Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TWO-WAYS DATABASE SYNCHRONIZATION IN HOMOGENEOUS DBMS USING AUDIT LOG APPROACH

[1]RAI GUDAKESA, [2]I MADE SUKARSA, [3]I GUSTI MADE ARYA SASMITA

[1]*Under Graduate Student, Departement of Information Technology, Udayana University, Bali, Indonesia*
[2,3]*Lecturer, Departement of Information Technology, Udayana University, Bali, Indonesia*

E-mail: [1]raigudakesa@gmail.com, [2]e_arsa@yahoo.com, [3]Aryasasmita@yahoo.com

ABSTRACT

Data integration is the most important part in applying distributed database, in which data from various data source can be united by implementing integration. Data replication is one of the data integration forms which is very popular nowadays, since it is convenient or it can do backup towards data in various different sites. The practice absolutely has shortcomings in data integration, as there is no control over the integration from replicated data, therefore it is necessary to do synchronization towards data. Data synchronization is a part of replication, it is a process to ensure each copy of database contains similar object and data. Data synchronization can be applied in numerous methods, one of them is utilizing Audit Log that is recorded every activities occur at database. Audit Log can be applied to almost any Database Management System (DBMS). This research utilized TCP Socket and client-server architecture to distribute data from Audit Log. The final result concludes that Audit Log can be utilized in synchronization with client-server implementation, yet it has limitation in recording data. This paper also showing how Audit Log created and managed to be used as replication and synchronization procedure.

Keywords: *Distributed Database*, *Data Integration*, *Data Replication*, *Data Synchronization*, *Audit Log*

1. INTRODUCTION

Distributed database system has some advantages, for instance, the ability to handle expansion (enhancement or widening) data and data availability, also the ability to manage where to distribute the data. Replication in distributed database is one useful way to distribute that data. To create fully consistent data while distribution process, it is not adequate by only utilizing replication process. To overcome that problem, it then uses data synchronization procedure. Data synchronization is a part of replication, it is a process to ensure each copy of database contains similar object and data [1]. Synchronization process enables a data at certain database updated in real-time or periodic at the other database [2].

The other fundamental thing in applying synchronization is by understanding how data of the replication final result can be accessed and utilized before moving to synchronization phase. Some DBMS, for example MySQL, replication can be done by manipulating existing replication feature and reading binary log file. Meanwhile, not every DBMS has similar feature in applying replication. To satisfy that condition, the utilization of *Audit Log* and *Audit Trail* can be used. *Audit Log* can be utilized to identify who accessed database, what kind of activities, and what data has been changed [3]. General category of auditing database includes monitoring user who attempts to access the database, Data Control Language (DCL) activity, Data Definition Language (DDL) activity, and Data Manipulation Language (DML) activity [4].

In this scenario, an audit table will be made in every configured database and data in audit table will be distributed through network to targeted database which also has been configured. Transferring the data will be done through Socket with client-server model.

2. SYNCHRONIZATION METHOD

According to the explanation previously, one possible method in applying synchronization is utilizing data produced by audit log. Each DBMS has different way to make audit log. The easy way to create audit log is by using Trigger in DBMS, in order to produce an audit table in every database.

2.1 Creating Audit Log Table

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Creating audit log with trigger needs all types of trigger action, such as INSERT, UPDATE and DELETE trigger. Trigger auditing has one weakness, which is not all activities of a database could be recorded, just as Data Definition Language (DDL) activity and user activities. Besides, trigger auditing also has virtues in applying at almost all DBMS.

Every synchronized configured database with this method will have one new table with auditlog_ prefix, and then followed by database name. Table 1 will show the field of the audit table.

Table 1: Field of Audit Log Table

Field	Explanation
execution_id	This *Field* is used to save log id
timestamp	This *Field* is used to save time of input data to *audit* table
table_name	This *Field* is used to record table name which has been changed
action	This *Field* is used to record type of alteration (Insert, Update or Delete)
field	This *Field* is used to record any columns in that current table.
field_pk	This *Field* is used to record *primary key* owned by current table
field_type	This *Field* is used to record data types of field owned by current table
old_data	This *Field* is used to save data before the alteration of current table. The record is valid for *query* Update and Delete, while the rest will result NULL
new_data	This *Field* is used to save new input data to current table. Recording DELETE statement will result NULL value

Every table on database will be divided into 3 triggers, each consists of AFTER INSERT, AFTER UPDATE, and AFTER DELETE, thus if there are 3 tables on database, there will be 9 triggers on that database. The utilization of AFTER aims to prevent data unsuccessfully gets into audit table. Data input which gets into audit table is data produced by recorded activities of trigger. Figure 1 shows how audit trigger is created.

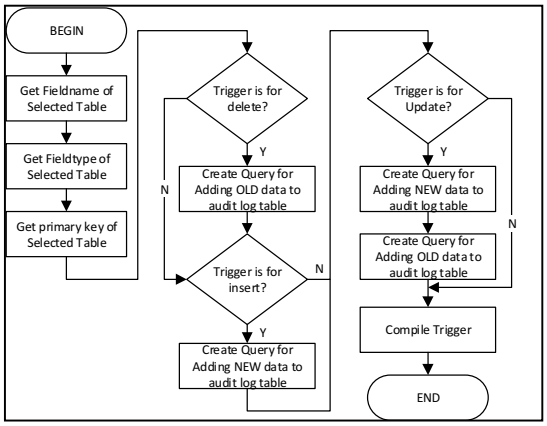

Figure 1: How Audit Log Trigger Created

Figure 1 explains how trigger loads to create audit table. It takes some data, such as field, data type and primary key, those will be a variable in trigger. Each type of trigger has different contents. Delete trigger merely has old data, since it is only old data needed in data deletion process. Insert trigger also merely uses new data, because it is only capable to obtain new data when there is new data input. Update trigger is able to record old and new data. Old data has function to search data that is going to be updated, while NEW is new data to replace old data. Data input passing through those every trigger will be marked with I code (for insert), U (for update), and D (for delete).

2.2 Synchronization Process

Before synchronization process, synchronized database in early initial state at each site must be similar. This is done to ensure synchronized data is still consistent [5]. If this condition has been satisfied, then synchronization process could be started.

To start synchronization by using data of audit log, id and last timestamp from audit table is necessary to record. Using timestamp is a very important thing and its utilization is very qualified in comparing [6], therefore timestamp is used to check whether there is new data, changed data, or deleted data from table at synchronized database according to id record and timestamp, also to make sure if there is overhaul towards the entire data in audit table which certainly will take long time.

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

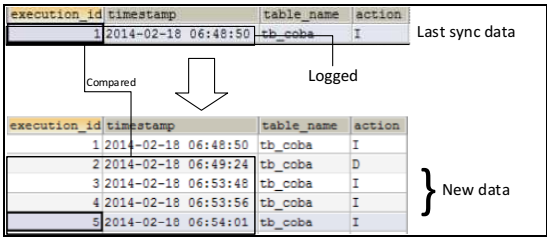

Figure 2: How audit table data logged and compared

Figure 2 points out how the reading process of data in audit table. The id and timestamp of last data that has been read will be saved and then used to investigate whether the newer data of that id and timestamp has been into audit table. If there is new data, data delivering process can be directly started and sent to all synchronized sites.

In the case of one way synchronization, the implemention of that model still produces merely replication mode, it is because when there is data deletion at slave site, thus that data will never be fetched and produce not synchronized database. It differs from two ways synchronization, where alteration at master or slave site will influence each other. To solve problems in that one way synchronization, it needs a third party application to control alteration at slave site in real-time, therefore rollback data can be conducted.

3. IMPLEMENTATION

Previous discussion of synchronization design will be implemented by using client-server architecture in exchanging data process and DBMS. This research used DBMS MySQL.

Regarding to Figure 3, synchronization process works by using client-server architecture and data deliverance is done by using *socket stream*. Client is a software that installed to every sites where the database that will be synchronized are used. Connection config on client is file containing connection configuration to DBMS where it has database existing which will be synchronized and connection configuration such as ip address, port and authentication user needed to do communication by using socket [7]. Client application uses one database, it consists of some configuration tables, they are table to save data received from the other clients, table to save data which will be sent to the other clients, also master and slave synchronization configuration table.

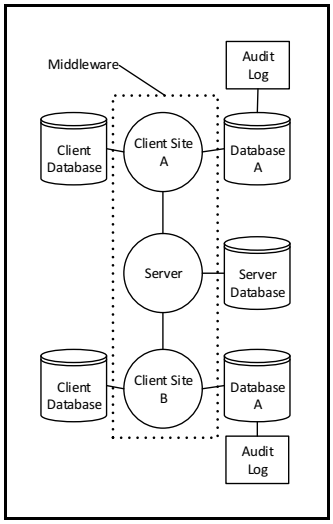

Figure 3: Scheme of Synchronization Architecture Implementation.

Sending and executing data uses traditional sort algorithm, it means first data input will be processed firstly (FIFO) [8]. Generally, client and server application has main function as data sender and receiver. In term of sending data, the application will determine what and where to send. In term of receiving data, the application will work more at waiting data to arrange what process should be conducted towards that data [9].

Moreover, each client can synchronize more than one database at one local site. There are two types of configuration can be conducted, those are configuration as master and as slave. When it is configuration as master, then the connection setting of database and synchronized table must be determined. For configuration as slave, the client has to register the other clients which control database as master at configuration database, and then do requesting data from master client in the form of database and table list which is necessary to be selected and synchronized. When slave assures to accomplish selecting database and table, client master will register slave into configuration database, this further will be used to know where to sent audit table. Figure 4 shows any components owned by client

Journal of Theoretical and Applied Information Technology
31st July 2014. Vol. 65 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

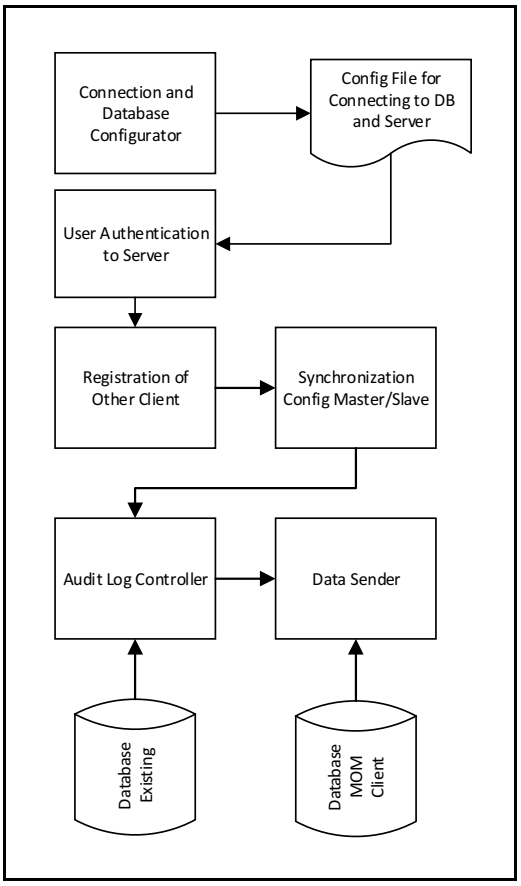

Figure 4: Components of Synchronization Client

Server synchronization has more moderate function than client. Server has duties to receive data from one client and send to directed client. Moreover, client also has function to supply service utilizing authority, therefore only client who inputs authentication correctly will be permitted to send data through server. Although it seems simple, server's duties are hardly difficult since it enables many clients connected at once and many inputs-outputs through the server to and fro. Therefore, server application is supposed to be installed in the computer with excellent hardware and network specification. Figure 5 illustrates any components owned by server synchronization.

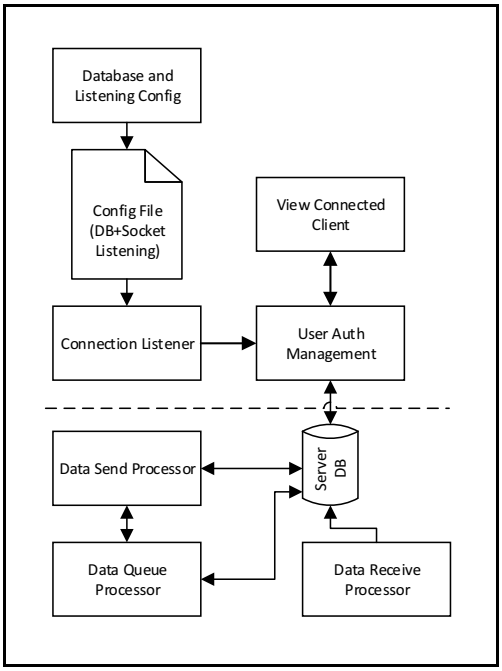

Figure 5: Components of Synchronization Server.

4. SYNCHRONIZATION ISSUES

When synchronization process is running, there are many possibilities occur toward the processed data. The emergence of the problems has to contribute the appropriate solution, so that, it can guarantee integration of data that has been synchronized. Some potential problems in implementing synchronization with client-server will be discussed below.

4.1 Client Fails Connecting to Server

Utilizing client-server model implementation obviously has a certain condition when client are not able connecting to server, particularly because of problem in network connectivity. That causes problems when a data must be sent and that data is not successfully sent. This problem can be solved by implementing deliverance status analysis process of data on the client application level. The message that is failed to send will be saved into a table and sent it back when client is fully connected to server.

4.2 Client at Destination Site is Offline

When data is already in server and will be forwarded to targeted client, there is possible problem, such as targeted client is offline, therefore server will fail to forward the data. This issue can be solved by saving data temporarily to server into

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

a message queue table. Sever will periodically attempt sending data to targeted client according data in queue table and it will give flag if data has been successfully sent.

4.3 There are Many Stacks of Log in One Audit Table

Log in audit table will be incessantly accumulating when there is constantly an increase, changing and deletion toward a table in a database. This circumstance will really influence the performance of synchronization process done by the application, since it searches numerous data by using query. To solve this problem, data in audit table will be deleted when the synchronization is successfully done, and then truncate at table will be executed when all data is considered synchronized to all sites successfully. It does not influence the new site to do synchronization configuration, because initial state at database will help to overcome possible problems in the stage of reading audit table.

4.4 The Possibility of Endless Loop when There is Two Ways Synchronizations

In applying two ways synchronization model, both database at master and slave site will have an active audit table. It will be used in synchronization process. The issue appears when data from audit table at master site is executed to slave site, and that data is recorded again in audit table at slave site. This could cause delivering data repeatedly and become an enormous problem if it is not rightly solved. To solve that problem, it is specifically regarded to audit table. It is necessary to make a jumper table that contains which data at audit table must be jumped when data is going to be sent to targeted client has similar identity with jumper table.

4.5 Data Reconciliation

The biggest problem in implementing database synchronization is when data that is going to be synchronized experiences a conflict in execution process. That conflict emerges when data with similar primary id, however it has another column with different data. In the case of adding data, this conflict can be possibly solved. Meanwhile, in the case of updating and deleting data, it may make this problem more complex. The solution in this implementation is making a data reconciliation model, so when there is a conflict on data, the user of client application will manually decide which data is correct. Data reconciliation needs confirmation of data selection from whole clients located in the two ways surroundings.

4.6 Data Security at Delivery Process

Data integrity can be assured through its accuracy, completeness, wholeness, health and conformity of data compared to source data [10]. Exchanged data through a communication network always has risk in its integrity problem. Changing and hijacking data by outsiders could happen when data passes through communication network. Utilizing AES encryption on data is one solution to overcome it. AES offers the combination of security, performance, efficiency, implementability and flexibility [11]. The security level could be gained based on how encryption implementation is conducted [12].

5. NEXT WORKS

Unfortunately, this does not mean that database Synchronization implementation has been fully accomplished. There are still many features need to apply in distributed database, thus it needs more effort to develop preceding implementation on previous model. Optimization in security, network and efficiency to process numerous data will certainly be another important issue to appear. The implementation of multi-server is also needed to be able to do load-balancing data which may pile up from the client.

Another aspect must be settled is implementing synchronization model in the form of multi-master. Therefore, one site does not merely use 1 master site, yet it can process similar database synchronization from the other master site. Synchronization model at fragmentation is also an extra work to add to this synchronization implementation. It is considering that fragmentation on column or row is very useful when that data is only used at some sites. Besides, it certainly will help increasing the reading performance toward data from database.

6. CONCLUSION

Synchronization implementation by utilizing this audit log method can be used to process replication and synchronization towards DML. Meanwhile, it has weakness in recording DDL activity although some DBMS like oracle could perform log to DDL.

Utilizing client-server model will absolutely encounter many problems when it is applied, for example problems in network, endless loop process, data conflict and data security. Problems

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

can be solved through correct solution just as previously elaborated. Even though, there are still many probabilities to solve it by applying better solution.

REFERENCES:

[1] Marius Christian. Database Replication. *Database Systems Journal*, Vol. I, No. 2, 2010, pp. 34.

[2] Kao, B. and Garcia-Molina, H. An Overview of Real-Time Database Systems. Technical Report. Stanford InfoLab. *Proceedings of NATO Advanced Study Institute on Real-Time Computing*. St. Maarten, Netherlands Antilles, October 9, 1992.

[3] Meg Coffin Muray. Database Security: What Students Need to Know. *Journal of Information Technology Education: Innovations in Practice*, Vol. 9, 2010, pp. 73.

[4] Yang, L. Teaching database security and auditing. *Proceedings of the 40th ACM Technical Sympo-sium on Computer Science Education*, Chattanooga, TN, USA, 2009.

[5] M. Tamer Özsu, Patrick Valduriez. Principles of Distributed Database System, 3rd edition, 2011, pp. 461-462.

[6] Majeed, Mahmud, Iqbal. Efficient Data Streams Processing in the Real Time Data Warehouse. *Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference*, Vol. 5, 2010, pp. 60.

[7] Abhijit A. Sawant, Dr. B. B. Meshram. Network Programming in Java Using Socket. *International Journal of Engineering Research and Applications* Vol 3, issue 1, 2013, pp. 1299-1304.

[8] JinGang Shi, YuBin Bao, FangLing Leng, Ge Yu. Priority-based Balance Scheduling in Real-Time Data Warehouse. *Hybrid Intelligent Systems, Ninth International Conference*, Vol. 3, 2009, pp. 301.

[9] Sameena Naaz, Afshar Alam, Ranjit Biswas. Load Balancing Algorithms for Peer to Peer and Client Server Distributed Environments. *International Journal of Computer Applications*, Vol. 47, No. 8, June 2012, pp. 18.

[10] Ed Gelbstein, Ph.D. Data Integrity: Information Security's Poor Relation. *ISACA Journal*, Vol. 6, 2011.

[11] A. Rudra, P.K. Dubey, C.S. Jutla, V. Kumar, J.R. Rao, P. Rohatgi, "Efficient Rijndael encryption implementation with composite field arithmetic,". *Lecture Notes in Computer Science 2162*, 2001, pp. 171-184.

[12] Samir El Adib, Naoufal Raissouni. AES Encryption Algorithm Hardware Implementation: Throughput and Area Comparison of 128, 192 and 256-bits Key. *International Journal of Reconfigurable and Embedded Systems (IJRES)*, Vol. 1, No. 2, July 2012, pp. 67-74.