

DB - HW14

19.1

Explain why log records for transactions on the undo-list must be processed in reverse order, whereas redo is performed in a forward direction.

Answer:

- Undo-List

Take a single transaction T_1 for example.

1. *< checkpoint >*
 2. *< T₁, begin >*
 3. *< T₁, A, 1, 2 >*
 4. *< T₁, A, 2, 3 >*
- *break down* –

The original value of A is 1.

- If we do the undo in forward order, we should first set A's value as 1, then reset it as 2.
- If we do the undo in backward order, we should first set A's value as 2, then reset it as 1.

It is obvious that, only when we do the undo in reverse order, can we ensure values that have been updated for many times could be recover to its original value at the end.

- Redo-List

Let's take a single transaction T_2 for example this time:

1. *< T₂, begin >*
 2. *< T₂, B, 1, 2 >*
 3. *< checkpoint >*
 4. *< T₂, B, 2, 3 >*
 5. *< T₂, commit >*
- *break down* –

The ultimate value of B is 3.

- If we do the redo in forward order, we should first set A's value as 2, then reset it as 3.
- If we do the redo in backward order, we should first set A's value as 2, then reset it as 1.

It is obvious that, only when we do the redo in forward direction, can we ensure values that have been updated for many times could be recover to its ultimate value at the end.

19.2

Explain the purpose of the checkpoint mechanism. How often should checkpoints be performed? How does the frequency of checkpoints affect:

- **System performance when no failure occurs?**
- **The time it takes to recover from a system crash?**
- **The time it takes recover from a media(disk) failure?**

Answer:

- Purpose:
 1. Checkpoint Mechanism is used to reduce the time required for recovery.
Working with the log-based schema, it avoid searching the entire log schema or undo/redo all the transactions. Besides, most of the log-records prior to the checkpoint can be ignored at the time.
 2. It enabled us to clear the log-record from stable storage when it is full.
- How often should it be performed:
 1. Reduce when fast recovery is not that needed.
 2. Increase when fast recovery is critical.
 3. Reduce when the amount of stable storage available is less.
- If no failure occurs, checkpoint mechanism would cause some loss in performance.
- Checkpoints can reduce the time taken in recovery from system crash.
- Checkpoints have no effect on recovery from disk crash.

19.22

Suppose there is a transaction that has been running for a very long time but has performed very few updates?

a. What effect should the transaction have on recovery time with the recovery algorithm of Section 19.4, and with the ARIES recovery algorithm?

b. What effect would the transaction have on deletion of old log records?

Answer:

a. A transactions has been running for a very long time with few updates → during the recovery, we should scan the log in backward order till the beginning of the transaction in undo phase, which would increase the time cost.

However, in ARIES, it would be not that bad, as ARIES considers the LastLSN and PrevLSN in the undo-list during the backward scanning, allowing it to skip intermediate records belonging to completed treansactions.

b. A long running transaction → no log records which are written after it started can be deleted till it's committed or aborted.

This might generate a pretty large log file, though most of the transactions in the log file have completed.

And this transaction becomes a obstacle when we want to delete old log records.