Timestamp Ordering Concurrency Control for Hotel Reservation System

Thin Thin Nwe; Khaing
University of Computer Studies, Yangon
thinthinnwe84@gmail.com;khaingaugust@gmail.com;

Abstract

A distributed database system is a database system that provides commands to read and write data that is stored at multiple site of network. If users access a database concurrency, they may interfere with each other by attempting to read or write the same data items. The goal of concurrency control is to prevent interference among users who are simultaneously accessing a database. Concurrency control ensures the consistency and reliability properties of transactions.

Conservative timestamp ordering is a technique for elimination restarts during time stamp scheduling. When a scheduler receives an operation; that might cause a future restarts; the scheduler delays operation until it is sure that no future restarts are possible. This system implements the on line hotel reservation system using time stamp ordering concurrency control. This system intends to avoid the several restart of the conflicting transactions and to avoid the dead lock operation and to facilitate the communication between the user and the web base hotel reservation system.

Keyword: Distributed Database System, timestamp, concurrency control.

1. Introduction

Now a day, organizations and multiple users need application able to coordinate activities and integrate information through a distributed network of participants. Most of the business applications are in the form of Distributed database applications. The number of users who access database applications has growing so fast and it becomes the highest priority in database fields to handle transaction of users. Businesses are beginning to rely on distributed rather than centralized computing; there is need to co0rdinate the incidental tasks by distributing the parts of process between the participants. Several systems architecture have been proposed and implemented to support and enhanced coordination and integration activities into distributed and dynamic environments. Distributed system techniques are more interest in the recent years due to the proliferation of the web and other online system and services.

The internet can be said to be a "network of networks". The internet is a network on a worldwide scale that is made up of large and small interconnected networks. The internet and World Wide Web can give huge amount of information. Several users can access the information from the different places at the same time. So the concurrency control for the distributed database system is the need to give the users reliability of the information.

2. Theory Background

A distributed database system is a database system that provides commands to read and write data that is stored at multiple site of network. If users access a database concurrency, they may interfere with each other by attempting to read or write the same data items. A distributed Database Management system is the software system that permits the management of the distributed database and makes the distribution transparent to the users.

Concurrency control is the activity of coordinating concurrent accesses to a database in a multi-user database management system. The goal of concurrency control is to prevent interference among users who are simultaneously accessing a database. Concurrency control ensures the consistency and reliability properties of transactions. Process of managing simultaneous execution of transaction in a shared database, to ensure the serializability of transactions, is known as concurrency control.

Concurrency control protocols generally are classified into two categories, pessimistic and optimistic. Pessimistic protocols prevent inconsistencies by disallowing potentially non-serilizable executions and ensuring that the effect of committed transactions need not be reversed or annulled. Pessimistic approached are often implemented using locks to control access to data items. A classical example of pessimistic protocol is the two-phase locking (2PL) protocol, which is widely implemented in both commercial as well as research prototype system.

The pessimistic group consists of locking based algorithms, ordering (transaction ordering) based algorithms, and hybrid algorithms. The optimistic group can be classified as locking based or timestamp ordering based. This classification is depicted in Figure 1.

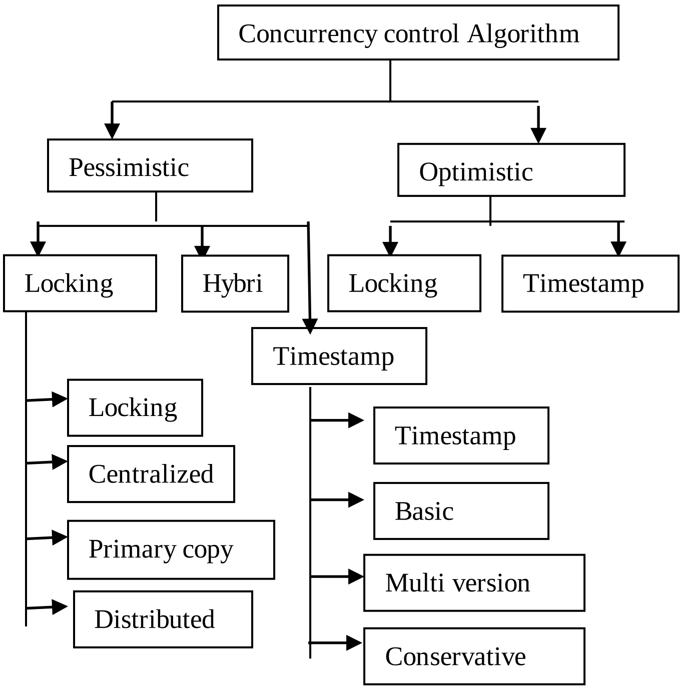

Figure 1. Type of concurrency control algorithm

Optimistic protocols, on the other hand, permit non- serializable executions to occur based on the assumption that no consistency will result due to the conflicting transactions. When such violation of inconsistency is detected during a validation phase prior to the commit of a transaction, the offending transactions are aborted. An example of the optimistic approach is the timestamp-based approach in which the timestamp of the data items accessed the transactions are checked at commit time to see if they have been modified by the other transactions since they first accessed by the transaction. If that is the case, then the transaction is aborted.

2.1 Timestamp Ordering

A timestamp is a simple identifies each transaction uniquely and to permit ordering. In timestamp ordering each transaction is assigned a globally unique timestamp by its transaction manager.

In a single server transaction, the coordinator issues a unique timestamp to each transaction when it starts. Serial equivalence is enforced by committing the versions of objects in the order of timestamp of transactions that accessed them. In distributed transactions, we require that each coordinator issues to the client by the first coordinator accessed by transaction. The transaction timestamp is passed to the coordinator at each server whose objects perform and operation in the transaction. The server of distributed transactions are jointly responsible for ensuring that they are performed in a serially equivalent manner.

Timestamp are used to avoid deadlock. For each transaction has a single timestamp. Timestamps are used to resolve conflicts between transactions.

Timestamp-based concurrency control involves using unique transaction timestamps in place of conventional locks. Concurrency control is based on the ordering of timestamps. So, for example, when a transaction accesses an item, the system checks whether this transaction is older than the last one which accessed the item. If this is the case the transaction proceeds; otherwise ordering is violated and the transaction is aborted. Such strict timestamp-based approaches can lead to the cyclic restart of transactions and starvation.

Conservative timestamp ordering is a technique for elimination restarts during time stamp scheduling. When a scheduler receives an operation; that might cause a future restarts; the scheduler delays operation until it is sure that no future restarts are possible. When the operation is received from a transaction manager, it is placed in its appropriate queue (buffered) in increasing time stamp order. The scheduler will choose an operation with the smallest timestamp and pass if on to the data processor. The operations of each transaction are buffered until an ordering can be established so that rejections are not possible.

Conservative Timestamp performs RW synchronization as follows:

Let: TS(Qi) denote the timestamp of the first operation in Qi
Let ROi is a read queue and WOi is a write queue
Let TS denotes the time stamp
read <object,TS>
if (non-empty(WQi) and TS(WQi) > TS for i = 1 ….N)
then execute the read operation
else add the read operation to RQi
write <object, val, TS>
if (non-empty (RQi) and non-empty (WQi) and TS(RQi) > TS for i = 1….N)
then execute the write operation
else add the write operation to WQi

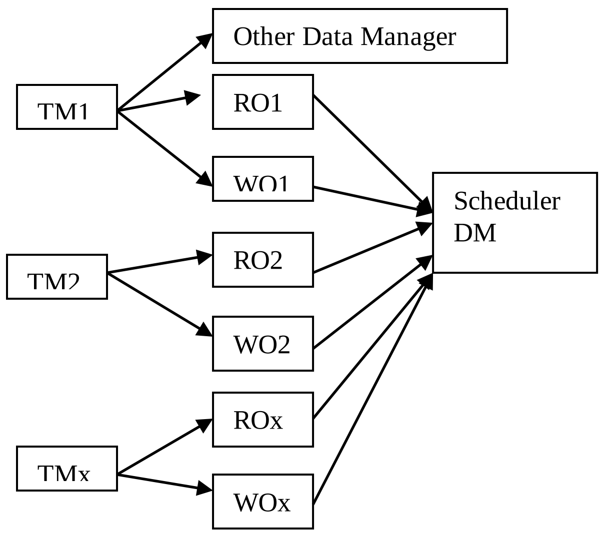

Figure 2. Conservative timestamp Ordering procedure

Figure 2 shows the conservative timestamp ordering procedure. When users request hotel reservation, transaction manager (TMi) defines a unique timestamp for each transaction and scheduler one write operation (for reservation), other (they make same reservation) may wait for a limited time because the data item (reservation room) is lock by data manager (Scheduler DM). Each transaction manager takes read operation (ROi) or Write operation(WOi).

3. System Design

To implement the hotel reservation system, Window XP professional and Window Vista are used as the operation system. Internet Information Services (ISS) provides the services necessary to publish information. Microsoft SQL Server is used for Database retrieving required data. Active Server Page (ASP) is used for browsing the resulted data in the proposed system.

In this system, there are two users involved. They are users and admin. Users can reserve hotel reservation for their required hotel. All of the user can view the hotel information and make the reservation. Users can confirm the hotel reservation of required hotel. Admin user can manage the hotel information, hotel type and service.

Figure 3 shows the process flow of the system. Any user can read hotel information. If user wants to reserve the hotel room, user can request reservation. If the required room of the required hotel is available, the user can reserve this room. If the preorder transaction has occurs, the next user will wait in Queue. When the user 1 reserve the room, another user 2 will wait in the buffer until the end of the specified time and the concurrent user3 will wait in buffer another specified time for the event handler. Waiting in buffer can increase the time rely the time events.

4. System Implementation

In the proposed system, the several users can access the hotel reservation system at the same time from the different places. This system uses the conservative time stamp ordering concurrency control. So the user no need much knowledge of the physical database design and can perform reservation easily for web base hotel reservation easily through the user interface. If the user wants to see the hotel information, to system shows the entire information. After the user views the hotel information, user wants to reserve the hotel room to enter the reservation section. In the reservation section, users can select the required hotel, room type, required date and services. For concurrency control, another user can enquire the hotel information of the same hotel. If the user sure to confirm the hotel, he/she confirm the hotel room number for specified type and will show the detail information.

While the user 1 reserve the room, another user 2 will wait in the buffer until the end of the specified time limit. After the user1 reserve the room, the schedule will be changed and the database will be update. After waiting specified time in buffer, the user 2 will receive the message that the schedule will change.

For admin user, there are three type of schedule can drew.

1. Hotel type
2. Hotel information and
3. Service.

Admin user can add the new hotel type, hotel information and schedule and can delete the any items. Admin level can change the available room of each hotel and the available services.

5. Advantage of the system

There are many advantage of the system. They are:

- To avoid deadlock operation and unnecessary transaction abort and restart
- To control concurrency control
- To get best performance of hotel reservation
- To facilitate the communication between the user and the web base hotel reservation system.

6. Conclusion

Many distributed system application are widely use in still and future. Concurrency control is attractive problem in this application area. Timestamp ordering algorithm is free deadlock but it may cause the operations to several restarts. It may lead to overhead cost and least throughput. To overcome these difficulties we proposed conservative timestamp ordering algorithm to control concurrent access operations.

N

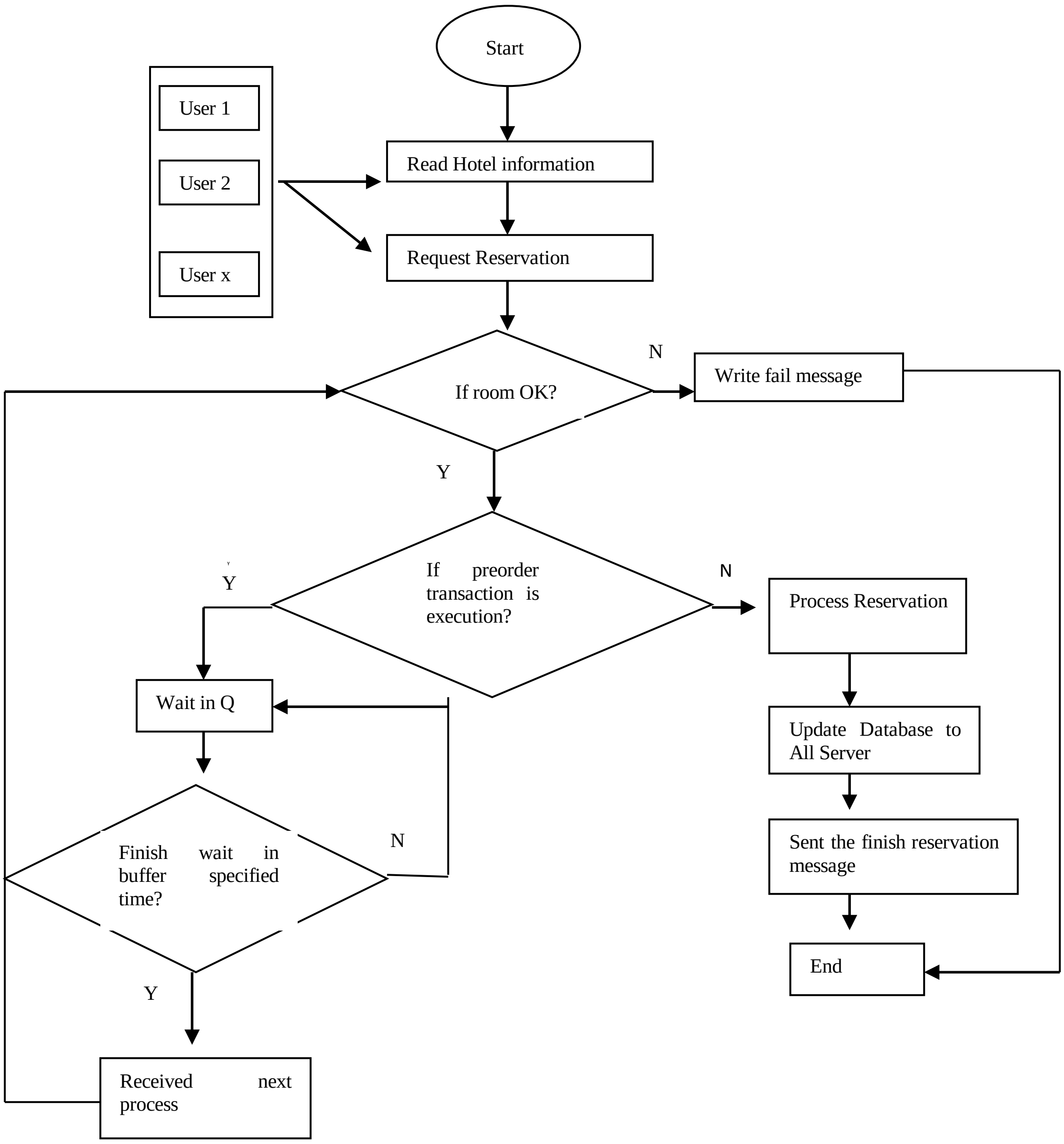

Figure 3. Process flow of the system

Reference

[1] Bernstein, P. and N.Goodman, “Timestamp Based Algorithms for concurrency control in distributed database systems”, Proceeding of 6th international conference on very large database, Oct 1980.

[2] Hoffer, Jeffrey A, MB. Prescott & FR. Mcfadden : "Modern Database Management", Pearson Prentice Hall, 2007

[3] Rucha Patel, “Concurrency control in distributed database”

[4] Silberschatz, “Database system concepts”, Mc-graw Hill, 2001.

[5] ” A Secure Time-Stamp Based Concurrency Control Protocol For Distributed Databases” Journal of Computer Science 3 (7): 561-565, 2007

[6] “A Sophisticated introduction to distributed database concurrency control”, Harvard University Cambridge, 1990.

[7] “Some Models of a Distributed Database Management System with Data Replication", International Conference on Computer Systems and Technologies - C*ompSysTech’07.*

[8] http://www.aaai.org/Conferences/National/ 1997/ aaai97.html