IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Fault Tolerance and Concurrency Control in Heterogeneous Distributed Database Systems

Sagar Patel[1], Meghna Burli[1], Nidhi Shah[1], Prof. (Mrs.) Vinaya Sawant[2]

Department of Information Technology, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India[1, 2]

Abstract: This project attempts to demonstrate the working of a Heterogeneous Distributed Database. It focuses primarily on fault tolerance and concurrency control in a distributed environment. Fragmentation, Replication and Query processing form the integral part of the program. The purpose of the software is that it allows the potential user to work in the heterogeneous environment. The potential user can access the database remotely. This software allows the user to fragment the data, to replicate the data and store it at different sites and lastly it enables user to fire a query from a remote pc and they can obtain the desired result set. It is a fault tolerant system which enhances its usability and acceptability in the database industry for applications in various sectors of the economy. The configuration file consists of metadata of the database and the list of IP address where the respective databases are stored.

Keywords: Distributed, fault tolerance, concurrency control, fragmentation, replication

I. INTRODUCTION

A distributed database is a database in which storage devices are not all attached to a common processing unit such as the CPU, controlled by a distributed database management system. It may be stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers. Unlike parallel systems, in which the processors are tightly coupled and constitute a single database system, a distributed database system consists of loosely coupled sites that share no physical components. System administrators can distribute collections of data (e.g. in a database) across multiple physical locations.

A distributed database can reside on network servers on the Internet, on corporate intranets or extranets, or on other company networks. Because they store data across multiple computers, distributed databases can improve performance at end-user worksites by allowing transactions to be processed on many machines, instead of being limited to one. In a heterogeneous distributed database, different sites may use different schema and software. Difference in schema is a major problem for query processing and transaction processing.

Sites may not be aware of each other and may provide only limited facilities for cooperation in transaction processing. In heterogeneous systems, different nodes may have different hardware & software and data structures at various nodes or locations are also incompatible. Different computers and operating systems, database applications or data models may be used at each of the locations. For example, one location may have the latest relational database management technology, while another location may store data using conventional files or old version of database management system.
Heterogeneous systems are usually used when individual sites use their own hardware and software. In this system, the users must be able to make requests in a database language at their local sites. Usually the SQL database language is used for this purpose.

If the hardware is different, then the translation is straightforward, in which computer codes and word-length is changed. The heterogeneous system is often not technically or economically feasible.

II. REVIEW OF LITERATURE

A. Existing System Study

A centralized database is a collection of information at a single location accessible from numerous points, in contrast with a distributed database where the information is spread out across multiple sites. There are advantages and disadvantages to this setup that can become considerations when people make decisions about how to configure databases. One advantage of the centralized database is the ability to access all the information in one location. Searches of the database can be fast because the search engine does not need to check multiple locations to return results. In a database upgrade to handle more information, servers can be added to the database site easily, and the company will not have to balance the needs of a distributed database.

Netezza is a Data warehousing appliance which can process your terabytes of data in glimpse. It is a focused appliance ideal for (but not limited to) departmental data warehouse solutions and for satellite data marts which extend an enterprise data warehouse to edge applications. Netezza has been successfully deployed in Automobile, Market research, banking sectors etc... Netezza's core value is to keep the things simple and accelerating high performance analysis of data to help clients uncover insights into their business. Technology plays a

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

transformative role helping organizations become smarter and more agile - ready to take the right action when most effective to do so. [1] Designed to deliver high performance, diverse queries, in-database analytics and sophisticated workload management, the Tera data Database supports and enables all Tera data Warehouse solutions.

With a Tera data Database driven warehouse, you can respond to changing and complex business requirements with greater speed and agility.[2]

III. PROPOSED SYSTEM

Designing software which implements a database which is distributed in nature and overcomes the drawbacks of the existing system which is usually used at enterprise level. We intend to merge the advantages of centralized database in the distributed architecture. The proposed system will increase the performance and will provide a user-friendly interface for the user.

The user is intended to give very basic inputs through the User interface. The given inputs will be then fed to the business logic in the background which will then produce the desired result.

With the inclusion of automated query processing feature in this software, we will try to make this as simple as possible so that people with even the basic technical knowledge can get the desired information they are looking for.

Some of the features that we will include in the software are as follows:

- Replication of available data on remote servers for greater availability of data.
- Fragmentation of data for greater speed and availability. Fragmentation of data depending on the user's choice i.e. Horizontal, Vertical or Mixed.
- Heterogeneous: A dynamic system to handle different types databases.
- Fault Tolerance: In case of a node failure, computation of final result in terms of consistency and speed will not be affected. Easy replication of data with only 2 copies of the fragment existing at any given point of time.
- Concurrency Control: Solution to the concurrency control problem, which can be bifurcated into two major sub problems: read-write and write-write synchronization using two phase locking, time stamping and multi-version timestamp.
- Security: Distributed databases compute by gathering data from different nodes, security of data is important to prevent from unauthorized access and malicious activity.
- Automated Query processing: User will be able to carry out computation even if the syntax of the SQL Query is unknown to them.

A. Proposed System Architecture

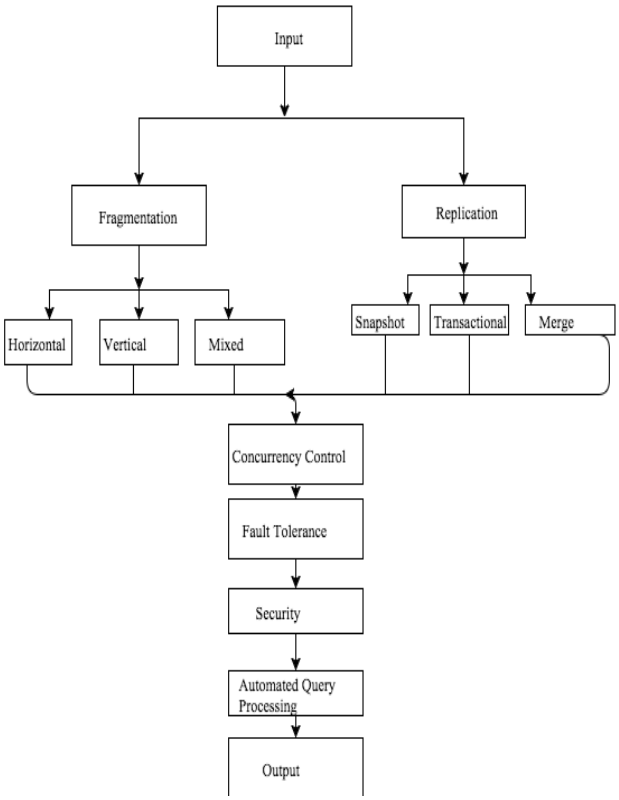

Figure A: Proposed Architecture

B. Expected Outcome

The software will initially provide four options namely, setup, fragmentation, replication and query processing. More than 1 computer needs to be connected to fulfill the distributed environment. On failure of any node, replication will be performed that will enable the software to provide 2 copies of the database at a time. The software also takes care of concurrent read and write of data. User friendly query processing will also be provided.

C. Benefits of Proposed System

This software overcomes most of the drawbacks of the existing system and can be used for heterogeneous environment.

Advantages over Centralized System:

- A big advantage of distributed DBMSs over centralized ones is that of scalability.
- Growth can be sustained more gracefully in a distributed system.
- Heterogeneous databases can be managed simultaneously.
- Greater availability of data through replication.
- Lesser storage space at site.
- Crash Recovery easy with no extra storage or backup.
- Fragmentation and Replication for databases as per user's needs.
- Transparency

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

- User friendly interface.
- High Performance standards.
- Less expensive infrastructure required.

IV IMPLEMENTATION

A. Module/ Component Description:
Module 1: Fragmentation
Fragmentation is the process of decomposing a database into multiple smaller units called Fragments, which are logically related and correct parts.
Characteristics of Fragmentation:

- Must be complete.
- Must be possible to reconstruct the original database

From the fragments

There are 2 types of Fragmentation
1. Vertical Fragmentation:
Each site or workstation may not need all the attributes of a relation. Thus we use Vertical Fragmentation which divides the relation vertically based on columns. It is a subset of a relation which is created by a subset of columns of that relation.

2. Horizontal Fragmentation:
It is a horizontal subset of a relation which contain those tuples which satisfy selection conditions.

Fragmentation is done to achieve the following:

- Reliability.
- Performance.
- Balanced storage capacity and costs.
- Communication costs.
- Security

Module 2: Concurrency Control:
Concurrency control is the activity of coordinating concurrent accesses to a database in a multiuser database management system (DBMS). Concurrency control permits users to access a database in a multi programmed fashion while preserving the illusion that each user is executing alone on a dedicated system. The main technical difficulty in attaining this goal is to prevent database updates performed by one user from interfering with database retrievals and updates performed by another. The concurrency control problem is exacerbated in a distributed DBMS (DDBMS) because [1].

1. Users may access data stored in many different computers in a distributed system.
2. A concurrency control mechanism at one computer cannot instantaneously know about interactions at other computers.

We shall be using Timestamp as a solution to prevent inconsistent data storage. Incorrect updation will not take place because of the use of timestamp. No user will be kept waiting for the updation to take place as they will be appropriately notified.

Module 3: Replication:
The term replication in a distributed database refers to the operation of copying and maintaining database objects in more than one location.
Replication is a cost effective way to increase availability and is used for both performance and fault tolerant purposes thereby introducing a constant trade-off between consistency and efficiency. Replication is the most appropriate way for travelling sales-people and roaming disconnected users and enables mobile users with laptops to be updated with current database information when they connect and to upload data to a server. Data is generated and then replicated. Replication is the process of copying data from a data store or file system to multiple computers to synchronize the data. Database replication is becoming more important for database applications. Replicated data are gaining more and more value as well as interest lately. Replication is desirable for at least, two reasons:

- Better performance.
- Better availability.

Module 4: Automated Query Processing:
To make this software as user friendly as possible, we are making a simple UI where users with very basic knowledge of the Database Languages can also use it to their benefit. Taking just specific user inputs, the software will process the query in the background and output the desired result. This includes a very user-friendly interface and proper query processing logic.

Module 5: Fault Tolerance:
Fault tolerance can be provided by maintaining multiple copies of data across the cluster of nodes. The degree of fault tolerance depends on the specified number of copies (with a minimum of two).
The following sections cover the possible failure cases. For this series of examples we will assume the following as shown in Figure 1

- A five-node cluster with separate front end and back end networks.
- A total of five slices of data with two replicas each. The primary replica is labeled with a letter (e.g. A), while the secondary replica is labeled with an apostrophe (e.g. A').

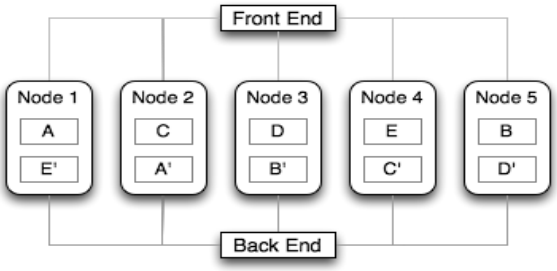

Figure 1: Basic Distributed System

Loss of a Node
Consider what happens when the cluster loses node #2 to some permanent hardware failure. Data on the failed node can no longer be accessed by the rest of the cluster.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

However, other nodes within the system have copies of that data. The Figure 2 demonstrates recovery from the node failure. Because our data distribution model is peer to peer and not master-slave:

- Multiple nodes participate in the recovery of a single failed node. This means that clusters with more nodes can recover faster from a node failure.
- Once the cluster makes new copies of slices A and C, we now have a fully protected system with two replicas of each slice. The cluster can now handle another failure without having to replace the failed node.

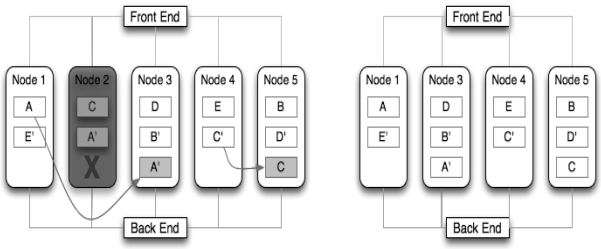

Figure 2: Handling a complete node failure. The failed state (left) and recovered state (right)

Module 6: Security:
There are two broad areas of security in distributed systems:

- Security Threats
- Authentication: Every user will be authenticated and password will be stored.

In traditional security model, all the data stored in database and the users who access that data belong to the same security level. A multilevel secure database system assigns security level to each transaction and data. Clearance level of a transaction is represented by security level assigned to it and the classification level of data is given by the classification level. A multilevel secure database management system (MLS/DBMS) restricts database operations based on the security levels.

B. Module wise algorithm / Pseudocode:
Main Module
This module is used to initialize the system. It has functions to check whether the system has all configuration files, cleans up temporary files, checks for drivers required. It sets the user directory so that system can perform its operations from anywhere.

Communication Module
It is the most important module of the system since it is used for communicating commands between host and is also used for transferring and receiving of files, arrays .It has functions to check whether other systems are up and ready on the network.

Setup Module
The user is supposed to enter the range of ip addresses and the application performs a search. If a node is found with the application then the user gets connected to it.

Fault Tolerance
On communicating with other nodes, the metadata file is checked for copies of the databases. The count is taken in order to ensure that a replicated copy of every database exists. A generated file i.e. metadata file gives the database that needs to be copied to the nearest node. The connected node is given the replicated file.

Fragmentation Module
This module is used for fragmentation of tables. The user has to choose the table that he wishes to fragment. Fragmentation of tables on basis of columns and rows i.e. horizontal and vertical fragmentation is provided. Only nodes supporting the table format are shown. The user is asked to input conditions for each of the fragments that he wishes to make.

Replication Module
This module is used for replication of databases. The user has to choose the database to be replicated and specify the IP addresses where he wishes to store the replicated copy. Only those IP addresses are shown that support the database to be replicated that is if it supports a sql database then nodes having sql are only shown. The replication and fragmentation module both have 2 phase roll back mechanism and handle all the file transfers of the databases/tables and the configuration files. During these processes the host is the server and is blocked to receive any request from any other nodes.

Query GUI
It is used to initialize the user interface of the query module. It is used for taking inputs from the user in the form of queries. It functions to initialize the database tree and the table tree. The database tree is initialized and shows all the databases and the table on the network. The table tree shows all the tables and their columns as one database on the network hiding the different databases the tables belong to. The query that is fired is converted to lower case and displays the result after the query has been executed. The GUI contains a dropbox that gives the keywords of the queries in SQL. On selecting any one of the keywords, the syntax of the same will be provided to make it easier for the user to perform queries on the databases.

Query Processing Module
The query processing module generates the following files:
a. Ts File
The ts file contains the details of the time when the query was fired on the application. It is then transferred to all the connected nodes.
****
Year
Month
Day
Hour
Minute
Second

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Millisecond
Read / Write
Name of the Database
****
2016
4
9
13
49
40
274
read
data
****

b. Ts final File
The ts final file contains the ts file of the other node that is sent from the other node on firing a query. This class is used for processing the queries inputted by the user. It has functions for different types of queries:
Functions

- Query_select()

Used for selecting data from different tables on the network. Accepts keywords as loc:, local:, org: just before the table name. The loc:, local: are used to specify to retrieve data from the local table if it exist else goes to the original table. The org: keyword is used to specify directly go to original table for retrieving data. The result is retrieved and stored in a result.txt file which is then shown to the user. The results are transferred in the form of files so that any developer can integrate our application into their application.

- Query_insert()

Used for inserting data into all instances of the table on the network .It modifies the query as per the columns present at each location of the table and then sends it to all the remote host for processing .It implements the two phase roll back algorithm. The queries are modified by referring to the cnffile.txt stored in system folder of the application.

- Query_delete

Used for deleting row/rows depending on the condition specified .If the table is horizontally fragmented then queries are not modified and are sent to remote host who have the tables. If the table is vertically fragmented then the query is sent to the original table to retrieve the primary keys affected by the query by converting the query to select query .The primary keys affected are retrieved and then appended to the delete query and the delete query is also modified depending on the columns at each location. The result is displayed to the user.

- Query_update

Used for updating row/rows depending on the condition specified .If the table is horizontally fragmented then queries are not modified and are sent to remote host who have the tables. If the table is vertically fragmented then the query is sent to the original table to retrieve the primary keys affected by the query by converting the query to select query .The primary keys affected are retrieved and then appended to the update query and the update query is also modified depending on the columns at each location. The result is displayed to the user.
All the above functions implement two phase roll back mechanism for data integrity.

Concurrency Control
On firing a query, the system compares the ts file with the ts final file to check whether there is any simultaneous change happening on the same database.
If the data matches, the user is given an error message to wait as simultaneous changes are made.

System function Module
This class has a wide range of functions which are used again and again for variety of purposes. It has functions for

1. File functions-Copy, Appending , deleting
2. Execution of queries.
3. Checking of fragment/replica tables.
4. Backing up of configuration files.
5. Deleting table from cnffile.txt, columns of a table, Initialization of table, adding background image to frames.

Initialization of table is used when a remote host request configuration files. It determines which tables are horizontally, vertically fragmented and which are original.
This class has static variables which are used for two phase roll back mechanism and these variables are used in communicate in communication module.

C. Working:
1. On running the main module, the four options that are seen:

- Setup
- Fragmentation
- Replication
- Query processing

Figure 1: On Startup

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

2. On Setup:

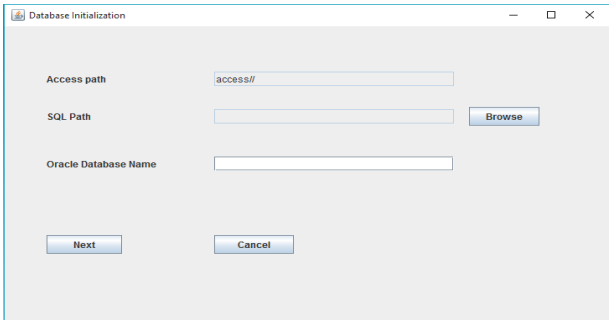

Figure 2.1: Setup Step

On putting in the path, all the databases on the local application are seen.

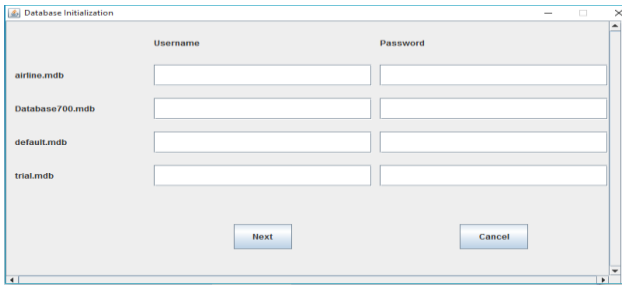

Figure 2.2: Initialize Database

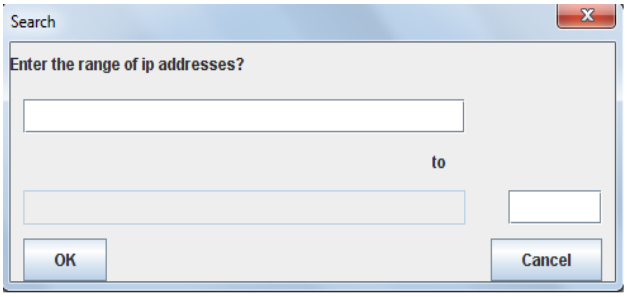

Figure 2.3: Search I.P.

3. Fault Tolerance

After communicating with the node, the theorem on fault tolerance is executed. The temp_Database620.mdb is the file that was generated through the fault tolerance module. The database 'Database620.mdb' was found on the near node and was the only copy so a replica of the same was generated. It is formed with a prefix 'temp_'.

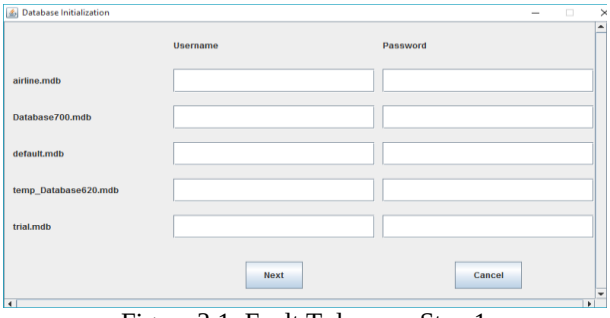

Figure 3.1: Fault Tolerance Step 1

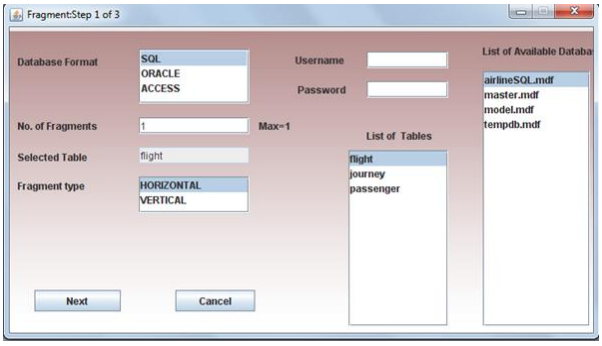

Figure 3.2: Fragmentation Step 2

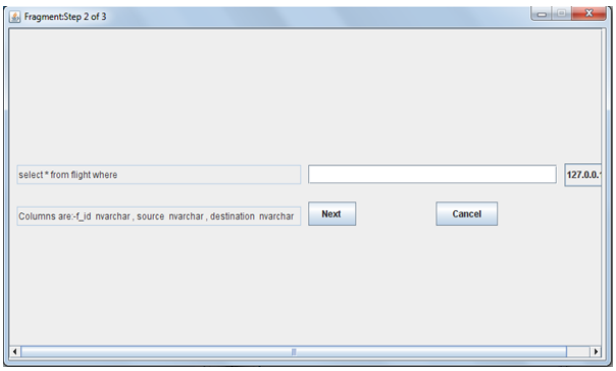

Figure 3.33: Fragmentation Step 3

The above images show the options to be selected for fragmentation. On specifying the IP address, the fragmentation process is executed and on the basis of the query, the fragment is made on the selected IP address.

4. Replication

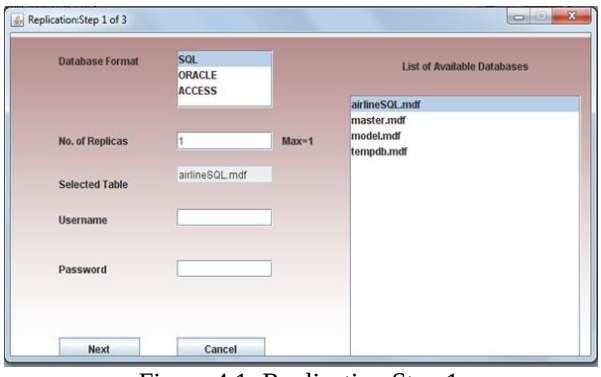

Figure 4.1: Replication Step 1

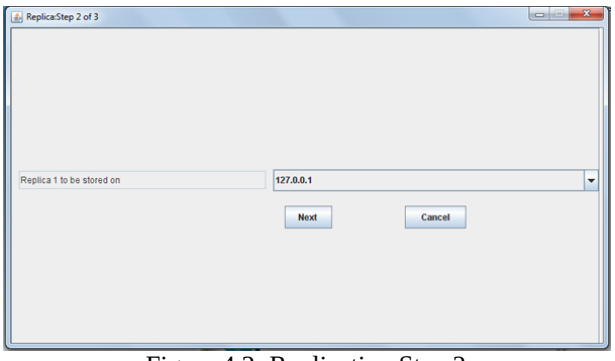

Figure 4.2: Replication Step 2

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

The above images show the options to be selected for replication. On specifying the IP address, the replication process is executed and a copy of the database selected is replicated on the selected IP address.

5. Query Processing

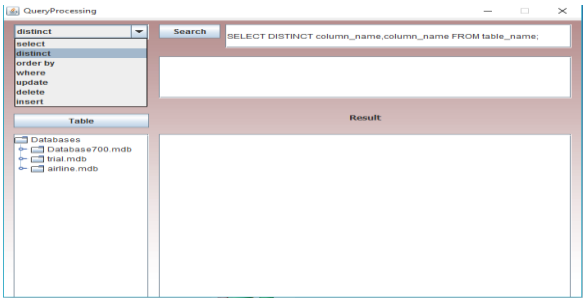

Figure 5.1: Query Processing Step 1

The user is provided with the appropriate syntax and this way any user who does not have knowledge of the SQL language can query the database.

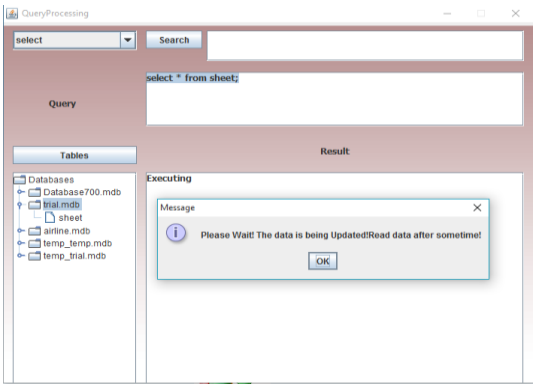

Figure 5.2: Query Processing Step 2

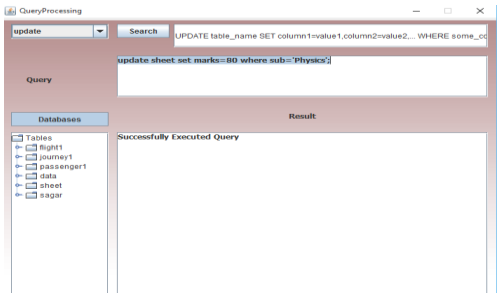

Figure 5.3: Concurrency Control on Node A

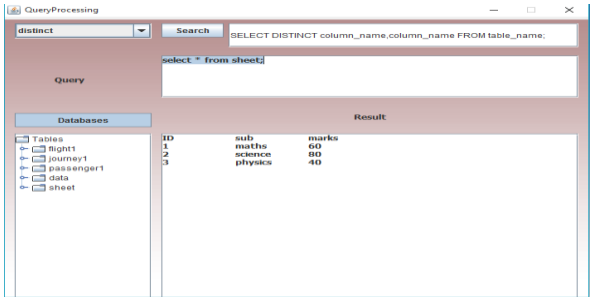

Figure 5.4: Concurrency Control on Node B

On executing the Query, the output is displayed.

While an update query is being written on Table A on Node A, if a read query is done on Table A on Node B, an appropriate message is displayed saying that a concurrent write is happening and hence the user has to wait.

REFERENCES

[1] http://netezzainformation.blogspot.in/2013/01/what-is-netezza.html
[2] http://in.teradata.com/products-and-services/TeradataDatabase/?LangType=16393&LangSelect=true
[3] P. A. Bernstein and N. Goodman, Concurrency Control in Distributed Database Systems, 1981.
[4] O. R. Liu Sheng, Database Allocation in Ethernet-based Local Area Networks: A Queueing Analytic Approach, IEEE, 1989.
[5] M. A. Adeboyejo and O. O. Adeosun, Fault Tolerance in Distributed Database Systems, IEEE, 2014.
[6] Thida, Fault Tolerance by Replication of Distributed Database in P2P System using Agent Approach, International Journal Of Computers, Issue 1, Volume 4, 2010.
[7] S. Wilbur, Distributed Systems Security, 2000.
[8] Harsh Mehta, Soham Mody, Dhrumil Joshi, Vinaya Sawant,"Methodology for implementation of heterogeneous distributed database system", ICTSM 2011, NMIMS, 25th-27th Feb 2011`