

Introduction to crash Recovery

Transactions (or units of work) against a database can be interrupted unexpectedly.

If a failure occurs before all the changes that are part of the unit of work are completed, committed, and written to disk, the database is left in an inconsistent and unstable state.

A system crash usually occurs when there is some sort of hardware or software breakdown. Some other problems which are external to the system and cause the system to abruptly stop or eventually crash include failure of the transaction, operating system errors, power cuts, main memory crash etc.

These types of failures are often termed soft failures and are responsible for the data losses in the volatile memory. It is assumed that a system crash does not have any effect on the data stored in the non-volatile storage and this is known as the fail-stop assumption.

Crash Recovery : is the process by which the database is moved back to a consistent and usable state. This is done by rolling back incomplete transactions and completing committed transactions that were still in memory when the crash occurred.

Recovery is a service which should be provided by all the DBMS for ensuring that the database is dependable and remains in a consistent state in the presence of failures time of the transaction or after the end of a process. In this context, dependability refers to both the flexibility of the DBMS to various kinds of failure and its ability to recover from those failures.

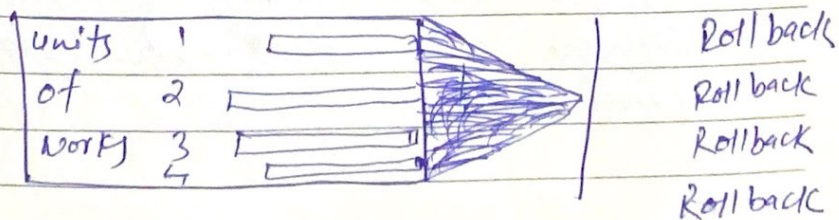


Fig: Rolling back units of work (Crash Recovery)

If the database or the database manager fails, the database can be left in an inconsistent state. The contents of the database might include changes made by transactions that were incomplete at the time of failure.

The database might also be missing changes that were made by transactions that completed before the failure but which were not yet flushed to disk.

A crash recovery operation must be performed in order to roll back the partially completed transactions and to write to disk the changes of completed transactions that were previously made only in memory.

Conditions that can necessitate a crash recovery include:

- * A power failure on the machine, causing the database manager and the database partitions on it to go down.
- * A hardware failure such as memory, disk, CPU, or network failure.
- * A serious operating system error that causes the database instance to end abnormally.

The Write-Ahead Log Protocol:

The log: is a sequence of records. Log of each transaction is maintained in some stable storage so that if any failure occurs, then it can be recovered from there. If any operation is performed on the database, then it will be recorded in the log.

But the process of storing the logs should be done before the actual transaction is applied in the database.

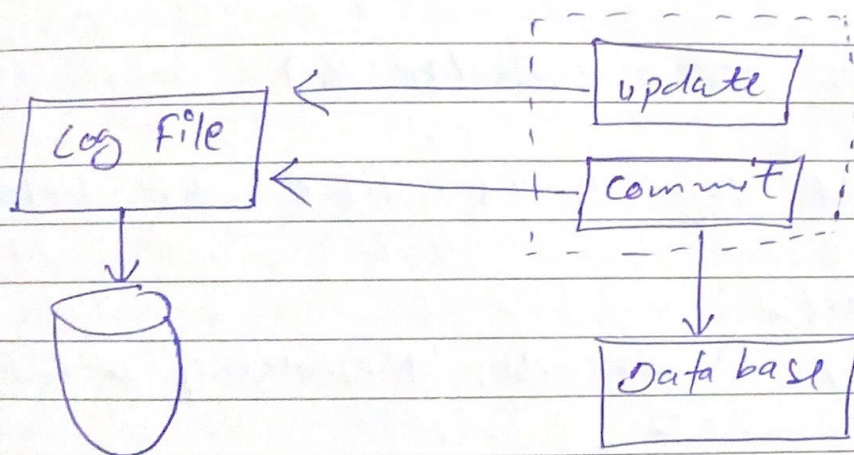


Fig: Log Based Recovery in DBMS

Let's assume there is a transaction to ~~edit~~ modify the city of a student from Butwar to Nepalgunj. The following logs are written for this transaction.

- (*) When the transaction is initiated, it writes 'star' log.
ie. $\langle T_n, \text{star} \rangle$



Mo Tu We Th Fr Sa Su

Memo No. _____

Date / /

- (*) When the transaction modifies the city from 'Butwal' to 'Nepalgunj', then another log is ~~written~~ written.
i.e. $\langle T_n, \text{city 'Butwal', 'Bangalore'} \rangle$
- (*) When the transaction is finished, then it writes another log to indicate the end of the transaction.
i.e. $\langle T_n, \text{commit} \rangle$
- (*) If transaction or modification is rolled back then it writes another log.
i.e. $\langle T_n, \text{rolled back} \rangle$

The final log looks like as below:

$\langle T_n, \text{start} \rangle$
 $\langle T_n, \text{city, 'Butwal', 'Nepalgunj'} \rangle$
 $\langle T_n, \text{commit} \rangle$
 $\langle T_n, \text{Rolled back} \rangle$

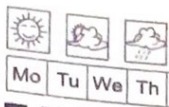
Write-Ahead logging: is a mechanism used to ensure that changes made by a transaction are in the redo log before they are written to the database. This makes sure that the changes are permanent and that they can be restored from the redo log in the event of a system failure.

Write-ahead logging (WAL) is a family of techniques for providing ~~atomic~~ atomicity and durability (two of the ACID properties) in database systems.

Here's how the write-ahead logging process typically works:

- (*) Logging changes
- (*) Writing changes to the Database
- (*) Commit Record
- (*) Redo and Undo operations
- (*) Durability Guarantee
- (*) Performance considerations

As a conclusion, write-ahead logging is a fundamental technique used in database systems to ensure data consistency and durability, especially in scenarios where system failures or crashes may occur.



Unit - 9 Back and Recovery

Memo No. _____

Date _____ / _____ / _____

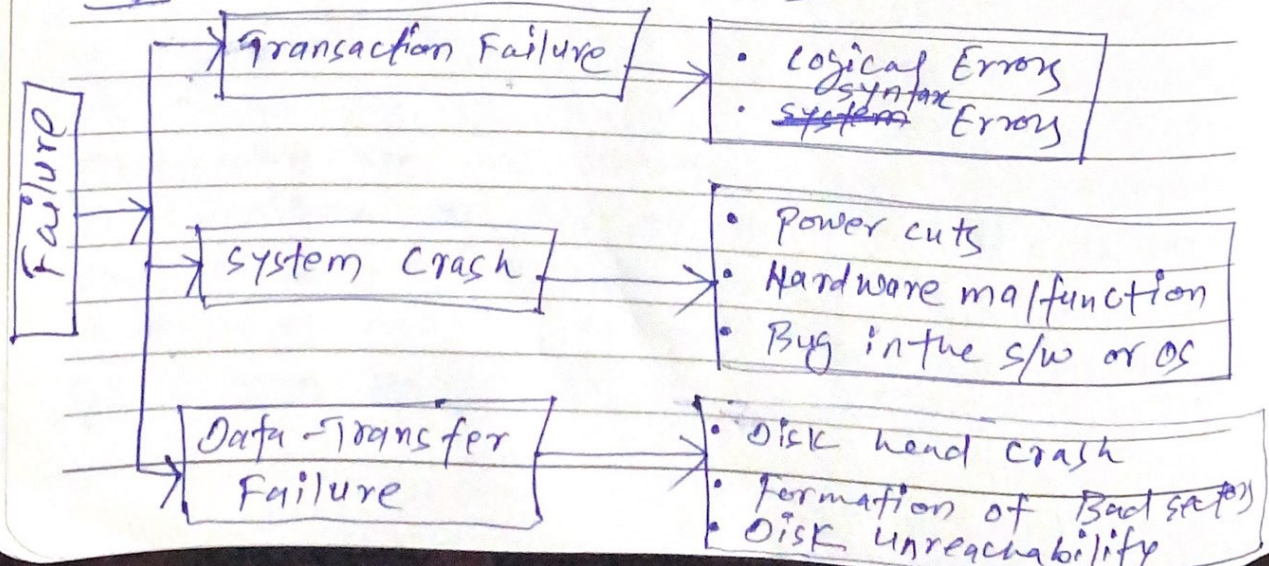
Failure of Database system:

A computer system can be failed due to variety of reasons like disk crash, power fluctuation, software error, sabotage and fire or flood at server site. These failures results in the lost of information. Thus, database must be capable of handling such situations to ensure that the atomicity and durability properties are preserved.

An integral part of a database system is the recovery manager that is responsible for recovering the data. The recovery manager deals with wide variety of database states because it is called during system failures. The database recovery is the process of restoring the database to a consistent state when a failure occurred.

A failure in DBMS can be classified as:

Fig: Failure classification in DBMS



(1) Transaction Failure:

A transaction failure occurs when it fails to execute or when it reaches a point from where it can't go any further. If a few transaction or process is hurt, then this is called as transaction failure.

Reasons for a transaction failure could be:

(a) Logical Errors:

If a transaction can't complete due to some code error or an internal error condition, then the logical error occurs.

(b) Syntax Error: It occurs when the DBMS itself terminates an active transaction because the database system is not able to execute it. i.e. the system aborts an active transaction, in case of deadlock or resource unavailability.

(2) System Crash:

A system crash usually occurs when there is some sort of hardware or software breakdown. In system crash that causes the loss in volatile memory of the computer and not in the persistent storage. Here, the information stored in the non-persistent storage like main memory, buffers, caches or registers, is lost.

The various types of system crash are:

- (*) operating system failure
- (*) ~~Main~~ Main memory crash



Mo Tu We Th Fr Sa Su

Memo No. _____

Date / /

- (*) Failure of supporting software
- (*) power failure
- (*) system generated error like integer overflow or divide-by-zero error.

(3) Data-Transfer Failure or Disk Failure:

When a disk failure occurs amid data-transfer operation resulting in loss of content from disk storage then such failures are categorized as data-transfer failures. Some other reasons for disk failure include disk head crash, disk unreachability, formation of bad sector, read-write errors on the disk, etc.

~~Some~~ Following are some causes of a disk failure:

- (*) Power failure
- (*) corruption of information on the disk.
- (*) Read/Write head crash of disk.
- (*) Formation of bad sectors etc.

In order to quickly recover from a failure caused amid a data-transfer operation, the backup copy of the stored on other tapes or disks can be used. Thus it's a good practice to backup our data frequently.