STUDY OF CONCURRENCY MECHANISM

Kuldeep Kumar
Department of Computer Science, Choudhary Devi Lal University, Sirsa, India
kuldeepkumarfatehabad@gmail.com

Abstract—In client server environment where multiple clients request parallel read write operation on same database, then data inconsistency may occur. Concurrency control is mechanism which provides solution to above mentioned problem. This paper gives detailed study of Concurrency control mechanism.

Index Terms—Concurrency, shared exclusive lock, interleaving, time stamp.

I. INTRODUCTION

To ensure serialization in shared database concurrency control may be used which is nothing but a process of managing simultaneous execution of transactions. Database when used simultaneously by more than one clients shows a lot problems. The study of such problem is included in the beginning sections of the paper.

II. TRANSACTION DEFINED

Transaction means action between client and server. In other words when any client pass any query to server for any read or write operation. The server accept the client query and provide data to client.

It is upto the nature of request made how the database will handle the request in hand.

III. TYPES OF TRANSACTION

In any database there are two types of transaction in client–server architecture.

1. Serial Transaction.
2. Concurrent Transaction.

In serial transaction if more than one client request data from server, then server will provide the data serially i.e. one by one. It is shown in FIG 1. Here transactions are taken one by one and are handled serially by the given server. Though the mechanism is simple yet it may not obtain the maximum utilization of available resources. Point to be noted is that maximum utilization at minimum cost of available resources is of prime concern in such an environment.

Fig 2 shows Parallel requests coming from multiple clients to the same server. Being efficient and effective this method is more in use yet it suffers some problems too.

FIG 1: Serial Transaction

FIG 2: Parallel Transaction

There are two operations:-

1. READ operation.
2. WRITE operation.

READ Transaction :- When client request any data from server for read purpose then it is called read...
transaction. For example in railway reservation system when any client wants to know how many seats are available at a specific time then he perform read transaction. Server provides data to client for read only not for any modification.

**FIG 3: Read Operation**

**WRITE Transaction**: When client request any data from server for write purpose then it is called write transaction. For example in railway reservation system when any client wants to book seats at a specific time then he perform write transaction. Server provides data to client for modification.

**FIG 4: Write Operation**

**IV. PROBLEMS OF CONCURRENT ACCESS**

When data is accessed in concurrent manner then there is problem of inconsistency of data. There are basically three main problem occur during the concurrent access of data.

1. **Lost Update Problem.**
2. **Dirty Read Problem.**
3. **Incorrect Summary Problem.**

**A. Lost Update Problem:**

When two client are requesting for same data at same time for write operation and sever provide them. But problem is that one client update data and commits it. But this updated data is not provided to 2nd client. So this updation is lost for 2nd client. This is called lost update problem.

**B. Dirty Read Problem**

When two client request for data at same time for write operation. Let us server has two variables with values as X=10, Y=20. Client -A want to subtract 5 from X and Add 5 into Y. Client -B tries to add 2 into X. Both request are triggered at same time. Server process these into concurrent manner. Client –A read X =10 and subtract 5 from X and send X=5 to sever at the same time Client-B read X=5 from server and add 2 into X. Now X=7 but on the other hand client –A rollback the transaction. In this situation client-B read data that is dirty.

**C. Incorrect Summary Problem:**

When two clients request for data at same time for write operation. Server has two variables X=10, Y=20. Client-A want to add X and Y. But Client-B add 5 into X both requests are put at the same time. If the request of client-A is executed first then X+Y=30 on the other hand if client-B’s request is performed first then X+Y=35. This type of problem is called incorrect summary problem.

**TABLE 1: CONCURRENCY PROBLEMS**

<table>
<thead>
<tr>
<th>Problems</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lost Update Problem</td>
<td>When two or more clients read data simultaneously and one client update it then second one will not able to detect the changes in common data</td>
</tr>
<tr>
<td>Dirty Read Problem</td>
<td>This problem also exists in environment where common data is processed by two clients in parallel manner.</td>
</tr>
<tr>
<td>Incorrect Summary Problem</td>
<td>Let One client want to add value of two variables whereas second client want to multiply one variable by some int. If two such transactions occur concurrently Incorrect problem arises.</td>
</tr>
</tbody>
</table>

**D. Binary Lock Method:**

In Binary lock method there is a binary lock on the sever side. Binary lock have possible two values 0 and 1. 0 represent the unlock and 1 represent lock. In this locking technique when any client request for data initially server check the lock value ‘if’ lock value is 0 then server put lock value 1 and provide data to client on the other hand if lock value is 1 then client in waiting mode.
E. Shared and Exclusive read/write

In this method there are two locks on server site one lock is called shared(READ) and other is called exclusive(WRITE) lock. If first request is of read nature then read lock value is set to 1, now if second request is also of read nature then server allow it but if the second request is of write nature then it will not be allowed.

On the other hand if first request is write request then write lock is set to 1 and second request will not be allowed at all (it may be read or write).

F. Two Phase Locking

As the name itself indicates this method contain two phases i.e. a growing phase and a shrinking phases. 

In the growing phases a transaction gets all the required locks without unlocking any data. After the acquisition of all the locks, the second phases starts.

The second phases is shrinking phases in which transaction releases all the locks and can not attain any new lock. It is shows in FIG.

References


<table>
<thead>
<tr>
<th>Method name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary lock</td>
<td>A lock with two binary values is applied here.</td>
</tr>
<tr>
<td>Shared/exclusive</td>
<td>Read lock may allow concurrent reading by another client. But Write lock is exclusive one.</td>
</tr>
<tr>
<td>Two-phase locking</td>
<td>Two Phases Growing and Shrinking are used.</td>
</tr>
</tbody>
</table>

TABLE II: CONCURRENCY CONTROL METHODS