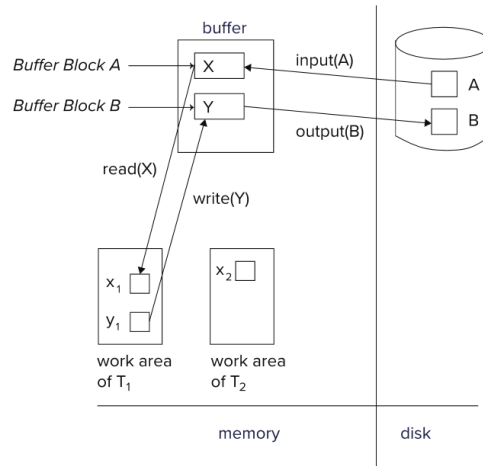


CSE411

Sample questions on Recovery

Q1: Explain the transaction processing by DBMS showing the transaction work area in memory, the buffer and the database in disk.



Q2: The accounts A, B, C has the balances 500, 1000 and 1500 respectively. Write a transaction where 20% of C account will deducted and equally divided into two accounts A and B. Write log records for the transaction.

<T1 start>

<T1, C, 1500, 1200>

<T1, A, 500, 200>

<T1, B, 1000, 700>

<T1 commit>

Q3: The log records are given bellow and the system failed:

<T₀ start>

<T₀, A, 1000, 950>

<T₀, B, 2000, 2050>

<T₀ commit>

<T₁ start>

<T₁, C, 700, 600>

>>>>>>>>> System failed

Explain the recovery from failure for the above case.

Q3. Ans.

redo (T_0) and undo (T_1): A and B are set to 950 and 2050 and C is restored to 700. Log records $\langle T_1, C, 700 \rangle$, $\langle T_1, \text{abort} \rangle$ are written out.

Q4:

- Why is checkpoint used in log-based recovery systems?
- How is a checkpoint performed? Explain with an example.
- In a database log file, there are two checkpoints in the same day:

First one: $\langle \text{checkpoint } L \rangle$ at time 8AM

Second one: $\langle \text{checkpoint } L \rangle$ at 4PM

The system failed at 5PM. There were 2000 transactions between first and second checkpoint and 100 transactions were active during second checkpoint. After second checkpoint, 500 transactions are committed before failure. There were 200 transactions running during failure. Describe the recovery from failure.

a.ans

- During recovery we need to consider only the most recent transaction T_i that started before the checkpoint, and transactions that started after T_i .
 - Scan backwards from end of log to find the most recent $\langle \text{checkpoint } L \rangle$ record
 - Only transactions that are in L or started after the checkpoint need to be redone or undone
 - Transactions that committed or aborted before the checkpoint already have all their updates output to stable storage.
- Some earlier part of the log may be needed for undo operations
 - Continue scanning backwards till a record $\langle T_i \text{ start} \rangle$ is found for every transaction T_i in L .
 - Parts of log prior to earliest $\langle T_i \text{ start} \rangle$ record above are not needed for recovery, and can be erased whenever desired.

b.ans.

Perform Checking

- Output all log records currently residing in main memory onto stable storage.
- Output all modified buffer blocks to the disk.
- Write a log record $\langle \text{checkpoint } L \rangle$ onto stable storage where L is a list of all transactions active at the time of checkpoint.

Example:

$\langle T_1 \text{ start} \rangle$

$\langle T_1 \text{ commit} \rangle$

<T2 start>
<T3 start>
<checkpoint {T2, T3}>
<T2 commit>
<T3 commit>
<T4 start>
<T5 strat>
<checkpoint {T4, T5}>
<T4 commit>
<T5 commit>
failed

c. ans.

Redo: 500 transactions

Undo: 200 transactions