Recovery

From System Crash
Types of Failure

• Catastrophic -- Physical Damage to data
  – Restore thru last backup from archived data
  – Restore to present using Log

• Non-catastrophic
  – Restore to present using Log
Logs, Buffers and Disks
Logs, Buffers and Disks

Updates written to log and buffers

Blocks read from disk
Logs, Buffers and Disks

- Blocks read from disk
- Updates written to log and buffers
- Commit written to log
Logs, Buffers and Disks

Updates written to log and buffers

Commit written to log

Blocks read from disk

Log written: transaction is committed
Logs, Buffers and Disks

Updates written to log and buffers

Commit written to log

Blocks read from disk

Log written: transaction is committed

Later
Forcewriting Log

• Why is log written first?
• Why delay in writing buffers?
• Promise can be kept after write to log
• Many changes preserved with one log write
• Other transactions can read buffers instead of disk
• No hurry to write to data disk
Forcewriting Log

Data Buffer

Data Buffer

Data Buffer

Log Buffer

Log Disk

Data Disk
Other transactions also write log buffer.
Other transactions also write log buffer

Many changes saved with one write
Other transactions use buffers instead of disk.

Other transactions also write log buffer.

Many changes saved with one write.
Other transactions use buffers instead of disk.

Popular page saved once, not many times.

Other transactions also write log buffer.

Many changes saved with one write.
Requirements on the Log

- Cannot tell from log whether data got to disk.
- After crash, state of the disk is unknown.
- After crash, all transactions are dead
- Must undo changes from uncommitted transactions
  - If uncommitted writes allowed to get to disk
- Must redo changes from committed transactions
- No cascading rollback
Note on Reads in Log

- Read information used only when cascading rollback possible
- Not used for system crash recovery
- Used only for normal processing.
Conceptual Phases of Recovery

- **ROLLBACK** -- go backward thru log
  - Note COMMITS
  - UNDO writes of uncommitted transactions
- **ROLL FORWARD** -- now go forward thru log
  - REDO writes of committed transactions
Recovery Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

Start, $T_3$

Read, $T_3$, C

Write, $T_3$, B, 15, 12 12

Start, $T_2$

Read, $T_2$, B

Write, $T_2$, B, 12, 18 18

Start, $T_1$

Read, $T_1$, A

Read, $T_1$, D

Write, $T_1$, D, 20, 25 25

Commit, $T_1$

Read, $T_2$, D

Write, $T_2$, D, 25, 26 26

Read, $T_3$, A

Write, $T_3$, A, 30, 35 35

End of Log
Recovery Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

Start, $T_3$
Read, $T_3, C$
Write, $T_3, B, 15, 12$

Start, $T_2$
Read, $T_2, B$
Write, $T_2, B, 12, 18$

Start, $T_1$
Read, $T_1, A$
Read, $T_1, D$
Write, $T_1, D, 20, 25$

Commit, $T_1$
Read, $T_2, D$
Write, $T_2, D, 25, 26$

Read, $T_3, A$
Write, $T_3, A, 30, 35$

End of Log

UNDO
Recovery Example

\[
\begin{array}{cccc}
A & B & C & D \\
30 & 15 & 40 & 20 \\
\end{array}
\]

start, \textcolor{blue}{T3}
read, \textcolor{blue}{T3}, C
write, \textcolor{blue}{T3}, B, 15, 12
start, \textcolor{blue}{T2}
read, \textcolor{blue}{T2}, B
write, \textcolor{blue}{T2}, B, 12, 18
start, \textcolor{blue}{T1}
read, \textcolor{blue}{T1}, A
read, \textcolor{blue}{T1}, D
write, \textcolor{blue}{T1}, D, 20, 25
Commit, \textcolor{blue}{T1}
read, \textcolor{blue}{T2}, D
write, \textcolor{blue}{T2}, D, 25, 26
\textcolor{red}{UND0}
read, \textcolor{red}{T3}, A
write, \textcolor{red}{T3}, A, 30, 35
\textcolor{red}{UND0}
End of Log
Recovery Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>15</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

start, \text{T3}
read, \text{T3}, C
write, \text{T3}, B, 15, 12
start, \text{T2}
read, \text{T2}, B
write, \text{T2}, B, 12, 18
start, \text{T1}
read, \text{T1}, A
read, \text{T1}, D
write, \text{T1}, D, 20, 25
Commit, \text{T1}
read, \text{T2}, D
write, \text{T2}, D, 25, 26
read, \text{T3}, A
write, \text{T3}, A, 30, 35
End of Log

Note commit
UNDO
UNDO
## Recovery Example

<table>
<thead>
<tr>
<th>Time</th>
<th>Transaction</th>
<th>Operation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>T3</td>
<td>start</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>T3</td>
<td>read</td>
<td>C</td>
</tr>
<tr>
<td>12</td>
<td>T3</td>
<td>write</td>
<td>B, 15, 12</td>
</tr>
<tr>
<td>18</td>
<td>T2</td>
<td>start</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>T2</td>
<td>read</td>
<td>B</td>
</tr>
<tr>
<td>18</td>
<td>T2</td>
<td>write</td>
<td>B, 12, 18</td>
</tr>
<tr>
<td>20</td>
<td>T1</td>
<td>start</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>T1</td>
<td>read</td>
<td>A</td>
</tr>
<tr>
<td>25</td>
<td>T1</td>
<td>read</td>
<td>D</td>
</tr>
<tr>
<td>25</td>
<td>T1</td>
<td>write</td>
<td>D, 20, 25</td>
</tr>
<tr>
<td>26</td>
<td>T1</td>
<td>Commit</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>T2</td>
<td>read</td>
<td>D</td>
</tr>
<tr>
<td>35</td>
<td>T2</td>
<td>write</td>
<td>D, 25, 26</td>
</tr>
<tr>
<td>35</td>
<td>T3</td>
<td>read</td>
<td>A</td>
</tr>
<tr>
<td>35</td>
<td>T3</td>
<td>write</td>
<td>A, 30, 35</td>
</tr>
</tbody>
</table>

**End of Log**
Recovery Example

\begin{align*}
\text{start, } & T_3 \\
\text{read, } & T_3, C \\
\text{write, } & T_3, B, 15, 12 \\
\text{start, } & T_2 \\
\text{read, } & T_2, B \\
\text{write, } & T_2, B, 12, 18 \\
\text{start, } & T_1 \\
\text{read, } & T_1, A \\
\text{read, } & T_1, D \\
\text{write, } & T_1, D, 20, 25 \\
\text{Commit, } & T_1 \\
\text{read, } & T_2, D \\
\text{write, } & T_2, D, 25, 26 \\
\text{read, } & T_3, A \\
\text{write, } & T_3, A, 30, 35 \\
\text{End of Log} & \end{align*}
Recovery Example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>T3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>T3</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td>T3</td>
<td>B, 15, 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>start</td>
<td>T2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>T2</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td>T2</td>
<td>B, 12, 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>start</td>
<td>T1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>T1</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td>T1</td>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td>T1</td>
<td>D, 20, 25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Commit</td>
<td>T1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note commit

| read  | T2 | D  |    |    |
| write | T2 | D, 25, 26 |    |    |
| read  | T3 | A  |    |    |
| write | T3 | A, 30, 35 | 35 |    |

End of Log
Note on Buffers and Disk

• UNDOING and REDOING operate on buffers read in from disk.
• Several REDOs to same data item will all be written at once
• System may crash during recovery
• Still don’t know if data got to disk so whole recovery process must proceed again.
Longer and longer Recoveries

- Say backups daily at 4 AM.
- Log in morning is small and recovery quick
- Log in evening is large and recovery slow
- Same data REDONE or UNDONE many times
- Solve with Checkpoints