January 17, 2019, 10am
Arnab Chakrabarti, Rihan Hai

Implementation of Databases (WS 18/19)

Exercise 7

Due until January 29, 2019, 10am.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 7.1 (Serializability) (10 points pts)

Consider the following two transactions running concurrently:

$$T1 = r_1(A)w_1(A)r_1(B)w_1(B)$$
$$T2 = r_2(B)r_2(A)w_2(A)w_2(B)$$

Assume that only exclusive lock and unlock actions are inserted by the scheduler, resulting in the following annotated transactions:

$$T1 = wl_1(A)r_1(A)w_1(A)wl_1(B)r_1(B)w_1(B)c_1wu_1(A)wu_1(B)$$
$$T2 = wl_2(B)r_2(B)wl_2(A)r_2(A)w_2(A)w_2(B)c_2wu_2(A)wu_2(B)$$

The two transactions are executed concurrently.

1. Is conflict serializability guaranteed? Why or why not? **(1 pts)**
2. Is cascading rollback possible? If not, explain why not. **(1 pts)**
3. Is deadlock possible? **(3 pts)**
4. Explain with example what are "Dirty Reads" and "Phantom Reads" ? **(5 pts)**

Exercise 7.2 (Recovery) (6 pts)

Consider the recovery scenario described in the following. There are three transactions T1, T2, T3 updating pages A, B, C. At the time of the crash, the log contains the following entries:

LSN	LAST_LSN	TRAN_ID	TYPE	PAGE_ID
1	0	T3	update	B
2	0	T2	update	A
3	0	T1	update	C
4	2	T2	abort	
5	begin CKPT			
6	end CKPT			
7	1	T3	update	A
8	3	T1	update	C
9	8	T1	commit	

The transaction table and dirty page table for the checkpoint are:

TRANSACTION_ID	LAST_LSN	STATUS
T1	3	active
T2	2	abort
T3	1	active

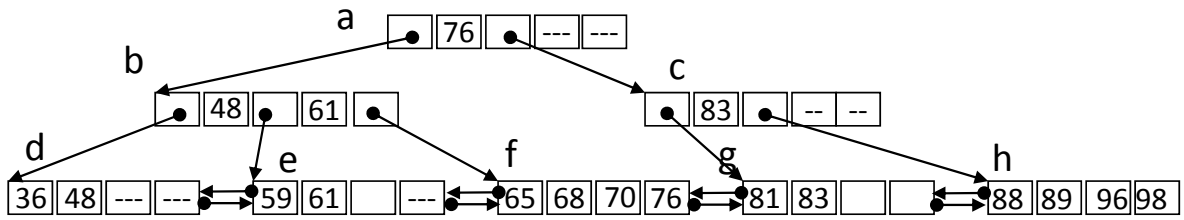
PAGE_ID	LSN
A	2
B	1
C	3

1. What is done during Analysis after the restart? Be precise about the points at which Analysis begins and ends and describe the contents of any tables constructed in this phase. **(2 pts)**

2. What is done during Redo? Be precise about the modifications to any tables due to prcoessing of log records. **(2 pts)**

3. What is done during Undo? Be precise about the modifications to the log and any tables due to processing of log records. **(2 pts)**

Exercise 7.3 (B+-tree Locking) (10 points pts)



Given the above B+-tree of degrees k=1 and k*=2, describe the steps involved in executing each of the following operations according to the simple tree locking algorithm. Redraw the result B+-tree if it is updated. Be specific about the kind of lock obtained and answer each question independently of the others, always starting with the tree above. Note that we would like to unlock a node as early as possible to maximize concurrency. We also would like to maximize throughput; i.e., releasing a higher-level node has priority over releasing a lower level node. Use the notation *rl(node), wl(node), ru(node), wu(node), r(node), w(node)* to indicate shared locking, exclusive locking, shared unlocking, exclusive unlocking, reading and writing a node respectively. Use *create(node)* to indicate creation of a new node and *delete(node)* to indicate the deletion of a node. Use *redistribute(node1, node2, node3)* to indicate redistribution of entries between siblings and *merge(child1,child2,resultNode)* (includes writing the result node) to indicate the merging of two siblings. List the actions in the order they occur, and add short explanations if necessary.

1. Search key 77 **(2 pts)**

2. Insert 82 **(3 pts)**

3. Delete 61 **(5 pts)**

Exercise 7.4 (Answer questions briefly) (4 points pts)

1. Which synchronization problem can be avoided by index locking? **(1 pts)**

2. Describe briefly the 3 phases of ARIES recovery method. What are log sequence numbers (LSNs) in ARIES? How are they used? **(3 pts)**