Evaluation Criterion

1. **Degree of Concurrency**

   Less reshuffle $\Rightarrow$ High degree of concurrency

2. **Resources used to recognize**
   - Lock tables
   - Time stamps
   - Read/write sets
   - Complexity

3. **Costs**
   - Programming ease
General Comments

- Information needed by Concurrency Controllers
  - Locks on database objects (System-R, Ingres, Rosenkrantz...)
  - Time stamps on database objects (Thomsa, Reed)
  - Time stamps on transactions (Kung, SDD-1, Schlageter, Bhargava...)

- Observations
  - Time stamps mechanisms more fundamental than locking
  - Time stamps carry more information
  - Checking locks costs less than checking time stamps
General Comments (cont.)

- **When to synchronize**
  - First access to an object (Locking, pessimistic validation)
  - At each access (question of granularity)
  - After all accesses and before commitment (optimistic validation)

- **Fundamental notions**
  - Rollback
  - Identification of useless transactions
  - Delaying commit point
  - Semantics of transactions
Probability that two transactions do not share an object

\[\frac{MC_{BS} * (M - BS)}{M} \cdot (M - BS - 1) \cdot \left(\frac{M - 2BS + 1}{M - BS + 1}\right)\]

Lower bound on this problem

\[=\left(\frac{M - 2BS + 1}{M - BS + 1}\right)^{BS}\]

Maximum problem that two transactions will share an object

\[=1 - \left(\frac{M - 2BS + 1}{M - BS + 1}\right)^{BS}\]

<table>
<thead>
<tr>
<th>BS</th>
<th>M</th>
<th>Probability of conflict</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>100</td>
<td>.0576</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>.0025</td>
</tr>
<tr>
<td>20</td>
<td>1000</td>
<td>.113</td>
</tr>
</tbody>
</table>

Probability of cycle

\[= 0(PC^2)\]

\[\approx \text{small}\]
Concurrency/Multiprogramming level is low

Example:

\[
\begin{align*}
\text{I/O} & = 0.005 \text{ seconds} \\
\text{CPU} & = 0.0001 \text{ seconds} \\
\text{Trans size} & = 5 \\
\text{Time to execute trans.} & = 0.0255 \text{ seconds}
\end{align*}
\]

For another trans. to meet this trans. in the system

\[
\text{Arrival rate} > \frac{1}{0.0255} \quad \text{or} \quad > 40 \text{ per second}
\]
Example: \( h = R_1 R_2 W_2 R_3 W_3 \ldots R_n W_n W_1 \)

\[
\begin{array}{ccccccc}
R_1 & R_2 & R_3 & \cdots & R_n & W_n & W_1 \\
W_1 & W_2 & W_3 & \cdots & \end{array}
\]

Locking: This history not allowed

- \( W_2 \) is blocked by \( R_1 \)
- \( T_2 \) cannot finish before \( T_1 \)

What if \( T_1 \) is a log trans. and \( T_2 \) is a small trans.?

- \( T_1 \) blocks \( T_2 \); can block \( T_3 \) \ldots \( T_n \) if \( R_2 \cap W_2 \neq \emptyset \)

Optimistic [Kung]

- \( T_i \) \((i = 2, \ldots, n)\) commit. \( W_i \) saved for \( valid_n \)
- \( R_1 \) validated with \( W_i \), \( T_1 \) aborted

\[
h = R_1 R_2 W_2 \ldots R_n W_n W_1
\]

switch to
Optimistic Validation (first modification)

\[ h = R_1 \ R_2 \ W_2 \ R_3 \ W_3 \ldots \ R_n \ W_n \ W_1 \]

Try this or this switch

T_i’s can commit, W_i and R_i saved from validation
W_1 validates with W_i and R_i
T_1 aborted if validation fails (second modification)

\[ h = R_1 \ R_2 \ W_2 \ R_3 \ W_3 \ldots \ R_n \ W_n \ W_1 \]

Switch R_1 to the right after W_2, W_3…W_n
Switch W_1 to the left before R_n, R_{n-1}…R_2
If R_1 and W_1 are adjacent, T_1 is successful

\[ h \equiv R_1 \ R_2 \ W_2 \ldots \ R_k \ W_k \ldots \ R_n \ W_n \ W_1 \]

\[ \equiv R_2 \ W_2 \ldots \ R_1 \ W_1 \ R_k \ W_k \ldots \ R_n \ W_n \]