Tutorial 9 — Isolation and Recovery

Richard Wong
rk2wong@edu.uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

March 26, 2018

For the following transaction schedule, fill in the RW-timestamps for data items a and b, assuming we use the simple timestamp-ordering protocol.

How would the answer change if we used the Thomas Write Rule?

T_1	T_2	T_3	$TS_r(a)$	$TS_w(a)$	$TS_r(b)$	$TS_w(b)$
r_a						
	r_b					
		r_a				
w_a						
	r_a					
		w_b				
r_b						
	w_b					
		w_a				

Exercise 9-1 Solution

T_1	T_2	T_3	$TS_r(a)$	$TS_w(a)$	$TS_r(b)$	$TS_w(b)$
r_a			1			
	r_b				2	
		r_a	3			
w_a				1<3, abort T_1		
	r_a		3			
		w_b				3
r_b						
	w_b					2<3, abort T_2
		w_a		3		

If we used the Thomas Write Rule, T_2 would not need to abort just because it wanted to write a value that would have already been overwritten by T_3, and the write would have been ignored instead.

Exercise 9-2

Under what conditions does the phantom read phenomenon occur?

Exercise 9-2 Solution

Phantom reads happen when reads and writes conflict on unisolated, non-tuple data.

e.g.
Query 1: SELECT a COUNT of the number of Math professors at Waterloo.
Query 2: INSERT a Waterloo Math professor.

Transaction 1: Run Query 1 twice.
Transaction 2: Run Query 2 once.

This poses a problem for serialized isolation. The result of the SELECT will differ based on whether it runs before or after the INSERT. If the SELECTs happen on either side of the INSERT, then Transaction 1 will see inconsistent information. In serialized isolation, each transaction should be able to assume that it is the only transaction running at a time.

This is allowed to happen since we can't lock rows that do not exist (yet). We fix this with a protocol that locks index leaf nodes, under the assumption that every query uses an index.

Exercise 9-3

Suppose we need to recover from a system failure, and have the transaction log below.

Assuming we use an immediate update protocol with checkpointing, what log entries does the recovery system need to add to restore the database to a consistent state?

action	transaction	item	val	val’	flags
start	T_1				
write	T_1	a	1	2	
write	T_1	a	2	3	
checkpoint	$[T_1]$				
want to abort	T_1				
write	T_1	a	-	2	redo-only
start	T_2				
write	T_2	b	5	6	
write	T_2	b	6	7	
(system failure)					

Exercise 9-3 Solution

action	transaction	item	val	val'	flags
start	T_1				
write	T_1	a	1	2	
write	T_1	a	2	3	
checkpoint	$[T_1]$				
want to abort	T_1				
write	T_1	a	-	2	redo-only
start	T_2				
write	T_2	b	5	6	
write	T_2	b	6	7	
(system failure)					
write	T_2	b	-	6	redo-only
write	T_2	b	-	5	redo-only
abort	T_2				
write	T_1	a	-	1	redo-only
abort	T_1				

Where do the following recovery protocols belong in the table below?

1. deferred update
2. immediate update (*can* persist prior to commit)
3. strict immediate update (persist changes immediately)

	redo	no-redo
undo		
no-undo		

Exercise 9-4 Solution

	redo	no-redo
undo	immediate update	strict immediate update
no-undo	deferred update	(this is called giving up)

Exercise 9-5

What data is logged in order for the ARIES protocol to restore from a checkpoint?

Exercise 9-5 Solution

In an ARIES checkpoint, the transaction table (TT) and dirty page table (DPT) at the time of checkpoint are written to the log.

The TT needs to know the IDs of the active transactions, the last LSN associated with each transaction, and the status of each transaction.

The DPT needs to know the IDs of the dirty pages (those that require updates), and the most recent (greatest) LSN associated with each page.

Recall that LSNs are written only for:

1. write an update
2. commit transaction
3. abort transaction
4. undo an update
5. end transaction

Exercise 9-6

Suppose a checkpoint is made between LSN 7 and 8 in the following schedule.

What data is stored in the transaction table and the dirty page table?

Where should the REDO phase start scanning for operations?

Lsn	Last_lsn	Tran_id	Type	Page_id	Other_information
1	0	T_1	update	C	. . .
2	0	T_2	update	B	. . .
3	1	T_1	commit		. . .
4	begin checkpoint				
5	end checkpoint				
6	0	T_3	update	A	. . .
7	2	T_2	update	C	. . .
8	7	T_2	commit		. . .

Exercise 9-7

Why is it important for the ARIES protocol to look for the most recent *end-checkpoint* log record as opposed to the most recent *start-checkpoint* log record during its analysis phase (finding TT and DPT at last checkpoint)?

Exercise 9-7 Solution

The *end-checkpoint* log entry tells us that the TT and DPT have been fully written and that the checkpoint completed successfully. The log entry also contains the LSN of the corresponding *start-checkpoint* log entry, which marks where to start reading the TT and DPT.