CS3223 Tutorial 9: Crash Recovery & Multigranular Locking

Wk 12, Sem 2, 2022/23

1. (Exercise 17.11, R&G) Consider a database organized in terms of the following hierarchy of objects: The database itself is an object (D), and it contains two files (F1 and F2), each of which contains 1000 pages (P1 $\cdots$ P1000 and P1001 $\cdots$ P2000, respectively). Each page contains 100 records, and records are identified as p:i, where p is the page identifier and i is the slot of the record on that page. Multiple-granularity locking is used, with S, X, IS, and IX locks, and database-level, file-level, page-level and record-level locking. For each of the following operations, indicate the sequence of lock requests that must be generated by a transaction that wants to carry out (just) these operations:
   (a) Read record P1200:5.
   (b) Read records P1200: 98 through P1205:2.
   (c) Read all (records on all) pages in file F1.
   (d) Read pages P500 through P520.
   (e) Read pages P10 through P980.
   (f) Read all pages in F1 and (based on the values read) modify 10 pages.
   (g) Delete record P1200:98. (This is a blind write.)
   (h) Delete the first record from each page. (Again, these are blind writes.)
   (i) Delete all records.

Solution:

(a) IS on D; IS on F2; IS on P1200; S on P1200:5.

(b) IS on D; IS on F2; IS on P1200; S on 1201 through 1204; IS on P1205; and S on each of the following records: P1200:98, P1200:99, P1200:100, P1205:1, and P1205:2

(c) IS on D; S on F1

(d) IS on D; IS on F1; S on P500 through P520.

(e) IS on D; S on F1. The performance hit of locking 971 pages (using IS on F1 & S on P10 through P980) is likely to be higher than performance penalty of blocking other transactions due to the lock on the entire file.

(f) IX on D; IX on F1; S on each page. If a page needs to be modified, upgrade its S-lock to X-lock.

(g) IX on D; IX on F2; X on P1200. Locking on the whole page is used (instead of IX on P1200 & X on P1200:98) as the record deletion also requires modification of the page header information and possibly page reorganization and compaction.

(h) IX on D; X on F1 and F2. There are many ways to do this with a tradeoff between overhead and concurrency.

(i) IX on D; X on F1 and F2.

2. (Exercise 18.3, R&G) Suppose the database system has just crashed with the log contents shown below. Assume that both the Dirty Page Table as well as the Transaction Table associated with the end_checkpoint_record log record are empty.

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
00	begin_checkpoint				
10	end_checkpoint				
20	update	T_1	P5		
30	update	T_2	P3		
40	commit	T_2		30	
50	end	T_2		40	
60	update	T_3	P3		
70	abort	T_1		20	

(a) Show the contents of the *Dirty Page Table* and *Transaction Table* at the end of the Analysis phase.

(b) What is the value of RedoLSN?

(c) Show all the log records that are generated by the Undo phase. For each log record, you only need to indicate the relevant information based on its type. Assume that the sequence of new log records have LSNs 80, 90, 100, etc.

Solution:

(a)

DIRTY PAGE TABLE

pageID	recLSN
P3	30
P5	20

TRANSACTION TABLE

XactID	lastLSN	status
T_1	70	U
T_3	60	U

(b) RedoLSN = 20.

(c)
1. Initially, $L = \{60, 70\}$
2. After processing LSN 70, $L = \{20, 60\}$
3. After processing LSN 60, $L = \{20\}$. Two new log records are created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
80	CLR	T_3	P3	60	
90	end	T_3		80	

T_3's entry is removed from transaction table.

4. After processing LSN 20, $L = \{\}$. Two new log records are created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
100	CLR	T_1	P5	70	
110	end	T_1		100	

T_1's entry is removed from transaction table.

3. (Exercise 18.5, R&G) Suppose the database system has just crashed with the log contents shown below. Assume that both the Dirty Page Table as well as the Transaction Table associated with the end_checkpoint_record log record are empty.

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
00	begin_checkpoint				
10	end_checkpoint				
20	update	T_1	P1		
30	update	T_2	P2		
40	update	T_3	P3		
50	commit	T_2		30	
60	update	T_3	P2	40	
70	end	T_2		50	
80	update	T_1	P5	20	
90	abort	T_3		60	

(a) Show the contents of the *Dirty Page Table* and *Transaction Table* at the end of the Analysis phase.

(b) What's the value of RedoLSN?

(c) Show all the log records that are generated by the Undo phase. For each log record, you only need to indicate the relevant information based on its type. Assume that the sequence of new log records have LSNs 100, 110, 120, etc.

Solution:

(a) 00.

(b)

DIRTY PAGE TABLE

pageID	recLSN
P1	20
P2	30
P3	40
P5	80

TRANSACTION TABLE

XactID	lastLSN	status
T_1	80	U
T_3	90	U

(c) RedoLSN = 20.

(d)

1. Initially, $L = \{80, 90\}$.
2. After processing LSN 90, $L = \{60, 80\}$.
3. After processing LSN 80, $L = \{20, 60\}$. One new log record is created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
100	CLR	T_1	P5	80	20

T_1's entry in TT is updated: lastLSN = 100.

4. After processing LSN 60, $L = \{20, 40\}$. One new log record is created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
110	CLR	T_3	P2	90	40

T_3's entry in TT is updated: lastLSN = 110.

5. After processing LSN 40, $L = \{20\}$. Two new logs record are created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
120	CLR	T_3	P3	110	
130	end	T_3		120	

T_3's entry in TT is deleted.

6. After processing LSN 20, $L = \{\}$. Two new logs record are created:

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
140	CLR	T_1	P1	100	
150	end	T_1		140	

T_1's entry in TT is deleted.

4. This question examines system recovery using the ARIES algorithm. A system failure has just occurred and the contents of the log file are shown in Figure 1(a), where for each of the log records, only the relevant information (based on the type of the log record) are indicated. The *Dirty Page Table* and *Transaction Table* associated with the *end_checkpoint* log record are shown in Figures 1(b) and (c), respectively.

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
00	update	T_3	P1		
10	update	T_3	P2	00	
20	update	T_1	P3		
30	update	T_2	P7		
40	update	T_2	P5	30	
50	update	T_1	P2	20	
60	update	T_1	P3	50	
70	begin_checkpoint				
80	end_checkpoint				
90	update	T_3	P5	10	
100	update	T_3	P4	90	
110	update	T_4	P5		
120	abort	T_3		100	
130	CLR	T_3	P4	120	90
140	update	T_1	P6	60	

(a)

DIRTY PAGE TABLE

pageID	recLSN
P3	20
P2	50

(b)

TRANSACTION TABLE

XactID	lastLSN	status
T_1	60	U
T_2	40	U
T_3	10	U

(c)

Figure 1: Log File

(a) Show the contents of the *Dirty Page Table* and *Transaction Table* at the end of the Analysis phase.

(b) What is the value of RedoLSN?

(c) In the Redo phase, to determine whether or not a redoable log record needs to be redone might require accessing the affected page. List down the LSNs of all the redoable log records that do not require a page access for this checking.

(d) Show all the log records that are generated by the Undo phase. For each log record, you only need to indicate the relevant information based on its type. Assume that the sequence of new log records have LSNs 150, 160, 170, etc.

Solution:

(a)

DIRTY PAGE TABLE

pageID	recLSN
P3	20
P2	50
P5	90
P4	100
P6	140

(b)

TRANSACTION TABLE

XactID	lastLSN	status
T_1	140	U
T_2	40	U
T_3	130	U
T_4	110	U

(c) Starting LSN for Redo phase: 20

(d) LSNs of redoable log records that not do require page access: 30 (P7 is not in the DPT), 40 (P5's recLSN > 40).

(e)

LOG

LSN	type	XactID	pageID	prevLSN	undoNextLSN
150	CLR	T_1	P6	140	60
160	CLR	T_4	P5	110	
170	end	T_4		160	
180	CLR	T_3	P5	130	10
190	CLR	T_1	P3	150	50
200	CLR	T_1	P2	190	20
210	CLR	T_2	P5	40	30
220	CLR	T_2	P7	210	
230	end	T_2		220	
240	CLR	T_1	P3	200	
250	end	T_1		240	
260	CLR	T_3	P2	180	00
270	CLR	T_3	P1	260	
280	end	T_3		270	