

MULTI MASTER REPLICATION IN ORACLE DATABASE

BOICEA, A[lexandru]; NICULA, A[drian] - I[onut]; SERBAN, A[lexandru] & GRIGORE, E[lena] M[ihaela]

Abstract: This paper presents the multi-master replication which is a method of database replication that allows data to be stored by a group of computers, and updated by any member of the group. The multi-master replication system is responsible for propagating the data modifications made by each member to the rest of the group, and resolving any conflicts that might arise between concurrent changes made by different members.

Key words: database, replication, multi-master

1. INTRODUCTION

The most important reason for which Oracle multi-master replication is used for geographically distributed systems is that it provides **multiple-node replication capabilities**. Replication can improve performance and increase availability of applications because alternate data access options becomes available. For example, users can access a local database rather than a remote server to minimize network traffic. Furthermore, the application can continue to function if parts of the distributed database are down as replicas of the data are still accessible.

Another benefit of multi-master replication is the ability to replicate stored procedures. In a system where all code is encapsulated inside Oracle stored procedures, you can replicate the stored procedures to remote sites, just like data. This capability allows the DBA to coordinate code changes with database changes. Once the Oracle stored procedures are written, you can easily replicate and distribute them to work groups and branch offices throughout the entire replicated network of systems.

Figure 1 shows how replication between two of the multi-master replication environment works:

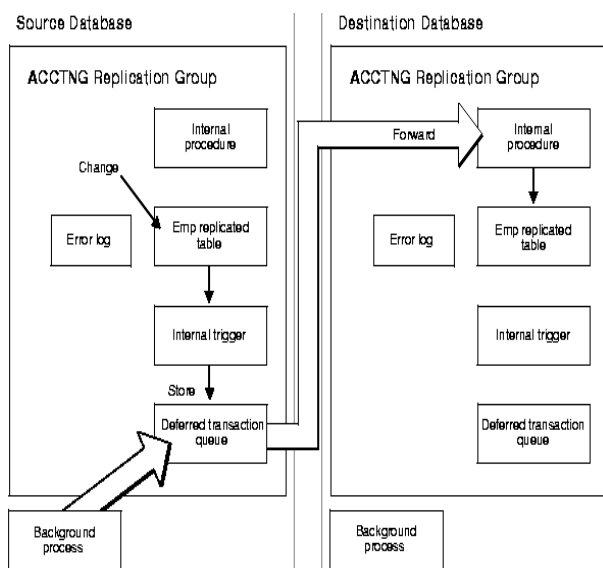


Fig. 1. How replication between two of the multi-master replication environment works

By using multi-master replication you can replicate: tables, indexes, procedures, functions, triggers and packages.

The advantages of multi-master replication are:

- replicates more objects, including user-defined objects
- if any master fails, other masters will continue to update the database

- updates and modifies the objects being replicated; adding a column to a table at the master definition site can be replicated to other master sites

- replicates with any number of other databases; any master site can replicate with other master sites

The disadvantages of multi-master replication are:

- potentially large network bandwidth requirements. Not only does multi-master push and pull changes between sites, it also sends acknowledgements and quite a bit of administrative data.

- reduced performance. Complexity and robustness comes at a price. MMR involves the use of triggers and procedures, and this can result in a database performance hit. Depending on how much data you are replicating, this performance hit can be substantial.

- significant increases in administration requirements. When problems appear in the database, the DBA must insure that replication is not the cause or that the cause is not replicated to other databases. Database performance tuning and problem resolution becomes more complicated by an order of magnitude.

- database changes require additional planning. Rolling out a new version of an application can be much more difficult. Each new version will require revisiting the design of the replication (Lewis, 2006).

2. MULTI-MASTER CONFLICTS AND POSSIBLE SOLUTIONS FOR THE CONFLICTS

The most common types of conflicts that appear in multi-master replication are:

Uniqueness conflict - results from an attempt from two different sites to insert records with the same primary key. To avoid uniqueness conflicts, you can choose from three available options. Those three pre-built methods are called Append Site Name To Duplicate Value, Append Sequence To Duplicate Value, and Discard Duplicate Value.

Update conflict - is caused by simultaneous update operations on the same record.

Delete conflict - occurs when one transaction deletes a row that another transaction updates (before the delete is propagated).

The possible solutions for the conflicts are shown in the Tab.1.

METHOD	DESCRIPTION
Latest Timestamp Value	With the latest timestamp value method, you define a column that contains a date data type to use in comparing multiple transactions. When a transaction fails because the before image has changed, the column timestamps of the transactions are

	compared, and if the second transaction is later than the one changing the before image, the second transaction is applied and overlays the current data (which contains the first transaction).
Earliest Timestamp Value	This is the opposite of the above method. The method is available but rarely used.
Minimum Value, Maximum Value	When a column group has a conflict between two transactions, the minimum value method evaluates the current and new values of a defined column and selects the lower (or higher for maximum) values to determine if the new columns are applied.
Group Priority Value	In this case, column groups are assigned a priority and conflicts are adjudicated to the highest priority group.
Site Priority Value	In this instance, sites are assigned a priority. When two transactions conflict, the transaction from the master site with the highest priority will be applied. This is actually a special case of the Group Priority Value method above.

Tab.1. Multi-master conflicts and possible solutions for the conflicts

3. PREPARING FOR MULTI-MASTER REPLICATION

Before starting to build a multi-master advanced replication environment, each participating database must be prepared with the following:

- A replication administrator.
- A replication propagator.
- A replication receiver.
- Database links to provide for interdatabase communication.

4. BUILDING A MULTI-MASTER REPLICATION ENVIRONMENT

By using Oracle, the minimum steps to complete, in order to build a multi-master replication environment are:

Step1: Design the advanced replication environment. Decide what tables and supporting objects to replicate to multiple databases, and organize replication objects in suitable master groups.

Step2: Use Replication Manager's setup wizard to configure a number of databases to support a multi-master replication environment. The Replication Manager setup wizard quickly configures all components necessary to support a multi-master replication system.

Step3: Using the Replication Manager database connection to the master definition site, create one or more master groups to replicate tables and related objects to multiple master sites.

Step4: Grant privileges necessary for application users to access data at each site.

Step5: Configure conflict resolution for replicated tables before resuming replication activity for a master group(Dye, 1999).

5. ORACLE STREAMS FOR MULTI-MASTER REPLICATION

Oracle Streams is an ideal solution for systems that are geographically distributed and have a high-speed connection

(e.g. a T1 line) between servers. As long as the server interconnect can keep-up with the changes you can implement systems that provide failover and disaster recovery, simply and reliably. However Oracle Streams replication is not for every database:

- High update systems – Busy update database may spend excessive resources synchronizing the distributed databases.
- Real-time replication required – If you require a two-phase commit (where the changes are committed on each database at exactly the same time), then RAC or on-commit replication are a better choice. For example, a banking system would require that all databases update each other simultaneously, and Oracle Streams might not be an ideal solution. On the other hand, Oracle Streams is perfect for geographically distributed applications where real-time synchronization is not required.

Oracle Streams will provide near real-time replication of important information and in case of a server outage, updates are automatically stored in update queues and applied automatically when service is restored to the crashed Oracle server.

Streams replication failure alerts are a critical component of the replication so that Oracle staff can be aware when there has been a server or network failure. Once the Oracle Streams, Data Guard and alerts are in-place, the only challenge for the Oracle DBA is ensuring that the Streams replication queues have enough room to hold updates on the surviving Oracle instance(Burleson, 2006).

6. CONCLUSIONS

Multi-master replication is a very powerful and flexible feature. However, with power and flexibility comes complexity. In order to create a multi-master replication environment it is necessary to analyze the type of informatic system where replication is made and to plan the implementation in detail.

The universal dispute that exists in a multi-master replication is transactional inconsistency avoidance or resolution. Most synchronous or keen replication systems do inconsistency avoidance, while asynchronous systems have to do inconsistency resolution. The resolution of such an inconsistency may be based on a timestamp of the transaction, on the ladder of the source servers or on much more intricate reason, which decides every time on all servers. By using replication, multiple disks, disk controllers, power supplies, and CPUs are used to ensure that your data modification makes it to stable storage. In other words, replication allows you to minimize the problem of a single point of failure by using more hardware to guarantee your data writes(Garmany et. al., 2003).

7. REFERENCES

- Burleson, D.K.(2006) *Oracle Replication: Expert Methods for Robust Data Sharing*, ISBN 0-9727513-3-5, Ed. Library of Congress, USA
- Dye, C. (1999) *Oracle Distributed Systems*, ISBN 978-1-56592-432-1, Ed. O'Reilly Media, USA
- Garmany, J., Freeman, R. (2003), *Oracle Replication: Snapshot, Multi-Master & Materialized Views Scripts*, ISBN 0-9727513-3-5, Ed. Rampant TechPress, USA
- Lewis, J.(2006), *Cost Based Oracle: Fundamentals* ISBN 1-59059-636-6, Ed. Apress, USA
- ***(2007)http://www.oracle.com/technology/books/pdfs/book_rep_chap6_ce2.pdf, *Accesed on:2010-05-25*