CS W186 - Spring 2020
Guerrilla Section 5
Recovery and Distributed Transactions

Sunday, April 26, 2020

1 Conceptual Recovery

1. Consider a scenario where we update the recLSN in the dirty page table to reflect each update to a page, regardless of when the page was brought into the buffer pool. What bugs might you see after recovery? Select all that apply. Explain your reasoning.

    (a) Some writes of committed transactions would be lost.

    (b) Some writes of aborted transactions would be visible in the database.

    (c) The system tries to commit or abort a transaction that is not in the transaction table.

2. Suppose that you are forced to flush pages in the DPT to disk upon making a checkpoint. Which of the following cases are now guaranteed? There is one correct answer. Explain your reasoning.

    (a) We can skip one of the three phases (analysis/redo/undo) completely

    (b) We must start analysis from the beginning of the log

    (c) Redo will start at the checkpoint.

    (d) Redo must start from the beginning of the log

    (e) Undo can start at the checkpoint

    (f) Undo must run until the beginning of the log

3. If the buffer pool is large enough that uncommitted data are never forced to disk, is UNDO still necessary? How about REDO? Explain your reasoning.

4. If updates are always forced to disk when a transaction commits, is UNDO still necessary? Will ARIES perform any REDOs? Explain your reasoning.

2 Recovery Practice

For this question, you will want to have the details of the ARIES protocol handy, so we suggest you have the slides or some notes to look at while doing this question.

The year is 2029. Power outages in Berkeley are so common now that PG&E does not even send out warnings anymore - instead, they just pull the plug whenever they want.

Our database has just restarted from one such power outage. You look at the logs on disk, and this is what you see:

LSN	Record		
10	T1	update	P1
20	T2	update	P2
30	T1	update	P2
40	T1	update	P3
50		begin-checkpoint	
60	T1	update	P4
70		end-checkpoint	
80	T1	commit	
90	T2	update	P1

You load up the checkpoint and see:

Transaction Table

Txn ID	Last LSN	Txn status
T1	40	running
T2	20	running

Dirty Page Table

Page	recLSN
P1	10
P2	30

1. What is the latest LSN that this checkpoint is guaranteed to be up-to-date to?

2. What do the transaction table and dirty page table look like at the end of analysis, and what log records do we write during analysis?

3. The next phase of ARIES is redo. What LSN do we start the redo from?

4. From that record, we will redo the effects of all the following records, except we will not redo certain records. What are the LSNs of the records we do NOT redo?

5. The last phase of ARIES is undo. What do we do for this phase? Answer this question by writing out the log records that will be recorded for each step.

   Stop after you write your first CLR record (make sure your CLR record specifies the nextLSN!).

Click! The lights go out, and you realize PG&E has pulled the power yet again... during ARIES recovery no less!

Five minutes later, the power comes back online. You inspect the log, and are glad to see that **all the logp records you wrote have made it to disk**.

6. You load up the checkpoint. What does the transaction table and dirty page table look like?

7. You run the analysis phase. What do the transaction table and dirty page look like at the end of analysis?

8. You run the redo phase. In order, what are the LSNs that we redo?

9. Now we run the undo phase. What do we do? (Answer again with the log records that you have to add.)

3 Conceptual Distributed Transactions

1. For each of the following statements, mark whether it's true or false and explain your reasoning.

   (a) Resolving a deadlock at a local node will always resolve distributed deadlock.

   (b) Suppose we make the changes needed so only a majority of participants need to vote yes for a transaction to commit. Participants can write an ABORT record in phase 1 of 2PC.

   (c) Presumed abort as presented in lecture no longer works in the scenario described in 1b.

   (d) If **any** machine sees COMMIT record in its log upon recovery, the transaction will commit. Additionally, describe how to distinguish whether a machine with a COMMIT record is a coordinator or participant.

2. For the following scenarios, describe what will happen without presumed abort, then describe what will happen with presumed abort.

   (a) Participant recovers and sees just a phase 1 ABORT record

   (b) Participant recovers and sees a PREPARE record

   (c) Participant recovers and sees a PREPARE and phase 2 ABORT record

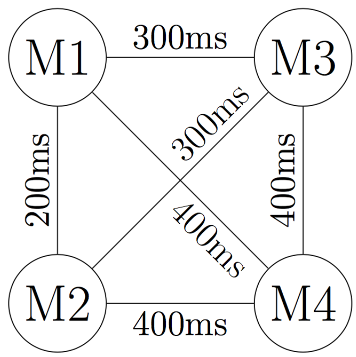
   (d) Coordinator recovers and sees an ABORT record

4 Two Phase Commit Practice (Fall 2017 Final Question 3)

Our database runs on 4 machines and uses Two-Phase Commit. Machine 1 is the Coordinator, while Machines 2, 3, and 4 are Participants.

Suppose our machines are connected such that the time it takes to send a message from Machine i to Machine j is 100·max(i,j) milliseconds (see graph below). Assume these communication latencies are symmetric: it takes the same amount of time to send from i to j as it takes to send from j to i. For example, sending a message between Machine 2 and Machine 4 takes 400 milliseconds in either direction.

Assume that the transaction will commit (i.e. all subordinates vote yes), and that everything is instantaneous except for the time spent sending messages between two machines.



1. What is the first message Machine 1 sends?
   (a) VOTE YES
   (b) PREPARE
   (c) COMMIT
   (d) None of the above

2. What is the second message Machine 1 sends?
   (a) VOTE YES
   (b) PREPARE
   (c) COMMIT
   (d) None of the above

3. How much time passes from when Machine 1 sends its first message to when Machine 1 sends its second message?

4. What is the first message Machine 2 sends?
   (a) VOTE YES
   (b) PREPARE
   (c) COMMIT
   (d) None of the above

5. What is the second message Machine 2 sends?
   (a) VOTE YES
   (b) PREPARE
   (c) COMMIT
   (d) None of the above

6. How much time passes from when Machine 2 sends its first message to when Machine 2 sends its second message?

7. True or False. A transaction is considered committed even if over half of the participants do not acknowledge the commit.

Now suppose that our implementation of 2-Phase Commit has an off-by-one bug where the Coordinator receives, but does not use, Machine 4's vote. That is, Machine 4's vote does not affect whether or not the transaction commits or aborts. Answer True or False for the following questions:

8. A transaction that should normally commit may be aborted instead.
9. A transaction that should normally abort may be committed instead.
10. A transaction that should normally commit may be committed properly.
11. A transaction that should normally abort may be aborted properly.