Distributed Transactions
Distributed DBMS Model

- **Transactions**
- **Data Manager (DM)**
- **Transaction Manager (TM)**
- **Physical Database**

Connections:
- Transactions to Transaction Manager
- Data Manager to Physical Database
- Transaction Manager to Physical Database
- Network connection between components
Serialization

\[ T_1 : \textcolor{red}{00000} \]
\[ T_2 : \square \square \square \]
\[ T_3 : \textcolor{brown}{\Delta \Delta \Delta \Delta} \]

`log :` 
\[ \square \textcolor{red}{0} \Delta \Delta \square \square \textcolor{brown}{0} \Delta \textcolor{red}{0} \textcolor{brown}{0} \textcolor{brown}{\Delta} \]

\[ \rightarrow \text{DB} \]
Transactions

Serialization

\[ \begin{align*}
T_1 : & \quad \text{O O O O} \\
T_2 : & \quad \square \square \square \\
T_3 : & \quad \triangle \triangle \triangle \triangle
\end{align*} \]

concurrent execution

log :

\[ \square \square \triangle \triangle \square \square \text{O} \triangle \text{O} \text{O} \text{O} \triangle \]

\[ \rightarrow \text{DB} \]

OPERATIONS

\begin{align*}
\text{READ}(X) : & \quad \text{read any one copy of X} \\
R_1 (X_3) \\
\text{WRITE} (Z) : & \quad \text{write all copies of } Z \\
W_3(Z_2) \text{ and } W_3 (Z_3)
\end{align*} \]
Serialization

T₁: 〇〇〇〇
T₂: □□□□
T₃: △△△△

log: □〇△△□□〇△〇〇△

DB is acceptable if it is guaranteed to have resulted from any one of:

T₁  T₂  T₃
T₂  T₁  T₃
T₂  T₃  T₁
T₁  T₃  T₂
T₃  T₁  T₂
T₃  T₂  T₁
Serialization

Consider two concurrent transactions executed at only one DM

LOG: \[ R_1(X) \ R_2(Y) \ R_1(Y) \ W_1(Z) \ W_1(X) \ W_2(X) \ R_2(Z) \]
Serialization

Consider two concurrent transactions executed at only one DM

LOG: \( R_1(X) \) \( R_2(Y) \) \( R_1(Y) \) \( W_1(Z) \) \( W_1(X) \) \( W_2(X) \) \( R_2(Z) \)

Serial Order:

\( R_2(Y) \) \( W_2(X) \) \( R_2(Z) \) ; \( R_1(X) \) \( R_1(Y) \) \( W_1(Z) \) \( W_1(X) \)
Serialization

Consider two concurrent transactions executed at only one DM

LOG: \( R_1(X) \ R_2(Y) \ R_1(Y) \ W_1(Z) \ W_1(X) \ W_2(X) \ R_2(Z) \ R_2(Y) \ W_2(X) \ R_2(Z) \ R_1(X) \ R_1(Y) \ W_1(Z) \ W_1(X) \)

Serial Order:

\( R_2(Y) \ W_2(X) \ R_2(Z) ; \ R_1(X) \ R_1(Y) \ W_1(Z) \ W_1(X) \)

① last write conflict
② read source conflict
Serialization

Consider two concurrent transactions executed at only one DM

LOG: $R_1(X) \ R_2(Y) \ R_1(Y) \ W_1(Z) \ W_1(X) \ W_2(X) \ R_2(Z)$

Serial Order: $R_1(X) \ R_1(Y) \ W_1(Z) \ W_1(X) \ ; \ R_2(Y) \ W_2(X) \ R_2(Z)$
Transactions

Serialization

Consider two concurrent transactions executed at only one DM

LOG:

\[ R_1(X) \quad R_2(Y) \quad R_1(Y) \quad W_1(Z) \quad W_1(X) \quad W_2(X) \quad R_2(Z) \]

Serial Order:

\[ \quad R_1(X) \quad R_1(Y) \quad W_1(Z) \quad W_1(X) \quad ; \quad R_2(Y) \quad W_2(X) \quad R_2(Z) \]
Distributed Transaction Processing

Transactions:

T₁ : READ(X); WRITE(Y);
T₂ : READ(Y); WRITE(Z);
T₃ : READ(Z); WRITE(X);
Distributed Transaction Processing

Transactions:

T_1 : READ(X); WRITE(Y);
T_2 : READ(Y); WRITE(Z);
T_3 : READ(Z); WRITE(X);

LOGS:

L_1 : R_2(Y_1) R_1(X_1) W_1(Y_1) W_3(X_1)
L_2 : R_3(Z_2) W_2(Z_2) W_1(Y_2)
L_3 : W_3(X_3) W_2(Z_3)
Transactions:

T_1 : READ(X); WRITE(Y);

T_2 : READ(Y); WRITE(Z);

T_3 : READ(Z); WRITE(X);

LOGS:  
L_1 : R_2(Y_1) R_1(X_1) W_1(Y_1) W_3(X_1)

L_2 : R_3(Z_2) W_2(Z_2) W_1(Y_2)

L_3 : W_3(X_3) W_2(Z_3)

Question: Are these logs equivalent to some serial execution of the transactions?
Serialization of Distributed Logs

Conflict: $P_j(A_x)$ and $Q_i(B_y)$ conflict if

1. $P$ and $Q$ are not both READ, and
2. $A = B$ (same data item), and
3. $i \neq j$ (different transactions), and
4. $X = Y$ (same data manager/log)
Serialization of Distributed Logs

**Conflict:** \( P_j(A_x) \) and \( Q_i(B_y) \) conflict if

(1) \( P \) and \( Q \) are not both READ, and
(2) \( A = B \), and
(3) \( i \neq j \), and
(4) \( X = Y \)

**LOGS:**

\[
\begin{align*}
L_1 & : R_2(Y_1) \ R_1(X_1) \ W_1(Y_1) \ W_3(X_1) \\
L_2 & : R_3(Z_2) \ W_2(Z_2) \ W_1(Y_2) \\
L_3 & : W_3(X_3) \ W_2(Z_3)
\end{align*}
\]
Serialization of Distributed Logs

**Conflict:** \( P_j(A_X) \) and \( Q_i(B_Y) \) conflict if

1. \( P \) and \( Q \) are not both READ, and
2. \( A = B \), and
3. \( i \neq j \), and
4. \( X = Y \)

**LOGS:**

\[ L_1 : R_2(Y_1) \ R_1(X_1) \ W_1(Y_1) \ W_3(X_1) \]
\[ L_2 : R_3(Z_2) \ W_2(Z_2) \ W_1(Y_2) \]
\[ L_3 : W_3(X_3) \ W_2(Z_3) \]
Transactions

Serialization of Distributed Logs

Conflict: \( P_j(A_X) \) and \( Q_i(B_Y) \) conflict if

1. \( P \) and \( Q \) are not both READ, and
2. \( A = B \), and
3. \( i \neq j \), and
4. \( X = Y \)

LOGS:

\[\begin{align*}
L_1 & : R_2(Y_1) R_1(X_1) W_1(Y_1) W_3(X_1) \\
L_2 & : R_3(Z_2) W_2(Z_2) W_1(Y_2) \\
L_3 & : W_3(X_3) W_2(Z_3)
\end{align*}\]

Contradictory
\[\therefore \text{ No total order} \]
\[\therefore \text{ Not serializable} \]
Theorem: Distributed logs are serializable if there exists a total ordering of the transactions such that for conflicting operations \( P_j \) and \( Q_i \) a log shows \( P_j \rightarrow Q_i \) only if \( T_j \rightarrow T_i \)

**LOGS:**

\[
\begin{align*}
L_1 & : R_2(Y_1) R_1(X_1) W_1(Y_1) W_3(X_1) \\
L_2 & : R_3(Z_2) W_2(Z_2) W_1(Y_2) \\
L_3 & : W_3(X_3) W_2(Z_3)
\end{align*}
\]

Contradictory

\[\therefore \text{No total order}\]

\[\therefore \text{Not serializable}\]
Transactions

**Locking**

- transactions must use Two Phase Locking (2PL)

### Locking phase

- transactions are in a locking phase
- no locks are released
- no new locks are requested

### Release phase

- transactions are in a release phase
- no new locks are requested

- only the following lock requests are granted

<table>
<thead>
<tr>
<th>lock request</th>
<th>current lock state</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock request</td>
<td>not locked</td>
</tr>
<tr>
<td>READ</td>
<td>OK</td>
</tr>
<tr>
<td>WRITE</td>
<td>OK</td>
</tr>
</tbody>
</table>
Transactions

Locking

- request lock before accessing a data item
- release all locks at the end of transaction

This guarantees serializability [ESWAREN]
Effects of Locking

Suppose the transactions have executed to this point:

$L_1 : R_2(Y_1)$
$L_2 : R_3(Z_2)$
$L_3 : W_3(X_3)$

The locks are then:

<table>
<thead>
<tr>
<th>Lock for</th>
<th>Lock state</th>
<th>Waiting for lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>write-locked by $T_3$</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>read-locked by $T_2$</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>read-locked by $T_3$</td>
<td>$T_2$</td>
</tr>
</tbody>
</table>

Only $T_3$ is able to execute; it will complete its write and release its locks, leading to …
Effects of Locking

L₁ : R₂(Y₁), W₃(X₁)
L₂ : R₃(Z₂)
L₃ : W₃(X₃)

...these logs, and...

...this lock state:

<table>
<thead>
<tr>
<th>Lock for</th>
<th>Lock state</th>
<th>Waiting for lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>unlocked</td>
<td>T₁</td>
</tr>
<tr>
<td>Y</td>
<td>read-locked by T₂</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>unlocked</td>
<td>T₂</td>
</tr>
</tbody>
</table>

Both T₁ and T₂ can acquire the locks needed to execute their next actions, resulting in...
Transactions

Effects of Locking

\[ L_1 : R_2(Y_1), W_3(X_1), R_1(X_1) \]

...these logs, and...

\[ L_2 : R_3(Z_2), W_2(Z_2) \]

\[ L_3 : W_3(X_3), W_2(Z_3) \]

...this lock state:

<table>
<thead>
<tr>
<th>Lock for</th>
<th>Lock state</th>
<th>Waiting for lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>read-locked by T_1</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>read-locked by T_2</td>
<td>T_1</td>
</tr>
<tr>
<td>Z</td>
<td>write-locked by T_2</td>
<td></td>
</tr>
</tbody>
</table>

Transaction T_2 completes, releases its lock - resulting in...
Transactions

Effects of Locking

...this lock state:

<table>
<thead>
<tr>
<th>Lock for</th>
<th>Lock state</th>
<th>Waiting for lock</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>read-locked by T₁</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>unlocked</td>
<td>T₁</td>
</tr>
<tr>
<td>Z</td>
<td>unlocked</td>
<td></td>
</tr>
</tbody>
</table>

At this point, T₁ can acquire the write-lock on Y, perform its write operations and complete, leading to the final serializable logs:

\[
\begin{align*}
L₁ : & \text{R}_2(Y₁), \text{W}_3(X₁), \text{R}_1(X₁), \text{W}_1(Y₁) \\
L₂ : & \text{R}_3(Z₂), \text{W}_2(Z₂), \text{W}_1(Y₂) \\
L₃ : & \text{W}_3(X₃), \text{W}_2(Z₃)
\end{align*}
\]