Assignment 3

16

- Posted on Avenue and MS Teams
- Due: December 3, 2021
- Recommend you start early, don't leave it until the last minute.

16

Scheduling Transactions

17

- *Serial schedule:* Schedule that does not interleave the actions of different transactions.
- *Equivalent schedules:* For any database state, the effect (on the set of objects in the database) of executing the first schedule is identical to the effect of executing the second schedule.
- *Serializable schedule:* A schedule that is equivalent to some serial execution of the transactions.

17

Serial Schedule

18

T1	T2
R(A)	
A := A+100	
R(B)	
B := B+100	
	R(A)
	A: = A* 2
	R(B)
	B:= B*2

S: R1(A),W1(A),R1(B), W1(B), R2(A), W2(A), R2(B), W2(B)

T1 T2

Adapted from M. Balazinska

18

A Serializable Schedule

19

T1	T2
R(A)	
A := A+100	
	R(A)
	A: = A* 2
R(B)	
B := B+100	
	R(B)
	B:= B*2

Notice: this is not a serial schedule, i.e., there is interleaving of operations

Net effect is the same as the serial schedule

S: R1(A),W1(A), R2(A), W2(A), R1(B), W1(B), R2(B), W2(B)

19

A Non-Serializable Schedule

20

T1	T2
R(A)	
A := A+100	
	R(A)
	A: = A* 2
	R(B)
	B:= B*2
R(B)	
B := B+100	

What's different?

Ordering of operations is not consistent

S: R1(A),W1(A), R2(A), W2(A), R2(B), W2(B), R1(B), W1(B)

20

Conflict operations

21

- Two operations in a schedule are said to be in *conflict* if they satisfy all three of the following conditions:
  (1) They belong to different transactions
  (2) They access the same item A;
  (3) At least one of the operations is a write(A)

Example in Sa: R1(A), R2(A), W1(A), W2(A), A1, C2;

- R1(A),W2(A) conflict, so do R2(A),W1(A),
- R1(A), W1(A) do not conflict because they belong to the same transaction,
- R1(A),R2(A) do not conflict because they are both read operations.

21

Conflicts

22

What can go wrong:

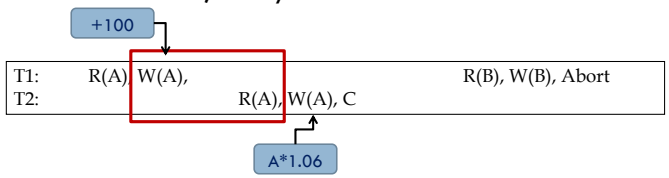
- Reading uncommitted data (WR), aka "dirty reads"
- Unrepeatable reads (RW)
- Lost updates (WW)

22

Reading Uncommitted Data

23

- WR Conflicts, "dirty reads"



- What's the problem?

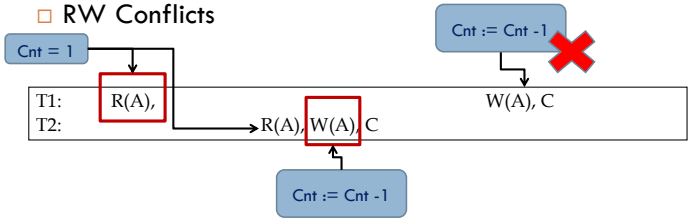
T2 has read an updated value A, which is later aborted!

23

Unrepeatable Reads

24

- RW Conflicts



- What's the problem?

T2 has changed A that has been read by T1, while T1 is still in progress

24

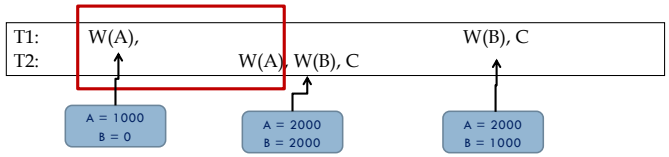
Lost Updates

25

T1: sets salaries to $1000
T2: sets salaries to $2000

Require: A and B salaries to be equal

- Overwriting uncommitted data
- WW Conflicts



- What's the problem?

T2 overwrites T1's update of A, while T1 is still in progress.

25

Schedules with Aborts

26

Withdraw $100 from A

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)
	W(B)
	Commit
Abort	

Bank reads A, B, and credits 6% interest

What now? (T2 read a value that is no longer valid)

26

Options

27

- If T2 did not commit, we abort T1 and cascade to T2.
- But T2 committed, so we cannot undo
- This schedule is *unrecoverable*

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)
	W(B)
	Commit
Abort	

27

Aborting a Transaction

28

- If a transaction *Ti* is aborted, all its actions have to be undone. Not only that, if *Tj* reads an object last written by *Ti*, *Tj* must be aborted as well!
- Most systems try to avoid such *cascading aborts*
  - If *Ti* writes an object, *Tj* can read this only after *Ti* commits.

28

Aborting a Transaction (cont'd)

29

- In order to *undo* the actions of an aborted transaction, the DBMS maintains a *log* in which every write is recorded.
- This mechanism is also used to recover from system crashes: all active Xacts at the time of the crash are aborted when the system comes back up.

29

Recoverable Schedules and Avoid Cascading Aborts

30

Recoverable

- Aborting T1 requires aborting T2
  - But T2 has already committed!
- A recoverable schedule is one in which this cannot happen.
  - i.e. a Xact commits only after all the Xacts it "depends on" (i.e. it reads from) commit.

Avoid cascading abort (ACA)

- Aborting T1 requires aborting T2!
- Aborting a Xact can be done without cascading the abort to other Xacts.
- A Xact only reads data from committed Xacts.
- ACA implies recoverable (but not vice-versa!)

No		Yes	
T1	T2	T1	T2
R(A)		R(A)	
W(A)		W(A)	
	R(A)		R (A)
	W(A)		W (A)
	Commit	Commit	
Abort			Commit

T1	T2	T1	T2
R (A)		R (A)	
W (A)		W (A)	
	R (A)	Commit	
Abort	W (A)		R (A)
			W (A)

30

Example

31

W1(X), R2(Y), R1(Y), R2(X), C2, C1

T1, T2: W1(X), R1(Y), R2(Y), R2(X)

- Serializable: Yes, equivalent to T1,T2
- Recoverable: No. Yes, if C1 and C2 are switched
- ACA: No.

  Yes, if T1 commits before T2 reads X.

31

Including Aborts in Serializability

32

- Extend the definition of a serializable schedule to include aborts
- Serializable schedule: a schedule that is equivalent to some serial execution of the set of *committed* transactions.

32

Conflict Serializable Schedules

33

- Two schedules are conflict equivalent if:
  - Involve the same actions of the same transactions
  - Every pair of conflicting actions is ordered the same way
- Schedule S is conflict serializable if S is conflict equivalent to some serial schedule

33

Recall Conflicts

34

- Two writes by Ti, Tj to same element
  - Wi(X); Wj(X)
- Read/write by Ti, Tj to same element
  - Wi(X); Rj(X)
  - Ri(X); Wj(X)

34

Conflict Equivalent

35

- Outcome of a schedule depends on the order of conflicting operations
- Can interchange non-conflicting ops without changing effect of the schedule
- If two schedules S1 and S2 are conflict equivalent then they have the same effect

  S1 ←→ S2 by swapping non-conflicting ops

35

Example Slide

36

36

Precedence Graph Test

37

Is a schedule conflict-serializable ?

Simple test:

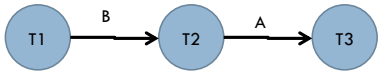
- Build a graph of all transactions Ti
- Edge from Ti to Tj if Ti comes first, and makes an action that conflicts with one of Tj
- The test: if the graph has no cycles, then it is conflict serializable !

37

Example 1

38

R2(A); R1(B); W2(A); R3(A); W1(B); W3(A); R2(B); W2(B)



This schedule is conflict serializable

38

Example Slide

39

39

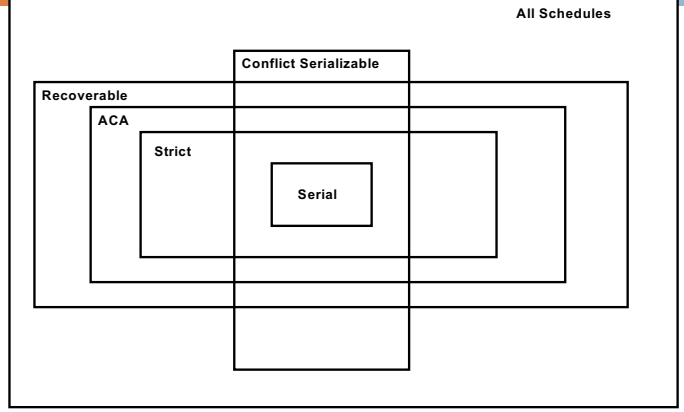
Strict Schedule

40

- A schedule S is strict if a value written by Ti is not read or overwritten by other Tj until Ti aborts or commits
- Example:
  W1(A); W1(B), C1; W2(A); R2(B); C2;
- Strict schedules are recoverable, and avoid cascading aborts.

40

Venn Diagram for Schedules



41