Replication Strategies in Mobile Environments

Salman Abdul Moiz[1] and **Lakshmi Rajamani**[2]

Abstract - Transaction management in wireless environment poses challenging issues in preserving data consistency and fault tolerance. As the mobile databases are prone to frequent disconnections, bandwidth limitations, mobility, etc. efficient execution of the transaction may not always be guaranteed. Replicating data at several sites is a powerful mechanism which not only increases performance but also provides fault tolerance for demanding database applications. However the major concern is to keep the replicated copies always consistent. In this paper various replication techniques and their applicability in the mobile environments are presented. Based on the requirements of the services provider and the type of execution model used, one of the replication strategies may be implemented.

Index Terms - Mobile Host (MH), Fixed Host (FH), Transaction, Replication, Synchronous, Asynchronous.

1. INTRODUCTION

Mobility gains more and more importance from a technological as well as social perspective. Since network bandwidth is an expensive resource in mobile environments, the transaction processing should reflect a much concern for bandwidth consumption and constraints than non-mobile environments [2].

The transaction requests is initiated at the Mobile Host (MH) but may be partially or completely executed at the Fixed Host (FH). In the traditional environment once the transaction request is initiated by a mobile host, the respective data item is locked at Fixed host. In case of failures or disconnections the system almost halts as there is no replica to proceed with the transaction processing.

Over the years, data replication is considered as a better solution to increase throughput(more replicas can serve more request), decrease response times (distribute the load and access the local replica) and provide fault tolerance[13]. The major challenge with the implementation of replication strategies is to keep the replicas consistent. In addition the mobility introduces several other challenges for the management & maintenance of replicas in a mobile distributed environment.

Several valuable attempts have been presented for efficient implementation of concurrency control and fault tolerance in mobile environments. However each attempt considers only a subset of the operational requirements.

[1]*Research Scientist, Centre for Development of Advanced Computing, Bangalore.*
[2]*Professor, CSE, University College of Engineering, Osmania University, Hyderabad*
E-Mail: [1]*salmanmca@gmail.com and*
[2]*lakshmiraja@yahoo.com*

For example in [9], the author proposes a scheme, to provide non-blocking protocol with restrained communication. It faces the problem of time lag between the local and global commit. In [7] concept of non-conflicting transactions is introduced such that if a conflicting transaction is detected, it may be aborted. The research in [8], proposes Mobile-2PC, which preserves the traditional 2 Phase Commit protocol while minimizing the impact of unreliable wireless communication. Fragmentation of relations is used to support semantic based concurrency control [4]. In [5] distributed lock management scheme is proposed. Most of these techniques presented produces low throughput due to inefficient implementation of replication strategies. However in [3] authors propose a directory structure to identify replicas rather than migrating the replicas. However the frequent mobility needs the directory structure to be updated. In this paper we propose various replication strategies which may be adopted my application providers based on the fidelity & importance of data.

The remaining part of this paper is organized as follows. Section 2 describes the general architectural view of mobile databases. Section 3 specifies the various replication strategies their advantages and challenges. Section 4 specifies mechanism to uniquely identify a record (replica) in the mobile distributed environment. Section 5 describes the proposed replication strategies of mobile environments and the proposed solutions for the conflict resolution. Section 6 concludes the paper.

2. MOBILE DATABASE ENVIRONMENT

The mobile computing environment generally consists of three entities Fixed Host (FH), Mobile Units (MU) and Base Stations (BS) respectively. Terminals, desktop, servers are the Fixed Host, which is interconnected by means of a fixed network.

Large databases can run on servers that guarantee efficient processing and reliable storage of database. Fixed hosts perform the transaction and data management functions with the help of data base servers (DBS). Mobile units are the portable computers which can retain the network connections through the support of the Base Stations (BS).

Mobile clients may vary from the thin to full clients based on their characteristics.

1. Thin client architecture: In this organization, the resources of the mobile clients are limited. The mobile client takes the request from the user and the execution of the transaction is done at the fixed host.
2. Full client architecture: In this architecture, clients can work in the disconnected mode. Full clients own the responsibility of server functions. They are portable and have enough resources for execution of an application.

In addition to the thin or full client architecture, transactions initiated by the mobile clients may also work on the Flexible client server architecture or the Client agent server architecture. In the flexible client server architecture the roles of client and

servers can be dynamically relocated. In order to increase the throughput of the system, the distinction between clients and servers may be temporarily blurred.

In the client-agent-server architecture, a three tier model introduces an agent similar to a proxy located on the fixed network [1]. In this paper we deal with the crash recovery mechanisms for the thin client and the full client architectures respectively.

3. REPLICATION STRATEGIES

Data Replication is the process that allows building a distributed environment through the management of multiple copies of data, caching one copy on each site. Replication however has the challenges of the replication control. Changes submitted to one replica have to be applied at the other replicas such that the different copies of the database remain consistent despite concurrent updates.

In general, there are two types of replication strategies: Synchronous and Asynchronous replication. In *Synchronous replication* (real time data replication [6]), performs updates on all replicas at the same time. In *Asynchronous replication* (store and forward replication [6]), operations performed on one site is stored locally and later on it is updated to other replicas.

Synchronous replication technology ensures highest level of data integrity but requires permanent availability of participating sites and transmission bandwidth. If one of the participating site holding a replica is not ready the transaction may not proceed. This scheme needs more resources but the replicas will be consistent at any instance of time. Asynchronous replication provides more flexibility than synchronous replication as the database synchronization time interval can be defined which can vary between the applications and from one service provider to another. Moreover a single site could work even if a remote server is not reachable or down.

As disconnections in mobile environments are treated as normal conditions rather than failures, asynchronous replication is better suited. If the mobile host holding a replica is disconnected for a longer time then in synchronous replication the transaction can't proceed unless the mobile host is connected. However techniques described in [11] may help in synchronous replication but it takes more time to keep the consistent copies of the data. Comparatively though asynchronous replication is better suited, efficient concurrency control and fault tolerance mechanisms are to be deployed. A Concurrency control algorithm proposed [10] is better suited for the implementation of concurrency control in mobile environment when the sites are updated based on asynchronous replication policy.

Based on the mobility, there are two approaches to the replication of data in mobile environments. In the first approach the data may be replicated at both the mobile host and the fixed host. Whenever the updates are preformed at mobile host (mobile client) the same has to be consistent with the data on fixed host (mobile server). If Synchronous replication is adopted both the mobile host and fixed host must be in connected mode at least when the updates on replicas are performed. The concurrency control can be effectively implemented using the dynamic timer management mechanism discussed in [10]. Since mobile hosts are prone to disconnections asynchronous replication can be adopted. The replication control is done at the client side. When the mobile host is in disconnected mode and a client replica is updated. These updations has to be performed on the fixed host within the specified time interval (this time interval differs from application to application).

The second approach to the replication deals with mobility. In this mechanism the MSS will maintain the replicas i.e it maintains the list of sites were replicas are present in home location as well it forwards one of the replica to remote location. If any update is performed by the replica in foreign location, the updates are conveyed to MSS which updates the replicas present in that cell.

4. IDENTIFICATION OF REPLICAS

In replicated environment, any database instance my contain the local data or it could be instance of another replica. In this section we followed the approach of Cavelleri et al[6] to distinguish the local site information and the replicated information of other sites.

Any table at any site either mobile host or fixed host can be represented in the following form:

$$T_{a,i} = P_{a,i} \; u \; U_{j=1,...N,\, J \neq i} \; R_{a,j,i}$$

where

a identifies a generic table of the database

i identifies the site (Mobile host or/and fixed host)

$T_{a,i}$ is the entire content of table a on site i

$P_{a,i}$ is the information entered in table a on site i.N is the number of participating sites i.e no of sites where the data is replicated.

$R_{a,j,i}$ is the replicated partitions of table a coming from site j cached into site i

Each site (Mobile host and Fixed host) is uniquely identified by site-id. $P_{a,i}$ and $R_{a,j,i}$ are partitions of table $T_{a,i}$. When an insertion occurs in table a on local site i the information is stored in local partition $P_{a,i}$ and later forwarded to every replicated partition $R_{a,j,i}$ of each remote site j.

Local and replicated sites could be the mobile host and fixed host or it could be one cell to another cell. To identify a record as to whether it is a local or remote partition site record ids are used i.e record id along with the site id will identify a particular record. This mechanism of record identification helps in knowing the origin of a record thereby reducing the overhead of directory structure to be maintained by MSS as discussed in[3].

Consider the following Table stored at site 1, which has local partitions as well as remote partitions. It has three fields A,B,C and the first field specifies the record id (site id + record id). The remote partitions for the purpose of updating of data can be identified using this identifier

Record_Id	A	B	C
(1,1)	10	20	30
(2,1)	23	34	56
(1,3)	33	12	21

Table 1: Schema representation using composite record id

The first two records belong the the site 1 and their record ids are 1, 2 respectively. The third record belongs to site 3 whose record id is 1. If an updation is performed for the third record on data item C. Apart from updating new value of C. The first record and partition 3 will also be updated. This is identified by composite record id.

5. MOBILE DATA OWNERSHIP MODELS

Data ownership models specifies various mechanisms that can be adopted to keep the replicas consistent. These mechanisms may be adopted based on different applications implemented in mobile environments. There are different types of Data ownership models viz., Workload partitioning data ownership model, Master/Slave data ownership model and Update any-where datownership model. The applicability of these models in mobile environments is discussed below:

5.1 Workload Partitioning data ownership Model

This model is implemented by read only access to $R_{a,j,i}$ and read write access to $P_{a,i}$ .The replicated partition of table from site j to will only have read access at site i. In this scheme all replicas can only read the information coming from remote partition. It can preform the write operation if the data item belongs to the local site.

If the mobile host needs to update the data item stored at some fixed host, the write operation has to be performed at fixed host. Even though a replica need to update the data item but since it has read only access the write operation has to be done at only local site for that data item. This mechanism preserves the integrity of the data as the updations to the data item is done at only one site.

If the mobile host moves from one cell-1 to cell-2 and if a write operation has to be performed on a data item then the updates are performed only in cell-1 as the fixed host of that mobile host is in cell-1.

5.2 Master/Slave Data Ownership Model

Master/Slave replication is an asynchronous replication where data is owned by one site (owner) and can be updated by only that site. It follows publisher-subscriber pattern. The other sites who own only read only data are slaves (subscribers).

In this approach whenever a data item has to be updated, the owner is identified using the composite record id scheme (record id + site id) so that the request for the update is given to the respective site and then the site updates the data items. However the site i where updations to the database item is performed has to propagate these updates to all replicas i.e all the slaves need to know the final value of the data item. This is similar to the Thin client architecture of the mobile environment where the mobile host requesting for a service send the request to the fixed host.

When a mobile host moves from once cell to another. The request for write operation for a particular data item may have to sent back to the home cell if the subscriber is in home cell. Concurrency control and failure tolerance is easily implemented as the udpations are done by only one site[10].

5.3 Update Anywhere Data Ownership Model.

In Update Anywhere Data Ownership Model, read write access is available to $R_{a,j,i}$ and $P_{a,i}$. This leads to replication conflicts because any mobile host or fixed host where the replica is present, a write operation on the data item can be performed. The database can be kept consistent by implementing efficient concurrency control techniques for mobile environments [5,10]. However these techniques differ from one replication strategy to another.

In Synchronous replication log based scheme[12] is implemented as the replicas need to be consistent all the time. Whereas in synchronous replication triggers are used to update the replicas after firing of events. The mobility issue can be managed by transfer of log information from one cell to another[10].

6. CONCLUSION

Replicating data at several sites is a powerful mechanism to increase the performance, throughput and can provide fault tolerance. However unlike synchronous replication, asynchronous replication may not keep the database consistent at every moment. If this time lag can be compromised it may require less resources. Further the problem of storage of log on the mobile will not be an issue. Based on the volume of replication and the number of transactions respective ownership models may be chosen. However update anywhere ownership model can make the replication strategies in increasing the throughput but has to be implemented as synchronous replication as any site can update data at any time. When the replication increases, performance will be an issue. Methodologies for the management of huge replicas are needed.

REFERENCES

[1]. Can Turker, Gabriele Zini, "A Survey of Academic & Commercial Approaches to Transaction Support in Mobile Computing Environments", DELOS, Network of Excellence on Digital Libraries, Project No.507618-A, PP. 8-35, 2003.

[2]. Evaggelia Pitoura, Bharat Bhargava, "Revising Transaction Concepts for Mobile Computing", wmcsa, Pp.164-168, 1994 First Workshop on Mobile Computing Systems and Applications, 1994.

[3]. Daniel Barbara Milla, Hector Garcia Molina, "Replicated Data Management in Mobile Environments: Anything New under the Sun?", IFIP Transactions; Vol. A-44 Proceedings of the IFIP WG10.3 Working Conference on

Applications in Parallel and Distributed Computing, PP. 237 – 246, 1994.
[4]. Gary D Walborn, Panos K Chrysanthis, "Management of Mobile Transactions", Proceedings of the 1999 ACM Symposium on Applied computing, PP. 389 – 398, 1999.
[5]. Jin Jing, Omran Bukhres, Ahmed Elmagarmid, "Distributed Lock Management for Mobile Transactions", Proceedings of 15th International conference on Distributed Computing Systems, PP. 118-125, 1995.
[6]. M. Cavalleri, R. Prudentino, U. Pozzoli, G. Veni, "A set of tools for building PostgreSQL distributed database in biomedical environment, Proceedings of the 22nd Annual International conference on "Engineering in Medicine and Biology society", PP. 540-544, 2000.
[7]. Minsoo Lee, Sumi Helal, "High Commit Mobile Transactions", Distributed & Parallel Databases, Vol. 11, Issue: 1, PP.73-92, 2002.
[8]. Nadia Nouali, Anna Doucet, Habiba Drias, "A Two Phase Commit Protocol for Mobile Wireless Environment" Proceedings of 16th Australasian Database Conference.Vol.39 (ADC 2005), PP. 135-143, 2005.
[9]. Patrica Serran-Alvarado, Claudia Roncancio, Michel Adiba, Cyril Labbe, "A Survey of Mobile Transactions" Distributed and ParallelDatabases, 16, 193-230,2004.
[10]. Salman Abdul Moiz, Dr. Lakshmi Rajamani, "Single Lock Manager Approach for acheiving Concurrency Control in Mobile Environments", Proceedings of 13 IEEE International Conference on High Performance Computing, 2007.Springer LNCS 4873, PP.650-660, 2007.
[11]. Salman Abdul Moiz, Dr. Lakshmi Rajamani, "Disconnected Modes of Operations in Mobile Environments", Proceedings of INDIACom-2008, 2nd National Conference on Computing for Nation Development, PP.253-256, 2008.
[12]. Salman Abdul Moiz, Dr. Lakshmi Rajamani, "Log Based Recovery in Mobile Environments", Proceedings of 1st International conference on Advance Computing, ICAC-08 (ACM), ISBN: 978-81-906457-1-3, PP. 530-533.
[13]. Shuqing Wu, Bettina Kemme, "Combining Replica Control with Concurrency Control based on Snapshot Isolation", Proceedings of 21st International Conference on Data Engineering (ICDE'05), PP. 422-433, 2005

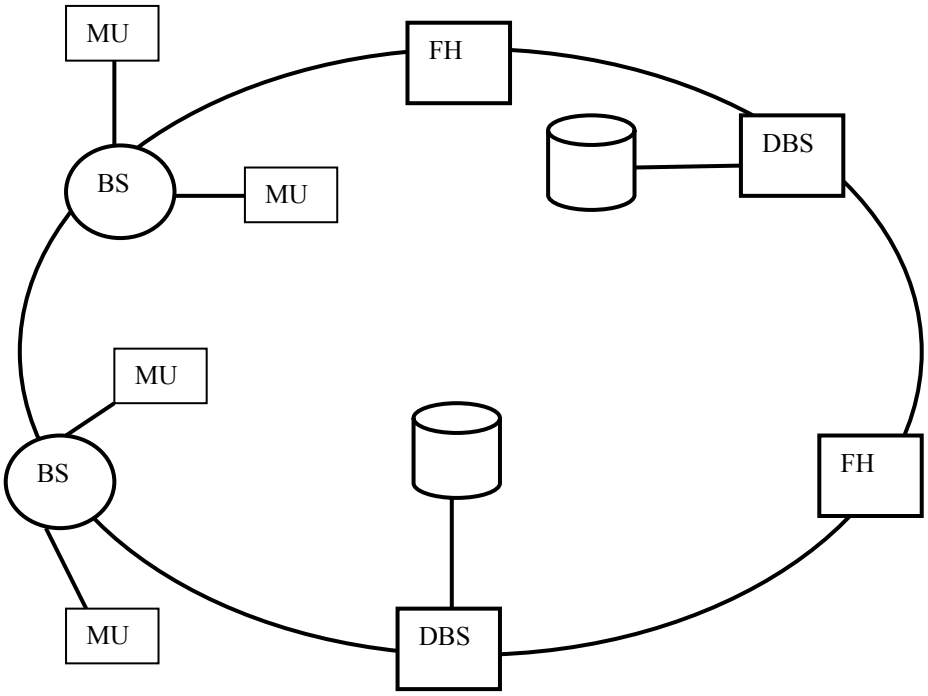

Figure 1: Architectural View of Mobile Environment

Continued on Page No. 169