Mobile Distributed Real-Time Database Systems Using Concurrency Approach

Dawodu Adekunle .A
D.S Adegbenro ICT Polytechnic
Itori-Ewekoro, Ogun State, Nigeria
E-mail: alandawodu@gmail.com
Phone: +2348055141696

ABSTRACT

With the increase in Mobile Computing Technology, applications and service creation are a motivated demand for real-time transactions in a mobile environment. There are two key words to note in this concurrency control way i.e. Mobile Distributed real-time database systems and High Priority Two Phase Locking. In a protocol environment, the characteristics of a mobile computing system are looked into a resolving back lock conflicts. There are several strategies that are proposed to improve the system performance and thus reduce the impact of mobile network on the performance of DHP-2PL. A transaction shipping approach is proposed to process transactions in a mobile domain by applying a well defined behavior of real time transactions. By applying the application of semantics in a real-time database application by adopting a similarity in concurrency of control in order to reduce the number of transactions restarts because of priority that is very expensive in a mobile network. A detailed model simulation of one of the two key words has been implemented and more so, a series of simulation experiments have been done to evaluate the performance of the intending approaches in a real time database application systems..

Keywords: Distributed Real-time Databases, Mobile Real-time Databases, Mobile Network, Concurrency Control, Transaction Scheduling, Data Similarity.

iSTEAMS Proceedings Reference Format
Dawodu, A.A. (2019): Mobile Distributed Real-Time Database Systems Using Concurrency Approach. Proceedings of the 18th iSTEAMS Multidisciplinary Cross-Border Conference, University of Ghana, Legon, Accra, Ghana. 28th – 30th July, 2019 Pp 187-196. www.isteams.net - DOI Affix - https://doi.org/10.22624/AIMS/iSTEAMS-2019/V18N1P22

1. BACKGROUND OF THE STUDY

Recently, many things occur in the wireless communication field in which mobile information services are fruitful. A large number of mobile computing systems that include tele-conference systems, real-time traffic information system, tele-medicine systems, mobile internet system and navigation systems are currently emerging as mobile users require instant access to information using their palm to PS, Personal Digital Assistant (PDA) and Note Book computers. The introduction of Mobile Computing does not only improve the distribution and flow of information but also at the same time increases the functionality of real-time database applications. The realization of instant information access over a mobile network relieves on real-time processing of transactions and it makes the timeliness of data accesses an important issue. However, as a matter of fact, research on processing soft real-time transactions in mobile distributed real-time database systems is receiving growing attention in recent years. Due to the limitations of mobile computing systems, such as limited bandwidth and frequent disconnection, the design of an efficient and cost effective mobile distributed real-time database systems (MDRTDBS) require techniques that are quite different from that in distributed real-time database systems (DRTDBS) which are supported with wired networks. It is extremely difficult to meet transaction deadliness in a mobile environment as there exist various factors, such as network performance, concurrency control and transaction scheduling, that can affect the transaction performance to a large extent.

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. However, the consequence of missing a transaction deadline in a tele-medicine system for ambulance system services may result in a loss of a human life.

1.1 Statement of Problem

Different issues are in the concurrency control of data access over a radio cellular network as though there exist several other mobile communication technologies such as satellite and wireless ATM. The MDRTDBS is a detailed performance study which was designed as a distributed real-time locking protocol in which the characteristics of an MDRTDBS are considered for resolving lock conflicts. A transaction shipping approach is proposed to reduce the impact of mobile network on the system performance and the performance of real-time locking protocol. In fact, different issues are in the implementation of transaction shipping approach which has been extended to a similarity-based protocol. However, a detailed simulation model is developed, in which the mechanics in mobile communications such as clients mobility and disconnection, and real-time transaction processing are involved. Simulation experiment would have been performed to investigate the performance of the proposed protocols and approaches, and the effectiveness of using similarity as the correctness criterion for concurrency control in MDRTBS.

1,2 Objective

The main objective of this study is to examine 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.

2. LITERATURE REVIEW

One of the most important issues to ensure timeliness of transaction execution i.e. 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 protocol method. However, 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 which is blocked by a lower priority transaction. In the past years, researchers have proposed various real-time concurrency control protocols for a single site as well as distributed RTDBS. The idea of priority inheritance was introduced, which allows a lower priority transaction inherit the priority of a higher priority transaction which is blocked by the lower priority transaction, to reduce the number of priority inversions of the higher priority transaction. The priority inversion problem may also be resolved by transaction restart. Also, the priority inversion problem, in general, can also be resolved by transaction restart in the optimistic concurrency control protocols.

In the optimistic currency control protocol, the execution of a transaction is divided into three phases; the read phase, the execution phase and the validation phase. Note that, Data conflicts among other different transactions will be resolved when one of them enters the validation phase. If the number of conflicting transactions, which priorities are higher than the validating transaction, and is not greater than 50% of the total number of conflicting transactions, the conflicting transactions will be restarted and the validating transaction is allowed to enter the write phase and then commit. Otherwise, the validating transaction will be blocked. In concurrency control for MDRTDBS, transactions are considered, which are simple flat transactions with read and write operations. This assumption on transaction structure is reasonable for many mobile soft (and firm) real-time applications with web-oriented interfaces, especially for applications running on mobile palmtop computers such as stock monitor systems and traffic information systems. The simplification of truncation behavior and characteristics may result in good strategies in real time transaction scheduling algorithms to improve the system performance. The two new strategies are transaction shipping approach, to reduce the performance dependency of a currency control protocol on the performance of underlying mobile network and also the notion of similarity to resolve data conflicts, which can be very costly in a mobile environment.

3. SYSTEM ARCHITECTURE MODELLING IN A MOBILE DISTRIBUTED REAL-TIME SYSTEM

A MDRTDBS consists of four (4) major components; the mobile clients (MCs), the base stations, the mobile network, and the main terminal switching office (MTSO) as shown in fig. 1 below. The mobile network is assumed to be a radio cellular network and the entire service area is divided into number of connected cell sites. Within each cell site, there is a base station, which is augmented with a wireless interface to communicate with the MCs within its cell site. The cellular radio network is assigned to be the Global systems for Mobile communication (GSM) in which two sub-bands each are defined. One of the sub-band is for uplink (for the mobile clients to transmit signals to the base station), another sub-band is for the downlink. Within each sub-band, a number of channels are defined for transmitting radio signals which can be data or control signals. Each channel is divided into several time frames by using the time division multiple access (TDMA) method. Usually, the data transmission rate of a channel is between 9.6 to 14.4 Kbps. The base stations at different cell sites are connected to the MTSO by a point-to-point wired network. Thus, the communications between the base stations and the MTSO are much more efficient and reliable than the communications between the base stations and the mobile clients.

The MTSO is responsible for active call information maintenance, performance of the handoff procedure, channel allocation and message routing. Attached to each base station is a real-time database system containing a local database which may be accessed by transactions issued by the MCs within the cell site or from the other cell sites through the MTSO. The mobile clients can move around within the same cell site, it can even cross the cell border into another cell. It sends a location signal to its base station through an uplink channel. The strength of signal received by a base station depends on several factors, such as the distance between the MC and the base station, and the surrounding buildings. When MC is crossing the cell border, the strength of signal received by a base station will be very weak. If the strength of signal is lower than a certain threshold level, the MTSO will be notified, and then the MTSO will perform a handoff procedure. It sends out requests to all the base stations, and the base stations respond by returning the strength of the location signals received from MC. The MTSO will then assign the MC to the base station which has received the strongest signal.

Due to noise and interference, thee signal which carries data, may be corrupted while it is being transmitted. In this situation, the data will be re-transmitted. More so, if the transmission of signals is corrupted consistently after several times, a disconnection may have occurred. In this case, the transaction may need to wait until a new channel is granted before it can proceed. However, because of high error rate and non stability of signal transmission, the effective data transmission rate is unpredictable.

3.1 The Database System and Transaction Modelling

The whole database is partitioned into local databases and distributed at different base stations. The database consists of two types of data objects in the external environment. Temporal data objects are used to record the status of the objects in the external environment. Each temporal data object is associated with a timestamp, which denotes the age of the data object. If a transaction may update a temporal data object, then the transaction is given a timestamp when it is initiated. If the transaction commits successfully before its deadline, the timestamp of any data object, which is updated by the transaction, will be set as the timestamp of the transaction. The validity of a temporal data object is defined by an absolute validity interval (avi). A temporal data object satisfies the avi constraints if its age is up to date i.e. the difference of the current time and the age is no more than the avi. A relative validity interval (rvi) may be given to a transaction which requires that the maximum age difference of the data objects read by the transaction is not larger than rvi. Non-temporal data objects are either derived by operations of transactions or are statistically set during system initialization. The transactions from the MCs are assumed to be simple flat transactions with a collection of read and write operations. More so, control statements may be defined to control the logic flow of transaction, in between the operations of a transaction.

3.2 The Protocol Used In MDRTDBS For Controlling Concurrency

While designing the concurrency control protocols for MDRTDBS, two important things are to be considered;

i. How to minimize the cost and overheads for resolving data conflicts.
ii. How to minimize the impact of mobile network on the performance of the protocol.

Restarting a transaction is highly expensive in a mobile environment, although priority inheritance is effective in managing the priority inversion problem in single site RTDBS. It may not be effective in MDRTDBS due to the slow network. In fact, it may take a long time before the priority of a transaction is inherited. Also, deadlock is possible when priority inheritance is used. Deadlock is highly undesirable to real-time systems especially in a distributed environment. It is not only because it greatly increases the response time of transactions, it also wastes a lot of system resources. The detection and resolution of a deadlock in a distributed environment may also consume a lot of resources. The most common method for distributed deadlock resolution is time-out, which is obviously not suitable to RTDBS due to difficulty in determining the appropriate timeout period.

The validation test required in the optimistic concurrency control protocols can be very complex in a distributed environment and it will be more complicated in a mobile network. The system consists of both a mobile network (connecting the mobile clients and the base stations) and a reliable wired network (connecting the base stations and MTSO). Since mobile network is vulnerable to disconnection, strategies for solving data conflict should consider the network quality connecting the conflicting transactions. A different strategy should be used if a mobile network instead of a reliable wired network connects the conflicting transactions especially when it is suspected that disconnection may have occurred.

DHP-2PL is a distributed locking Protocol. The local database system at each base station has a lock scheduler, which manages the lock requests for the data objects residing at the base station. The definition of the DHP-2PL is thus; where Tr and Tn are the lock-requesting transaction and lock-holding transaction respectively;

```
if      (the priority of the lock-requester > the priority of the lock-holder) and
        (the lock-holder is not committing)
        Restart the lock-holder (globally or locally, depending on the type of transaction)
Else
        if      location indicator of the lock-holder is "mobile client"
                if      the time already spent at the client side > threshold
                Ping the mobile client where the lock-holder is residing
                //*the base station sends a message to the mobile client to test whether the mobile
                client is disconnected or not*//
                if      no response from the mobile client
                        Restart lock-holder
                Else
                        Block the lock-requester
                //*repeat the checking after another threshold*//
                Endif
        Else
                Block the lock-requester
                //*the checking will be performed again when the time already spent at the client
                side is greater than the threshold value*//
        Endif
        Else
                Lock the lock-requester
        Endif
Endif
```

When a transaction is at the mobile client or it is waiting to move back to the mobile client, the location indicator of its processes will be set as mobile client. Otherwise, it is set as base station. When the location indicator of a transaction process is mobile client, the transaction is vulnerable to disconnection. Below are the cautions waiting scheme incorporated into DHP-2PL;

```
if      (the priority of the lock-requester > the priority of the lock-holder) and
        (the lock-holder is not committing)
        Restart the lock-holder (globally or locally, depending on the type of transaction)
Else
        if      location indicator of the lock-holder is “mobile client”
                if      the time already spent at the client side > threshold
                Ping the mobile client where the lock-holder is residing
                //*the base station sends a message to the mobile client to test whether the mobile
                client is disconnected or not*//
                if      no response from the mobile client
                        Restart lock-holder
                Else
                        Block the lock-requester
                //*repeat the checking after another threshold*//
                Endif
        Else
                Block the lock-requester
                //*the checking will be performed again when the time already spent at the client
                side is greater than the threshold value*//
        Endif
        Else
                Lock the lock-requester
        Endif
Endif
```

The threshold is a tuning parameter. It is a function of the average performance of the mobile network under normal situation.

4. METHODOLOGY

The method used in this paper involves literature review and the urgency to meet the transactions and how to satisfy the temporary constraints of the database and the mobile distributed real-time systems on this subject matter. A new approach called transaction shipping is proposed and the goal of this approach is to reduce the impact of mobile network on the performance of DHP-2PL and improve the system performance. A transaction hipping approach is a better transaction processing strategy of MDRTDBS which was designed to reduce the communication overheads between the mobile clients and the base stations for a transaction, and to alleviate the dependency of system performance on the performance of the underlying network. The main ideology here is to "ship" the entire transaction to the database server (base station) for processing instead of shipping all operations or data request to the database server. However, there exists many practical problems when it is applied to a MDRTDBS in order to identify the execution path and the required data objects of a transaction before its execution.

4.1 The Prediction in Transaction Ability

It is generally agreed that the functions and transactions behaviours in a real-time database application are more predictable. In most cases, they can be classified into different types of transaction, which have different pre-defined behavior. For example, in a medical information system, real-time transactions are for monitoring the physical status of a patient from various sensor devices such as the heart beat rate, body temperature and the blood pressure. The arrival pattern and data requirements of the transactions are pre-defined, that is, in some real-time database applications e.g. Programmed Stock trading, the transaction arrival pattern may be sporadic, their data requirements can be predicted with a high accuracy.

The diagram below shows the process Architecture under transaction shipping approach;

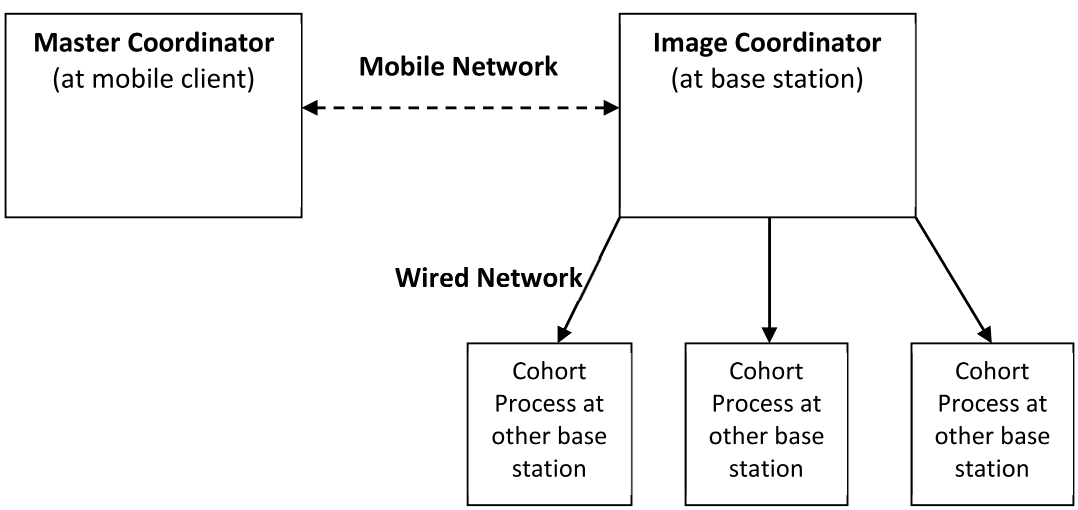

Fig. 1: Process Architecture under Transaction Shipping Approach

4.2 Pre-Analysis Phase

In transaction shipping approach, the execution of a transaction is divided into two phases; the pre-analysis phase and the execution phase. Figure 1 above shows the transaction architecture when it is processed under the transaction shipping approach. Once a transaction is initiated at a mobile client, a coordinator process, called Master Coordinator, will be created at the mobile client. We use the pre-analysis to predefine the execution path of a transaction in order to reduce the number of communications between the mobile clients and the base stations for a transaction. The pre-analysis of a transaction consists of two phases. In the first phase, the set of operations in the transaction will be identified and secondly, the execution path of the transaction, e.g. the precedence relationships of the operations will be determined. After the completion of the pre-analysis stage, a signature of the transaction will be created. A signature transaction, **S**i, for transaction, **T**i, consists of 4 turples;

Si = (**O**i, **D**i, **C**i, **<**i)

Where; **D**i is the deadlines of **T**i
Ci is the criticality of **T**i
Oi is a subset of the operations of **T**i
<I is the partial order relationship among operations in **O**i

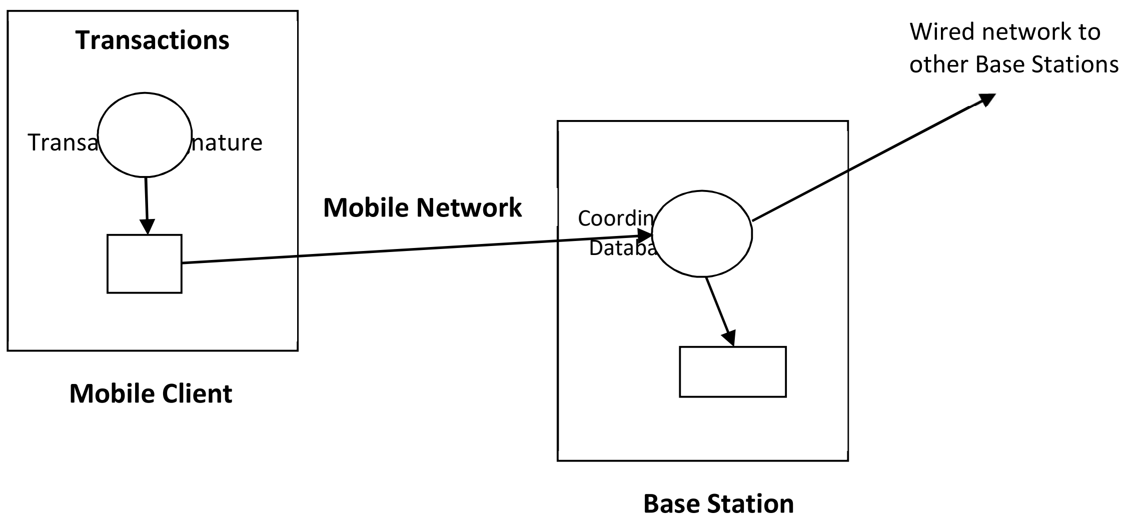

Fig. 2: The Transaction Shipping Approach

In the execution phase, the signature transaction is forwarded to the base station of the MC through the mobile network. Once the server at the base station receives the transaction signature, it will create a process called image coordinator for the transaction. The image coordinator will take over the job from the master coordinator to process the transaction. The benefit of defining an image coordinator at the base station is that the connection between the MCs and its base stations. Thus, it facilitates the management of the transactions and improves the performance of the atomic commitment protocol. The image coordinator is responsible for restarting the transaction from its beginning if its deadline has not been missed.

5. DISCUSSION

The findings from this research shows that the various tools used are not in any way adhering strictly to the common vulnerabilities and exposures (CVE) standard in assessing the problems in mobile distributed real-time Database systems. Although many researchers have done a lot of excellent work in concurrency control for a single-site and distributed RTDBS, there is a little work in concurrency control in MDRTDBS which is a fast growing and important areas in the mobile network which imposes a serious time burden on the performance of MDRTDBS, it can also affect the performance of the adopted concurrency control protocol. In fact, the benefit of a similarity based concurrency control protocol is the increase in system concurrency. Also, if the similarity bounds of the data objects are well chosen, the benefit obtained from an increased in concurrency control should be more than the overhead for the similarity checking. Note: an increase in concurrency may not necessarily result in better performance.

6. CONCLUSION

The design of mobile distributed real-time database systems (MDRTDBS) is receiving growing attention in recent years. Due to poor quality of services provided by mobile network, it is not easy to meet the deadlines of the transactions in a MDRTDBS. A detailed model for MDRTDBS in which the mobility of the mobile clients and characteristics of mobile network e.g. disconnection and low bandwidth. A distributed real-time locking protocol was designed called Distributed High Priority Two Phase Locking (DHP-2PL), where the characteristics of the mobile network are considered in resolving the conflicts in data accesses. However, certain strategies were introduced to improve the system performance and to reduce the impact of mobile network on the performance of concurrency control protocol. The concept of transaction shipping approach was proposed to reduce the dependency of a concurrency control protocol on the performance of the underlying network, also the communication overheads for processing a transaction can be much reduced using transaction shipping approach. However, similarity based locking protocol was introduced to resolve conflicts among data access that can be very costly over a mobile network.

7. FUTURE WORK

Pursuant to the findings from this research work, it is recommended that a relatively thorough assessment should be done on comparing the open source and propriety tools for efficiency, effectiveness and reliability in determining the use of similarity based algorithm for further research work and also to determine if there is any specific capabilities or limitations with open or closed source tools on mobile distributed real-time database.

REFERENCE

1. S. Gindraux, form 2G to 3G: A guide to mobile security, proceedings of third International conference on 3G Mobile Communications Technologies 2002.
2. Yang Kum, Guoxin, Liu Dayou, Security in mobile Agent system: Problems and approaches, ACM SIGOPS Operating Systems Review, 2000.
3. Asoke K Talukder, debabrata Das, Artificial Hygiene: Proliferation of virus in Cellular Network, Journal of System and Information Technology, Volumes, 1st December 2004, pp 10-22.
4. Wireless Transport Layer Security, Version 06-Apr-2001: WAP-261-WTLS-20010406-a; http://www.wapforum.com
5. Data Encryption standard (DES); Federal Information Processing standard Publication, 1999 October 25, U.S Department of Commerce/National Institute of Standards and Technology.
6. Williams Stallings, Network Security Essentials: Applications and Standards; Pearson Education, 2000.
7. Wireless Application Protocol Identity Module Specification, Version 05-Nov-1999; WAP forum
8. Henry .M Levy, Capability-Based Computer Systems, Digital Press, 1984.
9. Hussian Sandhya Sharma, and Vinita Sharma(2013),"Automated Intelligent Traffic Control Systems using Sensors". International Journal of soft computing and Engineering (IJSCE), 3(3), pp 2231-2307.
10. Lin Dong and Wushan Chen (2010),"Real-Time Traffic Signal Timing for Urban Road Multi-Intersection", Journal of Intelligent Informaation Management; 2(8), pp. 483-487.
11. Rostami, M., Kousharfar F, Rajndran, J, & Karri, R (2013). Hardware Security: Threat models and metrics, In K. Embler (Ed.), 2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (PP. 819-823), Hilton San Jose 300 Almaden Bvd. San Jose, CA, USA: IEEE Council on Electronic Design Automation