CSE 444: Database Internals

Section 6:

Transactions –
OCC(Multiversion) and Undo Recovery

Review in this section

1. Multiversion concurrency control
2. UNDO logging
3. Lab 3 questions

Multiversion Concurrency Control

- Maintains **old** versions of database elements in addition the current version in the database itself.
- The idea is to allow reads that would otherwise result in an abort (as the current version was written by future transaction)

Problem with Timestamp-Based Scheduling

T1	T2	T3	T4	A
150	200	175	225	RT = 0 WT = 0
$R_1(A)$				RT = 150
$W_1(A)$				WT = 150
	$R_2(A)$			RT = 200
	$W_2(A)$			WT = 200
		$R_3(A)$		
		Abort		
			$R_4(A)$	RT = 225

Had to abort because WT(A) is greater than my own timestamp

Would have been useful if I had access to an old version of A (from 150)…

Multiversion Timestamps

T1	T2	T3	T4	A_0	A_{150}	A_{200}
150	200	175	225	RT = 0 WT = 0		
$R_1(A)$				Read		
$W_1(A)$					Create	
	$R_2(A)$				Read	
	$W_2(A)$					Create
		$R_3(A)$			Read	
			$R_4(A)$			Read

Don't have to abort

Just read a previous value of A

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0 A_1 A_2 A_3 A_4 A_5
1	2	3	4	5	
			$W_4(A)$		

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0 A_1 A_2 A_3 A_4 A_5
1	2	3	4	5	
W1(A)			W_4(A)		Create

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$									

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$								

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort									

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$						

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$							

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$										

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C										

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C					X					

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C					X					

X means that we can delete this version

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C					X					
		C								

X means that we can delete this version

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C					X					
		C				X				

X means that we can delete this version

Second Example w/ Multiversion

T_1	T_2	T_3	T_4	T_5	A_0	A_1	A_2	A_3	A_4	A_5
1	2	3	4	5						
			$W_4(A)$						Create	
W1(A)						Create				
	$R_2(A)$					RT=2				
		$R_3(A)$				RT=3				
	$W_2(A)$									
	abort			$R_5(A)$					RT=5	
				$W_5(A)$						Create
			$R_4(A)$						RT=5	
$R_1(A)$						RT=3				
C					X					
		C				X				

X means that we can delete this version

Undo Logging

- Two Rules:
  - 1. If a transaction writes element **X**, then the log record of this update **<T,X,v>** must be written to disk before the new value of **X** is written to disk.
  - 2. If a transaction commits, then the **COMMIT** must be written to disk only after all elements changed by the transaction have been written to disk.

UNDO LOG RULES

1. <T, X, v> before OUTPUT(X)
2. OUTPUT(X) before <COMMIT>

Act				Disk A	Disk B	Log
						<START T>
INPU				8	8	
READ				8	8	
t:=t*2	16	8		8	8	
WRITE(A,t)	16	16		8	8	<T,A,8>
INPUT(B)	16	16	8	8	8	
READ(B,t)	8	16	8	8	8	
t:=t*2	16	16	8	8	8	
WRITE(B,t)	16	16	16	8	8	<T,B,8>
OUTPUT(A)	16	16	16	16	8	
OUTPUT(B)	16	16	16	16	16	
COMMIT						<COMMIT T>

When recovering (with UNDO logging)…

- We can not simply ignore the log before a recent commit
  - Many transactions interleave at once. If we truncate before a commit for a transaction, any information about those unfinished transactions would be lost.

- Instead, we can use checkpoint the log periodically…

Review: Checkpointing

- **Checkpointing (naïve)**
  - Write a <START CKPT(T1,…,Tk)>. Flush log to disk
  - **Stop accepting new transactions**
  - Wait until all active transactions abort/commit
  - Write <CKPT>. Flush log to disk.
  - Resume accepting transactions

- **Nonquiescent Checkpointing**
  - Write a <START CKPT(T1,…,Tk)>. Flush log to disk
  - Continue normal operation
  - When all of T1,…,Tk have completed, write <END CKPT>. Flush log to disk
  - **More efficient, system does not seem to be stalled**

Problem 1. UNDO Logging

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	<T2 X 9>
LSN6	<START T3>
LSN7	<T3 Z 11>
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	<T2 X 13>
LSN11	<T3 Y 15>
	CRASH

1.
Show how far back in the recovery manager needs to read the log

(which LSN do we need to read up to?)

UNDO: How far to scan log from the end?

- **Case 1**: See **<END CKPT>** first
  - All incomplete transactions began after <START CKPT…>
- **Case 2:** See **<START CKPT(T1..TK)>** first
  - Incomplete transactions began after <START CKPT…> or incomplete ones among T1.. TK
  - Find the earliest <START Ti> among them
  - At most we have to go until the previous <START CKPT>…<END CKPT>

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	<T2 X 9>
LSN6	<START T3>
LSN7	<T3 Z 11>
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	<T2 X 13>
LSN11	<T3 Y 15>
	CRASH

Problem 1. UNDO Logging

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	<T2 X 9>
LSN6	<START T3>
LSN7	<T3 Z 11>
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	<T2 X 13>
LSN11	<T3 Y 15>
	CRASH

1.
Show how far back in the recovery manager needs to read the log

(write the earliest LSN)

LSN3
(start of the earliest transaction among incomplete transactions)

Problem 1. UNDO Logging

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	<T2 X 9>
LSN6	<START T3>
LSN7	<T3 Z 11>
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	<T2 X 13>
LSN11	<T3 Y 15>
	CRASH

2.
Show the actions of the recovery manager during recovery.

Problem 1. UNDO Logging

LSN1	<START T1>
LSN2	<T1 X 5>
LSN3	<START T2>
LSN4	<T1 Y 7>
LSN5	**<T2 X 9>**
LSN6	<START T3>
LSN7	**<T3 Z 11>**
LSN8	<COMMIT T1>
LSN9	<START CKPT(T2,T3)>
LSN10	**<T2 X 13>**
LSN11	**<T3 Y 15>**
	CRASH

2.
Show the actions of the recovery manager during recovery.

Y = 15
X = 13
Z = 11
X = 9