CHAPTER SEVEN

CONCLUSIONS AND DISCUSSION

7.0 Overview

This thesis presented a data replication strategy for improving the availability and maintaining the consistency of replicated data in large scale mobile distributed database systems with large number of updates. The strategy is proposed by considering the characteristics of such systems, especially frequent disconnection of mobile hosts. This strategy consists of four components in order to better meet the requirements of data replication in these systems and act with accordance to their characteristics. This chapter summarizes the main conclusions, focusing on contributions, limitations and future work.

7.1 Contribution of Research

This research handles data replication issues in an emerging area (i.e. large scale mobile distributed database systems) in which fewer researches have been done. And compared to these researches, this research provides a comprehensive data replication strategy in terms of its services and outcomes. The strategy provides four services that are required for such replication systems, which are: replication architecture, replicas allocation, updates propagation, and updates ordering. Moreover, the research provides additional service for implementing those services, which is IIRA. The outcomes that are provided by this strategy are: availability of recent updates and consistency. In contrast, other strategies provide less services and outcomes. Moreover, our research provides new ideas for performing aforementioned services. Some of the specific notable contributions of this thesis are mentioned in the following subsections.

7.1.1 Four-Component Replication Strategy

A new replication strategy that is suitable for replicating data in LMDDBSs is proposed by considering a logical replication architecture, Wheel-based updates propagation protocol, updates ordering mechanism and IIRA-based propagation system. In this strategy, three-level replication architecture is provided, which specifies the organization and the distribution of the replicas in LMDDBSs in a manner that supports large number of replicas in order to better meet the requirements of LMDDBSs. The strategy supports frequent disconnections and the mobility of mobile hosts by enabling the users to perform their updates in a disconnected mode and then synchronizing their updates with the higher levels. The strategy provides mobile users with the ability of choosing what objects they want to be replicated on their databases for carrying out their duties in the initial replication process. Also, it enables them to change existing replicated objects according to their duties. The strategy combines both optimistic and pessimistic replication approaches in a hybrid manner that exploits the pertinent features of each in LMDDBSs. These features are supporting higher availability of recent updates and lower rate of inconsistencies as well as supporting the mobility of users.

7.1.2 Scalable Updates Propagation Protocol

A scalable Wheel-based updates propagation protocol is proposed for exchanging recent updates that occurred in large number of replicas in a manner that achieves the following:

- Load balance between the components of the replication system.
- Reduction of the updates propagation delay in a manner that bounds inconsistency between these components and unavailability of last updates.
- Lesser total number of messages that are required to propagate the information of recent updates.

The effectiveness of the proposed strategy with respect to updates propagation is verified through the comparative study with Roam replication system. The results revealed that the proposed strategy achieves better propagation delay and lesser total number of messages than Roam replication system. Both better propagation delay and lesser number of messages represent essential features that characterize scalable replication strategies with regard to updates propagation to all replicas. Moreover, the proposed propagation protocol achieves load balance in both propagation and ordering processes because these processes are shared by multiple hosts.

Also, the proposed updates propagation protocol is compared with N-ary tree based propagation protocol. The results showed that the horizontal extension provided by the proposed protocol is more suitable than the vertical extension for LMDDBSs, since such type of systems is extended to cover new geographic areas. Moreover, placing new hosts of those areas in a tree structure with a variable number of levels as in *N*-ary tree based propagation protocol (and also HARP protocol) is not reasonable due to the large values of updates propagation delay that are resulted by sending updates to the new hosts, especially when they are placed in either the last level of the tree or a new established level. Furthermore, this mode of extension (i.e. vertical extension) does not act in accordance with the characteristics of LMDDBSs that involved a combination of both fixed hosts and mobile hosts, which are distributed over large geographic areas. Accordingly, we argue that such type of systems requires a horizontal extension to encompass the new hosts according to their areas and types.

To bound inconsistency and unavailability of recent updates, the proposed strategy ensures fast propagation of updates (i.e. as compared to both Roam and *N*-ary strategies) and enabling all mobile hosts to propagate their data once they are connected with the fixed network. Similarly, the strategy enables all cell servers to provide the connected mobile hosts with recent updates that occurred in the other hosts. Accordingly, in the mobile network, the inconsistency is bounded by the disconnection period of mobile hosts. Also, the master server is enabled to propagate recent updates directly once they are ordered to underlying levels. Accordingly, the bound here is restricted by the completion of the updates collection and ordering period at the master server. Similar bound exists when cell servers and zone servers propagate their collected updates to the higher level, which here is restricted by the completion of the collection period. The enabling of a host to immediately propagate its updates (i.e. when the connection occurs or completion of collection period) is realized via the replication method, which is based on IIRA.

7.1.3 Updates Ordering Mechanism

To provide the level of consistency with ordering guarantee in LMDDBSs, updates ordering mechanism is proposed that ensures a unified causal and total ordering for all updates that have been occurred in all replicas in the replication system.

As compared with the ordering mechanisms that are based on version vectors, which are mentioned in the literature review (see section 2.7.2), we realized that the proposed ordering mechanism, which used Real-Like clock with periodical time checking and amendment is more suitable to support the characteristics of

LMDDBSs, especially scalability, disconnections, mobility, and limited storage and communication resources. The reason is that the implementation of version vectors based mechanisms in LMDDBSs imposes low storage, computation, and communication overheads. These overheads originated from the size of version vector and large amount of information that each message should carry in order to implement causal ordering. On the other hand, the proposed ordering mechanism imposes a small size of Real-Like clock and sending only the information that is needed to determine the causal order for updates to the higher level. This results in low communication and storage overheads in the higher level.

Moreover, the proposed mechanism acts in accord with the characteristics of LMDDBSs through providing a unified hierarchical ordering, which benefits from the hierarchical architecture for the distribution of replicas and the protocol of updates propagation that specifies certain paths for updates propagation between the replicas in the replication architecture. This protocol ensures that each host propagates and receives updates from a few hosts only and reduces both the storage and computation overheads on each host. Based on this hierarchical structure, updates collection periods are enforced in order to provide mobile hosts with a reasonable time periods for updates collection for propagating their updates to the higher levels in order to be unified ordered with other updates that come from fixed hosts. This support both mobility and disconnections of mobile hosts, since the mobile hosts can disconnect or move to another cell during the same collection period at the higher level.

The system clock on each host does not ensure a unified time on all hosts for assigning timestamps for the different events, since its value may be different from a host to another. Also, both logical clocks and vector clocks do not provide the suitable solution for LMDDBSs due to their weaknesses. Accordingly, the suitable solution is the one that satisfies the following requirements:

(i) Providing a method for ensuring a unified time in all hosts in the system.
This method is provided in our ordering mechanism through enforcing the time of the master server on all hosts in a hierarchical manner in order to ensure a unified timing on those hosts. For example, we do not enforce the time of GMT because we cannot prevent users from changing it. Moreover, if any change happens on it during the system operation, the amendment process may fail to correct the timestamps that are already assigned incorrectly to events, which occurred during the period between incorrect values of the clock. While in the case of enforcing

the time of the master server, there will be a reference that we can return to it even if the times are changed in the underlying levels. For example, if there is any change occurs in the values of the Real-Like clock on both a mobile host and its responsible cell server, the reference for the correct value will be the zone server, which will send TVM to the cell server to correct its clock's value. Then, this cell server will correct the value of underlying mobile host.

(ii) Ensuring that the timestamp values assigned by the clock to events should reveal if they are causally ordered.

In the proposed ordering mechanism, since events are timestamped with the values of Real-like clock when they occur, this means that each event with smaller timestamp happened before the event that has bigger timestamp.

(iii) Scalability. It means here that as the system increases in terms of the number of replicas, there is no implication on the clock size or additional overhead in computation for assigning timestamps. In the proposed mechanism, the size of the Real-like clock is not affected by any increasing in the number of replicas. Moreover, the amendment process will not be affected even if the number of replicas increased. This is due to the benefit that is provided by the hierarchical structure, since the load for performing amendment process is distributed among all hosts in the system. Furthermore, each server is responsible for sending MVT to limited number of hosts that represent the direct underlying hosts.

Most of the previous works on causal ordering do not handle the ordering of individual updates (e.g. in a single file or single data item in the case of database) that are generated from different hosts by comparing with them. They just consider the order of messages that carry update information as is assigned by the sender, but they do not consider the exact ordering of updates that are performed on same data items at different time instants. This means that each host delivers received updates to other replicas in the same order as the order of the messages that shipped those updates from the host where they are generated. Accordingly, the ordering of individual updates on those research works is based only on the order of the messages that shipped them to the other replicas, but this may not reflect the exact order of them by considering the timestamps information of generation operations for those updates. For example, consider a product x where many additions and withdraws are occurred in its quantity in different hosts. In order to that the value of the quantity to be consistent in all sites that replicated the object of products, this requires that all these additions and withdraws should be ordered according to their preceding in the actual scheduling of their occurring (or actual time instants when they occurred) and not on the ordering of the messages that shipped them. Our mechanism relies only on the timestamps information of the messages in case of the comparing between updates based on generation process has resulted in concurrent updates.

7.1.4 IIRA Based Replication Method

A replication method that is based on IIRA is introduced in this research to allocate replicas to new hosts, and automate propagation of updates between the components of the replication system in a manner that ensures propagation of only recent updates among the large number of replicas. Moreover, the propagation system that is based on IIRA is used to implement the updates propagation protocol, since this system provides the required data structures and algorithms for implementing updates propagation in the replication system.

The automated propagation of updates in LMDDBSs via IIRA types represents a new idea in the field of data replication, since it is not discussed in the literature. The previous strategies only described their mechanisms for propagating updates through messages without mentioning how these updates are tracked and retrieved from their sources, and applied on the other replicas in an automated manner.

In addition to the automation of both updates propagation and initial replication processes, IIRA provides better utilization of the connection time between two hosts. This is because the connection time will be consumed only by the migration of the instance to the other host, since the migrated instance is populated at once with recent updates that to be transferred to the other host when the connection is realized with it. This instead of direct synchronization of updates, since it consumes the connection time by checking which updates are missed from the other host, and preparing the updates that should be transferred accordingly.

As compared to the mobile agent (Lange and Oshima, 1999), IIRA has two same attributes as the mobile agent, which are autonomy and cooperation. IIRA is autonomous because it carries out its activities independently (without the needing for the user intervention). IIRA is cooperative according to the interaction between its four types for exchanging recent updates.

The differences between IIRA and mobile agent are:

1. IIRA is bound to the hosting system, and only an instance of it migrates to another host, while mobile agent is not bound to the hosting system.
2. IIRA's instance moves only to the host that is currently connected with the home host, while mobile agent is free to move through the hosts in the network.
3. The migrated instance removes itself after performing its task without needing to distribute its working results to the home host as mobile agent does in most cases. This leads to better consumption of the bandwidth and utilization of the connection time between two hosts.

Thus, to decrease the communication cost and provide a better utilization of the connection time in environments prone to more frequent disconnections and failures, the proposed strategy relies on instance immigration instead of the migration of the agent itself as in mobile agents' communities. When the connection takes place, the instance holding recent updates will migrate to the other host, and performs its task. Then it removes itself without needing to return back to its home host as in mobile agents' communities. This minimizes the connection cost to only the cost that is imposed to transfer one instance per connection time.

7.1.5 SPN Model

A SPN model is developed for modeling the synchronization process between the components of the replication system. This SPN represents a first attempt as compared with the literature to model the synchronization process. Moreover, the research has proved that the synchronization process has Markov property as a new fact that is revealed by this research.

The developed SPN model reveals the following outcomes:

- The availability of all recent updates that occurred in both fixed and mobile networks to the mobile hosts depends directly on the propagation of these updates from their sources to the fixed network.
- The IIRA-Based propagation system ensures achieving the consistency of the replicated data in the mobile network according to the time of the last connection happened with the fixed network. Therefore, data inconsistency here will depend on the difference between the connection times.

- The CA state of the mobile database in which availability and consistency are satisfied depends directly on two conditions as follows. The first condition is that all updates that are performed in both mobile and fixed networks are propagated to the fixed network before the synchronization of the mobile host with the fixed network. The second condition is that these updates are totally ordered by the master server and propagated to the cell servers in order to be available for MHs.
- The probability that the mobile database in CA state is proportional to the number of its synchronizations with the fixed network.
- The IIRA-Based propagation system achieves load balance in both propagation and ordering processes. This is because these processes are shared by multiple hosts, where each host propagates a set of the recent updates to another in either lower or higher level, and each host participates in ordering the updates that are issued in its replicated database or collected from underlying levels.

7.2 Limitations

The limitations of this thesis are as follows.

1. The proposed strategy allows replicated data to be temporarily inconsistent, since it does not provide a method for enforcing all sources of updates (especially mobile hosts) to connect with other hosts once updates are generated.
2. The strategy relies completely on the servers that exist in the fixed network for ordering updates and propagating them to other hosts in the other levels. This is because the research considers that each server has fixed and stable connectivity with the server in the higher level. However, this acts in accord with the characteristics of LMDDBSs, since the servers are distributed according to the areas that are covered by the system.
3. The performance evaluation and the behavior tracking are based on the comparative study with previous strategies and using of SPN theory, respectively. This means that an implementation in a real LMDDBS is needed to provide the better evaluation. However, in the previous strategies, such an implementation is not provided in order to compare implementation results of the proposed strategy with them.

7.3 Future Work

As a part of our future work, the research will continue to address the following issues:

1. Providing strong consistency guarantee in LMDDBSs

The proposed strategy can be extended to provide strong consistency guarantee through developing a method that is based on the quorum protocol (see section 2.4.4) for restricting each update and read operations to specific quorums of replicas.

Our basic idea for strong consistency is that the hosts in each rim delegate the responsibility of performing the update operation to their secondary/main center point in the next inner rim. The center point acts as a coordinator that forms a quorum by contacting its neighbours in its rim as well as contacting its parent center point in the next inner rim, which also will be responsible for contacting its neighbours and the parent. Accordingly, the quorum is formed from a subset of servers that exist in the cell and zone rims in addition to the main center point. This ensures fast dissemination of the result of updates requests to all replicas including the replicas that have submitted the requests.

Thus, this method can be implemented for forming a quorum of the servers in the higher levels (master and zone rims) due to their fixed locations and the stable connectivity between them. However, in the lower level (i.e. cell rim), the method should consider the disconnections and mobility of mobile hosts, especially in the construction of quorums, submission of their requests to quorums, and providing them with the results of their requests.

Accordingly, our future research will address extending of the strategy to encompass a quorum based method and another alternatives for achieving strong consistency in LMDDBSs.

2. Creating new propagation wheels

In this research, the method of how the propagation wheel extends horizontally is presented. According to the mechanism of the horizontal extension, new propagation wheels (master areas) are established only when the replication system covers new geographic areas that have same nature as the existing areas in that they can be divided into smaller areas. However, there is a need for a method that specifically determines when a new propagation wheel is created according to both the size of the new area and the number of users and their distribution over that area. Also, this need is realized when the system covers new sub areas in the same master area to determine exactly when a new zone or cell is established.

3. New Mechanisms for data management challenges in LMDDBSs

LMDDBSs as an emerging area demand new mechanisms that act as solutions for data management issues in such systems. These issues include transaction management, query processing and optimization, concurrency control, and recovery. Accordingly, our future research will concern on providing such mechanisms for LMDDBSs.

4. Registering new hosts and Hand-off processes

The processes of how new hosts are registered in the replication system and the Hand-off (i.e. transferring the responsibility of supporting mobile data processing to the MSS of the new cell when the mobile host leaves the current cell) (Madria et al., 2002) have not been discussed in thesis, since it is concerned only on the issues that are related to the problems of consistency and availability. However, these processes may represent important issues in the researches that concern on location management and the other data management issues in LMDDBSs.

5. Implementation plan

A plan will be provided to develop the required specifications, tools, and interfaces to implement the proposed strategy in large-scale mobile healthcare environments to provide healthcare practitioners with an efficient access to updates that are performed on healthcare data. The reason of focusing on mobile health care is due to the increasing trend of using portable devices enabled with wireless technology by the healthcare practitioners to access and distribute healthcare data.