Today’s Plan

- Exam information
- MVSR, MVTO, 2V2PL
- Layered protocols
- Hybrid protocols
- Escrow locking
- Homework
Exam Information

- 90 minutes, 90 points
- At most one proof. Other questions:
  - Apply an algorithm to a given schedule
  - Does the given schedule belong to CSR/FSR/...?
  - Construct a schedule that belongs to FSR but not CSR
  - Construct a schedule that belongs to CSR but cannot be generated by 2PL
  - Which algorithm works best in the given situation? Why?
  - ...
List of Classes and Algorithms (1/2)

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 (2/2)

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

ch. 6 (and further chapters:) not included yet
MVSR, MVTO, 2V2PL

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

- MVSR? version function, version order
- MVTO
- 2V2PL
Layered Locking

- Credits: Dr. Andrey Gubichev, 2013
Layered locking by example

- locks for high level operations
- page-level locks held only during corresponding high level operation
- this schedule would not be possible in page model
Layered locking: formally

- **Lock acquisition**: acquire $L_i$ lock on $x$ before the $f(x)$ can start at level $L_i$
- **Lock release**: once the $L_i$ lock is released, no other child of this subtransaction can get any lock
- **Subtransaction**: once the operation $f(x)$ of $L_i$ is finished, all locks at $L_{i-1}$ for the children of $f(x)$ are released
- **Modular-based lock manager**: for every level $L_i$
Layered locking: selective layered 2PL

- choose some layers, skip all the rest
- skip the layer $L_i$ by extending the scope of subtransactions above $L_i$
- less lock management overhead, but also less concurrency
Layered Locking

- Credits: Dr. Andrey Gubichev, 2013
Layered Locking

Investigate what kind of deadlocks can arise in a layered system with layered 2PL, and how they can be detected and eliminated.
Hybrid Protocols and Escrow Locking

- Credits: Dr. Andrey Gubichev, 2013
Hybrid protocols

- Idea of modularity: for every level use its own protocol
- Why? E.g., the fraction of read-only operation at a page level is higher than the fraction of read-only operations at the root level
- 2PL at object level, FOCC at the page level
- 2PL at $L_1$ and ROMV at $L_0$
Escrow locking

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

- \texttt{incr (x, D)}:
  
  \begin{verbatim}
  if x.sup + D \leq x.high then
    x.sup := x.sup + D; return ok
  else if x.inf + D > x.high then
    return no
  else wait fi fi;
  \end{verbatim}

- \texttt{decr (x, D)}:
  
  \begin{verbatim}
  if x.low \leq 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;
  \end{verbatim}
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
\[ x.\text{inf} := x.\text{inf} + D \]
end;
for each operation $\text{decr}(x, D)$ executed by $t$ do
\[ x.\text{sup} := x.\text{sup} - D \]
end;

Abort of transaction $t$:
for each operation $\text{incr}(x, D)$ executed by $t$ do
\[ x.\text{sup} := x.\text{sup} - D \]
end;
for each operation $\text{decr}(x, D)$ executed by $t$ do
\[ x.\text{inf} := x.\text{inf} + D \]
Hybrid Protocols and 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:

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

\[ \text{decr}_2(y, 60) \quad \text{incr}_2(x, 20) \quad a_2 \quad \text{decr}_1(y, 10) \quad c_1 \quad c_3 \]
Next Week’s Lecture

▶ Send questions
Homework

- Already uploaded to our website.