Distributed DBMS Model

Transactions

Data Manager

Logical database

Network

Physical database

Transaction Manager

Serialization

T₁: O O O O
T₂: □ □ □
T₃: △△△△

Concurrent execution

Log:

DB

OPERATIONS

READ(X): read any one copy of X

WRITE(Z): write all copies of Z

W₁(Z₂) and W₂(Z₂)

Consider two concurrent transactions executed at only one DM

LOG:

R₁(X) R₂(Y) R₁(Y) W₁(Z) W₁(X) W₂(X) R₂(Z)

Serialization

Consider two concurrent transactions executed at only one DM

LOG:

R₁(X) R₂(Y) R₁(Y) W₁(Z) W₁(X) W₂(X) R₂(Z)

Serial Order:

R₂(Y) W₁(X) R₂(Z) ; R₁(X) R₁(Y) W₁(Z) W₁(X)
Serialization
Consider two concurrent transactions executed at only one DM

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

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

Distributed Transaction Processing
Transactions:
\( T_1: \) READ(X); WRITE(X);
\( T_2: \) READ(Y); WRITE(Z);
\( T_3: \) READ(Z); WRITE(X);

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

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. \( A = B \)
3. \( i \neq j \)
4. \( X = Y \)

**LOGS:**

1. \( L_1 : R_2(Y_1) R_1(X_1) W_3(Y_1) W_3(X_1) \)
2. \( L_2 : R_3(Z_2) W_4(Z_2) W_4(Y_2) \)
3. \( L_3 : W_3(X_3) W_3(Z_3) \)

\[ \begin{align*}
\text{Contradictory} \\
\text{No total order} \\
\text{Not serializable}
\end{align*} \]

Theorem: Distributed logs are serializable if there exists a total ordering of the transactions such that for conflicting operations \( P_j \) and \( Q_i \) in a LOG only if \( T_j \rightarrow T_i \)

**LOGS:**

1. \( L_1 : R_2(Y_1) R_1(X_1) W_3(Y_1) W_3(X_1) \)
2. \( L_2 : R_3(Z_2) W_4(Z_2) W_4(Y_2) \)
3. \( L_3 : W_3(X_3) W_3(Z_3) \)

\[ \begin{align*}
\text{Contradictory} \\
\text{No total order} \\
\text{Not serializable}
\end{align*} \]

Locking

- transactions must use Two Phase Locking (2PL)
- only the following lock requests are granted

<table>
<thead>
<tr>
<th>lock request</th>
<th>current lock state</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>OK</td>
</tr>
<tr>
<td>WRITE</td>
<td>OK</td>
</tr>
<tr>
<td></td>
<td>DENY</td>
</tr>
</tbody>
</table>

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

This guarantees serializability [ESWAREN]