Transaction Systems
Exercise Session 08

Linnea Passing
linnea.passing@tum.de
www-db.in.tum.de

June 13, 2016
Today’s Plan

- Admin
- MVCC: Doctors on Duty
- Multiversion Conflict Graph vs. Multiversion Serialization Graph
- Last week’s homework
- Homework
Admin

- Written exam
- Thursday, July 14, 6pm
- CH 21010 (Hans-Fischer-Hrsaal, chemistry building)
- Closed book exam
- You may bring a dictionary
Situation: In our hospital, at least one doctor has to be on duty. This is ensured by a duty table, that contains the doctors’ unique names and a duty flag. When a doctor wants to go off duty, he runs a transaction that fetches the number of doctors on duty. If two or more doctors are on duty, the transaction sets the duty flag of its doctor to “off”.

Problem: Two doctors are on duty. Both simultaneously decide to go off duty. Thus, two transactions run in our database, and have to be handled by a concurrency control module/scheduler.

Required outcome: At least one doctor has to stay no duty, i.e., it is not allowed that both transactions succeed.

What you already know: Serializable protocols ensure that at least one doctor stays on duty, while Snapshot Isolations allows both doctors to leave.
Doctors on Duty in MVCC (2/2)

All MVTO protocols mentioned in the lecture ensure that at least one doctor stays on duty.

- MVTO: First doctor that removes his duty flag (write operation) fails because a newer transaction already read the data item.
All MVTO protocols mentioned in the lecture ensure that at least one doctor stays on duty.

- MVTO: First doctor that removes his duty flag (write operation) fails because a newer transaction already read the data item.
- 2V2PL and lock upgrades: Results in a deadlock.
- MVSGT: First write introduces edge $t_2 \not\rightarrow t_1$, second write creates $t_1 \not\rightarrow t_2$ and gets aborted because of the cycle.
- ROMV: There are no read-only transactions, thus only the rules of 2V2PL are used.
All MVTO protocols mentioned in the lecture ensure that at least one doctor stays on duty.

- MVTO: First doctor that removes his duty flag (write operation) fails because a newer transaction already read the data item.
- 2V2PL and lock upgrades: Results in a deadlock.
- 2V2PL and write locks from the beginning: Transaction 1 succeeds, transaction 2 has to wait, thus sees the new value and stays on duty.
Doctors on Duty in MVCC (2/2)

All MVTO protocols mentioned in the lecture ensure that at least one doctor stays on duty.

- **MVTO**: First doctor that removes his duty flag (write operation) fails because a newer transaction already read the data item.
- **2V2PL and lock upgrades**: Results in a deadlock.
- **2V2PL and write locks from the beginning**: Transaction 1 succeeds, transaction 2 has to wait, thus sees the new value and stays on duty.
- **MVSGT**: First write introduces edge $t_2 \rightarrow t_1$, second write creates $t_1 \rightarrow t_2$ and gets aborted because of the cycle.
Doctors on Duty in MVCC (2/2)

All MVTO protocols mentioned in the lecture ensure that at least one doctor stays on duty.

- **MVTO**: First doctor that removes his duty flag (write operation) fails because a newer transaction already read the data item.
- **2V2PL and lock upgrades**: Results in a deadlock.
- **2V2PL and write locks from the beginning**: Transaction 1 succeeds, transaction 2 has to wait, thus sees the new value and stays on duty.
- **MVSGT**: First write introduces edge $t_2 \rightarrow t_1$, second write creates $t_1 \rightarrow t_2$ and gets aborted because of the cycle.
- **ROMV**: There are no read-only transaction, thus only the rules of 2V2PL are used.
Multiversion Conflict Graph vs. Multiversion Serialization Graph

Multiversion Conflict Graph
- contains rw-conflicts
- for testing MCSR (acyclic $\Leftrightarrow$ MCSR)

Multiversion Serialization Graph
- contains edges for rw- and wr-pairs and for read operations
- for testing MVSR (NP-complete because you have to find an appropriate version order)
Homework

- Already uploaded to our website.