Transaction Systems
Exercise Session 10

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

June 27, 2016
Today’s Plan

- MVCC presentation
- Admin and exam
- Homework
- Escrow locking
- MVSR serial order
- No homework!
Admin and Exam

- Post-exam Review (Klausureinsicht): July 20, from 12:30pm, at this room. You will have to register and will get assigned a specific time
- Interest in resit (Wiederholungsprüfung)?
- Remaining lectures: June 27, July 4, (July 11)
- Remaining homeworks: 2
When specific knowledge about a red class/algorithm is needed in the exam, it will be provided. Beside classes and algorithms, you also need to know about e.g. ACID properties, page and object model, dirty read problem, relationships between classes, the MVCC version function, . . .

ch. 3 FSR (thus: Herbrand semantics, reads-from, . . .)  
VSR  
CSR (thus: conflict graph, . . .)  
OCSR, COCSR, CMFSR, CMVSR, CMCSR  
Interleaving Specifications
List of Classes and Algorithm (2/3)

**ch. 4** 2PL/C2PL/S2PL/SS2PL, deadlock prevention and resolution
- O2PL
- Altruistic Locking
- write-only and read-write tree locking
- BTO
- SGT
- BOCC, FOCC
- Hybrid protocols and TWR

**ch. 5** MVSR (thus: MVSG, . . .)
- MCSR (thus: MVCG, . . .)
- MVTO
- MV2PL, 2V2PL
- MVSGT
- ROMV
List of Classes and Algorithm (3/3)

ch. 6 State-independent CT
   Return-value CT
   Commutativity-based reducibility, CSR
   Tree reducibility

ch. 7 2PL for flat object schedules
   layered 2PL, selective-layed 2PL
   Hybrid algorithms, Escrow locking

ch. 8 ... not included yet!
Homework
Escrow Locking

- Credits: Dr. Andrey Gubichev, 2013
Escrow locking

- For counter objects: there are bounds low, high and current possible value range sup, inf

- incr (x, D):
  if x.sup + D <= x.high then
    x.sup := x.sup + D; return ok
  else if x.inf + D > x.high then
    return no
  else wait fi fi;

- decr (x, D):
  if x.low <= x.inf - D then
    x.inf := x.inf - D; return ok
  else if x.low > x.sup - D then
    return no
  else wait fi fi;
Escrow locking

- When committing: adjust $inf$, $sup$
- When aborting: roll back $inf$, $sup$
- Commit of transaction $t$:
  
  for each operation $\text{incr}(x, D)$ executed by $t$ do
  
  $x.inf := x.inf + D$
  od;
  
  for each operation $\text{decr}(x, D)$ executed by $t$ do
  
  $x.sup := x.sup - D$
  od;

- Abort of transaction $t$:
  
  for each operation $\text{incr}(x, D)$ executed by $t$ do
  
  $x.sup := x.sup - D$
  od;
  
  for each operation $\text{decr}(x, D)$ executed by $t$ do
  
  $x.inf := x.inf + D$
Escrow Locking

- Credits: Dr. Andrey Gubichev, 2013
Consider two counting objects $x$ and $y$, with initial values $x = 100$ and $y = 50$. Both counters have zero as lower bound and no upper bound. Apply the escrow locking method to the following schedule of three transactions, one of which aborts:

$$\text{decr}_1(x, 60) \quad \text{incr}_2(x, 20) \quad \text{incr}_1(x, 10) \quad \text{decr}_3(x, 50)$$

$$\text{decr}_2(y, 60) \quad \text{incr}_2(x, 20) \quad a_2 \quad \text{decr}_1(y, 10) \quad c_1 \quad c_3$$
MVSR Serial Order (1/2)

\[
\begin{align*}
\text{MVSR} & \quad x_0 \ll x_2 \\
& \quad y_0 \ll y_2 \\
& \quad r_2(y) \quad r_2(y_0) \\
& \quad r_2(y_2) \quad \lambda \geq 2 (\text{i}) \\
& \quad h(r_2(y)) = \Lambda_2(y) \\
& \quad h(r_3(x)) = \Lambda_2(x) \\
& \quad h(r_2(y)) = \ldots \Lambda_n(y_2) \\
& \quad \text{MVSR} \\
& \quad t_0 + t_2 + t_3
\end{align*}
\]
Example chapter 5 slide 12: How to find this order? Why must $t_2$ come before $t_1$?
No homework!