

## 数据库week15

### 19.2

Explain the purpose of the checkpoint mechanism. How often should check points 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 to recover from a media (disk) failure?

Checkpoint is done with log-based recovery schemes to reduce the time required for recovery after a crash. If there is no checkpoint, then the entire log must be searched after a crash, and all transactions undone/redone from the log. If checkpoint had been performed, then most of the log-records prior to the checkpoint can be ignored at the time of recovery.

Frequency to perform checkpoints is to clear log-records from stable storage as it gets full.

Because checkpoints cause some loss in performance while they are being taken, their frequency should be reduced if fast recovery is not critical. If we need fast recovery checkpoint frequency should be increased. If the amount of stable storage available is less, frequent checkpoint is unavoidable.

Checkpoints have no effect on recovery from a disk crash; archival dumps are the equivalent of checkpoints for recovery from disk crashes.

### 19.10

Explain the reasons why recovery of interactive transactions is more difficult to deal with than is recovery of batch transactions. Is there a simple way to deal with this difficulty? (Hint: Consider an automatic teller machine transaction in which cash is withdrawn.)

Interactive transactions are more difficult to recover from than batch transactions because some actions may be irrevocable. For example, an output (write) statement may have fired a missile, or caused a bank machine to give money to a customer. The best way to deal with this is to try to do all output statements at the end of the transaction. That way if the transaction aborts in the middle, no harm will have been done. Output operations should ideally be done atomically; for example, ATM machines often count out notes, and deliver all the notes together instead of delivering notes one-at-a-time. If output operations cannot be done atomically, a physical log of output operations, such as a disk log of events, or even a video log of what happened in the physical world can be maintained, to allow perform recovery to be performed manually later, for example by crediting cash back to a customers account.

### 19.21

Consider the log in Figure 19.5. Suppose there is a crash just before the log record  $\langle T_0$  abort  $\rangle$  is written out. Explain what would happen during recovery

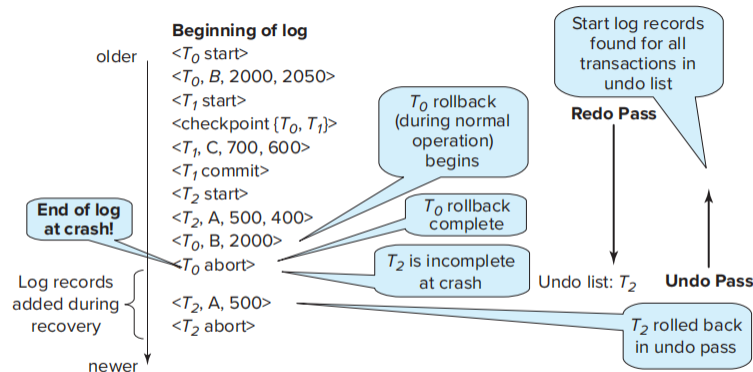


Figure 19.5 Example of logged actions and actions during recovery.

When Redo state:

Now ,ATT =  $\langle T_0, T_1 \rangle$

- $\langle T_1 \text{ Commit} \rangle$ : C  $\rightarrow$  600
- $\langle T_1 \text{ Commit} \rangle$ : remove  $T_1$  from ATT
- $\langle T_2 \text{ Start} \rangle$  : Add  $T_2$  to ATT
- $\langle T_2, A, 500, 400 \rangle$  : A  $\rightarrow$  400
- $\langle T_0, B, 2000 \rangle$  B  $\rightarrow$  2000

Now ,ATT =  $\langle T_0, T_2 \rangle$

- RollBack  $T_0, T_2$  : A  $\rightarrow$  500 & B  $\rightarrow$  2000

So A = 500, B = 2000 ,C = 600

The log records added during recovery are:

- $\langle T_2, A, 500 \rangle$
- $\langle T_2 \text{ abort} \rangle$
- $\langle T_0, B, 2000 \rangle$
- $\langle T_0 \text{ abort} \rangle$

## 19.25

In the ARIES recovery algorithm:

- If at the beginning of the analysis pass, a page is not in the checkpoint dirty page table, will we need to apply any redo records to it? Why?

When a page is not in the checkpoint dirty page table at the beginning of the analysis pass, redo records prior to the checkpoint record need not be applied to it as it means that the page has been flushed to disk and been removed from the `DirtyPageTable` before the checkpoint.

- What is RecLSN, and how is it used to minimize unnecessary redos?

The RecLSN is an entry in the `DirtyPageTable`, which reflects the LSN at the end of the log when the page was added to `DirtyPageTable`.

During the redo pass of the ARIES algorithm, if the LSN of the update log record encountered, is less than the RecLSN of the page in DirtyPageTable, then that record is not redone but skipped. Further, the redo pass starts at RedoLSN, which is the earliest of the RecLSNs among the entries in the checkpoint DirtyPageTable, since earlier log records would certainly not need to be redone.