Recovery

Chapter 6.3

V3.1

Dr Gordon Russell

NAPIER UNIVERSITY
EDINBURGH

Recovery

A database might be left in an inconsistent state when:

- deadlock has occurred.
- a transaction aborts after updating the database.
- software or hardware errors occur.
- incorrect updates have been applied to the database.

If the database is in an inconsistent state, it is necessary to recover to a consistent state. The basis of recovery is to have backups of the data in the database.

Recovery: the dump

The simplest backup technique is ‘the Dump’.

- entire contents of the database are backed up to an auxiliary store.
- must be performed when the state of the database is consistent - therefore no transactions which modify the database can be running
- dumping can take a long time to perform
- you need to store the data in the database twice.
- as dumping is expensive, it probably cannot be performed as often as one would like.
- a cut-down version can be used to take ‘snapshots’ of the most volatile areas.

Recovery: the transaction log

A technique often used to perform recovery is the transaction log or journal.

- records information about the progress of transactions in a log since the last consistent state.
- the database therefore knows the state of the database before and after each transaction.
- every so often the database is returned to a consistent state and the log may be truncated to remove committed transactions.
- when the database is returned to a consistent state the process is often referred to as ‘checkpointing’.

Deferred Update

Deferred update, or NO-UNDO/REDO, is an algorithm to support ABORT and machine failure scenarios.

- While a transaction runs, no changes made by that transaction are recorded in the database.
- On a commit:
  1. The new data is recorded in a log file and flushed to disk
  2. The new data is then recorded in the database itself.
- On an abort, do nothing (the database has not been changed).
- On a system restart after a failure, REDO the log.

Example

Consider the following transaction T1

Transaction T1

read(A)
write(10,B)
write(20,C)
commit

Example cont…

Using deferred update, the process is:

Time	Action	Log
t1	START	-
t2	read(a)	-
t3	write(10,b)	B => 10
t4	write(20,c)	C => 20
t5	COMMIT	COMMIT

Before

Disk

		B=6
A=5	C=2	

After

		B=10
A=5	C=20	

Example cont...

If the DMBS fails and is restarted:

1. The disks are physically or logically damaged then recovery from the log is impossible and instead a restore from a dump is needed.
2. If the disks are OK then the database consistency must be maintained. Writes to the disk which were in progress at the time of the failure may have only been partially done.

Example cont...

3. Parse the log file, and where a transaction has been ended with ‘COMMIT’ apply the data part of the log to the database.
4. If a log entry for a transaction ends with anything other than COMMIT, do nothing for that transaction.
5. flush the data to the disk, and then truncate the log to zero.
6. the process or reapplying transaction from the log is sometimes referred to as ‘rollforward’.

Immediate Update

Immediate update, or UNDO/REDO, is another algorithm to support ABORT and machine failure scenarios.

- While a transaction runs, changes made by that transaction can be written to the database at any time. However, the original and the new data being written must both be stored in the log BEFORE storing it on the database disk.
- On a commit:
  1. All the updates which have not yet been recorded on the disk are first stored in the log file and then flushed to disk.
  2. The new data is then recorded in the database itself.

- On an abort, UNDO all the changes which that transaction has made to the database disk using the log entries.
- On a system restart after a failure, REDO committed changes from log.

Example cont…

Using immediate update, and transaction T1 again, the process is:

Time	Action	LOG
t1	START	-
t2	read(a)	-
t3	write(10,b)	Was B == 6, Now 10
t4	write(20,c)	Was C == 2, Now 20
t5	COMMIT	COMMIT

	Before			During			After		
Disk			B=6			B=10			B=10
	A=5	C=2		A=5	C=2		A=5	C=20	

Immediate Update Example cont...

If the DMBS fails and is restarted:

1. The disks are physically or logically damaged then recovery from the log is impossible and instead a restore from a dump is needed.
2. If the disks are OK then the database consistency must be maintained. Writes to the disk which were in progress at the time of the failure may only have been partially done.

3. Parse the log file, and where a transaction has been ended with ‘COMMIT’ apply the ‘new data’ part of the log to the database.
4. If a log entry for a transaction ends with anything other than COMMIT, apply the ‘old data’ part of the log to the database.
5. flush the data to the disk, and then truncate the log to zero.

Rollback

The process of undoing changes done to the disk under immediate update is frequently referred to as rollback.

- Where the DBMS does not prevent one transaction from reading uncommitted modifications (a ‘dirty read’) of another transaction (i.e. the uncommitted dependency problem) then aborting the first transaction also means aborting all the transactions which have performed these dirty reads.
- as a transaction is aborted, it can therefore cause aborts in other dirty reader transactions, which in turn can cause other aborts in other dirty reader transaction. This is referred to as ‘cascade rollback’.