# Distributed Database System (CS-600)

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

**(Week 13) Lecture 5-26 (Chapter 20 of Book)**

**Objectives: Learning objectives of this lecture are**

- **Transactions and Recovery**
- **Read Buffer**
- **Update Strategies**
- **Recovery Mechanism**
- **Recovery Facilities**

# Distributed Database System (CS-600)

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

## Overview:

In this lecture, we will discuss how recovery system of a DBMS works? How data is read and written to storage device? What is buffer and what it does? How DBMS should react at the event of failure? And in the end we will see what recovery facilities a DBMS should provide to minimize the effect of failure on the system.

## Recap:

In previous lecture we have studied following

- **Deadlock detection**

  We created a Wait-for Graph (WFG) to detect a deadlock from a schedule.

- **Deadlock recovery**

  What are the important factors which must be considered while killing a transaction after a deadlock has occurred in a system? That includes choice of a victim, how far to rollback and avoid starvation.

- **Events of Failure**

  What are the events of failure that can cause a transaction to fail?

## Transactions and Recovery

- Transactions represent the basic unit of recovery in a database system.
- It is the role of the recovery manager to guarantee two of the four ACID properties of transactions, namely atomicity and durability, in the presence of failures.
- The recovery manager has to ensure that, on recovery from failure
  - either all the effects of a given transaction are permanently recorded in the database or none of them are.
- The situation is complicated by the fact that database writing is not an atomic (single-step) action
- It is therefore possible for a transaction to have committed but for its effects not to have been permanently recorded in the database
- Simply because they have not yet reached the database.

Consider an example where a transaction updates salary of an employee.

1- To implement the read operation, the DBMS carries out the following steps:

   a. Find the address of the disk block that contains the record with primary key value x.

   b. Transfer the disk block into a database buffer in main memory.

   c. Copy the salary data from the database buffer into the variable salary.

**Mr. Nauman Iqbal email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

2- For the write operation, the DBMS carries out the following steps.

   a. Find the address of the disk block that contains the record with primary key value x.

   b. Transfer the disk block into a database buffer in main memory.

   c. Copy the salary data from the variable salary into the database buffer.

   d. Write the database buffer back to disk.

- The database buffers occupy an area in main memory from which data is transferred to and from secondary storage.
- It is only once the buffers have been flushed to secondary storage that any update operations can be regarded as permanent.
- This flushing of the buffers to the database can be triggered by a specific command (for example, transaction commit) or automatically when the buffers become full.
- The explicit writing of the buffers to secondary storage is known as force-writing.
- If a failure occurs between writing to the buffers and flushing the buffers to secondary storage.
  - the recovery manager must determine the status of the transaction that performed the write at the time of failure.
- If the transaction had issued its commit, then to ensure durability the recovery manager would have to redo that transaction's updates to the database (also known as ROLLFORWARD).

---

Before we go into the update modes, let us first discuss concept of checkpoint.

## Checkpoint:

- Saving data into the permanent storage not on "WRITE/COMMIT" statement but when a specific time comes.

**Example 1:**

- Let us take the example of games.
- When a player is moving towards the end line.
- If players die before a certain point, then the game starts from the beginning.
- But if player cross a certain point (checkpoint) and dies, then games starts from the checkpoint not from the beginning.

- Which means that game does not save data unless a checkpoint appears.
- Whenever a checkpoint comes, all data that is unsaved is permanently saved in the system.
- We can say that data is moved from **RAM to Hard Drive** on arrival of checkpoint.

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

**Example 2:**

- Consider the given schedule having a transaction T1.
- T1 is reading and updating (writing) data-items x and y.
- Initial values of x and y are 50 and 20 respectively.

| Time | Local Variable Value (RAM) | T1 | Database Value (Hard Drive) |
|---|---|---|---|
| **t0** | x=50, y=20 | **Read (x)** | x=50,y=20 |
| **t1** | x=70, y=20 | **x = x + 20** | x=50,y=20 |
| **t2** | x=70, y=20 | **Write(x)** | x=50,y=20 |
| **t3** | x=110, y=20 | **x = x + 40** | x=50,y=20 |
| **t4** | x=110, y=20 | **Read(y)** | x=50,y=20 |
| **t5** | **CHECKPOINT** | | |
| **t5** | x=110, y=70 | **y = y + 50** | **x=110,y=20** |
| **t6** | x=110, y=70 | **Write(y)** | x=110,y=20 |
| **t7** | x=110, y=70 | **Commit** | x=110,y=20 |
| **t8** | | | x=110,y=20 |
| **t5** | **CHECKPOINT** | | |
| **t9** | | | **x=110,y=70** |

- Here we can there is a column that is representing RAM variables. Where value is changed with respect to operation.
- We have another column “Database Values”, where value only updates whenever Write statement is issued.
- But here values in database are not updated on “Write” statement.
- Values are only updated whenever a check point appears in the system.

## Update Methods:

- DBMS uses two modes for updating data into the permanent storage devices.
  - **Immediate Update**
    - Transaction does not wait for a checkpoint to update results in database.
    - As soon as transaction issues WRITE operations, values are saved into the database.
  - **Deferred Update**

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

- Transaction do not update value on issue of WRITE statement.
- Data is updated into the database whenever a "Checkpoint" appears in the system.
- Checkpoint appearance can vary and depend upon system.

**Example of Immediate and Deferred Update:**

- A student named "Amir" lives in a hostel.
- For some reasons, he has to shift to a new hostel.
- So he calls his friend "Usman" and ask for a favor to bring a lorry to shift his belongings from old hostel to new hostel.
- Amir belongings contains a pile of books, a laptop, a mattress and a blanket.
- Now Usman is very enthusiastic to help his friend and takes a 22 wheeler Truck for shifting to a new hostel.

NOTE: we can relate 22 wheeler Truck with the memory buffer, whose size is much larger than the data it is transferring from Hard Drive to RAM.

- **Immediate Update**
  - STEP 1: Amir will bring his blanket and place it in the vehicle.
  - STEP 2: Vehicle will start moving to drop the blanket at the new hostel.
  - STEP 3: Vehicle will return back to old hostel to pick up next item.

  NOTE: as soon as data is placed on buffer, it moves and stores it in hard drive. In this way, to and fro movement will increase which causes data transference slow.

- **Deferred Update**
  - STEP 1: Amir will bring his blanket, and place it on the vehicle.
  - STEP 2: Vehicle will wait for other items to be brought (checkpoint) and will not move.
  - STEP 3: Amir will bring rest of the items one by one and will place on the vehicle.
  - STEP 4: As soon as all the items will be loaded on the vehicle, then vehicle will move from old hostel to new hostel.

  **NOTE:** in this case, data will be placed in buffer and will not move into the hard drive for permanent save. In this case if data is in buffer and failure occurs, more data will be lost as compared to "Immediate update".

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

**DBMS Action:**

- In the light of above discussion, DBMS took following actions on transactions in the event of failure.
    - If data is not save on permanent storage (Hard Drive), then no changes will be made on Hard Drive. (Nothing)
    - If data is saved on permanent storage (Hard Drive) and commit statement has not been issued, then ROLLBACK statement will be issued to revert all changes made by transaction in database.
    - If data is not saved on permanent storage (still in buffer) and commit statement has already been issued by transaction and after that failure occurs, then ROLLFORWARD statement will be issued.

      **Reason:** when transaction issue commit statement it means that transaction was successful and completed all of its steps. If after complete transaction's data is not saved in database permanently then it is system's responsibility to make sure data is save. ROLLFORWARD statement re-executes all statements of a transaction is similar manner as it was executed earlier so that data is saved in database.

Example of Database Recovery:

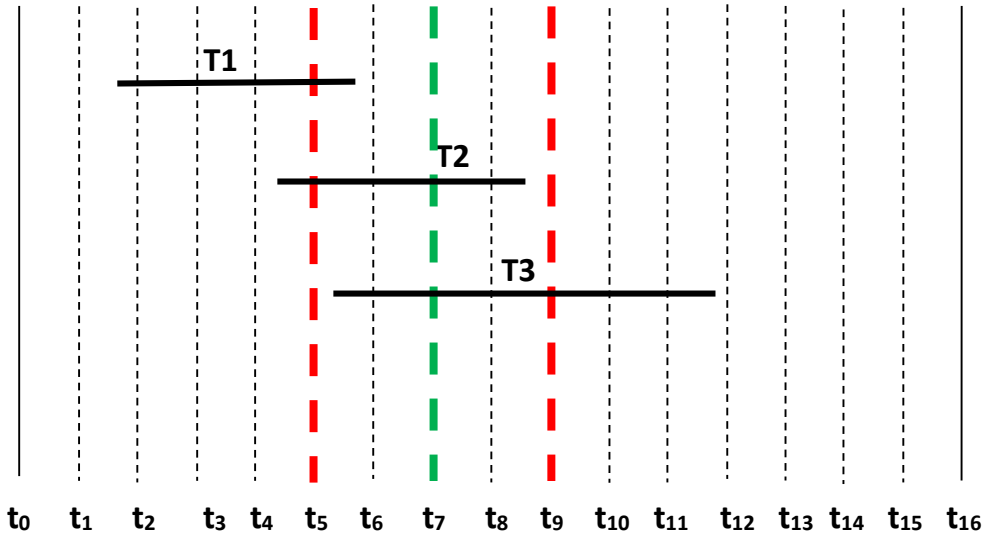
- Given is a schedule in which starting point and ending point of a transaction is mentioned.
- Failure points are mentioned for which we have to tell what operation will DBMS execute at the failure at this point.
  NOTE: there are multiple failure mentioned in schedule, it doesn't mean that there is more than one failure. It means that what will DBMS do if failure happened at that particular time. So each failure point is independent of each other.
- Checkpoints are also mentioned in the schedule. But as we know, checkpoints are only considered in "Deferred Update" mode.

**Example:**

# Distributed Database System (CS-600)

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---




- Above schedule has three transactions, S = {T1, T2, T3}.
- Red dotted lines are representing Failure points, which mean Failure is at Time $t_5$ and **$t_9$.**
- Green dotted line is representing checkpoint.
- Transaction T1 is starting before time **$t_2$** and ends before time **$t_6$**.
- Transaction T2 is starting before time **$t_5$** and ends before time **$t_9$**.
- Transaction T3 is starting before time **$t_6$** and ends before time $t_{12}$.
- So we have to tell what operation DBMS will perform at the event of failure which are **$t_5$** and **$t_9$**.

## 1. Immediate Update

| Transaction | Failure at t5 | Failure at t9 |
|---|---|---|
| T1 | Rollback | Nothing |
| T2 | Rollback | Nothing |
| T3 | Nothing | Rollback |

## Description:

- Now we will discuss the reason behind each operation.

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

| Transaction | Failure at t5 | Reason |
|---|---|---|
| **T1** | Rollback | As failure occurred during the transaction, which means data of the transaction was saved to database (hard drive). So at failure all data has to be reverted. |
| **T2** | Rollback | As failure occurred during the transaction, which means data of the transaction was saved to database (hard drive). So at failure all data has to be reverted. |
| **T3** | Nothing | As failure occurred before the starting of transaction, that means transaction was not in process so nothing will be done with transaction. |

| Transaction | Failure at t9 | Reason |
|---|---|---|
| **T1** | Nothing | As failure occurred after the transaction has been completed. All data of transaction has been saved to database so no changes are due. |
| **T2** | Nothing | As failure occurred after the transaction has been completed. All data of transaction has been saved to database so no changes are due. |
| **T3** | Rollback | As failure occurred during the transaction, which means data of the transaction was saved to database (hard drive). So at failure all data has to be reverted. |

**2- Deferred Update**

| Transaction | Failure at t5 | Failure at t9 |
|---|---|---|
| T1 | Nothing | Nothing |
| T2 | Nothing | Roll-Forward |
| T3 | Nothing | Roll-Back |

**Description:**

- Now we will discuss the reason behind each operation.
- Reminder: in deferred update, data is moved to hard drive only when checkpoint appears in the system.
- If checkpoint doesn't appear, data still remains in RAM.
- If data is in RAM, database will remain unchanged.

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

| Transaction | Failure at t5 | Reason |
|---|---|---|
| **T1** | Nothing | As failure occurred during the transaction, but checkpoint never occurred which means data is not updated in database (hard drive). Which means database is same as it was before the start of transaction. So no change is required. |
| **T2** | Nothing | As failure occurred during the transaction, but checkpoint never occurred which means data is not updated in database (hard drive). Which means database is same as it was before the start of transaction. So no change is required. |
| **T3** | Nothing | As failure occurred before the starting of transaction, that means transaction was not in process so nothing will be done with transaction. |

| Transaction | Failure at t9 | Reason |
|---|---|---|
| **T1** | Nothing | As failure occurred after the transaction has been completed. And after completion, checkpoint appeared which will save the data in database. Which means, all data is save to hard drive and transaction is completed so no operation is due on transaction. |
| **T2** | Roll-Forward | As transaction is completed, but checkpoint not appeared after the completion of transaction. Which means data was in RAM and failure occurred. Which will wash out transaction's data. Now we cannot perform ROLLBACK because after commit operation has been issued, transaction cannot ROLLBACK. So we will perform "ROLL-FORWARD" which means "Redo". This will re-execute all operations of transaction to save it permanently. |
| **T3** | Rollback | As failure occurred during the transaction, and checkpoint also appeared during transaction which means partial data of transaction has been copied to database (hard drive). So as failure occurred during transaction we have to revert all the changes made by transaction on the database. So we will have to "Rollback". |

## Recovery Facilities

A DBMS should provide the following facilities to assist with recovery:

- a backup mechanism, which makes periodic backup copies of the database
- logging facilities, which keep track of the current state of transactions and database changes.

**Mr. Nauman Iqbal email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

- a checkpoint facility, which enables updates to the database that are in progress to be made permanent.

- a recovery manager, which allows the system to restore the database to a consistent state following a failure.

## Backup mechanism

The DBMS should provide a mechanism to allow backup copies of the database and the log file (discussed next) to be made at regular intervals without necessarily having to stop the system first. The backup copy of the database can be used in the event that the database has been damaged or destroyed. A backup can be a complete copy of the entire database or an incremental backup, consisting only of modifications made since the last complete or incremental backup. Typically, the backup is stored on offline storage, such as magnetic tape.

## Log file

To keep track of database transactions, the DBMS maintains a special file called a log (or journal) that contains information about all updates to the database. The log may contain

the following data:

Transaction records, containing:

- transaction identifier

- type of log record (transaction start, insert, update, delete, abort, commit);

- identifier of data item affected by the database action (insert, delete, and update operations);

- before-image of the data item, that is, its value before change (update and delete operations only);

- after-image of the data item, that is, its value after change (insert and update operations only)

- log management information, such as a pointer to previous and next log records for that transaction (all operations).

- Checkpoint records, which we describe shortly.

The log is often used for purposes other than recovery (for example, for performance monitoring and auditing). In this case, additional information may be recorded in the log file

**Mr. Nauman Iqbal** **email id: nauman@biit.edu.pk Whatsapp# 0321-5821151**
**Mr. Ahsan** **email id: ahsan@biit.edu.pk, Whatsapp# 0333-5189656**

---

| Tid | Time | Operation | Object | Before image | After image | pPtr | nPtr |
|---|---|---|---|---|---|---|---|
| T1 | 10:12 | START | | | | 0 | 2 |
| T1 | 10:13 | UPDATE | STAFF SL21 | (old value) | (new value) | 1 | 8 |
| T2 | 10:14 | START | | | | 0 | 4 |
| T2 | 10:16 | INSERT | STAFF SG37 | | (new value) | 3 | 5 |
| T2 | 10:17 | DELETE | STAFF SA9 | (old value) | | 4 | 6 |
| T2 | 10:17 | UPDATE | PROPERTY PG16 | (old value) | (new value) | 5 | 9 |
| T3 | 10:18 | START | | | | 0 | 11 |
| T1 | 10:18 | COMMIT | | | | 2 | 0 |
| | 10:19 | CHECKPOINT | T2, T3 | | | | |
| T2 | 10:19 | COMMIT | | | | 6 | 0 |
| T3 | 10:20 | INSERT | PROPERTY PG4 | | (new value) | 7 | 12 |
| T3 | 10:21 | COMMIT | | | | 11 | 0 |

(for example, database reads, user logons, logoffs, and so on), but these are not relevant to recovery and therefore are omitted from this discussion.

## Checkpoint

> The point of synchronization between the database and the transaction log file. All buffers are force-written to secondary storage.

Checkpoints are scheduled at predetermined intervals and involve the following operations:

- writing all log records in main memory to secondary storage.
- writing the modified blocks in the database buffers to secondary storage.
- Writing a checkpoint record to the log file. This record contains the identifiers of all transactions that are active at the time of the checkpoint.

x----------x