Airline Reservation System of Replication Method Approach

Phu Mon Kyaw, Khayea Aung
Computer University (Maubin)
phumonkyaw@googlemail.com,linnlatstar@gmail.com

Abstract

Nowadays, most of the organizations need to share their related data to provide better performance and availability of data for users located at any site. Because of enterprises' negotiation of sharing their databases, the related distributed databases become developed to support the satisfaction of users who can access any data from any site transparently. The proposed system is a client-server system that provides sending, receiving and updating transactions on distributed databases. Actually each database stands on as central publisher and when they want to access other database, they behave as client and when other database tries to initiate them, they are being as a server. There are three types of client-server system, one client, one server: many clients, one server: and one client, many servers. In this system, one client, many servers format, merge replication system is presented. The main motivation is to improve location transparency by accessing any distributed database from one site. This system evolves to implement an airline reservation system by using merge replication method based on distributed database system to achieve data availability, high throughput, data consistency and location transparency.

Keywords: distributed database, merge replication method.

1. Introduction

Nowadays, distributed database system is being used increasingly in between many large scale organizations. The term "distributed database" is a really virtual related objects whose component parts are physically stored in a number of distinct real database at a number of distinct sites.

The system to the total number of aircraft, the schedule of the flight and the information of the passengers in each specific flight and give all needed information to the system. Database can manages this process without braking database.

Most of the organizations these days need to share their related data to provide better performance and availability of data for users. Because of enterprises' negotiation of sharing their databases, the related distributed databases become developed to support the satisfaction of users who can access any data from any site transparently. That's why merge replication is popular. In a one-copy distributed database system, each data item is stored at exactly one site. In a replicated database system, some or all data items may be stored at multiple sites. The main purpose of distributing database is improved reliability by storing important data at multiple sites. However, in order for the system to remain constant, all the transactions that update the database need to be synchronized and disseminated at all the replicas. Obviously, if most of the transactions in the system are updates, a replicated system trades off performance for availability and fault tolerance.

2. Replication

Database replication has been traditionally used as a basic mechanism to increase the availability (by allowing fail-over configurations) and the performance (by eliminating the need to access remote sites) of distributed databases.

Database replication allows database administrators to distribute data to various servers throughout an organization. The reasons why replication methods are used for organization are:

Load balancing. The user can be used the replication mechanism to disseminate data to a number of servers and then distribute the query load among those servers.

Redundancy. Replication can allow the user to build a fail-over database server that's ready to pick up the processing load at a moment's notice.

Offline processing. The user may wish to manipulate data from the database on a machine that is only frequently connected to the network.

3. Distributed Database Management System Architecture

There are a number of ways of related database have been distributed. Two main methods are:

(1) client/server system

This methods concentrates on data management duties at servers which the clients focus on providing the application environment including the user interface. The communication duties are shared between the client machines and servers.

Client-server describes the relationship between two computer programs in which one program, the client program, makes a service request to another, the server program. Standard networked functions such as email

exchange, web access and database access, are based on the client-server model.

For example, a web browser is a client program at the user computer that may access information at any web server in the world. There are a number of different types of client/server architecture. The simplest is multiple clients – single server where the database is stored on only one machine which hosts the software to manage it.

A most sophisticated client/server architecture is one where there are multiple servers in the system (multiple clients – multiple servers). In this case, two alternative strategies are possible: either each client manages its own connection to the appropriate server or each client knows of only its home server which then communicates with other servers as required. In client/server system, failure at the server site makes the entire system inoperable.

(2) peer to peer (merge)

In this system, there is no distinction of client versus server machines. Each machine has full DBMS functionality and can communicate with other machines to execute queries and transactions.P2P systems are useful and successful for several reasons, including:

Ease of use: The server code is bundled with a user interface application to publish, search and retrieve content**.**

Ease of deployment: Each user installs a single package that encompasses both client and server code; its initial configuration depends only on knowing a fixed index server or a single other installation; servers need not be continuously active.

Location and replication transparencies: Location and replication transparencies are supported by the definition of the local and global conceptual schemas and the mapping in between.

Network transparency: Network transparency is supported by the definition of global conceptual schema.

Fault tolerance: Failure or unavailability of a single server does not disable the system. A failure at one site of a DDBMS or a failure of a communication link only makes some sites inaccessible.

For these reasons, this system user peer to peer (merge) architecture to implement the airline reservation system based on distributed database.

3.1 Database Replication Methods

There are three main components:

Publishers have data to offer to other servers. Any given replication scheme may have one or more publishers.

Subscribers are database servers that wish to receive updates from the Publisher when data is modified.

Distributors are the servers that manage the flow of data through the replication system. This server contains the distribution database.

3.2 Three types of Replication Method

There are three types of database system: snapshot replication, transaction replication and merge replication. They trade off between the notions of latency, autonomy and some of the other considerations.

3.2.1. Snapshot Replication Method

This method can be used when the subscribing servers can function in read-only environment, and also when the subscribing server can function for some time without updated data.

Snapshot replication works by reading the published database and creating files in the working folder on the distributor.

Style: Single Master Replication
Asynchronous Type
Advantage: High Latency
Disadvantages: (i) Data may be out-of-date because of batch replication
(ii)Centralized mainly on primary replica
(iii) Only Read Access Available

3.2.2 Transactional Replication Method

This method of replication adds to subscribers the ability to update old data but not to create new data. Because of being Single Master Replication, the responsibility of sharing updated data and creating new data is mainly centralized on main server.

Style: Single Master Replication
Synchronous Type
Advantage: (i) Data are up-to-date
(ii) Read/Write Access Available
(iii) Location Transparency
Disadvantages: Still centralized mainly on primary server to create more new data and to disseminate updated data.

3.2.3 Merge Replication Method (Peer to Peer)

In situations where two or more servers must share most important information equally, and where all sites must be able to update data, merge replication is the answer. In merge replication, each server accepts changes from the local users, and these changes are later then transmitted to all negotiated servers. We can use merge replication in situations that require updates to data at more than one server location. Conflict handling is a matter of design.

However, we can take greater control over the process and create custom triggers that handle conflict resolution.

Moreover, because of being Multi Master Replication Technique, each site may be either publisher or subscriber at any time.

As all are Peer to Peer type, no centralized on any server for spreading updated data and each server can create its own site maintaining Local Autonomy.

Style: Multi-Master Replication
Synchronous Type

Advantages:

(i) Data are up-to-date
(ii) No Centralized on any server
(iii) Read/Write access available on local or global database
(iv) Location transparency

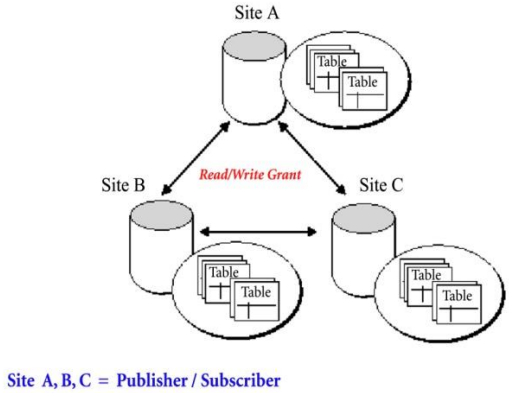

Figure 1. Merge Replication

In my system, uses merge replication methods. The snapshot replication method doesn't allow the updating data so it is only suitable for searching information and its style is single master and asynchronous. The transactional replication method centralized mainly on primary server to create more new data and to disseminate updated data. The transaction replication method is more powerful than snapshot replication method. But merge replication method are more useful than snapshot and transaction methods. At merge replication, two or more servers can share important information equally. All sites must be able to access updated data at any time. Each server accepts changes from local user and then transmitted to all negotiated servers immediately. There is no centralized server to handle the database.

So, my system is web based airline reservation system. In this system user need read/write access the database when user fulfills the reservation form. So the system database allows user read/write access. At two other methods, only transaction method access read/write available. But it has main centralized server. At the same time, many user can request the reservation form and fill. Like this time, only merge replication method can solve this situation. So in my system uses merge replication method any other than.

4. Overview of Proposed System Design

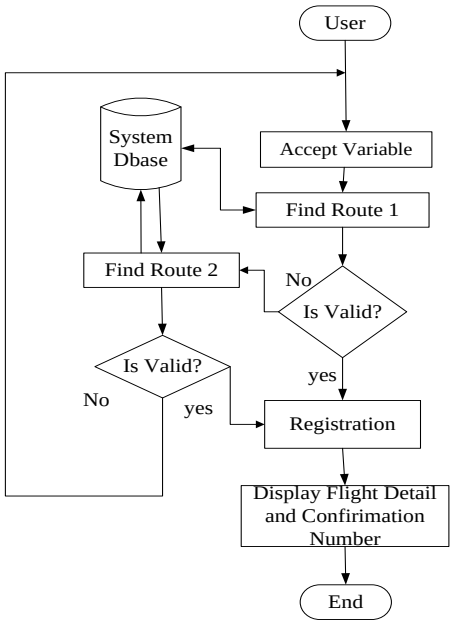

Figure 2. Proposed System Design

When user requests a reservation form by using web browser, server display form to user, user fulfills the reservation form and send it server. Server record the query and response display to user.

If user want to cancel the reservation, user will take canceling process and fill the confirmation number(s).The server update the database.

When the airports delay, the server sends the delay process to the user.

4.1. Database Design and Definitions

4.1.1. Plane type: This defines the physical type of the plane. It dictates the capacity of first, executive, business and economy seats that a *flight* can have.

4.1.2. Airport: An airport consists of a name, the city it is in, and its airport id.

4.1.3. Route: A route is simply of *airports: (Start Airport, End Airport),* and every route has a unique route id. A flight runs over a route only if it runs from the start airport to the end airport, possibly halting in between at other airports. A route is elementary for a flight if the flight runs nonstop from the start airport to the end airport.

4.1.4. Flight_Route_Ticket: A ticket is uniquely identified by a ticketid. The ticket may be a passenger ticket

or a cargo ticket, and can be booked under a passenger profile or a user profile. A ticket is booked *on a flight for a route* that the flight is associated with. *A passenger ticket* contains details about the passenger, and a *cargo ticket* about a cargo.

4.1.5. Passenger Profile: A profile denotes that a person has been verified to be genuine and can book tickets/ execute certain queries.

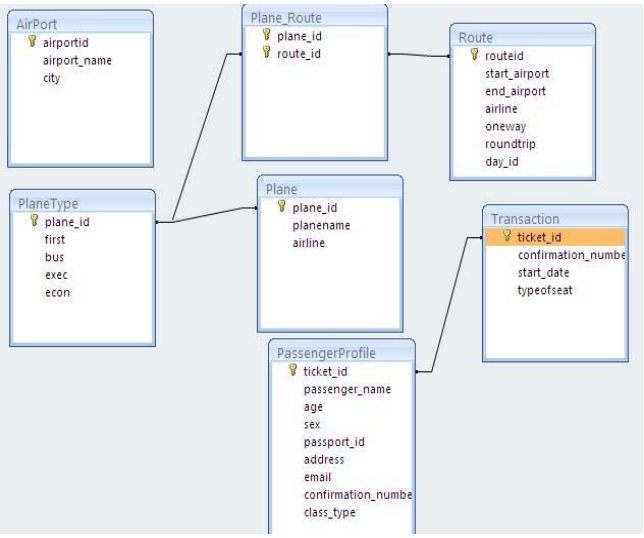

Figure 3. Database Design

4.2 Algorithm Description

In this paper, according to the merge replication method, Airline Site One can browse and check the Airline Site Two just for Route Table. It is the top row technique for nowadays economic process. But, other tables in each database are restricted to access and check from other site except Route Table. By doing like this, each web site can be one stop web site for ticket reservation. When end user requests the route that is not available in Airline Site One, he/she don't need to request other site and he/she can know other web site's information in the network.

5. Conclusion

Today, almost every organization needs to use distributed database to share their data for the purposes of co-operation, globalization, data availability, data maintenance, location transparency, etc. Merge replication appears as an elegant mechanism to achieve these goals. Most important, replication also must provide performance and scalability – characteristics which other replication methods often lack. Because of merge replication, the wish of organizations which want to expand their area is possible. But Merge Replication may lead to expensive costs because of communication overhead.

It will be necessary that everything concerning the database management should be put into consideration, and carried continuously, regaining with the rules sets in the performance, if we have to make a large numbers of access for the clients on any enterprise networks.

It will be able to construct a before and more secured network system, it we can perform monitoring on the system in details, such as the actual time duration spent on given access to the secured network.

In such a case, the above mentioned Web-Based application could be utilized conveniently and suitably without using too much time.

6. REFERENCES

[1] BETTINA KEMME and GUSTAVO ALONSO, Swiss Federal Institute of Technology (ETH), "A New Approach to Developing and Implementing Eager Database Replication Protocols", ACM Transactions on Database Systems, Vol. 25, No. 3, September 2000.

[2] Alexander Chigrik "Setting up Merge Replication: A step-by-step Guide", Database Journal, September 24, 2001.

[3] C.J Date, "An Introduction to Database Systems: seventh international edition.

[4] Goel, Sushant; buyya, Rajkumar; Data Replication Strategies in Wide Area Distributed System; Grid Computing and Distributed Systems (GRIDS) Laboratory; 2005.

[5] http://searchsqlserver.techtarget.com/ generic/0,295582,sid87_gci1103348,00.html(online document)

[6] Ozsu, Tamer.M, Valduriez, Patrick, "Principle of Distributed Database Systems", second edition, 1999. Prentice-Hall, Inc.

[7] Stanton, Jonathan; Amir, Yair; Danilov, Claudiu; Maslin-Amir, Michal; Tutu, Ciprian; On the performance of Wide-Area Synchronous Database Replication; November 18, 2002.

[8] Cardenas, Alfonso.F, "Database Management Systems", Second edition, 1984. Allyn and Bacon, Inc.

[9] Mailer,D, "Database Security, Integrity and Recovery", 2006. http:// www.hamiltion.bell.ac.uk. (online document)

[10] Ozau, Tamer.M; Valduriez, Patrick, "Principles of Distributed Database Systems", Second edition, 1999. Prentice Hall, Inc.