Transaction Systems
Exercise Session 6

Andrey Gubichev

December 9, 2013
Homework: Task 1

- $r_1(x)r_2(x)r_1(y)r_3(x)w_1(x)w_1(y)c_1r_2(y)r_3(z)w_3(z)c_3r_2(z)c_2$
Homework: Task 1

- $r_1(x) r_2(x) r_1(y) r_3(x) \omega_1(x) \omega_1(y) c_1 r_2(y) r_3(z) \omega_3(z) c_3 r_2(z) c_2$
- BOCC: $T_2, T_3$ abort
- FOCC: several options (abort $T_1$, abort $T_2$ and $T_3$, wait, ... )
Homework: Task 2

\[ w_0(x_0)w_0(y_0)w_0(z_0)c_0r_3(x_0)w_3(x_3)c_3w_1(x_1)c_1r_2(x_1)w_2(y_2)w_2(z_2)c_2 \]

▶ in fact, it is monoversion and already serial
Homework: Task 2

- \( w_0(x_0) w_0(y_0) c_0 w_1(x_1) c_1 r_3(x_1) w_3(x_3) r_2(x_1) c_3 w_2(y_2) c_2 \)
Homework: Task 2

- $w_0(x_0)w_0(y_0)c_0w_1(x_1)c_1r_3(x_1)w_3(x_3)r_2(x_1)c_3w_2(y_2)c_2$
- view equivalent monoversion schedule: $T_1 T_2 T_3$
- multiversion CG: empty. Schedule in MCSR and MVSR
Homework: Task 2

\[ w_0(x_0)w_0(y_0)c_0w_1(x_1)c_1r_2(x_1)w_2(y_2)c_2r_3(y_0)w_3(x_3)c_3 \]
Homework: Task 2

- $w_0(x_0)w_0(y_0)c_0w_1(x_1)c_1r_2(x_1)w_2(y_2)c_2r_3(y_0)w_3(x_3)c_3$
- View equivalent monoversion schedule: $T_3 \ T_1 \ T_2$
- Multiversion CG: $T_2 \rightarrow \ T_3$
How should the versions be written in an equivalent monoversion schedule?

Version order for $x$ – nonreflexive and total ordering of all versions of $x$

Multiversion Serialization Graph $\text{MVSG}(m, \ll)$: based on $\ll$
Homework: Task 3

- $m = w_0(x_0)w_0(y_0)c_0r_1(x_0)w_1(x_1)r_2(x_1)w_2(y_2)w_1(y_1)w_3(y_3)$
- $x_0 << x_1, y_0 << y_1 << y_2 << y_3$
- $T_0 \rightarrow T_1, T_1 \rightarrow T_2$
Another example of MVSR

\[ m = w_0(x_0)w_0(y_0)c_0r_1(x_0)w_1(x_1)r_2(x_1)w_2(y_2)w_1(y_1)w_3(x_3) \]

\[ x_0 << x_1 << x_3, \ y_0 << y_1 << y_2 \]

\[ T_0 \rightarrow T_1, \ T_1 \rightarrow T_2 \]

\[ T_2 \rightarrow T_3, \ T_1 \rightarrow T_3 \]
Formal definition of MVSG

- Nodes are transactions
- Edges: (consider $w_j(x_j)$, $r_k(x_j)$ and $w_i(x_i)$)
  - for $r_k(x_j)$ edge $T_j \rightarrow T_k$
  - if $x_i <<< x_j$: edge $T_i \rightarrow T_j$
  - if $x_j <<< x_i$: $T_k \rightarrow T_i$
- Edges: the order of transactions in the serial schedule
MVSR vs MCSR

MVSR
- Conflict graph $G(m)$
- Order function
- Multiversion serialization graph
- NP-complete to test

MCSR
- Multiversion conflict graph
- Test in polynomial time
Back to monoversion protocols

Hybrid protocols. Two subproblems – two components of the scheduler:

- rw (and wr) synchronization. Two operations are in conflict if one is read and another is write
- ww synchronization. Two operations are in conflict if both are writes

Example: 2PL (rw) + SGT (ww)
Hybrid protocols

To prove correctness:

- $G_{rw}(s)$, $G_{ww}(s)$ – conflict graphs
- Graphs have to be compatible: $T_i$ occurs before $T_j$ in both of them
- Then, the union of the graphs is acyclic
Hybrid protocols: example

SS2PL + TO:

- TO scheduler (ww): if \( w_i(x) \) came too late, simply ignore it (do not abort transactions) – *Thomas’ write rule*
  - \( w_1(x) r_2(y) w_2(x) w_2(y) c_2 w_1(y) c_1 \). Omit \( w_1(y) \)
  - \( w_1(x) r_2(y) w_2(x) w_2(y) c_2 r_1(y) w_1(y) c_1 \): omitting \( w_1(y) \) is not enough

- Union of two graphs is acyclic:
  - \((T_i, T_j) \in G_{rw}(s) \Rightarrow ts(T_i) < ts(T_j)\)
  - Or: if \( T_i \) commits before \( T_j \), then \( ts(T_i) < ts(T_j) \)
  - Block assignment of timestamps until the commit time.
  - Workspace concept again
Homework

- Prove: In the "no blind writes" model, where each data item written by a transaction must have been read before in the same transaction, \( \text{MCSR} = \text{MVSR} \)
- Prove: In the "action" model, where each step is a combination of a read operation immediately followed by a write operation on the same data item, \( \text{MVSR} = \text{VSR} \)
- For the schedule

\[ w_1(x)c_1r_2(x)r_3(x)c_2r_4(x)w_3(x)c_4c_3 \]

give the resulting schedule under the MVTO protocol
Consider the following schedule, given without a specific version function:

\[ r_1(x) r_2(x) r_3(y) w_2(x) w_1(y) c_1 w_2(z) w_3(z) r_3(x) c_3 r_2(y) c_2 \]

Show that it is multiversion serializable. Give a feasible version function and a feasible version order. Show the results of MVTO and 2V2PL.

For the schedule

\[ r_1(x) w_1(x) r_2(x) w_2(y) r_1(y) w_2(x) c_2 w_1(y) c_1 \]

give the resulting schedule under the 2V2PL protocol
Exercises due: 9 AM, December 16, 2012
Submit to andrey.gubichev@in.tum.de