Development of Efficient Temporal Databases in a Real time Database System.

.

Miss. Suyoga m. Mahalle

Dr.V. M. Thakre

Abstract — **This paper presents description of real time database system. Like conventional database, real time data-base system must handle transactions with timing constraint. Real-time database systems must maintain both logical and temporal data integrity so that transactions can respond to real-world events timely and predictably.**

This paper, first presents concurrency control mechanism including various techniques for achieving serializability and improving performance. It illustrates a concurrency control algorithm for accessing to temporal data model in real-time database. The concurrency control algorithm improves the performance of real-time systems by evaluating data-deadline and transactions execution time, adjusting transaction validation rule and committing order. The new concurrency control strategy for DRTDBS is improved 2PL-HP protocol.

This paper also Define the concept of absolute and relative temporal consistency from the perspective of transactions for discrete data objects. And discuss how a real-time database system should be designed to enforce different levels of temporal consistency. And also the concept of data-deadline is introduced and time cognizant transaction scheduling policies, forced wait policies are proposed. Also examine the implications of using run policies to better utilize real-time scheduling in a data base environment.

To improve performance of real time database system by avoiding unnecessary restart of transaction "OCC-DATI"(optimistic concurrency control with dynamic adjustment using timestamp intervals) method propose.

Key Words — **Concurrency control, Deadline, Optimistic, 2PL-HP Protocol, Real time, serializability.**

I. INTRODUCTION

Real time system can be defined as those computing system that are designed to work in timely manner. That is, performing certain actions within specific timing constraints.

A Real time database system is a database system

Providing all features of conventional database system such as data independence and concurrency control also some additional features. It is a transaction processing system designed to handle workloads in which transactions have deadline. Many real-world applications involve not only transactions with time constraints, but also data with time constraints. Typically there are two types of real time system first one is hard real time it is safety critical system that have no chance to miss any deadline or give wrong output and second is soft real time it allow to miss deadline for some applications[1].

In real-time database system only real-time transaction processing will be done. in this deadline to be effective, otherwise it is invalid, the system will immediately stop and abandoned.

Temporal databases incorporate the concept of time to create high-level abstractions useful in database applications. Because of the more use of real time database application in computer networks, and geographically dispersed companies, groups and organizations, there has been distributed real-time databases. The Requirements for real-time not simply use the set of general mechanisms of the distributed database, but must be the basis of the existing improvements. Therefore, it is necessary in the distributed database system based on the introduction of transaction in real-time requirements, here referred to as the distribution of real-time database system [2].

The database system in which time validity intervals are associated with the data introduce the need to maintain data temporal consistency in addition to logical consistency.

In a dynamic environment to monitor the status of real-world objects and to discover the occurrences of "interesting" events real-time database system is often used. The state of a dynamic environment is often modeled and captured by a set of base data items within the system [3].

This paper provides various existing methods to improve performance of Real time database system. And also present the propose method to improve performance of Distributed real time database system.

II. BACKGROUND

Several studies and work has done on performance of real time database system. This paper use concurrency control algorithm for accessing to temporal data model in real-time database. And a checking algorithm is used which is based on the semantic information of transaction deferrable time comparing transaction dead-line with data deadline. The concurrency control algorithm improves the performance of real-time systems by evaluating data-deadline and transactions execution time, adjusting transaction validation rule and committing order.

The concurrency control design space can be classified along several dimensions these are conflict detection, conflict resolution, serialization rule and order, and run policy. Such a classification provides us with a nice framework from which various techniques for achieving serializability and improving performance can be done. Concurrency control for transaction in DRTDBS improved 2PL-HP is used. It is a simple extension of the 2PL protocol 2PL basis in real-time transaction caused the issue of priority. Improved 2PL-HP is the simplest RTDBS concurrency control protocol. It is easy to expand.

III. Previous work done

Author R.K.Abbott, et al [1] had worked on how transaction scheduling will be done in real time data base system. and evaluate performance of RTDBS.

Author A.K.Mok, et al [2] has worked on Application semantics and concurrency control of real-time data-intensive applications.

Author J.Lindstrom, et al [3] proposed Optimistic concurrency control, multiversion concurrency control methods for real-time database system.

Author Yanqun Xiong, et al [4] had worked on Distributed Real-Time Database System Management.

Author B.Adelberg, et al [5] has worked on Applying Update Streams in a Soft Real-Time Database System.

Author M.Xiong, et al [5] has worked on Maintaining Temporal Consistency: and also describe issues and Algorithms.

IV. Existing methodology

There are so many Existing Methods to improve the Performance of real time data base system are as follows.

Concurrency control:

concurrency control is a mechanism to ensure non-interference of transaction execution; thus, isolation of concurrently executing transactions. Data-access scheduling policies used in a real time database system are commonly referred to as concurrency control protocols.

Conflict resolution:

This protocol determines which of the conflicting transactions will actually obtain access to a data item. Conflicts are usually the result of concurrent executions of transactions performing incompatible operations that is read vs. write on the same data item at the same time [1].

Priority-based Wound-Wait Conflict Resolution:

The Wound-Wait technique is used for avoiding deadlocks. Originally this scheme was designed to use timestamps. But modified the scheme so that it uses priorities instead of timestamps and applied the modified version to resolve conflicts in RTDB systems. The modified version is known as High-Priority (HP) and as Priority-Abort (PA) [2].

Two-phase locking (2PL):

In Two-phase locking (2PL), transactions set read locks on objects that they read and these locks are later upgraded to write locks for the objects that are updated [2].

Improved 2PL-HP protocol:

This concurrency control protocol is used for the Transaction to apply for resources, in accordance with their respective level of priority to the corresponding processing. In improved 2PL-HP protocol there is a need to give priority to every transaction at the beginning before the implementation of a system [3].

2PL Wait Promote (2PL-WP):

The 2PL Wait Promote algorithm is identical to basic 2PL in its resolution of conflicts, that is, transactions always block whenever a lock request is denied. a difference is that it includes a priority inheritance mechanism. With this mechanism, whenever a requester blocks behind a lower-priority lock holder, the lock holder's priority is promoted to that of the requester [4].

Real-Time Optimistic Algorithms:

OPT-SACRIFICE, OPT-WAIT algorithm. The goal of this algorithm is to prevent low priority validating transactions from unilaterally committing when they conflict with higher priority transactions. Two different mechanisms, priority sacrifice and priority wait are used to address this problem [5].

V. Analysis and Discussion

A. Analysis of Existing methods:

A conflict resolution mechanism needs to decide which candidate transaction to penalize, and choose an appropriate action and a suitable timing for the action. Two possible actions are most frequently used: blocking or wait and abort or restart. Simple conflict resolution does not used to avoid deadlock but the Priority-based Wound-Wait Conflict Resolution is used for avoiding deadlocks.

Improved 2PL-HP protocol is used in distributed real time database system. It is a simple extension of the 2PL protocol implemented to protect the relative urgency of the transaction at the expense of those low-priority transaction. The 2PL Wait Promote algorithm is identical to basic 2PL in its resolution of conflicts.

The OPT-SACRIFICE algorithm modifies the basic OPT protocol by incorporating a priority sacrifice mechanism. The OPT-WAIT algorithm modifies the basic OPT protocol by incorporating a priority wait mechanism.

B. Attribute and Parameter Considered:

To evaluate the performance real time database system following attributes and parameters are include in RTDBS:

System parameters are

- Database Size: this parameter gives no of pages in database.
- Num CPUs: gives no of processors required.
- Num Disks: no of disk
- Page CPU: CPU time for processing a data page.
- Page Disk: disk service time for a page

Transaction attribute and parameter are:

• The Arrival Rate: this parameter specifies the mean rate of transaction arrivals.

• TransSize: Average transaction size in pages.

• WriteProb: this parameter is used to write probability per accessed page.

• Execution time (CPU time): it is a time required for transaction to completes its execution.

• Access time: it is CPU time per object access.

• Throughput: it is no of transaction completed in one second.

• Deadline: a transaction deadline is the time constraint on transaction.

C. Effect of Outcome of Various Attribute and Parameter:

As the transaction size if small then it requires less time to execute. And less execution time of transaction mean more throughputs and thus it improves the performance of the system.

Deadline is the time in m/second assigned to transactions in real time database system.

D. How Attribute and Parameter Improved:

If transactions with least execution time set with higher priority for execution then it can help to increase throughput and in turn improves performance of the system.

E. Trends of Improvement:

Following list include trends of improvement and performance parameter:

- Throughput
- Database size
- Deadline (time in m/s)
- Access time
- Execution time of transaction.

F. Comparison and Drawback:

Locking based algorithm reduce the degree of concurrent execution of transaction as they construct serializable schedule. But optimistic approaches attempt to increase parallelism to its maximum, but they prune some of the transaction in order to satisfy serializability.

In 2PL_HP a transaction could be restarted or wait for another transaction that will be aborted later. Such wait or restart causes performance degradation. Standard 2PL protocol have many drawbacks are long and unpredictable delay, deadlock, unbounded priority inversion. So improvement in 2PL is 2PL-WP but sometime it behaves as badly when number of transactions and resource sharing are large. 2PL_HP overcome some of the drawbacks of 2PL_WP but may result in large amount of completed work. Optimistic protocol performs best under low resource constraint and low load situation and its main problem is unnecessary restart of transaction. The optimistic methods have attractive properties of being no blocking and deadlock free.

VI. PROPOSED METHODOLOGY

In optimistic methods one of the main problem is unnecessary restart. This is due to the late conflict detection that increases the restart overhead, since some near to completed transactions have to be restarted. Therefore to avoid unnecessary restart this paper proposed new optimistic concurrency control method named "optimistic concurrency control with dynamic adjustment using timestamp intervals" (OCC-DATI). In which number of transaction restart is reduced by dynamic adjustment of serialization order which is supported by similar time stamp intervals.

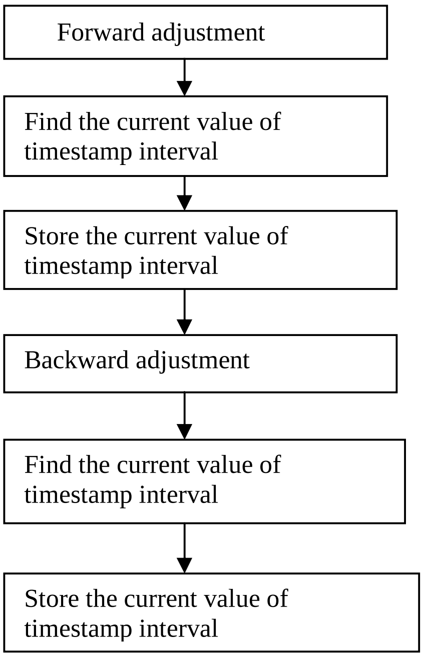
Forward adjustment
Find the current value of timestamp interval
Store the current value of timestamp interval
Backward adjustment
Find the current value of timestamp interval
Store the current value of timestamp interval

Fig. forward and backward adjustment in OOC-DATI

This method performs both forward and backward adjustment on the basis of time interval is as shown in fig.

In OOC-DATI all checking is performed at validation phase of each transaction. There is a no need to check for conflicts while the transaction is in its read phase. The conflict resolution between the transactions in OOC-DATI is delayed until a transaction is near completion**.**

VII. EXPECTED RESULT

In this way, this paper presents, how performance of real time database system is improves by improving the optimistic concurrency control method. "Optimistic concurrency control with dynamic adjustment using timestamp intervals" (OCC-DATI) method helps to improve performance by avoiding unnecessary restart of transaction. In OOC-DATI timestamp intervals have been adapted as the method to implement dynamic adjustment of the serialization order. Timestamp intervals allow transactions to be both forward and backward adjusted.

CONCLUSION

In this way this paper describes how real time database system is different from conventional database system. It illustrate various concurrency control methods, transaction scheduling, optimistic concurrency control schemes for RTDBS and evaluate the performance of these each scheme. Each method/scheme has its own advantages and disadvantages. There are number of parameters to improve the performance of real time database system. Thus to improve performance by avoiding unnecessary restart of transaction this paper propose a method is "OOC-DATI" (optimistic concurrency control with dynamic adjustment using timestamp intervals).
Applications of RTDBS are:

- Robotics process control
- Stock exchange
- Network and traffic management system
- Airline reservation, banking, stock market system(soft real time)
- Radar tracking system, patient care system, system working in airplanes(hard real time)
- Military command and control management system.
- Industrial control, Nuclear reaction control, Aircraft navigation system etc.

FUTURE SCOPE

The OOC-DATI method resolves conflicts between transactions by using timestamp interval. In this every transaction must be executed within a specific time interval, but for good performance the future work for OOC-DATI method involves extending the OOC-DATI conflict resolution method with deadline driven priority information. So to resolve conflict, higher deadline driven priority transactions are needed to be favored.

REFERENCES

[1] R. Abbott and H. Garcia-Molina, "Scheduling Real-Time Transactions: A Performance Evaluation," IEEE transaction on Database Systems, vol. 17, no. 3, pp. 513-560, Sept. 1992.

[2] Han qilong, hao zhongxiao. "Real-time Optimistic Concurrency Control based on Transaction Finish Degree"IEEE transaction on computer science,vol. 5, no. 2,pp .471-476, 20051.

[3] Q.N. Ahmed and S.V. Vrbsky, "Triggered Updates for Temporal Consistency in Real -Time Databases" IEEE transaction on Time-Critical Computing Systems, vol. 19, no. 3, pp. 209-243, Nov. 2000.

[4] Son, "On Real-Time Databases: Concurrency Control and Scheduling," Proc. IEEE transaction on computer , vol. 82, no. 1, pp. 140-157, 1994.

[5] M.Xiong, R.Sivasankaran, J. A. Stankovic, K. Ramamritham, and D. Towsley, "Scheduling Transactions with Temporal Constraints: Exploiting Data Semantics," IEEE transaction 0n computer science,vol. 51, no 6, Dec. 1996.

[6] X. Song and J.W.S. Liu, "Maintaining Temporal Consistency: Pessimistic versus Optimistic Concurrency Control," Proc. IEEE
Trans on Knowledge and DataEng., vol. 7, no. 5, pp. 786-796, Oct. 1995.

[7] M. Xiong, J. Stankovic, K. Ramamritham, D. Towsley, and R.M. Sivasankaran, "Maintaining Temporal Consistency: Issues and Algorithms," IEEE transaction on knowledge and data engineering, vol. 14, no. 5, October 2002.

[8] M. Xiong, R.M. Sivasankaran, J. Stankovic, K. Ramamritham, and D. Towsley, "Scheduling Transactions with Temporal Constraints: Exploiting Data Semantics," IEEE transaction on knowledge and data engineering, vol. 14, no. 5, pp. 240-251, Dec. 1996.

AUTHOR'S PROFILE

Suyoga Mukundrao Mahalle
She is completed her B.E degree in Computer Science, from PRMIT And R, Badnera. And pursuing her M.E degree in Computer Science and Information Technology from Department of Computer Science, S.G.B.Amravati University, Amravati