Gretel Rajamoney
rajamong@oregonstate.edu
933188305

Kaavya Subramanian
subramka@oregonstate.edu
933291513

1: Recovery

a. Describe the steal and no-force policies.
The steal policy is the chance of a buffer being stolen by a new transaction. The steal policy essentially implies that the rollback for a transaction requires access to pages on the disk in order to re-establish the old state in which it was in prior. The no-force policy refers to the disk pages in which the actual database objects are being modified. When utilizing a no-force policy, when a transaction commits, the modifications made to the actual objects are not "forced", meaning that it is required to be written to disk in-place. The no-force policy is considered to be in effect if when a transaction is committed, we do not need to ensure that all of the changes it has made to objects in the buffer pool are instantly forced into the disk. Most crash recoveries utilize the steal and no-force policies approach due to the fact that it accepts the risks of writing potentially uncommitted data to memory in order to gain the speed of not forcing all commit effects to memory. The steal and no-force policy allows for low input/output cost, and modified pages need not fit in the buffer pool.

2: Recovery and ARIES

a. Where in the log file does the ARIES algorithms start redoing log records in the redo phase of recovery after a crash?

The ARIES starts redoing log records at the lowest recLSN in the dirty page table constructed after the analysis phase of the ARIES algorithm. Then redo phase follows as such. In the redo pass, the ARIES essentially repeats history. This process is completed for the updates of all transactions, which includes the updates of the transactions which had missing updates as well as transactions which have never been committed. The log record's update will be redone if the impacted page's log sequence number is smaller than the log record's log sequence number. During this process, the redo pass will also obtain the locks necessary in order to protect the uncommitted updates of the transactions which have been distributed or in the in-doubt state at the end of the restart recovery phase.

b. Consider a system that uses ARIES for logging and recovery. Assume that a transaction commits during the normal execution of the database system. What log records of this transaction will be on the log file?

Log records of a transaction contain the prevLSN, transaction ID, type, pageID that is being updated, length, offset, and before and after image.bThe ARIES algorithm utilizes logs to keep record of the development of transactions as well as their moves which purpose adjustments to recoverable data objects. In a transaction log, there is a sequential document containing every single adjustment that has been made to the database, meanwhile all of the real records are contained within a separate document. Since the transaction log contains so much information, we can essentially utilize it to undo all modifications made to the facts document. The log is essentially the supply of fact and is utilized in order to ensure that all committed moves are contemplated in the database prior, and that all uncommitted movements are undone. Since the log is made up of a sequence of records, the log of each transaction is maintained within a few stable grades so that if any failures were to occur, then it can be recoverable at that point. Any operations that are performed on the database will be recorded within the log. Both the data as well as the transaction log documents are stored within the root of the database listing, and the database listing is the folder region exact from when the database was created.

3: Recovery and ARIES

a. For the actions listed above, show Transaction Table (XT) and Dirty Page Table (DPT) after each action. Assume that DPT holds pageID and recLSN, and XT contains transID and lastLSN.
At time 1

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T1	00	P7	00

At time 2 : P9 is brought into the buffer and flushed to disk, so does not enter the DPT.

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	00		

At time 3: P9 is brought into the buffer and flushed to disk, so does not enter the DPT.

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	20		

At time 4 -> begin checkpoint
At time 5 -> end checkpoint
At time 6

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	50	P9	50

At time 7

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P6	60
T1	50	P7	00
T2	60	P9	50

At time 8

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P5	70
T1	50	P6	60

T2	70	P7	00
		P9	50

At time 9

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P5	70
T1	70	P6	60
T2	60	P7	00
		P9	50

b. Simulate Analysis phase to reconstruct XT and DPT after the crash. Identify the point where the Analysis phase starts scanning log records and show XT and DPT after each action.

   We start scanning the table from the first checkpoint and reconstruct the DPT and XT at that point: (time 5)

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	20		

Scanning log forward

At time 6

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN

T0	10	P7	00
T1	20	P9	50
T2	60		

At time 7

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	20	P9	50
T2	60	P6	60

At time 8

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN
T0	10	P7	00
T1	20	P9	50
T2	60	P6	60
		P5	70

At time 9: P6 is flushed to disk.

Transaction Table (XT)		Dirty Page Table (DPT)	
transID	lastLSN	pageID	recLSN

T0	10	P7	00
T1	20	P9	50
T2	60	~~P6~~	~~60~~
		P5	70

c. Simulate Redo phase: first identify where the Redo phase starts scanning the log records. Then, for each action identify whether it needs to be redone or not.

We start at LSN 00, as that is the lowest LSN in the DPT.

Action at time 0 : P7 is in DPT and has not been written to disk. We redo this action.

Action at time 1: P9 is not in DPT. We do not redo this action.

Action at time 2: P8 is not in DPT. We do not redo this action.

Action at time 5: P9 is in DPT and is never flushed. We redo this action.

Action at time 6: P6 is in the DPT at this point, but is successfully flushed later on. We do not redo this action.

Action at time 7: P5 is in the DPT and is not written to disk. We redo this action.

Action at time 8: We do not redo this action.