Braden Ackles

CS 440

Assignment #6

1. Explain the actions taken in the two phase commit protocol after restart if the coordinator or subordinate nodes fail:
   1. The recovery process will start after restarting the coordinator. When the coordinator runs into a problem and fails or restarts no transactions are committed or work saved. After the restart the recovery process will find the transaction which was executing when the server failed. After finding the transaction it will undo all the actions using the undo log records, following this it will write abort to the log.
2. Each row in the following table shows vector clocks of different copies of the same data on a system with three nodes SX, SY , and SZ. Explain if the following copies have a conflict.
   1. There will be a conflict if anything else tries to read copy1 or copy2 values. This is because they are being written by the client so if anybod tried to read them their value could not be guaranteed to be correct. It could be read before or after the write.
3. Consider transactions T1, T2, T3 with timestamps 10, 20, and 30, respectively in a database system that follows multi-version concurrency control protocol. T1 is a reader and T2 and T3 are writer transactions. Provide a schedule in which the multi-version concurrency control method restarts T2.
   1. Schedule: T2, T1, T3
   2. This will make the transaction restart because T2, T1 will read timestamp of the database and return a number greater than the timestamp of T2
   3. This is because in multi version concurrency control, writing a transaction will force a restart if the read timestamp of the database is larger than the write timestamp for that transaction
4. MongoDB (Queries):
   1. Write a query that returns the cuisine type of the restaurant named ”The Dead Rabbit” (1 point).
      1. db.restaurants.find({name: "The Dead Rabbit"}, {cuisine:1})
   2. Write a query to create an index on the name attribute of the restaurants (1.5 points).
      1. db.restaurants.createIndex({name: "text"})
   3. Write a query that uses the index created in part b to return all the restaurants containing the term “Rabbit” in their name(1.5 points).
      1. db.restaurants.find($text: {$search: "Rabbit"}})
   4. Write an aggregate query to show total count of restaurants in each borough (2 points).
      1. Db.restaurants.aggregate([{$group: {"\_id":"$borough", "count": {$sum:1}}}]);