Logging and Recovery

444 Section, April 23, 2009

Why do we need to recover a DB?

- System crash
  - OS failure
  - DBMS crash
  - Power failure

How can logging make recovery easier/better?

Our undo log notation

- <START T>
  - Transaction T has begun
- <COMMIT T>
  - T has committed
- <ABORT T>
  - T has aborted
- <T,X,v>   -- Update record
  - T has updated element X, and its _old_ value was v

An undo logging problem

- Given this undo log, when can each data item be written to disk?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<T2,E,e>
9	<START T4>
10	<T4,F,f>
11	<T3,G,g>
12	<COMMIT T2>

Undo logging problem, continued

- After writing these log entries, the DBMS crashes. What does it do when it restarts?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<T2,E,e>
9	<START T4>
10	<T4,F,f>
11	<T3,F,f'>
12	<COMMIT T2>

Undo logging problem, part 3

- Now suppose the DBMS crashes after log entry 8. What does it do when it restarts?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<T2,E,e>
9	<START T4>
10	<T4,F,f>
11	<T3,G,g>
12	<COMMIT T2>

What if it was a redo log?

- Now, <T,X,v> means X's _new_ value is v
- So now when can we write each datum?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<T2,E,e>
9	<START T4>
10	<T4,F,f>
11	<T3,G,g>
12	<COMMIT T2>

Redo log problem, continued

- What do we do after the DBMS crashes and restarts?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<T2,E,e>
9	<START T4>
10	<T4,F,f>
11	<T3,G,g>
12	<COMMIT T2>

Log checkpointing

- Why would we add (non-quiescent) checkpoints to a log?
  - To avoid reading the whole log

Garcia-Molina, problem 17.2.7(i)

- Given this undo log, suppose we add a START CKPT after each of:

  2: <S,A,60>

  5: <T,A,10>

  7: <U,B,20>

  10: <U,D,40>

  13: <T,E,50>

- When is the earliest time that the END CKPT can be placed?

1	<START S>
2	<S, A, 60>
3	<COMMIT S>
4	<START T>
5	<T, A, 10>
6	<START U>
7	<U, B, 20>
8	<T, C, 30>
9	<START V>
10	<U, D, 40>
11	<V, F, 70>
12	<COMMIT U>
13	<T, E, 50>
14	<COMMIT T>
15	<V, B, 80>
16	<COMMIT V>

Checkpoints look different in undo and redo logs!

- Which is the undo log and which the redo log?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<END CKPT>
16	<COMMIT T2>
17	<COMMIT T4>

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<COMMIT T2>
16	<END CKPT>
17	<COMMIT T4>

Undo-log recovery with checkpoints

- The DBMS crashed with the previous undo log; what do we do to recover?
  - Which log entries are read?
  - Which data do we change?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<COMMIT T2>
16	<END CKPT>
17	<COMMIT T4>

Redo-log recovery with checkpoints

- Now we'd like to recover the redo log:
  - Which log entries are read?
  - Which transactions are redone?
  - Which data do we change?

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<END CKPT>
16	<COMMIT T2>
17	<COMMIT T4>

A slightly harder problem

- New notation: <OUTPUT X>
  - Write data item X to disk
- What are the legal sequences of events that consist of the entries in the following undo log, *plus* interspersed OUTPUT events?

1	<START T>
2	<T, A, 10>
3	<T, B, 20>
4	<COMMIT T>

A slightly harder problem, continued

- Can you derive a formula for the number of schedules given some number n of actions, modifying n distinct data items?
  - Hint, try listing the schedules for this 3-datum example

1	<START T>
2	<T, A, 10>
3	<T, B, 20>
4	<T, C, 30>
5	<COMMIT T>