Assignment 6

Transaction Chopping

Database Management and Tuning

Start date: May 14, 2013
Due date: May 28, 2013, 16:00
Grading: 1 point

In this assignment you will tune concurrent transactions by chopping them without trading in serializability.

1. A bank has two tables, `Account(accountID,branch,a_balance)` which stores accounts with their branch and their balance, and `Branch(branch,b_balance)` which stores the balance of each branch.

   The following types of transactions run concurrently:

   - T_1: Add money to an account and update the corresponding branch balance. No two transactions add money to the same account.
   - T_2: Read an account balance.
   - T_3: Compare the balance of each branch with the sum of the account balances in that branch.

   (a) Give the SQL queries (including pseudo code if necessary) for each transaction.

   (b) Model all transactions with read/write operations.

   (c) Show the chopping graph and give the finest possible correct chopping.

   (d) How does the chopping change if two concurrent transactions of type T_1 can update the same account? Explain.

   (e) The order of the atomic operations in T_3 has an impact on the chopping. Show two semantically equivalent implementations of T_3, one which favors chopping, the other which does not favor chopping. Explain.

2. Given the following transactions:

   - T_1: $R(a), R(b), W(b), R(e)$
   - T_2: $R(b), R(e)$
   - T_3: $R(a), W(a), R(e)$
   - T_4: $R(a), W(c)$
   - T_5: $R(c)$
   - T_6: $R(c), W(d), W(c), R(b)$

Find the finest chopping for the concurrent execution of the following transactions and show the respective chopping graphs.

(a) all transactions (i.e., $T_1, T_2, T_3, T_4, T_5, T_6$)

(b) all transactions except T_4 (i.e., T_1, T_2, T_3, T_5, T_6)

Please indicate the time that you spent solving this assignment in your report. The time that you indicate will have *no* impact on your grade.