Dynamic Adjustment of Serialization Order using Timestamp Interval in Real-Time Databases

Jan Lindström and Kimmo Raatikainen

Department of Computer Science,
University of Helsinki

Requirements for RTDBs

- Timeliness level
- Priority considerations rather than fairness
- Serializability for data consistency

Optimistic Concurrency Control (OCC)

- Read phase - normal execution
- Validation phase - check for conflicts
- Write phase - install changes

- Non-blocking
- Deadlock-free

Forward Validation

- Validating transaction is checked against other active transactions.
- Write set (WS) of validating transaction is compared to read set (RS) of active transactions.

Problems in traditional OCC-FV

- Late conflict detection
- Heavy restart overhead
- Serialization order is same as commit order
- Unnecessary restarts

Unnecessary restarts

- Major performance problem with OCC.
- Validation process erroneously concludes that a nonserializable execution has occured.
- Based on assumption that the serialization order of transactions is determined by the arriving order of the transactions at the validation phase (OCC-FV).

Key features for OCC-DATI

- Based on Forward Validation
- Write operations to private data copies
- Conflict checking in validation phase
- Serialization order may be different than commit order
- Dynamic timestamp assignment
- Dynamic adjustment of serialization order

Timestamps

- WTS(x) : timestamp of latest committed transaction writen object x.
- RTS(x) : timestamp of latest committed transaction read object x.
- TS(T_i): final timestamp of transaction T_i.

Time Intervals

- Every transaction must be executed within a specific time slot.
- Initialized as $[0,\infty[$
- Represents possible commitment time

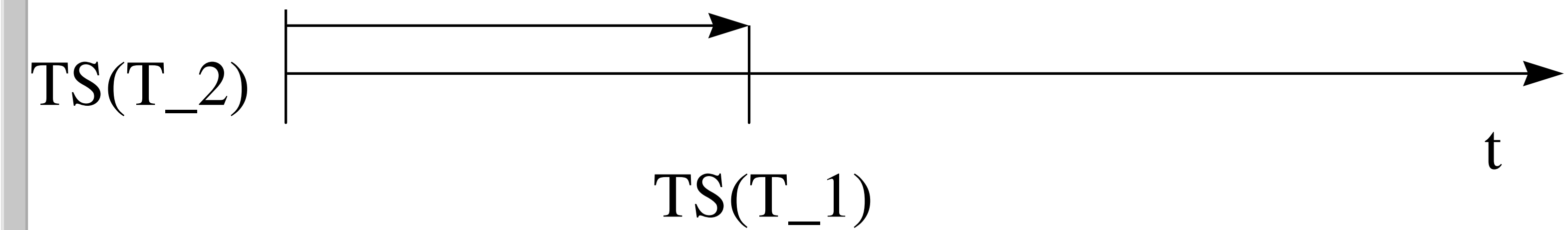
Example

History: r_2[x] w_1[x] c_1

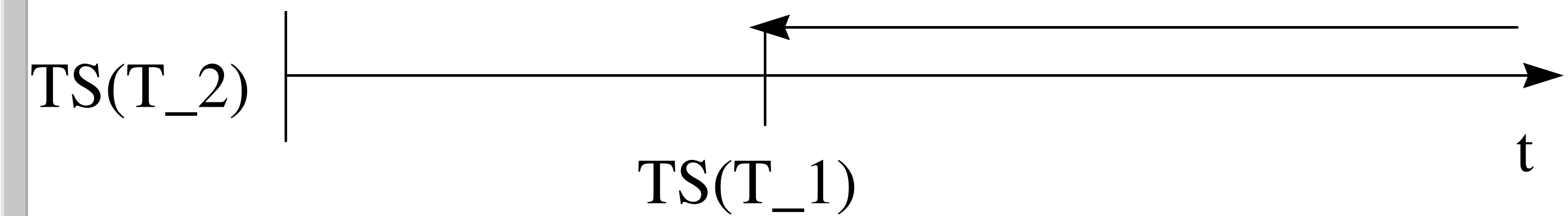
- OCC-FV will abort transaction T_2
- In 2PL w_1[x] operation will cause waiting
- In OCC-DATI both transactions are completed

Dynamic Adjustment

Forward ordering w_1[x] w_2[x] c_1



Backward ordering w_1[x] r_2[x] c_1



RODAIN = Real-Time Object-Oriented Database System for Intelligent Networks

- Real-Time, object-oriented, fault-tolerant, and distributed database management system
- Main-memory database
- Priority based scheduling
- Optimistic concurrency control

Conclusions

- OCC protocols suffer unnecessary restarts and heavy restart overhead
- OCC-DATI resolves unnecessary restart problem
- Result is less wasted resources
- Performance studies confirm that OCC-DATI is better than other OCC protocols