DATABASE RECOVERY TECHNIQUES

Engr. Laraib Siddiqui

Introduction

Database recovery is the process of restoring the database to the most recent consistent state that existed just before the failure.

Three states of database recovery:

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

Types of failures

1. **Transaction failures**
   - Logical error. The transaction can no longer continue with its normal execution because of some internal condition, such as bad input, data not found, overflow, or resource limit exceeded.
   - System error. The system has entered an undesirable state (e.g., deadlock), as a result of which a transaction cannot continue with its normal execution. The transaction, however, can be reexecuted at a later time.
2. **System crash**
   - A hardware, software, or network error (also called media failure)
3. **Disk crash**
   - A disk block loses its content as a result of either a head crash or failure during a data-transfer operation.

Recovery Approaches

- **Steal approach**-cache page updated by a transaction can be written to disk before the transaction commits.
- **No-steal approach** -cache page updated by a transaction cannot be written to disk before the transaction commits.
- **Force approach**- when a transaction commits, all pages updated by the transaction are immediately written to disk.
- **No-force approach**-when a transaction commits, all pages updated by the transaction are not immediately written to disk.

Log

The log is a sequence of log records, recording all the update activities in the database. There are several types of log records. An update log record describes a single database write. It has these fields:

- **Transaction identifier**, which is the unique identifier of the transaction that performed the write operation.
- **Data-item identifier**, which is the unique identifier of the data item written. Typically, it is the location on disk of the data item, consisting of the block identifier of the block on which the data item resides and an offset within the block.
- **Old value**, which is the value of the data item prior to the write.
- **New value**, which is the value that the data item will have after the write.

Database Modification

In order for us to understand the role of these log records in recovery, we need to consider the steps a transaction takes in modifying a data item:

1. The transaction performs some computations in its own private part of main memory.
2. The transaction modifies the data block in the disk buffer in main memory holding the data item.
3. The database system executes the output operation that writes the data block to disk.

Database Modification

Update strategies may be placed into two basic categories.

Deferred-modification technique

- a transaction does not modify the database until it has committed.
- These techniques need only to redo the committed transaction and no-undo is needed in case of failure.

Immediate modification technique

- database modifications occur while the transaction is still active.
- The updates are recorded in the log must contain the old values and new values.
- These techniques need to undo the operations of the uncommitted transactions and redo the operations of the committed transactions .

Database Modification

A recovery algorithm must take into account a variety of factors, including:

- The possibility that a transaction may have committed although some of its database modifications exist only in the disk buffer in main memory and not in the database on disk.
- The possibility that a transaction may have modified the database while in the active state and, as a result of a subsequent failure, may need to abort.

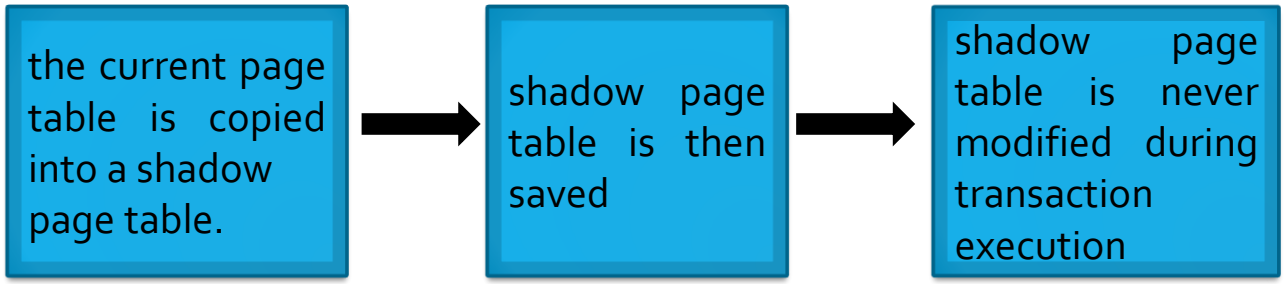
Because all database modifications must be preceded by the creation of a log record, the system has available both the old value prior to the modification of the data item and the new value that is to be written for the data item. This allows the system to perform *undo* and *redo* operations as appropriate.

- The **undo** operation using a log record sets the data item specified in the log record to the old value contained in the log record.
- The **redo** operation using a log record sets the data item specified in the log record to the new value contained in the log record.

Shadow Paging

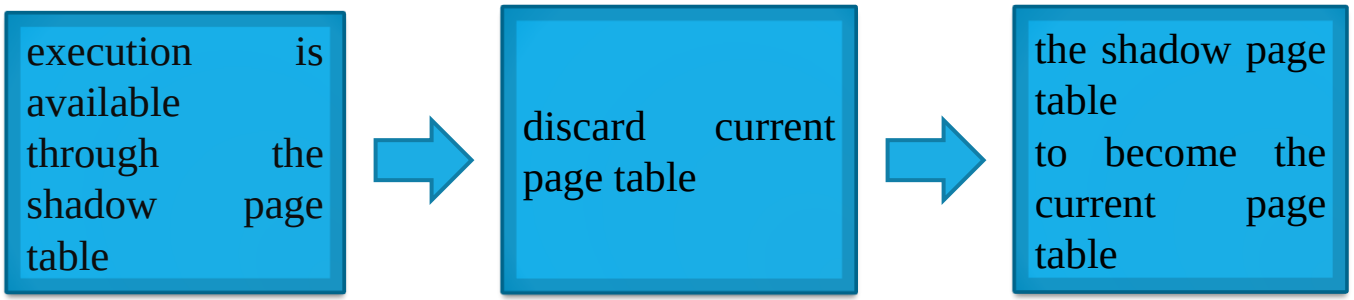
- In this technique, the database is considered to be made up of fixed-size disk blocks or pages for recovery purposes.
- Maintains two tables during the lifetime of a transaction-current page table and shadow page table.
- Store the shadow page table in nonvolatile storage, to recover the state of the database prior to transaction execution
- This is a technique for providing atomicity and durability.

When a transaction begins executing



Shadow Paging

To recover from a failure



Advantages

- No-redo/no-undo

Disadvantages

- Creating shadow directory may take a long time.
- Updated database pages change locations.
- Garbage collection is needed

“ARIES” Recovery algorithm.

Recovery algorithms are techniques to ensure database consistency ,transaction atomicity and durability without any failure.

Recovery algorithms have two parts

1. Actions taken during normal transaction processing to ensure enough information exists to recover from failures.
2. Actions taken after a failure to recover the database contents to a state that ensures atomicity, consistency and durability.

"ARIES" Recovery algorithm.

ARIES (*A*lgorithms for *R*ecovery and *I*solation *E*xploiting *S*emantics)

The ARIES recovery algorithm consist of three steps

- Analysis
- Redo
- Undo

"ARIES" Recovery algorithm.

- **Analysis -** Identify the dirty pages(updated pages) in the buffer and set of active transactions at the time of failure.

- **Redo -** Re-apply updates from the log to the database. It will be done for the committed transactions.

- **Undo -** Scan the log backward and undo the actions of the active transactions in the reverse order.

Recovery from disk crashes

- Recovery from disk crashes is much more difficult than recovery from transaction failures or machines crashes.
- Loss from such crashes is much less common today than it was previously, because of the wide use of redundancy in secondary storage (RAID technology).
  - (RAID - method of combining several hard disk drives into one logical unit.)

Typical methods are;

- The log for the database system is usually written on a separate physical disk from the database. or,
- Periodically, the database is also backed up to tape or other archival storage.

Practice

Suppose you are a manufacturer of product ABC, which is composed of parts A, B, and C. Each time a new product is created, it must be added to the product inventory, using the PROD_QOH in a table named PRODUCT. And each time the product ABC is created, the parts inventory, using PART_QOH in a table named PART, must be reduced by one each of parts A, B, and C. The sample database contents are shown in Table

Table name: PRODUCT

PROD_CODE	PROD_QOH
ABC	1,205

Table name: PART

PART_CODE	PART_QOH
A	567
B	498
C	549

1. How many database requests can you identify for an inventory update for both PRODUCT and PART?
2. Using SQL, write each database request you identified.
3. Write the complete transaction.
4. Write the transaction log.
5. Using the transaction log, trace its use in database recovery.

Practice

BEGIN TRANSACTION

UPDATE PRODUCT

SET PROD_QOH = PROD_OQH + 1

WHERE PROD_CODE = ‘ABC’

UPDATE PART

SET PART_QOH = PART_OQH -1

WHERE PART_CODE = ‘A’ OR PART_CODE = ‘B’ OR PART_CODE = ‘C’
COMMIT

Practice

Assume that product 'ABC' has a PROD_QOH = 23 at the start of the transaction and that the transaction is representing the addition of 1 new product. We also assume that PART components "A", "B" and "C" have a PROD_QOH equal to 56, 12, and 45 respectively.

TRL ID	TRX NUM	PREV PTR	NEXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
1	1A3	NULL	2	START	**START TRANSACTION				
2	1A3	1	3	UPDATE	PRODUCT	'ABC'	PROD_QOH	23	24
3	1A3	2	4	UPDATE	PART	'A'	PART_QOH	56	55
4	1A3	3	5	UPDATE	PART	'B'	PART_QOH	12	11
5	1A3	4	6	UPDATE	PART	'C'	PART_QOH	45	44
6	1A3	5	NULL	COMMIT	** END TRANSACTION				