Assignment 7

Transaction Chopping

Database Tuning

Due date: June 25, 2018, 23:55
Grading: 5 points

Notes

- It is suggested that you also have a look at the report template before you start working on the assignment.

Access Parameters for PostgreSQL

- Host: `biber.cosy.sbg.ac.at`
- Port: `5432`
- Database: `dbtuning_ss2018`
- User/Password: you should have received them via email

The database server (`biber.cosy.sbg.ac.at`) is accessible only from inside the university network. If you would like to work from home, please connect to `fanny.cosy.sbg.ac.at` via `ssh`. Java and the PostgreSQL client as well as Python are installed on this machine.

Support

If there are any ambiguities or problems of understanding regarding the assignment, you have the following possibilities to clarify them:

- Slack channel **#dbt**[1] (preferred way of communication)
- Office hours: Wednesday, 10am - 11am, Office 0.26 (ground floor)

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

1 Bank Accounts

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.

---

[1] `https://dbteaching.slack.com`

- T_2: Read an account balance.
- T_3: Compare the balance of each branch with the sum of the account balances in that branch.

Note that multiple instances of these transactions may run in parallel.

1. Give the SQL queries (including pseudo code if necessary) for each transaction.
2. Model all transactions with read/write operations.
3. Show the chopping graph and give the finest possible correct chopping.
4. How does the chopping change if two concurrent transactions of type T_1 can update the same account? Explain.
5. 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 Chopping Graphs

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.

1. all transactions (i.e., $T_1, T_2, T_3, T_4, T_5, T_6$)
2. 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.

Grading scheme:

Category	Max. Points
SQL Queries (1.1)	0.7
Model of transactions (1.2)	0.7
Chopping / finest possible (1.3)	0.7
Chopping / concurrent transactions (1.4)	0.7
Order of atomic operations (1.5)	0.7
Finest chopping (2.1)	0.75
Finest chopping (2.1)	0.75

Important: If the grading scheme is unclear, ask the lecturer!