ISSN 22773061

Performance evaluation of conflict serializability against 2PL in Homogenous Distributed Database

Harpreet kaur[1], Mritunjay Kumar[2], Gulshan kumar[3]
Computer Science and Engineering
Lovely Professional University, India
Preeti.kaler14@gmail.com
mritunjay.14555@lpu.co.in
gulshan.16865@lpu.co.in

Abstract— There are many techniques in the recent years that provide the synchronization among the transactions using shared data like 2 phase locking protocol and time stamping in distributed database system. These techniques are efficient and provide fast transactional operations and serializable execution of the transactions. But these existing techniques like 2PL abort the transaction when a conflict occurs in transaction actions that result in degradation in the performance. In this dissertation we analyse the purposed techniques that provide the serial execution of the concurrent transactions. To achieve the serializability among the transactions in the distributed database Conflict serializability model is used that will provide serial execution of the transactions with in conflict actions of the transactions. The objectives of the scheme are achieved that provide the better performance than the existing 2PL schemes and achieve the consistency among the concurrent transaction execution. Performance is evaluated of CS model against the 2PL model in distributed database to sustain the synchronization in the transactions. Results illustrate that the proposed new scheme increase the performance by decreasing the abort rate in distributed environment.

Keywords- concurrency control, CS (conflict serializability), synchronization, conflict equivalent, distributed database.

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY
Vol 7, No 1
editor@cirworld.com
www.cirworld.com, member.cirworld.com

ISSN 22773061

1. INTRODUCTION

Today most research and business environment deals with the database. Database is simply the structured collection of data. It is field that concern with the real time applications like aircraft control, network management and stock marketing etc. For reliable and scalable accessing of the data distributed database come into exist that handle or access the data. Distributed database is the system in which data is not placed on the single site or location like centralized database, here the data is distributed and replicated over the different physical locations.

When dealing with the distributed database several issues are to be considered. Concurrency control is the major issue that can handle the accessing of several transactions simultaneously and guarantee the serializablity executions of the given transactions. The goal of concurrency control is to prevent the inconsistency and interference among the transactions and allow simultaneously access the data from the different sites of the distributed database system. The serializability property ensures the consistency of the distributed transactions.

Two or more transaction performs the operation (READ-WRITE) using shared data cause the conflict among the transactions. To overcome that Conflict Serializablity (CS) model allow the transaction execute in serial schedule.

Various previous works are done to control the concurrency and provide the synchronization among the shared data in centralized database system as well as distributed database [15] [17].The concurrency control Protocol like 2PL, S2PL, 3PL aborts the transaction every time the conflict occur that reduce the availability and performance of the transaction processing as compared to the serializability model like conflict serializablity (CS).

In distributed database when several transactions are executed concurrently accessing the shared data then concurrency problem occurs. To handle the concurrent transaction to allow execute in some serial order, conflict serializability mechanism is used. It guaranteed the serializablity execution of the simultaneous transaction. If several transactions are executed concurrently, there result must be the same as they executed in some serial order.

In this paper we describe concurrency control problems among the transactions in distributed database system. To handle that we purposed the use CS model that allow transactions to commit even when the conflicting access to the shared data are given. It should achieve the high monitoring performance with conflicted transactions. The performance of that model is going to be measured in contrast to the 2PL model. As the serializability model construct the serial schedule to make the transactions execute in serial order results less abort rate of the transactions as compared to 2PL.The abort rates of the transaction are relatively decreased that is proven in this work.

DISTRIBUTED DATABASE SYSTEM

“Distributed database is define as a collection of multiple, logically interrelated database distributed on the computer network.”[1]. Distributed database is the system in which data is not placed on the single site or location like centralized database, here the data is distributed and replicated over the different physical locations that are connected together by some communication networks. The software that creates and administers the distributed database and provides data to the user is called the distributed database management system (DDBMS).ORACLE, Microsoft SQL server, MySQL, PostgreSQL are the example of DBMS used to manage the database.
It is important to distinguish between distributed and decentralized database. These two terms are seems similar but the functionality of both are different. The decentralized database is also stored on computers at multiple locations but unlike distributed database the computers are not interconnected by communication network.

In distributed database the data is physically stored on two or more computer systems, so DDBMS allow to manages or organize the whole database as a single collection of data as result the individuals able to access data from any of the database system because the replication of the data among all the systems in the network. And the DBMS periodically synchronized the database to remain consistency among the distributed environment.

ISSN 22773061

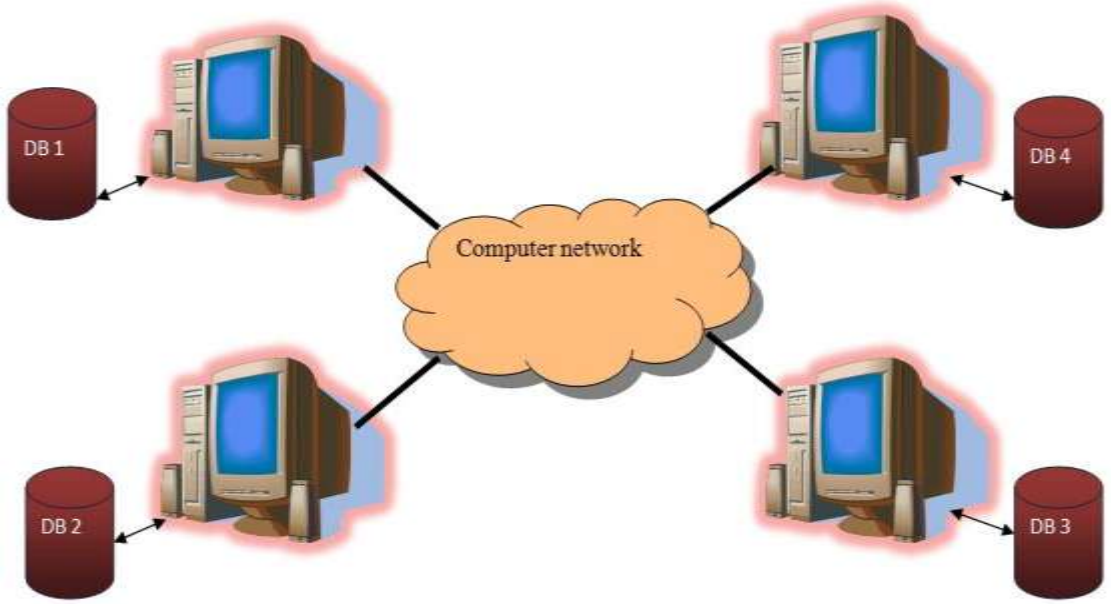

Figure 1 Distributed database system

In distributed database pair to pair connectivity among the systems is established and the data is replicated or distributed over the several locations. The replication of the data on different sites over the network is done by fragmentation of the data either using horizontal or vertical type of fragmentation. This ensures that the multiple copies of the data are placed on the different nodes of the network results in better data access and increases the data availability, if one site is down then the data is retrieved from another copy of the data.

2. RELATED WORK

Conflict serializability (CS) is the well-known concept in database domain to manage the concurrent transactions effectively. Our contribution is to implement this model to handle the conflicted access using shared data.

B.D.Carlstron et.al [7] describe the fact that transaction memory classes address the problems that occur in long running transaction that operate on the common shared data and the conflict can be eliminated. To supporting the transactions with high contention in database concurrency control is used that hold the ACID properties of the transaction. To provide the serializability property Strict Two Phase Locking (S2PL) method is use in database that provide the serial ordering of the commits in the transactions. But this isolation method limits the concurrency and result in high abort rate when conflict occurs.

U.Aydonat, T.S.Abdelrahman[13] purposes the CS model in the domain of STM system. The SON (serial order number) algorithm is used adjusting the serialization of transactions at runtime using time stamp intervals. This technique assumes the write buffering and adjusts the SON ranges while transactions are active and update their time stamp ranges only when transaction commits. The purposed algorithm is implemented in STM system and evaluate its performance on standard benchmarks that results in improvement of performance of the system with long transactions and high abort rate.

T.S.Abdelrahman[20] purposed the multi-versioning techniques for concurrency control in database. Multi-versioning increase the concurrency by allowing transactions to read only versions if shared data is used. This is implemented on obstruction –free STM system and result in increasing the performance and throughput of the running transactions using shared data.

David Lomet, Alan Fekete[12] described various concurrency control approaches are use in database that allows concurrent transaction access with high availability and performance. The author purpose the new concurrency control approach that enables all serializability to utilize multiple versions to increase the concurrency.TCM timestamp range conflict manager can handle the conflicts occur in the transactions and also overcome the blocking problems faced in S2PL models.TCM deals with the distributed transactions by handling both sub transactions and global transaction with timestamp range and provides read/write concurrent access while proving all SQL isolation levels, including serializability.
A.Srivastava, S.Tawari [5] To provide the synchronous transaction access by different user in distributed and replicated database author purposed the Two phase commit protocol that ensure the consistent termination of transaction in the distributed database system, where the local coordinator and the global coordinator collectively provide the consistence execution of the transactions. The 2 phase commit protocol can be implemented in distributed database system and achieve the better transaction throughput with complex transaction conflicts. But further study also point out the thing that 2PL provide the consistency of the dataset but with high abort rate it limits frequency of the committed transactions.

A.yadav, Dr. Aggerwal [4] defines the concurrency control methods to handle the conflicts among the transactions. The synchronization problem decomposes into Read-Write (RW) and Write –Write (WW) sub problems. The describe various synchronization techniques for solving each sub-problem and show 2PL implementation to handle the concurrency problem. Many performance issues are pointed out in it that gave way to our research in evaluate the performance of purposed model against the 2PL.

Sheetlani Jitendra et.al [18] describes the transaction management in distributed database system that provides the much flexibility and high performance as compared to the traditional centralized database system.Nested transactions techniques are used that overcome the limitation of the flat transaction model. Using nested transaction better concurrency control of transaction is achieved when the transaction from one system is interacting with the transaction of the other system. Nested transactions extend the notion that transactions are flat entities by allowing a transaction to invoke atomic transactions as well as atomic operations. They provide safe concurrency within transaction, allow possible internal parallelism to be exploited and offer an appropriate control structure to support their execution. Author describes advance-nested transactions where the transactions from one system interact with the transactions from another system. Nested transactions can expect to become more important with the introduction of network operating systems and heterogeneous distributed database systems.

3. DISTRIBUTED APPROACH FOR CONCURRENCY CONTROL

The management of the distributed transaction deals with the consistence, reliable, efficient and concurrent execution of transaction in the DDBMS. When the transactions access the shared data simultaneously concurrency problems occurs. Concurrent transactions causes conflicts if they acquiring lock for using the shared data at a time .To provide the synchronization among the shared data used by system and to handle the concurrency problem various algorithm and techniques are exist in the database system.[5][6][7][15]. Concurrency control is the major issue in the database either in centralized database or distributed database. Lock based techniques like 2 PL are implemented in distributed database to provide the consistency or serializability among the transactions performing read or write actions on the different system.

In 2PL there is pre declaration of lock to the data item, for example before reading data item (X) the transaction must have a READ lock on (X) and before writing to the data item (y), the transaction has a write lock on it. In distributed database each node have its local replicated database, so to provide the synchronization among the shared data used by the transaction, 2PL phase are applied on the local subtranaction of all the replicated database. Because of the redundancy between the data items in the DDBMS, conflict can occur when two transactions can hold the lock on the shared data. Blocking, deadlock problems can be occur, that can be overcome using various deadlock detection algorithms and provide synchronization among the shared data by executing transaction in serial schedule.

Transaction memory system implement 2PL that is lock based protocol used to coordinate the read and write operation to prevent the conflict occur in the transaction. There are two phases for locks in 2PL:-

(a) Growing phase :- the lock are acquire by the transaction
(b) Shrinking phase:- the locks are release by the transaction

Lock based 2PL can't allowed the conflicting access to the shared elements. The transaction conflicts are the RAW, WAR, WAW. Any conflict among the actions of the transactions becomes the cause of abort of the transaction in the 2PL model. Even this model based on the locking model that locks the data item until the data is in used. If the other transaction use the same data that already locked may cause the conflict in long transactions process. That result in high abort rate of the long transactions. To handle this problem a concurrency control algorithm is defined that is based Conflict serializability model [13][20].This model allow the transactions to commit even when contradictory access to familiar data is present. This is the more relaxed model as compared to 2PL that allow transaction to commit when they are aborted with 2PL model.

A SON (Serial Order Number) algorithm [13] defined to implement the CS model in distributed environment that allows the execution of the concurrent transactions in some serial schedule that ensures the consistency property of the transaction. This algorithm is base on the fact to if the transactions are construct the conflict equivalent schedule ,then the transaction are serializable. The schedules S1 and S2 are said to be conflict-equivalent if they satisfied the following conditions:-

1. Both schedules S1 and S2 involve the same set of transactions
2. The order of each pair of conflicting actions in S1 and S2 are the same.

The SON is the integer that indicates the relative order of the transactions conflict serial schedule among all the transactions. The order of the transaction action provides them serial order execution of the transaction and allows some transaction commit when conflict occurs.

4. EXPECTED RESULTS

- To improve the concurrency control of the distributed database.
- To maintain the consistency of the data in the database using the concurrency control techniques or algorithms
- The abort rate of the transactions is decreased by providing serial order to the transaction as compared 2PL.
- To achieve the better synchronization among the concurrent transactions using shared data items.
- Achieve the high monitoring performance with conflicted transactions.
- Not increase the communication overhead in the distributed network.

- The throughput of execution of the transactions will be increased with long running transactions and with high conflict rate.

With these objectives kept in consideration the conflict serializability model is used in homogenous distributed database system that will handle the concurrency among transactions and satisfy all these design objectives.

5. RESEARCH METHODOLOGY

Based on the synchronization problems associated with the distributed transactions that discuss so far in paper, and based on previous related work, we purposed the way to achieve the serializability in the distributed database system by using the Conflict serializablity model. This model provides the synchronization among the conflicted transactions and reduces the abort rate of the transactions as compared to the 2PL protocol. 2PL works well with short running transactions but with long running transactions including conflicted data results in high abort rate of the transactions.

To provide the synchronization among the concurrent transactions and reduce the abort rate of the distributed transactions in 2PL mechanism, the CS model is purposed that provide the serializability execution of the concurrent transactions. The Conflict Serializability model allows the some conflicts transactions to successfully commit. The different participant at the different resources have right to use the mutual data concurrently but result the conflicts. That conflicted access of the shared data will be allowed successfully execute transactions by Queued to SON for execution.

5.1 Proposed Scheme

Serial execution of the distributed transactions is achieved by Conflict Serializability method. The main concept of the purposed techniques is discussed below:-

- **Concept of Conflict Serializablity**

  The technique is based that if a conflict-equivalent serial schedule can be constructed, then the transactions are serializable. In the distributed database different number of participant can access the data from multiple resources simultaneously. In 2 Phase Locking protocol when the participant request for data access from particular recourses, that resource must be locked by that transaction which want to update or retrieve the data object from that database. When the other participant request the same resource that already locked by previous transaction, the transactions went to wait and further result is the transaction is aborted. Any participants that want to access the locked data item result as aborted.

  - SON (Serial Order Number)
    When the number of transactions is applied over the different resources in distributed environment the serial schedule is constructed by a serial order number (SON) for each transaction. SON's indicate the relative order of all transactions in the conflict-equivalent serial schedule that is being constructed. That is, a transaction that has a smaller SON is serialized earlier than a transaction that has a larger SON. When the participant requests the resources that are already occupied, the SON places the transaction in the Queue. That is the transactions are Queued to SON for execution.

This given structure point out the basic overview about the CS model

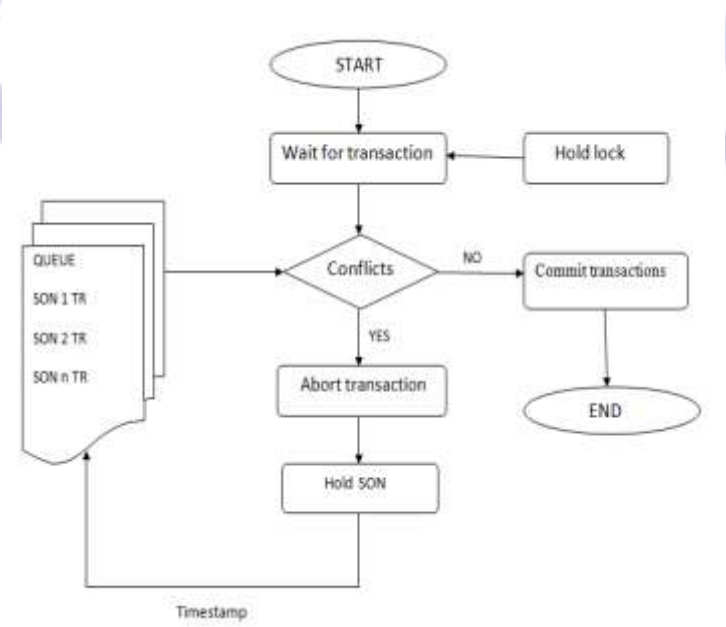

Figure 2 CS model with SON

ISSN 22773061

The 2 phase locking protocol acquire the locks on the data item in growing phase and no other transaction can access the shared data that is previously locked by some transaction. In distributed system if the two or more transaction wants to access the shared data conflict can occur in transactions. The conflicted transactions leads to high abort rate of the transactions in the database.
The purposed CS model handles the multiple transactions over the distributed computer systems and handles the conflicted shared data. The model leads to commit the transactions even they access the conflicted data items among two or more transactions. and reduce the abort rate of the transactions as compared to 2PL.

- **CS Model with SON**

CS model uses the SON (Serial order number) that constructs the distributed transaction in some serial schedule. The transactions participant that require the recourses that already locked by some other participant of the transactions put the transaction in SON (Serial Order Number) QUEUE. The transactions are queued to SON for executions. The timestamp is placed between the resources between the growing or released locked on the resources. The requested participant placed in queue that further can able to access the recourses efficiently.
Thus the conflicted transactions are able to access the recourses data. The SON algorithm can resolve the conflicts and make the transaction committed. The no. of transactions are executed simultaneously want to access the shared database. The CS model can make a serial schedule of the given transactions so that the different transactions can able to access the different data entity of the no. of resources.
The following steps the internal working of CS model.

1. The system first wait for the transactions. The number of distributed transactions from the different participants is given to the system.
2. The required shared or Exclusive locked are hold on the transactions.
3. Conflicts among the transactions must be checked. If more than one participant want to access the same database resources conflict occurs.
4. It conflicts is occur the transaction hold in SON.
5. The transaction queued in SON for execution.
6. Further transactions conflicts are checked and commit the transactions.

The following example shows the how the transactions are aborted in 2PL mechanism .and further shows CS model with SON can reduce the abort rate by making conflicted transactions committed. Three transactions are feed into system concurrently by different participants.

Transactions:-

TX1, TX2, TX3

Operations:-

READ, WRITE

Actions			Comments
Start (TX1)	Start (TX2)	Start (TX3)	
----------	Read(TX2,B)	-- ------	
----------	----------	----------	
Read(TX1,A)	----------	Read(TX3,B)	
----------	Read(TX2,A)	----------	
----------	A = A-100	Read(TX3,A)	A locked by TX2, Abort TX3
----------	Write(TX2,A)	----------	
----------	----------	ABORTED	

Figure 3. transaction schedule in 2PL

ISSN 22773061

Actions			Comments
Start (TX1)	Start (TX2)	Start (TX3)	
----------	Read(TX2,B)	----------	
----------	----------	----------	
Read(TX1,A)	----------	Read(TX3,B)	
----------	Read(TX1,A)	----------	
----------	A = A-100	Read(TX3,A)	A locked by TX2, hold TX3, queue as per SON for later execution
----------	Write(TX1,A)	Hold	A unlocked by TX2
		Read(TX3,A)	Resume TX3 A is unlocked by TX2
----------	----------		

Figure 4. Transaction schedule in CS with SON

The example shows the serial execution of the transactions where the aborted transactions TX3 is hold SON in Queue for later execution. The transactions TX2 released the locked then the TX3 that hold SON in queue can executed and result the committed transactions.

6. RESULTS AND DISCUSSIONS

The purposed scheme is implemented in JAVA programming language which provides the concurrent computing, object oriented programming structure.
The following results illustrate that the proposed new scheme increase the performance by decreasing the abort rate in distributed environment.

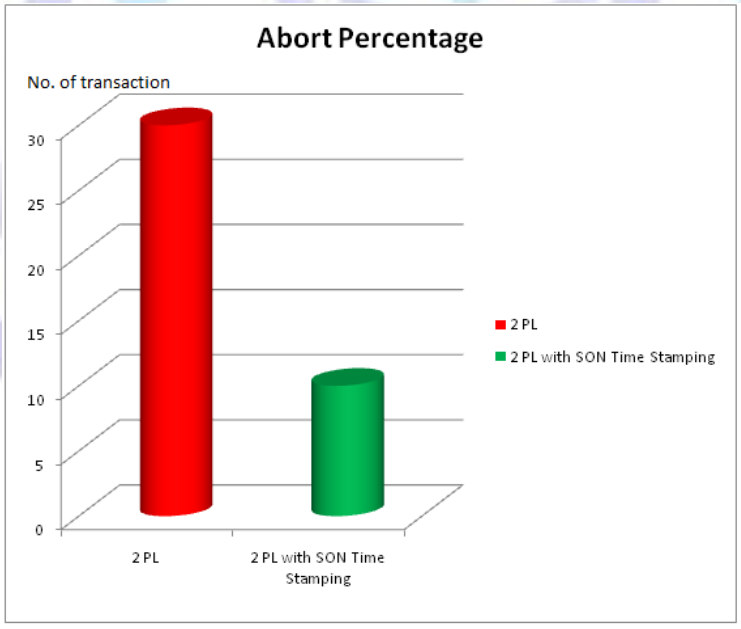

Figure 4.11 ABORT percentage of 2PL and SON model

The graph shows the abort rate in both the mechanisms. In 2PL the abort rate is high as the numbers of transactions are increased. With the long transactions the abort rate gets increased in 2PL model.
The serializablity model with SON allows the transactions to commit with in conflicted process that result in minimum abort rate. As the number of transactions is increased the transactions are process using SON queue allows conflicted transactions committed.

7. CONCLUSION

To provide the serializability execution of the transactions in the distributed database the purposed Conflict Serializability model is used that provide serial execution of the transactions with in conflict actions of the transaction. This model is leading to achieve better scaling performance with long running transactions. In developing the proposed model, we have taken into deliberation previous findings of related studies. The Conflict Serializability technique is proved that it is more reliable than other 2 Phase Locking protocol, that is conclude in this work. The CS model with SON mechanism provides the feasible execution of the transactions that may lead to conflicted transactions operations. The abort rates of the transaction are relatively decreased as compared to the 2PL that is proven in this work. The throughput of execution of the transactions is increased with long running transactions and with high conflict rate as compared to the previous 2 Phase Locking techniques.

ACKNOWLEDGEMENT

Working on my paper has been a milestone in my academic career. It has provided me an opportunity in learning new concepts, terminologies, and theories and expanding my academic knowledge beyond my classroom learning. I am grateful to my guides Mritunjay Rai and Gulshan kumar who guided and supported me throughout the research process and provided assistance for my venture. Any suggestions to further improvement of this topic are most welcome.

REFERENCES

[1] Stefano Ceri and Giuseppe Pelagatta “Distributed Databases Principles & Systems” 2008.
[2] Albert Burger, Vijay Kumar and Mary Lou Hines “Performance of Multiversion and Distributed Two-Phase locking Concurrency Control Mechanisms in Distributed Databases” Information sciences 96, No. 1, pp 129-152,1997.
[3] Alexander Thomasian and Erhard Rahm “A New Distributed Optimistic Concurrency Control Method and a Comparison of its Performance with Two-Phase Locking” IEEE Distributed Computing Systems. pp 294-301 May 1990.
[4] Arun Kumar Yadav and Ajay Agarwal “A Distributed Architecture for Transactions Synchronization in Distributed Database Systems” IJCSE Vol. 02, No. 06. pp 1984-1991, 2010.
[5] Ashish Srivastava, Udai Shankar and Sanjay Kumar Tiwari “Transaction Management in Homogeneous Distributed Real-Time Replicated Database system” IJARCSSE Vol 2, No. 6 pp 190-196, 2012.
[6] Bernstein Dr. Philip. “Concurrency Control” Database Hall of Fame. 2001.
[7] Carlstrom, Brian D., Austen McDonald, Michael Carbin, Christos Kozyrakis, and Kunle Olukotun. "Transactional collection classes." In Proceedings of the 12th ACM SIGPLAN , pp. 56-67. ACM, 2007.
[8] Christos H. Papadimitriou “The Serializability of Concurrent Database Updates” Journal of the Assoclauon for Computing Machinery. Vol 26. No 4. pp 631-653 October 1979.
[9] Dimitrios Georgakopoulos, Marek Rusinkiewicz, and Amit Sheth. "On serializability of multidatabase transactions through forced local conflicts." In Data Engineering, 1991, pp. 314-323. IEEE, 1991.
[10] Ghazi Alkhatid and Ronny S.Labban “Transaction Management in Distributed Database system: the case of Oracle’s two –phase commit” Journal of Information systems Education, Vol. 13(2) pp 94-103, 2004.
[11] Jones, Evan PC, Daniel J. Abadi, and Samuel Madden. "Low overhead concurrency control for partitioned main memory databases" pp. 603-614. ACM, 2010.
[12] Lomet, David, Alan Fekete, Rui Wang, and Peter Ward. "Multi-Version Concurrency via Timestamp Range Conflict Management." In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 714-725. IEEE, 2012.
[13] Utku Aydonat and Tarek S.abdelrahman “Relaxed concurrency control in software transaction memory” IEEE Transactions on Parallel and Distributed Systems, VOL. 23, No. 7, pp 1312-1325,July 2012.
[14] Maximiliano Canche1, Juan Lavariega and Erika Llanes “Design and Implementation of a Component-based Concurrency Control Mechanism for a Distributed Database” IJCSI, Vol. 9, Issue 6, No 3. pp 68-72, 2012.
[15] Neera Batra, and A. K. Kapil. "Concurrency Control Algorithms and its Variants: A Survey." In AIP Conference Proceedings, Vol. 1324, pp. 46-50, 2010.
[16] Omar Baakeel and Abdulaziz Alrashidi“A Distributed transaction model for a Multi-database Management system” International journal of Scientific & Engineering Research Vol 3, Issue 3, March -2012.
[17] Philip Bernstein and Nathan Goodman. "Concurrency control in distributed database systems." ACM Computing Surveys (CSUR) 13, no. 2, pp 185-221,1981.
[18] Sheetlani, Jitendra, and V. K. Gupta. "Concurrency Issues of Distributed Advance Transaction Process." Research Journal of Recent Sciences Vol. 1, pp 426-429 ,2012.
[19] Subhash Bhalla “Parallel Concurrency Control Activity for Transaction Management in Real-time Database Systems”, The Journal of Supercomputing, VOL .28 No.3, pp.345-369, June 2004.
[20] Utku Aydonat and Tarek Abdelrahman. "Serializability of transactions in software transactional memory." In Second ACM SIGPLAN Workshop on Transactional Computing. 2008.