Recovery system

1. Properties of transaction
   - (i) Atomicity ✓
   - (ii) Consistency X
   - (iii) Isolation X
   - (iv) Durability ✓

   ← Recovery.

2. Types of failure :
   - (i) Transaction failure : A transaction can be failed by
     - (a) logical error : The transaction can no longer processed due to some internal conditions such as bad input, data not found or resource limitation.
     - (b) system error : The system has entered an undesirable state such as deadlock then transaction can not continue with its normal execution.
   - (ii) System crash : There is a hardware failure. Then OS or database software will not run in continuation. By this running transaction will halt.
   - (iii) Disk failure : A disk block loses its content as a result of either a head crash or failure during a data transfer operation.

3. storage.:
   - (i) Volatile storage : RAM
   - (ii) Nonvolatile storage : ROM

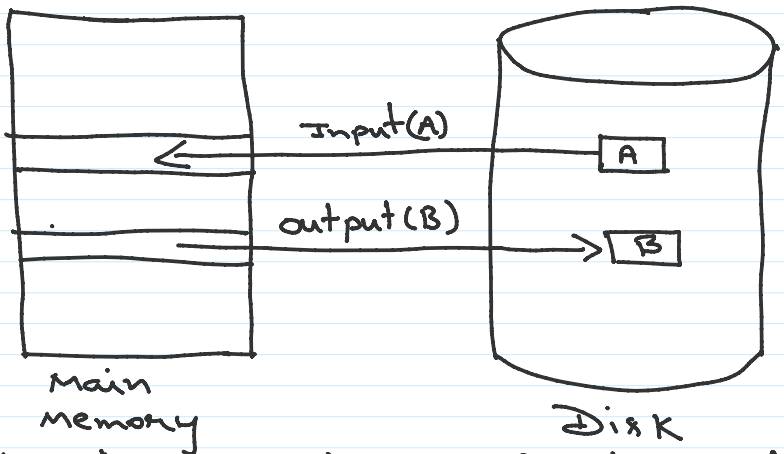
   } main memory

   - (iii) stable storage (secondary storage)

   RAID (Redundant Array of Inexpensive Disks)

   - Data is processed in volatile storage. Since O.S. directly communicates with RAM only.
   - Data is stored in stable storage.

✓ { O.S. directly communicates with RAM only.
→ { Data is stored in stable storage.



Data is transferred in terms of blocks from memory to disk & disk to memory.

There are two operation

(i) Input(A): It transfers the physical block A from disk to memory.

(ii) output(B): It transfers the buffer block B to the disk and replace the block B of disk.

* Block transfer between memory to disk can result in

(i) Successful completion: The transferred information arrived safely at the destination.

(ii) Partial failure: A failure occurred in the mid of transfer and destination block is incorrect.

(iii) Total failure: A failure occurred sufficiently early during the transfer that the destination block remains intact.

When system detects the failure, it invokes recovery procedure to restore the block to a consistent state.

To do this, the system maintains two physical blocks for each logical block of database.

When there is an output operation, it is executed as

When there is an output operation, it is executed as

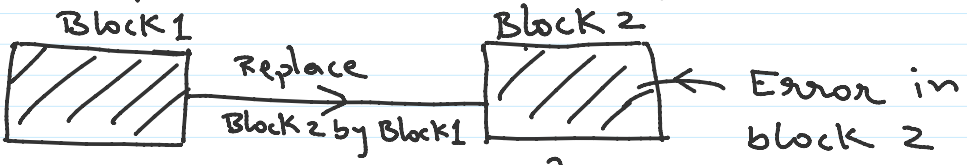
(i) write the information onto the first physical block. (local)
(ii) when first write operation is completed successfully, then the same information will be replicated to second physical block. (remote block)
(iii) the output is completed only after the second write completes successfully.

4. Recovery:

If there is an error, the system checks both the copies of the block.

(i) If both the blocks are same and no detectable error, then no action is needed.

(ii) If the system detects an error in one block, then it replaces its content by the other block.



* How error can be detected?

checksum of block 1 differs from block 2.

(iii) There is no error but content of both the blocks differ, the system replaces the content of first block with the second block.

There are two main operations associated with each data item.

(i) Read(x): It assigns the value of data item x to local variable x. It is executed as follows:

(a) If Bx block on which data item 'x' is present that is not in memory. It issues Input(Bx).

(b) It assigns the value of variable 'x' to local variable X.

(b) It assigns the value of variable x to local variable X.

(ii) write(x) : It copies the value of local variable x to data item 'x' in the buffer block. It is executed as follows:

(a) If block Bx on which x presented is not in main memory, then it issues Input(Bx).

(b) It copies the value of local variable x to data item 'x'.

5. Recovery techniques

(i) The mainly used recovery technique is log based recovery.

(ii) In log based recovery, all the modification of database are stored in a log.

(iii) If any kind of failure occur, then the log records are used to recover the database.

(iv) The log records keep

(a) Transaction identifier : Every transaction has a unique identifier which is keep in the log record.

(b) Data item identifier : Every data item has a unique identifier which is also keep in the log record.

(c) old value : It is the value of data item prior to write.

(d) New value : It is the value of data item after successful write.

(v) the log records are maintained as follows:

(a) <Ti, start> : Transaction Ti has started.

(b) <Ti, Xj, V1, V2> : Transaction Ti has updated

(b) $<T_i, X_j, V_1, V_2>$: Transaction T_i has updated value of data item X_j from old value V_1 to new value V_2.

(c) $<T_i, \text{commit}>$: Transaction T_i has completed.

(d) $<T_i, \text{abort}>$: Transaction T_i has aborted

(vi) There are two log based recovery techniques

(1) Deferred Database modification :

(a) In deferred modification technique, all the modification to the database are kept in the log record until transaction partially commits.

(b) If transaction has successfully committed, then database is updated by log records.

(c) If transaction has aborted, then simply discard the log record of that transaction but no change to the database.

(d) If any kind of failure occurs, then deferred modification required only redo(T_i) operation.

(e) redo(T_i) : It sets the value of all data items updated by transaction T_i to new values.

(d) The log record of deferred modification technique keeps only '3' parameters for each write

$<T_i, X_j, V_2>$

Transaction T_i has updated value of data item X_j to new value V_2.

e.g. Schedule $A = 1000, B = 2000$
$C = 5000$

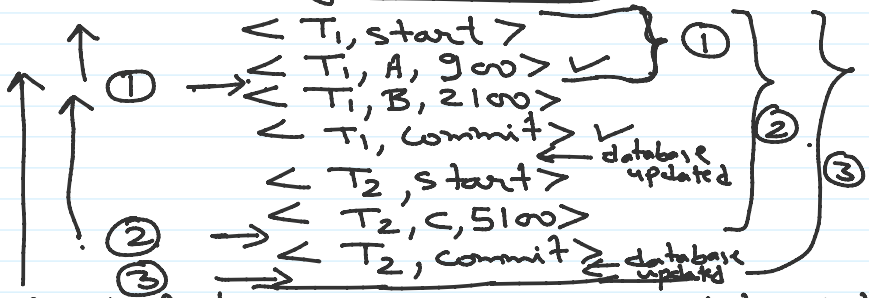
T_1	T_2
Read(A);	Read(C)

T_1	T_2

Read(A);
A := A − 100;
write(A);
Read(B);
B := B + 100;
write(B);
	Read(C)
	C := C + 100;
	write(C);

Transaction T_1 & T_2 are executed as $T_1 \rightarrow T_2$

Log records



If No failure occurs then update database by the log records.

Now. If any kind of failure occures at any point then following action will take place by recovery system.

Failure point	Recovery Action
1.	Discard the log records of T_1.
2.	Redo(T_1) & discard the log record of T_2
3.	Redo(T_1) & Redo(T_2)

For recovery of database in deferred modification technique, log records are scanned from last record to first log record in reverse order.

If we get $< T_i, commit >$ log record for any transaction T_i, then perform redo(T_i) for that transaction otherwise discard the log record of other transaction

(2) Immediate modification technique:

(i) All the modation done by transaction are

(2) Immediate modification technique:

(i) All the updation done by transactions are updated to the database parallelly at the same time.

(ii) A log record is also maintained for every modification to the database in the form of $< T_i, X_j, V_1, V_2 >$ where Transaction T_i has updated value of data item X_j from old value V_1 to new value V_2.

(iii) These log records is used later for recovery if any kind of failure occures.

(iv) In recovery there are two operations are performed by recovery system.

(a) redo(T_i): It sets the value of all the data items updated by the transaction T_i to new values.

(b) undo(T_i): It restores the value of all data items updated by the transaction T_i to old values.

Consider the same example

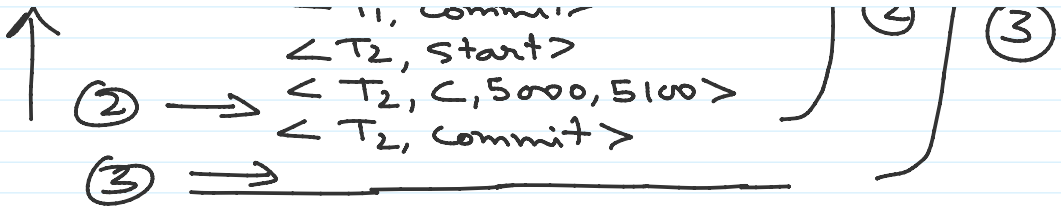
schedule $A = 1000, B = 2000, C = 5000$

T_1	T_2
Read(A);	Read(C)
A := A - 100;	C := C + 100;
write(A);	write(C)
Read(B);	
B := B + 100;	
write(B);	

Now, the log records of schedule $T_1 \to T_2$ as

Log record
$< T_1, start >$
$< T_1, A, 1000, 900 >$
$< T_1, B, 2000, 2100 >$
$< T_1, commit >$
$< T_2, start >$
$< T_2, C, 5000, 5100 >$

① ② ③



If no failure occures, then database is updated parallelly. so, we have to discard the log record.

Failure Point	Recovery Action
1	undo(T_1).
2	redo(T_1) & undo(T_2).
3	redo(T_1) & redo(T_2).

For recovery of database, we scan log record in reverse order.

(i) If system finds <T_i, commit> for any transaction T_i, then system performs redo(T_i) action.

(ii) Transactions for them log record <T_i, start> are presented but no <T_i, commit> are presented for them system performs undo(T_i) action.