A DYNAMIC REPLICATION FOR PERIODIC TRANSACTIONS IN WEATHER FORECAST DATA DISTRIBUTION BY ORDER-RS

EI EI KHAING

M.C.Sc. **JANUARY 2019**

A DYNAMIC REPLICATION FOR PERIODIC TRANSACTIONS IN WEATHER FORECAST DATA DISTRIBUTION BY (ORDER-RS)

By

EI EI KHAING

B.C.Sc.(Honours)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon

January 2019

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere thanks to those who helped me with various aspects of conducting research and writing this thesis. To complete this thesis, many things are needed like my hard work as well as the supporting of many people.

First and foremost, I would like to express my deepest gratitude and my thanks to**Prof.Dr. Mie MieThetThwin**, Rector, University of Computer Studies, Yangon, for her kind permission to submit this thesis.

I would like to express my appreciation to **Dr. ThiThiSoeNyunt**, Professor, Head of Faculty of Computer Science, University of Computer Studies, Yangon, for her superior suggestion, administrative supports and encouragement during my academic study.

My thanks and regards go to my supervisor, **DawKhaing,** Associate Professor, Faculty of Information Science Department, University of Computer Studies, Yangon, for her support, guidance, supervision, patience and encouragement during the period of study towards completion of this thesis.

I also wish to express my deepest gratitude to**DawKhine Yin Mon**,Lecturer, the Department of Language, University of Computer Studies,Yangon, for her editing this thesis from the language point of view.

Moreover, I would like to extend my thanks to all my teachers who taught me throughout the master'sdegree course and my friends for their cooperation.

I especially thank to my parents, all of my colleagues, and friends for their encouragement and help during my thesis.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodies in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

---------- ----------

Date EiEiKhaing

ABSTRACT

Both scientific and business applications today are generating large amount of data, typical applications, produce a lot of data per year. In many cases, data may be produced, or required to be accessed/shared, at geographically distributed sites. Sharing of data in a distributed environment gives rise to many design issues e.g. access permissions, consistency issues, security. Thus, effective measures for easy storage and access of such distributed data are necessary. One of the effective measures to access data effectively in a geographically distributed environment is replication.Many real-time applications need data services in distributed environments. Replication can help distributed real-time database systems meet the stringent time requirements of application transactions.By replicating temporal data items, instead of initiating remote data access requests, transactions that need to read remote data can now access the locally available replicas. In this system presents a dynamic replication control algorithm, On-demand Real-time Decentralized Replication with Replica Sharing (ORDER-RS), designed for distributed real- time database systems.the algorithm decides where and how often the replicas are updated. In the propose system, the weather forecast data replicas are dynamically created upon the requests by the incoming transactions and their update frequencies are determined by the data freshness requirements of these transactions.This system is implemented using by C# programming language with Microsoft Access 2010 database.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In distributed environment, replication is mainly used for data distribution. Distributed data are equally distributed in each site by replication.The main goals to use replication are :

- Availability increment
- Performance increment and
- Reliability increment.

Due to the data are distributed in different places, if one of the sites failed to connect, the data can still be available from other sites. As, the data are stored in more than one place, the user can access data from their nearest places to reduce user latency and network traffic.

In a real time distributed database system, data access consists of multiple network operation and takes more frequent than the local data accessing. This main favorstomany transaction to deadline and misses. Remaining problem is occurred due to accessing geographically remote data access time, by the time some of the data may be stale for long access time. So, replication is a good method to solve the above problems. By replication, the remote data can now access only in local site. This assist the transactions to meet the low access time and latest data requirements [1].

The On-demand Real-time Decentralized Replication with Replica Sharing (ORDER-RS) algorithm is developedfor afield where all transactions are updated transactions by the time. When a new transaction occured, it describes data period, data needs, processing time and expire time. There are different ways to control the replica data consistency. Many replication methods are more suitable for more workloads and database requirements. Variousways of replication management in middle-scale or high-scale real-time distributed database systems and this system describean algorithm for replication: On-demand Real-time Decentralized execution time and deadline. There are multiple ways to handle the replication control. Different Replication with replica sharing (ORDER-RS). With this facts, the proposed algorithm determine where and how update the data for replicas.

For more detail, this algorithm can improvethe replication by On-demand Real-time Decentralized Replication for Replica data Sharing (ORDER-RS). The proposed, whole distributed systems are categorized into groups which are called cliques based on the network technology. The clique members share the replicas within one clique.

This system is studied on the weather forecast system by using ORDER-RS. Weather forecasting is the protocol for science and technology to forecast the states of the weather for a specific location and time. Peoples are always try to forecast the weather conditions since the 19^{th} century. Weather conditions predicting is the collection of frequent data about the current condition of weather at a specific place and using meteorology tool and how the weather conditions will change.

There are enormous ways for weather forecasting. Mostly, weather forecasting is based on a specific place of atmosphere condition such as temperature, moisture, humidity, and so on. Temperature predictingarepredict for the demand over next days. Everyday based on the weather conditions, people forecasts the surrounding weather condition to predict what to live on a given day.

1.1 Objectives of the Thesis

The objectives of this thesis are as follows:

- To implement a data reliable system. (consistency and data freshness)
- To reduce the data response time to remote site.
- To understand how consistency is important in database systems with multiple users.
- To implement a system in which all objects remain in a consistent state when they are accessed by multiple transactions.

1.2Overview of the Thesis

The proposed system developed a weather forecast system. This system includes three weather forecast stations: Yangon Station, Mandalay Station and Naypyitaw Station which are located in geographically remote regions. Each station updates each region of real time weather update information in each 30 minutes. And

each station can replicates data updates to remaining stations by using ORDER-RS algorithm.

1.3 Organization of the Thesis

This thesis is organized in five chapters. They are as follows:

In Chapter 1, introduction of the proposed system, objectives,overview and organization of the thesis are presented.

Chapter 2, presents the theoretical background related to distributed system,system's database, properties of ACID, classification of replication techniques, criteria of classification, replication's techniques, Primary replica based protocol, replicated write protocol, quorum based,the concept of replication, database replication, to choose database replication,database replication should not used, methods of performing database replication.

Chapter 3, describes the controlling for dynamic replication,model of scheduling and transactions, ORDER algorithm (On-demand Real-time Decentralized Replication), dynamic replication of ORDER-RS algorithm, existing classification, classification of dynamic replication strategies.

Chapter 4, expresses the design and implementation of the proposed system,system overview, process flow of the system, weather forecast system of database design.

Finally, Chapter 5 concludes this thesis with its limitations and further extension of the proposed system.

CHAPTER 2

THEORETICAL OF BACKGROUND

2.1 Distributed Systems

A replicated database mayuse group communication is a logical choice. By promoting the toolkits for group communications, the replicated database design will be easy to use and allowthe functionality of network is derived from database application in Figure 2.1.

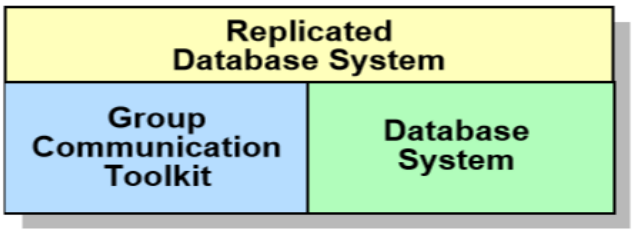

Figure 2.1 Basic Building Blocks

It applies the two segments from two communications: the distributed system communication, and the database communication. So, both segments depend on different architectures and pays different feature. Each communication tends to concern on specificcases and simple scope. The database communication leads to have a basic model of network, and the communication for distributed system is very simple for process model. Both communications have variouspoints, chances, but also variousarchitecture and metrics.

To establish a better system by using databases and group communication, both architectures must know and how reconcile. The output ofthe dedicate model can assist us and model of replication strategies in various ways. This means that communication primitives for all grouping in a single communication segment, nevertheless of the point that they started in one group or the other.

Duringsame data distribution, the system has been determined by both thedistributed system and database community for rare time, the stimulation for replicati-
on are very differ. This factsdescribe the goals for both communication.

The distributed system main goal is to make the consistent system by providingthepromising the facts on the passing of message and sequential. In the same techniques, the OS adds a basic layer on the top of hardware. This is gained by adding a basic layer on top ofnetwork. This model is shown in Figure 2.2.The system built on top of the group communication infrastructure can be an application, or another level of infrastructure, for instance a virtual shared memory environment [1].

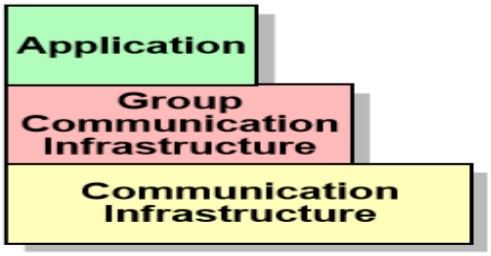

Figure 2.2 Group Communication Stack

Due to the communitymodel only emphasizeon it with sequential messages and reliable broadcasting, the replication techniquesrelated with these communication factorsto determine a basic model ofapplication. In multiprocessing, load balancing and enormous data migration are formal of complex designed application. This facts are often determined out of the focus of communications group. This is in order to multicast a complex application, handled by using the specific communication paradigm, a configuration is required.

2.2 System's Database

Data and its storage are handle by the database systems. Replication is not a mainscope of the database environments, and after a crash, databases are concerned to correctly recovered.

For performance and error resistant, replication has been determined for administrative facts. To warranty the facts of fault-tolerance used the specialized hardware. Level of hard- ware layer are often used for Replication.

Types of transactions are influence on database system. Transaction processingmust obey the ACID properties. During the crash in database system,transactions are discarded and rolled-back to a system consistent state.

Distributed databases are always controlled as more layer on top of existing systems. Distributed transactions divides one sub-transaction on each processor, those sub-transactions are parallel processing using an atomicity between all processors.

Distributed databases have been considered for some time , and are usually handled as an additional layer on top of existing systems. Distributed transactions span one sub-transaction on each node, those sub-transactions are synchronized using an atomic commitment protocol that ensures atomicity between all nodes.Figure 2.3 illustrates this architecture.

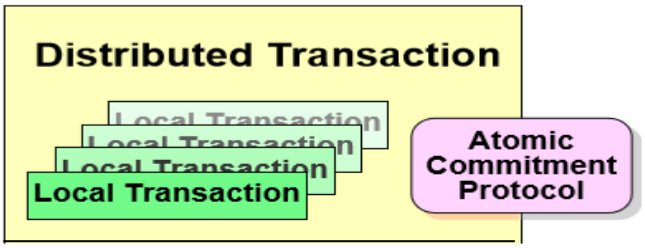

Figure 2.3 Distribution of transactions

Normally,replication is a sector of distributed databases system: distributed transaction refreshes all replicas of the system databases. While processing a transaction, if a nodedown for connection,the transaction is discarded and needs to be re-run. As the transaction discards for manycases (for concurrentconflict) so users have to tryaborted transaction in such a way. This wayis not to be precise, and the analyzeremphasize on constraintsrelaxation. Due to this unpredictable failures and discards make the distributed database systems unfit for essential and applications for real-time system [6].

As there is a greatconcern on performance, the database systems are massively used in core organizations.Basicarchitectures are released to get more performance.

2.3 Properties of ACID

In the distributed data system, every transaction processing must impact the following properties:

Atomicity **:**If a transaction commit in one node, this transaction must be commit in allremaining nodes. If not, none are done at every node.

Consistency **:**If a transaction commit in one node, this successful transaction result must know from all other remaining nodes. (i.e,to keep datatransa-rency)

Isolation: All sequentially processed commit transactions must propagatesequentiallyto all remaining nodes.

Durability: If a transaction processing commit for all nodes, the rollback of the successful transaction cannot be done.

2.4 Classification of Replication Techniques

The database replication techniques classification is described by Gray et al. [7]. The categorized ways are shown in Figure 2.4.First, object ownership presents "own" the dataof the nodes. Data can update by the owner only. This pointdeterminesthe node for the allowance of transactions processing. If all nodes have the data, any node can be processedtransactions. If the data is only in server node, only the server can process the transaction [4]. In replication technique of update everywhere, the server node is also called primary or backup.

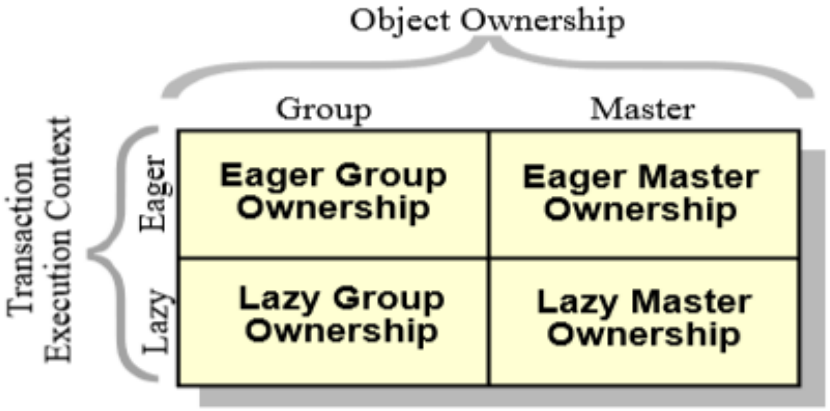

Figure 2.4 Gray of Classification

Secondly, transactions scopes are executed. In eagerly replication, the all replicas updates are done in distributed nodes: at any point the transaction can be discarded by any replica. This ensures all copy in consistent also called parallel replication. In the replicationof lazily, one replica updates the transaction, and the update are multicast to others remainingafter the transaction commit on the host replica. This situation may favor to become an inconsistent system in case of concurrent access or failure states, but even no failure conditions can also break the ACID properties. This type of replication also called asynchronous replication.

2.5 Criteria of Classification

Eager replication can be classifiedbased on nature and properties of the protocol and, its performance. They are detailed categorized as follows:

2.5.1 Architecture of Server

Server Architecture can becategorizedbased ownership of group and ownership of master.

2.5.1.1Primary Copy

The primary-copy isthe whole data associated with the whole system. At the primary-copy processing of any update can be done firstly. Then, primary copy multicast updated data to all remaining nodes. So, the primary-copy techniques are also called passive replication [6] and mainly depend on group communications.

Aprimary copy failure is a difficulty ofPrimary-copy technique.The primary server selection technique may additionally required if the primary crashes and one of cope must be serve the tasks of primary as primary role. To overcome these difficulties, all data items should not be kept as a single primary. So, the data must be replicated on more than a single server or single primary copy.

2.5.1.2Update Everywhere

Replication of update everywhere techniquepermits the updates to a data item from any- where of the system. Therefore, concurrent updates may occur on the same data. When a node failure is occurred,there is no selection protocol is need to continue accessing, Because of the property of update everywhere mechanism. Although update everywhere is support for system performance, there is a difficulties because of all copy must be executed for a transaction processing. So, this protocol may not efficient for processing cost and resource utilizing.

2.5.2Termination of Transaction

Transactions terminate is ensure the transaction atomicity. The transaction termination is an additional work on all copies.

Basically, transaction termination is related on two properties of replication.No voting termination is occurred if a primary-copy technique is usedand voting phase is required for update everywhere because of the management of concurrency and control for consistency of the distributed replicated data copy.

2.6 Replication Techniques

The various techniques for replication scheme of update everywhere. Clients can process their requested transaction at any server by update everywhere techniques. All classification of update everywheremechanisms are discussed as follow:

2.6.1 Update Everywhere, Constant Interaction in Non-Voting Techniques

Figure 2.5explains the infrastructure of update everywhere. Between distributed replicated data copy, the communication is only depend on network infrastructure. The discussion moresimple and only emphasize on critical point this technique. Transaction processing under the control of this protocol obeys the following steps of transaction processing [8]:

1. At the delegate server, the transaction processing is started.
2. The non-deterministic way is used for transaction processing.
3. This step shows the arrival point of determinism.
4. By the atomicity rule,all servers received the transactions.
5. By determinism, all replicas continues on processing.
6. By same way, the transaction is terminated at each replica.

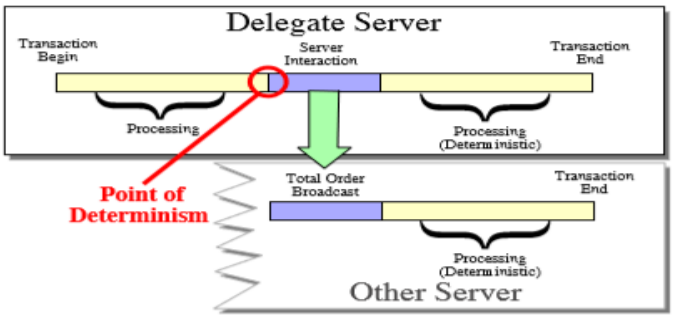

Figure 2.5 Non-voting,Update Everywhere, Constant Interaction

2.6.2 Update Everywhere, Constant Interaction in Voting Techniques

Figure 2.6 shows the basic structure of such a replication techniqueIn this discussion point of view, one communication link is usedfor all the interactions. Moreover, to maintain all replicas data consistency on the processing result,the system executedthe voting phase at the end of the execution. The requested transaction processing is executedto ensure that all replicas agree on the outcome [8]:

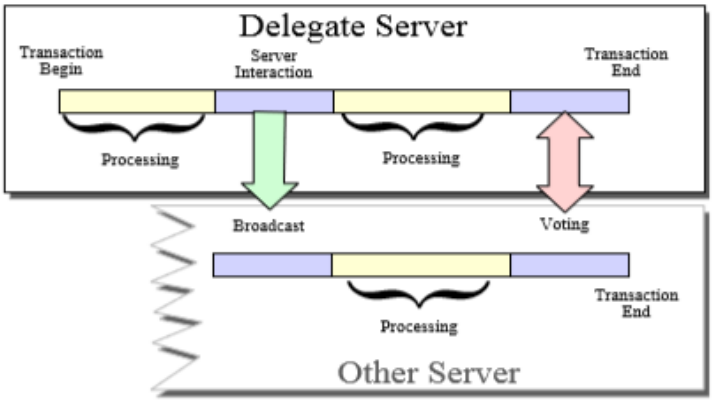

Figure 2.6 Voting,Update Everywhere Replication, Constant Interaction

1. At the delegate server, the transaction processing is started.

2. The non-deterministic way is used for transaction processing.

3. All servers received the multicast transactions.

4. All replicas carry on processing.

5. At the end of processing, voting phase is executed for termination

6. Based on the result of the voting phase, the transaction is terminated at each replica

2.6.3Update Everywhere,Linear Interaction,Non-Voting Techniques

This technique does not used voting. So, this is deterministic for full. There is no need to send the requested transaction to all replicas. But, the transaction is terminated at the delegate siteend. Although, this mechanismusesa message for termination, it is not a message for voting. The detail structure of this technique is described as follow [8]:

1. At the delegate server, the transaction processing is started.

2. By a total order broadcast, all servers received the initial operation.

3. All servers executed the initial operation.

4. Until the transaction is terminated phase 2 and 3 are repeated.

5. To notify the transaction termination, a termination message is sentby delegate.

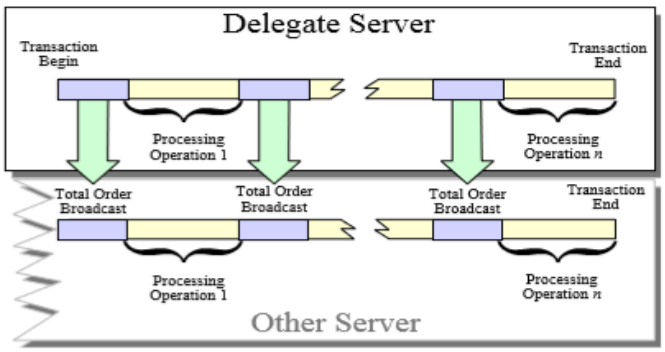

Figure 2.7 Non voting,Update Everywhere, Linear interaction

2.6.4 Update Everywhere,Linear interaction,Voting Techniques

This technique is mostly occurred in theoretical concept. This is also known as read one – write all technique. Figure 2.8 shows the detail transaction processing by the scope of this dimension [8].

1. At the delegate server, the transaction processing is started

2. Quorum sites is received the broadcast transactions.

3. Each quorum site perform the received transaction processing.

4. Until the transaction is terminated phase 2 and 3 are repeated.

5. For termination, the system executes the voting phase.

6. Based on the result of step 5, the transactions are determined for termination.

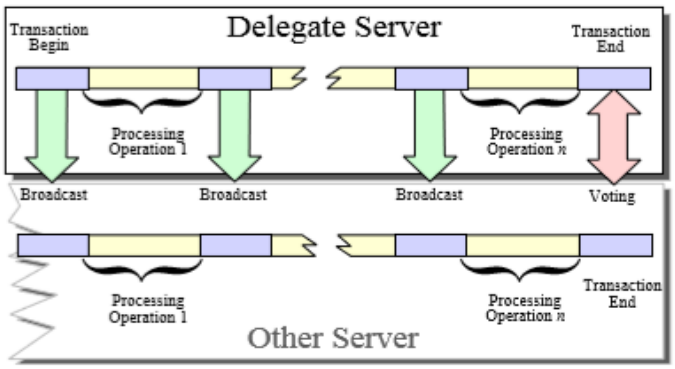

Figure 2.8 Voting,Update Everywhere, Linear Interaction

2.6.5 Primary Copy,Constant Interaction,Non-Voting Techniques

Techniques in this class are usually utilized for cold-standby replication. The protocols have the subsequent standard outline in Figure 2.9:

1. The transaction is executed at the primary.
2. While the transaction terminates, the corresponding log information are sent to all backups making use of a fifo dependable broadcast.
3. The number one commits the transaction without watching for the backups to put in the transmutations.
4. The backups eventually deploy the transmutations.

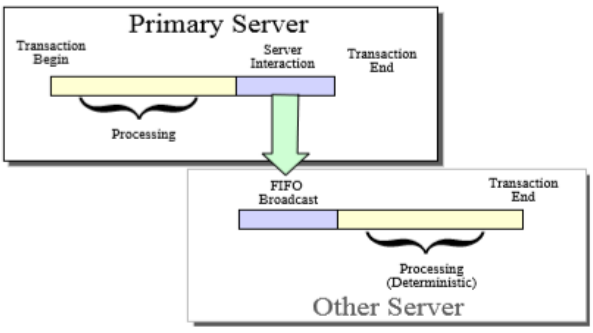

Figure 2.9 Primary copy, Constant Interaction, Non-voting

The concrete nature of the protocol depends at the type of broadcast primitive applied. In its handiest form, the protocol is predicated on fifo distribution, that allows the transaction changes are mounted on the backup inside the identical order they have been done at the primary [8].

2.6.6 Primary Copy,Constant Interaction,Voting Techniques

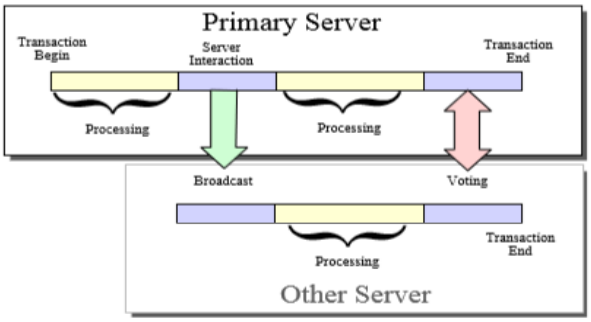

Figure2.10 Primary copy, Constant Interaction and Voting

The exordium of a voting phase sanctions us to examine that each the number one and the backups install the updates. At the same time as 2-secure belongings can be ascertained without a vote casting phase, having a vote casting phase is the traditional manner to enforce the two-protection assets. Balloting additionally gives flow manage and for that reason sultry- standby demeanor: the system must look ahead to all replicas to be yare to install the vicissitude of a transaction afore committing the transaction. Consequently, a reproduction can't be left behind. The protocol is the subsequent in Figure 2.10:

1. The transaction is achieved at the primary.
2. While the transaction terminates, the corresponding log information are broadcast to all backups.
3. The number one initiates an atomic commitment.
4. The transaction is established and devoted at all websites.

2.6.7Primary Copy, Linear Interaction,Non-Voting Techniques

With constant interaction strategies, ready until the transaction ends which will propagate the vicissitudes denotes that the replicas will have hassle keeping update with the primary (sultry-standby) [11]. The protocol might be extra

expeditious if the backups can system transaction in parallel to the primary. So that you can do this, the primary sends operations as they're completed, thereby sanctioning the backups to begin doing some paintings. If no balloting phase is involved, the protocol is as follows Figure 2.11:

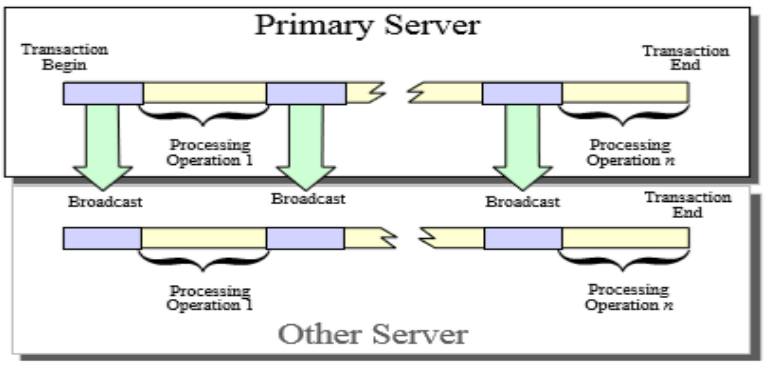

Figure 2.11 Primary-copy, Linear Interaction, Non-voting

1. The transaction commences on the primary.
2. Examine operations are achieved regionally.
3. The consequences of indie operations are broadcast to the backups.
4. A termination message betokens the cessation of the transaction.

2.6.8 Primary Copy,Linear Interaction,Voting Techniques

The purpose of introducing a vote casting phase is to envision sultry-standby comportment in the following Figure 2.12:

1. The transaction commences at the primary.
2. Examine operations are executed domestically.
3. The effects of indie operations are broadcast to the backups.
4. The primary commences an atomic commitment protocol.
5. The transaction is set up and devoted in any respect websites.

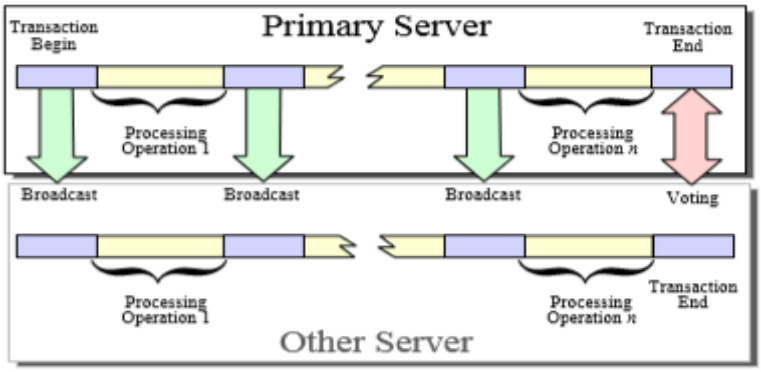

Figure 2.12 Primary copy, Linear Interaction, Voting

2.7 Primary Replica Based Protocol

Primary replica based protocol,all write operations to a data object x is attended via one particular duplicate that called primary replica. This primary reproduction is answerable for updating other replicas; the patron just cooperates by using this primary duplicate.Necessities should be came about for this beneficent of protocol:

- all study and write operations for updating a data item x must unfold and be carried out all replicas at some time.
- those operations ought to be finished within the identical order.

2.8 Replicated Write Protocol

Replicated write protocol, each write operations are sent to each replica to update procedure. There are two types for replicated write protocols.

2.8.1 Active Replication

In lively replication, each replica includes a concomitant technique that transports out the replace operations. Not like different protocols, update operations are generally propagated thru the write operation. This propagation reasons the operation is dispatched to each reproduction. Also there is required a complete order for all write operations that every duplicate execute the same order of write instructions [2].

2.8.2 Quorum Based

The quorum based protocol specifies that the customers reap the authorization of numerous servers earlier than any analyzing or writing a replicated statistics object x[16]. For instance, the write operations simplest need to be performed on fragment of all replicas earlier than go back to the purchaser. It makes use of elections to avoid write-read conflict and write-write battle:

- r is the range of replicas of each information object.
- rr is quantity of replicas that a customer need to touch with the aid of them for analyzing a fee.
- rw is range of replicas that a purchaser must touch by them for writing a cost.
- for preventing the write-write and write-study conflicts,
- rr + rw> r and rw + rw> r need to be happy [2].

2.9 The concept of Replication

Replication represents the manner of sharing facts to ensure consistency among redundant sources, including software program or hardware components, to enhance reliability, fault-tolerance, or accessibility. It can be records replication if the equal information is saved on more than one storage devices or computation replication if the same computing challenge is performed normally. The access to a replicated entity is commonly uniform with access to a single, non-replicated entity. The replication itself must be obvious to an external consumer. Further, in a failure state of affairs, a failover of replicas is hidden as much as feasible.In structures that mirror statistics, the replication itself is both active or passive.

An lively replication whilst the identical request is processed at every replicated instance and approximately passive replication when every request is processed on a unmarried duplicate and then its state is transferred to the opposite replicas. If at any time one master replica is unique to method all of the requests, then the number one-backup scheme (master-slave scheme) foremost in excessive-availability clusters. On the opposite side, if any replica approaches a request after which distributes a brand new nation, then that is a multi-primary scheme (known as

multi-grasp within the database field) [2]. Even thought, the manner of facts replication it’s used to create instances of the equal or parts of the same statistics, it with the manner of backup given that replicas are frequently updated and quickly loseany historic nation. Backup on the other hand saves a copy of facts unchanged for a protracted period of time.

2.10 Database Replication

Database replication is the process of creating and keeping multiple instances of the equal database and the system of sharing facts or database layout modifications between databases in one-of-a-kind places while not having to replicate the whole database.

In maximum implementations of database replication, one database server keeps the master copy of the database and the additional database servers keep slave copies of the database. The 2 or greater copies of a single database continue to be synchronized [6].The authentic database is called a design grasp and each replica of the database is referred to as a replica. Together, the layout grasp and the replicas make up a duplicate set. There may be most effective one layout grasp in a duplicate set. Synchronization is the system of making sure that each copy of the database carries the equal objects and data. When person synchronizes the replicas in a reproduction set, most effective the statistics that has changed is up to date.

Database writes are sent to the master database server and are then replicated by means of the slave database servers. Database reads are divided amongst all the database servers, which results in a large performance advantage because of load sharing. The transaction processing load may be distributed among all the replicas inside the device. This ends in a larger throughput (due to the fact queries and examine operations do now not trade the database nation, they can be independently finished in one replicas best) and a shorter reaction times for queries (due to the fact queries can be completed only in a single reproduction, that's typically inside the same area because the patron, and with none additional verbal exchange many of the replicas) [8]. Further, database replication also can improve availability due to the fact the slave database servers can be configured to take over the master role if the grasp database server turns into unavailable [3]. If one reproduction crashes due to a

software or hardware failure, the last replicas can nonetheless continue processing, whilst centralized database system will become absolutely unavailable after most effective one crash [7].

2.11 To Choose Database Replication

Implementing and keeping replication might not be a easy proposition. If numerous database servers that want to be concerned in diverse types of replication, a easy undertaking can speedy become complex.

Imposing replication also can be complicated by the software architecture. But, there are various situations wherein replication may be applied [4].

Database replication is nicely applicable to commercial enterprise solutions that want to:

- percentage records amongst far flung offices
- share facts amongst dispersed customers
- make server facts greater reachable
- distribute solution updates
- returned up facts
- offer internet or intranet replication

2.12 Database Replication Description

Although database replication has many blessings and may clear up many problems in disbursed-database processing, the reality that during a few conditions replication is much less than ideal. Database replication is not encouraged if:

- There are frequent updates of existing records at more than one replica.Solutions which have a large number of report updates in unique replicas are likely to have more document conflicts than solutions that clearly insert new facts in a database. If changes are made to the same report by distinctive users and on the identical time then file conflicts will really appear. This will be actual time consuming because the conflicts should be resolved manually.

- Facts Consistency is important at all times.Solutions that depend on facts being correct always, which include price range transfers, airline reservations, and the tracking of package deal shipments, normally use a transaction approach. Despite the fact that transactions can be processed within a replica, there may be no help for processing transactions across replicas. The facts exchanged between replicas for the duration of synchronization is the end result of the transaction, not the transaction itself.
- The replicas require additional conversation to make sure that adjustments are carried out to all of the database copies, which increases the load at the machines and within the verbal exchange community, accordingly degrades the general machine performance [13].
- Gadget complexity, synchronization of the database copies among replicas requires utilization of advanced conversation and transaction processing algorithms [13].

2.13 Methods of performing Database Replication

Database replication can be performed in at least three different ways [15]:

- **Snapshot replication:** Data on one database server is plainly copied to another database server, or to another database on the same server.
- **Merging replication:** Data from two or more databases is combined into a single database.
- **Transactional replication:** Users obtain complete initial copies of the database and then obtain periodic updates as data changes.

2.13.1 Snapshot replication

This kind of database replication is one of the most effective technique to installation, and perhaps the very best to understand.

The snapshot replication approach functions by means of periodically sending facts in bulk format. Normally it's far used whilst the subscribing servers can function in study- handiest surroundings, and also whilst the subscribing server can function

for a while without updated records. Functioning without updated records for a time frame is called latency.

As an instance, a retail keep uses replication as a method of retaining an correct stock at some stage in the district. For the reason that inventory can be controlled on a weekly or maybe monthly basis, the retail stores can function without updating the relevant server for days at a time. This state of affairs has a high degree of latency and is an excellent candidate for photo replication.

Extra reasons to use this form of replication encompass situations with low-bandwidth connections. Since the subscriber can final for some time without an replace, this gives a solution that is decrease in fee than different methods whilst nonetheless managing the necessities.

Snapshot replication also has the brought gain of being the most effective replication kind wherein the replicated tables are not required to have a number one key. Image replication works through studying the published database and developing documents within the running folder at the distributor. These files are known as snapshot documents and include the facts from the published database in addition to some additional information that will help create the preliminary replica on the subscription server[5].

Picture replication is often used whilst desiring to browse facts along with charge lists, on-line catalogs, or statistics for selection aid, wherein the maximum present day information is not essential and the facts is used as study-best.Figure 2.13 describes the snapshot replication. Snapshot replication is helpful when:

- Data is mostly static and does not change often.
- It is acceptable to have copies of data that are out of date for a period of time.
- Replicating small volumes of data in which an entire refresh of the data is reasonable.

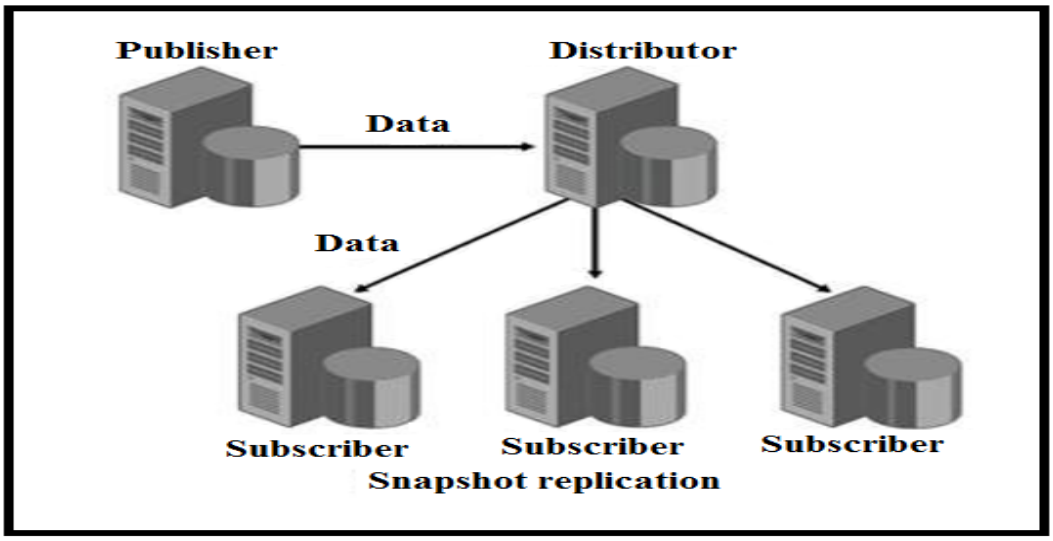

Figure 2.13 Snapshot Replication

2.13.2 Merging replication

Merge replication is the technique of dispensing records from publisher to subscribers, permitting the writer and subscribers to make updates whilst connected or disconnected, after which merging the updates among sites when they may be related.

Merge replication allows various sites to work autonomously and at a later time merge updates into a single, uniform end result. The preliminary picture is implemented to subscribers, after which adjustments are tracked to publish data on the publisher and at the subscribers. The data is synchronized between servers constantly, at a scheduled time, or on demand. Because updates are made at more than one server, the equal records can also were updated with the aid of the writer or by means of multiple subscriber. Consequently, conflicts can occur when updates are merged.

Merge replication consists of default and custom picks for war resolution that configure a merge book [16]. While a struggle happens, a resolver is invoked by means of the merge agent and determines which records may be established and propagated to other web sites.

Merge replication is useful whilst:

- A couple of subscribers need to update records at various instances and propagate the ones changes to the publisher and to different subscribers.

- Subscribers need to acquire information, make adjustments offline, and later synchronize changes with the publisher and different subscribers.
- Person does now not assume many conflicts when records are up to date at a couple of sites (because the information is filtered into partitions and then published to distinct subscribers or due to the uses of consumer's software). However, if conflicts do arise, violations of acid properties are ideal [15].

2.13.3 Transactional replication

The transactional replication works through sending changes to the subscriber as they happen. For example, square server strategies all moves within the database the use of transact-sq. Statements. Every finished assertion is referred to as a transaction.

In transactional replication, every dedicated transaction is replicated to the subscriber because it happens. The replication manner will acquire transactions and ship them at timed intervals, or transmit all adjustments as they occur. This type of replication has a decrease diploma of latency and better bandwidth connections.

Transactional replication requires a continuous and reliable connection, due to the fact the transaction log will grow quickly if the server is unable to attach for replication and can emerge as unmanageable. Transactional replication starts offevolved with a snapshot that units up the initial copy. That replica is then later updated by the copied transactions. How often to replace the photograph, or choose now not to update the picture after the first replica.

As soon as the preliminary snapshot has been copied, transactional replication makes use of the log reader agent to study the transaction log of the published database and shops new transactions in the distribution database. The distribution agent then transfers the transactions from the writer to the subscriber.

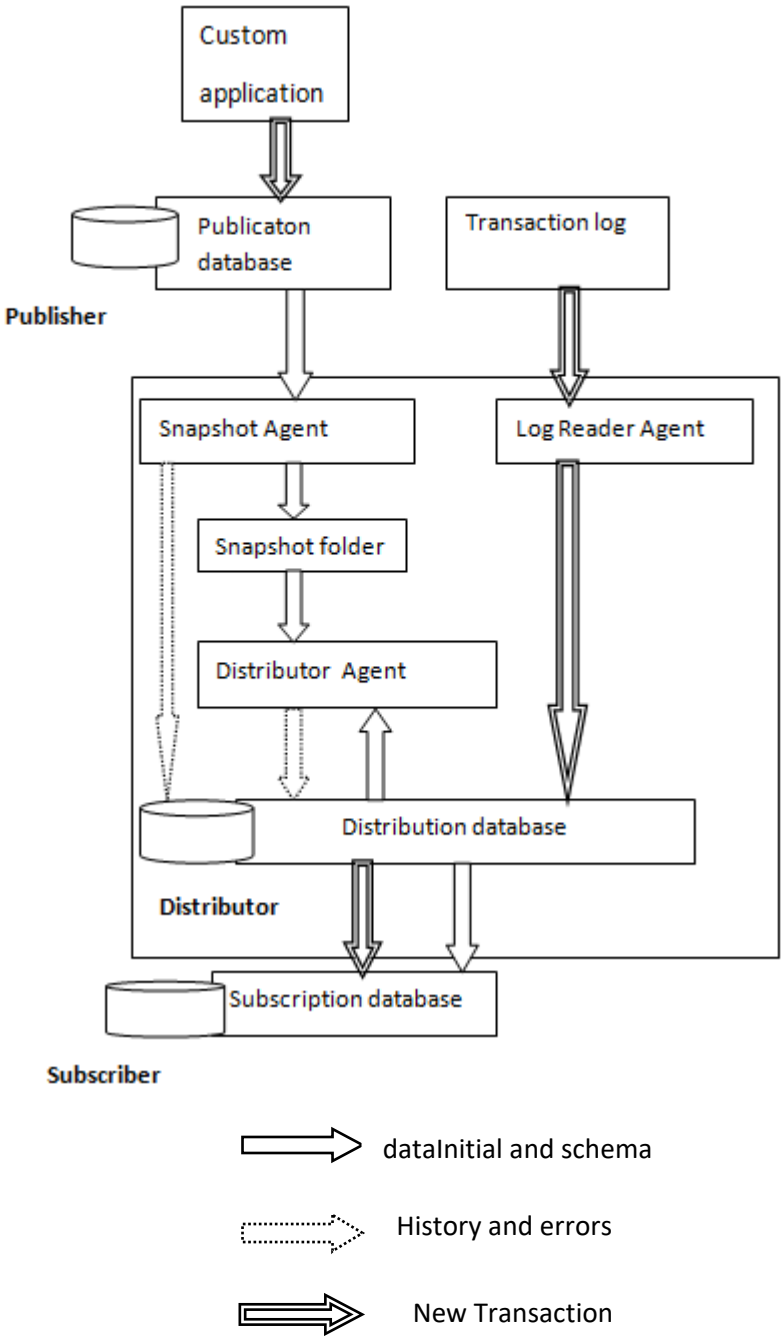

Figure 2.14 Transactional Replication

Transactional replication with updating subscribers: An offshoot of general transactional replication, this approach of replication basically works the same manner, however adds to subscribers the ability to replace information. While a subscriber makes a change to information locally, square server uses the microsoftdispensed transaction coordinator (msdtc), a component protected with square server, to execute the equal transaction on the writer. This manner permits for replication scenarios in which the posted information is taken into consideration read-

most effective most of the time, but can be modified on the subscriber from time to time if wanted [5].

Transactional replication with updating subscribers requires a permanent and dependable connection of medium to excessive bandwidth [6].

CHAPTER 3

DYNAMIC REPLICATION

Real time of many applications need to apportion data, these data are distributed to multiple sites.Example of controlling system in navel combat,scattered of different combat-ship and submarines are share and collect of sensor values and authentic time of database systems[1].In the distributed real timedatabase system of remote data and local data access consists multi hop network operations.In these local data are accesses more substantially time than the remote data accesses.Immense number's transaction deadline are misses this potentially.Another problem is that due to the long remote data access time, by the time a transaction gets all the data it needs, some of its data items may have already become stale.

Aforesaid quandaries,replication method to solve the effected.Initiating remote data of access requests instead of temporal items data by replicating,can nowavailable replicas of the locally data instead of remote data available.The replication help requisites of data freshness and meet the time of transactions. The control of replication are handle to multiple ways replication. Many different replications are more better for different data specifications and different data's workloads.In the work antecedent on replication,[2] the replication works distributed authentic time of database system well in scale-diminutives.Moreover,the system within the replication is utilized fully replication.Present a replication algorithm called On demand Real-time Decentralized Replication (ORDER),the paper in these,examine many replications control authentic time in distributed data of small scale and medium scale distributed data .

Designed in the ORDER algorithm to work in a environment, where all of the data types and the system of cognations are kenned as priori, and the transactions are kenned as short-term transactions periodic. The algorithm work a transaction arrives, it declares deadline, data needs and time of execution. How replicas update this information by deciding the algorithm.Greatly ameliorate system performance shows the result type by using ORDER algorithm. Term of scalability,the ORDER (On demand Real-time Decentralized Replication)algorithm can be enhanced to another replication algorithm.These algorithm are called ORDER-RS(On-demand Real-time

Decentralized Replication with Replica Sharing).These algorithm are suitable for immerse scale distributed system.On the network topology immensely distributed systems are divided into minute groups. The minute groups are called member of cliques.A clique by sharing the members of clique within the replicas.The ORDER algorithm compare the ORDER-RS algorithm,when theimmerse scale distributed system can further performance improve.

Authentic-time of distributed database system,by connecting the high speed networksconsists authentic-time databases a group of main recollection.A high performance of accesses recollection and decrementing of main recollection's costs are increasingly utilized for database management of authentic-time.

3.1 Model of Scheduling and Transaction

Thetransactionsinthesystemaredividedintotwotypes.These two types are transaction of utilizer and transaction of update or freshness transaction.The sensor of data or temporal of data and update transaction replica data are include transactions update system.From application,utilizer transactions takes queries or updates. The transactions must be access fresh of temporal data [15],the update transactions of temporal data must get firstly scheduled and then the update transaction replica.The transactions application incoming can get transaction update replica and update temporal data get after scheduled.

Operations as asequences on the data object are transactions.Sequential of fashion are executed one transaction of operations.All the antecedent operations are notfinished,one of the operation cannot be executed.The validation stage,the transactions are finished once all of the operations.Freshness of accessed data check the system in the validation stage.The transactions are restarted,if the items data not fresh of accessed data items.Transaction enters the commit stage,the system checks the accessed of data update on the validation stage.The system checks where gets data items freshness,the transaction needs to read another sites of data items and available of update local data copy [16].

If the operation is not read or there is no fresh local copy available, the transaction sends out data accommodation requests to establish cohorts atremotesites.

The cohorts read or indie data items on behalf of the transaction. At the validation stage, the transaction sends out a commit preparation message to all cohorts. As in the 2-phase commit protocol, the transaction commits if and only if all cohorts concur to commit.

3.2ORDER Algorithm(On-demandReal-time Decentralized Replication)

The goal of the On-demand Real-time Decentralized Replication**(**ORDER) algorithm is to gain efficiency over replication strategies such as full replication by dynamically changing the update frequency and update duration of replicas.

3.2.1 Update Frequency and Duration Definitions

The ORDER (On-demand Real-time Decentralized Replication) in database model, each node may contain multiple temporal data items and replicas. The primary replica of a data item is updated periodically at a given basic fundamental update frequency (BUF) while its replicas are updated at different elongated update frequencies (EUF) specified by the incoming application transactions. All replicas of a particular data item are updated utilizing the fresh value from their primary copy. The EUF upper bound for a given data item is the BUF of the primary copy. When a replica is updated periodically, it is called an active replica. Otherwise, it is called a dormant replica. An active replica becomes dormant if it is no longer needed by any incoming application transactions. The time that it turns into a dormant replica is called its closing time (CT) [17].

3.2.2 Definite Periodic Workload Model

The periodic transaction workload model where transactions are periodic with definite data requisites and accommodation durations. A transaction may arrive at any time and each transaction may contain requests for multiple data objects from different sites. The specification for a transaction is {TID,TD,EXETime,SF,DS}.

Eachtransactionspecificationcontainsatransaction identifier (TID),atransaction duration (TD), execution time (EXETime), slack factor (SF), and a data set (DS). The transaction’s dataset consistsof elements as DataObject{SiteID,DataType,DataID,FR}

The specification for one data object consists of information of the database identifier (site ID), data type, data identifier (data ID) and the freshness requirement (FR) of that data item. When a transaction arrives, it requests a data service with specific data freshness requirements and indicates the duration of the service. When this duration expires, the transaction no longer requests data objects at this site [19].

3.2.3 ORDER Algorithm Description

Incoming transactions of the freshness requirement data are changed dynamically the frequency update and replica update duration in the OADER algorithm.If the transactions can be admitted predicated on the current system of conditions,the algorithm evaluates and engenders replica data when the transactions receiving.And additionally, registration on the primary site of the frequency update replica and duration.It is the job of the algorithm,receiving the data site to register the replica active data to their primary site.Incoming transactions,obligation of primary site to push update to the replication active data.When the transaction on the local site admits,the algorithm evaluates the felicitous update frequency data and update data duration of the remote item data specification by the transaction.The algorithm suppose,receive the data request of their temporal items data site from the replication of the another site.

The algorithm decides, where are subsisting active replication data items for the remote item data,transaction request for each temporal item data of remote site when arrive the incipient transaction.By the incipient transactions, the algorithm compares the replication of the active frequency and the current frequency of the replication requested,if has already of the active replicas.By the transaction request, the update frequency of the active replica are transmuted,the replication to be updated at the higher update frequency if the request of the incipient transaction.In the case, current time add the incipient transaction’s duration are transmuted of the time closing of the replication.When the frequency update requested,the pristine update frequency is more than the incipient transaction, the algorithm not require to do any,the current frequency update is high enough. The algorithm engenders the active replica of the data item utilizing the incipient transaction and the frequency update,there is no replica active of the temporal remote site of the data items.

The algorithm,before the expire of active replication must track all of transactions that utilize, to get the minimum frequency of update data for all of the active replica.By expiring of transaction,these algorithm need to re-evaluatethe replicas of frequency update data items that are accessed.Example,consider the case,transaction requests the remote temporal data items are updated every 5 seconds for 100 seconds of the duration.And then,at the same data items to be updated for a duration of 20 seconds in every 2 seconds,different transaction arrived at 20 seconds.So,the replication of update frequency to once every 2 seconds the duration of 20 seconds ,the algorithm set the update frequency replica.The algorithm of the system must be recuperated the frequency update data to once every 5 seconds at the second transaction are expire at the 40 seconds[18].

3.3Dynamic Replication of ORDER-RS Algorithm

The ORDER algorithm not well perform in the authentic time databases are distributed in astronomically immense scale.Firstly, all of the data items of detail information to maintain in the immense colossal scale distributed data system .The system must be many cost and infeasible for the every site.Moreover,the algorithm not enough to get freshness items data directly at the primary site.More proximately,instead the site from some subsisting get the freshness replicas.And then,the request of data items can reach with the less transactions delay time.In Figure 3.1 the given example,two authentic time database (DB1 and DB2),connected high speed LAN.At the remote item data, database DB1 has the active replica.The DB1 active replica is update periodic utilize form the primary site of freshness items data values.In this time,database DB2 request the same data items need an incipient transaction.Visually perceive,it is not efficient,DB2 can get easily request data items form DB1because the DB1 has already for request of the same data DB2 needed,which can be facilely reached.Unnecessarily from the primary site get directly many workloads network of the site.More desirable, instead of freshness data items from DB1.The requirements of the new transaction,the replica of frequency update data can slake the requisites to the incipient transaction of DB2.The system needs DB1 of Active Replica1 to DB1 of Active Replica2.The system stop now replicating the Active Replica1 from primary site when the incipient transaction are higher frequency update data.And then, at the higher frequency update data,the primary data

to Active Replica1 is update directly instead of the data from Active Replica 2 to Active Replica 1 [12].

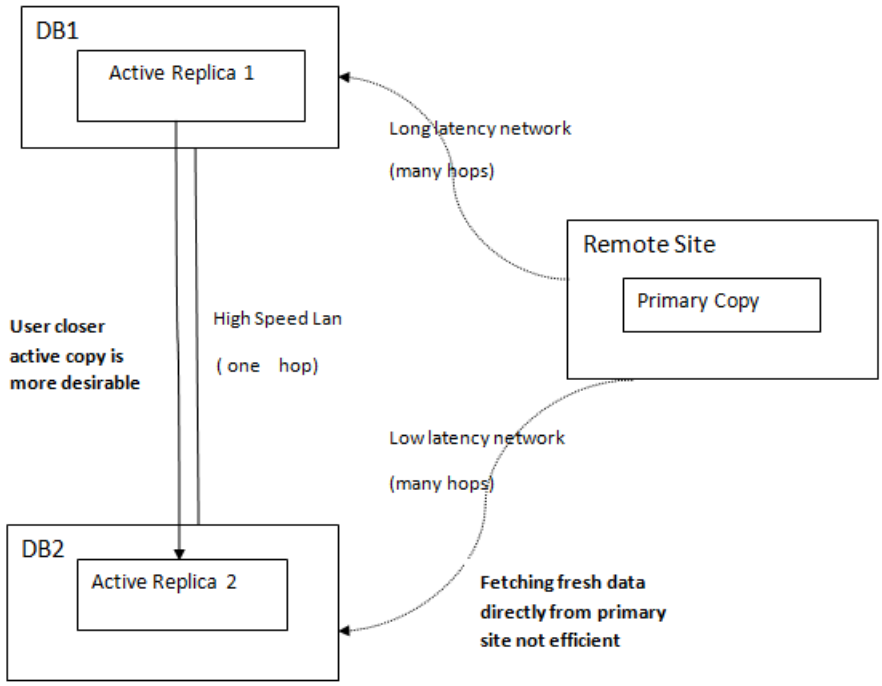

Figure 3.1 Getting the freshness data form closer active replicas

Utilize theconception,aforesaid,sharing the active replica those sites are proximate in the transaction delay time to each site,elongate in the algorithm.The algorithm is called ORDER-RS(On-demand Real-time Decentralized Replication with Replica Sharing).The Figure 3.2 is for the example,medium scale distributed authentic time database system,the database consists of 24 authentic time database serverssystem.By connecting the 8 transmission stations in the server of the authentic time database.On the network topology,the system divided into 6 cliques.Each clique are connected by wired more speed networks in the authentic time distributed database.Same clique are known the subsistence of the each other.The clique by multiple hop wireless networks are connected.The authentic time system configuration suitable the assumption.In the systems,high speed ethernet by connecting in the authentic time server database,the communications between the wireless network [6,14].

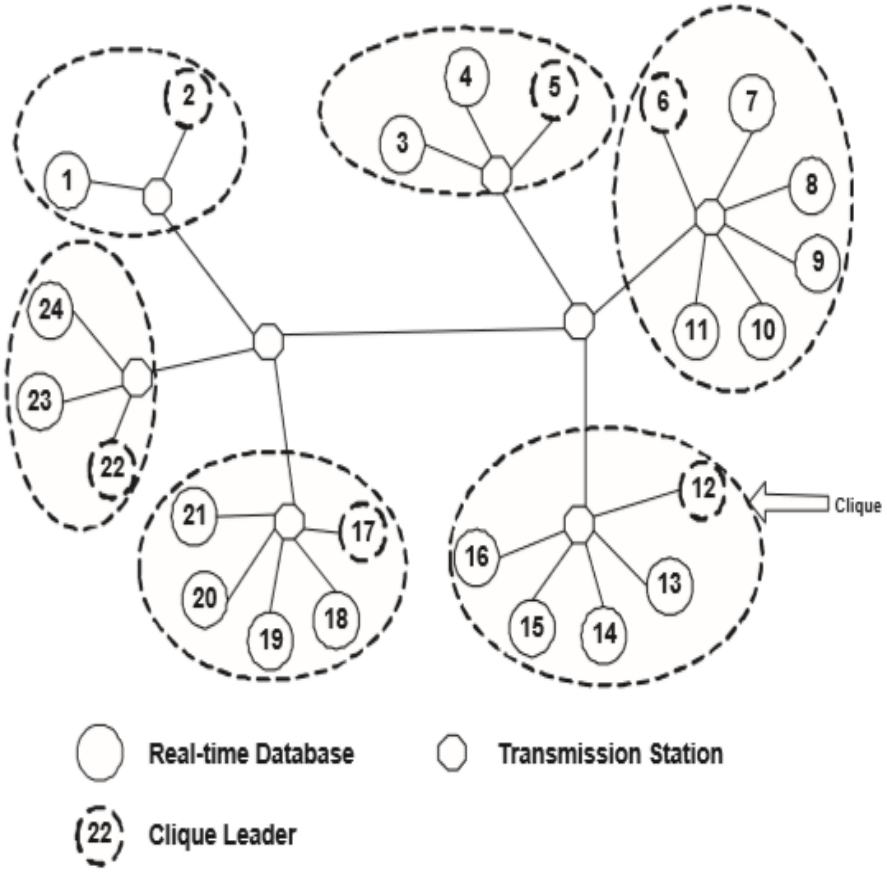

Figure3.2 Share the Replication ofImmerseScale in Distributed system

The replica of the one clique are share to the database server in that clique in the ORDER-RS algorithm.Each clique,the process of replication in the clique by managing a clique bellwether(leader).While the requirement of the temporal items data from the database server in another cliques,the clique bellwhether(leader) acts as proxy for the clique member and control the replica all of the data.While the clique member need the remote site of the temporal data items,firstly check where the requested items data resides.Within the same cliques of the requested items data resides to establish the process of replication, the clique reply the request items data to the member clique check has the request data items.The local clique bellwether(leader) receives the requested items data reside has in the different clique member of the database server.The local of clique bellwether checks where the data receiving the request,that has already the same data items of replica within the clique member.The replica highest frequency update data and all of the request within the cliques can share the replica to the clique bellwether transmutes.The requested frequency update data and duration by establishing an incipient replica of the clique

bellwether,when the requested items data in the remote site has not replica corresponding.

Transaction advent and departure are virtually the same,the evaluates the update frequency data and duration of the process transaction.Because of the directly handle by the themselves of the clique member within the cliques the replication of the two cliques member.The clique bellwethers handle in the different cliques members while the replication between server database.The clique bellwethers dynamically can be culled or definedstaticallyworkload at therun time consider on the system.Simple,in the paper,at the run time the clique bellwether considered and do not transmute.

3.4Existing Classifications

Replication strategies vastly vary, as all have different implementations. While every replication strategy is different, they may have common features with respect to certain aspects. Therefore, it is a sensible approach to classify replication strategies, as it helps building a coherent and organized foundation for studying them. In this section, existing classification schemes for dynamic replication are discussed.

3.4.1 Static Versus Dynamic Replication

In most general sense, replication strategies can be classified into two groups, namely static and dynamic replication. In static replication, all decisions regarding the replication strategy are made before the system is operational and not changed during operation. On the other hand, in dynamic replication, what, when, and where to replicate are decided as a response to the changing trends of the data grid. In a non-changing grid environment, where nodes do not join or leave the grid and file access patterns are not varied, static replication might be a good choice. Compared to dynamic replication, static replication does not have the overhead caused by replication decision and management.

On the other hand, when a replication scenario needs to be periodically reconfigured according to dynamic grid properties, it causes significant administrative overhead and affects scalability and optimal resource use of the system. In a dynamic environment where nodes are free to join or leave, and file access patterns change

over time, dynamic replication excels static replication due to its nature of adaptability [10].

3.4.2 Centralized Versus Decentralized Replication

Data replication involves many tasks, including but not limited to, choosing files to replicate, and deciding where to place replicas. Each task requires having a priori information about that particular state of the data grid environment. Which party or parties will collect this information, process it, and take actions regardingreplication is in the scope of this classification scheme. Centralized replication strategies contain a central authority to control all aspects of data replication. All metrics are either collected by or propagated to this central authority. Replication decisions are given by this point of control and all the other nodes report to it.

In contrast, decentralized approach encourages no central control mechanism to exist in the system. Nodes themselves decide on how replication will occur. With no central control, no single node can hold complete information about the entirety of the data grid. In a decentralized replication management strategy, coordination of a replication event is usually performed with the collaboration of a number of nodes. As the system scales up, the inter-node communication overhead should not increase to a point that surpasses the benefits of the replication.

Similarly, if a centralized replication management is chosen, the capabilities of the central replica manager should not cause bottlenecks if the system scales up in the future. Each approach has its advantages and drawbacks. Centralized replication is easier to implement and generally more efficient, as a single entity is responsible for all the decisions and has knowledge about every aspect of the data grid. On the other hand, central authority is also a point of failure and thus is not ideal for reliability and fault-tolerance [9]. Decentralized replication is good for reliability as there is no single point of failure in the system and the system can still behave predictably even a number of nodes are lost. However, having no central control and nodes acting on incomplete information about the state of the system may yield non-optimal results, e.g. excessive replication.

3.5Classification of Dynamic Replication Strategies

The existing dynamic replication strategies with respect to target data grid architecture. Each different data grid architecture has different properties, and these properties necessitate different strategies concerning data replication.

3.5.1 Dynamic Replication Strategies for Peer-To-Peer Architecture

In p2p architectures, nodes act in an independent manner. Nodes commonly possess enough capability to be both servers and clients at the equal time. This decentralized shape allows for even better volatility than different architectures, as nodes can connect to any part of the grid and go away without be aware. Replication techniques for p2p structure are advanced with the aid of retaining this fairly dynamic nature of p2p grids in mind [16].

Two dynamic replication techniques, course and requestor node placement method, and n-hop distance node placement method. Direction and requestor node placement method replicates files on all nodes at the route to the requestor node, consisting of the requestor node. In n-hop distance node placement strategy, the replicas are located on all acquaintances of the company nodewith a distance of n. In simulations, proposed techniques growth availability and reduce reaction time with the expense of the use of greater bandwidth [1].

Priori data replication approach for p2p data grid systems.Dynamic replication by using locating most efficient nodes to vicinity initial replicas earlier than the jobs are commenced. Maximizing the distance among identical replicas and minimizing distances among one of a kind replicas, they growth availability and ensure that every node has replicas of various record in its area. Of their simulations, a priori duplicate placement method is as compared with random preliminary reproduction placement and no preliminary replica placement. The proposed method improves job final touch instances and reduces record transfer times without growing bandwidth and storage costs [10].

3.5.2 Dynamic Replication Strategies for Hybrid Architecture

Hybrid information grid architectures commonly combine at the least two different architectures with special residences. For example, if a replication strategy is created for a sibling tree hybrid structure that combines p2p-like inter-sibling communique with hierarchical parenthood relationships, that particular method is studied in this subsection.

Hybrid replication strategy that combines the hierarchical structure with p2p functions. They implemented a value version and primarily based the replication selections on how the profits of the replication measure against the costs. A runtime issue constantly monitors the grid to gather crucial parameters. Reproduction size, and community status. Those facts are used within the calculation of the replication costs. The effects suggest that, common response time is advanced as replicas are positioned in the direction of the clients [2].

Two-way replication (TWR) that mixes a multi-tier architecture with p2p-like features. Within the target architecture, further to being linked to the discern node, every node (besides on the leaf level) is connected to its siblings as nicely. Replication decision is treated via a central authority, called grid replication scheduler (GRS) [7].

3.5.3DynamicReplicationStrategiesforGeneral Graph Architecture

Dynamic replication techniques which can be proposed for general graph architectures. In general graphs, nodes are freely connected. From a scalability point of view, those architectures are at an advantage because there is no strict challenge on the agency of the nodes. Scale-free, social network based totally data grid architectures, and other fashionable strategies that don't recognition on one specific architecture are classified on this subsection.The best replica is based on scheduling parameters. The scheduling parameters are bandwidth, load gauge, and computing capacity of the node. The scheduling in data grid helps in reducing the data access time [17].

Dynamic replication strategy, known as minimize facts lacking fee (mindmr).Mindmr measures and manages availability of the whole gadget. They introduce statistics availability metrics, system record lacking fee (sfmr) and system bytes lacking fee (sbmr). The former represents ratio of the lacking number of files to total documents requested by means of jobs and the latter represents the ratio of unavailable bytes to overall bytes requested by means of all jobs. With the objective of improving sfmr or sbmr, all documents are assigned weights by means of calculating the supply of the record, wide variety of expected destiny accesses, range of copies, and size of the record. The documents with lower weights are known as cold information and documents with higher weights are known as warm information. At some point of reproduction alternative, cold data are deleted first, and warm facts have the greater probability of replication. In overall performance assessment, mindmr accomplished better in phrases of job execution instances, sfmr, and sbmr as compared to different strategies [7]. Reproduction is created only under situations:

(i) while sufficient storage is available, or

(ii) duplicate to be created is extra critical than the replicas it's miles replacing.

The duplicate substitute decision is primarily based on a dynamic threshold that takes the range of requests, frequency of requests, size of the reproduction, and closing request time into consideration [10].

CHAPTER 4

DESIGN AND IMPLEMENTATION

The main target of this thesis is to analyze the effect of feature selection and the new trend of ORDER-RS algorithm(On-demand Real-time Decentralized Replication with Replica Sharing).The ORDER (On-demand Real-time Decentralized Replication) algorithm enhanced to ORDER-RS algorithm.In that work, full replication is used within the system.The algorithm decides where and how often the replicas are updated. The simulation results show that this type of on-demand replication algorithm can greatly improve the system performance. The proposed weather forecast data sharing system is developed on a wireless network. The weather forecast system has three main stations. The stations are Yangon station, Mandalay station and Naypyitaw Station.It is implemented on C# programming language with Microsoft Access 2010 database.

4.1 System Overview

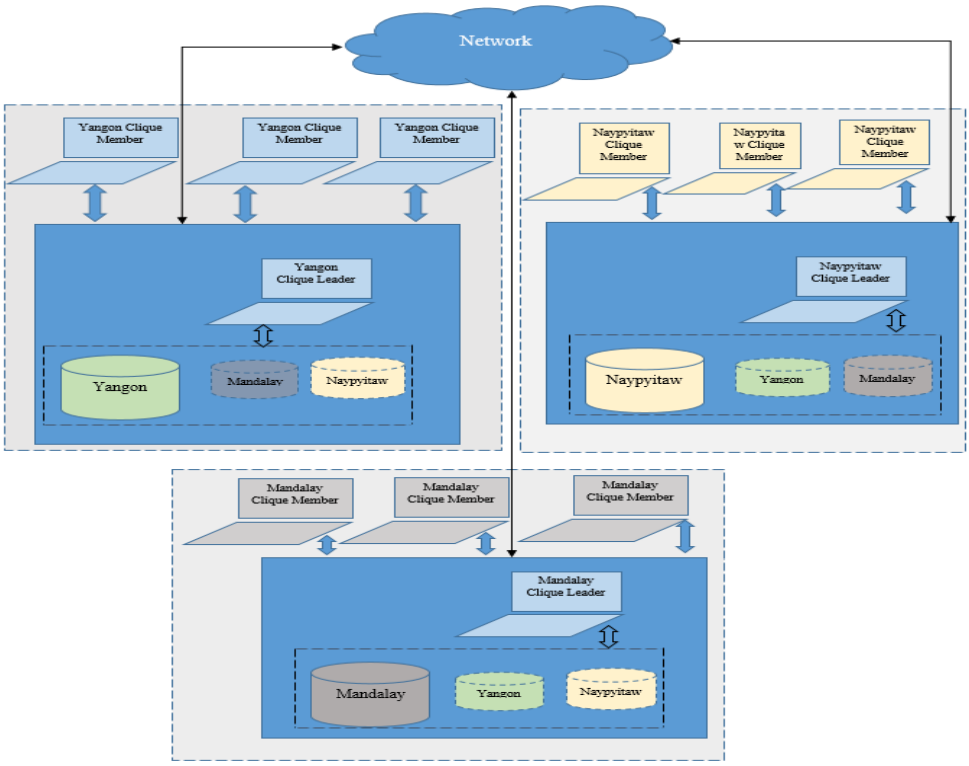

Figure 4.1 System Overview

The proposed weather forecasting system has three weather forecast stations (Yangon station, Mandalay station and Naypyitaw station) as shown in Figure 4.1.Each station is connected (Peer-to-Peer) to the remaining stations for the distributed data sharing. Every thirty minutes, each station uploads the updated weather forecast data to support the consistent weather information for other station.

4.2 Process Flow of the System

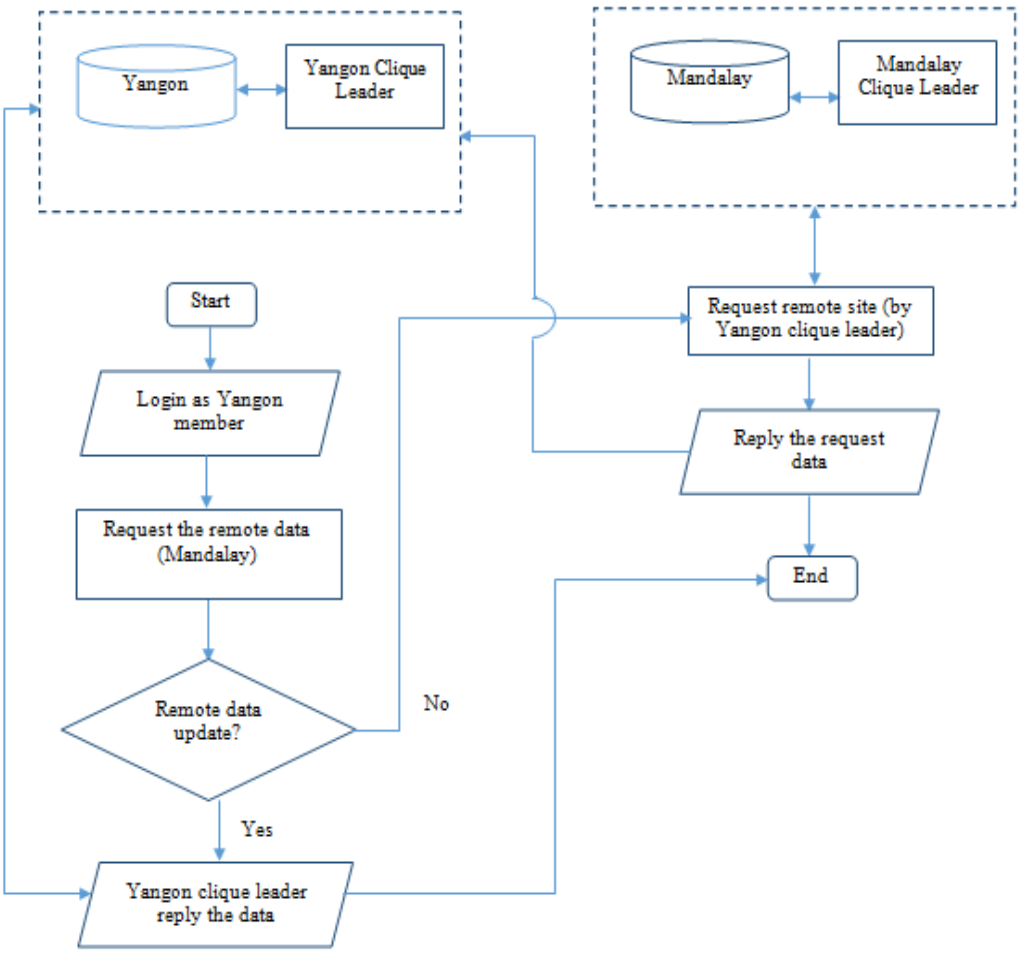

Figure4.2 The System Flow

The system flow is shown in Figure 4.2. When the user enters the system as Yangon station member and then the user wants to get the remote station data (Mandalay station's data), Yangon station checks the user requested data in local station. If Yangon station has Mandalay data, the system checks the Mandalay data in Yangon station is valid or not. If Yangon station has updated Mandalay data, Yangon station returns the result to user. If not, the ORDER-RS system will get the user

requested data from the nearest remote station and sends the result to user.By the use of ORDER-RS, the system maintains the consistent data update between each station and the data update in lowest latency.

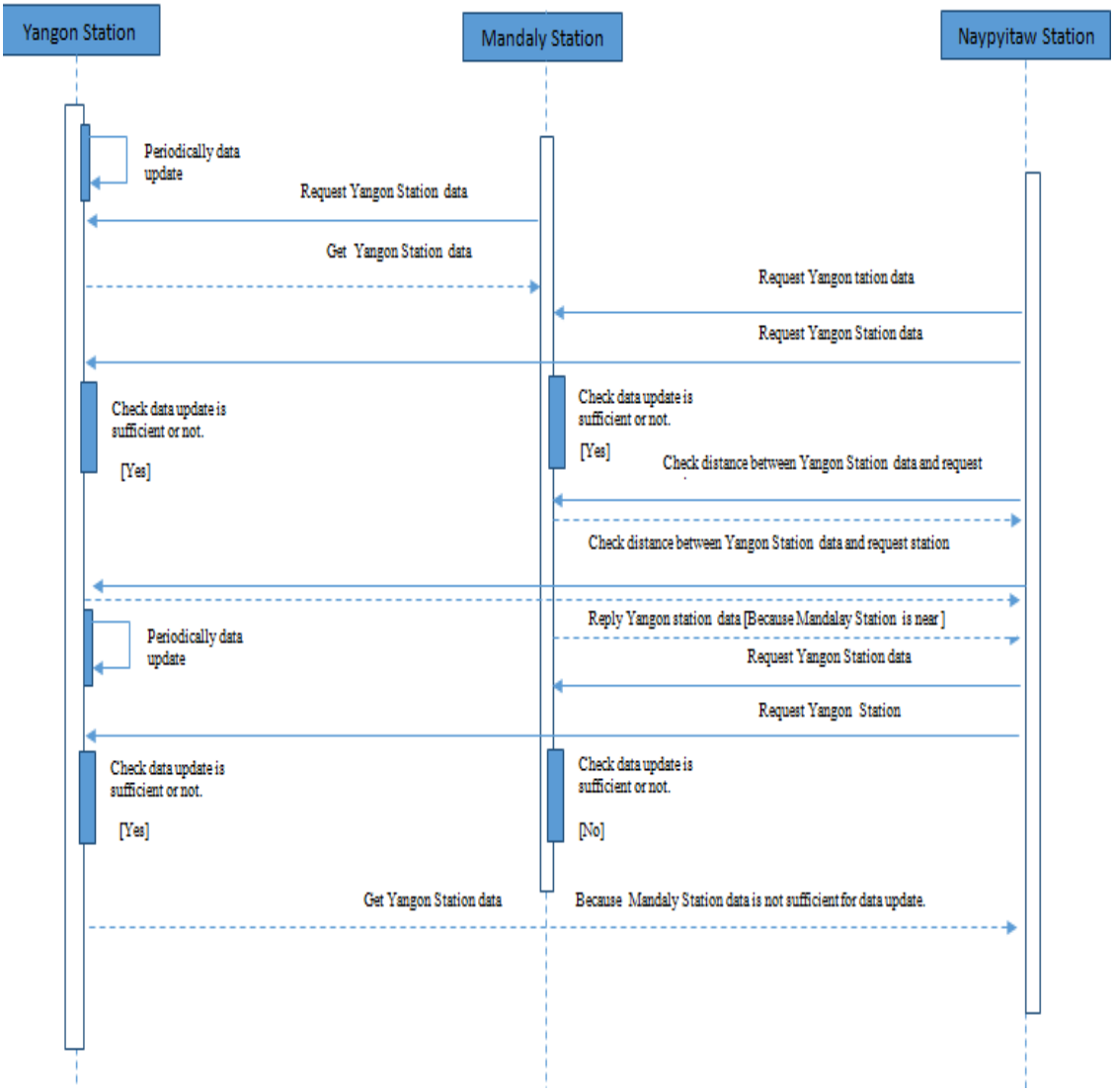

Figure 4.3 Sequence Diagram of the System

The detail operation steps are also shown in Figure 4.3. The sequence diagram has three stations: Yangon Station, Mandalay Station and Naypyitaw Station. Yangon Station periodically the data update. Mandalay Station request the Yangon Station data, Yangon Station check the requirement of data. Yangon Station data specify the requirement of Mandalay Station. So, Yangon Station reply the data to the Mandalay Station. Now, Mandalay Station has Yangon Station update data already. A new

transaction in Naypyitaw Station, the station want to Yangon Station data. Requests to Yangon Station and Mandalay Station check the requirement of data. Mandalay Station reply the data to the Naypyitaw Station because Mandalay Station has Yangon Station data already and more near than Yangon Station. After thirty minutes, Naypyitaw Station wants to Yangon Station data. In this case, Yangon and Mandalay Station check the requirement of data. In this time, Mandalay Station has replica of Yangon Station data, the data are not update. So, Naypyitaw Station gets data from Yangon Station.

4.3 Weather Forecast System of Database Design

The database design of each weather forecast stations are as shown in Figure 4.4. Each station has four data tables (Station table, WeatherInfo table, Distance table and User table). WeatherInfotable stores station information: StationID,FocusDate, Focus Time, Summary, Wind, Rain, MaxTemp, MinTemp, Location, WindDirection and Humidity.WeatherInfo table contains detail information of weather forecasting.User table store user information: UserID, UserName, Login, Password and Type. Station table store station information: StationID, StationName, Location, Dbaddress, Status, DistanceMileID. Distance table stores the distance miles between each station and this distance miles are used to determine to support the update data with lowest latency. User table stores the user information of each weather forecast station

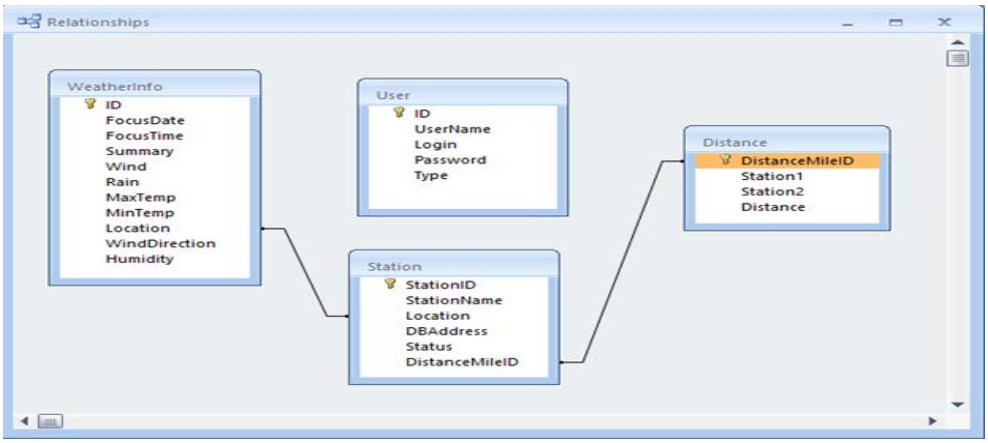

Figure 4.4 Weather Forecast systems of Database Design

4.4 Implementation of the System

The proposed weather forecast data sharing system is developed on a wireless network. The domain implementation is developed by C# programming language with Microsoft Access 2010 database. The system design and implementation is detail described in following.

4.4.1 The Login Page of System

Figure4.5 Login Page of Station

The login page of the station is shown in Figure 4.5. The proposed system has three main weather forecast data broadcasting station. So, the system user of the each station needs to enter the user name and password for system user authentication and additionally must choose their respective station name via the combo box.This system is developed for three weather forecast stations: Yangon Station, Mandalay Station and Naypyitaw Station. Although the Naypyitaw Station is situated between Yangon Station and Mandalay Station, Mandalay Station is near than Yangon Station for Naypyitaw Station.

4.4.2Main Page of Yangon Station

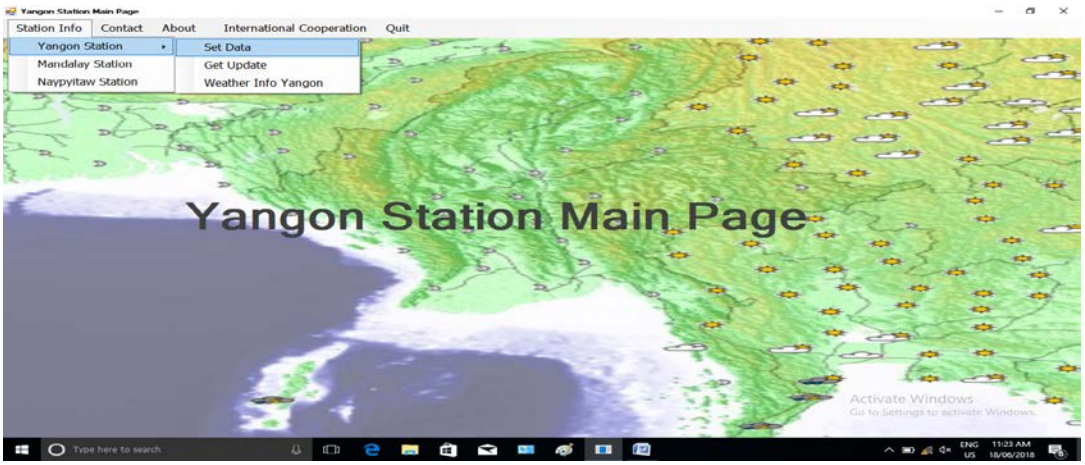

Figure 4.6 Main Page of Yangon Station

Figure 4.6,the main page of Yangon Station is composed of five main menus: “Station Info” menu, “Contact” menu, “About” menu, “International Cooperation” menu and “Quit” menu. Similarly, Yangon Station and Mandalay Station also consist of five main menus. “Station Info” menu has three sub menus for each station and these menus are capable for consistence data processing.

4.4.3 Getting Remote Station of Weather Information of Yangon Station

Weather Information Setting to base station (Yangon Station) is shown in Figure 4.7. In each station, base station data setting is done by every 30 minutes. In the data setting phase, all detail weather information of base station measurements (such as: Date, Time, Location, Summary, Wind, Rain, Max Temp, Min Temp, Wind Direction and Humidity) are submitted and uploaded by based station. Submitted weather information of each station is described in “WeatherInfo” page in figure 4.8. Getting Yangon Station’s date update from Other Stations are clearly described in following section.

Figure4.7 SetWeatherInfo Page

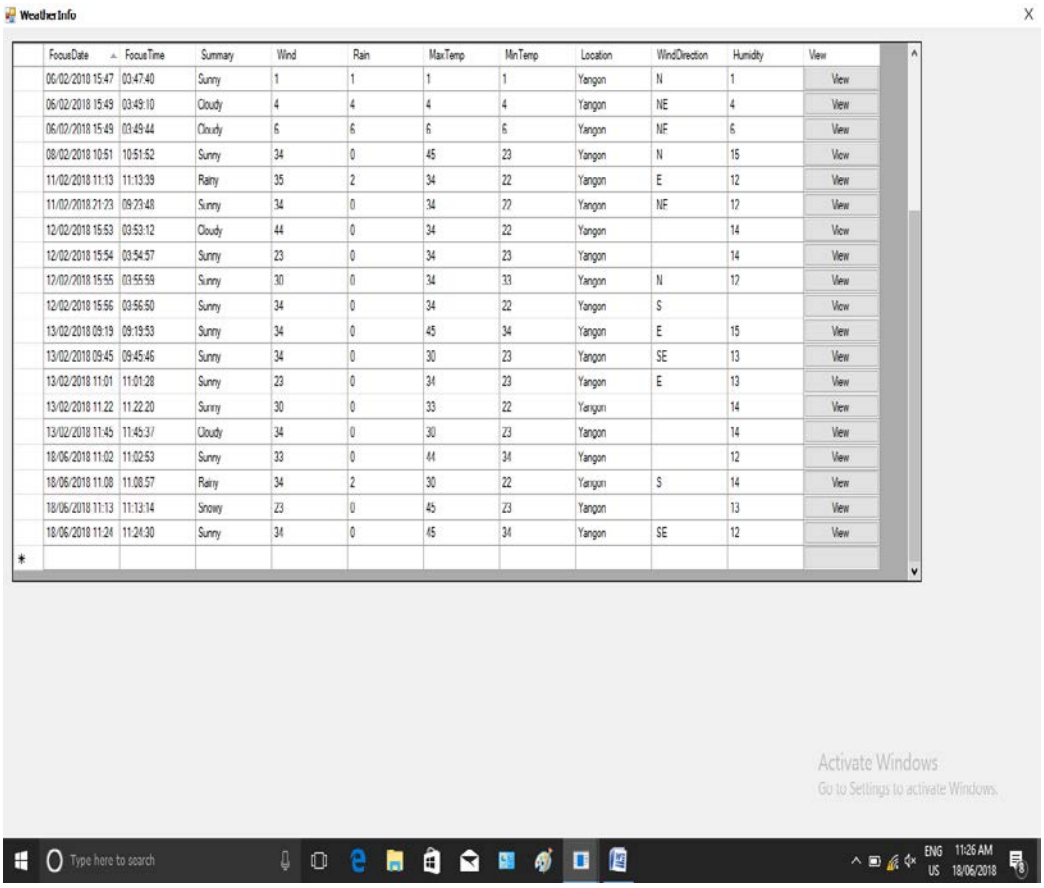

Figure4.8 WeatherInfoPage

4.4.4 Getting Remote Station of Weather Update Information

(At Mandalay Station)

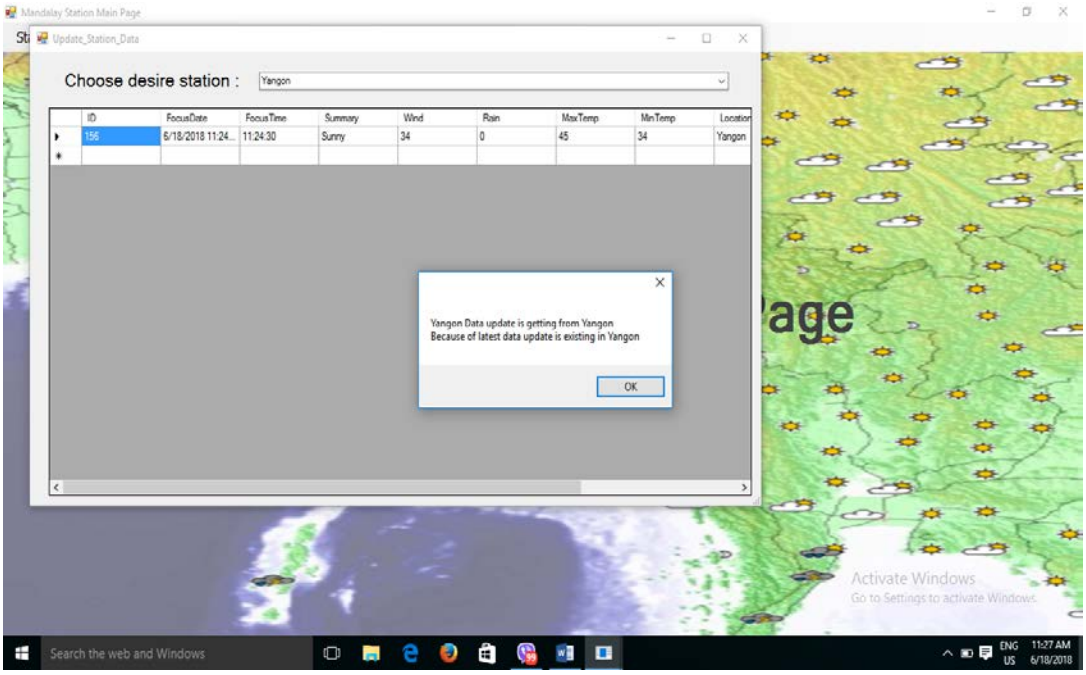

Figure4.9 Getting Yangon Station Data Update at Mandalay Station

"Update_Station_Data" page of Mandalay station has a combo box which contains remote station names (Yangon Station and Mandalay Station). When Mandalay Station chooses Yangon from combo box to get Yangon Station's update weather information, the system checks the nearest station's (Naypyitaw Station) data content. As the nearest station (Naypyitaw Station) does not has the Yangon Station's data update, the system gets the Yangon Station data update from source station (Yangon Station) and apply the data update at the Mandalay Station Figure 4.9.

4.4.5Getting Remote Station of Weather Update Information

(Getting Yangon Station Update Data from Naypyitaw Station)

When Naypyitaw Station chooses Yangon from combo box to get Yangon Station's update weather information, the system checks the nearest station's (Mandalay Station) data content. As the nearest station (Mandalay Station) has the Yangon Station's data update, the system gets the Yangon Station data update from

nearest station (Mandalay Station) and apply the data update at the Naypyitaw Station in the following Figure 4.10.

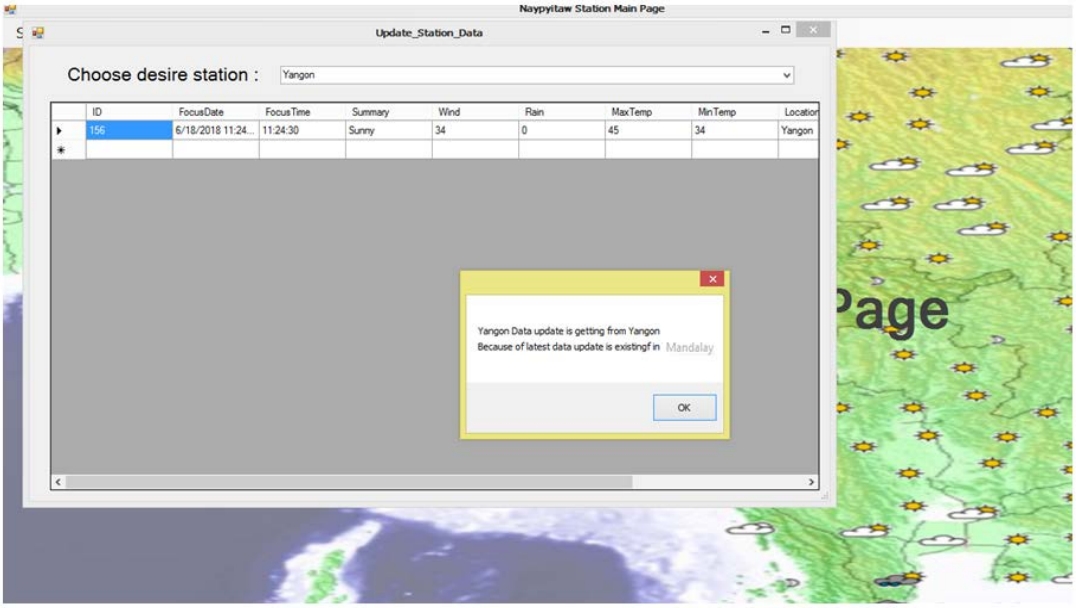

Figure4.10 Getting Remote Station's Data Update from Nearest Station

CHAPTER 5
CONCLUSION

5.1 Conclusion

In the distributed database environment many applications need authentic time of the accommodation data.Moreover,data accommodation is challenging task accessing delay of long remote data and requirement stringent time of the authentic time transactions.Replication can avail the requirement of application transaction stringent time.Weather forecast data is withal authentic time data and transmuting every time. Weather is highly volatile and must provide the most precise data for respective city. An astronomically immense network of weather stations distributed throughout the world coalesce with the felicitous replication algorithms. The propose on-demand weather forecast data sharing system is utilized for distribution, replication and data management by ORDER-RS algorithm.

5.2Advantages of using the system

In these fact, without replication cannot be distributed all of the system's performance not be accepted.Because of the remote data accessing form the high latency.And then,without replication the transaction executionand slack short time not meet deadlines in the system. Reduce the transactions workload of primary site due to the sharing of replication by using the ORDER-RS algorithm.So,the algorithm shows the better performance.More evident the effect of sharing replica get the workload higher.More replica can be shared,the reason is when the transaction's workload are higher.Addition advantage,the algorithm is the utilization network decremented dramatically.The utilization network of the ORDER algorithm is around moiety of the no replication algorithm.

5.3Limitation and Further Extension of the System

In ORDER-RS algorithms, the data need and durations by declaring the periodictransactions.On the requirement of the data,the replication of data

aredynamically engendered predicated.The freshness requisites of the incoming transcations to the frequency update data are dynamically adjusted. The proposed system shows,the ORDER-RS algorithm can greatly ameliorate.Can compare the system's performance without replication or the system with simple full replication strategy.Desirable to implement the replication control algorithms on the authentic time distribute database system and evaluate the workloads of the authentic transactions.Working system's prototype should improve currently.And additionally, in the distribute authentic time database into exploiting homogenous data attribute and imprecision the process of transaction.

This proposed system is mainly concern on data freshness with the minimal latency.So, this is not focus on the data crash or station data crash recovery.Add a development for the further extension,the system crash or database recovery technique should be controlled by the suitable recovery techniques.

REFERENCES

[1] Adelberg, B., Garcia-Molina, H., Kao, B.,"*Applying update streams in a soft real-time database system"*, Association for Computing Machinery Special Internet Group on Management of Datarecord, 1995.

[2] Andler, S., Hansson, J., Eriksson, J., Mellin, J., Berndtsson, M., Eftring, B,"*Towards a distributed and active real-time database systems*",15th International Conference on Distributed Computing Systems, 1996.

[3] Baulier, J., Bohannon, P., Gogate, S., Gupta, C., Haldar, S., Joshi, S., Khivesera, A., Korth, H., McIlroy, P., Miller, J., Narayan, P.P.S., Nemeth, M., Rastogi, R., Seshardi, S., Silberschatz, A., Sudarshan, S., Wilder, M., Wei, C., *"DataBlitz storage manager: Main memory database performance for critical applications"*,Association for Computing Machinery Special Internet Group on Management of Data Record 28, 1999.

[4] Cook, S., Pachl, J., Pressman, I, "*The optimal location of replicas in a network using a read-one-write-all policy"*,IEEE Internal Conference, 2002.

[5] ChanMya Aye,"*Online Stock Tracking System Using Active Replication"*,Master Thesis, University of Computer Studies,Taung-Ngu, 2010.

[6] Ericsson, P., "*An operational ship control system in a virtual environment*",Undersea Defense Technology Europe Conference,2003.

[7] Gray, J., Helland, P., O'Neil, P., Shasha, D.,"*The dangers of replication and a solution*", International Conference on Management of Data, Volume 25, 2 of Asscociation for Computing Machinery Special Internet Group on Management of Data Record, New York, 1996.

[8] Kemme, B., Alonso, G, "*A suite of database replication protocols based on group communication primitives*", 18thInternational Conference onDistributed Computing Systems Record 98, Amsterdam, Netherlands ,1998.

[9] Lee, V., Stankovic, J., Son, S., "*Intrusion detection in real-time database systems via time signatures*", 6thInternational Conferenceon Real-Time Technology and Applications Symposium, 2000.

[10] Mathiason, G., Andler, S.,"*Virtual full replication: Achieving scalability in distributed real-time main-memory systems*", 15thEuromicro Conference on Real-Time Systems,2003.

[11] MyaThidar, "*Update Propagation On Lazy and Eager Replication With Group and Master Replication Databases*",Master Thesis, University of Computer Studies, Yangon,2008.

[12] Nann Thin Thin New, "*Data Replication in Aircraft Components Database System using Distributed Database System*", December, Master Thesis, University of Computer Studies, Yangon,2010.

[13] Peddi, P., DiPippo, L.,"*Replication strategy for distributed real-time objectoriented databases*",5thInternational Symposium on Object-Oriented Real-Time Distributed Computing,2002.

[14] Shu, L.,Stankovic, J., Son, S.,"*Replicated data management in distributed database system"*, Special Internet Group on Management of Data Record,1988.

[15] Son, S., "*Supporting timeliness and security in real-time database systems*", 9thEuromicroConference on Real-Time Systems,1997.

[16] Wei, Y., Son, S., Stankovic, J., Kang, K., "*Qos management in replicated real-time databases*", 24thInternational Conference on Real-Time Systems Symposium,2003.

[17] Wolfson,O., Jajodia, S., Huang, Y., "*An adaptive data replication algorithm*",Association for Computing Machinery Conference Transactions on Database Systems, Volume 22,1997.

[18] Xiong, M., Ramamritham, K., Haritsa, J., Stankovic, J.,"Mirror a *stateconscious concurrency control protocol for replicated real-time databases"*, 5th IEEE International Conference of on Real-Time Technology and Applications Symposium ,1999.

[19] Yu, P., Wu, K., Lin, K., Son, S.,*"On real-time databases: concurrency control and scheduling"*, 14thIEEE International Conference 1994.