SNRU Journal of Science and Technology

Journal home page: snrujst.snru.ac.th

A Model Web-enable for OLTP with Replication Realtime Distributed Database

Klaokanlaya Silachan*, Pitiphol Pholpabu, Mongkol Rodjan

Department of Computer Technology, Faculty of Science and Technology, Nakhon Pathom Rajabhat University, Nakhon Pathom, 73000 Thailand

*Corresponding Author: klaokanlaya@gmail.com

Received: 8 January 2020; **Revised:** 15 August 2020; **Accepted:** 17 August 2020; **Available online:** 1 September 2020

Abstract

This research investigates the problem of distributed and update distributed database to be updated across all the site. Generally, a transaction database system distributes replication data with DBMS settings. The DBMS distributes the data to sites according to the distribution time settings. As a result, the data needs to be updated across all the sites. The larger the data set is, the bigger time consumption will take. Furthermore, the system becomes inconsistent and the data state maintenance becomes complicated. In this paper, a new model is presented to provide real-time distributed and update distributed replication to be updated across somesite and all sites by web code generation. The transaction database system can distribute data without DBMS. The study is conducted based on web-enable development for online shopping of the online transaction processing (OLTP), at the same time, distributed to other sites instantly. It results in keeping the information current and consistent when it is updated. The performance evaluation is based on the distributed and update efficiency which compared distribution response time of transaction connection access type on Linked server, remote access and direct access. The result shows that the module with linked server performs better than the other two.

Keywords: Web-enable; OLTP; Real-time distributed database

1. Introduction

The distributed database is a technology developed in order to store and access data with high accuracy, credibility, and reliability from more than one sources, locations, or sites. Users of this type of database can store and access data from anywhere via a computer network connection between database sites and the access terminal. Users may query any part of the database anytime without interfering with any other users [1, 2].

In terms of the distributed database, a replication is a process to copy a data object into a number of replicas on database sites in order to increase consistency, availability, and reliability of data. A replication can be ranged from a small replica with a single table to a big one containing a whole database. The replication process in distributed database leaves a main replica in the centralized database. Then, a number of replicas containing some particular relations are distributed to other database sites. Thus, users in each database site can access the data via the replica within their database site. Within these replicas, any change in any replica is updated to other replicas containing the same relation. The replication process provides a characteristic of replication transparency to the distributed database which data access is much faster with a number of copy replicas in different database sites. There are two main purposes in replication including 1) performance increasing as any closest location

of database can be used, and 2) multiple users support which several users can access the same data at the same time [3].

In the field of distributed database, there are a number of challenges including the difference types and procedures for transaction distribution. The transaction distribution procedures for each DBMS are designed differently. As a result, data distribution between different DBMS is not a simple matter. In distributing data with DBMS, it has the advantage of being able to set a time to distribute data from one database server to another database server. The disadvantage is that if there are additional distribution conditions, It has to create additional store procedures and execute them with DBMS. The administrator has to manage several steps. Furthermore, the cost of replication transparency, which provides a higher speed in transaction distribution, is a lot higher whenever new data is updated, especially for bigger data. It consumes longer update time and is very hard to maintain the same data in every site. Data replication is the process of storing data in more than one site or node. It is useful in improving the data availability for the replicate in consistently updated state. However, the overhead is increased on update operations as each site containing a replica which is needed to be updated in order to maintain consistency. The consistency of their databases are maintained by the distributed transactions [4]. In addition, there is a chance that some database sites with lower resources may become out of sync. In this case, other data which requires instant update or updates from the previous out of sync update could cause an update propagation problem. As the most important characteristic of distributed DBMS (DDBMS) is the data consistency, the latter problem can be seen as the most important issue of DDBMS. As a result, the present distributed DBMS development uses the real-time data distribution to fix the problem.

Real-time distributed database is one of an interesting issue for researching. In a development of real-time DDBMS, the DBMS requires real-time application support and real-time transaction execution which can be processed on every database sites with different time constrains. Nowadays, the real-time DDBMS is very important and is widely used in many fields, including, telecommunications, mobile communication, nuclear reactor control systems, traffic control systems, computer integrated manufacturing, robotics, military systems, and other related businesses [5 – 9]. Replication is the key factor to improve the data availability in real-time distribution systems. Replicated data is stored at multiple sites so that it can be accessed by the users. Real-time distributed transactions update data on two or more distinct nodes of a distributed database. A distributed transaction includes one or more statements that, individually or as a group, update data on two or more distinct nodes of a distributed database. When finishes executing its portion of a distributed and update, each real-time transaction is associated with a timing constraint, including, the time required for the commit processing and the remaining execution time of the transactions in running the transaction [10, 11].

Accordingly, in order to simplify DBMS setting procedures and ensure real-time transaction distribution with the online transactions, this research introduces a new model for distributed and update distribution which can distribute data and transaction in database system without DBMS on web-enable development. The data update and data distribution are done by a code generation on OLTP transaction database at the same time, distributed to other sites instantly which can be operated by either users or the implemented program. The case study in this research is a web application for online product ordering and regional product warehouse management. The data stores for OLTP are implemented and are used to support business process operations of an organization [8, 12]. The performance evaluation is based on runtime benchmark for distributed and update transaction in order to access local and global data. We measure the runtime variation of three connections which linked server in comparison with direct access and remote access.

The remainder of this paper is organized as the follows: section 1 discusses the existing researches which relate to our current research, section 2 presents methodology and related materials, section 3 covers the experimental studies and discussion, section 4 concludes the paper, and finally, suggestion is provided in section 5.

2. Materials and methods

Framework for Research

This research studies and experiments on data distribution and database update using code generation. The case study in this research is a web application for online product ordering and regional product warehouse management. Specifically, the case study company acts as a center receiving orders from customers. Then, the order transactions are first stored in the originating database site within the head quarter of the company. Afterwards, the data is distributed to regional product warehouses using horizontal distribution method. The data distribution of the order transaction is transmitted to a specific .database location based on the nearest location of a warehouse comparing to the customer delivery address. As a result, the number of available ordered products can be specified and further process the product delivery. In this case, the delivery can be very quick. In addition, the product list of each regional warehouse contains only the ordered product which will be delivered within the same region.

The analysis, design, and development of the system is aimed at developing OLTP commands in order to 1) set data access permissions of center host staffs, warehouse host staffs and customers, and 2) distribute and update data using distribution replication techniques. The purchase data is distributed to each site. Then, local staffs may manage the purchase transaction by themselves in order to reduce the process from the center host as shown in Fig. 1.

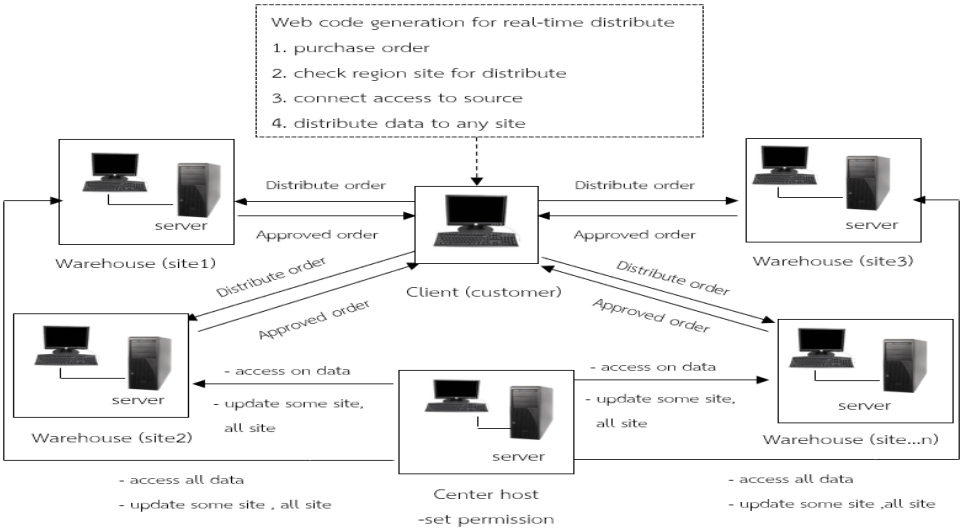

Fig. 1 A Model Framework for OLTP with Real-time Replication Distributed Database.

We analyze, design, and develop a procedure for a web application for online product ordering and regional product warehouse management and techniques for data distribution and update data as following:

Server connection host server

The data distribution between site is done by PHP commands via linked server access which can be controlled from a far using a computer network connection. The linked server access is proceeded by connecting to the hostname or IP address of the SQLserver for database server in the center site and other regional sites. The access permission is restricted by user accounts of each site. Each user account which belongs to either one of any warehouse or the center location can access the related data tables

depended on the permission of the user. There are three types of user account, including, center host staffs, warehouse host staffs, and customers.

Table 1 Site hostnames and addresses.

Host_user	Host_pw	Host_db	Host_site
Site1	827ccb0eea8a706c4c34	warehouse	C
Site2	93852ef2de19443046a9	warehouse	N
Site3	5e54fcc17b200f76b697	warehouse	S
Site4	9f6e6800cfae7749eb6c	warehouse	NE
Site5	ea416ed0759d46a8de58	warehouse	W
Site6	c1f68ec06b490b3ecb40	warehouse	E

Real time distribution module

In terms of real-time data distribution, the process depends on which site is needed data to be distributed to. In other words, when a customer makes an order, sets a delivery address, the real-time distribution replication process starts by distributing the order transaction to a specific regional location according to the delivery address, then, approve the order transaction on site. An order delivery document and an order receive document are created at the regional location, respectively. Afterwards, the center host staffs have to access the order and product from the center location to the all regional location and further deliver to the customer. To ensure success delivery, users have to specify the destination province and city in order to select a corresponding regional warehouse. Warehouse staffs can manage all order transaction within their location through the regional host without connecting to the center host as demonstrated in Table 2.

Table 2 Warehouses in different regions.

Region_id	Region_province	Region_district	Region_site
10510	Bangkok	KhlongSamWa	C
20000	ChonBuri	MeuangChonBuri	E
20150	ChonBuri	BangLamung	E
40000	KhonKaen	MeuangKhonKaen	NE

We can create a real-time distribution module as shown in Fig. 2 below:

```
Module1 :  Real-time distribution

Input   :  Order FROM customers (transaction)
Output :  distribute and approve orders
Description :   This module is a real-time distribution.
If a customer orders in any distributed server then order is replicated to region site
        BEGIN TRANSACTION
          REPEAT
                GET order FROM customer
                GET total price FROM order
                CHECK site on regional address for distribution order
                LINK server into regional warehouse from step2
         DISTRIBUTE transaction BY insert order record into regional warehouse
 APPROVED order by region site
        IF total price < xxxx
                APPROVED order
                do something
        ENDIF
        ELSE
                SEND order to CENTER FOR APPROVED
                SET status as waiting
                WHILE  status is waiting
                   Synchronize data;  // read data until status is set to approved
        ENDWHILE
        do something
END

UNTIL Committee
END
```

Fig. 2 Module 1: Real-time distribution

When the order transaction is confirmed by a customer, the data is distribute and insert in the region host. Then, the create-connection host module is executed to the regional site.

Distribution transaction update

In terms of basic information, including, products, prices, and other related data, the data distribution process is done to all regional locations. Conditionally, the data update can be done from the center host. Then, the system distributes the changes to all sites. Techniques for data distribution and data update are distributed and updated on all sites. The basic information, including, staffs, products, customers, and order lists can be updated here. The update procedure is done via an object using SQL commands.

Distribution update on some sites. Objects are created in the same number of host sites and connects to some host site. For example, purchase data on each site, customer data on each site, purchase transactions, etc. Then, the data in each site can be viewed and updated as shown in Fig. 3(a)

```
Module 2 : Updates distributed some sites
Input   : Order FROM customers (transaction)
Output : Update orders
Description : This module is provided for somesite manipulation.
              If customer orders in distributed server then order is updated , else
              order (transaction) will sent to update site.
      BEGIN
          REPEAT
          STORE the data for Update from table on center location
          CHECK data for update on regional site
          LINK server on regional_site_
             BEGIN
                CHECK for update data from table from condition
                        if condition true
                        Begin transaction
                           Update the data In table..1
                           Update the data In table..n
                      commit
                      else
                  end
        Until Link server on Region_site_N
        Until Update complete
      END
```

```
Module 3 : Updates distributed All-site
Input   : Dataset purchase order
Output : Set of distributed record S = { site , set of data }
Begin
 Repeat
        Update data from table on center location //
        Link server on Region_site1.....site N
        Begin transaction
            Check data for update from table from condition
               IF status Ok
                    Update the data In table..1 server#site1
                    Update the data In table.. n server_siteN
Until Commit
End
```

(a) (b)

Fig. 3 Module 2: (a) Global site distribution update and (b) All-site update

In order to distributed update on all-site, a sub-module for update replication from the center host is as shown in Fig. 3(b). The center host broadcasts data update, including, price per unit as the products on all sites should have the same price.

3. Results and Discussion

In this section, we discuss the result of our proposed approach based on the distribution process on the warehouse management systems.

Data Analysis and Design

From our analysis and design, we create several classes, relationship between classes and data for center and warehouse sites as shown in Fig. 4.

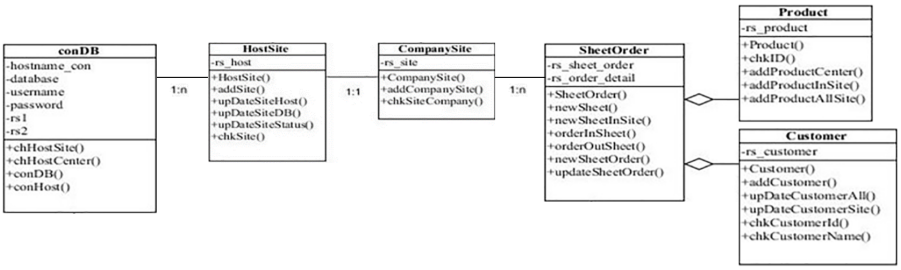

Fig. 4 Class diagram for center site and regional warehouses.

The system design, including the infrastructure of the database system and the data distributed system, are in the same pattern as the homogeneous distributed database system. Specifically, all sites are designed based on the same database structure and software. As a result, the procedures of the product stock counting, the delivery system, and the distributed system are simplified. So that, they can be easily and convenience used by all users.

Experiment Results

The distribution process starts when a customer confirms the order but the product distribution is done by staffs. The production distribution towards regional warehouses allows the delivery status checking on the regional sites. Accordingly, we conduct a data distribution test via websites for order transaction distribution and order receive document distribution as shows in Table 3 – 5.

Table 3 Warehouses and URL.

No.	Warehouse	Website	User
1	North (N)	https://northsite.co.th	Staff
2	South (S)	https://southernsite.co.th	Staff
3	West (W)	https://westsite.co.th	Staff
4	East (E)	https://eastsite.co.th	Staff
5	Northeast (NE)	https://northeastsite.co.th	Staff
6	Center (C)	https://centralsite.co.th	Customer, Staff

Table 4 Data distribution test via websites.

No.	Test format	Center	Regional
1	Product data distribution	☑	
2	Staff data distribution	☑	
3	Customer data distribution	☑	
4	Product receive data distribution	☑	
5	Order transaction distribution	☑	
6	Product check list	☑	☑
7	Product receive check list	☑	☑
8	Order check list	☑	☑
9	Delivery receipt generation	☑	☑

Table 5 Distribution of order transactions.

No.	Order (No. of products)	Warehouse	Result
1	0531 (1)	North (N)	✓Correct
2	0648 (3)	South (S)	✓ Correct
3	0791 (2)	West (W)	✓Correct
4	0854 (5)	East (E)	✓ Correct
5	0956 (3)	Northeast (NE)	✓ Correct
.	.	.	.
.	.	.	.
N(150)	xxxx (x)	X	✓ Correct

From Table 3 – 5, the system is tested by ordering products via center and regional websites. Then, staffs distribute basic information, send and receive the production, and check the product data via the website. The test results show that the system can distribute data modification to all sites correctly.

Evaluations

We implement the real- time distributed system for the linked server access. The performance evaluation is done based on the distributed and update efficiency comparing to the connection access type with transaction on linked server, remote access and direct access. The distributed and update transaction process provides response time which are gathered and updated to two table, including, committed time distributed table and committed time update table. When the process transaction is executed, we collect the start time and the time end of the process. The server used in the experiment is operated under Window 2010 with 3.0 GHz CPU and 1 GB RAM. The RDBMS is MS SQL Server connected with 28-Kbps link.

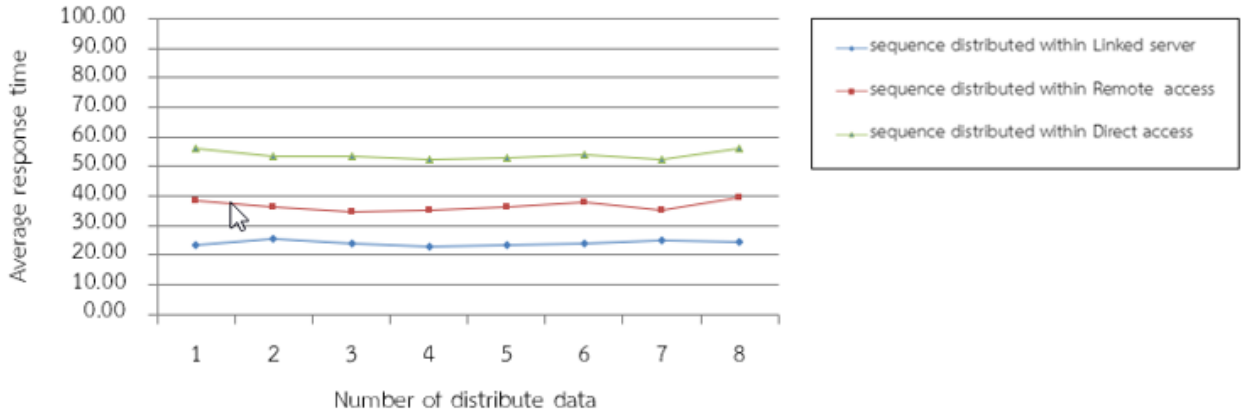

Fig. 5 Comparison in distribute response time for three connection types

The result of the experiment shows the distributed replication is executed for each purchase transaction. The distribution response time for each connection access type in a similar resource is not different in a big margin. Furthermore, the result also indicates that the connection speed of the linked server is better than the direct access and remote access transaction.

In comparison, several works have proposed the studies on other distributed database systems. Chueh et al. [13] propose a computer-based health-care record system accessible from remote sites using a transaction-based central database server based on controlled vocabularies on local databases. Diana et al. [14] develop a real time warehouse management system for industrial works which is focused on the design of WMS and integrated together with RFID technology in a cosmetics and

pharmaceutical warehouse. In addition, they focus on tools capturing real-time data, simulation models and optimization models. The system provides the best decisions according to system input in order to be allocate items and officiously move them around the warehouses. Singh et al. [15] emphasize different forms of real-time distribution transactions. A transaction management concept is introduced to manage database size in original site and remote sites rather than in computing parameters. In contrast, this research focuses on the study of the current data distribution and update. According to the test results, it can be seen that the developed OLTP web application can consistently distribute and update real-time data, including basic information and product orders, on all relevant sites. Moreover, the distribution and updates are done immediately without a DBMS administrator.

4. Conclusion

This research proposes a model of the web-based application for real-time data distribution and update using code generation, instead of DBMS settings. In terms of the distribution and update, the proposed model reduces the time consumption and the cost of data distribution and update. The model can rapidly and correctly distribute data and transaction to other site. The data management can be performed from the center site. The model consists of three modules, including real-time distribution module, some-site update distribution module, and all-site update module. The performance evaluation is based on the distributed and update efficiency which compared distribution response time of transaction connection access type on Linked server, remote access and direct access. The result shows that the module with linked server performs better than the other two. Furthermore, real-time distribution model increase performance in access and data distribution.

5. Suggestions

This suggest that real-time data distribution requires preparation on network system, connection types, and security system, such as, backup system on each site. Accordingly, when there is a problem on any site, the whole system can still process as normal. As for the future works, the query access replication is considered to be the next target in order to search through the needs of every site. Then, the other connection access type can be investigate along with the resource installation procedure which may affect to the overall distribution response time. Furthermore, the user authentication, user authorization, data encryption and data decryption should be provided on every site for a better data security.

6. References

[1] U.H. Waqar, Transaction processing in real-time database systems, Ph.D. Computer Science, Iowa State University, Iowa, 1993.

[2] C. Ulusoy, Processing Real-Time Transactions in a Replicated Database System, J. Dist. and P. Data. 2(1994), 405 – 436.

[3] S. R. Choudhary, C.K. Jha, A Systematic Study of Real time Distributed Systems in the Context of Distributed DBMS, IJRTER. 3(5) (2017) 134 – 148.

[4] A.K. Gupta1, V. Swaroop. Overload Handling in Replicated Real Time Distributed Databases, IJAER. 13(2018) 13969 – 13977.

[5] A.A.C. Fauzi, A. Noraziah, W.M.B.W. Mohd, A. Amer, T. Herawan, Managing Fragmented Database Replication for Mygrants Using Binary Vote Assignment on Cloud Quorum, J. App. Mech. and Mat. 490 – 491(2014) 1342 – 1346.

[6] F.A. Abdalhalim, A Survey on Transactions Management in Distributed Real Time Database Systems, IJACT. 4(6) (2015) 100 – 104.

[7] M.S. Rana, M.K. Sohel, M. S. Arman, Distributed Database Problems, Approaches and Solutions -A Study, IJMLC. 8 (5) (2018) 472 – 476.
[8] R. Salem, S. Saleh, H. A. Kader, Intelligent Replication for Distributed Active Real Time Databases Systems, IAJIT. 15(2018) 505 – 513.
[9] K.K. Ezechiel, S. Kant, R. Agarwal, A Systematic Review on Distributed Databases Systems and Their Techniques, JTAIT. 96 (2019) 236 – 266.
[10] A.K. Gupta, V. Swaroop. Overload Handling in Replicated Real Time Distributed Databases, IJAER. 13(18) (2018), 13969 – 13977.
[11] A. Srivastava, S.P.S. Kashyap. Issues in Distributed Real Time Replication Database System, The Opinion. 4(8) (2015) 11 – 20.
[12] S.R. Choudhary, C.K. Jha, A Study on Transaction Processing for Real Time Database System in Distributed Environment, IJCSIT. 4(5) (2013) 721 – 724.
[13] H.C. Chueh, G.O.Barnett, Client-server, Distributed Database Strategies in a Health-care Record System for a Homeless Population, JAMIA. 1(2) (1994) 186 – 198.
[14] R.R. Diana, R.P. Luis, J.B. Miguel, C.B. Lauren, M. Erik, The design of a real-time Warehouse Management System that integrates simulation and optimization models with RFID technology, TIJCAS. 2(04) (2013) 18 – 37.
[15] Y.J. Singh, Y.S. Singh, A. Gaikwad, S.C. Mehrotra, Dynamic management of transactions in distributed real-time processing system, IJDMS. 2(2) (2010) 161 – 170.