International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||12||Pages||1839-1845||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Security and Concurrency Control in Distributed Database System

Manoj Kumar Sah [1], Vinod Kumar [2], Ashish Tiwari [3]

[1] Assistant Professor, Dept. of Computer Science and Engineering
Shobhit University, Gangoh, UP, India.
emanojcse@gmail.com

[2] Assistant Professor, Dept. of Computer Science and Engineering
Shobhit University, Gangoh, UP, India.
vinod2911@yahoo.co.in

[3] Assistant Professor, Dept. of Computer Science and Engineering
Arya Institute of Engineering & Technology, Jaipur , India
er.ashish.tiwari89@gmail.com

Abstract: *This paper reviews the security issues and concurrency control in distributed database system. In this paper we have concerned about the features of distributed database system and its security issues. Distributed database is a collection of databases that can be stored at different computer network sites. This paper provides various aspects like fragmentation, types of fragmentation, replication and various problems that can be faced into distributed database system. The main goal of distributed database system design is to making fragmentation of the relations. Fragmentation is a design technique to divide two or more partitions such that the combination of the partitions provides the original database without any loss of data. By doing so, this reduces the amount of irrelevant data accessed by the applications of the database, thus reducing the disk accesses, which increases the performance of system.*

Keywords: Distributed Database, Deadlock, transaction, Fragmentation, Replication, locking, security, scalability, concurrency control, query optimization.

1. Introduction

In this paper we have presented concurrency control and security issues in distributed Database System. In today's world of Information technology, it is demand of society to available each information on hand. There is also need for secure and reliable communication of information. Distributed database provides such facility. A distributed data base is a database in which storage devices are not all attached to a common processing unit such as the CPU. It is stored in multiple computers. A distributed database system consists of loosely coupled sites which does not share physical component. In distributed database, users at any location can access information at anywhere in the network. The main goal of A distributed database system is to control the management of a distributed database in such a way that it seems for user as centralized database.

The purpose of this paper is to present an introduction to distributed databases, concurrency control and security issues which are becoming very popular now days. Today's business environment has an increasing need for distributed database. Distributed database systems provide an improvement on communication and data processing due to its data distribution throughout different network sites. Not only is data access faster, but a single-point of

failure is less likely to occur, and it provides local control of data for users. A distributed database is a single logical database that is spread physically across computers in multiple locations that are connected by data communication links. Concurrency control is also an important issue in database systems. Concurrency control is the process of coordinating concurrent access to a database in multi user fashion. We have discussed concurrency control algorithm as 2PL, BTO, Wound Wait and OPT in this paper. Concurrency control and security is important for transaction management.

2. Why distributed databases?

1. Organizational and economic reasons
2. Interconnection of existing databases
3. Incremental growth
4. Reduced communication overhead
5. Performance considerations
6. Reliability and availability
7. Greater control

2.1 Types of Distributed Database System:

1. Homogeneous Distributed Database System - In this data is distributed but all servers run on the same database Management system software.
2. Heterogeneous distributed database system: In this different site run under the control of different DBMS.

2.2 Advantages of Distributed database

1. Data is distributed, so network traffic reduced.
2. Local database still works even if the company Network is temporarily broken.

3. If there is problem in one branch then it does not Affect working of other branch.
4. Queries are local, so high performance.

2.3 Disadvantage of Distributed database

1. In Distributed System it is more complex to make sure that data and indexes are not corrupted.
2. Distributed System is not efficient if there is heavy interaction between sites.
3. It is more complicated to maintain Distributed database as compare to centralized database.
4. Security problem as data are distributed.

2.4 Architecture of Distributed Database System

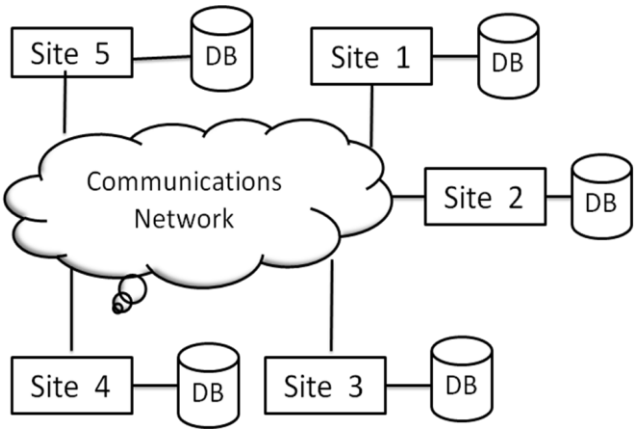


Fig 1: Distributed Database System

2.5 Centralized Database System

A centralised database is a database that is located, stored, and maintained in a single location

1. It Run on a single computer system and do not interact with other.
2. It works as General-purpose computer system.
3. Single-user system
4. Multi-user system: more disks, multiple CPUs, and a multi-user OS.

2.6 Architecture of Centralized DBMS

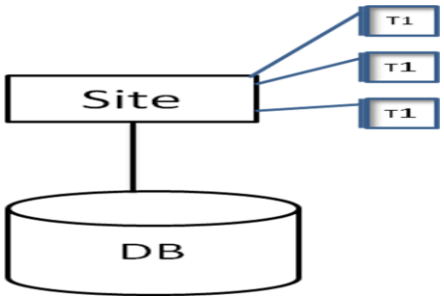


Fig 2: Centralized Database System

3. Design of Distributed Database Management System

A distributed database management system (DDBMS) is a set of multiple, logically interrelated databases distributed over a network. They provide a mechanism that makes the distribution of data transparent to users.DDBMS is widely used in data warehousing, where huge volumes of data are processed and accessed by numerous users or database clients at the same time. This database system is used to manage data in networks, maintain confidentiality and handle data integrity.

The design of a distributed database introduces three new issues:

- How to partition the database into fragments.
- Which fragments to replicate
- Where to locate those fragments and replicas.

A distributed database is database whose storage device are not all attached to a common processing unit since database are stored at multiple computer. A distributed database resides on network server on the internet.

3.1.1 Fragmentation

Fragmentation is a design technique to divide to divide a relation into two or more partitions in such that combination of partitions provides the original database without any loss of original data. Database fragmented into several pieces called fragments, which is stored on different site. In Distributed database, database is broken into logical units called fragments which will be stored at different sites. [1]. The simplest logical unit are tables.

3.1.1 Advantage of Fragmentation

Efficiency - Data are stored on nearest to those databases which are frequently used.

Parallelism - Due to fragmentation a transaction can be divided into several sub queries that operate on fragments. This increases the degree of concurrency or parallelism in the system.

Security - Data, which are not required by local applications, is not stored, and not available for unauthorized users.

3.1.2 Disadvantage of Fragmentation

Data are stored at different sites, so overall performance becomes slow. Integrity control becomes too difficult.

3.1.3 Types of Fragmentation

Fragmentation is of three types as discussed follows

3.1.4 Horizontal Fragmentation

It divides the relation into tuples or rows. It divides table horizontally by selecting the relevant rows and these fragments can be assigned to different sides in the distributed system. It allows a class to be partitioned into disjoint instances. In horizontal fragmentation table remains same, only rows gets split. Horizontal fragmentation is defined as selection operation, sigma Employee (Attribute).

Consider Relation: Relation R is: EmpTable E(Emp. No., Emp Name, Salary, Qualification)

Employee Table (E)

Emp No.	Emp Name	Salary	Qualification
1	Manoj	30000	M.Tech
2	Rahul	40000	M.Tech
3	Krishna	25000	M.Tech

4	Vikash	60000	P.hD.
5	Vivek	29000	M.Tech

The Employee Table E has four fields, as Employee Number, Employee Name, Salary and Qualification of Employee. We are fragmenting this Employee Table by using Horizontal Fragmentation below as we are taking rows based on some given condition , Condition is :

Employee Table (E_1): Fragment with Salary Less than equal to 30000.

Employee Table (E_2): Fragment with Salary Greater than 30000.

Employee Table (E_1)

Emp No.	Emp Name	Salary	Qualification
1	Manoj	30000	M.Tech
3	Krishna	25000	M.Tech
5	Vivek	29000	M.Tech

Employee Table (E_2)

Emp No.	Emp Name	Salary	Qualification
2	Rahul	40000	M.Tech
4	Vikash	60000	P.hD.

3.1.5 Vertical Fragmentation

Vertical fragmentation divides the relation into attributes called columns. In vertical fragmentation it is necessary to include primary key of table in each vertical fragmentation. If any time we need to construct the original table, then it is possible with the help of primary key. In vertical fragmentation one table is split into two or more table. The main objective of vertical fragmentation is to partition o relation into a set of smaller relations so that many of the applications will run on only one fragment.[2]. Vertical fragmentation of relation E produces E_1, E_2 each of which contains a subset of E's attributes. Vertical fragmentation is defined using the projection operation of the relational algebra.

$\Pi (E_1, E_2) (R)$

Employee Table (E)

Emp No.	Emp Name	Salary	Qualification
1	Manoj	30000	M.Tech
2	Rahul	40000	M.Tech
3	Krishna	25000	M.Tech
4	Vikash	60000	P.hD.
5	Vivek	29000	M.Tech

Condition:

Employee Table (E1): Fragment with Emp Name and Salary.

Employee Table (E2): Fragment with Qualification .

Employee Table (E_1)

Emp No.	Emp Name	Salary
1	Manoj	30000
2	Rahul	40000
3	Krishna	25000
4	Vikash	60000
5	Vivek	29000

Employee Table (E_2)

Emp No.	Qualification
1	M.Tech
2	M.Tech
3	M.Tech
4	P.hD.
5	M.Tech

3.1.6 Hybrid Fragmentation: It is the combination of both horizontal and vertical fragmentations. Hybrid fragmentation is also called mixed fragmentations. In this table is divided into arbitrary blocks, based on requirement. [8]. This fragmentation is very complex one, which needs more management. This fragmentation is defined using the selection and projection operations of relational algebra.

Emp No.	Emp Name	Salary	Qualification
	Fragment 1		Fragment 2
	Fragment 3		
			Fragment 4

3.2 Data Replication: In replication, multiple copies of the data are held in different locations, and different processes work with different copies. Caching is an Example of replication .The problem in managing replicated data is to maintain the consistency of the data. The Advantages of replication is to increased availability, increased reliability, improved response time, reduced network traffic, improved system throughput and better scalability. Replicated data are subject to the mutual consistency rule. The mutual consistency rule requires that all copies of data fragments be identical. Therefore, to maintain data consistency among the replicas, the DDBMS must ensure that a database update is performed at all sites where replicas exist.

Three replication scenarios exist:

1. A fully replicated database stores multiple copies of each database fragment at multiple sites.
2. A partially replicated database stores multiple copies of some database fragments at multiple sites.
3. An unreplicated database stores each database fragment at a single site. Therefore, there are no database fragments.

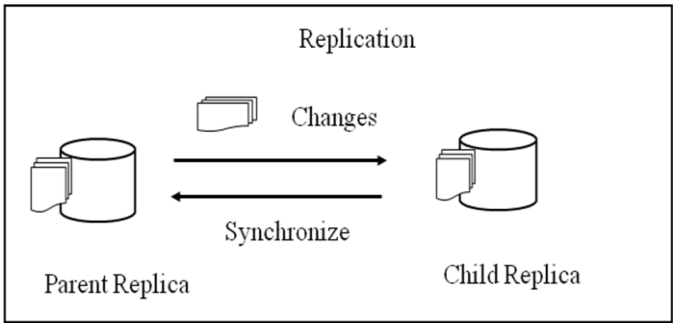


Fig 3 : Data Replication

3.3 Data Allocation: Data Allocation describes the process of deciding where to locate data.[7]. Data allocation strategies are as follows:

1. With centralized data allocation, the entire database is stored at one site
2. With partitioned data allocation, the database is divided into two or more disjoined parts and stored at two or more sites.
3. With replicated data allocation, copies of one or more database fragments are stored at several sites.

4. Distributed Concurrency Control:

Concurrency control: It is the activity of processing concurrent accesses to a database in Distributed Database System. The most common distributed concurrency control technique is strong strict two-phase locking. In database systems and transaction processing, distributed concurrency control refers primarily to the concurrency control of a distributed database.[4] The major goal for distributed concurrency control is distributed serializability for multi-database systems.

Transaction: It is set of read / writes operation used to perform a unit of work. Database transaction must be atomic, consistent, isolated and durable. Database practitioners often refer to these properties of database transactions using the acronym ACID.

Distributed Transaction: In this, each computer features a local transaction manager. If the transaction works at several computers, the transaction managers communicate with various other transaction managers by means of superior or subordinate relationships, which are accurate only for a specific transaction. A distributed transaction is a type of transaction with two or more engaged network hosts. Distributed transaction processing systems are designed to facilitate transactions that span heterogeneous, transaction - aware resource managers in a distributed environment. [4].

The ACID properties are primarily concerned with achieving the concurrency transparency.

1. Atomicity: Either all of the operations in a transaction are performed or none of them are, in spite of failures.
2. Consistency: The execution of interleaved transactions is equivalent to a serial execution of the transactions in some order.
3. Isolation: Partial results of an incomplete transaction are not visible to others before the transaction is successfully committed.
4. Durability: The system guarantees that the results of a committed transaction will be made permanent even if a failure occurs after the commitment.

4.1 Basic Timestamp Ordering Algorithm

Time Stamp can be used to determine the out datedness of a request with respect to the data object it is operating on and to order events with respect to one another. Timestamp is a unique identifier used to identify a transaction. In this algorithm transaction is ordered based on their timestamp values. The timestamp-ordering protocol ensures serializability among transaction in their conflicting Read and Write operations. [11]. This is the responsibility of the protocol system that the conflicting pair of tasks should be executed according to the timestamp values of the transactions.

Rule:

- Transaction T issue R (A) operation.
  If WTS>TS (T)
  Then rollback T
  Otherwise
  Execute R (A) successfully
  and set RTS (A) = max {RTS (A), TS (T)}.
- Transaction T issues W (A) operation.
  If RTS (A) > TS (T)
  Then rollback T
  If WTS (A) > TS (T)
  Then rollback T
  Otherwise
  Execute W (A) successfully
  and set WTS (A) =TS (T).

4.2 Distributed Two-Phase Locking (2PL):

Transaction are not allowed to request any lock if it is already performed any unlock. To ensure serializability of parallel executed transactions elaborated different methods of concurrency control. One of these methods is locking method. . Two phase locking protocol is basic concurrency control protocols in distributed database systems. The main approach of this protocol is "read any, write all". [6] The 2PL Protocol oversees locks by determining when transactions can acquire and release locks. In 2PL each transaction executed in two phases in two steps: Growing Phase: The transaction obtains locks. Shrinking Phase: The transaction releases locks the lock point is the moment when transitioning from the growing phase to the shrinking phase. It generates conflict serializable schedule. In Distributed 2PL Lock managers are distributed to all sites. Each lock manager responsible for locks for data at that site. Problem is deadlock handling more complex.

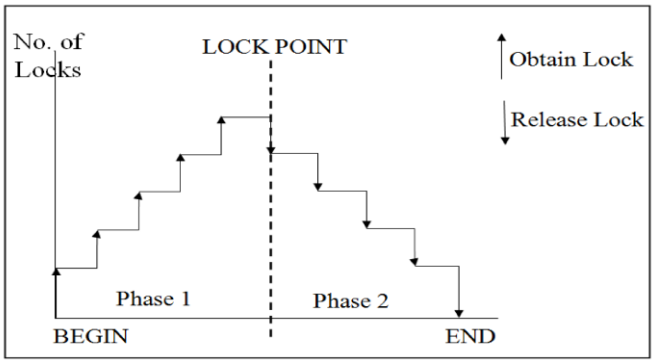


Fig 4 : Growing and shrinking phase in 2PL Protocol

4.3 Distributed Optimistic Protocol (OPT)

It operates by exchanging certification information during the commit protocol. For each data item WTS (write timestamp) and RTS (read timestamp) are maintained. For each read transaction must remember the version identifier associated with the item when it was read. When all the transaction's cohorts completed their work, and have reported back to the master, the transaction is assigned a globally unique timestamp. This time stamp is sent to each cohort in the "prepare to commit" message, and it is used to locally certify all of its reads and writes as follows [11].

A read request is certified if

- The version that was read is still the current version of the item
- No write with a newer timestamp has already been locally certified

A write request is certified if

- No later reads have been certified and subsequently committed
- No later reads have been locally certified already

4.4 Wait-Die & Wound-Wait Algorithms:

WAIT-DIE Rule: If T_i requests a lock on a data item which is already locked by T_j, then T_i is permitted to wait iff $ts(T_i)<ts(T_j)$. If $ts(T_i)>ts(T_j)$, then T_i is aborted and restarted with the same timestamp.[6],[11].

- if $ts(T_i)<ts(T_j)$ then T_i waits else T_i dies
- non-preemptive: T_i never preempts T_j
- prefers younger transactions

WOUND-WAIT Rule: If T_i requests a lock on a data item which is already locked by T_j , then T_i is permitted to wait iff $ts(T_i)>ts(T_j)$. If $ts(T_i)<ts(T_j)$, then T_j is aborted and the lock is granted to T_i.

- if $ts(T_i)<ts(T_j)$ then T_j is wounded else T_i waits
- preemptive: T_i preempts T_j if it is younger
- prefers older transactions

5. Security in distributed database system:

In distributed database system protection of data from unauthorized access is the main goal of security. To protect data or information from modification or misuse we are using security in distributed database system.

5.1 Data security aspects of client / server

There are five approaches in security aspect of client/server

- The work station approval mechanism of users may be partial or non – existent.
- It is possible to carry out automation of the Login procedure.
- The work station may be installed in a public area or in a high risk area.
- The work station may active strong utilities or development devices and thereby tries to bypass the security mechanism.
- In extreme cases the users may pretend to be another user and infiltrate the system.

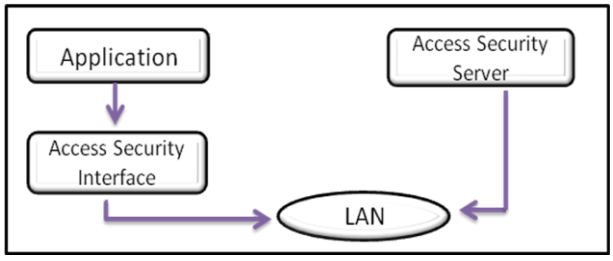


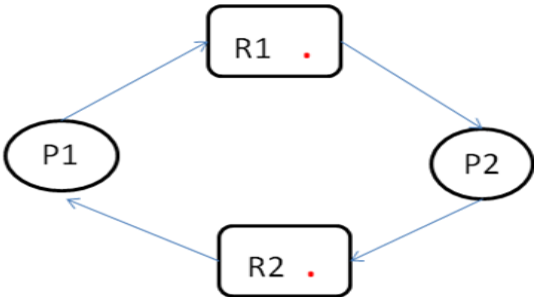
Fig 5 : Distributed System's Security Management

5.2 Distributed computer systems have four main security components:

- Security Authentication – Authentication stands for verifying the identity of someone as user or device who wants to use data, resources, or applications. Authentication also enables accountability by making it possible to link access and actions to specific identities. Authentications is realized by "smart token" which is a hardware device in the size of a pocket computer or credit card that creates a password and transfer it to the authentication server that is linked up to the network.
- Authorization – Authorization is a security mechanism. It is used to determine user/client privileges or access levels related to system resources. The only authorized users are allowed to access the service of system.
- Encryption - It is process to encoding message
- Access control – Access control permits management to specify what users can do, which resources they can access and what operations they can perform.

6 Problems in distributed Database System

Deadlock is the major problem in distributed system. Deadlock is the state of process where set of process are waiting for each other.[3]. Process P1 requesting the resource R1 which is hold by p2. Process p2 requesting the resource of R2 which is hold by P1. So this results in deadlock.



6.1 Deadlock Detection

To detect deadlock in distributed system each site maintain wait for graph. If there is any cycle in system then there is a deadlock. If there is no cycle in wait for graph then there may or may not be deadlock in system.

6.2 Deadlock Recovery

Process termination is the general approach for deadlock recovery. Process is terminated based on the priority of process, cost of restarting the process and the current state of the process. For recovery we identify that if there is a failure, what type was it, and at which site did that happen. Dealing with distributed recovery requires aspects include: database logs, and update protocols, transaction failure recovery protocol, etc.

7. Results: This is Throughput in the model of Distributed 2PL with Timestamps mechanism of concurrency control algorithm. We can say that 2PL algorithms with Time stamp mechanism are effectively enough for concurrency control in the DDBMS. Here it is shown that simulations of the developed timestamp model Algorithm for ordering in DDBMS considering the same input flow intensity and the same data element replicas distributions.

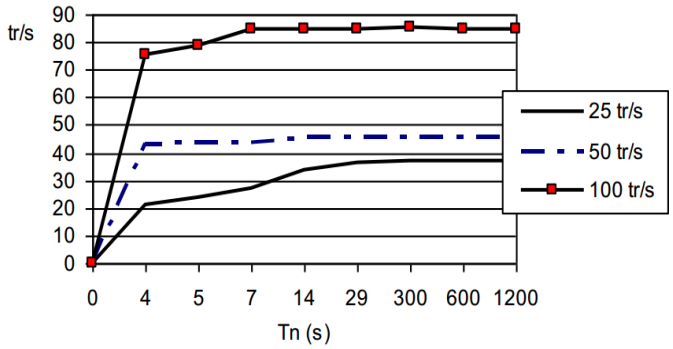


Fig 6 : Throughput in the model of Distributed 2PL with Time stamps mechanism

The Parallel transaction execution pattern investigates the performance of the four algorithms WW, 2 PL, BTO and OPT transaction algorithm.

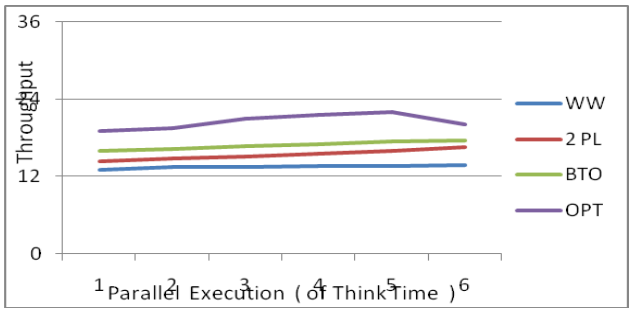


Fig 7: Performance of transaction Algorithm

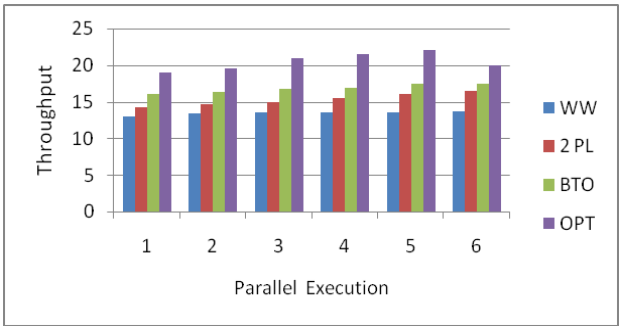


Fig 8: Performance chart of transaction Algorithm

The above comparison chart clearly shows the parallel performance comparisons between the algorithms as discussed above.

8. Conclusion:

In this paper we have described fragmentation, replication and security issues in distributed database system. Here we have shown advantageous side of distributed database system.

Distributed database system is considered to be more reliable than centralized database system. We also described the concurrency control algorithms as basic timestamp ordering, distributed 2PL, optimistic algorithm and wound-wait. It is really important for database to perform the ACID properties. We have discussed the basic concept of concurrency control in distributed database systems and also discussed the various techniques for concurrency control in distributed environments. Now a day's distributed database systems becomes very important for computer science. Many organizations are now deploying distributed database systems. To ensure that systems operate in a secure environment and integrity, Security is concerned with the assurance of confidentiality, integrity, and availability of information in all forms. It is clear from the study that distribution of data involves the problem of deadlock. We need to find out the methods to data distribution and accessing which leads to minimization of deadlock and thus resulting in proper utilization of resources.

References

[1] D. G. Shin, and K. B. Irani, "Fragmenting relations horizontally using a knowledge based approach," IEEE Transactions on Software Engineering (TSE), Vol. 17, No. 9, pp. 872–883, 1991.

[2] E. S. Abuelyaman, "An optimized scheme for vertical partitioning of a distributed database," Int. Journal of Computer Science & Network Security, Vol. 8, No.1, 2008.

[3] Gupta Dhiraj and Gupta V.K., Approaches for Deadlock Detection and Deadlock Prevention for Distributed, Res. J. Recent Sci., 1(ISC-2011), 422-425 (2012)

[4] Sheetlani Jitendra and Gupta V.K., Concurrency Issues of Distributed Advance Transaction Process, Res. J. Recent Sci., 1(ISC-2011), 426-429 (2012)
[5] M. Tamer Özsu, Patrick Valduriez, "Distributed Database Systems: Where Are We Now?" Appeared in IEEE Computer, Vol. 24, No. 8, August 1991.
[6] Bernstein P and Goodman N "Fundamental Algorithm for Concurrency Control in Distributed Database Systems"Tech. Rep., Computer Corp. of America, Cambridge, MA, 1980.
[7] Swati Gupta, Kuntal Saroha, Bhawna, Fundamental Research of Distributed Database, IJCSMS International Journal of Computer Science and Management Studies, Vol. 11, Issue 02, Aug 2011
[8] Arun Kumar Yadav& Ajay Agarwal, An Approach for Concurrency Control in Distributed Database System, Vol. 1, No. 1, January-June (2010)
[9] Tiwari Nitin, Solanki Rajdeepsingh and Pandya Gajrajsingh, Intrusion Detection and Prevention System(IDPS) Technoloy- Network Behavior Analysis System (NBAS), ISCA J. Engineering Sci., 1(1), 51-56 (2012)
[10] Md. Tabrez Quasim, (2013). An Efficient Approach For Concurrency Control In Distributed Database System. Indian Streams Research Journal, Vol. III, Issue. IX
[11] Bernstein P and Goodman N "Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems," Proc. 6th VLDB Cot& Mexico City, Mexico, Oct.1980
[12] M. AlFares et al, "Vertical Partitioning for Database Design: A Grouping Algorithm", in Proc. International Conference on Software Engineering and Data Engineering (SEDE), 2007, pp. 218-223.

Author Profile [1]

Manoj Kumar Sah received the M.Tech. Degrees in Computer Science and Engineering from *Indian School of Mines, Dhanbad* (**ISM Dhanbad**) in 2014. He now with **Shobhit University**, Gangoh, UP, India as Assistant Professor in Dept of Computer Science and Engineering.

Author Profile [2]

Vinod Kumar ,He now with **Shobhit University** , Gangoh , UP, India as Assistant Professor in Dept. of Computer Science and Engineering.

Author Profile [3]

Ashish Tiwari received the M.Tech. Degrees in Computer Science and Engineering from *Central University of Rajasthan, Kishangarh, India* in 2013. He now with Arya Institute of Engineering & Technology, Jaipur , India as Assistant Professor in Dept. of Computer Science and Engineering.