A Permutation Gigantic Issues in Mobile Real Time Distributed Database : Consistency & Security

Gyanendra Kr. Gupta
Asstt. Professor
Computer Science Deptt
Kanpur Institute of Technology
Kanpur, India

A. K. Sharma
Associate rofessor
Computer Sc. & Engg. Deptt.
M.M.M. Engg. College, Gorakhpur

Vishnu Swaroop
PhD Scholor
Computer Sc. & Engg. Deptt.
M.M.M. Engg. College, Gorakhpur

***Abstract*----Several shape of Information System are broadly used in a variety of System Models. With the rapid development of computer network, Information System users concern more about data sharing in networks. In conventional relational database, data consistency was controlled by consistency control mechanism when a data object is locked in a sharing mode, other transactions can only read it, but can not update it. If the traditional consistency control method has been used yet, the system's concurrency will be inadequately influenced. So there are many new necessities for the consistency control and security in Mobile Real Time Distributed Database (MRTDDB). The problem not limited only to type of data (e.g. mobile or real-time databases). There are many aspects of data consistency problems in MRTDDB, such as inconsistency between characteristic and type of data; the inconsistency of topological relations after objects has been modified. In this paper, many cases of consistency are discussed. As the mobile computing becomes well-liked and the database grows with information sharing security is a big issue for researchers. Mutually both Consistency and Security of data is a big confront for researchers because whenever the data is not consistent and secure no maneuver on the data (e.g. transaction) is productive. It becomes more and more crucial when the transactions are used in non-traditional environment like Mobile, Distributed, Real Time and Multimedia databases. In this paper we raise the different aspects and analyze the available solution for consistency and security of databases. Traditional Database Security has focused primarily on creating user accounts and managing user rights to database objects. But in the mobility and drifting computing uses this database creating a new prospect for research. The wide spread use of databases over the web, heterogeneous client-server architectures, application servers, and networks creates a critical need to amplify this focus.**

Keywords- Data sharing; data consistenc; concurrency control & recover; mobile database; real-time databas; security; authentication; access control.

I. INTRODUCTION

All The growing ability to communicate computers through internetworking, wireless network, high bandwidth satellite and cable network has spawned a new class of information centered applications based on data dissemination. In contrast to traditional, where data are delivered from servers to clients on demand, a wide range of emerging database applications benefit from a broadcasts mode for data dissemination. In such applications, the server repetitively broadcasts data to client inhabitants without a specific request. Clients monitor the broadcast channel and salvage the data items they need as they arrive on the broadcast channel. As the data dissemination systems continue to involve, more and more sophisticated client applications will require reading current and consistence data despite updates at the server. Since many data item in mobile computing systems are used to record the real time information. Trough this paper we appraise the problem of disseminating consistent data item to mobile transactions while allowing update to be executed concurrently on the database

server. Consistent data items are provided to mobile transactions by requiring the mobile transactions to read data items committed at the same point of time.

The important characteristics of the mobile database systems are that the values of the data items are highly dynamic and the arrival rates of the updates can be very high. In such a dynamic environment it is difficult to maintain a strict consistency between the external environment and the corresponding values of the data items n the database.

II. CONSISTENCY IN DATABASES

Data consistency summarizes the validity, accuracy, usability and integrity of related data between applications and across the IT venture. This ensures that each user observes a consistent view of the data, including visible changes made by the user's own transactions and transactions of other users or processes. Data Consistency problems may arise at any time but are frequently introduced during or following recovery situations when backup copies of the data are used in place of the original data [1]. Preserving consistency with acceptable performance under conditions of weak connectivity is a difficult challenge. Indeed, it has been widely accepted since the early days of mobile computing that shared data access involves a fundamental tradeoff between consistency, good performance, and tolerance of poor network quality. However, failing to preserve consistency undermines the very attribute that makes databases so attractive for many applications.[2]

Preserving consistency with acceptable performance under conditions of weak connectivity is a difficult challenge. Indeed, it has been widely accepted since the early days of mobile computing that shared data access involves a fundamental tradeoff between consistency, good performance, and tolerance of poor network quality [3]. This has led to a variety of approaches that relax consistency. However, failing to preserve consistency undermines the very attribute that makes databases so attractive for many applications.

Various kinds of data consistency have been identified. These include Point-in-Time Consistency, Transaction Consistency, Application Consistency, and Data Consistency.

A. Point in Time Consistency:

Data is said to be Point in Time consistent if all of the interrelated data components are as they were at any single instant in time. This type of consistency can be visualized by picturing a data center that has experienced a power failure. Before the lights come back on and processing resumes, the data is considered time consistent, due to the fact that the entire processing environment failed at the same instant of time. Different types of failures may create a situation where Point in Time consistency is not maintained. Even if the consistence data is received after the specified time it meaning is changed and the information loss at the point.

B. Transaction Consistency:

A transaction is a transformation of state which has the properties of atomicity (all or nothing), durability (effects survive failures) and consistency (a correct transformation). The transaction concept is a key to the structuring of data management applications. Any operation i.e. is read, write or update concerned with transaction that needs accurate data. The concept seems to have applicability to programming systems in general [4].

In some kinds of failure, the data will not be transaction consistent. In most cases what occurs is that once the application or database is restarted, the incomplete transactions are identified and the updates relating to these transactions are either “backed-out” or processing resumes with the next dependant write [5].

C. Application Consistency

Application Consistency is similar to Transaction consistency. Because application is formed to execute the transaction in a number of transaction grouped together. Instead of data consistency within the scope of a single transaction, data must be consistent within the confines of many different transaction streams from one or more applications. An application may be made up of many different types of data, such as multiple database components, various types of files, and data feeds from other applications. Application consistency is the state in which all related files and databases are in-synch and represent the true status of the application.

D. Data Consistency

Data Consistency refers to the usability of data and is often taken for granted in the single site environment. Data Consistency problems may arise even in a single-site environment during recovery situations when backup copies of the production data are used in place of the original data. Unusable data is consistence or not it does not means but once the data come in a meaning and useful form it must be consistence.

E. Data Loss vs. Data Consistency

How does one reconcile the possibility of lost data versus the integrity and consistency of the data? Often times, traditional backups were created while files were being updated. Eventually, backups created in this fashion were referred to as “fuzzy backups” as neither the consistency nor the integrity of the data could be assured.

One might think it is better to capture as many updates as possible, even if the end result is not consistent. Let us consider this point within the confines of a "typical" large systems data center. For the sake of discussion, let us assume that there are many applications sharing data on hundreds of logical volumes in many thousands of data sets. What happens to the integrity of the data if some updates are applied and others are not? Should this occur, the data is in an artificial state, one that is neither time, transaction nor application consistent? When the applications are restarted, it is likely that some data will be duplicated, while other data will still be missing. The difficulty here is in identifying which updates were successful, which updates caused erroneous results and which updates are missing.

In most cases it is preferable to have time consistent data, even if a few partial transactions are lost or rolled back in the process.

Data loss can be defined as data that is lost and cannot be recovered by another means. Often, individual transactions or files can be restored or recreated, which is inconvenient, but does not represent a true loss of data. Even in cases where some transactional data cannot be recreated or recovered by the data center support teams, it can sometimes be re-entered by the end user if necessary.

If considering an asynchronous Business Continuity and Disaster Recovery solution, it is important to understand that some updates may be lost in flight. However, the greater consideration is that the asynchronous solution you select provides you time consistent data for all of your interrelated applications. In this way, recovery is similar to the process necessary to achieve Transaction and Application Consistency following an outage at the primary site.

Data loss does not imply a loss of data integrity. However, given a choice, most organizations will protect data consistency—for example, ensuring that bank deposits and withdrawals occur in the proper sequence so that account balances reflect a consistent picture any given point in time. This is preferable to processing transactions out of sequence, or, to use our banking example again, to record the withdrawal and not the preceding deposit.

III. FUZZY BACKUP PROBLEM - AN IDEA

For a set of backup data to be of any value it needs to be consistent in some fashion; Time, Transaction or Application consistency is required. For an individual data set, one with no dependencies on any other data, this can be accomplished by creating a simple Point in Time copy of the data and ensuring that the data is not updated during the backup process.

In fact, there are three different possible outcomes, should this fuzzy backup be restored:

- The data is accidentally consistent and useable. This is a happy circumstance that may or may not be repeatable.
- The data is not consistent and not useable. A subsequent attempt to use the data detects the errors and abnormal end subsequent processing.
- The data is NOT consistent, but does not cause an ABEND and happens to be useable to the application. Subsequent processing uses it and any data errors go undetected and uncorrected. This is the worst possible outcome.

In this greater context, simple data consistency within individual data sets is no longer sufficient. What is required is time consistency across all of the interdependent data. As it is impossible to achieve this with the traditional backup methodologies, newer technologies are required to support time consistent data?

To guarantee the correct results and consistency of databases, the conflicts between transactions can be either avoided, or detected and then resolved. Most of the existing mobile database CC techniques use the (conflict) serializability as the correctness criterion. They are either pessimistic if they avoid conflicts at the beginning of transactions, or optimistic if they detect and resolve conflicts right before the commit time, or hybrid if they are mixed. To fulfill this goal, locking, timestamp ordering (TO) and serialization graph testing can be used as either a pessimistic or optimistic algorithm.

IV. SECURITIES IN DATABASES

Database security is the system, processes, and procedures that protect a database from unintended activity. Unintended activity can be categorized as authenticated misuse, malicious attacks or inadvertent mistakes made by authorized individuals or processes.

Traditionally databases have been protected from external connections by firewalls or routers on the network perimeter with the database environment existing on the internal network opposed to being located within a demilitarized zone. Additional network security devices that detect and alert on malicious database protocol traffic include network intrusion detection systems along with host-based intrusion detection systems.

One of the main issues faced by database security professionals is avoiding inference capabilities. Basically, inference occurs when users are able to piece together information at one security level to determine a fact that should be protected at a higher security level. Database security is more critical as networks have become more open [6].

Databases provide many layers and types of information security, typically specified in the data dictionary, including:

- Access control
- Auditing
- Authentication
- Encryption
- Integrity controls

Database security can begin with the process of creation and publishing of appropriate security standards for the database environment. The standards may include specific controls for the various relevant database platforms; a set of best practices that cross over the platforms; and linkages of the standards to higher level polices and governmental regulations.

Access Control is a term taken from the linguistic world of security. In general, it means the execution of limitations and constrictions on whoever tries to occupy a certain protected property. Guarding an entrance of a person is also a practice of access control. There are many types of access control. Some of them are mentioned in this article. You, the reader of this article, will have several types of access control around you. Nowadays, almost every computer user has a firewall or antivirus is running on every computer, a popup blocker and many other programs. All of these are with access control functions. All of these programs guard us from intruders of sorts. They inspect everything trying to enter the computer and let it in or leave it out. Computers have complicated access control abilities. They ask for authentication and search for the digital signatures. Also, there are different types of keypads and access control systems. In today's world the keys and locks are beginning to look different. With the passage of time, the key locks also got smarter. They can identify the patterns of your physical features, your voice, and fingerprint locks can read your fingerprints [7].

Access control is a rapidly growing market and soon may manifest itself in such ways we cannot even imagine. Nowadays, security access control is a necessary component for businesses [8]. There are many ways to create this security. Some companies hire a security guard to stand in the gateway. There are many security devices that prevent or permit access such as a turnstile. Computers operate the best most effective access control systems.

Auditing is a computer security audit is a manual or systematic measurable technical assessment of a system or application. Manual assessments include interviewing staff, performing security vulnerability scans, reviewing application and operating system access controls, and analyzing physical access to the systems. Automated assessments include system generated audit reports or using software to monitor and report changes to files and settings on a system. Systems can include personal computers, servers, mainframes, network routers, switches. Applications can include Web Services, Databases [9].

Authentication is the process of confirming a user or computer's identity. The process normally consists of four steps:

1. The user makes a claim of identity, usually by providing a username. For example, It might make this claim by telling a database that my username is something.
2. The system challenges the user to prove his or her identity. The most common challenge is a request for a password.
3. The user responds to the challenge by providing the requested proof. In this example, It would provide the database with my password.
4. The system verifies that the user has provided acceptable proof by, for example, checking the password against a local password database or using a centralized authentication server

Encryption is good. It helps make things more secure. However, the idea that strong cryptography is good security by itself is simply wrong. Encrypted messages eventually have to be decrypted so they are useful to the sender or receiver. If those end-points are not secured, then getting the plain-text messages is trivial. This is a demonstration of a crude process of accomplishing that. There is no dispute about the need for strong encryption, particularly for privileged communications. There is no way to have a high level of assurance that the entire path between endpoints of a message is secure, so the message has to be hidden in transit. While brute-force decryption is possible, modern forms of encryption have made this process too long to be valuable.

Computer security authentication means verifying the identity of a user logging onto a network. Passwords, digital certificates, smart cards and biometrics can be used to prove the identity of the user to the network. Computer security authentication includes verifying message integrity, e-mail authentication and MAC (Message Authentication Code), checking the integrity of a transmitted message [10]. Human authentication is the verification that a person initiated the transaction, not the computer. Challenge-response authentication is an authentication method used to prove the identity of a user logging onto the network [11]. When a user logs on, the network access server (NAS), wireless access point or authentication server creates a challenge, typically a random number sent to the client machine. The client software uses its password to encrypt the challenge through an encryption algorithm or a one-way hash function and sends the result back to the network. This is the response. Two-factor authentication requires two independent ways to establish identity and privileges. The method of using more than one factor of authentication is also called strong authentication. This contrasts with traditional password authentication, requiring only one factor in order to gain access to a system. Password is a secret word or code used to serve as a security measure against unauthorized access to data. It is normally managed by the operating system or DBMS. However, a computer can only verify the legality of the password, not the legality of the user.

Mobile agents are processes which can autonomously migrate to new hosts. Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. The primary added complication is that, as an agent traverses multiple hosts that are trusted to different degrees, its state can change in ways that adversely impact its functionality. Many of the most important applications of mobile agents will occur in fairly uncontrolled, heterogeneous Environments. As a consequence, we cannot expect that the participants will trust each other. More-over, interpreters may disclose the secrets of visiting agents, and may attempt to manipulate their state. Existing techniques, intended for distributed systems in general, certainly allow substantial protection within the broad outlines of these constraints. However, substantial investment in mobile agent systems may await further work on new security techniques specially oriented toward mobile agents. [12]

Threats, vulnerabilities, and countermeasures for the currently predominating static distributed systems have been studied extensively; sophisticated distributed system security architectures have been designed and implemented These architectures use the access control model, which provides a basis for secrecy and integrity security policies.[13.14]

The process of deducing which principal made a request is called authentication. In a distributed system, authentication is complicated by the fact that a request may originate on a distant host and may traverse multiple machines and network channels that are secured in different ways and are not equally trusted [15]. The process of deciding whether or not to grant a request once its principal has been authenticated is called authorization. The authentication mechanism underlies the authorization mechanism in the sense that authorization can only perform its function based on the information provided by authentication, while conversely authentication requires no information from the authorization mechanism. Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. The primary added complication is that, as an agent traverses multiple machines that are trusted to different degrees, its state can change in ways that adversely impact its functionality. [16, 17, 18]

V. DATABASE SECURITY ISSUES: DATABASE SECURITY PROBLEMS AND HOW TO AVOID THEM

Database security managers play a vital role in any organization and are required to multitask and juggle a variety of headaches that accompany the maintenance of a secure database. Once it is understood that how, where, and why database security can prevent [19, 20].

- Daily Maintenance: Database audit logs require daily review to make certain that there has been no data misuse. This requires overseeing database privileges and then consistently updating user access accounts.

- Varied Security Methods for Applications: More often it can create difficulty with creating policies for accessing the applications. The database must also possess the proper access controls for regulating the varying methods of security otherwise sensitive data is at risk.

- Post-Upgrade Evaluation: When a database is upgraded it is necessary for the administrator to perform a post-upgrade evaluation to ensure that security is consistent across all programs. Failure to perform this operation opens up the database to attack.

- Split the Position: Sometimes organizations fail to split the duties between the IT administrator and the database security manager. Instead the company tries to cut costs by having the IT administrator do everything.

- Application Spoofing: Hackers are capable of creating applications that resemble the existing applications connected to the database. These unauthorized applications are often difficult to identify and allow hackers access to the database via the application in disguise.

- Manage User Passwords: Password rules and maintenance needs to be strictly enforced to avoid opening up the database to unauthorized users.

- Windows OS Flaws: Windows operating systems are not effective when it comes to database security. Often theft of passwords is prevalent as well as denial of service issues. We discuss recent challenges for database security and some preliminary approaches that address some of these challenges.

i. Security Awareness and End-users
ii. Google Exposure
iii. Standard Compliance & Regulations Updates
iv. Vulnerability Management
v. Frequently Change of Management and Lack of Co-ordination in Management

VI. MDRTDB CONSISTENCY ISSUES

A real-time database system (RTDBS) is a transaction processing system that is designed to handle workloads where transactions have completion deadlines. The objective is of the system is to meet these deadlines, that is to complete processing transactions before their deadline expires [21]. In real-time database system (RTDBS), transactions have to be completed before their deadlines and all the accessed temporal data objects have to be valid. Besides meeting these timing constraints, a RTDBS needs to observe data consistency constraints as well [22]. Different transactions scheduling algorithms and concurrency control protocols have been proposed to satisfy these constraints [23, 24]. The problems become more complicated in a distributed real-time database system (DRTDSB) where a database is partitioned into a number of smaller local databases residing at different sites. The communication network, which is an essential component in a DRTDBS, is another important performance issue [25]. A Mobile transaction is distinguished by the features that the operations of it can be submitted by a portable (in small size, less memory and limited power backup) to the data servers from different locations [26].

The Sharing concepts of loading the database by more than one user for any network system are the big issues in MDRTDB system. The rapid growth of mobile computing technology provides a new, alternative platform for mobile distributed real-time database applications [27, 28]. With increasing uses of portable and mobile computers, mobile real-time database applications become more and more popular. We call the new systems as mobile distributed real-time database systems (MDRTDBS). Processing time-constrained transactions in MDRTDBS is a very new area and a lot of design issues still remain unresolved. Mobile transactions are largely limited to soft or firm deadlines because of the execution delays due to the system constraints [29].

The predictability of a MDRTDBS is affected by a number of factors such as concurrency control, priority scheduling, commitment and mobile wireless communication. In a mobile environment, the behavior of the underlying wireless network is highly unpredictable. Certain behavior of the mobile computing system creates an additional burden on the system performance and increases the unpredictability of the system. In particular, the narrow wireless bandwidth and the limitation of processing power in the mobile clients may be bottlenecks to the performance [30]. Mobile clients (MCs) face wide variance in network quality including unpredictable call setup probability. There are several factors which may affect a call to setup such as the availability of the receiver in the cell site and the utilization of the channel in the Main Terminal Switching Office (MTSO).

Even when a call can be set up, the setup time is also fluctuating. Furthermore, MCs always prefer a light, compact, and power-saving units. Methods have to be designed to save power in the MCs. Also, the mobility of the MCs affects the distribution of workload on different cells and thus the system predictability. Failures are not fully avoided in Mobile computing as well as mobility also raises the database dependency on servers moving across one geographical area to another.

Another source of problem is the cost in resolving data conflicts. In MDRTDBS, different real-time concurrency control protocols have been proposed. One of the commonest methods to resolve data conflict between transactions with different priorities is by restarting transactions. However, this will be very expensive under a mobile environment. The restarted transaction will have a very probability of deadline missing. Methods have to be designed to reduce the cost in resolving the conflict. One of the possible solutions to reduce the

number of restarts is to adopt less restrictive correctness criteria for database consistency such as using the concept of similarity or epsilon serializability. Similar problem will occur when a committing transaction has a data conflict with an executing transaction. Up to now, very little of work has been done on the design of real-time commitment protocol which is required to reduce the impact of transaction commitment on the system predictability. Some proposals are based on the optimistic method. Serializibility is widely accepted correctness criteria for controlling concurrent execution of transactions in database systems. Serializable schedules provide correct results and leave the database consistence. However serialzibilty can be restrictive for some mobile applications because of the limitation of concurrency allowed by serialzible executions.

Consistency guarantee for data processed by mobile clients are an important area of research. These guarantees provide the basis for any collaborative work and transaction processing done with these systems [31]. The mechanism to provide individual applications with a view of the database that is consistence with their own actions. This is important since in their environment, clients can read or write data from any one of the available server and these servers can contain in consistence views of the database [32,33].

VII. CONCLUSION

Concurrency elevates the intricacy of consistency and sharing grows the security issues in any environments. There are several techniques and protocols have been built for maintaining the consistency and security of database systems like Data Consistency. Two-Process Mutual Exclusion: Dekker's- and Peterson's Algorithms, N-Process Mutual Exclusion using Hardware, N-Reader, 1-Writer Mutual Exclusion using Head/Tail Flags. But the available techniques are not sufficient for the different database environment where the data is huge and complex for transactions including security system. Security is more than Just Good Crypto – The point here is not that encryption is worthless. The point is that encryption by itself is not helpful. The endpoints need to be secure, passwords need to be difficult to crack, and those who do have access to the system need to be trustworthy. System call traces can be used on any kind of process such as e-mail daemons, web servers, or encrypted chat programs. A new protocol must be needed to control both the situation for consistency and security in MDRTDB. In order for any security tool to be effective, it needs to be layered with other strong security tools, starting with a security policy. No one tool, by itself, can ever prevent information theft or attacks, but several layers of security provide the most solid defense against would-be hackers.[34, 35]. Encryption needs to be accompanied by server hardening, intrusion detection, firewalls, and auditing. Without it, encryption is easily compromised.

REFERENCES

[1] Joann J Ordille, Barton P Miller, "Database challenges in Global Information System", International Conf. on Management of Data Processing, ACM SIGMOD, Washington,DC,US, pp. 403-407, 1993.

[2] Improving Mobile Database Access Over Wide-Area Networks Without Degrading Consistency Niraj Tolia, M. Satyanarayanan, and Adam Wolbach Carnegie Mellon university {ntolia,satya,awolbach}@cs.cmu.ed

[3] BADRINATH, B. R., AND PHATAK, S. H. On clustering in database servers for supporting mobile clients. ClustercComputing 1, 2 (1998), 149–159.

[4] Jim Gray, "The Transaction Concept: Virtues and Limitations" , 7th International Conference, Cannes, France, Proceedings, IEEE Computer Society, pp. 144-54,Sep. 9-11,1981.

[5] Song X, Liu W. S., "Maintaining Temporal Consistency: Pessimistic vs Optimistic concurrency control", IEEE Transactions on Knowledge Engineering, 7(5):786-796, Oct 1995.

[6] S. Srinivasan, Anup Kumar,"Database security curriculum in InfoSec program", Proceedings of the 2nd annual conference on Information security curriculum development, Kennesaw, Georgia, Sep. 23-24, 2005.

[7] E. Bertino D. Leggieri and E. Terzi, "Securing DBMS: Characterizing and Detecting Query Flood", Proc. Ninth Information Security Conf. (ISC '04), Sept. 2004.

[8] Elisa Bertino , Barbara Catania , Elena Ferrari, "A nested transaction model for multilevel secure database management systems", ACM Transactions on Information and System Security (TISSEC), v.4 n.4, p.321-370, November 2001.

[9] B. Thuraisingham, "Database and Applications Security: Integrating Databases and Applications Security", CRC Press, Dec. 2004.

[10] A Kush, "Security Aspects in Adhoc Routing", CSI of India Communication, Vol No 32 Issue 11, pp. 21-33, March 2009.

[11] Walid Rjaibi, "An introduction to multilevel secure relational database management systems", Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, Markham, Ontario, Canada, p.232-241, October 04-07, 2004.

[12] Security for Mobile Agents: Issues and Requirements William M. Farmer, Joshua D. Guttman, and Vipin Swarup The MITRE Corporation 202 Burlington Road Bedford, MA 01730-1420.

[13] B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: Theory and practice. ACM Transactions on Computer Systems, 10:265{310, November 1992. http://DEC/SRC/research-reports/abstracts/src-rr-083.html.

[14] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In Proceedings of the Usenix Winter Conference, pages 191{202, 1988).

[15] Elisa Bertino , Laura M. Haas, "Views and Security in Distributed Database Management Systems", Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology, p.155-169, March 14-18, 1988.

[16] Elisa Bertino , Sushil Jajodia , Pierangela Samarati, "Database security: research and practice", Information Systems, v.20 n.7, p.537-556, Nov. 1995.

[17] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik. Itinerant agents for mobile computing. IEEE Personal Communications Magazine, 2(5):34{49, October 1995. http://www.research.ibm.com/massive.

[18] J. Tardo and L. Valente. Mobile agent security and Telescript. In IEEE CompCon, 1996. http://www.cs.umbc.edu/agents/security.html.

[19] C. Thirunavukkarasu, T. Finin, and J. May_eld. Secret agents | a security architecture for KQML. In CIKM Workshop on Intelligent Information Agents, Baltimore, December 1995.

[20] Jayant R Haritsa, "Data Access Sceduling in Firm Real time Database Systems", The Journal of Real Time Systems, Vol 4, pp 203-241, 1992.

[21] O. Ulusoy, "Processing Real-Time Transactions in Replicated database System", International Conf. on Distributed and parallel Databases, 2(4),pp. 405-436, 1994.

[22] P.S. Yu, K.L. Wu, K.L. Lin, and, S.H. Son, "On Real-time Databases: Concurrency Control and Scheduling", Proceedings of the IEEE, vol. 82, no. 1, pp. 140-157, 1994.

[23] Kin, L, Reoat, G, "Consistency issues of Real Time Systems", In proceeding of International Symposium on system Science and Technology, pp. 118-129, 1999.

[24] O. Ulusoy, "Real-time Data Management for Mobile Computing", International Workshop on Issues and Application of Database Technolgy, pp. 223-240, 1998.

[25] A. Elmagaemid J. Jing, O Bukhers, "An efficient and reliable reservation algorithm for mobile transactions", International Conf. on Information and Knowledge Management(CIKM), 1995.

[26] Evaggelia Pitoura and Bharat Bhargava, "Dealing with Mobility: Issues and Re-search Challenges", Technical Report, Purdue Univ., Nov. 1993.

[27] Tomasz Imielinski and B. R. Badrinath, Mobile Wireless Computing: Challenges in Data Management, Communications of the ACM, vol. 37, no. 10, Oct. 1994.

[28] Kayan, E, Ulusoy, O, "An evaluation of Real Time transaction management issues in Mobile Database Systems", The Computer Journal, Vol 42, pp. 501-510, 1999.

[29] Panos K. Chrysanthis, "Transaction Processing in Mobile Computing Environment", in Proceeding of IEEE Workshop on Advances in Parallel and Distributed Systems,pp. 77-82, USA, 1993.

[30] Dainel Barbara, "Mobile computing and Databases-A Survey", IEEE Transactions on Knowledge and data Engineering, Volume 11 No. 1 Jan-Feb,1999.

[31] D Terry, A J Demers, K Peterson, et al, "Session Guarantee for weakly consistent replicated data", Proc. Conf. Parallel and Distributed Computing, Austin, Texsas, Oct 1994.

[32] Y. Jayanta Singh, Yumnam Somananda Singh, Ashok Gaikwad and S C Melhotra, "Dynamic Management of Transactions in Distributed Real- Time Processing System", International Journal of Databse Management System(IJDMS), Vol. 2, No. 2 , pp. 161-170, May, 2010.

[33] Mario Guimaraes , Meg Murray , Richard Austin, "Incorporating database security courseware into a database security class", Proceedings of the 4th annual conference on Information security curriculum development, Kennesaw, Georgia, Sep. 28, 2007.

[34] Norjihan Abdul Ghani , Zailani Mohamed Sidek, "Personal information and privacy in E-commerce application", Proceedings of the 7th WSEAS international conference on Information security and privacy, Cairo, Egypt, p.28-32, December 29-31, 2008,

[35] Charles P. Pfleeger , Shari Lawrence Pfleeger, "Security in Computing", Prentice Hall Professional Technical Reference,200

AUTHORS PROFILE

Gyanendra Kumar Gupta received his Master degree in Computer Application in year 2001 and M.Tech in Information Technology in year 2004. He has worked as Faculty in different reputed organizations. Presently he is working as Asst. Prof. in Compute Science and Engineering Deptt. at KIT ,Kanpur. He has more than 10 years teaching experience. His area of interest includes DBMS, Networks and Graph Theory. His research papers related to Real Time Distributed Database and Computer Network are published in several National & International Conferences. He is pursuing his PhD in Computer Science.

Dr. A.K. Sharma received his Master degree in Computer Science in year 1991 and PhD degree in IIT, Kharagpur in 2005. Presently he is working as Associate Professor in Computer Science and Engineering Department, Madan Mohan Malaviya Engineering College, Gorakhpur. he has more than 23 years teaching experience. His areas of interest include Database Systems, Computer Graphics, Object Oriented Systems. He has published several National & International conferences & journals.

Vishnu Swaroop received his Master degree in Computer Application in year 2002 presently he is working as Computer Programmer in Computer Science and Engineering Department, Madan Mohan Malaviya Engineering College, Gorakhpur. He has more than 20 years teaching and professional experience. His area of interest includes DBMS, & Networks s research papers related to Mobile Real Time Distributed Database and Computer Network are published in several National & International conferences. He is pursuing his PhD in Computer Science.