Rapid Review!

Introduction to Databases
CompSci 316 Fall 2022
Announcement 1/2 : Thurs (Dec 8)

• Good luck with all the project presentations tomorrow! Do not forget to take a picture with your team 😊
  • Check out the project-related posts on Ed including presentation guideline
  • Make sure to have everything as mentioned in the project document’s grading outlines

• See Transactions problem set + Announcements 2-5 on Ed
  • Practice problems
  • Important topics for final – final is comprehensive
  • Gradiance reopened for practice

• GTA and my OH will continue next week
Announcements 2/2: Thurs (Dec 8)

• Final exam is required for all (“X” grade for missing the final, https://trinity.duke.edu/undergraduate/academic-policies/end-of-term-grades)

• For emergency situations like COVID, email your Dean, Sudeepa, Alex asap
  • For COVID, please do not come to class -- we will arrange for an online exam for you at the same time, but will wait for Dean’s excuse until it can be counted
  • No make-up exam at another time unless approved by a Dean’s excuse
    • Even then, may not be able to do it after the scheduled exam due to short timeline

• Course evaluations please by 12/12 (Monday)!
  • Very important step to improve a class in the future
  • Your feedback during and after the semester matters
  • We need to hear from each of you
  • Small token of thanks (75% submissions- +2 to all in the final, 90% + 4)
  • So far 29% = 64/236
Review: 2-phase locking
Review: 2-phase locking

• Consider the following schedule with three transactions $T_1,T_2,T_3$:

$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$

Q1. Is this schedule conflict-serializable? If yes, what is an equivalent serial schedule?
Review: 2-phase locking

• Consider the following schedule with three transactions T1, T2, T3:

\[ R2(B), W2(B), R3(C), W3(C), R3(A), W3(A), C3, R2(C), W2(C), R1(A), R1(B), W1(A), W1(B), C2, C1 \]

Q1. Is this schedule conflict-serializable? If yes, what is an equivalent serial schedule?

Yes - T3, T2, T1.
Draw the precedence graph – this is the only one feasible

T3 -> T2
T3 -> T1
T2 -> T1
Review: 2-phase locking

- Consider the following schedule with three transactions $T_1, T_2, T_3$:

$$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$$

Q2. Is this schedule possible under 2PL (two-phase locking)?
Review: 2-phase locking

• Consider the following schedule with three transactions $T_1, T_2, T_3$:

$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$

Q2. Is this schedule possible under 2PL (two-phase locking)?

*Ans: Yes.*

*Sample lock/unlock*

$L_2(B), R_2(B), W_2(B), L_3(C), R_3(C), W_3(C), L_3(A), R_3(A), W_3(A), C_3, U_3(A), U_3(C), L_2(C), R_2(C), W_2(C), U_2(C), U_2(B), L_1(A), L_1(B), R_1(A), R_1(B), W_1(A), W_1(B), U_1(A), U_1(B), C_2, C_1$
Review: 2-phase locking

- Consider the following schedule with three transactions $T_1, T_2, T_3$:

$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$

Q3. Is this schedule possible under strict 2PL (two-phase locking)?
Review: 2-phase locking

• Consider the following schedule with three transactions $T_1, T_2, T_3$:

$\text{R2(B), W2(B), R3(C), W3(C), R3(A), W3(A), C3, R2(C), W2(C), R1(A), R1(B), W1(A), W1(B), C2, C1}$

Q3. Is this schedule possible under strict 2PL (two-phase locking)?

no, as $T_2$ cannot release write locks on $B$ until it commits, so $T_1$ cannot read and write $B$. 
Review: 2-phase locking

- Consider the following schedule with three transactions $T_1, T_2, T_3$:

$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$

Q4. Is this schedule recoverable?

Recoverable = Each transaction commits after all transactions from which it has read has committed.
Review: 2-phase locking

- Consider the following schedule with three transactions $T_1,T_2,T_3$:

$$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$$

Q4. Is this schedule recoverable?

Recoverable = Each transaction commits after all transactions from which it has read has committed.

**Ans:**

Yes

$T_1$ reads $B$ written by $T_2$, commits after $T_2$. $T_2$ reads $C$ written by $T_3$, commits after $T_3$. $T_1$ reads $A$ written by $T_3$, commits after $T_3$. 
Review: 2-phase locking

- Consider the following schedule with three transactions $T_1, T_2, T_3$:

$$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), C_3, R_2(C), W_2(C), R_1(A), R_1(B), W_1(A), W_1(B), C_2, C_1$$

Q5. Does this schedule avoid cascading rollback?

Avoids Cascading Rollback = Each transaction reads only data written by committed transactions
Review: 2-phase locking

• Consider the following schedule with three transactions $T1, T2, T3$:

$R2(B), W2(B), R3(C), W3(C), R3(A), W3(A), C3, R2(C), W2(C), R1(A), R1(B), W1(A), W1(B), C2, C1$

Q5. Does this schedule avoid cascading rollback?
Avoids Cascading Rollback = Each transaction reads only data written by committed transactions

Ans:
$T1$ reads $B$ written by $T2$ when $T2$ has not committed. If $T2$ aborts and rolls back, $T1$ has to roll back too.

As one of you pointed out – if the answer to this question were ”Yes”, then the answer to the recoverable question must be “Yes” – convince yourself from the definitions!
Review: Recovery
(modified from practice problem set)
Review: Recovery

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)

Tips: Mark the START/END CKPTs to notice them easily!
Q1.
Which updates are “guaranteed” to be on the disk at the time of the crash?
Which updates may or may not go?
Review: Recovery

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)

Q1.
Which updates are “guaranteed” to be on the disk at the time of the crash?
Which updates may or may not go?

Ans:
Updates in steps 3, 4, and 7
Review: Recovery

Q2.
What is the earliest (smallest) Log Sequence Number that is relevant for recovery in the Analysis phase?

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)
Q2. What is the earliest (smallest) Log Sequence Number that is relevant for recovery in the Analysis phase?

Ans: Step 8 – START CKPT of the latest END CKPT
Review: Recovery

Q3. What are the updates (in order) in the recovery for Repeating History (REDO)?

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)
Q3.
What are the updates (in order) in the recovery for Repeating History (REDO)?

Ans:
Step 9: Y <- 20
Step 12: X <- 40
Step 14: Y <- 30
Step 18: Y <- 40
Q4. What is the value of the set “U” = uncommitted transactions at the end of repeating history?
Q4. What is the value of the set “U” = uncommitted transactions at the end of repeating history?

Ans:
U = {R, V}
Built during repeating history step
Q5.
What is the earliest (smallest) Log Sequence Number that is relevant for recovery in the UNDO phase?

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)
Review: Recovery

Q5.
What is the earliest (smallest) Log Sequence Number that is relevant for recovery in the UNDO phase?

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W, V))
17. (COMMIT W)
18. (V, Y, 30, 40)

Ans:
U = {R, V}
We have to go until the earliest START of transactions in this set U.
Here it is transaction R
We have to go until Step 2.
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(START S)</td>
</tr>
<tr>
<td>2.</td>
<td>(START R)</td>
</tr>
<tr>
<td>3.</td>
<td>(R, P, 10, 15)</td>
</tr>
<tr>
<td>4.</td>
<td>(S, X, 10, 20)</td>
</tr>
<tr>
<td>5.</td>
<td>(COMMIT S)</td>
</tr>
<tr>
<td>6.</td>
<td>(START T)</td>
</tr>
<tr>
<td>7.</td>
<td>(T, X, 20, 30)</td>
</tr>
<tr>
<td>8.</td>
<td>(START CKPT(R, T))</td>
</tr>
<tr>
<td>9.</td>
<td>(T, Y, 10, 20)</td>
</tr>
<tr>
<td>10.</td>
<td>(START W)</td>
</tr>
<tr>
<td>11.</td>
<td>(COMMIT T)</td>
</tr>
<tr>
<td>12.</td>
<td>(W, X, 30, 40)</td>
</tr>
<tr>
<td>13.</td>
<td>(END CKPT)</td>
</tr>
<tr>
<td>14.</td>
<td>(W, Y, 20, 30)</td>
</tr>
<tr>
<td>15.</td>
<td>(START V)</td>
</tr>
<tr>
<td>16.</td>
<td>(START CKPT(R, W, V))</td>
</tr>
<tr>
<td>17.</td>
<td>(COMMIT W)</td>
</tr>
<tr>
<td>18.</td>
<td>(V, Y, 30, 40)</td>
</tr>
</tbody>
</table>

Q6. What are the UNDO steps performed?
Review: Recovery

Q6. What are the UNDO steps performed?

1. (START S)
2. (START R)
3. (R, P, 10, 15)
4. (S, X, 10, 20)
5. (COMMIT S)
6. (START T)
7. (T, X, 20, 30)
8. (START CKPT(R, T))
9. (T, Y, 10, 20)
10. (START W)
11. (COMMIT T)
12. (W, X, 30, 40)
13. (END CKPT)
14. (W, Y, 20, 30)
15. (START V)
16. (START CKPT(R, W,V))
17. (COMMIT W)
18. (V, Y, 30, 40)

Ans:
U = {R, V}

18: Y <- 30
3: P <- 10
### Review: Recovery

<table>
<thead>
<tr>
<th></th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(START S)</td>
</tr>
<tr>
<td>2</td>
<td>(START R)</td>
</tr>
<tr>
<td>3</td>
<td>(R, P, 10, 15)</td>
</tr>
<tr>
<td>4</td>
<td>(S, X, 10, 20)</td>
</tr>
<tr>
<td>5</td>
<td>(COMMIT S)</td>
</tr>
<tr>
<td>6</td>
<td>(START T)</td>
</tr>
<tr>
<td>7</td>
<td>(T, X, 20, 30)</td>
</tr>
<tr>
<td>8</td>
<td>(START CKPT(R, T))</td>
</tr>
<tr>
<td>9</td>
<td>(T, Y, 10, 20)</td>
</tr>
<tr>
<td>10</td>
<td>(START W)</td>
</tr>
<tr>
<td>11</td>
<td>(COMMIT T)</td>
</tr>
<tr>
<td>12</td>
<td>(W, X, 30, 40)</td>
</tr>
<tr>
<td>13</td>
<td>(END CKPT)</td>
</tr>
<tr>
<td>14</td>
<td>(W, Y, 20, 30)</td>
</tr>
<tr>
<td>15</td>
<td>(START V)</td>
</tr>
<tr>
<td>16</td>
<td>(START CKPT(R, W,V))</td>
</tr>
<tr>
<td>17</td>
<td>(COMMIT W)</td>
</tr>
<tr>
<td>18</td>
<td>(V, Y, 30, 40)</td>
</tr>
</tbody>
</table>

Q7. What are the final values of variables?
Q7. What are the final values of variables?

Ans:
Y = 30 (note: = latest value assigned by committed transactions)
P = 10 (note: = reset to initial value)
X = 40 (as set by committed transactions)
Conclusions: 1/2

• We discussed using a database system (queries), designing a database, database internals, transactions, and approaches to handling big data
• You learnt writing queries in SQL, RA, XML, JSON/Mongo
• You developed a full-stack database-backed project and had teamwork experience (you should be very proud of what you did - I am proud of you!)

• If you explore more databases, you can go in any direction you like
  • Database systems (storage, query optimization, “learned” query optimization, transactions, db/systems for ML)
  • Database theory (http://webdam.inria.fr/Alice/, https://databasetheory.org/node/140)
  • Data mining/Data science
  • Usability of databases
  • Uncertain/noisy data handling – repair, probabilistic databases
  • ….
  • Check out recent papers in database research from main conferences “SIGMOD” and “VLDB” (https://2022.sigmod.org/sigmod_research_list.shtml, and http://vldb.org/pvldb/volumes/15/)
Conclusions: 2/2

• Two data-related courses (grad seminar-level) next semester – undergraduates are welcome!
  • Prof. Jian Pei: Data Science
  • Sudeepa: Causal Inference in Data Analysis with Applications to Fairness and Explanations

• Many undergraduate students work with us and contributed significantly to our projects - If you are interested in database research or projects, we would be happy to discuss with you!

• Good luck!