International Journal of Research and Development in Applied Science and Engineering (IJRDASE)

Priority Queue Concurrency Check Mechanism (PQCCM) Algorithm for Mobile Distributed Real-Time Database System

Saiyed Mohammad Yousuf Rizvi
Computer Science & Engineering,
BBDU, Lucknow, India
yousufrizvi786@gmail.com

S. K. Maurya
Computer Science & Engineering,
BBDU, Lucknow, India
shailmaurya@gmail.com

***Abstract*— In modern time the mobile computing based applications needs advanced data synchronization. They works on a complete set of data management services that uses very accurate data modelling, mobile and server-side support systems that can handle deployment and versioning, rules-based data distribution, bi-directional data transfers in a very fast and secure environment. The mobile device-based database services and tight transaction-level integration with multiple enterprise information sources are in great demand. Due to these reasons the mobile computing environment is developed as a distributed computing. In this work we have developed and investigated the complete database distributed among wireless components as in mobile switching stations. In this approach the entire database is being distributed in wireless components of the computer systems. Some of the parameters that influence and complicate database management are design of database and replication of database. We have developed a mobile environment protocol that can handle the distributed database of several clients using priority based concurrency control mechanism with considerations of Hand Off situation. For this case we have developed our algorithm using MATLAB 2010.An advanced priority queue mechanism is applied to reduce the transaction aborting issues.**

***Keywords*— Mobile Computing, MTSO, MATLAB, WSN.**

1. Introduction:

Recent advances in wireless communication networks and portable computers have led to the emergence of a new research area called mobile computing systems. An important part of the research conducted in mobile computing systems has been done on mobile data management. What make the mobile data management different from the conventional data management are the mobility of the users or the computers connected to the system, and the resource constraints such as wireless bandwidth and battery life. As a result of such distinctive features of mobile systems, the data management techniques developed for conventional distributed database systems may not work well in a mobile environment. Research contributions are required in a variety of areas, such as distribution of data on mobile and/or non-mobile computers, processing of queries and transactions submitted by mobile users, maintaining the consistency of data cached on mobile computers, and so on. Another important issue that needs to be considered in mobile data management is the requirement of processing queries and transactions within certain time limits in order to maintain the temporal validity of the data accessed by those queries and transactions. Our basic objective in this project is a thorough investigation of the issues to develop various types of methods for mobile data management in response to the requirements mentioned.

Many current researchers in the mobile computing arena share the same vision: ubiquitous access to information, data, and applications. Ubiquitous access refers to the ability of users to access these computing resources from almost any terminal. The idea behind the research is to provide dissemination of large amount of useful and needful information to different mobile user by designing the efficient data management policies. Recent developments relating to the Internet are establishing solid foundations for wide-area ubiquitous computing systems. [1, 2] Universal access and management of information has been one of the driving forces in the evolution of computer technology. Central computing gave the ability to perform large and complex computations and advanced information manipulation. Advances in networking connected computers together and led to distributed computing. Web technology and the Internet went even further to provide hyper-linked information access and global computing. However, restricting access stations to physical locations limits the boundary of the vision. The real global network can be achieved only via the ability to compute and access information from anywhere and anytime. This is the fundamental wish that motivates mobile computing. This evolution is the cumulative result of both hardware and software advances at various levels motivated by tangible application needs.[3]

With the rapid advances in mobile computing technology, there is an increasing demand for processing real-time transactions in a mobile environment. Owing to the intrinsic limitations of mobile computing systems, such as limited bandwidth and frequent disconnection, the design of an efficient and cost-effective MDRTDBS requires techniques that are quite different from that in distributed real-time database systems (DRTDBS) which are supported with wired networks [10]. It is much more difficult to meet transaction deadlines in a mobile environment as there exist various factors, such as network performance, concurrency control and transaction scheduling, which can seriously affect the transaction performance. Two of the most important performance objectives are how to meet the urgency of transactions and how to satisfy the temporal constraints of database, where temporal constraints refer to the freshness of data objects in the database [5]. Many real-time database applications are used to monitor the status of the objects in the external environment and they must generate timely responses to critical events. For example, in a stock trading system, a late response to a stock analysis transaction may result in a loss of a good trading opportunity. The consequence of missing a transaction deadline in a telemedicine system for ambulance services may result in a loss of a human life.

One of the most important issues to ensure timeliness of transaction execution is concurrency control. However, the concurrency control protocols for conventional database systems are not suitable to real time database systems. Real-time transactions are critical and have to be scheduled to meet their deadlines. Conventional concurrency control protocols, such as two phase locking (2PL) and optimistic concurrency control method [11], often schedule transactions on an equal basis. Higher-priority transactions may suffer from an unlimited amount of priority inversion time, where priority inversion is a situation in which a higher-priority transaction is blocked by a lower-priority transaction [5].

2. Related Work:

Kam-Yiu Lam et. al. (2000) [6] they proposed a distributed real-time locking protocol, called Distributed High Priority Two Phase Locking (DHP-2PL), for MDRTDBS. With the rapid advances in mobile computing technology, there is an increasing demand for processing real-time transactions in a mobile environment. Based on the High Priority Two Phase Locking (HP-2PL) scheme. In the protocol, the characteristics of a mobile computing system are considered in resolving lock conflicts. Two strategies are proposed to further improve the system performance and to reduce the impact of mobile network on the performance of the DHP-2PL: (1) A transaction shipping approach is proposed to process transactions in a mobile environment by exploring the well-defined behavior of real-time transactions. (2) We explore the application semantics of real-time database applications by adopting the notion of similarity in concurrency control to further reduce the number of transaction restarts due to priority inversion, which could be very costly in a mobile network. A detailed simulation model of a MDRTDBS has been developed, and a series of simulation experiments have been conducted to evaluate the performance of the proposed approaches and the effectiveness of using similarity for concurrency control in MDRTDBS.

The distributed transaction commit problem requires reaching agreement on whether a transaction is committed or aborted. The classic Two-Phase Commit protocol blocks if the coordinator fails. Fault-tolerant consensus algorithms also reach agreement, but do not block whenever any majority of the processes are working. The Paxos Commit algorithm runs a Paxos consensus algorithm on the commit/abort decision of each participant to obtain a transaction commit protocol that uses 2F + 1 coordinators and makes progress if at least F+1 of them are working properly. Paxos Commit has the same stable-storage write delay, and can be implemented to have the same message delay in the fault-free case, as Two-Phase Commit, but it uses more messages. The classic Two-Phase Commit algorithm is obtained as the special F = 0 case of the Paxos Commit algorithm proposed by Jim Gray et al. (2004) [7].

Salman Abdul Moiz et al. (2010), [8] they worked for any database environment either wired or wireless, if multiple host access similar data items it may lead to concurrent access anomalies. As disconnections and mobility are the common characteristics in mobile environment, preserving consistency in presence of concurrent access is a challenging issue. Most of the approaches use locking mechanisms to achieve concurrency control. This leads to increase in blocking and abort rate in mobile environments. However the dynamic timer adjustment strategies may use locking mechanism to efficiently implement concurrency control. To reduce deadlocks and blocking of resources an enhanced optimistic approach for concurrency control is proposed by Salman Abdul Moiz et al. (2010) [8]. To show the effectiveness of the commit protocols in mobile environments, a simulator is designed and implemented to demonstrate how the transactions are committed and how the data consistency is maintained when the transactions are executed concurrently. The simulator was tested for both pessimistic and optimistic approaches.

Managing the transactions in real time distributed computing system is not easy, as it has heterogeneously networked computers to solve a single problem. If a transaction runs across some different sites, it may commit at some sites and may failure at another site, leading to an inconsistent transaction. The complexity is increase in real time applications by placing deadlines on the response time of the database system and transactions processing. Such a system needs to process Transactions before these deadlines expired. A series of simulation study have been performed to analyze the performance under different transaction management under conditions such as different workloads, distribution methods, execution mode-distribution and parallel etc. The scheduling of data accesses are done in order to meet their deadlines and to minimize the number of transactions that missed deadlines. A new concept is introduced to manage the transactions in dynamic ways rather than setting computing parameters in static ways. With this approach, the system gives a significant improvement in performance this approach is proposed by Y. Jayanta Singh et.al. (2010) [9].

Recent advances in wireless communication networks and portable computers have led to the emergence of a new research area called mobile computing systems proposed by Vishnu Swaroop et. al. (2011) [12]. An important part of the research conducted in mobile computing systems has been done on mobile data management. What make the mobile data management different from the conventional data management are the mobility of the users or the computers connected to the system, and the resource constraints such as wireless bandwidth and battery life. As a result of such distinctive features of mobile systems, the data management techniques developed for conventional distributed database systems may not work well in a mobile environment. Research contributions are required in a variety of areas, such as distribution of data on mobile and/or non-mobile computers, processing of queries and transactions submitted by mobile users, maintaining the consistency of data cached on mobile computers, and so on. Another important issue that needs to be considered in mobile data management is the requirement of processing queries and transactions within certain time limits in order to maintain the temporal validity of the data accessed by those queries and transactions. Our basic objective in this project is a thorough investigation of the issues to develop various types of methods for mobile data management in response to the requirements mentioned.

3. Methodology:

We have developed a mobile environment protocol that can handle the distributed database of several clients using priority based concurrency control mechanism with considerations of Hand Off situation.

For this case we have developed our algorithm using MATLAB 2010.The main step of our algorithm is given as below:-

1. Request for data transaction.
2. Checking for nodes Cell Address for allotting Base Station.
3. Checking for the validity of the transaction.
4. Rejection of the node request which have not permission for data transaction.
5. Request acceptance and acknowledgement to the active valid nodes by the base station.
6. Priority listing of the nodes for data transaction.
7. Data transmission of higher priority nodes through the channel.
8. Queue allotment of the remaining nodes in priority listing for data transmission in next round.

The mobile database network that we have considered here has following specifications:

Table 1: Model Parameters and Their Baseline Values

Parameters	**Baseline Values**
System Level	
Number of MTSO	**1**
Number Of Cell Sites	**9**
Location Update Interval	**5**
Number Of Channels For Each Cell Site	
Mobile Network	
Number Of Mobile Clients	**63**
Database	
Number Of Local Databases	**9**

We have considered 'C' clients in a network of area N*N Km Square. The nodes are scattered in 9 different cells randomly. Each cell has a one base station hence we have 9 base station located at the centre of each cells.

The nodes are dynamic and charging their position by dx and dy displacement where dx and dy varies in the range of -0.5 to + o.5 kms. in each round maximum.

The coordinates of C nodes are defined as Cx and Cy hence in every round the position is updated as Cx=Cx + dx and Cy=Cy + dy in each round. The distance between each node from every base station is updated and minimum distance is determined for all the nodes with respect to base station. The nearest station .The nearest station the table below shows the node id of all the nodes and respective x and y coordinates cx and cy and their distance from all he base stations. Using this table for every round the algorithm generates the cell address of all the nodes.

4. Result and Discussion:

The network consists of nodes in different cells along with base station and MTSO is shown in fig. 1.

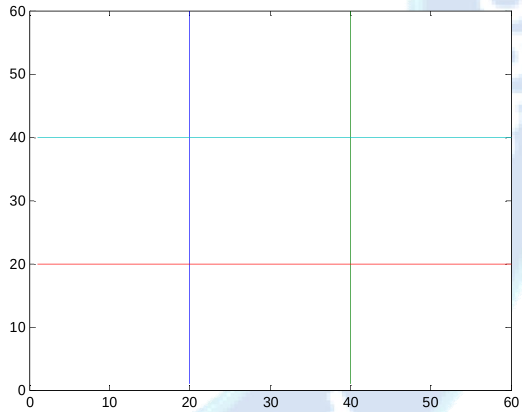

Fig. 1. Mobile Network With Cells

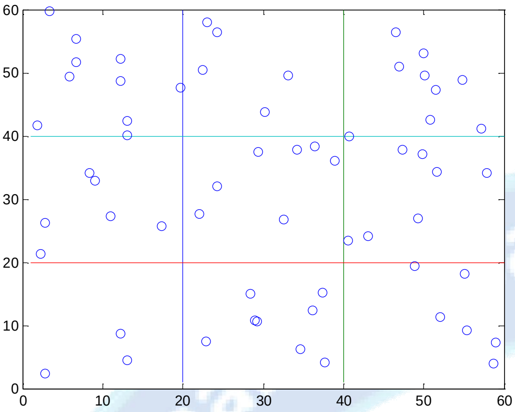

Fig. 1(b). Node Distributed in the network

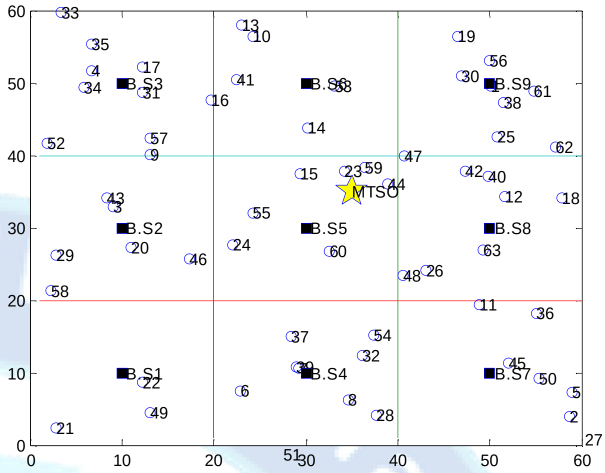

Fig. 1(e). Complete mobile network showing base station BS1 to BS9 nodes 1 to 63 and MTSO

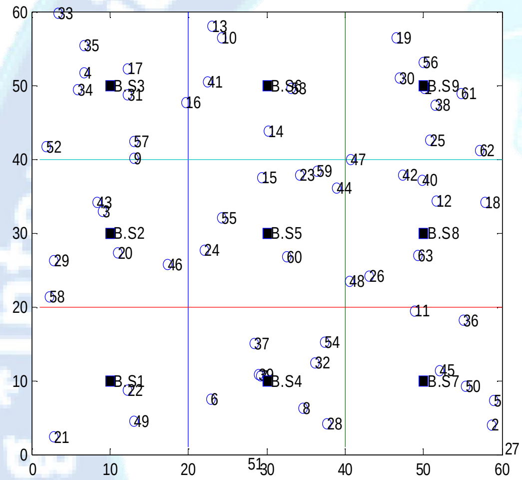

Fig 1(c). Node id allotment

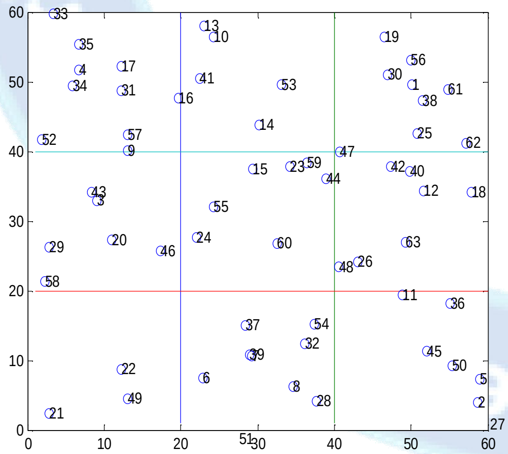

Fig. 1(d). Mobile network with Base station and nodes

Fig 1(a) shows the 9 different cells distributed in the network area. Fig 1(b) shows that the nodes are randomly distributed having coordinates Cx and Cy in different cells. Fig 1(c) shows the phase of our algorithm after allotment of ids to each node. C is the number of cells in the present fig 1(c) is 63. Fig. 1(d) shows the base station position of centre of each cell having number from BS1 to BS9.Fig 1(e) shows one of the arbitrary network whose data is provided in the table node.

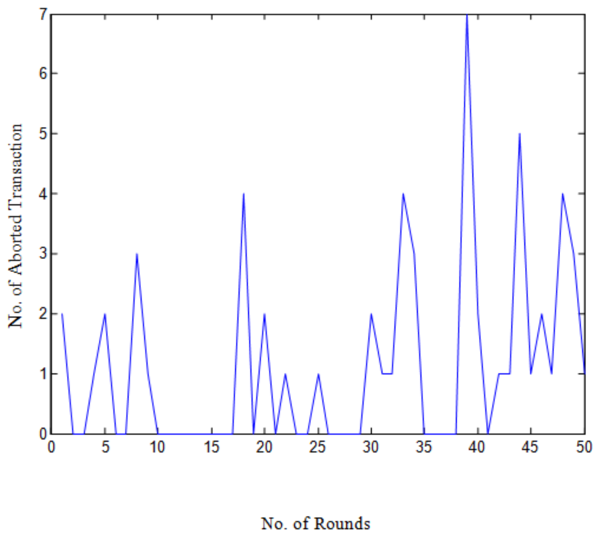

Fig 2. Aborted Transaction at different rounds without PQCCM at 4 channels

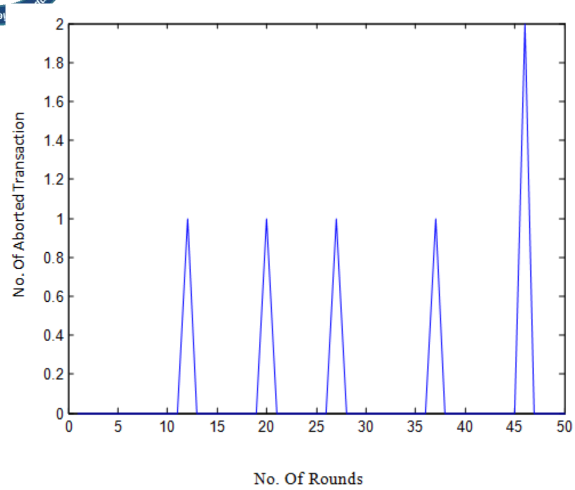

Fig. 3. Aborted Transaction at different rounds with PQCCM at 4 channels.

We have developed three type of algorithm for mobile distributed networks.The intial one the algorithm was without considering the PQCCM concept.In this case we considered 4 channels the transaction report his generated for upto 50 rounds and total number of dropout were observed for each round as shown in figure 2.We can see that there are about 7 transaction abort conditions and the abort condition are very frequent.Similarly figure 3 is for PQCCM based results for report generated for 50 rounds for 4 channels.It has been observed that maximum abort conditions are reduced to 2 packets requests only. Similarly we generated results for 5 channels for the PQCCM based results shown in figure 3
Here maximum abort conditions are also at max 2 but it has occurred only two times in total 50 rounds.

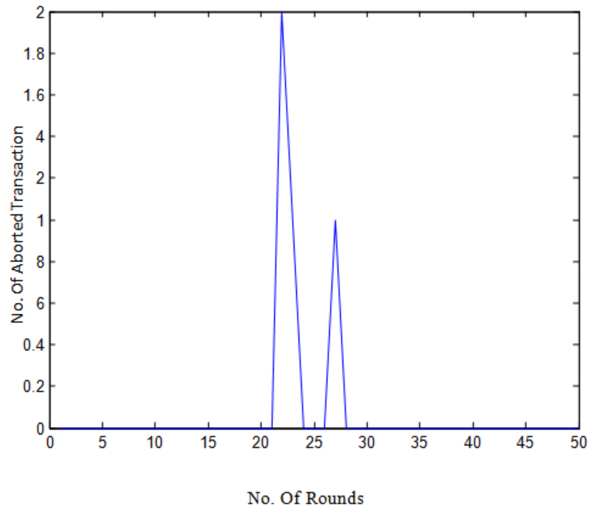

Fig. 4. Aborted Transaction at different rounds with PQCCM at 5 channels

5. Conclusion:

In this thesis we have developed three different algorithms for providing an environment of mobile distributed network system. For fulfilling the objectives of reducing packet drop out during the data transaction process. In our distributed system a transaction is performed by using collection of several step which are collectively called as Resource Management executed by Base Station to establish a connection and data transmission. In between different mobile node and the MTSO unit the transmission is assumed to be completed when any node request is either ends up with transaction commit or transaction dropout.
The program script is developed in MATLAB 2010 a. For the mobile distributed real time data base system We have considered that each node is mobile and during the changeover of cell by any arbitrary node Handoff mechanism is provided to transfer the data request and transaction control to the nearest base station for a particular cell. In first algorithm we have considered that we have only 4 channels and if more than 4 nodes are permitted for transaction then dropout occurs in the data of lower priority node. In this way it has been observed that there is 6 to 8 maximum dropout in each node. In another program we have considered that the nodes request which has not finished will be added in the higher priority in next priority queue list. By observing the result of this algorithm it has been observed that the maximum dropouts are decrease to 3 packets drop for different node. In the last part we have considered 5 channels priority queue concurrency check mechanism (PQCCM) algorithm the dropout become nil.
In this way we can conclude that our PQCCM based MDRTDBS protocol for 5 channels priority queuing running successfully and capable of eliminating the packets dropout in mobile network data transaction.

References

[1] Jensen, C. S. Research Challenges in Location-enabled M-services. In Proceedings of the 3rd International Conference on Mobile Data Management (MDM'02), 2002.
[2] Pitoura, Evaggelia, Samaras, George, "Data Management for Mobile Computing, Advances in Database Systems, Vol 10, 1998, pp-172.
[3] T. Imielinski and B. R. Badrinath, Wireless Mobile Computing: Challenges in Data Management, Communications of the ACM, 37(10), 1994, pp18-28.
[4] Smith Wayne Plourede: "Handling Concurrency Issues in .NET", http://www.15seconds.com/issue/030604.htm, Web retreieve on February 19, 2005).
[15] L. Sha, R. Rajkumar, S.H. Son and C.H. Chang. A Real-time Locking Protocol. IEEE Transactions on Computers, 40(7):793-800 (1991).
[16] kam-yiu lam et. al., "Concurrency Control In Mobile Distributed Real-Time Database Systems", Information Systems Vol. 25 No. 4, pp. 261–286, 2000.
[17] Jim Gray and Leslie Lamport, "Consensus on Transaction Commit" 1 January 2004 revised 19 April 2004, 8 September 2005.
[18] Salman Abdul Moiz et. al., "Commit Protocols in Mobile Environments: Design & Implementation" International Journal of Database Management Systems (IJDMS) Vol.2, No.3, August 2010.
[9] Y. Jayanta Singh et. al., "Dynamic management of transactions in distributed real-time processing system" 10.5121/ijdms.2010.2210.
[10] Salman Abdul Moiz, Dr. Lakshmi Rajamani, Supriya N. Pal, " Design and Implementation of Pessimistic Commit Protocols in Mobile Environments", The First International Workshop on Database Systems (DMS-2010), Springer Verlag Berlin Hiedelberg, CNSA 2010, CCIS 89, pp. 603- 612, 2010.
[11] J. Gray and A. Reuter. Transaction Processing: Concept and Techniques. Morgan Kaufmann (1993).
[12] Vishnu Swaroop, and Udai Shanker, "Data management in Mobile Distributed Real Time Database Systems: Reviews and Issues" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1517-1522.