

DATABASE REPLICATION IN HETEROGENOUS PLATFORM

Hendro Nindito, Evaristus Didik Madyatmadja and Albert Verasius Dian Sano

School of Information Systems, Bina Nusantara University, Jakarta, Indonesia

Received 2014-01-24; Revised 2014-03-05; Accepted 2014-07-03

ABSTRACT

The application of diverse database technologies in enterprises today is increasingly a common practice. To provide high availability and survivability of real-time information, a database replication technology that has capability to replicate databases under heterogenous platforms is required. The purpose of this research is to find the technology with such capability. In this research, the data source is stored in MSSQL database server running on Windows. The data will be replicated to MySQL running on Linux as the destination. The method applied in this research is prototyping in which the processes of development and testing can be done interactively and repeatedly. The key result of this research is that the replication technology applied, which is called Oracle GoldenGate, can successfully manage to do its task in replicating data in real-time and heterogeneous platforms.

Keywords: Replication, Heterogen, Real Time, SQL Server, MySQL

1. INTRODUCTION

Organizations need to provide updated data to users that might be geographically remote while handling vast amount of requested data distributed in multiple sites. The availability of a certain data at a huge network is one of the issues that still unsolved (Ahmad *et al.*, 2010).

People often use the terms *data* and *information* interchangeably. However, it is better to view data as raw material resources that will be processed into finished information products. Then we can define information (O'Brien and Marakas, 2011) as data that have been converted into a meaningful and useful context for specific end users.

Today in most industries, a majority of overwhelming business transactions is executed digitally as businesses expand the use of their online applications. To succeed in this real time (RTI, 2012) environment, companies must eliminate interruptions in data availability, sluggish application performance and stale data. There are four aspects to this challenge:

- Availability. Business-critical applications and underlying data must be accessible at or near

24/7/365 without service interruption or performance degradation

- Reduced latency. Data must remain fresh. As it ages, data becomes less relevant and less valuable
- Heterogeneity and IT flexibility. Integration and replication solutions must have the flexibility to be easily modified and distributed across diverse IT systems
- Transaction integrity. Data completeness and accuracy must be ensured as it moves through systems

There are essentially four kinds of replication styles -active replication, semi-active (leader-follower) replication, passive replication and coordinator-cohort replication. In active replication (state-machine approach), each server replica processes every client invocation and returns the response to the client. With active replication the availability of system is more when comparing to any other replication technique (Saravanan and Ramaraj, 2009).

This study gives an overview of an active replication technology in which users can replicate from any RDBMS to another RDBMS in heterogeneous environments using third party software, that is, Oracle Golden Gate.

Corresponding Author: Hendro Nindito, School of Information Systems, Bina Nusantara University, Jakarta, Indonesia

2. STUDY LITERARURE

2.1. Previous Researcher on Heterogeneous Database Replication

Large amounts of work have already been done on distributed database systems (Silberschatz *et al.*, 2011; Murthy *et al.*, 2012). Research on cross-platform environment is similar to that of heterogeneous database systems, in which different sites (nodes that are equipped with a computational and storage resource) may use different ways of storing and representing data, file formats, access protocols, query languages. The sites in this cross-platform environment communicate through an insecure channel. This study provides a solution in which relational database can be replicated from server to client based on replication or fragmentation (vertical) without any loss of information. This replication is independent of any platform and secured by strong cryptographic algorithm. Here, each site may have different Relational Database Management Systems (RDBMSs), such as Oracle, MySql, DB2, MS Access.

2.2. Distributed Database

Data distribution is a very important issue in distributed database, the database fragments need to be assigned to nodes in the computer network (Khder *et al.*, 2013).

Distributed database can be clasified on homogenous and heterogenous database. In literature (Silberschatz *et al.*, 2011) homogeneous distributed database system, all sites have identical database management system software, are aware of one another and agree to cooperate in processing users' requests. In contrast, in a heterogeneous distributed

database, different sites may use different schemas and different database-management system software.

Consider a relation r that is to be stored in the database. There are two approaches to storing this relation in the distributed database:

- Replication. The system maintains several identical replicas (copies) of the relation and stores each replica at a different site
- Fragmentation. The system partitions the relation into several fragments and stores each fragment at a different site

In literature (Charron-Bost *et al.*, 2010) the generic architecture for practical database replication is composed of the following building blocks (**Fig. 1**):

- The *Application*, which might be the end-user or a tier in a multi-tiered application
- The *Driver* provides a standard interface for the application
- The *Load Balancer* dispatches client requests to database replicas using a suitable load-balancer algorithm
- The *DBMS*, or *Database Management System*, which holds the database content and handles remote requests to query and modify data expressed in standard SQL
- The *Reflector* is attached to each DBMS and allows inspection and modification of on-going transaction processing
- The *Replicator* mediates the coordination among multiple reflectors in order to enforce the desired consistency criteria on the replicated database. The *Group Communication* supports the communication and coordination of local replicators

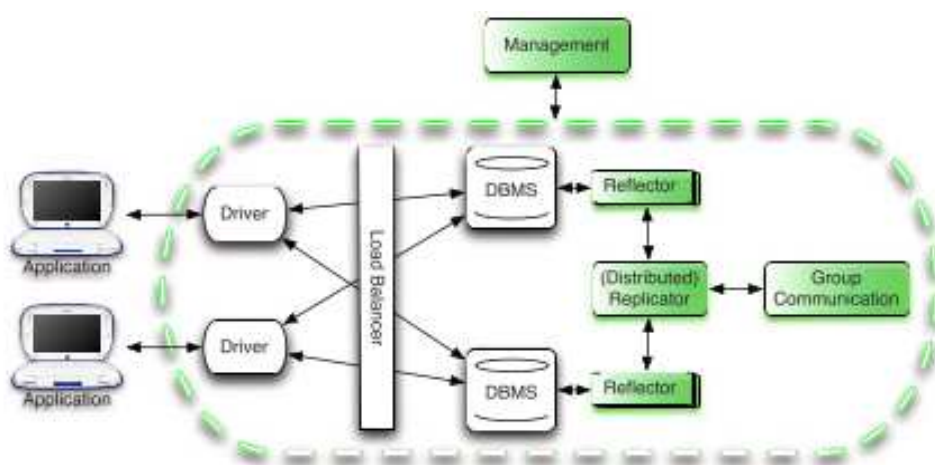


Fig. 1. Generic architecture for replication (Charron-Bost *et al.*, 2010)

2.3. Benefits of Database Replication

Some of the main benefits associated with database replication (Connolly and Begg, 2010) are:

- Availability, refers to how replication increases the availability of data for users and applications through the provision of alternative data access options. If one site becomes unavailable, then users can continue to query or even update the remaining locations
- Reliability, refers to the fact that with multiple copies of the data available over the system, this provides excellent warm standby recovery facilities in the event of failure at one or possibly more sites
- Performance, refers to the improvement for query transactions when a system suffers from a significant overloading of centralized resources. Replication provides fast, local access to shared data because it balances activity over multiple sites
- Load reduction, refers to how replication can be used to distribute data over multiple remote locations. This configuration can significantly reduce network traffic. Also, users can access data from the replicated site that has the lowest access cost, which is typically the site that is geographically closest to them
- Disconnected computing, refers to how replication can be supported by snapshots
- Supports many users, refers to how organizations increasingly need to deploy many applications that require the ability to use and manipulate data
- Supports advanced applications, refers to how organizations increasingly need to make the corporate data available not only for traditional Online Transaction Processing (OLTP) systems but also for advanced data analysis applications such as data warehousing, Online Analytical Processing (OLAP) and data mining

2.4. System Architecture

In this study, we apply the heterogenous database replication using Oracle Golden Gate solution. Golden Gate (Jeffries, 2012) can achieve heterogeneous and homogeneous-real-time transactional Change Data Capture and integration by decoupling itself from the database architecture.

Therefore it provides a performance boost as well as flexibility through its modular components. A number of system architecture solutions are offered for data replication and synchronization:

- One-to-one (source to target)
It's the simplest and the most common configurations, here we are performing real-time or batch change data replication between two sites. This could be one example between a primary and standby site for disaster recovery or an OLTP to datawarehouse for business intelligence
- One-to-many (one source to many targets)
This architectures provide two solutions, one data replication feed for reporting and one for backup and disaster recovery
- Many to one
This configurations is common in all industries, for example from retailing outlet taking customer orders to centralized database in head office
- Cascading
this offer s data replication at n sites, originating from a single source. As the data flows from the originating source database, parts or all of it are replicate at each site in cascading fashions until the final target populated
- Bi-directional (active)
This is an active-active configurations. In this model sites A sends changed data to site B vice versa where conflict handling is an important considerations
- Bi-directional (active passive)
This is also known as live standby where site A sends changed data to site B only

3. THE PROPOSED MODEL

There are two primary objectives and needs addressed by PT.XYZ:

- Continuous availability and disaster tolerance. This is to meet end-users' expectations for 24/7 system availability and to surpass and enhance the service level
- Real-time data integration for live reporting. This is to serve the purposes of real-time, operational reporting, freeing up the resources of the critical source systems

The source data is on MS SQL Server database running on MS Windows platform. The target is an MySQL database running on Linux platform.

In order to meet the continuous availability and disaster tolerance requirements, we propose Oracle GoldenGate solution as depicted in the **Fig. 2** below.

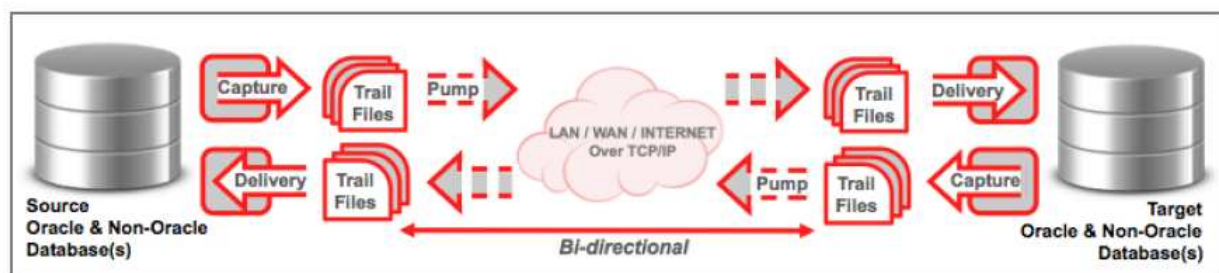


Fig. 2. Oracle GoldenGate architecture (Oracle, 2012)

Oracle GoldenGate consists of decoupled modules that are combined to create the best possible solution for business requirements as follows (**Fig. 2**):

3.1. On the Source System (s)

GoldenGate’s Capture (Extract) process reads data transactions as they occur, by reading the native transaction log, typically the redo log. Oracle GoldenGate only moves changed and committed transactional data, which is only around 30% of all transactions-therefore it operates with extremely high performance and very low impact on the data infrastructure.

Filtering can be performed at the source or target-at tables, columns and/or rows.

Transformations can be applied at the capture or delivery stages.

3.2. Trail Files

To move transactional data efficiently and accurately across systems, Oracle GoldenGate converts the captured data into an Oracle GoldenGate data format in “trail” files. With both source and target trail files, Oracle GoldenGate’s unique architecture eliminates any single point of failure and ensures data integrity is maintained-even in the event of a system error or outage.

3.3. Routing

Data is sent via TCP/IP to the target systems. Data compression and encryption are supported. Thousands of transactions can be moved per second and without distance limitations.

3.4. On the Target System(s)

A Server Collector process (not shown) reassembles the transactional data into a target trail.

The Delivery (Replicat) process applies transactional data to the designated target systems using native SQL calls.

3.5. Bi-Directional (if Needed)

In bi-directional configurations/solutions, this process runs reciprocally, to concurrently synchronize data between the source and target systems.

Manager processes perform administrative functions at each node.

To meet real-time data integration for live reporting, a secondary site can be deployed to serve the purposes of real-time, operational reporting, instead of running on primary site. In this case, the secondary site would be MySQL running on Linux platform.

4. RESULTS AND DISCUSSION

To perform this experiment we built two Virtual Machines on X series server, where one VM server represents the source server and the other one represents the target server as shown in the **Table 1** below.

After the environment ready, we do some test scenario in several conditions to check wheter the data could replicate without any lost. The test scenarios is describe in **Table 2** below. As result of the scenario we can see at **Table 3** below. As we see the test scenari and result from **Table 2** and **Table 3**, the GoldenGate can replicate data event in condition when the network is off. This can be seen as we turn off one of the replicate component. The data will be pool at trial file at server 1 and continue deliver to server 2 as the network is on without data lost.

Table 1. Test environment

Component	Server 1	Server 2
Host	Server 1	Server 2
Ip	192.168.137.10	192.168.137.20
Port	7809	7809
Platform	Windows server 2003 32 bit	Linux ubuntu 32 bit
Database	SQL Server 2008 R2 enterprise	MySQL Server 5.1.x
GoldenGate	11.1.1.1	11.1.1.1
User	Gguser	Gguser
Envorontment	Intel Core i5-2410M 2.3Ghz CPU, 512 MB RAM, 20GB HD	Intel Core i5-2410M 2.3Ghz CPU, 512 MB RAM, 8GB HD

Table 2. Test scenarios

Test description	Scenarios
Replication components on one server in a state STOP	Stop the replication components on server 2 in a time-span of 10 min
Data transaction remains running when one server replication components in a state STOP	Update all value on fields in Server 1
Run the replication components on server 2	Activate replication components on server 2 and query on the modified tables
Server maintenance	Stop the replication components on server 1 and then turn off server
Data transaction remains running during one replication components server is in a maintenance state	Update, add, delete data on server 1
One server is active	Activate server 1, but replication components are not running
Run the replication components	Run on replication components in server1, perform a query on the server 1

Table 3. Test results

Expected results	Test results
Replication components on server 1 must be in a running state	Succeed
There were changes in the data on the server 1 and no change on the server 2	Succeed
After the components on server 2 are active, the data transactions that occur on the server 1 are replicated on the server 2	Succeed
On the server 2 all components must be in a running state	Extract data pump abended
There were changes in the data on the server 2 and no change in server 1	Succeed
Replication components that are in an ABENDED state can be up automatically	Succeed with 1minute delay while server 1 in an active state
After the components on server 1 are active, the data transactions that occur on the server 2 are replicated on server 1	Succeed

5. CONCLUSION

Replication of tables across different database management systems running on different platforms can be efficiently done. Data is effectively transferred and stored without any loss of information. With the third party tool, such as Oracle GoldenGate, the complexity of the processes is hidden from user.

The Oracle GoldenGate technology can be applied to heterogeneous operating system environments and DBMS platforms. It has been successfully proven by performing a

case study on uni-directional replication topology applying MSSQL Server DBMS running on Windows platform and MySQL DBMS running on Linux platform.

6. REFERENCES

Ahmad, N., A.N. Abdalla and R.M. Sidek, 2010. Data replication using read-one-write-all monitoring synchronization transaction system in distributed environment. *J. Comput. Sci.*, 6: 1095-1098. DOI: 10.3844/jcssp.2010.1095.1098

- Charron-Bost, B., F. Pedone and A. Schiper, 2010. Replication: Theory and Practice. 1st Edn., Springer, Berlin, ISBN-10: 3642112935, pp: 290.
- Connolly, T. and C. Begg, 2010. Database System: A Practical Approach to Design, Implementation and Management. 5th Edn., Addison Wesley, ISBN-10: 0321523067, pp: 1400.
- Jeffries, J.P., 2012. Oracle GoldenGate 11g Implementer's Guide. 1st Edn., Packt Publishing Ltd., Birmingham, ISBN-10: 1849682011, pp: 280.
- Khder, M.A., Z.A. Othman, A.M. Zin, and S. Abdullah, 2013. Improving data availability in mobile environment using data allocation. *J. Comput. Sci.*, 9: 1019-1029. DOI:10.3844/jcssp.2013.1019.1029
- Murthy, P.K., S. Agarwal, T. Garbyal and P. Chakrobarty, 2012. Secure migration of tables across different database management systems over a cross-platform environment. *Int. J. Inform. Electron. Eng.*, 2: 387-390. DOI: 10.7763/IJIEE.2012.V2.121
- O'Brien, J.A. and G.M. Marakas, 2011. Management Information Systems. 10th Edn., McGraw-Hill.
- Oracle, 2012. Oracle: Using Oracle GoldenGate 11g For Oracle Database.
- RTI, 2012. Oracle: Oracle GoldenGate 11g: Real-Time Access to Real-Time Information.
- Silberschatz, A., H.F. Korth and S. Sudarshan, 2011. Database System Concepts. 6th Edn., McGraw-Hill, New York, ISBN-10: 0071289593, pp: 1349.
- Saravanan, R. and N. Ramaraj, 2009. Providing reliability in replicated middleware applications. *J. Comput. Sci.*, 5: 11-22. DOI: 10.3844/jcssp.2009.11.22