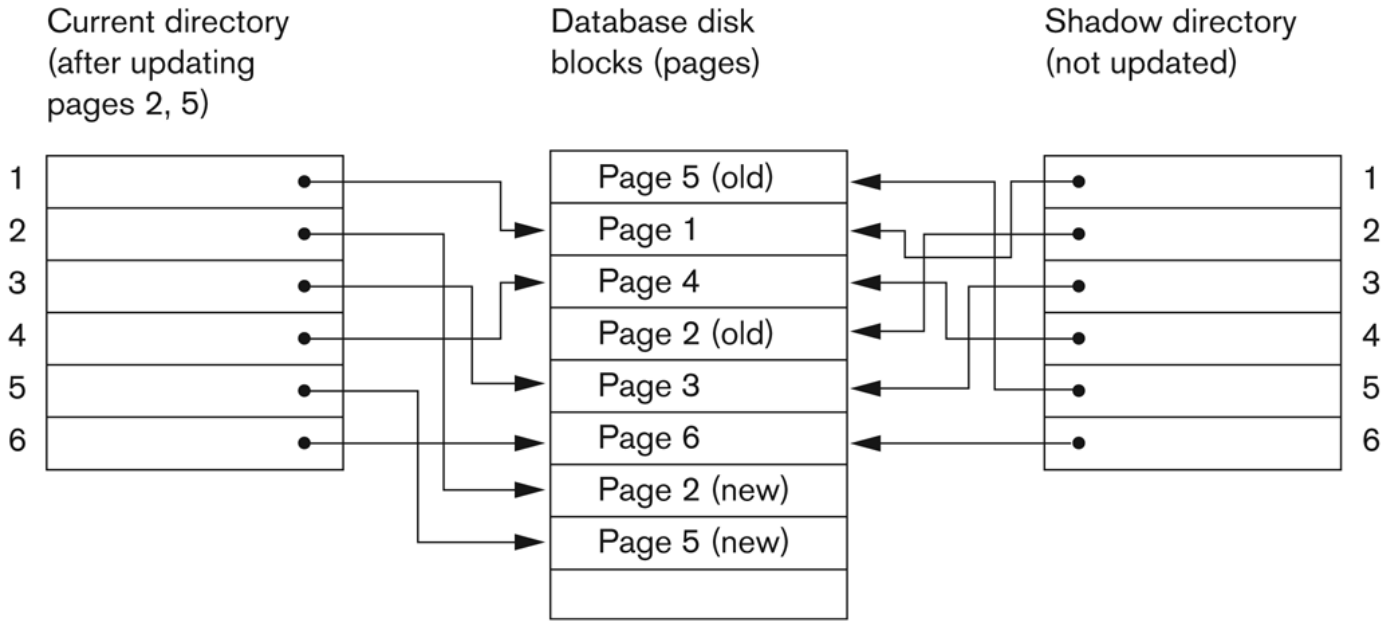

Figure 19.5
An example of shadow paging.

Checkpoints

Checkpointing:

- Time to time (randomly or under some criteria) the database flushes its buffer to database disk to minimize the task of recovery. The following steps defines a checkpoint operation:

  1. Suspend execution of transactions temporarily
  2. Force write modified buffer data to disk
  3. Write a [checkpoint] record to the log, save the log to disk.
  4. Resume normal transaction execution

- During recovery redo or undo is required to transactions appearing after [checkpoint] record

Recovery Techniques Based on Deferred Update(No Undo/Redo)

- The idea is to postpone any actual updates to the DB until the transaction completes its execution and commits (it follows the No-Steal approach)
- The updates are recorded only in the log and in the buffer
- After transaction reaches its commit point and the log is force-written to disk, the updates are recorded in the disk
- If the transaction fails before commit, no need to undo any operations

Recovery Techniques Based on Immediate Update(Undo/No-Redo)

- In this algorithm AFIMs of a transaction are flushed to the database disk under WAL before it commits
  - For this reason the recovery manager **undoes** all transactions during recovery
  - No transaction is **redone**
  - It is possible that a transaction might have completed execution and ready to commit but this transaction is also **undone**